

EX5

EX5 - diamonds

Vyberte si vhodná data (například Animals2 v knihovně robustbase nebo cars v knihovně datasets) a pokuste se zvolit vysvětlující proměnné (případně jejich transformace) tak, aby byla závisle proměnné co nejlépe vysvětlená.

Nakreslete graf s původními daty včetně proložených hodnot (fitted values) z navrženého lineárního modelu.

Okomentujte statistickou významnost odhadnutých koeficientů. Otestujte nulovou hypotézu, že jsou všechny regresní koeficienty nulové (například pomocí funkce anova()).

Vypočítejte residua a na základě vhodných grafů okomentujte splnění předpokladů použitého modelu (heteroskedasticita, normalita, tvar regresního modelu, apod).

```
library(yarrrr)
```

```
## Loading required package: jpeg
```

```
## Loading required package: BayesFactor
```

```
## Loading required package: coda
```

```
## Loading required package: Matrix
```

```
## *****  
## Welcome to BayesFactor 0.9.12-4.3. If you have questions, please contact Richard  
## Morey (richarddmorey@gmail.com).  
##  
## Type BFManual() to open the manual.  
## *****
```

```
## Loading required package: circlize
```

```
## Warning: package 'circlize' was built under R version 4.0.5
```

```
## =====
## circlize version 0.4.14
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: https://jokergoo.github.io/circlize_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
##   in R. Bioinformatics 2014.
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(circlize))
## =====
```

```
## yarrv v0.1.5. Citation info at citation('yarrv'). Package guide at yarrv.guide()
```

```
## Email me at Nathaniel.D.Phillips.is@gmail.com
```

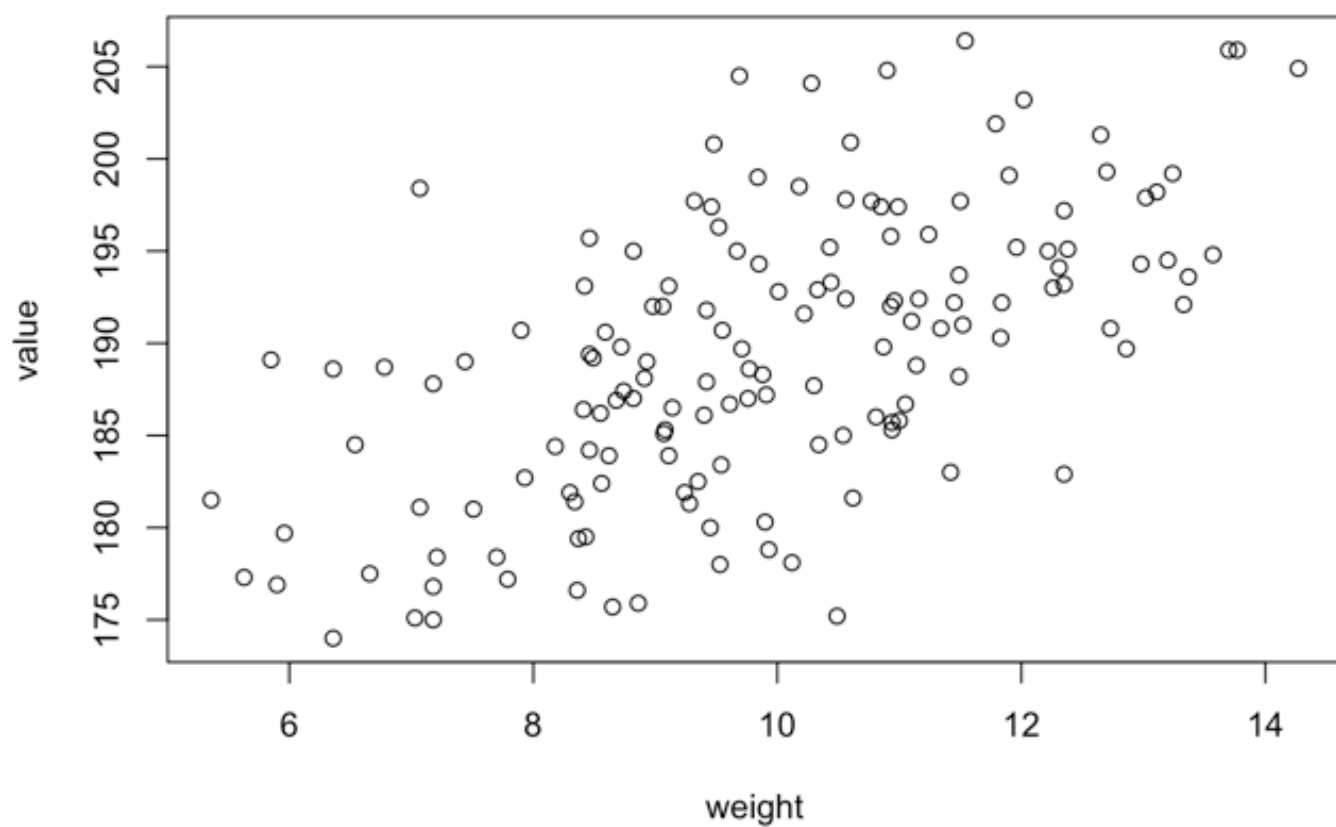
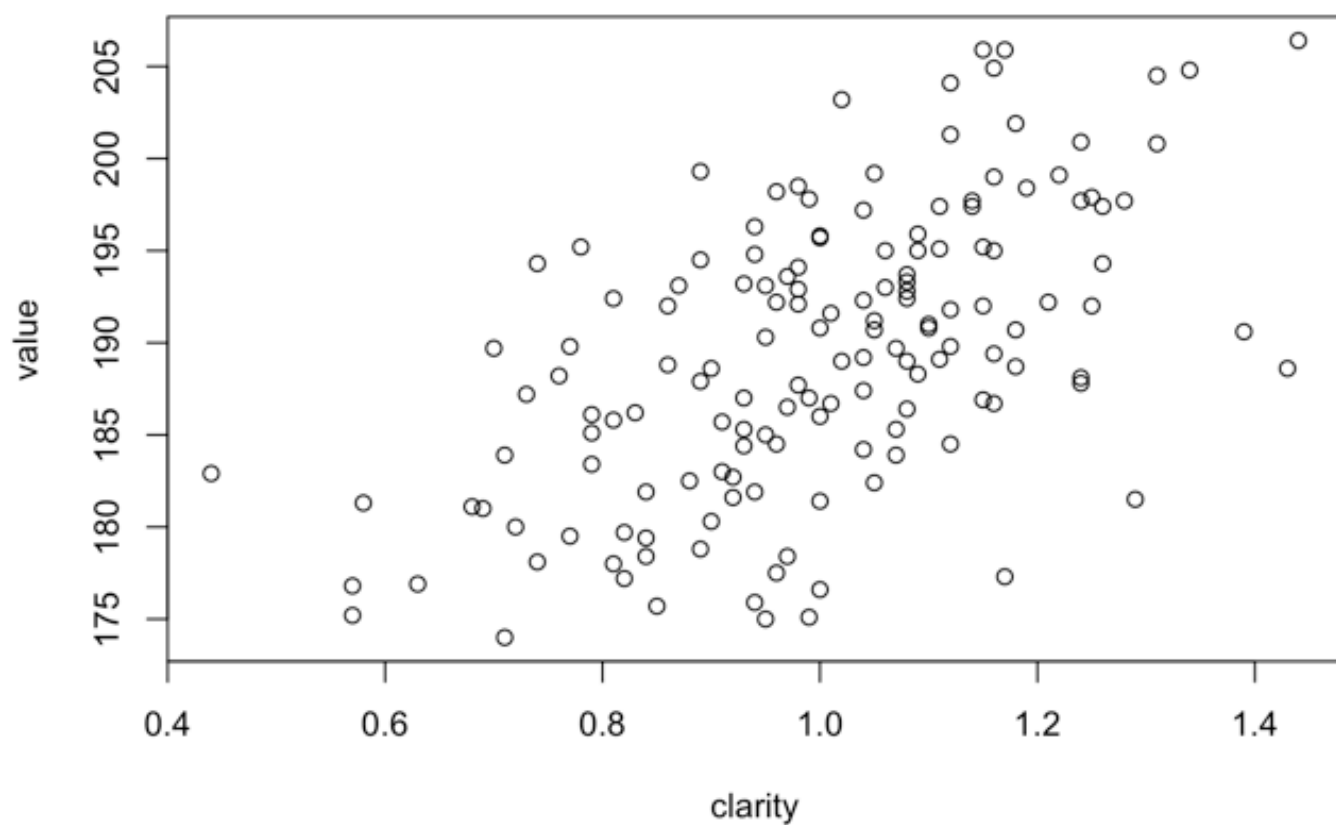
```
head(diamonds)
```

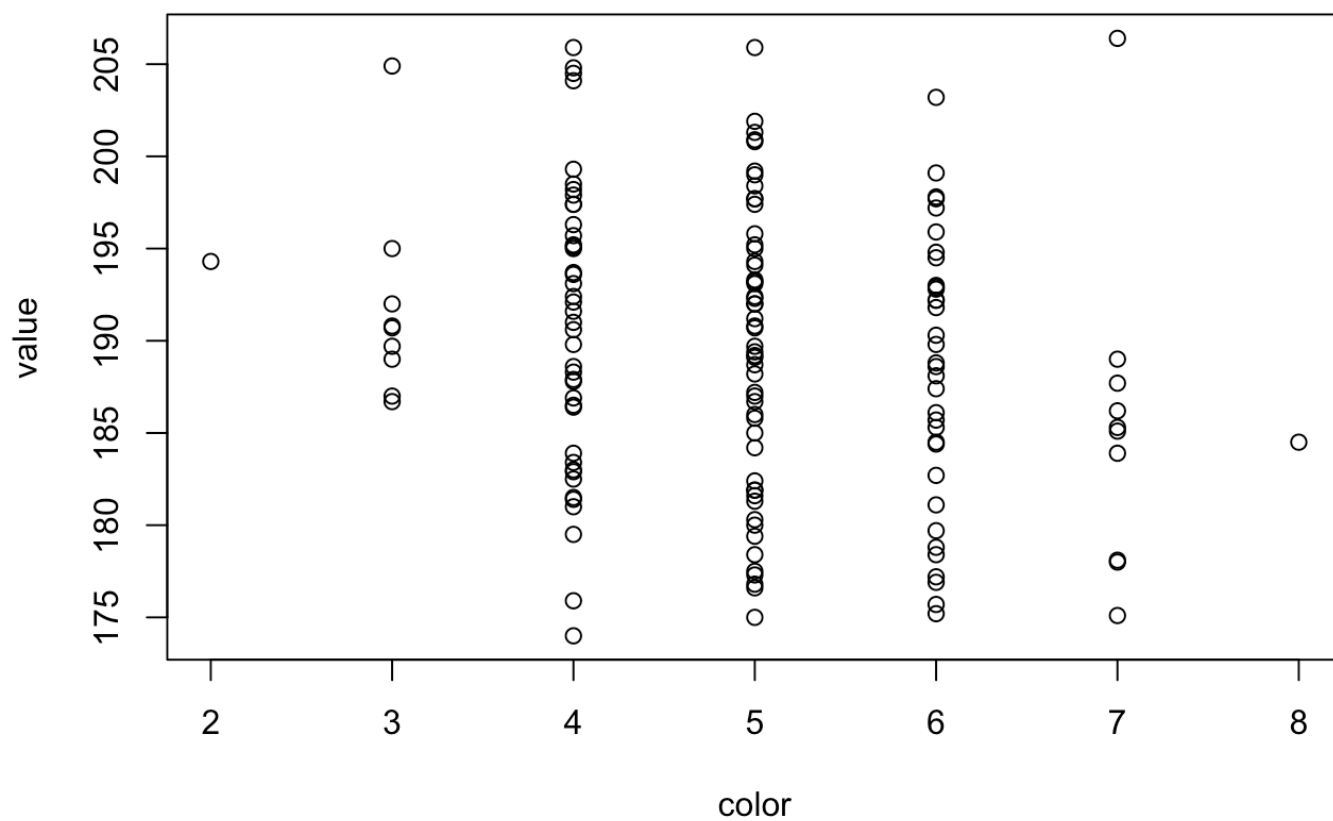
```
##   weight clarity color value
## 1    9.35    0.88     4 182.5
## 2   11.10    1.05     5 191.2
## 3    8.65    0.85     6 175.7
## 4   10.43    1.15     5 195.2
## 5   10.62    0.92     5 181.6
## 6   12.35    0.44     4 182.9
```

```
summary(diamonds)
```

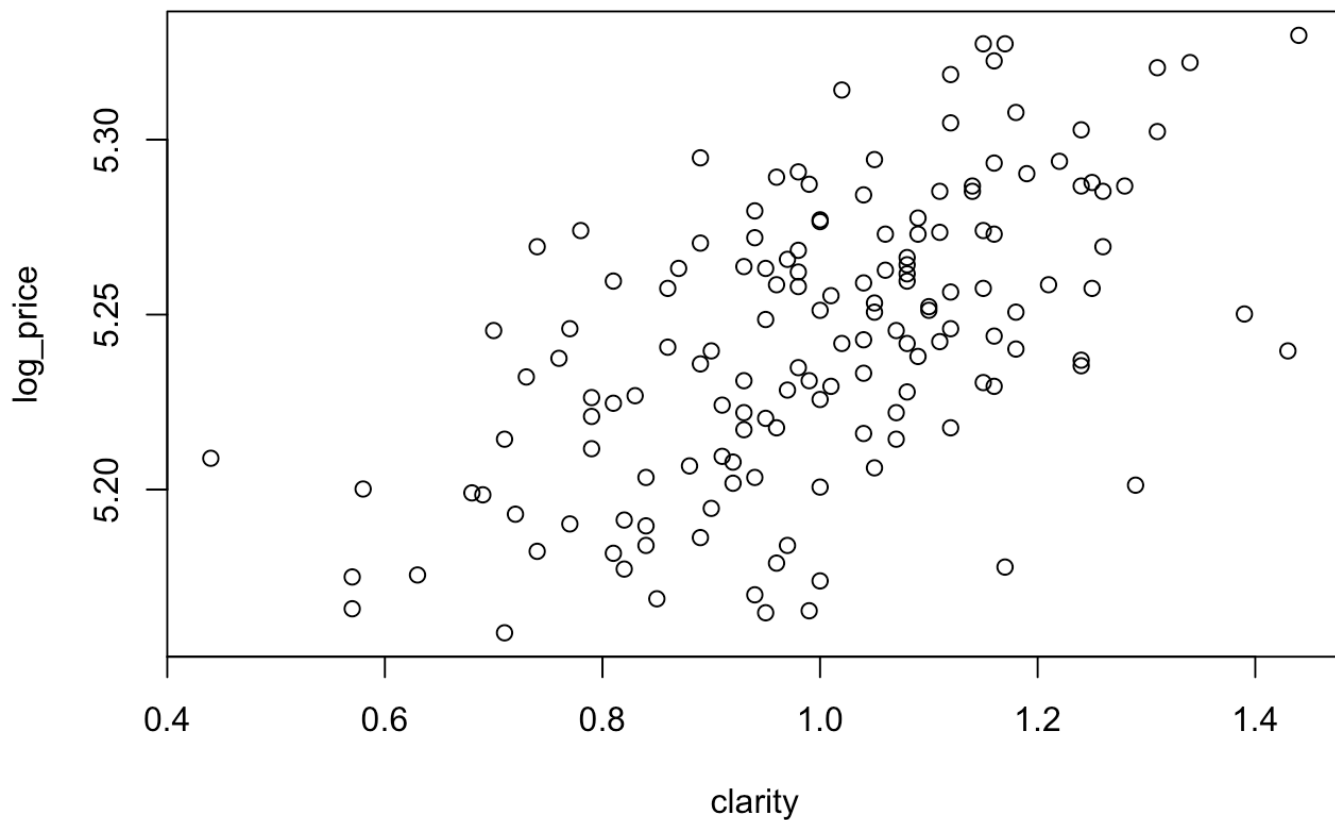
```
##      weight      clarity      color      value
## Min.   : 5.360   Min.   :0.4400   Min.   :2.00   Min.   :174.0
## 1st Qu.: 8.598   1st Qu.:0.8900   1st Qu.:4.00   1st Qu.:184.0
## Median : 9.805   Median :1.0000   Median :5.00   Median :189.6
## Mean    : 9.901   Mean    :0.9996   Mean    :4.96   Mean    :189.4
## 3rd Qu.:11.155   3rd Qu.:1.1200   3rd Qu.:6.00   3rd Qu.:194.9
## Max.    :14.270   Max.    :1.4400   Max.    :8.00   Max.    :206.4
```

```
plot(value ~ clarity + weight + color, data = diamonds)
```

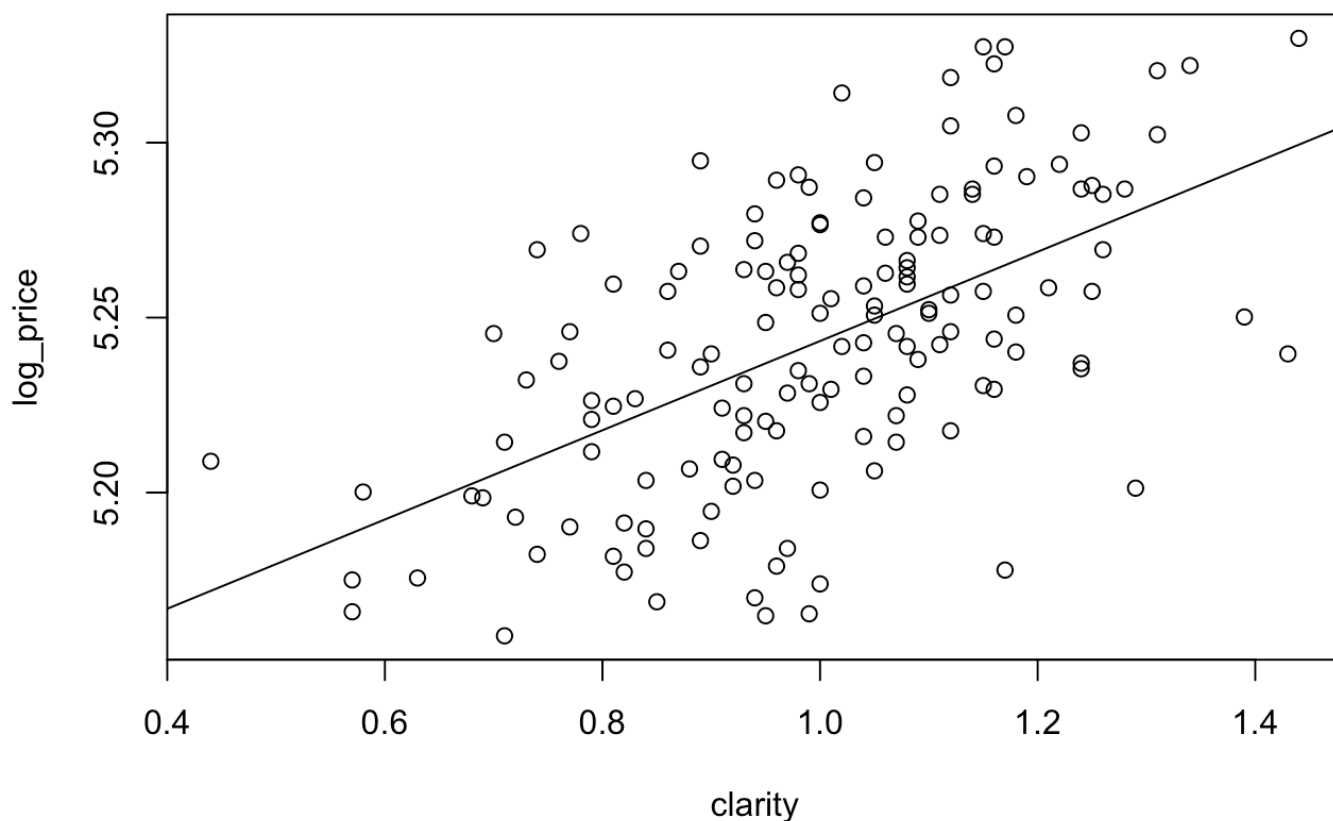


```
#logarimus ceny:  
log_price = log(diamonds$value)  
plot(log_price ~ clarity, data = diamonds)
```



Jednoduchý lineární model

```
par(mfrow=c(1,1))  
lm.2=lm(log_price~clarity,data=diamonds)  
plot(log_price~clarity,data=diamonds)  
abline(lm.2)
```



```
diamonds.lm <- lm(formula = value ~ weight + clarity , data = diamonds)
summary(diamonds.lm)
```

```
##
## Call:
## lm(formula = value ~ weight + clarity, data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.034  -3.802  -0.196   3.207  11.166
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   145.446     2.795   52.04  <2e-16 ***
## weight         2.219     0.199   11.15  <2e-16 ***
## clarity       22.036     2.129   10.35  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.681 on 147 degrees of freedom
## Multiple R-squared:  0.6334, Adjusted R-squared:  0.6284
## F-statistic: 127 on 2 and 147 DF, p-value: < 2.2e-16
```

```
names(diamonds.lm)
```

```
## [1] "coefficients" "residuals" "effects" "rank"
## [5] "fitted.values" "assign" "qr" "df.residual"
## [9] "xlevels" "call" "terms" "model"
```

```
diamonds.lm$coefficients
```

```
## (Intercept) weight clarity
## 145.446330 2.218574 22.036231
```

```
summary(diamonds.lm)$coefficients
```

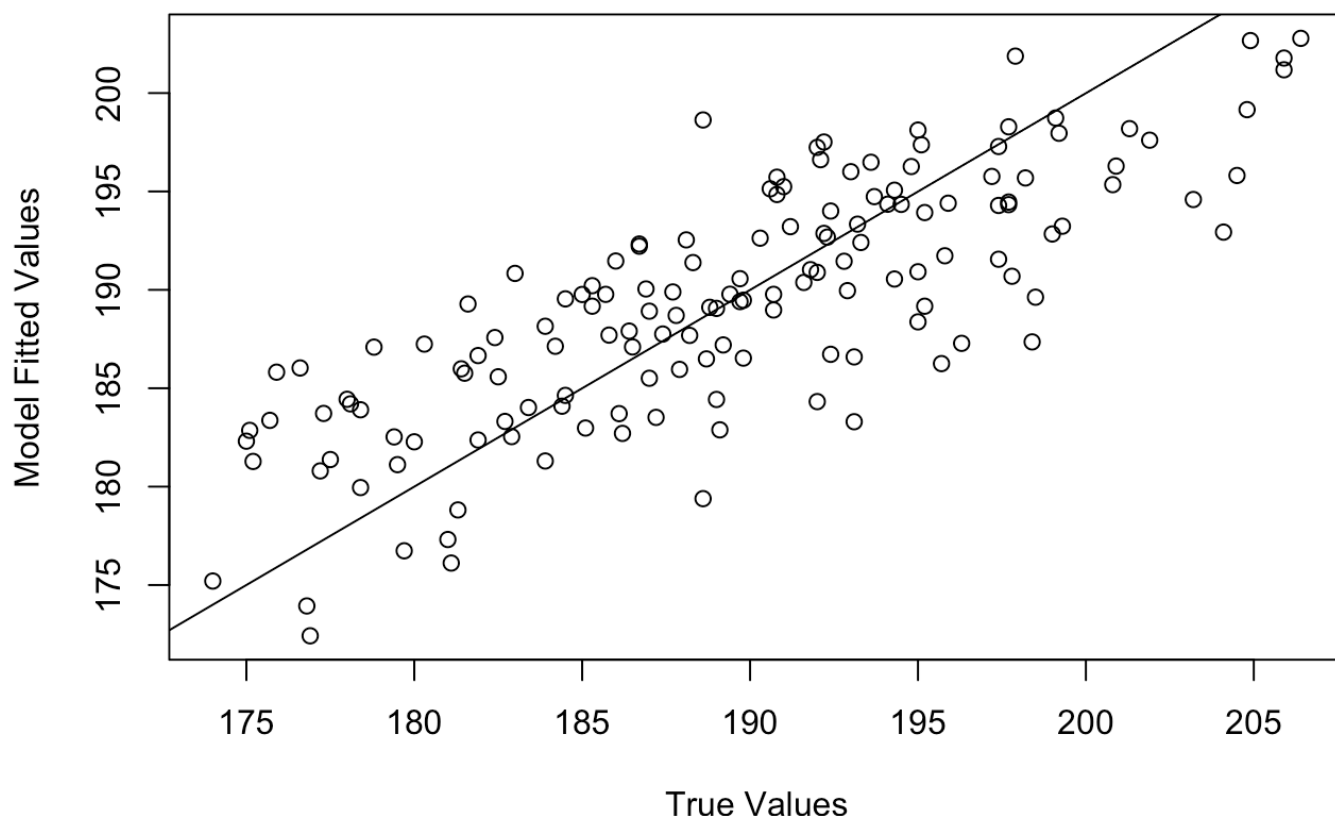
```
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 145.446330 2.794869 52.04049 1.371703e-96
## weight 2.218574 0.198992 11.14906 2.646844e-21
## clarity 22.036231 2.129080 10.35012 3.373243e-19
```

```
diamonds$value.lm <- diamonds.lm$fitted.values
head(diamonds)
```

```
## weight clarity color value value.lm
## 1 9.35 0.88 4 182.5 185.5819
## 2 11.10 1.05 5 191.2 193.2105
## 3 8.65 0.85 6 175.7 183.3678
## 4 10.43 1.15 5 195.2 193.9277
## 5 10.62 0.92 5 181.6 189.2809
## 6 12.35 0.44 4 182.9 182.5417
```

```
plot(x = diamonds$value, y = diamonds.lm$fitted.values, xlab = "True Values", ylab = "Model Fitted Values", main = "Regression fits of diamond values")
abline(b = 1, a = 0)
```


Regression fits of diamond values



```
diamonds.aov <- aov(diamonds.lm)
summary(diamonds.aov)
```

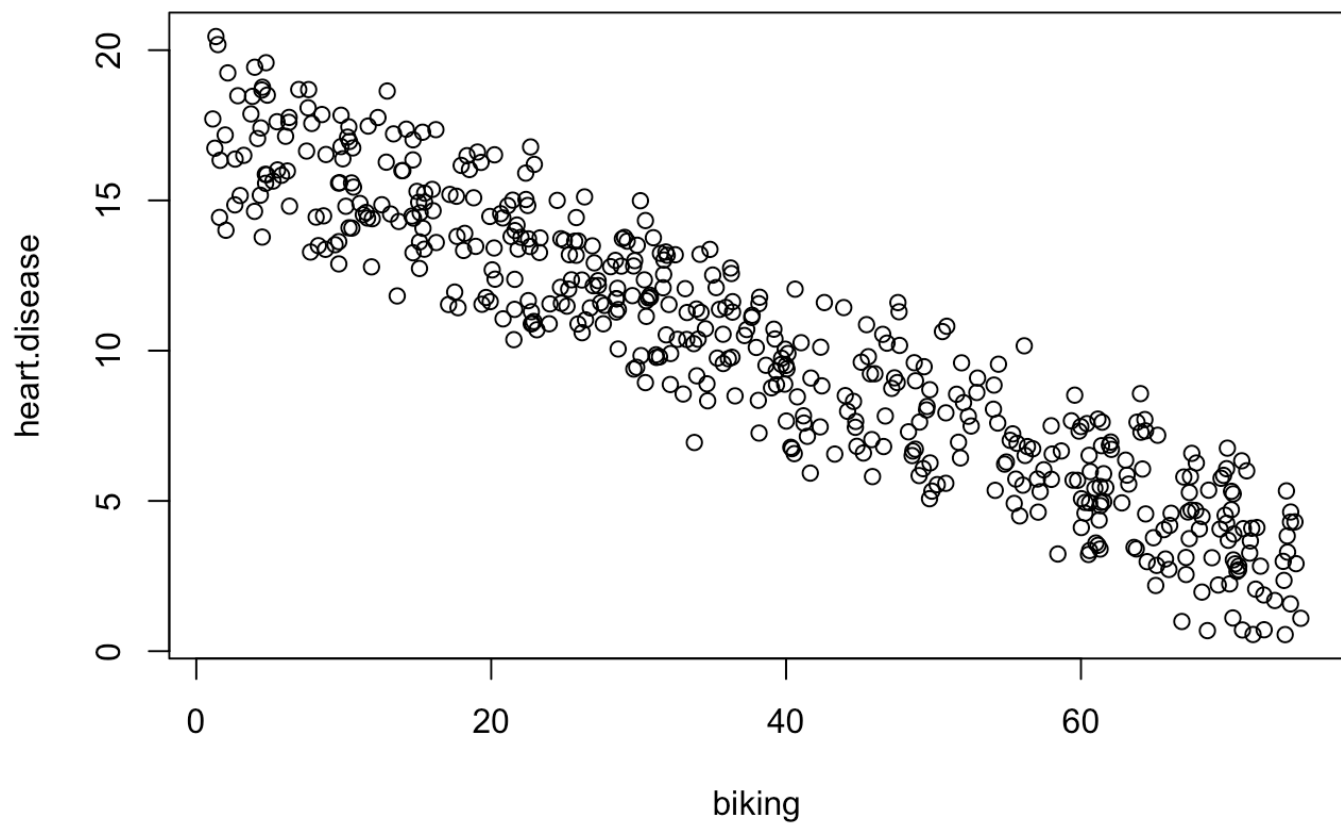
```
##              Df Sum Sq Mean Sq F value Pr(>F)
## weight         1    3218     3218   146.8 <2e-16 ***
## clarity         1    2347     2347   107.1 <2e-16 ***
## Residuals    147     3221         22
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

EX5 - heart disease

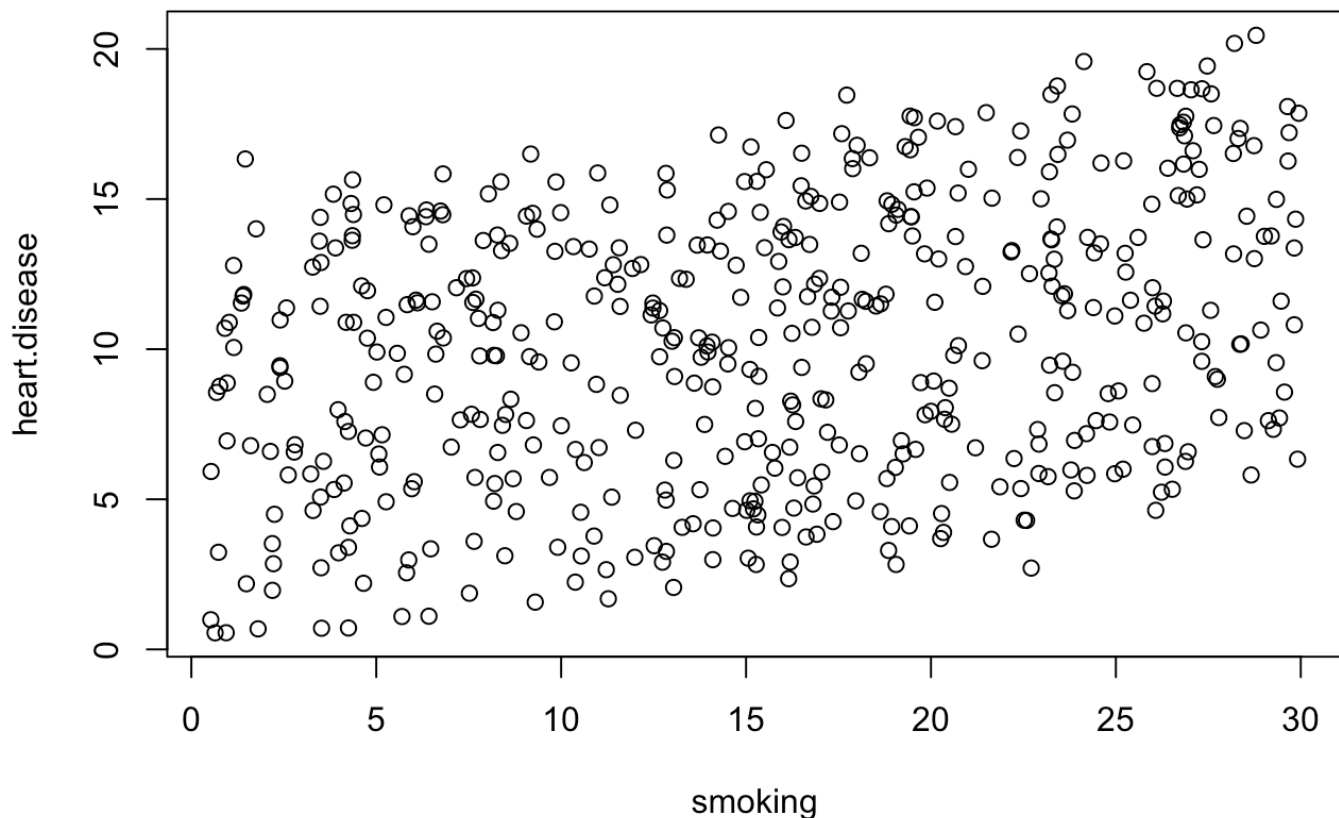
```
data_heart <- read.csv('h.csv')
summary(data_heart)
```

```
##           X           biking           smoking           heart.disease
## Min.      : 1.0      Min.      : 1.119      Min.      : 0.5259      Min.      : 0.5519
## 1st Qu.:125.2      1st Qu.:20.205      1st Qu.: 8.2798      1st Qu.: 6.5137
## Median :249.5      Median :35.824      Median :15.8146      Median :10.3853
## Mean     :249.5      Mean     :37.788      Mean     :15.4350      Mean     :10.1745
## 3rd Qu.:373.8      3rd Qu.:57.853      3rd Qu.:22.5689      3rd Qu.:13.7240
## Max.     :498.0      Max.     :74.907      Max.     :29.9467      Max.     :20.4535
```

```
plot(heart.disease ~ biking, data=data_heart)
```



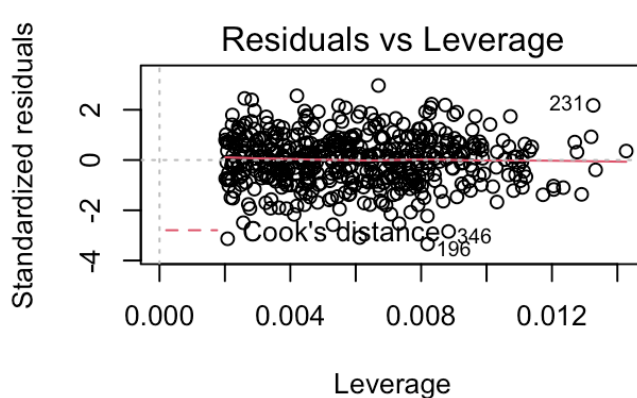
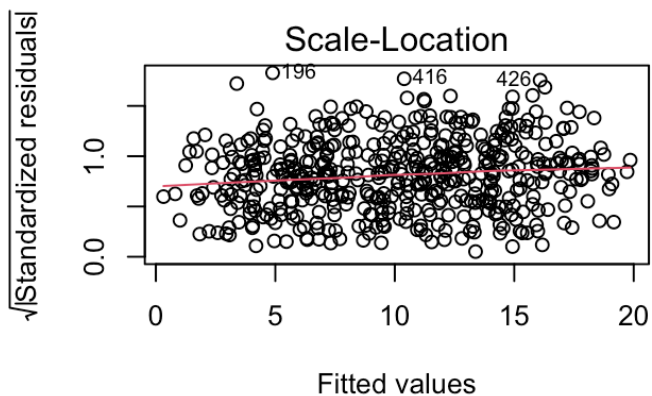
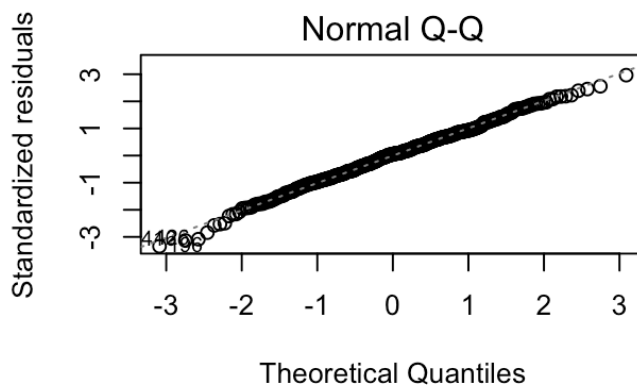
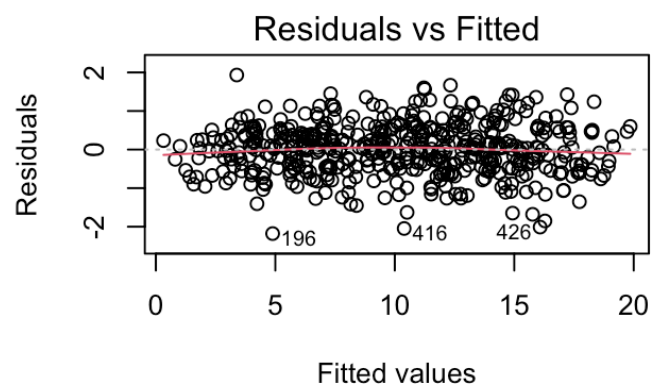
```
plot(heart.disease ~ smoking, data=data_heart)
```



```
heart.disease.lm<-lm(heart.disease ~ biking + smoking, data = data_heart)
summary(heart.disease.lm)
```

```
##
## Call:
## lm(formula = heart.disease ~ biking + smoking, data = data_heart)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.1789 -0.4463  0.0362  0.4422  1.9331
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.984658   0.080137  186.99  <2e-16 ***
## biking       -0.200133   0.001366 -146.53  <2e-16 ***
## smoking       0.178334   0.003539   50.39  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.654 on 495 degrees of freedom
## Multiple R-squared:  0.9796, Adjusted R-squared:  0.9795
## F-statistic: 1.19e+04 on 2 and 495 DF, p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(heart.disease.lm)
```



```
par(mfrow=c(1,1))
```

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

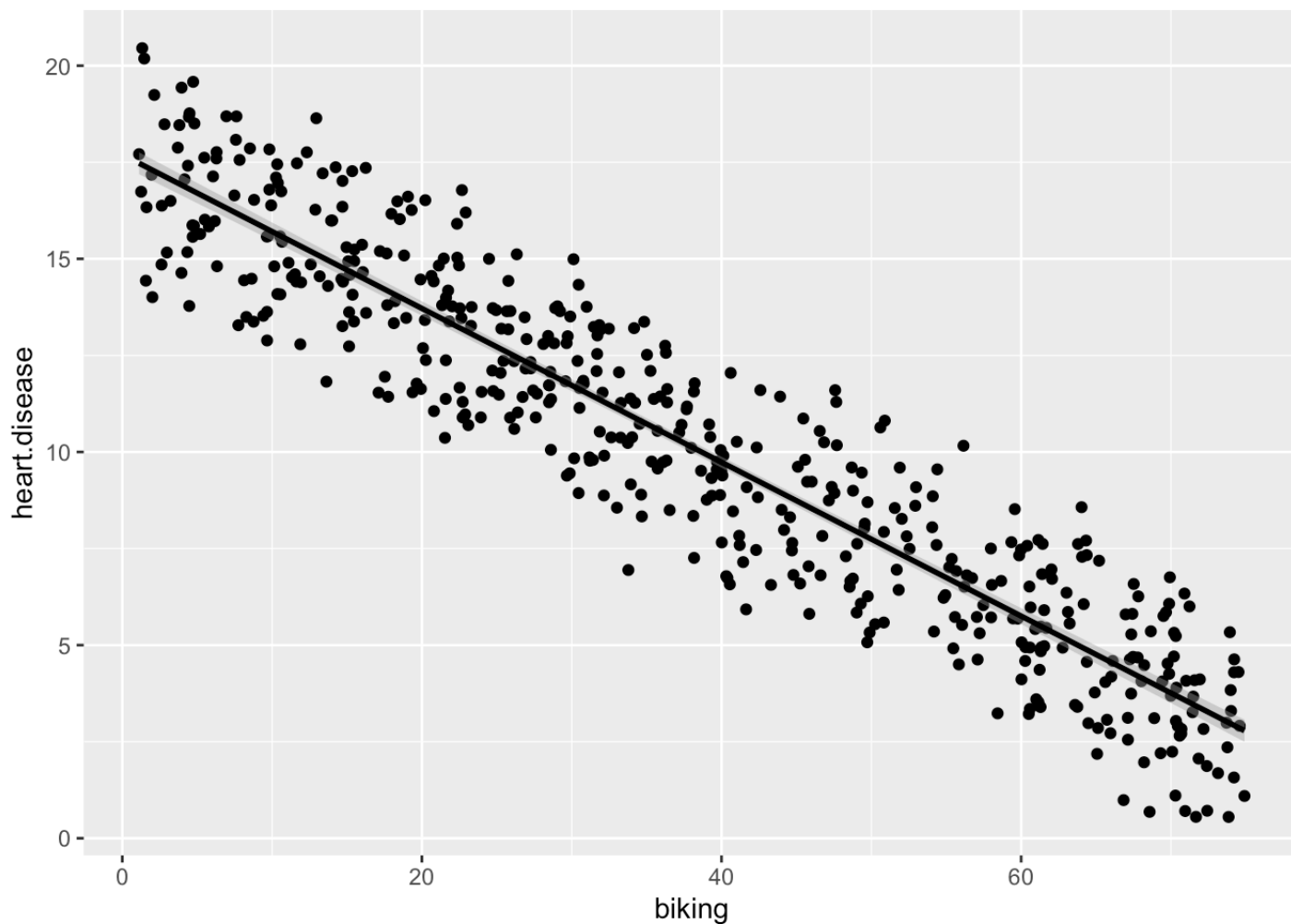
```
## The following object is masked _by_ '.GlobalEnv':
##
## diamonds
```

```
## The following object is masked from 'package:yarr':
##
## diamonds
```

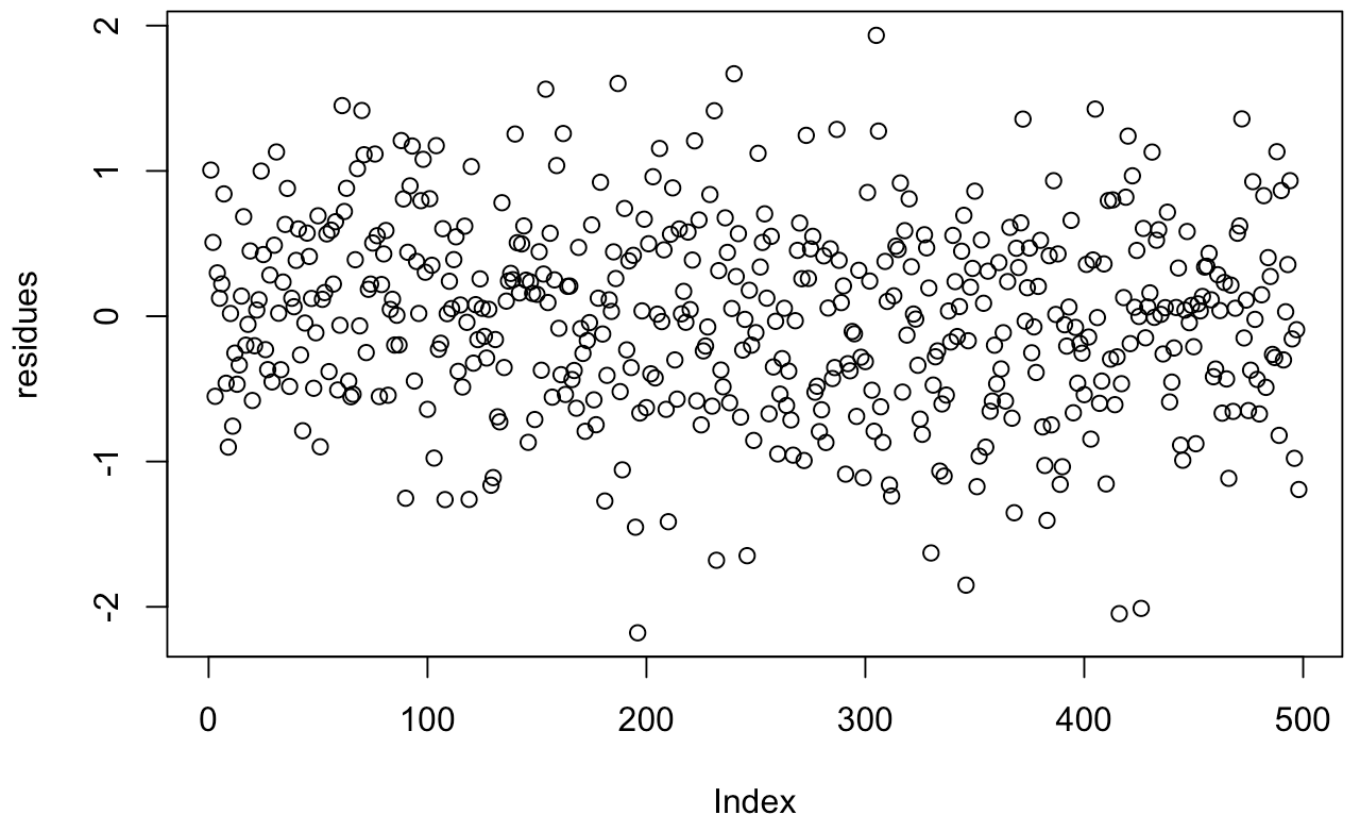
```
heart.plot <- ggplot(data_heart, aes(x=biking, y=heart.disease)) + geom_point()  
heart.plot <- heart.plot + geom_smooth(method="lm", col="black")
```

```
heart.plot
```

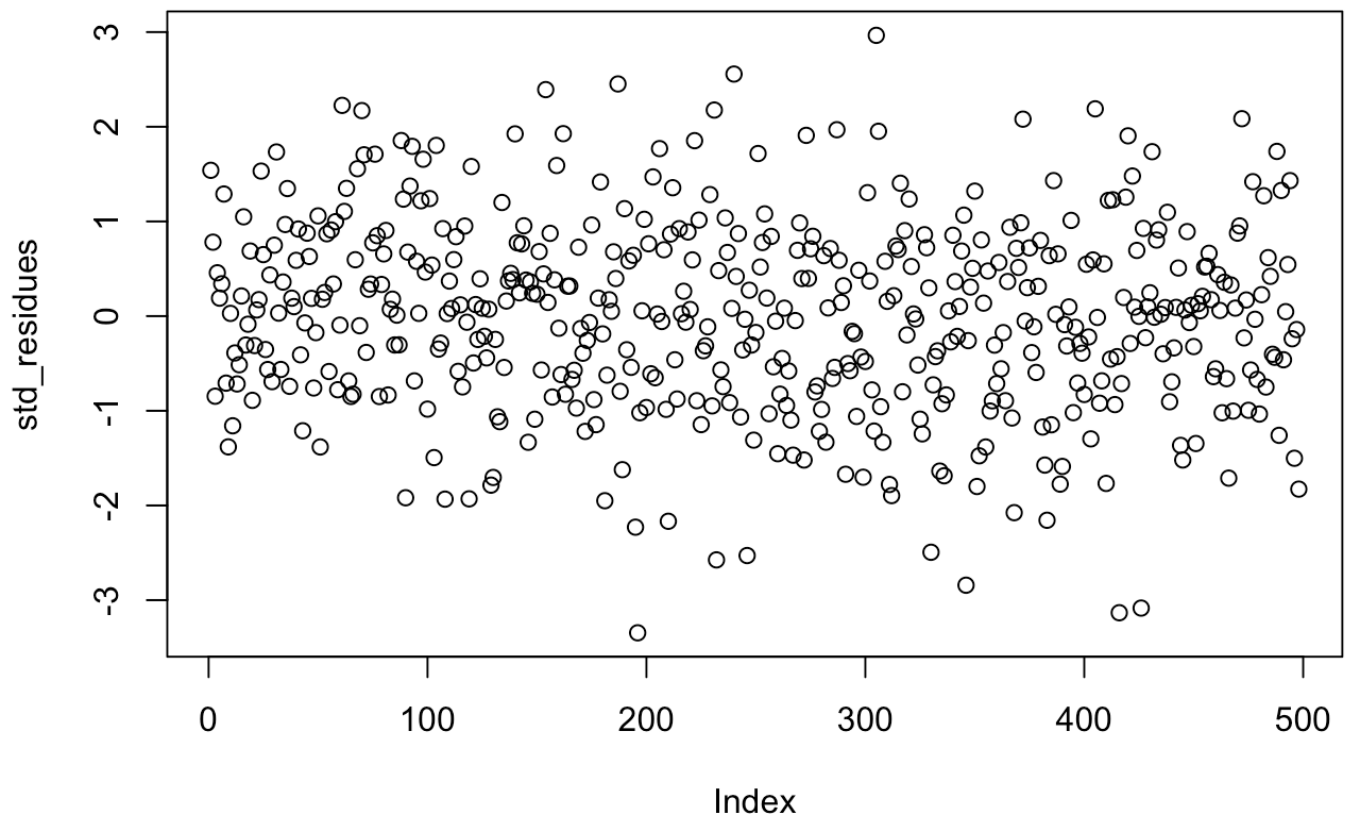
```
## `geom_smooth()` using formula 'y ~ x'
```



```
residues = resid (heart.disease.lm)  
plot(residues)
```



```
std_residues = rstandard (heart.disease.lm)
plot(std_residues)
```



```
s_residues = rstudent (heart.disease.lm)
plot(s_residues)
```

