431 Class 24

Thomas E. Love

2018-11-29

Today's Agenda

- Regression Analysis: What is new today?
  - Box-Cox approach to identifying sensible re-expressions
  - Analysis of Variance to compare models
  - Collinearity and the Variance Inflation Factor
  - Stepwise Regression to help identify predictor sets
  - Imputation and its impact on the model
  - Making predictions / prediction vs. confidence intervals
- Modeling in the National Youth Fitness Survey data

Today's R Setup

```
library(GGally); library(car); library(simputation)
library(janitor); library(broom); library(magrittr)
library(tidyverse) # always load tidyverse last

nnyfs_raw <- read_csv("data/nnyfs.csv") %>%
  clean_names() %>%
  mutate_if(is.character, as.factor) %>%
  mutate(seqn = as.character(seqn)) %>%
  mutate(bmi_cat = fct_relevel(bmi_cat, "Obese",
          "Overweight", "Normal weight", "Underweight")) %>%
  select(seqn, plank, age, gender, reth, inc_cat, incvspov,
         bmi_cat, waist, mealsout, calories, sugar)
```

Initial Look at the Data

```
glimpse(nnyfs_raw)
```

```
Observations: 1,352
Variables: 12
$ seqn      <chr> "71918", "71919", "71920", "71921...
$ plank     <int> 45, 121, 45, 11, 107, 127, 44, 18...
$ age       <int> 8, 14, 15, 3, 12, 12, 8, 7, 8, 9,...
$ gender    <fct> Female, Female, Female, Male, Mal...
$ reth      <fct> Non-Hispanic Black, Non-Hispanic ...
$ inc_cat   <fct> Above 75K, Above 75K, 20 to 44K, ...
$ incvspov <dbl> 5.00, 5.00, 0.87, 4.34, 5.00, 5.0...
$ bmi_cat   <fct> Obese, Normal weight, Obese, Norm...
$ waist     <dbl> 71.9, 79.4, 96.4, 46.8, 90.0, 72....
$ mealsout <int> 2, 3, 2, 1, 1, 2, 1, 0, 2, 0, 0, ...
$ calories <dbl> 1725, 2304, 1114, 1655, 2920, 175...
$ sugar     <dbl> 118.68, 81.38, 119.25, 90.35, 309...
```

Codebook

Name	Type	Description	Original Source
seqn	character	ID code	SEQN
plank	integer	# of seconds plank position is held, range from 1 to 450	MPXPLANK
age	integer	3 to 16, although just a few were 16, in years	RIDEXAGY
gender	2-level	Female or Male	RIAGENDR
reth	4-level	Hispanic, Non-Hispanic Black, Non-Hispanic White, Other Race	RIDRETH1
inc_cat	4-level	Below $20K, 20 to 44K, 45 to 74K, Above 75K	INDFMIN2
incvspov	quantity	ranges from 0 to 5.00 [multiple of poverty level]	INDFMPIR
bmi_cat	4-level	Underweight, Normal weight, Overweight, Obese	BMDBMIC
waist	quantity	ranges from 44.1 to 144.7, in cm	BMXWAIST
mealsout	quantity	# of meals not home prepared last 7 days, ranges from 0 to 20	DBD895
calories	quantity	ranges from 257 to 5265 kcal, from dietary recall	DR1TKCAL
sugar	quantity	ranges from 1 to 405.5 grams, from dietary recall	DR1TSUGR

- Data Source: https://www.cdc.gov/nchs/nnyfs/index.htm
- Demographics (DEMO), Dietary (DR1TOT), Examination (BMX and PLX), Questionnaire (DBQ, HUQ) files imported into R as SAS .xpt files using the `haven` package's `read_xpt` function.

Plank Details

Participants were instructed to lie face down on the mat resting on their elbows with their hands on the floor and their toes curled under their feet so that some of their weight was on the balls of their feet. Then they were told to tighten their stomach muscles and the muscles along the front of their thighs. Next, they were told to push off the floor and rise up onto their toes, keeping their elbows on the floor and their back straight.

Participants were instructed to hold this position for as long as they could without letting their hips drop towards the floor or their knees bend. They were given one practice plank test before beginning the measured test. Participants were instructed to correct their position if they wobbled or moved out of position during the measured test. If it happened a second time the test was stopped. The test ended either when participants could no longer maintain the correct position, or when they requested the test be stopped.

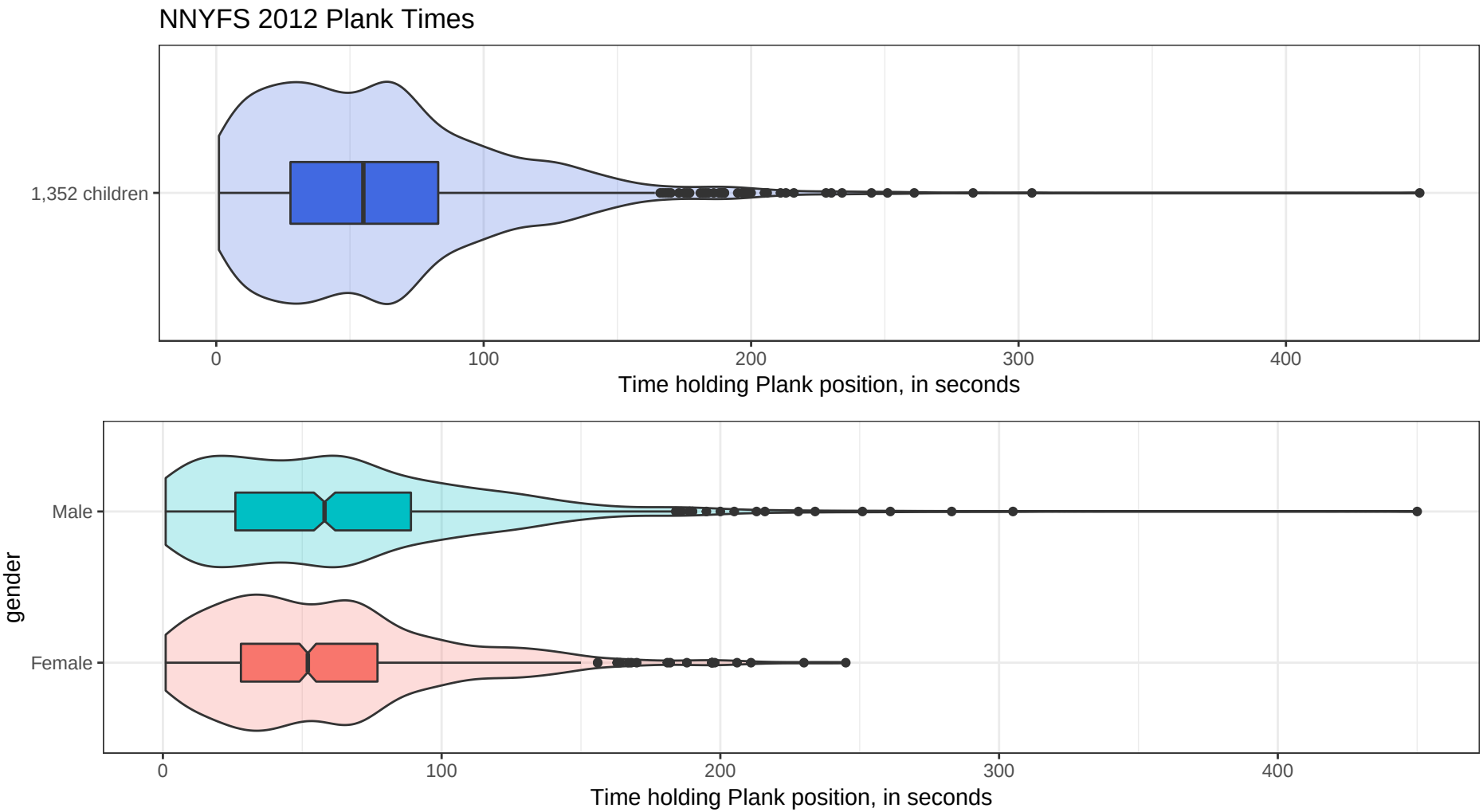
The number of seconds the plank position was held was recorded.

Our Modeling Goal

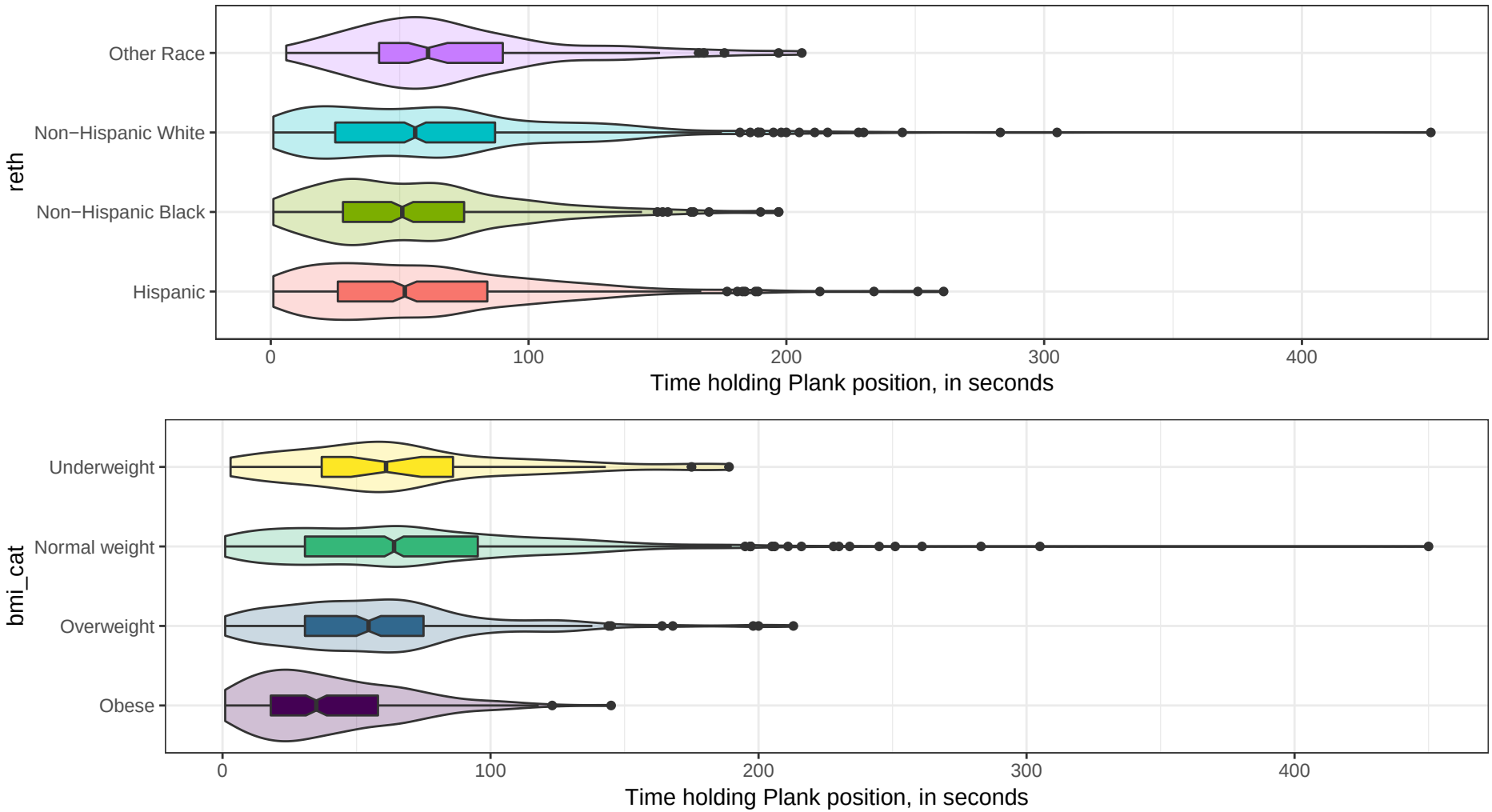
Can we predict `plank` time using some/all of these 9 predictors?

- `age`, `gender`, `reth`
- `incvspov`, `mealsout`
- `bmi_cat`, `waist`
- `calories`, `sugar`

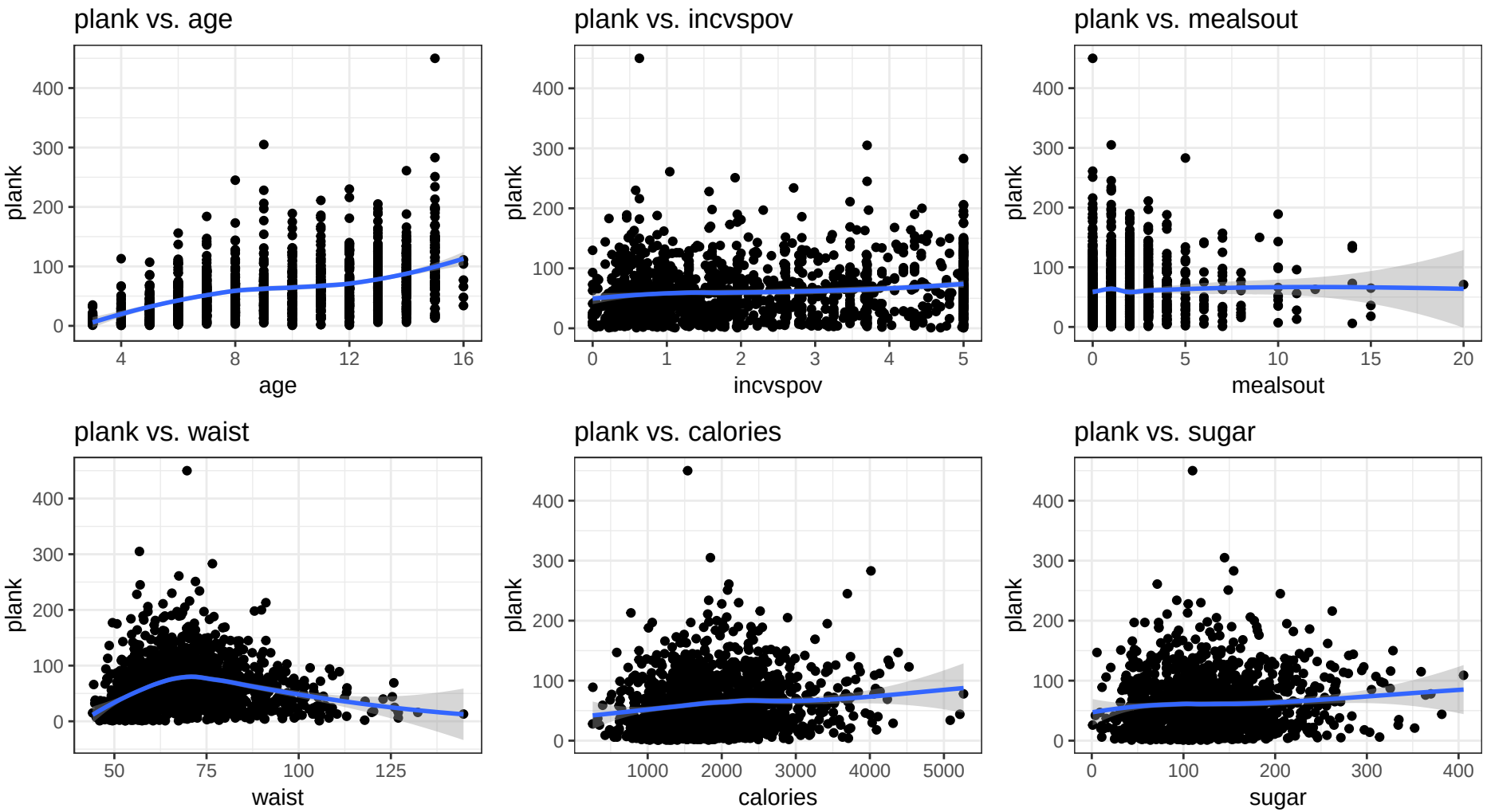
plank Times, and plank Times by gender



plank Times by reth and by bmi_cat



plank vs. the other 6 candidate predictors



Several warning messages suppressed here. (`warning = FALSE`)

Is Imputation Necessary?

```
colSums(is.na(nnyfs_raw))
```

```
    seqn    plank      age   gender     reth  inc_cat
       0        0        0        0        0        0
incvspov  bmi_cat    waist mealsout calories    sugar
      48        0        0        7        0        0
```

```
nnyfs_raw %>% count(is.na(incvspov), is.na(mealsout))
```

```
# A tibble: 4 x 3
  `is.na(incvspov)` `is.na(mealsout)`     n
  <lgl>             <lgl>             <int>
1 FALSE             FALSE              1299
2 FALSE             TRUE                  5
3 TRUE              FALSE                46
4 TRUE              TRUE                  2
```

Impute missing values for incvspov and mealsout

```
set.seed(20181129)
nnyfs_imp <- nnyfs_raw %>%
  impute_rlm(incvspov ~ inc_cat) %>%
  impute_rlm(mealsout ~ age + incvspov + calories)

colSums(is.na(nnyfs_imp))
```

```
    seqn    plank      age   gender     reth  inc_cat
       0        0        0        0        0        0
incvspov  bmi_cat    waist mealsout calories    sugar
       0        0        0        0        0        0
```

Partition the data

The nnyfs_imp data set contains 1352 observations on 12 variables.

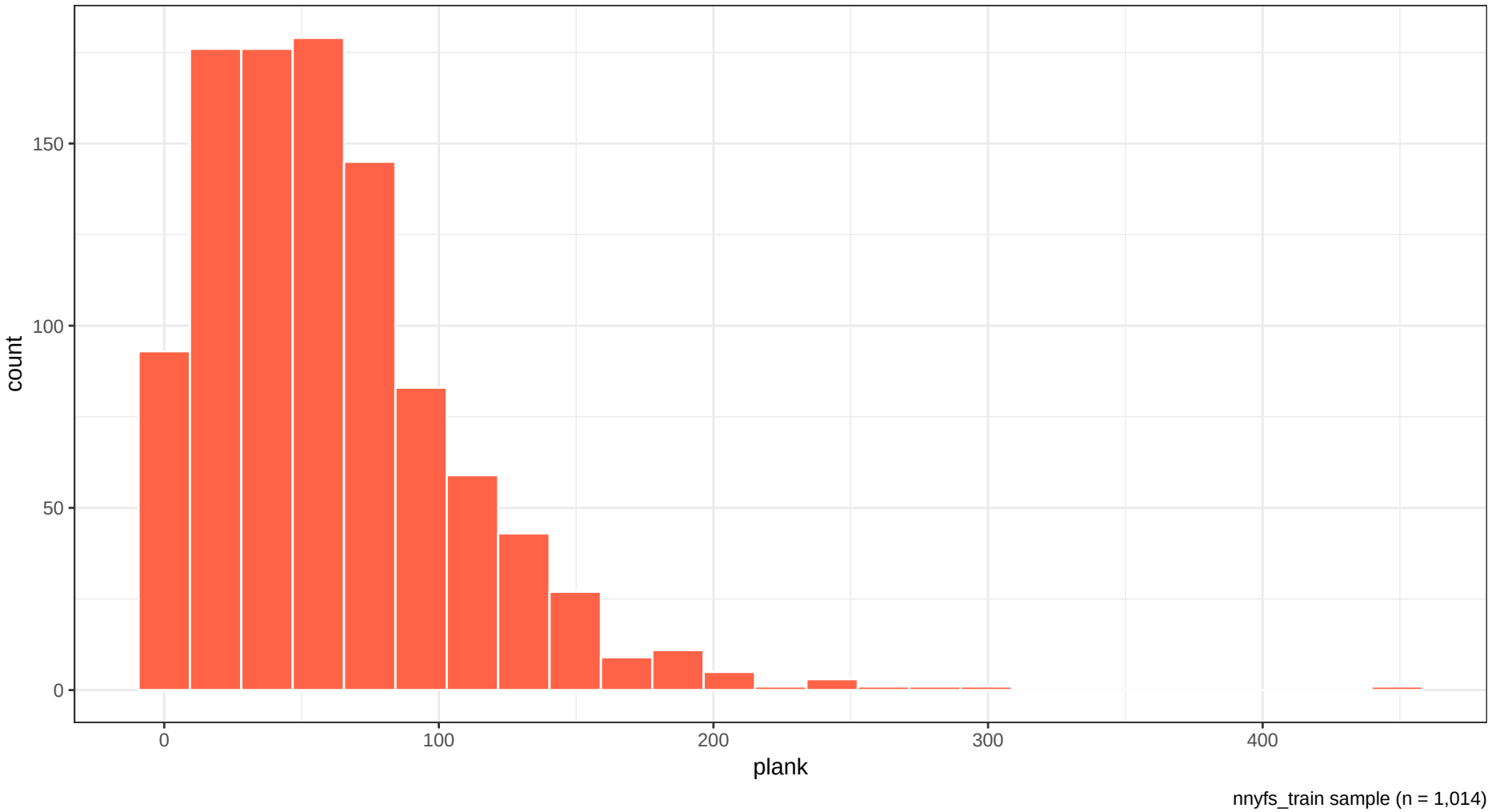
```
set.seed(20181129)
nnyfs_train <- sample_frac(nnyfs_imp, size = 0.75)
nnyfs_test <- anti_join(nnyfs_imp, nnyfs_train,
                        by = "seqn")
```

```
dim(nnyfs_train); dim(nnyfs_test)
```

```
[1] 1014   12
```

```
[1] 338  12
```

Distribution of `plank` - do we need to transform?



Transforming / Re-expressing our Outcome?

We can use the Box-Cox family of transformations to isolate specific choices of the power transformation parameter λ for re-expressing our quantitative outcome which might lead to a more effective (yet still interpretable) model.

This approach is appropriate for strictly **positive** outcomes. If our minimum value is -14, we might add 15 to each observation before using Box-Cox.

Ladder of Power Transformations

Power (λ)	Transformation
2	y^2
1	y (untransformed)
0.5	$\sqrt{y}$
0	$\log y$
-1	$\frac{1}{y}$

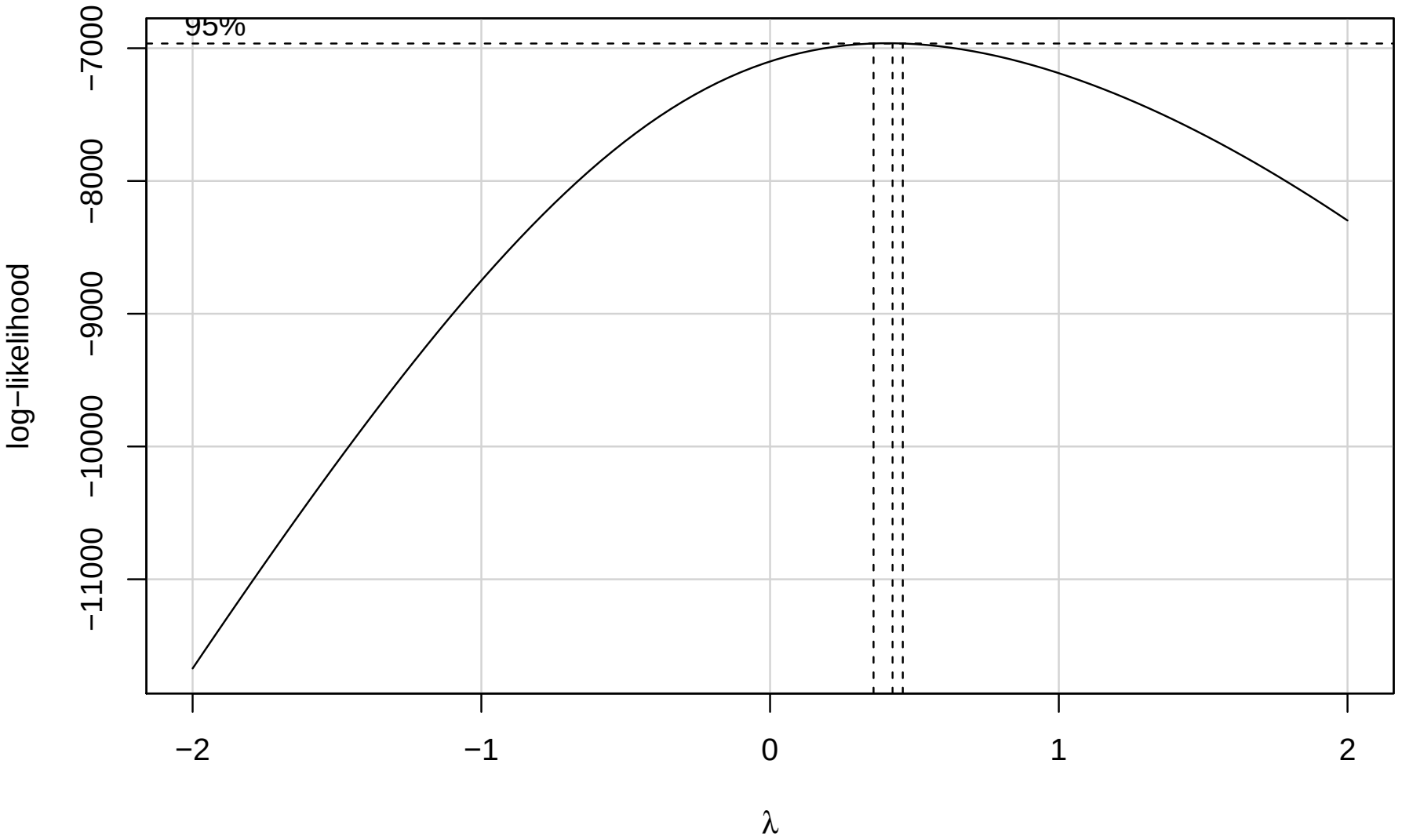
From the `car` package, we use `boxCox` and `powerTransform`.

Using the Box-Cox approach to pick a transformation

```
m_start <- lm(plank ~ age + gender + reth + incvspov +
                mealsout + bmi_cat + waist + calories +
                sugar, data = nnyfs_train)

boxCox(m_start)

powerTransform(m_start)
```

- Results on next two slides

Box-Cox Plot based on `m_start`



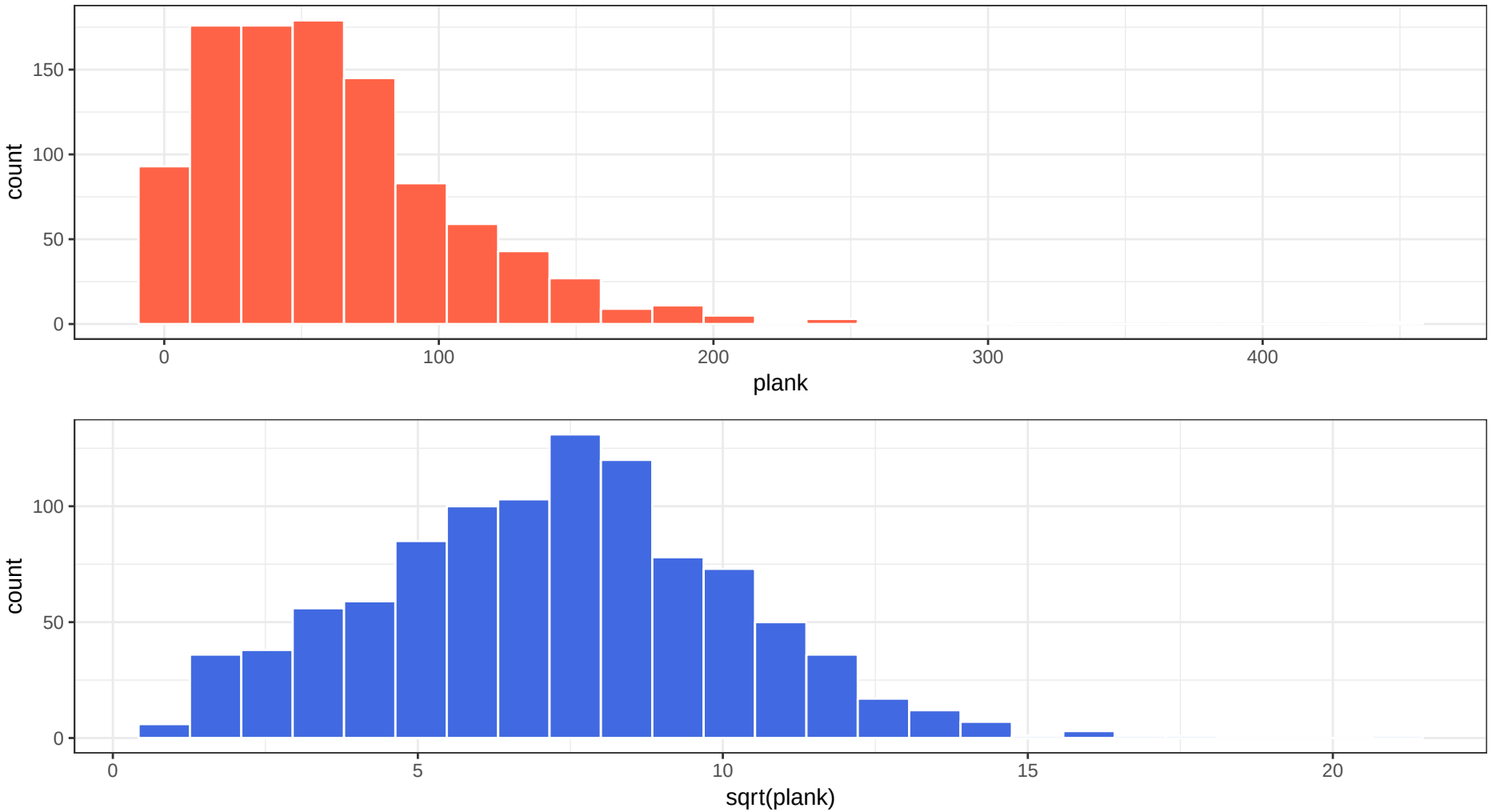
Suggested Power Transformation

```
powerTransform(m_start)
```

```
Estimated transformation parameter
       Y1
0.4075796
```

Power (λ)	Transformation
2	y^2
1	y (untransformed)
0.5	$\sqrt{y}$
0	$\log y$
-1	$\frac{1}{y}$

Training Sample: `plank` and its square root



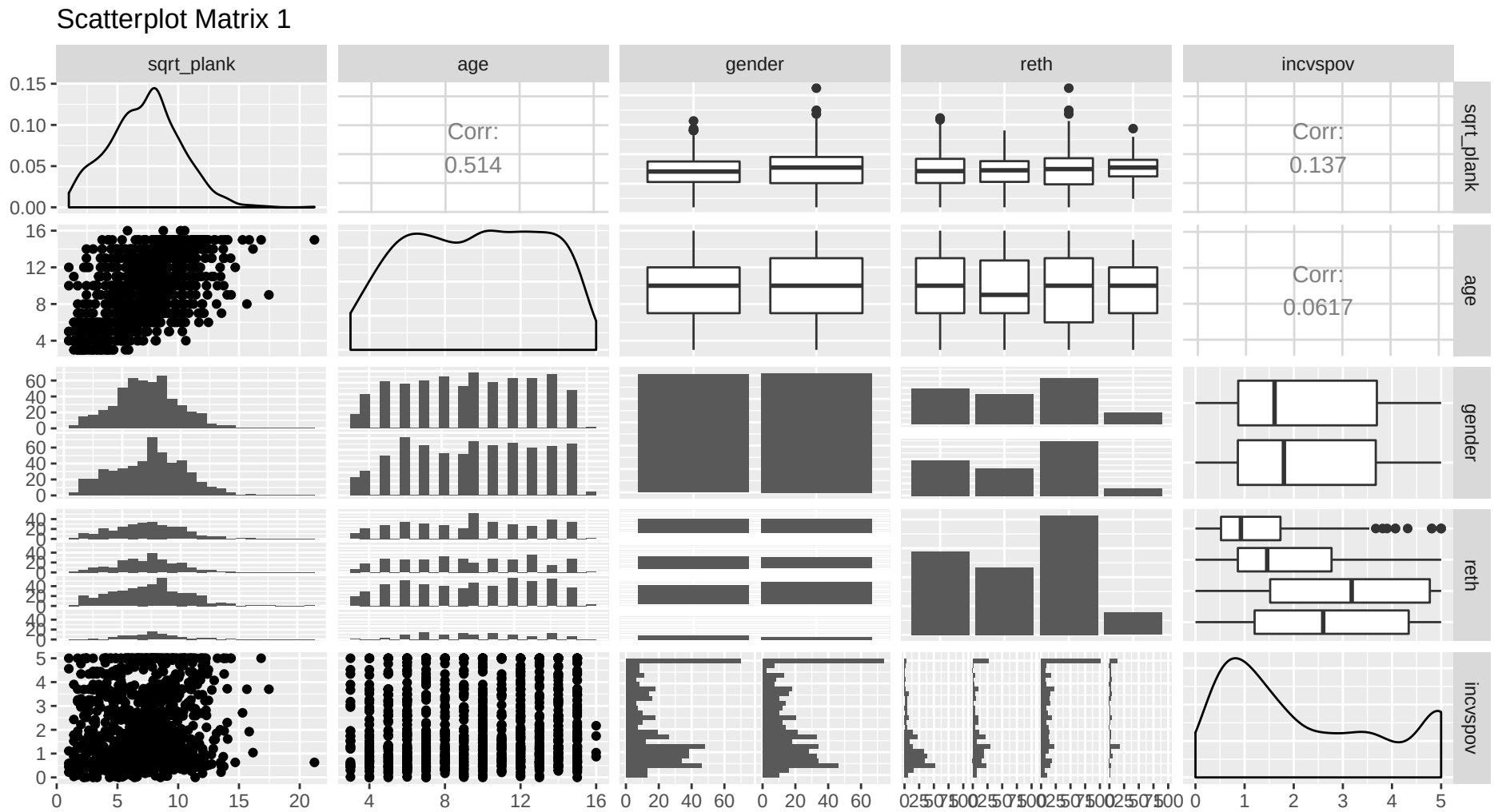
Code for Scatterplot Matrix Plots

```
nnyfs_train <- nnyfs_train %>%
  mutate(sqrt_plank = sqrt(plank))

nnyfs_train %>%
  select(sqrt_plank, age, gender, reth, incvspov) %>%
  ggpairs(., title = "Scatterplot Matrix 1",
          lower = list(combo = wrap("facethist", bins = 25)))
```

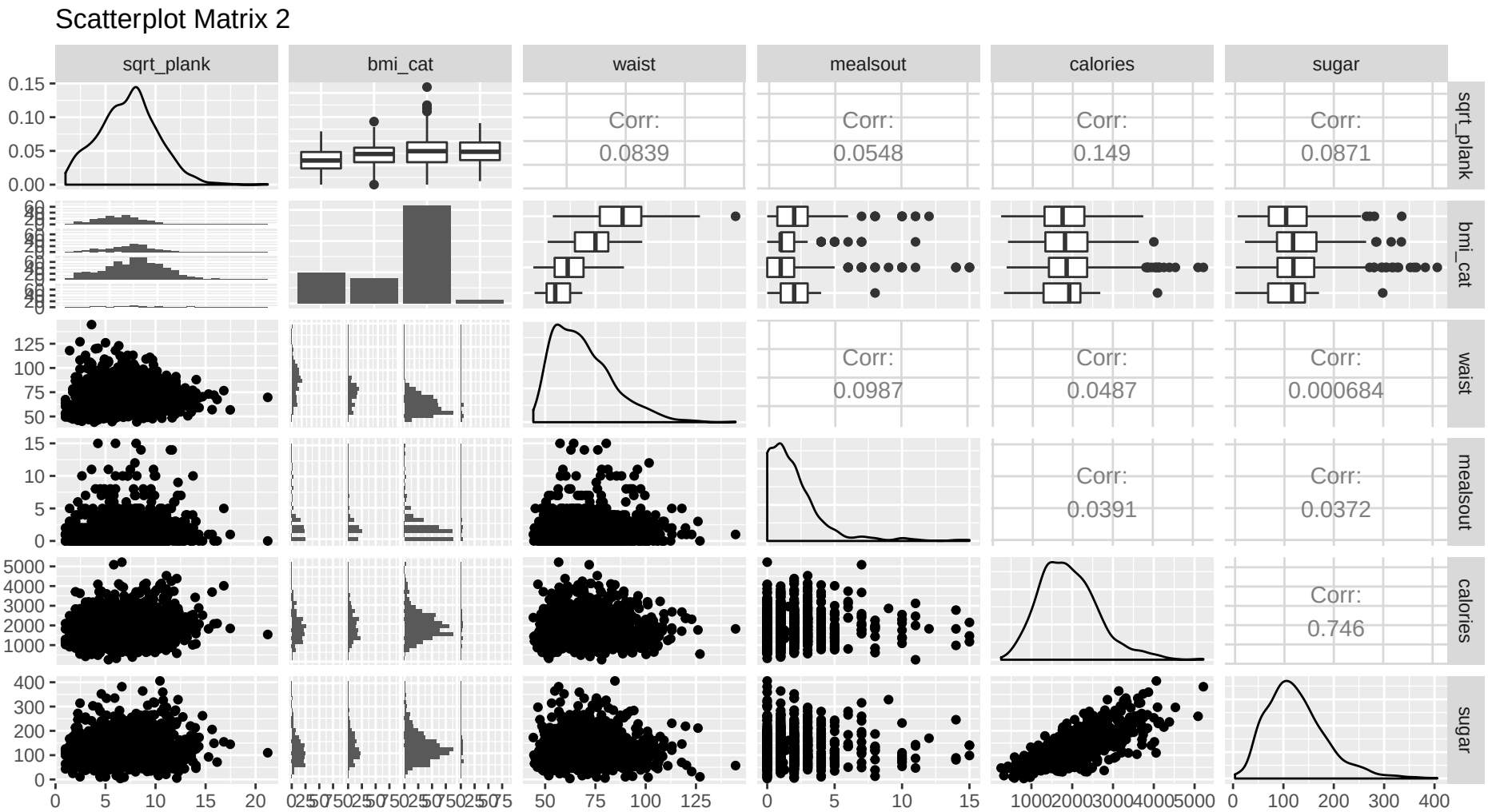
just building sqrt_plank for these plots, then...

```
nnyfs_train %>%
  select(sqrt_plank, bmi_cat, waist, mealsout,
         calories, sugar) %>%
  ggpairs(., title = "Scatterplot Matrix 2",
          lower = list(combo = wrap("facethist", bins = 25)))
```

Scatterplot Matrix 1 (outcome + 4 predictors)



Matrix 2 (outcome + other 5 predictors)



Kitchen Sink Model

```
m_ks <- lm(sqrt(plank) ~ age + gender + reth +
             incvspov + mealsout + bmi_cat + waist +
             calories + sugar, data = nnyfs_train)
```

Tidied `m_ks` coefficients

term	estimate	conf.low	conf.high	p.value
(Intercept)	6.410	4.859	7.961	0.000
age	0.660	0.588	0.732	0.000
genderMale	0.253	-0.041	0.548	0.092
rethNon-Hispanic Black	-0.303	-0.719	0.113	0.153
rethNon-Hispanic White	-0.093	-0.487	0.302	0.645
rethOther Race	0.047	-0.562	0.655	0.880
incvspov	0.140	0.039	0.242	0.007
mealsout	0.024	-0.045	0.093	0.494
bmi_catOverweight	0.030	-0.546	0.606	0.919
bmi_catNormal weight	-0.168	-0.825	0.489	0.616
bmi_catUnderweight	-0.808	-1.995	0.378	0.182
waist	-0.087	-0.110	-0.065	0.000
calories	0.000	0.000	0.001	0.106
sugar	-0.002	-0.005	0.002	0.339

Snowflakes become seductive. . .

```
Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                6.2420789  0.5383140  11.596  < 2e-16 ***
age                        0.6601699  0.0365789  18.048  < 2e-16 ***
genderMale                 0.2531160  0.1500823   1.687  0.09201 .
rethNon-Hispanic Black    -0.3028281  0.2118678  -1.429  0.15322
rethNon-Hispanic White    -0.0926926  0.2008961  -0.461  0.64461
rethOther Race             0.0466566  0.3102282   0.150  0.88048
incvspov                   0.1404188  0.0515328   2.725  0.00655 **
mealsout                   0.0241090  0.0352040   0.685  0.49360
bmi_catObese               0.1678230  0.3349159   0.501  0.61642
bmi_catOverweight          0.1976904  0.2378628   0.831  0.40611
bmi_catUnderweight        -0.6406177  0.4832587  -1.326  0.18527
waist                     -0.0870328  0.0114502  -7.601 6.75e-14 ***
calories                   0.0002521  0.0001559   1.617  0.10625
sugar                     -0.0017759  0.0018553  -0.957  0.33869
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Summarizing the Fit of the Kitchen Sink Model

```
glance(m_ks) %>%
  select(r.squared, adj.r.squared, sigma,
         AIC, BIC, p.value) %>%
  knitr::kable(digits = 3)
```

	r.squared	adj.r.squared	sigma	AIC	BIC	p.value
value	0.379	0.371	2.334	4612.306	4686.131	0

or, from `summary(m_ks)`, we have:

```
Residual standard error: 2.334 on 1000 degrees of freedom
Multiple R-squared: 0.3787, Adjusted R-squared:  0.3707
F-statistic:  46.9 on 13 and 1000 DF,  p-value: < 2.2e-16
```

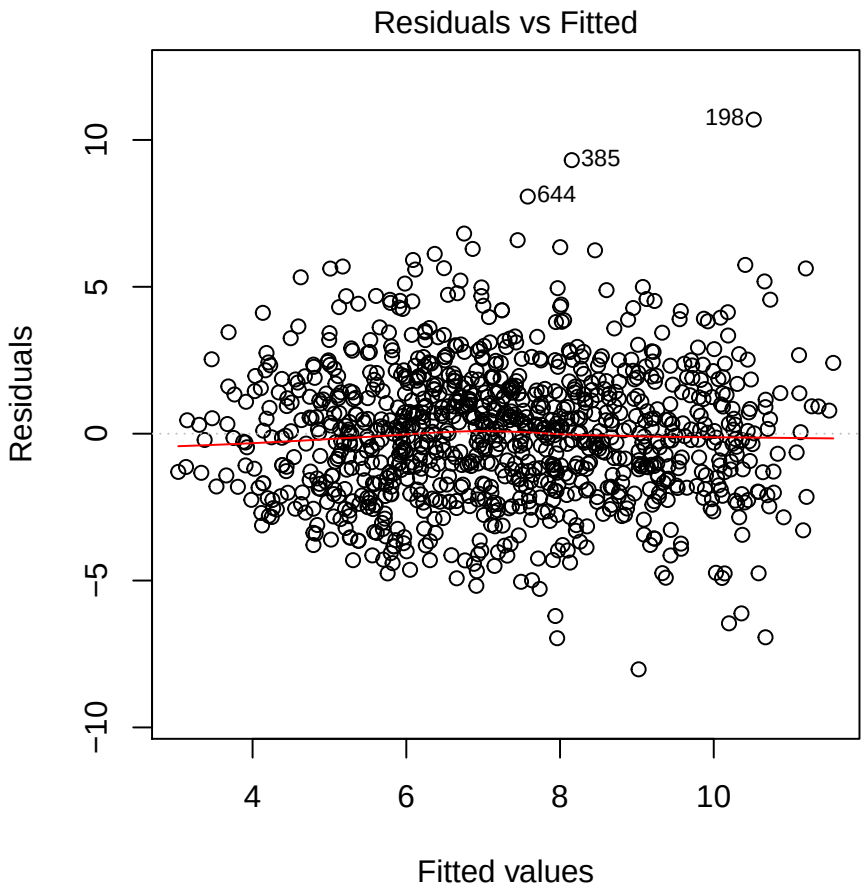
ANOVA testing for Kitchen Sink Model

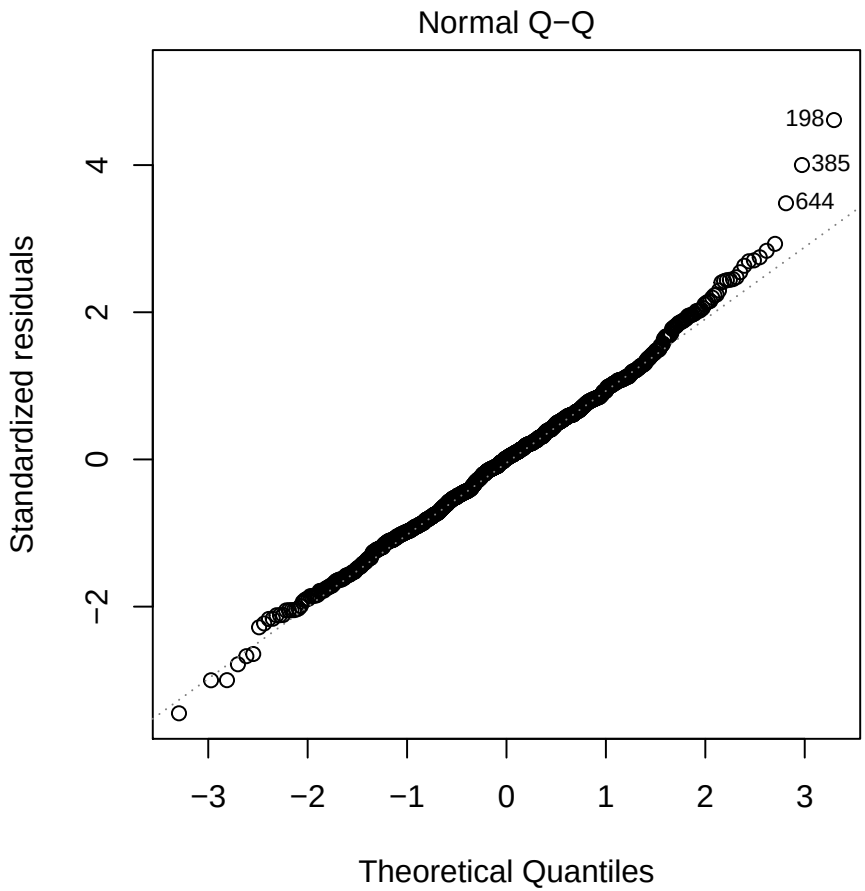
```
anova(m_ks)
```

```
Analysis of Variance Table

Response: sqrt(plank)
             Df Sum Sq Mean Sq  F value    Pr(>F)
age           1 2314.0 2314.01 424.8231 < 2.2e-16 ***
gender        1   12.7   12.74   2.3383    0.1265
reth          3   22.1    7.38   1.3554    0.2551
incvspov      1   84.0   84.01  15.4241 9.178e-05 ***
mealsout      1    0.3    0.25   0.0466    0.8291
bmi_cat       3  560.6  186.86  34.3045 < 2.2e-16 ***
waist         1  312.0  312.04  57.2869 8.535e-14 ***
calories      1   10.0    9.96   1.8277    0.1767
sugar         1    5.0    4.99   0.9163    0.3387
Residuals  1000 5447.0    5.45
---
```

Residual Plots for `m_ks`? (training sample)

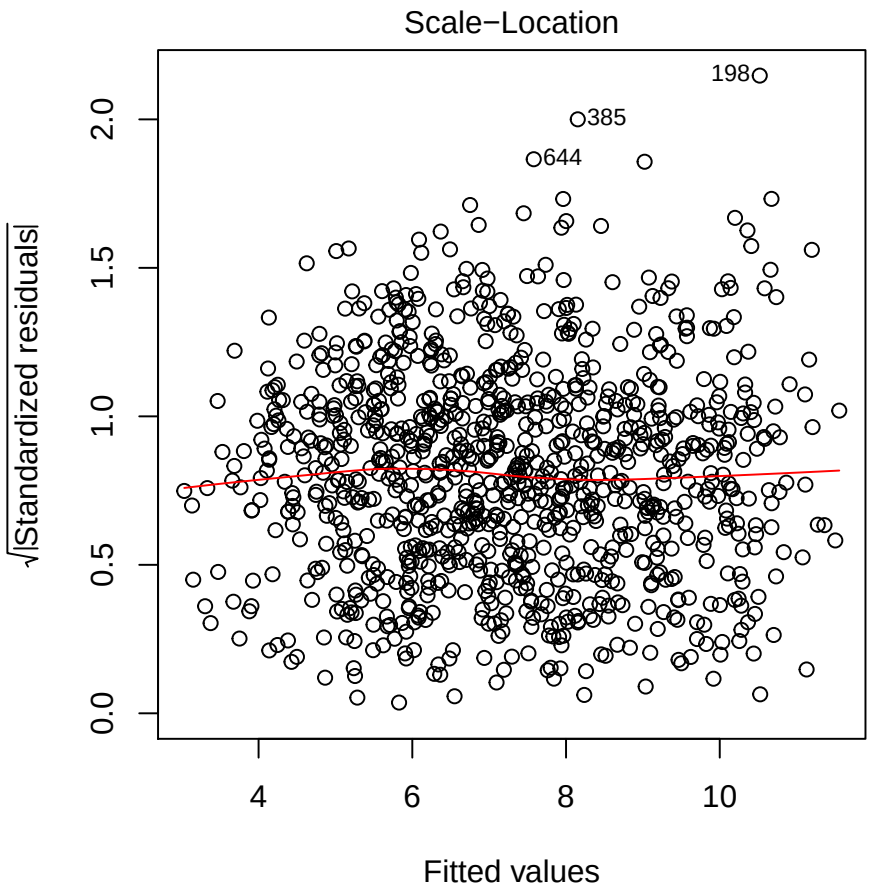
```
par(mfrow = c(1,2))
plot(m_ks, which = 1:2)
```

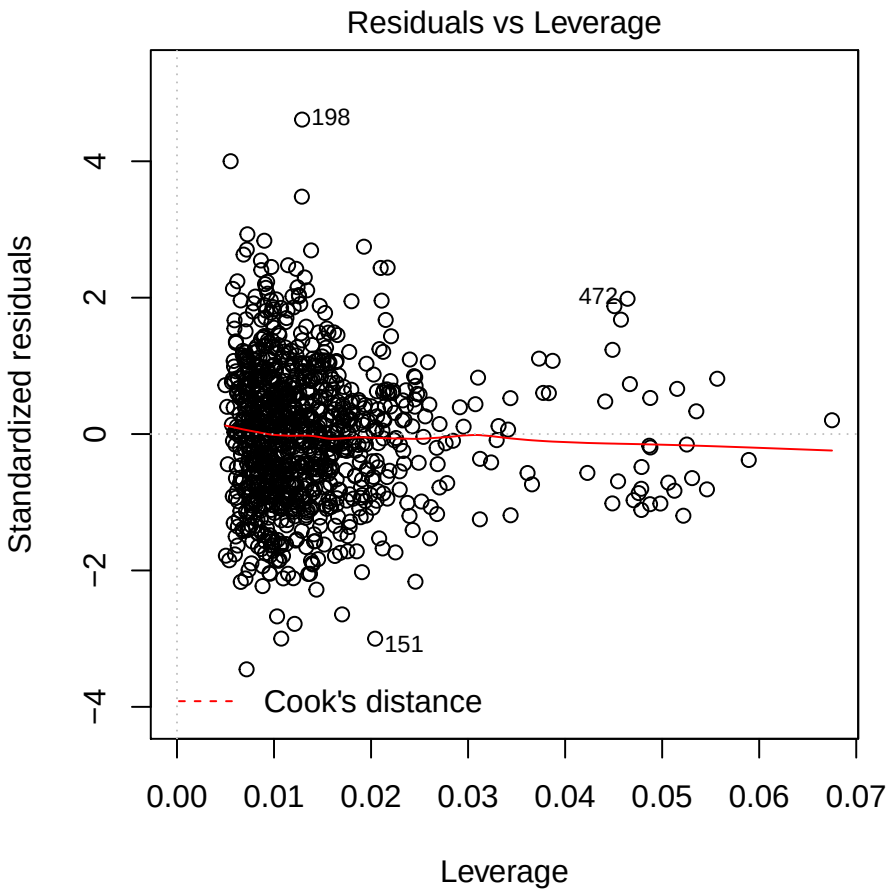




Checking m_ks Residual Plots (training sample)

```
par(mfrow = c(1,2))
plot(m_ks, which = c(3, 5))
```





Who is observation 198?

```
nnyfs_train %>% slice(198) %>%
  select(seqn, plank, age, gender, reth, bmi_cat) %>%
  knitr::kable()
```

seqn	plank	age	gender	reth	bmi_cat
72184	450	15	Male	Non-Hispanic White	Normal weight

```
mosaic::favstats(~ plank, data = nnyfs_train)
```

```
 min Q1 median Q3 max     mean       sd    n missing
   1 28     54 83 450 61.30276 46.34275 1014       0
```

Collinearity (Correlated Predictors) and VIF

If two predictors (say A and B) are highly correlated (collinear) with each other, then the predictive value of the second one into the model (B) will be masked by its strong correlation with A (since A is already in the model.)

- When we have larger models, it's helpful to look at the impact on the standard errors for the coefficient estimates that collinearity contributes. We'll do this using the **variance inflation factor** or VIF.

The `car` package provides a VIF calculation for us, that applies to both simple settings (all quantitative or binary variables) and a generalized VIF (when multi-categorical predictors are involved)

- In either case, a VIF exceeding 5 is of some concern.
- Should we see a large VIF, it usually indicates that we would be better off including fewer of the predictors in the model, so that we avoid some of this masking.

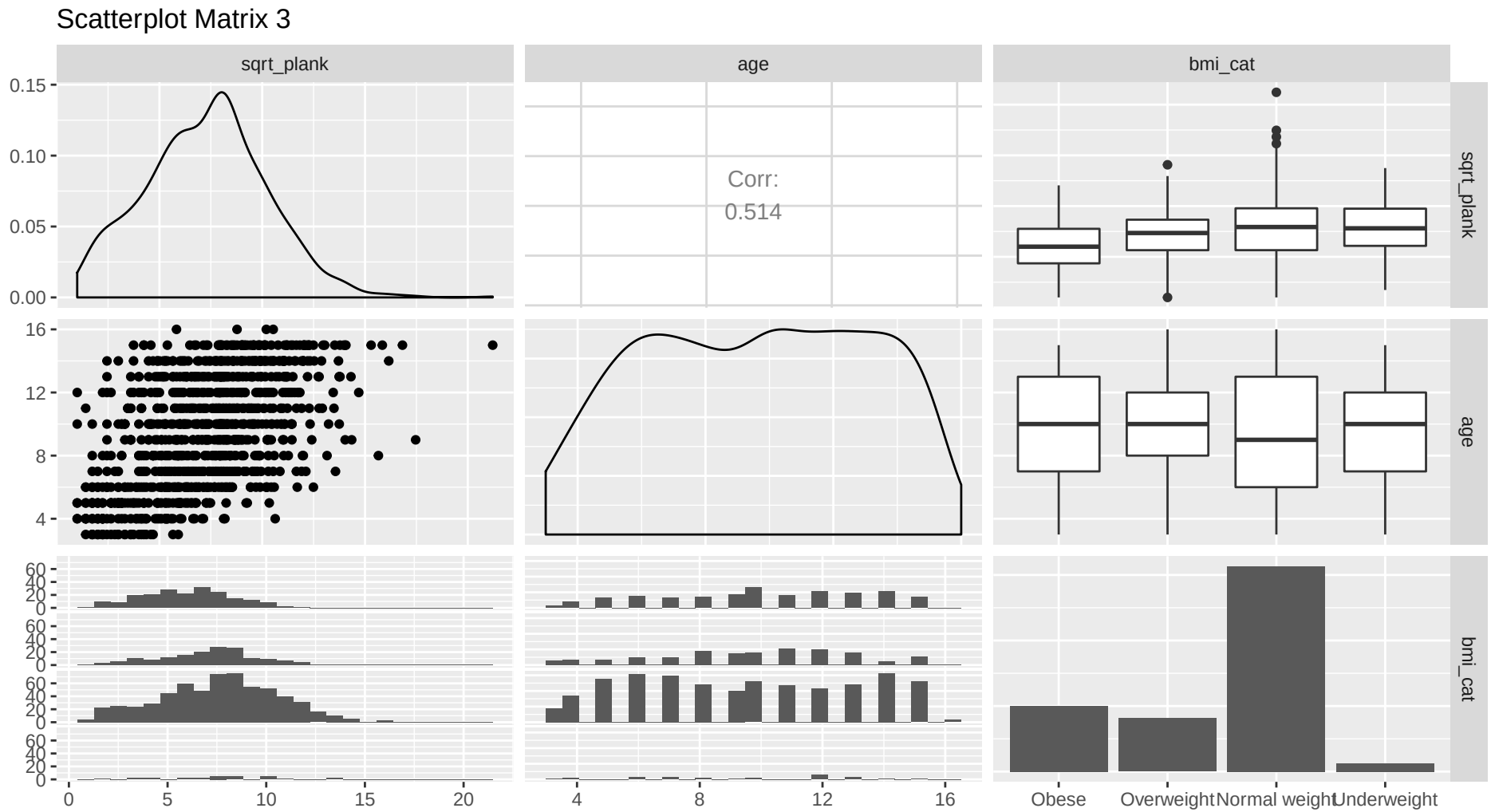
Using `vif` from `car` to assess collineaity

```
vif(m_ks)
```

```
                GVIF Df GVIF^(1/(2*Df))
age         3.017279  1         1.737032
gender      1.048247  1         1.023840
reth        1.394710  3         1.057014
incvspov    1.317244  1         1.147712
mealsout    1.059032  1         1.029093
bmi_cat     3.356962  3         1.223651
waist       5.516333  1         2.348687
calories    2.375051  1         1.541120
sugar       2.316816  1         1.522109
```

- Lack of collinearity isn't technically an assumption of regression, but avoiding collinearity is a way to make things more interpretable, and avoid certain kinds of overfitting.

A Smaller (Two-Predictor) Model?



Build Two-Predictor Model

```
m_2 <- lm(sqrt(plank) ~ age + bmi_cat, data = nnyfs_train)

m_2
```

```
Call:
lm(formula = sqrt(plank) ~ age + bmi_cat, data = nnyfs_train)

Coefficients:
          (Intercept)                   age
               1.5359                0.4475
   bmi_catOverweight  bmi_catNormal weight
               1.2214                1.9356
  bmi_catUnderweight
               1.9604
```

Is collinearity still an issue in m_2?

```
vif(m_2)
```

```
           GVIF Df GVIF^(1/(2*Df))
age      1.003932  1        1.001964
bmi_cat  1.003932  3        1.000654
```

Tidied m_2 coefficients

```
tidy(m_2, conf.int = TRUE) %>%
  select(term, estimate, conf.low, conf.high, p.value) %>%
  knitr::kable(digits = 2)
```

term	estimate	conf.low	conf.high	p.value
(Intercept)	1.54	0.99	2.08	0
age	0.45	0.40	0.49	0
bmi_catOverweight	1.22	0.72	1.72	0
bmi_catNormal weight	1.94	1.55	2.32	0
bmi_catUnderweight	1.96	0.96	2.96	0

m_2 predictions for four new kids?

Consider new kids, ages 6, 10, 10 and 14, and the first two are of Normal weight while the latter two are obese. Who does the model predict will hold the plank position longest?

```
newkids <- data_frame(
  age = c(6, 10, 10, 14),
  bmi_cat = c("Normal weight", "Normal weight",
              "Obese", "Obese"))

predict(m_2, newdata = newkids, interval = "prediction",
        level = 0.95)
```

```
       fit      lwr      upr
1 6.156886 1.416619 10.89715
2 7.947083 3.209001 12.68516
3 6.011437 1.265380 10.75749
4 7.801634 3.052421 12.55085
```

ANOVA for m_2?

Are both age and bmi_cat important in my model?

```
anova(m_2)
```

```
Analysis of Variance Table

Response: sqrt(plank)
            Df Sum Sq Mean Sq F value    Pr(>F)
age          1 2314.0 2314.01 397.562 < 2.2e-16 ***
bmi_cat      3  580.8  193.61  33.263 < 2.2e-16 ***
Residuals 1009 5872.9    5.82
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Summarizing the Fit of m_2

```
glance(m_2) %>%
  select(r.squared, adj.r.squared, sigma,
         statistic, p.value) %>%
  knitr::kable(digits = 3)
```

	r.squared	adj.r.squared	sigma	statistic	p.value
value	0.33	0.328	2.413	124.338	0

and we can compare this to the kitchen sink results, below.

	r.squared	adj.r.squared	sigma	statistic	p.value
value	0.379	0.371	2.334	46.896	0

ANOVA comparison of our first two models

Is there a statisically significant difference in prediction quality between the two models?

```
anova(m_ks, m_2)
```

```
Analysis of Variance Table

Model 1: sqrt(plank) ~ age + gender + reth + incvspov + mealsc
    waist + calories + sugar
Model 2: sqrt(plank) ~ age + bmi_cat
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1   1000 5447.0
2   1009 5872.9 -9   -425.89 8.6876 1.234e-12 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Which model does this significance test prefer?

AIC/BIC comparison of our first two models

Which of these models does AIC prefer? How about BIC?

```
AIC(m_ks, m_2); BIC(m_ks, m_2)
```

```
     df      AIC
m_ks 15 4612.306
m_2   6 4670.642
```

```
     df      BIC
m_ks 15 4686.131
m_2   6 4700.172
```

AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) estimate the relative quality of statistical models for the same set of data.

- Each trades off goodness of fit of the model and its simplicity (parsimony), as does adjusted R^2.
- We often use AIC to help us select a model with **stepwise regression**

Using stepwise regression and the `step` function

The default choice is to use an idea called "backwards elimination" with the AIC as the key criterion to help you select a model. - Step-by-step, the machine will consider whether removing each of the variables currently in the model will improve AIC. - Then it will remove the "least useful" predictor, and repeat, until it cannot improve the AIC further.

So we specify a "big model" and then let the stepwise algorithm assess how we can prune it down to a smaller set of variables. . .

Here's how we get started. . .

```
step(m_ks)
```

First Step in Stepwise Output

```
Start:  AIC=1732.7
sqrt(plank) ~ age + gender + reth + incvspov + mealsout +
              bmi_cat + waist + calories + sugar

            Df Sum of Sq    RSS    AIC
- bmi_cat    3     13.37 5460.4 1729.2
- reth       3     13.60 5460.6 1729.2
- mealsout   1      2.55 5449.6 1731.2
- sugar      1      4.99 5452.0 1731.6
<none>                   5447.0 1732.7
- calories   1     14.24 5461.2 1733.3
- gender     1     15.49 5462.5 1733.6
- incvspov   1     40.44 5487.4 1738.2
- waist      1    314.70 5761.7 1787.7
- age        1   1774.22 7221.2 2016.6
```

Next Step in Stepwise Output

```
Step:  AIC=1729.19
sqrt(plank) ~ age + gender + reth + incvspov + mealsout +
              waist + calories + sugar

            Df Sum of Sq    RSS    AIC
- reth       3     12.44 5472.8 1725.5
- mealsout   1      2.22 5462.6 1727.6
- sugar      1      4.29 5464.7 1728.0
<none>                   5460.4 1729.2
- calories   1     13.60 5474.0 1729.7
- gender     1     15.12 5475.5 1730.0
- incvspov   1     40.26 5500.6 1734.6
- waist      1    846.01 6306.4 1873.2
- age        1   2878.78 8339.2 2156.6
```

Step 3 in Stepwise Output

```
Step:  AIC=1725.49
sqrt(plank) ~ age + gender + incvspov + mealsout + waist +
              calories + sugar

            Df Sum of Sq    RSS    AIC
- mealsout   1      1.59 5474.4 1723.8
- sugar      1      4.63 5477.4 1724.3
<none>                   5472.8 1725.5
- calories   1     13.55 5486.4 1726.0
- gender     1     15.29 5488.1 1726.3
- incvspov   1     48.61 5521.4 1732.5
- waist      1    842.86 6315.7 1868.7
- age        1   2892.77 8365.6 2153.8
```

Step 4 in Stepwise Output

```
Step:  AIC=1723.79
sqrt(plank) ~ age + gender + incvspov + waist +
              calories + sugar

            Df Sum of Sq    RSS    AIC
- sugar      1      4.46 5478.9 1722.6
<none>                   5474.4 1723.8
- calories   1     13.43 5487.8 1724.3
- gender     1     15.52 5489.9 1724.7
- incvspov   1     53.61 5528.0 1731.7
- waist      1    841.99 6316.4 1866.9
- age        1   2894.40 8368.8 2152.2
```

Step 5 in Stepwise Output

```
Step:  AIC=1722.61
sqrt(plank) ~ age + gender + incvspov + waist + calories

            Df Sum of Sq    RSS    AIC
- calories   1      9.77 5488.6 1722.4
<none>                   5478.9 1722.6
- gender     1     15.63 5494.5 1723.5
- incvspov   1     56.46 5535.3 1731.0
- waist      1    838.29 6317.1 1865.0
- age        1   2894.12 8373.0 2150.7
```

Step 6 in Stepwise Output

```
Step:  AIC=1722.42
sqrt(plank) ~ age + gender + incvspov + waist

            Df Sum of Sq    RSS    AIC
<none>                   5488.6 1722.4
- gender     1     20.24 5508.9 1724.2
- incvspov   1     59.22 5547.9 1731.3
- waist      1    856.27 6344.9 1867.4
- age        1   3033.61 8522.2 2166.6
```

Final Step in Stepwise Output

```
Call:
lm(formula = sqrt(plank) ~
       age + gender + incvspov + waist, data = nnyfs_train)

Coefficients:
(Intercept)         age  genderMale    incvspov       waist
    6.06270     0.64878     0.28291     0.14873    -0.07971
```

We'll call this model `m_step`

```
m_step <- lm(sqrt(plank) ~ age + gender + incvspov + waist,
             data = nnyfs_train)
```

Tidied Coefficients of m_step

```
tidy(m_step, conf.int = TRUE) %>%
  select(term, estimate, conf.low, conf.high, p.value) %>%
  knitr::kable(digits = 2)
```

term	estimate	conf.low	conf.high	p.value
(Intercept)	6.06	5.35	6.78	0.00
age	0.65	0.59	0.70	0.00
genderMale	0.28	0.00	0.57	0.05
incvspov	0.15	0.06	0.24	0.00
waist	-0.08	-0.09	-0.07	0.00

Is collinearity an issue in m_step?

```
vif(m_step)
```

```
     age   gender incvspov    waist
1.704285 1.002302 1.009277 1.700507
```

glance results for all 3 models

```
a1 <- glance(m_ks) %>% mutate(model = "m_ks")
a2 <- glance(m_2) %>% mutate(model = "m_2")
a3 <- glance(m_step) %>% mutate(model = "m_step")

bind_rows(a1, a2, a3) %>%
  select(model, df, r.squared, "Adj. R^2^" = adj.r.squared,
         sigma, p.value, AIC, BIC) %>%
  knitr::kable(digits = 3)
```

model	df	r.squared	Adj. R^2	sigma	p.value	AIC	BIC
m_ks	14	0.379	0.371	2.334	0	4612.306	4686.131
m_2	5	0.330	0.328	2.413	0	4670.642	4700.172
m_step	5	0.374	0.372	2.332	0	4602.027	4631.557

ANOVA comparison of m_step and m_ks

Does stepping down from m_ks to m_step have a significant impact on predictive quality of the model?

```
anova(m_ks, m_step)
```

```
Analysis of Variance Table

Model 1: sqrt(plank) ~ age + gender + reth + incvspov + mealsc
    waist + calories + sugar
Model 2: sqrt(plank) ~ age + gender + incvspov + waist
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1   1000 5447.0
2   1009 5488.6 -9   -41.631 0.8492 0.5707
```

Setting up Out of Sample Validation

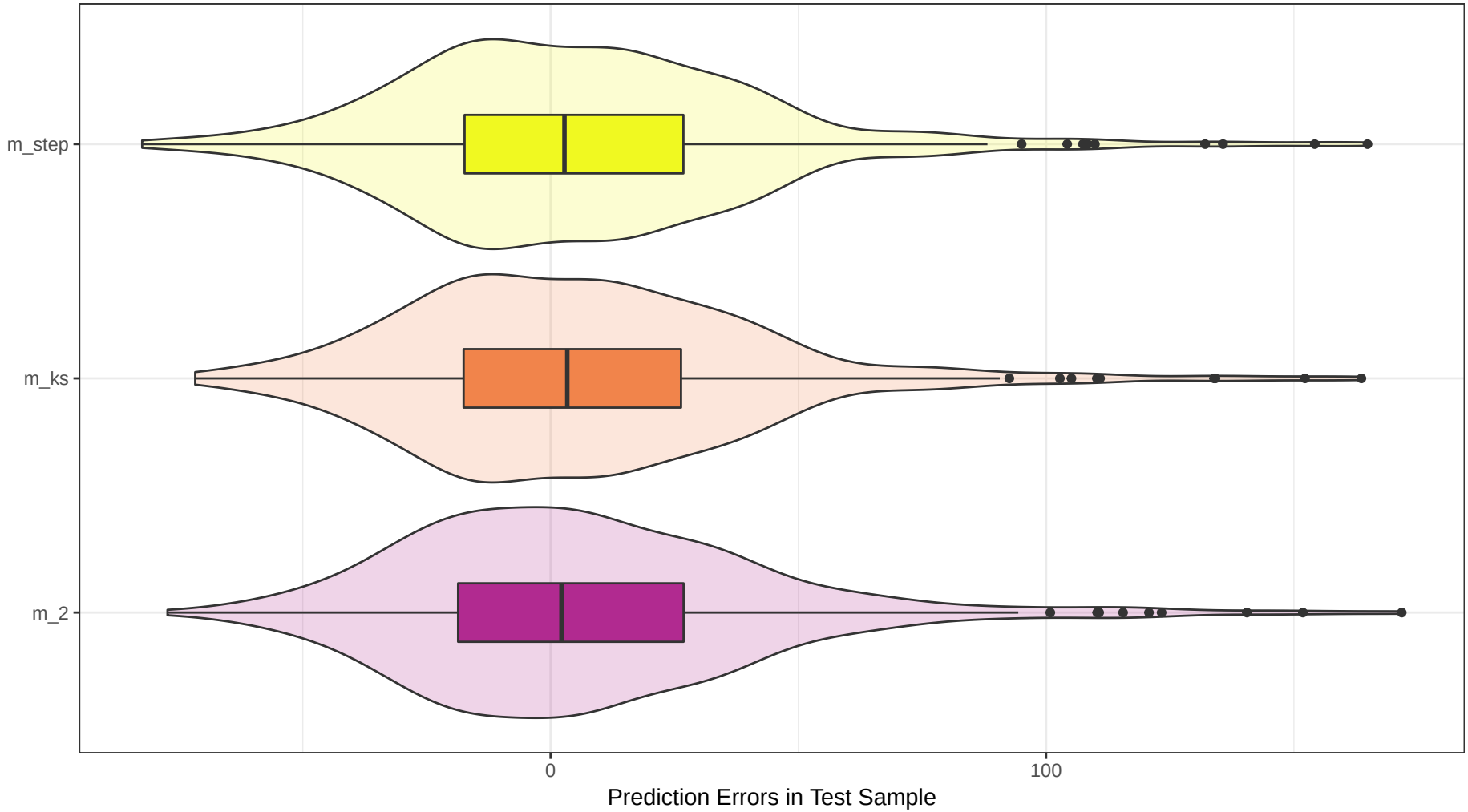
```
test_m_ks <- augment(m_ks, newdata = nnyfs_test) %>%
  mutate(modname = "m_ks", .fitsqr = .fitted^2,
         .resid = plank - .fitsqr) %>%
  select(seqn, modname, plank, .fitsqr, .resid, .fitted)
test_m_step <- augment(m_step, newdata = nnyfs_test) %>%
  mutate(modname = "m_step", .fitsqr = .fitted^2,
         .resid = plank - .fitsqr) %>%
  select(seqn, modname, plank, .fitsqr, .resid, .fitted)
test_m_2 <- augment(m_2, newdata = nnyfs_test) %>%
  mutate(modname = "m_2", .fitsqr = .fitted^2,
         .resid = plank - .fitsqr) %>%
  select(seqn, modname, plank, .fitsqr, .resid, .fitted)

temp <- union(test_m_ks, test_m_step)
test_comp <- union(temp, test_m_2) %>%
  arrange(seqn, modname)
```

test_comp result

```
test_comp %>% head() %>% knitr::kable(digits = 2)
```

seqn	modname	plank	.fitsqr	.resid	.fitted
71922	m_2	107	47.70	59.30	6.91
71922	m_ks	107	58.74	48.26	7.66
71922	m_step	107	59.30	47.70	7.70
71923	m_2	127	66.06	60.94	8.13
71923	m_ks	127	84.26	42.74	9.18
71923	m_step	127	83.02	43.98	9.11

Three Models: Distribution of Errors (Test Sample)



Three Models: Test Sample Error Summaries

```
test_comp %>%
  group_by(modname) %>%
  summarize(n = n(),
            MAPE = mean(abs(.resid)),
            MSPE = mean(.resid^2),
            max_error = max(abs(.resid))) %>%
  knitr::kable(digits = 2)
```

modname	n	MAPE	MSPE	max_error
m_2	338	27.42	1378.02	171.76
m_ks	338	27.39	1331.79	163.62
m_step	338	27.64	1360.74	164.85

Training Sample and Test Sample Results

- In the Training Sample, we had . . .

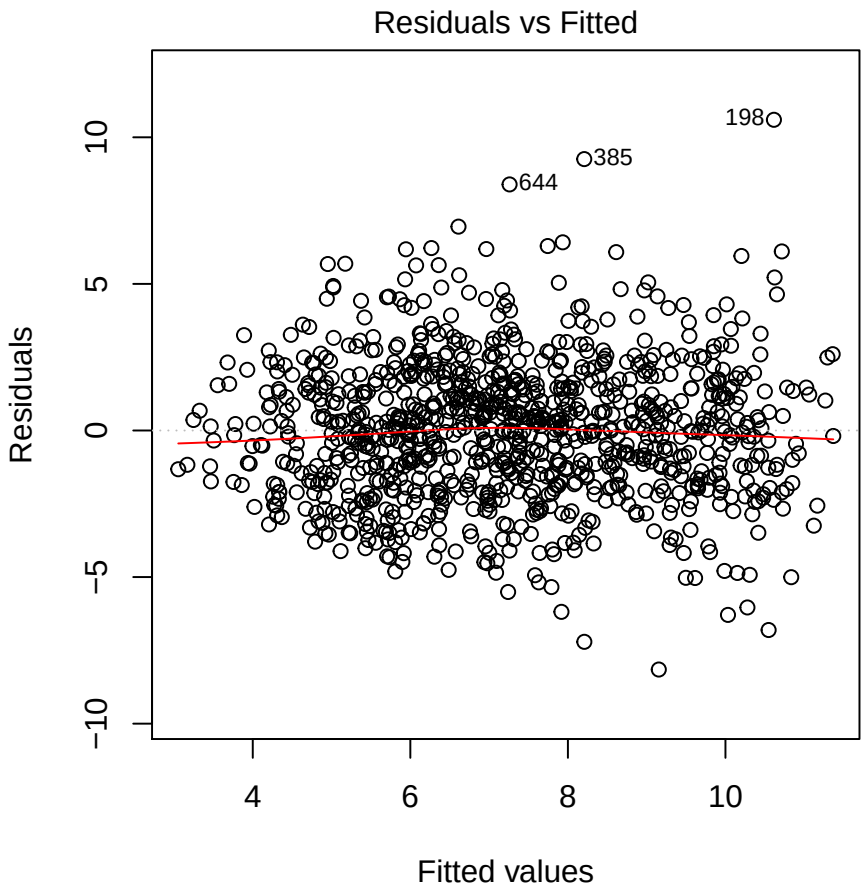
model	df	r.squared	Adj. R^2	sigma	p.value	AIC	BIC
m_2	5	0.330	0.328	2.413	0	4670.642	4700.172
m_ks	14	0.379	0.371	2.334	0	4612.306	4686.131
m_step	5	0.374	0.372	2.332	0	4602.027	4631.557

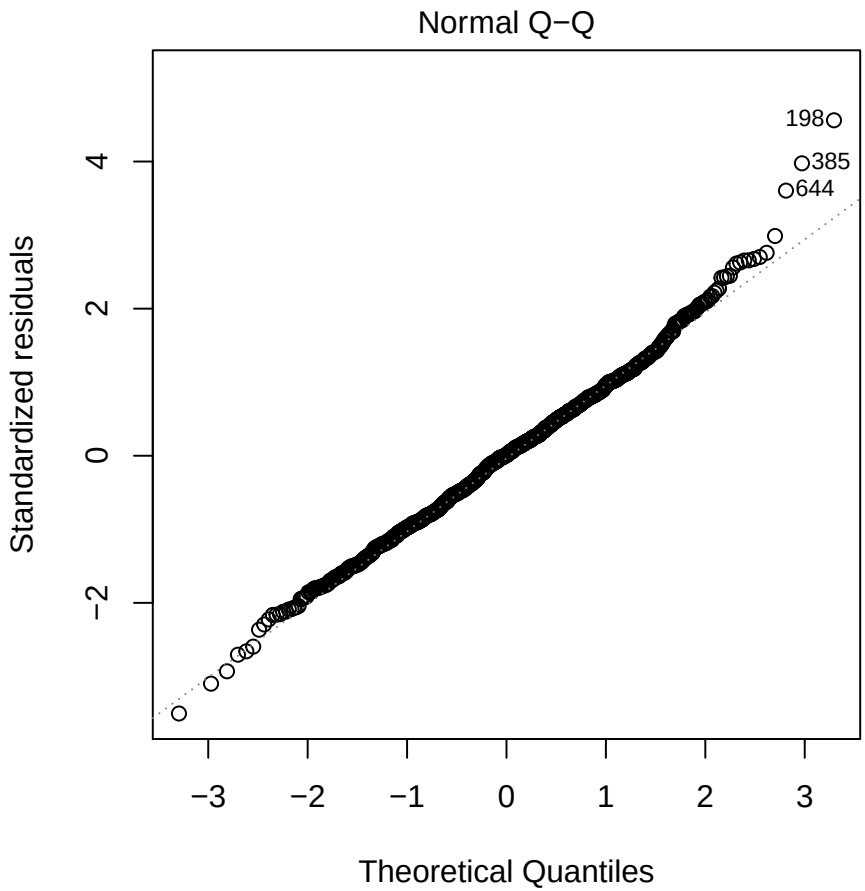
- In the Test Sample, we had. . .

modname	n	MAPE	MSPE	max_error
m_2	338	27.42	1378.02	171.76
m_ks	338	27.39	1331.79	163.62
m_step	338	27.64	1360.74	164.85

Residual Plots for m_step? (training sample)

```
par(mfrow = c(1,2))
plot(m_step, which = 1:2)
```





Checking `m_step` Residual Plots (training sample)

```
par(mfrow = c(1,2))
plot(m_step, which = c(3, 5))
```