

Figyelem!

Minden itt található kód megtalálható a `melléklet.txt`-ben!

0. Feladat:

Tisztelt Hallgató!

Ha ezt olvassa, akkor remélem sikeresen letöltötte a feladatsort.

Figyelmesen olvassa el a feladatokat és csak arra válaszoljon amire kell.

Az R felületén futassa le a következő R kódot, amelybe be kell helyettesítenie a saját NEPTUN kódját az `xyz63v` helyére (kisbetűket és számjegyeket használjon!

A kapott `ax`, `ay`, `az`, `av`, `ss`, `ev`, részvény értékeket jegyezze meg. Sőt a megoldás fájlba ezeket is adja meg!

Mielőtt elkezdjük a ZH megoldását, generálnunk kell adatokat amelyekkel dolgozni tudunk. A következőhöz hasonló kódot fog majd biztosítani a vizsgáztató, amelyet a Console felületére kell bemásolni, ezzel megkapva a ZH-hoz használandó paramétereket:

```
> x="y86i0i";#neptun kód
> z=charToRaw(iconv(x, "latin1", "UTF-8"))
> for (i in 1:6) v=paste("0x",z,sep="")
> e=strtoi(v)
> ax=e[1];ay=e[2];az=e[3];av=e[4];ss=sum(strtoi(v))+9
> cat("ax=",ax,"\n")
ax= 121
> cat("ay=",ay,"\n")
ay= 56
> cat("az=",az,"\n")
az= 54
> cat("av=",av,"\n")
av= 105
> cat("ss=",ss,"\n")
ss= 498
> ar=c("FB","AAPL","AMZN","GOOG","NFLX","TSLA")
> ai=ss-6*floor(ss/6)
> ev=2019-(ss-10*floor(ss/10))
> cat("ev=",ev,"\n")
ev= 2011
> cat("reszveny=",ar[ai+1],"\n")
reszveny= FB
> |
```

Feladat paramétereinek generálása

Ne felejtjük el, hogy a ZH-t valamilyen dokumentumban is kell rögzíteni (pontosabban a megoldások kódjait, adatait, paramétereit), így célszerű feladatról feladatra vezetni egy word dokumentumot. Ennek formátuma nem kötött, de ahogy a feladat is kijelentette:

“Sőt a megoldás fájlba ezeket is adja meg!”

Úgyhogy figyeljetelek oda ezekre, ne így menjenek el a pontok!

A kód Console-ban történő futtatását követően haladhatunk tovább az első feladatra, amely a korábban generált paramétereket felhasználja majd!

1. Feladat

Készítse el a következő 1100 elemű kétdimenziós mintarealizációt:

A generálás előtt állítsa be a `set.seed(ss)` értéket.

```
set.seed(ss)
nx=600
v=matrix(c(ax,abs(ax-ay),abs(ax-ay),ay),2)
w=chol(v)
z1=-log(runif(nx))
z2=-log(runif(nx))
zm=matrix(c(z1,z2),ncol=2)
zn=zm%*%w
```

Az első feladat megint csak könnyű eset, egyszerűen csak másoljuk ki a mintarealizációt létrehozó kódrészletet, és futtassuk le!

```
> set.seed(ss)
> nx=600
> v=matrix(c(ax,abs(ax-ay),abs(ax-ay),ay),2)
> w=chol(v)
> z1=-log(runif(nx))
> z2=-log(runif(nx))
> zm=matrix(c(z1,z2),ncol=2)
> zn=zm%*%w
```

Mintarealizáció generálása

Ha megvizsgáljuk a **zn** mátrixot - tehát a mátrixot amely tartalmazza a feladatok megoldásához szükséges adatokat - rájövünk, hogy Sanyi bácsi nem 1100 hanem 1200-as méretű mátrixot generáltatott ki velünk:

```
> length(zn)
[1] 1200
Rare Sanyi bácsi L
```

Sebaj, ez nem hátráltat minket a további munkában!

2. Feladat

Készítsen az előző kétdimenziós **zn** mintarealizációról statisztikai elemzést, azaz becsülje meg a paramétereket, ferdeséget, lapultságot! Továbbá vizsgálja meg a peremek függetlenségét!

Itt hirtelen úgy tűnhet, hogy sokat kér a feladat, így bontsuk részekre! Először is, csináljunk a **zn** mátrixról **általános statisztikai elemzést**:

```
> summary(zn)
      V1      V2
Min.   : 0.02409 Min.   : 0.4419
1st Qu.: 3.06816 1st Qu.: 5.4005
Median : 7.51216 Median : 9.3168
Mean   :10.91901 Mean   :10.6251
3rd Qu.:15.39214 3rd Qu.:14.1196
Max.   :79.04646 Max.   :52.8759
.
```

Általános statisztikai elemzés

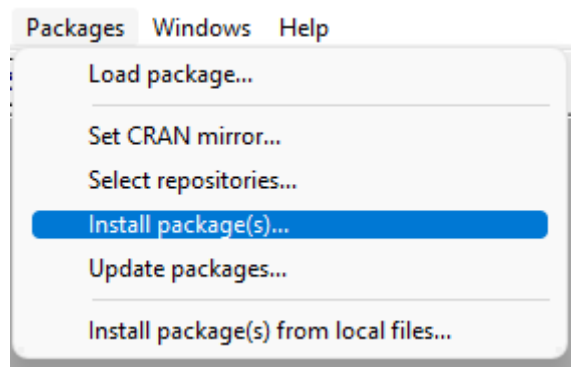
Az elemzés részleteit érdemes rögzíteni a megoldást tartalmazó fájlunkba:

- **Min.** : Az adott vektorban tapasztalt legkisebb érték
- **1st Qu.** : Első kvartilis, a megfigyelések 25%-a ettől az értéktől kisebb

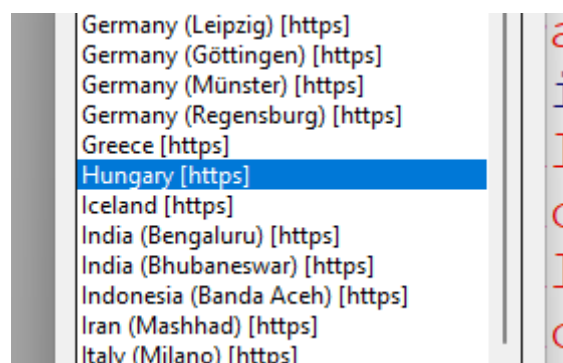
- **Median** : Az adott vektorban tapasztalt medián
- **Mean** : Átlag, (ínyenceknek empirikus közép)
- **3rd Qu** : Harmadik kvartilis, a megfigyelések 75%-a ettől az értéktől kisebb
- **Max**: Az adott vektorban tapasztalt legnagyobb érték

Következő részfeladat a paraméterek ferdeségének megállapítása, amelyhez a **skewness()** parancsot érdemes használni, ez viszont nem jár az R alapvető verziójával. Nem esünk kétségbe: A **skewness()** parancs az **moments** csomagban található, így ezt le kell töltenünk! (Mindenféleképpen érdemes először kipróbálni, hogy a **skewness()** parancs elérhető-e, mielőtt időt pazarolunk a csomag letöltésére!)

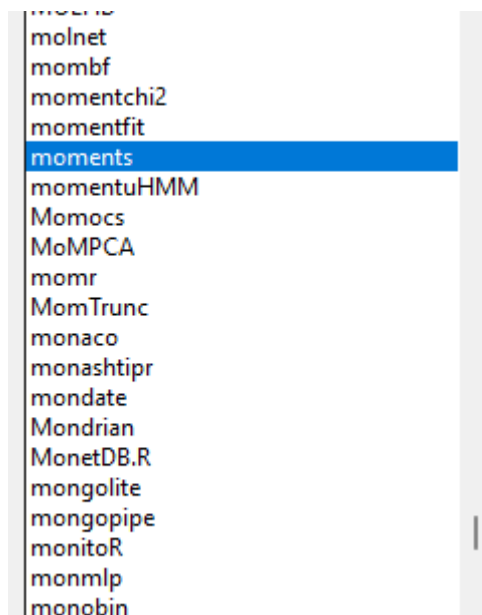
A csomagot a következőképpen tudjuk beszerezni:



moments csomag beszerzése (1)



moments csomag beszerzése (2)



moments csomag beszerzése (3)

Az "OK" gombra kattintva elindul a gyors letöltés, és már használható is a csomag a következőképpen:

```
> library(moments)
> skewness(zn)
[1] 1.927752 1.477755
```

Ferdeség megállapítása

A **lapultság** meghatározását a **kurtosis()** parancs hajtja végre, amely ugyancsak az moments csomagot hívja segítségül:

```
> kurtosis(zn)
[1] 8.271304 6.633745
```

Lapultság vizsgálata

Az utolsó részfeladat a peremek függetlenségét kéri. Erre a **cor()** parancs ad megoldást, amelyet így használunk:

```
> cor(zn)
      [,1]      [,2]
[1,] 1.0000000 0.7736075
[2,] 0.7736075 1.0000000
```

Peremek függetlenségének vizsgálata

3. Feladat

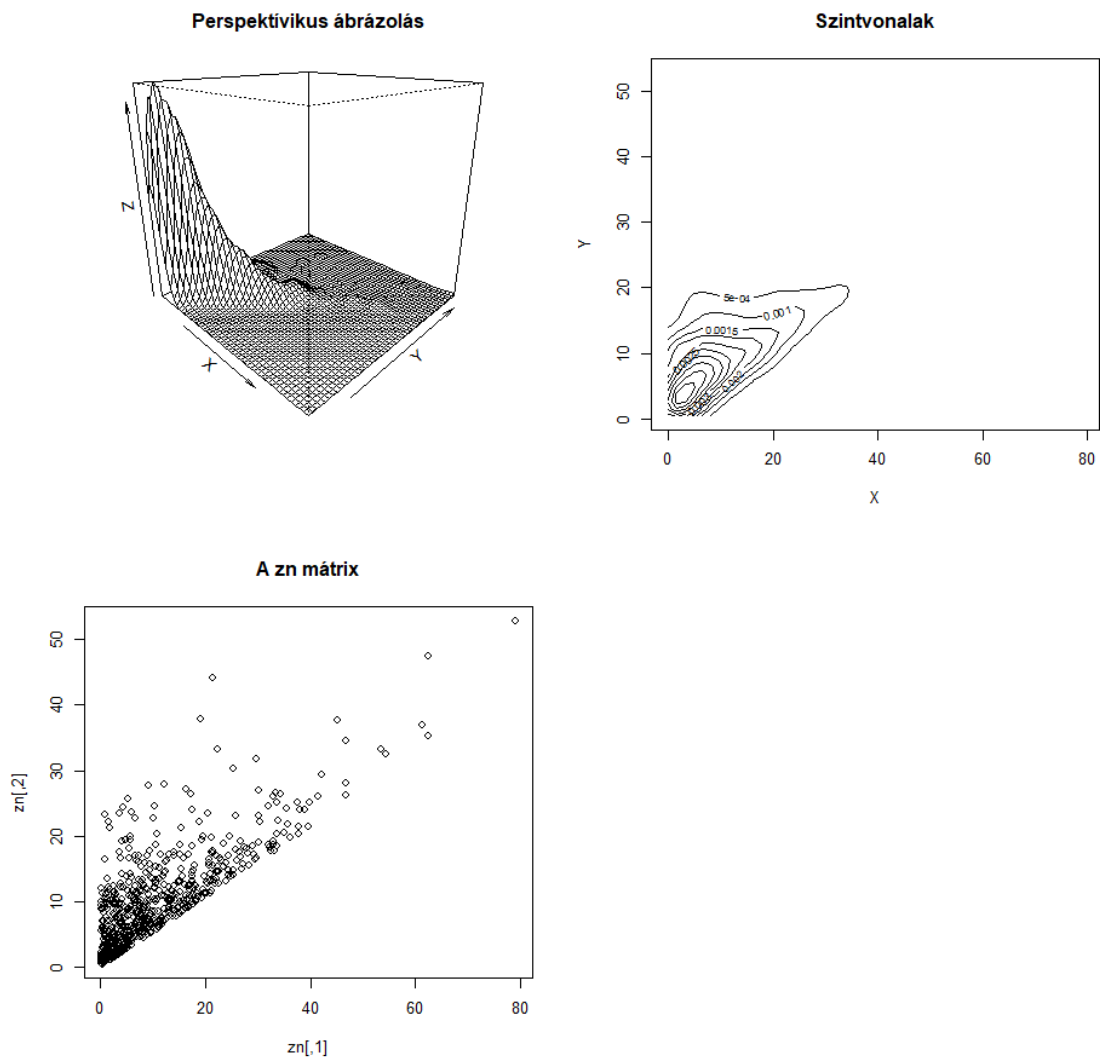
Ezenkívül készítsen többdimenziós ábrázolást szintvonalakkal és perspektívikusan is (feliratozással, a kétdimenziós eloszlás jól látható legyen) !

A szintvonalak ábrázolását a **contour()** paranccsal, míg a perspektivikus ábrázolást a **persp()** paranccsal tudjuk megvalósítani. Mivel érdemes az ábrázolásokat egy ablakban prezentálni, a **par()** parancs segítségével hívjuk, amelynek meg tudunk adni egy mátrix méretet, amely alapján az feltudja osztani a plot ablakot $X * Y$ nagyságúra (teszem azt, 2x2 ablakban 4 db plot fér el). A megvalósítás a következő:

```
> # Adatok betöltése
> x <- zn[,1]
> y <- zn[,2]
>
> # Rács készítése
> grid_size <- 50
> x_range <- seq(min(x), max(x), length.out = grid_size)
> y_range <- seq(min(y), max(y), length.out = grid_size)
>
> # Kétváltozós sűrűség becslése
> library(MASS) # csomag a kde2d() függvényhez
> fhat <- kde2d(x, y, n = grid_size)
>
> # Plot-ok 2x2-es elhelyezése
> par(mfrow=c(2,2))
>
> # Perspektivikus ábrázolás
> persp(x_range, y_range, fhat$z, theta = 45, phi = 20, xlab = "X", ylab = "Y", zlab = "Z",
+ main="Perspektivikus ábrázolás")
>
> # Szintvonalas ábrázolás
> contour(x_range, y_range, fhat$z, xlab = "X", ylab = "Y", main="Szintvonalak")
> plot(zn, main="A zn mátrix")
```

Szintvonalak, perspektivikus ábrázolás megvalósítása

és az eredmény:



A kiszámított adatokból kapott ábrák, feliratozva

4. Feladat:

Generáljon geometriai Brown folyamatot (várható érték: $\mu = ax$, szórás: $\sigma = (ax+ay)/(ax+ay+az)$ értékkel, az időintervallum 100 egység)!

A generálás előtt állítsa be a `set.seed(ss+27)` értéket.

Ábrázolja és vizsgálja meg a statisztikai jellemzőket!

Mielőtt elkezdenénk, figyelmesen olvassuk el a feladatot: A **Geometriai Brown folyamat** és a **Brown folyamat** két különböző gráf, nehogy

rosszat számoljunk! A feladat most Geometriai Brown folyamatot kér tőlünk, de a Brown folyamat is benne van a mellékletek között!

Story time a megértéshez:

A Brown folyamatok a részecskék mozgásának véletlenszerűségét használja ki annak reményében, hogy ha ezt a véletlenszerű mozgást ráillesztjük egy részvény ismert adataira, sikeresen tudjuk előre jelezni a részvény árának haladását. Természetesen ez is csak egy becslés, így nem lehetünk milliomosok belőle! Inkább arra használják, hogy szimuláljanak egy hiteles részvény haladást.

A Geometriai Brown folyamatot kétféleképpen tudjuk generálni, csomaggal vagy anélkül.

a) Csomag megoldás

Az **LSMRealOptions** nevű R könyvtár tartalmaz egy úgynevezett **GBM_simulate()** függvényt, amely célja egy Geometrikus Brown folyamat szimulálása. A használatához ismernünk kell a paramétereit.

Használat:

```
GBM_simulate(n, t, mu, sigma, S0, dt)
```

Paraméterek:

n - Mennyi szimulációt szeretnénk

t - A becslés mérete években (az utolsó becslés napja mikor van)

mu - A Geometriai Brown folyamat várható értéke

sigma - A Geometriai Brown folyamat szórása

S0 - A szimuláció kezdő értéke (részvény ára)

dt - Megfigyelések nagysága. A kezdő értéktől (**S0**) mekkora lépésekkel érünk el az utolsó naphoz (**t**).

Példák:

Csináljunk egy szimulációt egy részvény 1 éven keresztül tartó haladására úgy, hogy havonta egyszer vizsgálódunk!

```
set.seed(1) <- 1 # Reprodukálás
```

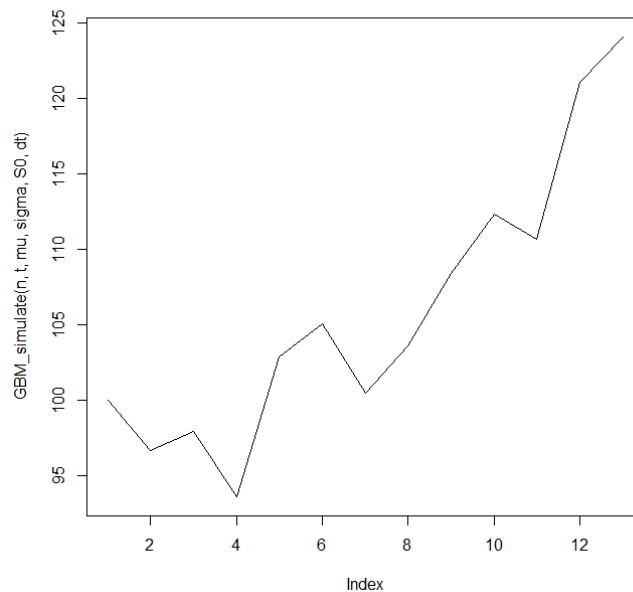
```
n <- 1 # Egy szimuláció kell most csak
```



```

t <- 1 # Egy évre szeretnénk előrevetíteni
mu <- 0.05 # Várható érték
sigma <- 0.2 # Szórás
S0 <- 100 # A részvény kezdő értéke
dt <- 1/12 # Havonta egyszer vizsgálódunk
plot(GBM_simulate(n, t, mu, sigma, S0, dt), type='l')

```



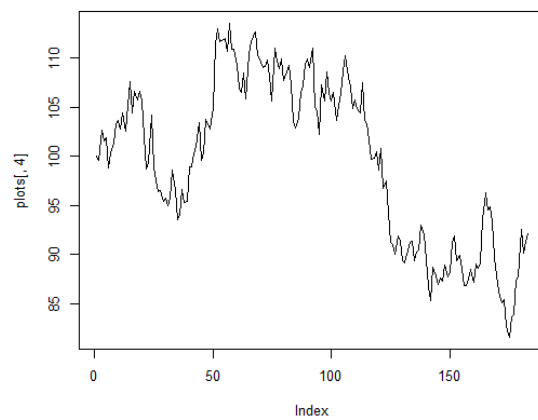
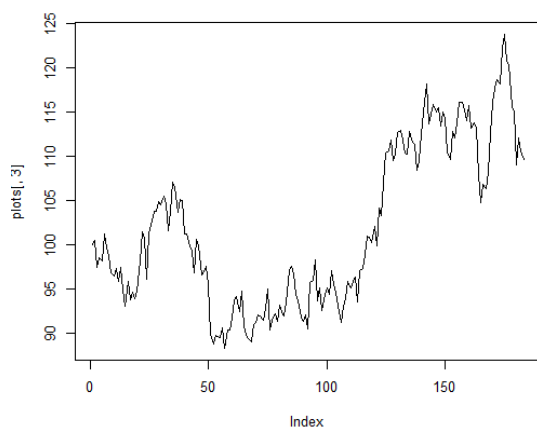
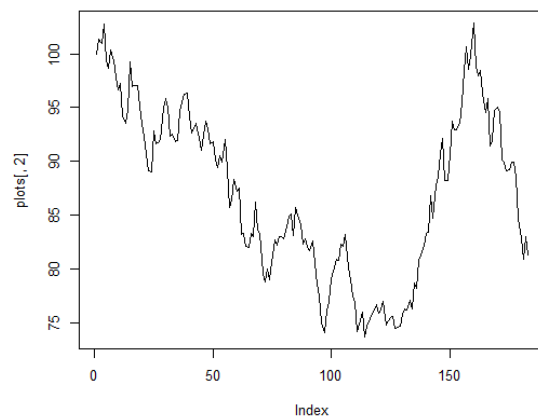
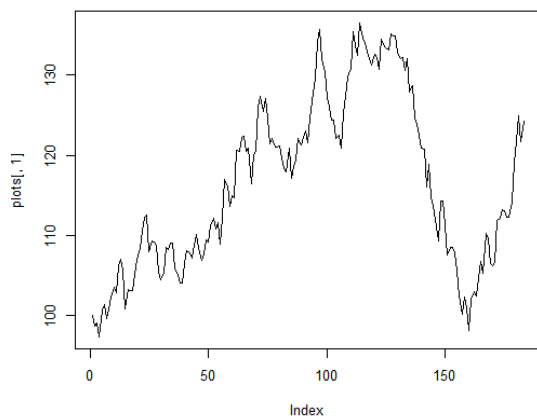
Eredmény

Csináljunk négy szimulációt egy részvény fél éven keresztül tartó haladására úgy, hogy naponta egyszer vizsgálódunk!

```

set.seed(1) # Reprodukálás
par(mfrow=c(2,2)) # 2x2 plot
n <- 4 # Négy szimuláció kell
t <- 1/2 # Félévre szeretnénk előrevetíteni
mu <- 0.09 # Várható érték
sigma <- 0.4 # Szórás
S0 <- 100 # A részvény kezdő értéke
dt <- 1/365 # Naponta egyszer vizsgálódunk
plots <- GBM_simulate(n, t, mu, sigma, S0, dt);
# Több plot kell, mivel több szimuláció
plot(plots[,1], type='l') # 1. generáció
plot(plots[,2], type='l') # 2. generáció
plot(plots[,3], type='l') # 3. generáció
plot(plots[,4], type='l') # 3. generáció

```



Eredmény

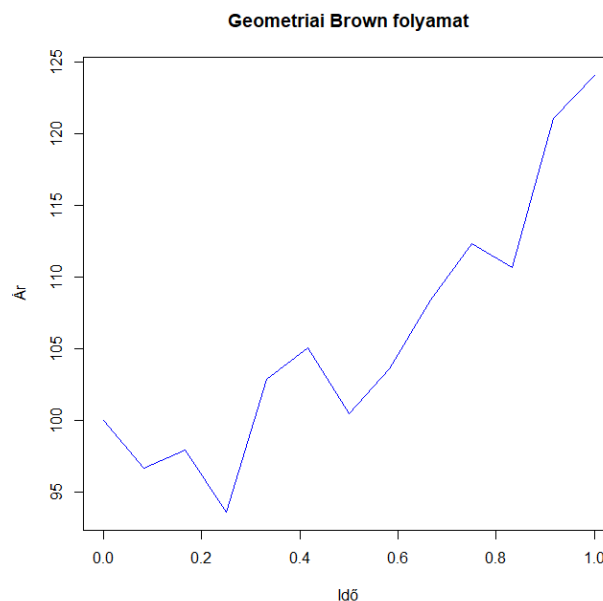
b) Kézi megoldás

A kézi megoldás ugyanazt csinálja mint a csomag, csak bele tudunk nézni a “motorháztető alá” és meg tudjuk vizsgálni mi is történik igazán. Ez vizuálisan szebb látvány / megoldás, nem csak egy command kiadása.

A kód (megtalálható a mellékletben):

```
> gbm <- function(mu, sigma, T, S0, dt) {  
+   set.seed(1);  
+   # Lépések megadása  
+   t <- seq(0, T, dt)  
+  
+   # Inicializáció  
+   S <- numeric(length(t))  
+   S[1] <- S0  
+  
+   # Generáljuk le a GBM szimuláció eredményeit  
+   for(i in 2:length(t)) {  
+     dW <- rnorm(1, mean = 0, sd = sqrt(dt))  
+     S[i] <- S[i-1]*exp((mu - 0.5*sigma^2)*dt + sigma*dW)  
+   }  
+  
+   # Rajzoljuk ki a gráfot  
+   plot(t, S, type = "l", main = "Geometriai Brown folyamat",  
+        xlab = "Idő", ylab = "Ár", col = "blue")  
+ }  
>  
> # Hívjuk meg a függvényt  
> gbm(mu = 0.05, sigma = 0.2, T = 1, S0 = 100, dt=1/12)
```

Kivitelezés (első példa megoldása)



Eredmény, 1:1 ugyanaz, mint a csomag megoldás első példája

Az így szerzett tudásunkkal oldjuk meg a feladatot!

Emlékeztetőként:

Generáljon geometriai Brown folyamatot (várható érték: $\mu = ax$, szórás: $\sigma = (ax+ay)/(ax+ay+az)$ értékkel, az időintervallum 100 egység)!

A generálás előtt állítsa be a `set.seed(ss+27)` értéket.

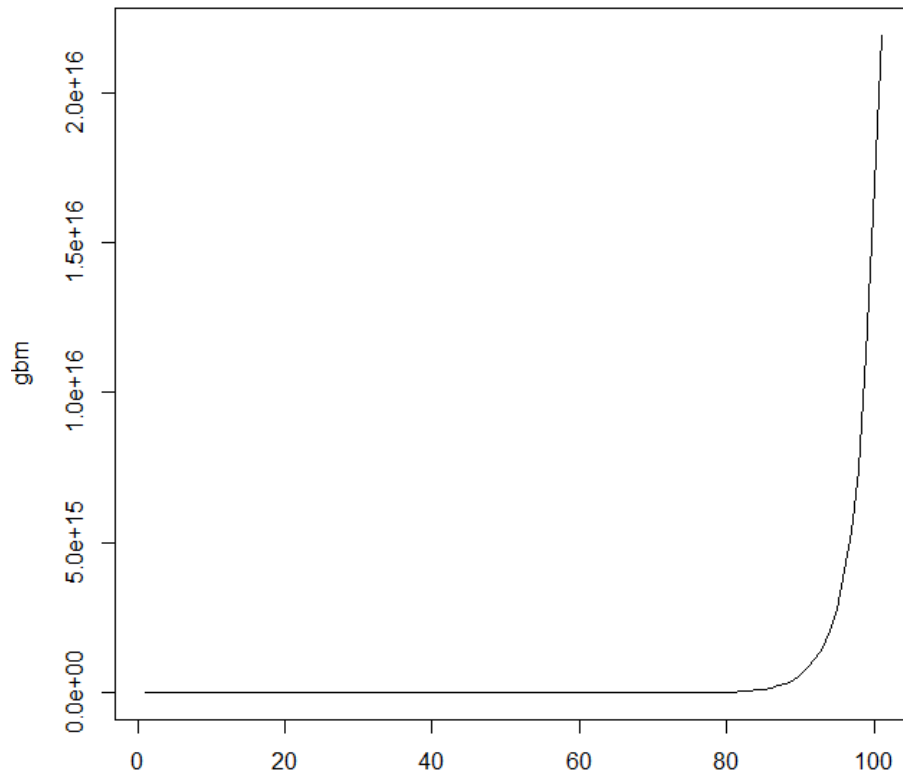
Ábrázolja és vizsgálja meg a statisztikai jellemzőket!

Csomag megoldás:

```
> set.seed(ss+27) # Reprodukálás
> n <- 1 # Egy szimuláció kell most csak
> t <- 100/365 # 100 napra szeretnénk előre vetíteni
> mu <- ax # Várható érték
> sigma <- (ax+ay)/(ax+ay+az) # Szórás
> S0 <- 100 # A részvény kezdő értéke
> dt <- 1/365 # Naponta egyszer vizsgálódunk
> gbm <- GBM_simulate(n, t, mu, sigma, S0, dt);
> plot(gbm, type='l')
> summary(gbm)
      Min.    1st Qu.    Median      Mean   3rd Qu.      Max.
1.000e+02 3.900e+05 1.532e+09 7.681e+14 5.292e+12 2.192e+16
> skewness(gbm)
[1] 5.134
> kurtosis(gbm)
[1] 27.89741
> |
```

Megoldás csomaggal

A kapott eredmény egy exponenciális görbe lesz a magas várható érték miatt, ami nem meglepő, mivel a Geometriai Brown folyamatot más néven Exponenciális Brown folyamatnak is nevezik, exponenciális természetéből kifolyólag.



Exponenciális (Geometriai) Brown folyamat

Ugyanígy oldjuk meg a feladatot a kézzel írt megoldással:

```
> gbm <- function(mu, sigma, T, S0, dt, ss) {
+   set.seed(ss+27);
+   # Lépések megadása
+   t <- seq(0, T, dt)
+
+   # Inicializáció
+   S <- numeric(length(t))
+   S[1] <- S0
+
+   # Generáljuk le a GBM szimuláció eredményeit
+   for(i in 2:length(t)) {
+     dW <- rnorm(1, mean = 0, sd = sqrt(dt))
+     S[i] <- S[i-1]*exp((mu - 0.5*sigma^2)*dt + sigma*dW)
+   }
+
+   # Rajzoljuk ki a gráfot
+   plot(t, S, type = "l", main = "Geometriai Brown folyamat",
+        xlab = "Idő", ylab = "Ár", col = "blue")
+   return(S);
+ }
>
> # Hívjuk meg a függvényt
> S <- gbm(mu = ax, sigma = (ax+ay)/(ax+ay+az), T = 100/365, S0 = 100, dt=1/365, ss)

> summary(S);
      Min.    1st Qu.      Median        Mean     3rd Qu.        Max.
1.000e+02 3.900e+05 1.532e+09 7.681e+14 5.292e+12 2.192e+16
> skewness(S);
[1] 5.134
> kurtosis(S);
```

Kézi megoldás

A kódot kissé átszerkesztettem annak érdekében, hogy az eredményeket tudjuk statisztikailag elemezni. A változtatás annyiban merül ki, hogy az S paramétert (amely az árakat tartalmazza) **return()**-el visszaadom, majd kint meghívom rá a már megszokott statisztikai elemző metódusokat.

5-6. Feladat

Töltse le az R kód futtatásával kapott részvény adatait a

<https://finance.yahoo.com/quote/reszveny/history?p=reszveny>

honlapról az ev változó értékének megfelelően (január 01-től december 31-ig)!

Vizsgálja meg milyen eloszlású a napi záró árak megváltozásának logaritmusa (javasolt a logreturn, azaz $\ln(x_{(n+1)}/x_n)$ értékek vizsgálata) (minimum khi négyzet próba, ez azt jelenti, hogy meg kell adni az eloszlást paraméterekkel)!

Grafikus ábrázolás, pontbecslések és intervallumbecslések!

Ha visszaemlékezünk, amikor az adatokat generáltuk a **0. Feladatban**, kaptunk egy "ar" változót is, amely a részvény nevét tartalmazza. Ennek a részvénynek a nevét kell beilleszteni a biztosított linkbe. Az én esetemben a "FB" (Facebook) részvényei jutottak, így az "FB"-t kell bemásolnom a link-be:

<https://finance.yahoo.com/quote/FB/history?p=FB>

(különleges eset lehet, hogy a FB mostmár META, de hála a jó égnek az FB-re is tudott reagálni a yahoo finance, így nem hiszem hogy időközben történő névváltoztatás itt bármilyen akadályt okozna)

Ez a felület fogad minket:

yahoo!
finance

Search for news, symbols or companies

Meta Platforms, Inc. (META)
NasdaqGS - NasdaqGS Real Time Price. Currency in USD

☆ Follow

245.64 -1.21 (-0.49%)
At close: May 19 04:00PM EDT

245.80 +0.16 (+0.07%)
After hours: 07:59PM EDT

Start Trading >>

Plus500 86% of retail CFD accounts lose money

[Summary](#) [Chart](#) [Conversations](#) [Statistics](#) **[Historical Data](#)** [Profile](#) [Financials](#) [Analysis](#) [Options](#) [Holders](#) [Sustainability](#)

Advertisement

AD

Webhosting kedvező árakon
Webhosting folyamatos támogatással, évtizedes tapasztalattal

Megnyitás

Time Period: [May 20, 2022 - May 20, 2023](#) Show: [Historical Prices](#) Frequency: [Daily](#) [Apply](#)

Currency in USD [Download](#)

Date	Open	High	Low	Close*	Adj Close**	Volume
May 19, 2023	247.47	248.69	243.41	245.64	245.64	21,599,800

Yahoo finance: META (korábban FB) részvények

Ha a feladatot tovább olvassuk, megtudjuk, hogy a generált adatok között ott van egy “ev” nevű változó is, amely nálam 2011. Kicsit korrigálok ezen, mivel a yahoo finance 2013-ig tartja számon a részvényeket, így ezzel az évvel dolgozom. A feladat azt is megmondja, hogy Január 1 és December 31 közötti információkat kell beszerezni, így megvan minden információ amire szükségünk van. Töltsük le a megfelelő adatokat!

Time Period: [May 20, 2022 - May 20, 2023](#)

Currency in USD

ate

May 19, 2023

May 18, 2023

May 17, 2023

May 16, 2023

1D 5D 3M 6M

YTD 1Y 5Y Max

Start Date

2013. 01. 01.

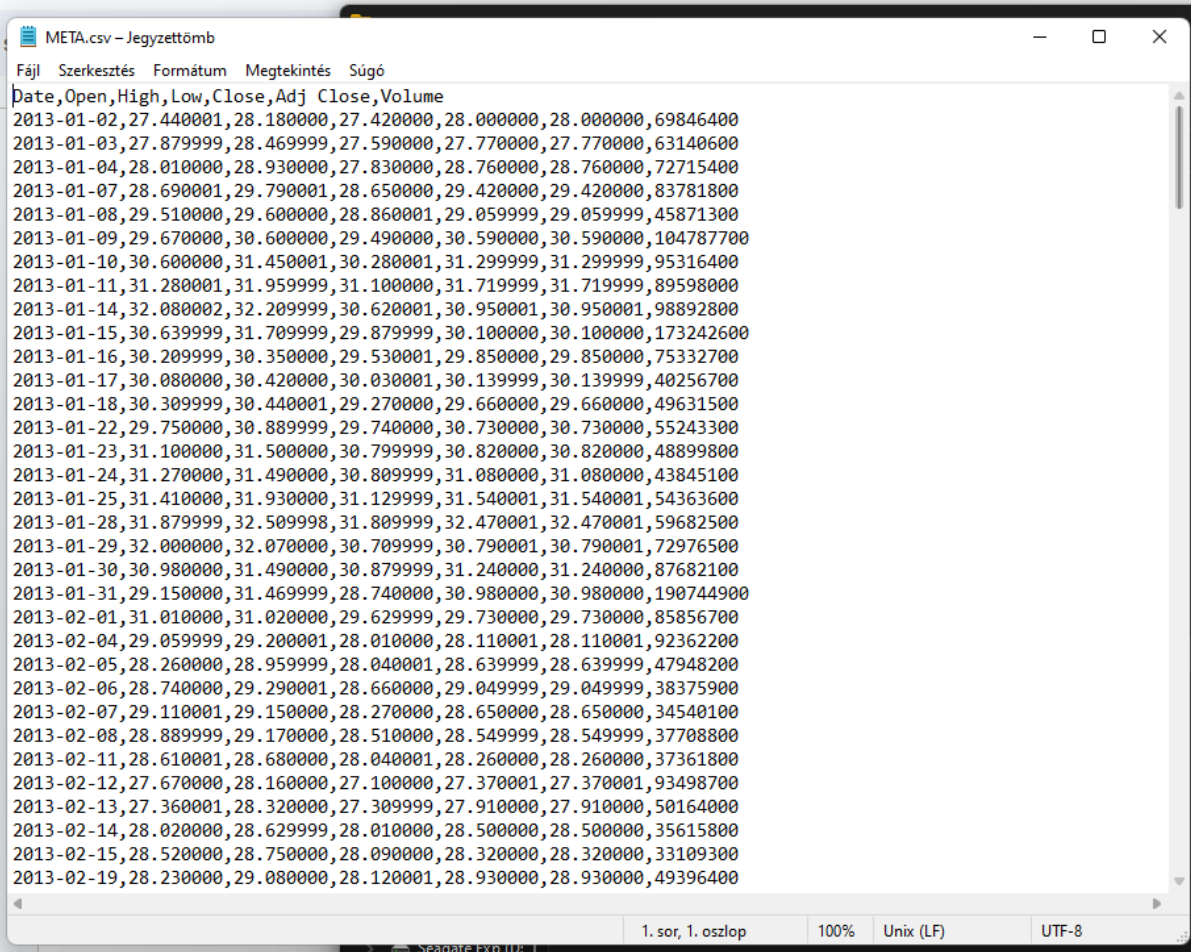
End Date

2013. 12. 31.

Done Cancel

Időperiódus kiválasztása

A dátum kiválasztása után nyomjunk a Download gombra, és letöltésre kerül egy **META.csv** fájl (természetesen ez mindenkinek a saját részvénye lesz, nálam most META). **Mielőtt elkezdünk dolgozni, bizonyosodjunk meg róla hogy a letöltött fájl tényleg a kívánt évnek megfelelő adatokat tartalmaz-e!**



Date	Open	High	Low	Close	Adj Close	Volume	
2013-01-02	27.440001	28.180000	27.420000	28.000000	28.000000	69846400	
2013-01-03	27.879999	28.469999	27.590000	27.770000	27.770000	63140600	
2013-01-04	28.010000	28.930000	27.830000	28.760000	28.760000	72715400	
2013-01-07	28.690001	29.790001	28.650000	29.420000	29.420000	83781800	
2013-01-08	29.510000	29.600000	28.860001	29.059999	29.059999	45871300	
2013-01-09	29.670000	30.600000	29.490000	30.590000	30.590000	104787700	
2013-01-10	30.600000	31.450001	30.280001	31.299999	31.299999	95316400	
2013-01-11	31.280001	31.959999	31.100000	31.719999	31.719999	89598000	
2013-01-14	32.080002	32.209999	30.620001	30.950001	30.950001	98892800	
2013-01-15	30.639999	31.709999	29.879999	30.100000	30.100000	173242600	
2013-01-16	30.209999	30.350000	29.530001	29.850000	29.850000	75332700	
2013-01-17	30.080000	30.420000	30.030001	30.139999	30.139999	40256700	
2013-01-18	30.309999	30.440001	29.270000	29.660000	29.660000	49631500	
2013-01-22	29.750000	30.889999	29.740000	30.730000	30.730000	55243300	
2013-01-23	31.100000	31.500000	30.799999	30.820000	30.820000	48899800	
2013-01-24	31.270000	31.490000	30.809999	31.080000	31.080000	43845100	
2013-01-25	31.410000	31.930000	31.129999	31.540001	31.540001	54363600	
2013-01-28	31.879999	32.509998	31.809999	32.470001	32.470001	59682500	
2013-01-29	32.000000	32.070000	30.709999	30.790001	30.790001	72976500	
2013-01-30	30.980000	31.490000	30.879999	31.240000	31.240000	87682100	
2013-01-31	29.150000	31.469999	28.740000	30.980000	30.980000	190744900	
2013-02-01	31.010000	31.020000	29.629999	29.730000	29.730000	85856700	
2013-02-04	29.059999	29.200001	28.010000	28.110001	28.110001	92362200	
2013-02-05	28.260000	28.959999	28.040001	28.639999	28.639999	47948200	
2013-02-06	28.740000	29.290001	28.660000	29.049999	29.049999	38375900	
2013-02-07	29.110001	29.150000	28.270000	28.650000	28.650000	34540100	
2013-02-08	28.889999	29.170000	28.510000	28.549999	28.549999	37708800	
2013-02-11	28.610001	28.680000	28.040001	28.260000	28.260000	37361800	
2013-02-12	27.670000	28.160000	27.100000	27.370001	27.370001	93498700	
2013-02-13	27.360001	28.320000	27.309999	27.910000	27.910000	50164000	
2013-02-14	28.020000	28.629999	28.010000	28.500000	28.500000	35615800	
2013-02-15	28.520000	28.750000	28.090000	28.320000	28.320000	33109300	
2013-02-19	28.230000	29.080000	28.120001	28.930000	28.930000	49396400	

2013-as META részvények adatai

Az adatok helyesek, haladjunk tovább! Használjuk az alábbi kódot a csv fájl beolvasására, valamint a logreturn értékek kinyerésére és elemzésére!

```
> details <- read.csv("E:/Letöltések/misc/gazd/META.csv") #csv fájl beolvasása
> logreturn = c() #logreturn létrehozása
> zaro <- details$Close #A záró értékek kinyerése
> for (i in 1:length(zaro)-1){
+   logreturn[i] = abs(log(zaro[i+1]/zaro[i])) #logreturn értékek begyűjtése
+ }
> chisq.test(logreturn) #Khi-négyzet teszt
```

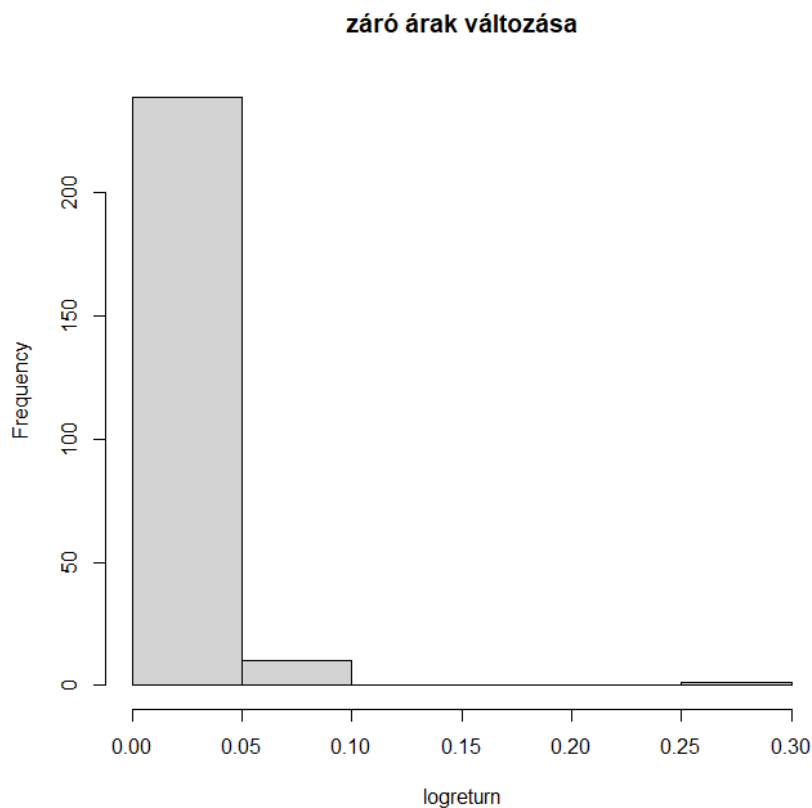
Chi-squared test for given probabilities

```
data: logreturn
X-squared = 5.8058, df = 249, p-value = 1
```

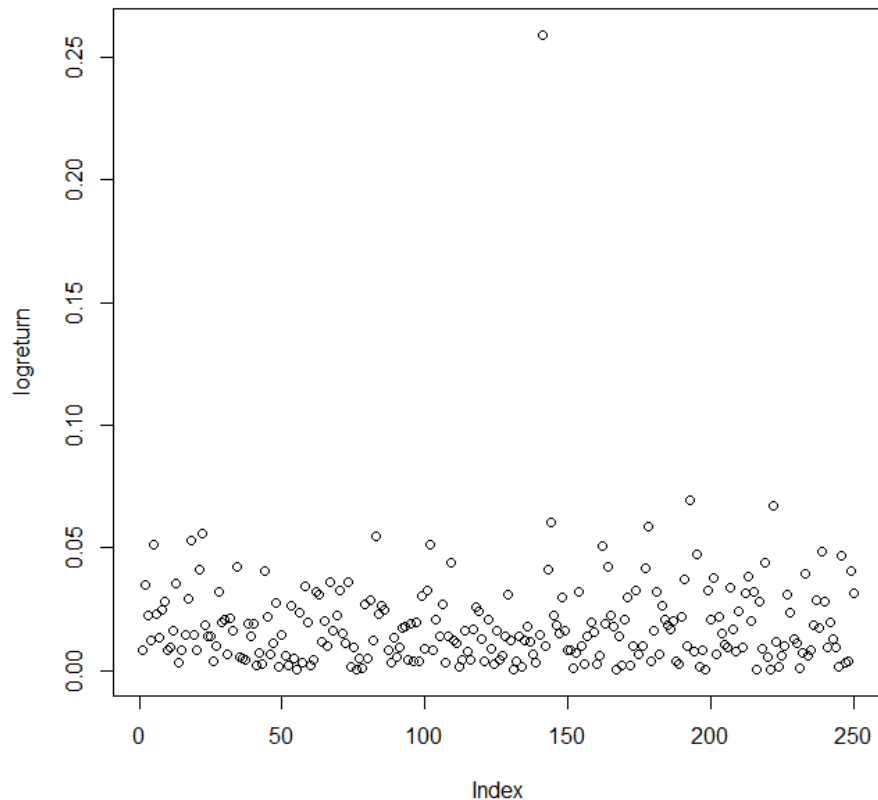
```
Warning message:
In chisq.test(logreturn) : Chi-squared approximation may be incorrect
>
> #Statisztikák
> hist(logreturn, main="záró árak változása")
> plot(logreturn)
```

5. feladat megoldása

A kapott plot-ok:



logreturn hisztogramja



logreturn értékeinek kirajzolása

BÓNUSZ

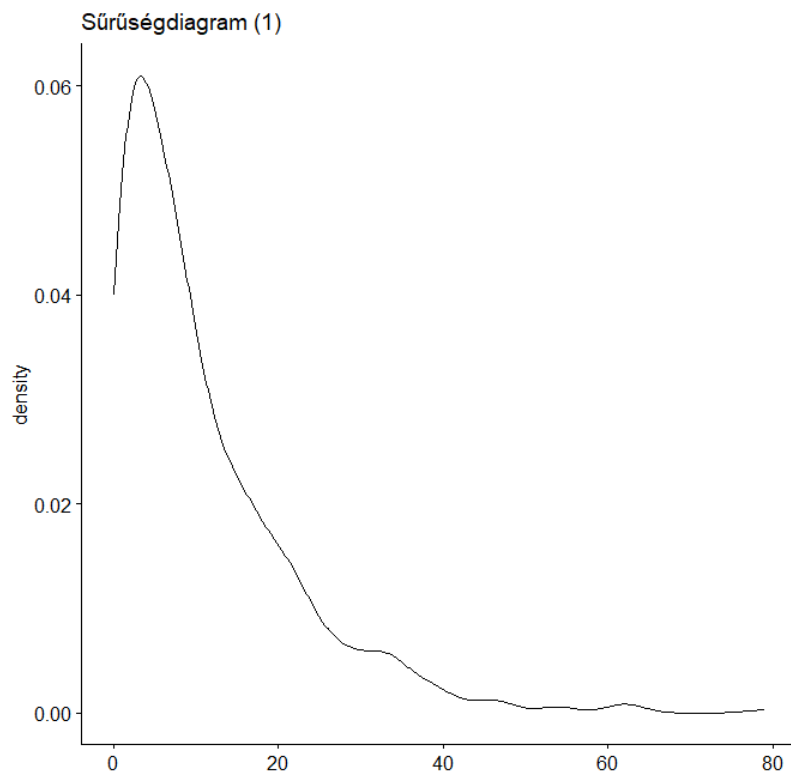
Ezek a kérdések is előfordulhatnak, bár csak egy esetet találtam ahol ez történt. Minden feladathoz oda írom hol fordulhat elő.

2. feladat+ : Eloszlás vizsgálat

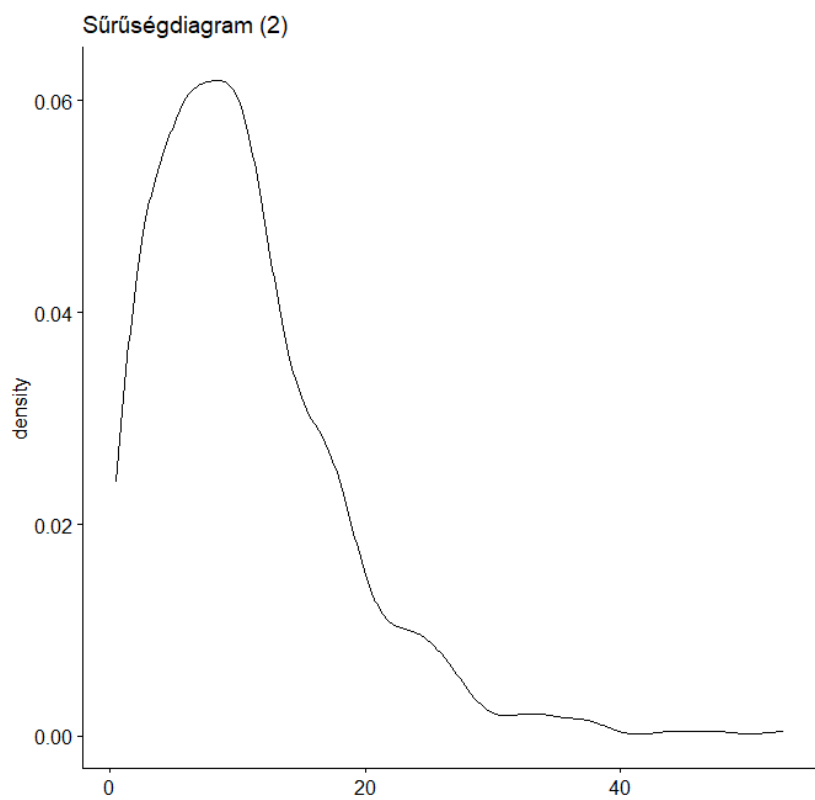
Eloszlás vizsgálatot a **ggdensity()** függvénnyel tudunk végezni, amelyhez be kell szereznünk a **ggpubr** csomagot! Használata egyszerű, csak egy oszlopot kell megadnunk neki a **zn** mátrixból! A sűrűségfüggvényből meg tudjuk állapítani az eloszlást!

```
> library(ggpubr)
Loading required package: ggplot2
Warning messages:
1: package 'ggpubr' was built under R version 4.2.3
2: package 'ggplot2' was built under R version 4.2.3
> ggdensity(zn[,1], main="Sűrűségdiagram (1)")
```

Eloszlás vizsgálata



zn [, 1] eloszlás vizsgálata, ez egy Poisson eloszlás



zn [, 2] eloszlás vizsgálata, ez is egy Poisson eloszlás

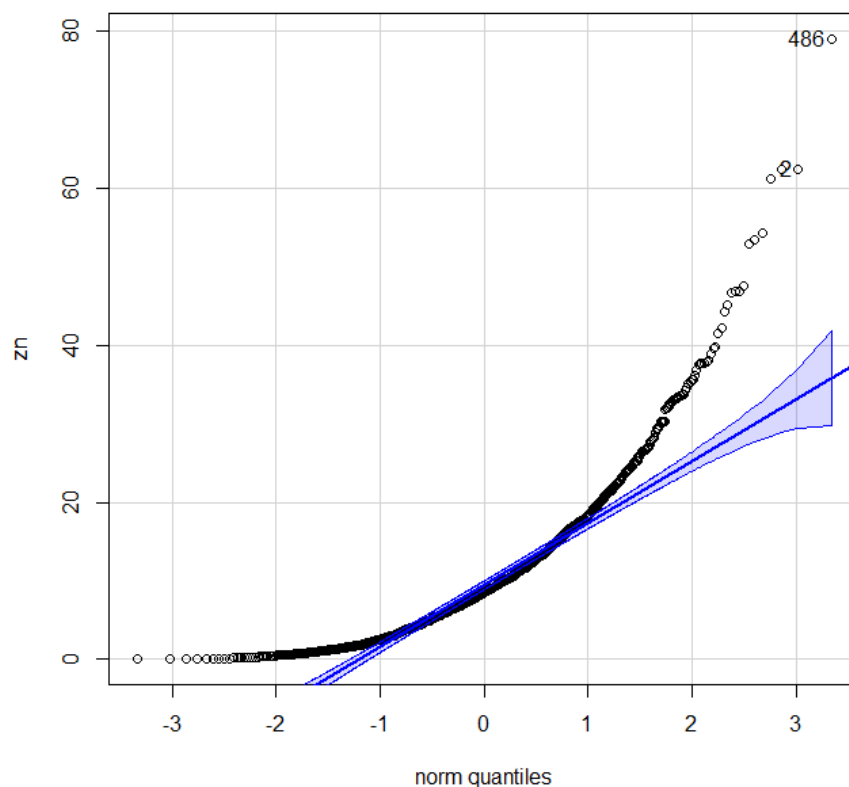
Az eloszlások vizsgálatához kvantilis diagramot is alkalmazhatunk, amelyből ugyancsak megmondható hogy egy sűrűség milyen eloszlásból származik. Segítségül a **qqplot()** függvényt hívjuk, amely a **car** csomagban található meg.

```
> library(car)
Loading required package: carData
Warning messages:
1: package 'car' was built under R version 4.2.3
2: package 'carData' was built under R version 4.2.3
```

```
> qqPlot(zn)
[1] 486 2
```

qqPlot() használata

És a kapott eredmény:



A kapott kvantilis-kvantilis plot

Az eredményből csak azt tudjuk megmondani, hogy normális eloszláshoz lényegében köze nincs a **zn** adathalmaznak, annak természete erősen ferdtett. Pontosabb becslést az eloszlás

természetére a korábban bemutatott módszer adhat nekünk, a **sűrűségdiagram**.

3. feladat+ : Poisson folyamat

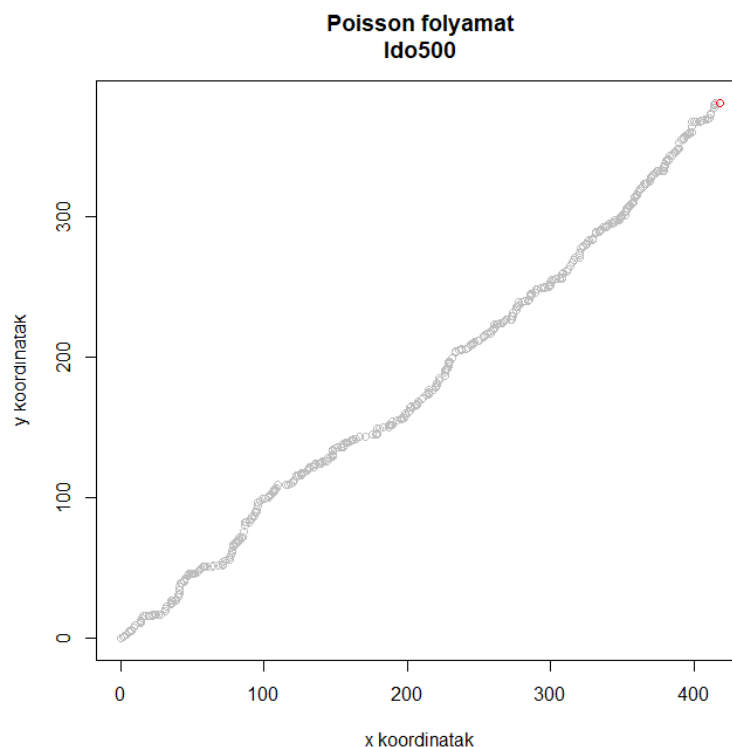
Őszintén szólva ez nem tudom miről szól de a generálás az legitnek tűnik szóval ezt is beleteszem ide, hátha kérdezi valamikor. A kódot amint elindítjátok, szépen ki fog rajzolódni a a folyamat lépésről lépésre.

A felhasznált kód, (should work):

```
> poisson <- function () {  
+   set.seed(ss+17)  
+   x <- y <- x.new <- y.new <- x.new.p <- y.new.p <- vector ()  
+   for (i in 1:500) {  
+     x <- rpois (1, (ax+az) / (ax+ay+az))  
+     y <- rpois (1, (ax+az) / (ax+ay+az))  
+     x.new <- c(x.new, x)  
+     y.new <- c(y.new, y)  
+     x.new.p <- cumsum(x.new)  
+     y.new.p <- cumsum(y.new)  
+     plot (x.new.p, y.new.p, type="b", main=paste("Poisson folyamat\nIdo", i, sep=""),  
+         xlab="x koordinatak", ylab="y koordinatak", col=c (rep ("gray", i-1), "red"), pch=c (rep (20, i-1), 1))  
+   }  
+   poisson_g <- matrix(c(x.new.p, y.new.p), ncol= 2)  
+   return (poisson_g)  
+ }  
> poisson_generalt <- poisson ()
```

Poisson folyamat generálása

A kapott gráf:



Poisson folyamat

A függvény visszatérési értéként visszaadja a kirajzolt értékeket, így tudunk rajtuk különböző statisztikai elemzéseket végrehajtani:

```
> summary(poisson_generalt)
      V1      V2
Min.   : 0.0   Min.   : 0.0
1st Qu.:106.8  1st Qu.:104.0
Median :222.0  Median :184.0
Mean   :217.0  Mean   :188.1
3rd Qu.:327.5  3rd Qu.:283.0
Max.   :418.0  Max.   :381.0
> skewness(poisson_generalt)
[1] -0.07481129  0.02818425
> kurtosis(poisson_generalt)
[1] 1.752028 1.865649
```

Poisson folyamat elemzése