Idősorok analízise

Lukovszki Csaba

2021. április

Tartalomjegyzék


```
require(graphics)
```

1. Idősor adatok analízise

Idősor adatok alatt kronológiailag egymást követő adatokat értünk. Ilyen adatok lehetnek a tőzsdeindex mozgása, a napi hőmérséklet adatok, az utasok száma egy járaton, vagy éppen egy EKG eredmény. Az idősorok adatok olyan regisztrátumok, melyek *sorrendje fontos szerepet tölt be az értelmezés, illetve a feldolgozás során.*

Formálisan az idősor adatok egymást megadott időegységenként követő, **reguláris**, folytonos értékkészletű sorrendezett adatok egymásutánja.

Az idősor adatokat úgy is modellezhetjük, hogy egy **véletlen folyamat** adott időpontbeli realizációinak sorozata.

$$\{y_0, y_1, y_2, y_3, ...y_N\} = \{Y(t = k)\}, \quad k \in (0, 1...N)$$

Az R-ben az idősorokat *time series (ts)* típusú objektumok tárolják. Az idősorok időben egyenközű minták tárolására alkalmasak. Egy évhez (mint időegységhez) viszonyítva egy adott gyakoriságú (frekvenciájú) minták tárolására alkalmas.

Az R-ben lehetőség van **irreguláris** idősorok tárolására és kezelésére is. Ezekre leggyakrabban az `xts` és `zoo` csomagokat használhatjuk. Erre jelen jegyzet nem tér ki.

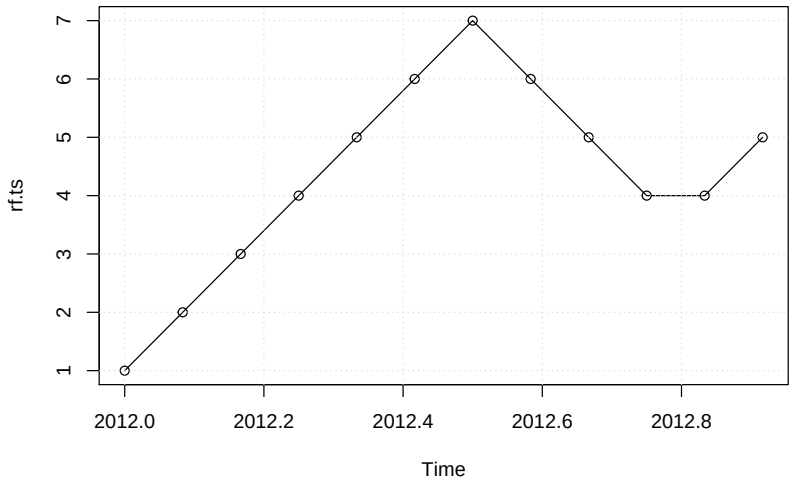
1.1. Idősorok létrehozása és ábrázolása

A következő példák mutatják, hogy miként tudunk létrehozni idősorokat és miként tudjuk ábrázolni azt.

```
rf <- c(1,2,3,4,5,6,7,6,5,4,4,5)
# start: év, index az éven belül
rf.ts <- ts(rf, start=c(2012,1), frequency=12)
print(rf.ts)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2012   1   2   3   4   5   6   7   6   5   4   4   5
```

```
plot.ts(rf.ts, type = "o")
grid()
```



1.2. Idősorok tulajdonságai

Az idősorok tulajdonságait a következő függvényekkel kérdezhetjük le.

```
# Az adatok típusa
typeof(rf.ts) # valójában egy double vektor
```

```
## [1] "double"
```

```
# Osztálya
class(rf.ts)
```

```
## [1] "ts"
```

```
# Az összes attribútuma
attributes(rf.ts)
```

```
## $tsp
## [1] 2012.000 2012.917   12.000
##
## $class
## [1] "ts"
```

```
# Kezdő időpontja
start(rf.ts)
```

```
## [1] 2012    1
```

```
# Az utolsó időpontja
end(rf.ts)
```

```
## [1] 2012   12
```

```
# Az adatok évenkénti frekvenciája
frequency(rf.ts)
```

```
## [1] 12
```

```
# Az időskála (double-lé konvertálva)
time(rf.ts)
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2012 2012.000 2012.083 2012.167 2012.250 2012.333 2012.417 2012.500 2012.583
##           Sep      Oct      Nov      Dec
## 2012 2012.667 2012.750 2012.833 2012.917
```

1.3. Műveletek idősorokon

Az idősorok egyszerű vektorokként viselkednek.

```
rf2 <- c(8,8,8,8,7,7,7,7,6,6,6,6)
rf2.ts <- ts(rf2, start = c(2012,1), frequency = 12)
rf + rf2
```

```
##  [1]  9 10 11 12 12 13 14 13 11 10 10 11
```

```
rf.ts + rf2.ts
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2012   9  10  11  12  12  13  14  13  11  10  10  11
```

Ugyancsak alkalmazhatók rá vektorokon értelmezett műveletek, de az idősorra speciális műveletek is.

```
# kiválasztás
# egyszerű vektorművelet
subset(rf.ts,  rf.ts > 4)
```

```
## [1] 5 6 7 6 5 5
```

```
# ts
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
?subset.ts
subset(rf.ts, start=2, end=11)
```

```
##      Feb Mar Apr May Jun Jul Aug Sep Oct Nov
## 2012   2   3   4   5   6   7   6   5   4   4
```

```
subset(rf.ts, season=2)
```

```
## Time Series:
## Start = 2012.083
## End = 2012.083
## Frequency = 1
## [1] 2
```

```
#subset(rf.ts, quarter=3)
# ts
library(stats)
window(rf.ts, frequency=4)
```

```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 2012    1    4    7    4
```

```
subset(window(rf.ts, frequency=4), quarter=c(1,2))
```

```
## Time Series:
## Start = c(2012, 1)
## End = c(2012, 2)
## Frequency = 2
## [1] 1 4
```

```
window(rf.ts, frequency=6)
```

```
## Time Series:
## Start = c(2012, 1)
## End = c(2012, 6)
## Frequency = 6
## [1] 1 3 5 7 5 4
```

```
window(rf.ts, frequency=6, start=c(2012,1), end=c(2012,6))
```

```
## Time Series:
## Start = c(2012, 1)
## End = c(2012, 3)
## Frequency = 6
## [1] 1 3 5
```

```
# eltolás
library(stats)
lag(rf.ts, k=1)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2011                                               1
## 2012   2   3   4   5   6   7   6   5   4   4   5
```

```
lag(rf.ts, k=-3)
```

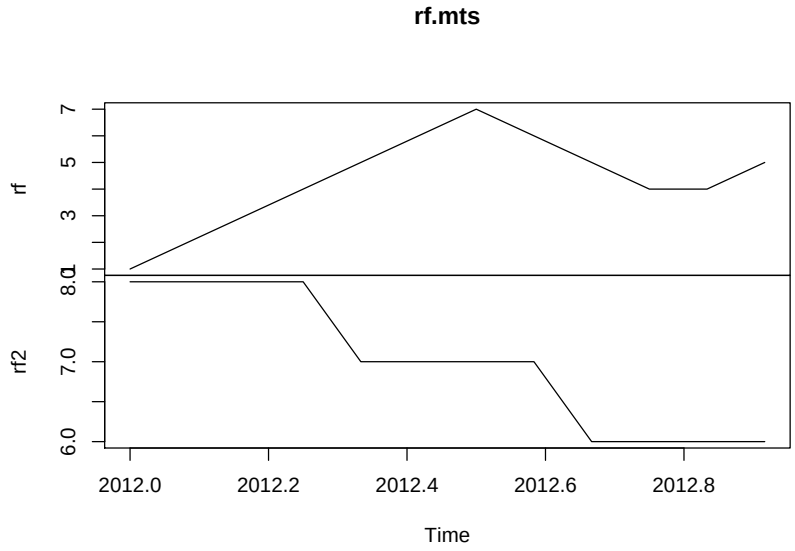
```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
```

```
## 2012                 1    2    3    4    5    6    7    6    5
## 2013    4    4    5
```

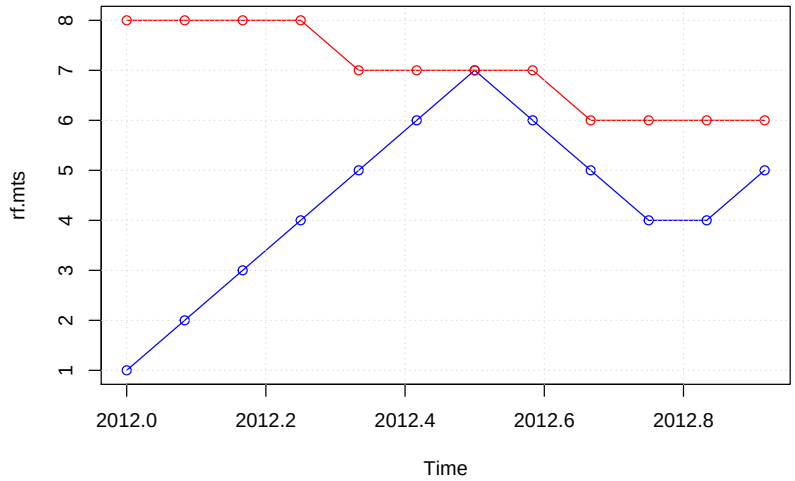
1.4. Idősorok kombinálása és ábrázolása

Az idősorokat oszlopokként értelmezzük és aszerint is kombináljuk őket.

```
rfc <- cbind(rf, rf2)
rf.mts <- ts(rfc, start=c(2012,1), frequency=12)
plot.ts(rf.mts)
```




```
plot.ts(rf.mts, plot.type="single", type="o", col=c("blue", "red"))
grid()
```



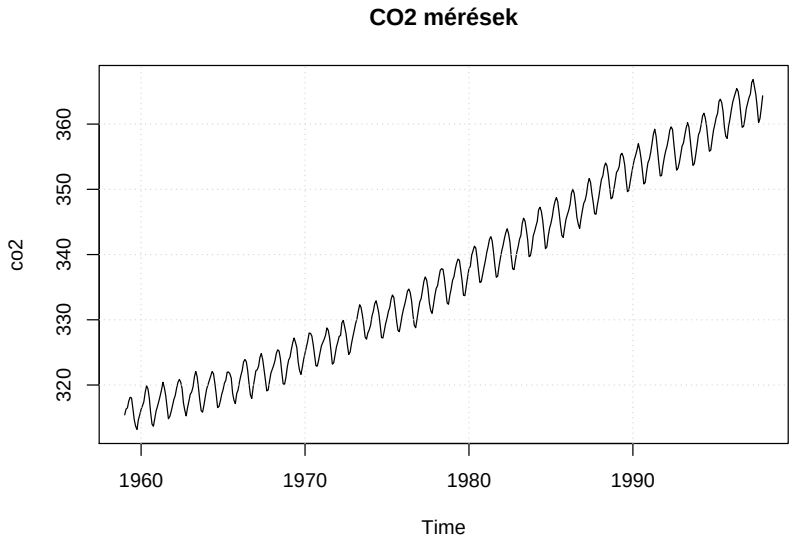
1.5. A CO2 adathalmaz

A következőkben vizsgáljuk meg a Hawaii Mauna Loa-nál mért atmoszférikus CO2 koncentráció értékeket. Láthatjuk, hogy a `co2` adathalmaz esetében az idősor frekvenciája 12, ami annyit tesz, hogy évente 12 mérést tartalmaz az idősor, tehát havonta tárol mintákat. Az idősor első értéke 1959. januári, míg az utolsó 1997. decemberi.

```
data(co2)
attributes(co2)
```

```
## $tsp
## [1] 1959.000 1997.917   12.000
##
## $class
## [1] "ts"
```

```
plot.ts(co2, main = "CO2 mérések")
grid()
```



Fontos észrevenni a **különbséget** az idősor, valamint a pontszerű adatok között. Míg egy mennyiség mért értékeinek regisztrációjánál a statisztikai jellemzők (várható érték, szórás, egyéb momentumok, eloszlás) jellemzi az adott mennyiséget, addig az idősoroknál az egyszerű statisztikai jellemzők nem használhatók fel az adatok jellemzésére. Úgy is mondhatjuk, hogy *az idősor adat egyszerű (egyváltozós) statisztikai jellemzése nem ad leírást az idősorok mért adatainak jellemzésére, és segítségével nem jósolhatók meg a következő mért értékek.*

A széndioxid (CO2) koncentráció alakulásánál a következőket vehetjük észre:

- Egy jellegzetes emelkedést mutat hosszú távon
- Ismétlődő peiódusokat fedezhetünk fel

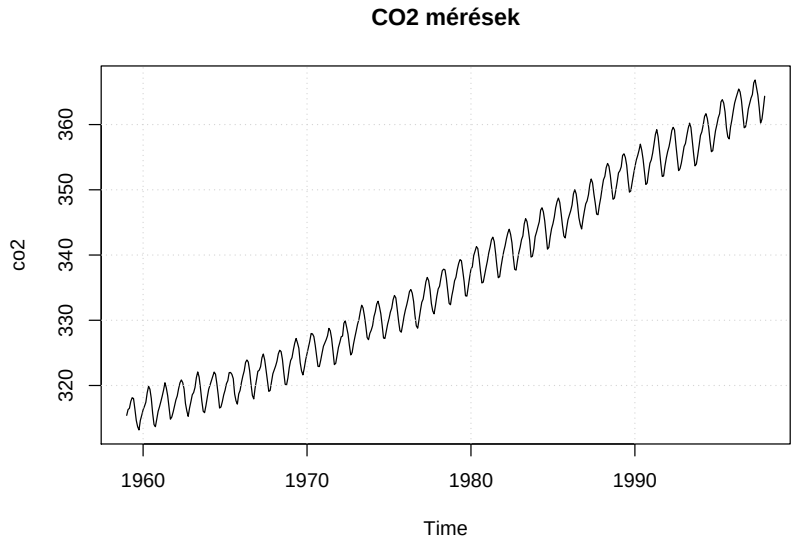
Ennek következtében két dolgot állíthatunk a megjelenített idősorról:

- *Nem memóriamentes*: Egy adott időpont utáni adatok függenek a korábbi adatoktól
- *Nem stacionárius*: Különböző időpontokban vett statisztikai leírók nem ugyanazok, változnak

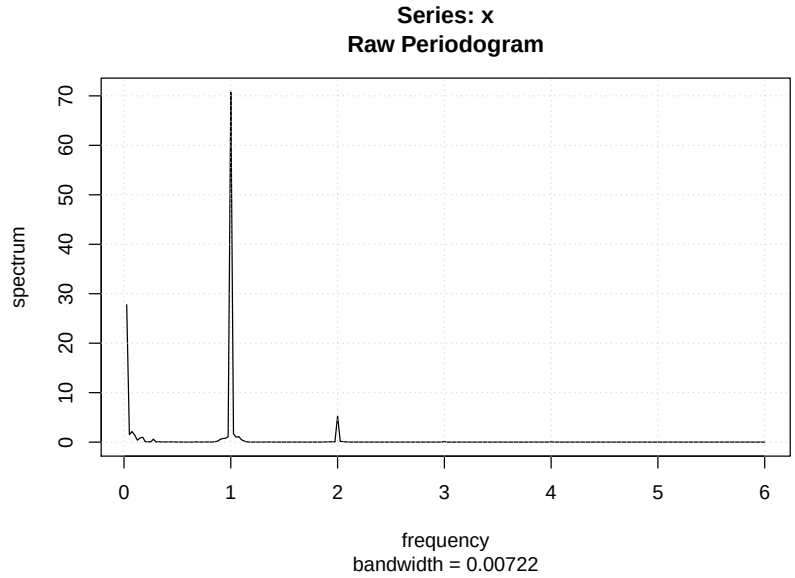
1.6. Idősorok a frekvenciatartományban

Az idősorokat nem csak az időtartományban, de a frekvenciatartományban is vizsgálhatjuk. A következőkben nézzük meg a mintahalmazunk frekvenciakomponenseit, mely segítségével időtartományból frekvenciatartományba transzformáljuk az idősor adatokat.

```
plot.ts(co2, main = "CO2 mérések")
grid()
```




```
library(stats)
spectrum(co2, log = "no")
grid()
```



Az ábrából látató, hogy az adatsorunk legnagyobb frekvencia komponense az 1 egységnyi frekvenciánál található, ami az 1 évet jelenti, hiszen az idősorunk időbélyegének mértéegysége az 1 év. (ls. attribútumok, `attributes()`)

1.7. Autó korreláció

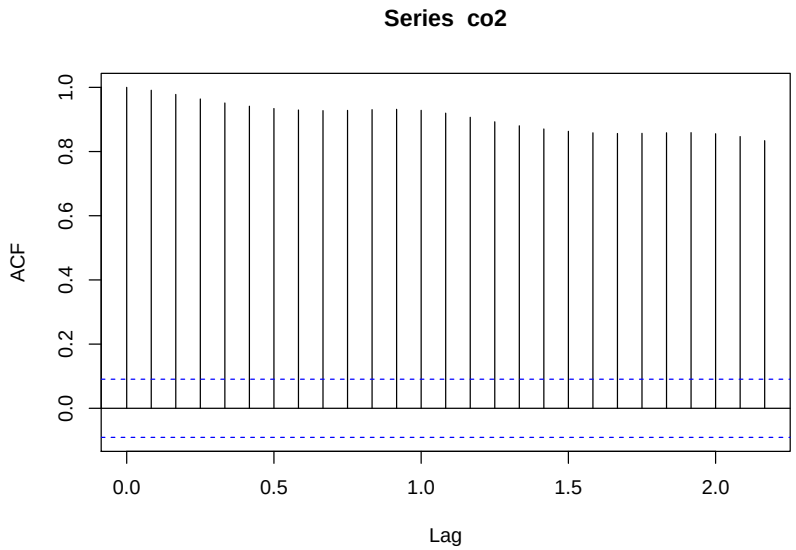
Az idősorok egyes mintái közötti összefüggéseket az **autókorrelációs függvény** (ACF - Auto Correlation Function) segítségével írhatjuk le. A függvény megmutatja, hogy az egyes mintákból a következő mintákra milyen megbízhatósággal következtethetünk.

Ehhez először nézzük az autó korreláció definícióját

$$\rho_Y(s,t) = cor(Y_s, Y_t) = \frac{\gamma_Y(s,t)}{\sigma_s \sigma_t} = \frac{cov(Y_s, Y_t)}{\sigma_s \sigma_t}$$

A tapaszatalati autokorrelációs függvényt (ACF) a következőképpen ábrázolhatjuk:

```
library(stats)
acf(co2)
```

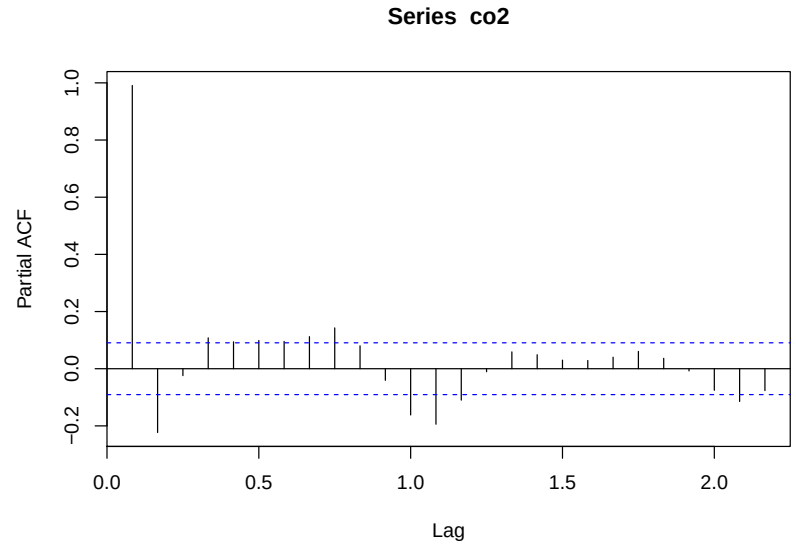


Az ábrából látható, hogy még az egymástól nagy távolságra lévő minták közötti korreláció is igen jelentős, azaz az idősor mintái összefüggnek.

Az autokorreláció vizsgálatához tartozik a tapasztalati **parciális autokorrelációs függvény** vizsgálata (PACF - Partial Auto Correlation Function) is.

A tapasztalati parciális autokorrelációs függvény két változó közötti kapcsolat erősségét mutatja akkor, ha a köztes változókon keresztül terjedő hatásokat kiszűrjük.

```
library(stats)
pacf(co2)
```



1.8. Idősorok modellezése

Az idősor adatok feldolgozásának **célja**:

- Megismerés: Analízis
- Előrejelzés: Predikció

Az idősor adatok elemzésének részletesebb feladata, hogy a fenti komponenseket meghatározza,

valamint hogy ezen értékek alapján *következtetéseket vonjon le* az idősorok mögött lévő folyamatokról, valamint *predikálja*, jósolja meg a következő értékeket. Ebben nagy jelentősége van a trend és szenzonális változásnak, viszont a véletlen ingadozás változó értékével nem számolhatunk, azt az analízisünk kapcsán bizonytalannak kell tekintsük, vagy csak sztochasztikusan tudjuk értelmezni.

Az idősorok elemzését kétféle **módszerrel** végezhetjük el:

- **Determinisztikus**: Főként hosszú távú előrejelzésekre használjuk
- **Sztochasztikus**: Főként rövid távú előrejelzések esetében használható

Determinisztikus módszer esetében az idősort alakító tényezőket vesszük figyelembe, melyek alapján az idősor időbeni alakulása jól, determinisztikusan leírható. Előnye, hogy az adott feladatnak megfelelően az egyes tényezők jól meghatározhatók, megérthetők, hátránya, hogy a folyamatokban rejlő hibákat nem veszi számításba, így az eredmény egy megfoghatatlan mértékű hibával terhelt.

Sztochasztikus módszer esetében az idősorba beépülő változások, kisebb eltérések, hibák alapján sztochasztikus folymatot építünk, melyek a valószínűségszámítás és statisztika elemeivel jól kezelhetők. Ennek eredményeképpen rövid távon pontos és az esetleges változásokra, megbízhatósági intervallumokra jó becslést ad.

1.9. Dekompozíció

Az idősorok legegyszerűbb elemzése alapján az idősor adatokat a következő **komponensekre** bonthatjuk:

1. *Trend* (T): A trend az idősor hosszútávú változását mutaja, melyek alapvetően mentesek az apróbb ingadozásoktól.
2. *Ciklikus változás* (C): A trend körüli hosszabb időtávlató (jellemzően egy éven túli) indazozás, mely nem feltétlenül periodikus.
3. *Szezonális ingadozás* (S): A megfigyelt értékek állandó periódushosszúságú és jellemzően azonos mintájú változása.
4. *Irreguláris, véletlen változás* (V): A véletlen ingadozás értéke tartalmazza azokat az összetevőket, melyek nem sorolhatóak sem a trendszerű változáshoz, sem a szezonális változáshoz. Ezeket okozhatják az adott mért értéktől független, külső körülmények, események.

Az egyes komponensek együtt alkotják az eredeti idősor értékeket. Attól függően, hogy közöttük *additív*, vagy *multiplikatív* hatás értvényesül-e a tényleges, valóságban tapasztalt mintaértékek az egyes komponensek összegeként, vagy szorzataként alakítható ki. A gyakorlatban az idősorokatban az összefüggésekre az aditív hatást feltételezzük, mivel multiplikatív esetben egy logaritmikus transzformációval egyszerűen additívvá alakíthatók az összefüggések.

Additív idősor esetében:

$$Y = T + C + S + V$$

Multiplikatív idősor esetében:

$$Y = T * C * S * V$$

$$log(Y) = log(T) + log(C) + log(S) + log(V)$$

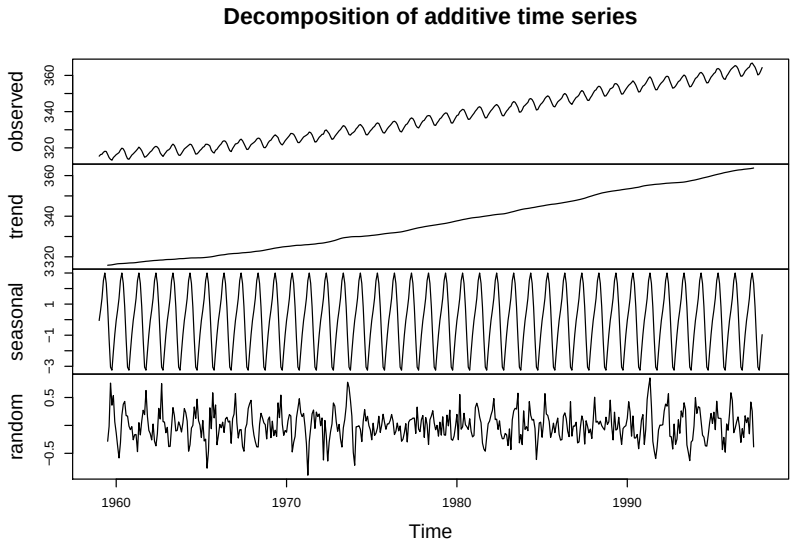
1.9.1. Dekompozíció additív esetben

A következőkben bonsuk komponensekre a CO2 idősorunkat.

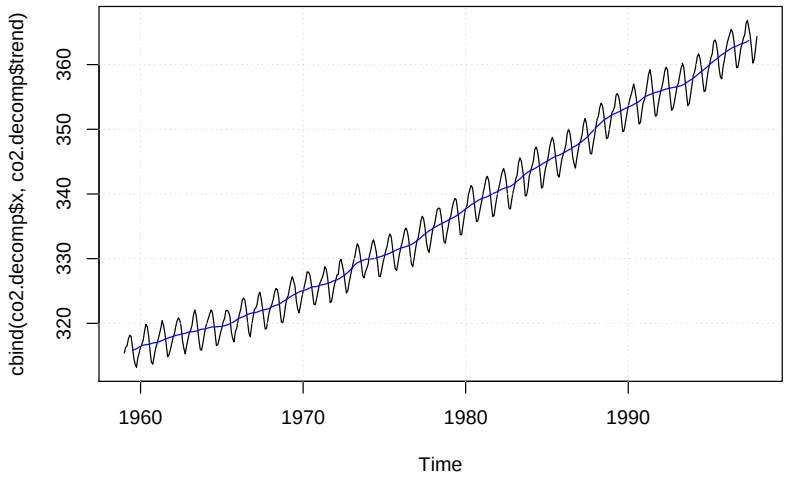
```
co2.decomp <- decompose(co2, type="additive")
objects(co2.decomp)
```

```
## [1] "figure"   "random"   "seasonal" "trend"    "type"     "x"
```

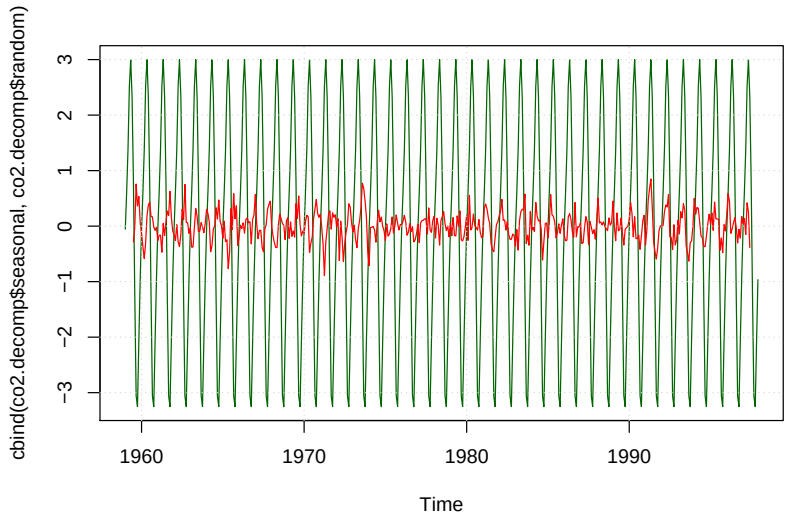
```
# az eredeti adathalmaz
#co2.decomp$x
# a dekompozíció eredménye
plot(co2.decomp)
```




```
# az eredeti idősor és a trend ábrázolása
plot.ts(cbind(co2.decomp$x, co2.decomp$trend), plot.type="single", col=c("black", "blue"))
grid()
```

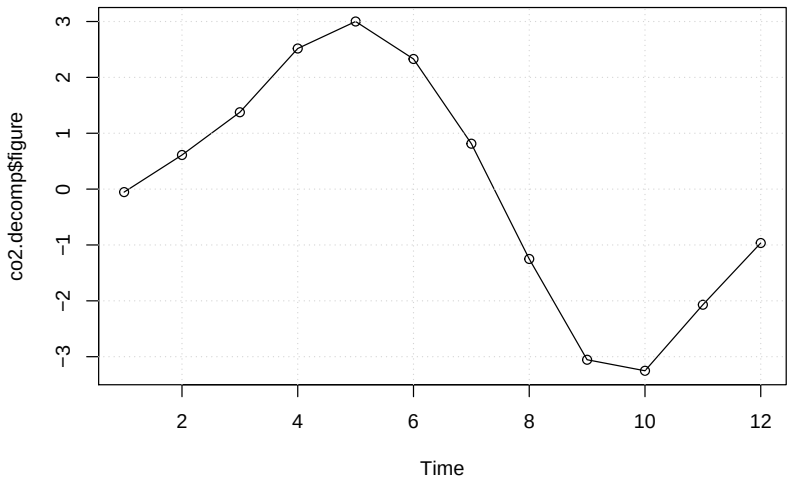



```
# a szezonális és a véletlen változás kirajzolása
plot.ts(cbind(co2.decomp$seasonal, co2.decomp$random), plot.type="single", col=c("darkgreen"
grid()
```



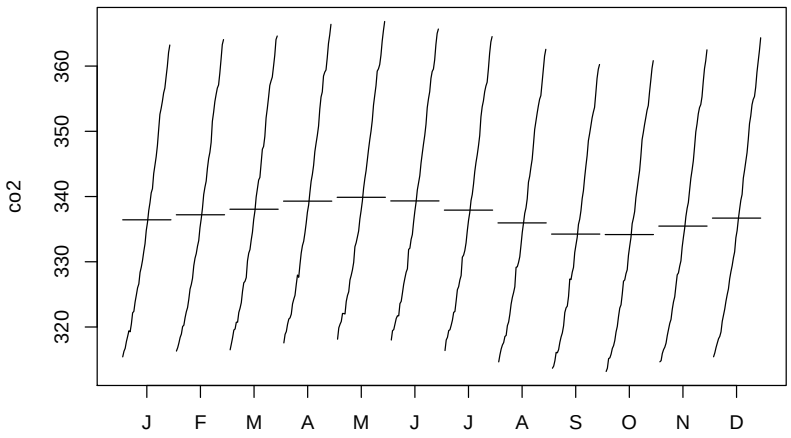
A dekompozíció során megkapjuk a szezonális változás egy periódusát is.

```
plot.ts(co2.decomp$figure, type="o")
grid()
```

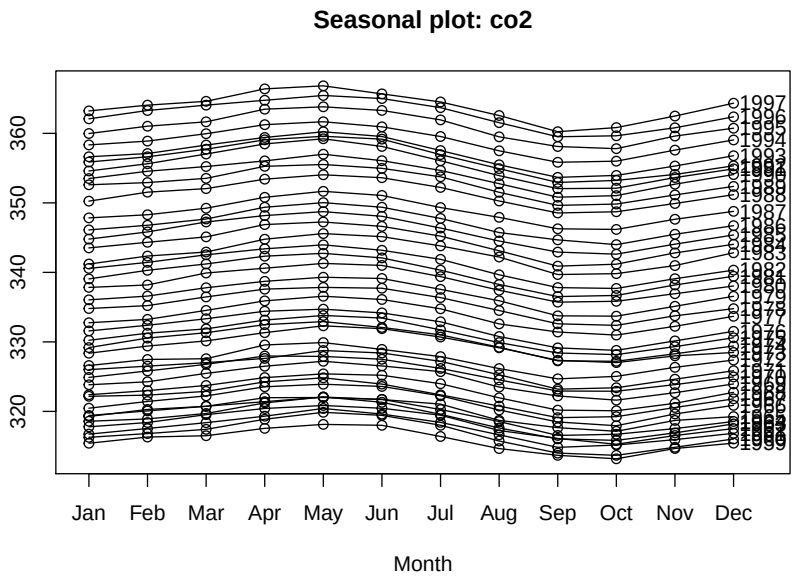


Az idősor változását vizsgálhatjuk adott hónapora vetítve, valamint az egyes szezonális periódusokra is.

```
library(stats)
monthplot(co2)
```



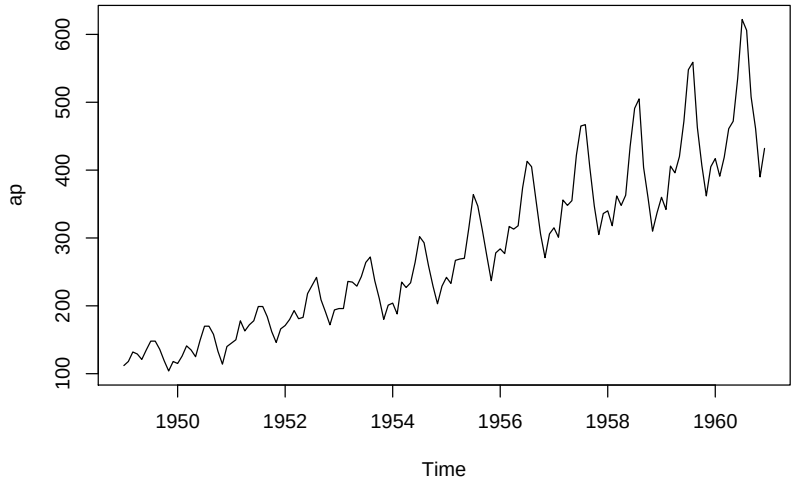
```
library(forecast)
seasonplot(co2, year.labels = T)
```



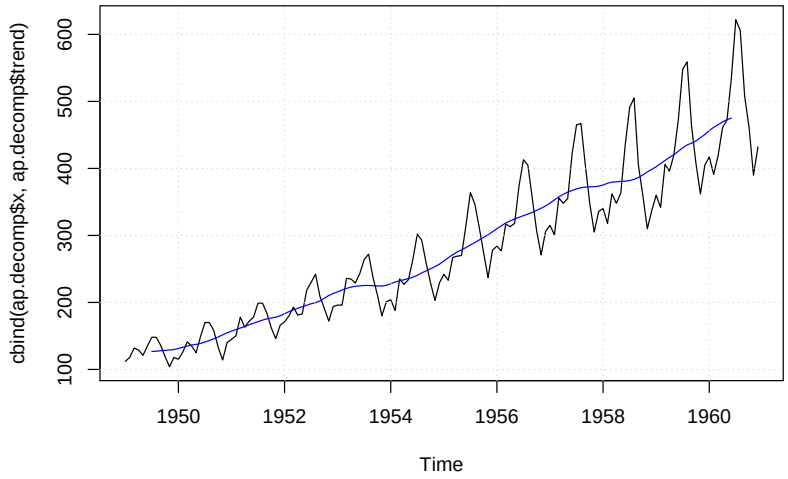
1.9.2. Dekompozíció multiplikatív esetben

A következőkben nézzünk meg egy olyan adathalmazt, mely multiplikatív az egyes komponensekre nézve. A multiplikatív dekomponzíciót elvégezhetjük a `decomp()` függvénnyel.

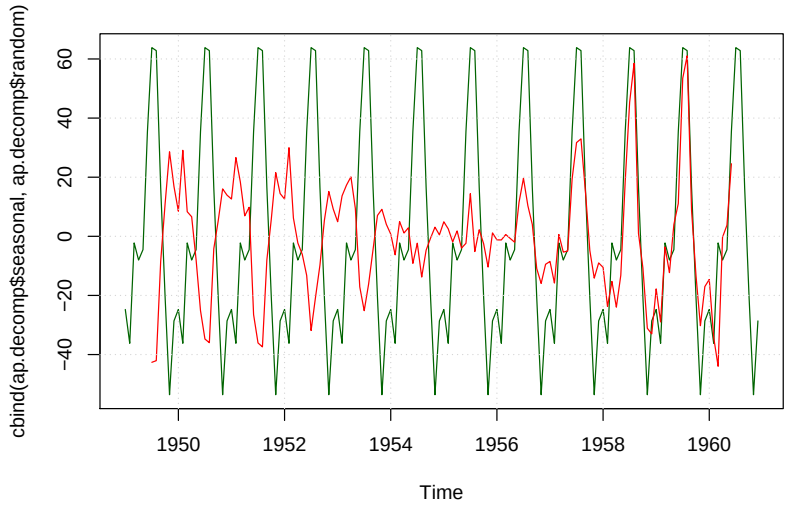
```
data("AirPassengers")
ap <- AirPassengers
plot(ap)
```




```
# dekompozíció
ap.decomp <- decompose(ap)
plot.ts(cbind(ap.decomp$x, ap.decomp$trend), plot.type="single", col=c("black", "blue"))
grid()
```



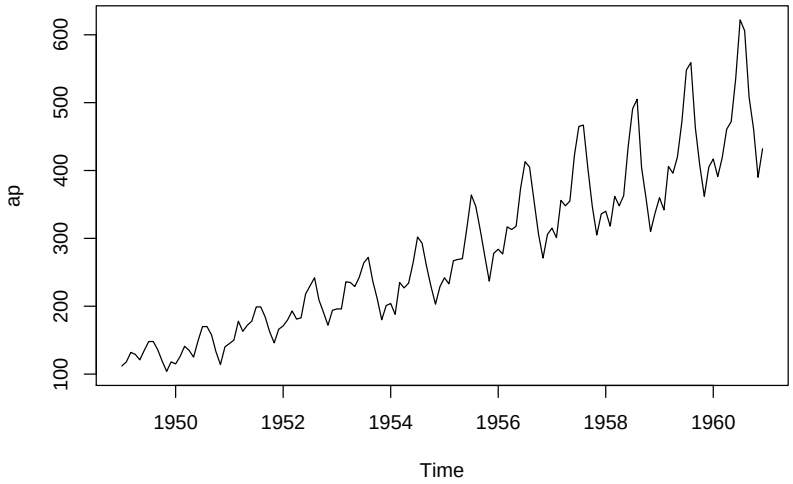

```
plot.ts(cbind(ap.decomp$seasonal, ap.decomp$random), plot.type="single", col=c("darkgreen","
grid()
```



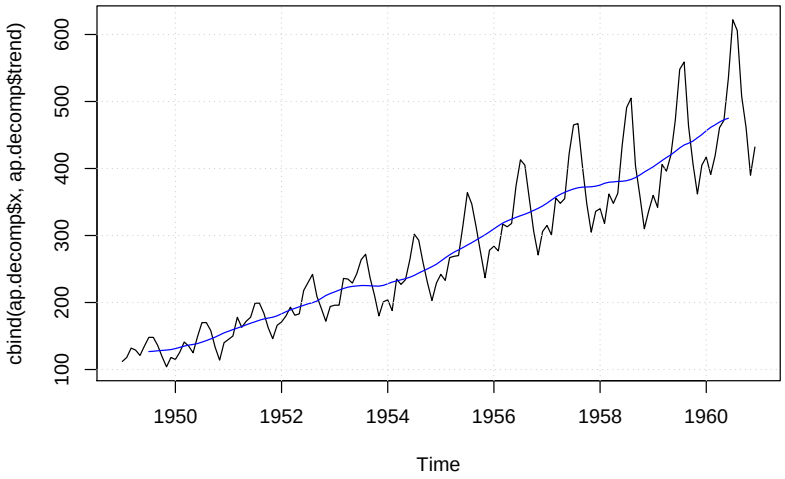
Már az eredeti idősor ábrázolásából feltűnő, hogy az adatsor trendjének növekedésével szinte egyenesarányosan növekszik a szezonális komponens amplitúdója is. Amennyiben elvégezzük az additív dekompozíciót, a szezonális és véletlen értékeket ábrázolva jól látható, hogy a véletlen komponens az idősor elején tartalmazza az átlagos szezonális komponens ellenütemű, az idősor végén az ugyanolyan ütemű értékeit.

Ezért próbálkozzunk a multiplikatív dekompozícióval.

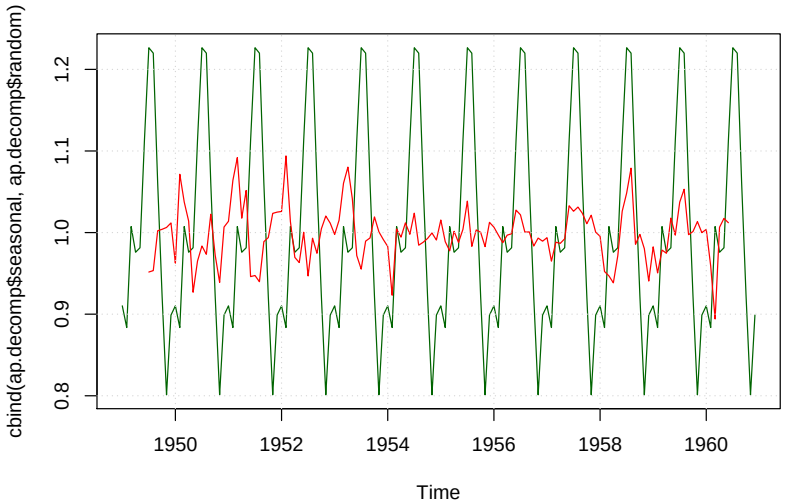
```
data("AirPassengers")
ap <- AirPassengers
plot(ap)
```




```
# dekompozíció
ap.decomp <- decompose(ap, type = "multiplicative")
plot.ts(cbind(ap.decomp$x, ap.decomp$trend), plot.type="single", col=c("black", "blue"))
grid()
```

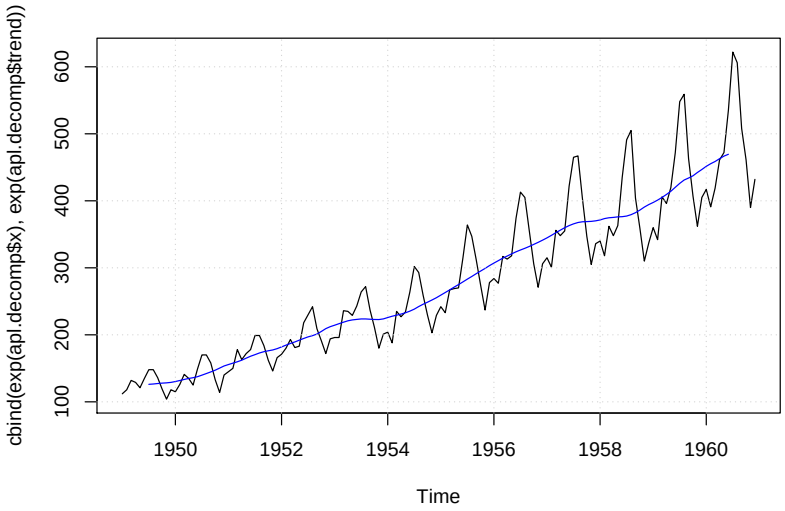



```
plot.ts(cbind(ap.decomp$seasonal, ap.decomp$random), plot.type="single", col=c("darkgreen","
grid()
```

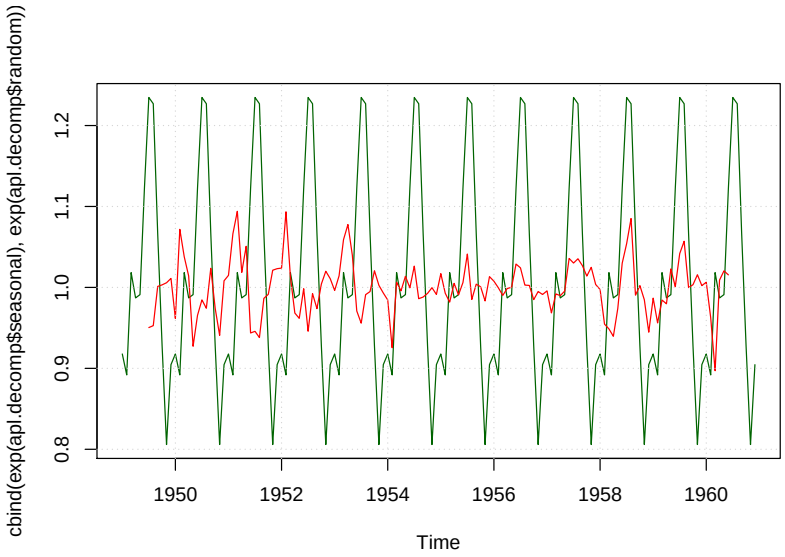


A multiplikatív dekompozíciót elvégezhetjük a `log()` függvény használatával is.

```
apl <- log(ap)
apl.decomp <- decompose(apl, type="additive")
plot.ts(cbind(exp(apl.decomp$x), exp(apl.decomp$trend)), plot.type="single", col=c("black",
grid()
```




```
plot.ts(cbind(exp(apl.decomp$seasonal), exp(apl.decomp$random)), plot.type="single", col=c('
grid()
```



1.10. Trend számítás

A trend nem tartalmazza a periodikusságot. A trend meghatározása esetében azt keressük, hogy a idősor hosszú távú változása milyen függvénnyel írható le.

Az idősor terndjének meghatározását többféle módon is meghatározhatjuk. Ebből jellemzően a kettő legfontosabb a következő:

- *Mozgó átlagolás*: Ebben az esetben egy vizuálisan értelmezhető változást kapunk, nem egy függvényszerű változást.
- *Regresszió*: Regresszió számítással. Ebben az esetben egy meghatározott függvénnyel közelítjük az idősor adatokat a regressziószámítás felhasználásával.

1.10.1. Mozgó átlagolás

A *mozgó átlagolás* egy ablak alapján történik, mely lehet *szimmetrikus*, vagy *asszimetrikus*. Szimmetrikus esetben az átlagolt érték az intervallum közepén, míg asszimetrikus esetben - a gyakorlati alkalmazásoknak megfelelően - az intervallum későbbi szélénél helyezkedik el.

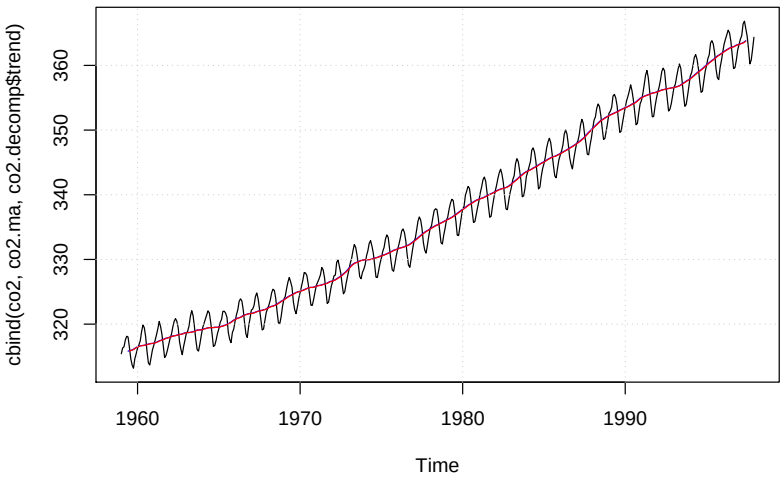
A mozgó átlag számítását a β_i súlyokkal a következőképpen számolhatjuk:

$$\hat{y}_t = \frac{1}{sum_{i=-k}^{q}\beta_i}\left[\beta_{-k}y_{t+k} + ... + \beta_0 y_t - \beta_1 y_{t-1} - .... - \beta_q y_{t-q}\right]$$

A mozgó átlag egyszerűen számítható a `TTR` csomagban lévő `SMA()` függvénnyel, vagy a `forecast` csomagban lévő `ma()` fügvénnyel. A következőkben ugyanakkor a `stats` csomag `filter()` függvényét alkalmazzuk.

A mozgóátlag alkalmazásánál figyelmesen kell megválasszuk az ablak méretét. Korábban láthattuk, hogy a leginkább meghatározó frekvenciakomponens a 12 hónap, ezért próbálkozzunk 12 méretű ablak használatával.

```
library(stats)
co2.ma <- filter(co2, filter=rep(1/12,12), sides=2)
# plot
plot(cbind(co2, co2.ma, co2.decomp$trend), plot.type = "single", col = c("black", "blue", "r
grid()
```




```
head(cbind(co2, co2.ma, co2.decomp$trend), 10)
```

```
##             co2   co2.ma co2.decomp$trend
## Jan 1959 315.42       NA               NA
## Feb 1959 316.31       NA               NA
## Mar 1959 316.50       NA               NA
## Apr 1959 317.56       NA               NA
## May 1959 318.13       NA               NA
## Jun 1959 318.00 315.8258               NA
## Jul 1959 316.39 315.8967         315.8613
## Aug 1959 314.65 315.9383         315.9175
## Sep 1959 313.68 316.0150         315.9767
## Oct 1959 313.18 316.1242         316.0696
```

A csúszóablakos átlagolásnál szükséges legalább az ablak mérettel megegyező minta, minek köszönhetően az idősorunk elején és végén összesen 11 időpontra nem tudtunk átlagot számolni.

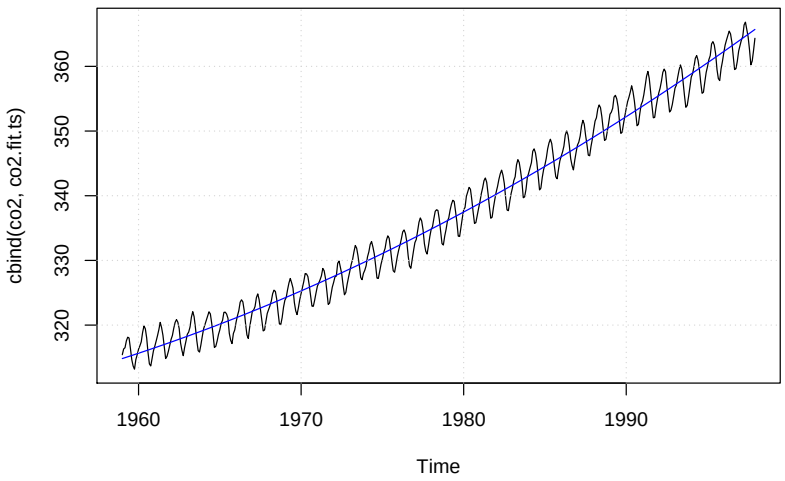
1.10.2. Regresszió

A trendszerű változás determinisztikus meghatározása felhasználható a jövőbeli értékek predikálására is. Ebben az esetben a trend-et, egy adott függvénnyel kell leírjuk. Ebben az esetben a korábban kalulált trend értékeket felbonthatjuk függvényszerűen leírható trend, valamint ciklikus komponensekre.

```
t <- as.numeric(time(co2))
co2.fit <- lm(co2 ~ poly(t, 2, raw = T))
summary(co2.fit)
```

```
##
## Call:
## lm(formula = co2 ~ poly(t, 2, raw = T))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0195 -1.7120  0.2144  1.7957  4.8345
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            4.770e+04  3.483e+03   13.70   <2e-16 ***
## poly(t, 2, raw = T)1 -4.919e+01  3.521e+00  -13.97   <2e-16 ***
## poly(t, 2, raw = T)2  1.276e-02  8.898e-04   14.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.182 on 465 degrees of freedom
## Multiple R-squared:  0.9788, Adjusted R-squared:  0.9787
## F-statistic: 1.075e+04 on 2 and 465 DF,  p-value: < 2.2e-16
```

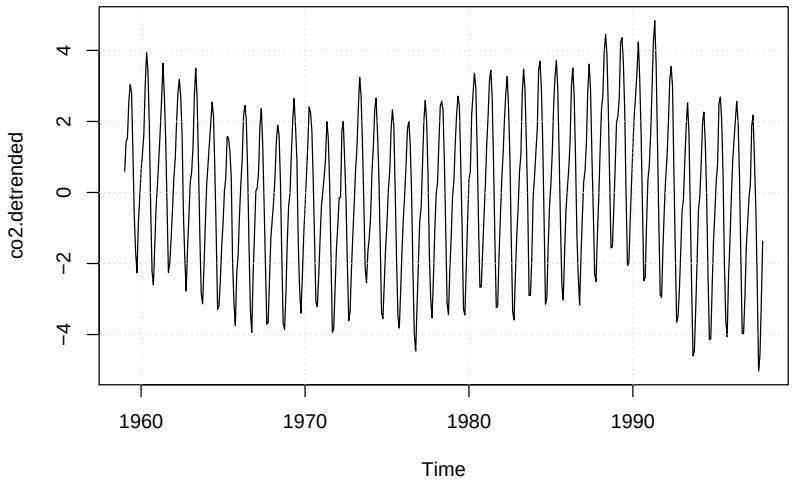
```
co2.fit.ts <- ts(co2.fit$fitted.values, start = start(co2), frequency = frequency(co2))
plot.ts(cbind(co2, co2.fit.ts), plot.type = "single", col = c("black", "blue"))
grid()
```



1.10.3. Detrending

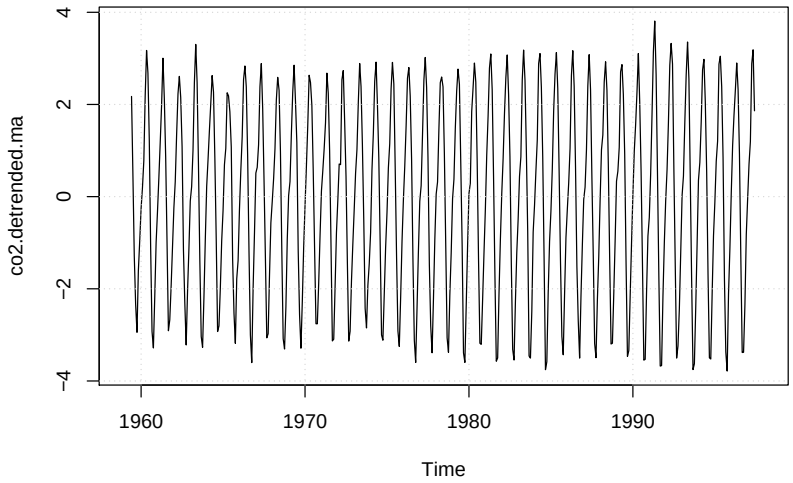
A detrending a trendszerű viselkedés megszüntetését jelenti. Ennek során eltávolítjuk a trendet a már ismert trendet az idősorunkból. Ezt a regresszió eredményével könnyen megtehetjük a következő módon.

```
co2.detrended <- co2 - predict(co2.fit)
plot.ts(co2.detrended)
grid()
```



Amennyiben mozgó átlagolást használtunk a trendszerű viselkedés meghatározásához, természetesen a trend kivonását csak a trend által meghatározott szakaszon végezhetjük el.

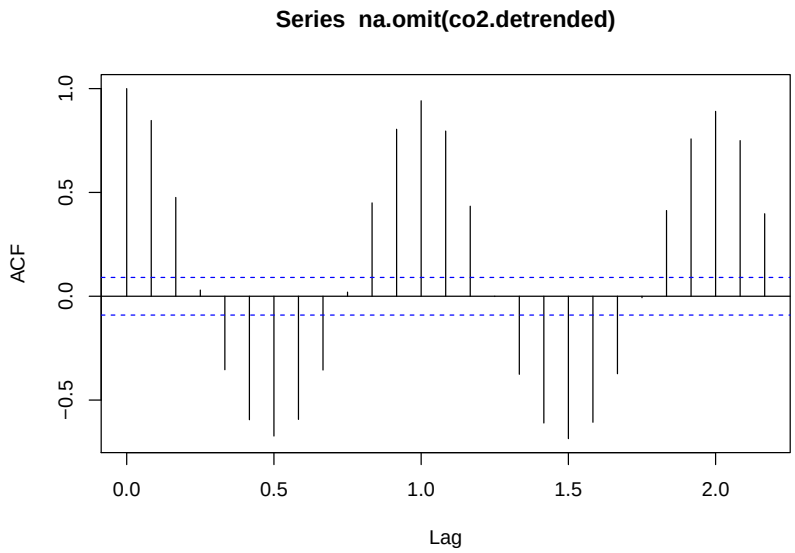
```
co2.detrended.ma <- co2 - co2.ma
plot.ts(co2.detrended.ma)
grid()
```



1.11. Szenzonális változás

A szezonális változás egy adott időintervallumra (jellemzően egy év) periodikus. A szezonális változás meghatározásához meg kell határozni azt a periódust, mely minden periódusban ugyanúgy viselkedik. A szezonális változás ugyancsak nem móriamentes, nem stacionárius. Ennek bizonyítására nézzük meg a a trend mentesített CO2 idősor autokorreláció függvényét.

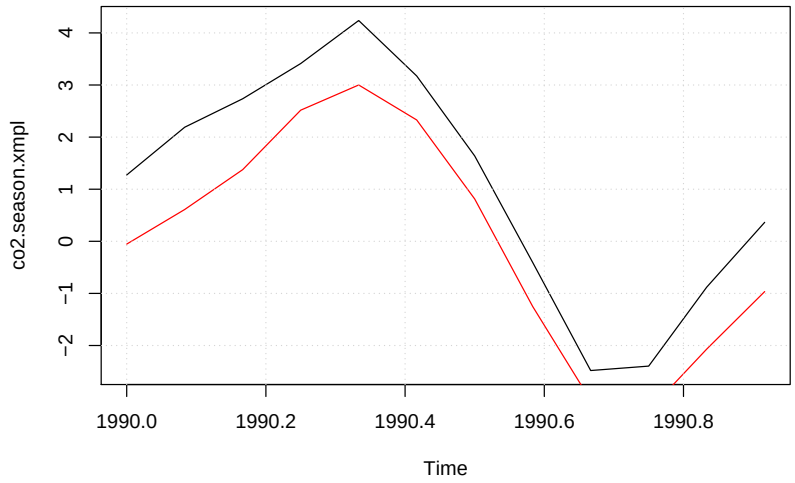
```
acf(na.omit(co2.detrended))
```



Látható, hogy a legnagyobb korrelációt a $k = 1(12)(lag)$, valamint ennek többszöröseinél tapasztalhatjuk. Mivel ebben az esetben a kovariancia értéke 1, ebben az esetben igencsak nagy mértékű összefüggést tapasztalhatunk.

Nézzünk meg egy szezont, hogy az értékei miként változnak.

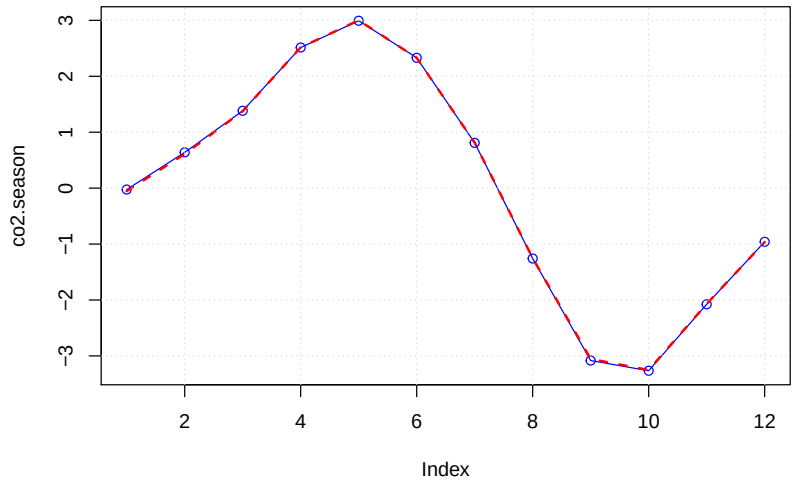
```
co2.season.xmpl = window(co2.detrended, start = c(1990,1), end = c(1990,12))
plot(co2.season.xmpl)
lines(ts(co2.decomp$figure, start = c(1990,1), frequency = 12), col = "red")
grid()
```



A szezonális értékeket az egyes periódusok átlagolásával kalkulálhatjuk ki. Ezután vessük össze az átlagos szezonális értéket a `decompose()` függvény által számolt értékkel.

```
# mátrixszá alakítjuk
co2.mat <- matrix(as.numeric(co2.detrended), ncol = 12, byrow = T)
# Egyes hónapora számoljuk az összeget és az érvényes elemek számát
co2.sum <- apply(co2.mat, 2, sum, na.rm = T)
co2.n <- apply(!is.na(co2.mat), 2, sum)
# Kiszámoljuk az átlagot
co2.season <- co2.sum / co2.n
# Nulla várható értékűvé transzformáljuk
co2.season <- co2.season - sum(co2.season) / 12
# Kirajzoljuk
```

```
plot(co2.season, type = "o", col = "blue")
lines(co2.decomp$figure, col = "red", lty = "dashed", lwd = 2)
grid()
```

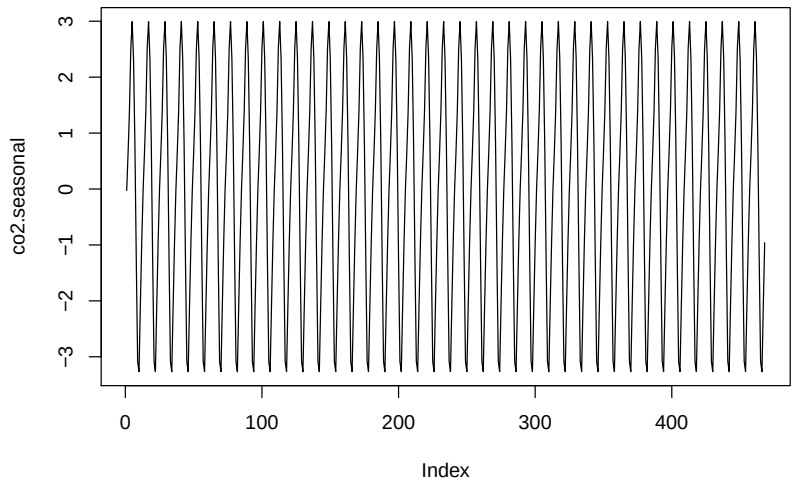


Mivel a szezonális változás nulla érték körül ingadozik, azaz a várható értéke nulla. A detrended értéket csökkentve, a trendet pedig növelve a fent számolt várható értékkel a `decompose()` függvénnyel kalkulált *trend* és *season* értékekhez juthatunk.

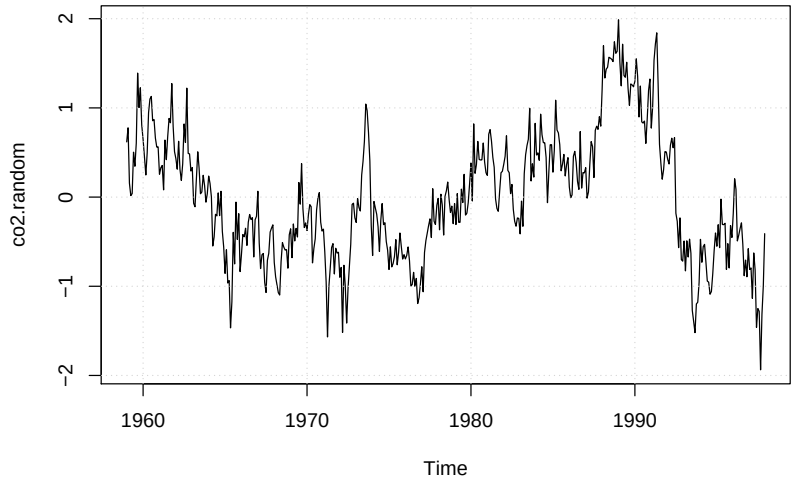
1.11.1. Szezonalitás megszüntetése

A szezonalitást az periódusonkénti szezonális értékek kivonásával kaphatjuk meg.

```
co2.seasonal <- rep(co2.season, length(co2) / frequency(co2))
plot(co2.seasonal, type = "l")
```



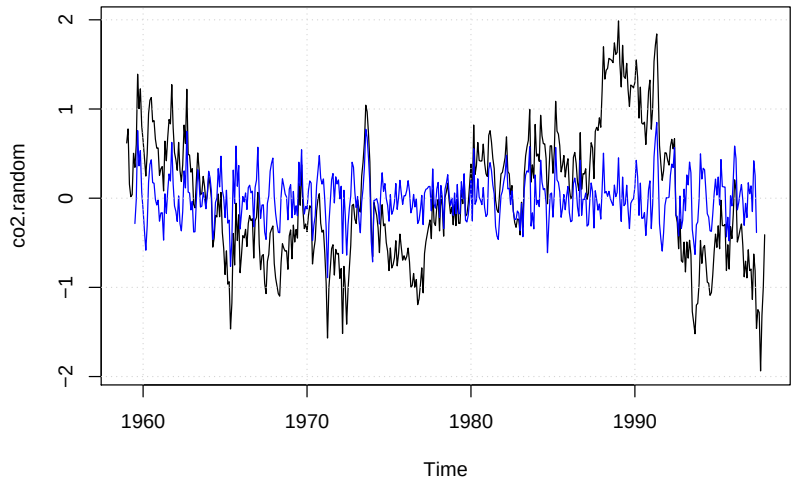

```
# A szezonalitás megszüntetése
co2.random <- co2.detrended - co2.seasonal
co2.random <- ts(co2.random, start = start(co2), frequency = frequency(co2))
plot(co2.random)
grid()
```



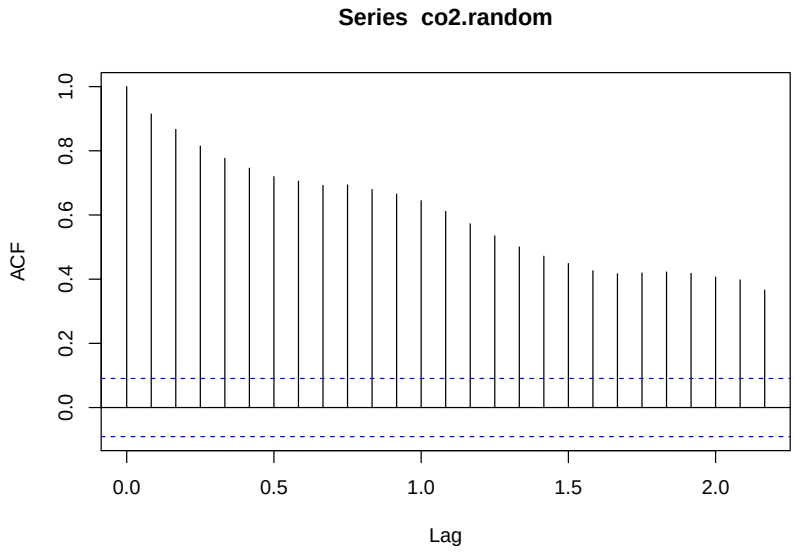
1.12. Véletlen változás

Determinisztikus esetben a véletlen, irreguláris változást a detrended, valamint a kiátlagolt szezonális változás különbségeként számoljuk.

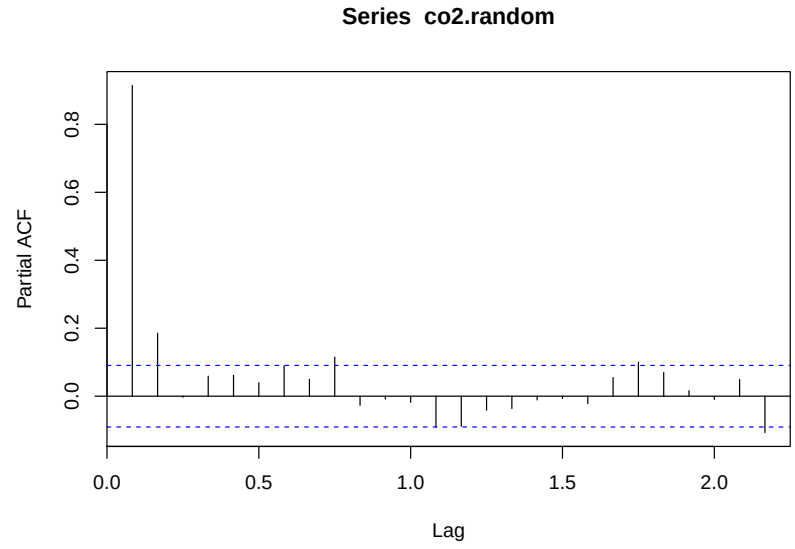
```
#co2.random
plot(co2.random)
lines(co2.decomp$random, col = "blue")
grid()
```




```
acf(co2.random, na.action = na.pass)
```




```
pacf(co2.random, na.action = na.pass)
```



1.13. Predikció

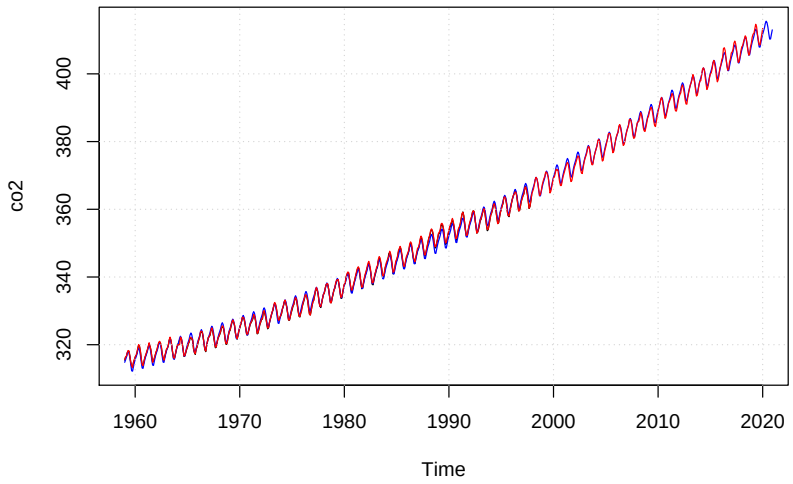
A predikció nem más, mint egy idősor jövőbeli adatainak előrejelzése a már regisztrált minták alapján. A jövőbeli értékek előrejelzését csak bizonyos valószínűséggel, megbízhatósággal tehetjük meg.

A predikciót elvégezhetjük az előző fejezetekben származtatott determinisztikus komponsensek, vagy sztochasztikus modell alkalmazásának segítségével.

A következőkben prediktáljuk a determinisztikus komponensek segítségével a CO2 koncentráció értékét. Ennek során a trend és szezonális értékekek alapján határozzuk meg a jövőbeli értékeket.

```
co2.pred.t <- time(ts(1, start = start(co2), end = c(2020,12), frequency = 12))
# trend és szezonális változás összegzése
co2.pred <- predict(co2.fit, newdata = data.frame(t = co2.pred.t))
co2.pred <- co2.pred + rep(co2.season, 2020-1959+1)
# idősor létrehozása
co2.pred.ts <- ts(co2.pred, start = start(co2), end = c(2020,12), frequency = 12)
# /// valós adatok
co2.real <- read.csv("data/co2-2020b.csv", sep=",", head=FALSE)
```

```r
co2.realts <- ts(co2.real[,3], start=c(1959,1), frequency=12)
# ábrázolás
plot(co2, xlim = c(1959, 2020), ylim = c(min(co2.pred), max(co2.pred)))
lines(co2.pred.ts, type = "l", col = "blue")
lines(co2.realts, col="red")
grid()
```



Azon eredményre jutottunk, hogy a determinisztikus predikció meglepően jól közelíti a 2020-ig mért tényleges értékeket. Tehát az analízis alapján meghatározott trendet a valóság is igazolta. Ez általában nem igaz az idősorokra, mivel a véletlen modellező hatása teljesen hiányzik a megközelítésben.

Sztochasztikus esetben az esetben az idősor adatokra, mint sztochasztikus folyamatra tekintünk. Ennek során felhasználjuk a véletlen modellépítő hatását is.

Egy **szochasztikus folyamat** egy véletlenszerű folyamat, melyet valószínűségi változókkal jellemezhetünk. Számos sztochasztikus folyamat létezik, ezek közül tekintsük át a leginkább jellemzőket.

- A *fehér zaj* (White Noise) egy olyan sztochasztikus folyamat, megy független azonos eloszlású (i.i.d) valószínűségi változókból áll wt. A várható értéke $\mu = 0$, a szórása korlátos σ_w^2. Jelölése $w_t \sim wn(0, \sigma_w^2)$. A fejér zaj a sztochaszikus viselkedés, így az idősorok sztochasztikus leírására is jól használható.
- A *Gauss-i fehér zaj* (Gaussian White Noise) egy olyan fehér zaj, melyben a változók normális eloszlásúak. Jeölése $w_t \sim N(0, \sigma_w^2)$.
- A *véletlen bolyongás* (Random Walk) egy folyamat, mely esetében az egyes szomszédos változók közötti differencia véletlen folyamatot követ. Amennyiben megengedjük a várható érték változását (δ, drift), valamint a kezdő érték 0 ($X_0 = 0$) a folyamat a következőképpen írható le.

$$X_t = X_{t-1} + w_t + \delta = \delta t + \sum_{i=1}^{t} w_i$$

Egy folyamatot **stacionárius**, melynél az egyes minták eloszlása független az adott időponttól, tehát egy adott minta értékeinek bekövetkezési valószínűsége független az időtől.

Egy *folyamat értékei közötti összefüggését* stacionárius szotchasztikus folyamatok esetében **autokorreláció függvény** segítségével vizsgálhatjuk. Az autokorrelációs függvény ugyanazon

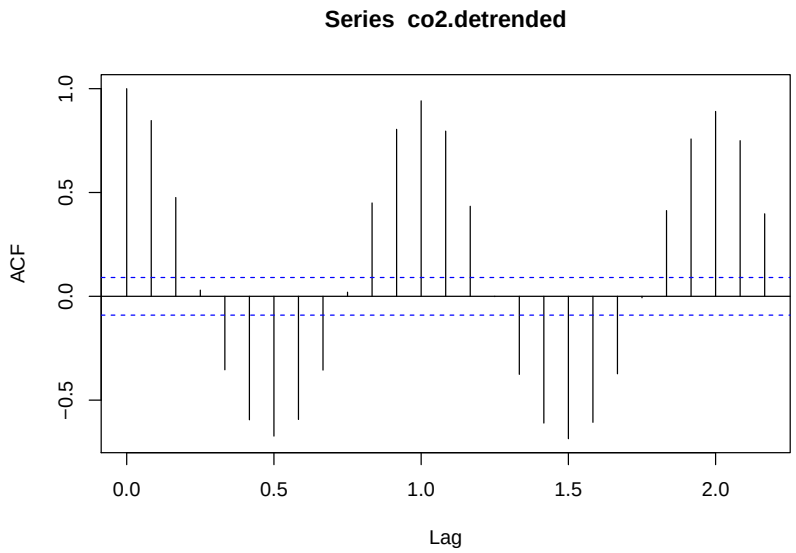
mintahalmaz időben eltolt változatával való korrelációjaként definiálhatjuk.

$$r_{X_t,k} = \frac{\sum_{i=1}^{N-k}(x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

Hogy a szezonális változást vizsgálhassuk alakítsuk át az idősorunkat stacionáriussá, azaz szüntessük meg azokat a mennyiségeket, melyek elrontják a stacionárius viselkedést. Ezt jellemzően a trend kiküszöbölésével (detrending) valósíthatjuk meg.

A következőkben nézzük meg a szezonális változás autokorreláció függvényét:

```
library(forecast)
acf(co2.detrended)
```



Egy véletlen folyamatot stacionáriussá tehetünk a differenciaképzés segítségével. Ennek során az időben szomszédos minták közötti különbségekből képezük új véletlen folyamatot. Ezt a `diff()` függvénnyel végezhetjük el.

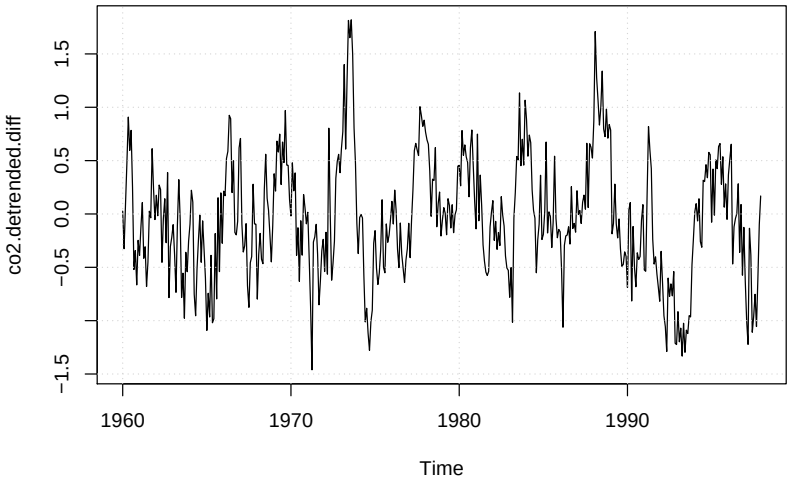
Az `nsdiff()` függvény segítségével megtudhatjuk, hogy egy adott folyamat esetében hányszor kell differeciát képeztni ahoz, hogy a folyamat stacionárius legyen.

A következőkben vizsgáljuk meg a detrended folyamat stacionaritását és tegyük a folyamatot stacionáriussá.

```
library(forecast)
nsdiffs(co2.detrended)
```

```
## [1] 1
```

```
co2.detrended.diff <- diff(co2.detrended, lag = 12, differences = 1)
plot(co2.detrended.diff)
grid()
```



Ezzel szemben a véletlen folyamatunkra a következő adódik.

```
library(forecast)
nsdiffs(co2.random)
```

```
## [1] 0
```

Azaz a véletlen folyamatunk már stacionárius.

1.13.1. Autoregresszív modell (AR)

Az autoregresszív modell azzal a feltételezéssel él, hogy egy adott idősor minta (X_t) kifejezhető p számú korábbi minta ($X_{t-1}, X_{t-2}, ...X_{t-p}$) lineáris kombinációjaként egy véletlen hiba erejéig.

Az AR(p) modellt a következőképpen formalizálhatjuk:

$$X_t = \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + ... + \alpha_p X_{t-p} + w_t = \sum_{i=1}^{p} \alpha_i X_{t-i} + w_t$$

1.13.2. Mozgó átlag modell (MA)

A mozgó átlag modell, habár kifejezésében és nevében is hasonlít, nem összekeverendő a mozgó átlag simítással (ld. fent). Az MA modell feladata, hogy az aktuális és korábbi időpontokban lévő az additív zajt $w_t, w_{t-1}, ..., w_{t-q}$ is figyelembe vegye.

Ennek alapján a mozgó átlag modell MA(q) a következőképpen definiálható:

$$X_t = w_t + \beta_1 w_{t-1} + ...\beta_q w_{t-q} = w_t + \sum_{j=1}^{q} \beta_j w_{t-j}$$

Két fontos megállapítás:

- Az egyes hibatagok (w_t) nem megfigyelhetők
- Az MA modell mindenképpen stacionárius, hiszen a megfigyelés csak független azonos eloszlású tagok súlyozott átlagai

1.13.3. ARMA modell

Egy (x_t) idősor adat ARMA(p,q) modell szerinti, amennyiben stacionárius és

$$X_t = w_t + \sum_{i=1}^{p} \alpha_i X_{t-i} + \sum_{j=1}^{q} \beta w_{t-j}$$

1.13.4. Az ARIMA modell

Látható, hogy az ARMA modell alkalmazásához szükséges, hogy a folyamat stacionárius legyen. Egy véletlen folyamat stacionáriussá tehető a következő módszerekkel:

- A trend kivonása (detrending)
- Differenciálással: Mivel a differenciálás megszünteti a folymat egyes időpontja közötti várható érték változásokat, és ezeket egymás után alkalmazva a linárisnál magasabb hatványú összetevők is eltüntethetők, a többszörös differenciálással a folyamat stacionáriussá tehető.

Az ARIMA (Auto-Regressive Integrated Moving Average) modell a következő módszereket tartalmazza:

1. AR (Auto Regression): Olyan regressziós modell, mely a mintahalmaz adott pontja és megelőző pontjai közötti összefüggést tartalmazza.
2. I (Integraged): A minták differenciáját képzi annak érdekében, hogy a megfigyeléseket stacionáriussá tegye.
3. MA (Moving Average): Egy olyan modell, mely kihasználja a megfigyelések és a modell közötti hibák, valamint a korábbi megfigyelések mozgó átlaga közötti összegüggéseket.

Az ARIMA modell három számmal jellemezhető:

1. p: Az AR modell hány korábbi változót vesz figyelembe.
2. d: Az I modellben milzen szintű a differenciálás.
3. q: Az MA modell milyen nagyságú ablakot használ.

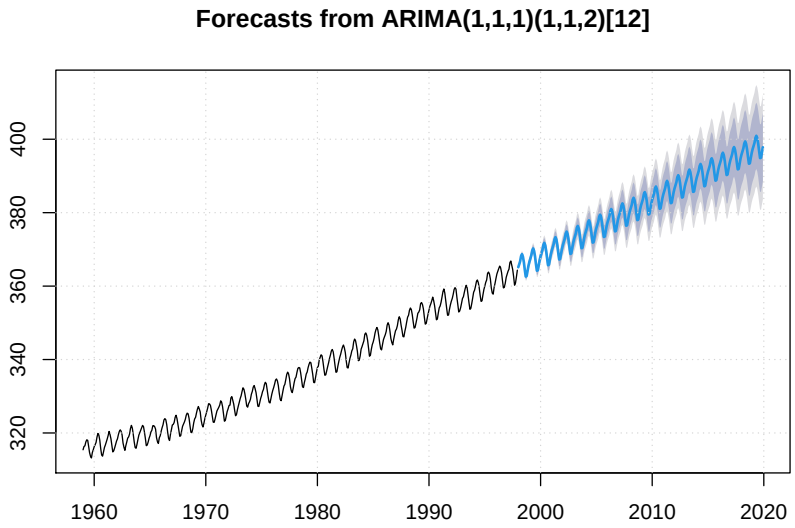
Az ARIMA modell alkalmazásának lépései a következők:

1. Modell azonosítás: A stacionáriusság és a szezonalitás ellenőrtése. Amennyiben szükséges differenciálás alkalmazása, az ARIMA modell p, d, q értékeinek meghatározása.
2. Paraméter becslés: A paraméterek (α_i, β_i) becslése. Erre a gyakorlatban kétféle lehetőség van:

- Maximum Likelihood Estimation (ML)
- Non-linear Least-Squares Estimation (NL-LS)

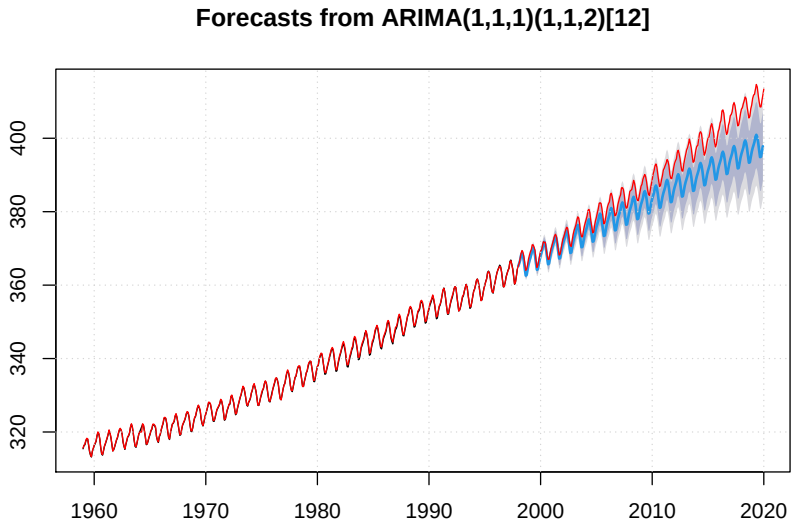
A következőkben végezzük el a `co2` adathalmazon a stacionárius folyamat megalkotását a `forecast` csomagban található `auto.arima()` függvénnyel, majd végezzük el a predikciót 2020-ig, azaz 1998-hoz képest $22 * 12$ hónapig a `forecast()` függvény segítségével.

```
library(forecast)
co2.model <- auto.arima(co2)
co2.forecast <- forecast(co2.model, h = 22*12)
plot(co2.forecast)
grid()
```



A becsült értékeket hasonlítsuk össze a ténylegesen regisztrált értékekkel.

```
co2.real <- read.csv("data/co2-2020b.csv", sep=",", head=FALSE)
co2.realts <- ts(co2.real[,3], start=c(1959,1), frequency=12)
plot(co2.forecast)
lines(co2.realts, col="red")
grid()
```



1.14. Gyakorló feladatok

Végezzük el a jegyzetben részletezett determinisztikus dekompozíciót, valamint determinisztikus és sztochasztikusa predikciót a következő adathalmazokra.

1) *AirPassangers*: Az adathalmaz a `data(AirPassangers)` utasítással tölthető be.

2) *Kings*: Az adathalmaz egymás utáni királyok életkorát tartalmazza és a következő utasításokkal tölthető be:

```
kings <- scan("/opt/datasets/kings.dat")

kings <- ts(kings)
```

3) *Births*: Az adathalmaz 1946-tól a NY-i születések számát mutatja havonta.

```
births <- scan("data/births.dat")
```

```
births <- ts(births, start=c(1946,1), frequency=12)
```

4) *Shop*: Az adathalmaz egy tengerparti shop forgalmát mutatja.

```
shop <- scan("data/shop.dat")

shop <- ts(shop, start=c(1987,1), frequency=12)
```

5) *EuStockMarkets*: Az adathalmaz négy tőzsde árfolyamindexét mutatja. Egyesével érdemes a vizsgálatokat elkészíteni.

```
data("EuStockMarkets")
```