

Generisanje slučajnih procesa

Statistički softver 2

26. april, Beograd

Generisanje lanaca Markova

- Slučajan proces $\{X_t, t \text{ iz } T\}$ je lanac Markova ako važi:

$$P\{X_{t+1}=j \mid X_t=i\} = P\{X_{t+1}=j \mid X_0=i_0, \dots, X_{t-1}=i_{t-1}, X_t=i\}.$$

- Svojstvo izraženo formulom naziva se svojstvo Markova.
- „Budućnost“ procesa zavisi samo od „sadašnjosti“, ne i od „prošlosti“.
- Postoje procesi Markova sa diskretnim vremenom, kao i sa neprekidnim.

Generisanje lanaca Markova

- Kod procesa sa diskretnim vremenom, fazni prostor (skup stanja) je prebrojiv.
- Najčešće se uzima skup celih brojeva Z .
 - X_t - stanje sistema u trenutku t
 - $p_{ij} = P\{ X_{t+1}=j \mid X_t=i \}$
 - verovatnoća prelaska sistema iz stanja i u stanje j
- Kod homogenih lanaca Markova, p_{ij} ne zavisi od trenutka t .

Generisanje lanaca Markova

- $\mathbf{P} = (p_{ij})_{|Z| \times |Z|}$ - stohastička matrica
- Osobine stohastičke matrice:
- $p_{ij} \geq 0$
- $\sum_j p_{ij} = 1$

Stanje i je povratno ako je $P\{X_n = i \mid X_0 = i\} = 1$, a prolazno ako $P\{X_n = i \mid X_0 = i\} < 1$.

Može se pokazati da je nulto stanje povratno u primeru slučajnog lutanja po pravoj.

Generisanje lanaca Markova

- Neki primeri lanaca Markova:

1. **prost proces grananja** - broj potomaka zavisi od broja jedinki prethodne generacije

2. **niz nezavisnih, jednakoraspodeljenih slučajnih veličina** - ovde važi „jače“ svojstvo od svojstva Markova; budućnost ne zavisi ni od „sadašnjosti“, već samo od „budućnosti“

3. **slučajno lutanje po pravoj**

Generisanje lanaca Markova sa dva stanja

Pretpostavimo da imamo dva stanja, 0 i 1, kao i

stohastičku matricu: $\begin{pmatrix} 1 - p_1 & p_1 \\ 1 - p_2 & p_2 \end{pmatrix}$

Postupak generisanja:

1. Neka je $X[1]=0$
2. $X[2] \sim \text{Ber}(p_1)$
3. $X[i], i \geq 3$ razlikujemo dva slučaja:

Ako je $X[i-1]=0$, onda $X[i] \sim \text{Ber}(p_1)$

Ako je $X[i-1]=1$, onda $X[i] \sim \text{Ber}(p_2)$


```
generating<-function(p) {  
  gama<-runif(1)  
  if(gama ==> p)  
    U<-0  
  if(gama < p)  
    U<-1  
  return(U)  
}
```

```
two_states<-function(n, p1, p2){  
  X=vector()  
  X[1]=0  
  for (k in 2:n){  
    if (X[k-1] == 0)  
      X[k]=generating(p1)  
    else if (X[k-1] == 1)  
      X[k]=generating(p2)  
  }  
  return(X)  
}
```


Generisanje lanaca Markova u opštem slučaju

- Pretpostavimo da je fazni prostor tročlan skup $S = \{1, 2, 3\}$, i da je data stohastička matrica reda 3×3 :

$$\begin{pmatrix} 0,5 & 0,2 & 0,3 \\ 0,4 & 0,1 & 0,5 \\ 0,3 & 0,4 & 0,3 \end{pmatrix}$$

1. Uzmemo da je $X[1] = S[1]$
2. Generišemo Y_1 i $X_2 = Y_1$, gde Y_i uzima vrednosti iz S sa verovatnoćama iz i -te vrste stohastičke matrice; $n=2$
3. N -dužina lanca, dok važi $n < N$: $i = X_n$, generišemo Y_i , uvećavamo brojač $n = n + 1$, $X_n = Y_i$.


```
P=t(matrix(c(0.5, 0.2, 0.3, 0.4, 0.1, 0.5, 0.3, 0.4, 0.3),  
nrow=3, ncol=3))  
S=c(1, 2, 3)
```

```
generateY=function(P, S){  
  n=length(P)  
  gamma=runif(1)  
  suma=P[1]  
  
  if (gamma<=P[1])  
    return(S[1])  
  
  for (i in 2:(n-1)) {  
    if (gamma>suma & gamma<=(suma+P[i]))  
      return(S[i])  
    suma=suma+P[i]  
  }  
  if (gamma>suma)  
    return(S[n])  
}
```



```
Markov=function(N){
```

```
  X=vector()
```

```
  X[1]=1
```

```
  X[2]=generateY(P[1, ], S)
```

```
  n=2
```

```
  while(n<N){
```

```
    i=X[n]
```

```
    n=n+1
```

```
    X[n]=generateY(P[i, ], S)
```

```
  }
```

```
  return(X)
```

```
}
```


Slučajno lutanje po pravoj

```
afterNSteps=function(n){
```

```
  u=sample(c(-1, 1), n, replace=T)
```

```
  S=0
```

```
  for (i in 2:n) {
```

```
    s=rep(u, length.out=i-1)
```

```
    S[i]=sum(s)
```

```
  }
```

```
  plot(S, xlab= „Broj koraka slučajnog lutanja“,  
type="l")
```

```
  return(S)
```

```
}
```


Paket u R-u za lance Markova

Metodi:

new- generisanje novih objekata klase 'markovchain'. Argumenti su: naziv paketa, states, byrow, transitionMatrix..

as- konvertovanje lanca u data.frame objekat ili tabelu

plot, show, print..- prikaz stohastičke matrice, podataka o lancu i dijagrama lanca

Primer:

```
weatherStates <- c("sunny", "cloudy", "rain")  
byRow <- TRUE
```

```
weatherMatrix <- matrix(data = c(0.70, 0.2, 0.1,  
0.3, 0.4, 0.3, 0.2, 0.45, 0.35), byrow = byRow,  
nrow = 3, dimnames = list(weatherStates,  
weatherStates))
```

```
mcWeather <- new("markovchain", states =  
weatherStates, byrow = byRow, transitionMatrix =  
weatherMatrix, name = "Weather")
```



```
weathersOfDays <- rmarkovchain(n = 365, object = mcWeather, t0 =  
"sunny")
```

```
WeathersOfDays[1:30]
```

Dijagram lanca:

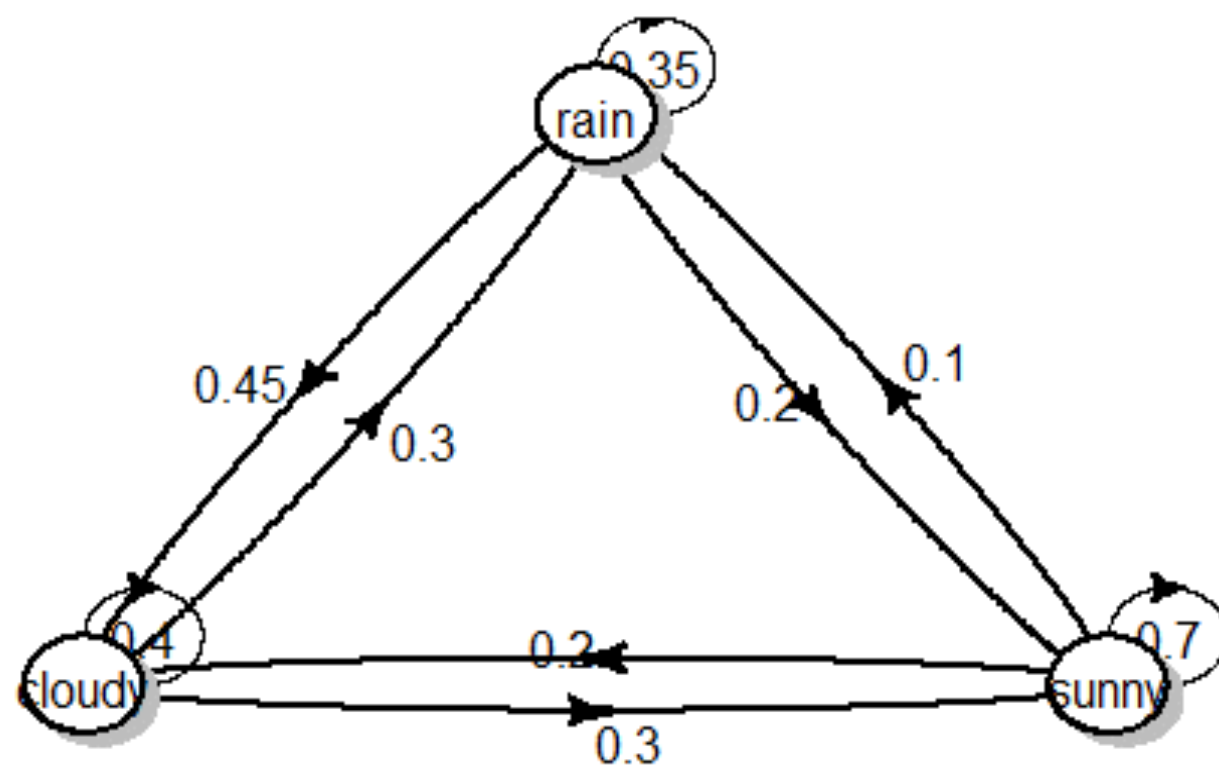
```
install.packages("diagram")
```

```
library("diagram")
```

```
plot(mcWeather, package="diagram", box.size = 0.04)
```

Ocena metodom maksimalne verodostojnosti:

```
weatherFittedMLE <- markovchainFit(data = weathersOfDays,  
method = "mle", name = "Weather MLE")
```

Generisanje Puasonovog procesa

Puasonov proces je slučajni proces definisan na skupu $T=[0, +\infty)$ ako važi:

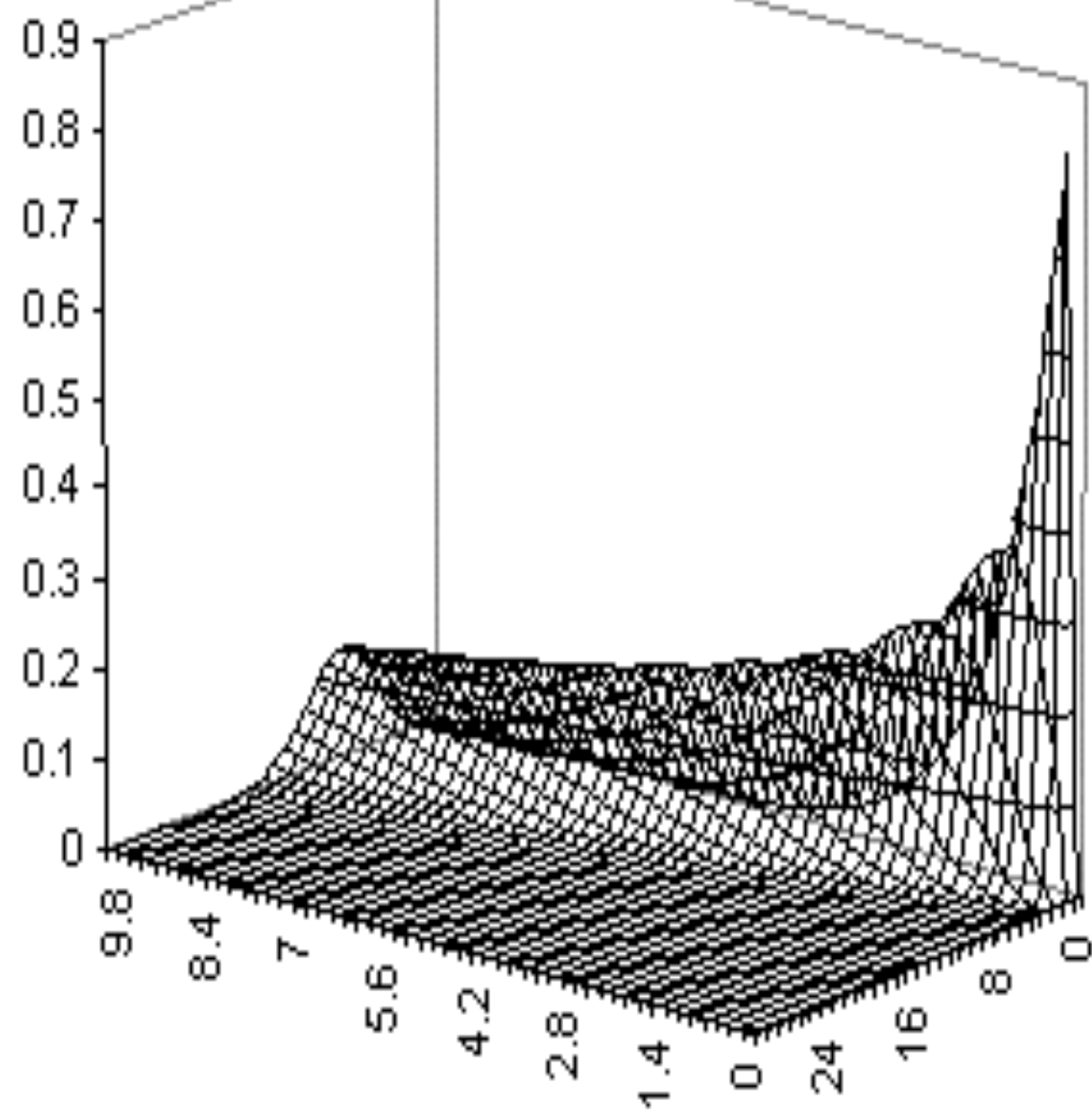
1. $P(X_0=0)=1,$

2. priraštaji $X_t - X_s$ i $X_v - X_u$ su nezavisni za svako $s < t \leq u < v,$

3. za svako s, t ($s < t$), priraštaj $X_t - X_s$ ima Poasonovu raspodelu sa parametrom $\lambda(t-s)$ gde je λ pozitivna realna konstanta ($\lambda > 0$).

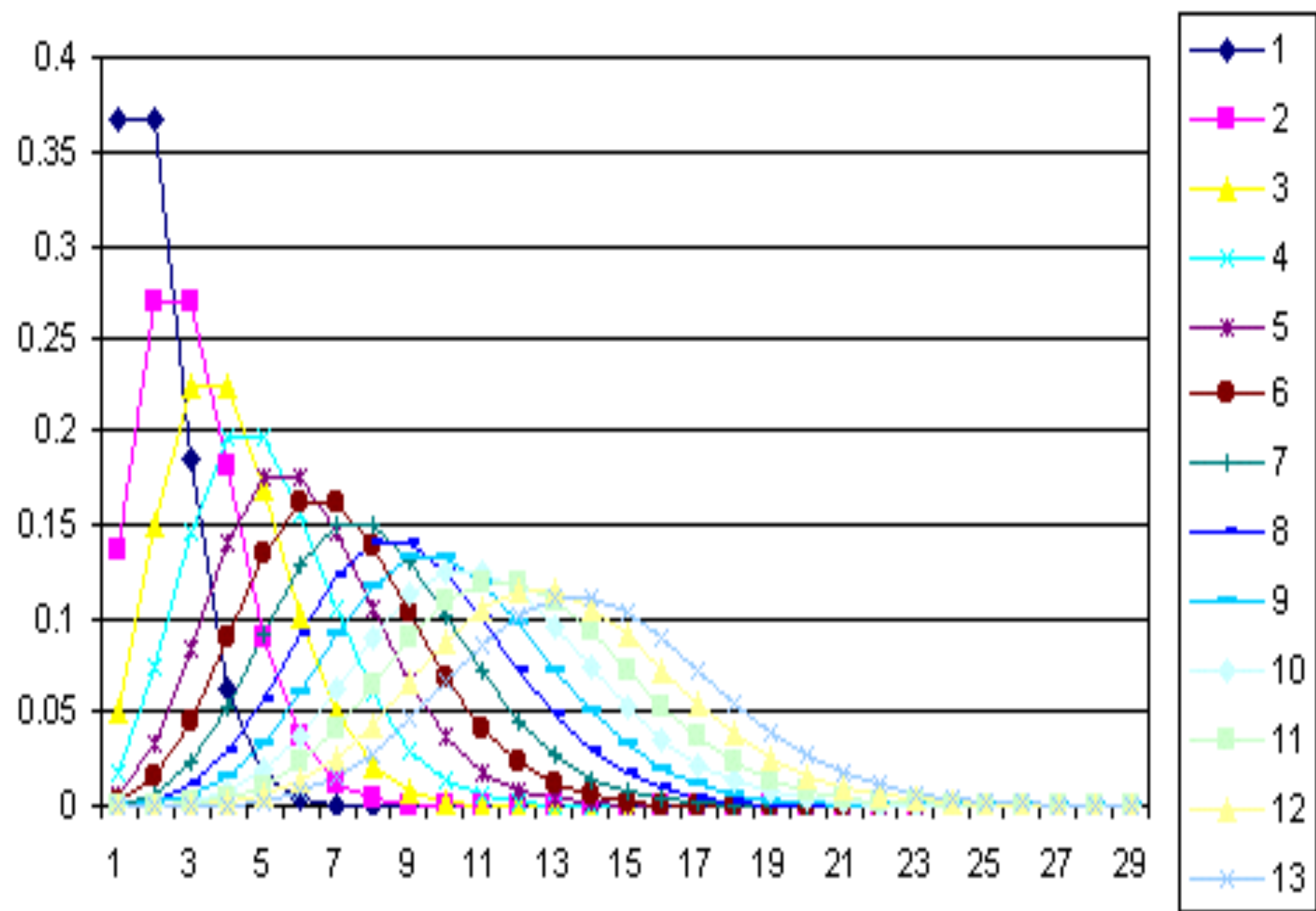
Generisanje Puasonovog procesa

- Puasonovi procesi se primenjuju za modelovanje broja tzv. “retkih događaja” :
 - broj telefonskih razgovora do trenutka t sa neke telefonske govornice
 - broj autobusa gradskog saobraćaja koji prođu pored nekog mesta
 - broj zahteva koje korisnik uputi računarskom sistemu



Generisanje Puasonovog procesa

- Na gornjoj slici prikazana je funkcija $P_x(i,t)$.
- Puasonova raspodela dostiže svoj maksimum za $i = \lambda t$
- Funkcija $P_x(i,t)$ rastuća pa potom opadajuća po oba svoja argumenta zasebno.
- Na donjoj slici prikazane su funkcije $P_x(i,t)$, u zavisnosti od i za razne vrednosti λt .



Generisanje Puasonovog procesa

```
getN <- function(x, tt=1) {  
  if (sum(x) < tt)  
    return(NA)  
  if (x[1] > tt)  
    return(0)  
  else  
    return(max(which(cumsum(x) < tt)))  
}
```


Generisanje Puasonovog procesa

```
simPP <- function(ntrials,tt=1,lambda=2) {  
  maxN <- round(qpois(0.99999,tt*lambda))+10  
  #dovoljno veliko da su šanse da nam treba  veći uzorak  
  jako male  
  x <- matrix(rexp(ntrials*maxN,lambda),ntrials,maxN)  
  return(apply(x, 1, getN, tt=tt))  
}
```


Paket u R-u za Puasonov proces

-Naziv paketa: NHPoisson

-Funkcije:

1. fitPP.fun
2. simNHP.fun
3. glovalval.fun
4. data (BarTxTn)
5. addAIC.fun
6. emplambda.fun & emplabdaD.fun
7. POTevents.fun
8. graphResX.fun & graphResCov.fun
9. resQQplot.fun

Paket u R-u za Puasonov proces

- `fitPP.fun(covariates = NULL, start, fixed=list(), posE = NULL,...)`
- `simNHP.fun(lambda, fixed.seed=NULL)`
- `globalval.fun(mlePP, lint = NULL, nint = NULL, Xvar = NULL, namXvar = NULL, Xvart = NULL, namXvart = NULL,...)`

Primeri korišćenja ugrađenih funkcija:

```
data(BarTxTn)
```

```
covB<-cbind(cos(2*pi*BarTxTn$dia/365), sin(2*pi*BarTxTn$dia/  
365), BarTxTn$TTx, BarTxTn$Txm31, BarTxTn$Txm31**2)
```

```
modB<-fitPP.fun(tind=TRUE,covariates=covB, POTob=list(T=BarTxTn  
$Tx, thres=318), tit="BAR Tx; cos, sin, TTx, Txm31, Txm31**2",  
start=list(b0=-100,b1=1,b2=10,b3=0,b4=0,b5=0),Clty="Transf",m  
odCI=TRUE, modSim=TRUE,dplot=FALSE)
```

```
aux<-globalval.fun(mlePP=modB,lint=153, type1="Disjoint",  
resqqplot=FALSE)
```

```
t<-runif(500, 0.01,0.1)
```

```
aux<-simNHP.fun(lambda=t)
```


Vinerov proces (Braunovo kretanje)

-Slučajni proces $\{W(t), t > 0\}$ definisan na nekom prostoru verovatnoća je Vinerov proces ako važe uslovi:

1. $W(0)=0$ s.s

2. W ima nezavisne priraštaje

3. za $0 < s < t$: $W(t)-W(s) \sim N(0, t-s)$

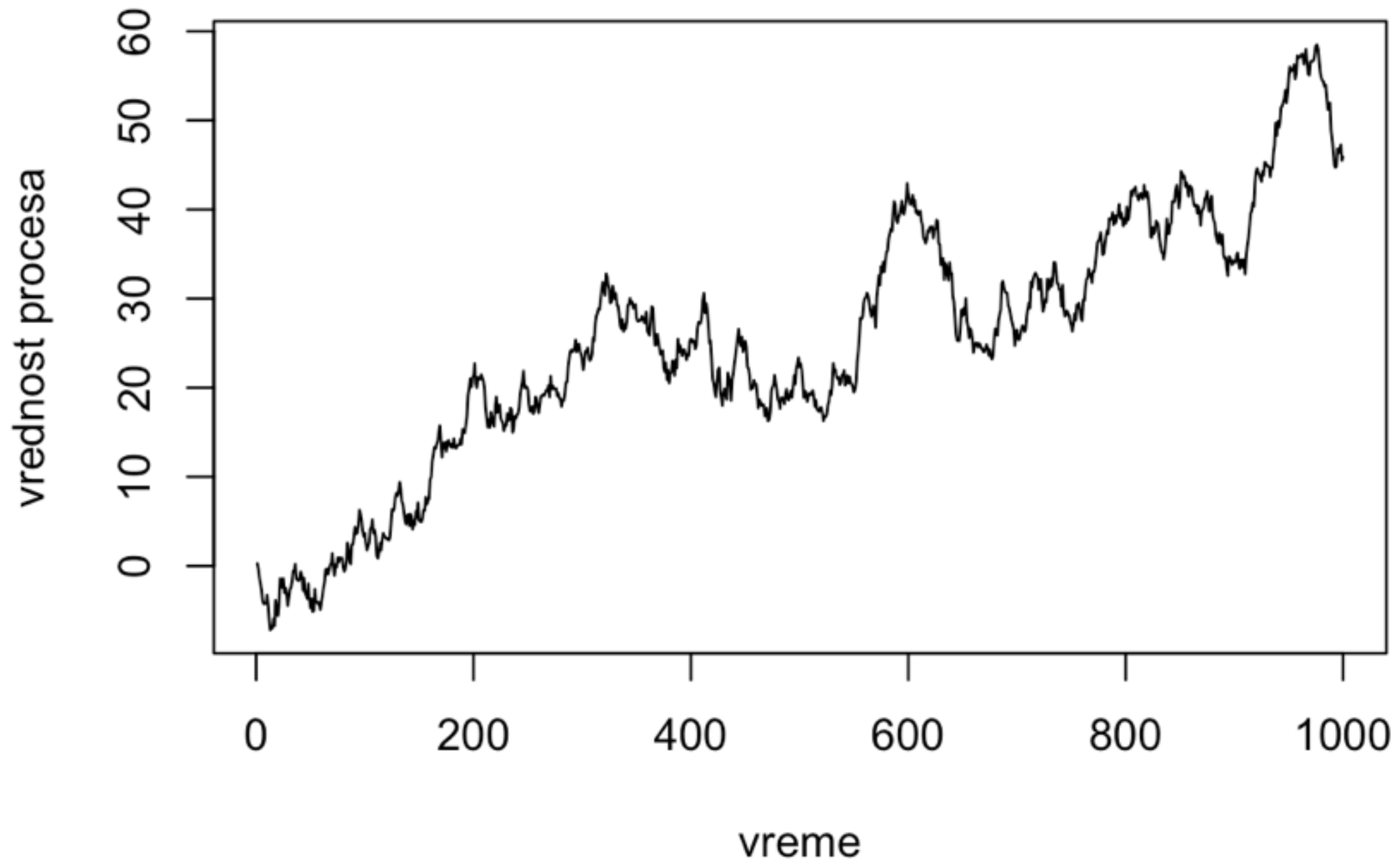
-Vinerov proces je proces Markova.

Generisanje Vinerovog procesa

Primer generisanja jednodimenzionog Braunovog kretanja:

```
BMinOneDim<-function (N){  
  dis<-cumsum(rnorm(N))  
  plot(dis, type="l", main="Jednodimenziono  
  Braunovo kretanje", xlab="vreme",  
  ylab="vrednost procesa")  
}
```


Jednodimenziono Braunovo kretanje, $N=1000$



Generisanje Vinerovog procesa

Druga mogućnost crtanja grafika ove funkcije je pomoću zadavanja veličine koraka:

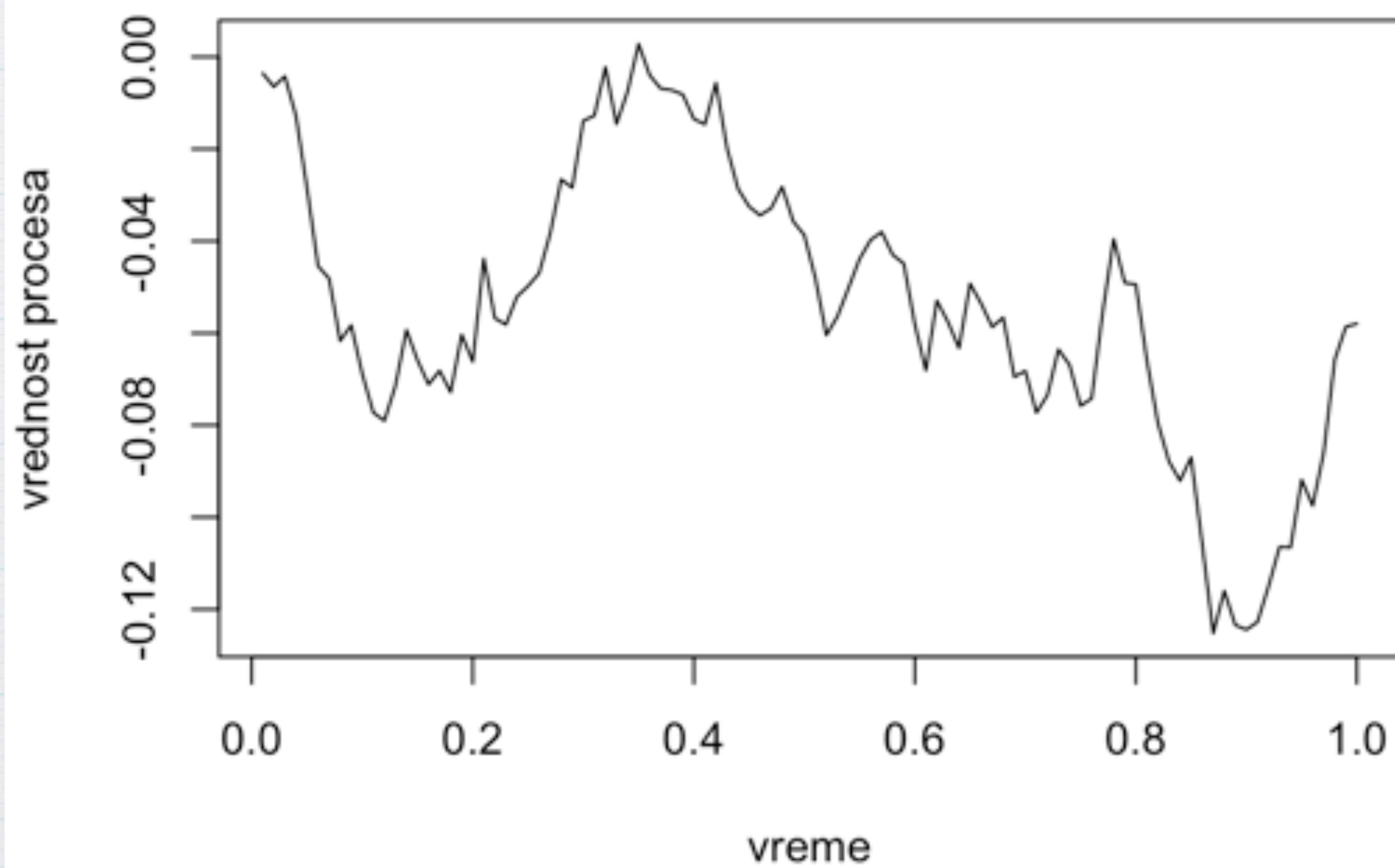
```
x<-c(1:100)
```

```
x<-x/100
```

```
y<-rnorm(100, 0, 0.01)
```

```
y<-cumsum(y)
```

```
plot (x, y, type="l", xlab="vreme",  
ylab="vrednost procesa")
```

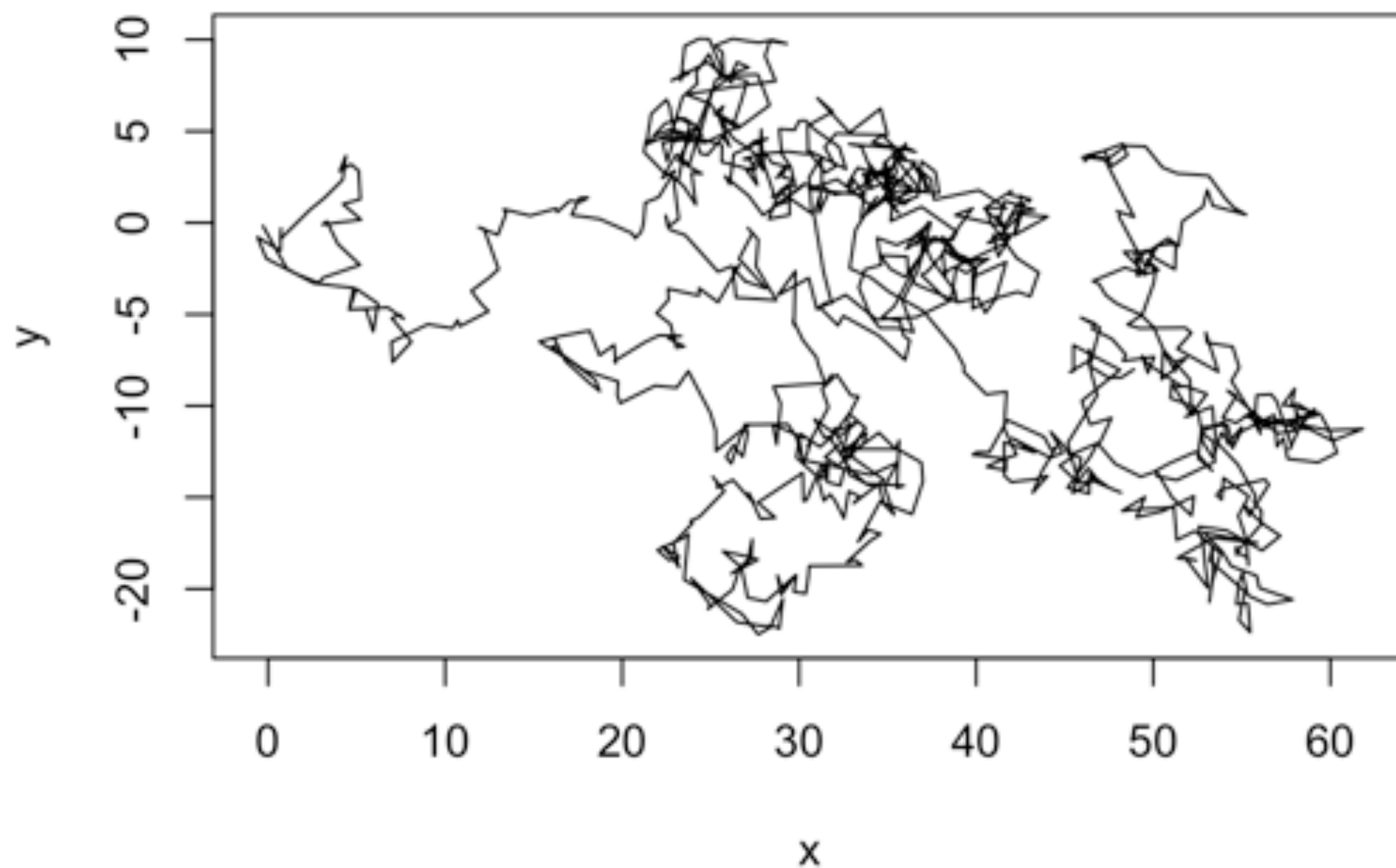



Generisanje Vinerovog procesa

Primer generisanja dvodimenzionog Braunovog kretanja:

```
BMinTwoDim<- function (N){  
  x<-cumsum(random(N))  
  y<-cumsum(random(N))  
  plot (x, y, type="l", main="Dvodimenziono  
  Braunovo kretanje, N=1000")  
}
```


Dvodimenziono Braunovo kretanje, $N=1000$



Paket u R-u za Vinerov proces

- Naziv paketa: RWiener
- Način instaliranja: `install.packages(RWiener)`
- Korisne funkcije:
 1. `dwiener(q, alpha, tau, beta, delta, resp="upper", give_log=TRUE)`
 2. `pwiener (q, alpha, tau, beta, delta, resp="upper")`
 3. `qwiener (q, alpha, tau, beta, delta, resp="upper")`
 4. `rwiener (n, alpha, tua, beta , delta)`

Paket u R-u za Vinerov proces

-Argumenti:

q - vektor kvantila

p - vektor verovatnoća

n - broj posmatranja

alpha - parametar granice

beta - parametar nepristrasnosti

delta - parametar drifta (brzine pomeranja)

resp - “upper”, “lower”, “both”

give_log - funkcija vraća log, ako je ovaj

argument TRUE

Primeri korišćenja ugrađenih funkcija:

```
library(RWiener)
```

```
#generisanje slučajnih vrednosti procesa  
dat <- rwiener(100,2,.3,.5,0)
```

```
# crtanje grafika vrednosti  
wiener_plot(dat)
```

```
dwiener(1.45, 2,0.3,0.5,0)
```

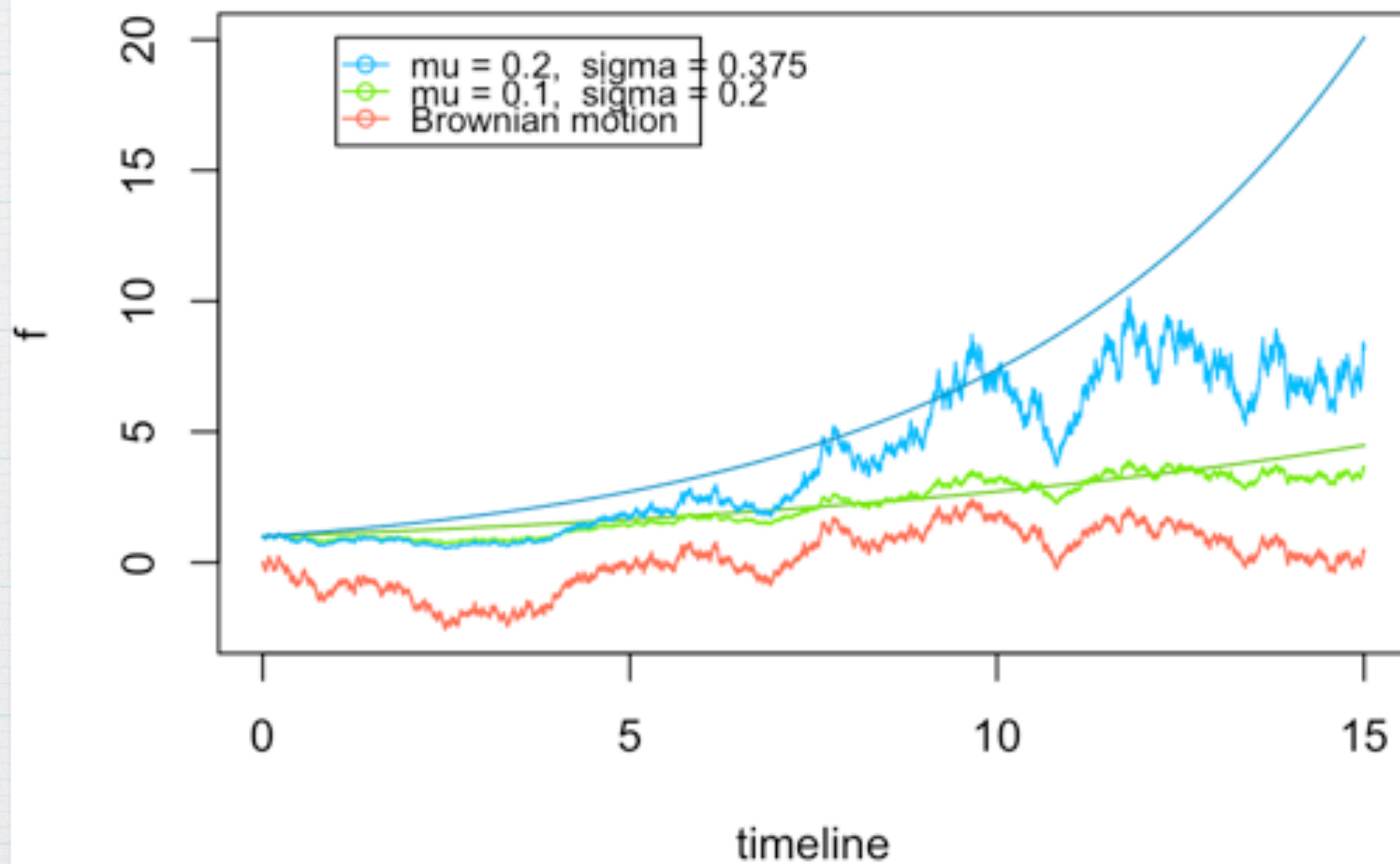
```
pwienner(1.45, 2,0.3,0.5,0)
```

```
qwiener(0.5, 2,0.3,0.5,0)
```


Geometrijsko Braunovo kretanje

- Modifikacija standardnog Braunovog kretanja
- Definiše se kao $\{\exp(W(t)), t > 0\}$, gde je $W(t)$ standardno Braunovo kretanje.
- Primena: finansijsko modelovanje rasta cena akcija

Geometric Brownian Motion trajectories



Hvala na pažnji! :)

Ozrenka Dragić
Isidora Đurić
Katarina Jagličić
Anica Stamenić