431 Class 26

Thomas E. Love

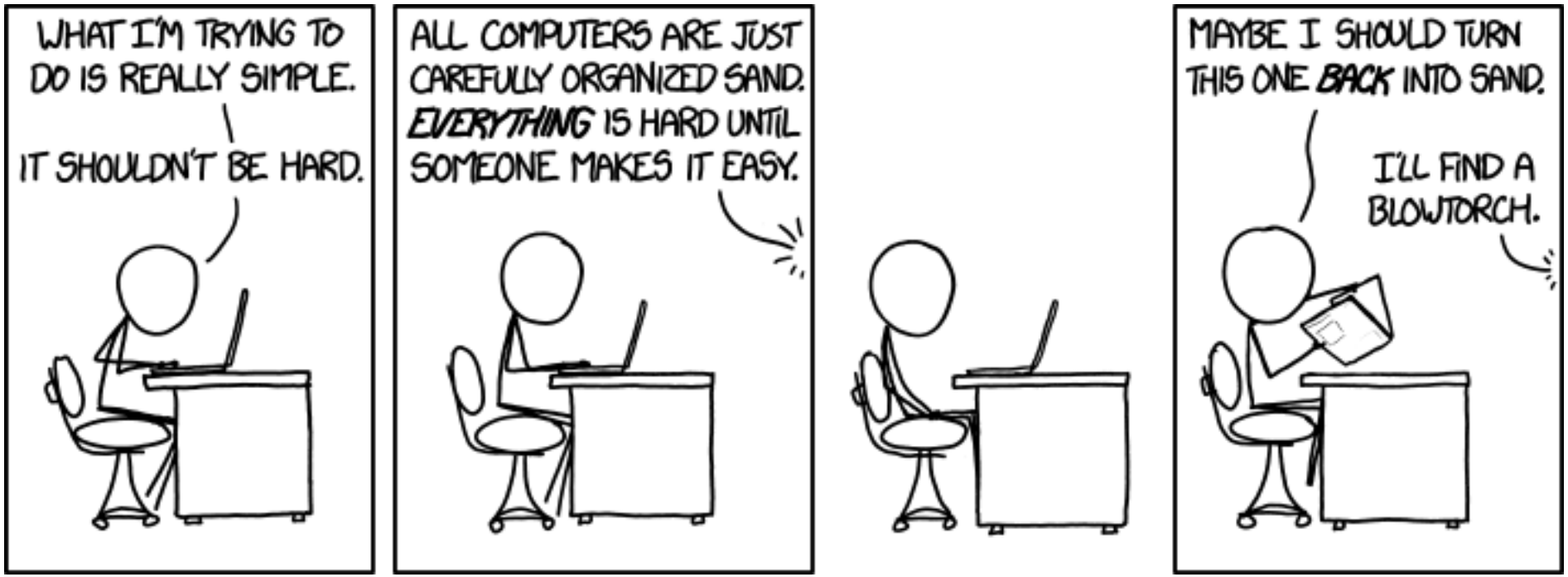
2017-12-05

Today's Agenda

- Discussion of Assignment 6
- Reasoning About Data
  - A data analysis allow you to understand how the *data*, as opposed to other aspects of an analysis like assumptions or models, played a role in producing the outputs. (Roger Peng)
- Anscombe's Data: A Quartet of Simple Regressions
- Calibrating Yourself on Residual Plots
- The `dm192` data
  - Approach 1. We have 7 regression inputs. How well can we predict today's systolic BP?
  - Approach 2. A new input (statin) is of special interest. What is its relationship with today's systolic BP, after we control for other inputs?

Assignment 6 Discussion

https://xkcd.com/1349/



Every detail matters. Computers aren't smart enough to guess our intent.

Reasoning About Data (Roger Peng)

"...I think it's tempting to think of the goal of methods development as removing the need to think about data and assumptions. The"ultimate" method is one where you don't have to worry about distributions or nonlinearities or interactions or anything like that. But I don't see that as the goal. Good methods, and good analyses, help us think about all those things much more efficiently. So what I might say is that...

When doing large-scale data analyses, the data analyst always has to think about the data and assumptions, and as such, some approaches can actually make that harder to do than others. The goal of the good data analysis is to make it easier to reason about how the data are related to the result, relative to the assumptions you make about the data and the models."

Roger Peng: https://simplystatistics.org/2017/11/20/follow-up-on-reasoning-about-data/

The tidyverse can do just about everything.

Except think.

R Setup for Today

```
library(car); library(broom); library(magrittr)
library(tidyverse)

dm192 <- read_csv("data/dm192.csv")
```

Anscombe's Data: A Famous Example

A Quartet of Simple Linear Regressions

```
anscombe
```

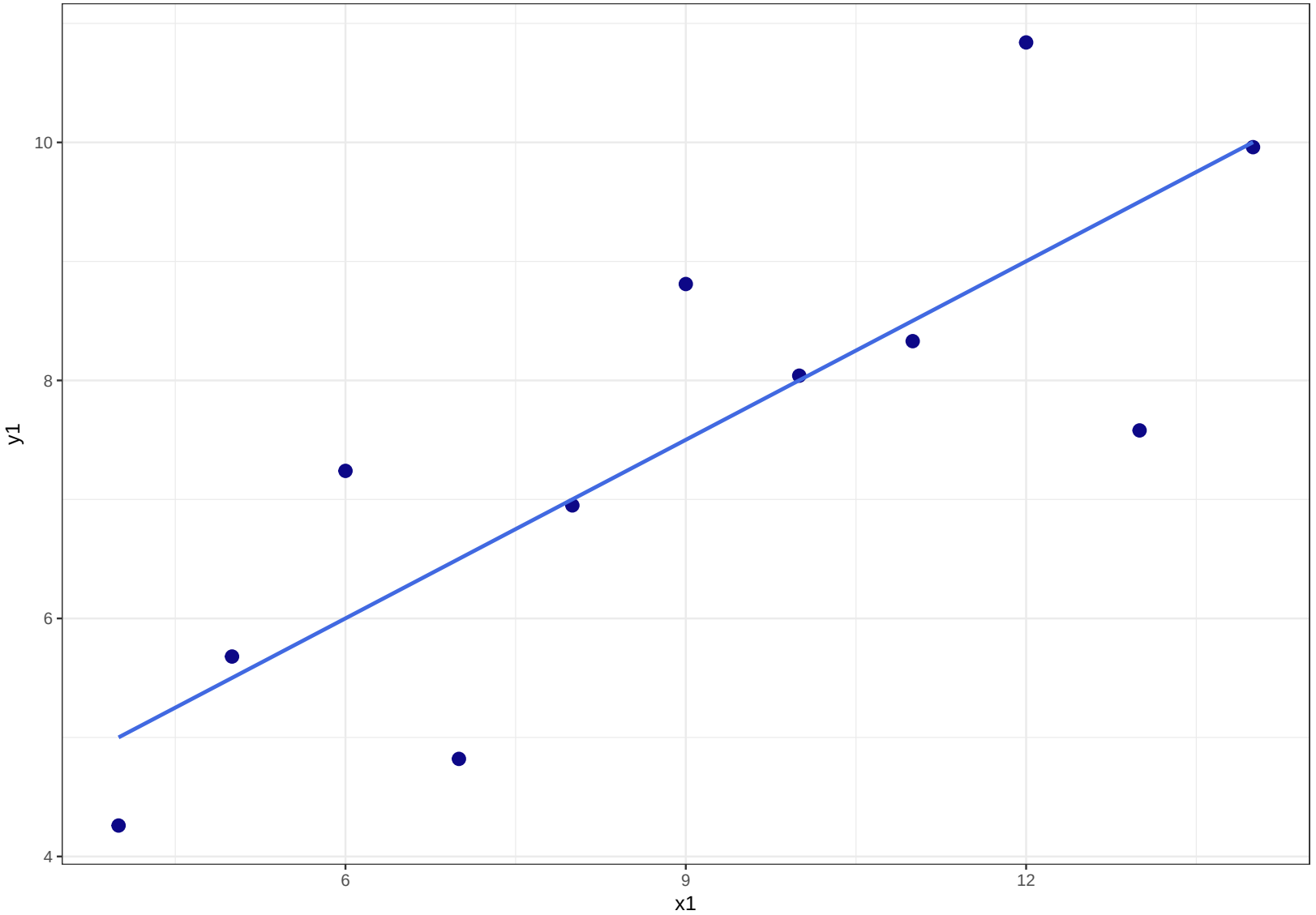
```
   x1 x2 x3 x4    y1   y2    y3    y4
1  10 10 10  8  8.04 9.14  7.46  6.58
2   8  8  8  8  6.95 8.14  6.77  5.76
3  13 13 13  8  7.58 8.74 12.74  7.71
4   9  9  9  8  8.81 8.77  7.11  8.84
5  11 11 11  8  8.33 9.26  7.81  8.47
6  14 14 14  8  9.96 8.10  8.84  7.04
7   6  6  6  8  7.24 6.13  6.08  5.25
8   4  4  4 19  4.26 3.10  5.39 12.50
9  12 12 12  8 10.84 9.13  8.15  5.56
10  7  7  7  8  4.82 7.26  6.42  7.91
11  5  5  5  8  5.68 4.74  5.73  6.89
```

Anscombe Model 1

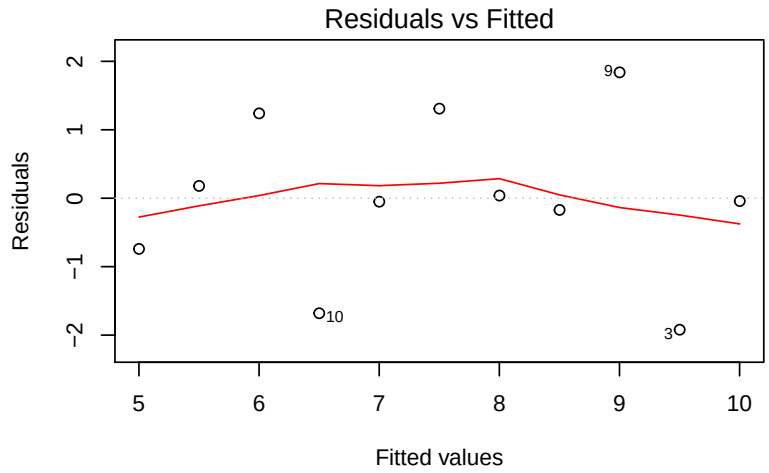
```
arm::display(lm(y1 ~ x1, data = anscombe))
```

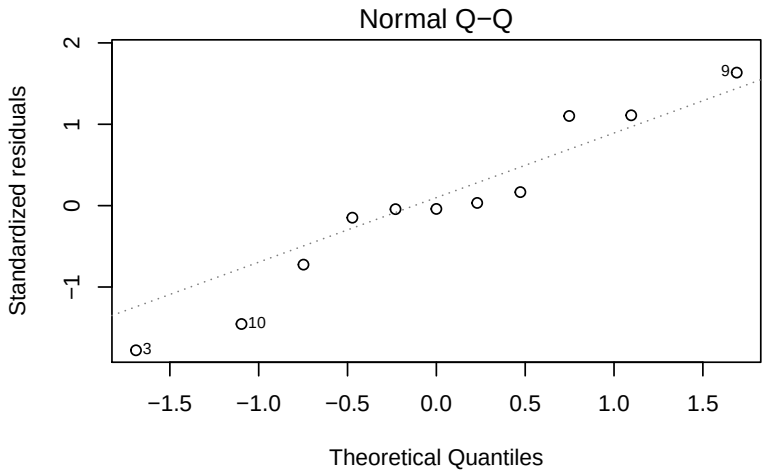
```
lm(formula = y1 ~ x1, data = anscombe)
            coef.est coef.se
(Intercept) 3.00     1.12
x1          0.50     0.12
---
n = 11, k = 2
residual sd = 1.24, R-Squared = 0.67
```

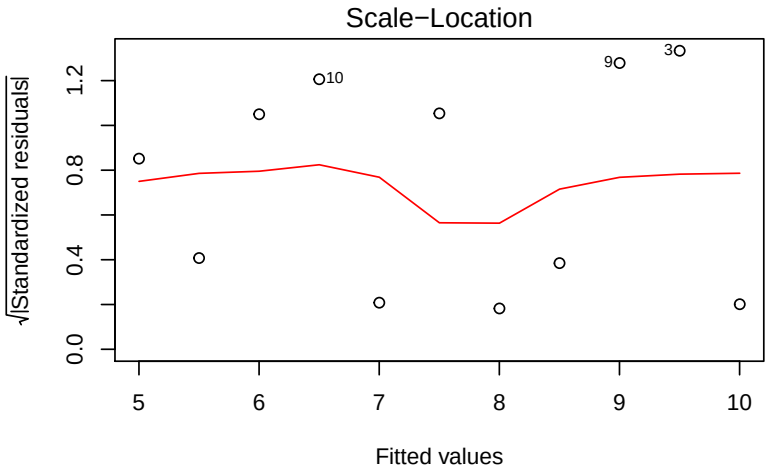
Plot of `y1` vs. `x1`

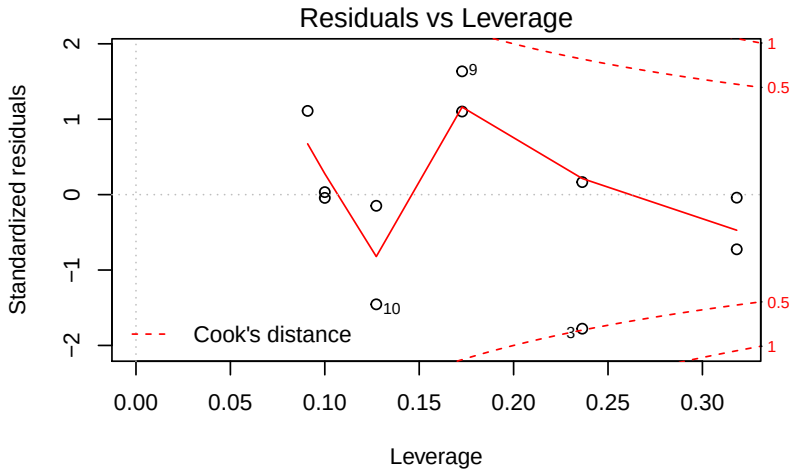


Residual Plots for Anscombe Model 1









Anscombe Model 2

```
arm::display(lm(y2 ~ x2, data = anscombe))
```

```
lm(formula = y2 ~ x2, data = anscombe)
            coef.est coef.se
(Intercept) 3.00     1.13
x2          0.50     0.12
---
n = 11, k = 2
residual sd = 1.24, R-Squared = 0.67
```

Anscombe Model 3

```
arm::display(lm(y3 ~ x3, data = anscombe))
```

```
lm(formula = y3 ~ x3, data = anscombe)
            coef.est coef.se
(Intercept) 3.00     1.12
x3          0.50     0.12
---
n = 11, k = 2
residual sd = 1.24, R-Squared = 0.67
```

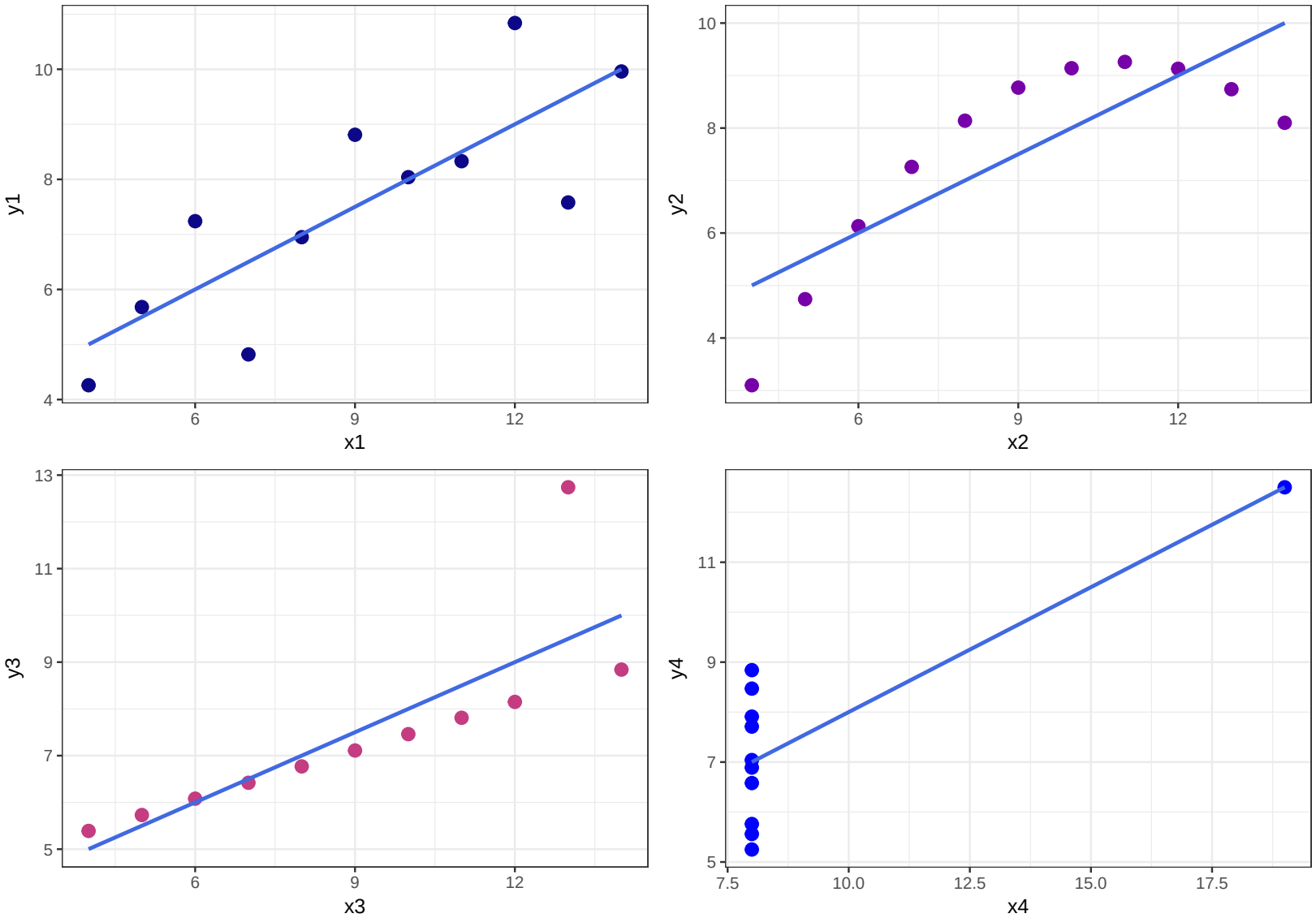
Anscombe Model 4

```
arm::display(lm(y4 ~ x4, data = anscombe))
```

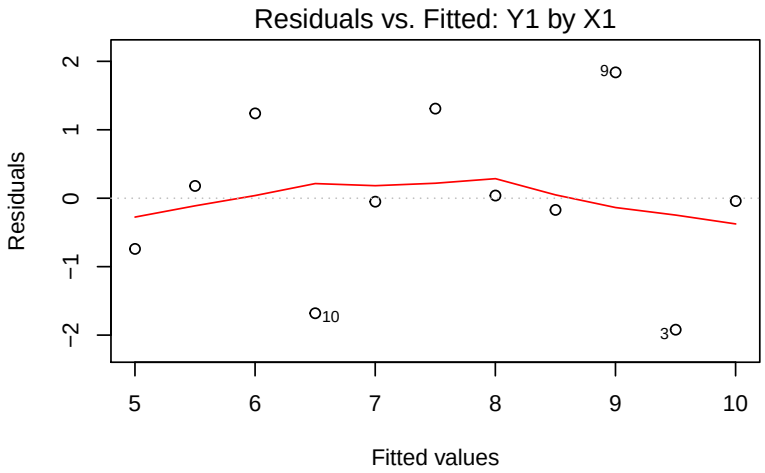
```
lm(formula = y4 ~ x4, data = anscombe)
            coef.est coef.se
(Intercept) 3.00     1.12
x4          0.50     0.12
---
n = 11, k = 2
residual sd = 1.24, R-Squared = 0.67
```

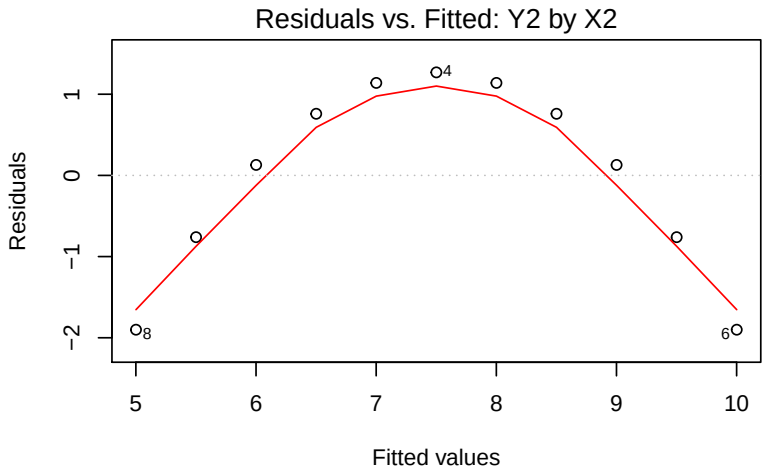
Models 1-4 all look about the same, but what happens if we plot the data?

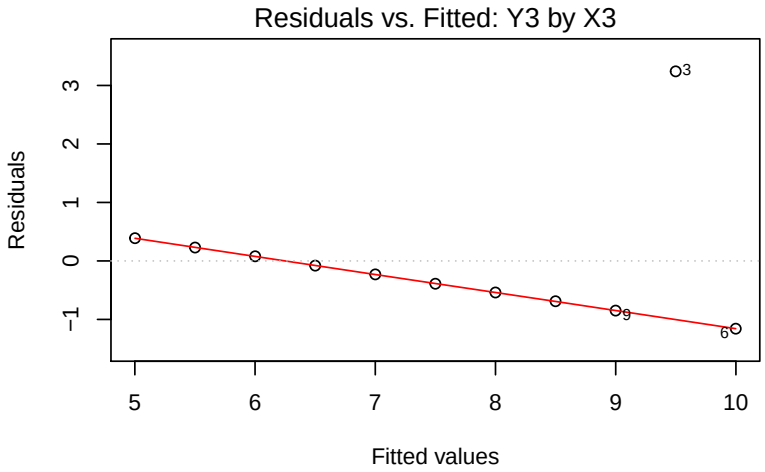
Plot the Data (and the regression lines)

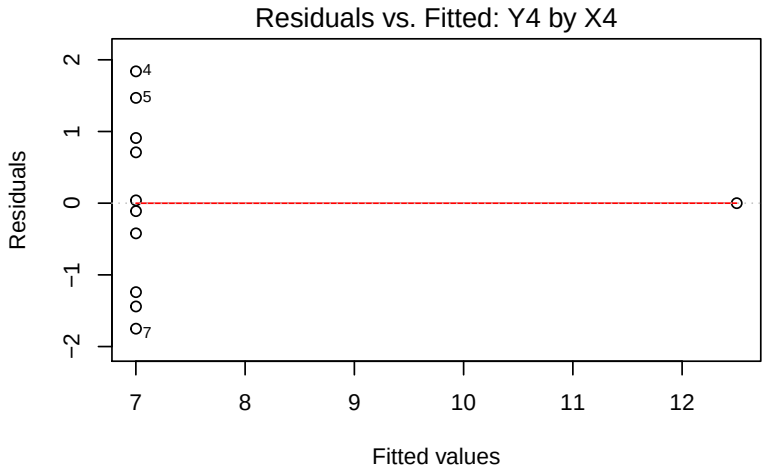


Do Residuals vs. Fitted Plots reveal the problems?









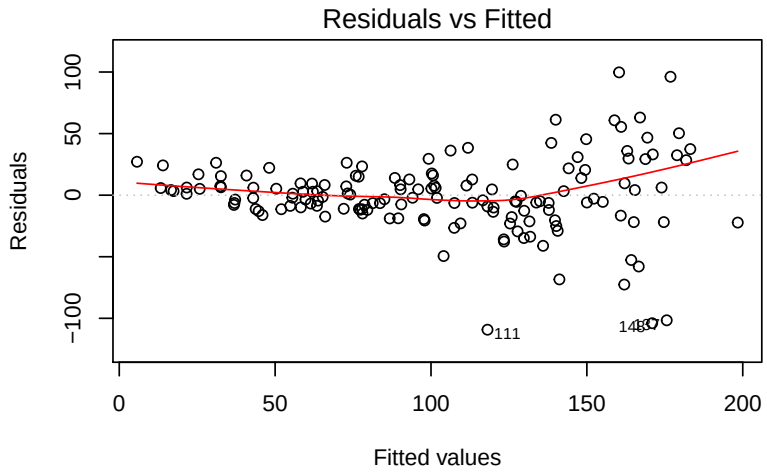
Calibrating Yourself on Residual Plots: Five New Examples

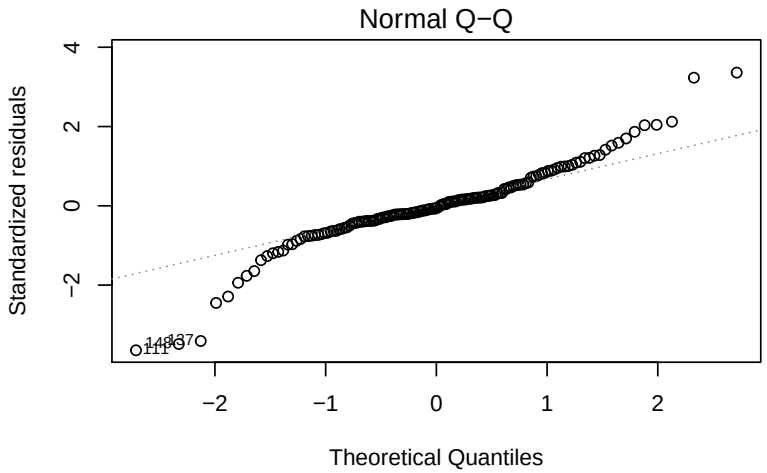
Your Response Options are

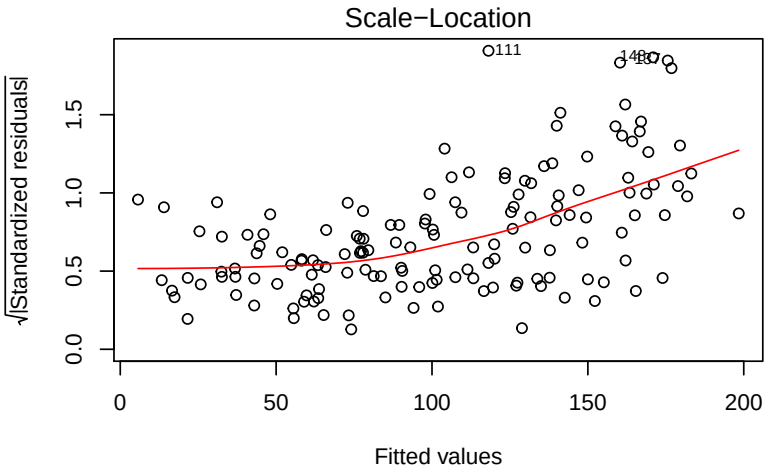
1. Assumptions of regression look reasonable.
2. Biggest problem is Linearity.
3. Biggest problem is non-constant variance.
4. Biggest problem is Normality
5. Some other problem is the biggest issue.

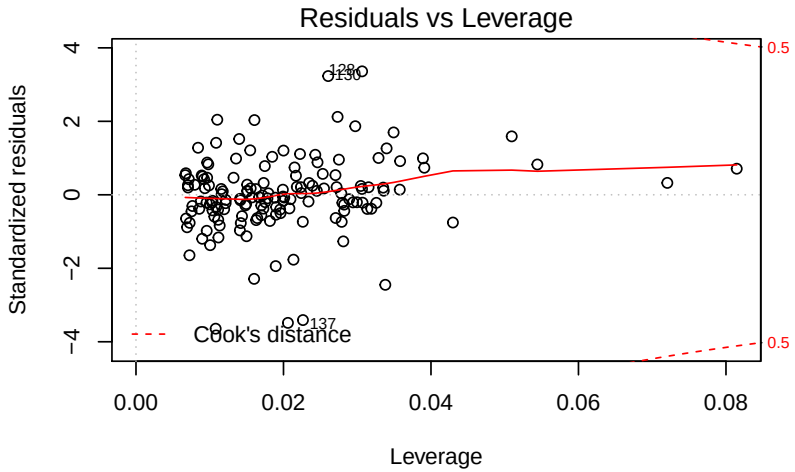
All of these models describe cross-sectional data, and so there's no issue with independence possible.

Example A

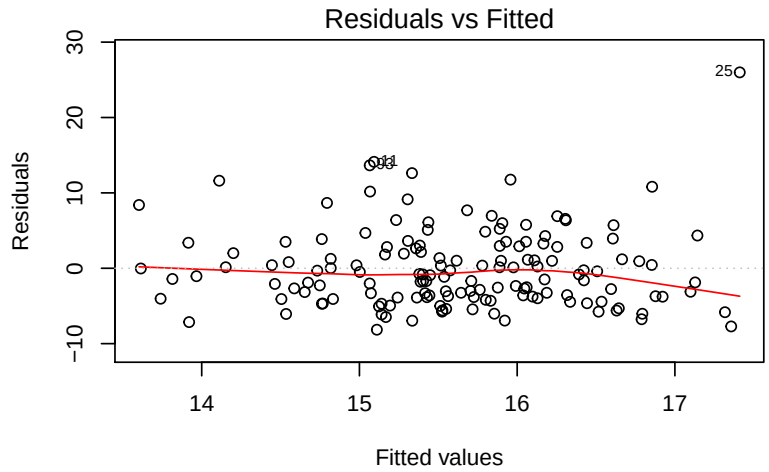

Residuals vs Fitted
Residuals
Fitted values
111

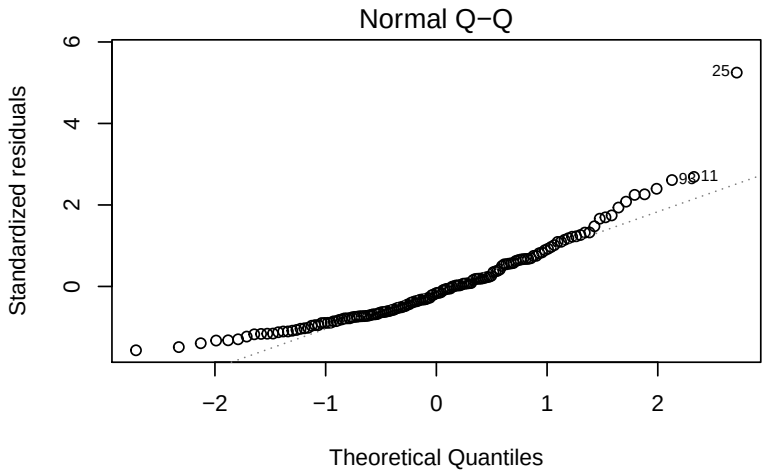

Normal Q−Q
Standardized residuals
Theoretical Quantiles

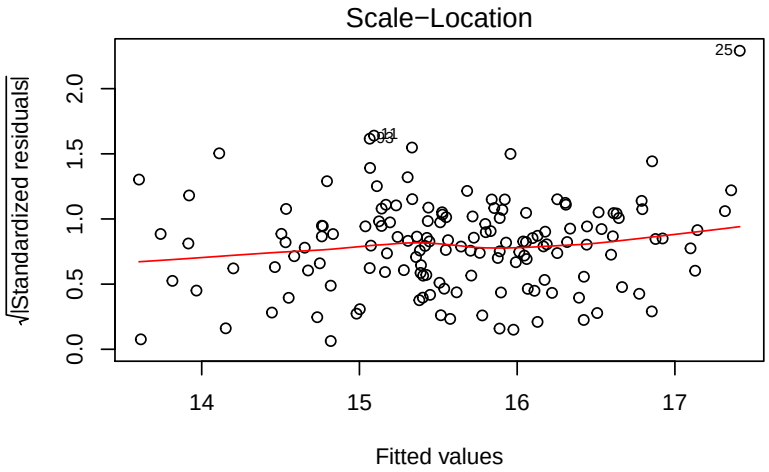

Scale−Location
√|Standardized residuals|
Fitted values
111

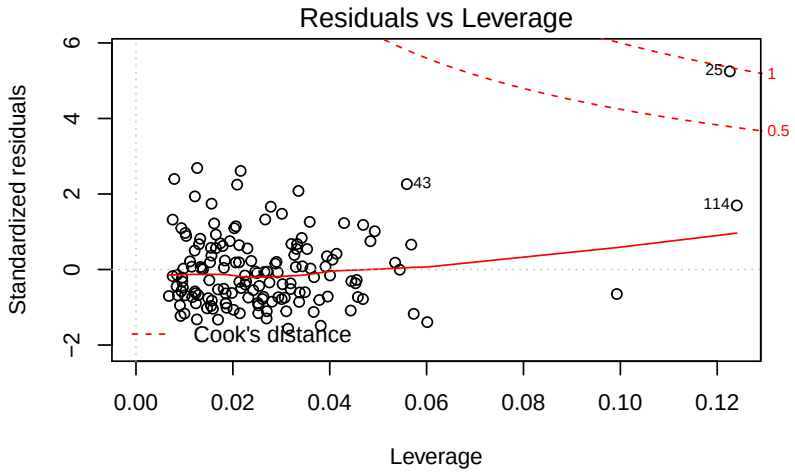

Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage

Example B

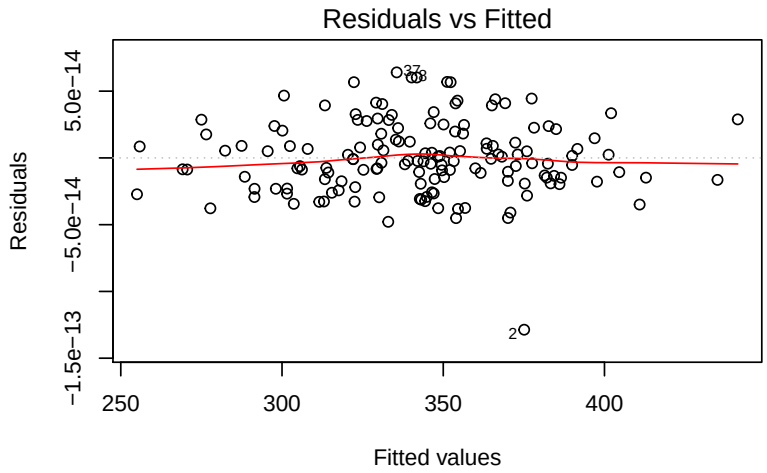

Residuals vs Fitted
Residuals
Fitted values
25
11
93

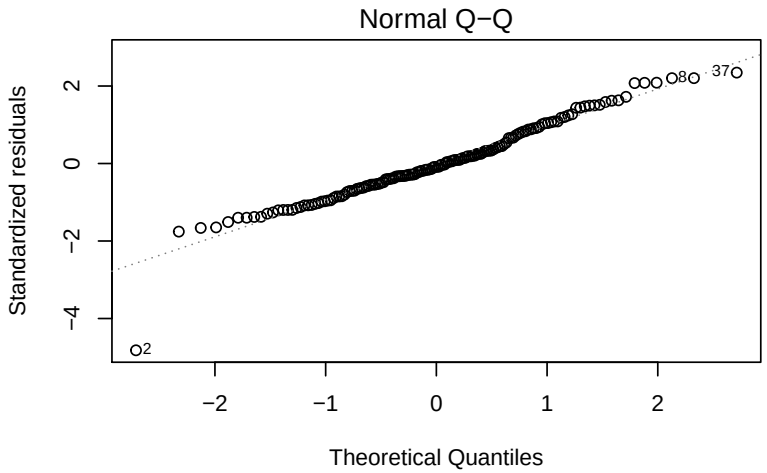

Normal Q−Q
Standardized residuals
Theoretical Quantiles
25
99
11

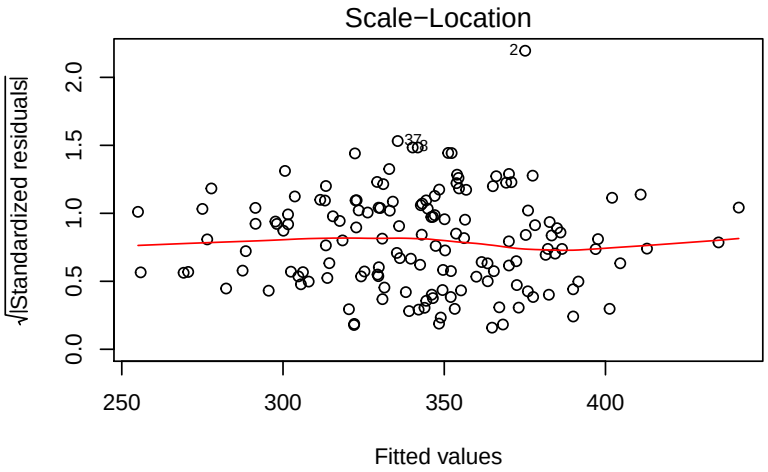

Scale−Location
√|Standardized residuals|
Fitted values
25
11
93

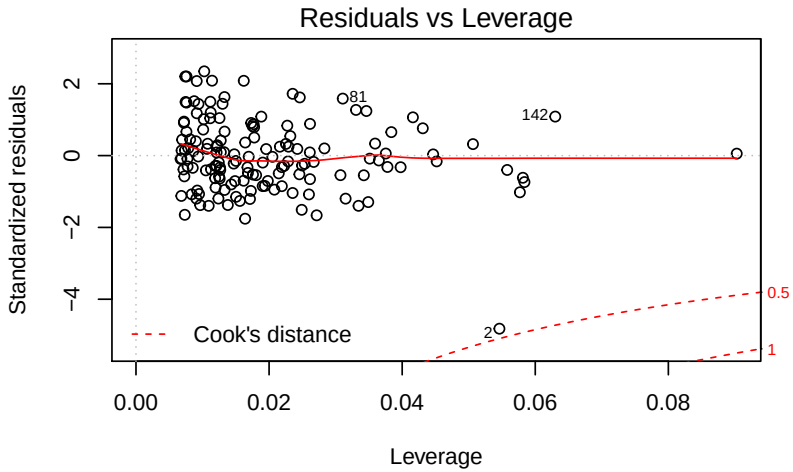

Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
25
43
114
1
0.5

Example C

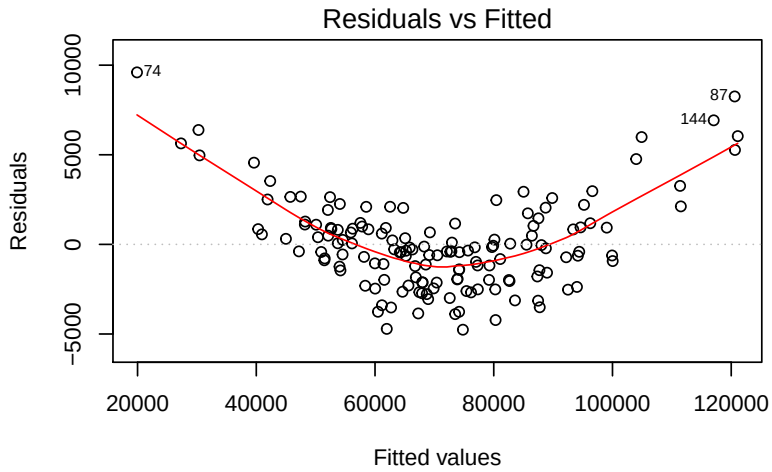

Residuals vs Fitted
Residuals
Fitted values
5.0e−14
−5.0e−14
−1.5e−13
250
300
350
400
37
8
2

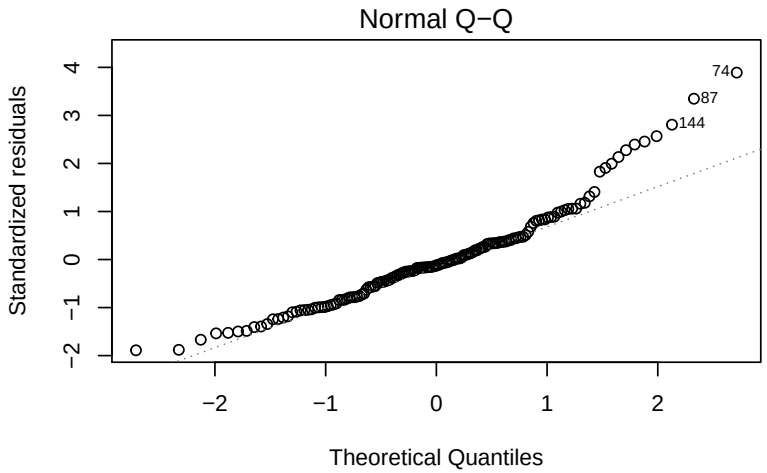

Normal Q−Q
Standardized residuals
Theoretical Quantiles
2
0
−2
−4
−2
−1
0
1
2
37
8
2

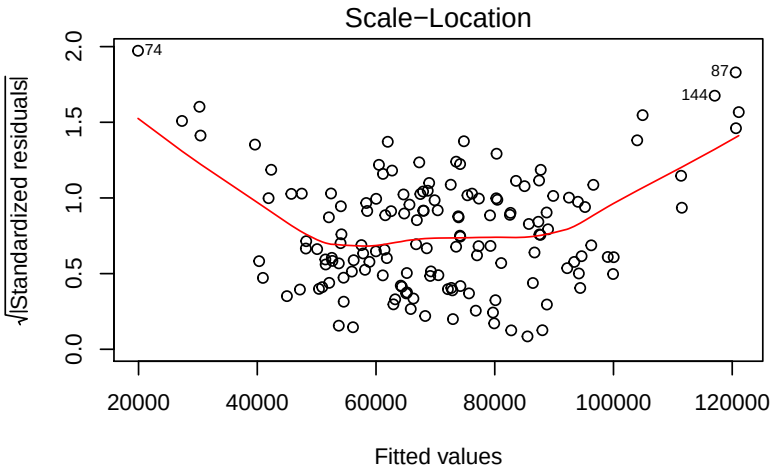

Scale−Location
√|Standardized residuals|
Fitted values
0.0
0.5
1.0
1.5
2.0
250
300
350
400
2
37
8

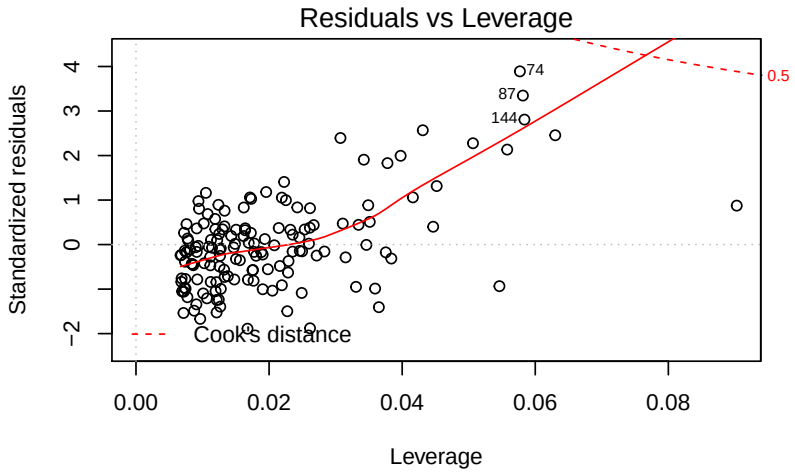

Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
2
0
−2
−4
0.00
0.02
0.04
0.06
0.08
81
142
2
0.5
1

Example D

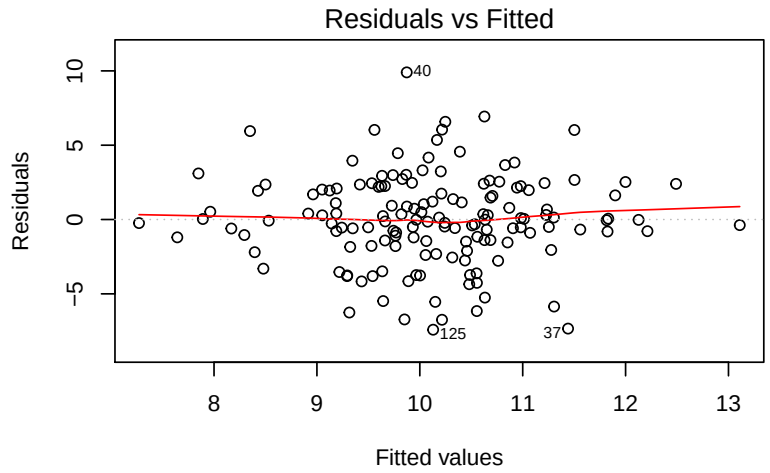

Residuals vs Fitted
Residuals
Fitted values
74
87
144
−5000
0
5000
10000
20000
40000
60000
80000
100000
120000

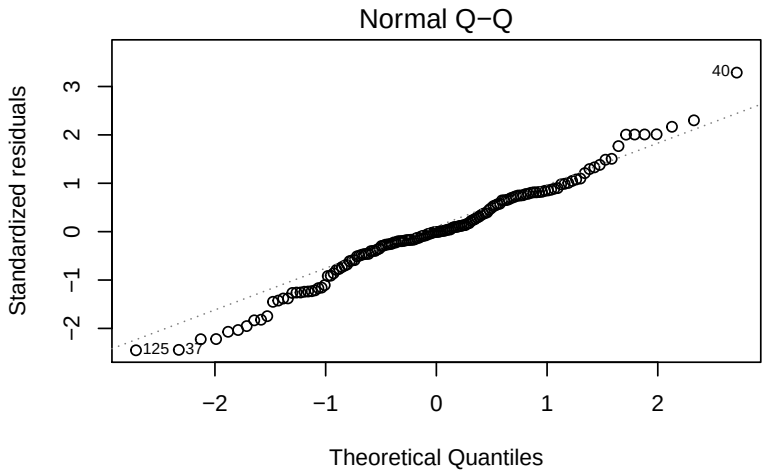

Normal Q−Q
Standardized residuals
Theoretical Quantiles
74
87
144
−2
−1
0
1
2
3
4

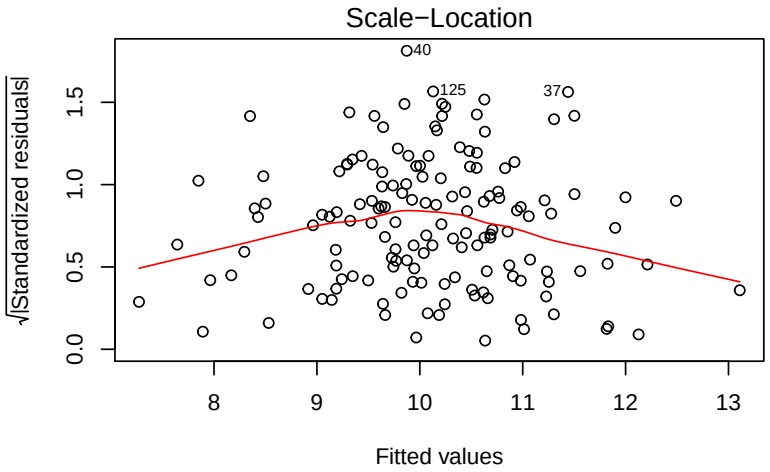

Scale−Location
√|Standardized residuals|
Fitted values
74
87
144
0.0
0.5
1.0
1.5
2.0
20000
40000
60000
80000
100000
120000

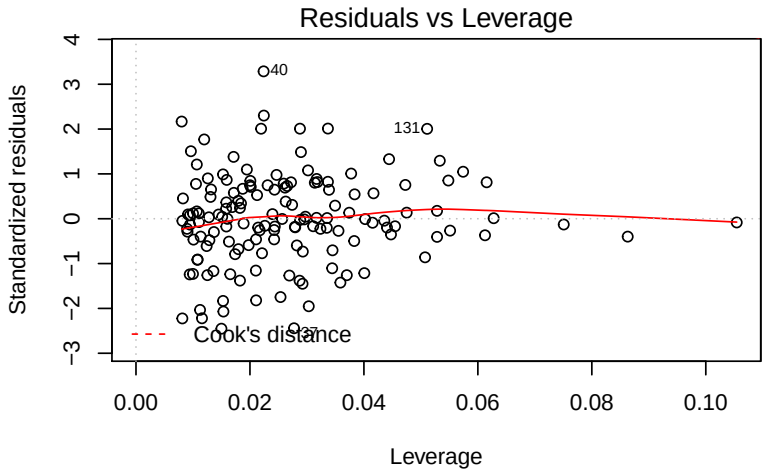

Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
0.5
74
87
144
−2
0
1
2
3
4
0.00
0.02
0.04
0.06
0.08

Example E

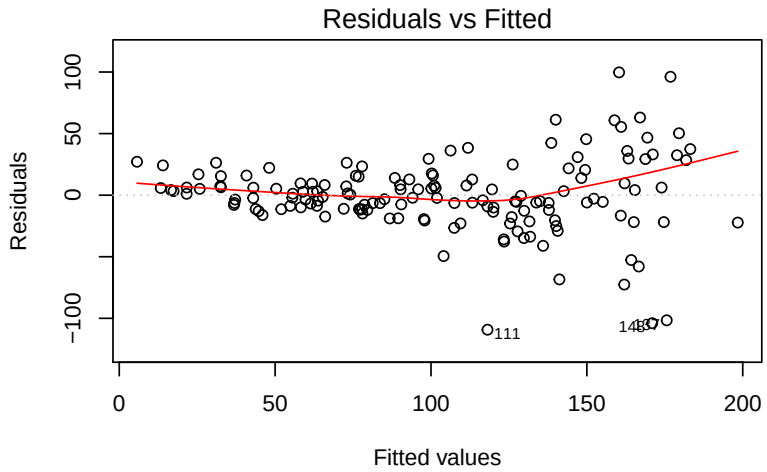

Residuals vs Fitted
Residuals
Fitted values
40
125
37

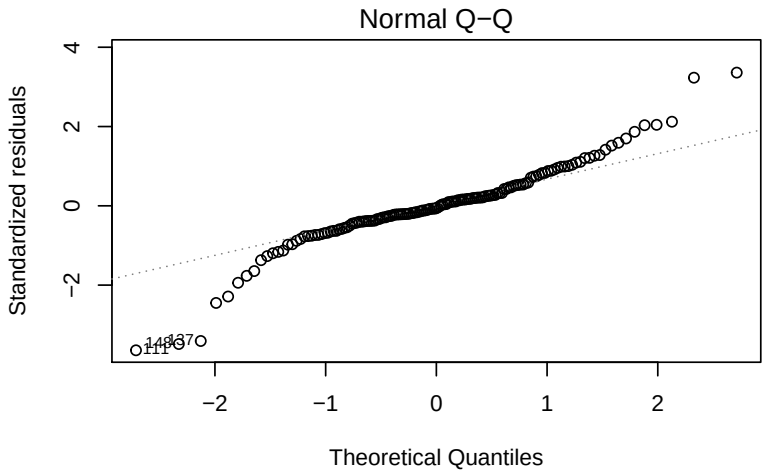

Normal Q−Q
Standardized residuals
Theoretical Quantiles
40
125
37

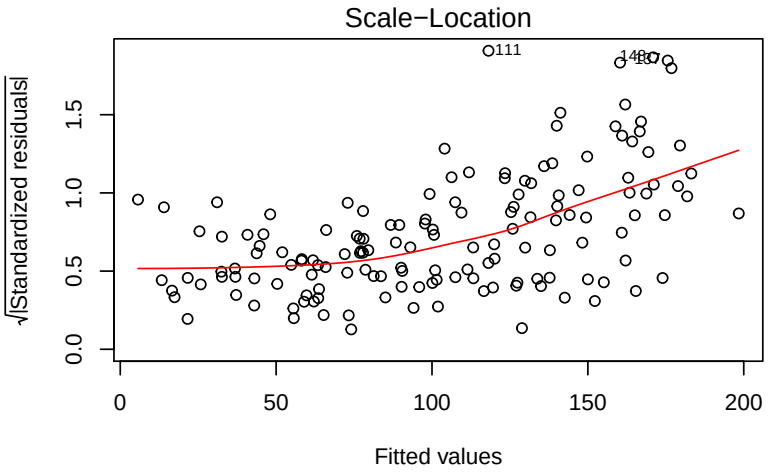

Scale−Location
√|Standardized residuals|
Fitted values
40
125
37

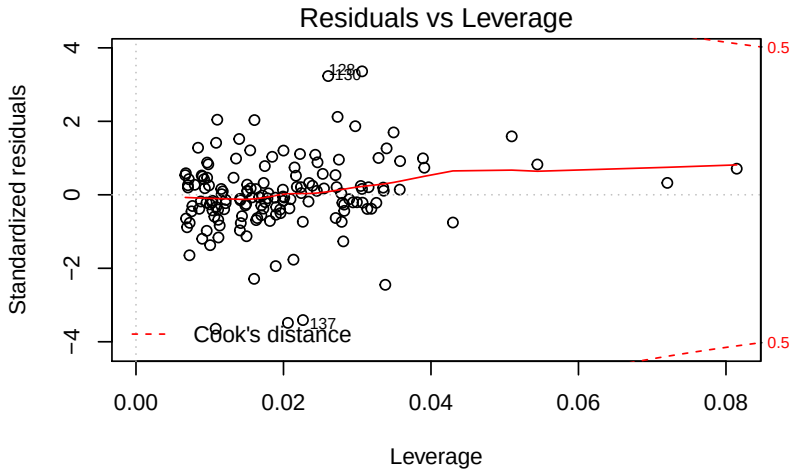

Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
40
131
37

OK, back to Example A

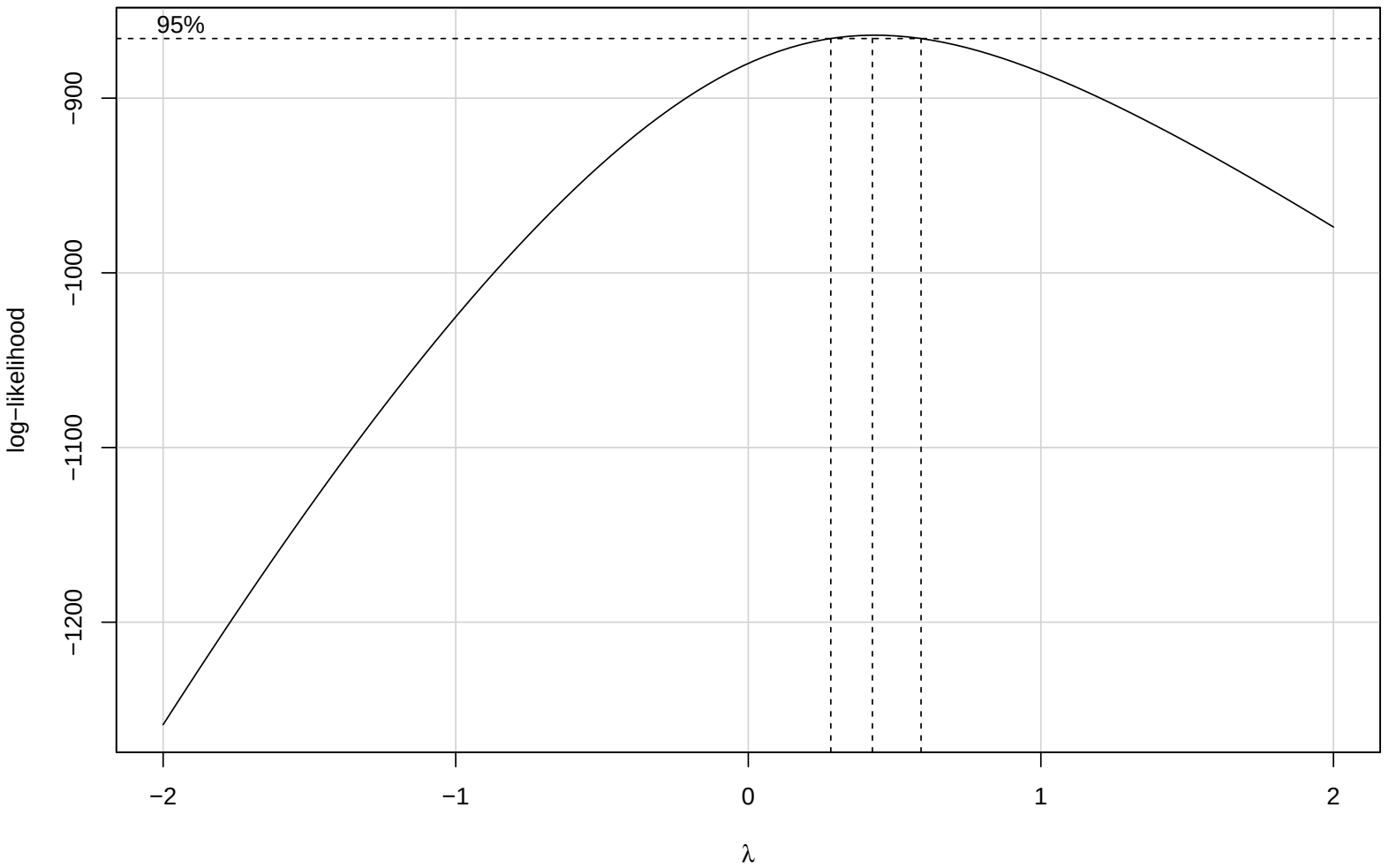




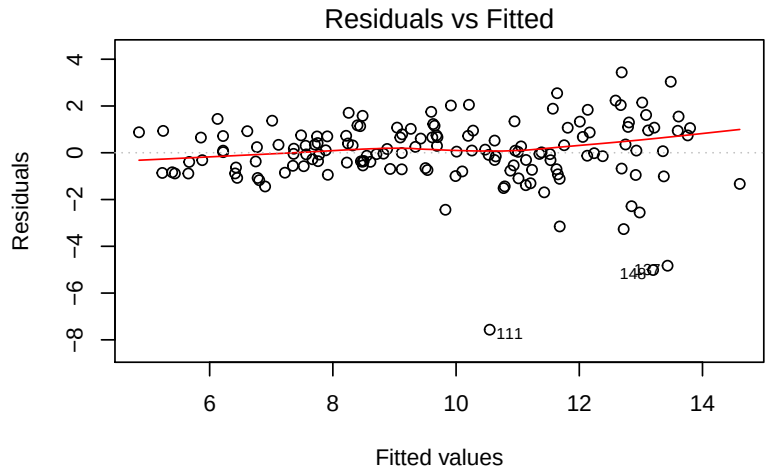


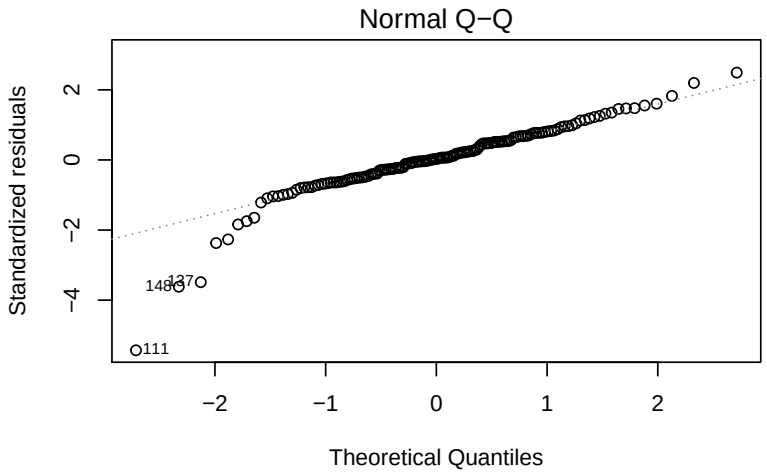


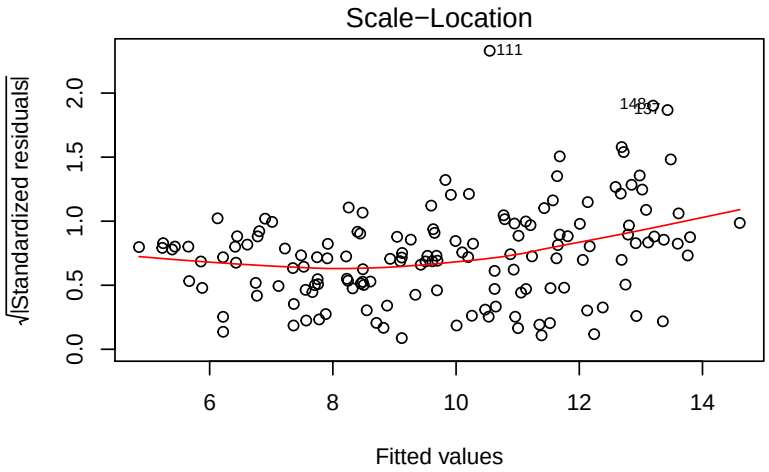
Box Cox for Model A

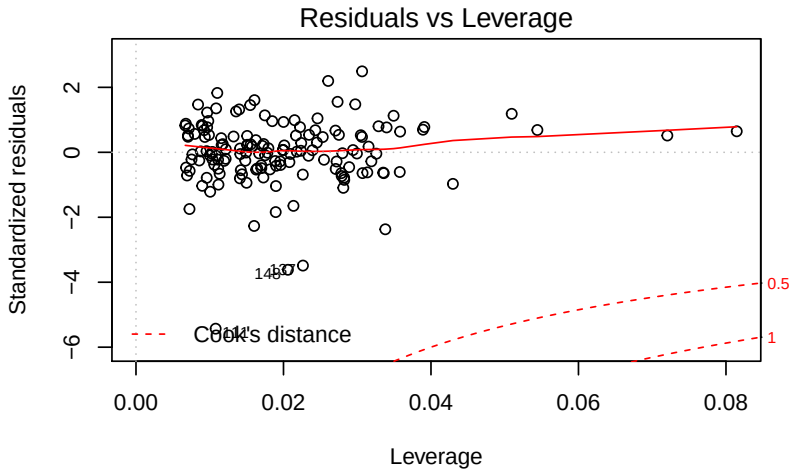


Model A, but with square root of Y

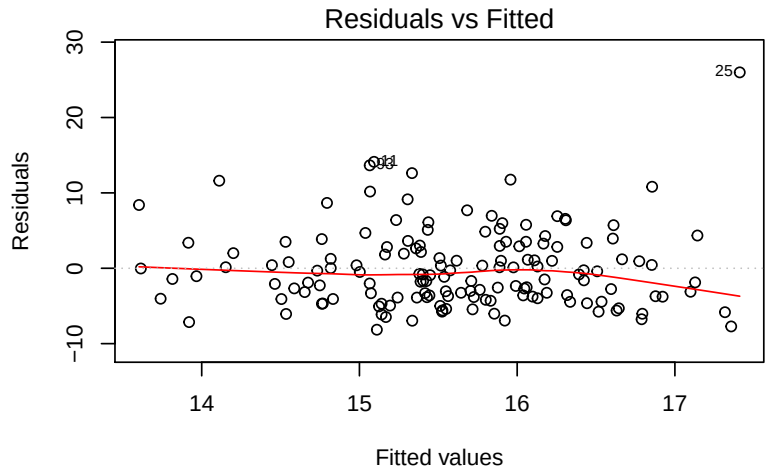


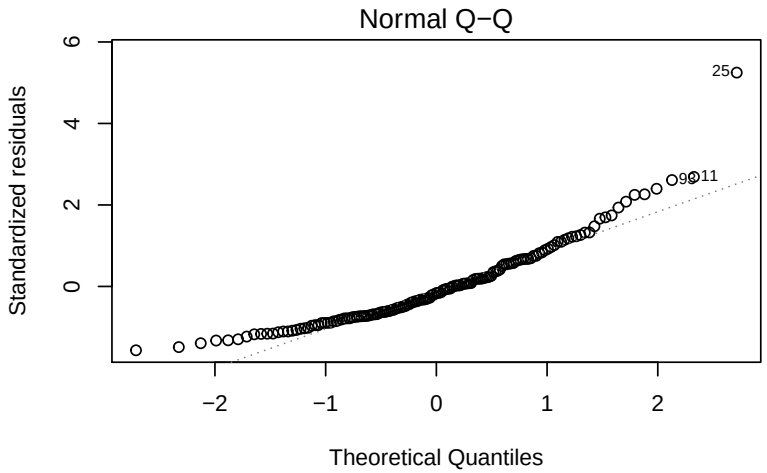


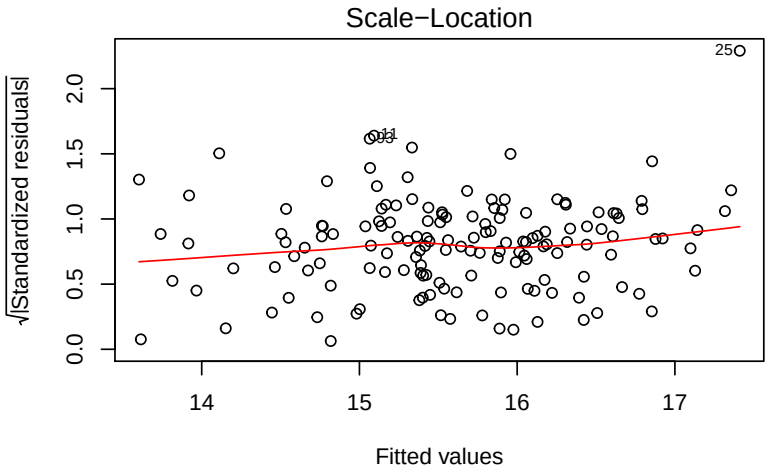


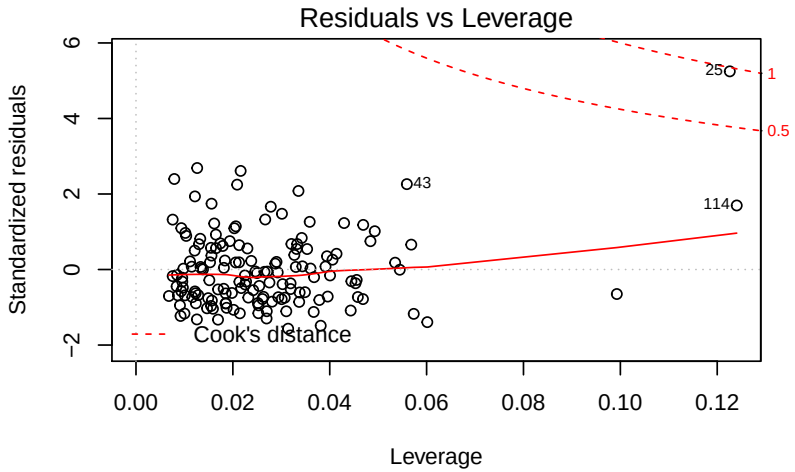


New Example B

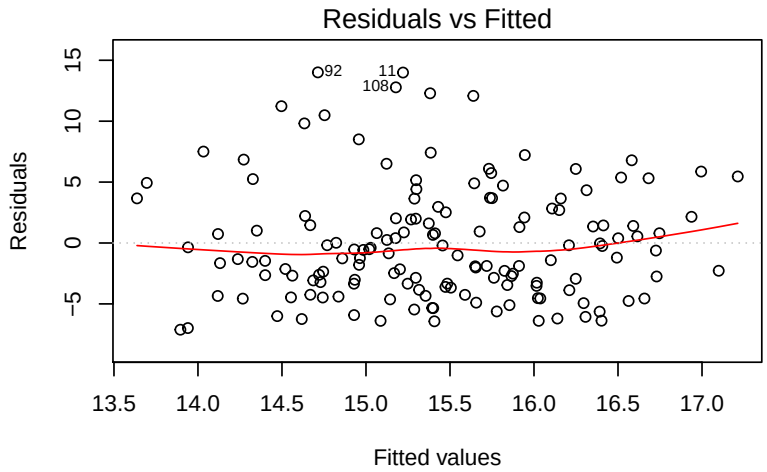


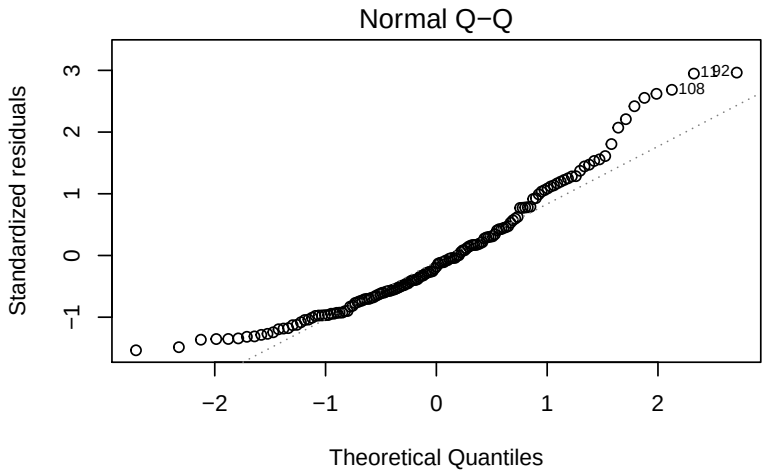


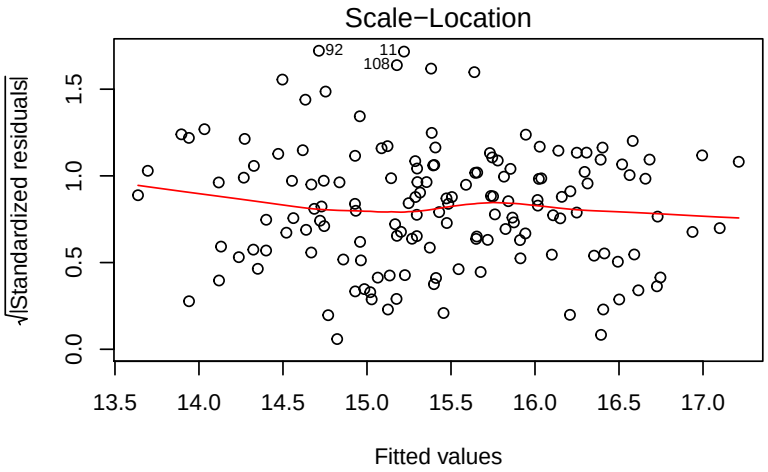


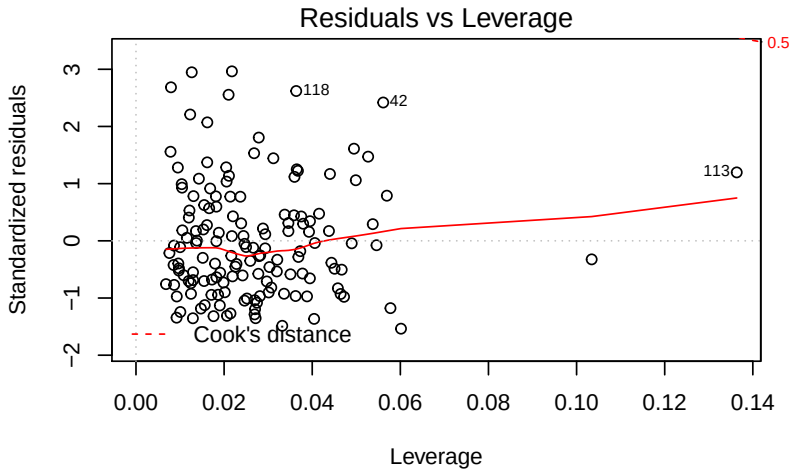


Example B without point 25

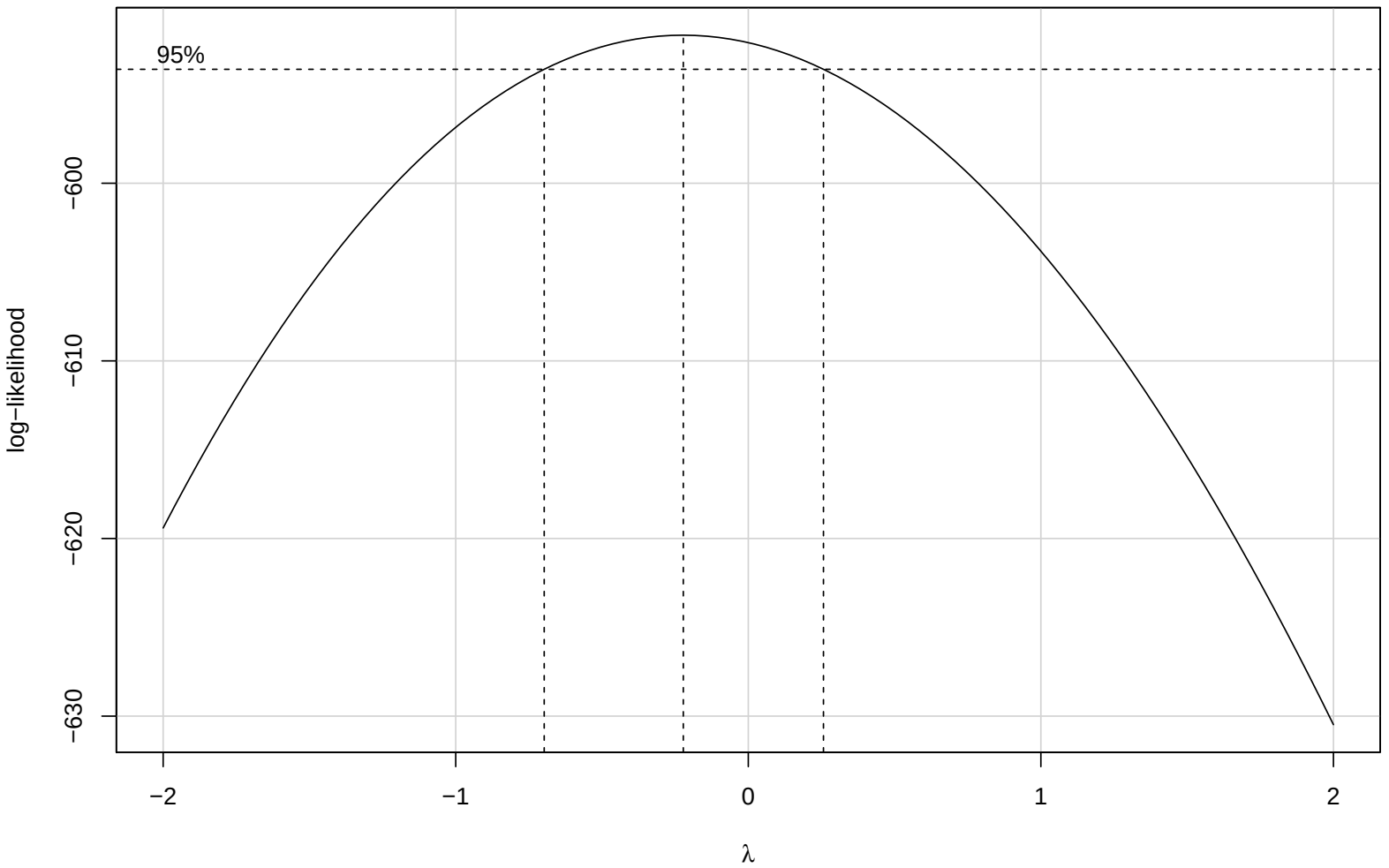




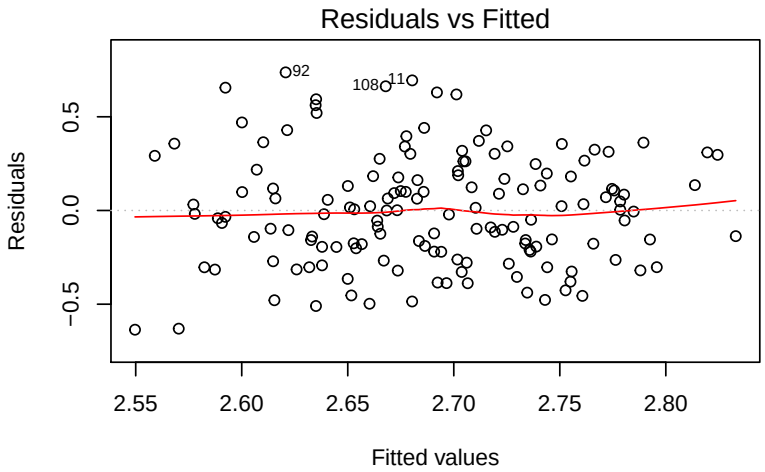


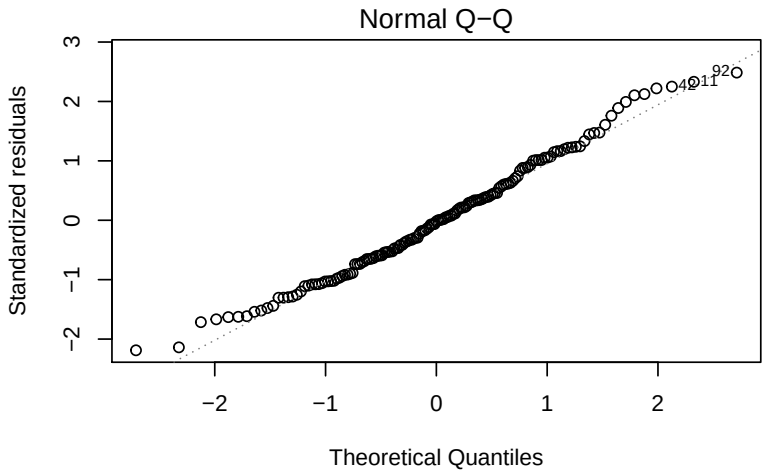


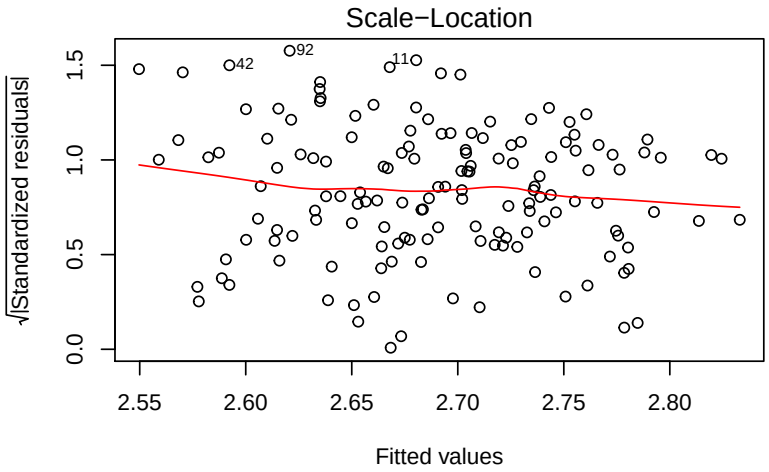
Box-Cox for new Model B (-25)

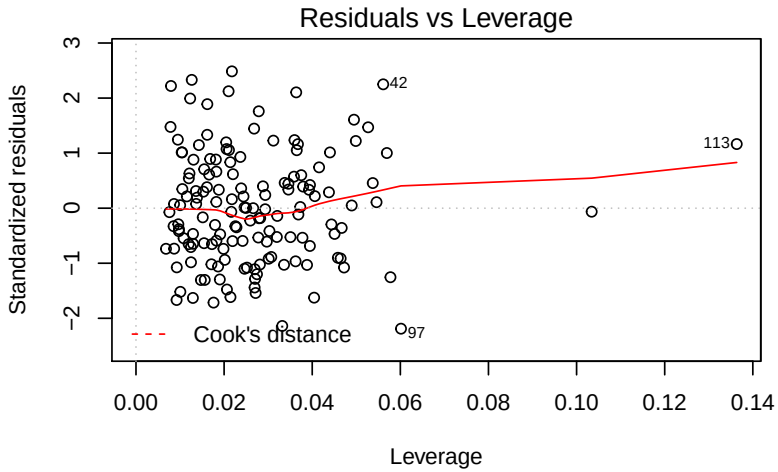


log(Y) model B (-25)

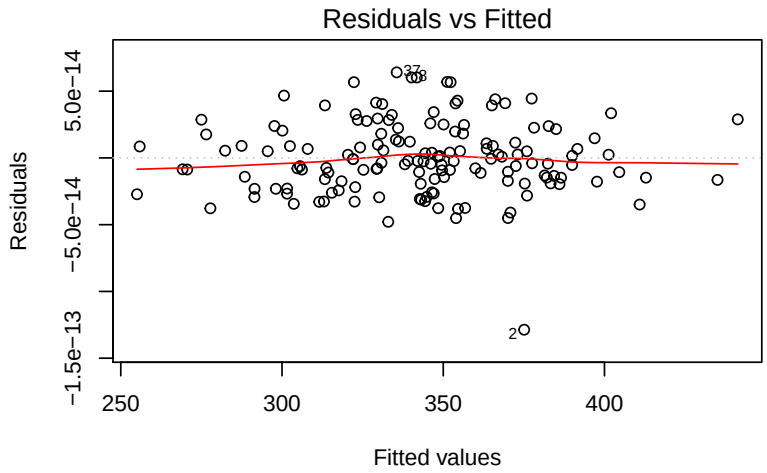


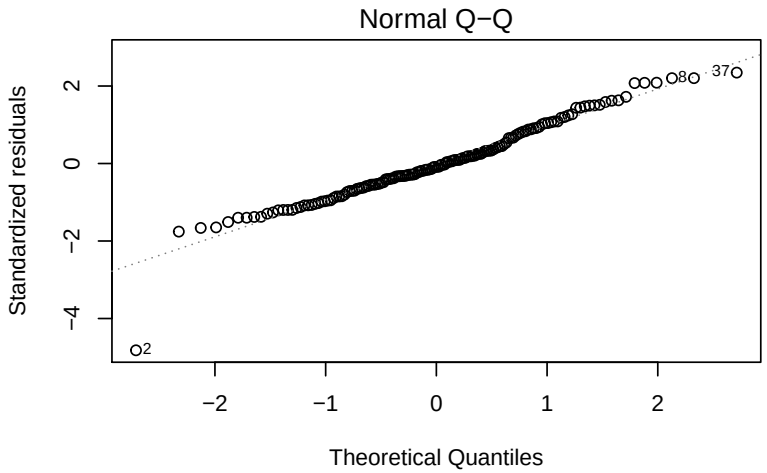


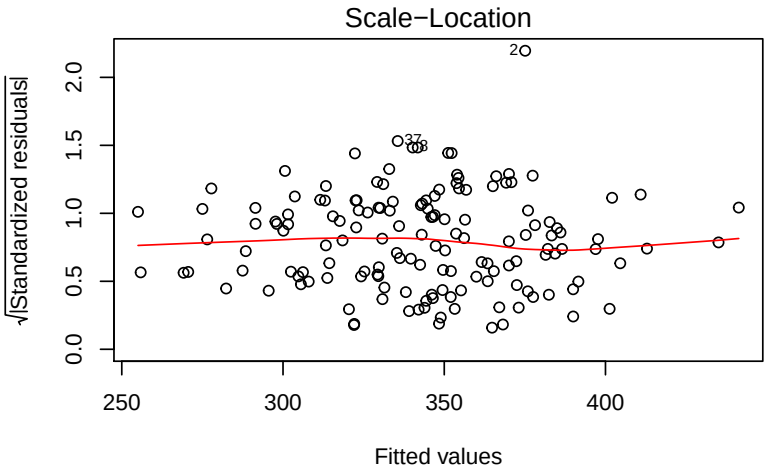


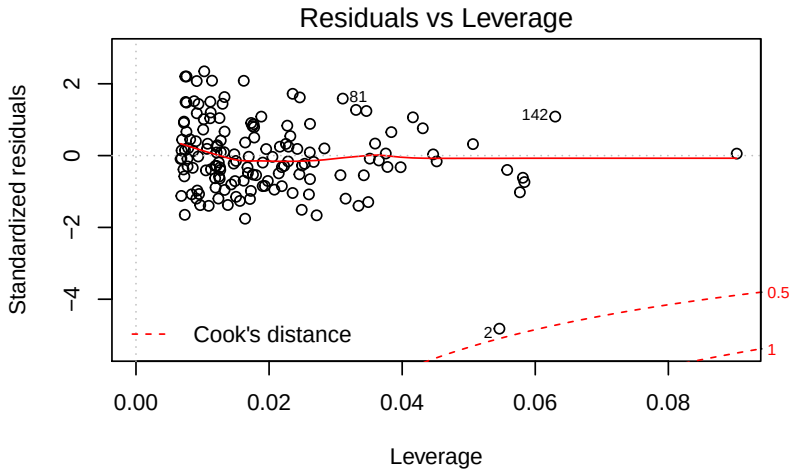


New Example C





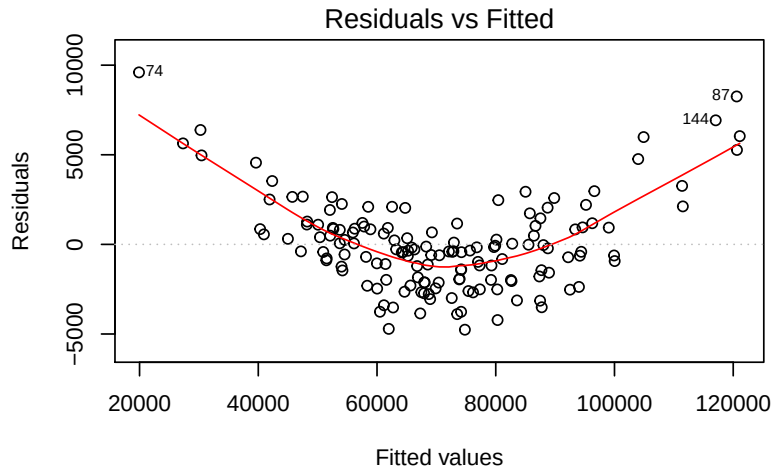


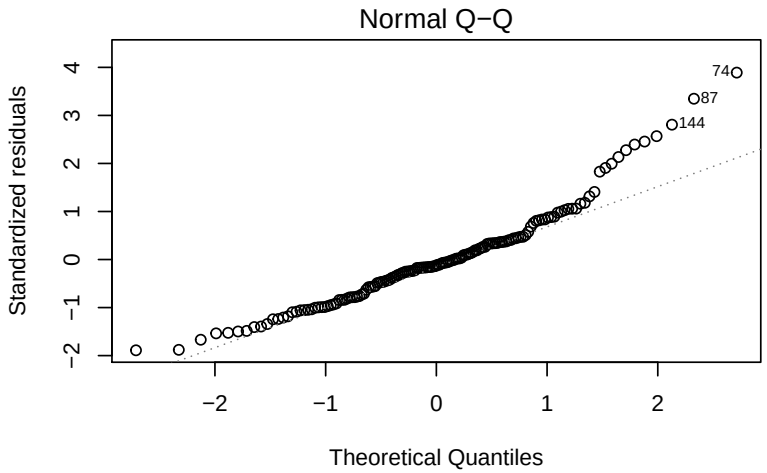


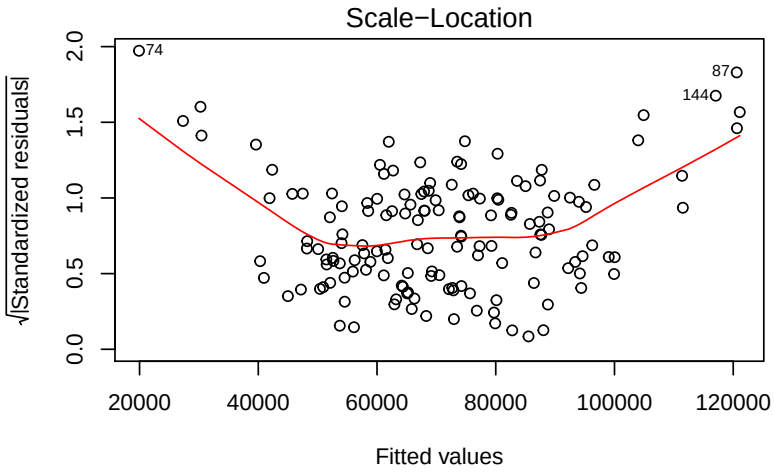
summary(mC)

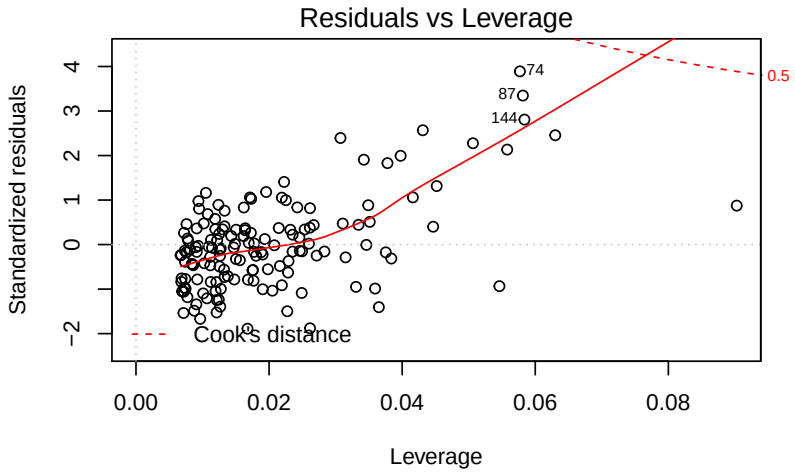
```
Warning message:
In summary.lm(mC) : essentially perfect fit:
summary may be unreliable

Coefficients:
             Estimate Std. Error   t value Pr(>|t|)
(Intercept) 7.426e-14  2.275e-14 3.264e+00  0.00137 **
x1          3.000e+00  2.020e-16 1.485e+16  < 2e-16 ***
x2          2.000e+00  4.526e-16 4.419e+15  < 2e-16 ***

Residual standard error: 2.745e-14 on 147 df
Multiple R-squared:      1, Adjusted R-squared:      1
F-stat: 1.176e+32 on 2 and 147 DF,  p-value: < 2.2e-16
```

New Example D

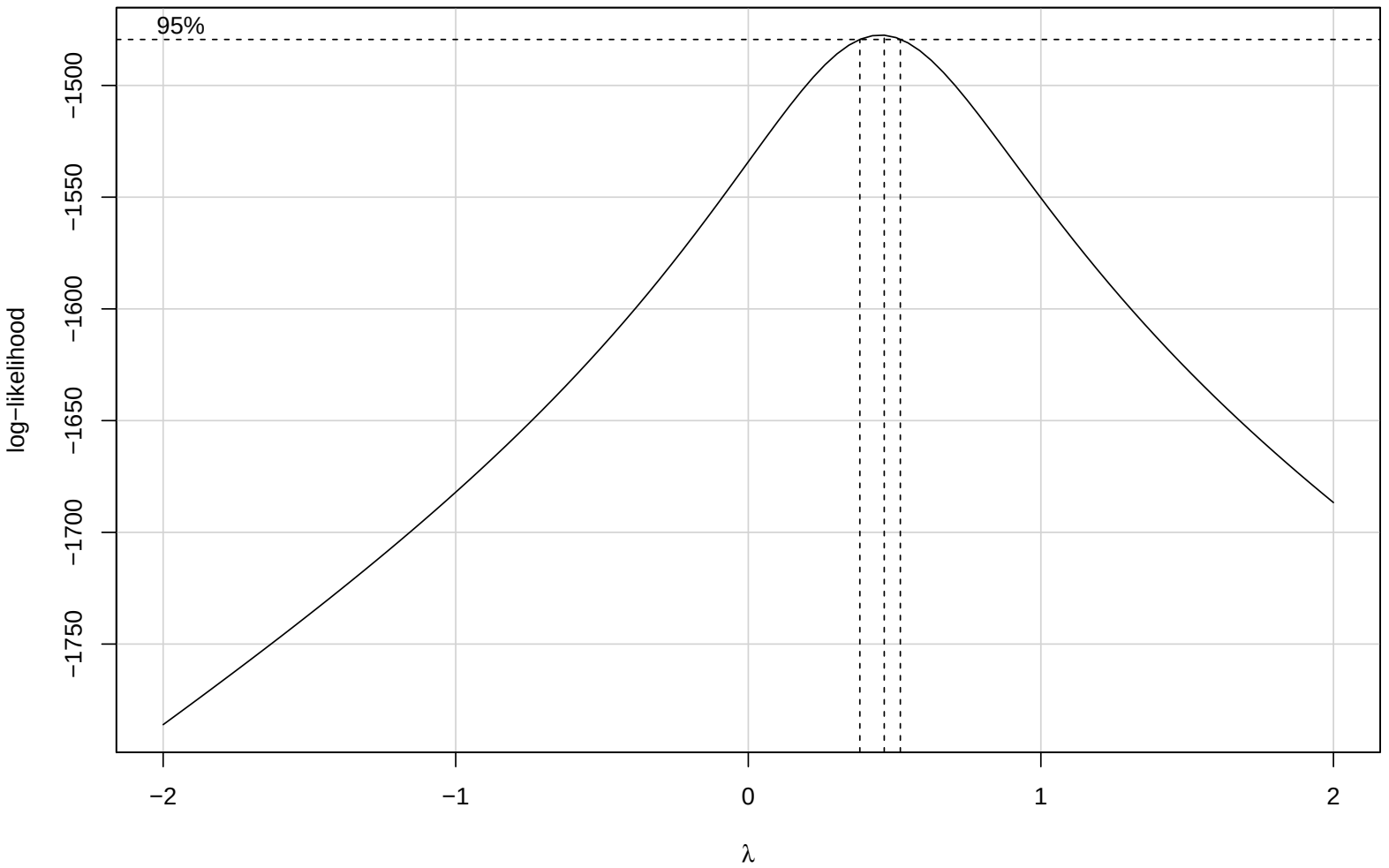




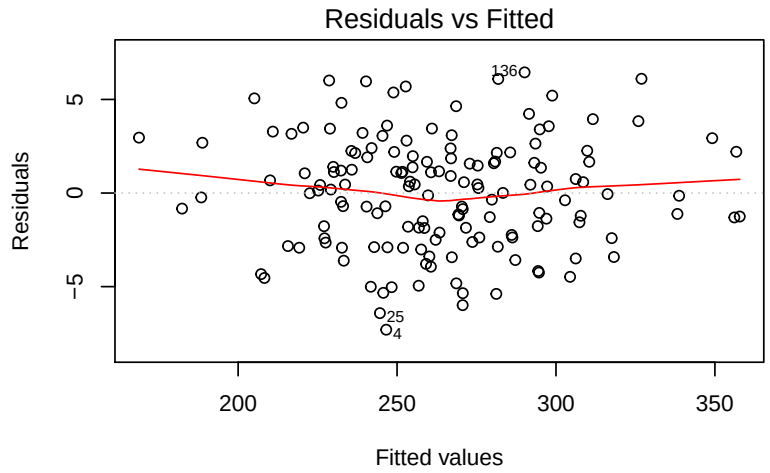


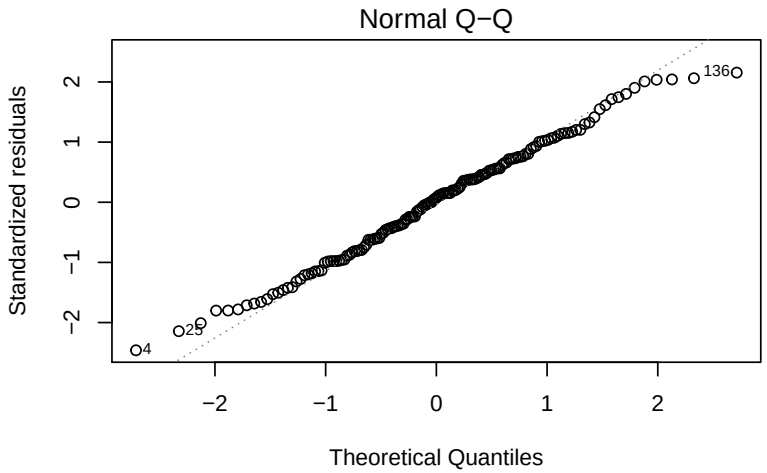


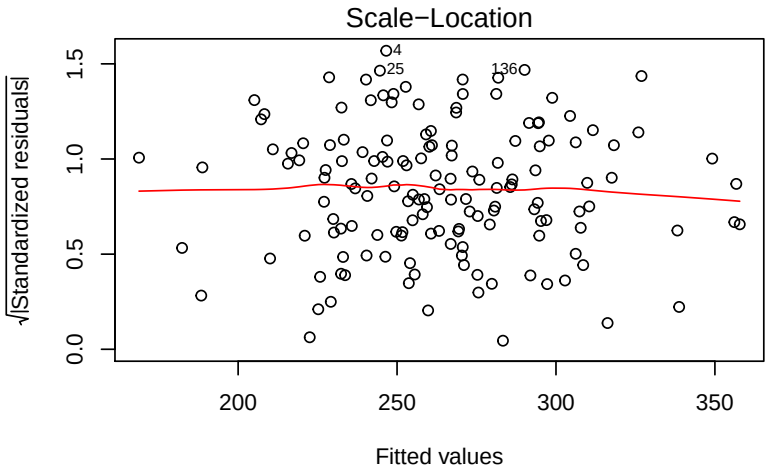
Box-Cox for Model D

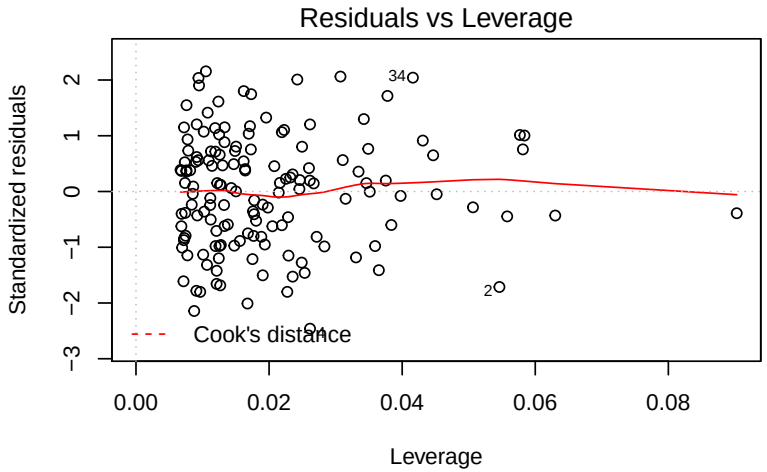


New Example D (sqrt of Y)

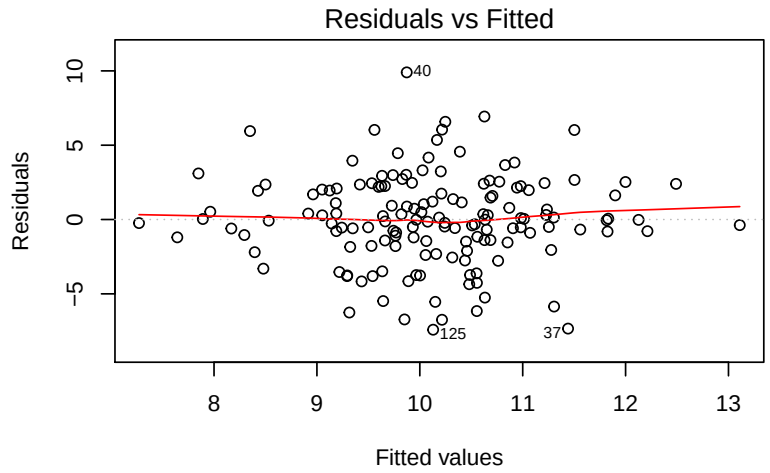


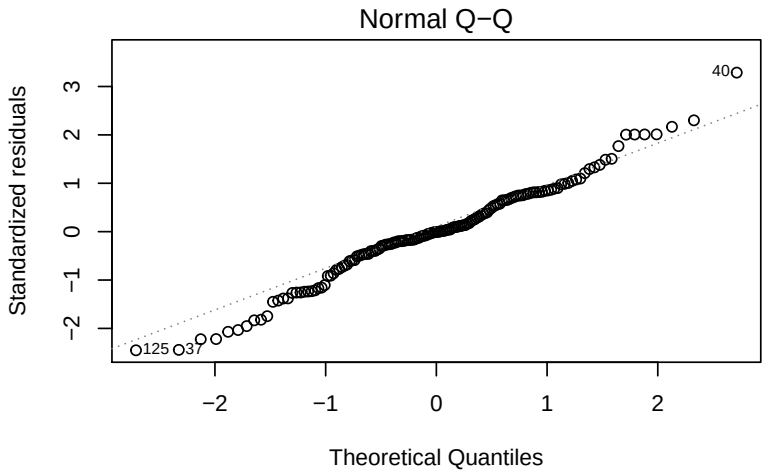


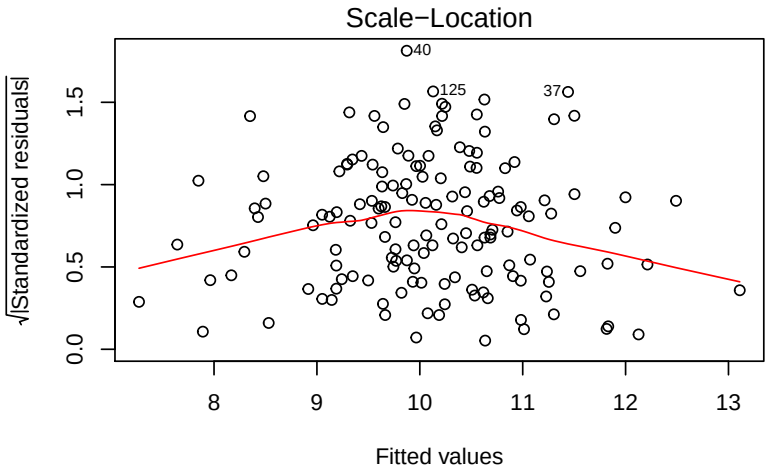


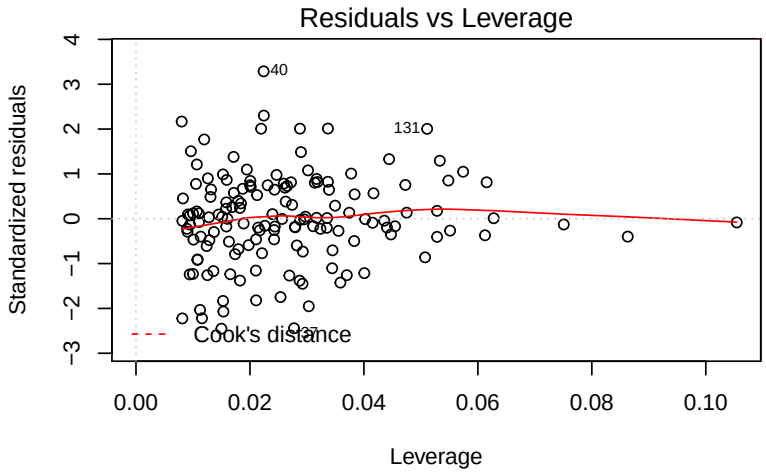


New Example E









Regression and the dm192 data

Our Research Question

Can we predict a patient's `sbp` level today, if the seven features we can use to predict that are:

- their `sbp` level one year ago
- their `a1c` level now
- their `age`, `race`, `sex` and `insurance` type
- and the `practice` where they are seen

We want to use some or all of these seven regression inputs to do the best possible job of predicting today's `sbp`, regardless of which predictors fall in or out of the model.

The dm192 data

```
head(dm192, 4)
```

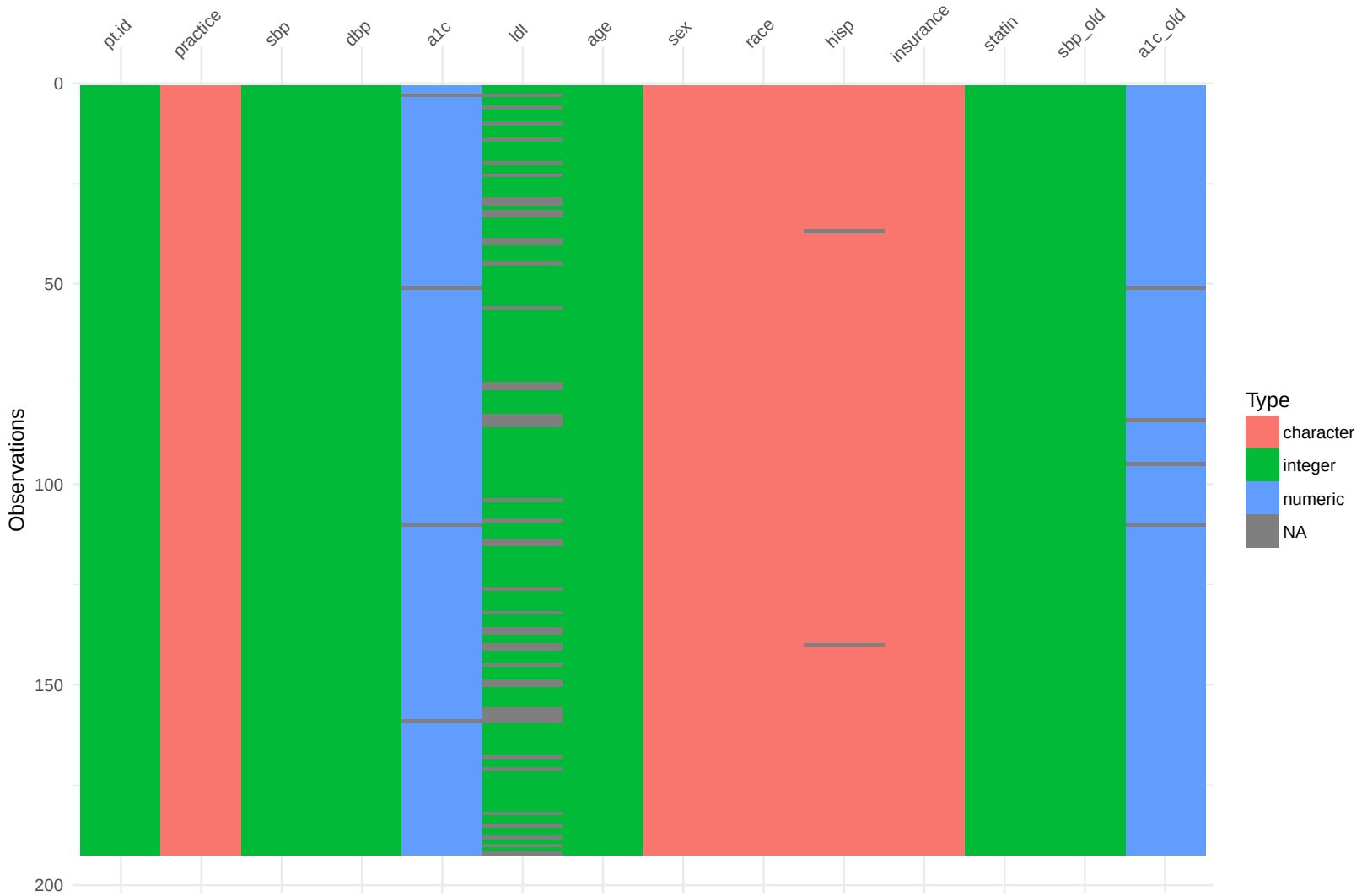
```
# A tibble: 4 x 14
  pt.id practice   sbp   dbp   a1c   ldl   age     sex
  <int>    <chr> <int> <int> <dbl> <int> <int>   <chr>
1     1        A   108    71   5.8    58    44    male
2     2        A   162    92  11.6    54    28  female
3     3        B   135    84    NA    NA    58  female
4     4        C   133    87  12.7   112    56    male
# ... with 6 more variables: race <chr>, hisp <chr>,
#   insurance <chr>, statin <int>, sbp_old <int>,
#   a1c_old <dbl>
```

- We may want to change some of those chr variables to factors.
- We probably want to address the missingness, too.

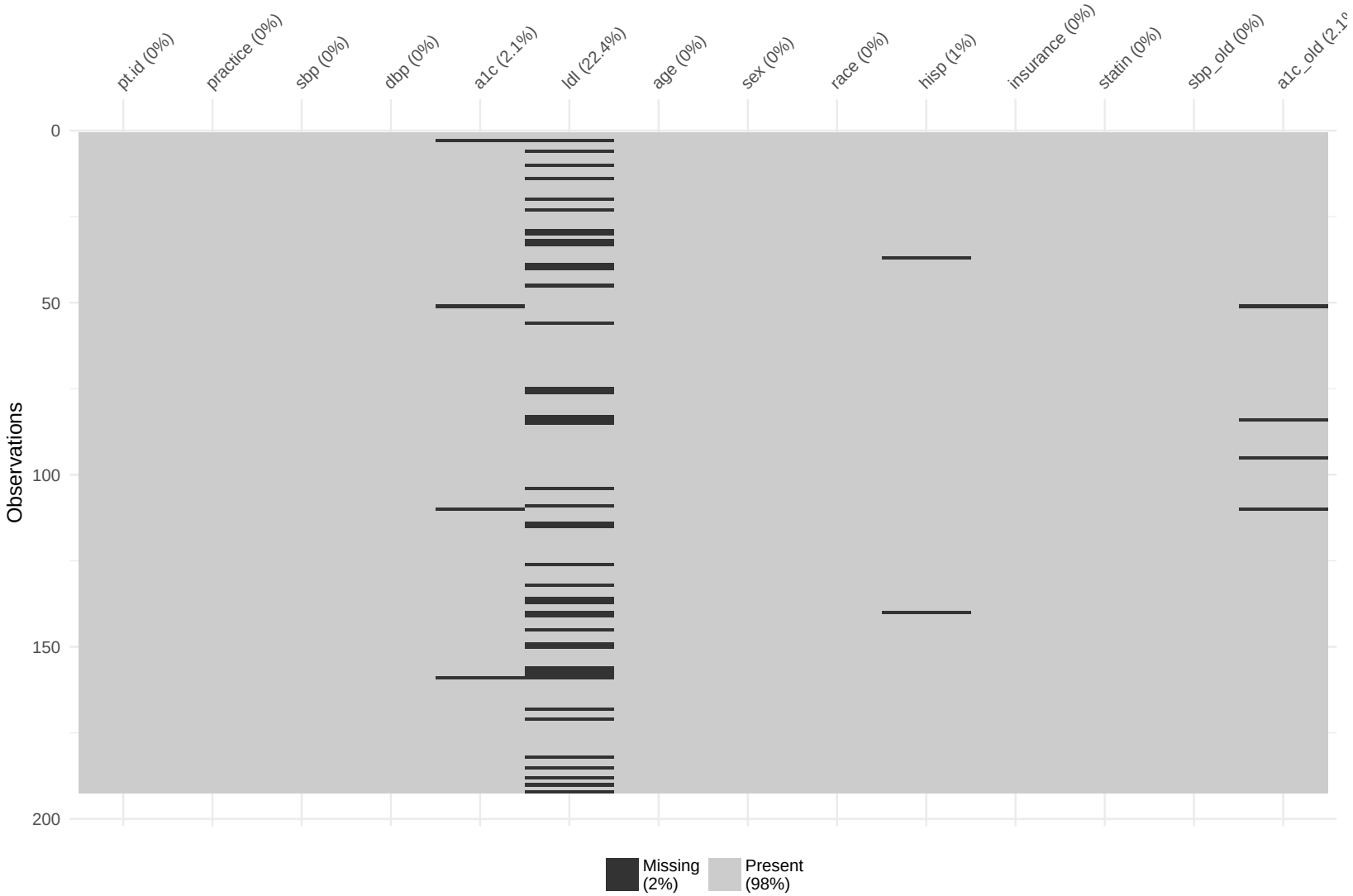
visdat to get a first look?

```
## assumes visdat package is installed from CRAN
library(visdat)
vis_dat(dm192, sort_type = FALSE)
```

vis_dat(dm192, sort_type = FALSE)



vis_miss(dm192)



For 431, we're working with complete cases only

```
dm192_work <- dm192 %>%
  select(pt.id, sbp, sbp_old, a1c, age, race,
         sex, insurance, practice) %>%
  filter(complete.cases(.))

head(dm192_work,3)
```

```
# A tibble: 3 x 9
  pt.id   sbp sbp_old   a1c   age  race    sex
  <int> <int>   <int> <dbl> <int> <chr>  <chr>
1     1   108     110   5.8    44 black   male
2     2   162     158  11.6    28 black female
3     4   133     145  12.7    56 black   male
# ... with 2 more variables: insurance <chr>,
#   practice <chr>
```

Change several character variables to factors

```
cols_temp <- c("race", "sex", "insurance", "practice")

dm192_work[cols_temp] <- lapply(dm192_work[cols_temp], factor)

head(dm192_work,3)
```

```
# A tibble: 3 x 9
  pt.id   sbp sbp_old   a1c   age   race    sex
  <int> <int>   <int> <dbl> <int> <fctr> <fctr>
1     1   108     110   5.8    44  black   male
2     2   162     158  11.6    28  black female
3     4   133     145  12.7    56  black   male
# ... with 2 more variables: insurance <fctr>,
#   practice <fctr>
```

Are the factor levels sensible and sensibly ordered? (1)

```
dm192_work %>% count(race)
```

```
# A tibble: 4 x 2
    race     n
  <fctr> <int>
1  asian     5
2  black   119
3  other    16
4  white    48
```

Auto-collapse to most common 2 levels, plus "Others"

```
dm192_work$race <- dm192_work$race %>%
  fct_lump(n = 2, other_level = "Others")

table(dm192_work$race)
```

```
 black  white Others
   119     48     21
```

Are the factor levels sensible and sensibly ordered? (2)

```
dm192_work %>% count(sex)
```

```
# A tibble: 2 x 2
     sex     n
  <fctr> <int>
1 female    96
2   male    92
```

Are the factor levels sensible and sensibly ordered? (3)

```
dm192_work %>% count(insurance)
```

```
# A tibble: 4 x 2
   insurance     n
      <fctr> <int>
1 commercial    39
2   medicaid    67
3   medicare    76
4  uninsured     6
```

Collapse Medicaid and Uninsured together

```
dm192_work$insurance <-
  fct_collapse(dm192_work$insurance,
               Medicare = "medicare",
               Commercial = "commercial",
               Medicaid_Unins = c("medicaid", "uninsured"))

table(dm192_work$insurance)
```

```
    Commercial Medicaid_Unins       Medicare
            39             73             76
```

Reorder Factor Levels by Hand

```
dm192_work$insurance <-
  fct_relevel(dm192_work$insurance,
              "Medicare", "Commercial")

table(dm192_work$insurance)
```

```
     Medicare     Commercial Medicaid_Unins
           76             39             73
```

Are the factor levels sensible and sensibly ordered? (4)

```
dm192_work %>% count(practice)
```

```
# A tibble: 4 x 2
  practice     n
    <fctr> <int>
1        A    48
2        B    45
3        C    47
4        D    48
```

Predict `sbp` as well as you can, in new data

Stage 1. Partition the Data

```
set.seed(43123)
dm192_train <-
  sample_frac(dm192_work, 0.8, replace = FALSE)
dm192_test <-
  anti_join(dm192_work, dm192_train)
```

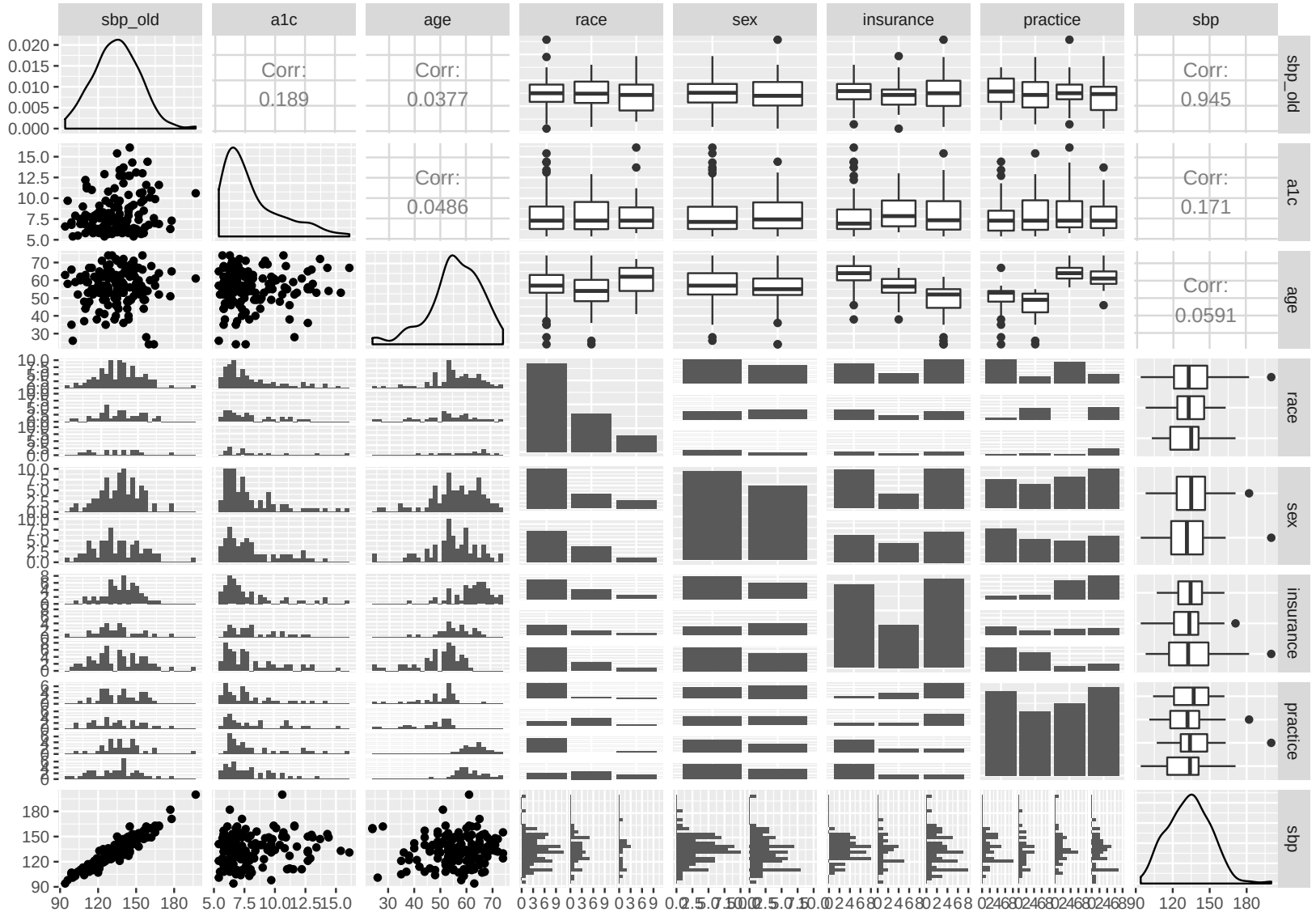
```
Joining, by = c("pt.id", "sbp", "sbp_old", "a1c", "age", "race
```

```
dim(dm192_train); dim(dm192_test)
```

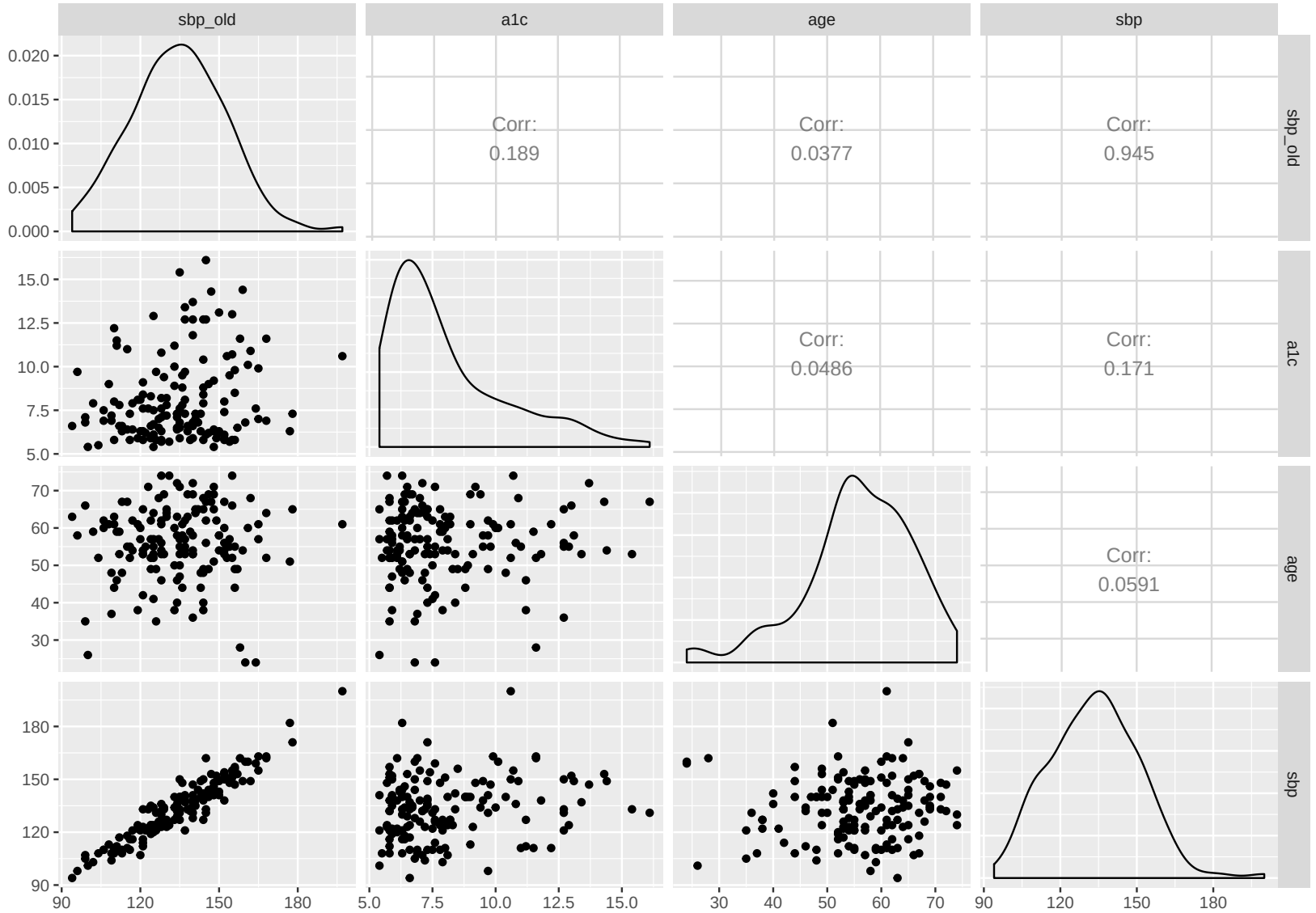
```
[1] 150   9
```

```
[1] 38  9
```

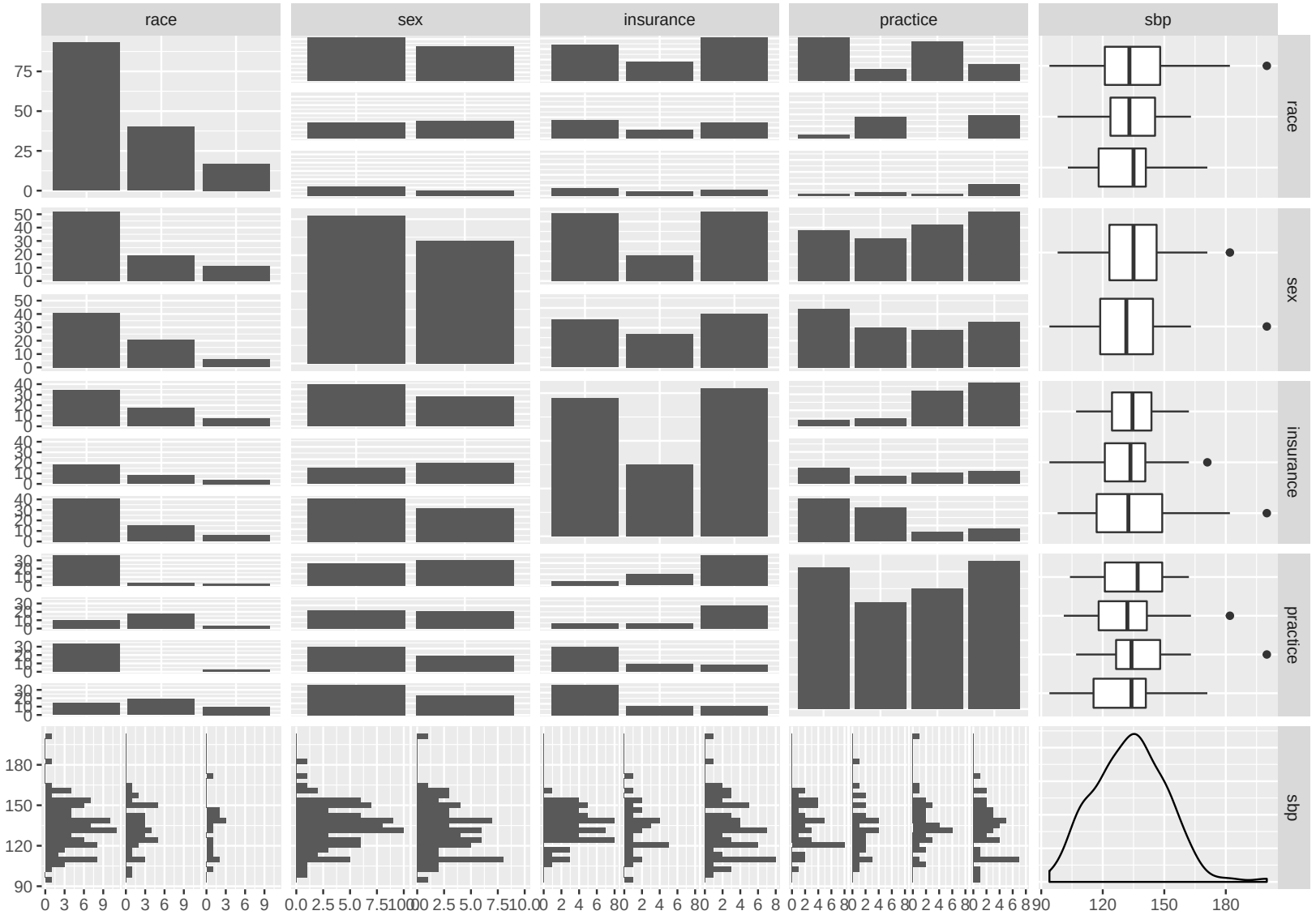
Stage 2. DTDP (everything in training set)



Stage 2. DTDP (quantitative predictors)



Stage 2. DTDP (categorical predictors)



Stage 3. Exploratory Data Analysis

```
mosaic::favstats(dm192_train$sbp)
```

```
 min     Q1 median    Q3 max   mean       sd   n
  94 121.25    133 145.5 200 133.42 17.48605 150
 missing
       0
```

```
mosaic::favstats(dm192_train$sbp ~ dm192_train$sex)
```

```
  dm192_train$sex min     Q1 median     Q3 max
1          female  98 123.25  135.0 146.25 182
2            male  94 118.75  131.5 144.50 200
      mean       sd  n missing
1 134.6463 16.19966 82       0
2 131.9412 18.93814 68       0
```

Stage 4. Fit Kitchen Sink Model in Training Sample

```
mod_ks1 <- lm(sbp ~ sbp_old + a1c + age + race +
                sex + insurance + practice,
              data = dm192_train)

round(glance(mod_ks1),3)
```

```
  r.squared adj.r.squared sigma statistic p.value df
1     0.898          0.89 5.791   110.954       0 12
    logLik     AIC      BIC deviance df.residual
1 -470.034 966.067 1005.206  4627.97         138
```

arm::display(mod_ks1) (n = 150, r-sq = 0.90)

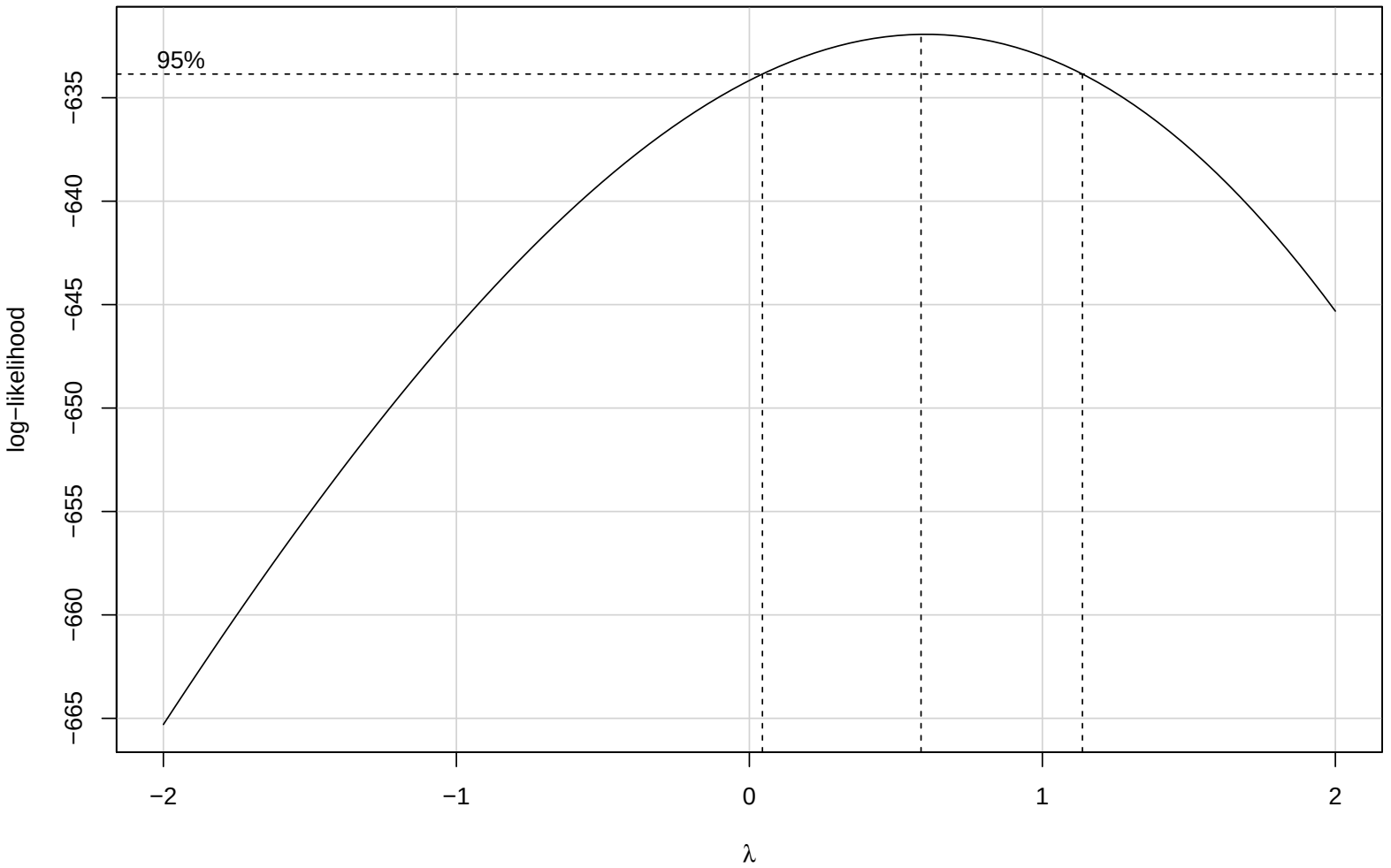
```
lm(formula = sbp ~ sbp_old + a1c + age + race + sex + insuranc
    practice, data = dm192_train)
                          coef.est coef.se
(Intercept)                6.70     5.70
sbp_old                    0.93     0.03
a1c                       -0.09     0.21
age                        0.02     0.08
racewhite                 -1.16     1.39
raceOthers                -1.16     1.66
sexmale                   -0.67     0.97
insuranceCommercial        1.59     1.40
insuranceMedicaid_Unins    1.85     1.38
practiceB                  1.29     1.59
practiceC                  2.27     1.73
practiceD                  2.91     1.75
---
n = 150, k = 12
```

Stage 5. Consider collinearity, residual plots, potential transformations of the outcome

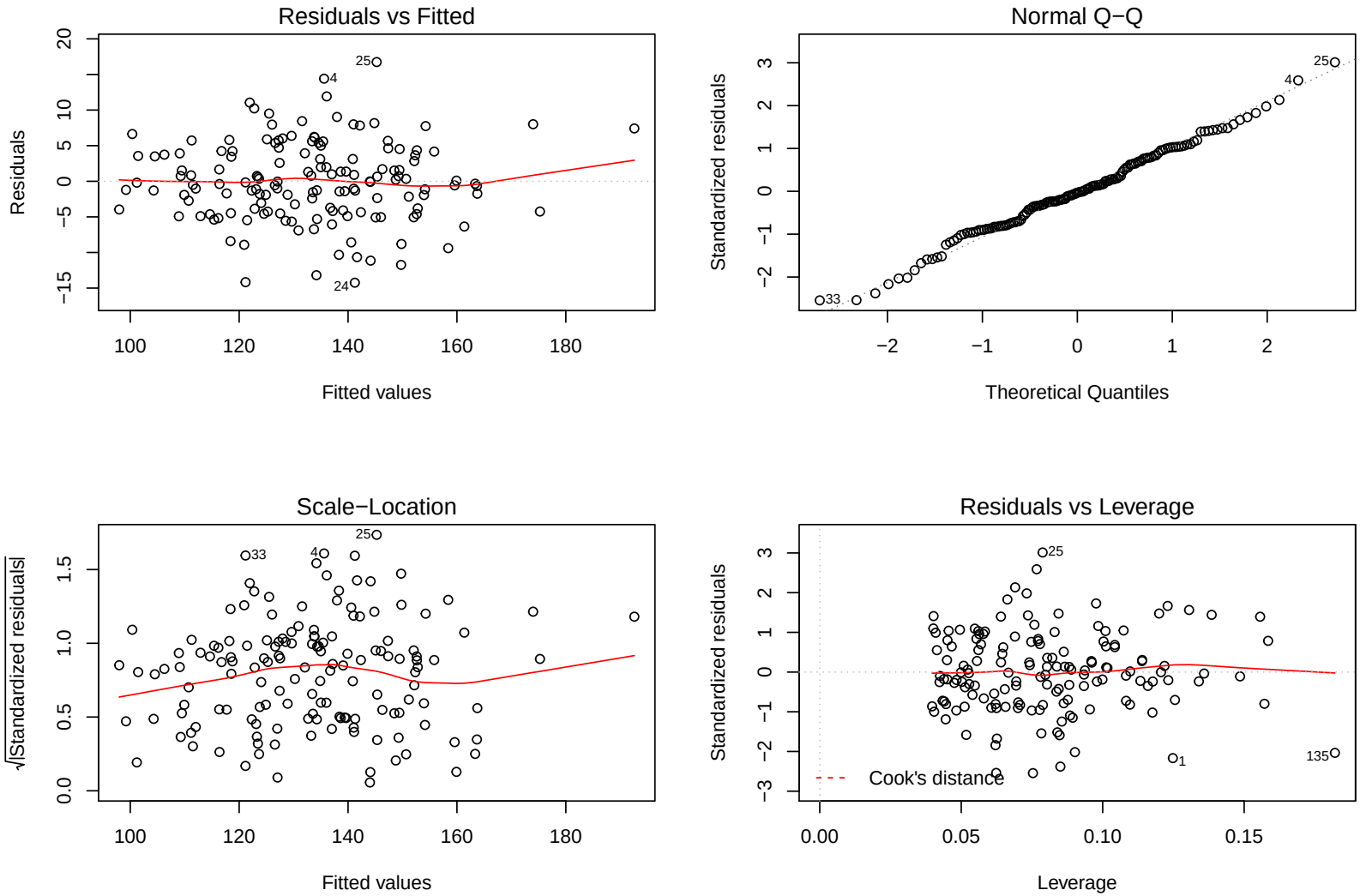
```
vif(mod_ks1)
```

```
                GVIF Df GVIF^(1/(2*Df))
sbp_old     1.082669  1         1.040514
a1c         1.075402  1         1.037016
age         2.645307  1         1.626440
race        1.745582  2         1.149437
sex         1.052235  1         1.025785
insurance   1.757957  2         1.151468
practice    4.032427  3         1.261618
```

boxCox(mod_ks1) ($\lambda = 0.6$, round to 1)



plot(mod_ks1)


Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance

Stage 6. Consider stepwise regression to prune the model

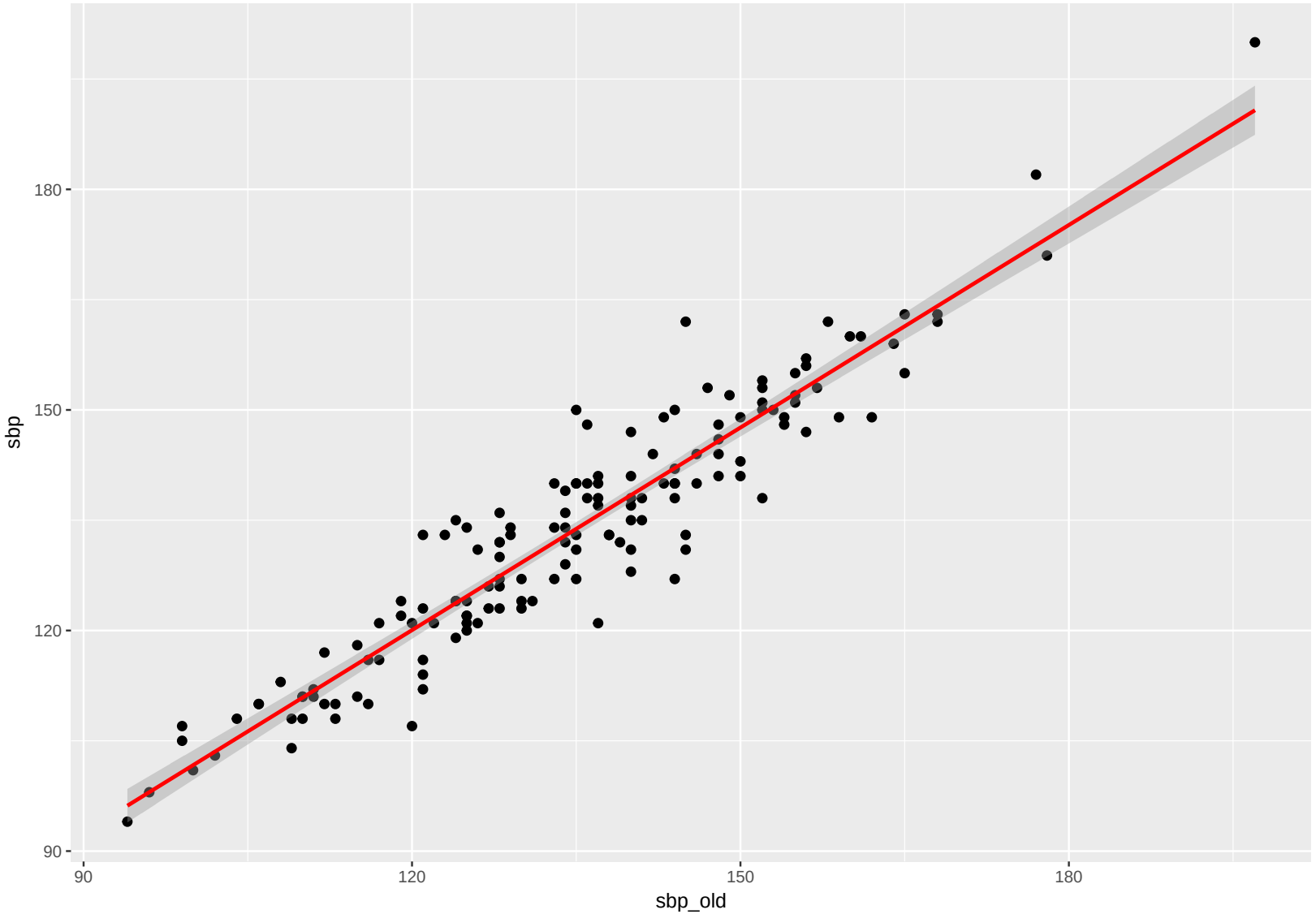
```
step(mod_ks1)
```

```
Start:  AIC=538.39
sbp ~ sbp_old + a1c + age + race + sex + insurance + practice

             Df Sum of Sq   RSS    AIC
- race        2         30  4658 535.35
- practice    3         98  4726 535.53
- age         1          1  4629 536.43
- a1c         1          7  4635 536.61
- insurance   2         69  4697 536.61
- sex         1         16  4644 536.89
<none>                     4628 538.39
- sbp_old     1      38268 42896 870.39
```

Suggested model from `step` is

```
Step:  AIC=524.98
sbp ~ sbp_old

           Df Sum of Sq   RSS    AIC
<none>                    4836 524.98
- sbp_old  1     40722 45559 859.42

Call:
lm(formula = sbp ~ sbp_old, data = dm192_train)

Coefficients:
(Intercept)      sbp_old
     9.8485       0.9183
```

So that's just . . .



Stage 7. Compare potential models in-sample

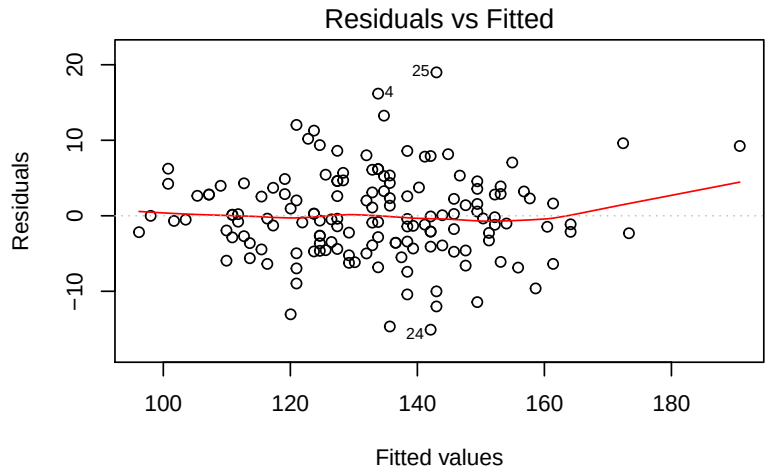
```
mod_simple <- lm(sbp ~ sbp_old, data = dm192_train)
glance(mod_simple) %>% select(r.squared, adj.r.squared, AIC, B
```

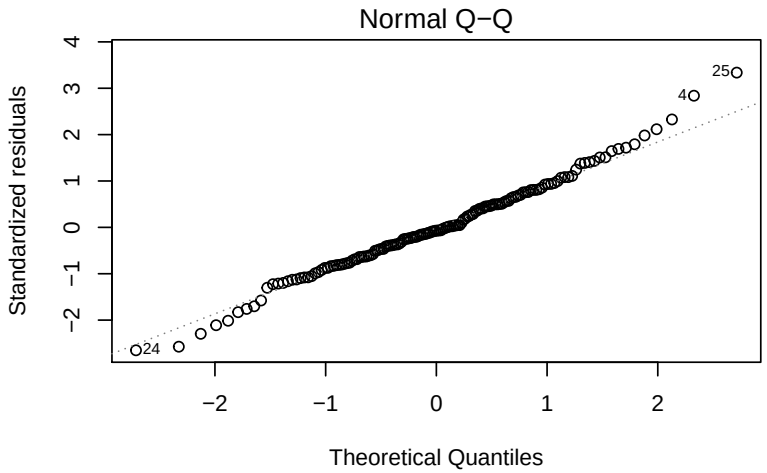
```
  r.squared adj.r.squared      AIC     BIC
1 0.8938499     0.8931326 952.6641 961.696
```

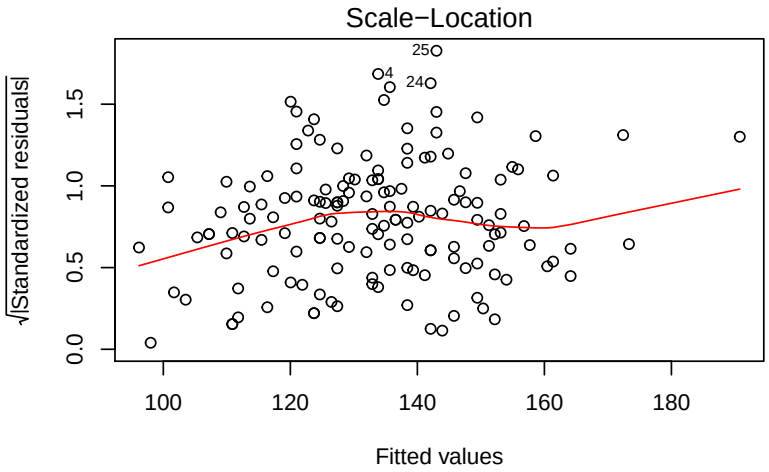
```
glance(mod_ks1) %>% select(r.squared, adj.r.squared, AIC, BIC)
```

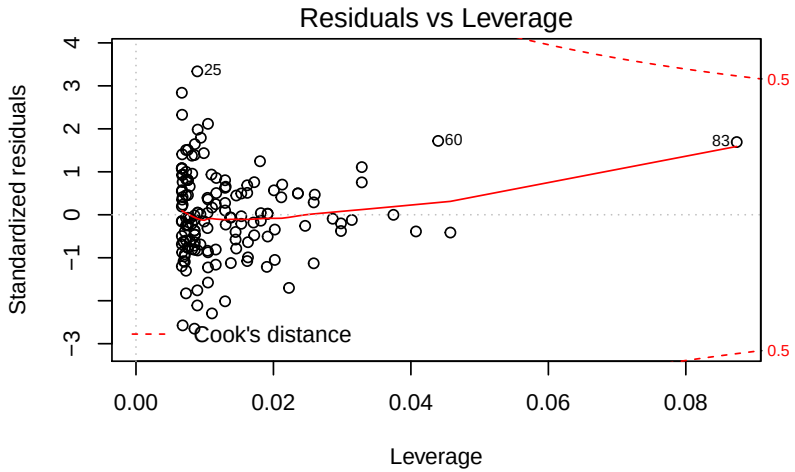
```
  r.squared adj.r.squared      AIC      BIC
1 0.8984171     0.8903199 966.0673 1005.206
```

Residual Plots for Simple One-Predictor Model









Stage 8. Compare potential models on test data

```
pred_ks <- predict(mod_ks1, newdata = dm192_test)
err_ks <- dm192_test$sbp - pred_ks
summary(abs(err_ks))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.5776  2.8760  4.2873  5.1298  6.8480 14.4801
```

```
summary(err_ks^2)
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
  0.3336   8.2714  18.3960  37.1017  46.9061 209.6720
```

```
cor(pred_ks, dm192_test$sbp)^2
```

```
[1] 0.9162891
```

Simple Model

```
pred_simple <- predict(mod_simple, newdata = dm192_test)
err_simple <- dm192_test$sbp - pred_simple
summary(abs(err_simple))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.1507  2.3341  4.2984  4.9942  6.2981 14.2776
```

```
summary(err_simple^2)
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.
  0.02272   5.46800  18.47813  37.96536  39.67679
     Max.
203.85108
```

```
cor(pred_simple, dm192_test$sbp)^2
```

```
[1] 0.9149048
```

MAPE and MSPE results

Model	MAPE	MSPE	Max Abs. Error	Out of Sample R^2
Kitchen Sink	5.13	37.1	14.48	0.916
Simple	4.99	38	14.28	0.915

Remember that the training sample here has only 38 observations.

Stage 9. Re-combine sample and fit final model

```
model_all <- lm(sbp ~ sbp_old, data = dm192_work)

glance(model_all)
```

```
  r.squared adj.r.squared    sigma statistic
1 0.8962414     0.8956835 5.785442  1606.622
       p.value df    logLik      AIC      BIC deviance
1 1.907078e-93  2 -595.7599 1197.52 1207.229 6225.669
  df.residual
1         186
```

Tidied `model_all` Coefficients

```
tidy(model_all)
```

```
         term  estimate  std.error statistic
1 (Intercept) 7.1213074 3.19565828  2.228432
2     sbp_old 0.9410682 0.02347817 40.082692
       p.value
1 2.704951e-02
2 1.907078e-93
```

Residual Plot for `model_all`