431 Class 23

Thomas E. Love

2017-11-16

Today's Agenda

- Discussing Past Activities
  - Quiz 2
  - Assignment 5
  - "After Class 22" Google Form
- Changes in the Tidyverse
- Regression Modeling in the Tidyverse
  - The National Youth Fitness Survey

Today's R Setup and Data Set

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

```
-- Attaching packages ---------------------------------------
v ggplot2 2.2.1     v purrr   0.2.4
v tibble  1.3.4     v dplyr   0.7.4
v tidyr   0.7.2     v stringr 1.2.0
v readr   1.1.1     v forcats 0.2.0

-- Conflicts ------------------------------------------------
x tidyr::extract()   masks magrittr::extract()
x dplyr::filter()    masks stats::filter()
x dplyr::lag()       masks stats::lag()
x purrr::set_names() masks magrittr::set_names()
```

```
nyfs1 <- read.csv("data/nyfs1.csv") %>% tbl_df
```

Regression, or "You've Got To Draw The Line Somewhere"

The National Youth Fitness Survey, 2012

See https://thomaselove.github.io/431notes/NYFS-Study.html for details.

```
summary(select(nyfs1, bmi, waist.circ, sex))
```

```
      bmi          waist.circ          sex
 Min.   :11.9   Min.   : 42.50   Female:707
 1st Qu.:15.8   1st Qu.: 55.00   Male  :709
 Median :17.7   Median : 63.00
 Mean   :18.8   Mean   : 65.29
 3rd Qu.:20.9   3rd Qu.: 72.92
 Max.   :38.8   Max.   :112.40
```

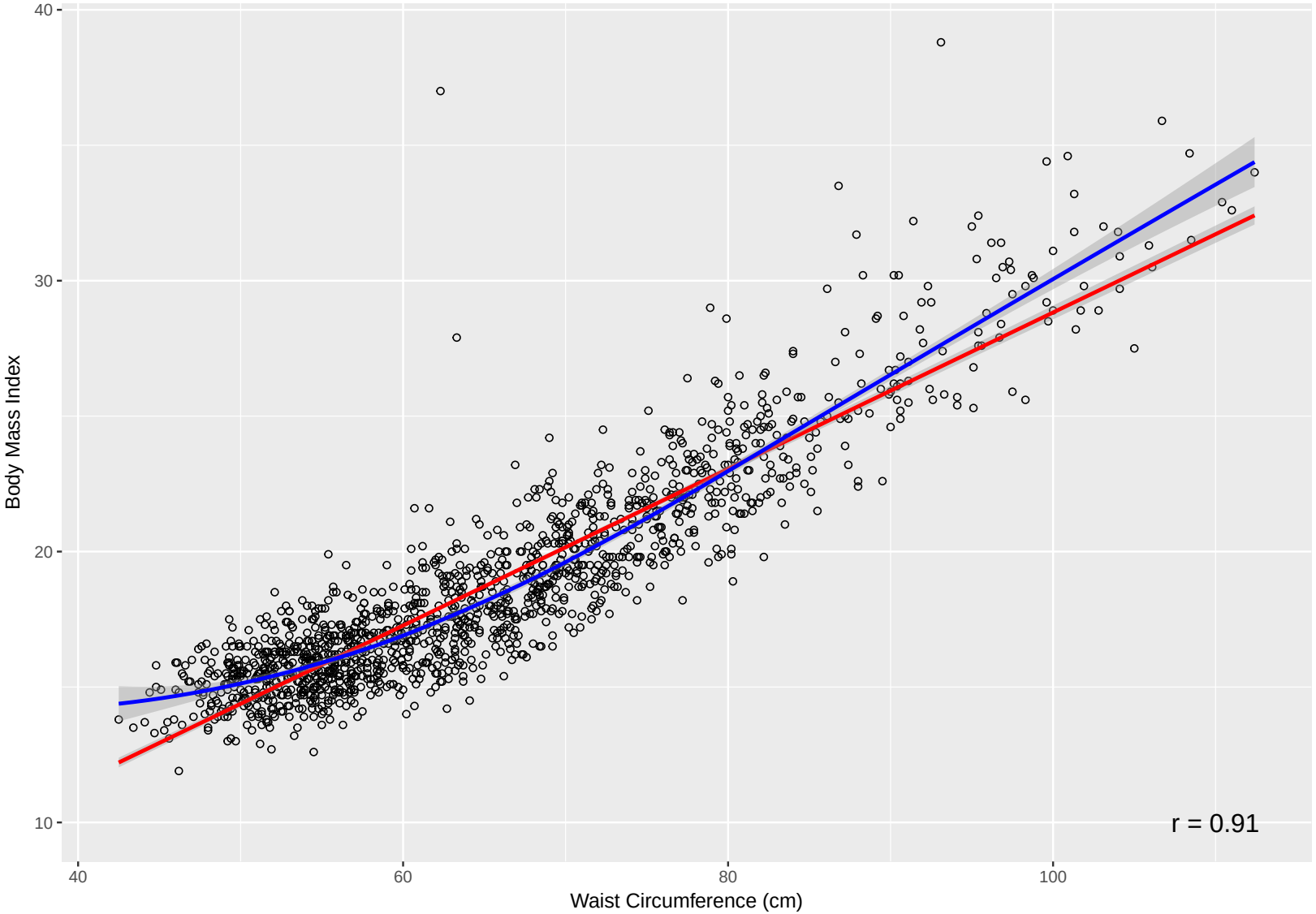
Summarizing Associations Graphically (Code)

```
cor_lab <- nyfs1 %$%
  cor(waist.circ, bmi) %>%
  round(.,2)

ggplot(nyfs1, aes(x = waist.circ, y = bmi)) +
  geom_point(pch = 1) +
  geom_smooth(method = "lm", col = "red") +
  geom_smooth(method = "loess", col = "blue") +
  labs(x = "Waist Circumference (cm)",
       y = "Body Mass Index") +
  annotate("text", x = 110, y = 10, size = 5,
           label = paste("r =",
                         cor_lab))
```

Plot on next slide. . .

Summarizing Associations Graphically



Summarizing Associations Numerically

```
nyfs1 %$% cor(waist.circ, bmi)
```

```
[1] 0.9100287
```

The Pearson Correlation, *r*

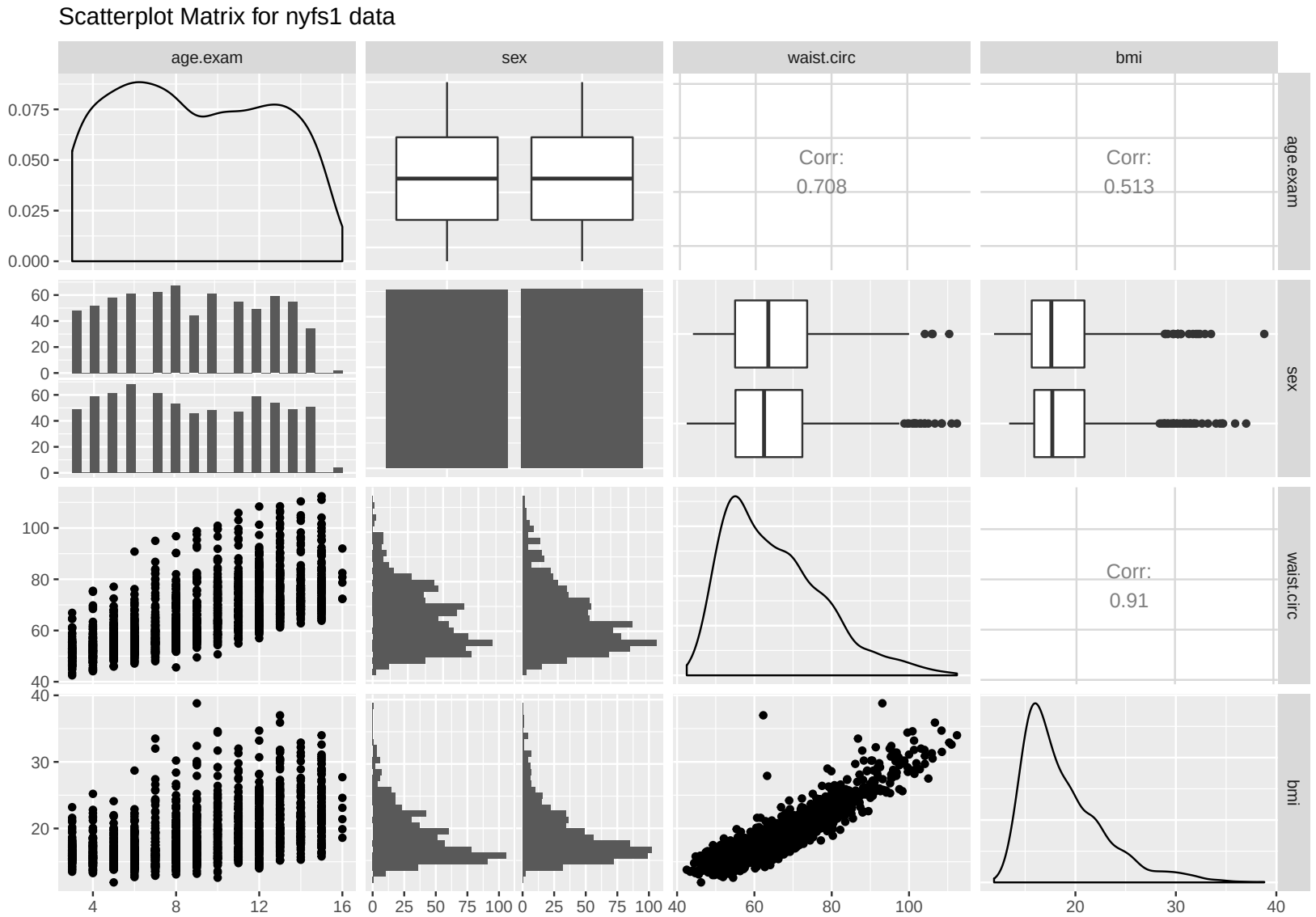
- Unitless (scale-free) measure of bivariate linear association
- r_{XY} = (slope of Y~X regression line) times SD_X / SD_Y
- $-1 \leq r \leq +1$
  - -1 indicates straight line relationship with negative slope
  - +1 indicates straight line relationship with positive slope
  - 0 indicates no linear association
- r^2 = key regression summary - % of variation in Y explained by X

Scatterplot Matrix for some NYFS1 variables (Code)

```
GGally::ggpairs(select(nyfs1,
                       age.exam, triceps.skinfold,
                       sex, waist.circ, bmi),
        title = "Scatterplot Matrix for nyfs1 data")
```

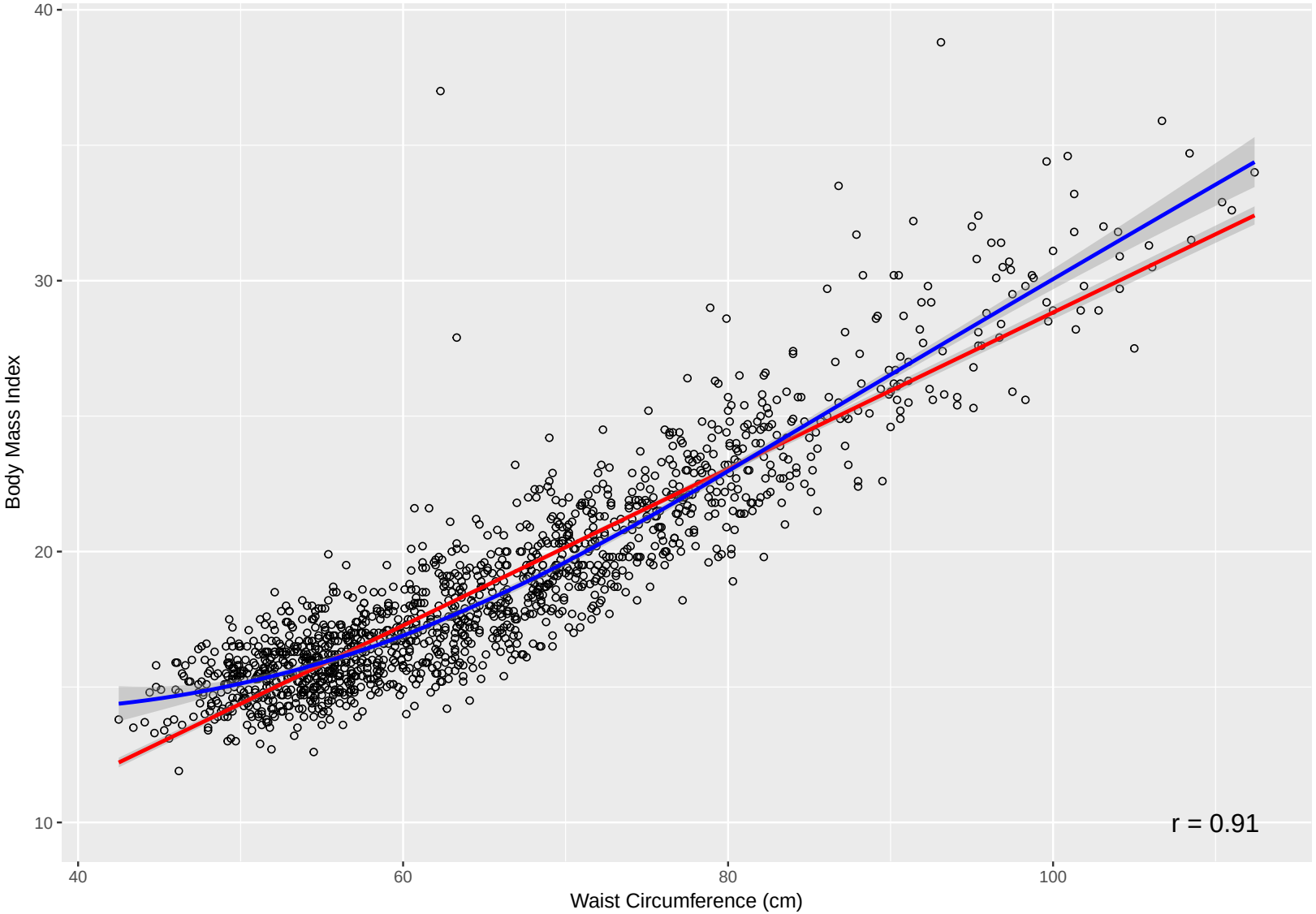
Note that I usually run this with `message = FALSE` in the chunk label. Otherwise, you get some irritating messages in your output.

Scatterplot Matrix for some NYFS1 variables



Model 1: Predicting `bmi` using `waist.circ` alone

Summarizing the `bmi-waist.circ` association



Linear Model 1: bmi ~ f(waist.circ)

```
m1 <- nyfs1 %$% lm(bmi ~ waist.circ)

m1
```

```
Call:
lm(formula = bmi ~ waist.circ)

Coefficients:
(Intercept)   waist.circ
   -0.06646      0.28893
```

```
summary(m1) ## see next slide
```

Summary for Linear Model (rearranged, a bit)

```
Call: lm(formula = bmi ~ waist.circ)

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.06646    0.23292  -0.285    0.775
waist.circ     0.28893    0.00350  82.548   <2e-16 ***
---
Sig. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Multiple R-squared:  0.8282
F-statistic:  6814 on 1 and 1414 DF,  p-value: < 2.2e-16

Residuals:     Min      1Q  Median      3Q     Max
           -4.2343 -1.0941 -0.0744  0.9254 19.0664
Residual standard error: 1.692 on 1414 degrees of freedom
Adjusted R-squared:  0.828
```

95% Confidence Intervals for `m1` Coefficients

```
m1
```

```

Call:
lm(formula = bmi ~ waist.circ)

Coefficients:
(Intercept)   waist.circ
   -0.06646      0.28893
```

```
confint(m1, level = 0.95)
```

```
                 2.5 %    97.5 %
(Intercept) -0.5233582 0.3904439
waist.circ   0.2820601 0.2957920
```

Tidying the Coefficients (with `broom::tidy`)

Places the coefficient summary into a tibble.

```
tidy(m1) ## from broom package
```

```
         term    estimate   std.error statistic
1 (Intercept) -0.06645719 0.232917523 -0.285325
2  waist.circ  0.28892604 0.003500087 82.548253
   p.value
1 0.775437
2 0.000000
```

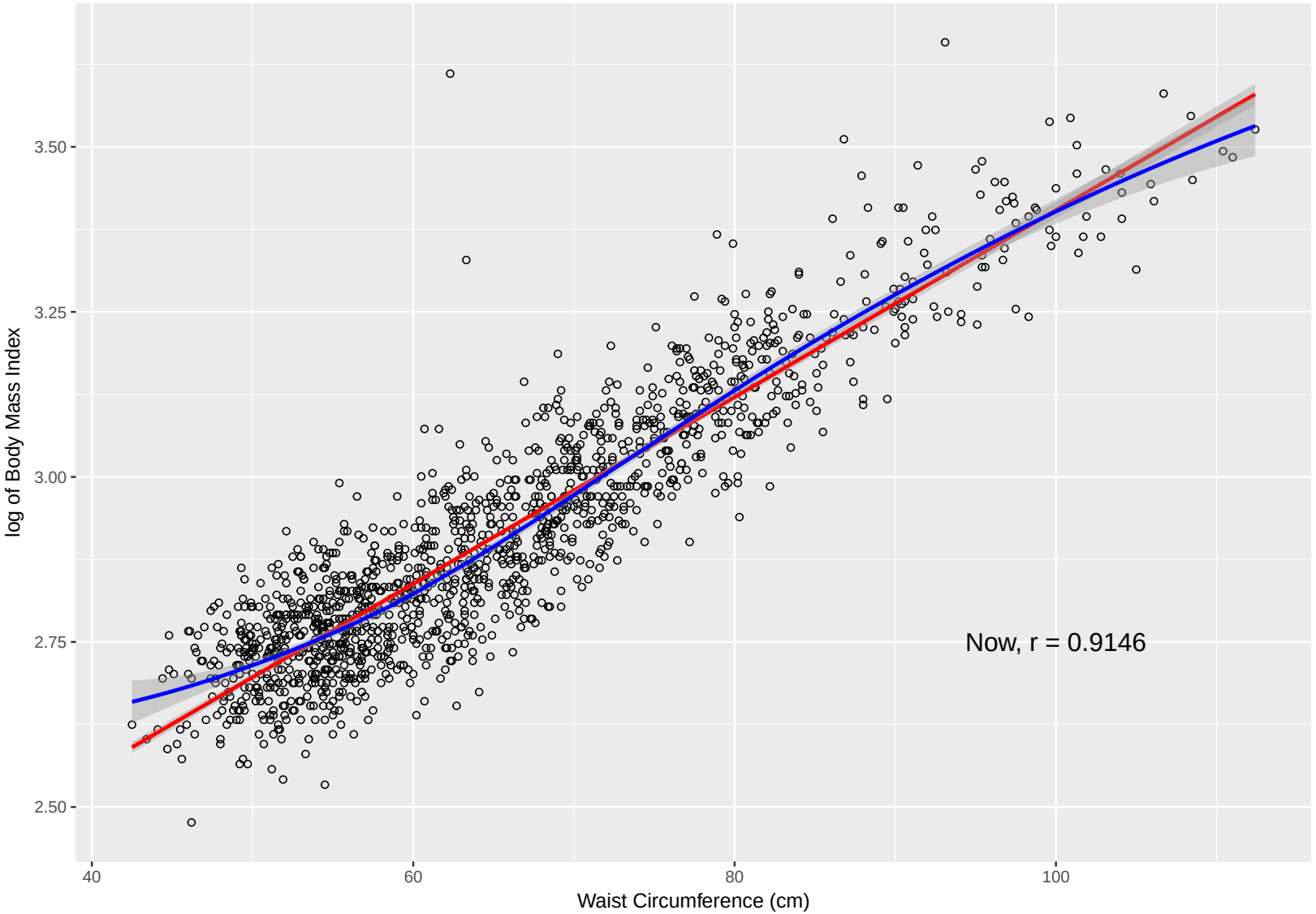
Model Summaries, at a glance (with broom::glance)

Places detailed model summaries into a one-row tibble.

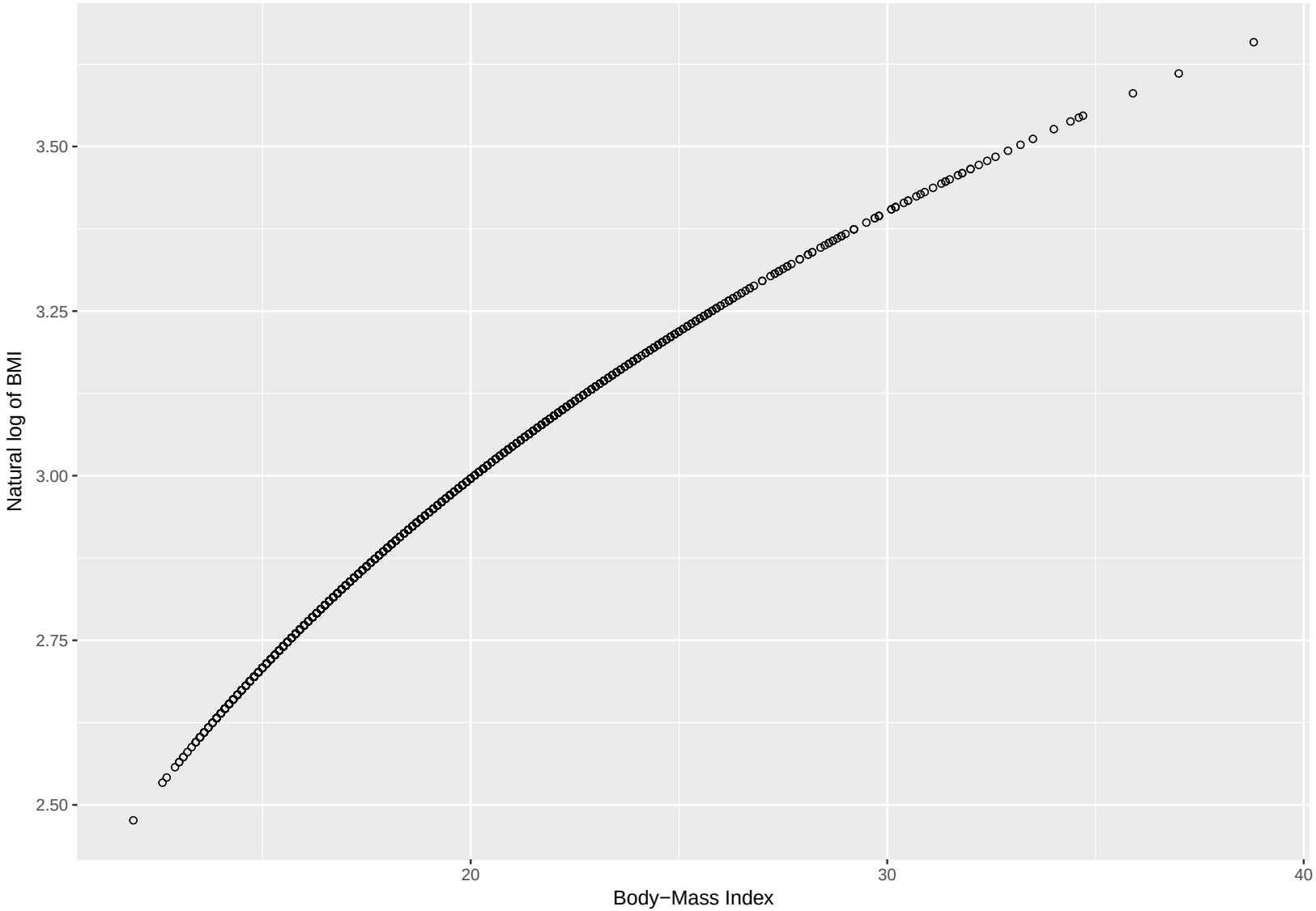
```
glance(m1) ## also from broom
```

```
  r.squared adj.r.squared    sigma statistic p.value
1 0.8281523     0.8280307 1.692336  6814.214       0
  df    logLik      AIC      BIC deviance df.residual
1  2 -2753.188 5512.376 5528.143   4049.7        1414
```

What if we tried log(BMI)?



log(BMI) vs. BMI: The effect of the log



But what if we tried modeling the log of BMI?

```
m1.log <- nyfs1 %$% lm(log(bmi) ~ waist.circ)

m1.log
```

```
Call:
lm(formula = log(bmi) ~ waist.circ)

Coefficients:
(Intercept)   waist.circ
    1.98892      0.01415
```

m1.log vs. m1 (Coefficients)

```
tidy(m1)
```

```
         term    estimate   std.error statistic
1 (Intercept) -0.06645719 0.232917523 -0.285325
2  waist.circ  0.28892604 0.003500087 82.548253
   p.value
1 0.775437
2 0.000000
```

```
tidy(m1.log)
```

```
         term   estimate    std.error statistic
1 (Intercept) 1.98891965 0.0110708236  179.6542
2  waist.circ 0.01415173 0.0001663629   85.0654
  p.value
1       0
2       0
```

m1.log vs. m1 (Summaries)

```
glance(m1)
```

```
  r.squared adj.r.squared    sigma statistic p.value
1 0.8281523     0.8280307 1.692336  6814.214       0
  df    logLik       AIC       BIC deviance df.residual
1  2 -2753.188 5512.376 5528.143   4049.7         1414
```

```
glance(m1.log)
```

```
  r.squared adj.r.squared     sigma statistic p.value
1 0.8365341     0.8364185 0.0804386  7236.122       0
  df   logLik       AIC      BIC deviance df.residual
1  2 1560.474 -3114.947 -3099.18   9.1491         1414
```

Augmenting the Data with Model m1 Results (broom::augment) run with warning = FALSE

```
newdata <- augment(m1) ## yet again from broom

head(newdata,3)
```

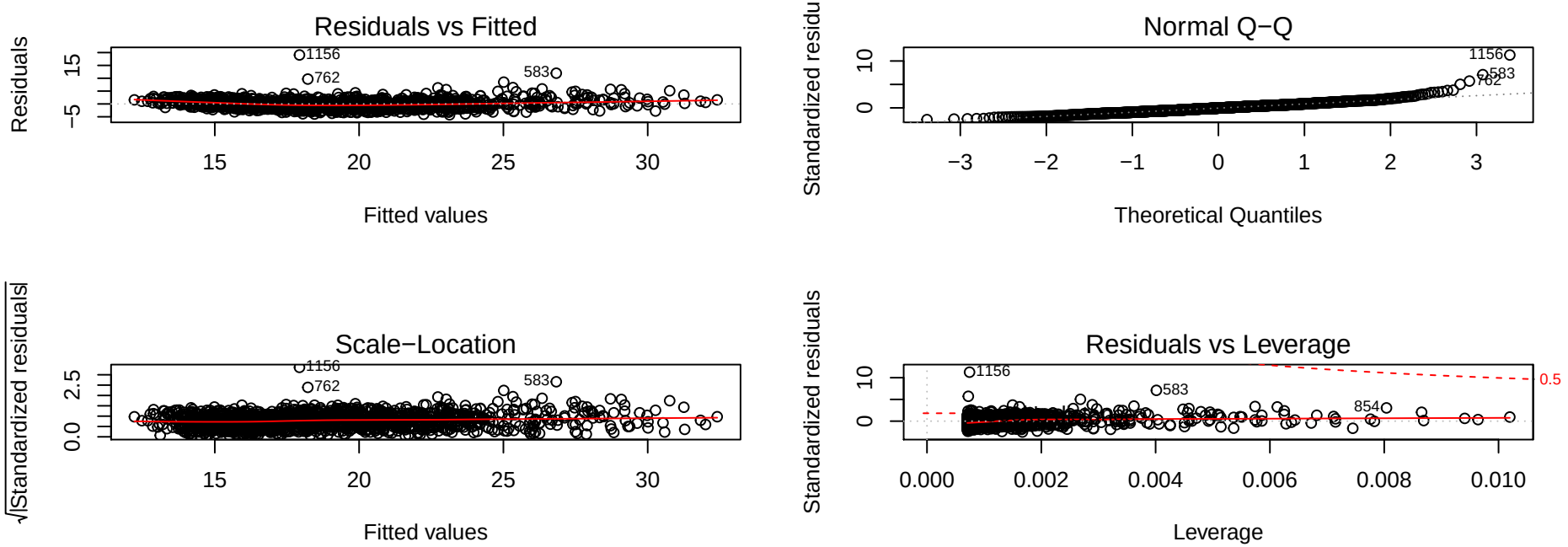
```
   bmi waist.circ  .fitted    .se.fit     .resid
1 22.3        71.9 20.70732 0.05056896  1.592675
2 19.8        79.4 22.87427 0.06678505 -3.074270
3 15.2        46.8 13.45528 0.07882016  1.744719
           .hat   .sigma      .cooksd .std.resid
1 0.0008928832 1.692404 0.0003961152  0.9415307
2 0.0015573458 1.690955 0.0025776230 -1.8179994
3 0.0021692083 1.692297 0.0011578035  1.0320726
```

Assumption Checking: The Role of Residuals

1. ("Linearity") The residuals from the linear model should show no particular curved relationship when plotted against the fitted values, or when plotted against individual predictors.
2. ("Independence") Particularly if the data describe a series in time or space, we want to see no clear cycles in the residuals when plotted against time/space.
3. ("Homoscedasticity" / "Constant Variance") The residuals from the linear model should display a similar amount of spread across levels of the fitted values. Sometimes we plot the square root of the residuals against the fitted values to assess this.
4. ("Approximately Normal") The residuals from the linear model should follow an approximately Normal distribution. Often, we'll plot the residuals in a normal Q-Q plot to assess this. We'd like to avoid [a] fitting points poorly (exceptionally large positive or negative residuals), [b] points that show enormous influence over the model (removing them causes the coefficients to change substantially)

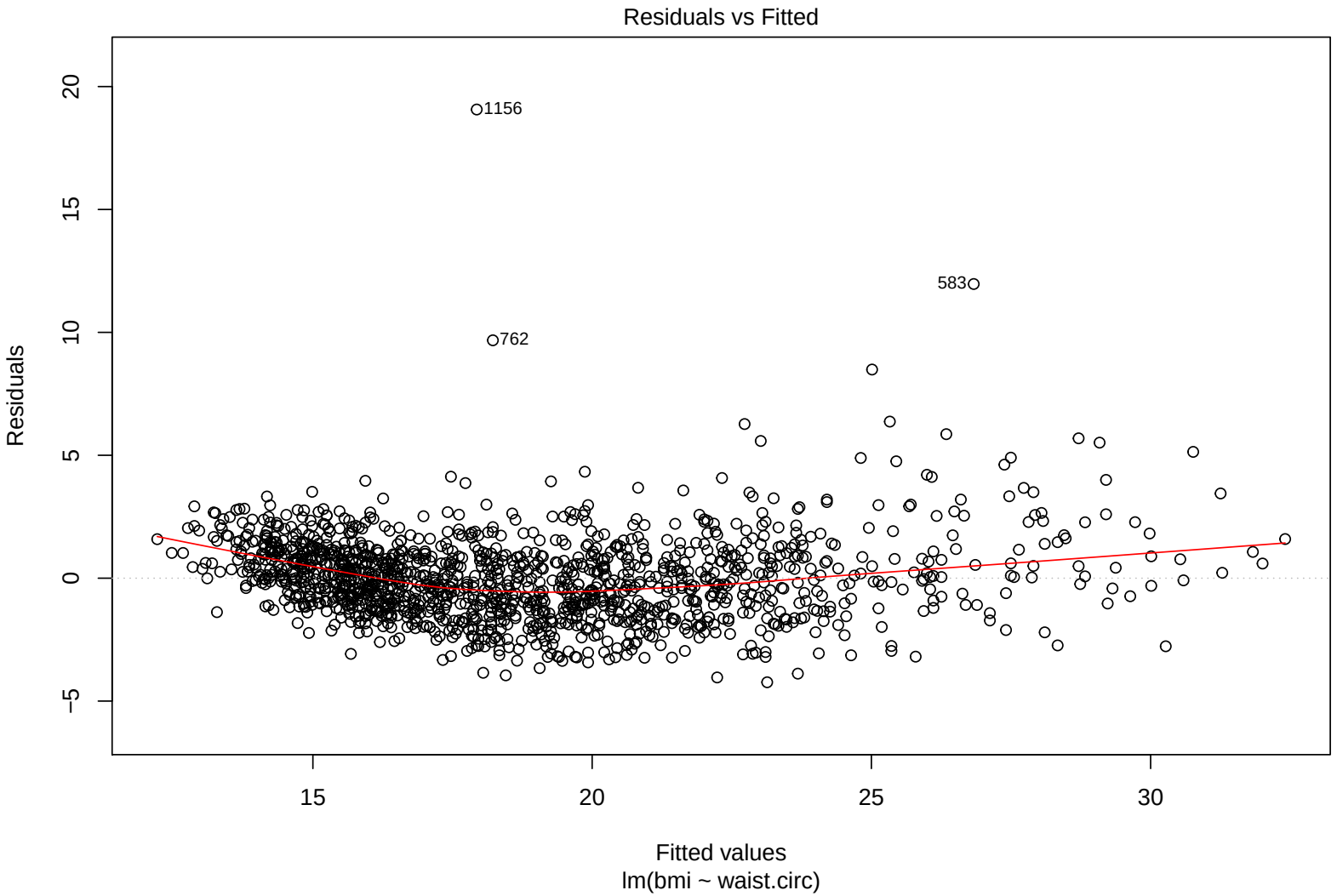
Plotting the Model Residuals (squeezed too much)

```
par(mfrow = c(2,2))
plot(m1)
```

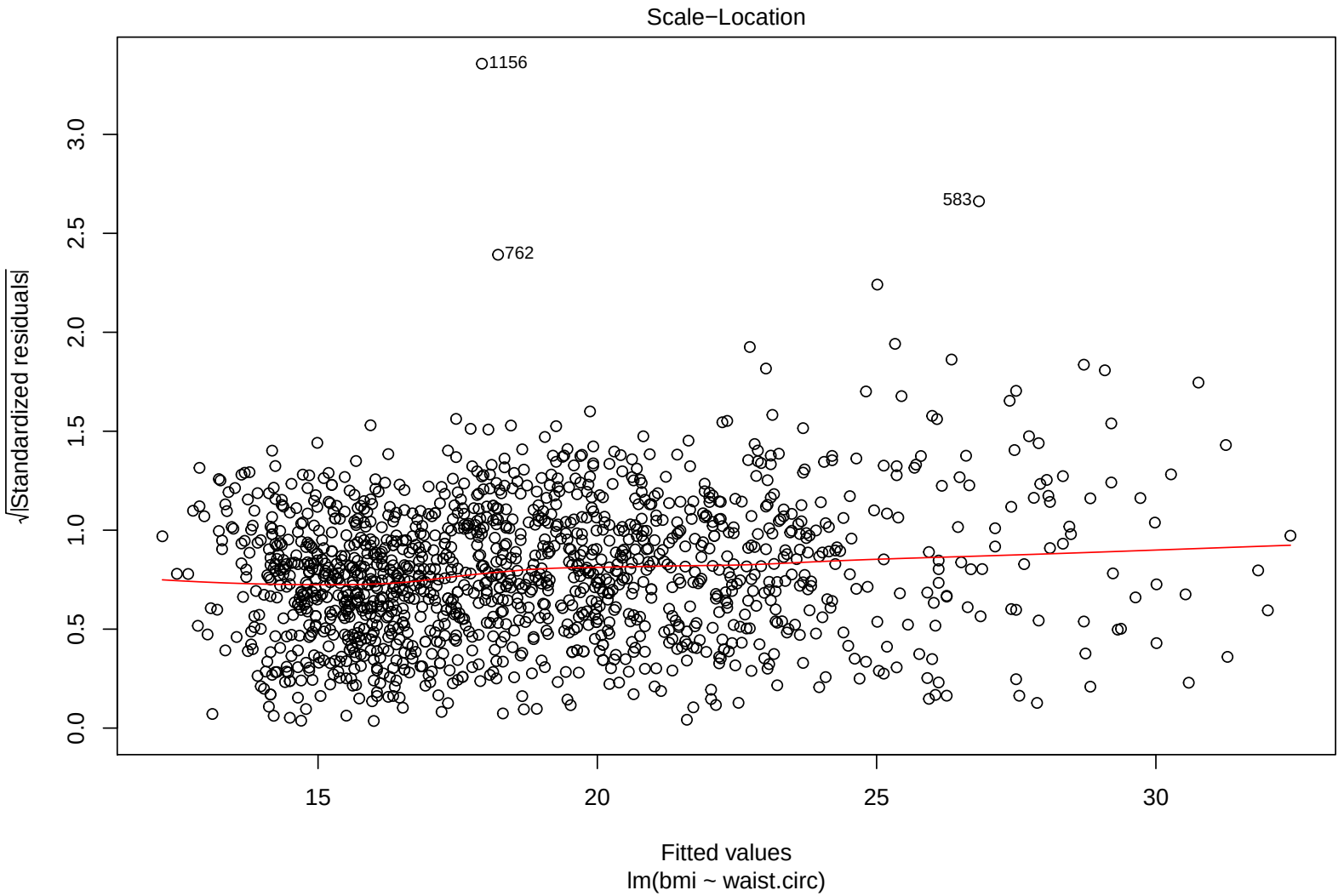



```
par(mfrow = c(1,1))
```

`plot(m1, which = 1)` Residuals vs Fitted Values



plot(m1, which = 3) Scale-Location Plot



Subjects labeled 583, 762 and 1156

```
nyfs1 %>% slice(c(583, 762, 1156)) %>%
  select(subject.id, bmi, waist.circ)
```

```
# A tibble: 3 x 3
  subject.id   bmi waist.circ
       <int> <dbl>      <dbl>
1      72560  38.8       93.1
2      72758  27.9       63.3
3      73202  37.0       62.3
```

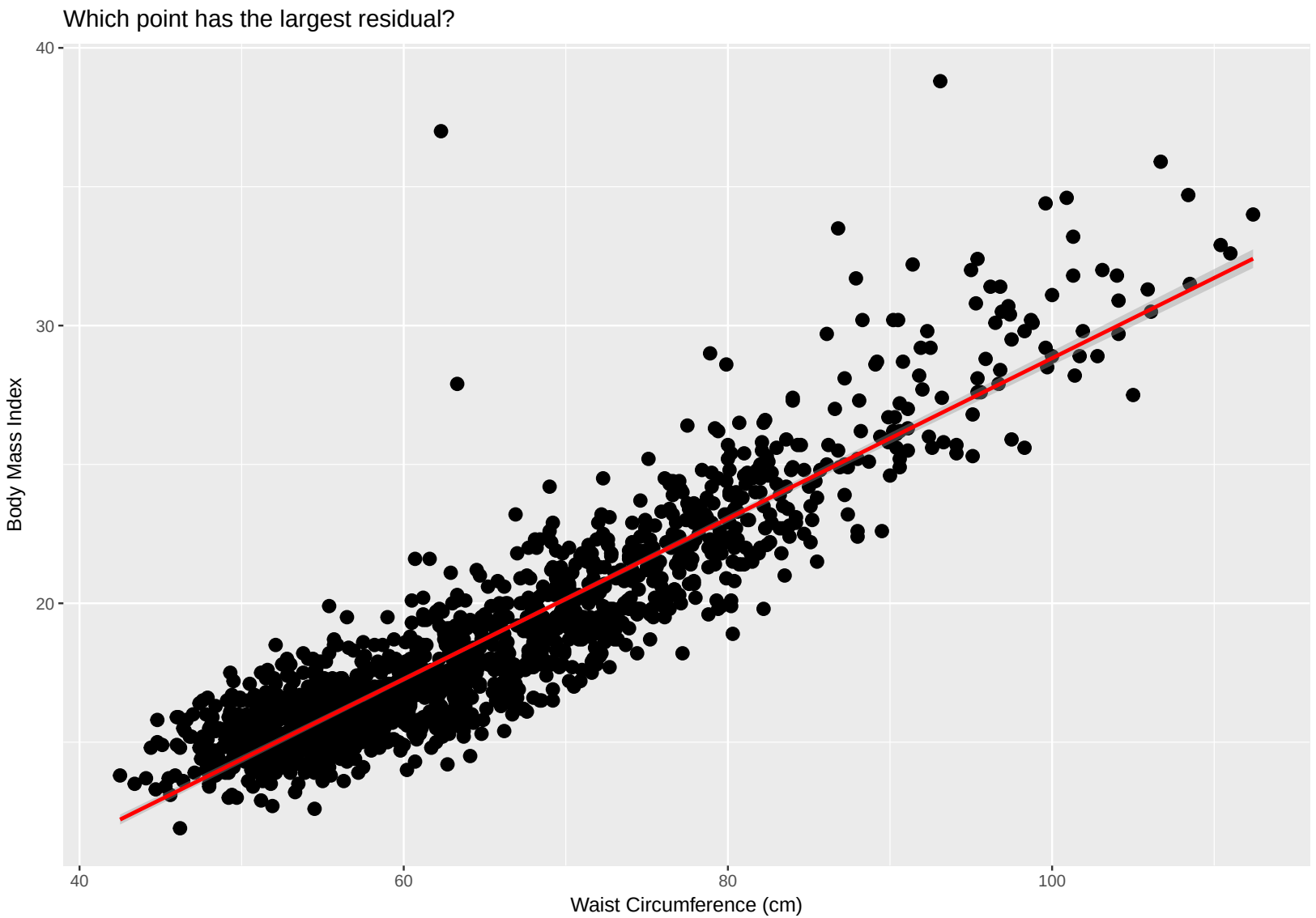
What about our augmented data?

Subjects labeled 583, 762 and 1156

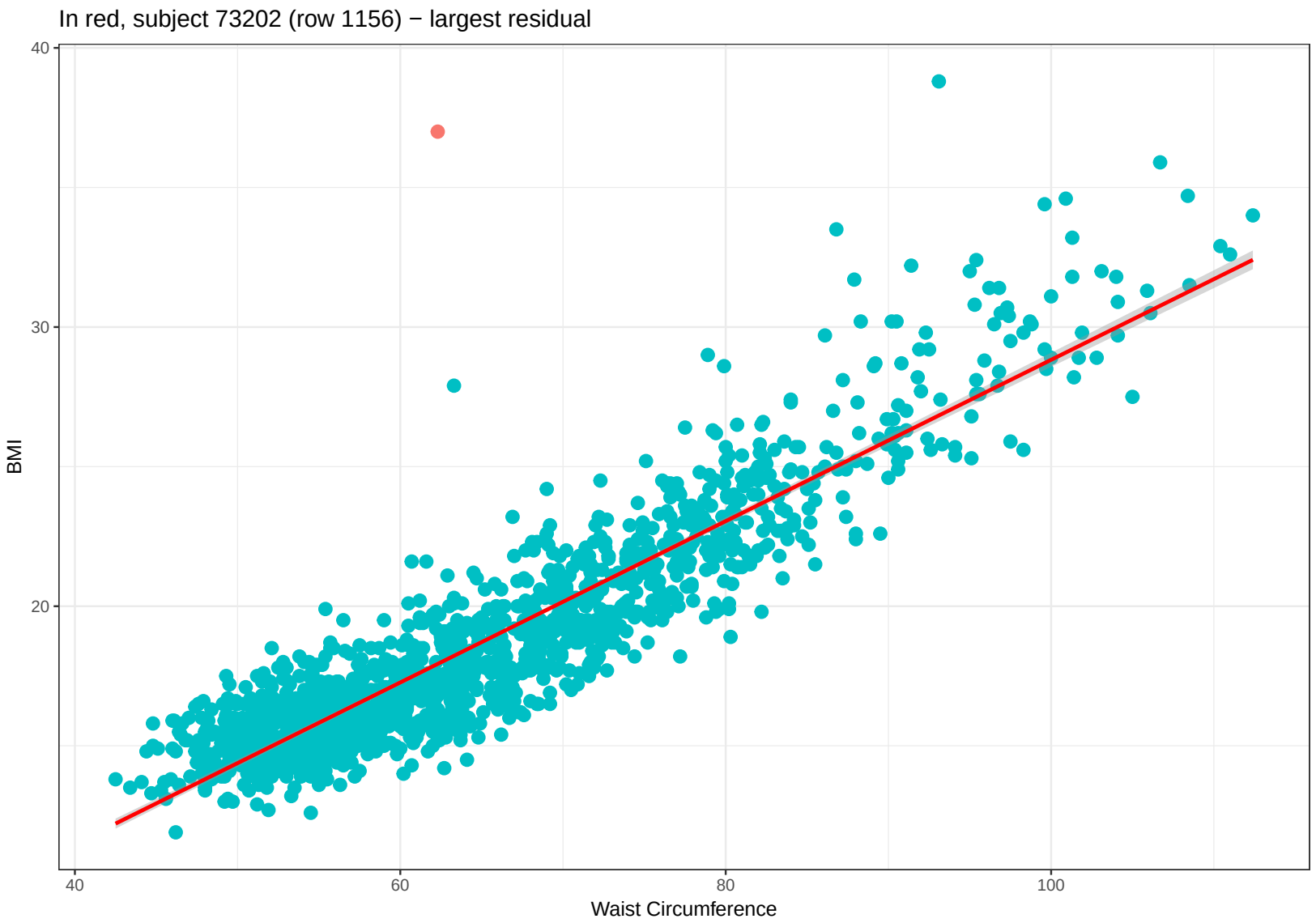
```
augment(m1) %>%
  slice(c(583, 762, 1156)) %>%
  select(bmi, waist.circ, .fitted, .resid,
         .std.resid, .cooksd) %>%
  round(., 2)
```

```
# A tibble: 3 x 6
    bmi waist.circ .fitted .resid .std.resid .cooksd
  <dbl>      <dbl>   <dbl>  <dbl>      <dbl>   <dbl>
1  38.8       93.1   26.83  11.97       7.09    0.10
2  27.9       63.3   18.22   9.68       5.72    0.01
3  37.0       62.3   17.93  19.07      11.27    0.05
```

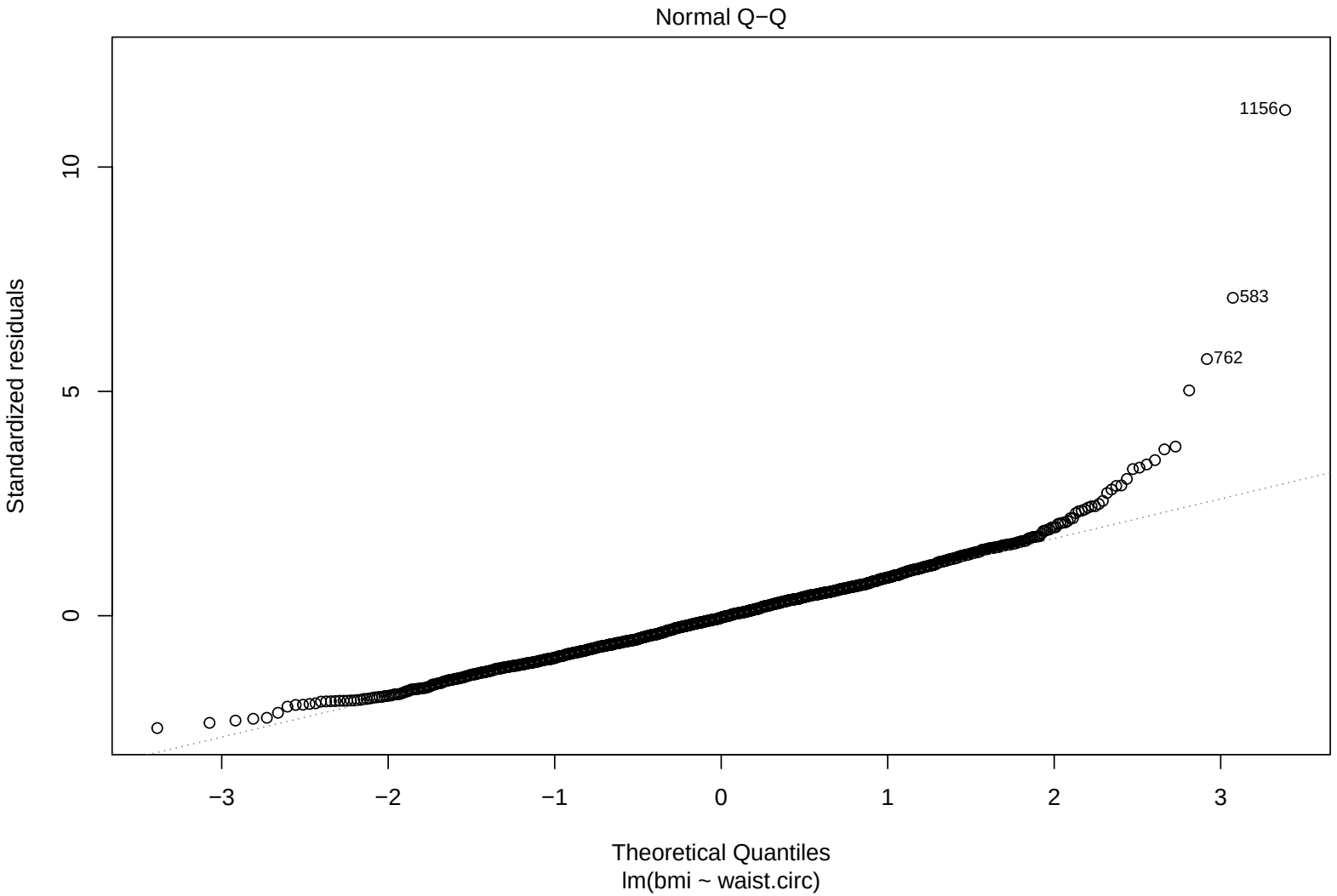
Which point has the largest residual in the data?



Row 1156 (ID 73202) has the largest residual

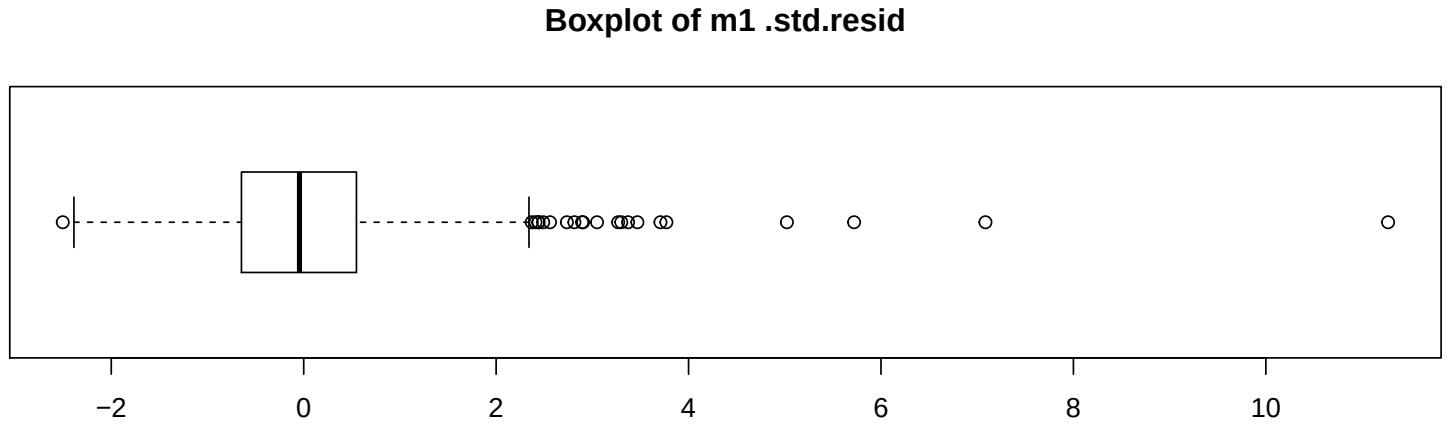


plot(m1, which = 2)



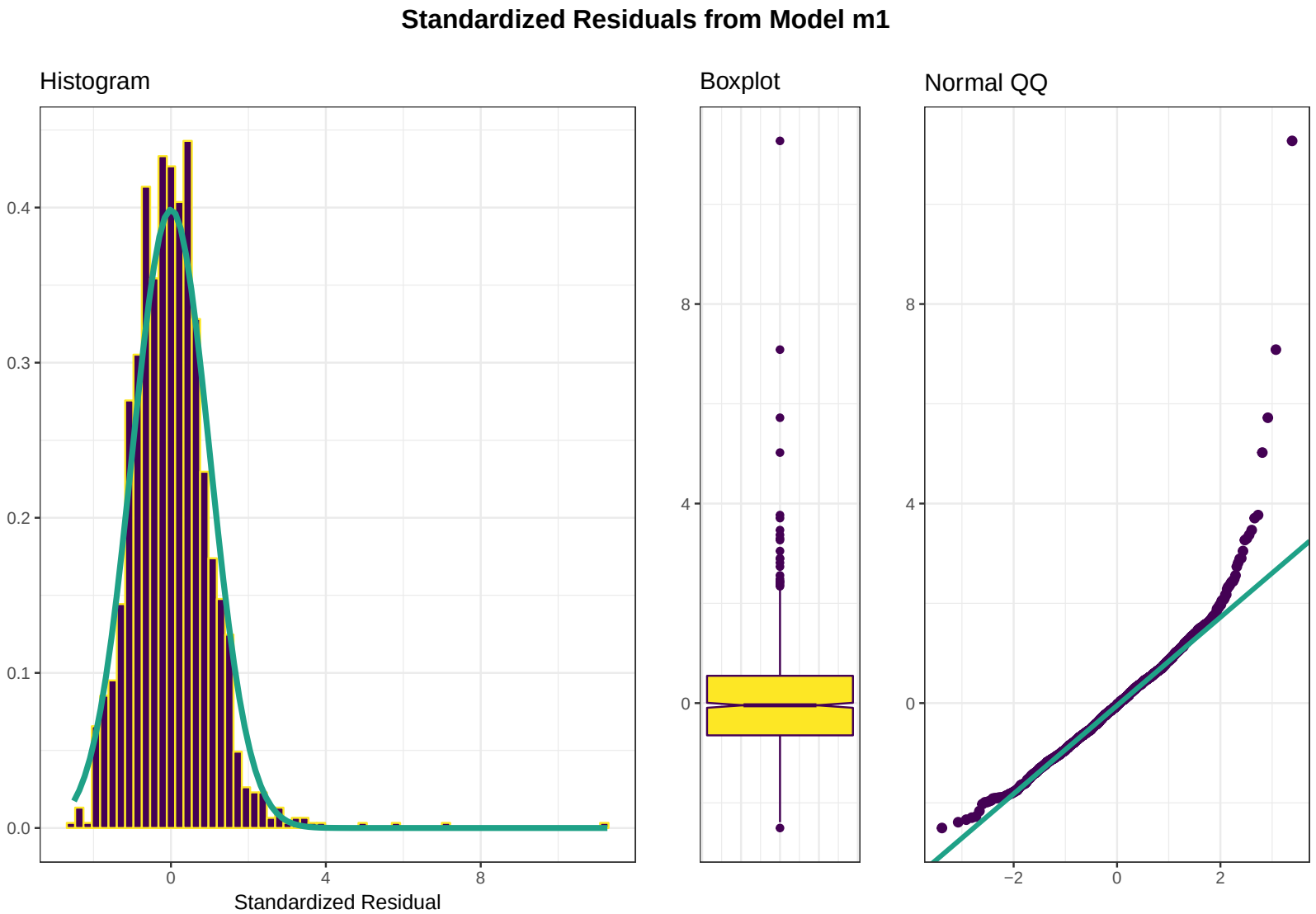
Standardized Residuals just rescale Raw Residuals

.resid = Model m1 Raw Residuals



.std.resid = Model m1 Standardized Residuals

.std.resid from Model m1 (Notes, Section 39.2)



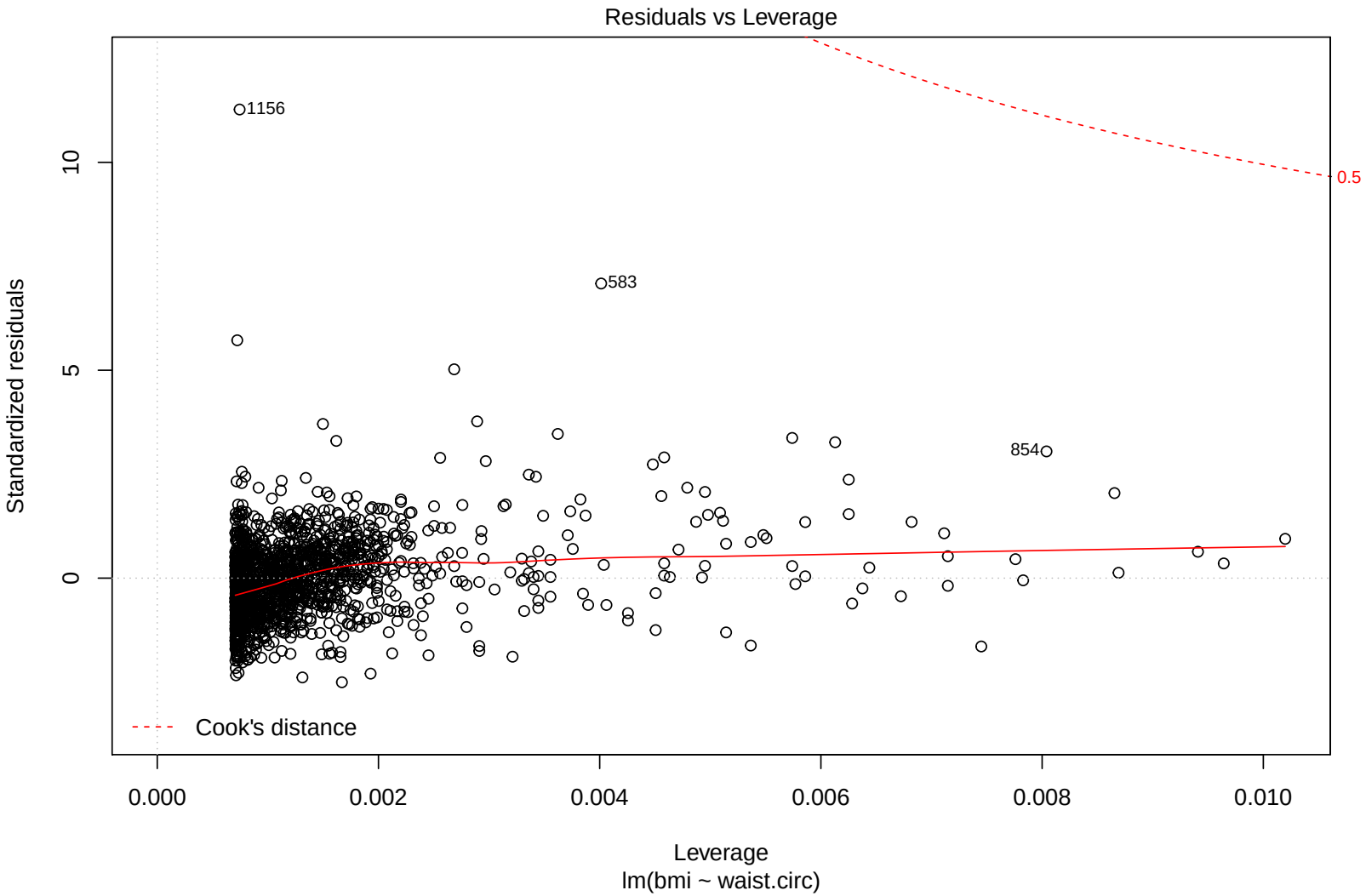
Outliers, Leverage, Influence

We will also examine the **leverage** of individual points (`.hat` in the augment output) where large values indicate unusual combinations of predictors, especially when we have more than one predictor.

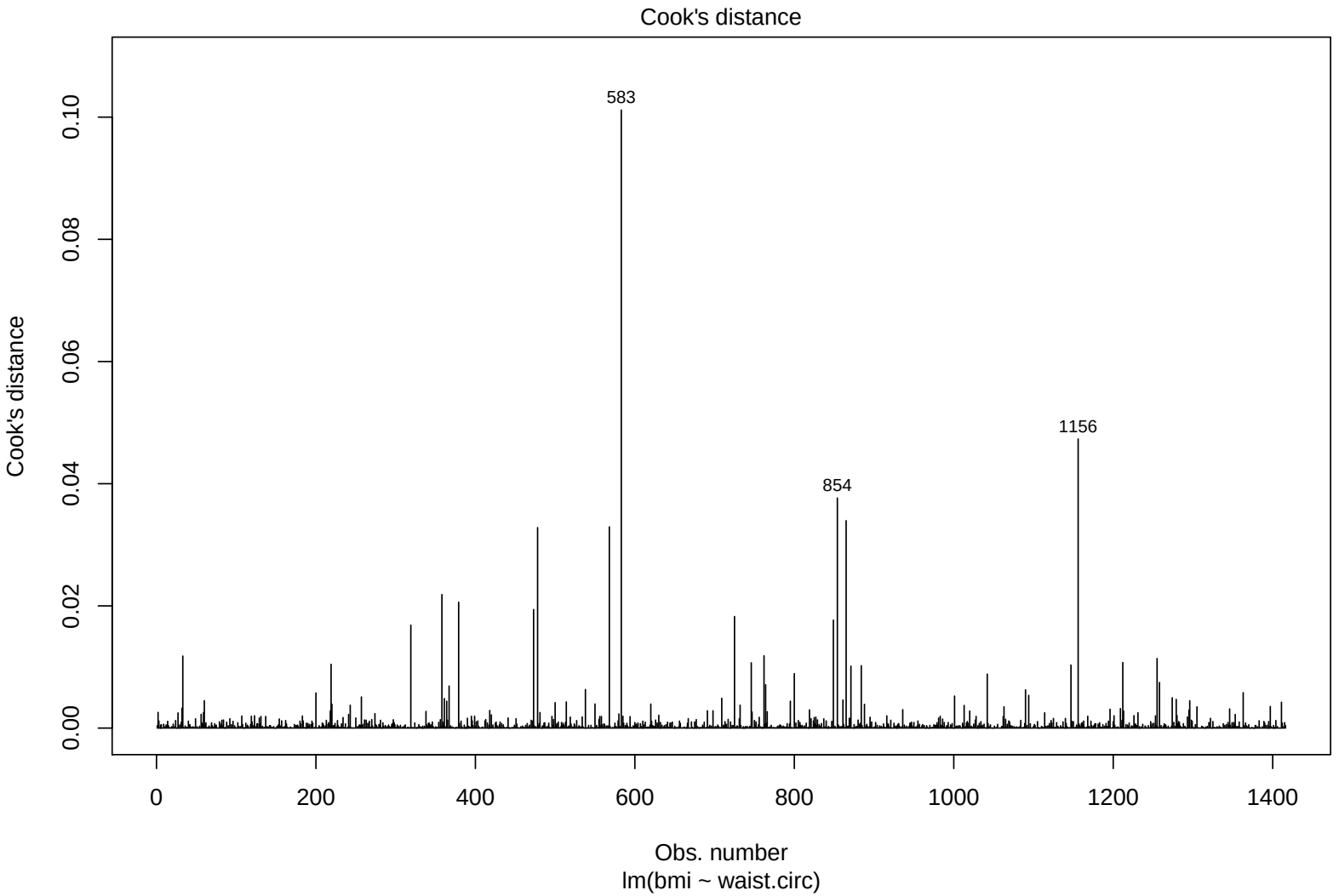
```
augment(m1) %>%
  slice(c(583, 762, 1156)) %>%
  select(bmi, waist.circ, .hat,
         .std.resid, .cooksd)
```

```
# A tibble: 3 x 5
    bmi waist.circ          .hat .std.resid    .cooksd
  <dbl>      <dbl>         <dbl>      <dbl>      <dbl>
1  38.8       93.1 0.0040134378   7.085784 0.10116002
2  27.9       63.3 0.0007232207   5.720458 0.01184177
3  37.0       62.3 0.0007445560  11.270493 0.04732348
```

`plot(m1, which = 5)` Residuals vs. Leverage Plot



`plot(m1, which = 4)` Cook's Distance Plot



Why does 583 have the largest Cook's distance?

```
summary(select(nyfs1, waist.circ, bmi))
```

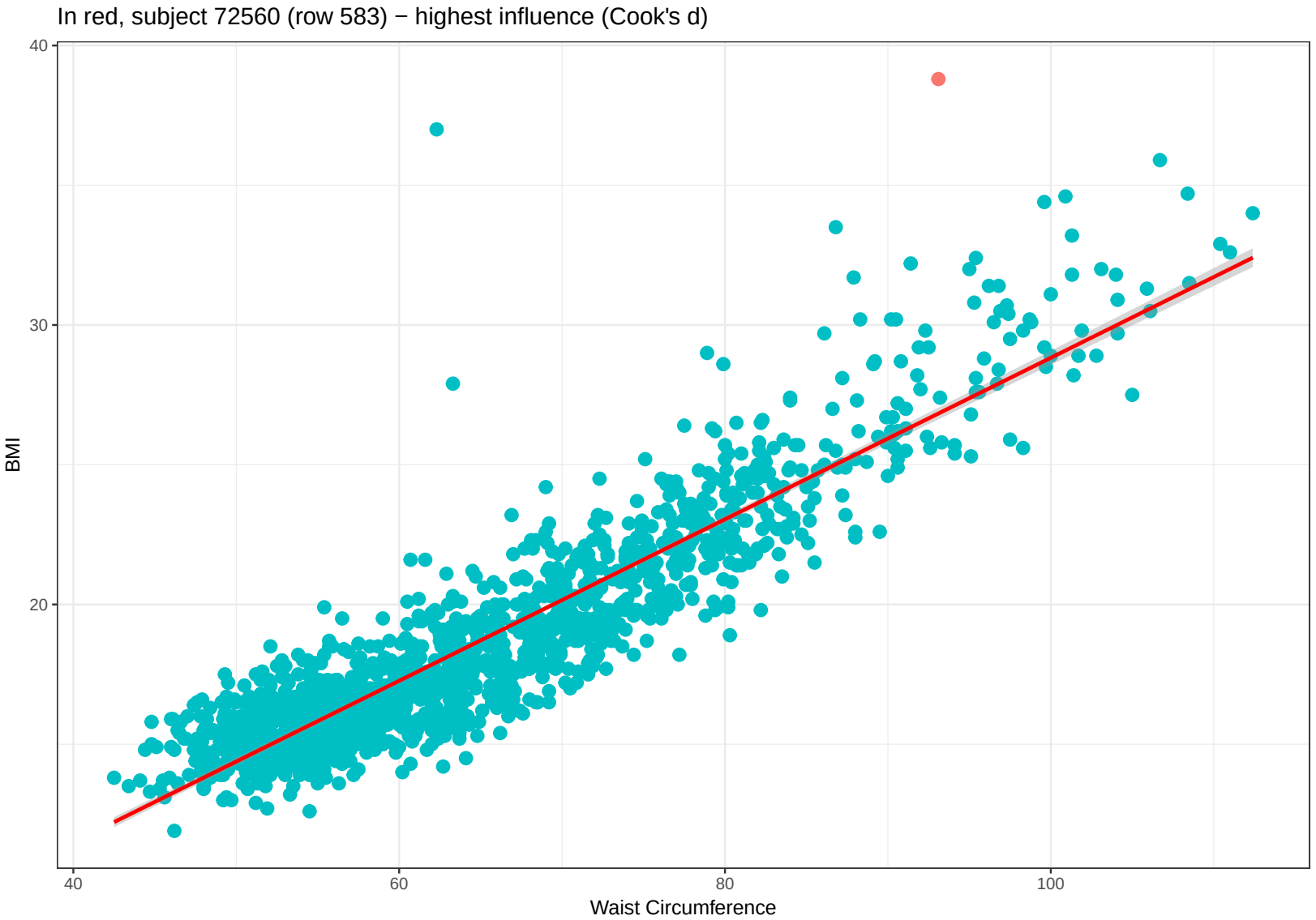
```
   waist.circ          bmi       
 Min.   : 42.50   Min.   :11.9  
 1st Qu.: 55.00   1st Qu.:15.8  
 Median : 63.00   Median :17.7  
 Mean   : 65.29   Mean   :18.8  
 3rd Qu.: 72.92   3rd Qu.:20.9  
 Max.   :112.40   Max.   :38.8  
```

Why does 583 have the largest Cook's distance?

```
nyfs1 %>% slice(583) %>%
  select(subject.id, waist.circ, bmi)
```

```
# A tibble: 1 x 3
  subject.id waist.circ   bmi
       <int>      <dbl> <dbl>
1      72560       93.1  38.8
```

Why does 583 have the largest Cook's distance?



Model m1, without point 583

```
m1.no583 <- lm(bmi ~ waist.circ, data = slice(nyfs1, -583))

m1.no583
```

```
Call:
lm(formula = bmi ~ waist.circ, data = slice(nyfs1, -583))

Coefficients:
(Intercept)   waist.circ
    0.01837      0.28750
```

m1 with and without 583, coefficients

```
tidy(m1)
```

```
         term    estimate   std.error statistic
1 (Intercept) -0.06645719 0.232917523 -0.285325
2  waist.circ  0.28892604 0.003500087 82.548253
   p.value
1 0.775437
2 0.000000
```

```
tidy(m1.no583)
```

```
         term   estimate   std.error   statistic
1 (Intercept) 0.01837068 0.229127911  0.08017652
2  waist.circ 0.28749691 0.003444305 83.47022689
    p.value
1 0.9361082
2 0.0000000
```

m1 with and without 583, summaries

```
glance(m1)
```

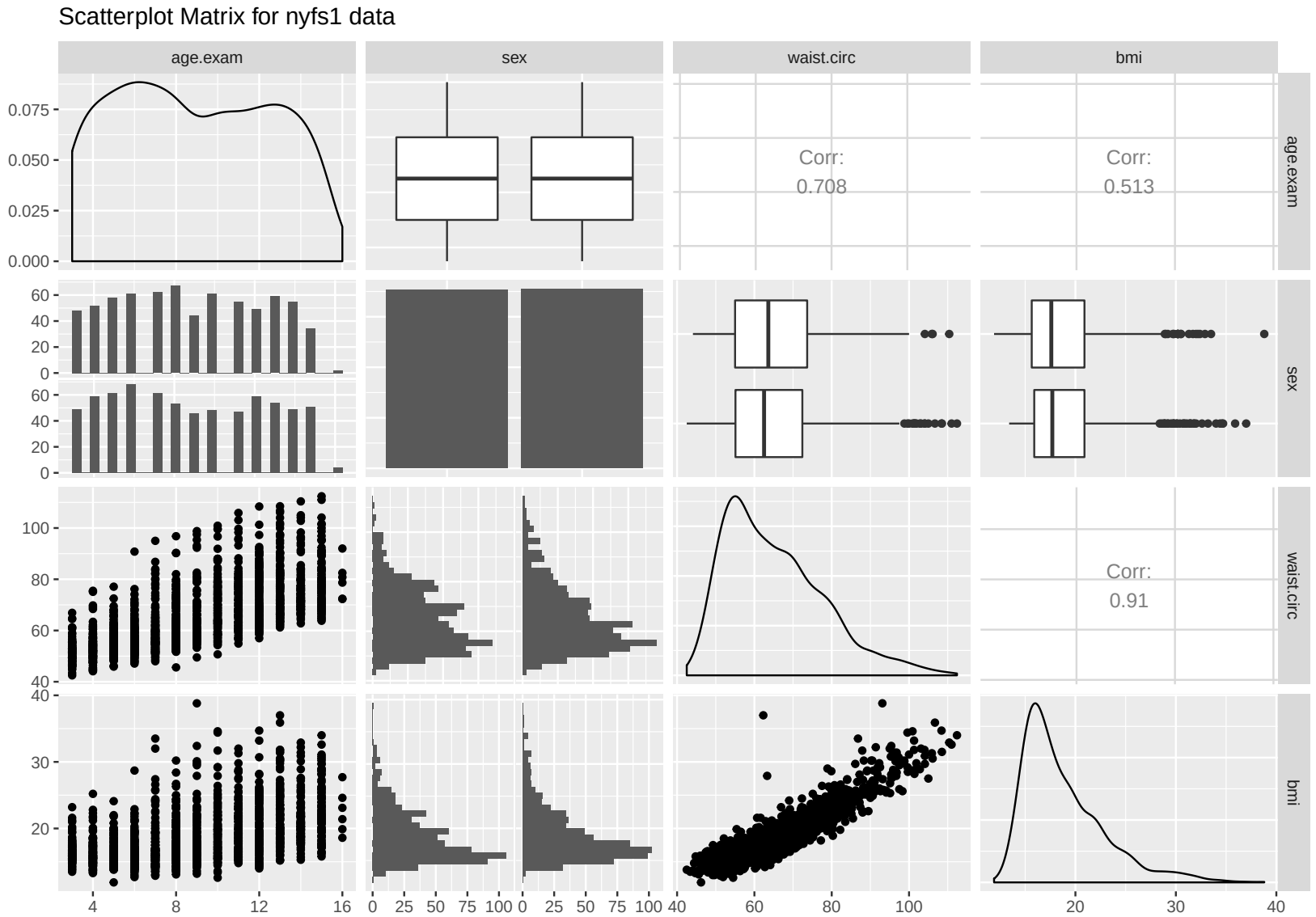
```
  r.squared adj.r.squared    sigma statistic p.value
1 0.8281523     0.8280307 1.692336  6814.214       0
  df    logLik      AIC      BIC deviance df.residual
1  2 -2753.188 5512.376 5528.143   4049.7        1414
```

```
glance(m1.no583)
```

```
  r.squared adj.r.squared    sigma statistic p.value
1 0.8313899     0.8312705 1.662607  6967.279       0
  df    logLik     AIC      BIC deviance df.residual
1  2 -2726.165 5458.33 5474.094 3905.903        1413
```

Model 2: Predicting `bmi` using `waist.circ` as well as `age.exam` and `sex`

Scatterplot Matrix for some NYFS1 variables



Building Model 2

```
m2 <- nyfs1 %$%
  lm(bmi ~ waist.circ + age.exam + sex)

m2
```

```
Call:
lm(formula = bmi ~ waist.circ + age.exam + sex)

Coefficients:
(Intercept)   waist.circ     age.exam      sexMale
    -1.4452       0.3484      -0.2932       0.1881
```

Summary of Model m2 (rearranged a little)

```
Call: lm(formula = bmi ~ waist.circ + age.exam + sex)

Multiple R-squared:  0.8635,    Adjusted R-squared:  0.8633
F-statistic:  2979 on 3 and 1412 DF,  p-value: < 2.2e-16

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.445179   0.222018  -6.509 1.05e-10 ***
waist.circ    0.348370   0.004421  78.801  < 2e-16 ***
age.exam     -0.293247   0.015440 -18.992  < 2e-16 ***
sexMale       0.188094   0.080208   2.345   0.0192 *
---
Sig. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residuals:     Min      1Q  Median      3Q     Max
           -4.0232 -0.8879 -0.0802  0.7836 20.3659
Residual standard error: 1.509 on 1412 degrees of freedom
```

Summary of m1 (for reference, rearranged)

```
Call: lm(formula = bmi ~ waist.circ)

Multiple R-squared:  0.8282,    Adjusted R-squared:  0.828
F-statistic:  6814 on 1 and 1414 DF,  p-value: < 2.2e-16

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.06646    0.23292  -0.285    0.775
waist.circ     0.28893    0.00350  82.548   <2e-16 ***
---
Sig. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residuals:     Min      1Q  Median      3Q     Max
           -4.2343 -1.0941 -0.0744  0.9254 19.0664
Residual standard error: 1.692 on 1414 degrees of freedom
```

95% Confidence Intervals for m2 Coefficients

```
confint(m2, level = 0.95)
```

```
                   2.5 %     97.5 %
(Intercept) -1.88069873 -1.0096587
waist.circ   0.33969754  0.3570418
age.exam    -0.32353535 -0.2629586
sexMale      0.03075409  0.3454346
```

Tidying the Coefficients (with `broom::tidy`)

Places the coefficient summary into a tibble.

```
tidy(m2) ## from broom package
```

```
         term   estimate   std.error  statistic
1 (Intercept) -1.4451787 0.222017690  -6.509295
2  waist.circ  0.3483697 0.004420855  78.801422
3    age.exam -0.2932470 0.015440287 -18.992327
4     sexMale  0.1880943 0.080208291   2.345073
        p.value
1 1.046338e-10
2 0.000000e+00
3 8.257321e-72
4 1.916119e-02
```

Model Summaries, at a glance (with broom::glance)

```
glance(m2) ## also from broom
```

```
  r.squared adj.r.squared    sigma statistic p.value
1 0.8635437     0.8632538 1.509103  2978.545       0
  df   logLik      AIC      BIC deviance df.residual
1  4 -2589.92 5189.84 5216.118 3215.678        1412
```

```
glance(m1) ## for comparison
```

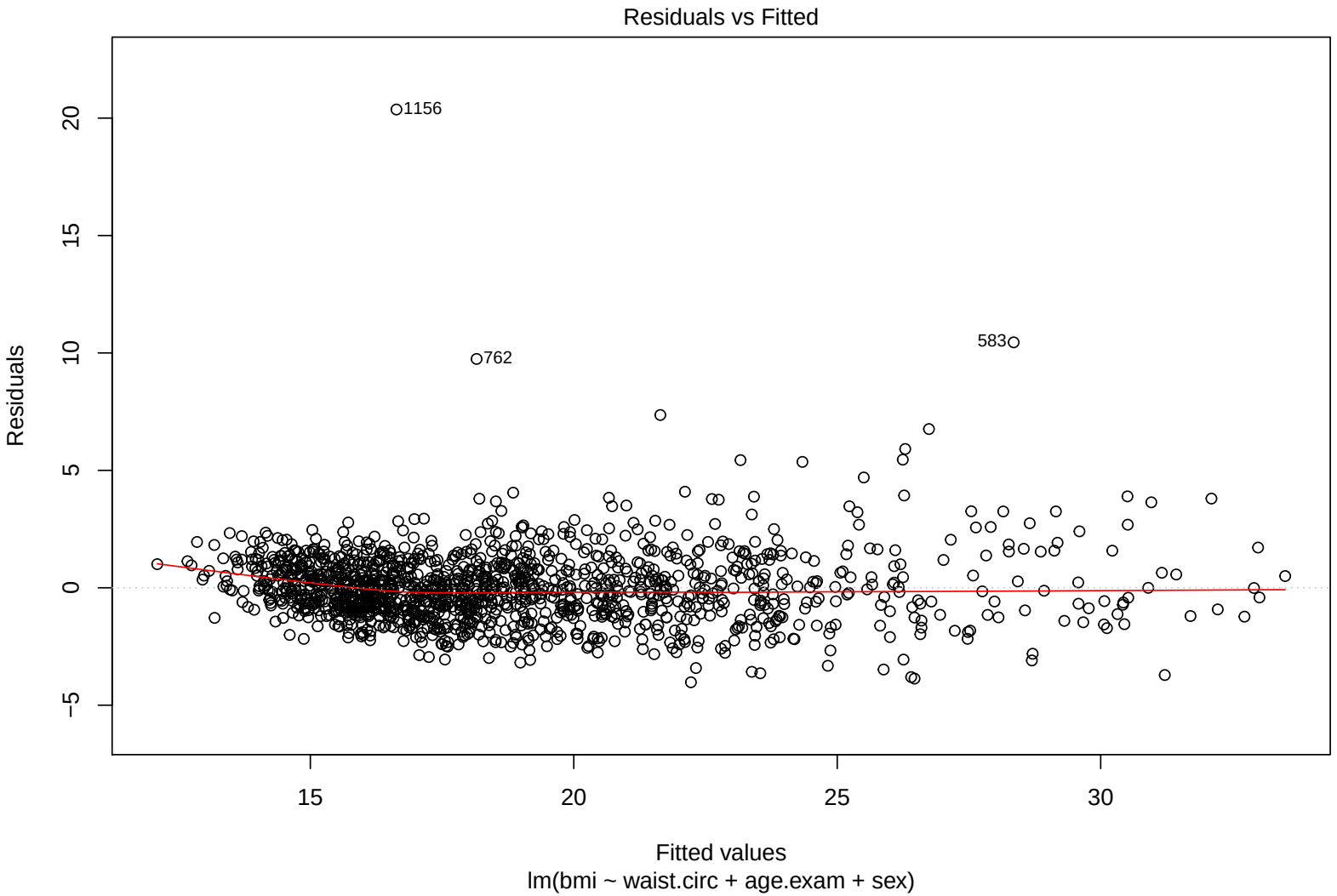
```
  r.squared adj.r.squared    sigma statistic p.value
1 0.8281523     0.8280307 1.692336  6814.214       0
  df    logLik       AIC      BIC deviance df.residual
1  2 -2753.188 5512.376 5528.143   4049.7        1414
```

Augmenting the Data with Model Results (broom::augment) run with warning = FALSE

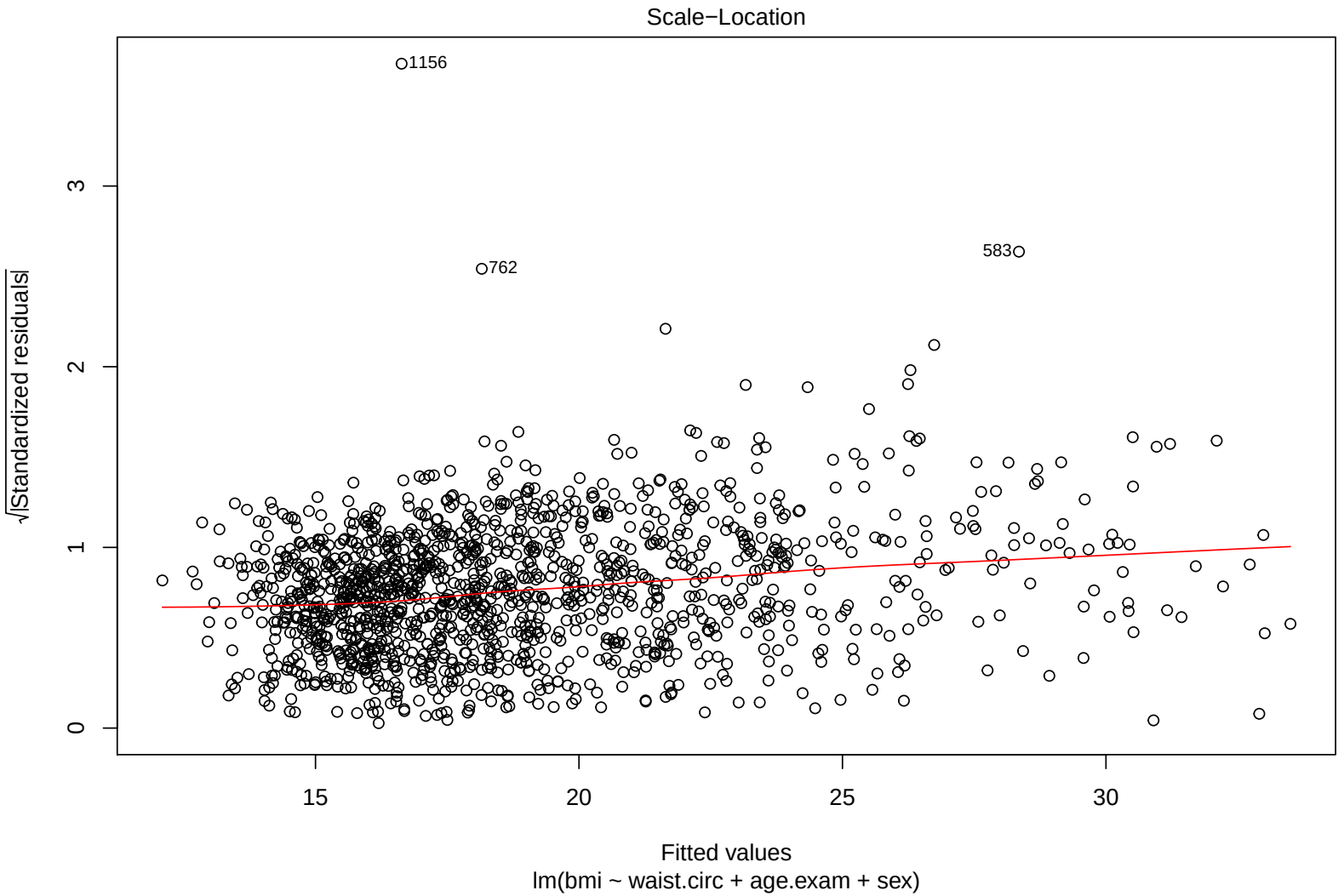
```
newdat2 <- augment(m2) ## again from broom
head(newdat2,3)
```

```
   bmi waist.circ age.exam    sex  .fitted    .se.fit
1 22.3        71.9        8 Female 21.25663 0.06927752
2 19.8        79.4       14 Female 22.10992 0.08007933
3 15.2        46.8        3   Male 14.16688 0.08724867
      .resid         .hat    .sigma       .cooksd
1  1.043374 0.002107399 1.509382 0.0002529070
2 -2.309917 0.002815807 1.508381 0.0016586209
3  1.033124 0.003342564 1.509387 0.0003942708
  .std.resid
1  0.6921161
2 -1.5328151
3  0.6857415
```

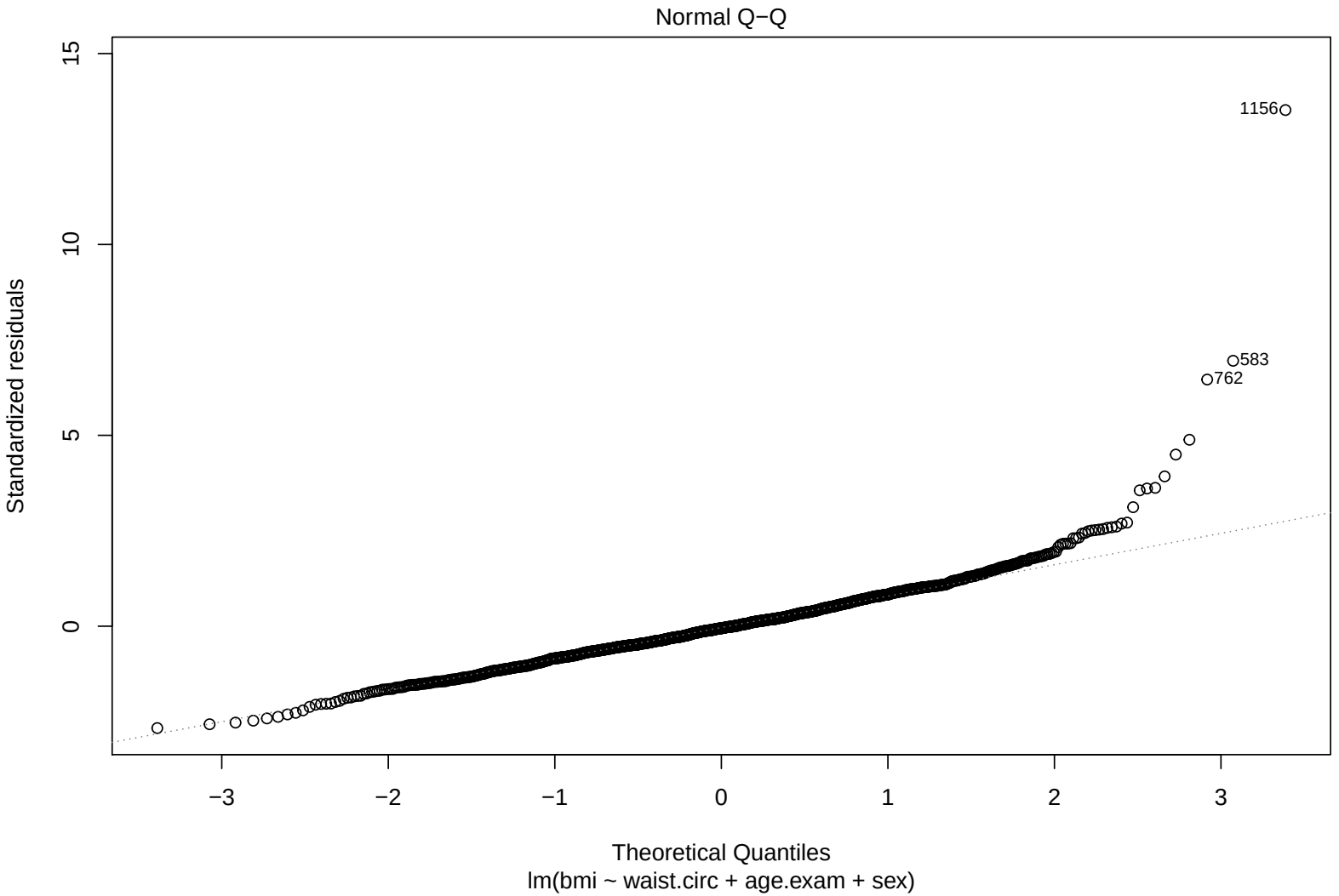
Residuals vs. Fitted Values `plot(m2, which = 1)`



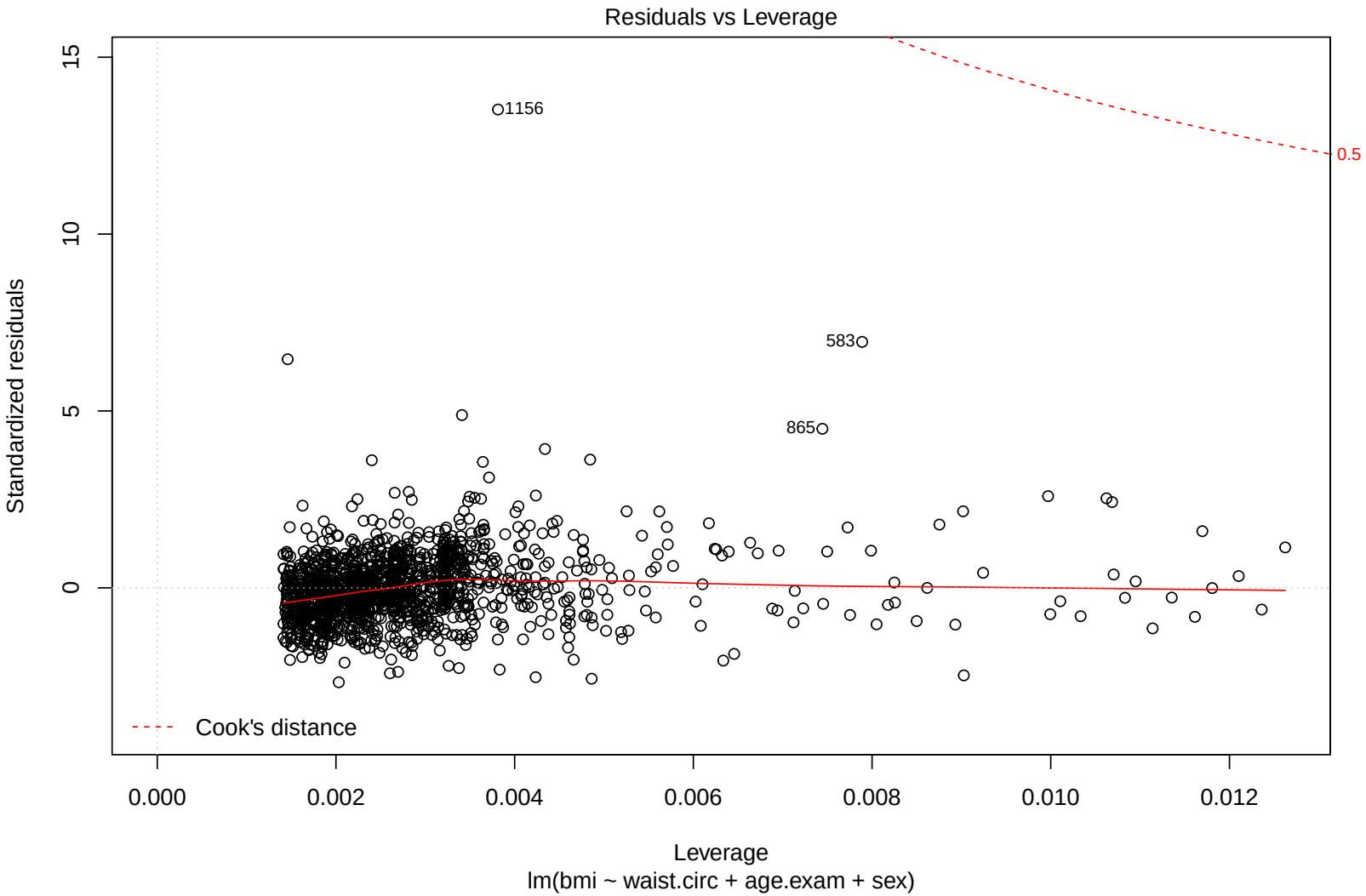
Scale-Location Plot `plot(m2, which = 3)`



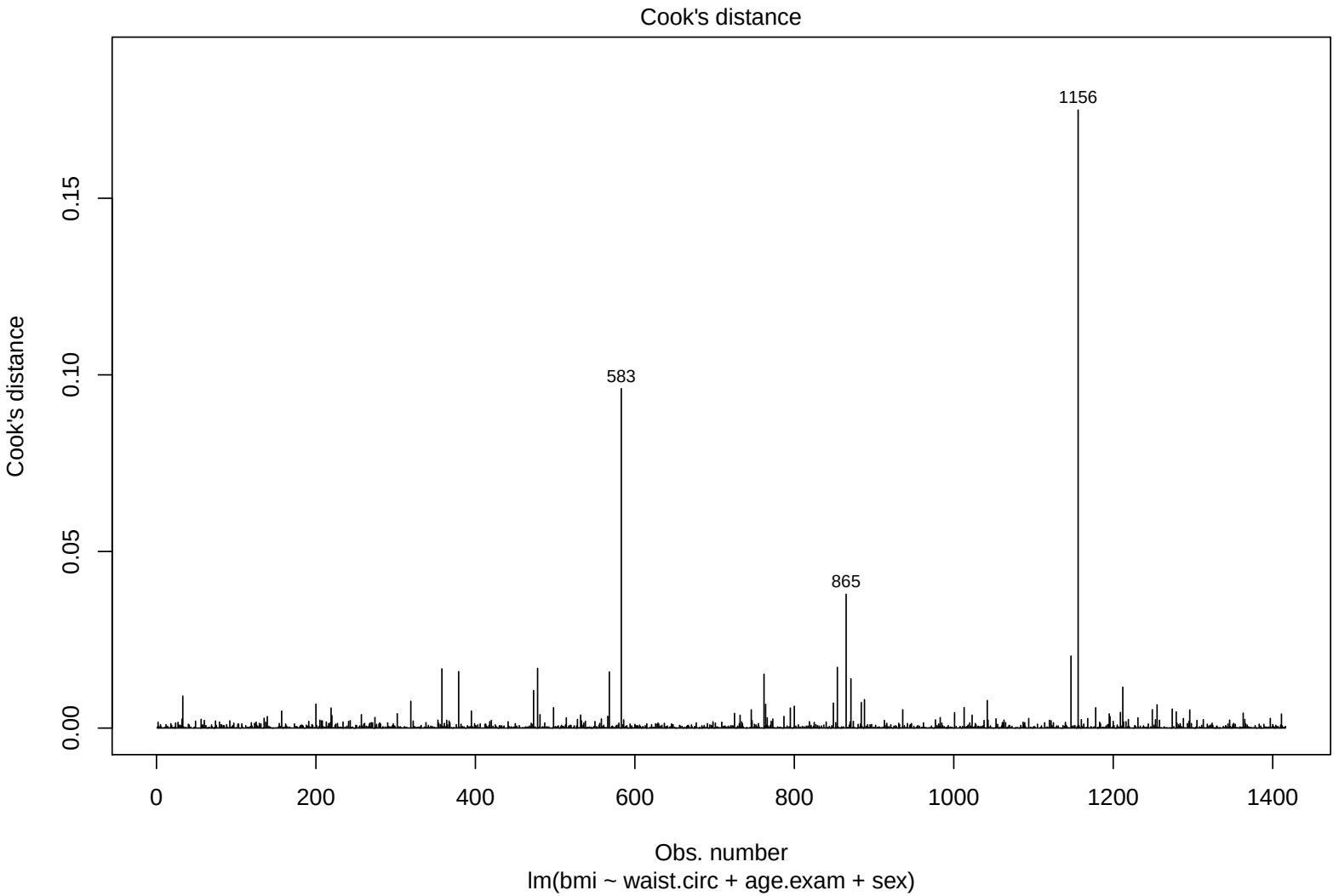
Standardized Residuals `plot(m2, which = 2)`



Residuals, Leverage, Influence `plot(m2, which = 5)`



Cook's Distance Index Plot `plot(m2, which = 4)`



Coming Soon (and see Notes, Sections 37-end)

- What happens when a linear model doesn't fit so well?
  - Should we be modeling a transformed outcome?
  - Box-Cox approach to transformation decisions
- Incorporating multi-categorical predictors
- Model Validation: Making predictions in new data
  - Simplest Smart Approach: Split data into Training, Test samples
- Variable Selection: Stepwise and better approaches
- Standardizing Regression Coefficients
- Dealing with Missingness Sensibly