

**Környezetszennyezéssel kapcsolatos gyakorlati feladatok
megoldásának statisztikai támogatása SAS adatelemző szoft-
ver használatával**

Oktatási segédanyag

v1.0

Készítette:

Csicsman József

Gálfi Márta

László Anna

Sipos Szabó Eszter

Soltész Gábor

Szeged

2011. augusztus 9.

Tartalomjegyzék

1. ÁLTALÁNOS ISMERTETŐ	5
1.1 A DOKUMENTUM CÉLJA	5
1.2 VERZIÓKÖVETÉS.....	5
2. TERMÉSZETTUDOMÁNYOS ISMERETEK RENDSZEREZÉSE A KÖRNYEZET IRÁNYÍTÁSI RENDSZEREK ÉS A VÁLLALATI TEVÉKENYSÉGEK KOCKÁZATELEMZÉSI SZÜKSÉGESSÉGÉNEK MEGÉRTÉSÉHEZ.	6
2.1 AZ ÖNSZERVEZŐDÉS SZABÁLYOZÓ LÉPÉSEI	9
2.1.1 <i>Karbon (C) ciklus</i>	10
2.1.2 <i>Nitrogén (N₂) ciklus</i>	11
2.1.3 <i>Oxigén (O₂) ciklus</i>	13
2.1.4 <i>Víz (H₂O) és hidrogén (H₂) ciklus</i>	13
2.1.5 <i>Foszfor (P) ciklus</i>	14
2.1.6 <i>Kén (S) ciklus</i>	16
2.1.7 <i>A többi elem ciklusa</i>	17
2.2 A TÁRSADALMI SZINTEN ÉRTELMEZETT ALKALMAZKODÁS.....	19
2.3 PÉLDÁK	21
3. A SAS BASE KÖRNYEZET MEGISMERÉSE.....	23
3.1 RENDSZER.....	23
3.2 FELÜLET	24
3.2.1 <i>LOG</i>	25
3.2.2 <i>Editor</i>	25
3.2.3 <i>Output</i>	26
3.2.4 <i>Explorer</i>	27
3.2.5 <i>Results</i>	30
3.3 ADATIMPORT	31
4. KÉKSZALAG SAS ADATÁLLOMÁNY MEGISMERÉSE, ADATMANIPULÁCIÓ.....	37
5. VÁLTOZÓK KÉPZÉSE, ADATOK RENDEZÉSE, CSOPORTOK SZERINTI KÜLÖN ADATÁLLOMÁNYOK LÉTREHOZÁSA ÉS GRAFIKUS MEGJELENÍTÉSE.....	58
6. VALÓS FELADAT – LEVEGŐSZENNYEZÉS ESETTANULMÁNY.....	65
7. TISZTÍTOTT ÉS NYERS SZENNYVÍZ PH ÉRTÉKEINEK ÖSSZEHASZNÁLÁSA	75
8. SZENNYVÍZTISZTÍTÁS - ADATGENERÁLÁS	82
9. ADATGENERÁLÁS MÁS MÓDON.....	90
10. ADATGENERÁLÁS, GRAFIKONKÉSZÍTÉS, TÁBLÁZATOS MEGJELENÍTÉS	99
11. VALÓSZÍNŰSÉGSZÁMÍTÁS ÉS ALAPSTATISZTIKÁK KOCKÁZATBECSLÉSHEZ.....	106

12.	TÖBB VÁLTOZÓ ADATAINAK MEGJELENÍTÉSE EGY KÖZÖS DIAGRAMON.....	113
12.1	1. MEGOLDÁS.....	113
12.2	2. MEGOLDÁS.....	120
12.3	3. MEGOLDÁS.....	121
13.	MELLÉKLET.....	123
13.1	KÉKSZALAG EXCEL ADATÁLLOMÁNY	123
13.2	NOX MÉRT ÉRTÉKEK	123
13.3	SZENNYVÍZTISZTÍTÁSELŐTT ÉS UTÁN MÉRT JELLEMZŐK ADATAI.....	123
13.4	HIÁNYOS ADATOK SZENNYVÍZTISZTÍTÓ ÜZEMBŐL.....	123
13.5	TALAJVÍZ KUTAK PÁRHUZAMOSAN MÉRT NITRÁT KONCENTRÁCIÓI.....	124
14.	FÜGGELEK.....	125
14.1	MATEMATIKAI ALAPOK AZ ADATBÁNYÁSZATI SZOFTVEREK ELSŐ MEGISMERÉSÉHEZ	125
14.2	SAS / GRAPH GYAKORLATI BEMUTATÓ	125
15.	ÁBRAJEGYZÉK.....	126

1. ÁLTALÁNOS ISMERTETŐ

1.1 A DOKUMENTUM CÉLJA

Jelen dokumentum célja a nem informatikus hallgatók statisztikai adatelemzésének támogatása a SAS (Statistical Analysis System) adatelemző szoftver felhasználásával.

Ez egy oktatási segédanyag, mely egy 2011. tavaszi gyakorlati képzés tapasztalatain alapul. Konkrétan a Szegedi Tudományegyetem környezettan szakos végzős hallgatóinak a szakdolgozatának kockázatbecsléséhez készült a tananyag. Célunk, hogy összefoglaljuk az elhangzottakat konkrét feladatmegoldásokon keresztül, illusztrációkkal, illetve kiegészítsük a környezeti szakmai ismeretekkel.

Csicsman József informatikus hallgatóknak tart az SZTE és mérnököknek a BME-n is „Statisztikai szoftverek” c. kurzust, melynek oktatási anyagait megtaláljuk az egyetemek weboldalaiban¹. Ezen oktatási segédanyag fontos kiegészítője az első függelékben megtalálható, a statisztikai fogalmak és módszerek alapjait ismertető dokumentum.

1.2 VERZIÓKÖVETÉS

Dátum	Verzió	Leírás	Módosította
			Csicsman József,
			Gálfi Márta
2011.08.09.	v1.0	Első verzió	László Anna,
			Sipos Eszter,
			Soltész Gábor

¹ <http://www.inf.u-szeged.hu/~csicsman/oktatas/adatbanyaszat/adatbanyaszat.html>
<http://www.math.bme.hu/~csicsman/oktatas/adatbanyaszat/adatbanyaszat.html>

2. Természettudományos ismeretek rendszerezése a Környezet Irányítási Rendszerek és a vállalati tevékenységek kockázatelemzési szükségességének megértéséhez.

A földi viszonyok zajló evolúciót tanulmányozva a biológiai evolúció eseményeit emeljük vizsgálataink fókuszába. Tesszük ezt azért, mert ezen folyamatokról mindenkinek vannak előzetes információi, másrészt ezen folyamatok objektív bizonyítékokon alapszanak, s mint ilyenek számos szempontból akár modellként is tanulmányozható környezet és biológiai rendszerviszonyokat képviselnek.

Az eseményeket átvizsgálva, azok megértése kapcsán a következő megállapításokat tehetjük:

A,

- Az élet kialakulásához és fennmaradásához szükséges feltételek (igények), azaz a klasszikus értelemben vett földi környezet folyamatosan változik, módosul az idő függvényében.

Mindennek bizonyítéka pl. a prokariota anaerob szervezetek, mint kozmopolita élőlények elterjedése a Földön. Viszont, e nagy biomassza tömeget képviselő említett élő anyagforma a napsugárzás hullámhossz-függő elektromágneses energiájának hasznosítására alakított ki biológiai struktúrát, megindult a fotoszintézis. Mindez egyrészt azt eredményezte, hogy az organizmusok egyre komplexebb szervezeteikként (egyre magasabb rendű élőlényekként) a Földi körülményekhez adaptálódtak, aminek következtében egyre több oxigén (O_2) halmozódott fel a Föld légkörében. Másrészt a redukáló légkör pontosan a folyamatos légköri parciális O_2 tenzió (pO_2) emelkedés miatt átalakult oxidatívvá, ami megkövetelte az ehhez a körülményhez való alkalmazkodást. Így azután a szárazföldi élőlények oxidatív adaptációra képes biológiai struktúrákat tartanak fenn, mert ennek hiányában elpusztulnának. Azon élőlények amelyek megőrizték az anaerob körülményekkel való egyensúlytartás strukturális lehetőségeit, a mai napig is úgy élnek a földi viszonyok mellett, hogy annak anaerob, azaz emelkedett parciális CO_2 tenzió (CO_2) mellett tartják fenn életjelenségeiket.

A vázolt evolúciós eseménysorozat (folyamat) kapcsán a következő megfigyeléseket tehetjük:

(1) Évezredek alatt zajló folyamatkaszkádként értelmezhető az említett földi légkör redukáló jellegének oxidatívvá válása - (az idődimenzió meghatározható, amely alatt a változás lezajlott).

(2) Az említett folyamat az élettelen légkört - mint környezeti elemet -, az egész földi rendszert érintően alakította át. Mindezt az élettelen rendszeren, mint feltételen alkalmazkodni képes biológiai anyagforma (amit jelen esetben élő organizmusokként is értelmezzünk) jelenléte gerjesztette (tehát az élő rendszerek hatására alakult ki az összes földi.

(3) A földi körülmény-változásokat az élő organizmusok folyamatos jelenléttel képesek a konkrét biológiai rendszerállapotok változássorozatán keresztül követni, (biológiai rendszerállapotuk egyensúlyi tartományai környezetéhez iteráltatásával).

(4) A légköri változás a biológiai anyagforma akkomodálódott rendszerében (oxigenizált légkörhöz való alkalmazkodásban) is globális a földi viszonyok tekintetében.

(5) Az új alkalmazkodási mintázat lassú turn-over mellett, optimalizálódott, úgy, hogy a régi rendszerállapotok is megőrződtek, a nekik alkalmas kritériumok esetén (megszüntetve megőrző mód)

B,

- A biotikus rendszerek (biológiai anyagformák, melyeket infra- és szupraindividuális szerveződésekként is értelmezhetünk) együttesen (globálisan) és külön-külön (lokálisan) is hatással vannak környezetükre és egymásra, ami a környezet sajátosságait meghatározó abiotikus tényezők változásait fenntartja. (Az állandó tehát a változás maga!)

C,

- Az abiotikus tényezők (élettelen környezeti elemek és feltételek) alapvetően azt a háttér anyag- és energiamező-rendszert képviselik, amelynek jelenléte nélkülözhetetlen az aktuálisan optimalizálódott élő anyagforma kialakulásához, fennmaradásához.

Következményesen kimondható tehát, hogy,

- a biotikus rendszerek strukturális szerveződésének feltétele a környezet, azaz a „háttér” jelenléte.

- a dinamikus anyagi struktúrákat hordozó élő anyagformák alkalmazkodásukkal követték a háttér, vagy ha úgy tetszik a környezet változásait.

- az alkalmazkodást meghatározó törvényszerűségek az egyre komplexebb biomateriális rendszerek révén valósultak meg.

- a korai embertípusok megjelenésével bezárólag a belső alkalmazkodási módozatok (szervezeti strukturális elemekkel történő adaptációk) tárházát vonultatták fel.

Belátható tehát, hogy az alkalmazkodás minősége és mértéke döntő fontosságú és az evolúció folyamaként értelmezhető. Az evolúció szempontjából sikeresebb, („fejlettebb”) élő rendszerek a létező összes változással képviselt kihívásnak jobban meg tudnak felelni, ugyanis strukturális felépítésük biológiai funkciójukat tökéletesebb, szervezettebb (komplexebb) módon szolgálja. Ez a magyarázata annak, hogy az evolúció folyamán változatos felépítésű életformák jelentek (és jelennek) meg, melyek eltérő módon képesek az optimális alkalmazkodásra.

Az életteleniség összes kritériumát hordozó alacsony komplexitású rendszerek, az élettelen rendszerek, melyek végtelen szabadságfokkal rendelkeznek az említett fő szempontrendszer megtartásában. (Az életteleniségnek végtelen a feltételrendszere, azaz a szabadságfoka.) Külső alkalmazkodással, az adaptációval valósulnak meg a konkrét, ott és akkor egyensúlytartásra képes egyensúlyi állapotok az életteleniség rendszerfolyamában.

A biológiai struktúrák magas komplexitású kontextusaival jellemezhető rendszereiben, az élettelen (ké-sőbb) és/vagy élő környezeti feltételeken, már kritikus határokon belül értelmezett csak az élet. Tehát az evolúció ezen szintéren konfidencia határokkal jellemezhető tartományokon belül értelmezett akkomodációs folyamatokkal tartja fenn egyensúlyi állapotát. Az élő anyagforma létezésének feltételrendszere szigorúan definiált, tehát határokkal jellemzett az élet szabadságfoka.

Az evolúció során az adaptáció, akkomodáció folyamában az ember, mint biológiai organizmus vált alkalmassá organikus struktúrája és erre épülő funkcionalitása (mentális képességei) révén, a társadalmi szintérre terelt evolúció megvalósítására. Jelenleg a legmagasabb komplexitású evolúciós szintér a társadalom, melyben a humán szereplők kooperatív cselekvési mintázataikkal képesek a környezetet olyan módon átalakítani, hogy létezésük és evolúciós szintjük igényeit folyamatosan optimalizáltan fenn tudják tartani.

Az evolúciós feltételek, az élettelen, élő valamint társadalmi összes kommunikáció és kontextus fenntartását, és potenciális alkalmazkodási mintázatainak minél nagyobb szabadságfokkal történő megőrzését jelentik.

A földi viszonyok mellett, az anyagi szerveződés bármely szintjén megjelenő általános törvényszerűség (amely az élő rendszerekre is igaz) az önszerveződési folyamatok által triggerelt komplexitások folyamatos kialakítása és optimalizált fenntartása; - ami az alkalmazkodást, ha úgy tetszik, magát az evolúciót eredményezi.

2.1 AZ ÖNSZERVEZŐDÉS SZABÁLYOZÓ LÉPÉSEI

A földi viszonyrendszer egyensúlyi folyamatának fenntartásán keresztül egy, a környezetével (Föld és a komplement téregységek) részben nyílt-dinamikus anyag- és energiacsere rendszert alakított ki. A földtudományok és csillagászat szakmai módszertani tárházával meghatározható Földre jellemző optimális mérettartományok lehetővé tették, hogy e földi viszonyok mellett (annak lég, víz és szárazföldi tereiben) olyan egységek alakuljanak ki, melyekben kommunikációs szempontból eltérő sebességgel zajló információk (anyag és energia quantumokban) továbbítódnak.

Így, ha időszakosan is, de részlegesen kompartmentalizálódtak a földi viszonyrendszerek, azaz azon belül alrendszerek stabilizálódtak, s tovább ezeken belül is al-alrendszerek alakultak ki, egyensúlyi ciklusokként. Jellemzően megtalálható minden szinten a rendszerstabilitást biztosító törvényszerűség (algoritmus).

Az teljes földi viszonyrendszerre jellemző a rendezettség, mely körfolyamatokkal értelmezhető, antikaotikus állapot. Igaz rá, hogy addig, ott és akkor értelmezett az egyensúlyi ciklusfolyamattal leírható rendezettség, amíg azt az aktuális szintet definiáló feltételekkel való egyensúlytartás megengedi.

A magasabb szinten definiált egyensúlyi folyamatcikluson belül „kicsiny” változásokként értelmezhető eltérések, (pl. kozmikus sugárzás zuhatag, nyomásváltozás eltolódások, stb.) újabb belső egyensúlyi ciklusokat generálhatnak, mint azok feltétel kritériumai.

Tehát nagy Föld egyensúlyi ciklusrendszerén, mint egységes egészen belül az, egyensúlyt definiáló törvényszerűségek, azaz, általános algoritmusok (szabályok) nyilvánvalóan ugyanazok. Az ezek szerint értelmezett a működésben, az al- és al-al, ... és így tovább értelmezhető egyre kisebb komplexitású belső-rendszerciklusok eltéréseiben találjuk meg a specialításokat, amelyek a lokális tulajdonságokkal jellemezettek.

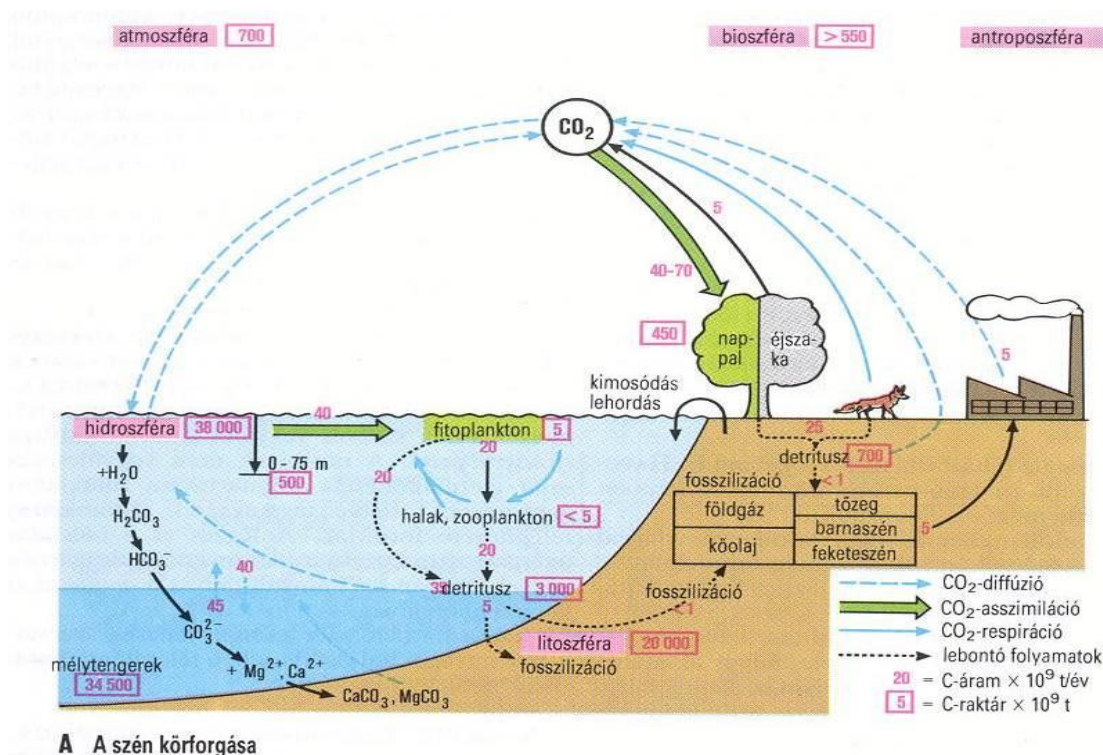
Így pl. a légkör specialításai az óceánok felett, más áramlási rendszerekkel értelmezhetők, mint a szárazföldi területeken. De ezek mindegyike eltér egyes elemeiben az egész Földre jellemző nagy légköri

viszonyoktól, noha alaptörvényszerűségeik, azaz természettudományos fizikai-kémiai folyamataik azonos módon definiáltak, csak az értelmezési tartományaik különböznek..

Az evolúció során megismert rend, (pl. az élettelen anyagforgalom tekintetében, de az élőlények esetén is) a pontosan összerendezett, folyamatosan zajló bio-geo-kémiai folyamatok eredményeként akár modellezhető is.

Vizsgáljunk meg néhány, földi viszonyrendszeren belül értelmezett ciklust a következő alfejezetekben.

2.1.1 Karbon (C) ciklus



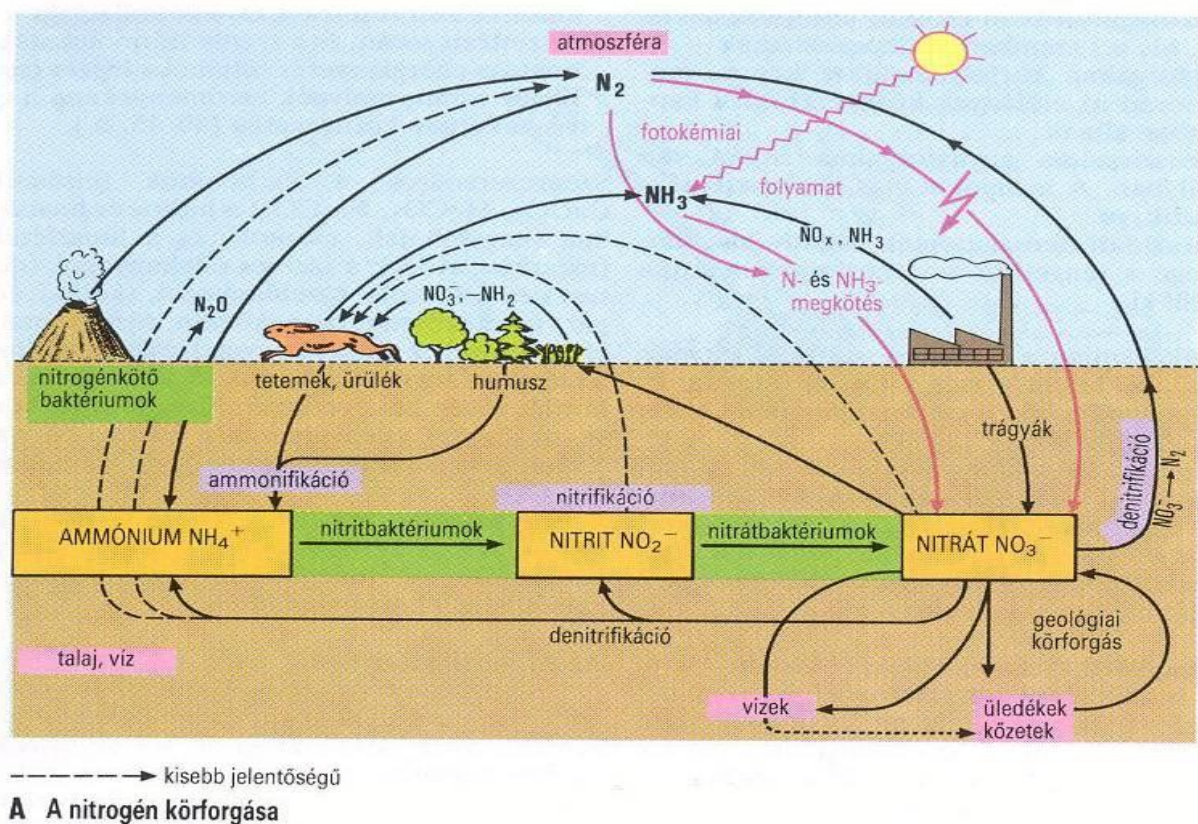
1. ábra: A szén körforgása

A karbon, azaz a szén-ciklus a földi viszonyok mellett egy esszenciális egyensúlyi al-ciklusrendszer. Ugyanis a földi viszonyok között csak az élettelen evolúcióban nem kapott esszenciális tulajdonságot a szén. A többi evolúciós szintéren (élő evolúció, társadalom) igen. A szén körforgás jelenleg az élő és élettelen anyagformákban való megoszlása szerint is vizsgálható, de a környezeti elemekben (talaj, víz, levegő) való jelenléte alapján, (stb.) is. Földünkön rezervoárként tartjuk számon a különböző környezeti elemek élővilágát, de a légkörben lévő 735 mrd. tonna CO₂ –ot, továbbá a talajban raktározott 1200-

1400 mrd. tonna szenet a vegyületeiben, vagy a tengerekben oldott 754 mrd. tonna CO_2 –ot és egyéb aljzati közetekben és ásványokban, üledékekben előforduló széntartalmak képviselte egyensúlyi nagy szén-ciklus anyagtartalmának jelentős részét.

A körforgalom (rendszer-ciklus) tanulmányozás evolúciós szintér szerint: élettelen adaptív folyamatokkal jellemzeten, élő, akkomodációs folyamatokkal jellemzeten, és társadalom minőségügyi tevékenység-folyamatokkal értelmezeten.

2.1.2 Nitrogén (N_2) ciklus

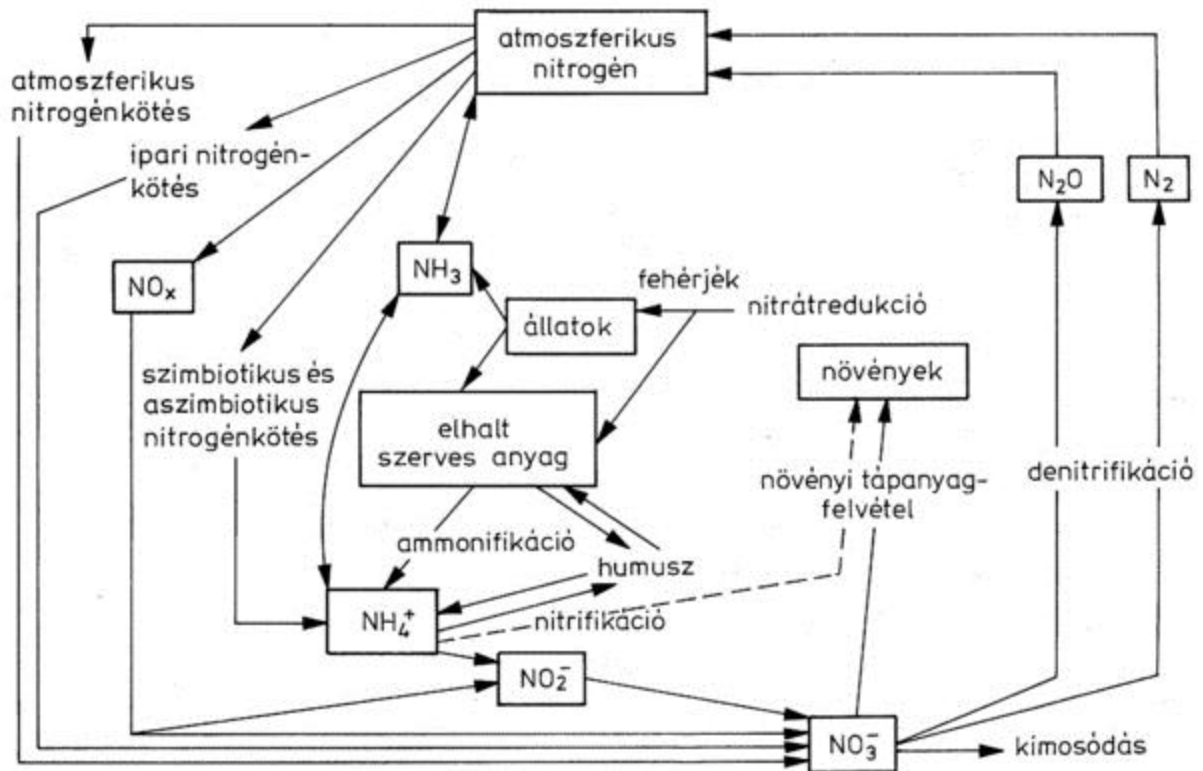


2. ábra: A nitrogén körforgása

A Föld viszonyrendszerében a Nitrogén igen nagy mennyiségben van jelen. Nagy rezervoárjai a litoszféra és a légkör. De a biológiai rendszerekben is tetemes mennyiségben van jelen (pl. fehérjék, nukleinsavak, egyéb nagy és kismolekulájú szerves és szervetlen vegyületek, élő szervezetek, stb.)

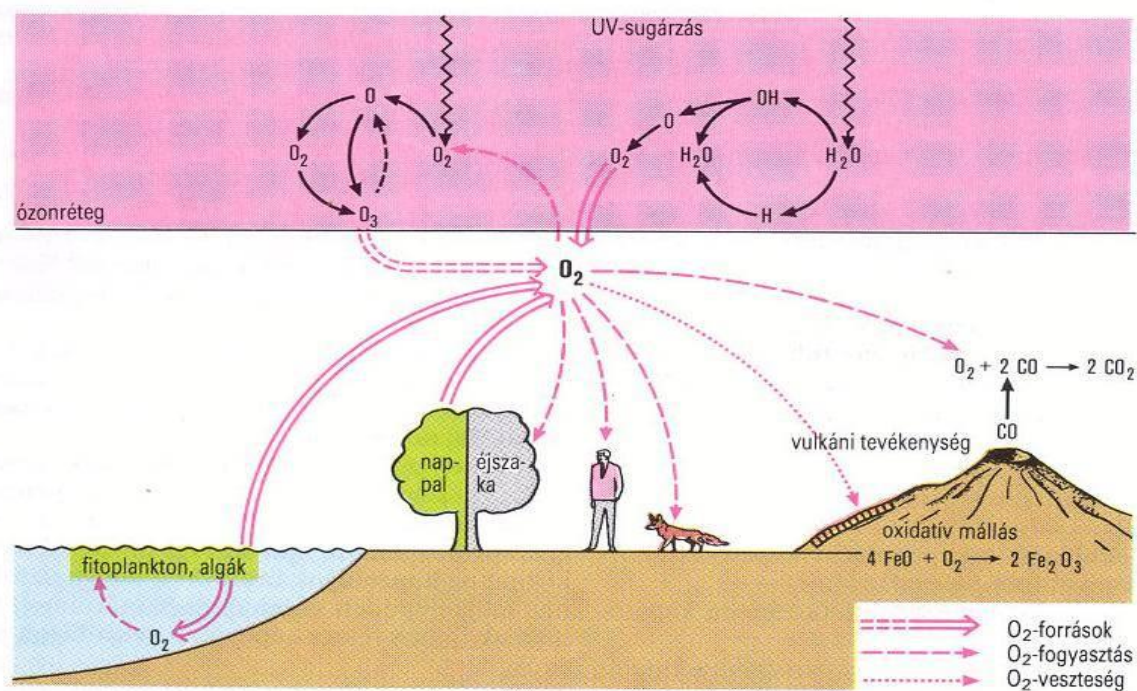
Annak ellenére, hogy a légköri nitrogén igen tetemes nitrogénforrás lehetne, mégis az élő evolúcióba a nitrogén a mikrobiális rendszerek nitrogén fixálásának eredményeként kapcsolódik be

(nitrogénmineralizáció: ammonifikáció, nitrifikáció, nitrátredukció, denitrifikáció), majd a producensek közreműködése révén halad tovább a nitrogén folyamat.



3. ábra: Nitrogén-körforgalom (Delwiche, 1970 után Gisi, 1990)

2.1.3 Oxigén (O₂) ciklus



B Az oxigén körforgása

4. ábra: Az oxigén körforgása

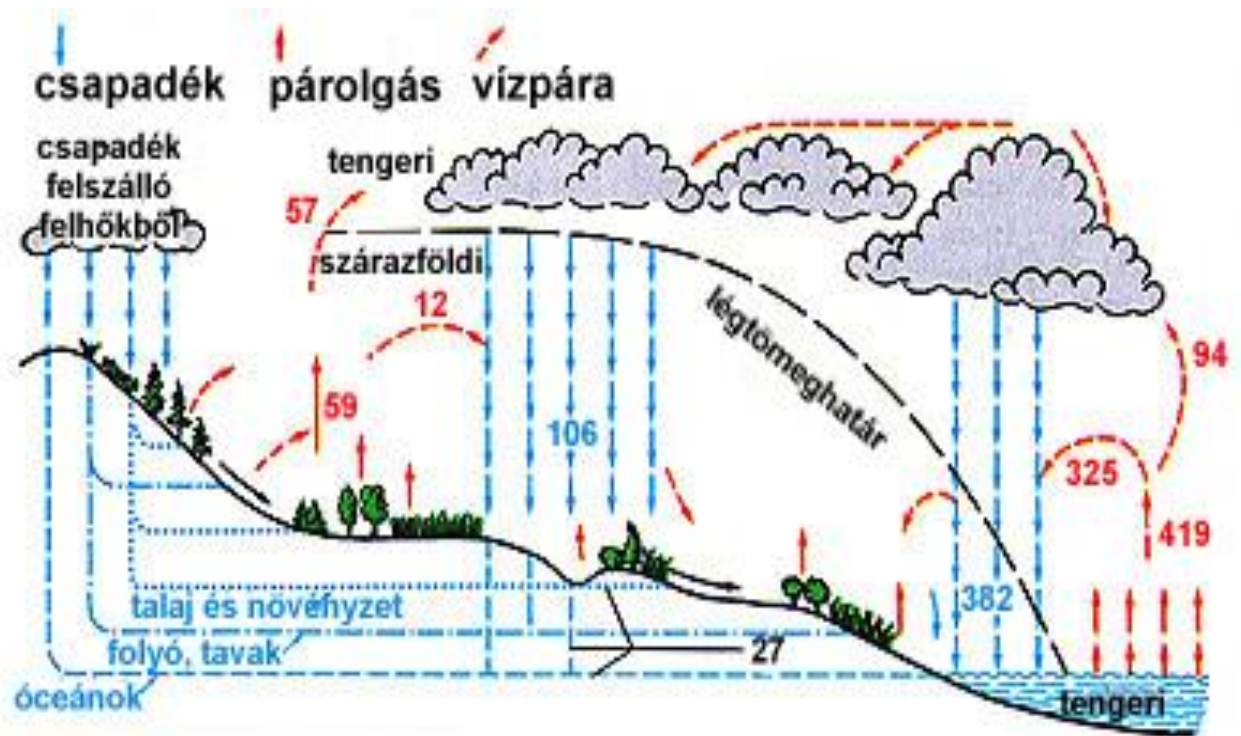
Az oxigén a földi viszonyrendszerben kémiaiilag kötött, tehát vegyületeiben (SiO₂, H₂O; stb.) és szabad formában (O₂) van jelen. A halmazállapota mutatja a környezeti elemekben való előfordulási gyakoriságát is. Az oxigén háromatomos formája az ózon (O₃), mely a földi viszonyok mellett jelentős alrendszer tulajdonsággal bír, ugyanis hiányában megszűnne a jelenleg értelmezett földi evolúciós egyensúlyi ciklus.

(Vegyünk észre, hogy lokális tulajdonság változások eredményezte al-ciklusok kialakulása biztosítja azokat a további feltételeket, amelyeken a rákövetkező egyensúlyi ciklusrendszerek megjelenése értelmezett.)

2.1.4 Víz (H₂O) és hidrogén (H₂) ciklus

A víz a hidrogén leggyakoribb vegyülete a földi viszonyrendszer ciklusban. Itt az élettelen és élő evolúciós folyamatban található a legnagyobb rezervoár a víz és a hidrogén tekintetében.

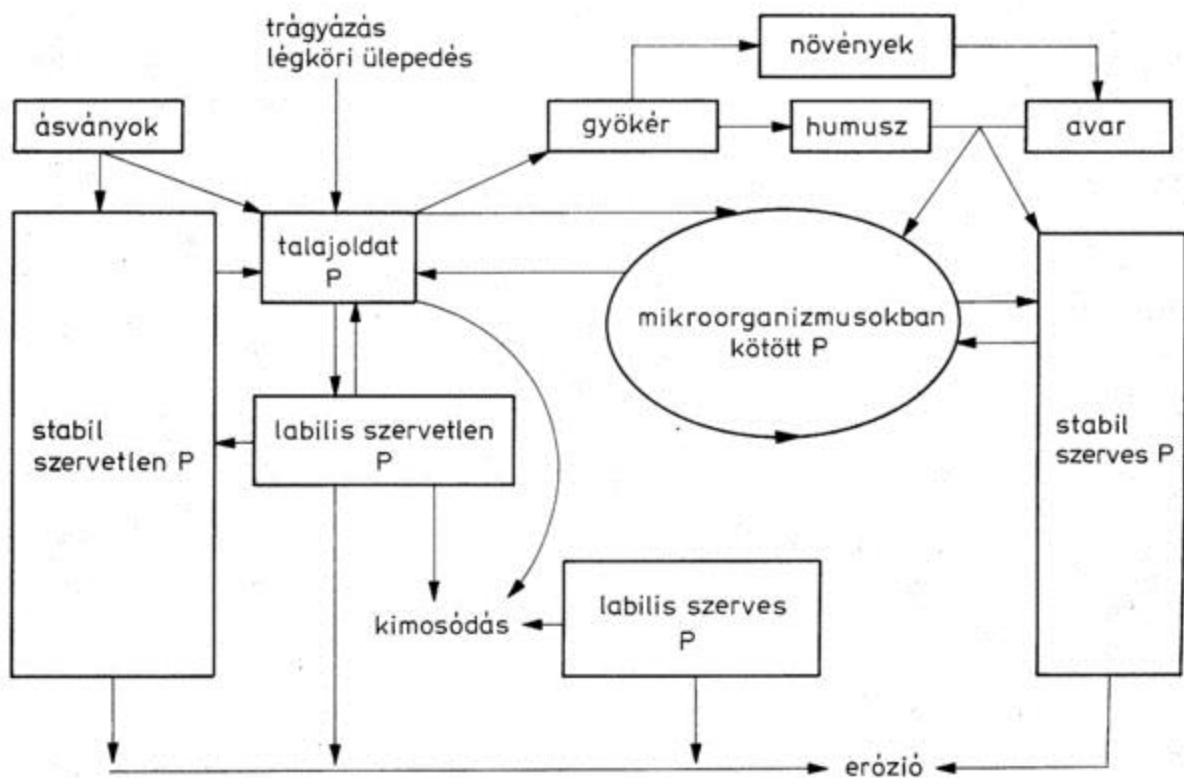
A víz különleges fizikai és/vagy kémiai tulajdonságai tették lehetővé, hogy az életfolyamban is esszenciális szerepe legyen.



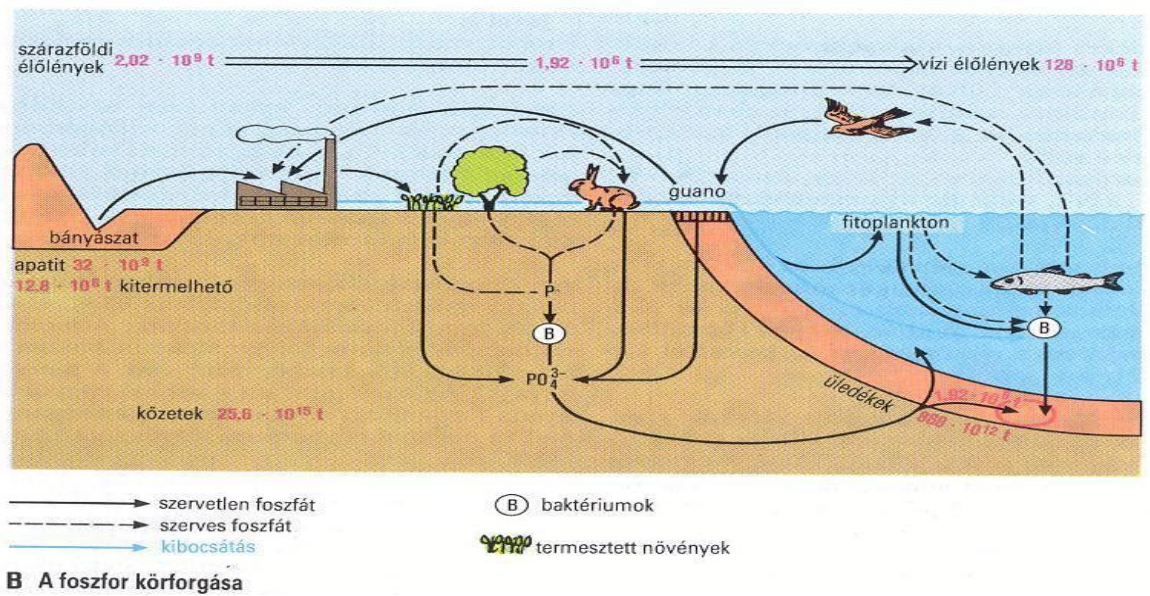
5. ábra: A víz és hidrogén körforgása

2.1.5 Foszfor (P) ciklus

A földi viszonyrendszerben jelenleg értelmezhető foszfor ciklus tipikus példája az élettelen (adaptív) evolúciós viszonyrendszer és az élő (akkomodációs) viszonyrendszer kooperatív eseményeinek. Ugyanis ma a foszfor készletek az üledékes (mészkö, homok, vagy foszforittelep, guano formájában) és mélységi kőzetben (gabbro - ahonnan nehéz a reciklizációja) vannak jelen. A Föld két evolúciós álfolyam-rendszere, mely adaptív és akkomodációs komplexitásként került eddig elkülönítésre (de az adaptív evolúciós folyamat lokális tulajdonságainak diszkrét részterületein, mint feltételeken értelmezett életfolyam is megjelölésre kerülhetett volna), egy teljes, új adaptó-akkomodatív tulajdonsággal bíró alciklusként fogható fel éppen a foszfor ciklus példáján.



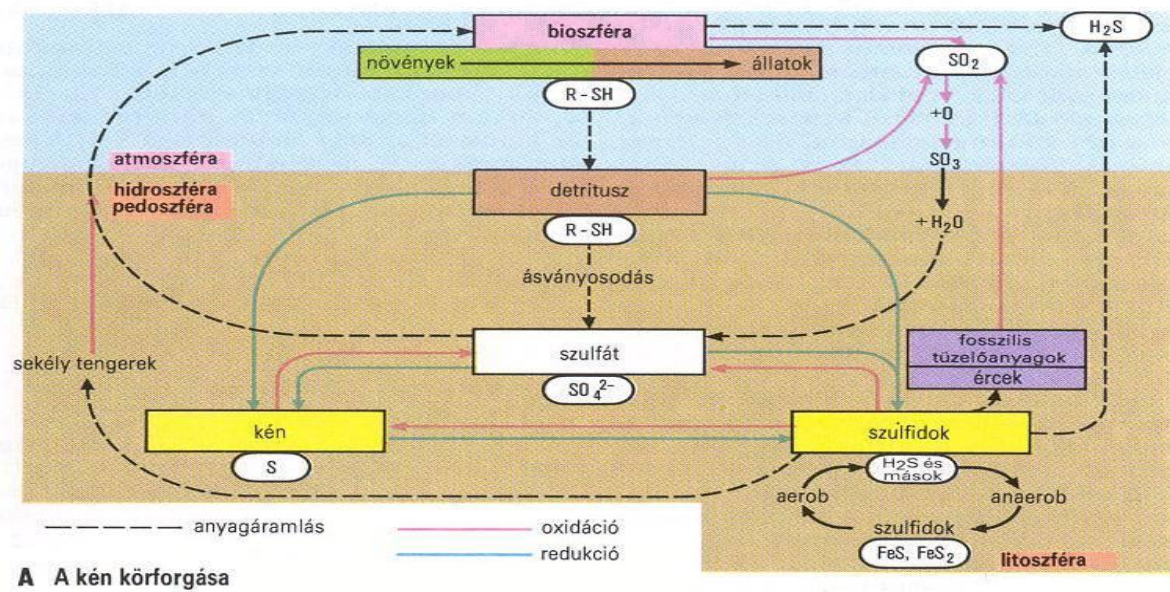
6. ábra: Foszforkörforgalom (Stewart, 1981 után módosítva)



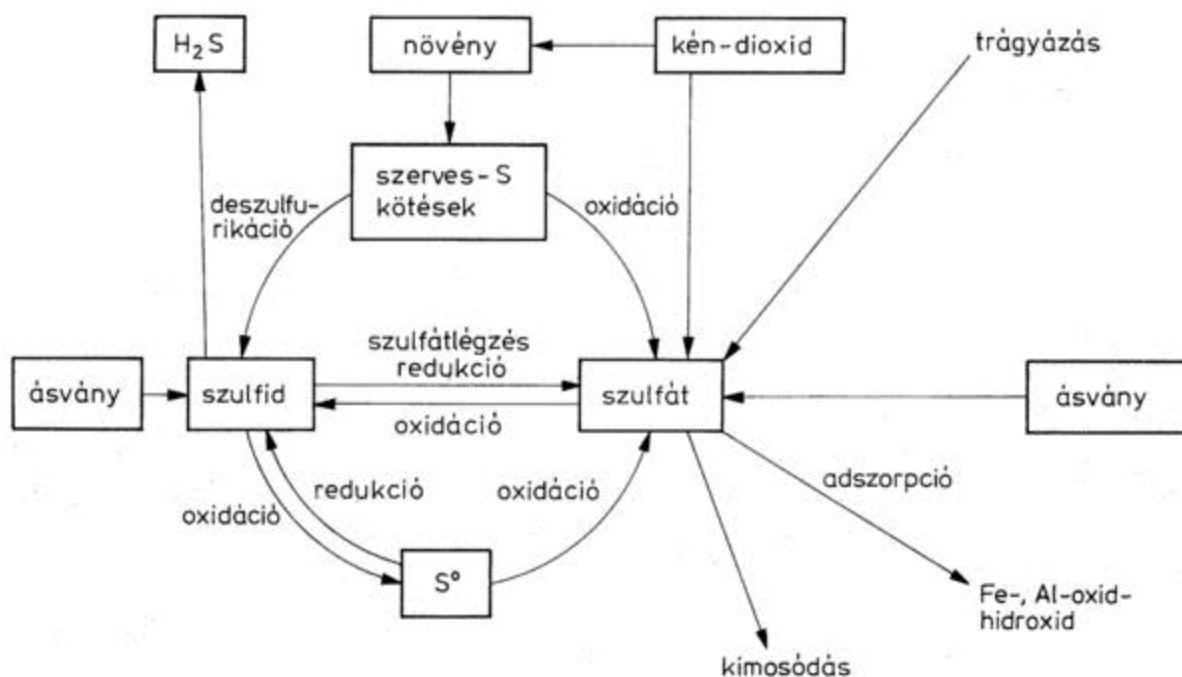
7. ábra: A foszfor körforgása

2.1.6 Kén (S) ciklus

A kén geokémiai és biológiai forrásokból származva van jelen a jelenlegi ciklusrendszerében. Mikrobiális folyamatokkal alakult ki a biológiai rendszerekbe kapcsolódása (N/S: 9/1; C/S: 15/1)



8. ábra: A kén körforgása



9. ábra: Kénkörforgalom (Brady, 1974 után módosítva)

2.1.7 A többi elem ciklusa

A makro- (K, Ca, Mg, Na,) és mikroelemként (Fe, B, Zn, Cu, Mn, Mo, Cl stb.) azonosított anyagok is ciklusfolyamatokkal jellemzetten vesznek részt az egységes földi viszonyrendszer stabilizálásában.

Jelenleg ezeket nem mutatjuk be, de visszautalunk arra a törvényszerűsége, hogy bármely, az alkalmazkodás szempontjából legkisebb mennyiségben jelenlévő elem által képviselt feltétel, mint egy újabb akkomodációs ciklus értelmezési tartománya, ha eltolódik (megváltozik), akkor az, az adott egyensúlyi-ciklusok teljes rendszer kaszkádjának megváltozását is generálhatja. Mindez azon múlik, hogy az illető változás

milyen mértékű?

Ennek a megfontolásnak megfelelően elkülönítünk „*kiszavarokat*”, amikor az illető ciklus-folyamatrendszer fennmaradását meghatározó feltételeinek módosulása kapcsán, még reverzibilis formában zajló változásokat detektálhatjuk. Azaz, a tapasztalt rendszerváltozás során a vizsgált állapotciklus rendszerét jellemző tulajdonságok még megmaradnak (annak anyag – energetikai tulajdonságai és struktúrája ugyanabban az állapotciklusban marad). Viszont mindez bizonyos tűréshatárokon belül, erősen diszkrét konfidencia intervallummal jellemzetten értelmezett.

(pl. Nincs elegendő foszfát felvételre lehetősége egy sejtnek. Így a sejtben azon anyagok szintézise, amelyekhez foszfát igényelt, biztosan korlátozott lesz, mert a limitet mindig a legkisebb koncentrációban jelen lévő anyag fogja képezni. (Liebig-féle minimum törvény, ami a növényi anyagfelvétel kapcsán került megfogalmazásra.) Viszont ez nemcsak azt jelenti, hogy kevés lesz a foszfátot igénylő szintézistermékek mennyisége, hanem azt is, hogy mindazon funkciók, amelyek ezen, limitált mennyiségben jelen lévő szintézis termékekkel hordozottak (feltételen stabilizálódott al-alrendszer ciklusok) már csak lecsökkent formában valósulnak meg (hypo-funkciók).

Belátható az iménti példa kapcsán is, hogy a tartósan jelenlévő kis zavar eredményeként, a rendszer mégis elhagyhatja saját állapotciklusát. Ugyanis egy új feltételrendszerhez kell adaptálnia a rendszernek, ha nincs elég foszfor, mert következményesen nem lesznek jó kommunikációs folyamatok sem a ciklusrendszeren belül. Ezért először új stabilitási al-alciklusok jönnek majd létre, melyek finom változásrendszerének eredményeként a fő ciklus megváltozott feltételtartományához való alkalmazkodás alakul ki. Így viszont arra visszahatva, annak működése is egy egészen új rendszerállapot megjelenését váltja ki. Ezek a kiszavarok okozta lavinaesemények: a sérülések vagy máshogy fogalmazva a megváltozott feltételek kiváltotta következmény-módosulások, különböző mértékben terjedhetnek szét egy ciklusrendszerben.

Bekövetkezik a folyamatos „kiszavar” terhelések, vagy nagyobb hatás eredményeként „strukturális” zavar, ami azt jelenti, hogy az illető ciklusrendszerben az azt alkotó elemek közötti kapcsolat végérvényesen megváltozik. Ennek következménye a rendszer megváltozás lesz. Így az adott, bármely szintű ciklusrendszer folyamataiban ez a zavar minél inkább szétterjed, annál nagyobb valószínűséggel következik be az új alkalmazkodási mintázat kialakulása, ami egy új állapotciklus megjelenését képviseli.

Megállapíthatjuk tehát, hogy egy adott szintű ciklusfolyamat értelmezési tartományában (vonzási tartomány) azaz, feltételrendszerében, ha változás történik, akkor az új állapothoz való alkalmazkodási mintázat kialakulása,

- vagy egy új, de még az eredet rendszer működési szabályait és elemkapcsolatait megőrző formában és diszkrét módon megváltozva kerül követésre;
- vagy egy teljesen új struktúra alakul ki, azáltal, hogy hirtelen, vagy fokozatosan új elemkapcsolatok determinálják a volt rendszert, ami ebből adódóan egy új rendszerré alakul át.

Az átalakulás folyamatában jelentős szerepe van az új rendszerre jellemző struktúra szétterjedésének. Ha ezt lokalizálni lehet, a rendszer állapotciklus változása megakadályozható a kiterjedéstől függő mértékben. Ha nem sikerül a struktúraváltozásokat, és a folyamatok lavinarendszerét a kiindulási állapotciklus feltételeihez való alkalmazkodás szintjén minimalizálni, vagy megszüntetni, úgy az állapotciklus egy új állapot megjelenésével alkalmazkodik a változásokhoz.

A rendszerek belső stabilizáló állapotciklusaik változásaival iterálnak a feltételek módosulásaihoz. Így a változások az esetek nagy részében nem látványosak, viszont a bekövetkezés pillanatában sokszor menthetetlenek.

2.2 A TÁRSADALMI SZINTEN ÉRTELMEZETT ALKALMAZKODÁS

Az evolúció jelenleg ismert legmagasabb komplexitású szintere a társadalom. Ebben a rendszerben a már megismert adaptációs, akkomodációs feltételeken, mint értelmezési tartományon működik a társadalom.

Hogyan értelmezhető az összefüggésrendszer? Miért szükséges meglássuk az egységes rendszert a földi viszonyok között, melynek a társadalmi állapotciklusok is csupán alkalmazkodni képes részei?

Könnyen belátható, ha a földi viszonyrendszerek tanulmányozásakor alkalmazott ok-okozati összefüggések kutatási módszerével szisztematikusan és tematikusan vizsgálódunk.

Korábban beláttuk, hogy az élő komplexitások megjelenése előtti élettelen rendszerek folyamatos változásai az élettelen anyagi minőségek sokféleségének megjelenésével jellemezhető, alkalmazkodási mintázattal alkalmazkodtak a folyton változó környezethez.

pl. Ami azt jelenti, hogy ha egy adott anyagi rendszer környezetében egy kritikus értéknél nagyobb energiaközlés jelent meg, és ezzel találkozott az illető anyag, akkor más rendszerré, anyaggá alakult át. Tehát kialakult egy másik élettelen anyagrendszer, növekedett az élettelenség Földön megjelent, és stabilizálódott mintázatainak száma. Az új anyagi minőség kialakulásával biztosítottá vált egy új, ott és akkor fenntartható egyensúlyi állapot, amit a megváltozott feltételekhez való adaptív alkalmazkodásként

értékelhetünk. Ebben az esetben nem az történt, hogy az azonos anyagi minőségen belül alakult tovább az élettelen anyagi struktúra, hanem egy másik anyagi struktúra jött létre, ennek megfelelően (ugyan nem a leghelyesebben), külső alkalmazkodásról beszélünk, melyet az élettelenség megőrzésével teljesen eltérő anyagi rendszerek állapotciklusain keresztül értelmezhetünk.

Az élettelenséghez rendelt külső alkalmazkodási mintázaton, mint adaptáción keresztül az anyagi állapotciklusok olyan mintázatokat is kialakítanak, amelyek segítségével a lokális mintázatok teljesen lehatárolt formában, az egységes földi viszonyrendszerekben belül, azoknak csak diszkrét állapotaira érvényes szerves-anyag mintázatokat képviselnek. Ezen mintázatok diszkrét, önálló működési egységként értelmezett kompartmenteket képeznek, amelyekben az egyensúlyi feltételek megtartására, az élettelen állapotciklusokra nem jellemző tulajdonságok is felfedezhetők. Az így kialakult rendszerek, az élő rendszerek (véletlenszerű párhuzamos működésű rendszerek, bináris rendszerekként modellezve vizsgálhatók).

A folyton változó feltételrendszer, amelyen értelmezett az élet (az élő anyagforma működése), igen diszkrét területét képviseli az élettelen adaptációs élettelen környezet földi viszonyrendszerében.

Az élő anyagforma alkalmazkodása az akkomodáció, amely rendkívüli tulajdonsággal, az élő struktúrán belüli változékonyság megjelenítésével alakítja ki az egyensúlyi állapotokat (pl. biológiai organizmusok, szerveződési szintek révén). Az élő anyagforma hihetetlen gazdagságban mutatja be az életjelenségek prezentálását a különféle élő egyensúlyi állapotciklusain keresztül, ha úgy tetszik a komplexitásában növekvő genetikai folyamatok expressziójában funkcionális szinten.

Az életfolyam tehát az élő anyagformák nagy diverzitásának prezentálásával akkomodálódott a földi viszonyrendszer csak bizonyos feltételein belül.

Az ember, mint jól definiálható biológiai komplexitás olyan evolúciós eredményt képvisel, ahol az energia-transzformáció a tudati energiaszinteken is értelmezett. Ez az energiatranzíció tette lehetővé, hogy az evolúció színtere a társadalom legyen. A magas biológiai komplexitást képviselő ember, nem alkalmas azon létezők betöltésére, amelyeken pl. napjainkban szétterjedve jelen van. Ehhez csak evolúciós szerepének betöltésével, a külső környezet társadalmi szinten történő átalakításával juthat el. Így folyamatosan, evolúciós determinizmusként alakítja át külső környezetét a társas ember, ami a környezet-átalakító külső alkalmazkodásként értelmezhető.

Ennek értelmében a társadalom szereplői evolúciós hivatásukként folyamatosan alakítják át a környezetüket, mely az adaptálódott és akkomodálódott feltételrendszerek szükségszerű megváltoztatását jelenti.

Ez a környezethasználat minden határon nem működhet, mert a rendszerállapotok, állapotciklusok csak a vonzási tartományokon értelmezettek, melyek ha megváltoznak maguknak a rendszerállapotoknak a változásait okozzák, a már leírt kis-zavaroktól a strukturális zavarokig.

A társadalmak tehát abban érdekeltek, hogy a természet rendszerállapot-ciklusait ismerjék, azok feltételeit jelentő állapotciklus vonzási tartományokat olyan módon nem „sérítsék”, hogy az evolúció folyamatát a társadalmi szintről, egy ma még értelmezhetetlen szintre taszítsák.

Ennek a megfontolásnak megfelelően, valamint a földi viszonyok között értelmezett állapotciklusokkal összeegyeztethető törvényszerűségekkel harmonizáló társadalmi tevékenység rendszermintázatot kell kialakítania az emberiségnek, melyben a megjelölt követelményeket a minőségügyi rendszertevékenységek fedik le.

Ezek olyan komplex, feltételeken, mint vonzási tartományokon optimalizált állapotciklusok, amelyben a tevékenységek eredményeként a társadalom számára hasznos termékek kerülnek kialakításra, miközben az evolúciós folyamat nem sérül. Tehát az adaptív és akkomodatív lehetőségek nem sérülnek.

Tekintsünk erre példákat a következő alfejezetben.

2.3 PÉLDÁK

A társadalmak tevékenységei kapcsán pl. a termőföld (talaj, mint környezeti elem) igen sokrétűen kerül hasznosításra (mezőgazdaság, ipar, kulturális jelleg, stb.). A társadalmi igények szerinti hasznosítás során megváltozik a föld (talaj) elemtartalmának lokális összetétele. Megváltozhat a nitrogén tartalma, a foszfát tartalma, kálium tartalma, stb, de változhat pl. a nehézfém tartalma is.

A földi viszonyok között értelmezett összes természetes anyagtartalom közel állandó a földön, de azok eloszlása megváltozik, ami különféle energia és anyagáramok kialakulását eredményezheti.

Ezen túlmenően, megváltozik a lokális jellegek eltolódása miatt a talajok élettérként való hasznosulása is, hiszen e megváltozott feltételekhez megváltozott igényű életközösségek képesek alkalmazkodni, akkomodálódni. De megváltozik ez egyes élőek alkalmazkodási potenciálja is, mert a környezeti stresszek hatására különféle „betegségekkel” reagálnak az élőek, ami egyedi életciklusaik mellett akár életközösségek pusztulásához is vezethet.

Így, ha ismert tevékenységgel pl. növeljük a talajok nitrogén és foszfát tartalmát, akkor az azon élő producens szervezetek minőségének átalakulását, továbbá a vizek elszennyeződését, majd élő vizek eutrofizációját okozhatjuk.

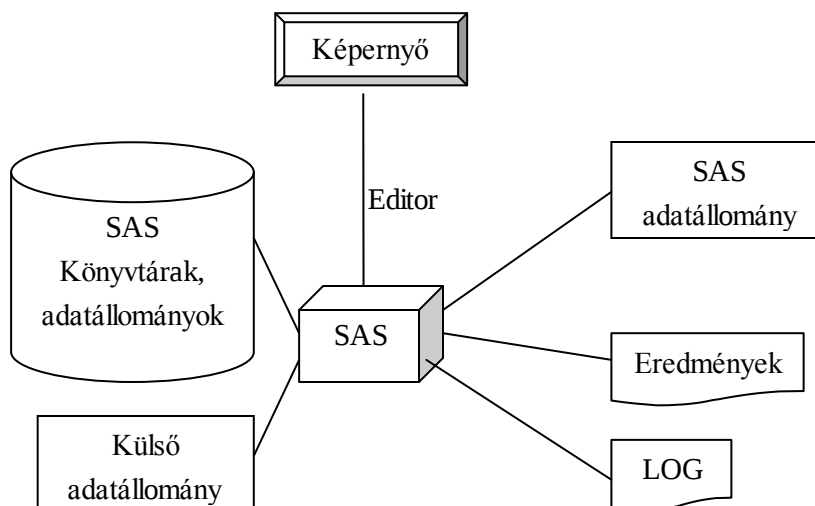
Ezért fontos, hogy amikor pl. egy mezőgazdasági tevékenység kapcsán Környezetirányítási rendszert működtetünk, akkor a tevékenység kapcsán a talaj állapotát kövessük, s szűk lokális adatainkat a nagyobb rendszerekbe csatlakoztassuk. Ezzel a nagyobb rendszerek követelményeihez illesztjük tevékenységünk körülményeit, nevezetesen pl. a nitrogén és foszfát-szennyezéseket.

A környezetirányítási rendszerek és a vállalati tevékenységek kockázatelemzési feladatainak mérésére szolgáló adatok kezelését, és elemzését mutatják be a következő fejezetek.

3. A SAS Base környezet megismerése

3.1 RENDSZER

A következő ábra a SAS rendszer felépítését szemlélteti:



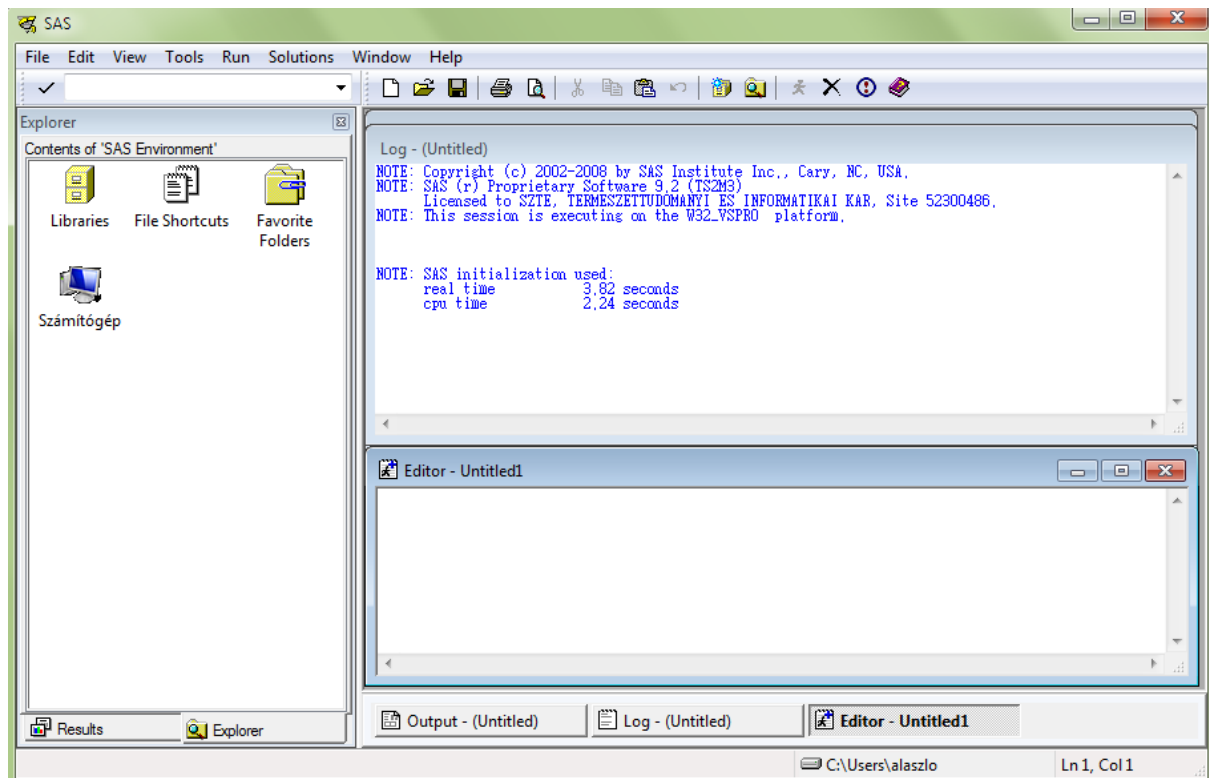
10. ábra: SAS rendszer

A SAS rendszer SAS adatállományokkal dolgozik, melyeket könyvtár struktúrában szerveződnek. Terméyzetesen mód van külső adatállományok beolvasására is. Az adatok fizikai elérésére logikai könyvtárnévvel hivatkozunk, mert a SAS-nak logikai adatrendszert használ. A SAS adatállományokat a Program Editor ablakban szerkesztett SAS base programokkal szólítjuk meg, ezen keresztül történik a kommunikáció a programozó és a SAS rendszer között a képernyőn. A SAS programok futását a LOG ablakban naplózza a SAS, melyben a programozó nyomon követheti, hogy a számítógép hogy értette meg, amit megadtunk neki. A futási eredményeket a rendszer kinyomtatja (pl. listing nyomtatási, vagy HTML output), melyeket utólag a Results ablakban előhívhatunk.

Lehetőség van arra, függően a számítási kapacitástól, hogy a SAS helyi gépre legyen telepítve (saját PC-n fusson), de akár távoli szerveren vagy mainframe-en is futhat. Mainframe-re akkor van szükség, amikor nagy számítási kapacitás kell (pl. banki terminálok ugyanarra a gépre kötődnek, ehhez hatalmas kapacitás kell).

3.2 FELÜLET

A következő ábrán láthatjuk a SAS 9.2 Base kezdőfelületét, amit a program elindítása után látunk.



11. ábra: SAS 9.2 Base kezdőfelület

A SAS Base szoftver felhasználói felülete hasonló más programok kezelő felületeihez. Könnyen áttekinthető, ablakos szerkezetű (több ablakból, részből áll).

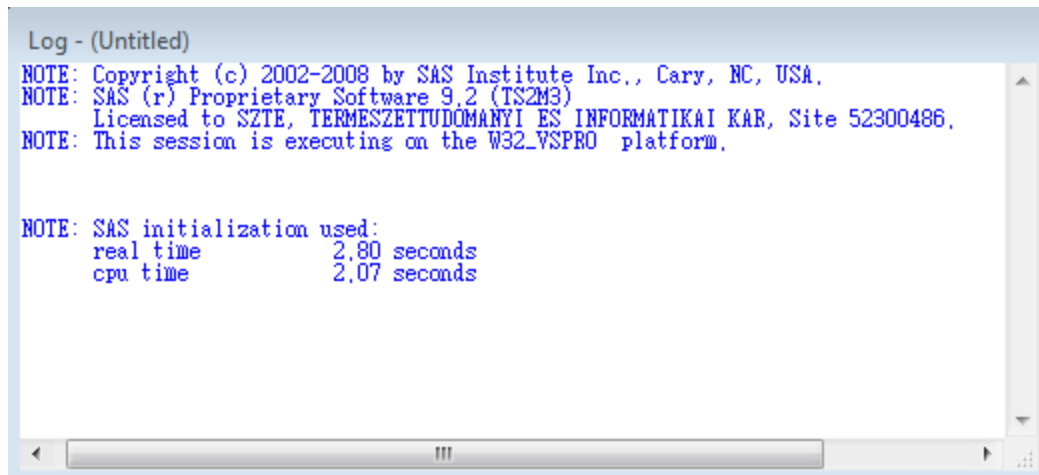
Felül van egy menüsor, mint azt sok esetben megszokhattuk (pl. Microsoft Office, SPSS, stb.), melyből néhány fontosabb funkció ikonja az alatta lévő sorban ki van helyezve. Az egyes ablakokra (Results, Explorer, Output, Log, Editor) kattintva az annak megfelelő menü és ikonsor jelenik meg a felső részen.

Az egyes ablakok külön igény szerint átméretezhetők.

3.2.1 LOG

A LOG a naplózó ablak, ahol nyomon követhetjük, hogy egy-egy programfuttatást hogyan értelmezett a SAS. Innen könnyen kiszűrhetjük, hogy mely sorokban vétettünk hibát, illetve, hogy mi lehetett az oka. Az eredményről és a futási időről is információt ad.

Érdeemes nézni munka közben ezt az ablakot, hisz ezzel gyorsan tudunk ellenőrizni, hogy a program valóban azt csinálta-e, amit szerettünk volna.



12. ábra: LOG naplózó ablak: a program betöltését ismertető sorokkal

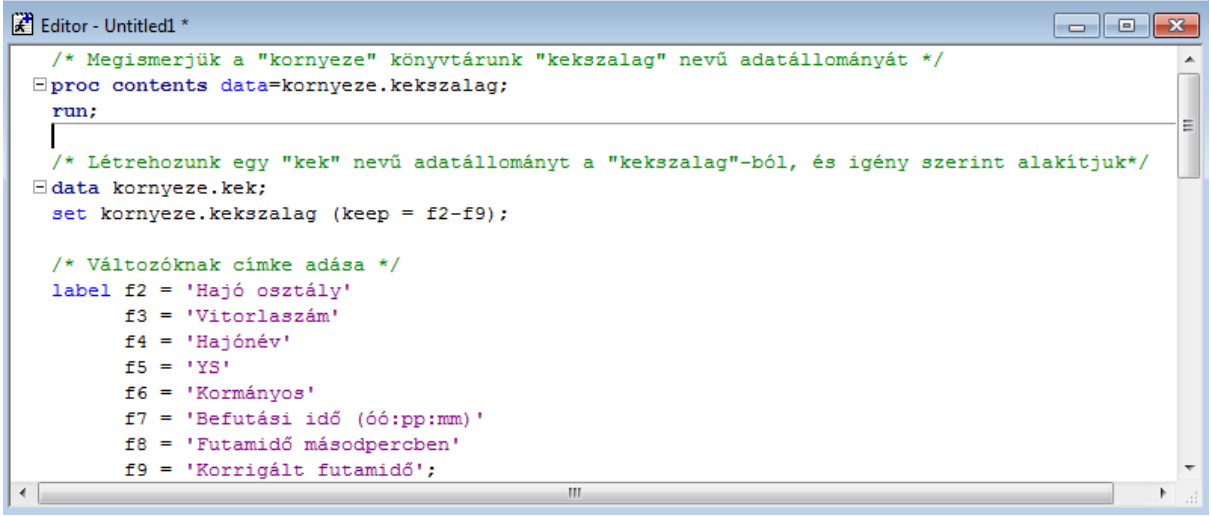
A LOG ablakban a kékkel írt sorok jelzik, ha minden „rendben van”. A SAS program elindításakor (inicializálás) a licenszről illetve az inicializációs (betöltődési) időről (jelen esetben 2,8 másodperc alatt töltött be) ad információt.

Ha hamarosan lejár a licensz, vagy valamiféle WARNING üzenetet kapunk, amitől még a programok tudnak futni, de észrevételként megjegyzi nekünk a szoftver, azt zölddel jelzi a LOG. A piros hibákat kell javítanunk, azok az ERROR üzenetek.

3.2.2 Editor

Az Editor ablak, ahol a SAS programokat szerkeszthetjük. Az alábbi ábra egy részletet mutat egy SAS kódsorból. Láthatjuk, hogy értelmezi az Editor a parancsokat, és aszerint színezi a szavakat, részeket. Így a főbb utasításokat kékkel jelzi, a futásban nem szereplő, de az olvasó számára hasznos információ-

ókat tartalmazó kommenteket (/* */ jelek közötti részek) zölddel, a szövegesként értelmezett részeket lilával, a hibás, nem ismert részeket pirossal színezi. A többi utasításrészt feketével írja a program editor.



```
/* Megismerjük a "kornyeze" könyvtárunk "kekszalag" nevű adatállományát */
proc contents data=kornyeze.kekszalag;
run;

/* Létrehozunk egy "kek" nevű adatállományt a "kekszalag"-ból, és igény szerint alakítjuk */
data kornyeze.kek;
set kornyeze.kekszalag (keep = f2-f9);

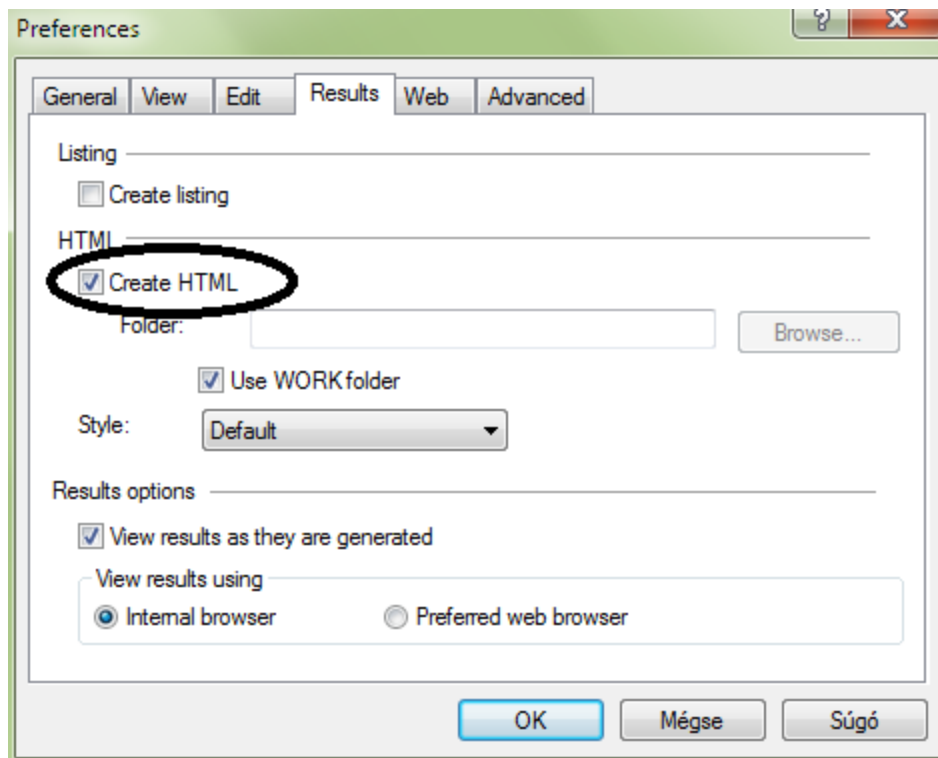
/* Változóknak címke adása */
label f2 = 'Hajó osztály'
      f3 = 'Vitorlaszám'
      f4 = 'Hajónév'
      f5 = 'YS'
      f6 = 'Kormányos'
      f7 = 'Befutási idő (óó:pp:mm)'
      f8 = 'Futamidő másodpercben'
      f9 = 'Korrigált futamidő';
```

13. ábra: Program Editor: SAS program szerkesztő ablaka

Ha lenyomjuk a  „Submit” ikont, akkor minden utasítás, ami az Editorban szerepel, lefut. Ha csak részleteket szeretnénk lefuttatni, akkor jelöljük ki a kívánt kódrészletet, és azután nyomjuk le a futtatást. A futás folyamatát és eredményét a LOG-ban nyomon követhetjük, magát az eredményt az *Output*, a *Results* illetve a *Results Viewer* ablakokban tudjuk megtekinteni.

3.2.3 Output

Alapértelmezetten a SAS-ban a listing output van beállítva. Ez egyszerű nyomtatási, karakteres output formátum, mely mellett szebb kimenetet ad a HTML output. Ezt a *Tools\Options\Preferences* menüben lehet beállítani a *Results* munkalapon a „Create HTML” opció bepipálásával (lásd következő ábrán).



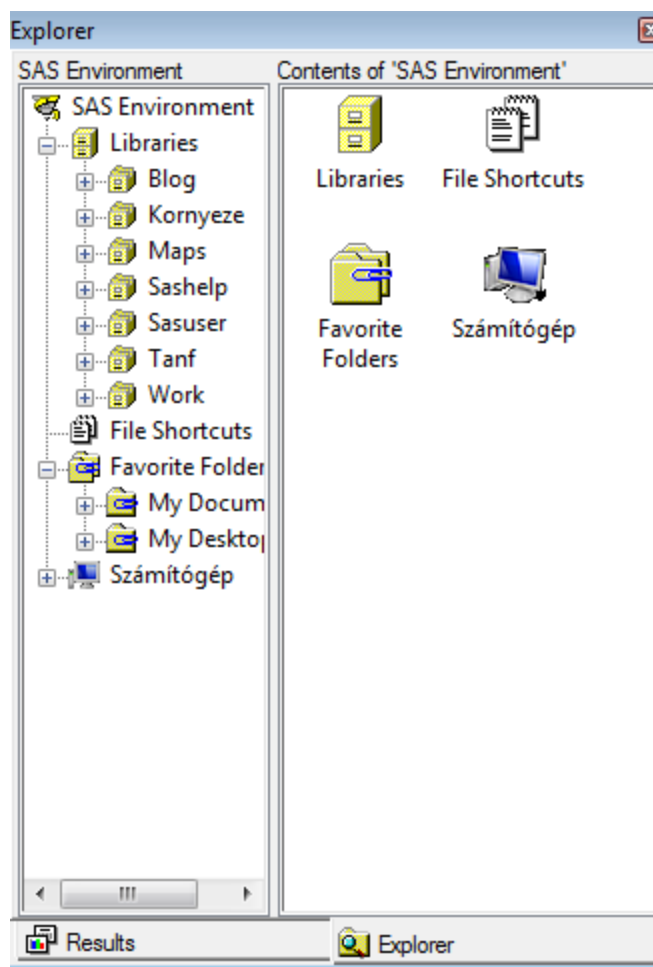
14. ábra: HTML megjelenítés beállítása (*Tools\Options\Preferences*)

Ezzel a programok futtatásának eredménye egy ún. *Results Viewer* ablakban jelenik meg, HTML formátumban.

3.2.4 Explorer

Az Explorer ablak olyan, mint a Windows Explorer. Itt találjuk az adatállományainkat. Láthatjuk a számítógép meghajtóinak fájlstruktúráját, de a lényeg a SAS adatállományok láthatósága a SAS könyvtárban.

Lehetőség van a *View\ Show Tree* menüponttal kettős ablak-megjelenítésre, ami könnyebb átláthatóságot eredményez ebben az ablakban (a bal oldalon a mappaszerkezet jelenik meg, a jobb oldali részen pedig a bal oldalon kiválasztott elem tartalmát láthatjuk). Ezt szemlélteti a következő ábra:



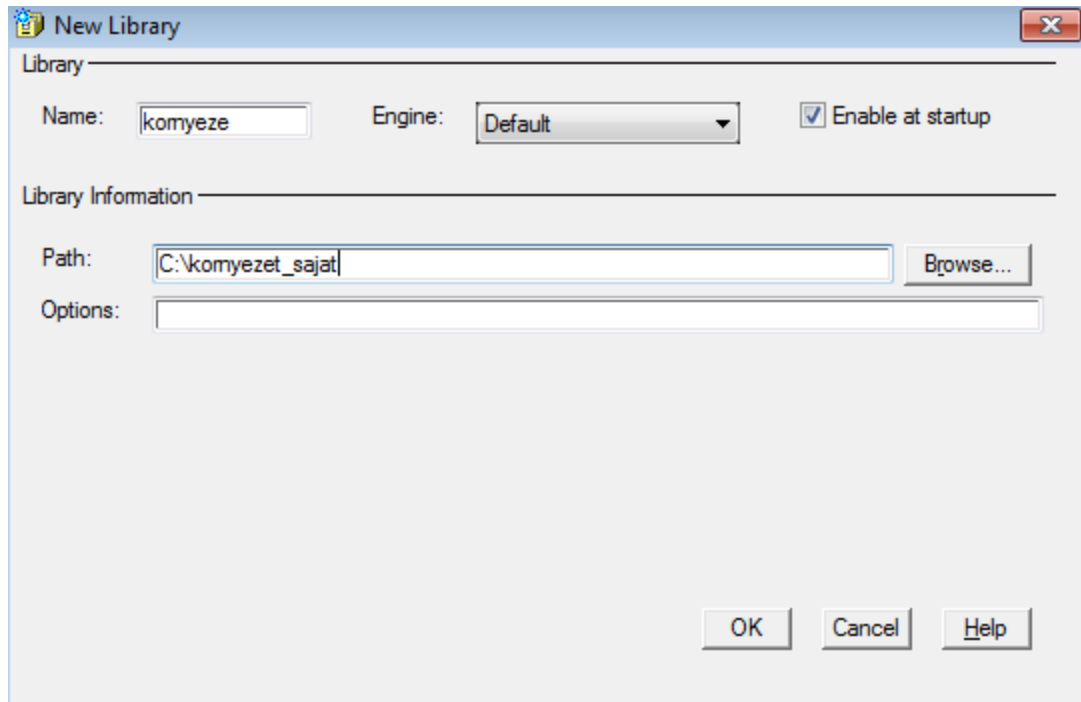
15. ábra: Explorer ablak két részes megjelenítésben

Itt jól látható, hogy a SAS környezet (*SAS Environment*) legelső (és legtöbbet használt) része a könyvtárak (*Libraries*). Alapértelmezetten 4 könyvtár van a SAS-ban: *Maps*, *Sashelp*, *Sasuser* és a *Work*. A *Maps*-ben olyan adatállományok vannak, melyekben országok térkép geo-koordinátái szerepelnek. A *Sashelp* és a *Sasuser* előre létrehozott minta adatállományokat tartalmaz. A *Work* az ideiglenes tárolója a SAS-nak (munkakönyvtár). Ha nem adunk meg könyvtárnevet, ahova mentsen, akkor alapértelmezetten ide ment. Ennek tartalma viszont a SAS program bezárásával törlődik. Erre figyeljünk! (Ha szükségünk van valamely adatállományra, ami csak a WORK-ben szerepel ideiglenesen, akkor a SAS program bezárása előtt mentjük el egy másik könyvtárba.)

(A képen látható többi könyvtárat már pluszban hoztuk létre, azokban dolgozunk különböző témák kapcsán.)

A Libraries főkönyvtáron jobb egérgombbal megjelenő menüben a New lenyomásával tudunk új könyvtárat létrehozni. (A  „New Library” ikon is ugyanezt a dialog ablakot éri el.) Mi jelen dokumentum fontosabb adatállományait a „kornyeze” nevű könyvtárba mentjük.

Az alábbi ablak jelenik meg, amikor saját könyvtárat hozunk létre:



16. ábra: Új könyvtár létrehozása SAS-ban

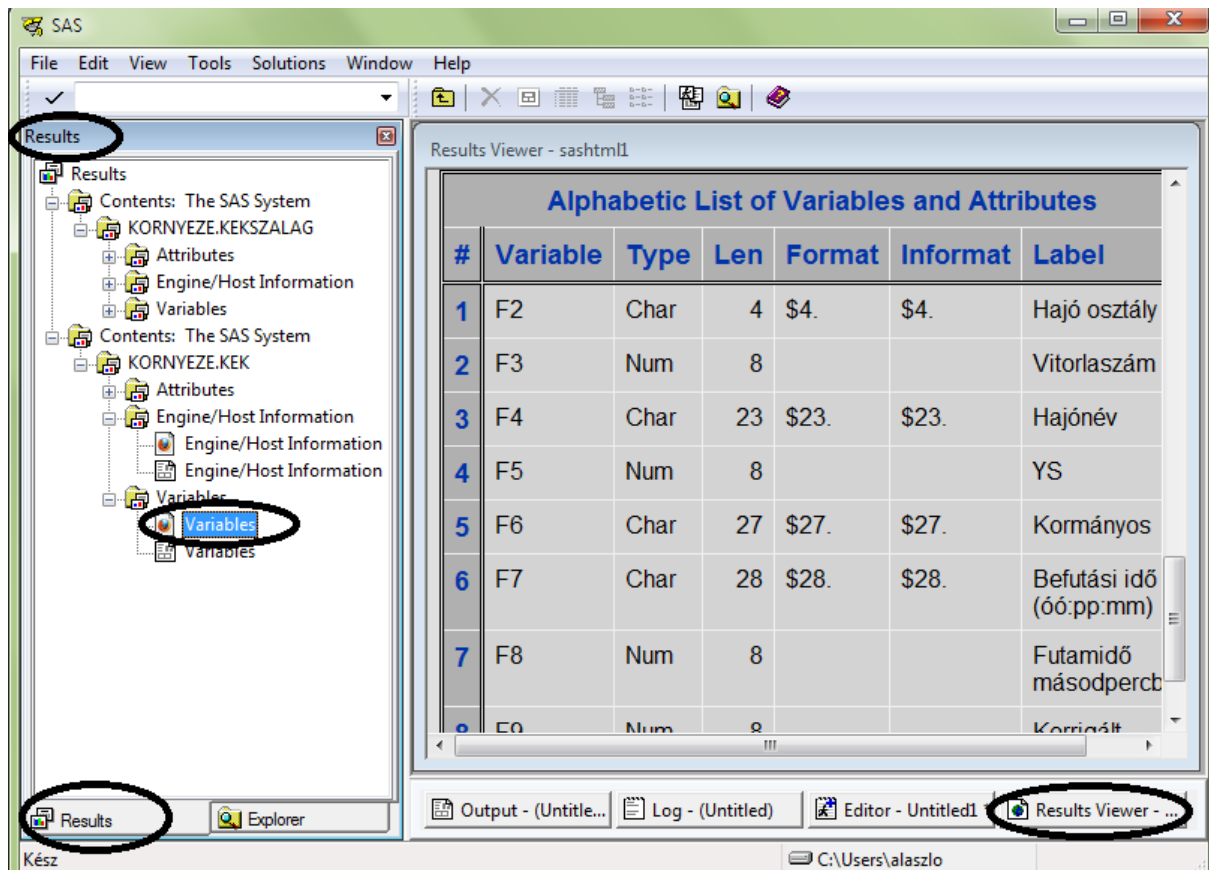
A *Name* sorába adjunk egy logikai nevet, amiként azt a SAS-ban a Libraries-ben látni fogjuk a könyvtárunkat. Az „*Enable at startup*” opció bepipálásával a legközelebbi alkalommal ugyanazon a gépen alapértelmezetten megjelenik ez a „*kornyeze*” könyvtár hivatkozás.

Ez a könyvtár egy logikai elérést biztosít a számítógépen, jelen esetben a C meghajtón lévő korábban létrehozott „*kornyezet_sajat*” nevű fizikai könyvtárba (ezt az elérési utat válasszuk ki a *Path* sorában a *Browse* gomb lenyomásával megjelenő fájlkereső ablakban). Vagyis a SAS-ban a „*kornyeze*” logikai névvel hivatkozunk az adatállományainkra, melyeket a SAS-ból kilépve is megtalálunk a *C:\kornyezet_sajat* könyvtárban. (Természetesen mindenki oda hoz létre könyvtárat és olyan névvel a számítógépen, ahol az szükséges.)

Az OK gomb lenyomásával jön létre az új könyvtár, és jelenik meg a „*kornyeze*” név a könyvtárak között. (A 2. ábrán azért látható már ez a könyvtár, mert korábban létrehoztuk, hisz ebben dolgoztunk a félév során.)

3.2.5 Results

A Results ablakban tudjuk a futtatásaink eredményeit visszakeresni, és megtekinteni.

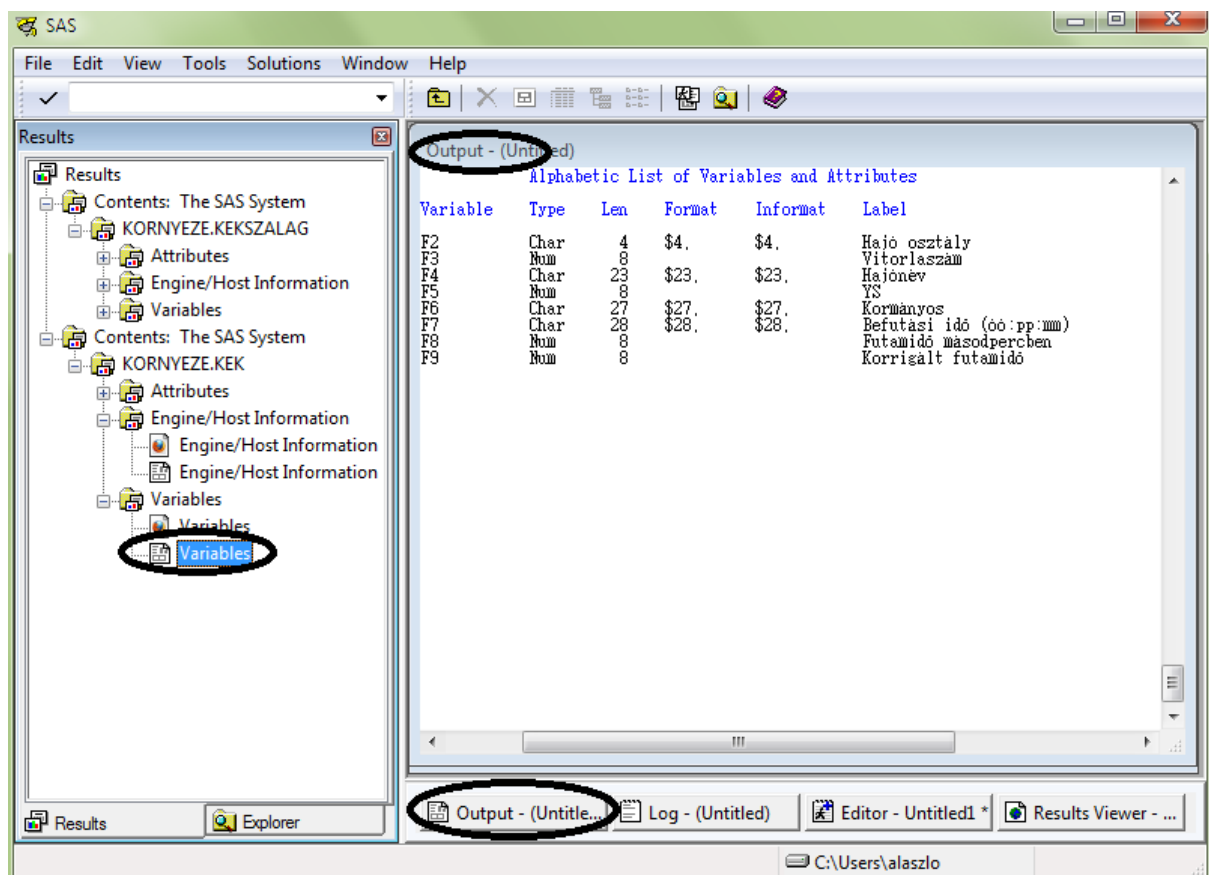


17. ábra: Results ablak

A fenti képen a bal oldali részen láthatjuk a *Results* ablakot (az Explorer melletti munkalap fülre kattintva érjük el, illetve a *View* menüben is megjeleníthetjük, ha véletlen bezártuk ezt az ablakot).

A jobb oldali részen, a *Results Viewer* ablakban jeleníti meg a *kornyeze* könyvtárban található *kek* nevű adatállományon futtatott *contents* eljárás változóit (*Variables*) HTML formátumban (mert korábban beállítottuk a HTML output megjelenítést, ezért ez a *Results Viewer*-ben jelenik meg).

Ha nem a bal oldali *Results* ablakban bekarikázott *Variables* HTML eredményt választjuk ki, hanem az alatta lévő sort, akkor karakteresen jeleníti meg az *Output* ablakban ez eredményt. Ezt szemlélteti a következő ábra:



18. ábra: Listing output karakteres megjelenítése

3.3 ADATIMPORT

Jelen fejezetben egy minden évben megrendezett, balatoni vitorlásverseny, a Kékszalag verseny Excel adatállományát importáljuk be a SAS környezetbe.

Ehhez ismerjük meg a kékszalag adatállományunkat, melyet az első mellékletben találhatunk meg.

Microsoft Excel - kekszalag2008_ere_jav

Fájl Szerkesztés Nézet Beszúrás Formátum Eszközök Adatok Ablak Súgó

K535 100% Times New Roman 12

	A	B	C	D	E	F	G	H	I	J	K	L
1	T-Mobile Nagydíj 40. Kékszalag											
2	Végeredmény											
3	Rajt ideje: 18-án 9.00 ora											
4				Hajónév	YS	Kormányos	Befutási idő (óó.pp.ss)	Futott idő (sec)	Korrigált idő (sec)			
5	1.	YS1	2002	Brokernet- Uniqua	63	Litkey Farkas	22.13.40	47 620	75 587			
6	2.	YS1	92	Raffica	63	Király Zsolt	22.29.22	48 562	77 083			
7	3.	YS1	91 GER	Telebox Due	63	Gerhard Müller	22.53.17	49 997	79 360			
8	4.	YS1	115	Principessa	63	Rauschenberger Miklós	22.53.36	50 016	79 390			
9	5.	YS1	18	Liberty Sailing Team	67	Újhelyi Gáspár Miklós	00.36.27	56 187	83 861			
10	6.	YS1	2	AC Sailing Team	63	Láng Róbert SVE	00.42.57	56 577	89 805			
11	7.	YS1	1212	Mediacontact- Pioneer	67	Soponyai Géza KMP	00.56.42	57 402	85 675			
12	8.	YS1	1153	Sponzor Wanted	67	Vándor Róbert	00.58.44	57 524	85 857			
13	9.	YS1	42	Raiffeisen	71	Pfeninberger András	01.07.27	58 047	81 756			
14	10.	70	6	Anna	78	Csőregh Zoltán	01.27.35	59 255	75 968			
15	11.	70	1	Orpheus	78	Herkó Dezső	01.28.14	59 294	76 018			
16	12.	Ö	75/2	Sirocco	81	Cittel Lajos	01.43.32	60 212	74 336			
17	13.	70	7	Irókéz	78	Litkey Bence KMP	01.49.22	60 562	77 644			

Ábszolút / Hárdoztálvonnként /

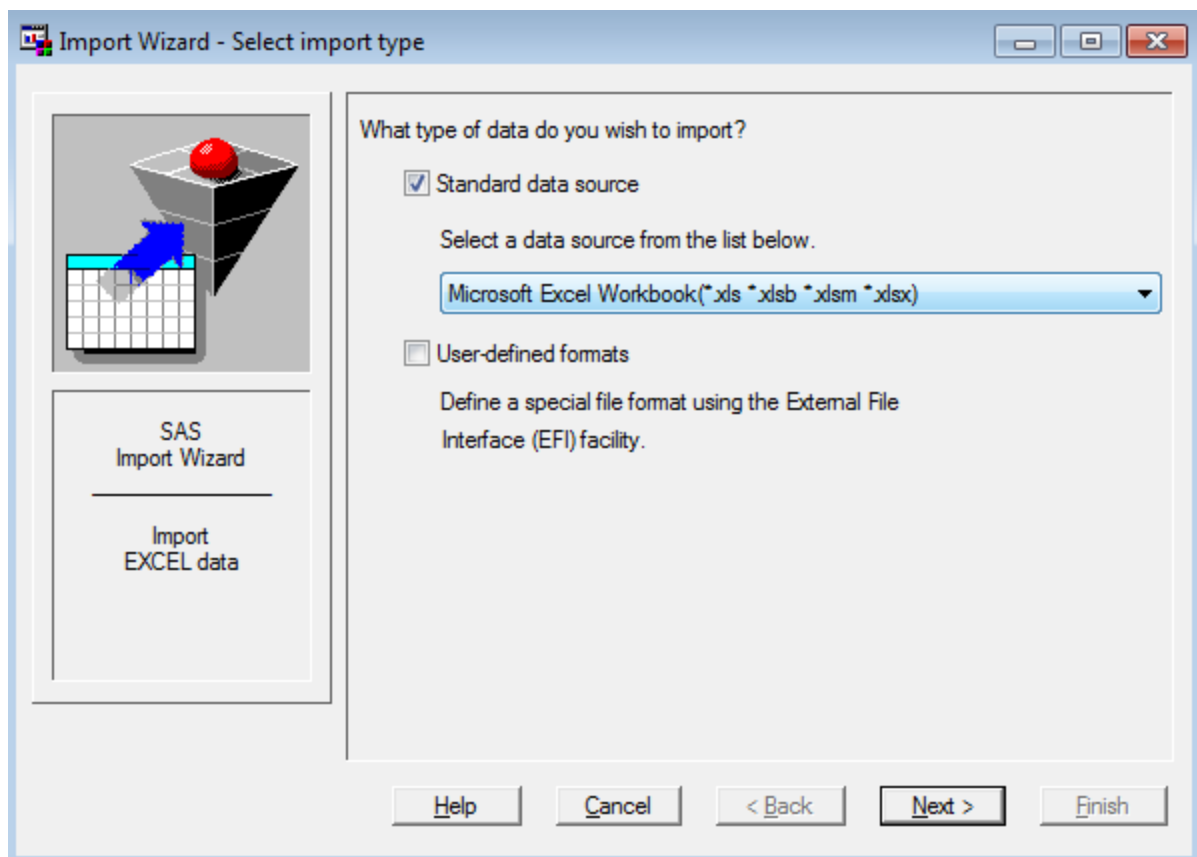
19. ábra: Kékszalag Excel adatállomány részlet

Kétféleképpen járhatunk el. Vagy az Excelben formázzuk úgy az adatokat, hogy SAS-ba beimportálva könnyen kezelhető legyen, vagy úgy, ahogy van beimportáljuk SAS-ba, és ott formázzuk igény szerint. A gyakorlat során alakul ki igazán, hogy melyik célravezetőbb, illetve a kettőt lehet vegyítve is alkalmazni.

A SAS-ban egy-egy adatállomány táblázatos szerkezetű, ahol a sorokban a megfigyeléseink vannak, az oszlopokban pedig a változóink. Minden üres cellát hiányzó értéknek érzékel a SAS, ezért arra figyelniünk kell, hogy a valóságnak megfelelően olvassuk be az adatokat a SAS-ba. Figyeljünk arra is, hogy az Excelben is vannak automatikus formátumok, melyek bezavarhatnak (pl. egy számot van, hogy dátum formátumban érzékel, és utána SAS-ba beolvasva azt már nem az eredeti adattal számolunk)!

Mielőtt elkezdenénk beolvasni az Excelől az adatokat SAS-ba, ellenőrizzük le, hogy ne legyen kinyitva a beolvasandó Excel fájl.

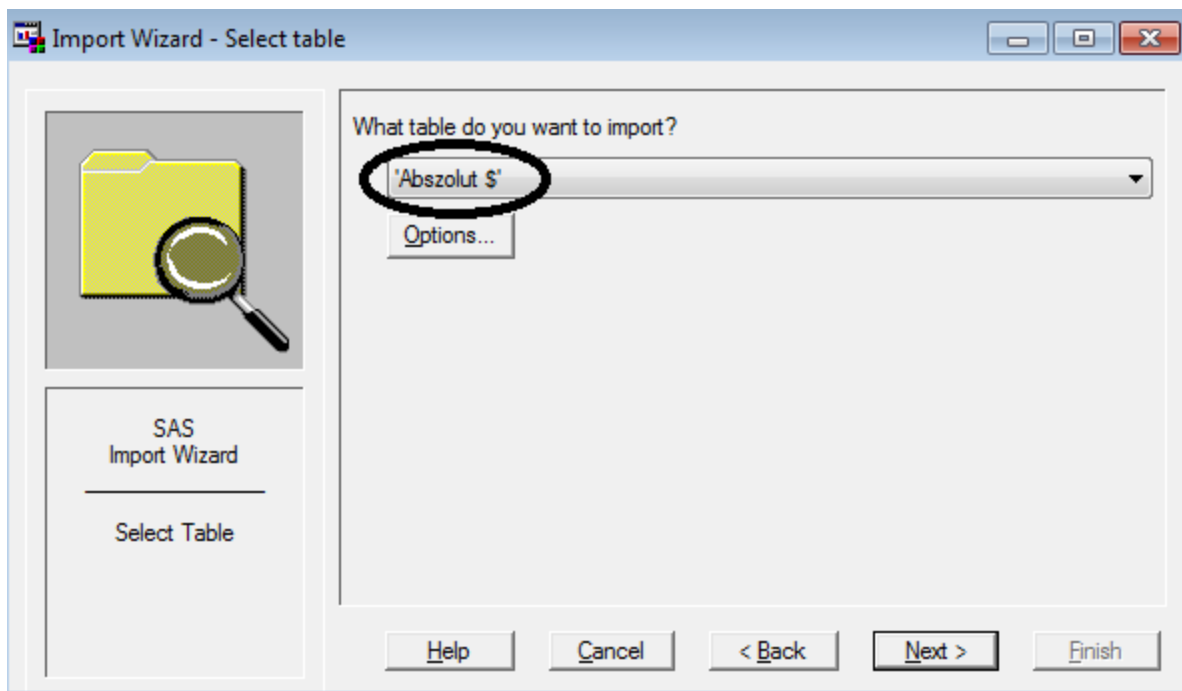
A *File\Import Data* menüpontban válasszuk először a *Microsoft Excel* inputot, ahogy azt az alábbi ábra mutatja és menjünk tovább (*Next*).



20. ábra: Import varázsló 1. lépés: beolvasandó file típusának kiválasztása

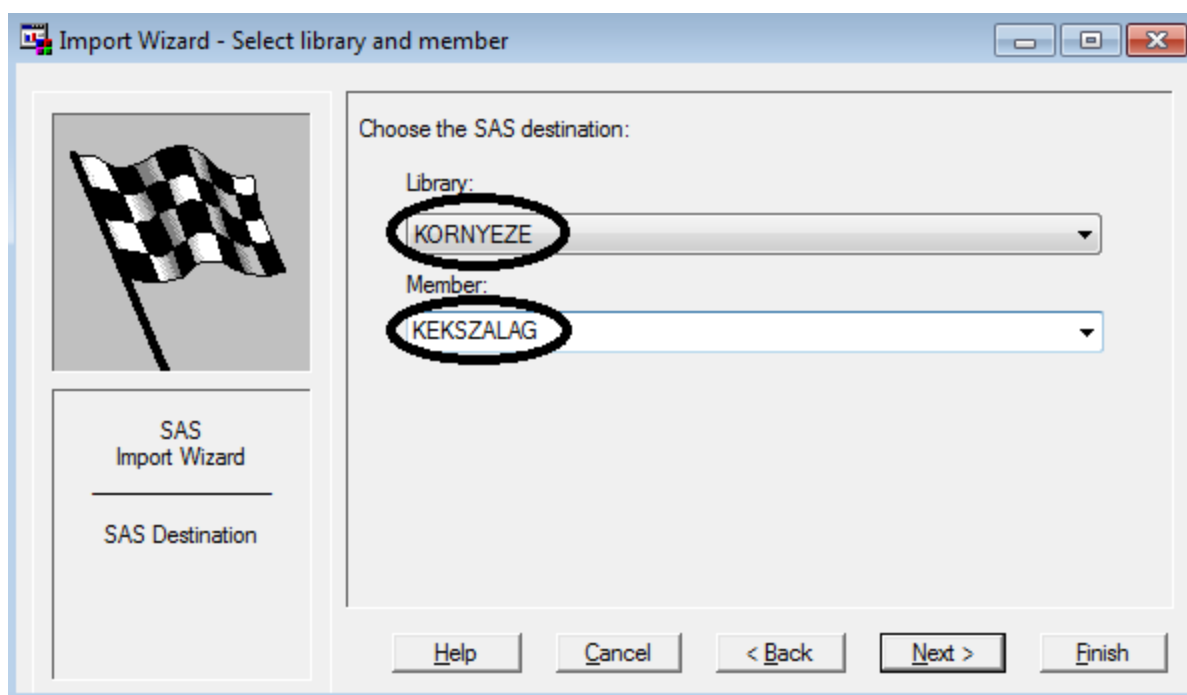
A megjelenő „*Connect to MS Excel*” ablakban a *Browse* lenyomásával keressük ki a *kekszalag2008.xls* Excel állományunkat (lásd első melléklet), majd nyomjuk le az *OK* gombot.

A következő lépésben válasszuk ki azt a munkalapot az Excel dokumentumból, aminek a tartalmát be kívánjuk olvasni. Jelen esetben ez most az első, az Abszolút munkalap. Alapértelmezetten az elsőt kínálja fel.



21. ábra: Import varázsló 2. lépés: a kiválasztott Excel fájl munkalapjának megjelölése, ahonnan az adatokat szeretnénk beolvasni a SAS-ba

A *Next* lenyomásával az utolsó lépéshez jutunk:

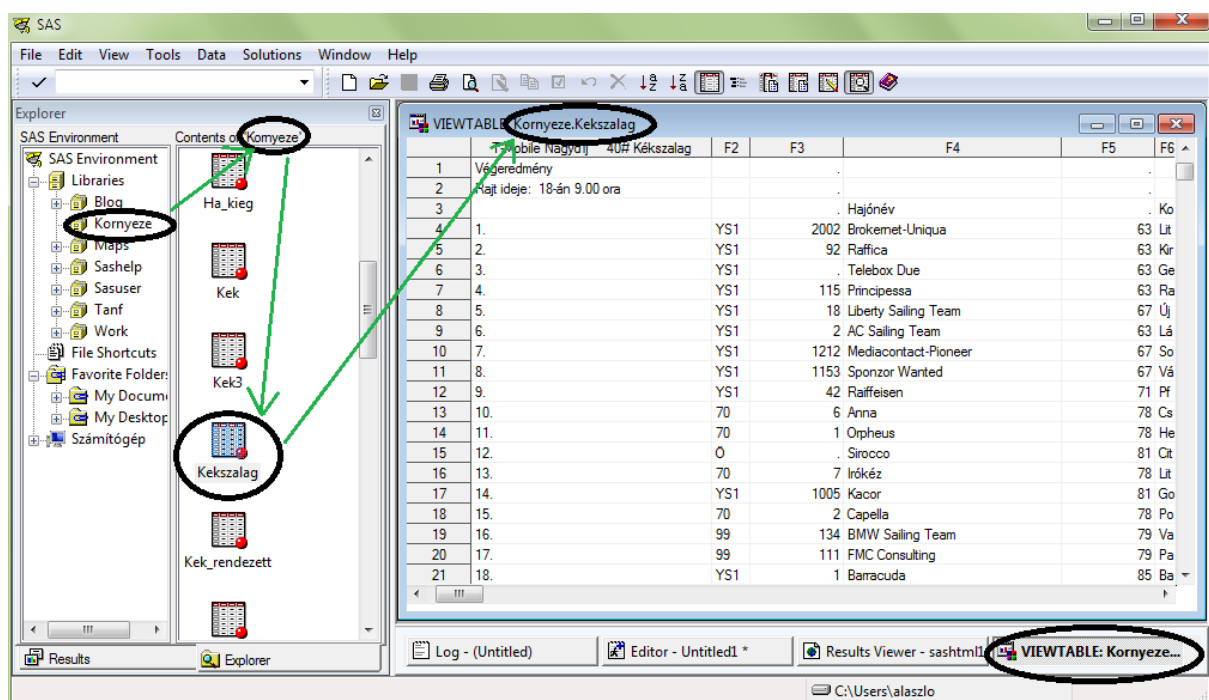


22. ábra: Import varázsló 3. lépés: könyvtár és adatállomány nevének megadása

A harmadik lépésben válasszuk ki, hogy melyik könyvtárba, milyen néven szeretnénk elmenteni az éppen beolvasott SAS adatállományt. Jelen esetben ez a *kornyeze* könyvtár *kekszalag* adatállománya lesz, ahogy azt a fenti ábra mutatja.

Itt már nyomhatunk „Finish”-t (befejezés), mert a főbb lépések végére értünk. (Ha a Next-et nyomjuk, akkor lehetőségünk van a beimportálás SAS forráskódjának kimentésére.)

Az importálás befejezésével létrejön a „kornyeze” nevű könyvtárunkban egy „kekszalag” nevű SAS adatállomány, melyet az *Explorer*-ben is megtalálunk:



23. ábra: Excelből beimportált Kekszalag SAS adatállomány a "kornyeze" könyvtárban

A bal oldali részén az *Explorer*-nek a „kornyeze” könyvtárra duplán kattintva, a jobb oldali részben megjelenik a „kornyeze” könyvtár tartalma (*Contents of 'Kornyeze'*). Ebben láthatjuk a SAS adatállományok sorát, így a beimportált „kekszalag” nevűt is. Láthatjuk, hogy a SAS adatállományokat általában egy  ikon jelöli. Erre duplán kattintva a jobb oldali részén a SAS programnak megnyílik egy *Viewtable*-ben az adatállomány táblázatos formája.

Itt átfuthatjuk, hogy valóban azt importáltuk-e be, amit szerettünk volna.

Ugyanez az adatállomány, mivel saját könyvtárba mentettük, nem pedig ideiglenesen a WORK könyvtárba, ezért a „kornyeze” könyvtárnak beállított fizikai elérési úton konkrétan is megtalálhatjuk a *kekszalag.sas7bdat* fájlt. Ez az a fájl, amit a SAS rendszerben az imént a Viewtable-ben megtekintettünk.

A SAS adatállományok kiterjesztése tehát *.sas7bdat*, egyfajta szöveges állományok ezek. A SAS programokat is kimenthetjük, ezek kiterjesztése *.sas*.

4. KÉKSZALAG SAS ADATÁLLOMÁNY MEGISMERÉSE, ADATMANIPULÁCIÓ

Az előzőleg beimportált Kékszalag adatállományt fogjuk megismerni és kicsit alakítani igény szerint. Ehhez nézzük sorban az alapvető SAS kódokat, amiket az *Editor* ablakba írhatunk be, és a menüsoron a *Submit* ikon lenyomásával futtathatunk.

Ahogy azt már az előző fejezetben említettük, a kódot elláthatjuk **kommentekkel** (lásd a következő programban a zöld részeket), melyeket */** és **/* jelek közé kell tenni, hogy a SAS annak is értelmezze. Ide akár ékezeteket is írhatunk, ebben a részben magyarázhatjuk a programunkat. Mi is ezt fogjuk alkalmazni leginkább a dokumentumban.

A program egyéb részében kerüljük az ékezetes betűket, szóközöket és speciális karaktereket a változók elnevezésében és utasítások használatakor (kivéve a szöveges részeket az aposztróf jelek vagy idézőjelek között, illetve egyéb indokolt eseteket).

Alapvetően a SAS-ban két lépést, főbb utasítást használunk: ezek a DATA és a PROC step-ek. A **DATA lépésben** adatállományokat hozunk létre, olvasunk be, kezelünk és/vagy módosítunk.

A **PROC lépésben** a SAS-ba beépített eljárások (PROCedure angol szóból) közül hívunk meg, mely a DATA lépésben létrehozott SAS adatállományokon végez különféle műveleteket (pl. kiírás, rendezés, különféle statisztikák számítása, tartalmak megjelenítése, stb.).

Általános szintaktikájuk (a parancsok „nyelvi” helyessége) a következő:

```
DATA konyvtarnev.SAS_adatallomany ;  
    UTASITAS1 ; /* opcionális */  
    UTASITAS2 ; /* opcionális */  
    ...  
RUN ;
```

```
PROC konyvtarnev.SAS_adatallomany ;  
    UTASITAS1 ; /* opcionális */  
    UTASITAS2 ; /* opcionális */  
    ...  
RUN ;
```

A SAS utasításokat mindig kulcsszóval kezdjük és **pontosvesszővel zárjuk**. Az utasításokban lévő szavakat szóközzel tagoljuk.

A következőkben a *kekszalag* nevű adatállomány megismerésén keresztül gyakoroljuk a SAS programozás alapjait.

```
/* Megismerjük a "kornyeze" könyvtárunk "kekszalag" nevű adatállományát */  
proc contents data=kornyeze.kekszalag;  
run;
```

Ez tehát egy PROC lépés volt, amiben a CONTENTS nevű eljárást hívtuk meg a „kornyeze” könyvtárban található *kekszalag* nevű adatállományunkra. Ahogy azt az előbb az általános szintaktikában láhattuk, mind a DATA, mind a PROC lépést RUN; utasítással zárjuk le. A CONTENTS eljárás eredménye az alábbi:

LOG ablakban:

```
1 /* Megismerjük a "kornyeze" könyvtárunk "kekszalag" nevű adatállományát */  
2 proc contents data=kornyeze.kekszalag;  
NOTE: Writing HTML Body file: sashtml.htm  
3 run;
```

```
NOTE: PROCEDURE CONTENTS used (Total process time):  
      real time           2.34 seconds  
      cpu time            0.71 seconds
```

A LOG-ból láthatjuk, hogy „csak kék” NOTE-okat kaptunk, vagyis leegyszerűsítve szintaktikailag (nyelvtanilag) helyes volt a beírt parancssorunk.

Mivel korábban az alapértelmezett listing output mellett beállítottuk a HTML kimenetet is (Tools \ Options \ Preferences \ Results \ Create HTML), ezért két outputot kaptunk:

Output ablakban:

```
1  
  
The SAS System 09:52 Wednesday, May 18, 2011  
  
The CONTENTS Procedure  
  
Data Set Name      KORNYEZE.KEKSZALAG      Observations  
1392
```

66	Member Type	DATA	Variables	
	Engine	V9	Indexes	0
	Created	2011. február 14. hétfő 17:12:38	Observation Length	
216	Last Modified	2011. február 14. hétfő 17:12:38	Deleted Observations	0
	Protection		Compressed	
NO	Data Set Type		Sorted	
NO	Label			
	Data Representation	WINDOWS_32		
	Encoding	wlatin2 Central Europe (Windows)		

Engine/Host Dependent Information

Data Set Page Size	16384
Number of Data Set Pages	20
First Data Page	1
Max Obs per Page	75
Obs in First Data Page	39
Number of Data Set Repairs	0
Filename	C:\Anna\SAS\kornyezettan\2011\kekszalag.sas7bdat
Release Created	9.0202M3
Host Created	W32_VSPRO

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat	Label
2	F2	Char	4	\$4.	\$4.	F2
3	F3	Num	8			F3
4	F4	Char	23	\$23.	\$23.	F4
5	F5	Num	8			F5
6	F6	Char	27	\$27.	\$27.	F6
7	F7	Char	28	\$28.	\$28.	F7
8	F8	Num	8			F8
9	F9	Num	8			F9
10	F10	Char	1	\$1.	\$1.	F10
11	F11	Char	1	\$1.	\$1.	F11
12	F12	Char	1	\$1.	\$1.	F12
13	F13	Char	12	\$12.	\$12.	F13
14	F14	Char	1	\$1.	\$1.	F14
15	F15	Char	1	\$1.	\$1.	F15
16	F16	Char	1	\$1.	\$1.	F16
17	F17	Char	1	\$1.	\$1.	F17
18	F18	Char	1	\$1.	\$1.	F18
19	F19	Char	1	\$1.	\$1.	F19
20	F20	Char	1	\$1.	\$1.	F20
21	F21	Char	1	\$1.	\$1.	F21
22	F22	Char	1	\$1.	\$1.	F22
23	F23	Char	1	\$1.	\$1.	F23
24	F24	Char	1	\$1.	\$1.	F24

25	F25	Char	1	\$1.	\$1.	F25
26	F26	Char	1	\$1.	\$1.	F26
27	F27	Char	1	\$1.	\$1.	F27
28	F28	Char	1	\$1.	\$1.	F28
29	F29	Char	1	\$1.	\$1.	F29
30	F30	Char	1	\$1.	\$1.	F30
31	F31	Char	1	\$1.	\$1.	F31
32	F32	Char	1	\$1.	\$1.	F32
33	F33	Char	1	\$1.	\$1.	F33
34	F34	Char	1	\$1.	\$1.	F34
35	F35	Char	1	\$1.	\$1.	F35
36	F36	Char	1	\$1.	\$1.	F36
37	F37	Char	1	\$1.	\$1.	F37
38	F38	Char	1	\$1.	\$1.	F38
39	F39	Char	1	\$1.	\$1.	F39
40	F40	Char	1	\$1.	\$1.	F40

The SAS System

09:52 Wednesday, May 18, 2011

2

The CONTENTS Procedure

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat	Label
41	F41	Char	1	\$1.	\$1.	F41
42	F42	Char	1	\$1.	\$1.	F42
43	F43	Char	1	\$1.	\$1.	F43
44	F44	Char	1	\$1.	\$1.	F44
45	F45	Char	1	\$1.	\$1.	F45
46	F46	Char	1	\$1.	\$1.	F46
47	F47	Char	1	\$1.	\$1.	F47
48	F48	Char	1	\$1.	\$1.	F48
49	F49	Char	1	\$1.	\$1.	F49
50	F50	Char	1	\$1.	\$1.	F50
51	F51	Char	1	\$1.	\$1.	F51
52	F52	Char	1	\$1.	\$1.	F52
53	F53	Char	1	\$1.	\$1.	F53
54	F54	Char	1	\$1.	\$1.	F54
55	F55	Char	1	\$1.	\$1.	F55
56	F56	Char	1	\$1.	\$1.	F56
57	F57	Char	1	\$1.	\$1.	F57
58	F58	Char	1	\$1.	\$1.	F58
59	F59	Char	1	\$1.	\$1.	F59
60	F60	Char	1	\$1.	\$1.	F60
61	F61	Char	1	\$1.	\$1.	F61
62	F62	Char	1	\$1.	\$1.	F62
63	F63	Char	1	\$1.	\$1.	F63
64	F64	Char	1	\$1.	\$1.	F64
65	F65	Char	1	\$1.	\$1.	F65
66	F66	Char	1	\$1.	\$1.	F66

1 T_Mobile_Nagydj_____40__K_kszal Char 27 \$27. \$27. T-Mobile Nagydi 40#
Kékszalag

Ebből kiolvashatunk néhány hasznos információt. Többek között jól látszik, hogy 1392 megfigyelésünk („Observations”), sorunk és 66 változónk („Variables”), oszlopunk van a „kornyeze” könyvtár „kekszalag” nevű adatállományában („Data Set Name”).

Információkat láthatunk a nyomtatási papírméretre vonatkozóan (eszerint is jelenítette meg nekünk 2 lapon az adatokat: „The SAS System” kezdetű sorok végén a számozásból látjuk: „1” és „2”), ill. különféle egyéb adatokat, de ezek nem igazán relevánsak. A lényegi információ még a változók felsorolásában van („Alphabetic List of Variables and Attributes”), melyből láthatjuk a változó nevét („Variable”), címkéjét („Label” ha van – jelen esetben nincs, de ebben lehetne ékezetes, értelmezhető elnevezést adni egy-egy változónak, amit nemsokára meg is teszünk), típusát („Type”) és hosszát („Len”).

Most megnézzük, hogy a HTML outputban ugyanezt az információt hogyan jeleníti meg a SAS, és a továbbiakban már csak ezt fogjuk használni:

Results Viewer ablakban:

The SAS System			
The CONTENTS Procedure			
Data Set Name	KORNYEZE.KEKSZALAG	Observations	1392
Member Type	DATA	Variables	66
Engine	V9	Indexes	0
Created	2011. február 14. hétfő 17:12:38	Observation Length	216
Last Modified	2011. február 14. hétfő 17:12:38	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data	WINDOWS_32		

Representation			
Encoding	wlatin2 Central Europe (Windows)		

Engine/Host Dependent Information	
Data Set Page Size	16384
Number of Data Set Pages	20
First Data Page	1
Max Obs per Page	75
Obs in First Data Page	39
Number of Data Set Repairs	0
Filename	C:\Anna\SAS\kornyezettan\2011\kekszalag.sas7bdat
Release Created	9.0202M3
Host Created	W32_VSPRO

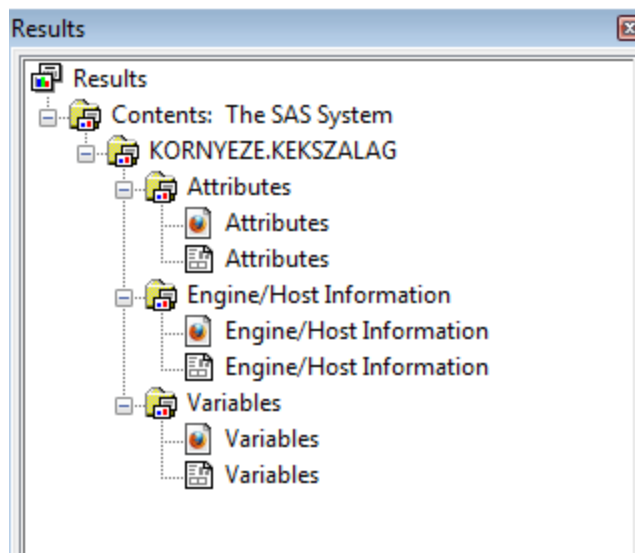
Alphabetic List of Variables and Attributes						
#	Variable	Type	Len	Format	Informat	Label
2	F2	Char	4	\$4.	\$4.	F2
3	F3	Num	8			F3
4	F4	Char	23	\$23.	\$23.	F4
5	F5	Num	8			F5
6	F6	Char	27	\$27.	\$27.	F6
7	F7	Char	28	\$28.	\$28.	F7
8	F8	Num	8			F8
9	F9	Num	8			F9

Alphabetic List of Variables and Attributes						
#	Variable	Type	Length	Format	Information	Label
10	F10	Char	1	\$1.	\$1.	F10
11	F11	Char	1	\$1.	\$1.	F11
12	F12	Char	1	\$1.	\$1.	F12
13	F13	Char	12	\$12.	\$12.	F13
14	F14	Char	1	\$1.	\$1.	F14
15	F15	Char	1	\$1.	\$1.	F15
16	F16	Char	1	\$1.	\$1.	F16
17	F17	Char	1	\$1.	\$1.	F17
18	F18	Char	1	\$1.	\$1.	F18
19	F19	Char	1	\$1.	\$1.	F19
20	F20	Char	1	\$1.	\$1.	F20
21	F21	Char	1	\$1.	\$1.	F21
22	F22	Char	1	\$1.	\$1.	F22
23	F23	Char	1	\$1.	\$1.	F23
24	F24	Char	1	\$1.	\$1.	F24
25	F25	Char	1	\$1.	\$1.	F25
26	F26	Char	1	\$1.	\$1.	F26
27	F27	Char	1	\$1.	\$1.	F27
28	F28	Char	1	\$1.	\$1.	F28
29	F29	Char	1	\$1.	\$1.	F29
30	F30	Char	1	\$1.	\$1.	F30
31	F31	Char	1	\$1.	\$1.	F31

Alphabetic List of Variables and Attributes						
#	Variable	Type	Length	Format	Information	Label
32	F32	Char	1	\$1.	\$1.	F32
33	F33	Char	1	\$1.	\$1.	F33
34	F34	Char	1	\$1.	\$1.	F34
35	F35	Char	1	\$1.	\$1.	F35
36	F36	Char	1	\$1.	\$1.	F36
37	F37	Char	1	\$1.	\$1.	F37
38	F38	Char	1	\$1.	\$1.	F38
39	F39	Char	1	\$1.	\$1.	F39
40	F40	Char	1	\$1.	\$1.	F40
41	F41	Char	1	\$1.	\$1.	F41
42	F42	Char	1	\$1.	\$1.	F42
43	F43	Char	1	\$1.	\$1.	F43
44	F44	Char	1	\$1.	\$1.	F44
45	F45	Char	1	\$1.	\$1.	F45
46	F46	Char	1	\$1.	\$1.	F46
47	F47	Char	1	\$1.	\$1.	F47
48	F48	Char	1	\$1.	\$1.	F48
49	F49	Char	1	\$1.	\$1.	F49
50	F50	Char	1	\$1.	\$1.	F50
51	F51	Char	1	\$1.	\$1.	F51
52	F52	Char	1	\$1.	\$1.	F52
53	F53	Char	1	\$1.	\$1.	F53

Alphabetic List of Variables and Attributes						
#	Variable	Type	Length	Format	Informat	Label
54	F54	Char	1	\$1.	\$1.	F54
55	F55	Char	1	\$1.	\$1.	F55
56	F56	Char	1	\$1.	\$1.	F56
57	F57	Char	1	\$1.	\$1.	F57
58	F58	Char	1	\$1.	\$1.	F58
59	F59	Char	1	\$1.	\$1.	F59
60	F60	Char	1	\$1.	\$1.	F60
61	F61	Char	1	\$1.	\$1.	F61
62	F62	Char	1	\$1.	\$1.	F62
63	F63	Char	1	\$1.	\$1.	F63
64	F64	Char	1	\$1.	\$1.	F64
65	F65	Char	1	\$1.	\$1.	F65
66	F66	Char	1	\$1.	\$1.	F66
1	T_Mobile_Nagydj____40__K_kszal	Char	27	\$27.	\$27.	T-Mobile Nagydj 40# Kék- szalag

Megjegyzés: természetesen az előbbi két output bekerült a Results ablakba is, ahol utólag is visszakereshető:



24. ábra: PROC CONTENTS eljárás eredménye a Results ablakban

Jól látjuk, hogy lefutott egy CONTENTS eljárás a KORNYEZE.KEKSZALAG adatállományon, melynek 3 táblázat az eredménye (ahogy azt láttuk az Output és Results Viewer ablakokban: „Attributes”, „Engine/Host Information”, „Variables”), melyek kétféle kimenetben jelentek meg: HTML és nyomtatási outputok.

A lényeg, amit láthatunk az outputokból (és ezt már akkor látjuk, ha dupla kattintással megnyitjuk magát az adatállományt a Viewtable-ben), hogy sok plusz változót és megfigyelést (oszlopot és sort) importált be a SAS Excelből, melyekben nincs használható információ, illetve vannak olyan sorok, amiben rossz, nem használható adatok szerepelnek. Ezekről tisztítjuk meg a következő lépésben az adatainkat. Nincs szükségünk ugyanis az adatelemzéshez az első változóra, melyben sorszámok szerepelnek, illetve az F10-es változótól felfelé egyik üres változóra sem.

A sorokat illetően az első három sor nem értelmes az elemzéshez, illetve az 529. sor után nincs használható értékadatunk, ezért ezeket is töröljük az elemzésre szánt adatállományunkból.

```
/* Létrehozunk egy "kek" nevű adatállományt a "kekszalag"-ból, és igény szer-
int alakítjuk */
data kornyeze.kek;
  set kornyeze.kekszalag (keep = f2-f9); /* csak az f2-f9 változókat tartjuk
meg */

  /* Változóknak címke adása */
  label f2 = 'Hajó osztály'
```

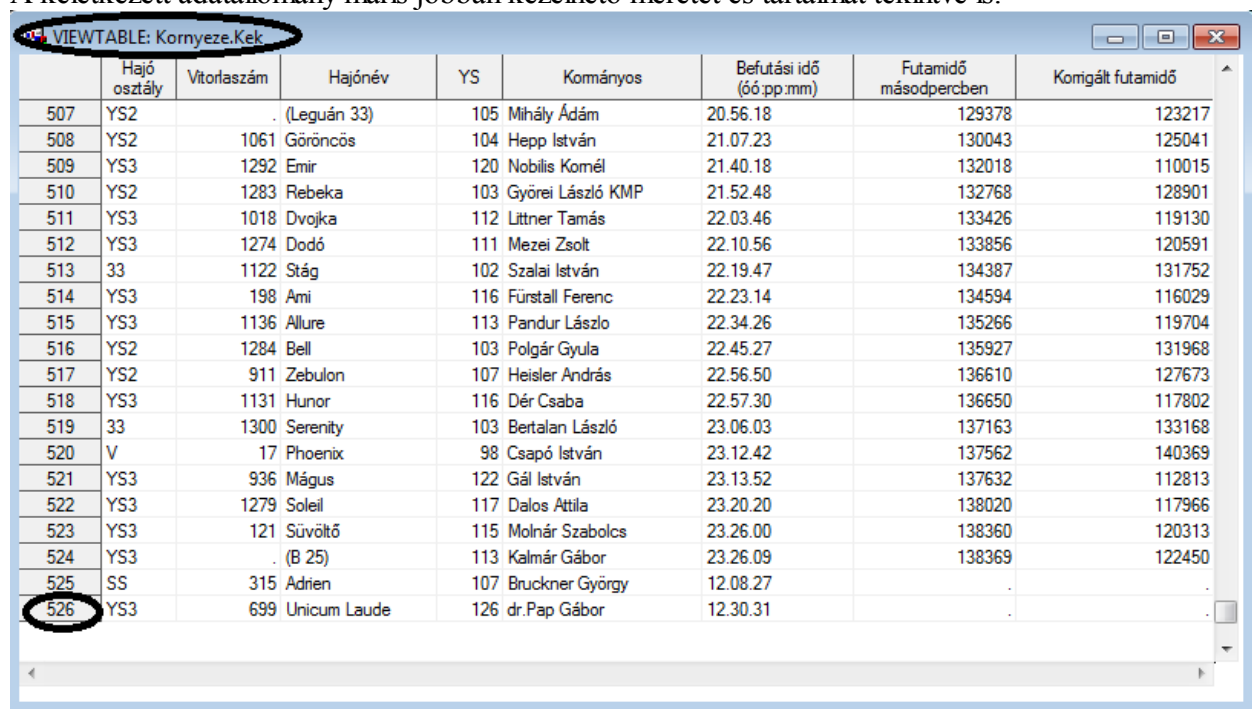
```

f3 = 'Vitorlaszám'
f4 = 'Hajónév'
f5 = 'YS'      /* YardStick szám - a hajó súlyozására használt szám */
f6 = 'Kormányos'
f7 = 'Befutási idő (óó:pp:mm)'
f8 = 'Futamidő másodpercben'
f9 = 'Korrigált futamidő';

/* Kitöröljük a fölösleges sorokat, és csak a 4 és 529 közötti sorokat tart-
juk meg az eredeti kekszalag adatállományból. _N_ a sorszám,
   ge = "greater than and equal to" : "nagyobb vagy egyenlő, mint"
   le = "less than and equal to" : "kisebb vagy egyenlő, mint" */
/* az utolsó két sort gyakorlásként hagyjuk csak meg,
   ez a két hajó nem ért célba - nincs helyezési száma a kekszalag első osz-
lopában */
if _N_ ge 4 and _N_ le 529;
run;

```

A keletkezett adatállomány máris jobban kezelhető méretét és tartalmát tekintve is:



	Hajó osztály	Vitorlaszám	Hajónév	YS	Kormányos	Befutási idő (óó:pp:mm)	Futamidő másodpercben	Korrigált futamidő
507	YS2	.	(Leguán 33)	105	Mihály Ádám	20.56.18	129378	123217
508	YS2	1061	Göröncös	104	Hepp István	21.07.23	130043	125041
509	YS3	1292	Emir	120	Nobilis Komél	21.40.18	132018	110015
510	YS2	1283	Rebeka	103	Györei László KMP	21.52.48	132768	128901
511	YS3	1018	Dvojka	112	Littner Tamás	22.03.46	133426	119130
512	YS3	1274	Dodó	111	Mezei Zsolt	22.10.56	133856	120591
513	33	1122	Stág	102	Szalai István	22.19.47	134387	131752
514	YS3	198	Ami	116	Fürstall Ferenc	22.23.14	134594	116029
515	YS3	1136	Allure	113	Pandur László	22.34.26	135266	119704
516	YS2	1284	Bell	103	Polgár Gyula	22.45.27	135927	131968
517	YS2	911	Zebulon	107	Heisler András	22.56.50	136610	127673
518	YS3	1131	Hunor	116	Dér Csaba	22.57.30	136650	117802
519	33	1300	Serenity	103	Bertalan László	23.06.03	137163	133168
520	V	17	Phoenix	98	Csapó István	23.12.42	137562	140369
521	YS3	936	Mágus	122	Gál István	23.13.52	137632	112813
522	YS3	1279	Soleil	117	Dalos Attila	23.20.20	138020	117966
523	YS3	121	Süvöltő	115	Molnár Szabolcs	23.26.00	138360	120313
524	YS3	.	(B 25)	113	Kalmár Gábor	23.26.09	138369	122450
525	SS	315	Adrien	107	Bruckner György	12.08.27	.	.
526	YS3	699	Unicum Laude	126	dr.Pap Gábor	12.30.31	.	.

25. ábra: Az elsőként megtisztított KORNYEZE.KEK adatállomány

Jól látható az ábrán, hogy 526 sorból és 8 oszlopból áll a táblázatunk (mátrix), vagyis 526 vitorlás versenyző szerepel az adatállományban, mely megfigyelésekre (egyedekre) 8 jellemzőt (változót) vizsgáltak.

Azt is tapasztalhatjuk, hogy a két utolsó sorban a futamidő és a korrigált futamidő változók hiányoznak. Ez a két versenyző nem ért célba, vagyis a végső létszámban nem szerepel, de példaként bent hagyjuk őket, és majd kalkulálunk értékeket a hiányzó adatok helyére.

Ha most ismét lefuttatjuk az előző CONTENTS eljárást, akkor kezelhetőbb képet kapunk:

```
proc contents data=kornyeze.kek;
run;
```

Ha most ismét lefuttatjuk az előző CONTENTS eljárást, akkor kezelhetőbb képet kapunk, ami alátámasztja az előbbi következtetéseinket:

The SAS System

The CONTENTS Procedure

Data Set Name	KORNYEZE.KEK	Observations	526
Member Type	DATA	Variables	8
Engine	V9	Indexes	0
Created	2011. május 18. szerda 11:34:39	Observation Length	120
Last Modified	2011. május 18. szerda 11:34:39	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_32		
Encoding	wlatin2 Central Europe (Windows)		

Engine/Host Dependent Information

Engine/Host Dependent Information	
Data Set Page Size	12288
Number of Data Set Pages	6
First Data Page	1
Max Obs per Page	102
Obs in First Data Page	86
Number of Data Set Repairs	0
Filename	C:\Anna\SAS\kornyezettan\2011\kek.sas7bdat
Release Created	9.0202M3
Host Created	W32_VSPRO

Alphabetic List of Variables and Attributes						
#	Variable	Type	Length	Format	Informat	Label
1	F2	Char	4	\$4.	\$4.	Hajó osztály
2	F3	Num	8			Vitorlaszám
3	F4	Char	23	\$23.	\$23.	Hajónév
4	F5	Num	8			YS
5	F6	Char	27	\$27.	\$27.	Kormányos
6	F7	Char	28	\$28.	\$28.	Befutási idő (óó:pp:mm)
7	F8	Num	8			Futamidő másodpercben
8	F9	Num	8			Korrigált futamidő

Az első táblázatból látjuk, hogy valóban 8 változónk és 526 megfigyelésünk van, míg az utolsó részletezi a már címkékkel ellátott megtartott 8 változót.

Általában az életben és a SAS-ban is két adattípust / változó típust különítünk el: kód- és értékadatokat / változókat. A **kódadatok** (más néven **nómenklatúrák**) lehetnek numerikusak is, de leginkább karakteresek, ilyenek például a nevek, csoportok elnevezései, melyeket nem használunk semmiféle számításra (nem vonjuk ki egymásból, nem képezünk arányokat), mert nincs értelmük! (Pl. a fenti példánál maradva nincs értelme átlagos *vitórlaszám*ról beszélni, pedig az egy numerikus érték, mégis a vitórlaszám jelentésénél fogva azonosításra szolgál, ezért értelmetlen, sőt súlyos alkalmazási hiba mondjuk azok átlagolása, összeadása, vagy bármely rajtuk végzett aritmetikai műveletvégzés. Ez a változó jelentését tekintve egyenértékű a *hajónév* vagy *kormányos* változókkal, hiszen azok is egyedi azonosításra szolgálnak. A *hajó osztály*, vagy a *yardstick szám* ugyanakkor már olyan kódváltozók, amelyek csoportosításra szolgálnak, az egyedeket csoportokba, kategóriákba sorolják. Van értelme beszélni a „J24”-es hajó osztályba tartozó versenyzőkről pl., vagy a 102-es yardstick számmal rendelkező hajókról. De figyelni kell arra, hogy ezeket csakis karakteresként értelmezzük.)

A másik típusú adatok az **értékadatok**, értékváltozók (más néven **mutatók**), melyek már numerikus adatok, és aritmetikai számításokat végezhetünk rajtuk. Van értelme azt mondani, hogy az egyik érték nagyobb a másikonál, hogy mennyivel nagyobb, illetve arányok felállítása is elképzelhető. A mi esetünkben például a másodpercben megadott *futamidő* ilyen változó, de, ha figyelembe vesszük a tartalmát, akkor ez csak a ténylegesen futott versenyzőket jelenti az egyes versenyzőkre nézve másodpercben. A *korrigált futamidőt*, mint értékadatot érdemesebb használni adatelemzésre, hiszen ebben már bele van kalkulálva, hogy melyik hajó gyorsabb, vagyis a *yardstick számmal* javított érték ez, miszerint összehasonlíthatóvá válnak az adatok.

A változóinkat és azok közötti tartalmi és egyéb összefüggéseket minden esetben meg kell ismernünk, hogy használható elemzéseket és azokból érvényes következtetéseket vonhassunk le.

A következőben megismerünk néhány alapvető statisztikát számító eljárást a SAS-ban:

```
/* Néhány statisztika beépített eljárásokkal (PROCedure).  
   A VAR opcióval lehet megadni, hogy mely változókra szeretnénk futtatni az  
   eljárást. */  
proc means data=kornyeze.kek;  
  var f5 f8 f9; /* Figyelünk arra, hogy mely változókra van értelme futtatni  
    az elemzést! Karakteres változóra nem is engedi a SAS ezt lefuttatni, és hi-  
    bát dob a LOG-ban. */  
run;
```

The MEANS Procedure

Variable	Label	N	Mean	Std Dev	Minimum	Maximum
F5	YS	526	98.7110266	10.4115390	63.0000000	126.0000000
F8	Futamidő másodpercben	524	95954.00	15092.57	47620.00	138369.00
F9	Korrigált futamidő	524	97027.41	9649.72	74336.00	140369.00

A MEANS eljárás az egyedszámokat, átlagot, szórást és legkisebb valamint legnagyobb értékeket számítja ki a megadott változókra. Elmondható tehát, hogy 524 versenyzőre van korrigált futamidő adatunk, melynek átlagos értéke 97027,41 másodperc. Ettől az átlagos korrigált futamidőtől átlagosan 9649,72 másodperccel térnek el a megfigyelt egyedek, a vitorlás versenyzők.

```
proc univariate data=kornyeze.kek;
  var f5 f8 f9;
run;
```

A fenti programsorban az UNIVARIATE eljárást 3 változóra futtattuk, de csak a korrigált futamidő (f9) jelentését nézzük meg. A többi változóra is ugyanilyen felépítésű táblázatokat kapunk. (Figyeljünk, hogyan értelmezzük, mert pl. a yardstick számnál, ami csoportosító kategorikus változó, bár numerikusként, számként van megadva, annak az átlagos értékét ne értelmezzük, mert nem lehet! Középértéknek ilyen esetben használjuk a mediánt, illetve vegyük a percentilis értékeket.)

Az UNIVARIATE a MEANS eljárásban megtalálható minden statisztikát kiszámít, de annál jóval több információt nyújt.

Ezekon felül a **skewness** például az eloszlás ferdeségére, míg a **kurtózis** a csúcsosságára hoz mutatószámot. A „kis” értékek nem támasztják alá, hogy a vizsgált sokaság eloszlása nem lenne normális. Ez persze sokban függ magától az adatoktól, és az egyedszámtól.

A **standard error**, a szórás (standard deviáció: SD) osztva a megfigyelések számának (N) gyökével. Ez, ha nem adják meg, hogy minek a standard hibája, akkor az átlagra vonatkozó érték, az átlag szórását jelenti (ha a populációból vennénk több ilyen mintát, versenyzők eredményeit, akkor azt mutatja

meg, hogy az egyes minták átlagai hogyan szóródnának: SE, vagy SEM: Standard Error of Mean). Figyelem: nem tévesztendő össze az SE az SD-vel, mert SD a minta szórása, míg SE az átlag szórása! A **variancia** a szórás értékének a négyzete, a szóródás mérőszáma. Ezeket és egyéb eredményeket különféle mutatószámok és statisztikák képleteiben szokták használni.

A **terjedelem** (range) a minimum és a maximum elem különbsége, most 66033 (ennyi másodperc telt el a leggyorsabb és a leglassabb verseny hajó futamideje között már a yardstick számmal korrigálva). Az **interkvartilis terjedelem** (interquartile range) az alsó és felső kvartilis (a sorbarendezett adatok 25%-ánál és 75%-ánál lévő adatok) különbsége, ami most 10577 másodperc. Azaz a középső 50%-a az adatoknak 10577 terjedelmű intervallumba esik.

A **percentilisek** (századoló pontok) közül találjuk a fontosabbakat a kvantilisek (Quantiles) táblázatban. Pl. a felső 5%-a az adatoknak (a szélén) 114696 feletti érték.

Az utolsó előtti táblázat a legkisebb és legnagyobb 5-5 értéket tünteti fel, ezek a **kiugró értékek** a korrigált futamidőre másodpercben, vagyis a leggyorsabb és a leglassabb 5-5 hajó. (Ezekről bővebben lásd az első függelék.)

Az utolsó táblázat a hiányzó értékek számát (SAS-ban a hiányzó érték jele '.' - pont) tünteti fel. A mi esetünkben 2 hiányzó érték van, amit már korábban láthattunk is, a két utolsó sorban nem volt számítva futamidő és korrigált futamidő, mert a két versenyző nem fejezte be a versenyt, csak kalkuláció miatt hagytuk benne az adatokban (amúgy, ha csak azok között vizsgálódunk, akik végigmentek szabályosan a pályán, és célba értek, akkor ezt a két sort ki is kéne törölni, nem lenne szabad bevenni az adatelemzésbe).

The SAS System

The UNVARIATE Procedure Variable: F9 (Korrigált futamidő)

Moments			
N	524	Sum Weights	524
Mean	97027.4084	Sum Observations	50842362
Std Deviation	9649.71935	Variance	93117083.5
Skewness	0.82612272	Kurtosis	2.26193969

Moments			
Uncorrected SS	4.9818E12	Corrected SS	4.87002E10
Coeff Variation	9.94535411	Std Error Mean	421.549947

Basic Statistical Measures			
Location		Variability	
Mean	97027.41	Std Deviation	9650
Median	96142.00	Variance	93117083
Mode	85042.00	Range	66033
		Interquartile Range	10577

Note: The mode displayed is the smallest of 9 modes with a count of 2.

Tests for Location: $\mu_0=0$				
Test	Statistic		p Value	
Student's t	t	230.1682	Pr > t	<.0001
Sign	M	262	Pr >= M	<.0001
Signed Rank	S	68775	Pr >= S	<.0001

Quantiles (Definition 5)	
Quantile	Estimate
100% Max	140369
99%	128901
95%	114696
90%	107657
75% Q3	101793
50%	96142

Quantiles (Definition 5)	
Quantile	Estimate
Median	
25% Q1	91216
10%	85932
5%	81525
1%	76018
0% Min	74336

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
74336	12	131752	513
74411	18	131968	516
75337	14	133168	519
75587	1	134860	505
75968	10	140369	520

Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	2	0.38	100.00

```
/* A SORT eljárással sorba rendezzük az f2 változó értékeit,
   hogy utána a FREQ (frequency - gyakoriság) eljárást le tudjuk futtatni.
   Vigyázzunk, mert a FREQ-t azokra a változókra érdemes csak futtatni, ahol
   a legtöbb érték nem különböző!
   Egy névsor változóra például értelmetlen, és több ezres adat esetén
```

```

nem jó! */
proc sort data=kornyeze.kek;
  by f2;
run;
proc freq data=kornyeze.kek;
  by f2;
run;

```

Mivel az f2 változó a hajóosztály, ezért a FREQ eljárás eredménye hajóosztályonként kiszámol minden változó gyakoriságát. (Ha több változó szerint szeretnénk gyakoriságokat számítani, akkor mindegyik szerint rendezni is kell az adatállományt.)

Az output rengeteg táblázatot tartalmaz, ezek közül választottunk egyet:

The SAS System

The FREQ Procedure Hajó osztály=C

...

YS				
F5	Frequency	Percent	Cumulative Frequency	Cumulative Percent
101	2	28.57	2	28.57
108	5	71.43	7	100.00

Ez azt mutatja meg, hogy a C hajóosztályban kétféle yardstickszám fordul elő: a 101-es, amiből 2 db volt, illetve a 108-as, amiből 5 db hajó szerepelt a versenyben. Összesen tehát (kumuláltan, ha összeadjuk) 7 hajó volt a C hajóosztályból, melyek 28,57%-a 101-es YS számmal rendelkező, és a többi 108-as YS számú.

Folytonos változók közötti lineáris kapcsolat szorosságot számíthatunk a korrelációval. Erre szolgál a SAS CORR nevű eljárása. Erre nézünk most példát:

```

/* Korreláció számítás - van-e kapcsolat az f8, f9 folytonos változók között,
és, ha igen milyen erősségű és irányú? */

```

```
proc corr data=kornyeze.kek;
  var f8 f9;
run;
```

The SAS System

The CORR Procedure

2 Variables:

F8 F9

Simple Statistics

Variable	N	Mean	Std Dev	Sum	Minimum	Maximum	Label
F8	524	95954	15093	50279897	47620	138369	Futamidő másodpercben
F9	524	97027	9650	50842362	74336	140369	Korrigált futamidő

Pearson Correlation Coefficients, N = 524 Prob > |r| under H0: Rho=0

	F8	F9
F8 Futamidő másodpercben	1.00000	0.78272 <.0001
F9 Korrigált futamidő	0.78272 <.0001	1.00000

A „Simple Statistics” táblázat a már a MEANS és UNIVARIATE eljárásokban látott alap statisztikákat mutat be a két változóra nézve. A Pearson korrelációs együttható értékét (Rho, vagy r) és a hozzá tartozó p-értéket mutatja az utolsó táblázat.

Az r egy -1 és 1 közötti szám, ami, ha -1 vagy 1-hez közeli értéket vesz fel, akkor erős függvényszerű kapcsolatot mutat a két változó között. Negatív szám esetén negatív irányú (ha az egyik értéke nő, a

másiké csökken), míg pozitív esetén pozitív irányú a kapcsolat (ha az egyik értéke növekszik, az a másik változó növekedését vonja magával).

Ha nullához közelít, akkor nem tudjuk elvetni azt a hipotézist, hogy a két vizsgált változó között nem a véletlen eredménye a kapott p-érték. A most kapott 0.78-as érték, erős pozitív kapcsolatot mutat a két változó között, amit a $p < 0.0001$ igen kicsi p érték alá is támaszt (mivel p kisebb 0,05, ezért azt mondhatjuk, hogy 5%-os szinten szignifikáns az eredményünk, vagyis elutasítjuk a nullhipotézist, miszerint nem mutatható ki kapcsolat a két változó között). Erre az eredményre számítottunk is, hiszen tudjuk, hogy a korrigált futamidőt a futamidőből számítottuk, tehát nyilván lesz a két változó között valamiféle függvényszerű kapcsolat.

5. VÁLTOZÓK KÉPZÉSE, ADATOK RENDEZÉSE, CSOPORTOK SZERINTI KÜLÖN ADATÁLLOMÁNYOK LÉTREHOZÁSA ÉS GRAFIKUS MEGJELENÍTÉSE

```
/* Harmadik gyakorlat, ahol az előző órán használt "kek" nevű adatállományon  
dolgozunk tovább. */  
data kornyeze.kek3 (keep=f2-f9); /* A keep miatt az ora, perc, masodperc vál-  
tozókat nem tartalmazza az adatállomány */  
set kornyeze.kek;  
  
/* Hiányzó értékeknek értékadás - hiányzó érték a SAS-ban: "." */  
if f8 = . then  
  do;  
    /* A LENGTH-szel egyenként numerikusnak definiáljuk az idő elemeit  
    és az ezekből számított másodperceket összeadjuk */  
    length ora perc masodperc 3.;  
    /* substr() függvény: első paramétere a karakteres változó, aminek a máso-  
    dik paraméterben megadott karakterétől a harmadik paraméterben megadott  
    számú karaktert olvas ki */  
    ora = substr(f7, 1, 2); /* f7 változó értékének az első karaktertől számí-  
    tott 2 hosszú karaktersora - ez maga az óra értéke az 'óó:pp:mm' formátumú  
    karaktersorból */  
    perc = substr(f7, 4, 2);  
    masodperc = substr(f7, 7, 2);  
    /* másodpercben a futott idő f8 nevű változóban */  
    f8 = (15+ora)*3600 + perc*60 + masodperc;  
    /* f9 változóban számoljuk ki a yardstick számmal (f5-ös változó) korri-  
    gált futamidőt másodpercben */  
    f9 = round( f8 / f5 * 100, 1); /* kerekítés egész számra round() függvény-  
    nyel */  
  end;  
run;  
  
/* Rendezzük egy másik adatállományba (hogya az eredetit ne írjuk felül) az  
f9=korrigált futamidő szerint az adatainkat,  
hogya lássuk, milyen értékeket vesz fel ez a változó.  
Ezt persze a PROC MEANS-szel is megtehetnénk.  
A rendezés eredményeként láthatjuk, hogy a legkisebb érték az f9 változóra  
74336 - ezt fogjuk később használni a befutk nevű változóban */  
proc sort data=kornyeze.kek3 out=kornyeze.kek_rendezett;  
  by f9;  
run;  
  
/* Hasonlóan rendezzük a kek3 adatállományt az f8 változó szerint  
A rendezés eredményeként láthatjuk, hogy a legkisebb érték az f8 változóra  
47620 - ezt fogjuk  
később használni a befuti nevű változóban.
```

```

    Vigyázat, ezzel, most felülírtam az előbb f9 szerint rendezett
    kornyeze.kek_rendezett nevű adatállományom,
    most f8 szerint rendezve! */
proc sort data=kornyeze.kek3 out=kornyeze.kek_rendezett;
  by f8;
run;

/* Új csoportosító változó ("csoport") képzése. */
data kornyeze.kek3;
  set kornyeze.kek3; /* ezzel felülírjuk az előző kek3 adatállományunkat */

  if (f9 le 95000) then csoport = 'gyors';
  else if (f9 ge 120000) then csoport = 'lassu';
  else csoport = 'kozep';

  befuti = round((f8-47620)/1000, 1);
  befutk = round((f9-74336)/1000, 1);
run;

/* Rendezés csoport szerint */
proc sort data=kornyeze.kek3 out=kornyeze.kek_rendezett2;
  by csoport;
run;

/* Három adatállomány létrehozása a csoport változó szerint: gyors, lassú,
közepes */
data gyors lassu kozepes;
  set kornyeze.kek_rendezett2;
  if csoport='gyors' then output gyors;
  else if csoport='kozep' then output kozepes;
  else output lassu;
run;

```

A következőkben grafikus megjelenítéseket fogunk alkalmazni a gyakoriságok szemléltetésére. Ehhez érdemes a 2. függelékben megadott dokumentumot elolvasni, ami segítséget nyújt a SAS grafikus elemeinek megértéséhez. Ebben a SAS Graph-N-Go „kattintós” grafikonkészítő felületén keresztül ismerjük meg a forráskódot, és tanuljuk meg, hogy ezt hogyan tudjuk aktualizálni saját igényeink szerint.

```

/* YS szerinti gyakoriságok - már saját formázás, színezés szerint */
options reset=all;

axis1 minor=none label=("YS") ;
axis2 minor=none label=("(Frequency)")

```

```

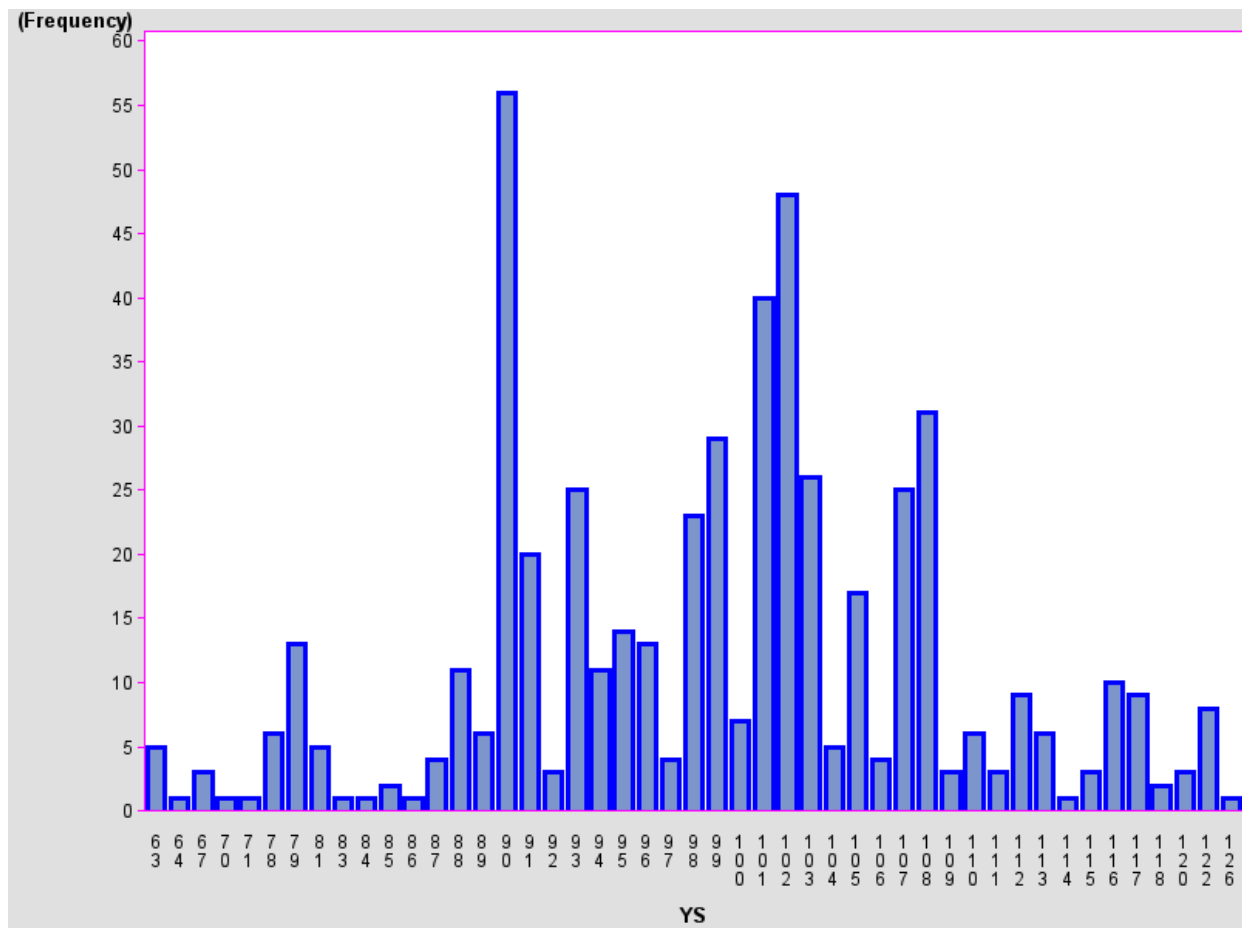
order=(0 to 60 by 5) ;

proc gchart data=KORNYEZE.KEK3;
  vbar F5 /          /* Az f5 változóra futtatunk egy vertikális oszlopdiag-
ramot. */
    type=FREQ /* Mely oszlopdiagramon a gyakoriságokat tüntetjük föl az
y, függőleges tengelyen. */
    discrete /* Minden f5-beli értéket szeretnénk megjeleníteni, va-
gyis minden YS számot, nem akarunk összevonni közülük egyet sem. */
    maxis=axis1 /* A vízszintes tengely beállításai (alosztások, cím-
ke) az "axis1"-en beállítottak szerinti. */
    frame /* Legyen kerete a diagramunknak. */
    cframe=white /* A háttér színe legyen fehér. */
    woutline=2 /* Az oszlopok kontúrvonalának vastagsága legyen 2 pi-
xel. */
    coutline=blue /* Az oszlop kontúrvonalának színe legyen kék. */
    caxis=magenta /* A tengelyek színe legyen lila. */
    raxis=axis2 ; /* A függőleges tengely (response axis) beállítá-
sai az "axis2" szerintiek. */

    /* Ha szeretnénk, adhatunk címet a grafikonunknak. */
    /* title 'A 40. kékszalag verseny résztvevőinek gyakorisága yardstick (YS)
szám szerint' c=blue h=12pt; */
run;
quit;

goptions reset=all;

```



26. ábra: Yardstick szám szerinti gyakoriságok

```
/* Befutási idő szerinti gyakoriságok - hagyva a Graph-N-Go szerinti beállítá-
tásokat. Nézzük meg, hogy melyik hasonlít jobban egy harang görbéhez - a
tényleges befutási idők gyakorisági grafikonja, vagy a következő diagramban
készített korrigált futamidők gyakoriságát szemléltető diagram! */
```

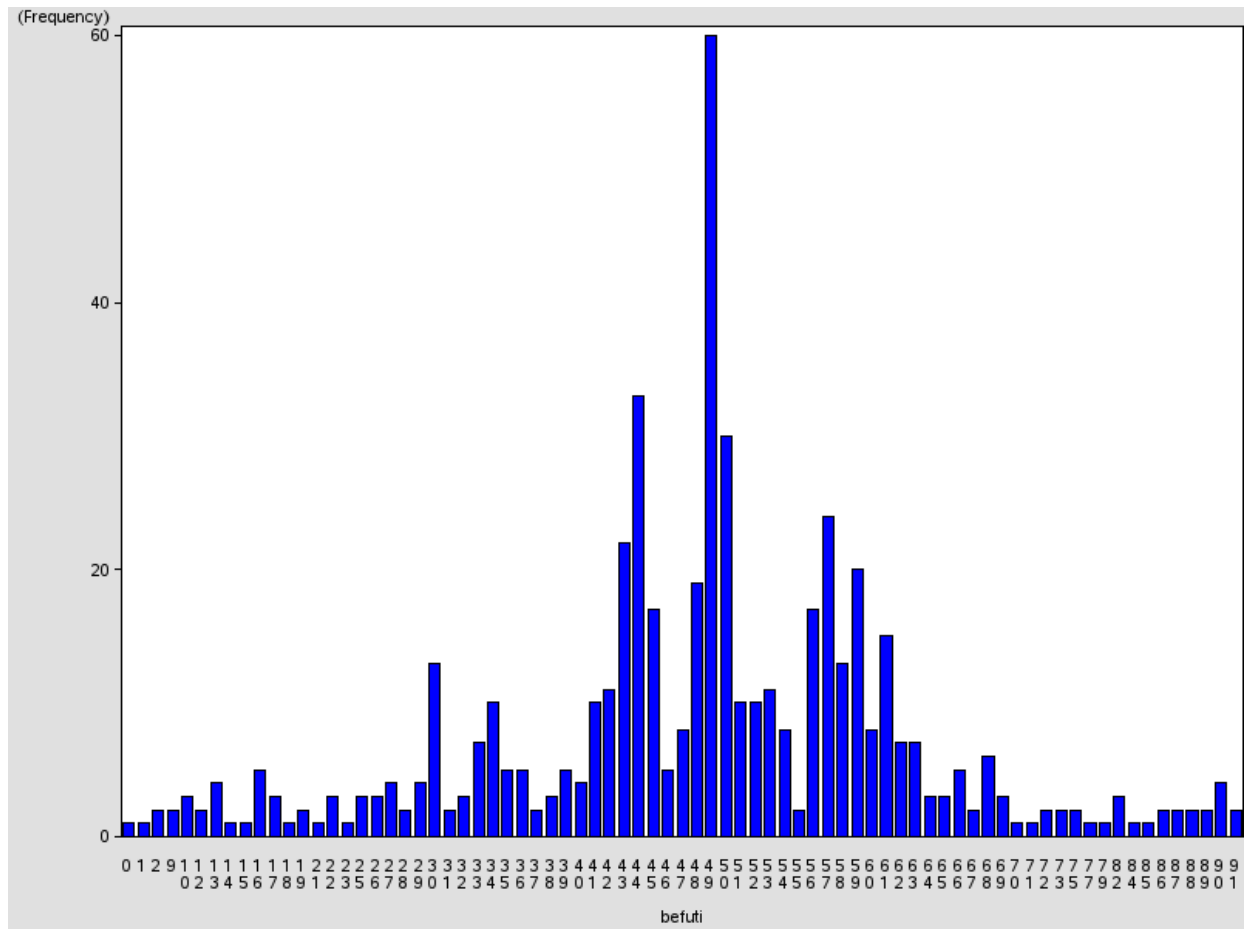
```
goptions reset=all
      ctext=CX000000 ftext="MS Sans Serif" htext=8 pt
      colors=(CX0000FF CXFF0000 CX008080 CX00FF00
              CXFF00FF CXFFFF00 CX00FFFF CX800000
              CX008000 CX800080 CX000080 CX808000
              CXFFFFFF CX808080 CXC0C0C0 CX000000);
```

```
axis1 minor=none label=("befuti") ;
axis2 minor=none label=(" (Frequency) ")
      order=(0 to 60 by 20) ;
```

```
proc gchart data=KORNYEZE.KEK3;
  vbar BEFUTI /
    type=FREQ discrete
    maxis=axis1 frame cframe=CXFFFFFF
    woutline=1 coutline=CX000000 caxis=CX000000
    raxis=axis2 ;
```

```
run;
quit;
```

```
goptions reset=all;
```



27. ábra: Befutási idő szerinti gyakoriságok

```
/* Korrigált futamidő szerinti gyakoriságok - hagyva a Graph-N-Go szerinti  
beállításokat. */
```

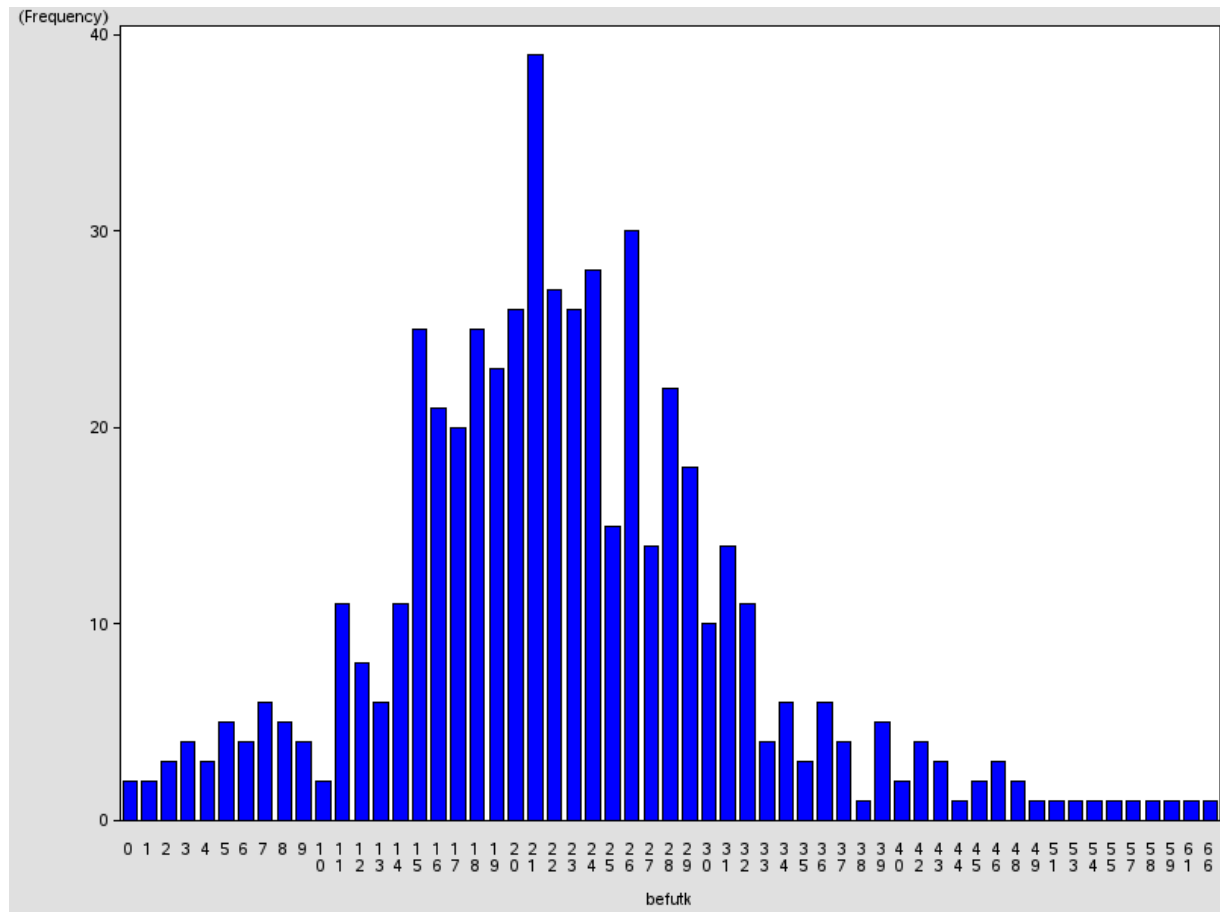
```
goptions reset=all  
  ctext=CX000000 ftext="MS Sans Serif" htext=8 pt  
  colors=(CX0000FF CXFF0000 CX008080 CX00FF00  
          CXFF00FF CXFFFF00 CX00FFFF CX800000  
          CX008000 CX800080 CX000080 CX808000  
          CXFFFFFF CX808080 CXC0C0C0 CX000000);
```

```
axis1 minor=none label=("befutk") ;  
axis2 minor=none label=("Frequency")  
  order=(0 to 40 by 10) ;
```

```
proc gchart data=KORNYEZE.KEK3;  
  vbar BEFUTK /  
    type=FREQ discrete  
    maxis=axis1 frame cframe=CXFFFFFF  
    woutline=1 coutline=CX000000 caxis=CX000000  
    raxis=axis2 ;
```

```
run;
quit;

goptions reset=all;
```



28. ábra: Korrigált futamidő szerinti gyakoriságok a teljes mintára

```
/* Korrigált futamidő szerinti gyakoriságok a gyors hajókra - hagyva a Graph-
N-Go szerinti beállításokat.*/
goptions reset=all
  ctext=CX000000 ftext="MS Sans Serif" htext=8 pt
  colors=(CX0000FF CXFF0000 CX008080 CX00FF00
          CXFF00FF CXFFFF00 CX00FFFF CX800000
          CX008000 CX800080 CX000080 CX808000
          CXFFFFFF CX808080 CXC0C0C0 CX000000);

axis1 minor=none label=("befutk") ;
axis2 minor=none label=("(Frequency)")
order=(0 to 40 by 10) ;

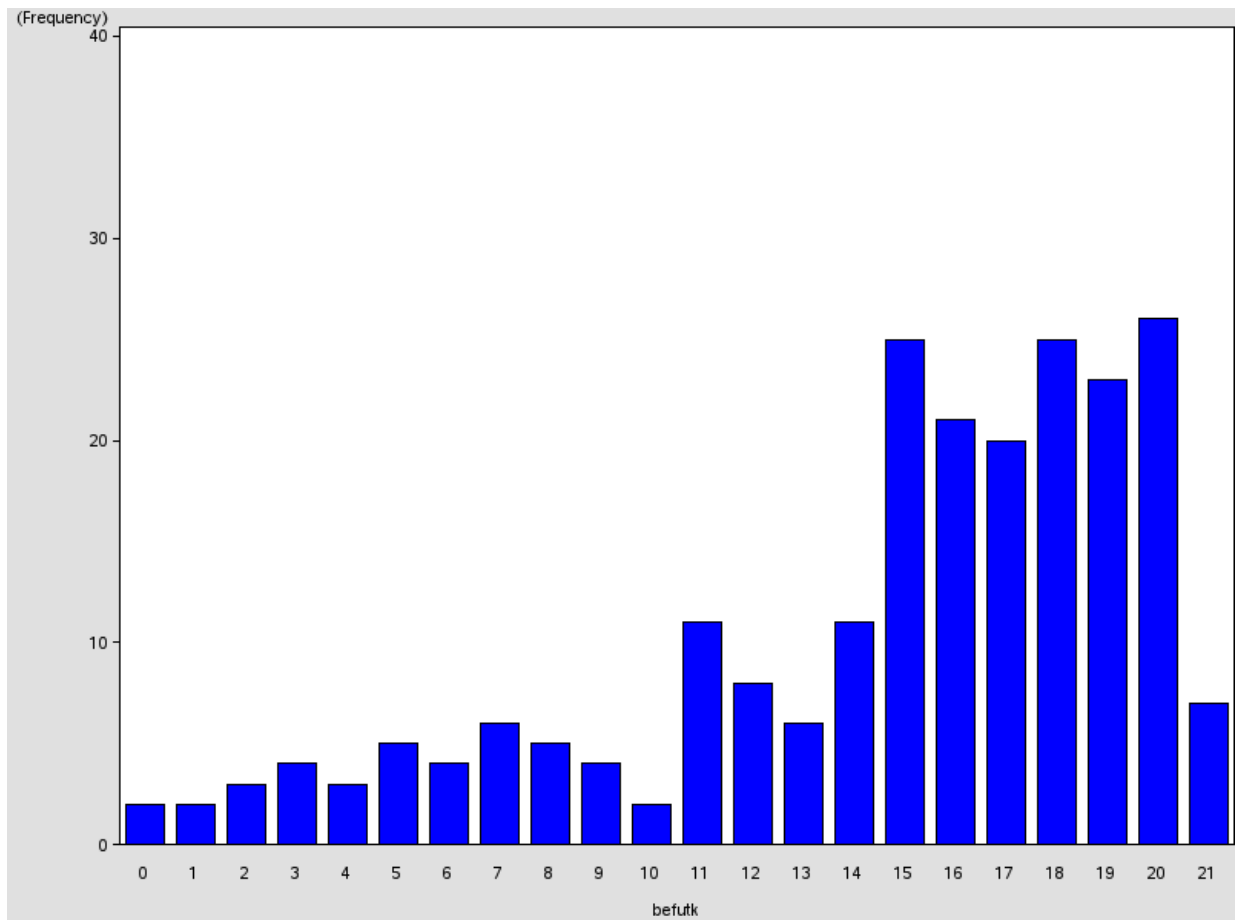
proc gchart data=gyors;
  vbar BEFUTK /
    type=FREQ discrete
    maxis=axis1 frame cframe=CXFFFFFF
```

```

woutline=1 coutline=CX000000 caxis=CX000000
raxis=axis2      ;
run;
quit;

goptions reset=all;

```



29. ábra: Gyors hajók korrigált futamidőinek gyakorisága

6. VALÓS FELADAT – LEVEGŐSZENNYEZÉS

ESETTANULMÁNY

A Tiszai Erőmű jogszabályi kötelezettségének eleget téve rendszeresen szolgáltat levegőtisztaságvédelmi mérési adatokat a Környezetvédelmi Felügyelőség felé. Több paramétert mérnek, többek között a NOx mérés is folyamatos. Az adattáblában 365 nap mérési adatai láthatóak (nem kötelező minden nap mérni, ezért nincs az év minden napjára adatunk).

NOx határérték (törvényileg előírt): 80 ug/m³

Kérdések:

I.

1. Az Erőmű alkalmas-e a jelenleg működtetett technológiával a jogszabályi kötelezettségek teljesítésére? (tűrészhatár 1,5 %) – hatáság mondja pl., hogy 60-nál jelezzen: „48 órán belül szűrőcsere”, 65-nél: „van 36 órája”, stb.
2. Mennyi idő múlva tolódhat el úgy a rendszer, hogy le kell állítani (ha az adatok 40%-a 75 fölötti)?
3. Szükség van-e technológia váltásra a határértékek betartásának biztosítása érdekében?
4. Amennyiben egyedi határértékként 50 ug/m³ kerül megállapításra, akkor is maradhat ez a technológia?

II.

A mérések minden páros napon az „A” jelű mérőhelyen és eszközzel történtek, minden páratlan napon a „B” jelű mérőeszközzel zajlottak.

Észlelhető-e a két eszköz között különbség a mérések szórását illetően?

Melyik eszközből érdemes póteszközt vásárolnunk?

**A cég, az adatsor és a határérték fiktív, csak gyakorlási célra alkalmas.*

Maguk az adatok a 2. mellékletben csatolt dokumentumban találhatóak.

```
/* A Microsoft Word-ből CTRL+c és CTRL+v segítségével másoljuk át notepad-be  
a táblázat tartalmát.  
Mentsük el ezt a file-t a saját könyvtárunkba.  
Érdemes a változókat az első sorban átnevezni könnyen  
kezelhetővé: datum és nox.  
Mentsük ezt a változást is.  
Az így kapott txt file-t importáljuk be a "kornyeze" könyvtárba "nox"  
névvel.
```

Ismerkedjünk meg az adatállománnyal (contents és univariate eljárások)! */

```
proc contents data = kornyeze.nox;
run;
```

Az output egy része:

The CONTENTS Procedure

Data Set Name	KORNYEZE.NOX	Observations	332
Member Type	DATA	Variables	2

...

Alphabetic List of Variables and Attributes					
#	Variable	Type	Length	Format	Informat
2	NOx	Num	8	BEST12.	BEST32.
1	datum	Num	8	YMMDD10.	YMMDD10.

```
proc univariate data = kornyeze.nox;
var nox;
run;
```

The SAS System

The UNIVARIATE Procedure Variable: NOx

Moments			
N	331	Sum Weights	331
Mean	51.1286103	Sum Observations	16923.57
Std Deviation	39.9334462	Variance	1594.68013
Skewness	2.41159179	Kurtosis	8.53628405
Uncorrected SS	1391523.06	Corrected SS	526244.442

Moments			
Coeff Variation	78.1039148	Std Error Mean	2.19494071

Basic Statistical Measures			
Location		Variability	
Mean	51.12861	Std Deviation	39.93345
Median	38.40000	Variance	1595
Mode	16.60000	Range	266.20000
		Interquartile Range	39.72000

Note: The mode displayed is the smallest of 8 modes with a count of 2.

Tests for Location: $\mu_0=0$				
Test	Statistic		p Value	
Student's t	t	23.29385	Pr > t	<.0001
Sign	M	165.5	Pr >= M	<.0001
Signed Rank	S	27473	Pr >= S	<.0001

Quantiles (Definition 5)	
Quantile	Estimate
100% Max	272.07
99%	237.08
95%	118.88
90%	96.24
75% Q3	65.40
50% Median	38.40

Quantiles (Definition 5)	
Quantile	Estimate
25% Q1	25.68
10%	17.95
5%	13.14
1%	8.23
0% Min	5.87

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
5.87	143	208.81	285
7.08	92	237.08	286
7.90	146	253.27	225
8.23	134	270.42	326
8.63	168	272.07	265

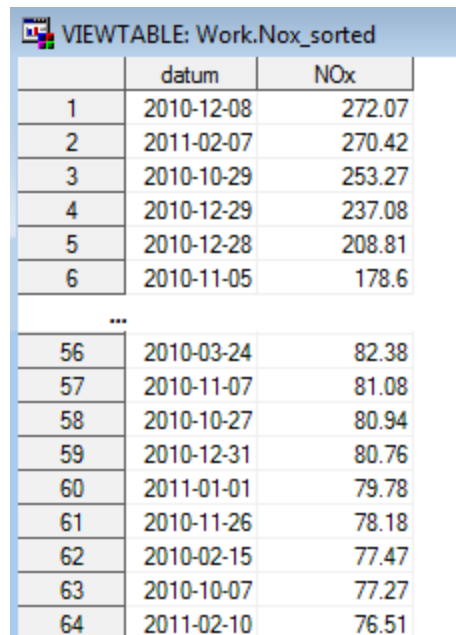
Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	1	0.30	100.00

/* Rendezés után jól látható a Viewtable-ben, hogy mennyi elem van egy-egy határérték felett.

Jelen esetben pl. a legnagyobb érték (a legelső a sorban 272,07, dec. 8-án mért adat, a megadott 80-as határérték felett pedig jól látjuk, hogy 59 nap mért értéke van.)

A rendezés eredményét a "nox_sorted" nevű adatállományba tettük, ezzel nem írjuk felül az eredeti kornyeze.nox adatállományunk. Mivel hosszú távon erre a rendezett nox_sorted adatállományra nincs szükségünk, ezért ideiglenesen hozzuk csak létre, vagyis nem adunk meg könyvtárat, hogy hova mentjük ezt a file-t, ezért a SAS szoftver futása alatt a WORK könyvtárban találjuk meg. De vigyázat, a SAS program bezárásával a WORK tartalma törlődik! */

```
proc sort data=kornyeze.nox out=nox_sorted;
  by descending nox;
run;
```



VIEWTABLE: Work.Nox_sorted

	datum	NOx
1	2010-12-08	272.07
2	2011-02-07	270.42
3	2010-10-29	253.27
4	2010-12-29	237.08
5	2010-12-28	208.81
6	2010-11-05	178.6
...		
56	2010-03-24	82.38
57	2010-11-07	81.08
58	2010-10-27	80.94
59	2010-12-31	80.76
60	2011-01-01	79.78
61	2010-11-26	78.18
62	2010-02-15	77.47
63	2010-10-07	77.27
64	2011-02-10	76.51

30. ábra: Rendezett NOx adatállomány

```
/* 80-as határérték esetén vizsgálódva: */
data kornyeze.nox1;
  set kornyeze.nox END=vege;

  /* A RETAIN-ben megadott változót nem írja felül a program, ha új rekord ke-
  letkezik, hagyja az előző értéken. */
  retain k k75;
  if _N_ eq 1 then do; k=0; k75=0; end; /* Az első sorban 0 értéket állítunk
  be mindkét számlálásra felvett változónkra: k és k75 */

  /* A feladat szerint a mérőműszer tűréshatára 1,5%.
  Ezért hozunk létre egy min és max határokkal meghatározott intervallumot,
  ami a mért NOx értékek +/- 1,5%-os értékeket jelentik. */
  nox_max = nox * 1.015;
  nox_min = nox * 0.985;
```

```

/* Ha a mérőműszer hibájával (tűréshatárával) korrigált mért érték nagyobb
vagy egyenlő, mint 80, vagyis a törvényileg előírt határérték, akkor k vál-
tozó értékét növeljük eggyel: tehát k-ban számláljuk azon mérések számát,
melyek a határérték szerint nem megfelelőek. */
if nox_max ge 80 then k=k+1;

/* Hasonlóan a k75 nevű változó értékét akkor növeljük eggyel, amikor olyan
mérést találunk, amelynek a mérőműszer hibájával korrigált értéke (nox_max)
nagyobb vagy egyenlő, mint 75 */
if nox_max ge 75 then k75 = k75+1;

/* I/1. Ha a mért értékek adott határértéket elérnek, azt jelezzük a LOG-ban
a PUT utasítással */
if nox_max ge 70 then put datum ': 24 óra múlva eléri a határértéket!';
else if nox_max ge 65 then put datum ': 36 órája van a szűrőcserére!';
else if nox_max ge 60 then put datum ': 48 órán belül szűrőcsere!';

/* I/2. Amikor az adatok 40%-a 75 fölötti, akkor kiírjuk a PUT utasítással a
LOG-ba, hogy intézkedni kell. */
if k75 > 332*0.4 then put 'Tessék leállítani a gyárat!!!';

if vege eq 1 then
do; /* Ha elértük az adataink végét, írjuk ki a LOG-ba, hogy miket szám-
láltunk a k és k75 változókban */
put 'A határértéket meghaladó mérések száma: ' k;
put 'A 75 fölötti értékek száma: ' k75 ;
end;

```

run;

```

/* k / _N_ = 60/332 = 0,1807 = 18,07%
Ez a törvényileg előírt 80 ug/m3-es határérték feletti.
N=332, mert 332 nap kellett, hogy mérjenek (bár egy hiányzó érték,
de aznapra is várnánk értéket).

```

I/1. Mivel van olyan, a mérőműszer hibájával (tűréshatár) korrigált érték, amely a törvényileg előírt 80 ug/m3 feletti, ezért valamit tenni kell a gyártás során a levegő tisztaságának érdekében. Valami nincs rendben. Ezt jelzi is nekünk, hogy a LOG-ban több sorban megjelennek üzenetek, hogy mennyi idő múlva kellene pl. szűrőcserét alkalmaznunk.

I/2. Jelen esetben nem lépünk túl a megfigyeléseinkben az előírás szerinti határt (az adatok 40%-a 75 fölötti), ezért nem kapunk "Tessék leállítani a gyárat" üzenetet a LOG-ban. A mi esetünkben k75 értéke 68, vagyis 68 mérés értéke van 75 vagy afölött (a mérőműszer hibáját figyelembe véve), viszont $332 \cdot 0.4 = 132.8$, és ezen érték alatt vagyunk.

I/3. Ha a 80-as határértéket vesszük figyelembe, akkor szükség van technológia váltásra, mert találtunk legalább egy (de sokkal több is ráadásul!) mért értéket a törvényileg előírt felett.

Ha a 2-es feladatban leírt szabályt tekintjük a technológia váltás alapjának, akkor nem kell még leállítani a rendszert, mert nem értük el a megadott százalékot.

A feladat kiírásakor nem volt ismert az adatállomány, csak "értelmes", életszerű kérdéseket foglalmaztunk meg. Ezért van, hogy néhol nem következetes a fogalmazásmód. (Pl. már I/4-ben úgy kérdezi, hogy akkor "is" maradhat-e a technológia - valójában már a 80-as határértéknél sem maradhat, de ezt a feladat megfogalmazásakor még nem gondoltuk.

A feladat gyakorlati jellegű, és így használható.)

*/

```
/* I/4. 50-es határérték esetén vizsgálódva: mivel lejjebb toltuk a korábbi 80-as értékünket, természetesen ezzel növeltük mind k (az 50-es határértéket meghaladó értékek száma), mind k47 (a 47-es értéket meghaladó értékek száma) értékét, ami több szennyezést jelent, rosszabb arányokat jelent. */
```

```
data kornyeze.nox2;
```

```
set kornyeze.nox END=vege;
```

```
/* A RETAIN-ben megadott változót nem írja felül a program, ha új rekord keletkezik, hagyja az előző értéken. */
```

```
retain k k47;
```

```
if _N_ eq 1 then do; k=0; k47=0; end; /* Az első sorban 0 értéket állítunk be mindkét számlálásra felvett változónkra: k és k47 */
```

```
/* A feladat szerint a mérőműszer tűréshatára 1,5%.
```

```
  Ezért hozunk létre egy min és max határokkal meghatározott intervallumot, ami a mért NOx értékek +/- 1,5%-os értékeket jelentik. */
```

```
nox_max = nox * 1.015;
```

```
nox_min = nox * 0.985;
```

```
/* Ha a mérőműszer hibájával (tűréshatárával) korrigált mért érték nagyobb vagy egyenlő, mint 50, vagyis a törvényileg előírt határérték, akkor k változó értékét növeljük eggyel:
```

```
tehát k-ban számláljuk azon mérések számát, melyek a határérték szerint nem megfelelőek. */
```

```
if nox_max ge 50 then k=k+1;
```

```
/* Hasonlóan a k47 nevű változó értékét akkor növeljük eggyel, amikor olyan mérést találunk, amelynek a mérőműszer hibájával korrigált értéke (nox_max) nagyobb vagy egyenlő, mint 47 */
```

```
if nox_max ge 47 then k47 = k47+1;
```

```

/* I/1. Ha a mért értékek adott határértéket elérnek, azt jelezzük a LOG-ban
a PUT utasítással */
if nox_max ge 43 then put datum ': 24 óra múlva eléri a határértéket!';
else if nox_max ge 39 then put datum ': 36 órája van a szűrőcserére!';
else if nox_max ge 35 then put datum ': 48 órán belül szűrőcsere!';

/* I/2. Amikor az adatok 40%-a 47 fölötti, akkor kiírjuk a PUT utasítással a
LOG-ba, hogy intézkedni kell. */
if k47 > 332*0.4 then put 'Tessék leállítani a gyárat!!!';

if vege eq 1 then
do; /* Ha elértük az adataink végét, írjuk ki a LOG-ba, hogy miket szám-
láltunk a k és k47 változókbán */
put 'A határértéket meghaladó mérések száma: ' k;
put 'A 47 fölötti értékek száma: ' k47 ;
end;

run;

data kornyeze.nox_a;
set kornyeze.nox1;

/* Azokat a sorokat írjuk ki, amelyeknek a sorszáma páros, vagyis kettővel
osztva 0 maradékot adnak.
Ezek lesznek a 2., 4., 6., 8., stb. sorok. */
if mod(_N_, 2)=0 then output;
run;

data kornyeze.nox_b;
set kornyeze.nox1;

/* Azokat a sorokat írjuk ki, amelyeknek a sorszáma páratlan, vagyis ket-
tővel osztva 1 maradékot adnak.
Ezek lesznek az 1., 3., 5., 7., stb. sorok. */
if mod(_N_, 2)=1 then output;
run;

/* Készítsünk a két adatállományból egyet.
A set utasítással egymás után illeszti a két adatállományt, nox_a-t és
nox_b-t,

```

```

    ezért utána egy SORT eljárással rendezzük a dátum szerint sorba az értéke-
    ket. */
data kornyeze.nox_ab;
    set kornyeze.nox_a kornyeze.nox_b;
run;
proc sort data=kornyeze.nox_ab;
    by datum;
run;

/* A vonal diagramon történő megjelenítés miatt ideiglenesen sorba te-
szem egy "uj" változóba a felső tűréshatár, nox és alsó tűréshatár
értékeket a páratlan napok adatát tartalmazó adatállományra. */
data nox1_a;
    set kornyeze.nox_a;
    drop nox nox_max nox_min k i; /* töröljük ki ezeket a változókat */
    uj=nox_max; output;          /* és vegyük fel helyettük az "uj" változót */
    uj=nox; output;
    uj=nox_min; output;
    label uj='NOx mért érték'; /* a változónak adjunk címkét - a diagramon
    majd ez jelenik meg, nem az "uj" semmitmondó név */
run;

/* Vonaldiagramon szemléltetve */
options reset=all border nodash; /* Grafikus alapbeállítások */

title 'NOx mért adatok egy év páros napjain'; /* Diagram címe */

/* Egyik tengely beállításai: címke legyen Dátum, színe fekete, tengely
alosztások ne legyenek */
axis1 label=("Dátum") color=black minor=none;
/* Másik tengely beállításai: címke legyen ug/m3, színe fekete, 0-tól 280-ig
20-asával legyenek a főosztások */
axis2 label=("ug/m3") color=black order=(0 to 280 by 20);
/* Használhatnánk harmadik tengelyt is, pl. a jobb oldali függőleges tengely
beállításai az alábbiak lehetnének */
/* axis3 label=none color=black order=(0 to 280 by 20); */

/* Jelmagyarázat beállításai: a keret legyen fekete, címkéje: Jelmagyarázat,
az ábrán fent (a diagram felett) középen helyezkedjen el */
legend1 cborder=black label=('Jelmagyarázat:') position=(top center);

/* Kék vonaldiagram, ahol narancssárga szálkákkal jelöljük az alsó és felső
tűréshatárokat.
A "hiloctj" típusú görbe használata miatt kellett sorba tenni egy "uj" válto-
zóban a kívánt értékeket. */
symbol interp=hiloctj cv=orange ci=blue width=2;

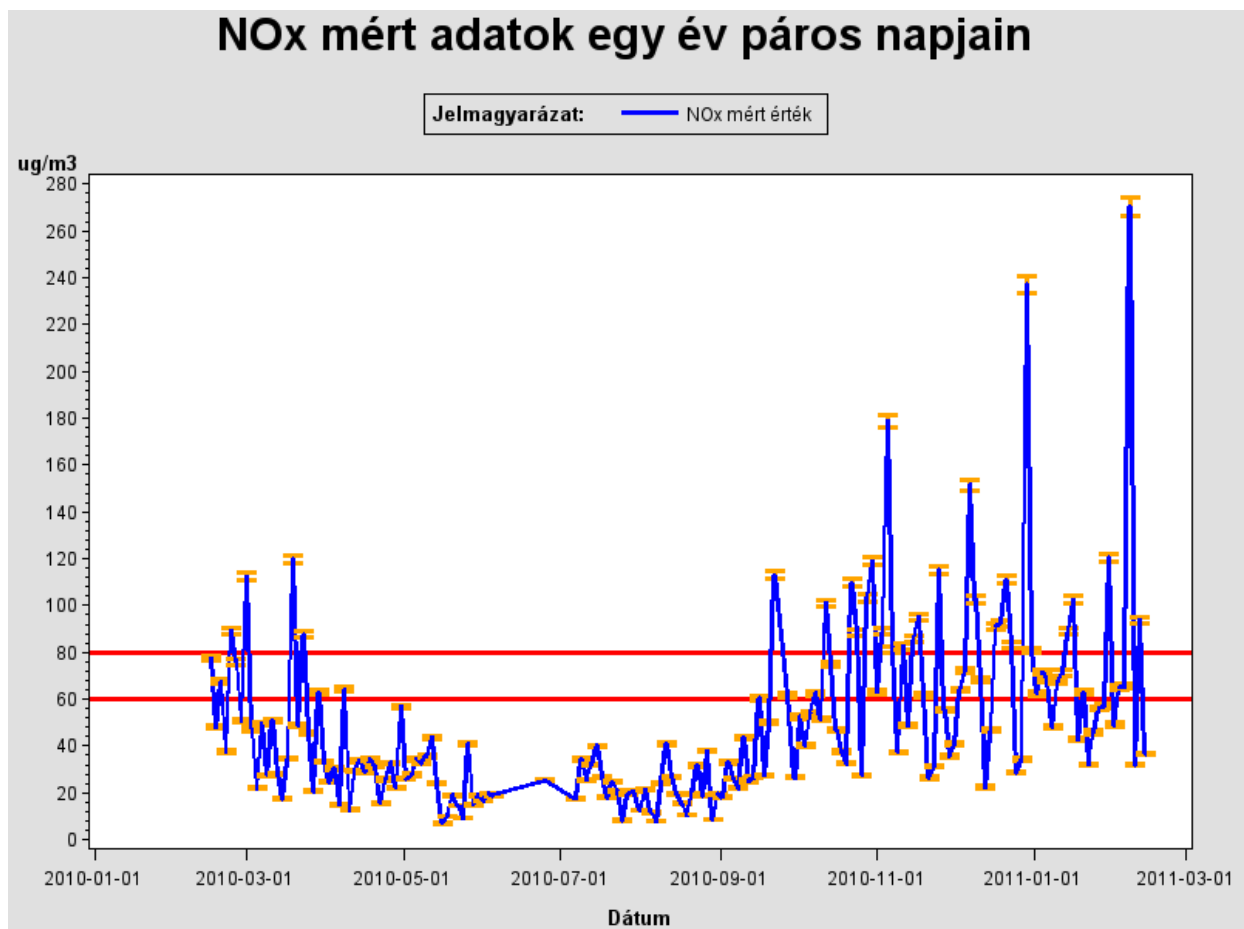
```

```

proc gplot data=nox1_a;
  plot uj*datum
  / legend=legend1          /* Jelmagyarázat használata a fentebb definiált
  "legend1" beállításával */
  haxis=axis1 vaxis=axis2 /* A horizontális tengely beállítása legyen az
  axis1 szerinti, a vertikálisé az axis2 szerinti */
  vref=80 60 cvref=red wvref=2 /* Jelöljük a 80-as és 60-as határérté-
  küket egy referencia vonallal */
  cframe=white caxis=black; /* Háttér fehér, tengely fekete */
run;
quit;

goptions reset=all; /* Grafikus alapbeállítások visszaállítása */

```



31. ábra: Levegőszennyezés vonaldiagramja tőrés határokkal

7. TISZTÍTOTT ÉS NYERS SZENNYVÍZ PH ÉRTÉKEINEK ÖSSZEHASONLÍTÁSA

Feladat:

Szennyvíztisztító (egy fiktív cég) beérkező nyers szennyvizének mintázásából van egy nagyon hiányos táblázat, illetve a tisztított (Tiszába) kimenő víz mintázásából származó adatok állnak rendelkezésre.

1. A táblázatot **ki** kellett **egészíteni**, és **pontosítani** minden érték és szöveg jelentését.
2. **Ha ugyanazon mintának** szeretnénk **többszöri időpontban mért értékét** használni (mint most előbb nyers érték, majd ugyanezen nyersnek egy későbbi időpontban vett tisztított mért értéke), akkor figyeljünk arra, hogy tényleg ugyanahhoz a mintához, **(ugyanabba a sorba) írjuk** mindegyik értéket. (A mi esetünkben egy napon belül van az egy mintának a nyers és tisztított értéke is, ezért ugyanannak a dátumnak a sorában szerepelnek ezek az értékek.)
3. Tisztázzuk a **határértékeket**! (A törvényileg előírt egy nagyobb régióra (pl. országra) vonatkozik, azon belül lehet ugyanazon területet vizsgálva (levegőszennyezés, vízszennyezés, egyéb) több üzem is, akiknek összességében kell a szennyező anyag kibocsátásukban betartani a törvényt. Ezért általában minden cég meghatároz egy-egy sajátos/speciális határértéket, amiben egy biztonsági sávot is hagy, ha bármi történne, ne legyen baj. Ezen egyedi, üzemekre meghatározott helyi határértékek összege nem haladhatja meg a törvényileg előírt összesített határértéket. A szabályozást folyamatosan figyelni kell, mert változhat a törvény, alakulhat új üzem, vagy bővíülhet, de akár el is költözhet ill. bezár. Ekkor a helyi határértékek módosulhatnak a termelés mennyisége és milyensége szerint.)
4. Az egyes mért értékek **mérő műszereinek** hiba határait / **tűrőhatárait** is gyűjtjük össze! A mérő anyagok sokfélesége miatt ezek igen eltérőek lehetnek! Ilyen értékek ismeretében a tűréshatárokkal korrigált értékekkel dolgozunk (**minimum** vagy **maximum**, de **akár mindkettő**, ha felső és alsó határértéket is megszab a törvény ill. az üzem)!
5. A **határértékeket** és tűréshatár értékeket **NEM kell feltüntetni az Excelben** (esetleg a tűréshatárokkal kapott intervallumokat (min és max értékek) itt is kiszámíthatjuk, ha azt nem szeretnénk a SAS-ban – de ott is megtehető a már látott módon könnyedén).
6. Ha esetleg az adataink között vannak számított értékek, akkor azokat ellenőrizzük, hogy pontosak legyenek (pl. tudjuk, hogy E és F oszlopok mért értékeit összegezzük, vagy átlagoljuk a G oszlop-

ban, akkor azt Excelben, vagy SAS-ban is kiszámíthatjuk, hogy biztos értékeink legyenek, mert sokszor van elgépelés).

7. Ezután úgy kell **Excel-ben formázni**, hogy azt a SAS-ba könnyen be lehessen importálni. (Figyeljünk, hogy ne legyenek elgépelések, üres sorok és oszlopok az adattábla előtt. Az első sorban ékezet, szóköz és speciális karakterektől mentes, de egyértelmű változó neveket adjunk meg, melyeket utána a SAS-ban tudunk használni. Nem kell e megfigyelések sorszámát külön oszlopban feltüntetni.)

8. **Importáljuk be SAS-ba** a megfelelő Excel munkalap tartalmát, és ellenőrizzük, hogy látszólag jó adatokat kaptunk-e!

Két fő kérdés merült fel a feladattal kapcsolatban:

1. A nyers szennyvíz egyes alkotóinak összehasonlítása a tisztított szennyvíz ugyanazon alkotóival. Ezeket szeretném egymáshoz viszonyítva ábrázolni.
2. A tisztított szennyvíz egyes alkotóinak ábrázolása és a határérték ábrán való megjelenítése. Ezzel azt szeretném bemutatni, hogy az értékek mennyivel a határérték alatt vannak, esetleg mi az, ami átlépi a határértéket.

Az Excel rendezett adatállomány:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Datum	pH ny	KOI ny	BOI ny	NH4 ny	NO2 ny	NO3 ny	oN ny	oP ny	oLebA ny	oNszlen ny	pH t	KOI t	BOI t	NH4 t	NO2 t	NO3 t	oN t	oP t	oLebA t	oNszlen t	
2	2010.01.06	7.54	235	140	47.02	0.070	0.250	49.73	4.6	251.0	47.340	6.72	26	16	0.05	0.008	4.99	5.36	0.34	10	5.048	
3	2010.01.12	7.2	259	150	32.61	0.030	0.681	36.1	5.6	256.0	33.321	6.92	21	13	0.04	0.031	3.50	3.66	0.28	11	3.571	
4	2010.01.20	7.21	436	325	40.23	0.054	1.836	46.5	6.7	200.0	42.120	6.96	19	12	0.10	0.075	2.19	2.98	0.26	5	2.365	
5	2010.01.24	7.34	538	315	48.98	0.042	1.498	55.5	8.5	135.0	50.520	7.25	18	10	0.21	0.060	3.13	3.54	0.23	5	3.400	
6	2010.02.03	7.87	550	298	29.01	0.045	0.401	34.98	6.4	531.0	29.456	7.41	22	11	0.05	0.006	2.30	3.38	0.23	5	2.356	
7	2010.02.09	7.55	569	300	52.20	0.021	0.250	56.7	7.4	228.0	52.471	7.51	18	3	0.04	0.011	3.36	3.58	0.48	6	3.411	
8	2010.02.17	7.98	375	168	38.23	0.063	2.361	47.89	7.0	241.0	40.654	7.03	30	13	0.05	0.030	3.02	3.29	0.48	22	3.100	
9	2010.02.23	7.13	637	360	34.23	0.054	1.478	37.34	5.6	578.0	35.762	7.11	30	14	0.04	0.032	3.35	3.97	0.29	16	3.422	
10	2010.03.05	7.34	188	115	49.65	0.057	0.316	52.40	4.6	255.0	50.023	6.87	29	19	0.08	0.034	4.03	4.28	0.22	12	4.144	
11	2010.03.11	7.83	313	190	47.87	0.024	1.982	54.74	6.6	373.0	49.876	7.05	28	18	0.08	0.058	6.92	7.84	0.56	10	7.058	
12	2010.03.17	7.02	331	221	29.45	0.046	0.680	35.65	7.8	257.0	30.176	7.01	29	18	0.07	0.125	5.46	6.50	0.27	16	5.655	
13	2010.03.22	7.25	782	480	36.70	0.012	0.316	50.59	6.7	416.0	37.028	6.97	32	23	0.07	0.045	3.20	5.22	0.31	10	3.315	
14	2010.03.31	6.25	502	331	36.02	0.056	1.756	42.31	4.5	258.0	37.832	7.32	27	11	0.07	0.02	3.32	4.46	0.73	5	3.408	
15	2010.04.08	7.11	450	306	32.44	0.047	0.925	39.87	6.5	512.0	33.412	7.16	22	5	0.09	0.062	5.1	5.53	0.44	15	5.252	
16	2010.04.11	7.10	585	406	20.40	0.012	0.316	46.50	5.8	250.0	20.728	7.28	24	14	0.04	0.041	3.94	5.52	0.36	13	4.021	
17	2010.04.21	6.12	343	232	28.67	0.054	1.126	38.50	3.9	532.0	29.850	7.25	19	5	0.06	0.100	4.40	7.00	0.33	9	4.560	
18	2010.04.26	7.51	705	481	48.75	0.045	1.967	60.40	3.8	245.0	50.762	7.01	26	14	0.12	0.049	3.72	4.04	0.29	9	3.889	
19	2010.05.05	7.32	547	289	33.56	0.075	1.039	44.56	4.0	411.0	34.674	7.09	27	17	0.03	0.156	3.75	4.79	0.62	5	3.936	
20	2010.05.10	7.18	462	299	40.00	0.004	0.316	51.04	4.9	176.0	40.320	7.06	24	14	0.05	0.04	2.39	2.57	0.36	14	2.481	
21	2010.05.19	7.36	404	216	27.01	0.005	0.878	39.90	5.8	124.0	27.893	7.22	27	7	0.15	0.06	5.30	6.36	0.20	10	5.510	

Az adatokat megtaláljuk harmadik mellékletben.

Az ábrán is jól látható, hogy olyan adatstruktúrát kell kialakítanunk, hogy az első sorban legyenek ékezet-, speciális karakter- és szóközmentesen a változónevek, a sorokban pedig az egyes megfigyelt mért értékek a megadott napokon. A színek jelen esetben csak arra vonatkoznak, hogy a szennyvíztisztítás

előtt vagy után mért értékekről van szó. Így 10 jellemző van megadva nyers és tisztított értékével (minden színből tehát kettő szerepel a bordótól a kékig).

Megoldás SAS-ban:

```
/* Formázzuk még Excelben az adatokat, hogy ne legyenek üres sorok és oszlopok, és csak azok az adatok szerepeljenek, amiket használnunk kell: vagyis PH, KOI, BOI, NH4, NO2, NO3, oN, oP, oLebA, oNsztlen 10 db mutató (érték változó) van és mindegyikből van ny: nyers és t: tisztított mért érték. Ismerkedjünk meg az adatállománnyal (contents és univariate eljárások)! */  
proc contents data = kornyeze.kk;  
run;  
/* 21 változónk van: dátum és 10 nyers plusz 10 tisztított mért érték 52 megfigyelésünk (sorunk) van a kk adatállományban. Minden változó (oszlop) numerikus típusú, a dátum speciális numerikus típus: DATE9 */
```

```
/* Csak néhány változót emelek ki most, de érdemes lehet mindegyikre futtatni leíró statisztikát, vagy arra, amelyekre különösen kíváncsi vagyok. A futás eredményét értelmezzük a korábban tanult módon. */  
proc univariate data = kornyeze.kk;  
var ph_ny ph_t koi_ny koi_t;  
run;
```

```
/* Minden változóra van előírt határérték:  
PH: 6-9.5 között jó  
KOI: 75  
BOI: 25  
NH4: 5  
NO2:  
NO3: 10  
oN: 10 ? - nem a nitrogének össz határértéke? 10+5 már több, mint 10...  
oP: 1  
oLebA: 20  
oNsztlen:
```

Rendezés után jól látható a Viewtable-ben, hogy mennyi elem van egy-egy határérték felett.

Jelen esetben, a teljesség igénye nélkül csak a PH tisztított értékre futtatva

pl. a legnagyobb érték (a legelső a sorban 7.98, dec. 29-én mért adat,

a legkisebb 6, vagyis minden érték az előírt 6-9.5 intervallumon belül van.)

Miért nem PROC MEANS-t futtatunk?

Mert ugye ott is lehetne min és max értékeket látnunk, de, ha lenne kiugró érték, azt nem tüntetné fel.

Az így rendezett adatállományban pedig minden értéket láthatunk, és ha lenne kiugró, azokat könnyebb lenne megszámolni sorok számozása révén.

A rendezés eredményét a "kk_sorted" nevű adatállományba tettük, ezzel nem írjuk felül az eredeti kornyeze.kk adatállományunk. Mivel hosszú távon erre a rendezett kk_sorted adatállományra nincs szükségünk, ezért ideiglenesen hozzuk csak létre, vagyis nem adunk meg könyvtárat, hogy hova mentjük ezt a file-t, ezért a SAS szoftver futása alatt a WORK könyvtárban találjuk meg. De vigyázat, a SAS program bezárásával a WORK tartalma törlődik!

```
*/
```

```
proc sort data=kornyeze.kk out=kk_sorted;  
  by descending ph_t;  
run;
```

```
/* A tisztított PH határérték intervallum esetén vizsgálódva: */
```

```
data kornyeze.kk1;  
  set kornyeze.kk END=vege;
```

```
/* A RETAIN-ben megadott változót nem írja felül a program, ha új rekord ke-  
letkezik, hagyja az előző értéken. */
```

```
retain k6 k9;  
if _N_ eq 1 then do; k6=0; k9=0; end; /* Az első sorban 0 értéket állítunk  
be mindkét számlálásra felvett változónkra: k6 és k9 */
```

```
/* Változóknak címkék adása: */
```

```
label  datum = 'Dátum'  
       ph_ny = 'Nyers PH érték'  
       koi_ny = 'Nyers Kémiai Oxigén Igény'  
       boi_ny = 'Nyers Biológiai Oxigén Igény'  
       nh4_ny = 'Nyers NH4'  
       no2_ny = 'Nyers Nitrogén-dioxid'  
       no3_ny = 'Nyers Nitrogén-trioxid'  
       oN_ny = 'Összes Nyers Nitrogén'  
       oP_ny = 'Összes Nyers Foszfor'  
       oLebA_ny = 'Összes Nyers Lebegő Anyag'  
       oNsztlen_ny = 'Összes Nyers Szervetlen Nitrogén'  
       ph_t = 'Tisztított PH érték'  
       koi_t = 'Tisztított Kémiai Oxigén Igény'  
       boi_t = 'Tisztított Biológiai Oxigén Igény'
```

```

    nh4_t = 'Tisztított NH4'
    no2_t = 'Tisztított Nitrogén-dioxid'
    no3_t = 'Tisztított Nitrogén-trioxid'
    oN_t = 'Összes Tisztított Nitrogén'
    oP_t = 'Összes Tisztított Foszfor'
    oLebA_t = 'Összes Tisztított Lebegő Anyag'
    oNszetlen_t = 'Összes Tisztított Szervetlen Nitrogén';

/* A feladat szerint a PH-ra nincs tűréshatár / mérőműszer hibája, ezért az
eredeti ph_t változóval dolgozunk */

/* Ha a nyers mért PH érték nagyobb, mint 9.5, vagy kisebb, mint 6,
vagyis a törvényileg előírt intervallumon kívül esik, akkor k6 ill. k9
változók értékét növeljük eggyel:
aszerint, hogy amikor 6 alatti értéket találunk, legyen k6 értéke eggyel
nagyobb,
ha pedig 9.5 fölötti a mért értékünk, akkor növeljük 1-gyel k9 változónk
értékét.
Tehát külön számláljuk a fölfelé és lefelé kiugró értékeket. */
if ph_ny gt 9.5 then k9=k9+1;
if ph_ny lt 6 then k6=k6+1;

if vege eq 1 then
    do; /* Ha elértük az adataink végét, írjuk ki a LOG-ba, hogy miket szám-
láltunk a k6 és k9 változókban */
        put 'A felső határértéket meghaladó nyers PH mérések száma: ' k9;
        put 'Az alsó határértéket el nem érő nyers PH értékek száma: ' k6;
    end;

run;

/* Vonaldiagramon szemléltetve */
goptions reset=all border nodash; /* Grafikus alapbeállítások */

title 'Nyers és tisztított PH mért adatok egy év 52 napján'; /* Diagram címe
*/

/* Egyik tengely beállításai: címke legyen Dátum, színe fekete, tengely
alosztások ne legyenek */
axis1 label=("Dátum") color=black minor=none;
/* Másik tengely beállításai: címke legyen PH, színe fekete, 0-tól 10-ig 0.5-
esével legyenek a főosztások */
axis2 label=("PH") color=black order=(5 to 10 by 0.5);

/* Jelmagyarázat beállításai: a keret legyen fekete, címkéje: Jelmagyarázat,

```

```

    az ábrán fent (a diagram felett) középen helyezkedjen el */
legend1 cborder=black label=('Jelmagyarázat:') position=(top center);

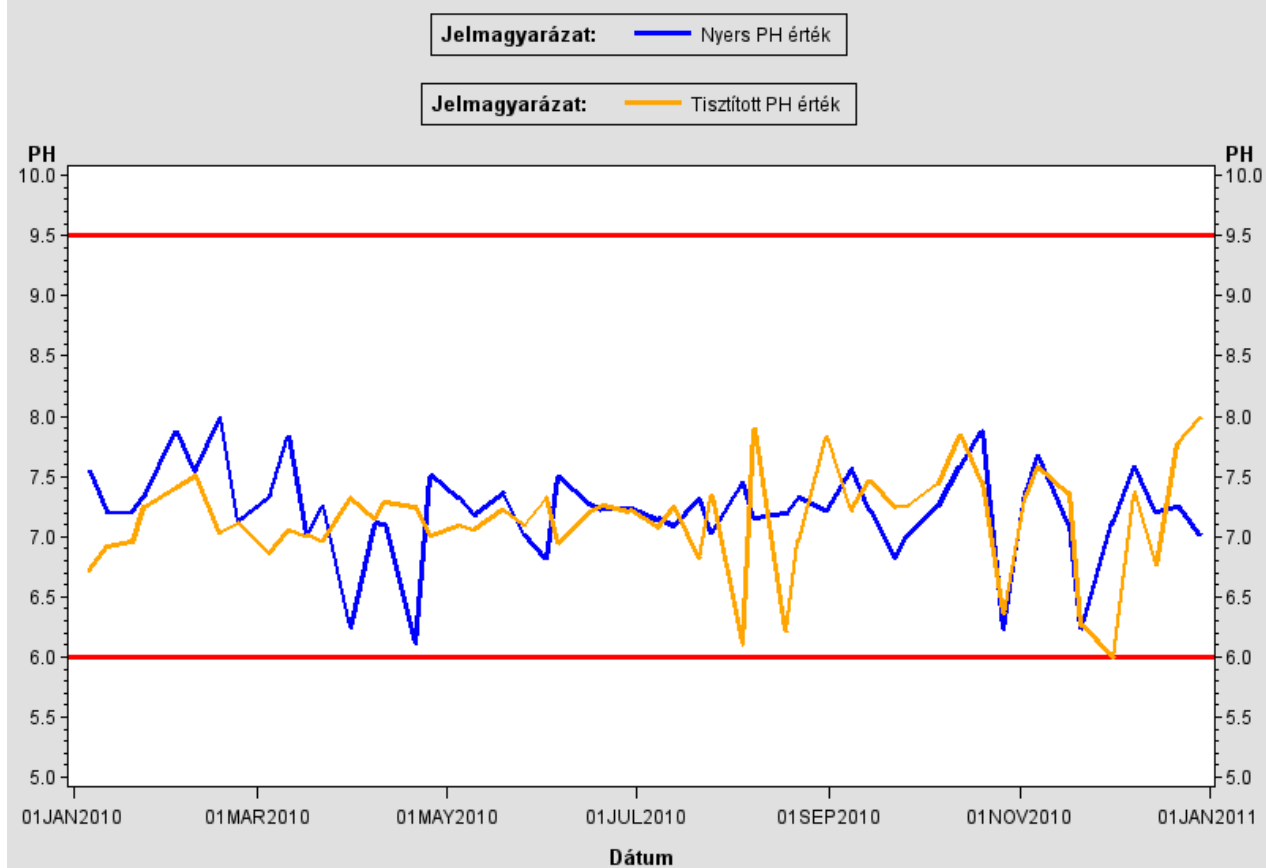
/* Kék vonaldiagram, ahol narancssárga szálkákkal jelöljük az alsó és felső
tűrészhatárokat.
A "hiloctj" típusú görbe használata miatt kellett sorba tenni egy "uj" válto-
zóban a kívánt értékeket. */
symbol1 interpol=join line=1 ci=blue width=2;
symbol2 interpol=join line=1 ci=orange width=2;

proc gplot data=kornyeze.kk1;
    plot ph_ny*datum=1 /* Az első görbe a symbol1 szerinti formában jelenjen
meg */
    / legend=legend1 /* Jelmagyarázat használata a fentebb definiált
"legend1" beállításával */
    haxis=axis1 vaxis=axis2 /* A horizontális tengely beállítása legyen az
axis1 szerinti, a vertikálisé az axis2 szerinti */
    vref=6 9.5 cvref=red wvref=2 /* Jelöljük a 6-os és 9.5-ös határérté-
keinket egy-egy 2 pont vastagságú piros referencia vonallal */
    cframe=white caxis=black; /* Háttér fehér, tengely fekete */
    plot2 ph_t*datum=2 /* A második görbe a symbol2 szerinti formában je-
lenjen meg */
    / legend=legend1 /* Jelmagyarázat használata a fentebb definiált
"legend1" beállításával */
    haxis=axis1 vaxis=axis2 /* A horizontális tengely beállítása legyen az
axis1 szerinti, a vertikálisé az axis2 szerinti */
    vref=6 9.5 cvref=red wvref=2 /* Jelöljük a 6-os és 9.5-ös határérté-
keinket egy-egy referencia vonallal - ez most ki is hagyható, mert az
előző két referencia vonallal egybe esik */
    cframe=white caxis=black; /* Háttér fehér, tengely fekete */
run;
quit;

goptions reset=all; /* Grafikus alapbeállítások visszaállítása */

```

Nyers és tisztított PH mért adatok egy év 52 napján



32. ábra: Nyers és tisztított PH értékek vonaldiagramja SAS-sal

8. SZENNYVÍZTISZTÍTÁS - ADATGENERÁLÁS

Feladat rövid leírása: Egy gépipari vállalat felületkezelő üzemében fém alkatrészek eloxálása, kromátozása és impregnálása történik. Az eloxálás illetve kromátozás után a keletkező szennyvizet a vállalat saját szennyvízkezelő berendezésében tisztítják, mely így a közüzemi csatornahálózatba kerül. A veszélyes anyagokat tartalmazó iszapot szállító járművekkel szállítják el.

Kérdés: Mekkora a valószínűsége annak, hogy a szennyvíztisztítóból kilépő tisztított szennyvíz nem lépi túl a meghatározott határértékeket?

A szennyvízben található, környezeti szempontból releváns komponensek illetve értékek a következők:

pH

KOI – Kémiai oxigénigény

BOI – Biológiai oxigénigény

SZOI – Szerves oldószer extrakt

Össz só

Vas

Cink

Nikkel

Kadmium

Króm (összes)

Réz

Ólom

A mellékelt Excel táblázatban (4. melléklet) látható, hogy az egy évre vonatkozó adatok a fent említett komponensek tekintetében igen hiányosak. Havonta történik belső mérés, melynek eredményeképp komponensenként évi 6 adattal rendelkezünk. Emellé további virtuális adatokat szeretnénk generálni, a már meglévőket 365-re kiegészíteni - így napi egy adattal számolhatunk a kockázatelemzés során.

A generáláshoz szükséges tűréshatár adatokat a következő táblázat utolsó oszlopában találjuk:

Jogszabályban előírt határértékek: 28/2004. (XII. 25.) KvVM rendelet 4. melléklete alapján, valamint az ezek alapján alkotott intervallumok (range-ek):

<i>Komponens</i>	<i>Jogszabályi határérték [mg/l]</i>	<i>Mérés alapjául szolgáló szabvány</i>	<i>Tűrőhatár [mg/l]</i>
KOI	1000	MSZ ISO 6060:1991	200 – 950
BOI	500	MSZ en 1899-1	100 – 450
SZOE	50 ⁽¹⁾	MSZ 1484-12:2002	5– 45
Össz só	2500	MSZ 260-3:1973	650 – 2250
Vas	20	MSZ 1484-3:2006	0,1– 18
Cink	2	MSZ 1484-3:2006	0.1 – 0.18
Nikkel	1	MSZ 1484-3:2006	0,05 – 0.9
Kadmium	0.1	MSZ 1484-3:2006	0,01 – 0.095
Króm (összes)	1	MSZ 1484-3:2006	0,05– 95
Réz	2	MSZ 1484-3:2006	0,1 – 1.8
Ólom	0.2	MSZ 1484-3:2006	0,005 – 0.018
	<i>Mértékegység nélkül</i>		
pH	6-9.5	MSZ 1484-22:2009	5.8-9.7

(1) 100 m³/d kibocsátás alatt a határérték növényi és állati eredet esetén háromszoros, fölötte kétszeres.

A pH-n kívüli összes érték mg/l-ben adott.

Adatsor

Mintavétel dátuma: III.01.			
Komponens neve	Vizsgálati eredmények	Határérték	Mértékegység
pH	7,74	10,00	
KOI	683,00	1000,00	mg/l
BOI (5)	260,00	500,00	mg/l

SZOE	16,00	50,00	mg/l
Króm (összes)	0,80	1,00	mg/l
Mintavétel dátuma: IV.29. Kontrollminta, Mérési eredmények			
Komponens neve	Vizsgálati eredmények	Határérték	Mértékegység
pH	7,52	10,00	
KOI	520,00	1000,00	mg/l
BOI (5)	275,00	500,00	mg/l
SZOE	43,00	50,00	mg/l
Össz. Só	895,00	2500,00	mg/l
Vas	0,41	10,00	mg/l
Cink	0,36	2,00	mg/l
Króm (összes)	0,08	1,00	mg/l
Mintavétel dátuma: VI. 02.			
Komponens neve	Vizsgálati eredmények	Határérték	Mértékegység
pH	7,52	10,00	
KOI	524,00	1000,00	mg/l
BOI (5)	300,00	500,00	mg/l
SZOE	8,50	50,00	mg/l
Össz. Só	795,00	2500,00	mg/l
Vas	0,85	10,00	mg/l
Cink	0,33	2,00	mg/l
Króm (összes)	0,06	1,00	mg/l
Mintavétel dátuma: IX. 02.			
Komponens neve	Vizsgálati eredmények	Határérték	Mértékegység
pH	7,37	10,00	
KOI	962,00	1000,00	mg/l
BOI (5)	450,00	500,00	mg/l
SZOE	42,00	50,00	mg/l
Össz. Só	970,00	2500,00	mg/l
Vas	0,98	10,00	mg/l
Cink	0,95	2,00	mg/l
Króm (összes)	0,71	1,00	mg/l
Mintavétel dátuma: X. 18.			
Komponens neve	Vizsgálati	Határérték	Mértékegység

eredmények			
pH	7,53	10,00	
KOI	207,00	1000,00	mg/l
BOI (5)	110,00	500,00	mg/l
SZOE	17,00	50,00	mg/l
Össz. Só	1660,00	2500,00	mg/l
Vas	0,44	10,00	mg/l
Cink	0,15	2,00	mg/l
Króm (összes)	0,09	1,00	mg/l
Mintavétel dátuma: XII. 01.			
Komponens neve	Vizsgálati eredmények	Határérték	Mértékegység
pH	7,36	10,00	
KOI	312,00	1000,00	mg/l
BOI (5)	185,00	500,00	mg/l
SZOE	26,00	50,00	mg/l
Össz. Só	1430,00	2500,00	mg/l
Vas	0,36	10,00	mg/l
Cink	0,26	2,00	mg/l
Króm (összes)	0,14	1,00	mg/l
Hatósági mérési eredmények 2010			
Mintavétel dátuma: IV. 29.			
Komponens neve	Vizsgálati eredmények	Határérték	Mértékegység
pH	7,30	10,00	
KOI	560,00	1000,00	mg/l
SZOE	42,00	50,00	mg/l
Össz. Só	692,00	2500,00	mg/l
Vas	0,52	10,00	mg/l
Cink	0,17	2,00	mg/l
Nikkel	0,10	0,10	mg/l
Kadmium	0,01	0,05	mg/l
Króm (összes)	0,11	1,00	mg/l
Réz	0,10	2,00	mg/l
Ólom	0,06	0,20	mg/l
Mintavétel dátuma: XI. 25.			
Komponens neve	Vizsgálati eredmények	Határér- ték	Mértékegység

pH	7,90	10,00	
KOI	600,00	1000,00	mg/l
SZOE	26,00	50,00	Mg/l
Össz. Só	425,00	2500,00	Mg/l
Vas	0,65	10,00	Mg/l
Cink	0,16	2,00	Mg/l
Nikkel	0,10	0,10	Mg/l
Kadmium	0,01	0,05	Mg/l
Króm (összes)	0,50	1,00	Mg/l
Réz	0,10	2,00	Mg/l
Ólom	0,05	0,20	Mg/l
Mintavétel dátuma: XI.25. Kontrollminta, Mérési eredmények			
Vizsgálati			
Komponens neve	eredmények	Határérték	Mértékegység
pH	7,90	10,00	
KOI	560,00	1000,00	Mg/l
SZOE	32,00	50,00	Mg/l
Össz. Só	510,00	2500,00	Mg/l
Vas	0,56	10,00	Mg/l
Cink	0,20	2,00	Mg/l
Nikkel	0,05	0,10	mg/l
Kadmium	0,01	0,05	mg/l
Króm (összes)	0,30	1,00	mg/l
Réz	0,13	2,00	mg/l
Ólom	0,05	0,20	mg/l

A mellékelt Excel tábla ezen adatokat tartalmazza rendezett formában.

```

/* Az Excelben törölve az elejéről a fölösleges sorokat, oszlopokat,
rendezve az adatokat, ékezet nélküli változóneveket adva az első sorban
beimportáljuk SAS-ba az erre előkészített „SAS” Excel munkalapot.
Előfordulhat, hogy az adatállományban a számok után még sok üres sor sze-
repel, tele hiányzó értékekkel. Ekkor az alábbi programot futtassuk le, és
csak a 7-es sor előtti értékeket hagyjuk meg. */
data kornyeze.ha;
  set kornyeze.ha;
  if _N_ lt 7 then output;
run;

```

Az adataink SAS adatállományban az alábbi képen néznek ki:

	Datum	pH	KOI	BOI	SZOE	OsszSo	Vas	Cink	Nikkel	Kadmium	OsszKrom	Rez	Olom
1	01MAR2010	7.74	683	260	16	9999.99	9999.99	9999.99	9999.99	9999.99	0.8	9999.99	9999.99
2	29APR2010	7.52	520	275	43	895	0.41	0.36	9999.99	9999.99	0.08	9999.99	9999.99
3	02JUN2010	7.52	524	300	8.5	795	85	0.33	9999.99	9999.99	0.06	9999.99	9999.99
4	02SEP2010	7.37	962	450	42	970	0.98	0.95	9999.99	9999.99	0.71	9999.99	9999.99
5	18OCT2010	7.53	207	110	17	1660	0.44	0.15	9999.99	9999.99	0.09	9999.99	9999.99
6	01DEC2010	7.36	312	185	26	1430	0.36	0.26	9999.99	9999.99	0.14	9999.99	9999.99

33. ábra: Hat soros eredeti szennyvíztisztításnál mért adatállomány SAS-ban (kornyeze.ha)

```

/* A következő DATA lépésben generálunk 365 napra adatot. */
data kornyeze.ha_gen (drop=ev ho nap); /* Az újonnan bevezetett ev, ho, nap
változókat nem íratjuk ki az adatállományba, ezeket eldobjuk DROP-pal */
set kornyeze.ha ; /* betöltjük a HA adatállomány tartalmát, ezen dolgozunk
tovább a HA_GEN adatállományban */

/* a datum nevű változónak lekérdezhetjük az év, hónap és nap értékeit.
Ezek azok a változók, amiket a SAS adatállományban (kornyeze.ha_gen) nem
íratunk ki a DATA sorában lévő DROP utasítás miatt. */
ev = year(datum);
ho = month(datum);
nap = day(datum);

format datum 9. ; /* numerikussá alakítjuk a dátum formátumot, hogy azzal
tudjunk számolni */

if _N_ eq 1 then /* ha az adatállomány első sorában vagyunk, akkor */
do ; /* tegyük a a következőket */
put datum ev ho nap; /* A LOG-ból kiolvashatjuk, hogy 2010 március 1-je
numerikusan 18322
Ez az 1960 január 1 óta eltelt napok száma.
Valójában csak ezért a kiíratásért használtuk a 3 új változót. */

do i=18322-59 to 18322-59+364 ; /* március 1-től visszaszámolunk 59 napot,
hogy január 1-et kapjunk és összesen 1+364 sort/megfigyelést készítünk */
datum = i; /* mivel a dátum számformátum, ezért a mindig egy-
gyel növekvő i változó értékével tesszük egyenlővé */
ph = 5.8 + ranuni(1)*3.9 ; /* a tűrészatárok szerint generálunk adatokat.
A ranuni(1) függvény 0 és 1 közötti valós számot generál. Ezt alakítjuk a
saját igényeink szerint. */
koi = 200 + ranuni(1)*750 ;
boi = 100 + ranuni(1)*350;
szo = 5 + ranuni(1)*40;
osszso = 650 + ranuni(1)*1600;
vas = 0.1 + ranuni(1)*17.9;
cink = 0.1 + ranuni(1)*0.08;
nikkel = 0.05 + ranuni(1)*0.85;
kadmium = 0.01 + ranuni(1)*0.085;
osszkrom = 0.05 + ranuni(1)*0.9;
rez = 0.1 + ranuni(1)*1.7;
olom = 0.005 + ranuni(1)*0.013;
output ;
end ; /* a DO ciklus végét END zárja */
end ;

```

```

format datum date9.; /* visszaalakítjuk numerikusról dátum formátummá a
datum változót, hogy értelmezni tudjuk*/

run ;

/* Teljes év adatai, ahol van eredeti adatokkal, egyébként generáltakal: */
data kornyeze.ha_kieg;
/* A MERGE utasítással a BY-ban megadott változó szerint rendezettek a meg-
adott adatállományaink: ha_gen és ha.
Ha nem lennének datum szerint rendezve, akkor a PROC SORT eljárást kellene
használnunk külön-külön mindkét adatállományra
(PROC SORT data=kornyeze.ha_gen; BY datum; RUN;
PROC SORT data=kornyeze.ha; BY datum; RUN;)
Ezeket természetesen nem a DATA lépésben, hanem az előtt!
A MERGE a rendezett datum szerint párosítja a változóinkat, vagyis a megfele-
lő dátumú sorokba beteszi azokat a változókat, amik a ha_gen adatállományban
vannak, majd megkeresi, hogy vannak-e megegyező dátumú sorok a ha adatállo-
mányból, és azok tartalmával kiegészíti a megegyező nem üres oszlopokat.*/
merge kornyeze.ha_gen kornyeze.ha;
by datum;
run;

```

A következő ábrán láthatjuk a generált adatállomány egy részletét, amiből jól látszódik, hogy a generált adatok több tizedes jeggyel rendelkeznek, míg az eredeti adatok is megmaradtak (lásd 2010 március 1-jei adatsort). Ebben is láthatjuk, hogy az OsszSo, Vas, Cink, Nikkel, Kadmium, Rez és Olom változókhoz generált adatok szerepelnek, míg a többi az eredetileg begyűjtött adat. Az előző ábrával összevetve kitűnik, hogy valóban az ott hiányzó értéként '9999.99' érték helyett szerepelnek itt csak generált adatok, a többi maradt az eredeti.

	Datum	pH	KOI	BOI	SZOE	OsszSo	Vas	Cink	Nikkel	Kadmium	OsszKrom	Rez	Olom	i
56	25FEB2010	8.17901	898.2958	383.25499	43.8410499	1682.659	15.915485	0.13130035	0.7488438	0.0462	0.59549	1.693594	0.0122533	18318
57	26FEB2010	8.8244	319.1785	439.16825	40.0314114	2066.1229	4.0871957	0.17895355	0.1395512	0.03447	0.3381206	1.15226	0.0111698	18319
58	27FEB2010	9.37647	500.3752	416.11487	30.8190087	2159.4668	13.626954	0.17436434	0.0905164	0.038956	0.91777812	0.602694	0.0119273	18320
59	28FEB2010	8.32926	465.8033	102.86937	6.49165414	1154.1301	7.5735696	0.16238003	0.6408066	0.077773	0.1845381	1.338009	0.0137684	18321
60	01MAR2010	7.74	683	260	16	945.94011	17.464588	0.13198594	0.2704888	0.088336	0.8	1.747771	0.0120587	18322
61	02MAR2010	6.15511	232.8257	431.3265	30.9981265	1094.8565	7.6778272	0.13212527	0.1623531	0.01291	0.3036038	0.20735	0.0130912	18323
62	03MAR2010	8.48969	412.2964	263.87961	44.6380537	1973.9251	8.5332228	0.15256119	0.4498698	0.063865	0.7774972	0.65446	0.0091951	18324
63	04MAR2010	6.38892	215.406	349.05482	33.1150024	928.76036	11.737241	0.16245231	0.6080256	0.069251	0.644862	1.146085	0.015163	18325
64	05MAR2010	7.139	286.4429	274.73294	11.6168445	1443.4933	3.5233995	0.1037584	0.6136525	0.052198	0.2932552	0.670521	0.0080255	18326
65	06MAR2010	9.10758	407.8882	142.3284	21.9004163	1697.6683	0.3422308	0.12461728	0.0918102	0.044973	0.9427001	1.458971	0.007415	18327
66	07MAR2010	7.69572	383.3105	242.38186	36.4322552	1677.13	11.464087	0.14876747	0.0775816	0.017636	0.5230096	0.501819	0.0062391	18328
67	08MAR2010	8.01314	742.7984	443.52953	10.133161	936.02834	3.0865474	0.15475736	0.5404372	0.090974	0.3335229	1.310664	0.0121315	18329
68	09MAR2010	9.57919	731.0091	339.77781	37.7187857	2172.5759	15.527941	0.1013844	0.8669326	0.090719	0.6312651	1.155956	0.0101233	18330
69	10MAR2010	9.27641	726.1694	232.12957	22.2785107	1686.0739	3.5677077	0.11074167	0.1852974	0.074583	0.4349367	1.670069	0.0147323	18331
70	11MAR2010	9.1842	622.438	231.402	22.6221873	1991.3931	2.1784885	0.10048809	0.3097741	0.010204	0.8048344	0.277431	0.0075309	18332
71	12MAR2010	7.34251	905.2035	275.64863	41.3323335	1313.2449	12.189568	0.1196628	0.7648168	0.082238	0.7421357	1.126837	0.0075073	18333

34. ábra: Az eredeti hat napi mért hiányos szennyvíztisztítási adatok kiegészítése egy év minden napjára generált adatokkal (kornyeze.ha_kieg)

Természetesen lehetőség van adott időszakra vonatkoztatva adott értékeket határértéken túlnak is generálni, ez teljesen tőlünk függ, hogy milyen szabályokat adunk meg.

9. ADATGENERÁLÁS MÁS MÓDON

Ebben a fejezetben az előző fejezetben használt kiinduló adatsort vesszük alapul (lásd negyedik melléklet), de más úton próbálunk adatot generálni.

Egyetlen nagy DATA lépést tartalmaz a következő SAS programunk, amiben a MERGE használata nélkül oldjuk meg az adatgenerálást. Ebben minden hónapra külön készítünk adatsort, és, amikor eljünk a 6 eredetileg lemért adatsorunk napjaihoz, azokat másoljuk be az adatállományba (kornyeze.a_kieg).

```
data kornyeze.a_kieg (drop=datum i) ;

/* Az egész programra vonatkozó beállítások */
Length dat $9. ; /* definiálunk egy DAT nevű változót, ami 9 hosszú karakteres. Ebben tároljuk majd a dátumot. */
Substr(dat,6,4)='2010' ; /* Ennek a DAT dátum változónak a hatodik karaktertől 4 karakter hosszan '2010'-et írunk - hisz 01JAN2010 formátumú dátumokat készítünk */

/***** Márc 1-i adat előttiék */
If _N_ eq 1 then do ;

    /* Január 1-30. */
    substr(dat,3,3)='JAN' ;
    do i=1 to 31 ;
        substr(dat,1,2)=put(i,Z2.) ;

        ph= 5.8 + ranuni(1)*3.9 ;
        koi= 200+ ranuni(1)*750 ;
        boi = 100 + ranuni(1)*350;
        szoe = 5 + ranuni(1)*40;
        osszso = 650 + ranuni(1)*1600;
        vas = 0.1 + ranuni(1)*17.9;
        cink = 0.1 + ranuni(1)*0.08;
        nikkel = 0.05 + ranuni(1)*0.85;
        kadmium = 0.01 + ranuni(1)*0.085;
        osszkrom = 0.05 + ranuni(1)*0.9;
        rez = 0.1 + ranuni(1)*1.7;
        olom = 0.005 + ranuni(1)*0.013;
        output ;
    end ;

    /* Február 1-28. */
    substr(dat,3,3)='FEB' ;
    do i=1 to 28 ;
        substr(dat,1,2)=put(i,Z2.) ;
        ph= 5.8 + ranuni(1)*3.9 ;
        koi= 200+ ranuni(1)*750 ;
        boi = 100 + ranuni(1)*350;
        szoe = 5 + ranuni(1)*40;
```

```

    osszso = 650 + ranuni(1)*1600;
    vas = 0.1 + ranuni(1)*17.9;
    cink = 0.1 + ranuni(1)*0.08;
    nikkel = 0.05 + ranuni(1)*0.85;
    kadmium = 0.01 + ranuni(1)*0.085;
    osszkrom = 0.05 + ranuni(1)*0.9;
    rez = 0.1 + ranuni(1)*1.7;
    olom = 0.005 + ranuni(1)*0.013;
    output ;
end ;

end ;  /***** Márc 1-i adat előttiék vége */

set környeze.Ha ;

/***** _N_=1 */

if _N_ eq 1 then do ;

    /* Március 1. a file-ból */
    i=.;
    substr(dat,1,9)= put(datum,date9.) ;
    substr(dat,1,2)='01' ;
    /* Adatpótlás a létező méréseknél */
    osszso = 650 + ranuni(1)*1600;
    vas = 0.1 + ranuni(1)*17.9;
    cink = 0.1 + ranuni(1)*0.08;
    nikkel = 0.05 + ranuni(1)*0.85;
    kadmium = 0.01 + ranuni(1)*0.085;
    rez = 0.1 + ranuni(1)*1.7;
    olom = 0.005 + ranuni(1)*0.013;
    output ;

    /*További márciusi adatok 2-31. */
    substr(dat,3,3)='MAR' ;
    do i=2 to 31 ;
        substr(dat,1,2)=put(i,Z2.) ;
        ph= 5.8 + ranuni(1)*3.9 ;
        koi= 200+ ranuni(1)*750 ;
        boi = 100 + ranuni(1)*350;
        szoe = 5 + ranuni(1)*40;
        osszso = 650 + ranuni(1)*1600;
        vas = 0.1 + ranuni(1)*17.9;
        cink = 0.1 + ranuni(1)*0.08;
        nikkel = 0.05 + ranuni(1)*0.85;
        kadmium = 0.01 + ranuni(1)*0.085;
        osszkrom = 0.05 + ranuni(1)*0.9;
        rez = 0.1 + ranuni(1)*1.7;
        olom = 0.005 + ranuni(1)*0.013;
        output ;
    end ;

    /*A hiányzó áprilisi adatok 1-28. */
    substr(dat,3,3)='APR' ;
    do i=1 to 28 ;
        substr(dat,1,2)=put(i,Z2.) ;

```

```

    ph= 5.8 + ranuni(1)*3.9 ;
    koi= 200+ ranuni(1)*750 ;
    boi = 100 + ranuni(1)*350;
    szoe = 5 + ranuni(1)*40;
    osszso = 650 + ranuni(1)*1600;
    vas = 0.1 + ranuni(1)*17.9;
    cink = 0.1 + ranuni(1)*0.08;
    nikkel = 0.05 + ranuni(1)*0.85;
    kadmium = 0.01 + ranuni(1)*0.085;
    osszkrom = 0.05 + ranuni(1)*0.9;
    rez = 0.1 + ranuni(1)*1.7;
    olom = 0.005 + ranuni(1)*0.013;
    output ;
end ;

end ; /***** _N_=1 vége */

/***** _N_=2 */

If _N_ eq 2 then do ;

    /* Április 29-i adat beolvasása */
    dat = put(datum,date9.) ;
    i=.;
    /* Adatpótlás a létező méréseknél */
    nikkel = 0.05 + ranuni(1)*0.85;
    kadmium = 0.01 + ranuni(1)*0.085;
    rez = 0.1 + ranuni(1)*1.7;
    olom = 0.005 + ranuni(1)*0.013;
    output;

    /*Április 30-i adat kigenerálása*/
    i=30; /* hogy követhető legyen */
    substr(dat,1,2)='30' ;
    substr(dat,3,3)='ÁPR';
    ph= 5.8 + ranuni(1)*3.9 ;
    koi= 200+ ranuni(1)*750 ;
    boi = 100 + ranuni(1)*350;
    szoe = 5 + ranuni(1)*40;
    osszso = 650 + ranuni(1)*1600;
    vas = 0.1 + ranuni(1)*17.9;
    cink = 0.1 + ranuni(1)*0.08;
    nikkel = 0.05 + ranuni(1)*0.85;
    kadmium = 0.01 + ranuni(1)*0.085;
    osszkrom = 0.05 + ranuni(1)*0.9;
    rez = 0.1 + ranuni(1)*1.7;
    olom = 0.005 + ranuni(1)*0.013;
    output ;

    /*Májusi adatok 1-31. */
    Substr(dat,3,3)='MAJ' ;
    do i=1 to 31 ;
        substr(dat,1,2)=put(i,Z2.) ;
        ph= 5.8 + ranuni(1)*3.9 ;
        koi= 200+ ranuni(1)*750 ;
        boi = 100 + ranuni(1)*350;

```

```

    szoe = 5 + ranuni(1)*40;
    osszso = 650 + ranuni(1)*1600;
    vas = 0.1 + ranuni(1)*17.9;
    cink = 0.1 + ranuni(1)*0.08;
    nikkel = 0.05 + ranuni(1)*0.85;
    kadmium = 0.01 + ranuni(1)*0.085;
    osszkrom = 0.05 + ranuni(1)*0.9;
    rez = 0.1 + ranuni(1)*1.7;
    olom = 0.005 + ranuni(1)*0.013;
    output ;
end ;

/*Június 1-i adat létrehozása*/
i=1; /* hogy követhető legyen */
substr(dat,3,3)='JUN' ;
substr(dat,1,2)='01' ;
ph= 5.8 + ranuni(1)*3.9 ;
koi= 200+ ranuni(1)*750 ;
boi = 100 + ranuni(1)*350;
szoe = 5 + ranuni(1)*40;
osszso = 650 + ranuni(1)*1600;
vas = 0.1 + ranuni(1)*17.9;
cink = 0.1 + ranuni(1)*0.08;
nikkel = 0.05 + ranuni(1)*0.85;
kadmium = 0.01 + ranuni(1)*0.085;
osszkrom = 0.05 + ranuni(1)*0.9;
rez = 0.1 + ranuni(1)*1.7;
olom = 0.005 + ranuni(1)*0.013;
output ;

end ; /****** _N_=2 vége */

/***** _N_=3 */

If _N_ eq 3 then do ;

/*Június 2-i adat beolvasása*/
i=.;
dat = put(datum,date9.) ;
substr(dat,1,2)='02' ;
/* Adatpótlás a létező méréseknél */
nikkel = 0.05 + ranuni(1)*0.85;
kadmium = 0.01 + ranuni(1)*0.085;
rez = 0.1 + ranuni(1)*1.7;
olom = 0.005 + ranuni(1)*0.013;
output;

/*További júniusi adatok 3-30. */
Substr(dat,3,3)='JUN' ;
do i=3 to 30 ;
    substr(dat,1,2)=put(i,Z2.) ;
    ph= 5.8 + ranuni(1)*3.9 ;
    koi= 200+ ranuni(1)*750 ;
    boi = 100 + ranuni(1)*350;
    szoe = 5 + ranuni(1)*40;
    osszso = 650 + ranuni(1)*1600;

```

```

vas = 0.1 + ranuni(1)*17.9;
cink = 0.1 + ranuni(1)*0.08;
nikkel = 0.05 + ranuni(1)*0.85;
kadmium = 0.01 + ranuni(1)*0.085;
osszkrom = 0.05 + ranuni(1)*0.9;
rez = 0.1 + ranuni(1)*1.7;
olom = 0.005 + ranuni(1)*0.013;
output ;
end ;

/*Juliusi adatok generálása 1-31.*/
Substr(dat,3,3)='JUL' ;
do i=1 to 31 ;
    substr(dat,1,2)=put(i,Z2.) ;
    ph= 5.8 + ranuni(1)*3.9 ;
    koi= 200+ ranuni(1)*750 ;
    boi = 100 + ranuni(1)*350;
    szoe = 5 + ranuni(1)*40;
    osszso = 650 + ranuni(1)*1600;
    vas = 0.1 + ranuni(1)*17.9;
    cink = 0.1 + ranuni(1)*0.08;
    nikkel = 0.05 + ranuni(1)*0.85;
    kadmium = 0.01 + ranuni(1)*0.085;
    osszkrom = 0.05 + ranuni(1)*0.9;
    rez = 0.1 + ranuni(1)*1.7;
    olom = 0.005 + ranuni(1)*0.013;
    output ;
end ;

/*Augusztusi adatok generálása 1-31. */
Substr(dat,3,3)='AUG' ;
do i=1 to 31 ;
    substr(dat,1,2)=put(i,Z2.) ;
    ph= 5.8 + ranuni(1)*3.9 ;
    koi= 200+ ranuni(1)*750 ;
    boi = 100 + ranuni(1)*350;
    szoe = 5 + ranuni(1)*40;
    osszso = 650 + ranuni(1)*1600;
    vas = 0.1 + ranuni(1)*17.9;
    cink = 0.1 + ranuni(1)*0.08;
    nikkel = 0.05 + ranuni(1)*0.85;
    kadmium = 0.01 + ranuni(1)*0.085;
    osszkrom = 0.05 + ranuni(1)*0.9;
    rez = 0.1 + ranuni(1)*1.7;
    olom = 0.005 + ranuni(1)*0.013;
    output ;
end ;

/*Szeptember 1-i adatok generálása*/
Substr(dat,3,3)='SEP' ;
substr(dat,1,2)='01' ;
ph= 5.8 + ranuni(1)*3.9 ;
koi= 200+ ranuni(1)*750 ;
boi = 100 + ranuni(1)*350;
szoe = 5 + ranuni(1)*40;
osszso = 650 + ranuni(1)*1600;
vas = 0.1 + ranuni(1)*17.9;

```

```

cink = 0.1 + ranuni(1)*0.08;
nikkel = 0.05 + ranuni(1)*0.85;
kadmium = 0.01 + ranuni(1)*0.085;
osszkrom = 0.05 + ranuni(1)*0.9;
rez = 0.1 + ranuni(1)*1.7;
olom = 0.005 + ranuni(1)*0.013;
output ;

end ;      /***** _N_=3 vége */

/***** _N_=4 */

If _N_ eq 4 then do ;

    /*Szeptember 2-i adat beolvasása */
    i=.;
    substr(dat,3,3)='SEP' ;
    substr(dat,1,2)='02';
    /* Adatpótlás a létező méréseknél */
    nikkel = 0.05 + ranuni(1)*0.85;
    kadmium = 0.01 + ranuni(1)*0.085;
    rez = 0.1 + ranuni(1)*1.7;
    olom = 0.005 + ranuni(1)*0.013;
    output;

    /*További szeptemberi adatok 3-30. */
    do i=3 to 30 ;
        substr(dat,1,2)=put(i,Z2.) ;
        ph= 5.8 + ranuni(1)*3.9 ;
        koi= 200+ ranuni(1)*750 ;
        boi = 100 + ranuni(1)*350;
        szoe = 5 + ranuni(1)*40;
        osszso = 650 + ranuni(1)*1600;
        vas = 0.1 + ranuni(1)*17.9;
        cink = 0.1 + ranuni(1)*0.08;
        nikkel = 0.05 + ranuni(1)*0.85;
        kadmium = 0.01 + ranuni(1)*0.085;
        osszkrom = 0.05 + ranuni(1)*0.9;
        rez = 0.1 + ranuni(1)*1.7;
        olom = 0.005 + ranuni(1)*0.013;
        output ;
    end ;

    /*A hiányzó októberi adatok 1-17. */
    substr(dat,3,3)='OCT' ;
    do i=1 to 17 ;
        substr(dat,1,2)=put(i,Z2.) ;
        ph= 5.8 + ranuni(1)*3.9 ;
        koi= 200+ ranuni(1)*750 ;
        boi = 100 + ranuni(1)*350;
        szoe = 5 + ranuni(1)*40;
        osszso = 650 + ranuni(1)*1600;
        vas = 0.1 + ranuni(1)*17.9;
        cink = 0.1 + ranuni(1)*0.08;
        nikkel = 0.05 + ranuni(1)*0.85;
        kadmium = 0.01 + ranuni(1)*0.085;

```

```

        osszkrom = 0.05 + ranuni(1)*0.9;
        rez = 0.1 + ranuni(1)*1.7;
        olom = 0.005 + ranuni(1)*0.013;
        output ;
    end ;
end ;          /***** _N_=4 vége */

/***** _N_=5 *****/

If _N_ eq 5 then do ;

    /*Október 18-i adat beolvasása */
    i=.;
    substr(dat,3,3)='OCT' ;
    substr(dat,1,2)='18';
    /* Adatpótlás a létező méréseknél */
    nikkel = 0.05 + ranuni(1)*0.85;
    kadmium = 0.01 + ranuni(1)*0.085;
    rez = 0.1 + ranuni(1)*1.7;
    olom = 0.005 + ranuni(1)*0.013;
    output;

    /*További októberi adatok 19-31. */
    substr(dat,1,7)=datum ;
    substr(dat,3,3)='OCT' ;
    do i=19 to 31 ;
        substr(dat,1,2)=put(i,Z2.) ;
        ph= 5.8 + ranuni(1)*3.9 ;
        koi= 200+ ranuni(1)*750 ;
        boi = 100 + ranuni(1)*350;
        szoe = 5 + ranuni(1)*40;
        osszso = 650 + ranuni(1)*1600;
        vas = 0.1 + ranuni(1)*17.9;
        cink = 0.1 + ranuni(1)*0.08;
        nikkel = 0.05 + ranuni(1)*0.85;
        kadmium = 0.01 + ranuni(1)*0.085;
        osszkrom = 0.05 + ranuni(1)*0.9;
        rez = 0.1 + ranuni(1)*1.7;
        olom = 0.005 + ranuni(1)*0.013;
        output ;
    end ;

    /*A hiányzó novemberi adatok 1-30. */
    substr(dat,3,3)='NOV' ;
    do i=1 to 30 ;
        substr(dat,1,2)=put(i,Z2.) ;
        ph= 5.8 + ranuni(1)*3.9 ;
        koi= 200+ ranuni(1)*750 ;
        boi = 100 + ranuni(1)*350;
        szoe = 5 + ranuni(1)*40;
        osszso = 650 + ranuni(1)*1600;
        vas = 0.1 + ranuni(1)*17.9;
        cink = 0.1 + ranuni(1)*0.08;
        nikkel = 0.05 + ranuni(1)*0.85;
        kadmium = 0.01 + ranuni(1)*0.085;
        osszkrom = 0.05 + ranuni(1)*0.9;

```

```

    rez = 0.1 + ranuni(1)*1.7;
    olom = 0.005 + ranuni(1)*0.013;
    output ;
end ;
end ;          /***** _N_=5 vége */

/***** _N_=6 */

If _N_ eq 6 then do ;

    /*Decembet 1-i adat átvétele*/
    i=.;
    substr(dat,3,3)='DEC' ;
    substr(dat,1,2)='01';
    /* Adatpótlás a létező méréseknél */
    nikkel = 0.05 + ranuni(1)*0.85;
    kadmium = 0.01 + ranuni(1)*0.085;
    rez = 0.1 + ranuni(1)*1.7;
    olom = 0.005 + ranuni(1)*0.013;
    output;

    /*További decemberi adatok*/
    do i=2 to 31 ;
        substr(dat,1,2)=put(i,Z2.) ;
        ph= 5.8 + ranuni(1)*3.9 ;
        koi= 200+ ranuni(1)*750 ;
        boi = 100 + ranuni(1)*350;
        szoe = 5 + ranuni(1)*40;
        osszso = 650 + ranuni(1)*1600;
        vas = 0.1 + ranuni(1)*17.9;
        cink = 0.1 + ranuni(1)*0.08;
        nikkel = 0.05 + ranuni(1)*0.85;
        kadmium = 0.01 + ranuni(1)*0.085;
        osszkrom = 0.05 + ranuni(1)*0.9;
        rez = 0.1 + ranuni(1)*1.7;
        olom = 0.005 + ranuni(1)*0.013;
        output ;
    end ;

end ;          /***** _N_=6 vége */

run ;

```

Részlet a generált adatállományból:

	dat	pH	KOI	BOI	SZOE	OsszSo	Vas	Cink	Nikkel	Kadmium	OsszKrom	Rez	Olom
54	23FEB2010	6.3533833	922.849	206.5629	10.49219	1510.274	11.40367	0.12909	0.11431	0.0432	0.5956	0.688	0.0082
55	24FEB2010	8.5295266	833.7287	384.3976	42.44023	2240.133	9.182798	0.16412	0.63564	0.0399	0.67433	1.124	0.012
56	25FEB2010	8.1790078	898.2958	383.255	43.84105	1682.659	15.91549	0.1313	0.74884	0.0462	0.59549	1.694	0.0123
57	26FEB2010	8.8243955	319.1785	439.1682	40.03141	2066.123	4.087196	0.17895	0.13955	0.0345	0.33812	1.152	0.0112
58	27FEB2010	9.376474	500.3752	416.1149	30.81901	2159.467	13.62695	0.17436	0.09052	0.039	0.91778	0.603	0.0119
59	28FEB2010	8.3292621	465.8033	102.8694	6.491654	1154.13	7.57357	0.16238	0.64081	0.0778	0.18454	1.338	0.0138
60	01MAR2010	7.74	683	260	16	1752.357	11.93846	0.13354	0.69108	0.0392	0.8	1.305	0.0178
61	02MAR2010	5.9249425	369.4829	220.3217	21.85615	1709.997	1.729862	0.1035	0.85465	0.0652	0.30023	0.82	0.0102
62	03MAR2010	6.3155025	225.6757	198.6237	7.525876	1645.835	12.44497	0.12264	0.44799	0.0942	0.79471	0.901	0.0135
63	04MAR2010	7.6346969	675.2806	382.9156	18.04611	1166.317	2.803006	0.10164	0.65485	0.0697	0.2068	1.205	0.0151
64	05MAR2010	8.3603527	722.8046	331.3352	29.61377	1900.836	6.245668	0.10922	0.47435	0.0241	0.49634	0.425	0.0056
65	06MAR2010	8.3861703	572.3333	194.5993	18.42403	1022.366	15.28092	0.12217	0.1528	0.0459	0.63931	0.123	0.009
66	07MAR2010	5.991835	508.5828	447.1611	36.97578	947.2337	8.800857	0.11955	0.39578	0.0768	0.62776	1.179	0.0129
67	08MAR2010	5.9265507	267.3746	283.9482	14.45456	802.5069	10.25776	0.1579	0.88429	0.0209	0.21089	0.384	0.0139
68	09MAR2010	8.0502411	914.4748	210.2589	33.48622	1527.719	17.44553	0.15664	0.63232	0.0795	0.90645	1.565	0.0052
69	10MAR2010	9.5482791	912.2241	326.0475	29.84601	1280.562	16.05585	0.15612	0.37089	0.0467	0.63279	0.429	0.0067
70	11MAR2010	6.4207761	769.8473	249.6976	41.94281	1847.82	15.63262	0.14506	0.36912	0.0474	0.80453	0.297	0.0051

35. ábra: 365 napi generált szennyvíztisztítással kapcsolatos adatok

Itt is láthatjuk amit az előző fejezetben is tapasztaltunk, hogy a 2010 március 1-jei ('01MAR2010' formátumban megadva) adatokból az eredetieket használtuk fel, míg a hiányzó változóknak generáltunk értéket.

10. ADATGENERÁLÁS, GRAFIKONKÉSZÍTÉS, TÁBLÁZATOS MEGJELENÍTÉS

Az alábbi fejezetben a korábbiakban tanult adatgenerálási és grafikus megjelenítési eszközöket alkalmazzuk egy újabb gyakorlati probléma kapcsán, majd kiegészítjük táblázat készítésével.

Probléma: Egy sertéstelep nem bocsátott rendelkezésre semmiféle számadatot. Amit tudunk, hogy 1975 óta bocsát ki hígtrágyát szántóföldekre. Az is ismert, hogy meghaladja a nitrát szint a határértéket.

```
/* Adatgenerálás
   1951 - 2008 -ig félévente egy adat
   határérték: 50 mg/l
   1975-től átlépve a határértéket, 80-90 mg/l közötti értékekkel */
data adatgener (drop=i); /* Az i változót csak számlálásra használjuk, feleslegesen nem tartjuk meg, töröljük. */
  do i=1951 to 1974.5 by 0.5; /* 1951-től évente két adatot generálunk, vagyis felesével lépünk a DO ciklusban */
    datum = int(i); /* A dátum (év) a számlálásra használt "i" változó egészrészese. */
    adat = ranuni(1)*50; /* Az "adat" nevű változóban generálunk 50-ig pozitív számot. */
    output;
  end; /* DO ciklus vége */

  do i=1975 to 2008 by 0.5; /* 1975-től a határértéket túllépő értékeket generálunk, ezért külön vesszük ezt. */
    datum = int(i);
    adat = 80 + ranuni(1)*10; /* A generált adat minimum 80 és maximum 90 (80+10) értéket vehet fel. */
    output;
  end; /* DO ciklus vége */
run; /* DATA step vége */
```

	datum	adat
41	1971	8.326305
42	1971	43.55524
43	1972	14.93948
44	1972	46.73209
45	1973	45.02354
46	1973	28.43917
47	1974	2.477283
48	1974	6.779413
49	1975	85.11318
50	1975	84.33205
51	1976	81.76106
52	1976	86.65036
53	1977	84.04819
54	1977	81.24549
55	1978	84.53487
56	1978	81.99553
57	1979	85.74837

36. ábra: Generált adatok két változóra: év és adat

```

/* Vonaldiagramon szemléltetve */
goptions reset=all border nodash; /* Grafikus alapbeállítások */

title 'Generált féléves adatok 1951-2008 között'; /* Diagram címe */

/* Egyik tengely beállításai: címke legyen „Dátum”, színe fekete, tengely
alosztások ne legyenek */
axis1 label=("Dátum") color=black minor=none;
/* Másik tengely beállításai: címke legyen „mg/l”, színe fekete, 0-tól 90-ig
10-esével legyenek a főosztások */
axis2 label=("mg/l") color=black order=(0 to 90 by 10);

/* Jelmagyarázat beállításai: a keret legyen fekete, címkéje: Jelmagyarázat,
az ábrán fent (a diagram felett) középen helyezkedjen el */
legend1 cborder=black label=('Jelmagyarázat:') position=(top center);

symbol1 interpol=join line=1 ci=blue width=2;

proc gplot data=adatgener;
  plot adat*datum=1
  / legend=legend1 /* Jelmagyarázat használata a fentebb definiált
"legend1" beállításaiával */
  haxis=axis1 vaxis=axis2 /* A horizontális tengely beállítása legyen az
axis1 szerinti, a vertikálisé az axis2 szerinti */

```

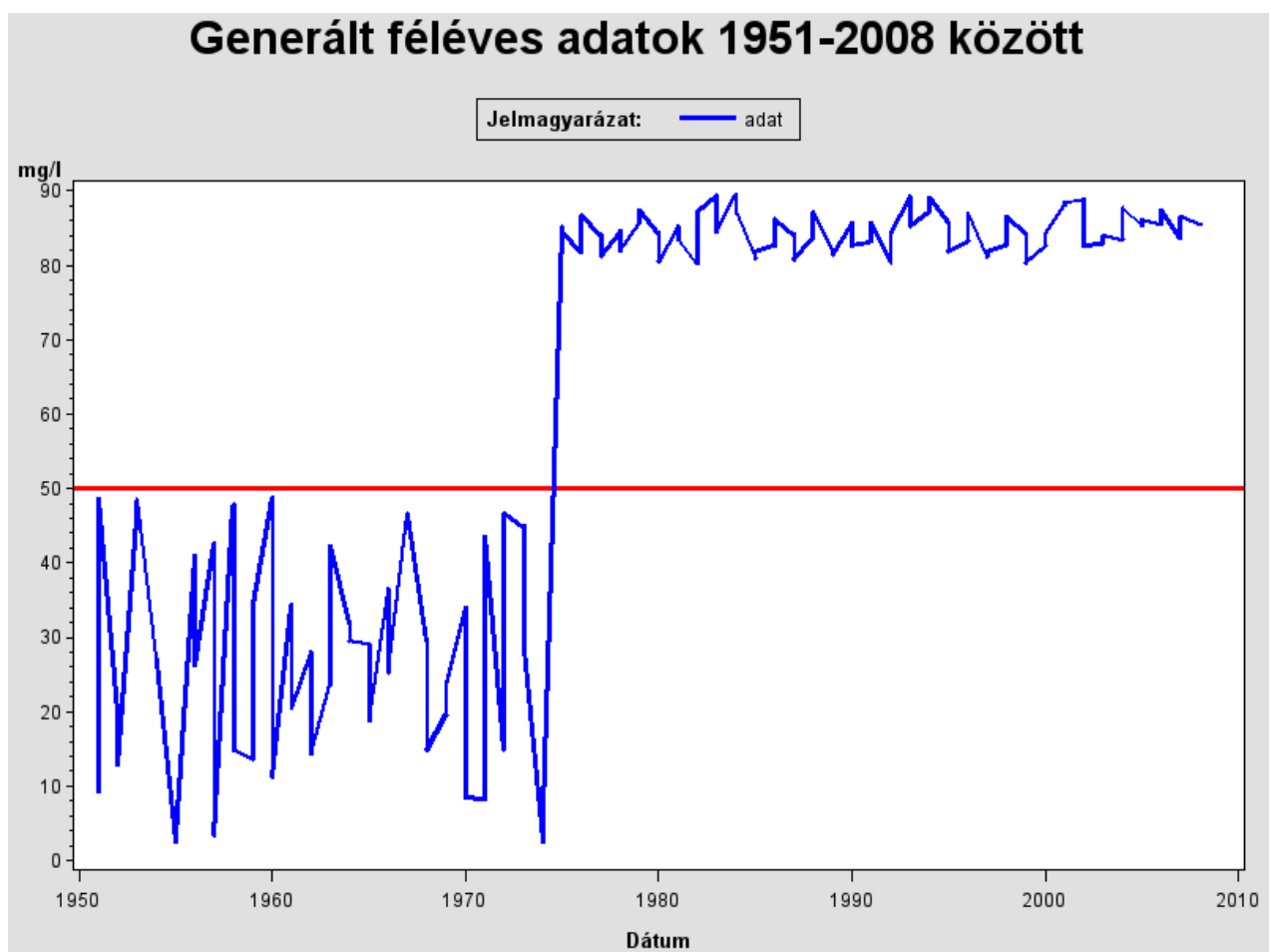
```

vref=50 cvref=red wvref=2    /* Jelöljük az 50-es határértéket egy piros
referencia vonallal */
cframe=white caxis=black;    /* Háttér fehér, tengely fekete */
run;
quit;

options reset=all;          /* Grafikus alapbeállítások visszaállítása */

```

A futtatás eredményét a következő ábra szemlélteti. Itt jól látszódik, hogy nagy az „ugrás” az 1975-ös határérték túllépéskor. Ezen lehet finomítani, és időszakonként, „finomabban” emelve is lehet adatokat generálni, ahogy azt látni fogjuk.



37. ábra: Generált adatok grafikus megjelenítése

```

/* Táblázatos megjelenítéshez készítünk egy csoportosító változót a dátumból
(év) */
data adatgener2;
set adatgener;

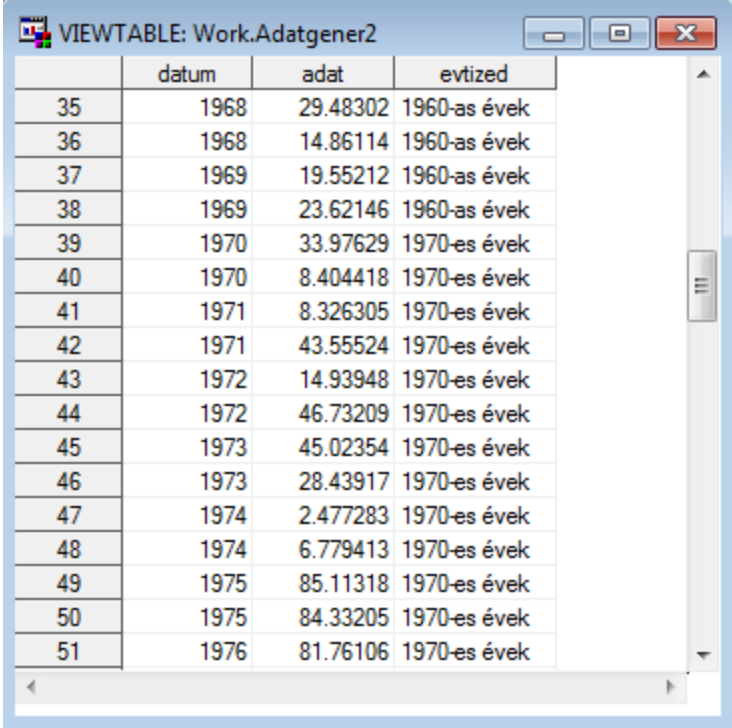
```

```

/* új, csoportosító változó bevezetése az évtizedekre */
if datum >= 1950 and datum < 1960 then evtized='1950-es évek';
else if datum >= 1960 and datum < 1970 then evtized='1960-as évek';
else if datum >= 1970 and datum < 1980 then evtized='1970-es évek';
else if datum >= 1980 and datum < 1990 then evtized='1980-as évek';
else if datum >= 1990 and datum < 2000 then evtized='1990-es évek';
else if datum >= 2000 and datum < 2010 then evtized='2000-es évek';

run;

```



	datum	adat	evtized
35	1968	29.48302	1960-as évek
36	1968	14.86114	1960-as évek
37	1969	19.55212	1960-as évek
38	1969	23.62146	1960-as évek
39	1970	33.97629	1970-es évek
40	1970	8.404418	1970-es évek
41	1971	8.326305	1970-es évek
42	1971	43.55524	1970-es évek
43	1972	14.93948	1970-es évek
44	1972	46.73209	1970-es évek
45	1973	45.02354	1970-es évek
46	1973	28.43917	1970-es évek
47	1974	2.477283	1970-es évek
48	1974	6.779413	1970-es évek
49	1975	85.11318	1970-es évek
50	1975	84.33205	1970-es évek
51	1976	81.76106	1970-es évek

38. ábra: Évtized csoportosító változó készítése a generált adatállományba

```

/* Táblázatos megjelenítés */
title 'Átlagos mért értékek évtizedenként 1951 és 2008 között mért féléves
adatokból'; /* táblázat címe */
proc tabulate data=adatgener2; /* az adatgener2 nevű adatállományon dolgo-
zunk, mert ebben készítettünk "evtized" változót */
class evtized; /* csoportosító karakteres változó az "evtized" */
var adat; /* érték változó az "adat" */
table evtized=' ' all='Összesen', /* a sorokban az "evtized" szerint il-
letve összesítve jelenítjük meg */
adat='Átlagos mért érték'*mean; /* az átlagos (mean) adatokat */
keylabel mean=' '; /* a "mean" kulcsszót nem szeretnénk feltüntetni a
táblázatban, ezt már magyaráztuk az "adat" elnevezésekor, illetve a címben
is, ezért üres értéket adunk neki */

```

`run;`

A lefuttatott TABULATE eljárás eredményét a következő táblázat mutatja:

Átlagos mért értékek évtizedenként 1951 és 2008 között mért féléves adatokból

	Átlagos mért érték
1950-es évek	26.05
1960-as évek	28.75
1970-es évek	54.07
1980-as évek	84.10
1990-es évek	84.38
2000-es évek	85.50
Összesen	60.42

Most nézzünk egy olyan adatgenerálást, ahol folyamatosan növekedő adatsort készítünk, vagyis sűrűbben adunk meg finomabb intervallumokat.

```
data adatgener3 (drop=i);  
  do i=1951 to 1955 by 0.5;  
    datum = int(i);  
    adat = 30 + ranuni(1)*5;  
    output;  
  end;  
  
  do i=1955 to 1960 by 0.5;  
    datum = int(i);  
    adat = 35 + ranuni(1)*5;  
    output;  
  end;  
  
  do i=1960.5 to 1965 by 0.5;  
    datum = int(i);  
    adat = 40 + ranuni(1)*5;
```

```

    output;
end;

do i=1965.5 to 1970 by 0.5;
    datum = int(i);
    adat = 45 + ranuni(1)*5;
    output;
end;

do i=1970.5 to 1975 by 0.5;
    datum = int(i);
    adat = 50 + ranuni(1)*5;
    output;
end;

do i=1975.5 to 1980 by 0.5;
    datum = int(i);
    adat = 55 + ranuni(1)*5;
    output;
end;

do i=1980.5 to 1985 by 0.5;
    datum = int(i);
    adat = 60 + ranuni(1)*5;
    output;
end;

do i=1985.5 to 1990 by 0.5;
    datum = int(i);
    adat = 65 + ranuni(1)*5;
    output;
end;

do i=1990.5 to 1995 by 0.5;
    datum = int(i);
    adat = 70 + ranuni(1)*4;
    output;
end;

do i=1995.5 to 2000 by 0.5;
    datum = int(i);
    adat = 75 + ranuni(1)*3;
    output;
end;

do i=2000.5 to 2008 by 0.5;
    datum = int(i);
    adat = 79 + ranuni(1)*2;
    output;
end;
run;

/* Vonaldiagramon szemléltetve */
options reset=all border nodash;

title 'Generált fél éves adatok 1951-2008 között';

```

```

axis1 label=("Dátum") color=black minor=none;
axis2 label=("mg/l") color=black order=(0 to 90 by 10);

legend1 cborder=black label=('Jelmagyarázat:') position=(top center);

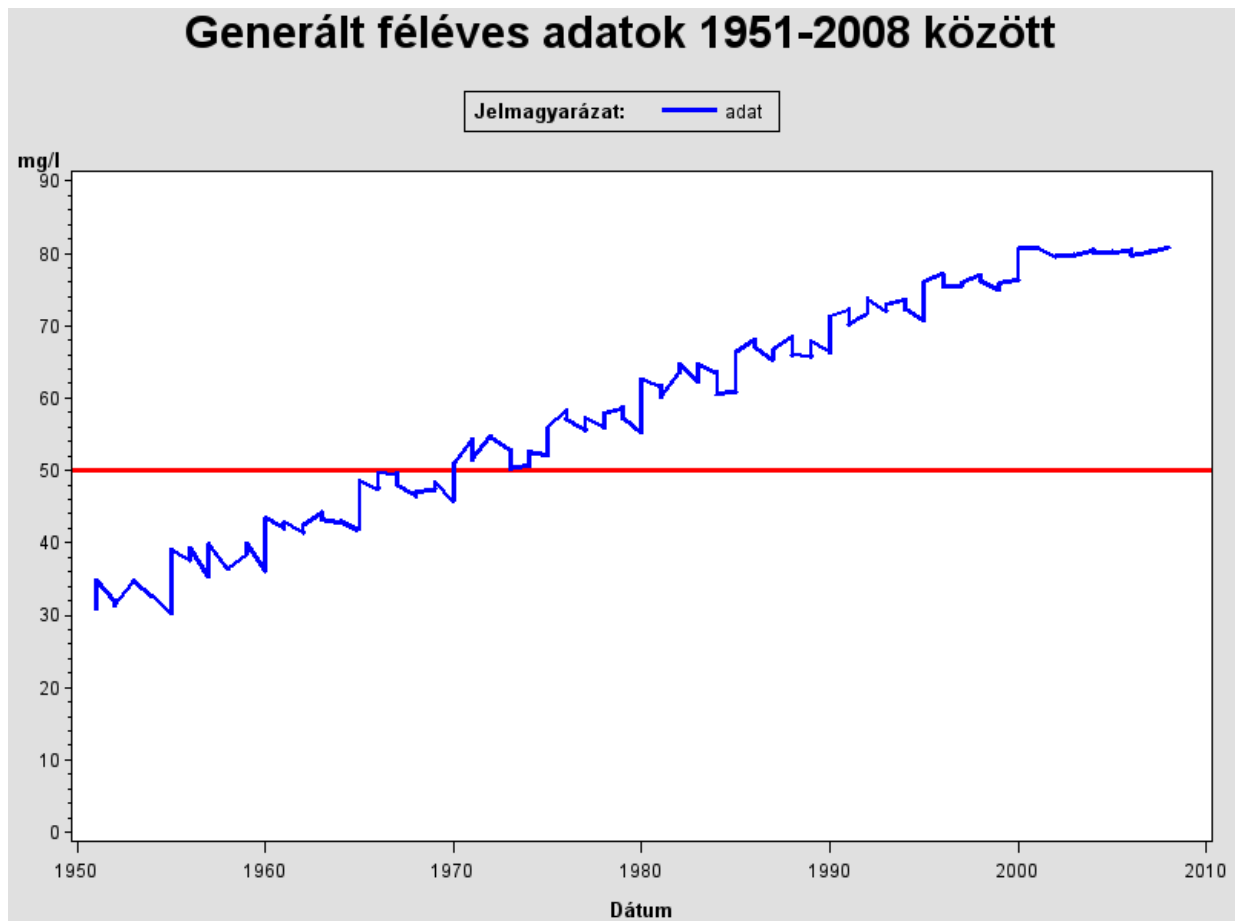
symbol1 interpol=join line=1 ci=blue width=2;

proc gplot data=adatgener3;
  plot adat*datum=1
  / legend=legend1
    haxis=axis1 vaxis=axis2
    vref=50 cvref=red wvref=2
    cframe=white caxis=black;
run;
quit;

goptions reset=all;

```

Az eredmény grafikont a következő ábra szemlélteti:



39. ábra: Folyamatosnak ható generált adatsor

11. VALÓSZÍNŰÉGSZÁMÍTÁS ÉS ALAPSTATISZTIKÁK KOCKÁZATBECSLÉSHEZ

A valószínűségszámítással és statisztikai módszerekkel kapcsolatosan találhatunk néhány hasznos információt az alábbi oldalon: <http://www.biostat.hu/biostat/indit1.asp?p=szotar2&k=123> (ezt érdemes olvasgatni, mert jó, rövid, de tömör statisztikai leírások vannak benne!)

Valószínűségszámítás

Valamely esemény **kockázata** tehát az adott esemény bekövetkezésének valószínűségét jelenti. A 8-9. fejezet adatain alapulva két mutatószámot lehet érdekes számítani: tapasztalati valószínűség és esetleg esélyérték. (Persze van több is, de a tapasztalati valószínűséget érdemes lehet. Az esélyhányados két faktor esetén mondhat valamit, annak itt nincs értelme.)

Relatív gyakoriság = Tapasztalati valószínűség = (bekövetkezések száma) / (összes lehetséges esemény száma).

P1. van 365 mért érték egy változóra (mondjuk cink), és ezekből 15 lépi túl a megadott határértéket. Jelen esetben a határérték túllépését vizsgáljuk, a megfigyelt 365 esemény függvényében/alapján. Ekkor $15/365=0,041=4,1\%$ a tapasztalati valószínűség, ami annyit mond, hogy a megfigyelt 365 adatnak 4,1%-a lépi túl a határértéket. (Ilyet számláltunk a NO_x-os feladat kapcsán a k75 és k47 változókban.) A valószínűség jele P az angol Probability szóból. Értéke 0 és 1 közötti lehet. Ha pont 0-val egyenlő, akkor lehetetlen az esemény, ha 1, akkor biztosan bekövetkezik.

Esélyérték = esély = O = odds(E) = odds(esemény) = $P(\text{esemény}) / (1 - P(\text{esemény}))$ = (esemény bekövetkezésének valószínűsége) / (esemény nem bekövetkezésének valószínűsége)

Ez megmutatja, hogy hányszor akkora a valószínűsége annak, hogy bekövetkezik egy esemény, mint annak, hogy nem következik be.

P1. Folytassuk az előző példát. Annak valószínűsége, hogy a mért cink érték határérték feletti a megfigyelt 365 adatban 0,041. Mivel a teljes valószínűség, ami mindig 1-gyel egyenlő, a bekövetkezés valószínűségének és a nem bekövetkezés valószínűségének összege (hiszen egy esemény vagy bekövetke-

zik, vagy nem), ezért most annak valószínűsége, hogy a mért cink érték nem határérték feletti: $1 - 0,041 = 0,959$.

Ezért odds(cink mért értéke határérték feletti) = $0,041 / 0,959 \approx 0,043$ (kerekítéssel, elég 3 tizedes jegyet megadni).

Vagyis kb. 4,3% az esélye annak, hogy a cink mért értéke határérték feletti, mint, hogy nem határérték feletti érték.

Az esélyérték 0 és végtelen közötti értékeket vehet fel (nem csak 0 és 1 közöttieket, mint a valószínűség az előbb). Ha pont 0, akkor az a lehetetlen eseményt tükrözi. Ha 1-gyel egyenlő, akkor azonos eséllyel fordul elő a bekövetkezés és a nem bekövetkezés, az igen, vagy nem. Végtelen esetén beszélünk biztos eseményről.

Futtathatunk még alap statisztikákat, mint azt már korábban láthattuk (4-5. fejezetek):

```
proc means data=kornyeze.ha_kieg;  
  var ph kadmium vas;  
run;
```

The MEANS Procedure

Variable	Label	N	Mean	Std Dev	Minimum	Maximum
pH	pH	365	7.8075455	1.1540308	5.8083016	9.6971480
Kadmium	Kadmium	365	0.0526640	0.0254661	0.0102037	0.0949114
Vas	Vas	365	9.2166950	6.4939758	0.1705047	85.0000000

N: egyedszám

Mean: átlag: átlagos érték

Std Dev: szórás: átlagtól való átlagos eltérés

Minimum: legkisebb mért érték

Maximum: Legnagyobb mért érték

Bővebb statisztikát számol:

```
proc univariate data=kornyeze.ha_kieg;  
  var ph kadmium vas;  
run;
```

A pH változóra (csak értékváltozóra van értelme számítani, csoportosító változóra, mint pl. nem (férfi, nő), vagy szennyezett-e (nem, kicsit, közepesen, nagyon) arra nincs!) az univariate outputja:

The UNIVARIATE Procedure

Variable: pH (pH)

Moments			
N	365	Sum Weights	365
Mean	7.80754549	Sum Observations	2849.7541
Std Deviation	1.1540308	Variance	1.33178709
Skewness	-0.027497	Kurtosis	-1.186716
Uncorrected SS	22734.3553	Corrected SS	484.770502
Coeff Variation	14.780968	Std Error Mean	0.06040473

Basic Statistical Measures			
Location		Variability	
Mean	7.807545	Std Deviation	1.15403
Median	7.812654	Variance	1.33179
Mode	7.520000	Range	3.88885
		Interquartile Range	2.00082

Tests for Location: $\mu_0=0$				
Test	Statistic		p Value	
Student's t	t	129.2539	Pr > t 	<.0001
Sign	M	182.5	Pr >= M 	<.0001

Tests for Location: $\mu_0=0$				
Test	Statistic		p Value	
Signed Rank	S	33397.5	Pr $\geq S $	<.0001

Quantiles (Definition 5)	
Quantile	Estimate
100% Max	9.69715
99%	9.68004
95%	9.54901
90%	9.42610
75% Q3	8.79860
50% Median	7.81265
25% Q1	6.79778
10%	6.21181
5%	5.97210
1%	5.82361
0% Min	5.80830

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
5.80830	292	9.67857	173
5.81538	78	9.68004	287

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
5.82016	87	9.68005	88
5.82361	106	9.68259	303
5.82641	211	9.69715	321

Itt lehet a különféle középértékek (medián, módusz, átlag), kvantilisek (kvartilisek, percentilisek) kiugró értékek (extreme observations) közül felhasználni néhányat az adatelemzésbe (ezt mindenki a maga témája szerint tudja értelmezni).

Korrelációs számítás

Akkor használjuk a korrelációs számítást, ha két változó között kapcsolat szorosságot van értelme számolni (pl. érdekes lehet, és van értelme figyelni, hogy a pH változásával (növekedésével vagy csökkenésével) hogyan változik a kadmium vagy vas mért értéke). Persze, ha generáltak az adataink, akkor ennek nagy eredménye nem lesz. De érdemes tudni, hogy ilyet is lehet futtatni.

```
proc corr data=kornyeze.ha_kieg;
  var ph kadmium vas;
run;
```

Pearson Correlation Coefficients, N = 365 Prob > r under H0: Rho=0			
	pH	Kadmium	Vas
pH	1.00000	0.02883	-0.03200
pH		0.5830	0.5422

Pearson Correlation Coefficients, N = 365 Prob > r under H0: Rho=0			
	pH	Kadmium	Vas
Kadmium	0.02883	1.00000	0.07336
Kadmium	0.5830		0.1619
Vas	-	0.07336	1.00000
Vas	0.03200	0.1619	
	0.5422		

Output értelmezése:

A Pearson korrelációs együttható két változó közötti lineáris kapcsolatot vizsgál. -1 és 1 közötti értéket vehet fel.

Ha -1, akkor azt jelenti, hogy negatív teljes függvényszerű kapcsolat van a két megfigyelt változó között (tehát, ha jelen esetben a pH és a vas között -1 lenne, akkor azt mondhatnánk, hogy a pH és a vas fordítottan arányos). Ha +1 a korrelációs együttható, akkor pozitív teljes függvényszerű kapcsolat van a két változó között, vagyis az egyik változására, ugyanannyival nő a másik is (egyenest arányosság: teljes lineáris kapcsolat).

Ha 0, akkor azt mondjuk, hogy nem mutatható ki a mintából (vizsgált egyedekből, adatokból, most a 365 adatból), hogy lenne a két változó között lineáris kapcsolat. Minél inkább közelít az érték abszolút értékben egyhez, annál erősebb, szorosabb a kapcsolat a két változó között. 0.5-0.6 közepes érték, de a 0.7-0.8 és előlött már erősnek számít. Nyilván függ a változó értelmezésétől, és magától az adatoktól.

Jelen esetben ugyanis generált adatokról beszélünk, melyek között nem állítottunk fel összefüggést, ezért is vannak igencsak 0-hoz közeli értékek.

A korrelációs együttható alatti érték a táblázatban a 0 és 1 között értelmezett szignifikancia szint (hiba-faktor). Mert minden statisztikában számolnunk kell valamennyi hibalehetőséggel. Ez közzismerten 5% szokott lenni, amit megengedünk a mintában, ami mellett számolva következtetést vonhatunk le egy adott sokaságra nézve. (A mi mintánk a 365 db szám, a sokaság lehet több év adata is, vagy akár napi

több adat is. Vagy lehet mi egy telepen mértünk, de a teljes sokaság, vagyis, amire következtetést szeretnénk levonni ezen telephely adataiból, az egy megye összes telephelyén, ugyanilyen körülmények között mért értékek.)

Szóval, a mi esetünkben mind a 3 szignifikancia szint 16% feletti, amire azt mondhatjuk, hogy a kapott korrelációs értékünk 5%-os szinten nem szignifikáns (nem jelentős az eltérés attól a feltevéstől, hogy nincs lineáris kapcsolat a két változó között).

A főatlóban értelemszerűen csupa 1-es van, hiszen önmagával erősen korrelál minden változó (amennyire nő a ph, ugyanannyira nő a ph!).

A főátlóra pedig szimmetrikus ez a korrelációs táblázat, hiszen ugyanaz az érték a ph és vas közötti korrelációra, mint a vas és a ph közötti korrelációra: -0.032 ($p=0.54 \rightarrow$ nem szignifikáns). (A szignifikancia szintet a p-érték adja itt meg, ami a statisztikai próba (most Pearson korreláció) megbízhatóságát mutatja.)

12. TÖBB VÁLTOZÓ ADATAINAK MEGJELENÍTÉSE EGY KÖZÖS DIAGRAMON

Az alábbi fejezetben három megoldást láthatunk egy az ötödik mellékletben csatolt adatállomány grafikus elemzésére.

Talajvíz kutakban mértek nitrát koncentrációt mg/l mértékegységgel.

12.1 1. MEGOLDÁS

Kiinduló adatállomány: félévente öt kúton mért nitrát koncentráció 2005 és 2010 között

	datum	Kut1	kut2	kut3	kut4	kut5
1	05.01.01.	0	0	1.32	13.39	0
2	05.06.12.	0	3.7	0	19.1	0
3	06.01.01.	0	0	1.22	12.32	0
4	06.06.12.	0	0	12.2	0	0
5	07.01.01.	0	1.33	13.39	0	0
6	07.06.12.	0	1.22	12.32	0	0
7	08.01.01.	0	3.7	0	19.1	0
8	08.06.12.	0	0	1.1	0.9	0
9	09.01.01.	0	3.7	0	9.1	0
10	09.06.12.	0	1.22	20.2	15.3	0
11	10.01.01.	0	1.23	12.34	0	0
12	10.06.12.	0	3.7	0	19.1	0

40. ábra: Öt talajvíz kútnál mért nitrát koncentráció (mg/l) 2005 és 2010 között félévente

A GPOLT szintaxisa:

```
PROC GPLOT DATA = SAS-data-set;
```

```
    PLOT y-változó * x-változó;
```

```
    RUN;
```

```
    QUIT;
```

X: független változó

Y: függő változó

Azonban alap esetben a gplot csak 2 változót képes megjeleníteni.

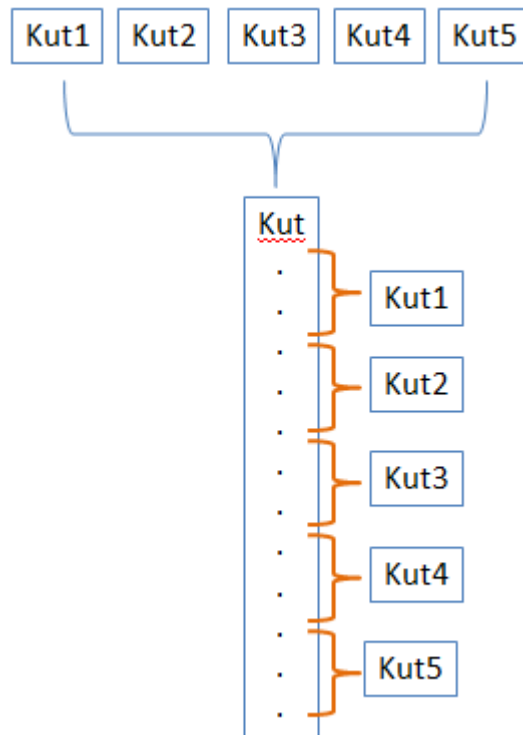
Ha egy harmadik osztályozó változó szerint szeretnénk megjeleníteni az adatokat, akkor az alábbi szintaxist kell alkalmazni.

```
PROC GPLOT DATA = SAS-data-set;  
    PLOT y-változó * x-változó = osztályozó_változó;  
RUN;  
QUIT;
```

A fenti adatállomány esetén ez nem alkalmazható, mert nincs osztályozó változó. Gondoljunk csak bele, hogy melyik változó lehet osztályozó. A „dátum” változó nem alkalmas, mert annak minden értéke különböző egy változó esetén.

Y-változó: a dátum

X-változó: Képeznünk kell egy numerikus változót („Kut” néven), ami nem más, mint a „Kut1”-„Kut5” változók értékei. (lásd 31. ábra)



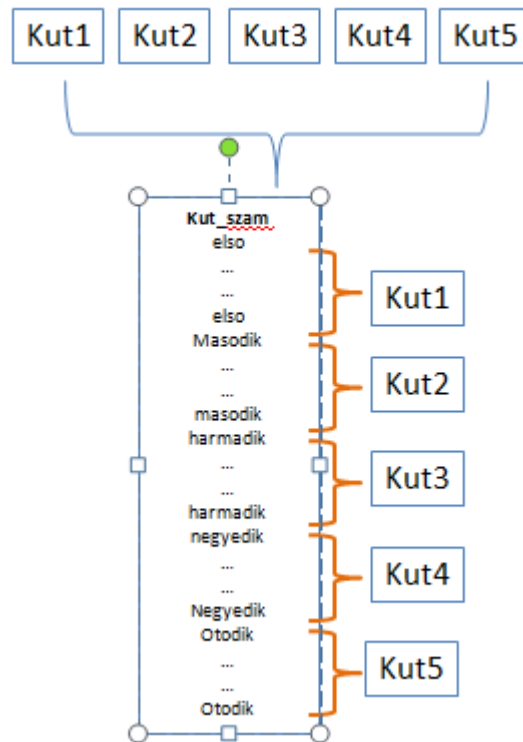
41. ábra: Kut változó generálása

Ezzel megkonstruáltuk az X-változót.

A harmadik változót azonban nekünk kell megszerkesztenünk úgy, hogy osztályozni lehessen a kutakat.

Ezt a változót úgy kell elképzelni, hogy a SAS olvassa a sorokat az adott sorhoz tartozó további értékeket. Ennek a változónak minden különböző értéke egy grafikont fog eredményezni.

Öt kút értéke áll rendelkezésre, így öt különböző grafikont kell ábrázolnunk. Ennek a változónak csak annyi a szerepe, hogy jelzi, melyik sorok melyik kúthoz tartoznak.



42. ábra: `Kut_szam` változó generálása

SAS kód az első táblához:

```
data otkut1;  
  set kornyeze.otkut(keep=kut1 datum);  
  length kutszam $8.;  
  kutszam='első';  
  rename kut1=kut;  
run;
```

Eredmény:

	datum	kut	kutszam
1	05.01.01.	0	első
2	05.06.12.	0	első
3	06.01.01.	0	első
4	06.06.12.	0	első
5	07.01.01.	0	első
6	07.06.12.	0	első
7	08.01.01.	0	első
8	08.06.12.	0	első
9	09.01.01.	0	első
10	09.06.12.	0	első
11	10.01.01.	0	első
12	10.06.12.	0	első

43. ábra: Az első kút adatai külön adatállományban: kornyeze.otkut1

Az előző kódot módosítani kell és le kell generálni a további négy kút tábláját.

```
data otkut2;  
  set kornyeze.otkut(keep=kut2 datum);  
  kutszam='második';  
  rename kut2=kut;  
run;
```

```
data otkut3;  
  set kornyeze.otkut(keep=kut3 datum);  
  kutszam='harmadik';  
  rename kut3=kut;  
run;
```

```
data otkut4;  
  set kornyeze.otkut(keep=kut4 datum);  
  kutszam='negyedik';  
  rename kut4=kut;  
run;
```

```
data otkut5;  
  set kornyeze.otkut(keep=kut5 datum);  
  kutszam='ötödik';  
  rename kut5=kut;  
run;
```

Ezek után össze kell fűzni az öt táblát.

```
data vegso;  
  set otkut1  
      otkut2  
      otkut3  
      otkut4  
      otkut5;  
run;
```

	datum	kut	kutszam
1	05.01.01.		0 elso
2	05.06.12.		0 elso
3	06.01.01.		0 elso
4	06.06.12.		0 elso
5	07.01.01.		0 elso
6	07.06.12.		0 elso
7	08.01.01.		0 elso
8	08.06.12.		0 elso
9	09.01.01.		0 elso
10	09.06.12.		0 elso
11	10.01.01.		0 elso
12	10.06.12.		0 elso
13	05.01.01.		0 masodik
14	05.06.12.		3.7 masodik
15	06.01.01.		0 masodik
16	06.06.12.		0 masodik
17	07.01.01.		1.33 masodik

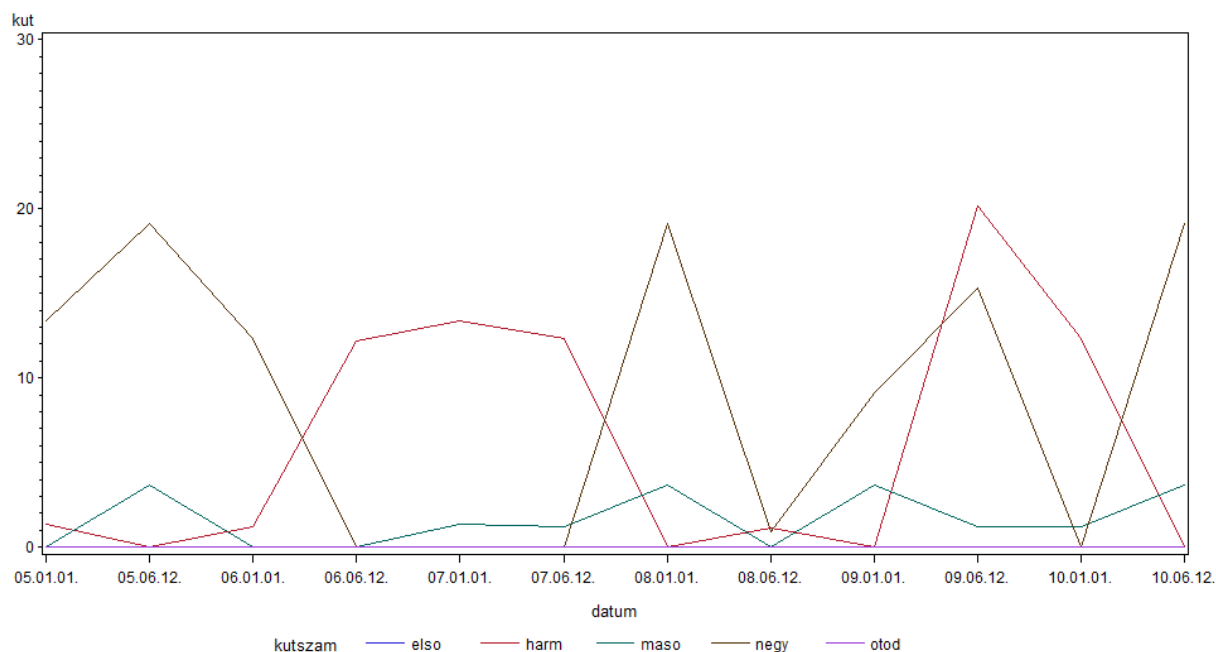
44. ábra: Végző tábla részlete

Ezt a táblát már könnyedén tudjuk ábrázolni.

```
goptions reset=all; /* Minden grafikus elem visszaállítása */  
symbol1 i=join; /* A grafikonok görbéjének a stílusát lehet megadni */  
proc gplot data=vegso;  
  plot kut * datum = kutszam;  
  title 'Teszt grafikon';  
run;  
quit;
```

Eredmény:

Teszt grafikon



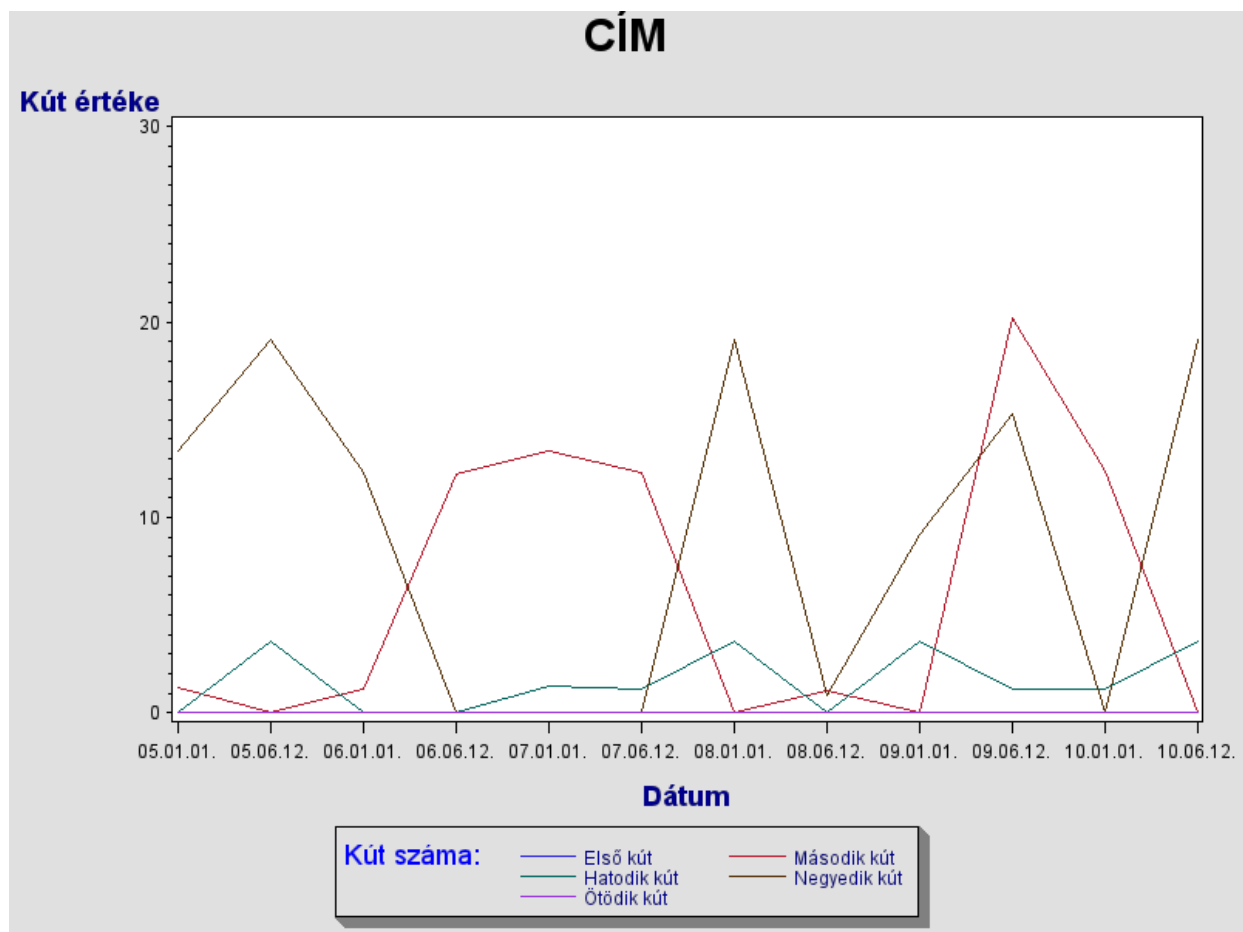
45. ábra: Teszt grafikon

Grafikon formázása

```
/* formázott diagram */
goptions reset=all;

symbol1 i=join;
axis1 label=(color=darkblue height=1.5 'Dátum');
axis2 label=(color=darkblue height=1.5 'Kút értéke');
legend1 frame cblock=gray down=3
    label=(color=blue font='Arial' height=1.5 'Kút száma:')
    value=(color=darkblue t=1 'Első kút' t=2 'Második kút' t=3 'Hatodik
kút' t=4 'Negyedik kút'
    t=5 'Ötödik kút');
proc gplot data=vegso;
    plot kut * datum = kutszam /
        legend=legend1
        haxis=axis1
        vaxis=axis2;
    title 'CÍM';
run;
quit;
```

Eredmény:



46. ábra: Formázott grafikon

12.2 2. MEGOLDÁS

Ebben az esetben nincs szükség a tábla átalakítására. Ilyenkor az adatokat idősorosan jelenítjük meg. A `patternid=t`, azért kell megadni, mert különben a SAS minden grafikon vonalát külön stílusként jeleníti meg.

```
/* 2. megoldás: nem összesített váltzóban, hanem az eredeti 5 változó fel-  
használásával */  
goption reset=all;
```

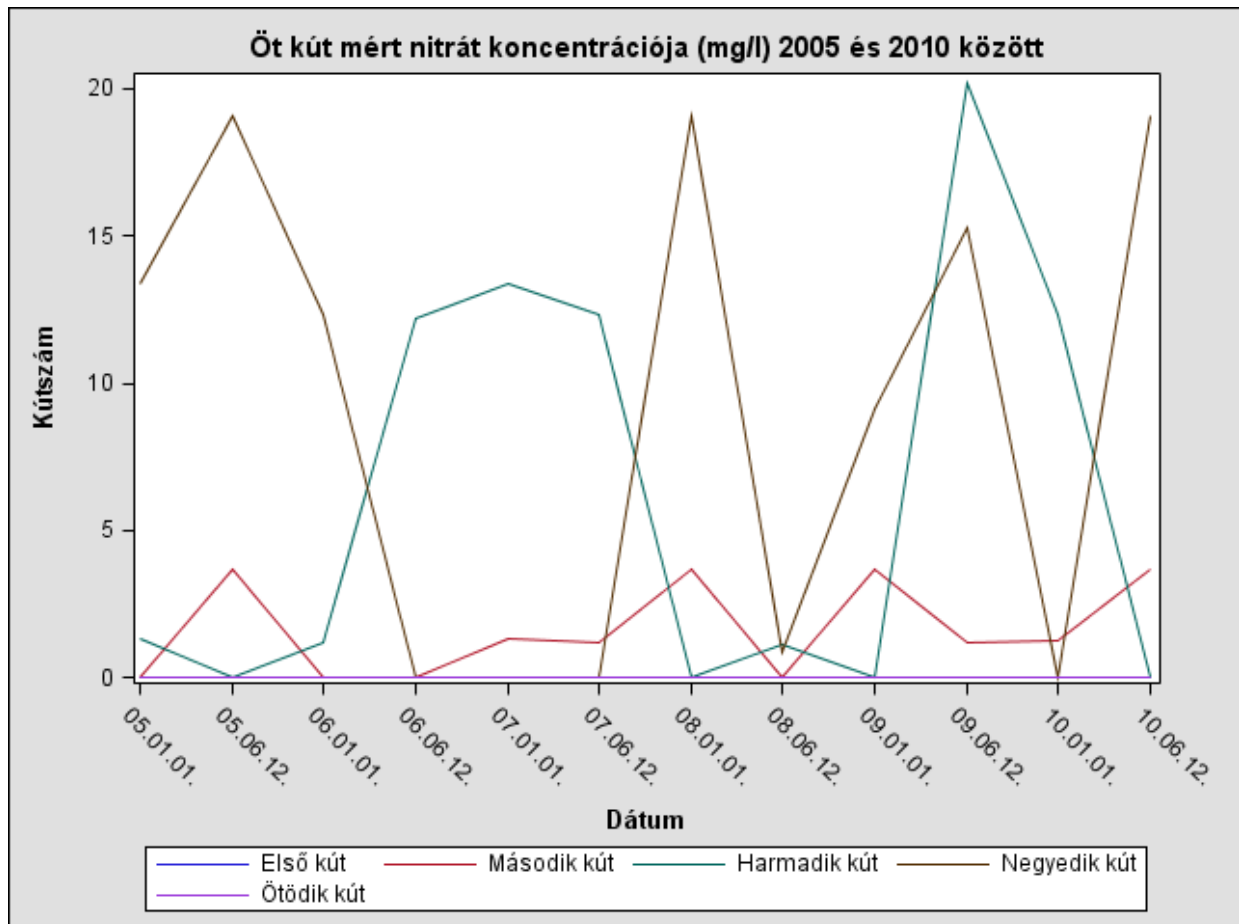
```
proc sgplot data=kornyeze.otkut;  
  series y=kut1 x=datum / legendlabel="Első kút";  
  series y=kut2 x=datum / LINEATTRS=(pattern=solid)  
                           legendlabel="Második kút";  
  series y=kut3 x=datum / LINEATTRS=(pattern=solid)  
                           legendlabel="Harmadik kút";
```

```

series y=kut4 x=datum / LINEATTRS=(pattern=solid)
                        legendlabel="Negyedik kút";
series y=kut5 x=datum / LINEATTRS=(pattern=solid) legendlabel="Ötödik kút";
label datum='Dátum' kut1='Kútszám';
title 'Öt kút mért nitrát koncentrációja (mg/l) 2005 és 2010 között';
run;
quit;

```

Eredmény:



47. ábra: Az 5 kút vonaldiagramja idősoros megjelenítéssel

12.3 3. MEGOLDÁS

A grafikonokat külön-külön jelenítjük meg és összefogjuk egy közös panelba.

```

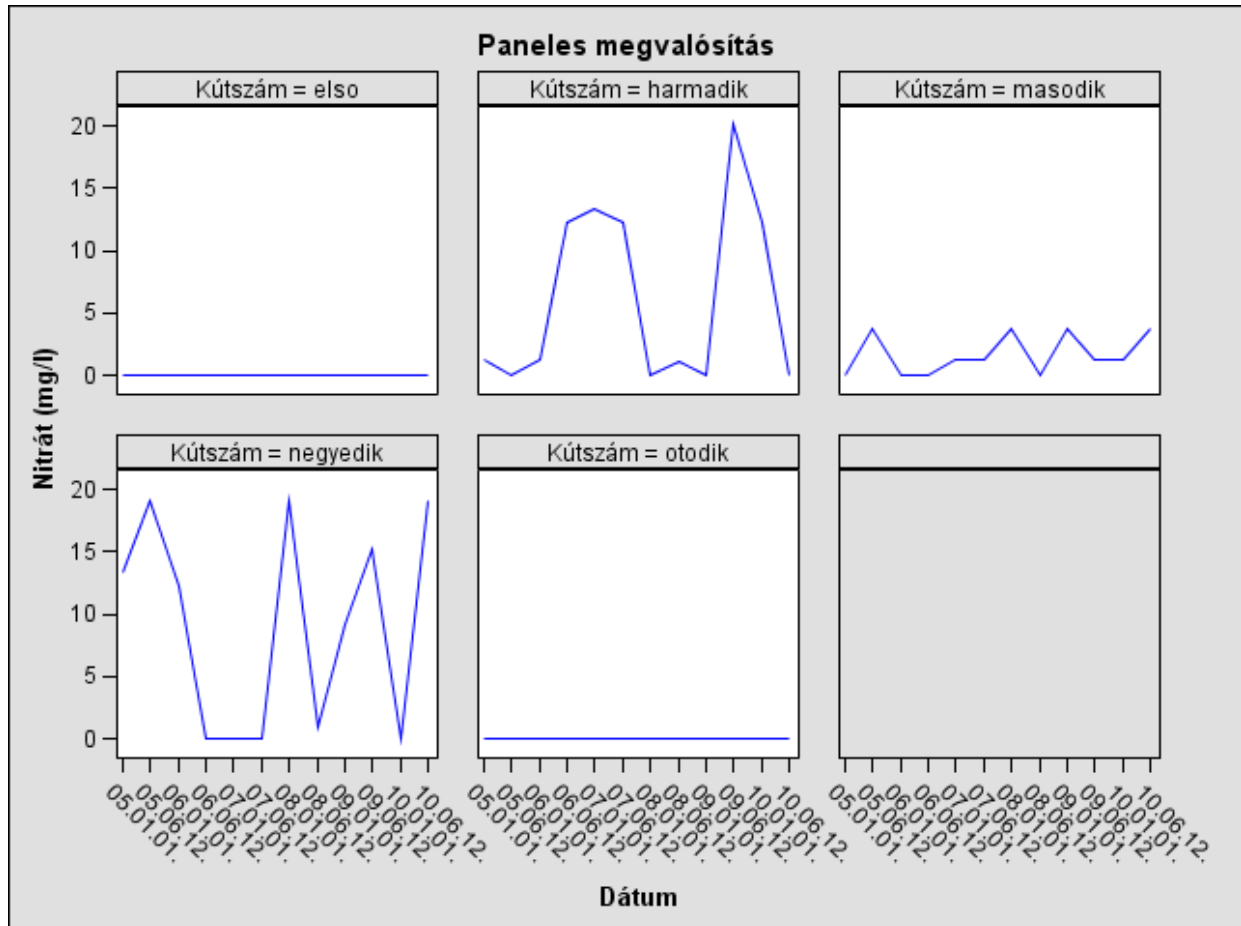
/* 3. megoldás: panelesen kirajzolva egyesével az 5 görbét */
goption reset=all;

proc sgpanel data=vegso noautolegend;

```

```
run;  
quit;
```

Eredmény:



48. ábra: Paneles megjelenítés

13. MELLÉKLET

13.1 KÉKSZALAG EXCEL ADATÁLLOMÁNY

Az első melléklet tartalmazza a balatoni Kékszalag vitorlás verseny adatait, amit az elemzéshez kiinduló adatállományként felhasználunk (kekszalag2008.xls).

13.2 NOX MÉRT ÉRTÉKEK

A második melléklet tartalmazza a levegőszennyezés valós feladat adatait Word dokumentumban (NOx_levego_feladat.doc).

13.3 SZENNYVÍZTISZTÍTÁS ELŐTT ÉS UTÁN MÉRT JELLEMZŐK ADATAI

A harmadik mellékletben szennyvíztisztítás során 2010-ben mért 52 napi 10 jellemző mért adatát találjuk meg nyers és tisztított formában. Az Excel dokumentum első munkalapja a kapott eredeti adathalmaz, amit az „m2” munkalapon javítottunk (lásd piros értékek), illetve az „m3” munkalapon előkészítettünk a SAS-ba való beimportálásra (nyers-szennyviz.xls).

13.4 HIÁNYOS ADATOK SZENNYVÍZTISZTÍTÓ ÜZEMBŐL

A negyedik mellékletben csatolt Excel táblázat az első munkalapján az eredetileg a szennyvíztisztító üzemből összegyűjtött igen hiányos adatokat tartalmazza, míg a második munkalap „SAS” néven a SAS-ba való beimportálásra lett az előbbiből előkészítve (Szennyviz_HA.xls).

13.5 TALAJVÍZ KUTAK PÁRHUZAMOSAN MÉRT NITRÁT KONCENTRÁCIÓI

Az ötödik melléklet egy SAS adatállomány (otkut.sas7bdat), melyben 5 talajvíz kút mért nitrát koncentrációs értékei (mg/l) szerepelnek 2005 és 2010 között félévente.

14. FÜGGELÉK

14.1 MATEMATIKAI ALAPOK AZ ADATBÁNYÁSZATI SZOFTVEREK ELSŐ MEGISMERÉSÉHEZ

Az első függelékben matematikai és statisztikai alapokat jelen dokumentum könnyebb megértéséhez (stat_book.pdf).

14.2 SAS / GRAPH GYAKORLATI BEMUTATÓ

A második függelékben a SAS grafikus megjelenítésének alapjait ismertető, gyakorlati példákkal alátámasztott dokumentumot találjuk (SAS-GRAPH_gyak_bemutato_SAS9.1_2010-04-26.doc).

15. ÁBRAJEGYZÉK

1. ábra: A szén körforgása	10
2. ábra: A nitrogén körforgása	11
3. ábra: Nitrogén-körforgalom (Delwiche, 1970 után Gisi, 1990).....	12
4. ábra: Az oxigén körforgása	13
5. ábra: A víz és hidrogén körforgása	14
6. ábra: Foszforkörforgalom (Stewart, 1981 után módosítva)	15
7. ábra: A foszfor körforgása.....	15
8. ábra: A kén körforgása.....	16
9. ábra: Kénkörforgalom (Brady, 1974 után módosítva)	17
10. ábra: SAS rendszer.....	23
11. ábra: SAS 9.2 Base kezdőfelület.....	24
12. ábra: LOG naplózó ablak: a program betöltését ismertető sorokkal.....	25
13. ábra: Program Editor: SAS program szerkesztő ablaka	26
14. ábra: HTML megjelenítés beállítása (<i>Tools\Options\Preferences</i>).....	27
15. ábra: Explorer ablak két részes megjelenítésben.....	28
16. ábra: Új könyvtár létrehozása SAS-ban.....	29
17. ábra: Results ablak	30
18. ábra: Listing output karakteres megjelenítése	31
19. ábra: Kékszalag Excel adatállomány részlet.....	32
20. ábra: Import varázsló 1. lépés: beolvasandó file típusának kiválasztása	33
21. ábra: Import varázsló 2. lépés: a kiválasztott Excel fájl munkalapjának megjelölése, ahonnan az adatokat szeretnénk beolvasni a SAS-ba	34
22. ábra: Import varázsló 3. lépés: könyvtár és adatállomány nevének megadása	34
23. ábra: Excelből beimportált Kékszalag SAS adatállomány a "kornyeze" könyvtárban	35
24. ábra: PROC CONTENTS eljárás eredménye a Results ablakban	46
25. ábra: Az elsőként megtisztított KORNYEZE.KEK adatállomány	47
26. ábra: Yardstick szám szerinti gyakoriságok	61
27. ábra: Befutási idő szerinti gyakoriságok	62
28. ábra: Korrigált futamidő szerinti gyakoriságok a teljes mintára	63
29. ábra: Gyors hajók korrigált futamidőinek gyakorisága.....	64
30. ábra: Rendezett NOx adatállomány.....	69

31. ábra: Levegőszennyezés vonaldiagramja túréshatárokkal	74
32. ábra: Nyers és tisztított PH értékek vonaldiagramja SAS-sal.....	81
33. ábra: Hat soros eredeti szennyvíztisztításnál mért adatállomány SAS-ban (kornyeze.ha)	87
34. ábra: Az eredeti hat napi mért hiányos szennyvíztisztítási adatok kiegészítése egy év minden napjára generált adatokkal (kornyeze.ha_kieg)	88
35. ábra: 365 napi generált szennyvíztisztítással kapcsolatos adatok	98
36. ábra: Generált adatok két változóra: év és adat	100
37. ábra: Generált adatok grafikus megjelenítése.....	101
38. ábra: Évtized csoportosító változó készítése a generált adatállományba.....	102
39. ábra: Folyamatosnak ható generált adatsor.....	105
40. ábra: Öt talajvíz kútban mért nitrát koncentráció (mg/l) 2005 és 2010 között fél évente	113
41. ábra: Kut változó generálása	115
42. ábra: Kut_szam változó generálása	116
43. ábra: Az első kút adatai külön adatállományban: kornyeze.otkut1	117
44. ábra: Végző tábla részlete	118
45. ábra: Teszt grafikon	119
46. ábra: Formázott grafikon.....	120
47. ábra: Az 5 kút vonaldiagramja idősoros megjelenítéssel.....	121
48. ábra: Paneles megjelenítés	122