# MD2_Racinskis

Pēteris Račinskis pr20015

5/18/2021

## 2. mājas darbs Mate6029

### 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.

```
summary(df)
```

```
##       ell              Cl               se            measerr
##  Min.   :  2.0   Min.   :-17046   Min.   : 215.2   Min.   :   3.381
##  1st Qu.:226.5   1st Qu.:  1452   1st Qu.: 332.7   1st Qu.:  37.437
##  Median :451.0   Median :  2449   Median : 456.8   Median : 325.755
##  Mean   :451.0   Mean   :  2688   Mean   :1679.8   Mean   :1499.883
##  3rd Qu.:675.5   3rd Qu.:  4195   3rd Qu.:2034.4   3rd Qu.:1959.961
##  Max.   :900.0   Max.   : 26481   Max.   :9726.3   Max.   :9657.755
##      cosmic
##  Min.   : 68.57
##  1st Qu.: 91.84
##  Median :117.10
##  Mean   :179.89
##  3rd Qu.:258.56
##  Max.   :759.26
```

#### 1.1. Regresijas modeļu ģenerēšana

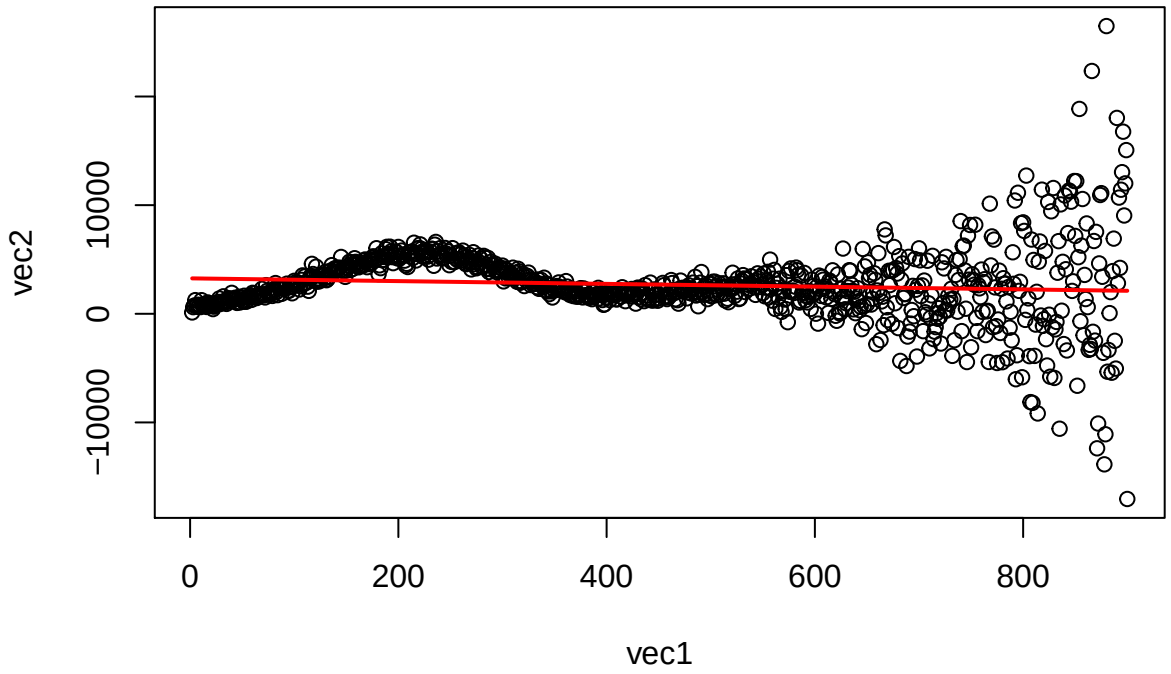
Vispārināma lineārās regresijas funkcija grafiku zīmēšanai, rezultātu izvadei uz konsoles un darbam ar augstākas pakāpes modeļiem:

```
general_lreg <- function(vec1,vec2,degree=1,plot=F,print=F) {
  fit<-lm(vec2~poly(vec1,degree,raw=T))
  if(plot){
    plot(vec1,vec2)
    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(summary(fit))
  }
  fit
}
```

Vienkārša lineārā regresija parametriem 'ell' un 'Cl':

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




```
##
## Call:
## lm(formula = vec2 ~ poly(vec1, degree, raw = T))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19163.1  -1330.3   -265.5   1405.5  24338.5
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 3261.1891   220.7702  14.772  < 2e-16 ***
## poly(vec1, degree, raw = T)   -1.2714     0.4243  -2.996  0.00281 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3301 on 897 degrees of freedom
## Multiple R-squared:  0.009911,   Adjusted R-squared:  0.008807
## F-statistic: 8.979 on 1 and 897 DF,  p-value: 0.002806
```

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:

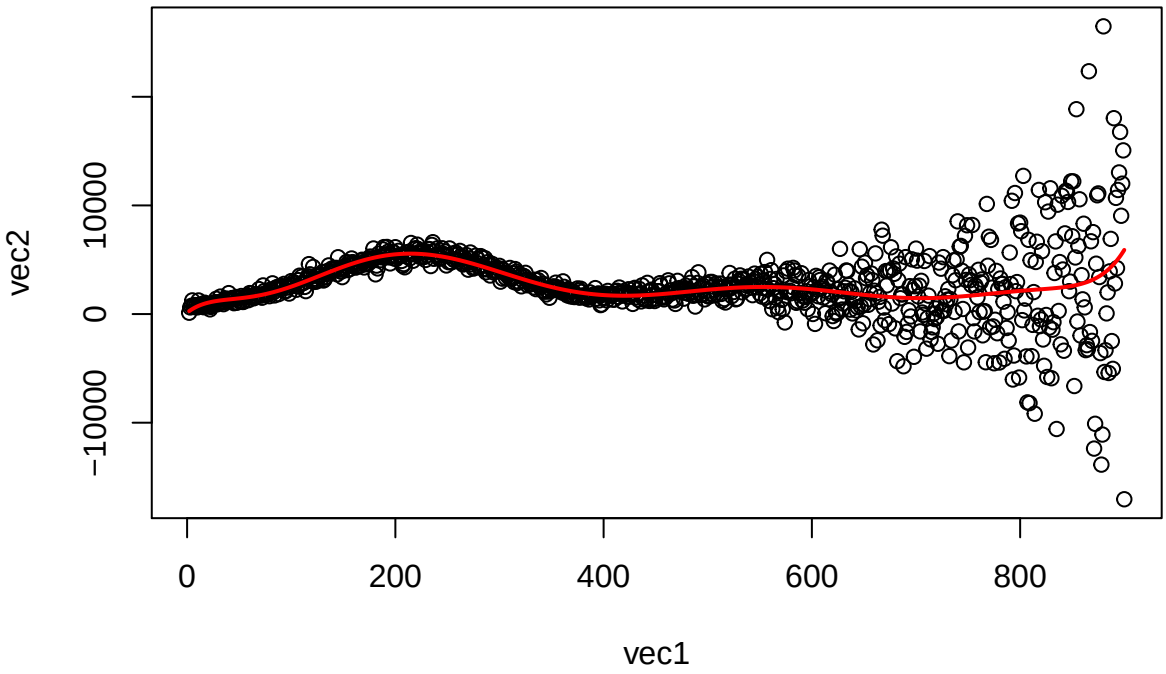
```
res <- bestfit(ell,Cl)
paste("stopped at x^", res$d,sep="")
```

```
## [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)
```



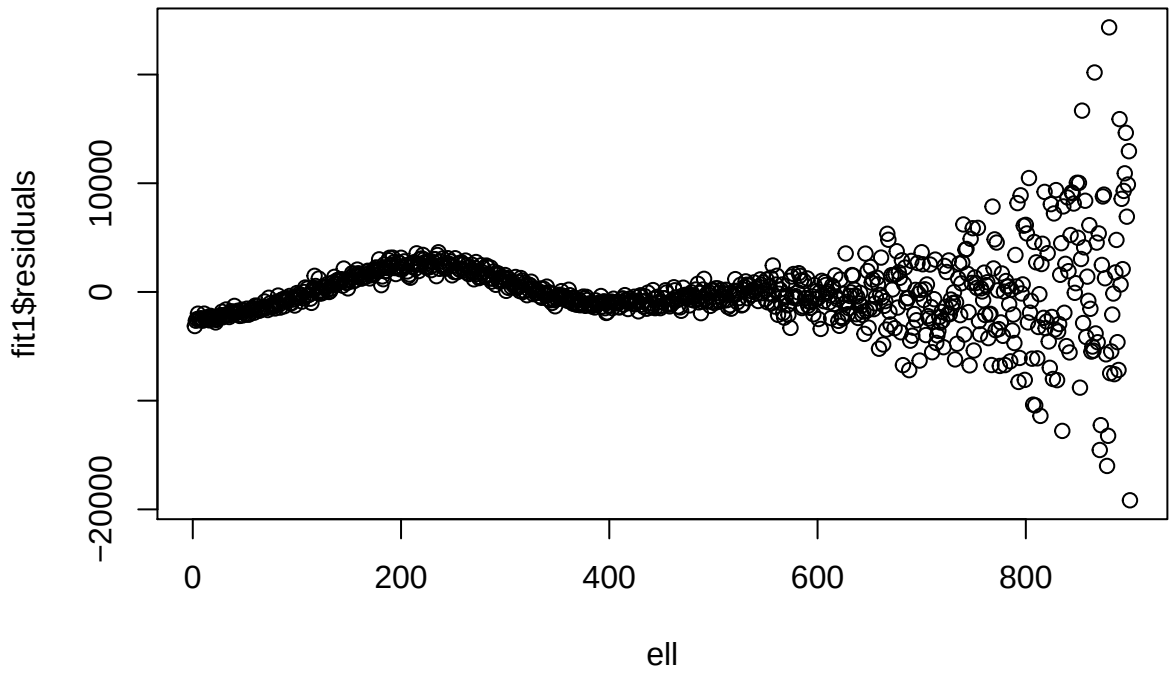

```
##
## Call:
## lm(formula = vec2 ~ poly(vec1, degree, raw = T))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22948.7   -454.9    -26.9    499.6  22741.0
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                   1.283e+02  1.113e+03   0.115  0.90826
## poly(vec1, degree, raw = T)1  7.506e+01  6.918e+01   1.085  0.27824
## poly(vec1, degree, raw = T)2 -1.927e+00  1.405e+00  -1.371  0.17080
## poly(vec1, degree, raw = T)3  2.600e-02  1.321e-02   1.969  0.04929 *
## poly(vec1, degree, raw = T)4 -1.642e-04  6.748e-05  -2.433  0.01518 *
## poly(vec1, degree, raw = T)5  5.515e-07  2.024e-07   2.725  0.00656 **
## poly(vec1, degree, raw = T)6 -1.058e-09  3.664e-10  -2.887  0.00398 **
## poly(vec1, degree, raw = T)7  1.165e-12  3.932e-13   2.964  0.00312 **
## poly(vec1, degree, raw = T)8 -6.874e-16  2.302e-16  -2.986  0.00291 **
## poly(vec1, degree, raw = T)9  1.685e-19  5.664e-20   2.975  0.00301 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3073 on 889 degrees of freedom
## Multiple R-squared:  0.1499, Adjusted R-squared:  0.1413
## F-statistic: 17.42 on 9 and 889 DF,  p-value: < 2.2e-16
```

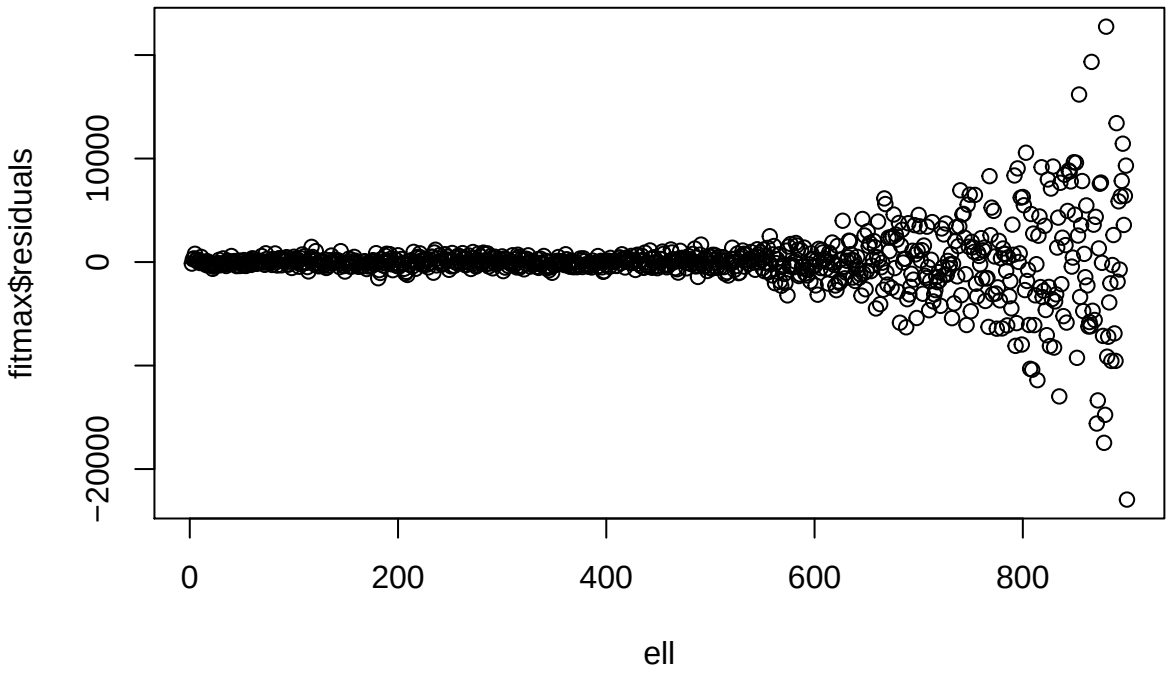
### 1.2. Diagnostika

Atlikumu neatkarība - grafiski:

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




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




Statistisko testu bibliotēkas:

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

Durbin-Watson tests autokorelācijai:

```
durbinWatsonTest(fit1)
```

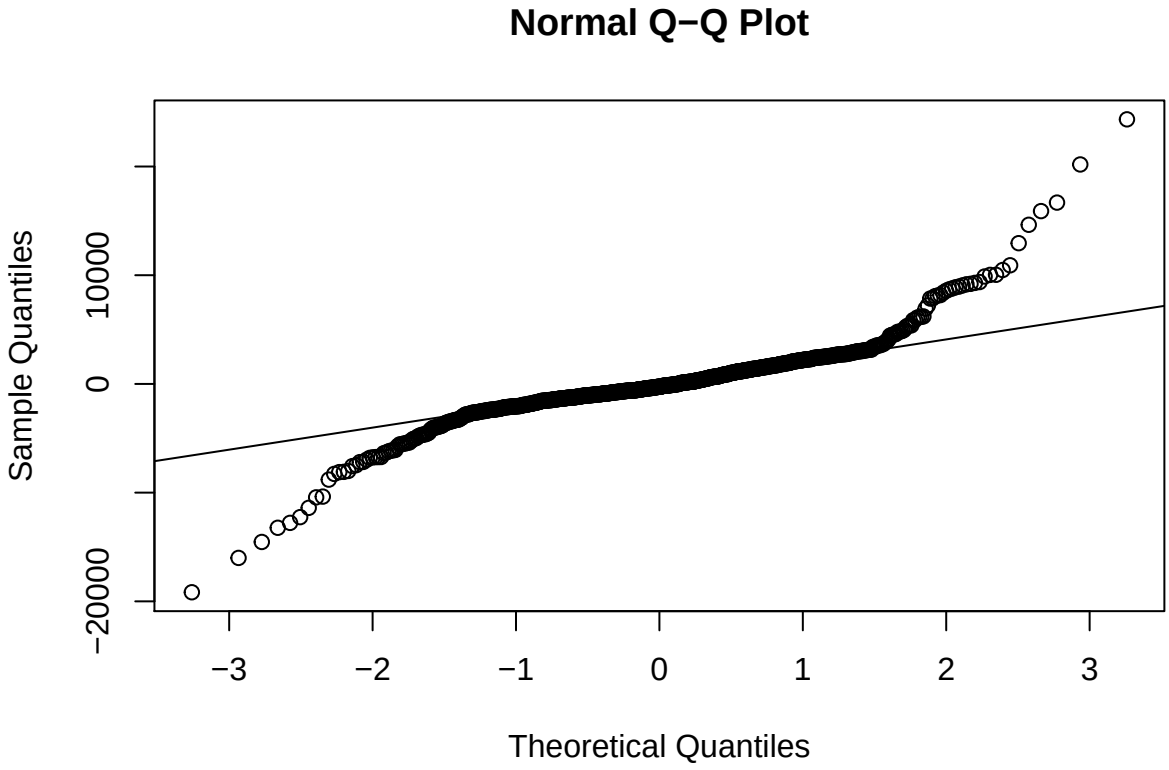
```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.05972642      1.841981   0.022
##  Alternative hypothesis: rho != 0
```

```
durbinWatsonTest(fitmax)
```

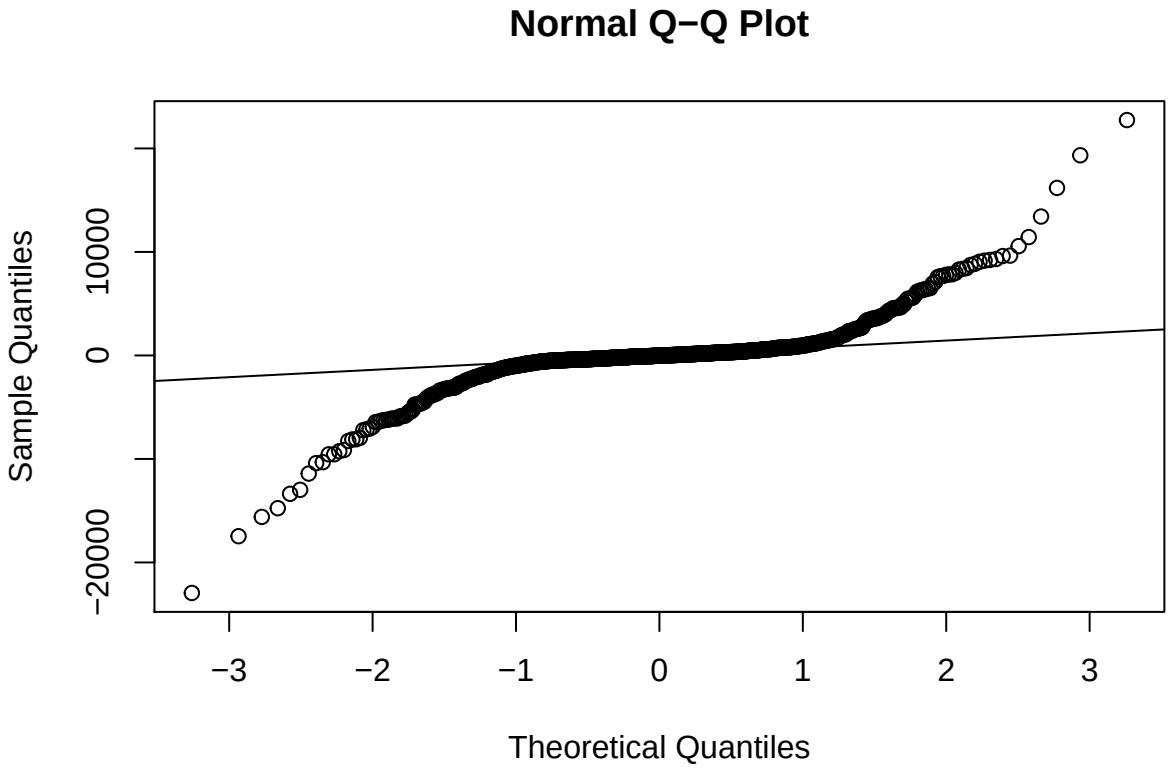
```
##  lag Autocorrelation D-W Statistic p-value
##    1      -0.1044366      2.146133   0.046
##  Alternative hypothesis: rho != 0
```

Normalitātes testi - grafiski:

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




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




Normalitātes testi - Kolmogorova-Smirnova tests:

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

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

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

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

Dispersijas vienmērīguma testi:

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

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

```r
# 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.