MD2_Racinskis

Pēteris Račinskis pr20015

5/18/2021

2. mājas darbs Mate 6029

1. uzdevums - lineāras regresijas, modeļu novērtējumi

Datu kopas ielāde:

```
df <- read.table('CMB.dat',header=TRUE)
attach(df)
```

Datu kopa - kosmiskā mikroviļņu fona novērojumi. Svarīgie parametri šajā gadījumā ir 'ell' - multipolu moments (rupji runājot, lenķiskais ekvivalents starojuma frekvencei) un starojuma spektra nobīde 'Cl' (rupji runājot, spektra temperatūras nobīde no vidējā). Pārējās trīs kolonnas ir statistiski novērojumu trokšņa u.c. raksturotāji.

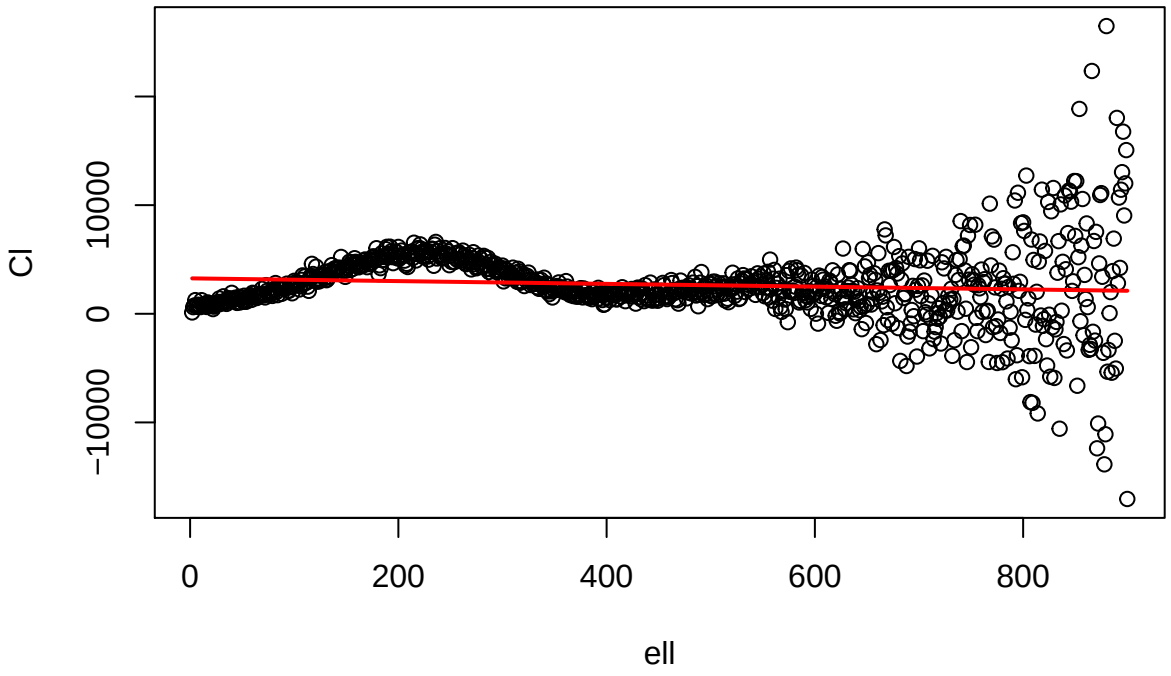
1.1. regresijas modeļu ģenerēšana

Vispārināma lineārās regresijas funkcija grafiku zīmēšanai, rezultātu izvadei uz konsoles un darbam ar augstākas pakāpes modeļiem:

```
general_lreg <- function(vec1,vec2,degree=1,plot=F,print=F,names=c("","")) {
  fit<-lm(vec2~poly(vec1,degree,raw=T))
  if(plot){
    plot(vec1,vec2,xlab=names[1],ylab=names[2])
    x <- seq(min(vec1),max(vec1),length.out = length(vec1))
    f <- predict(fit, newdata = data.frame(vec1 = x))
    lines(x,f,col="red",lwd=2)
  }
  if(print){
    print(paste("R-squared:",summary(fit)$r.squared))
  }
  fit
}
```

Vienkārša lineārā regresija parametriem ‘ell’ un ‘Cl’:

```
fit1 <- general_lreg(ell,Cl,plot=T,print=T,names=c("ell","Cl"))
```




```
## [1] "R-squared: 0.00991069470200286"
```

Funkcija labākās atbilstības polinoma meklēšanai (apstājas, kad ANOVA tests liecina, ka jaunas brīvības pakāpes pievienošana būtisku uzlabojumu modeļa atbilstībā datiem vairs nesniedz). Virzība - pa divām pakāpēm, lai ļautu modelim piekārtoties simetriskiem/asimetriskiem sadalījumiem pēc vajadzības:

```
bestfit <- function(vec1,vec2,deg=1,last_deg=1,P=0,p=0.05,max=27) {
  if((P>0.05) || (deg>max)){
    list(d=last_deg,p=P)
  } else {
    f1<-general_lreg(vec1,vec2,deg)
    next_deg<-deg+2
    f2<-general_lreg(vec1,vec2,next_deg)
    P<-anova(f1,f2)$'Pr(>F)'[2]
    bestfit(vec1,vec2,next_deg,deg,P)
  }
}
```

Labākais polinoms:

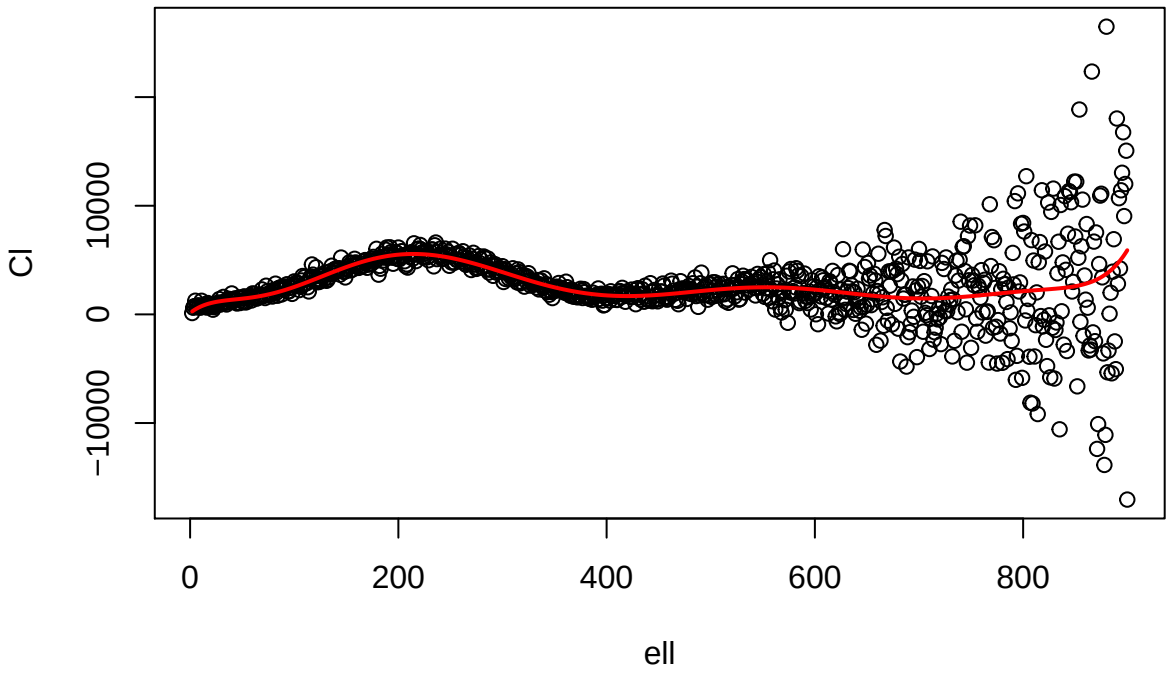
```
res <- bestfit(ell,Cl)
paste("stopped at x^", res$d,sep="")
```

```
## [1] "stopped at x^9"
```

```
paste("p value:",res$p)
```

```
## [1] "p value: 0.9633917284375"
```

```
fitmax<-general_lreg(ell,Cl,degree=res$d,plot=T,print=T,names=c("ell","Cl"))
```

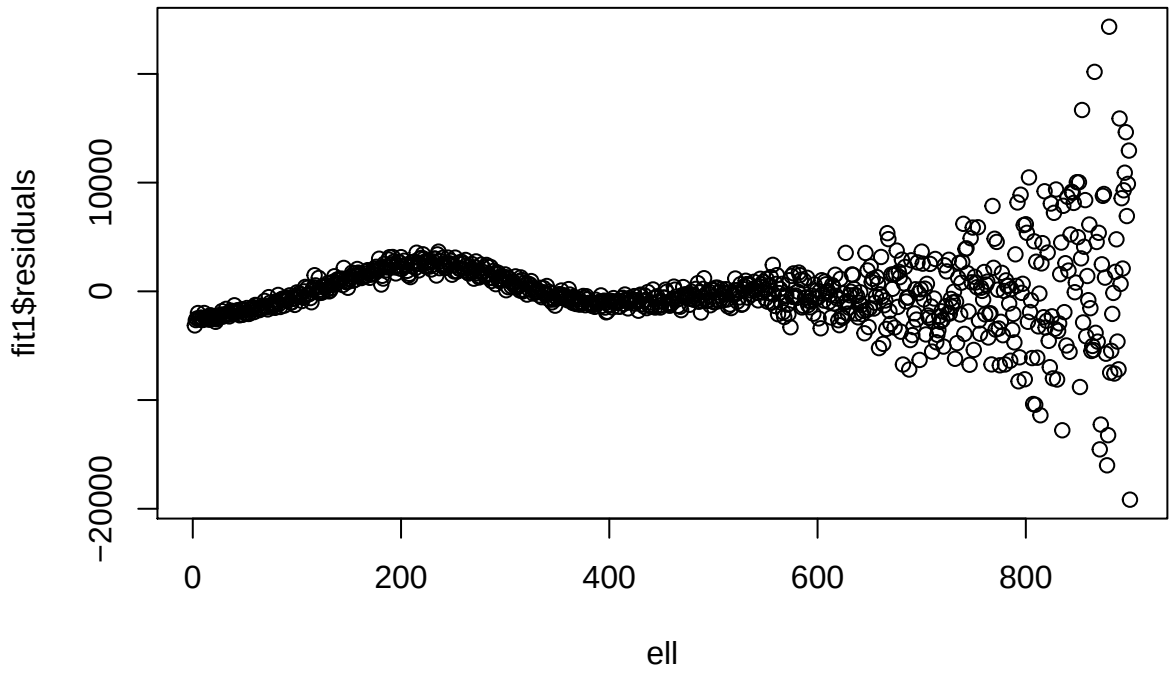



```
## [1] "R-squared: 0.14991001889"
```

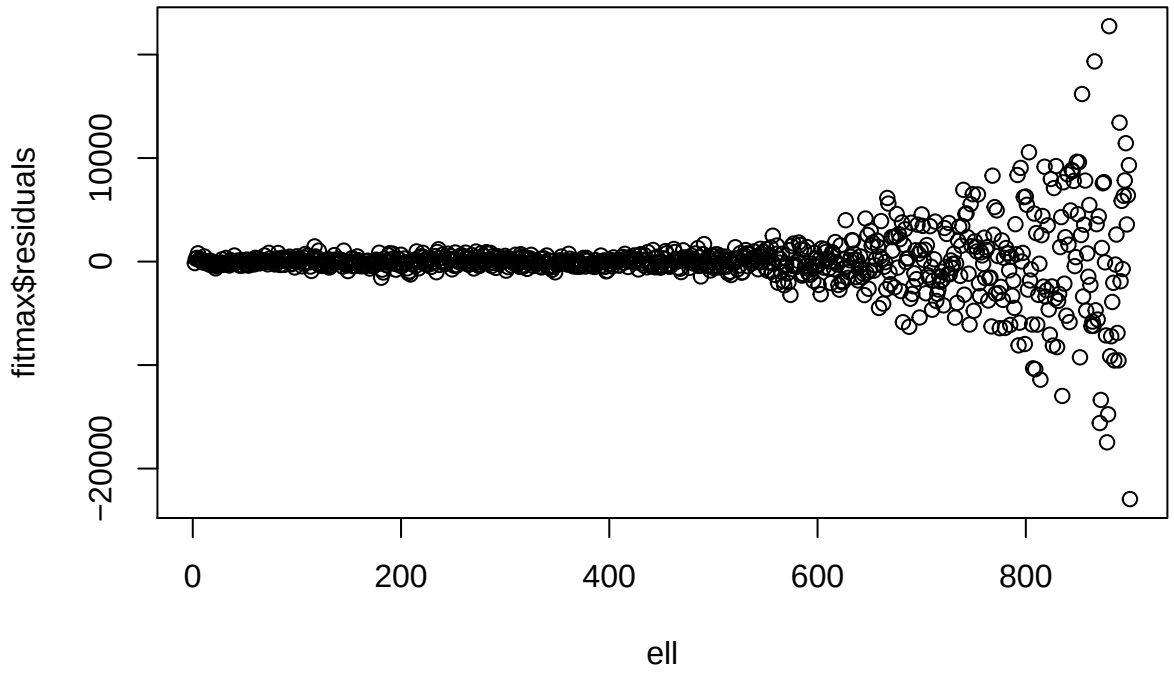
1.2. diagnostika

Atlikumu neatkarība - grafiski:

```
plot(ell,fit1$residuals)
```



```
plot(ell,fitmax$residuals)
```



Statistisko testu bibliotēkas:

```
library(car)
library(nortest)
```

Durbin-Watson tests autokorelācijai:

```
durbinWatsonTest(fit1)
```

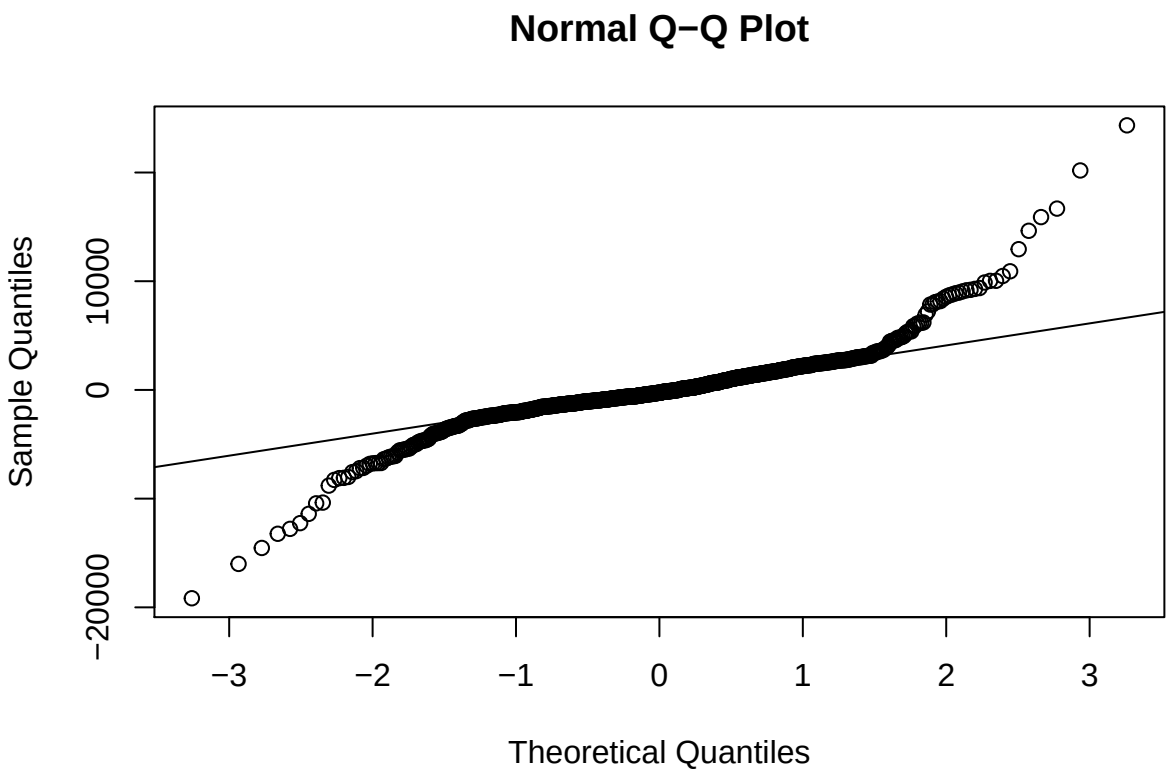
```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.05972642      1.841981   0.012
##  Alternative hypothesis: rho != 0
```

```
durbinWatsonTest(fitmax)
```

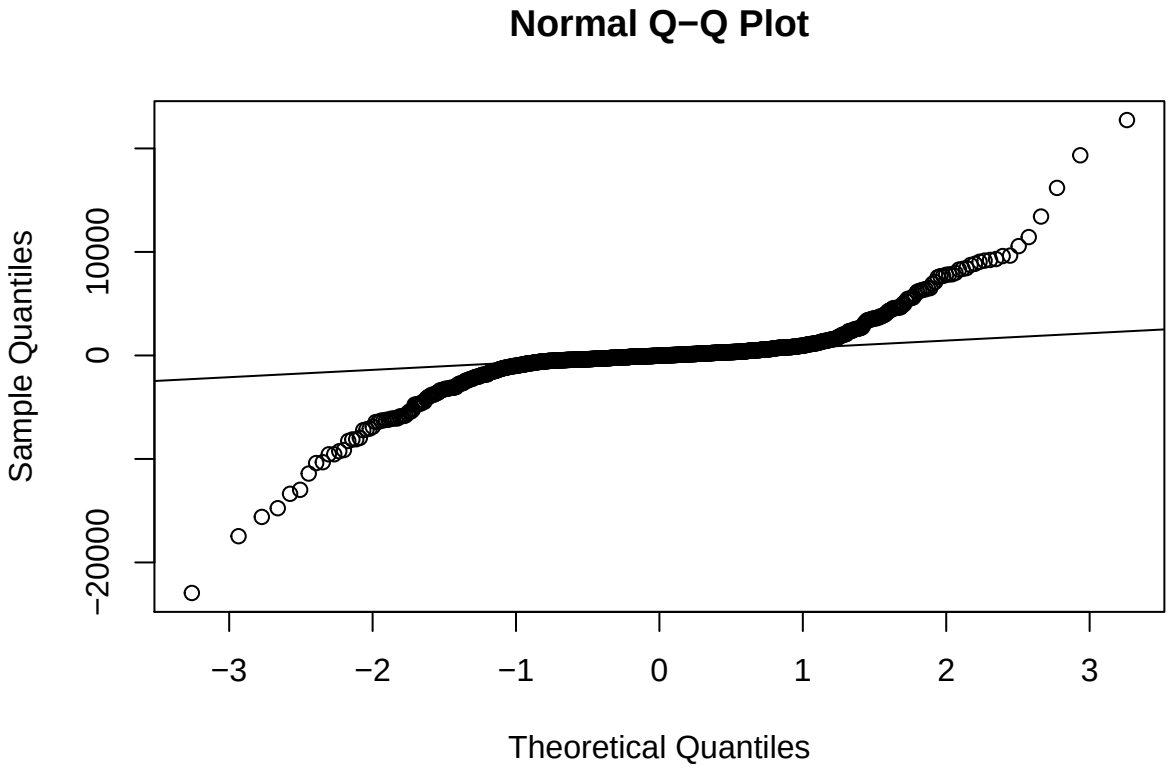
```
##  lag Autocorrelation D-W Statistic p-value
##    1      -0.1044366      2.146133   0.074
##  Alternative hypothesis: rho != 0
```

Normalitātes testi - grafiski:

```
qqnorm(fit1$residuals)
qqline(fit1$residuals)
```




```
qqnorm(fitmax$residuals)
qqline(fitmax$residuals)
```



Normalitātes testi - Kolmogorova-Smirnova tests:

```
# degree-1 approximation normality
(lillie.test(fit1$residuals)$p.value > 0.05)
```

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

```
# max degree approximation normality
(lillie.test(fitmax$residuals)$p.value > 0.05)
```

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

Dispersijas vienmērīguma testi:

```
# degree-1 approximation homoscedacity
(ncvTest(fit1)$p > 0.05)
```

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

```
# max degree approximation homoscedacity
(ncvTest(fitmax)$p > 0.05)
```

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

1.3. secinājumi

Dati acīmredzami nav lineāri sakarīgi, un to apliecina arī visas formālās metrikas. Cits jautājums ir par reģionu [0:500], kur tie diezgan cieši seko līknei, ko labi varētu aprakstīt samērā nelielas pakāpes polinoms (sk. sekciju "atlikumu neatkarība - grafiski", kur šajā reģionā atlikumi 9. pakāpes regresijas līknei ir vienmērīgi sadalīti ap 0). Taču ap 'ell' = 500 ļoti strauji pieaug novērojumu dispersija, kas pilnībā izgāž jebkādus mēģinājumus aproksimēt visu datu kopu ar vienu līkni. Šī radikālā izmaiņa dispersijā nomāc arī jebkādus dziļākus ieskatus, ko pār visu datu kopu veiktie testi varētu sniegt par sadalījuma dabu.

2. uzdevums

Datu kopas ielāde:

```
df <- LifeCycleSavings
attach(df)
```

Datu kopas kolonnas:

1. sr - uzkrājumi
2. pop15 - % iedzīvotāju zem 15
3. pop75 - % iedzīvotāju virs 75
4. dpi - ienākumi
5. ddpi - IKP pieaugums

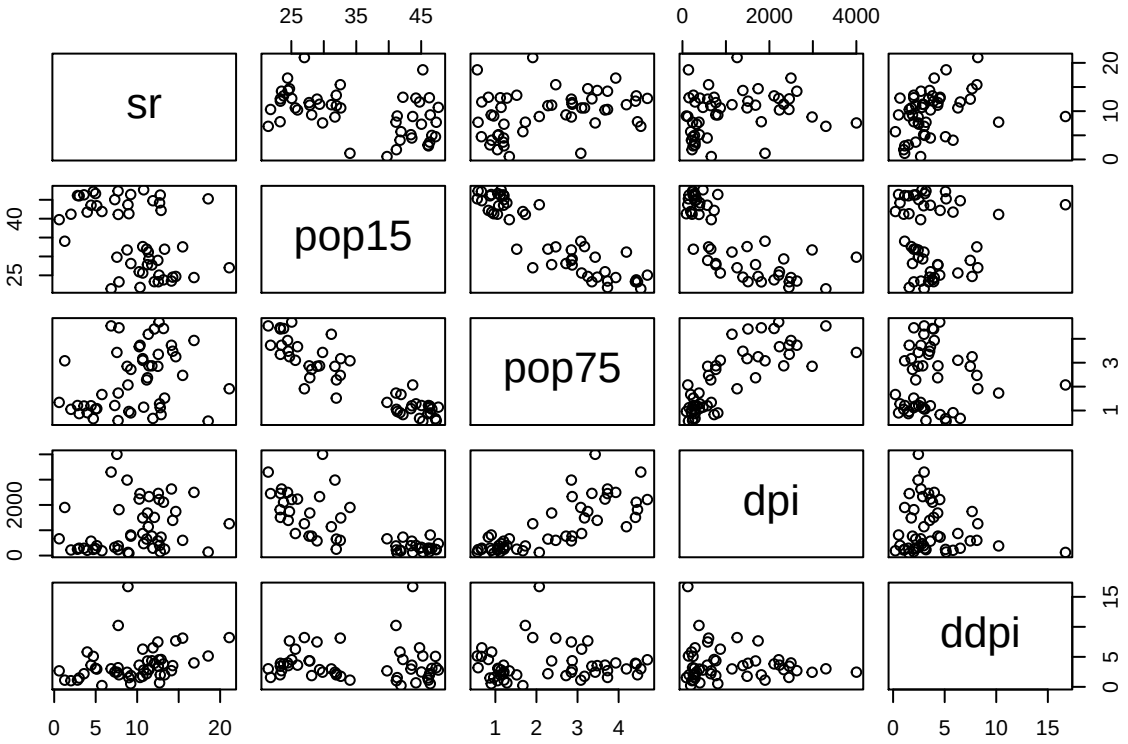
2.1. vispārīgu sakarību meklēšana

Izmantojot iebūvētās funkcijas cor() un pairs(), var ātri gūt vispārīgu priekštatu par datu kopā pastāvošajām sakarībām:

```
cor(df)
```

```
##                sr       pop15       pop75        dpi        ddpi
## sr     1.0000000 -0.45553809  0.31652112  0.2203589  0.30478716
## pop15 -0.4555381  1.00000000 -0.90847871 -0.7561881 -0.04782569
## pop75  0.3165211 -0.90847871  1.00000000  0.7869995  0.02532138
## dpi    0.2203589 -0.75618810  0.78699951  1.0000000 -0.12948552
## ddpi   0.3047872 -0.04782569  0.02532138 -0.1294855  1.00000000
```

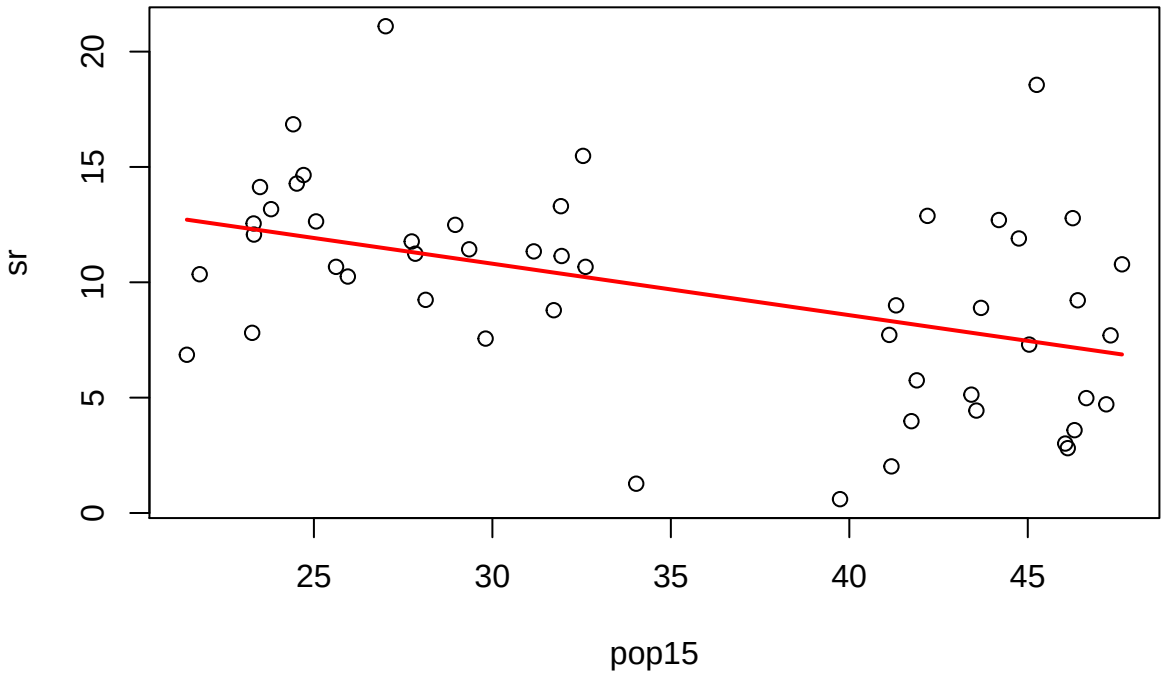
```
pairs(df)
```



Kā redzams, izteiktas sakarības nav starp nevienu parametru un uzkrājumiem, taču redzama neliela negatīva korelācija starp pop15 un sr, un nelielas pozitīvas korelācijas visos citos gadījumos.

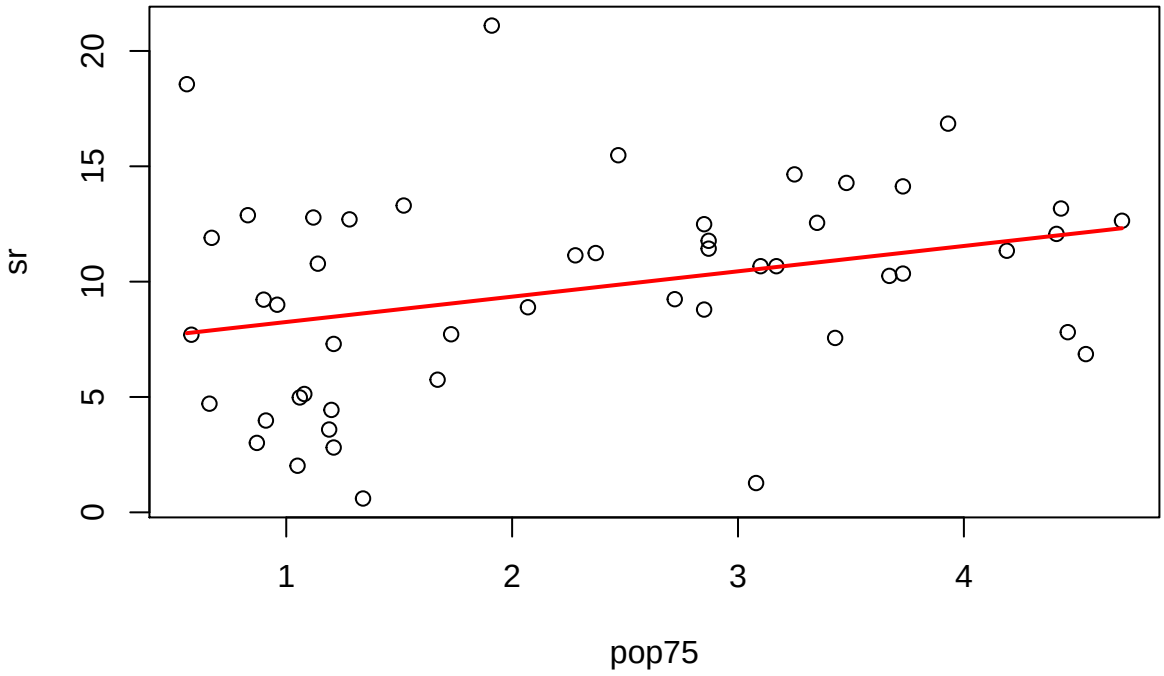
2.2. regresijas modeļu konstruēšana

```
f_pop15<-general_lreg(pop15,sr,plot=T,print=T,names=c("pop15","sr"))
```



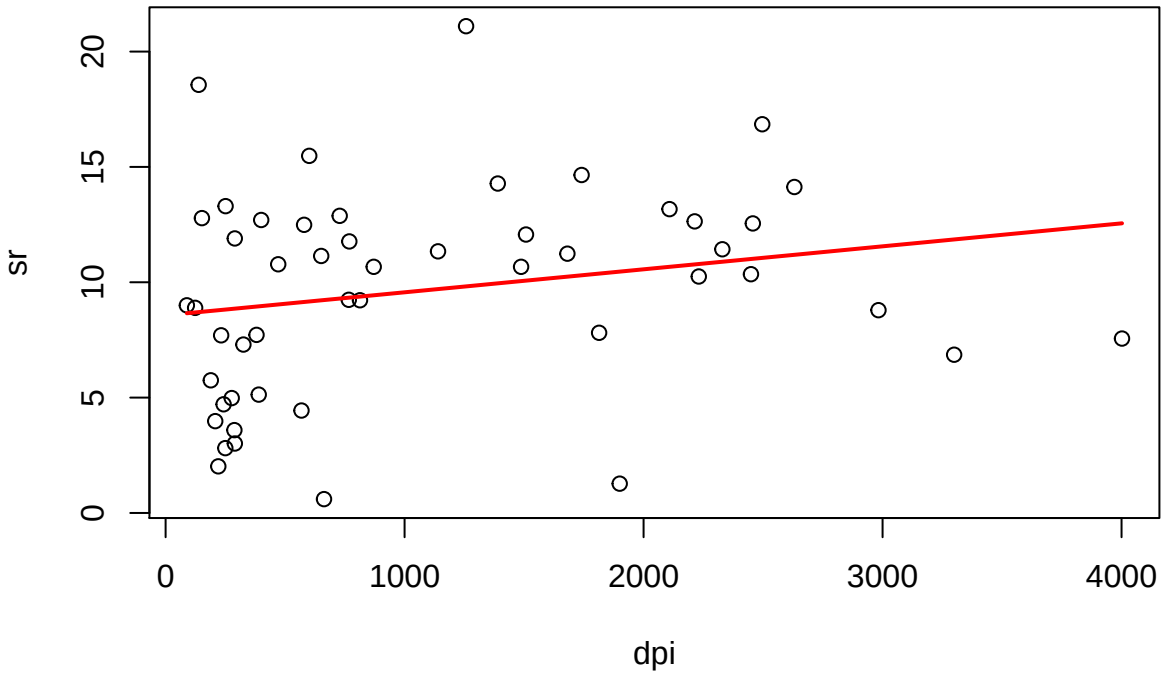

```
## [1] "R-squared: 0.20751494822826"
```

```
f_pop75<-general_lreg(pop75,sr,plot=T,print=T,names=c("pop75","sr"))
```



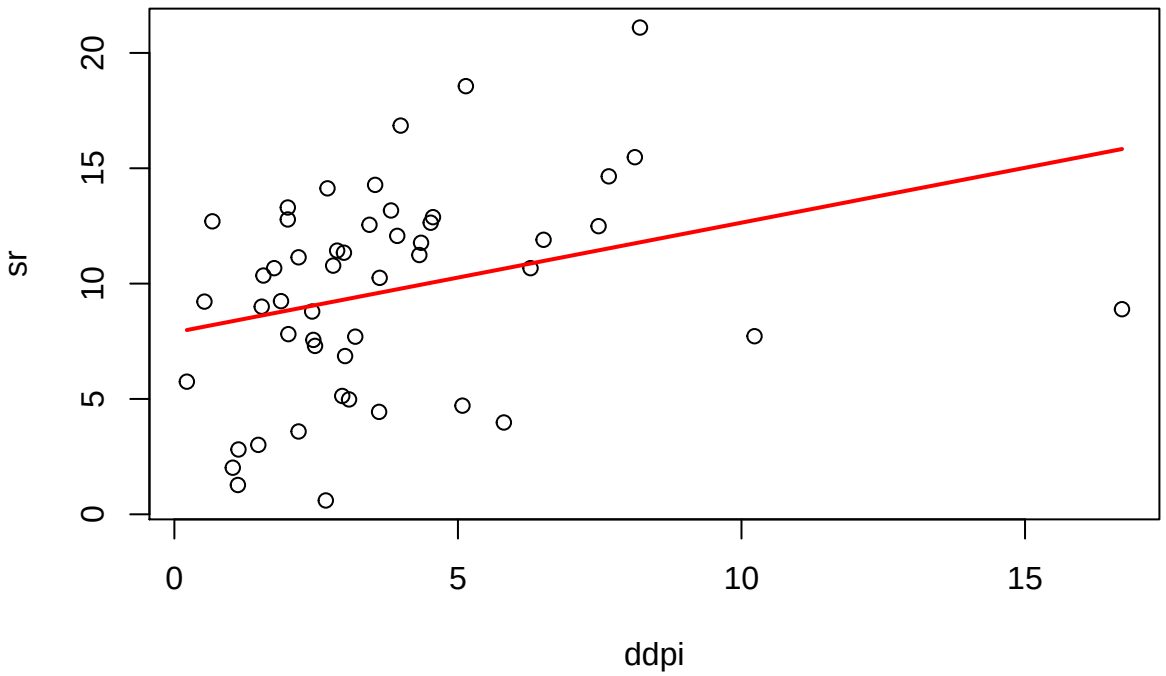

```
## [1] "R-squared: 0.100185621919712"
```

```
f_dpi<-general_lreg(dpi,sr,plot=T,print=T,names=c("dpi","sr"))
```




```
## [1] "R-squared: 0.0485580524006459"
```

```
f_ddpi<-general_lreg(ddpi,sr,plot=T,print=T,names=c("ddpi","sr"))
```



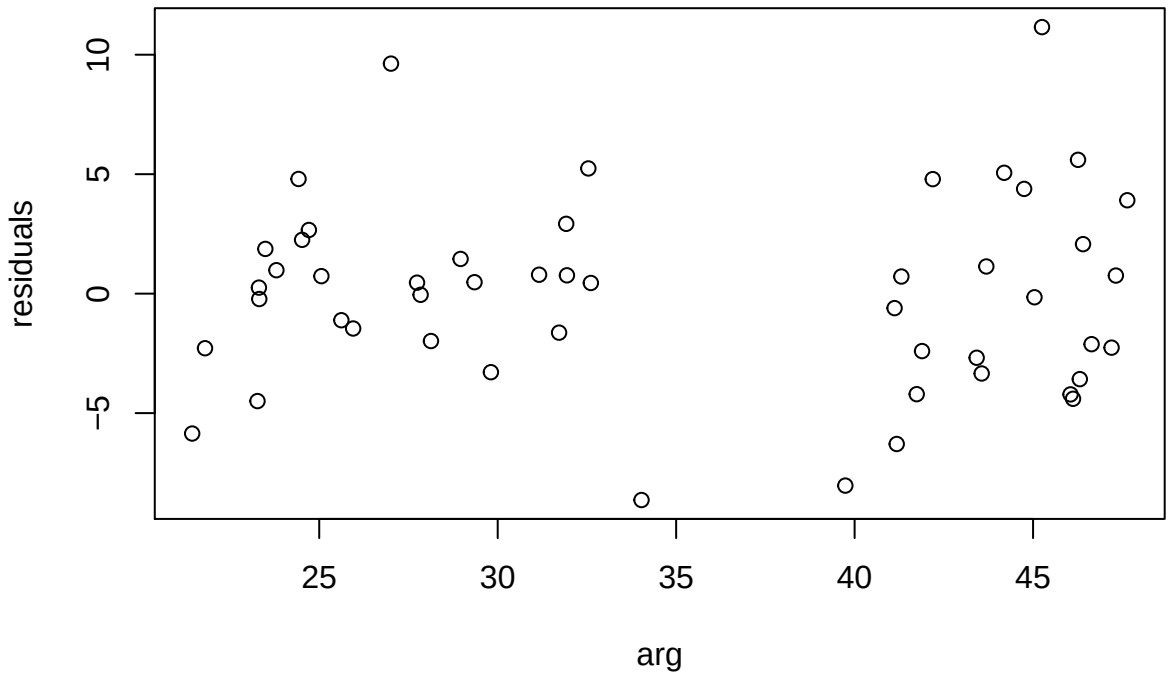

```
## [1] "R-squared: 0.092895211669384"
```

Kā jau vizuāli redzams, pop15 izskaidro lielāko frakciju (~20%) no sr dispersijas un ir negatīvi korelēts ar sr. Pārējie izskaidro ne vairāk kā 10%, bet ir pozitīvi korelēti.

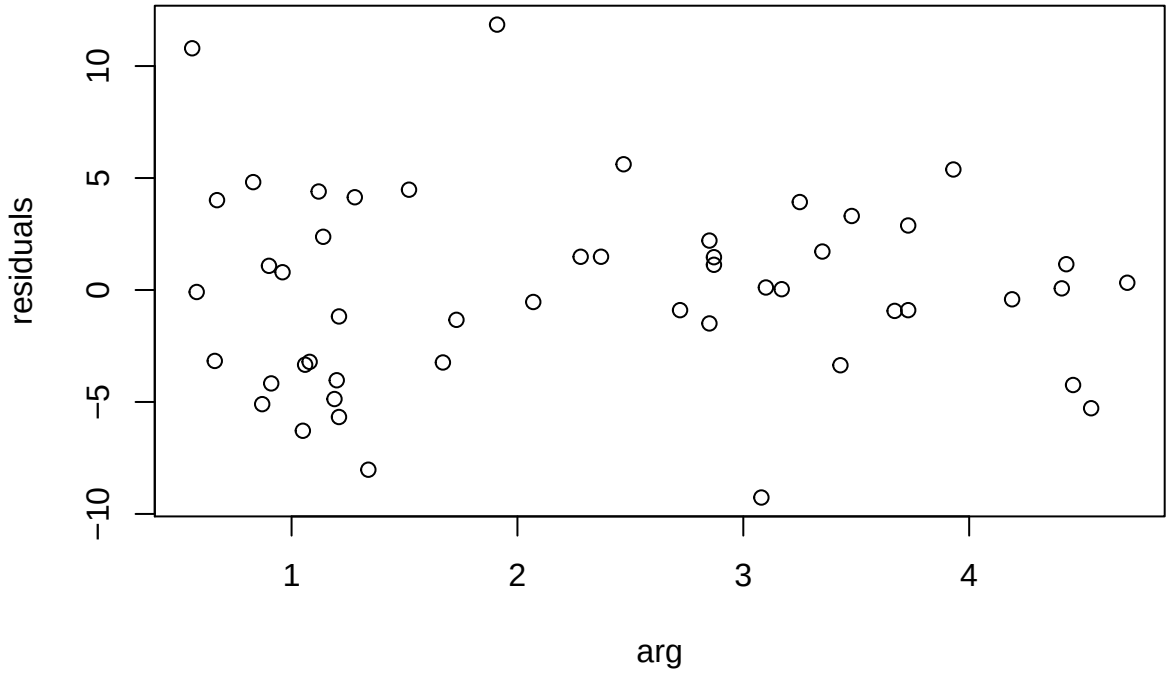
2.3. regresijas modeļu nosacījumu analīze, diagnostika

Atlikumu neatkarība (grafiski):

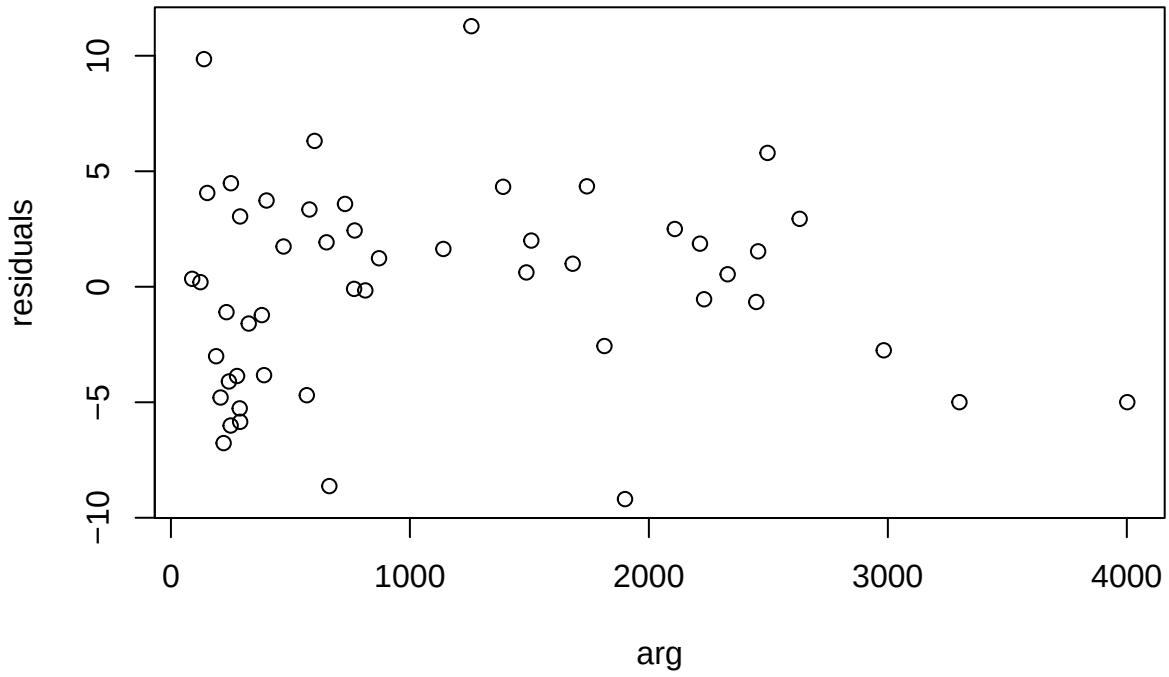
```
plot(pop15,f_pop15$residuals,xlab="arg",ylab="residuals")
```



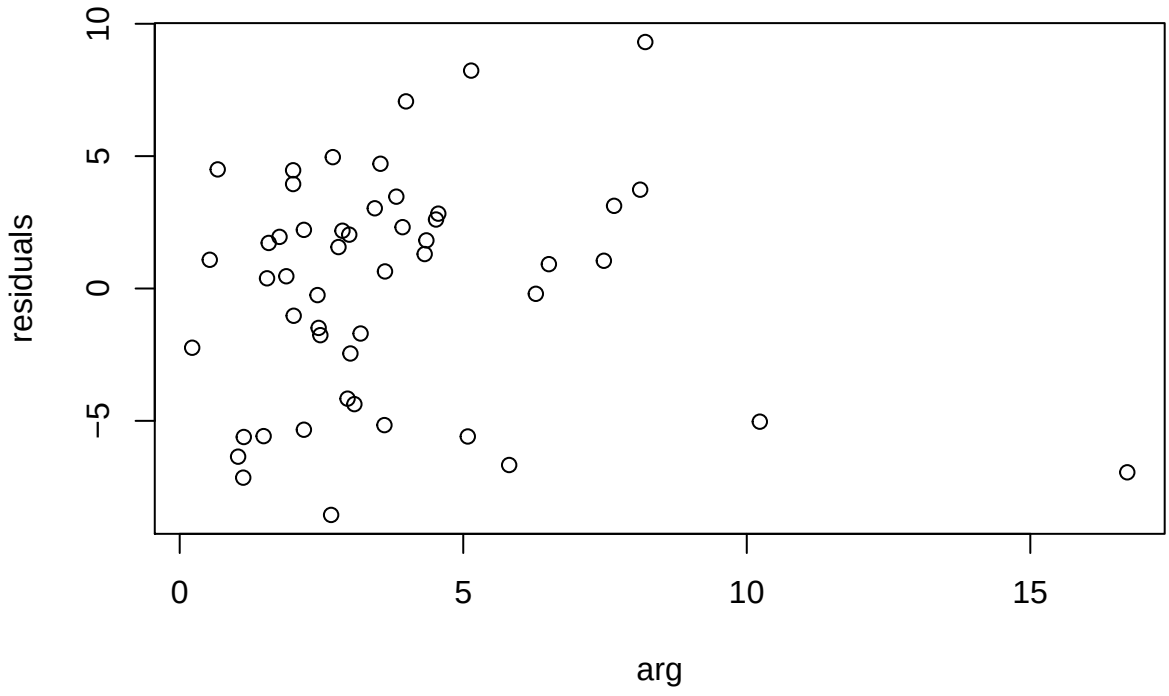

```
plot(pop75,f_pop75$residuals,xlab="arg",ylab="residuals")
```



```
plot(dpi,f_dpi$residuals,xlab="arg",ylab="residuals")
```




```
plot(ddpi,f_ddpi$residuals,xlab="arg",ylab="residuals")
```



Izteiktas sakarības nav redzamas.

Atlikumu neatkarība (Durbin-Watson autokorelācijas tests, TRUE - pastāv autokorelācija):

```
(durbinWatsonTest(f_pop15)$p < 0.05)
```

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

```
(durbinWatsonTest(f_pop75)$p < 0.05)
```

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

```
(durbinWatsonTest(f_dpi)$p < 0.05)
```

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

```
(durbinWatsonTest(f_ddpi)$p < 0.05)
```

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

Atlikumu normalitāte (Kolmogorov-Smirnov tests, TRUE - normāli sadalīti):

```
(lillie.test(f_pop15$residuals)$p.value > 0.05)
```

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

```
(lillie.test(f_pop75$residuals)$p.value > 0.05)
```

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

```
(lillie.test(f_dpi$residuals)$p.value > 0.05)
```

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

```
(lillie.test(f_ddpi$residuals)$p.value > 0.05)
```

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

Dispersijas vienmērība (TRUE - dispersija nav atkarīga no argumenta)

```
(ncvTest(f_pop15)$p > 0.05)
```

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

```
(ncvTest(f_pop75)$p > 0.05)
```

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

```
(ncvTest(f_dpi)$p > 0.05)
```

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

```
(ncvTest(f_ddpi)$p > 0.05)
```

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

Secinājumi

Visos gadījumos lineārās regresijas modelis ir ne īpaši tuvs datiem, taču nav novērotas nozīmīgas autokorelācijas, atlikumi ir normāli sadalīti un to dispersijas ir vienmērīgas, kas neliecina par viegli atrodamām sistemātiskām nobīdēm.