

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.

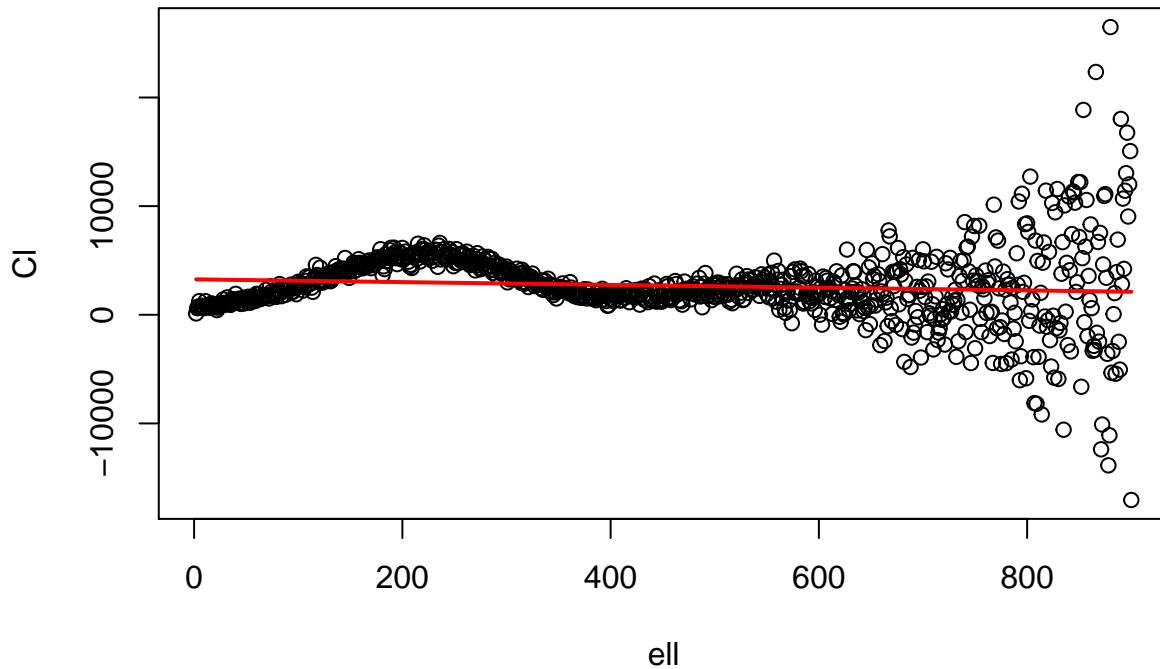
1.1. regresijas modeļu generēšana

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:

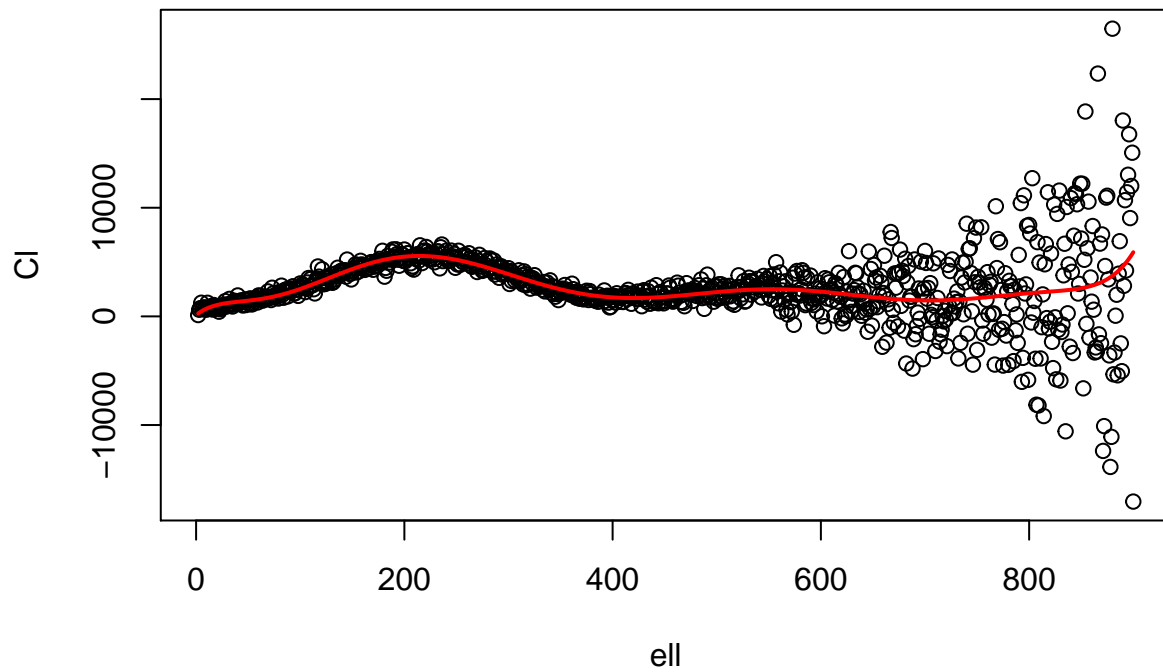
```
res <- bestfit(ell,Cl)  
paste("stopped at x^", res$d,sep="")
```

```
## [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"))
```

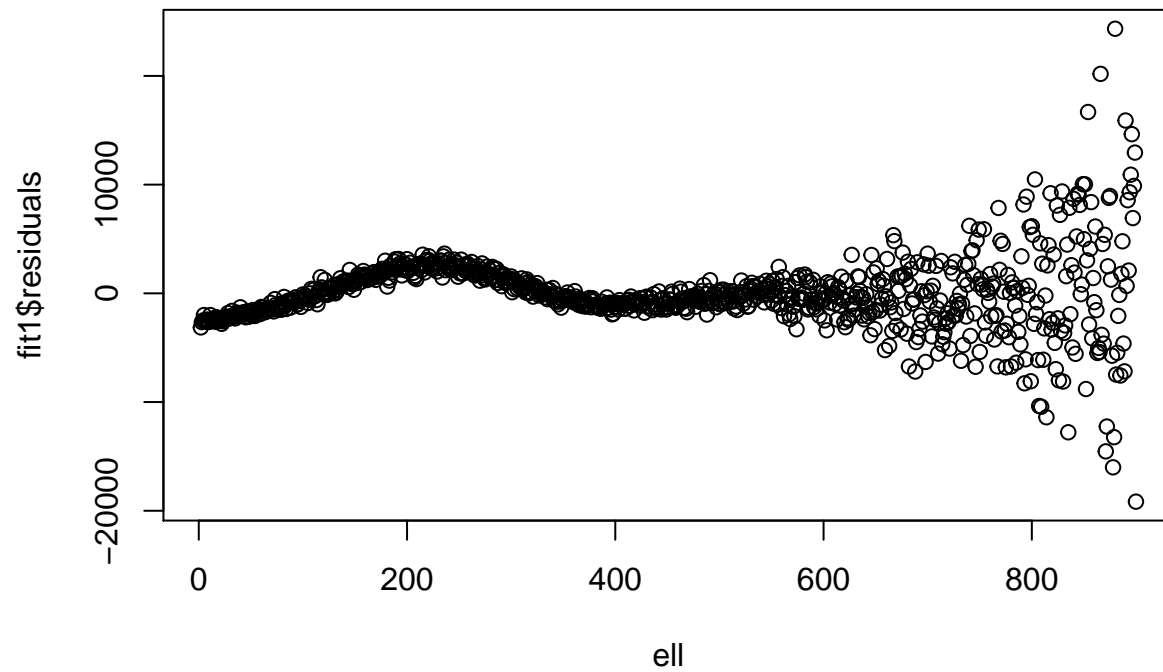


```
## [1] "R-squared: 0.14991001889"
```

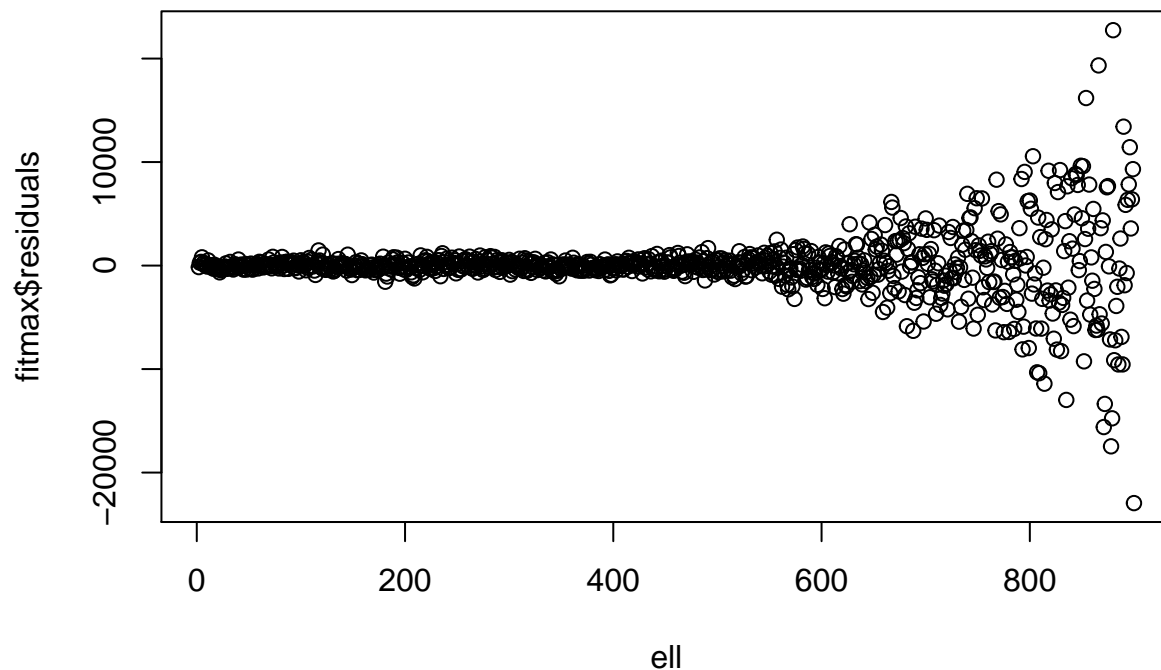
1.2. diagnostika

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 0.014  
## Alternative hypothesis: rho != 0
```

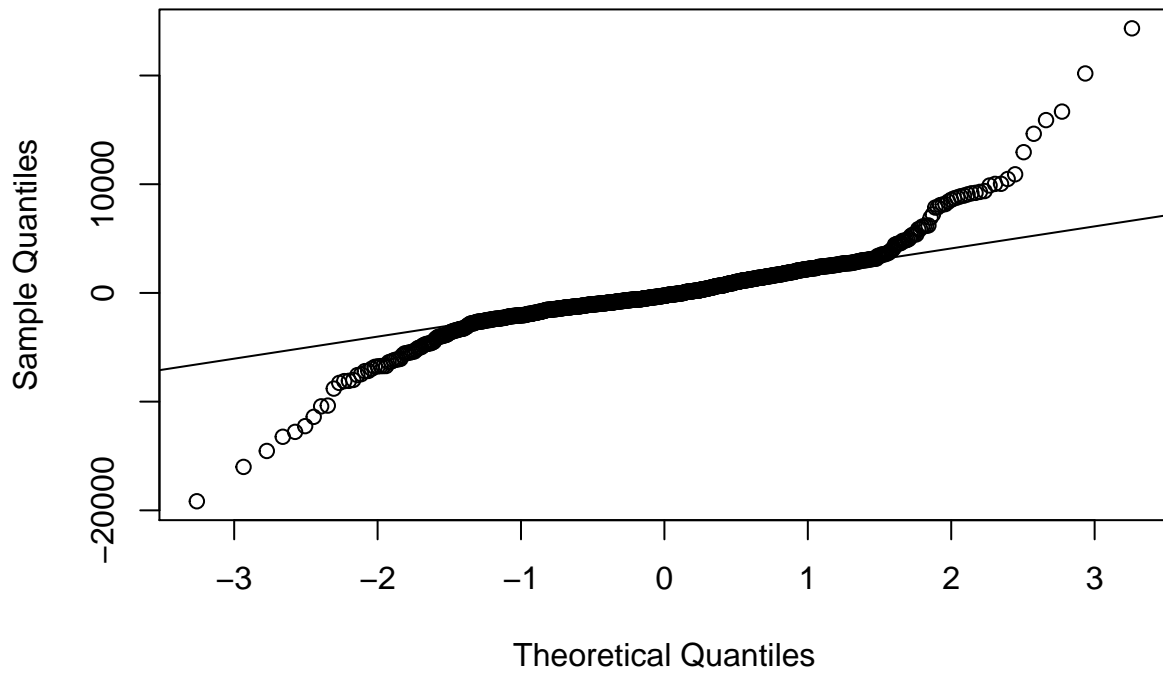
```
durbinWatsonTest(fitmax)
```

```
## lag Autocorrelation D-W Statistic p-value  
## 1 -0.1044366 2.146133 0.056  
## Alternative hypothesis: rho != 0
```

Normalitātes testi - grafiski:

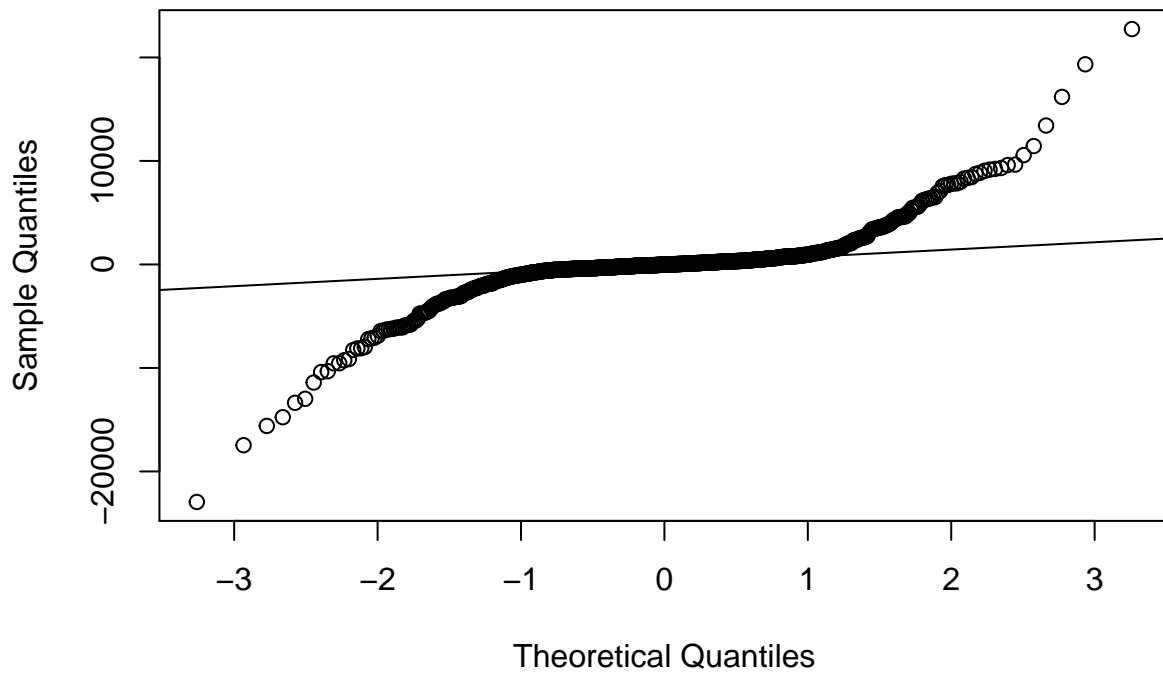
```
qqnorm(fit1$residuals)  
qqline(fit1$residuals)
```

Normal Q-Q Plot



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

Normal Q-Q Plot



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 liknei, ko labi varētu aprakstīt samērā nelielas pakāpes polinoms (sk. sekciju “atlikumu neatkarība - grafiski”, kur šajā reģionā atlikumi 9. pakāpes regresijas liknei ir vienmērīgi sadalīti ap 0). Taču ap $\text{'ell'} = 500$ ļoti strauji pieaug novērojumu dispersija, kas pilnībā izgāž jebkākus mēģinājumus aproksimēt visu datu kopu ar vienu līkni. Šī radikālā izmaiņa dispersijā nomāc arī jebkākus 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

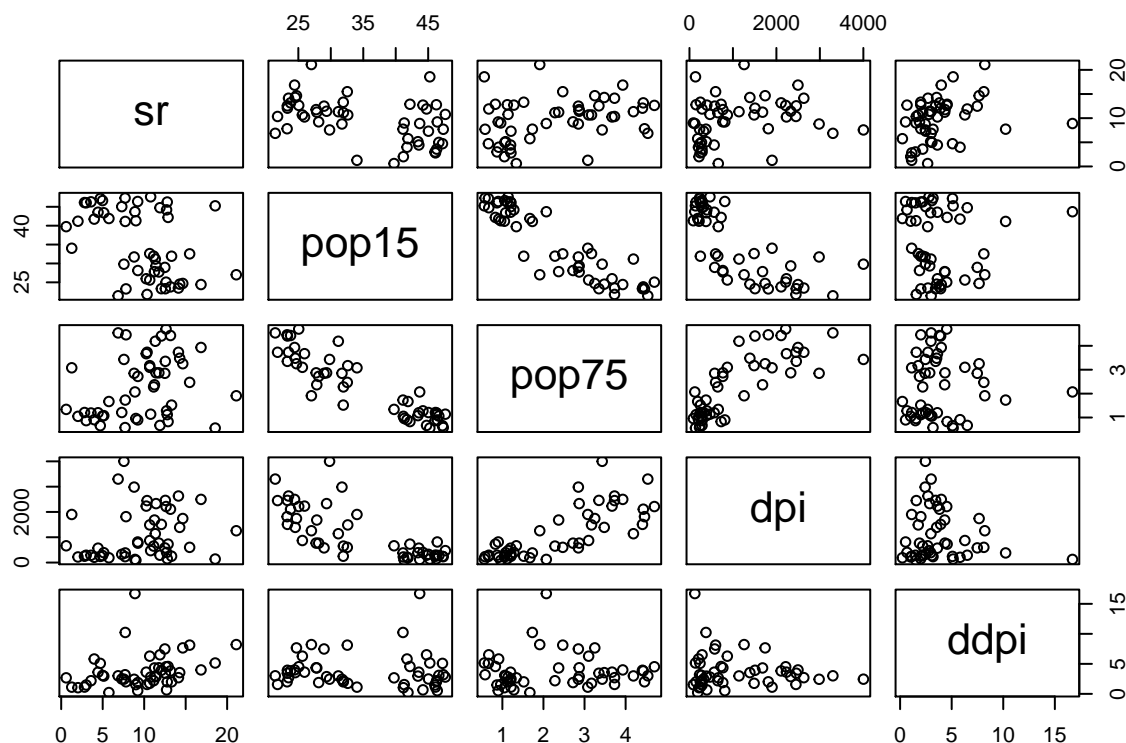
2.1. vispārīgu sakarību meklēšana

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

```
cor(df)
```

```
##           sr      pop15      pop75      dpi      ddpi
## sr      1.000000 -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
```

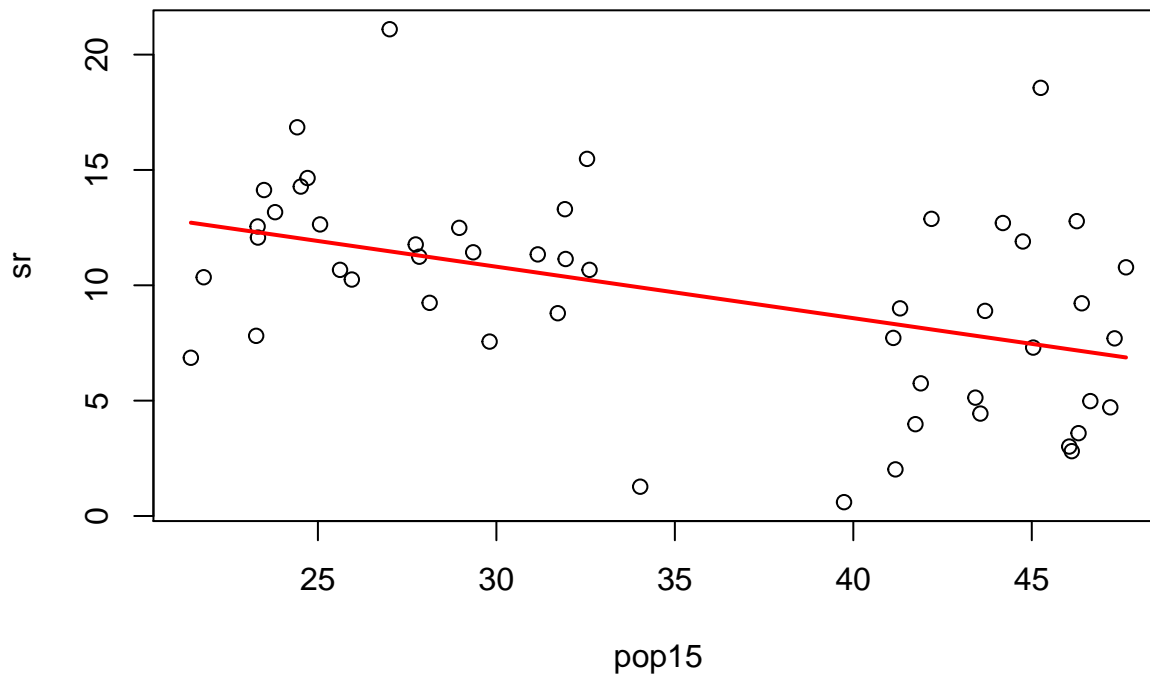
```
pairs(df)
```



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.

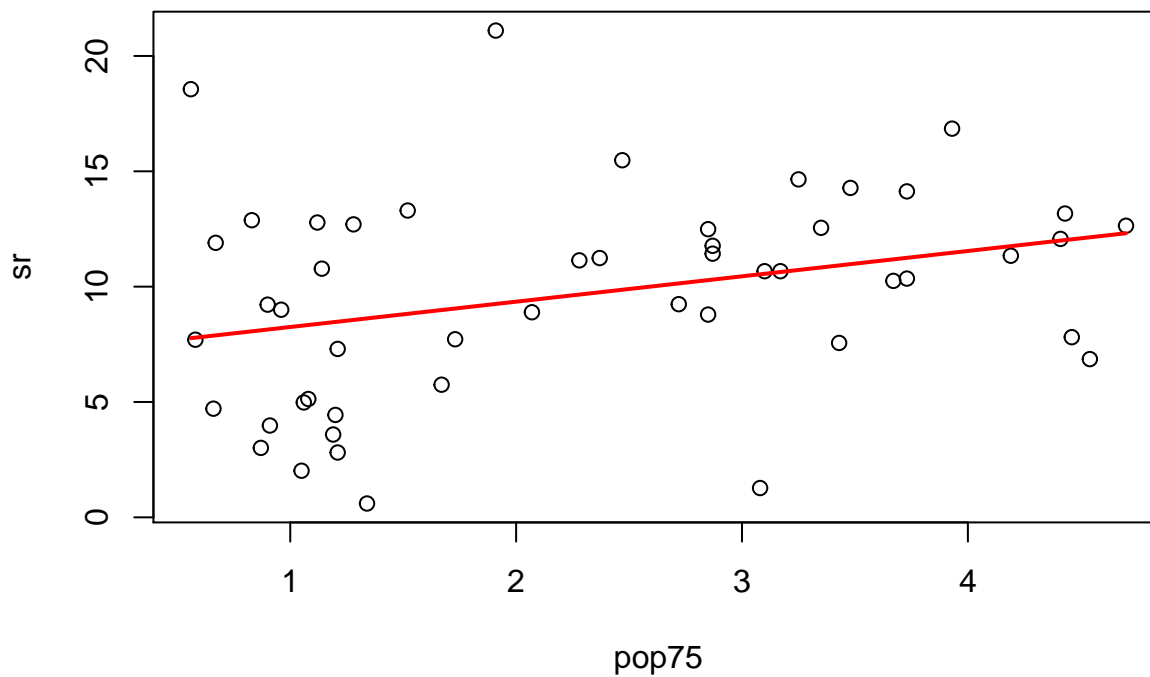
2.2. regresijas modeļu konstruēšana

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



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

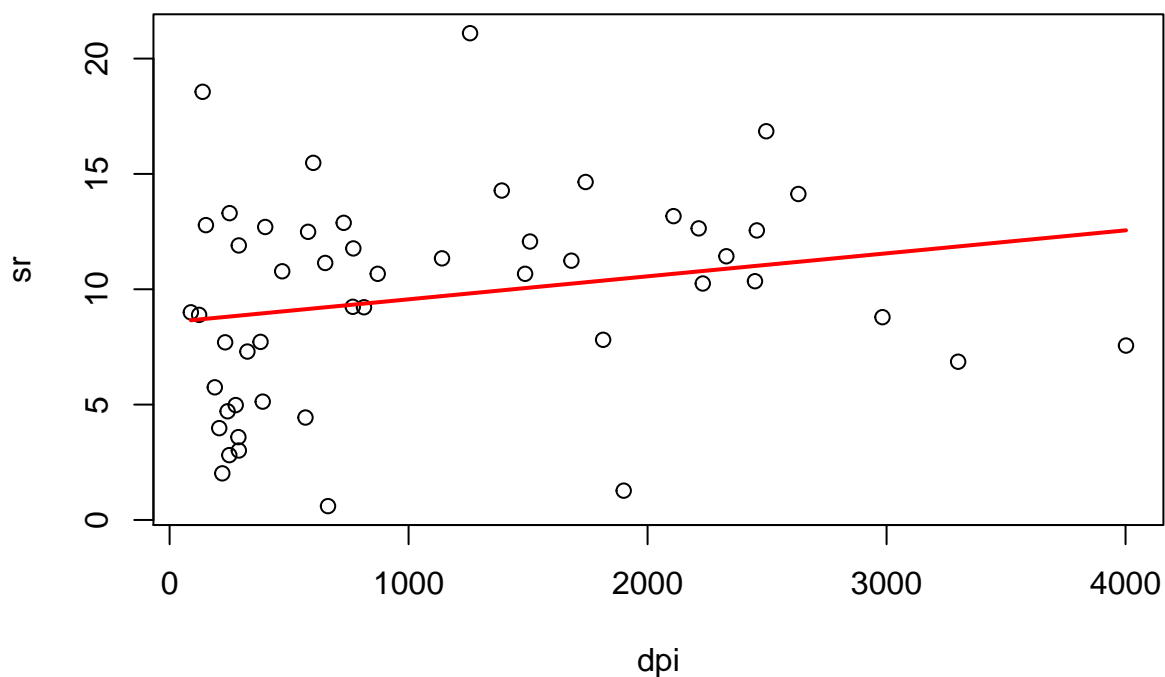
```
f_pop75<-general_lreg(pop75,sr,plot=T,print=T,names=c("pop75","sr"))
```



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

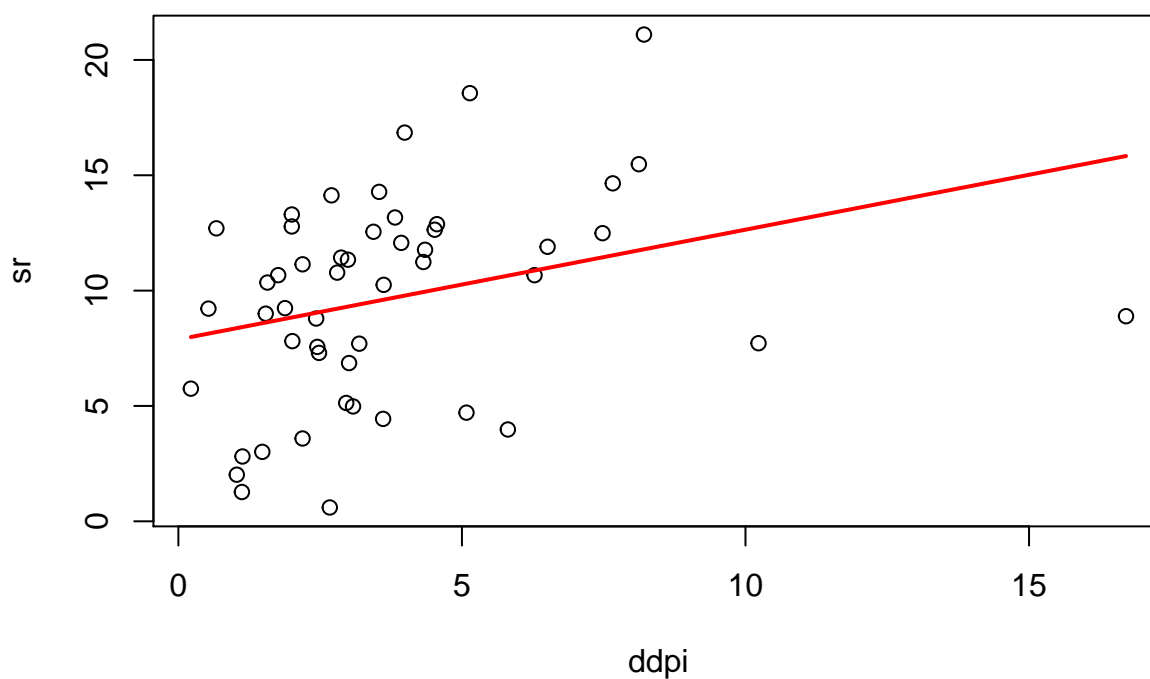


```
f_dpi<-general_lreg(dpi,sr,plot=T,print=T,names=c("dpi","sr"))
```



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

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



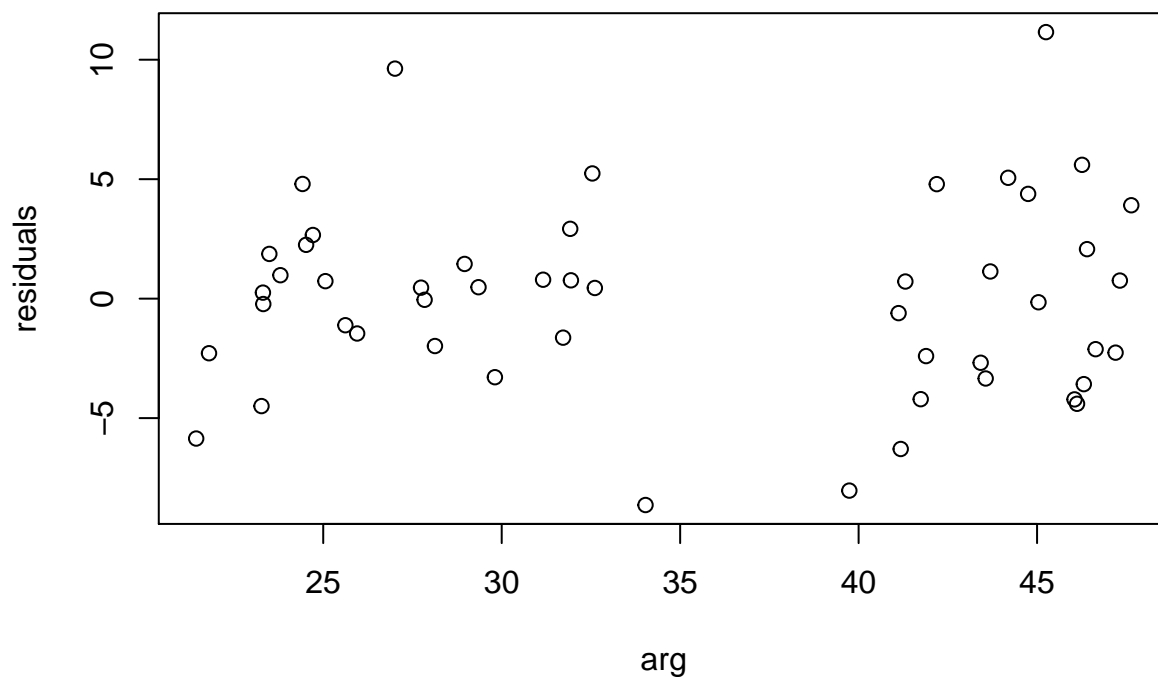
```
## [1] "R-squared: 0.092895211669384"
```

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.

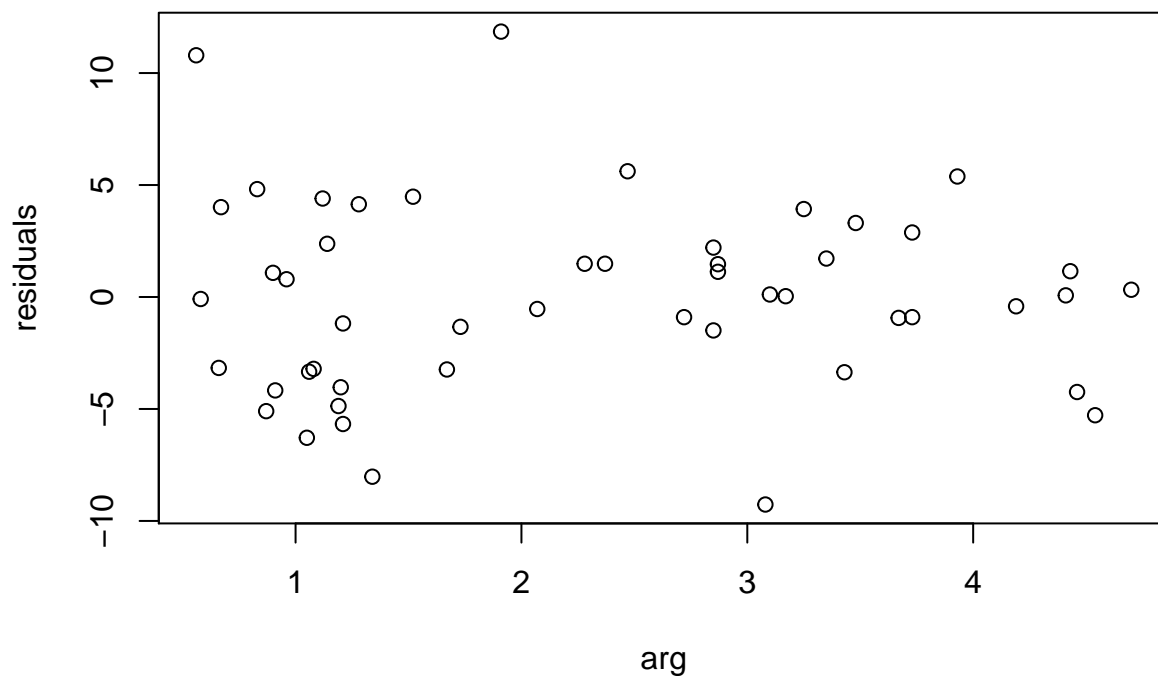
2.3. regresijas modeļu nosacījumu analīze, diagnostika

Atlikumu neatkarība (grafiski):

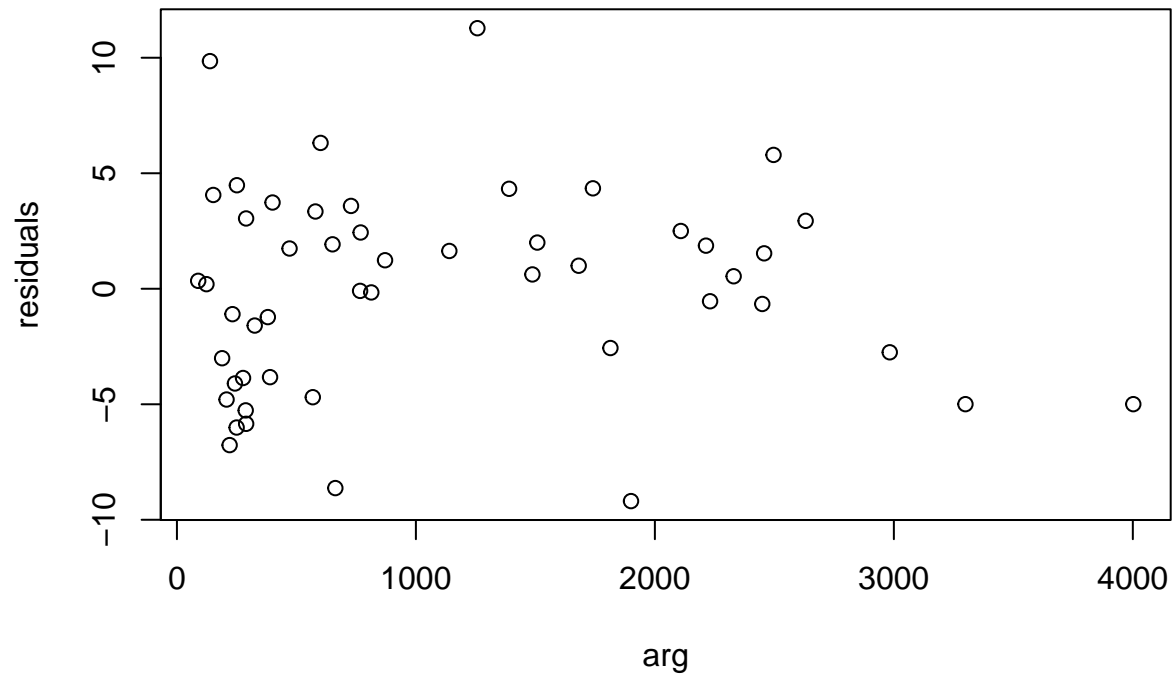
```
plot(pop15,f_pop15$residuals,xlab="arg",ylab="residuals")
```



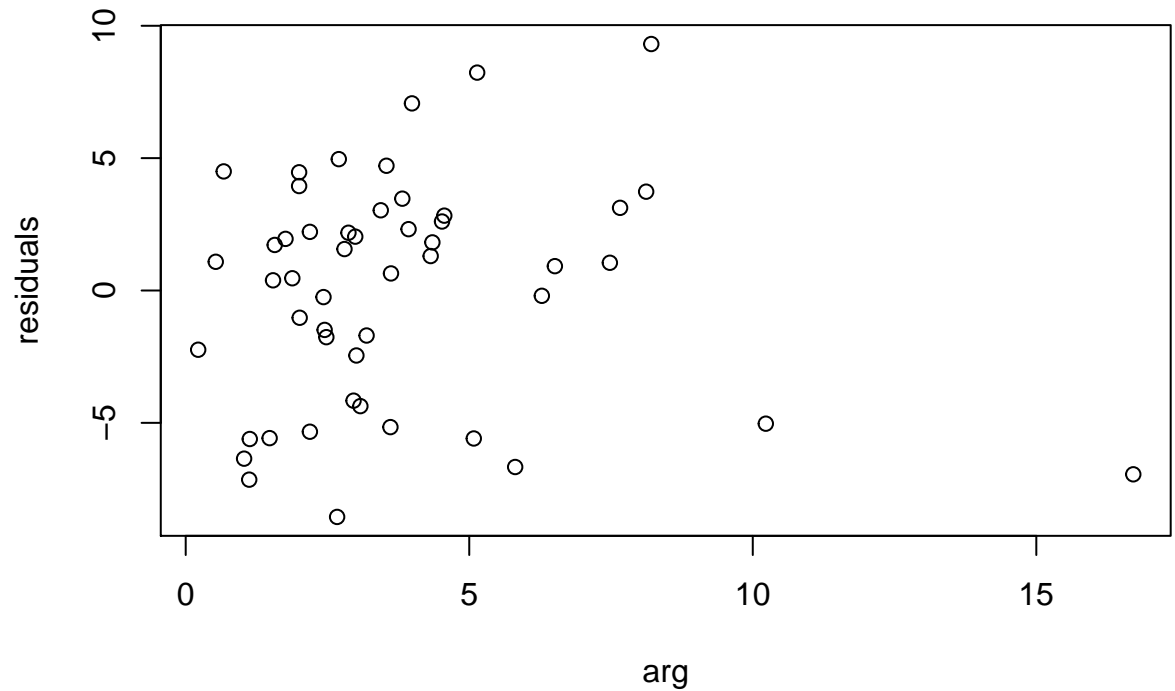
```
plot(pop75,f_pop75$residuals,xlab="arg",ylab="residuals")
```



```
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

```

2.4. daudzfaktoru regresija

Šo apakšuzdevumu gandrīz palaidu garām, jo uzdevuma nosacījumos prasīts veikt “vienkāršas lineāras regresijas”, ko var pārprast kā nosacījumu veikt individuālas viena faktora regresijas. Jebkurā gadījumā, šeit veikta daudzfaktoru regresija un rezultātu analīze kā mtcars piemērā lekcijās:

```

fit_multi<-lm(sr~., data=df)
summary(fit_multi)

##
## Call:
## lm(formula = sr ~ ., data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max

```

```
## -8.2422 -2.6857 -0.2488 2.4280 9.7509
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 28.5660865  7.3545161   3.884 0.000334 ***
## pop15       -0.4611931  0.1446422  -3.189 0.002603 **
## pop75       -1.6914977  1.0835989  -1.561 0.125530
## dpi         -0.0003369  0.0009311  -0.362 0.719173
## ddpi         0.4096949  0.1961971   2.088 0.042471 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.803 on 45 degrees of freedom
## Multiple R-squared:  0.3385, Adjusted R-squared:  0.2797
## F-statistic: 5.756 on 4 and 45 DF,  p-value: 0.0007904
```

VIF analyze:

```
vif(fit_multi)
```

```
##      pop15      pop75      dpi      ddpi
## 5.937661 6.629105 2.884369 1.074309
```

Neviena vērtība netiek izmesta.

Iteratīvā uzlabošana pēc AIC metrikas:

```
fit_multi<-step(fit_multi)
```

```
## Start:  AIC=138.3
## sr ~ pop15 + pop75 + dpi + ddpi
##
##           Df Sum of Sq  RSS   AIC
## - dpi      1      1.893 652.61 136.45
## <none>                 650.71 138.30
## - pop75    1     35.236 685.95 138.94
## - ddpi     1     63.054 713.77 140.93
## - pop15    1    147.012 797.72 146.49
##
## Step:  AIC=136.45
## sr ~ pop15 + pop75 + ddpi
##
##           Df Sum of Sq  RSS   AIC
## <none>                 652.61 136.45
## - pop75    1     47.946 700.55 137.99
## - ddpi     1     73.562 726.17 139.79
## - pop15    1    145.789 798.40 144.53
```

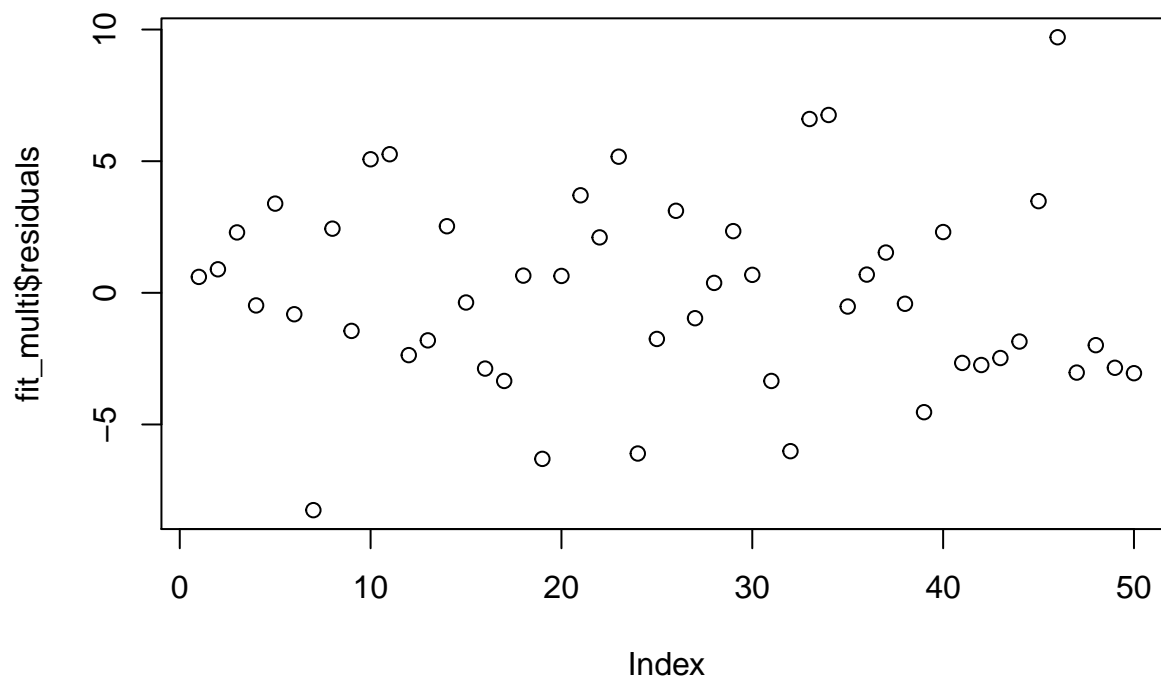
```
summary(fit_multi)
```

```
##
## Call:
## lm(formula = sr ~ pop15 + pop75 + ddpi, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.2539 -2.6159 -0.3913  2.3344  9.7070
##
```

```
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  28.1247     7.1838   3.915 0.000297 ***
## pop15        -0.4518     0.1409  -3.206 0.002452 **
## pop75        -1.8354     0.9984  -1.838 0.072473 .
## ddpi          0.4278     0.1879   2.277 0.027478 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.767 on 46 degrees of freedom
## Multiple R-squared:  0.3365, Adjusted R-squared:  0.2933
## F-statistic: 7.778 on 3 and 46 DF,  p-value: 0.0002646
```

Testi tāpat kā viena faktora regresijās:

```
plot(fit_multi$residuals)
```



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

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

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

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

```
(ncvTest(fit_multi)$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. Daudzfaktoru regresijas modelis sniedz mērenu uzlabojumu (det.koef 20% -> 29%).

3. uzdevums - ANOVA

Datu ielasišana

```
df <- chickwts
attach(df)
summary(df)
```

```
##      weight      feed
## Min.   :108.0 casein  :12
## 1st Qu.:204.5 horsebean:10
## Median :258.0 linseed  :12
## Mean   :261.3 meatmeal :11
## 3rd Qu.:323.5 soybean  :14
## Max.   :423.0 sunflower:12
```

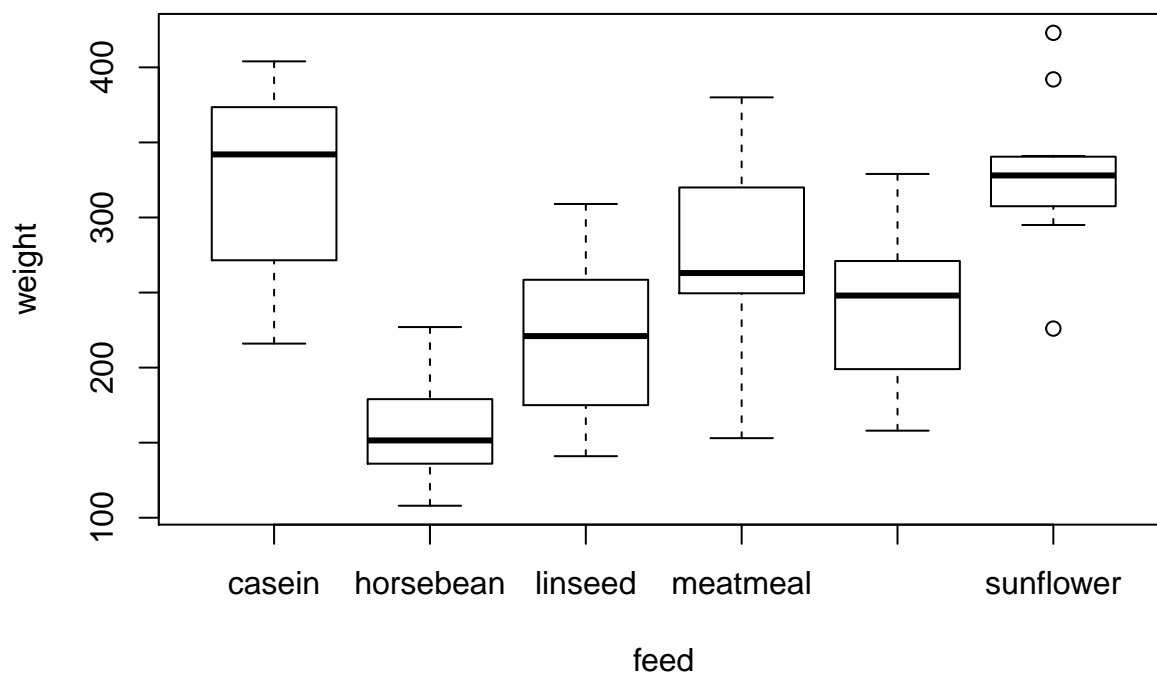
a) Kastu grafiki, aprakstošās statistikas

Bibliotēka

```
library(psych)
```

Kastu grafiki, aprakstošās statistikas:

```
boxplot(weight~feed)
```



```
describe(weight[feed=="casein"])
```

```
##      vars  n  mean    sd median trimmed  mad min max range skew kurtosis  se
## X1      1 12 323.58 64.43   342   326.3 63.01 216 404   188 -0.46    -1.37 18.6
```

```
describe(weight[feed=="horsebean"])
```

```
##      vars  n  mean    sd median trimmed  mad min max range skew kurtosis  se
## X1      1 10 160.2  38.63  151.5  158.38 32.62 108 227   119  0.47    -1.19 12.21
```

```
describe(weight[feed=="linseed"])

##      vars  n   mean    sd median trimmed  mad min max range skew kurtosis   se
## X1      1 12 218.75 52.24   221   217.5 58.56 141 309   168 0.01    -1.33 15.08

describe(weight[feed=="meatmeal"])

##      vars  n   mean    sd median trimmed  mad min max range skew kurtosis   se
## X1      1 11 276.91 64.9   263   279.22 77.1 153 380   227 -0.25    -0.93 19.57

describe(weight[feed=="soybean"])

##      vars  n   mean    sd median trimmed  mad min max range skew kurtosis   se
## X1      1 14 246.43 54.13   248   246.92 53.37 158 329   171 0.03    -1.17 14.47

describe(weight[feed=="sunflower"])

##      vars  n   mean    sd median trimmed  mad min max range skew kurtosis   se
## X1      1 12 328.92 48.84   328   329.8 18.53 226 423   197 -0.05     0.06 14.1
```

b) ANOVA modelis

```
fit<-aov(weight~feed)
summary(fit)

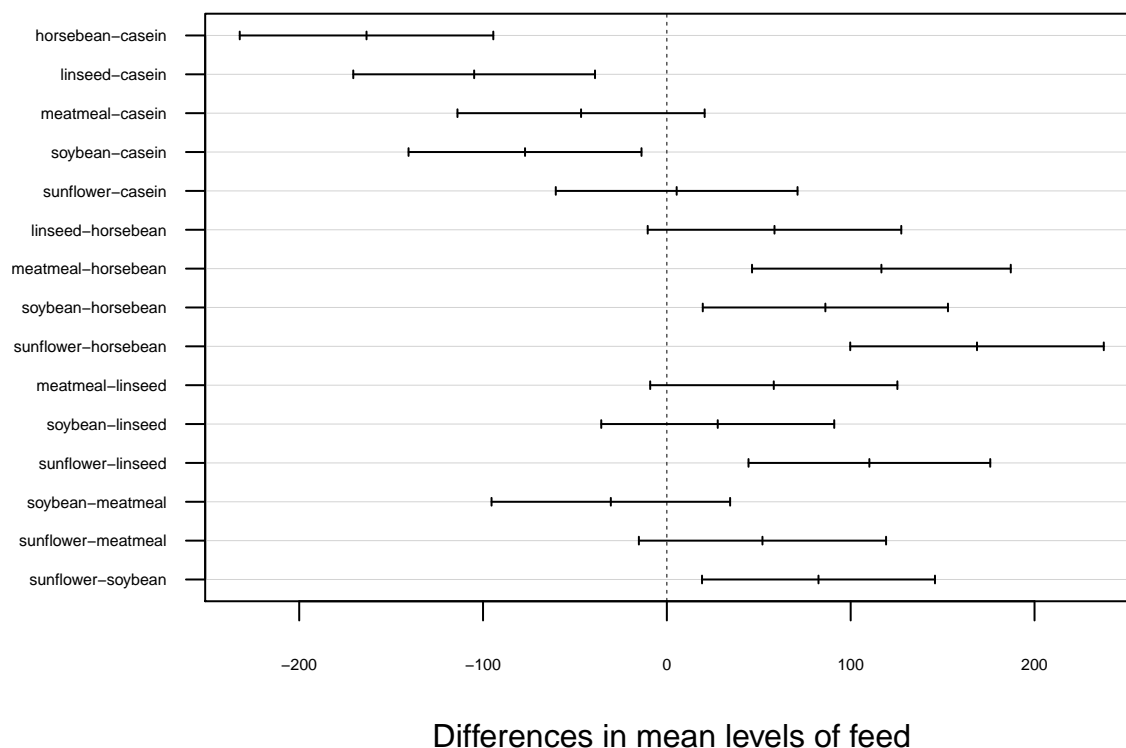
##              Df Sum Sq Mean Sq F value    Pr(>F)
## feed              5 231129   46226   15.37 5.94e-10 ***
## Residuals        65 195556    3009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Rezultāts ir gaidītais - grupas nepieder pie viena sadalījuma.

c) Post-hoc salīdzinājums pa pāriem

```
r<-TukeyHSD(fit)
op <- par(mar= c(4,5,3,3) + 0.1, cex.axis=0.5)
plot(TukeyHSD(fit),las=1)
```


95% family-wise confidence level



```
par(op)
```

Grupas, kas varētu pārklāties:

- meatmeal-casein;
- sunflower-casein;
- linseed-horsebean;
- meatmeal-linseed;
- soybean-linseed;
- soybean-meatmeal;
- sunflower-meatmeal.

d) ANOVA pieņēmumu pārbaude

Normalitāte:

```
library(dplyr)
library(rstatix)
df %>% group_by(feed) %>% shapiro_test(weight)
```

```
## # A tibble: 6 x 4
##   feed      variable statistic    p
##   <fct>      <chr>      <dbl> <dbl>
## 1 casein    weight         0.917 0.259
## 2 horsebean weight         0.938 0.526
## 3 linseed   weight         0.969 0.903
## 4 meatmeal  weight         0.979 0.961
## 5 soybean   weight         0.946 0.506
## 6 sunflower weight         0.928 0.360
```

Nevienai grupai nevar noraidīt.

Dispersijas vienmērība:

```
leveneTest(weight,feed)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group  5  0.7493 0.5896
##      65
```

```
bartlett.test(weight,feed)
```

```
##
## Bartlett test of homogeneity of variances
##
## data: weight and feed
## Bartlett's K-squared = 3.2597, df = 5, p-value = 0.66
```

Arī to nevar noraidīt.

e) neparametriskā ANOVA procedūra

Tā kā šajā gadījumā grupu sadalījumu normalitāti un dispersiju vienmērību noraidīt nevar, stingri runājot neparametriskās metodes nav nepieciešamas. Taču Kruskal-test procedūru var veikt jebkurā gadījumā:

```
kruskal.test(weight~feed)
```

```
##
## Kruskal-Wallis rank sum test
##
## data: weight by feed
## Kruskal-Wallis chi-squared = 37.343, df = 5, p-value = 5.113e-07
```

Kā iepriekš, visu sadalījumu vienādības hipotēze tiek pārliecinoši noraidīta.

Pāru salīdzināšanai var lietot pāru Vilkoksa testu:

```
pairwise.wilcox.test(weight,feed,p.adjust.method="BH")
```

```
##
## Pairwise comparisons using Wilcoxon rank sum test
##
## data: weight and feed
##
##      casein  horsebean linseed meatmeal soybean
## horsebean 0.00016 -          -          -
## linseed   0.00305 0.01191 -          -
## meatmeal  0.11355 0.00096 0.05451 -
## soybean   0.01110 0.00227 0.27306 0.28035 -
## sunflower 1.00000 9.3e-05 0.00025 0.09384 0.00334
##
## P value adjustment method: BH
```

Grupas, kas varētu pārklāties:

- meatmeal-casein;
- sunflower-casein;
- meatmeal-linseed;
- soybean-linseed;

- soybean-meatmeal;
- sunflower-meatmeal.

Salīdzinot ar iepriekšējo, parametrisko metodi atkrīt:

- linseed-horsebean.

f) neparametriskais post-hoc tests

Izmantojot Dunn testu:

```
dunn_test(df, weight~feed)
```

```
## # A tibble: 15 x 9
##   .y.   group1 group2    n1    n2 statistic      p    p.adj p.adj.signif
## * <chr> <chr>  <chr>  <int> <int>    <dbl>    <dbl>    <dbl> <chr>
## 1 weight casein  horseb~    12    10    -4.81  1.49e-6  2.08e-5 ****
## 2 weight casein  linseed    12    12    -3.31  9.39e-4  1.03e-2 *
## 3 weight casein  meatme~    12    11    -1.42  1.57e-1  6.27e-1 ns
## 4 weight casein  soybean    12    14    -2.50  1.24e-2  9.94e-2 ns
## 5 weight casein  sunflo~    12    12     0.183  8.55e-1  9.90e-1 ns
## 6 weight horsebe~ linseed    10    12     1.66  9.72e-2  5.83e-1 ns
## 7 weight horsebe~ meatme~    10    11     3.36  7.68e-4  9.22e-3 **
## 8 weight horsebe~ soybean    10    14     2.60  9.27e-3  8.34e-2 ns
## 9 weight horsebe~ sunflo~    10    12     4.99  6.12e-7  9.17e-6 ****
## 10 weight linseed meatme~    12    11     1.82  6.88e-2  4.82e-1 ns
## 11 weight linseed soybean    12    14     0.933  3.51e-1  9.90e-1 ns
## 12 weight linseed sunflo~    12    12     3.49  4.81e-4  6.25e-3 **
## 13 weight meatmeal soybean    11    14    -0.974  3.30e-1  9.90e-1 ns
## 14 weight meatmeal sunflo~    11    12     1.59  1.11e-1  5.83e-1 ns
## 15 weight soybean  sunflo~    14    12     2.69  7.15e-3  7.15e-2 ns
```

Grupas, kuru pārklāšanos nevar izslēgt:

- meatmeal-casein;
- soybean-casein;
- sunflower-casein;
- linseed-horsebean;
- soybean-horsebean;
- meatmeal-linseed;
- soybean-linseed;
- soybean-meatmeal;
- sunflower-meatmeal;
- sunflower-soybean.

Salīdzinot ar parametrisko metodi, klāt nākušas:

- soybean-casein;
- soybean-horsebean;
- sunflower-soybean.

g) secinājumi un komentāri

- Kaut gan ANOVA normalitātes un dispersijas vienmērīguma nosacījumi it kā reti izpildās, šajā datu kopā nekādas problēmas tie nesagādā;
- Dažas grupas ļoti robusti iztur vienādības pārbaudes visās testu kategorijās, bet citas ir uz robežas - dažādi testi sniedz dažādas atbildes - kaut gan tās ir ~5% p-vērtības visos variantos.

4. uzdevums - 2-faktoru ANOVA

Ceru, ka esmu pareizi sapratis uzdevuma nosacījumus, jo šķiet, ka šajā punktā jādara daudz mazāk nekā citos.

Datu ielasišana:

```
df <- ToothGrowth
attach(df)
summary(df)
```

```
##          len          supp          dose
## Min.      : 4.20    OJ:30    Min.      :0.500
## 1st Qu.:13.07    VC:30    1st Qu.:0.500
## Median :19.25                Median :1.000
## Mean      :18.81                Mean      :1.167
## 3rd Qu.:25.27                3rd Qu.:2.000
## Max.      :33.90                Max.      :2.000
```

Datu kolonnas:

- len - zobu garums;
- supp - uztura bagātinātājs (2 kategorijas);
- dose - doza (3 līmeņi - {0.5, 1, 2}).

2-faktoru ANOVA novērtējums:

```
summary(aov(len ~ supp*dose))
```

```
##          Df Sum Sq Mean Sq F value    Pr(>F)
## supp      1  205.4    205.4   12.317 0.000894 ***
## dose      1 2224.3   2224.3  133.415 < 2e-16 ***
## supp:dose  1   88.9     88.9    5.333 0.024631 *
## Residuals 56  933.6     16.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANOVA tests liecina, ka katra kategorija ir statistiski nozīmīga ($p \rightarrow 0$). Abu kategoriju mijiedarbība izdalīta kā neatkarīgs faktors nav ne tuvu tik nozīmīga kā katrs atsevišķi, taču p-vērtība tik un tā ir zem 5%, kas ļauj šo mijiedarbību arī atzīt par statistiski nozīmīgu.

Izmantojot HSD.test, var redzēt, kā kategorijas grupējas:

```
library(agricolae)
tx<-with(df, interaction(supp,dose))
amod <- aov(len ~ tx)
HSD.test(amod, "tx", group=TRUE,console=TRUE)
```

```
##
## Study: amod ~ "tx"
##
## HSD Test for len
##
## Mean Square Error:  13.18715
##
## tx,  means
##
##          len          std  r  Min  Max
## OJ.0.5 13.23 4.459709 10  8.2 21.5
## OJ.1   22.70 3.910953 10 14.5 27.3
```

```

## OJ.2    26.06 2.655058 10 22.4 30.9
## VC.0.5   7.98 2.746634 10  4.2 11.5
## VC.1    16.77 2.515309 10 13.6 22.5
## VC.2    26.14 4.797731 10 18.5 33.9
##
## Alpha: 0.05 ; DF Error: 54
## Critical Value of Studentized Range: 4.178265
##
## Minimum Significant Difference: 4.798124
##
## Treatments with the same letter are not significantly different.
##
##          len groups
## VC.2    26.14      a
## OJ.2    26.06      a
## OJ.1    22.70      a
## VC.1    16.77      b
## OJ.0.5  13.23      b
## VC.0.5   7.98      c

```

Grupās, kas izdalāmas arī ar grafisko metodi, kas apskatīta lekcijā, taču konfliktē ar Markdown kompilatoru, redzams, ka pieaugot dozai, mazinās atšķirības starp uztura bagātinātājiem.

5. uzdevums - joslas platuma meklēšana

Datu ielāde (dots speciāls 5-modāls sadalījums, kas lekcijās izmantots, lai ilustrētu pdf novērtējuma metožu trūkumus):

```
df <- as.numeric(read.delim("dati2_5.txt",header=F,sep=" "))
```

5.1. uzdevums - joslas platuma atrašana

Izmantojot R iebūvēto krosvalidācijas optimizatoru:

```
cross_validated<-bw.ucv(df)
cross_validated
```

```
## [1] 0.05095563
```

5.2. ilustrēt KDE aproksimāciju pret histogrammu

Lai uzskatāmāk parādītu arī “nepiegludinātus” un “pārgludinātus” PDF tuvinājumus, zīmētas līknes ar $h = 1$ un $h = 0.01$:

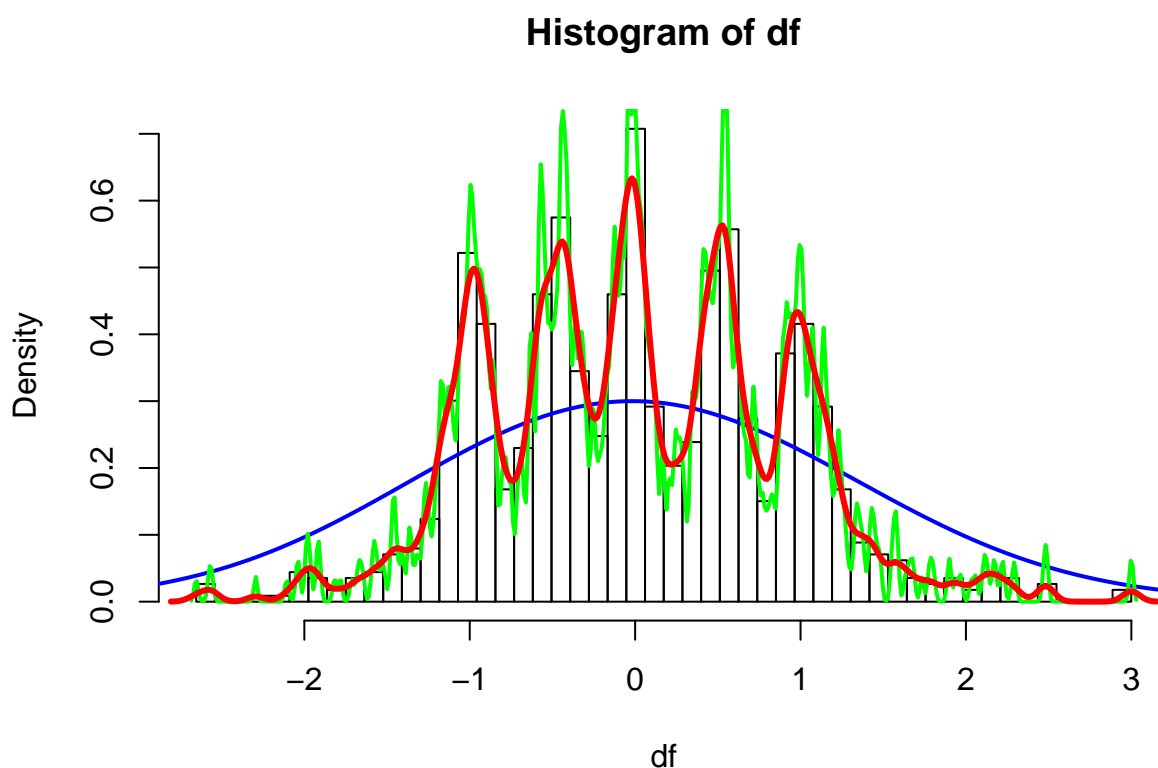
```
library(histogram)
hh<-histogram(df,type="regular",penalty="cv")
```

```
## Building regular histogram with maximum number of bins 144.
```

```
## - Choosing number of bins via leave-1-out cross validation. Using formula 1.
```

```
## - Number of bins chosen: 50.
```

```
lines(density(df,bw=1),col="blue",lwd=2)
lines(density(df,bw=0.01),col="green",lwd=2)
lines(density(df,bw="ucv"),col="red",lwd=3)
```



5.3. salīdzināt ar citām joslas platuma novērtējuma metodēm

Grāmatā “Nonparametric and Semiparametric Models” dota formula t.s. “Silverman rule of thumb” heuristikai, kas pieņem normālā sadalījuma otro atvasinājumu un līdz ar to ļauj iegūt joslas platumu no datu kopas standartnovirzes novērtējuma. Šo un nedaudz paplašinātu heuristiku arī piedāvā R iebūvētais joslas platuma noteikšanas rīks:

```
bw.nrd0(df)
```

```
## [1] 0.1952054
```

```
bw.nrd(df)
```

```
## [1] 0.2299086
```

Kā redzams, iegūtie rezultāti ir ļoti līdzīgi. Salīdzinot tuvāko (mazāko) ar krosvalidācijas optimizatora iegūto:

```
hh<-histogram(df,type="regular",penalty="cv")
```

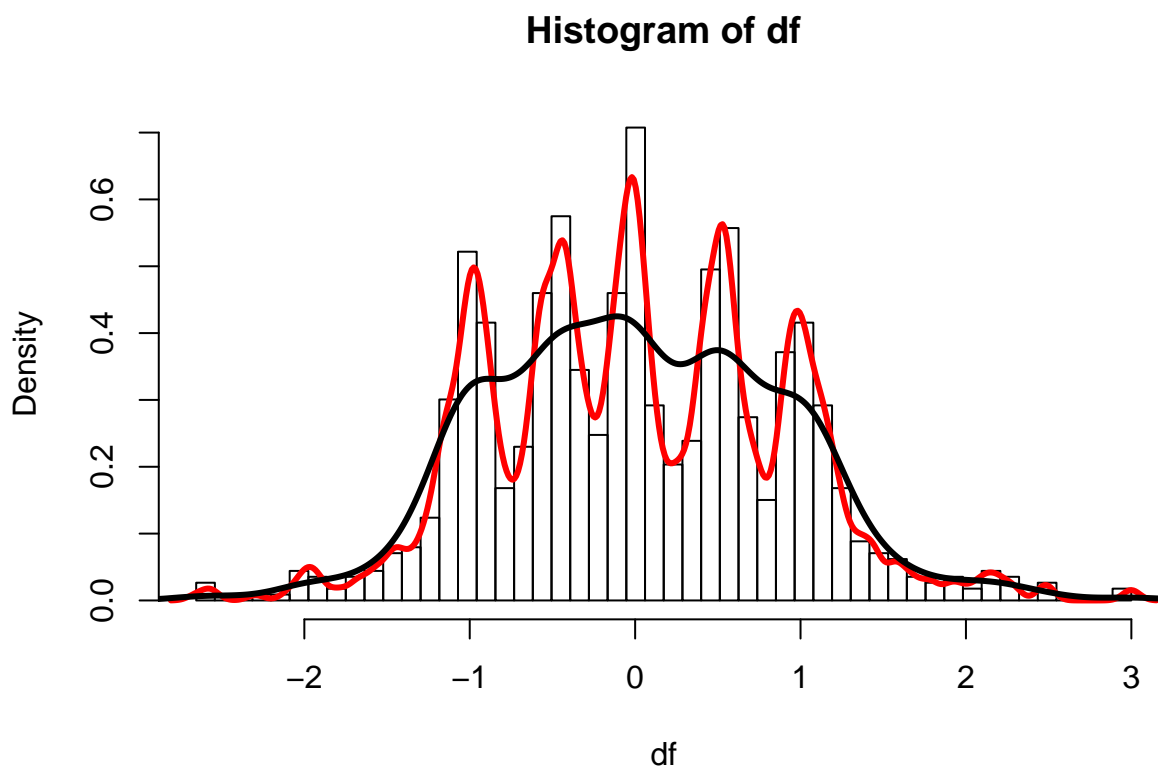
```
## Building regular histogram with maximum number of bins 144.
```

```
## - Choosing number of bins via leave-1-out cross validation. Using formula 1.
```

```
## - Number of bins chosen: 50.
```

```
lines(density(df,bw="ucv"),col="red",lwd=3)
```

```
lines(density(df,bw="nrd0"),col="black",lwd=3)
```



Kā redzams, ar krosvalidācijas metodi iegūtā KDE līkne labāk seko īstajam sadalījumam.

6. uzdevums - neparametriskā regresija

Datu ielāde:

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

6.1. regresiju ģenerēšana

Polinomiālā regresija ar jau iepriekš noskaidroto “labāko” vērtību:

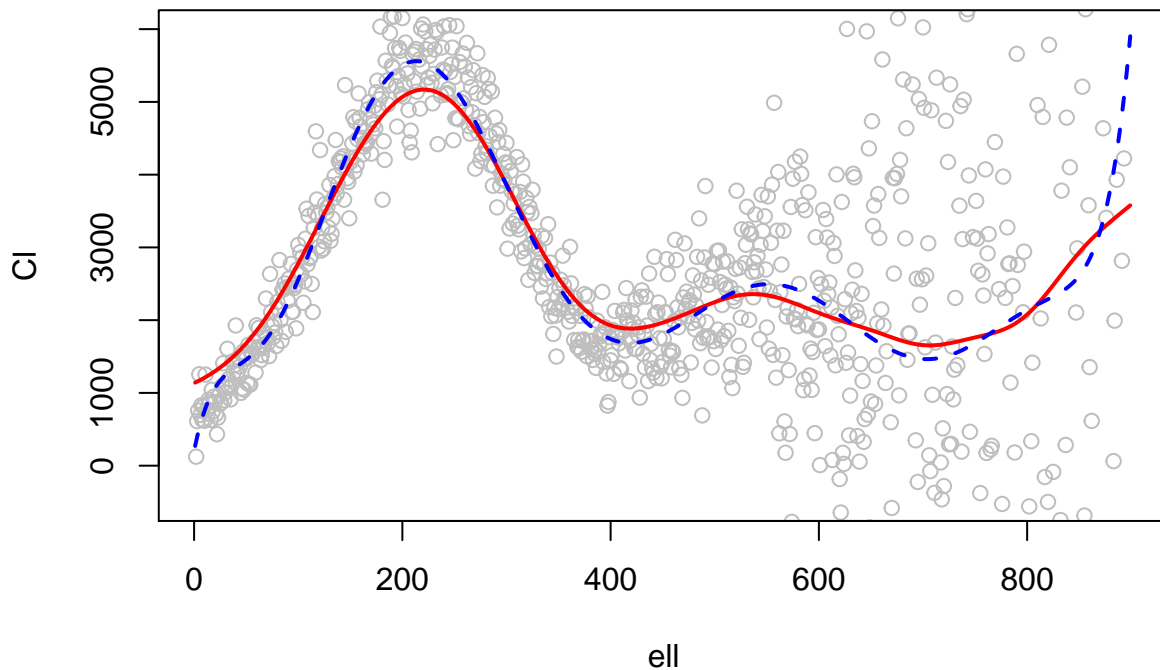
```
p <- lm(Cl~poly(ell,9,raw=T))
```

Neparametriskā regresija ar NW metodi bez papildus korekcijām (pirmā funkcija nosaka joslas platumu, otrā ģenerē pašu regresijas likni):

```
library(np)
bw <- npregbw(Cl~ell,bwmethod="cv.ls",regtype="lc")
n <- npreg(bw,residuals=T)
```

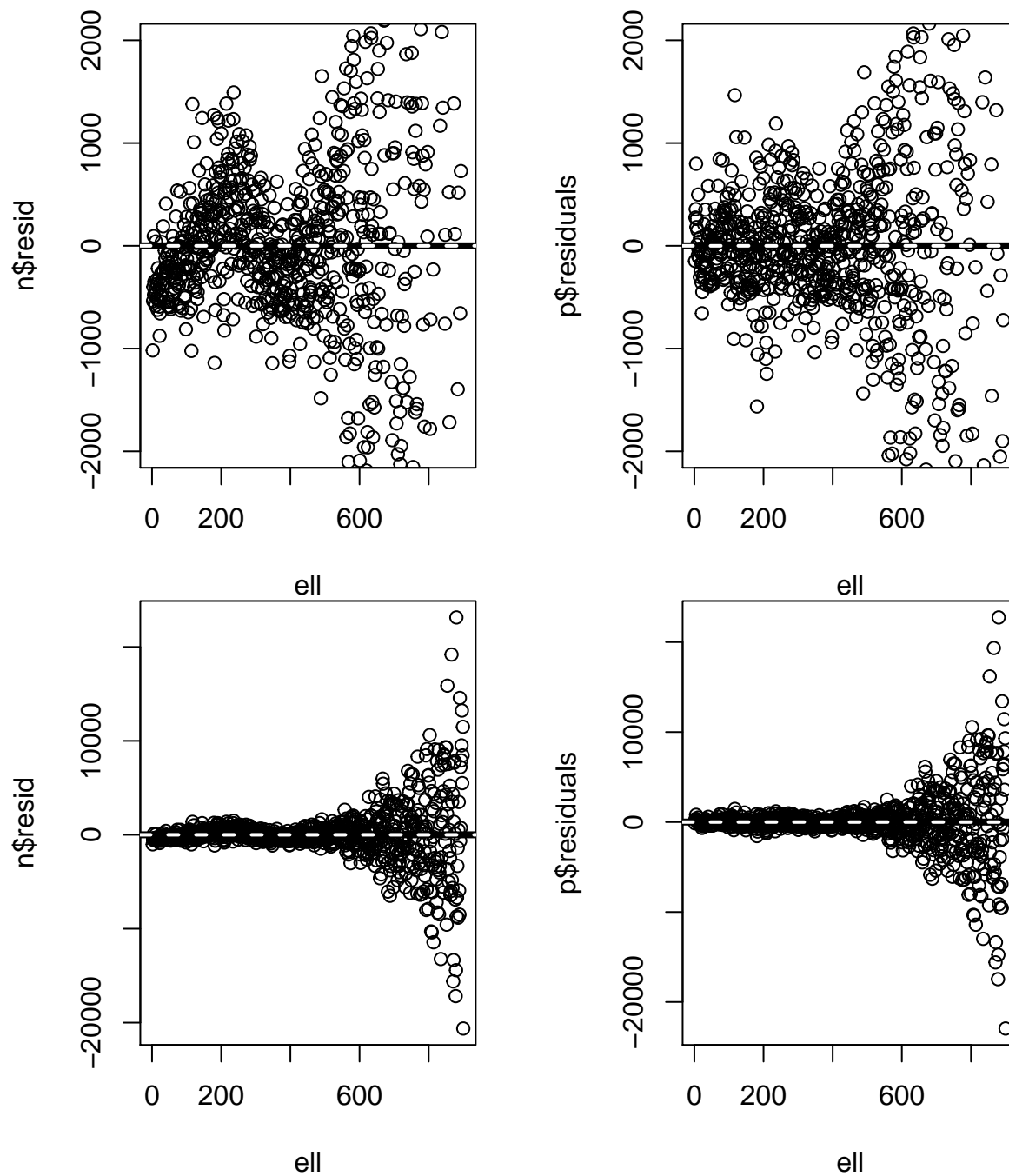
Grafisks attēlojums:

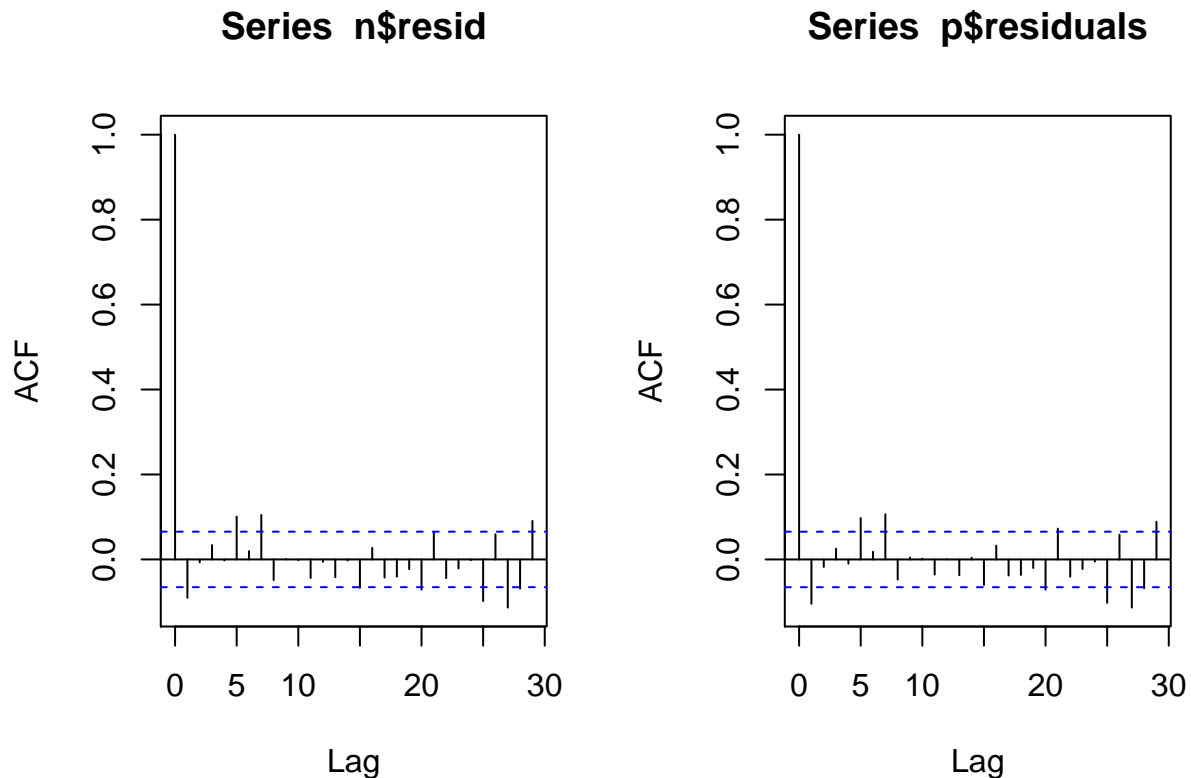
```
plot(ell,Cl,col="grey",ylim=c(-500,6000))
lines(n$mean,col="red",lwd=2)
lines(p$fitted.values,col="blue",lwd=2,lty="dashed")
```



6.2. nosacījumu pārbaude, salīdzinājums

Grafiski attēlojumi:





Grafiski redzams, ka definīcijas intervālā polinoms labāk tuvināts datiem, taču asi maina virzienu abos galos, kas to padara nederīgu paredzējumiem ārpus intervāla. Autokorelācijas abos gadījumos ir diezgan līdzīgas, taču neparametriskās regresijas gadījumā grafiski redzams, ka saglabājusies sistemātiska, periodiska nobīde starp modeli un datiem. Polinoma gadījumā atlikumi veido (vismaz vizuāli) nejauši izkļiedētu punktu mākonī.

Jau pirmajā uzdevumā noskaidrots, kas neizpildās ne normalitātes, ne dispersijas vienmērīguma nosacījumi atlikumos, kas nepieciešami lineārām un polinomiālām regresijām.

6.3. joslas platuma novērtējumi

NW nav vienīgā metode, ar kuru var ģenerēt neparametriskas regresijas. Lai iegūtu joslas platumus, var izmantot t.s. “plug-in” metodes, kas izmanto vienkāršotas heuristikas. Kodola regresijas jēdzienu iespējams vispārināt, padarot izteiksmi par polinomu katrā punktā - iegūstot lokāli lineāru vai polinomiālu modeli.

```
bwl1 <- npregbw(C1~ell,regtype="ll",bwmethod="cv.aic")
# would be dpill(ell,C1), but wilcox test crashes the compiler.
bwrt <- 34.61873
bwrt <-npregbw(C1~ell,bws=bwrt,bandwidth.compute=F)
```

```
bwl1

##
## Regression Data (899 observations, 1 variable(s)):
##
##              ell
## Bandwidth(s): 41.83265
##
## Regression Type: Local-Linear
## Bandwidth Selection Method: Expected Kullback-Leibler Cross-Validation
## Formula: C1 ~ ell
## Bandwidth Type: Fixed
```

```
## Objective Function Value: 17.07873 (achieved on multistart 1)
##
## Continuous Kernel Type: Second-Order Gaussian
## No. Continuous Explanatory Vars.: 1
```

```
bwrt
```

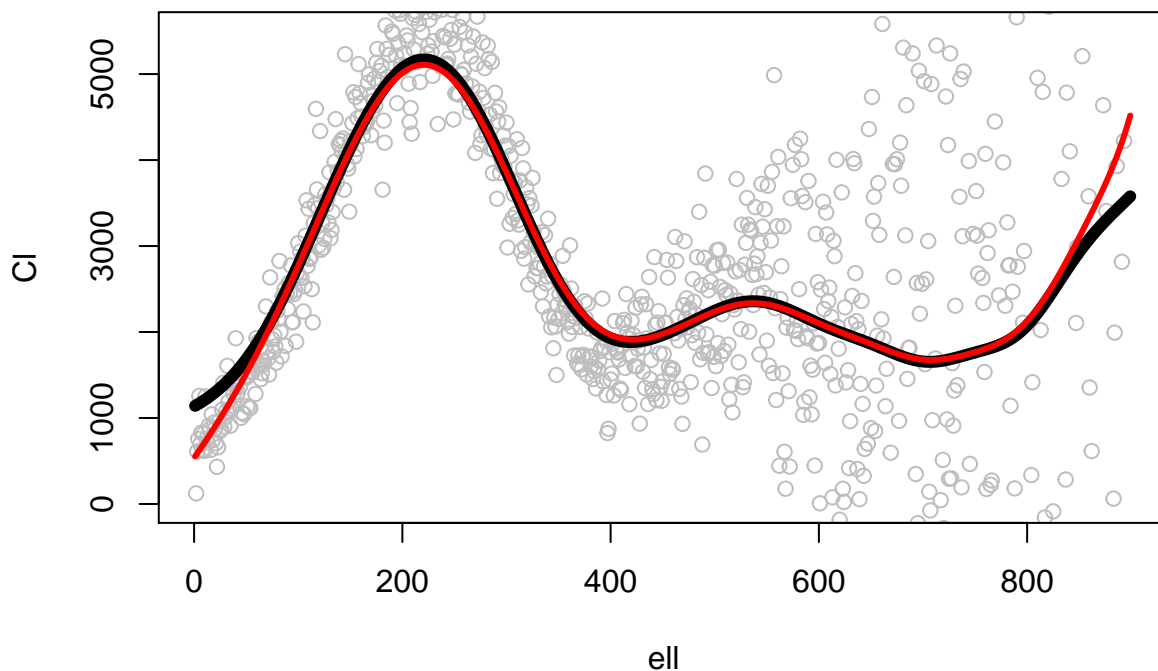
```
##
## Regression Data (899 observations, 1 variable(s)):
##
##           ell
## Bandwidth(s): 34.61873
##
## Regression Type: Local-Constant
## Bandwidth Selection Method: Manual
## Formula: Cl ~ ell
## Bandwidth Type: Fixed
##
## Continuous Kernel Type: Second-Order Gaussian
## No. Continuous Explanatory Vars.: 1
```

Salīdzinot regresijas metodes grafiski:

```
nll <- npreg(bwll,residuals=T)
nrt <-npreg(bwrt,residuals=T)
```

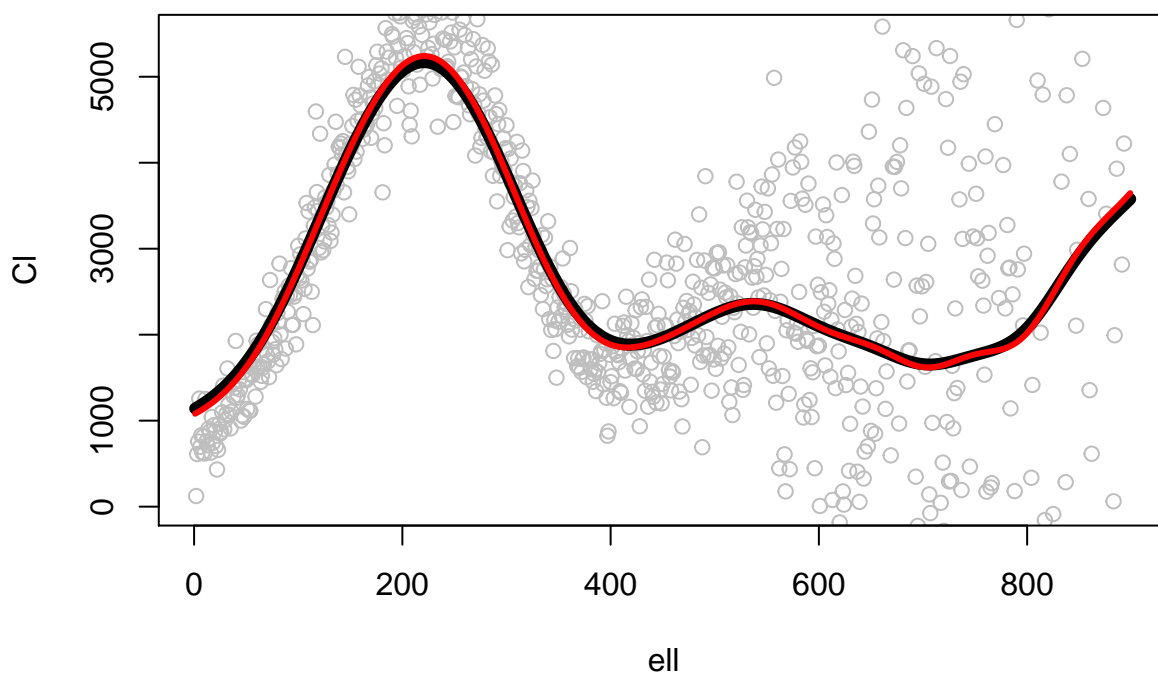
NW pret lokāli polinomiālo regresiju (metode ir ekvivalenta locpoly funkcijai, bet bez patvaļīgās argumentu kopas pārkārtošanas):

```
plot(ell,Cl,col="grey",ylim=c(0,5500))
lines(n$mean,col="black",lwd=6)
lines(nll$mean,col="red",lwd=3)
```



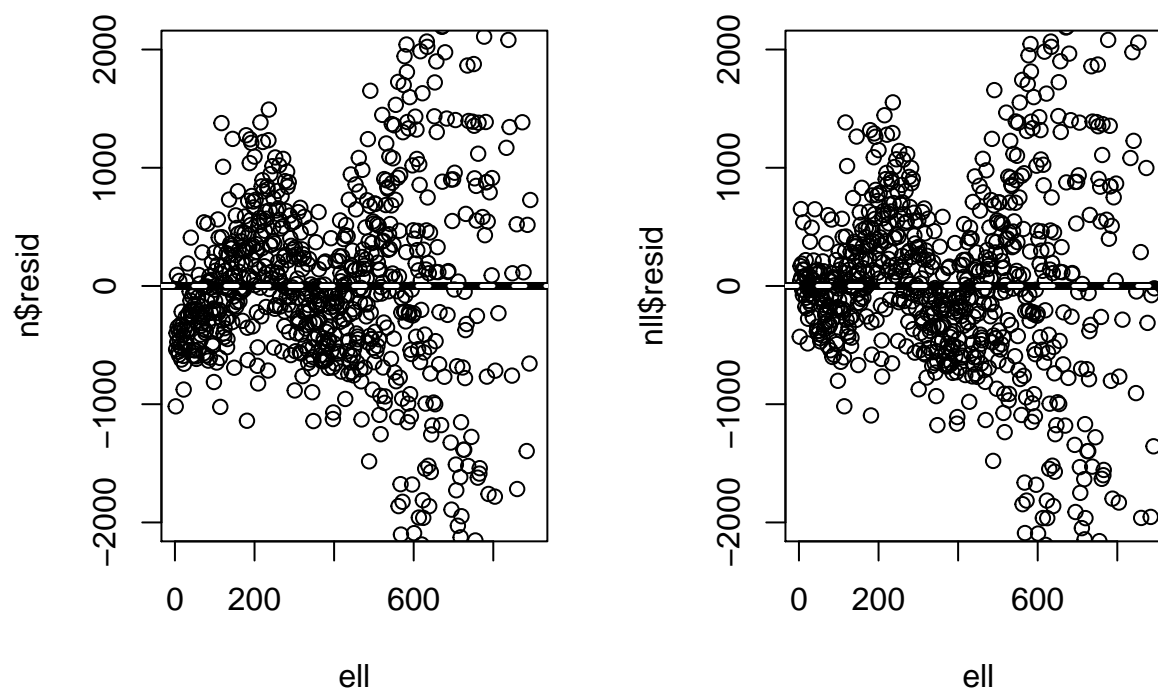
Krosvalidācija pret “plug-in” kodola aplēšanu:

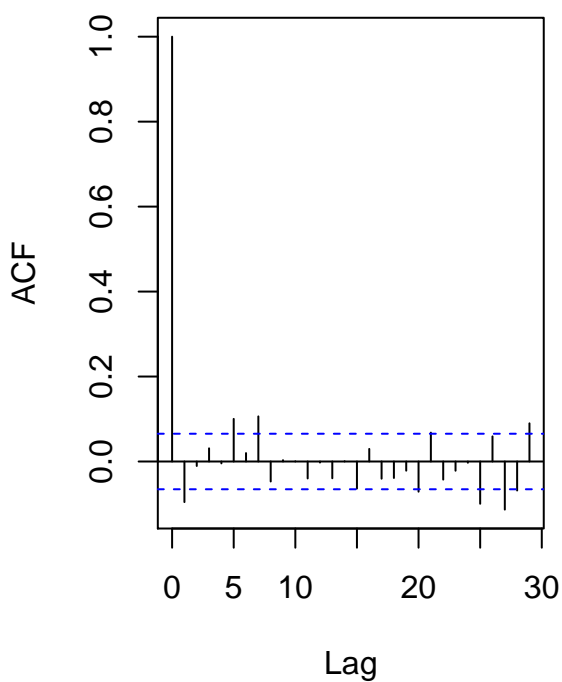
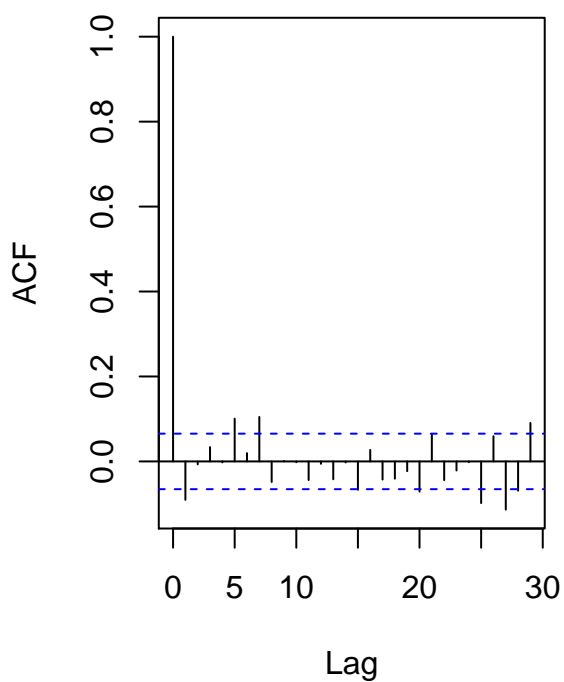
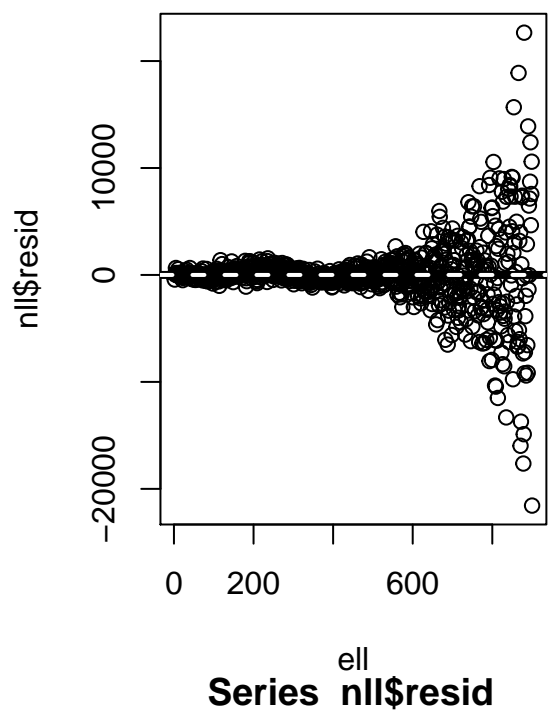
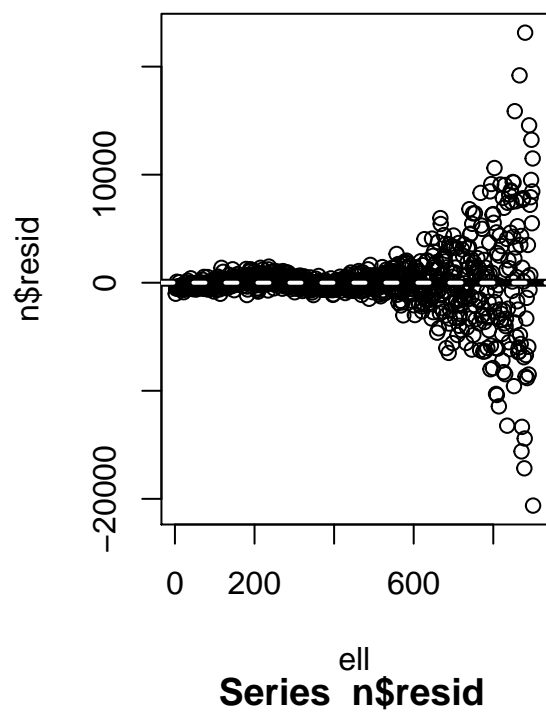
```
plot(ell,Cl,col="grey",ylim=c(0,5500))
lines(n$mean,col="black",lwd=6)
lines(nrt$mean,col="red",lwd=3)
```



Redzams, kas joslas platuma izvēles metodei ir minimāls iespaids uz rezultējošo līkni šajā gadījumā, taču lokāli polinomiālā regresija datu kopas galos uzvedas ievērojami savādāk nekā parastā NW metode.

Salīdzinot atlikumus:





Redzams, ka mazliet mazinājusies pārpalikusi struktūra atlikumos, taču tuvinājums vēl joprojām nav tik labs kā polinoma gadījumā.