Example-Ch9: Diagnostic

Assumptions about the multiple linear regression model

1. Linearity
2. Constant variance (homogeneous variance)
3. Independence
4. Distribution
5. Lack of outliers

Residuals

R Package: GLMsData Dataset: lungcap

Use FEV as the dependent variable

```
> reg1 <- lm (FEV ~ Ht + Gender + Smoke, data = lungcap)
```

The residuals (raw residuals, studentized residuals, studentized deleted residuals)
(In R, rstandard is the studentized residual (internally studentized residual);
rstudent is the studentized deleted residual (externally studentized residual))

```
> r1 <- resid(reg1)
> r2 <- rstandard(reg1)
> r3 <- rstudent(reg1)
```

$\hat{\varepsilon}_i = y_i - \hat{y}_i$

studentized residuals r_i

studentized deleted residuals t_i

```
> c(mean(r1), mean(r2), mean(r3))
[1] -8.066157e-18  2.131402e-04  3.872182e-04

> c(var(r1), var(r2), var(r3))
[1] 0.1812849 1.0027232 1.0083382
```

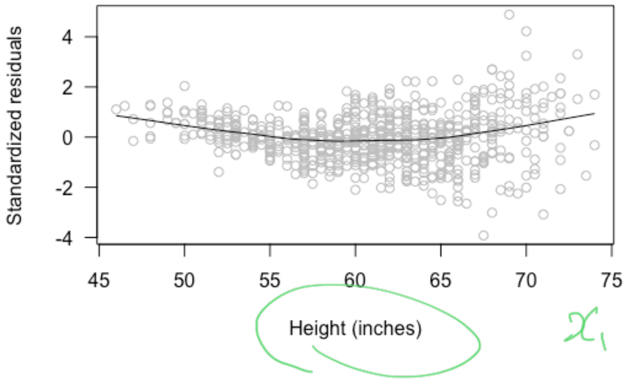
x_{ij}, $j=1,\ldots,k$.

Residual plot against a predictor variable

Example: vs height

```
> scatter.smooth(rstandard(reg1) ~ lungcap$Ht, col="grey", las=1,
ylab="Standardized residuals", xlab="Height (inches)")
```

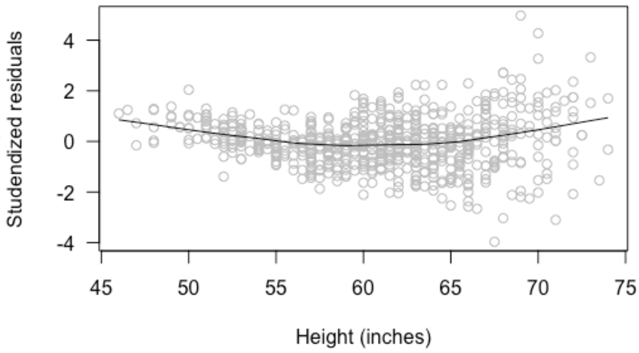
r_i



x_1

```
> scatter.smooth(rstudent(reg1) ~ lungcap$Ht, col="grey", las=1, ylab="Studendized
residuals", xlab="Height (inches)")
```

t_i



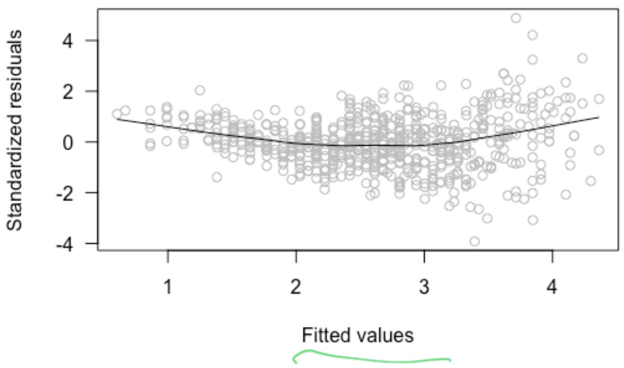
x_i

Residual plot against predicted value

```
> scatter.smooth(rstandard(reg1) ~ fitted(reg1), col="grey", las=1,
ylab="Standardized residuals", xlab="Fitted values")
```

plot

r_i

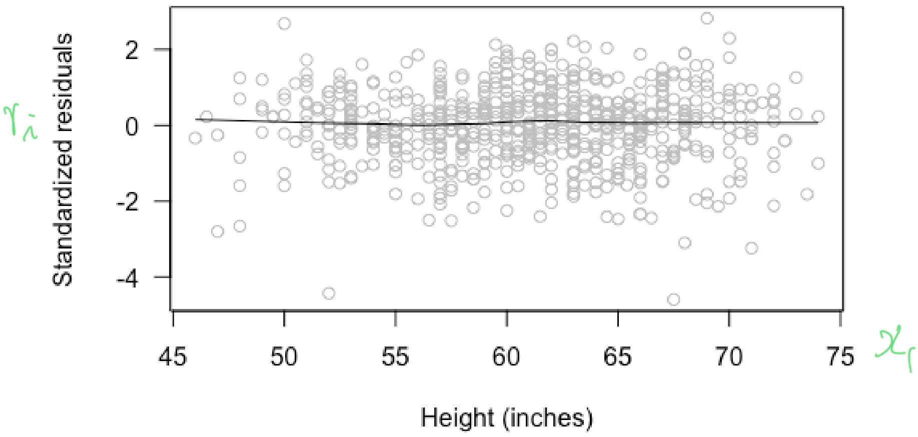


$\hat{y}_i$

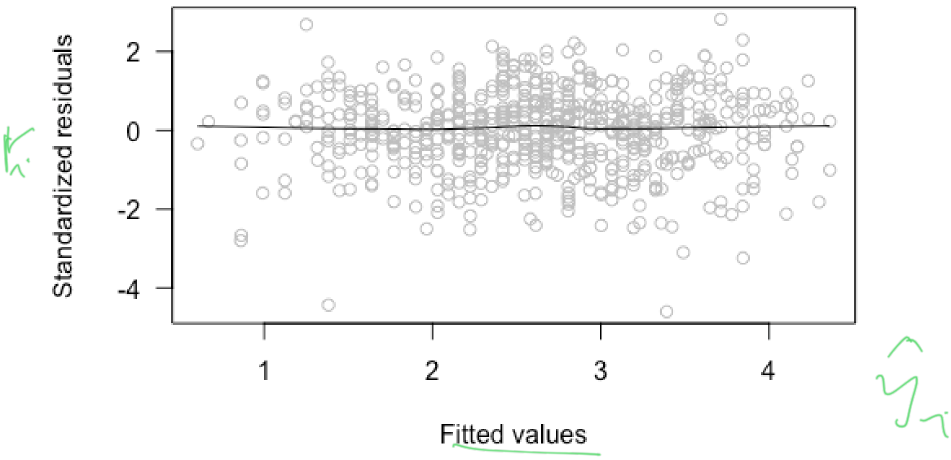
Transformation: FEV to log(FEV)

```
> reg2 <- lm (log(FEV) ~ Ht + Gender + Smoke, data = lungcap)

> scatter.smooth(rstandard(reg2) ~ lungcap$Ht, col="grey", las=1,
ylab="Standardized residuals", xlab="Height (inches)")
```




```
> scatter.smooth(rstandard(reg2) ~ fitted(reg1), col="grey", las=1,
ylab="Standardized residuals", xlab="Fitted values")
```



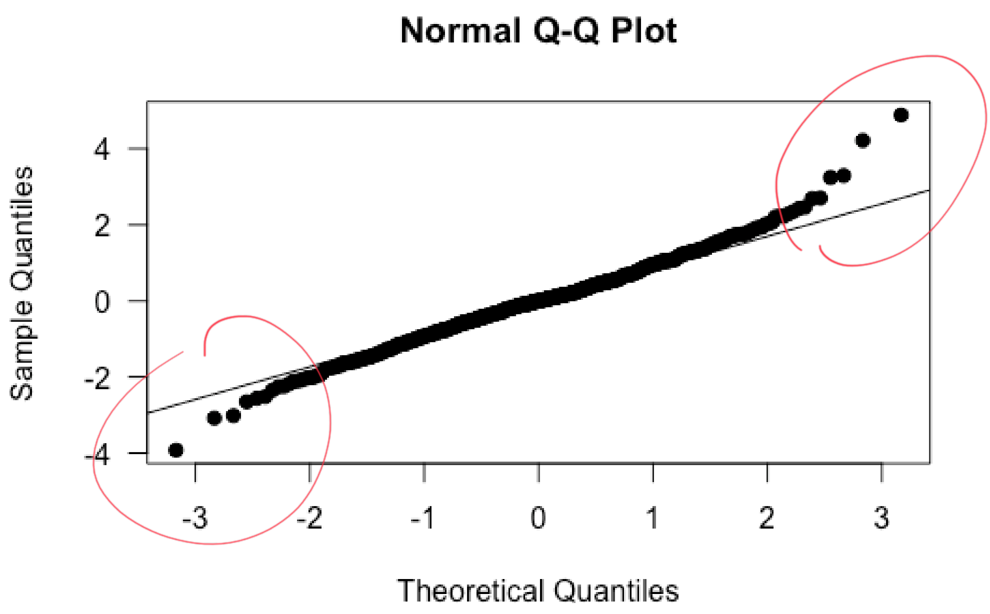
Identification of outliers

$|r_i|$ in descending order

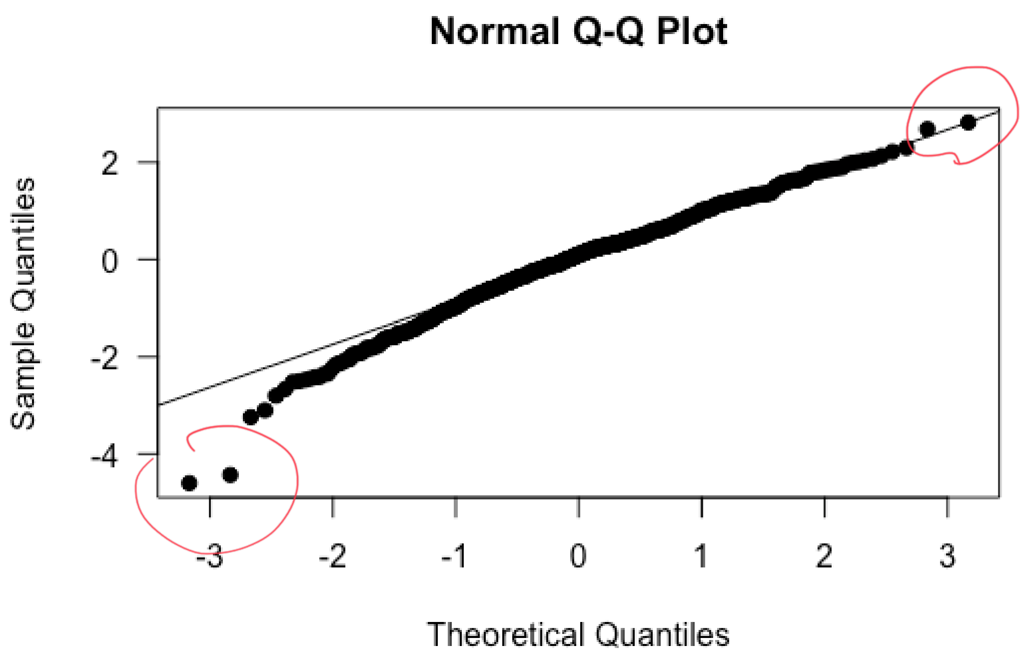
```
> abrstandard <- abs(rstandard(reg2))
> sort(abrstandard, decreasing = TRUE) [1:2]
     111      19
4.602118 4.432858
> sort(abrstandard, decreasing = TRUE) [1:8]
     111      19      269      613      576      281      285        2
4.602118 4.432858 3.242654 3.101003 2.822101 2.800127 2.684379 2.664408
```

Q-Q plots and normality

```
> qqnorm(rstandard(reg1), las=1, pch=19)
> qqline(rstandard(reg1))
```




```
> qqnorm(rstandard(reg2), las=1, pch=19)
> qqline(rstandard(reg2))
```



Leverage and extreme covariate values

Use reg1

h_{ii}

```
> h <- hatvalues(reg1)
> sort(h, decreasing = TRUE) [1:5]    # The largest 5 leverages
       629        631        633        636        643 
0.02207842 0.02034224 0.01882431 0.01882431 0.01882431 
```

The two five leverages are listed above. Compare them to the average leverage $(k+1)/n$.

$\geq 2(k+1)/n = 2 \times 4/654 = 0.0122$

```
> mean(h); length(coef(reg1))/length(lungcap$FEV)    # average leverage
[1] 0.006116208
```

$\frac{k+1}{n}$

```
> sort(h, decreasing = TRUE) [1:5] / mean(h)
     629      631      633      636      643 
3.609822 3.325956 3.077774 3.077774 3.077774 
```

≥ 2

Identify the large leverage points

```
> sort.h <- sort(h, decreasing=TRUE, index.return=TRUE)
> large.h <- sort.h$ix[1:5]  # Provide the index where these occur
> lungcap[large.h,]
    Age   FEV Ht Gender Smoke
629   9 1.953 58      M     1
631  11 1.694 60      M     1
633  11 4.637 72      M     1
636  12 3.751 72      M     1
643  14 3.957 72      M     1

> plot(FEV ~ Ht, main="Male smokers", data=subset(lungcap, Gender=="M" & Smoke==1),
las = 1, xlim=c(55, 75), ylim=c(0,5), xlab="Height (inches)", ylab="FEV (L)")
> points(FEV[large.h] ~Ht[large.h], data=lungcap, pch=19) # Large values
> legend("bottomright", pch=19, legend=c("Large leverage points"))
```

$\hat{\beta}_{j(i)} - \hat{\beta}_j$

Influential observations

```
infl <- influence.measures(reg1)
```

h_{ii}

```
> infl$infmat[1:5, 1:8]
       dfb.1_       dfb.Ht     dfb.GndM      dfb.Smok        dffit    cov.r       cook.d         hat
1  0.117124532 -0.109447749 -0.024484146  0.0154425438  0.127223620 1.011895 4.045093e-03 0.013092705
2 -0.005201569  0.004799092  0.001416715 -0.0005472614 -0.005845598 1.016771 8.555874e-06 0.010438869
3  0.051386692 -0.047410587 -0.013995834  0.0054064367  0.057749104 1.014813 8.346179e-04 0.010438869
4  0.113246447 -0.104483871 -0.030844144  0.0119147530  0.127267986 1.007226 4.045952e-03 0.010438869
5  0.115718262 -0.105902822 -0.036128118  0.0102352241  0.133115946 1.003826 4.423882e-03 0.009270865
```

```
> infl$is.inf[1:5, 1:8]
  dfb.1_ dfb.Ht dfb.GndM dfb.Smok dffit cov.r cook.d   hat
1  FALSE  FALSE    FALSE    FALSE FALSE FALSE  FALSE FALSE
2  FALSE  FALSE    FALSE    FALSE FALSE FALSE  FALSE FALSE
3  FALSE  FALSE    FALSE    FALSE FALSE FALSE  FALSE FALSE
4  FALSE  FALSE    FALSE    FALSE FALSE FALSE  FALSE FALSE
5  FALSE  FALSE    FALSE    FALSE FALSE FALSE  FALSE FALSE
```

```
> index=(rowSums(infl$is.inf)>0)
> infl$infmat[index,]
          dfb.1_        dfb.Ht    dfb.GndM     dfb.Smok       dffit     cov.r       cook.d         hat
111  0.1979650250 -2.310150e-01  0.204914171  0.1297105399 -0.327526959 0.9204818 2.622407e-02 0.006772166
152  0.0633867730 -8.167752e-02  0.106862287  0.0588849812 -0.154875769 0.9799958 5.959365e-03 0.004714197
257 -0.0464712060  6.436666e-02 -0.102349940 -0.0532689920  0.144870248 0.9803407 5.215325e-03 0.004240093
269  0.2337948553 -2.603960e-01  0.175481580  0.1252040652 -0.317628035 0.9586145 2.489178e-02 0.010378225
281 -0.0163426402  1.643190e-02 -0.008741674 -0.0033822900 -0.018539665 1.0206860 8.605907e-05 0.014357727
282  0.0030757521 -3.091748e-03  0.001722972  0.0006156514  0.003544940 1.0192845 3.146486e-06 0.012864807
493  0.0613287434 -6.283558e-02 -0.080629381  0.0431974667 -0.153967862 0.9710763 5.877757e-03 0.003694372
528 -0.0461332587  4.739228e-02  0.072802495 -0.0357619743  0.132974962 0.9792386 4.393577e-03 0.003527078
538  0.1283296618 -1.303401e-01 -0.058114066  0.0606470040 -0.181048120 0.9814632 8.143298e-03 0.006391871
539 -0.1976992030  2.007178e-01  0.081923737 -0.0913852282  0.270946008 0.9490351 1.808351e-02 0.006824022
567 -0.1208997542  1.229109e-01  0.066084721 -0.0601840179  0.183050698 0.9756246 8.313674e-03 0.005607028
576 -0.2647167744  2.689834e-01  0.131443123 -0.1282122460  0.385982987 0.8715554 3.593339e-02 0.005986207
580 -0.0722285624  7.374548e-02  0.070012146 -0.0441660006  0.147329850 0.9782179 5.391175e-03 0.004108417
585 -0.2584842282  2.624310e-01  0.107112186 -0.1194827284  0.354251655 0.9066401 3.056181e-02 0.006824022
587 -0.2734921283  2.772169e-01  0.069666016 -0.1146775428  0.333424787 0.9500270 2.737042e-02 0.009973122
589 -0.1769636829  1.794523e-01  0.052676702 -0.0762459275  0.222446396 0.9810602 1.228438e-02 0.008817479
590  0.0057669015 -4.876618e-03 -0.008790978  0.0360387765  0.040596543 1.0225566 4.125926e-04 0.016609249
591 -0.0005922271  1.411267e-04  0.004676383 -0.0167967276 -0.019522243 1.0227789 9.542294e-05 0.016370437
592  0.0067206614 -1.281585e-02  0.064589709 -0.2177039090 -0.258409344 0.9981139 1.661709e-02 0.016409945
593 -0.0014373800  2.740987e-03 -0.013814110  0.0465613771  0.055267243 1.0218109 7.645780e-04 0.016409945
596  0.0034230261 -2.530935e-03 -0.009033332  0.0346318709  0.039556188 1.0224023 3.917192e-04 0.016436872
597 -0.0003057978  7.287108e-05  0.002414661 -0.0086730281 -0.010080354 1.0228848 2.544229e-05 0.016370437
599  0.0108660427 -1.326114e-02  0.026164443 -0.0803786458 -0.099132992 1.0195845 2.458427e-03 0.016667849
600  0.0076900526 -6.969646e-03 -0.006825368  0.0310618549  0.034490919 1.0232155 2.978327e-04 0.017066457
602  0.0044061607 -5.695945e-03  0.013952041 -0.0441927493 -0.053764677 1.0220336 7.235822e-04 0.016555395
603 -0.0101666685  1.132268e-02 -0.013097591  0.0357027172  0.046904485 1.0231924 5.507496e-04 0.017382519
606 -0.0093351409  8.145042e-03  0.011596470 -0.0491970690 -0.055107961 1.0221781 7.601827e-04 0.016735166
607  0.0412319493 -3.486663e-02 -0.062853365  0.2576685271  0.290255452 0.9923111 2.093362e-02 0.016609249
608  0.0018010724 -4.291924e-04 -0.014221750  0.0510819658  0.059370764 1.0215891 8.822918e-04 0.016370437
609  0.0018010724 -4.291924e-04 -0.014221750  0.0510819658  0.059370764 1.0215891 8.822918e-04 0.016370437
610  0.0012198331 -2.326140e-03  0.011723350 -0.0395143304 -0.046902567 1.0221335 5.506982e-04 0.016409945
612  0.0114734290 -1.400241e-02  0.027626974 -0.0848716236 -0.104674294 1.0191663 2.740668e-03 0.016667849
613  0.0543073008 -6.394615e-02  0.106304408 -0.3168295801 -0.396473032 0.9673744 3.880839e-02 0.016806788
615 -0.0002596004  6.186232e-05  0.002049874 -0.0073627794 -0.008557498 1.0228955 1.833578e-05 0.016370437
616  0.0079314603 -1.512476e-02  0.076226234 -0.2569255921 -0.304964546 0.9885599 2.308835e-02 0.016409945
617 -0.0167267270  1.562506e-02  0.009964126 -0.0506213648 -0.055805584 1.0232859 7.795586e-04 0.017762036
620  0.0008344854 -6.170063e-04 -0.002202199  0.0084427611  0.009643240 1.0229573 2.328364e-05 0.016436872
621  0.0039513829 -2.921595e-03 -0.010427661  0.0399774290  0.045661833 1.0222061 5.219536e-04 0.016436872
623  0.0115041043 -1.321878e-02  0.019086387 -0.0552032894 -0.070151160 1.0217533 1.231651e-03 0.016972213
624 -0.0063834798  5.692823e-03  0.006637777 -0.0291577400 -0.032505111 1.0230735 2.645275e-04 0.016887569
625  0.0026280257 -2.222316e-03 -0.004006123  0.0164231746  0.018500187 1.0230439 8.569340e-05 0.016609249
626 -0.0020797607  1.758691e-03  0.003170356 -0.0129969328 -0.014640634 1.0230917 5.366856e-05 0.016609249
627  0.0001165259  1.150548e-03 -0.013282706  0.0462075323  0.054248730 1.0218164 7.366642e-04 0.016376948
628 -0.0022786711  4.345273e-03 -0.021899438  0.0738135104  0.087614875 1.0200686 1.920687e-03 0.016409945
629 -0.0546776792  5.913679e-02 -0.051995290 -0.1196168486 -0.131315999 1.0240647 4.312540e-03 0.022078420
630  0.0029550208 -1.689413e-03 -0.012996786 -0.0327534635 -0.038244193 1.0237261 3.661720e-04 0.017635202
631 -0.1009541779  1.114137e-01 -0.119360223 -0.2788094846 -0.303492413 0.9994645 2.290583e-02 0.020342236
632  0.0113986203 -7.813100e-03 -0.036531937 -0.0925958343 -0.109328835 1.0201194 2.989745e-03 0.017691141
633 -0.0466641340  4.173173e-02  0.047301914  0.1253718908  0.163689596 1.0167018 6.694486e-03 0.018824305
634 -0.0014998176  1.356014e-02 -0.126676788 -0.3140381237 -0.356427785 0.9807276 3.146584e-02 0.017626302
635  0.0067478278 -3.857796e-03 -0.029678328 -0.0747929523 -0.087331103 1.0215627 1.908369e-03 0.017635202
636  0.0361069476 -3.229044e-02 -0.036600438 -0.0970080425 -0.126656838 1.0202137 4.011500e-03 0.018824305
637 -0.0279676010  1.598934e-02  0.123007234  0.3099930107  0.361959661 0.9794357 3.243938e-02 0.017635202
638 -0.0435444038  3.275533e-02  0.109045945  0.2780283234  0.332222437 0.9867627 2.737814e-02 0.017773564
639 -0.0123547875  9.293619e-03  0.030939440  0.0788845536  0.094260967 1.0212903 2.223023e-03 0.017773564
640 -0.0108110844  2.255571e-02 -0.124251985 -0.3064233043 -0.345089232 0.9835476 2.951653e-02 0.017676306
641 -0.0281417117  1.608888e-02  0.123773008  0.3119228541  0.364213019 0.9788917 3.283997e-02 0.017635202
643  0.0168583916 -1.507646e-02 -0.017088803 -0.0452932102 -0.059136280 1.0243303 8.753762e-04 0.018824305
644  0.0004992581 -4.513890e-03  0.042168071  0.1045367663  0.118647404 1.0192932 3.520468e-03 0.017626302
645 -0.0026458768  1.512672e-03  0.011637108  0.0293269096  0.034243218 1.0238280 2.935716e-04 0.017635202
646 -0.0379410771  3.294647e-02  0.048785905  0.1275561054  0.161194859 1.0162791 6.492064e-03 0.018368116
647 -0.0056874162 -2.634630e-03  0.086770699  0.2162618008  0.247577324 1.0028942 1.526677e-02 0.017602783
648 -0.0016919681  9.673139e-04  0.007441622  0.0187537812  0.021897630 1.0240717 1.200563e-04 0.017635202
649 -0.0129018314  1.027195e-02  0.026362386  0.0676224259  0.081843701 1.0221785 1.676228e-03 0.017882474
650 -0.0687082146  6.144574e-02  0.069647281  0.1845974207  0.241016364 1.0065661 1.447706e-02 0.018824305
652  0.0142744393 -1.073764e-02 -0.035746722 -0.0911414114 -0.108906968 1.0202566 2.966754e-03 0.017773564
```

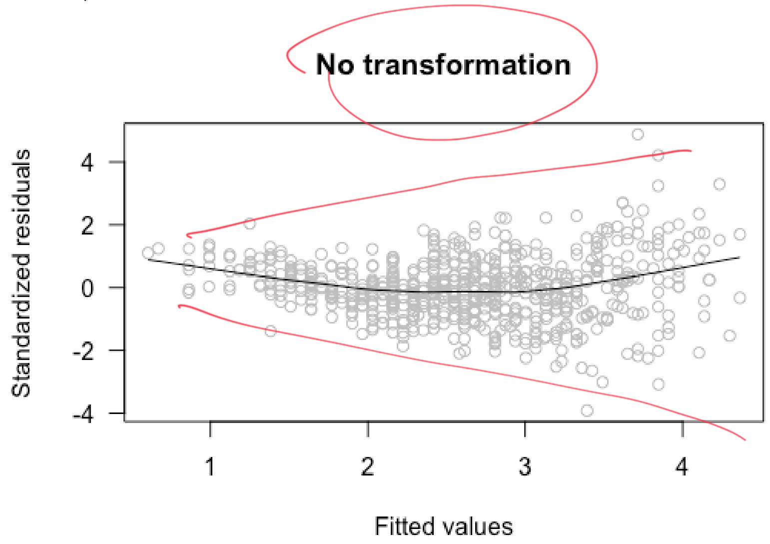
```
> dim(infl$infmat[index,])
[1] 66  8
> infl$is.inf[index,]
    dfb.1_ dfb.Ht dfb.GndM dfb.Smok dffit cov.r cook.d   hat
111  FALSE  FALSE    FALSE    FALSE  TRUE  TRUE  FALSE FALSE
152  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
257  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
269  FALSE  FALSE    FALSE    FALSE  TRUE  TRUE  FALSE FALSE
281  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
282  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
493  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
528  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
538  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
539  FALSE  FALSE    FALSE    FALSE  TRUE  TRUE  FALSE FALSE
567  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
576  FALSE  FALSE    FALSE    FALSE  TRUE  TRUE  FALSE FALSE
580  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
585  FALSE  FALSE    FALSE    FALSE  TRUE  TRUE  FALSE FALSE
587  FALSE  FALSE    FALSE    FALSE  TRUE  TRUE  FALSE FALSE
589  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
590  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
591  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
592  FALSE  FALSE    FALSE    FALSE  TRUE FALSE  FALSE FALSE
593  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
596  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
597  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
599  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
600  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
602  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
603  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
606  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
607  FALSE  FALSE    FALSE    FALSE  TRUE FALSE  FALSE FALSE
608  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
609  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
610  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
612  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
613  FALSE  FALSE    FALSE    FALSE  TRUE  TRUE  FALSE FALSE
615  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
616  FALSE  FALSE    FALSE    FALSE  TRUE FALSE  FALSE FALSE
617  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
620  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
621  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
623  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
624  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
625  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
626  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
627  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
628  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
629  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE  TRUE
630  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
631  FALSE  FALSE    FALSE    FALSE  TRUE FALSE  FALSE  TRUE
632  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
633  FALSE  FALSE    FALSE    FALSE FALSE FALSE  FALSE  TRUE
634  FALSE  FALSE    FALSE    FALSE  TRUE  TRUE  FALSE FALSE
635  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
636  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE  TRUE
637  FALSE  FALSE    FALSE    FALSE  TRUE  TRUE  FALSE FALSE
638  FALSE  FALSE    FALSE    FALSE  TRUE FALSE  FALSE FALSE
639  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
640  FALSE  FALSE    FALSE    FALSE  TRUE FALSE  FALSE FALSE
641  FALSE  FALSE    FALSE    FALSE  TRUE  TRUE  FALSE FALSE
643  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE  TRUE
644  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
645  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
646  FALSE  FALSE    FALSE    FALSE FALSE FALSE  FALSE  TRUE
647  FALSE  FALSE    FALSE    FALSE  TRUE FALSE  FALSE FALSE
648  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
649  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
650  FALSE  FALSE    FALSE    FALSE  TRUE FALSE  FALSE  TRUE
652  FALSE  FALSE    FALSE    FALSE FALSE  TRUE  FALSE FALSE
```

```
> colSums(infl$is.inf[index,])
  dfb.1_    dfb.Ht dfb.GndM dfb.Smok    dffit    cov.r   cook.d      hat
       0         0        0        0       18       56        0        7
```

Variance-Stabilizing Transformations

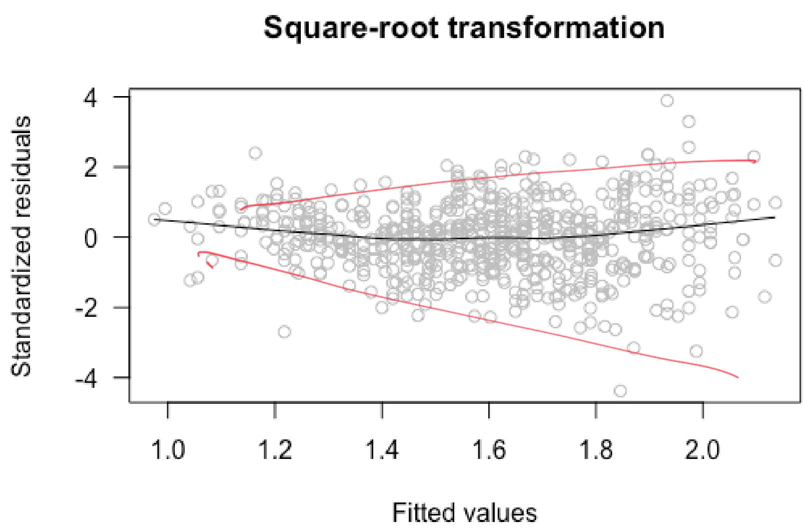
1. No transformation

```
> scatter.smooth(rstandard(reg1) ~ fitted(reg1), col="grey", las=1,
ylab="Standardized residuals", xlab="Fitted values", main="No
transformation")
```



2. Square-root transformation

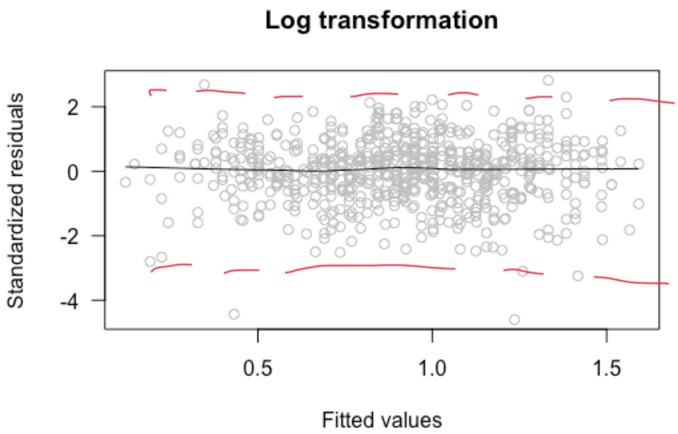
```
sqrtreg <-update(reg1, sqrt(FEV) ~ .)
> scatter.smooth(rstandard(sqrtreg) ~ fitted(sqrtreg), col="grey", las=1,
ylab="Standardized residuals", xlab="Fitted values", main="Square-root
transformation")
```



3. Log transformation

log(y_i) ~

```
> logreg <-update(reg1, log(FEV) ~ .)
> scatter.smooth(rstandard(logreg) ~ fitted(logreg), col="grey", las=1,
ylab="Standardized residuals", xlab="Fitted values", main="Log
transformation")
```



4. Box-Cox Transformations

$$y^* = \begin{cases} \dfrac{y^\lambda - 1}{\lambda} & \text{for} \quad \lambda \neq 0 \\ \log y & \text{for} \quad \lambda = 0 \end{cases}$$

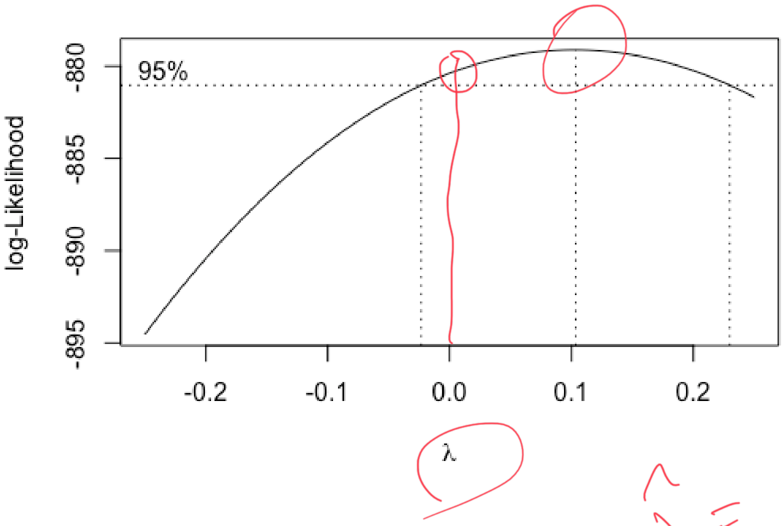
Can estimate λ using the data!

```
> boxcox(FEV~Ht+Gender+Smoke, lambda=seq(-0.25, 0.25, length=11),
data=lungcap)
```

$y_i^* \sim N(x_i'\beta, \sigma^2)$

$L(\theta)$, $\theta = (\beta, \sigma^2, \lambda)$

$L(\theta)$



$\hat{\lambda} = 0.1$

$y^* = \frac{y^{0.1} - 1}{0.1}$