Regression (Linear + Multiple)

IMT 573A - Data Science 1 - Learning from Data

12-Nov-2020 (Week 7, Day 12)

Our Zoom class sessions will be recorded.

Today’s topic

- Making predictions
- Extrapolating predictions (Limits of predictions) + Lab part 1

- Working with model objects (broom) + lab part 2

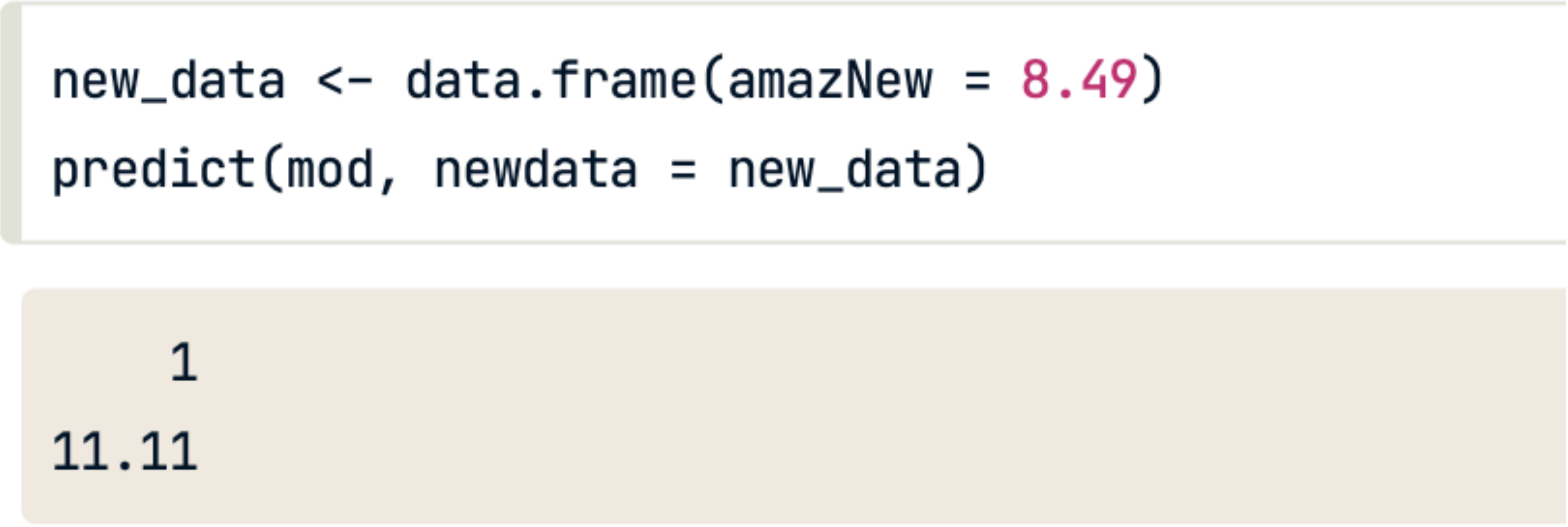
- Visualizing model fit & Extracting outliers + lab part 3

- Multiple Linear regression + lab part 5

Making predictions (last class & lab)

Using our model to make predictions about new observations. **POWERFUL TECHNIQUE in ML.**

predict(lm)



```
new_data <- data.frame(amazNew = 8.49)
predict(mod, newdata = new_data)
```

```
    1
11.11
```

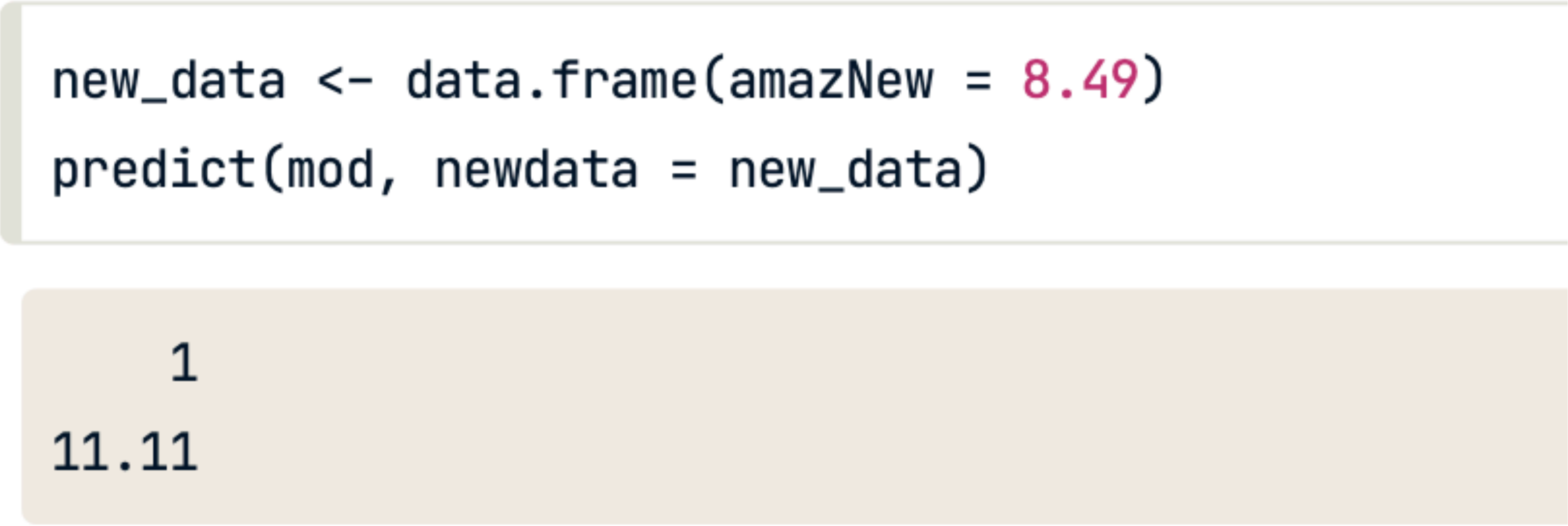
predict(lm, newdata)

fitted values for any new data

Making predictions & **Showing Predictions**

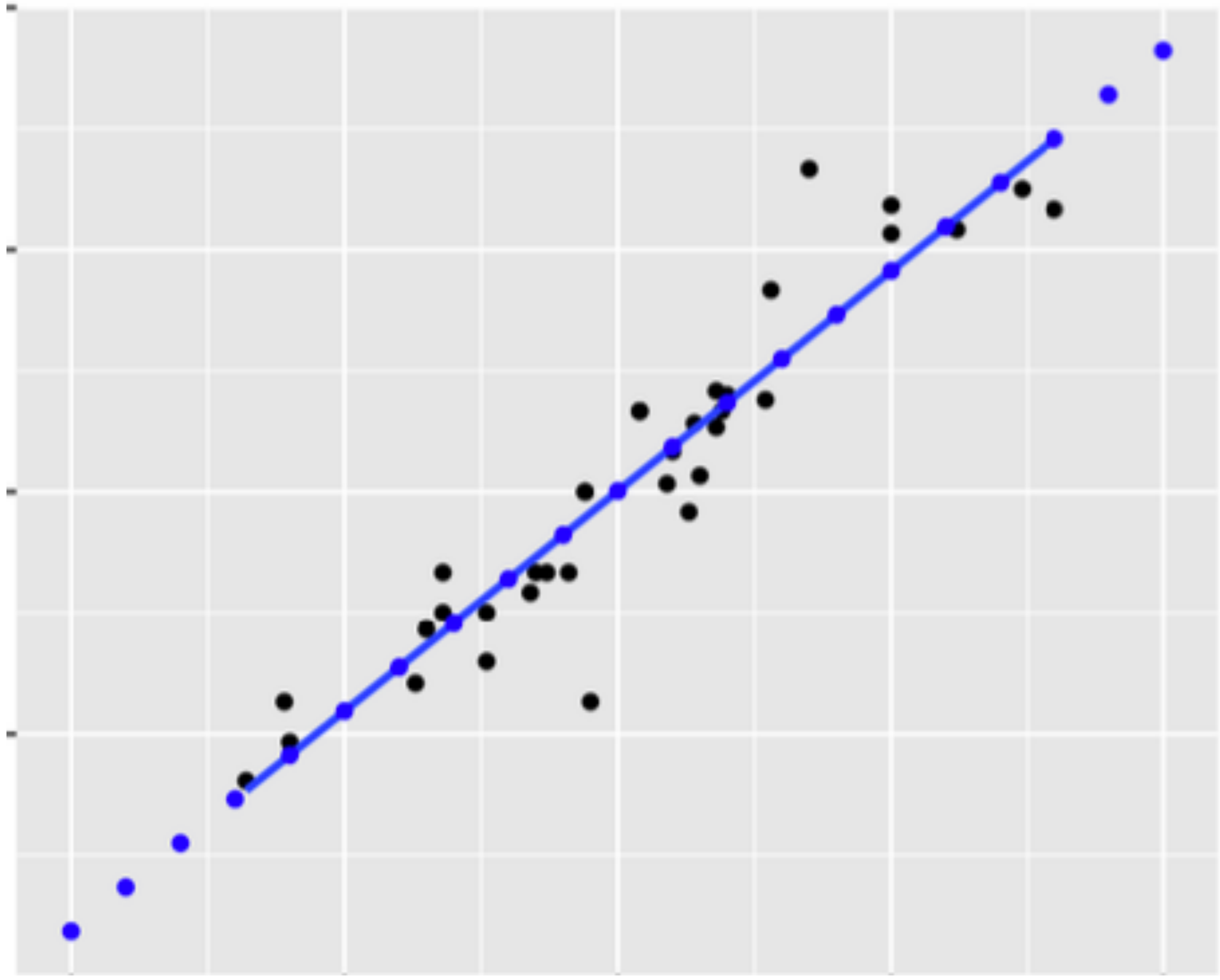
Using our model to make predictions about new observations. **POWERFUL TECHNIQUE in ML.**

predict(lm)



```
new_data <- data.frame(amazNew = 8.49)
predict(mod, newdata = new_data)
```

```
    1
11.11
```



blue dots are the predicted data

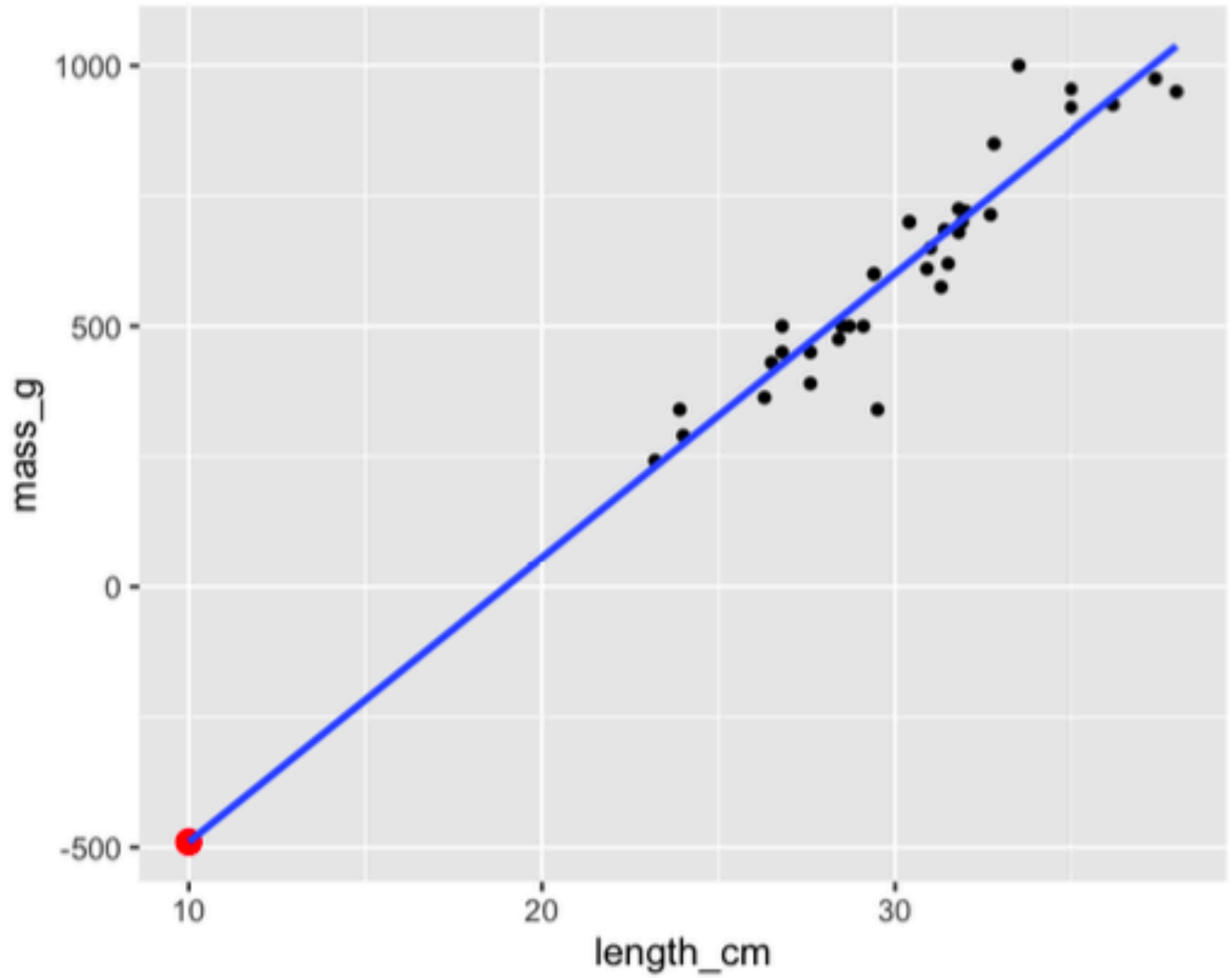
Predictions lie exactly on the trend line

Extraploting (Limits of prediction)

Extrapolating means making predictions outside the range of observed data

Extrapolating is sometimes appropriate, but can lead to misleading or ridiculous results.

You need to understand the context of your data in order to determine whether it is sensible to extrapolate



Length of a fish used to predict mass of a fish.

Extrapolating to length 10cm gives non-sensical results.

Lab - part 1

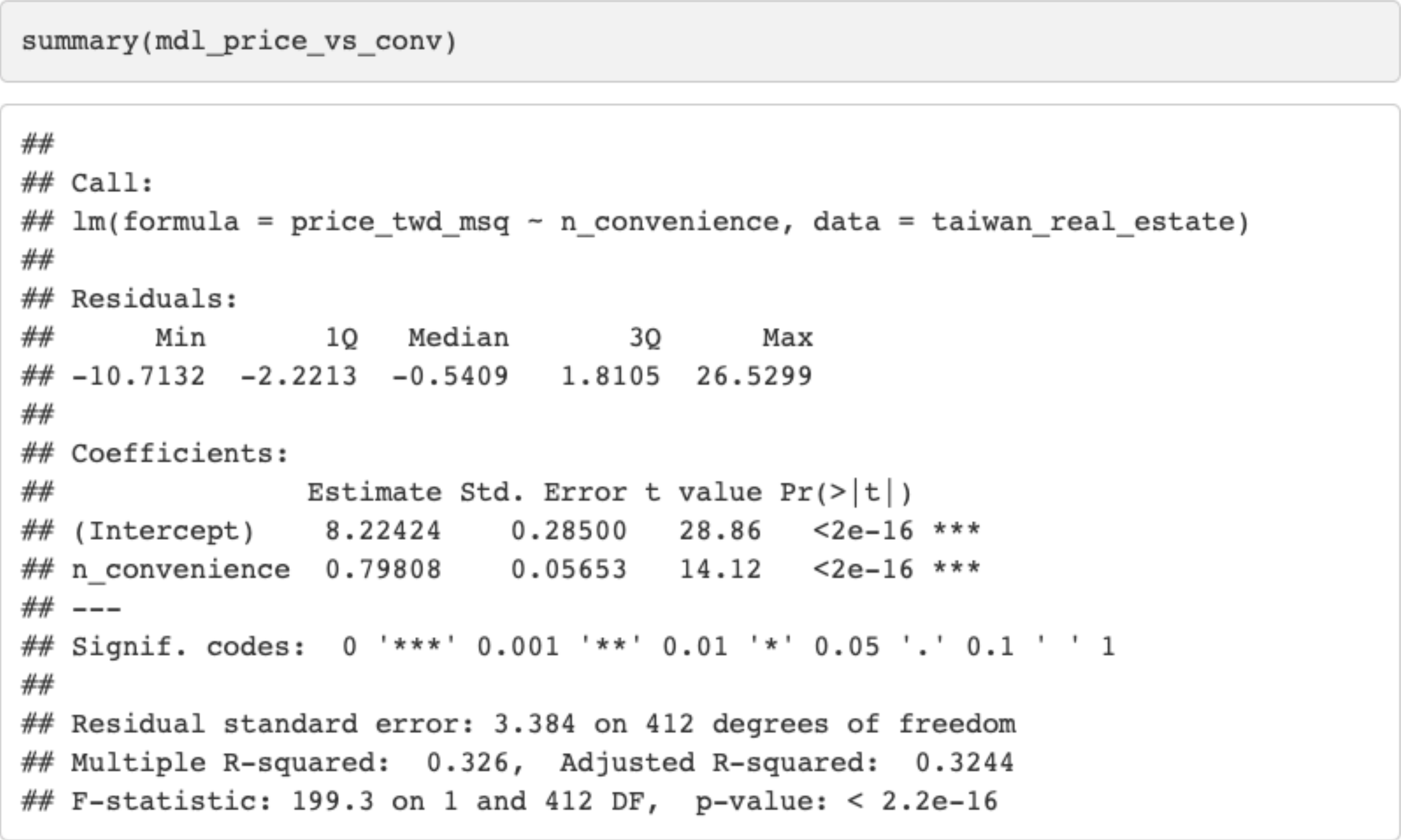
Working with model objects

broom package

While summary shows a lot of information, it is designed to be read, and not to be manipulated.

You need either vector or a data frame to manipulate data.

broom package comes in handy. Returns data frame. This allows you to manipulate data with dplyr, ggplot and other tidyverse packages

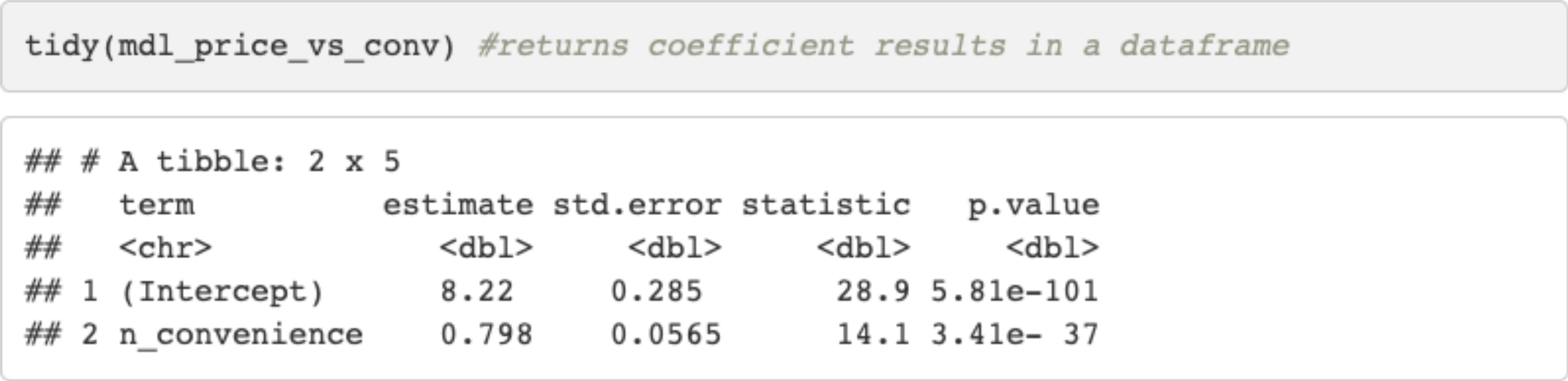


```
summary(mdl_price_vs_conv)
```

```
##
## Call:
## lm(formula = price_twd_msq ~ n_convenience, data = taiwan_real_estate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7132  -2.2213  -0.5409   1.8105  26.5299
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.22424    0.28500   28.86   <2e-16 ***
## n_convenience  0.79808    0.05653   14.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.384 on 412 degrees of freedom
## Multiple R-squared:  0.326,  Adjusted R-squared:  0.3244
## F-statistic: 199.3 on 1 and 412 DF,  p-value: < 2.2e-16
```

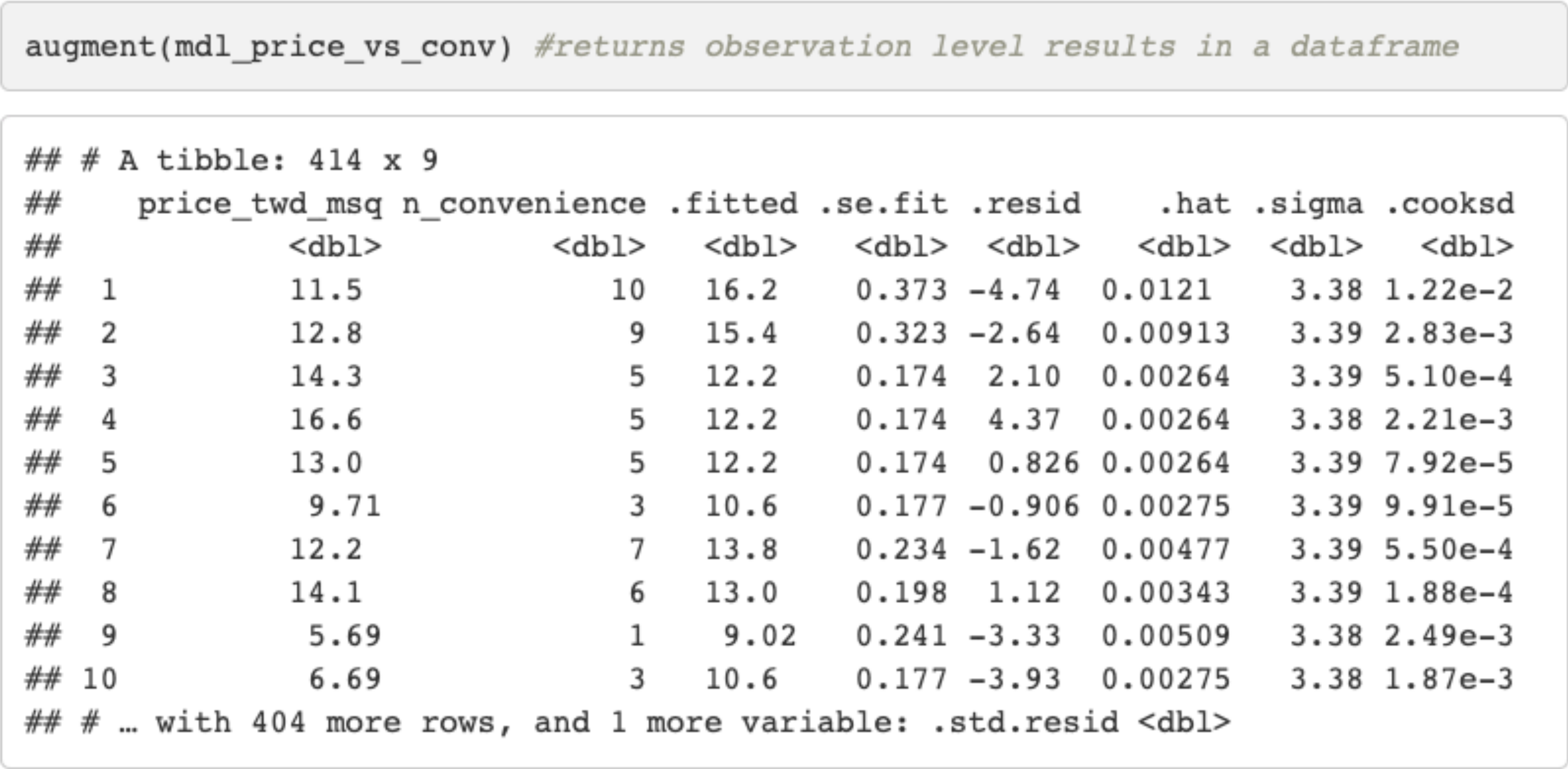
Working with model objects

broom package: tidy, augment, glance



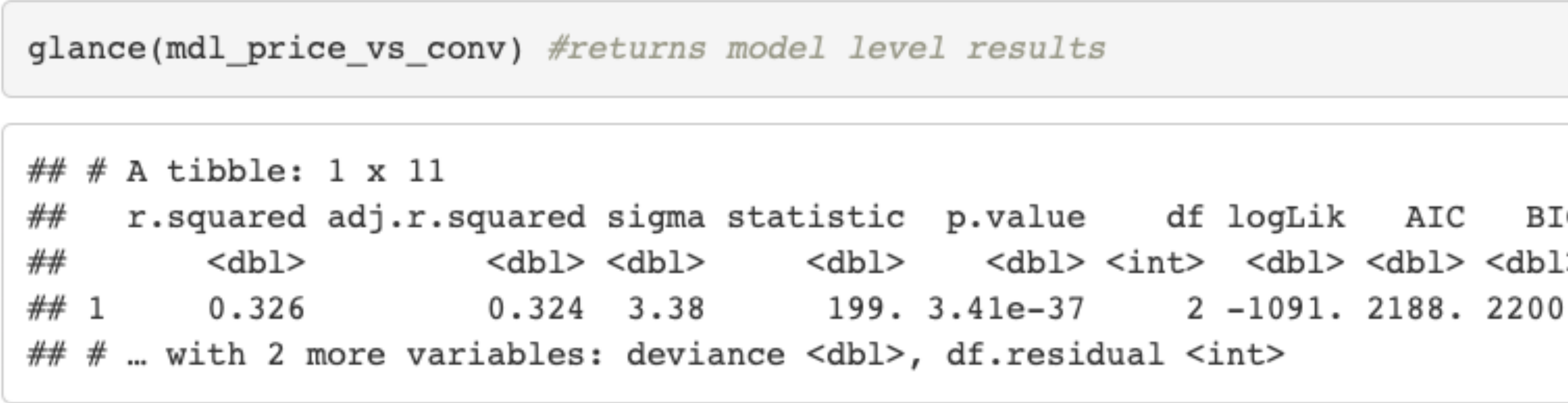
```
tidy(mdl_price_vs_conv) #returns coefficient results in a dataframe
```

```
## # A tibble: 2 x 5
##   term          estimate std.error statistic   p.value
##   <chr>            <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)      8.22     0.285       28.9 5.81e-101
## 2 n_convenience    0.798    0.0565      14.1 3.41e- 37
```



```
augment(mdl_price_vs_conv) #returns observation level results in a dataframe
```

```
## # A tibble: 414 x 9
##    price_twd_msq n_convenience .fitted .se.fit .resid    .hat .sigma .cooksd
##            <dbl>         <dbl>   <dbl>   <dbl>  <dbl>   <dbl>  <dbl>   <dbl>
##  1         11.5             10   16.2    0.373 -4.74  0.0121    3.38 1.22e-2
##  2         12.8              9   15.4    0.323 -2.64  0.00913   3.39 2.83e-3
##  3         14.3              5   12.2    0.174  2.10  0.00264   3.39 5.10e-4
##  4         16.6              5   12.2    0.174  4.37  0.00264   3.38 2.21e-3
##  5         13.0              5   12.2    0.174  0.826 0.00264   3.39 7.92e-5
##  6          9.71             3   10.6    0.177 -0.906 0.00275   3.39 9.91e-5
##  7         12.2              7   13.8    0.234 -1.62  0.00477   3.39 5.50e-4
##  8         14.1              6   13.0    0.198  1.12  0.00343   3.39 1.88e-4
##  9          5.69             1    9.02   0.241 -3.33  0.00509   3.38 2.49e-3
## 10          6.69             3   10.6    0.177 -3.93  0.00275   3.38 1.87e-3
## # … with 404 more rows, and 1 more variable: .std.resid <dbl>
```



```
glance(mdl_price_vs_conv) #returns model level results
```

```
## # A tibble: 1 x 11
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BI
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <int>  <dbl> <dbl> <dbl
## 1     0.326         0.324  3.38      199. 3.41e-37     2 -1091. 2188. 2200
## # … with 2 more variables: deviance <dbl>, df.residual <int>
```

sigma in glance output is RSE (residual standard error)
RSE = difference b/w a prediction and an observed response, i.e., how much the predictions are typically wrong by.

It has the same unit as the response variable.

Lab - part 2

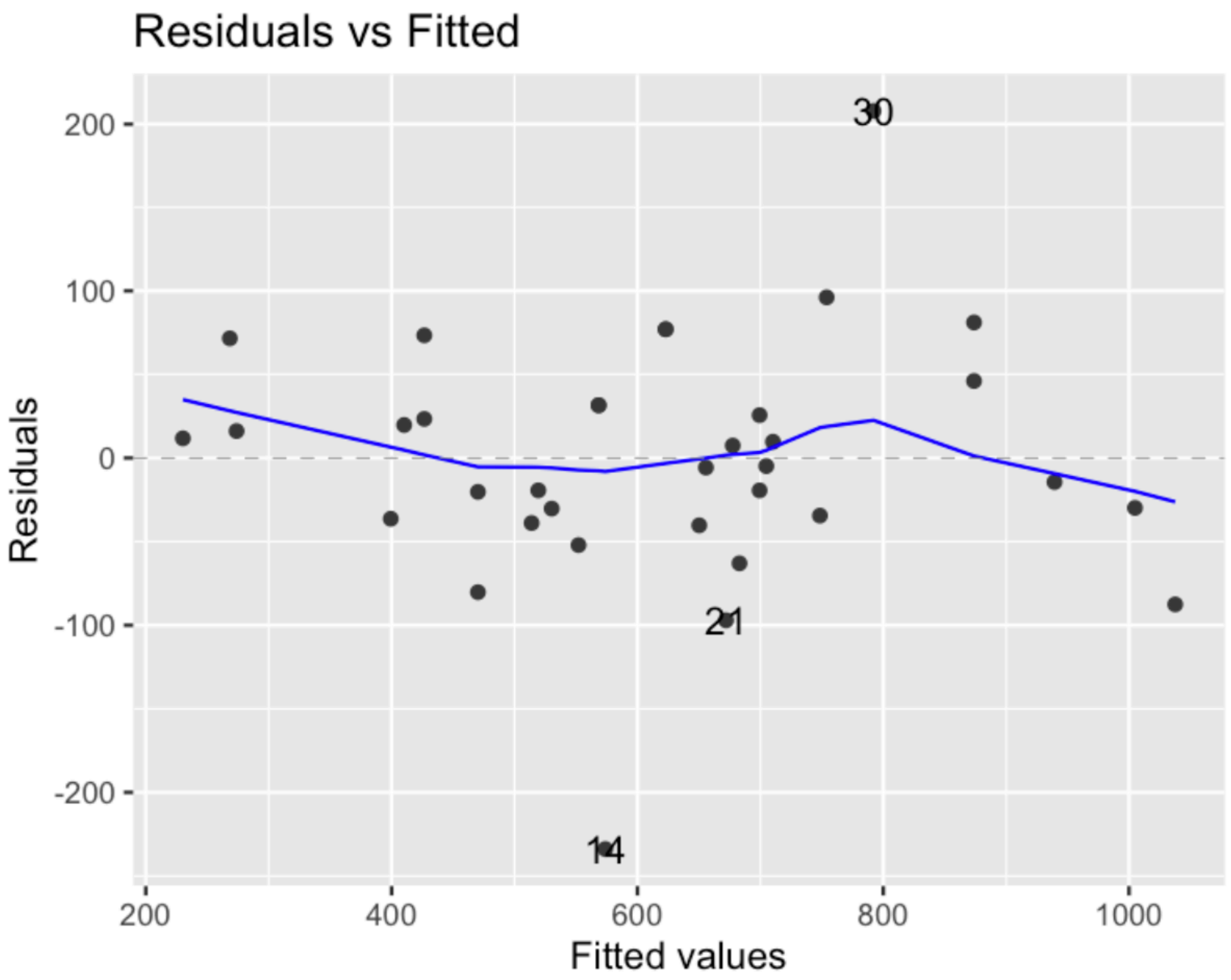
broom package

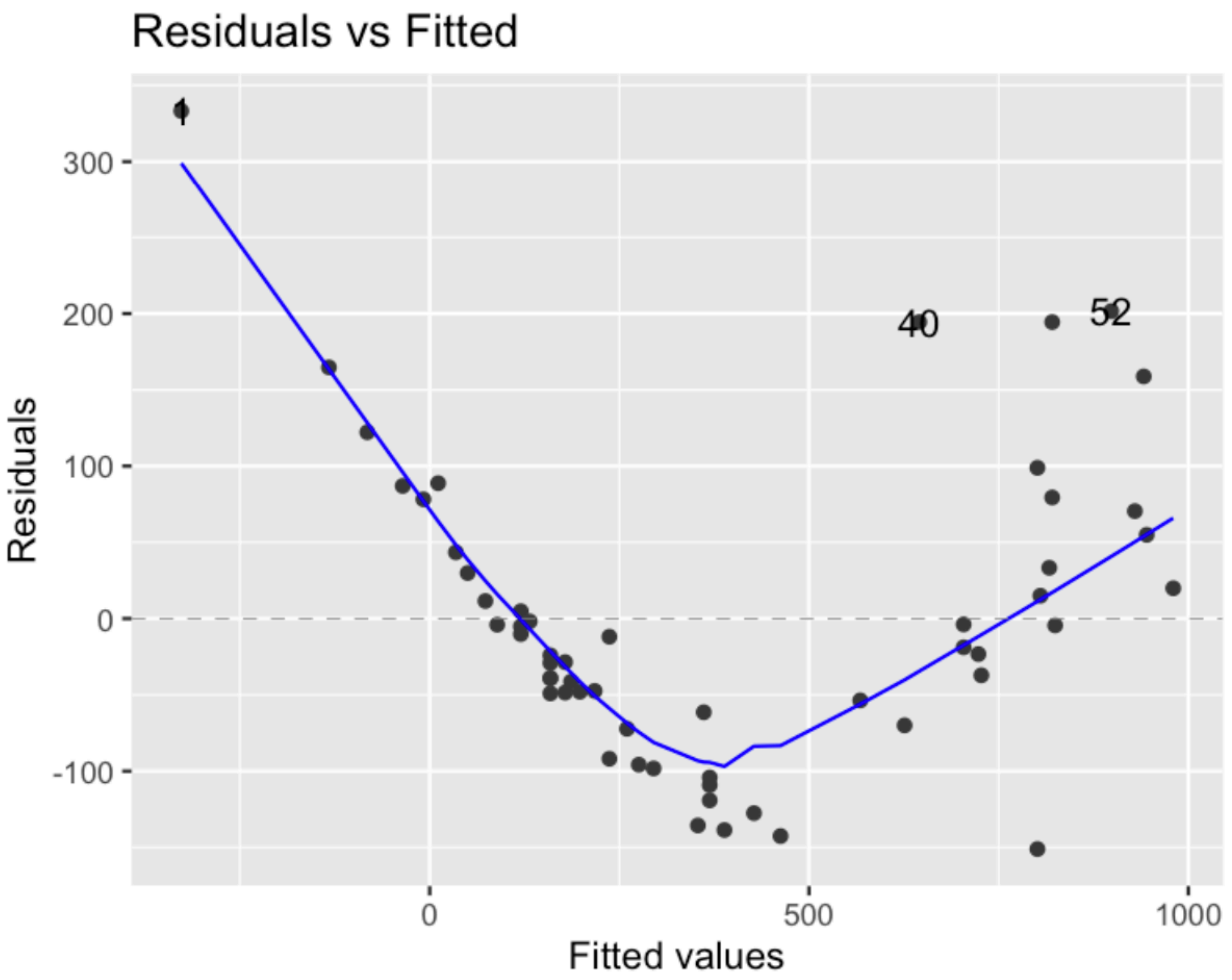
Visualizing model fit

Residual vs. Fitted model

If the residuals met the assumption that they are normality distributed with mean zero, then the trend line should closely follow the y = 0 line on the residuals vs. fitted values plot.

True for left plot, not right plot.

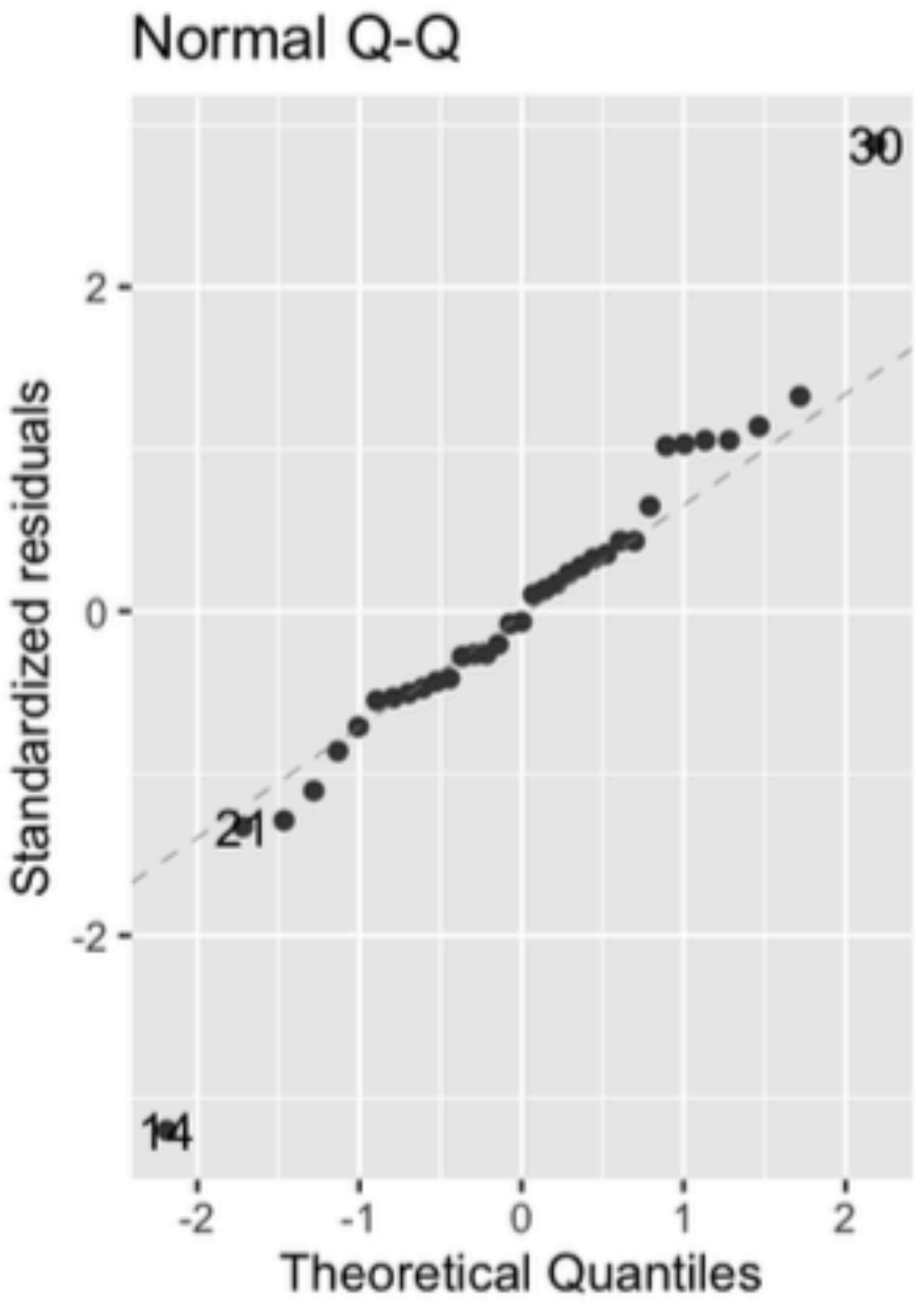


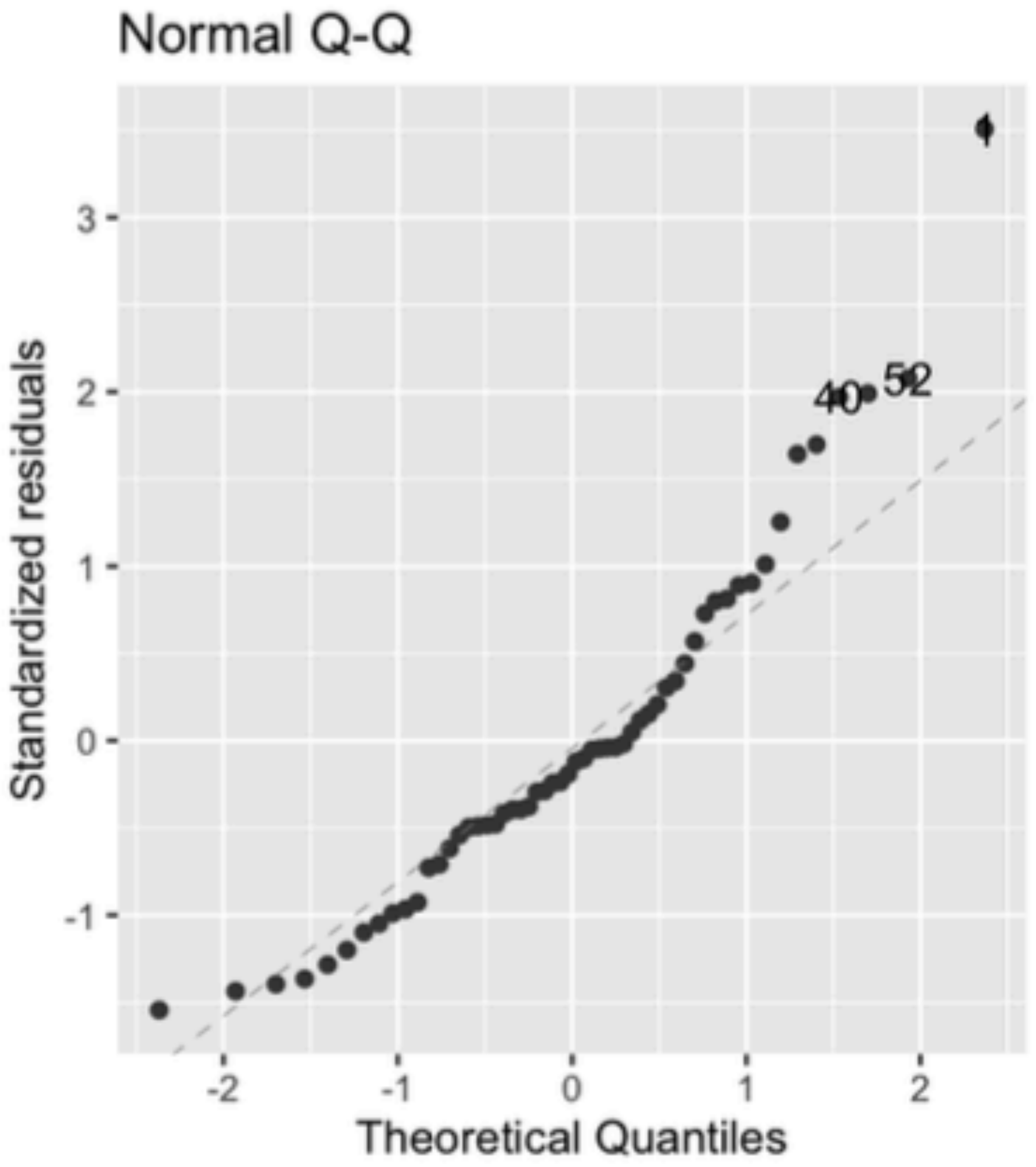


Visualizing model fit

Q-Q plot: shows whether or not the residuals follow a normal distribution.

If the points follow the straight line then it follows a normal distribution



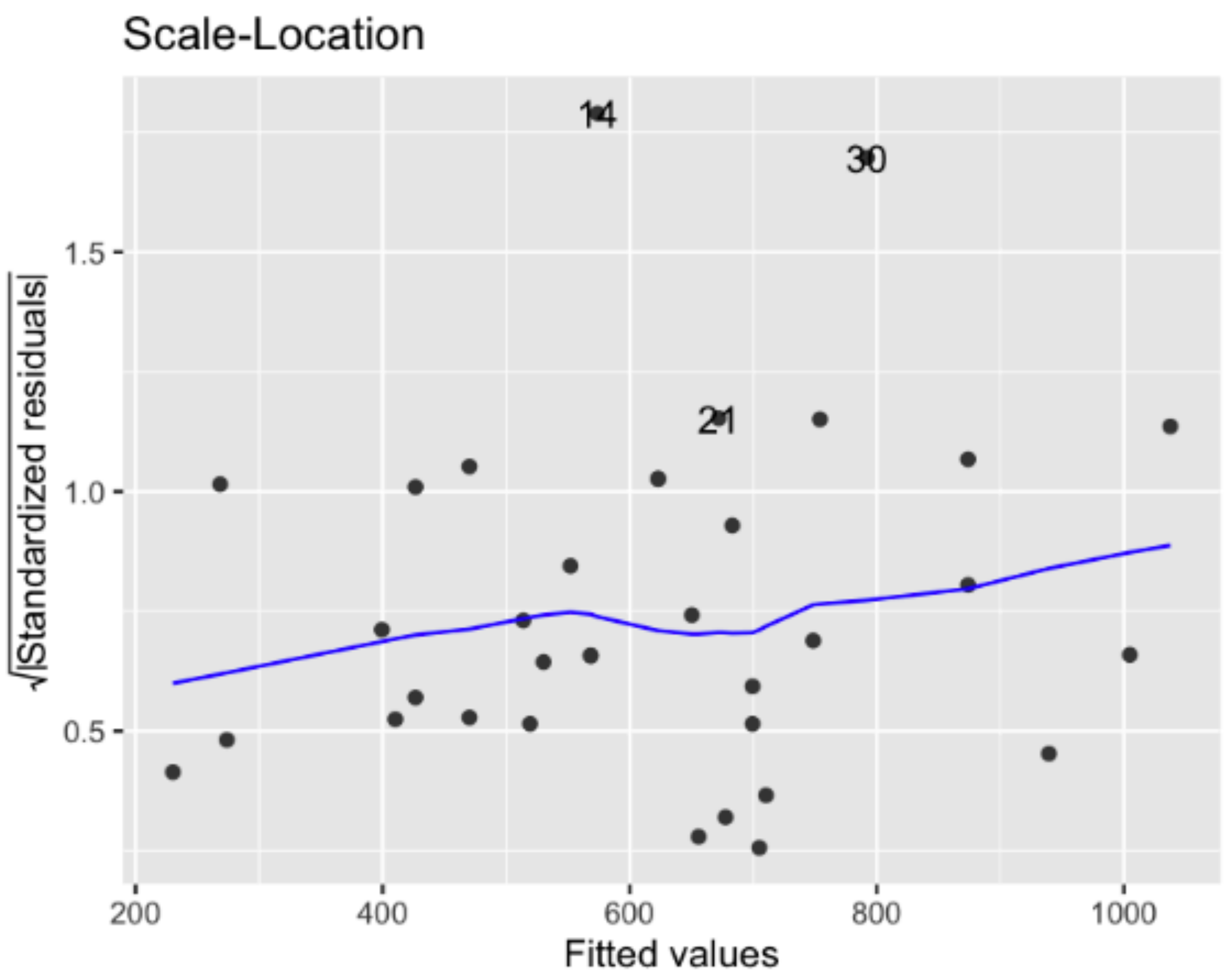


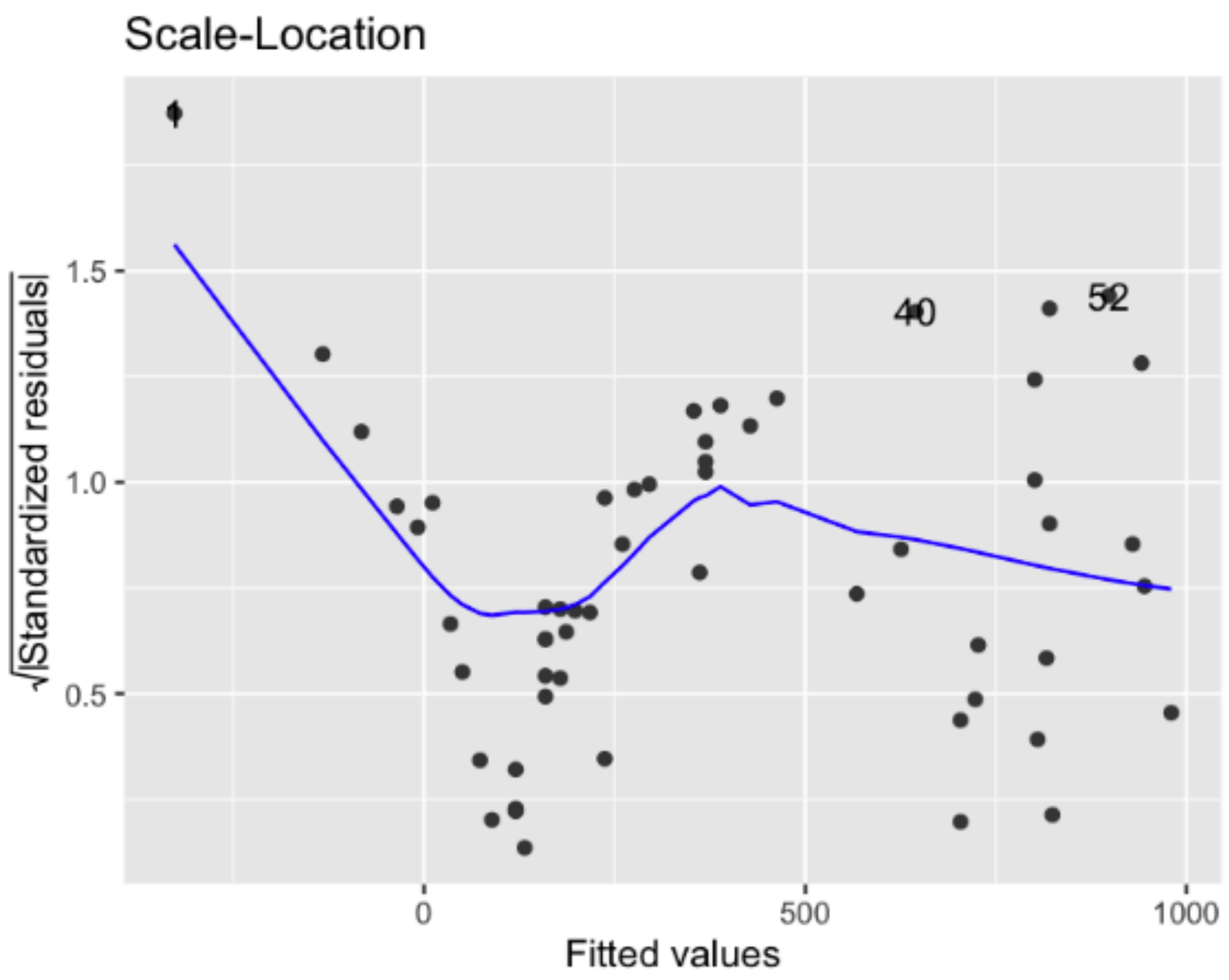
Visualizing model fit

Scaled-location: plot shows the square root of the standardized residuals vs. the fitted values.

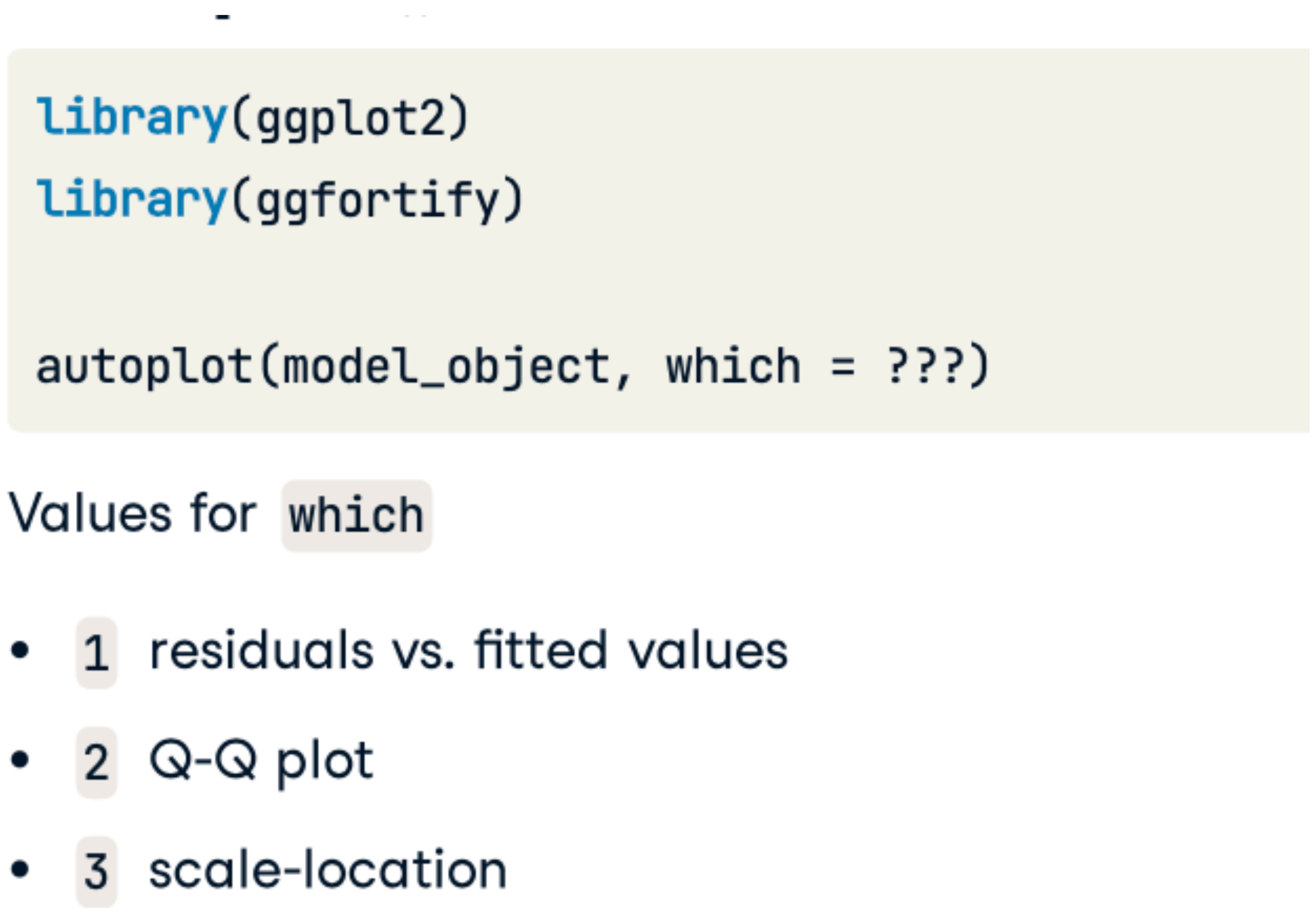
This plot shows whether the size of the residuals gets bigger or smaller.

Left side trend (not a huge change), Right side: trend line goes up and down all over the place, indicating poor fit.





Visualizing model fit



```r
library(ggplot2)
library(ggfortify)

autoplot(model_object, which = ???)
```

Values for `which`

- `1` residuals vs. fitted values
- `2` Q-Q plot
- `3` scale-location

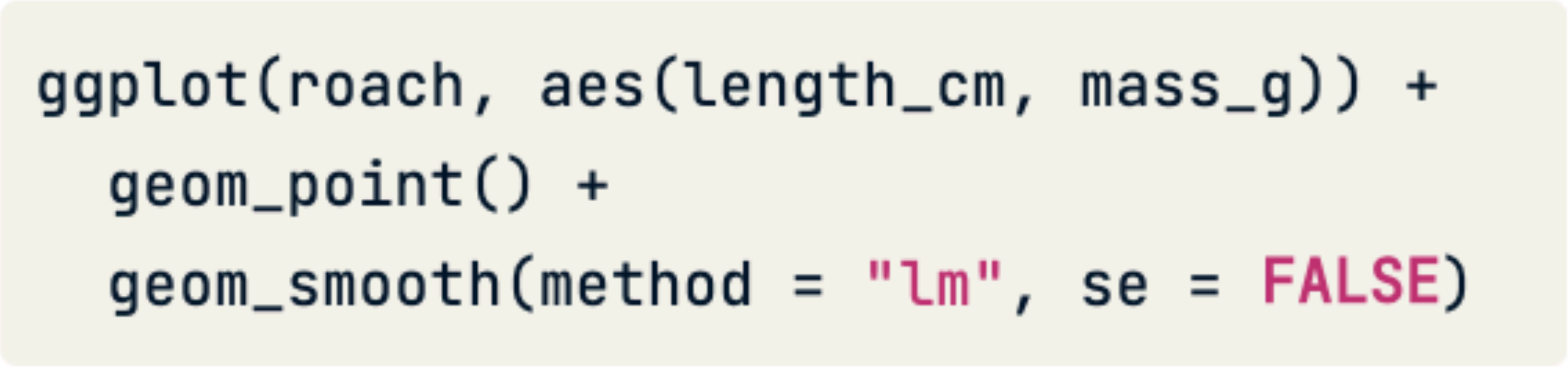
These three diagnostic plots are excellent for sanity-checking the quality of your models.

Outliers, leverages, and influence

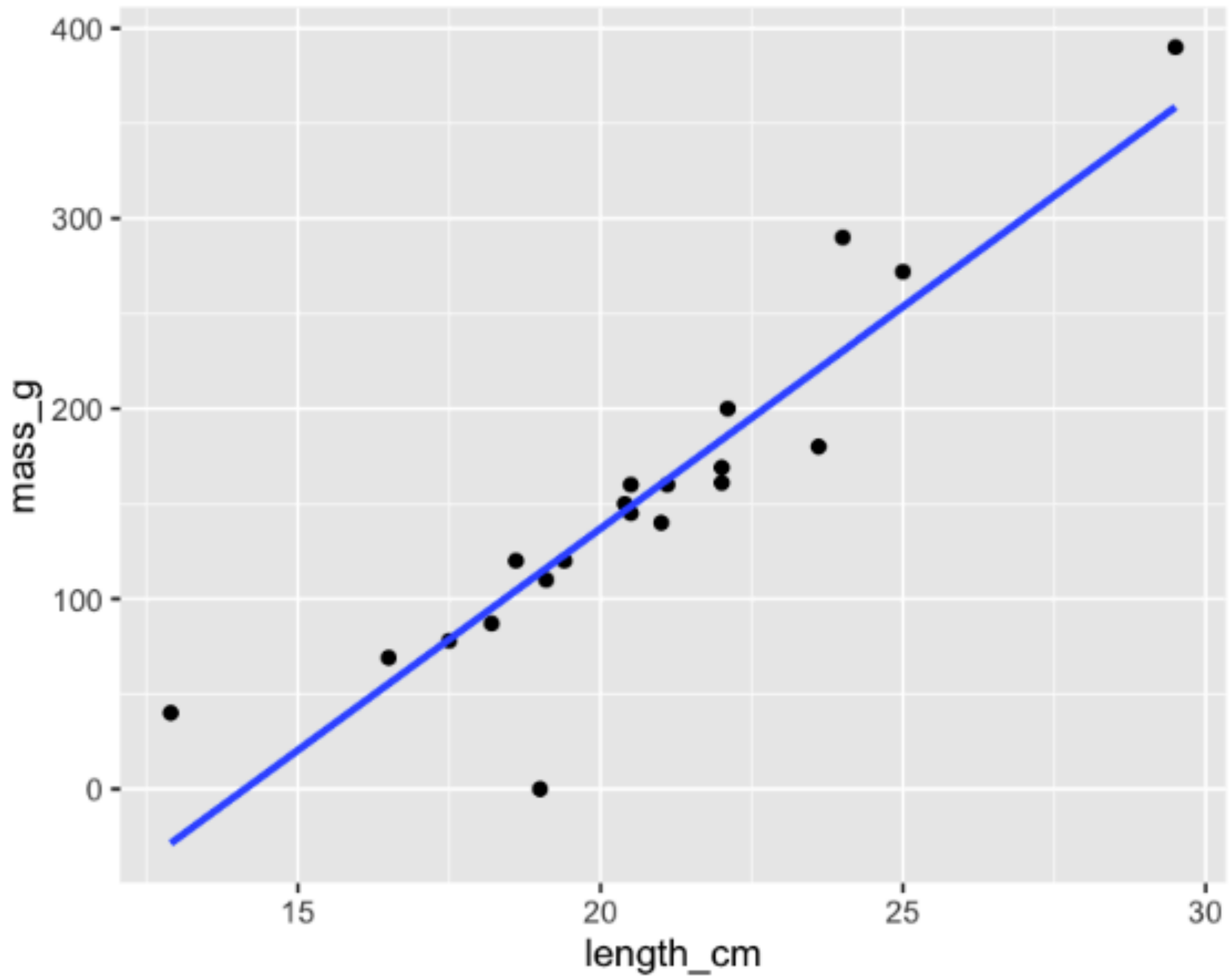
Spotting unusual values in the dataset

Plotting mass versus length of a dataset (fish data)

species	length_cm	mass_g



```
ggplot(roach, aes(length_cm, mass_g)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```



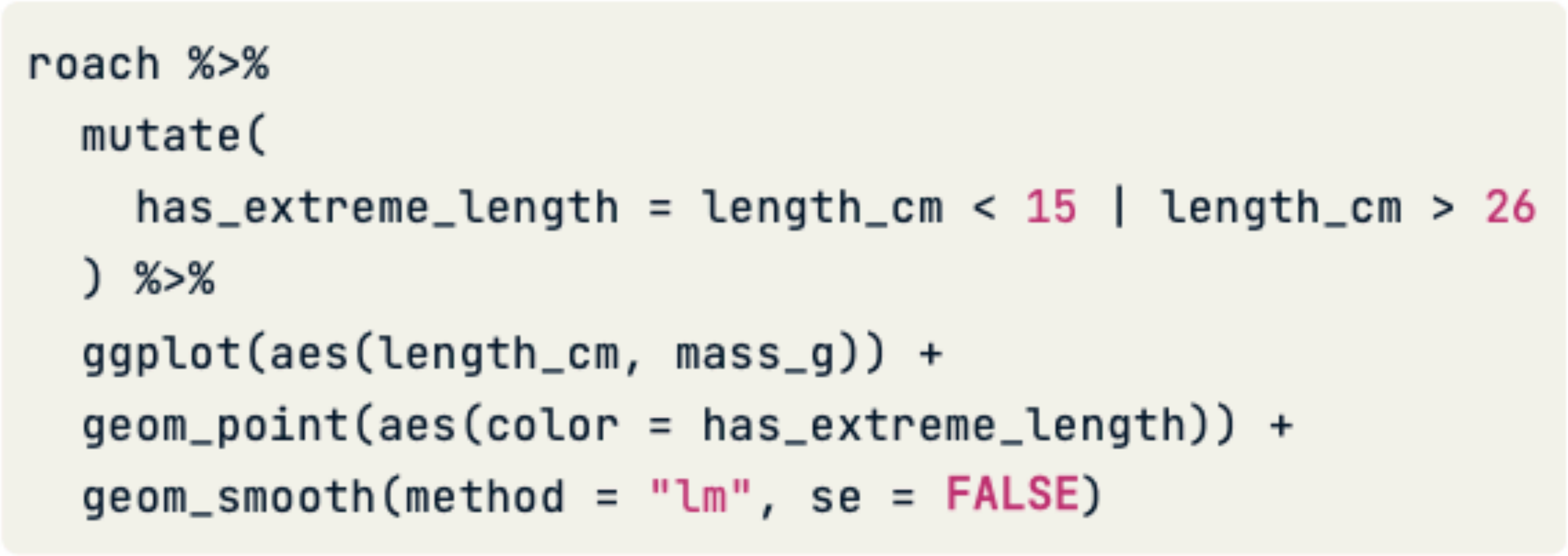
Which of these points constitute an outlier?

Explanatory variable: length, Response variable: mass

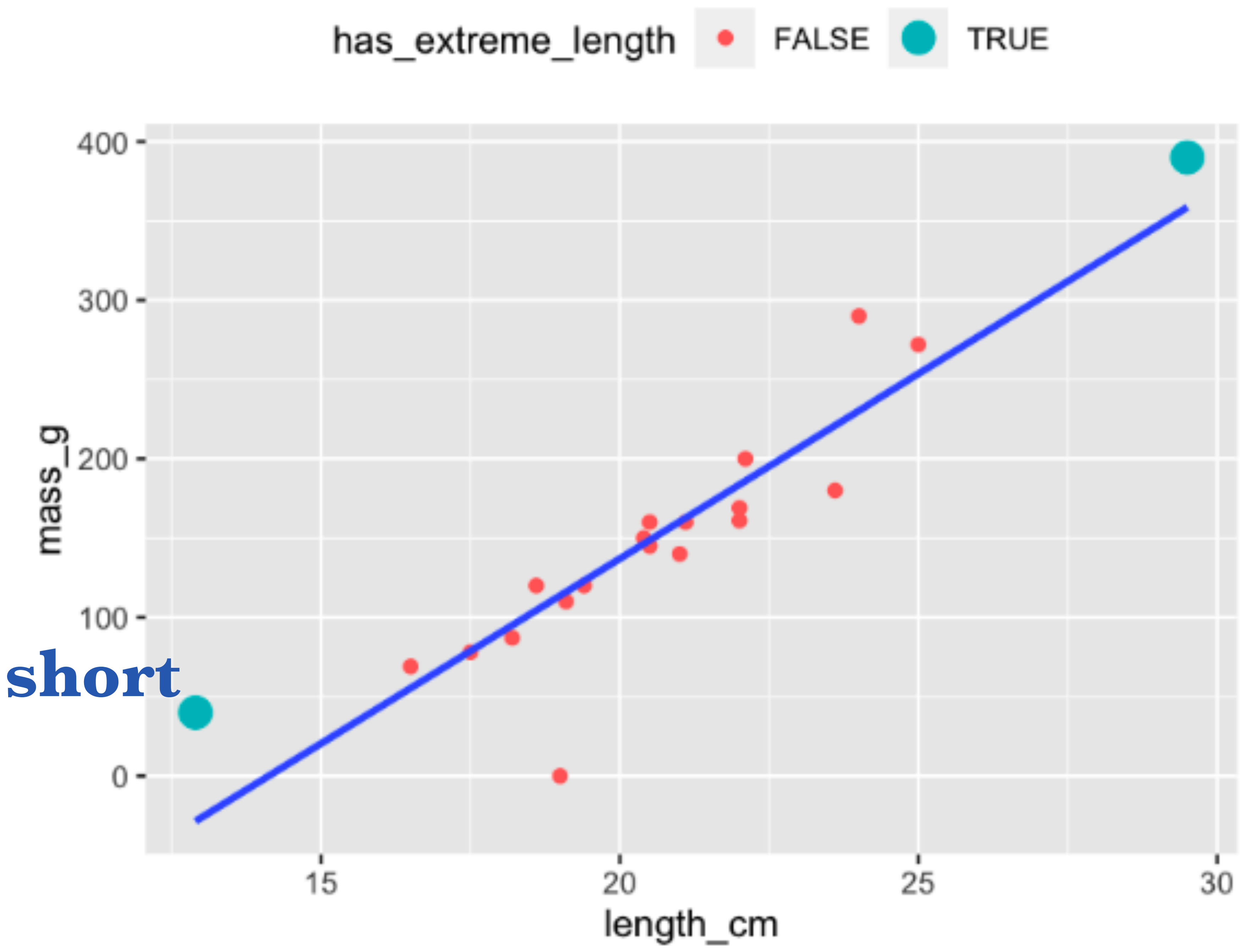
Leverage

First type of outlier is when you have explanatory variables that are extreme

Leverage measures how unusual or extreme the explanatory variables are for each observation.



```
roach %>%
  mutate(
    has_extreme_length = length_cm < 15 | length_cm > 26
  ) %>%
  ggplot(aes(length_cm, mass_g)) +
  geom_point(aes(color = has_extreme_length)) +
  geom_smooth(method = "lm", se = FALSE)
```



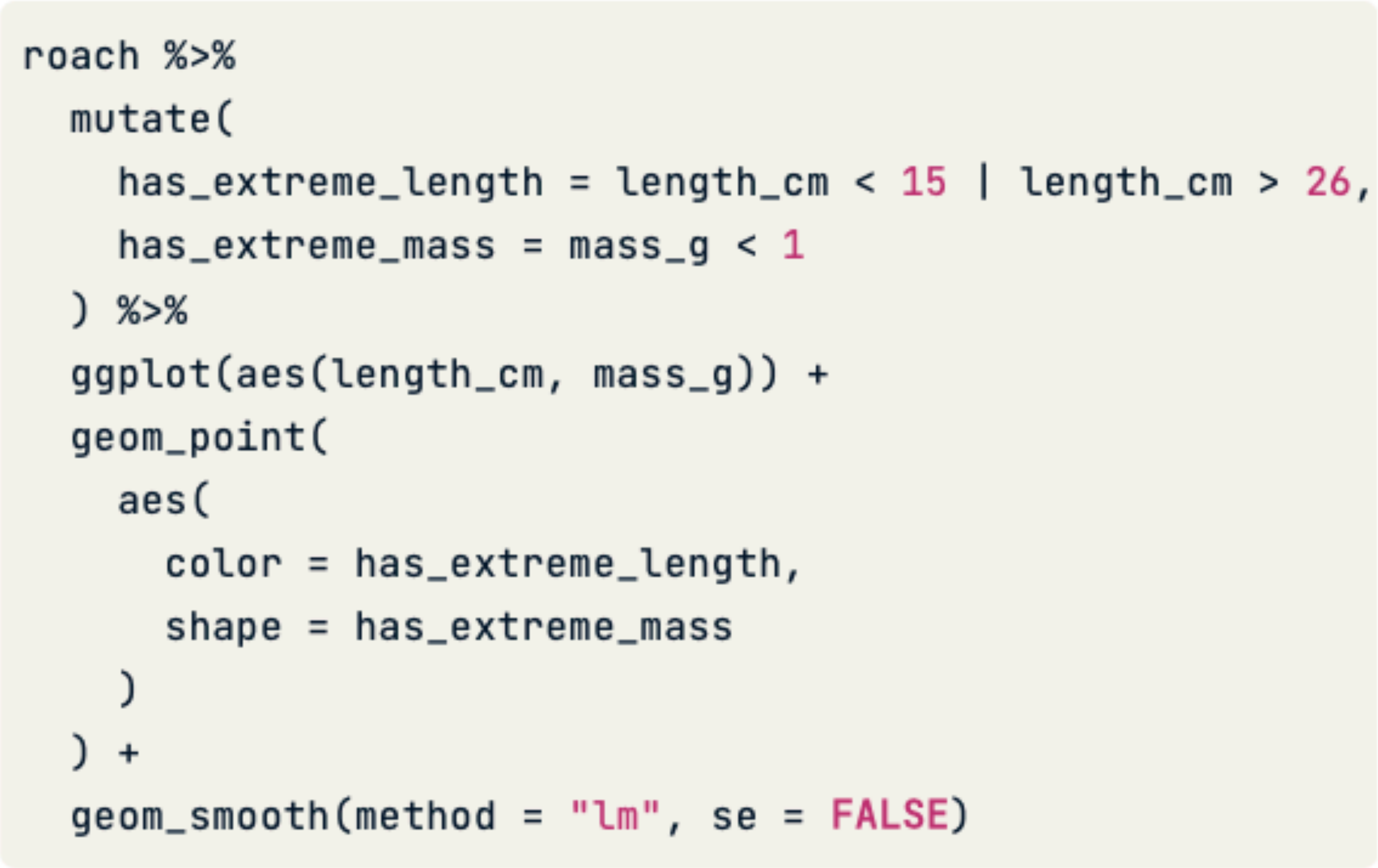
Extremely long

Extremely short

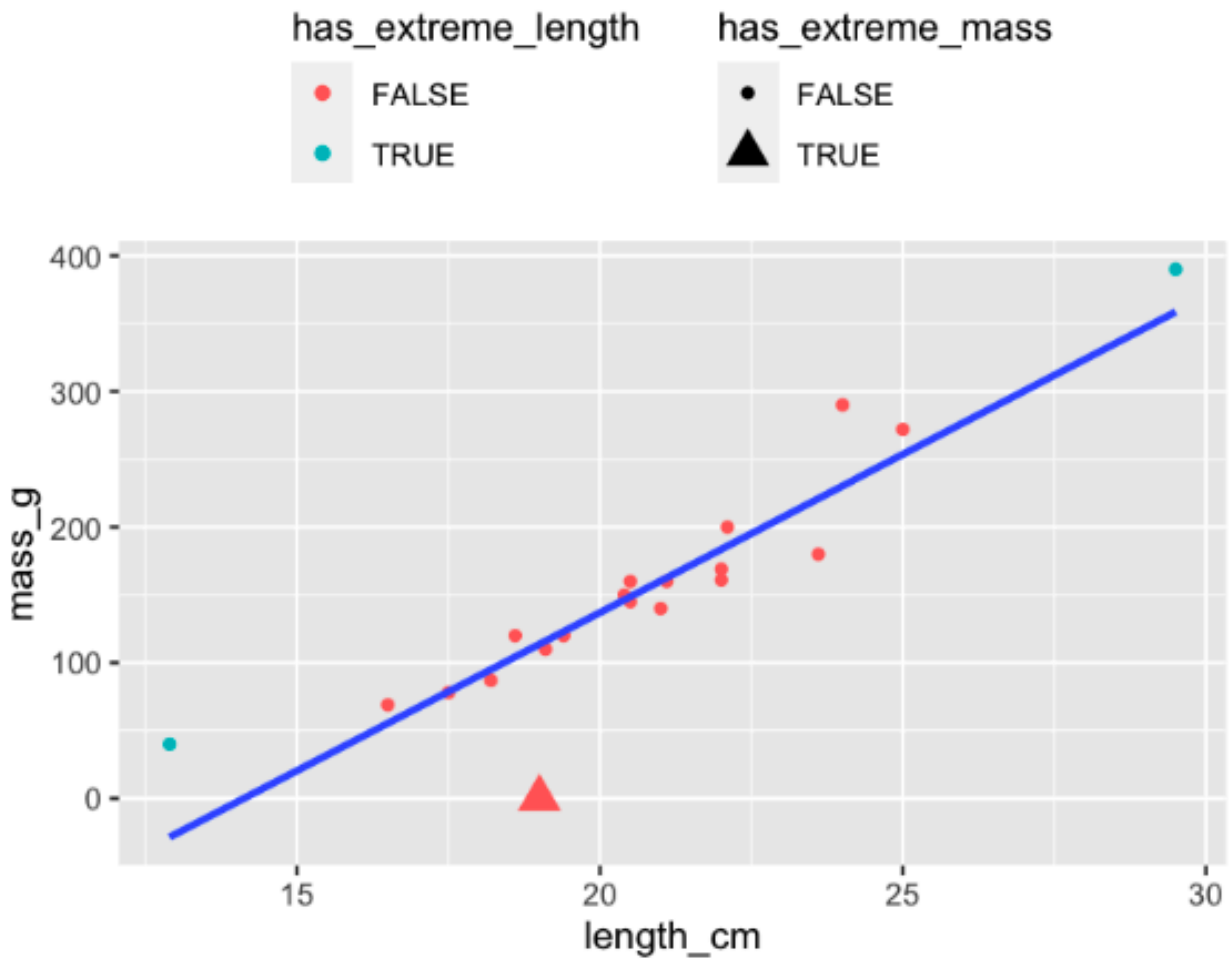
Influence

Another type of outlier is when the point lies a long way from the model predictions

Influence measures how much the model would change if you left the observation out of the dataset when modeling.



```
roach %>%
  mutate(
    has_extreme_length = length_cm < 15 | length_cm > 26,
    has_extreme_mass = mass_g < 1
  ) %>%
  ggplot(aes(length_cm, mass_g)) +
  geom_point(
    aes(
      color = has_extreme_length,
      shape = has_extreme_mass
    )
  ) +
  geom_smooth(method = "lm", se = FALSE)
```



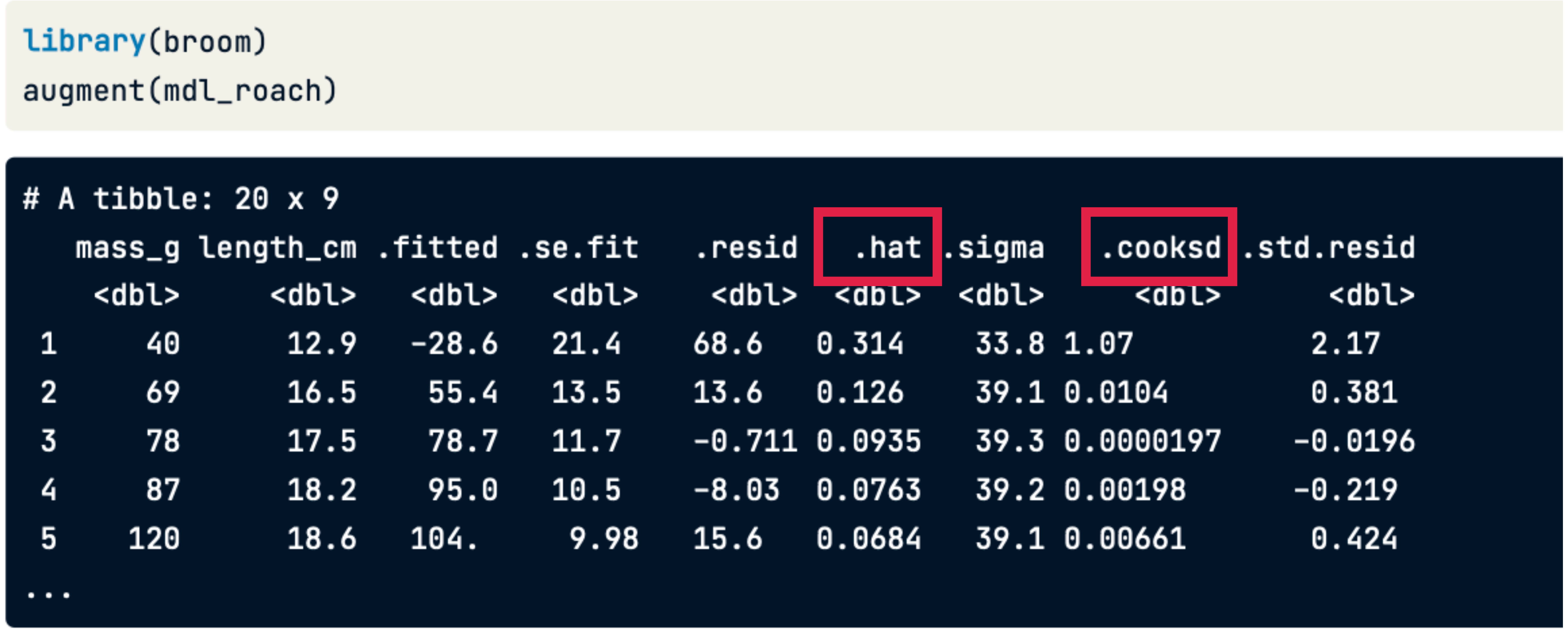
Far away point — mass = 0 for a certain length (mass is unlikely to be 0)

Outliers, leverage, and influence

broom augment(model object) to retrieve

leverage: **.hat** column

influence: **.cooksd** column



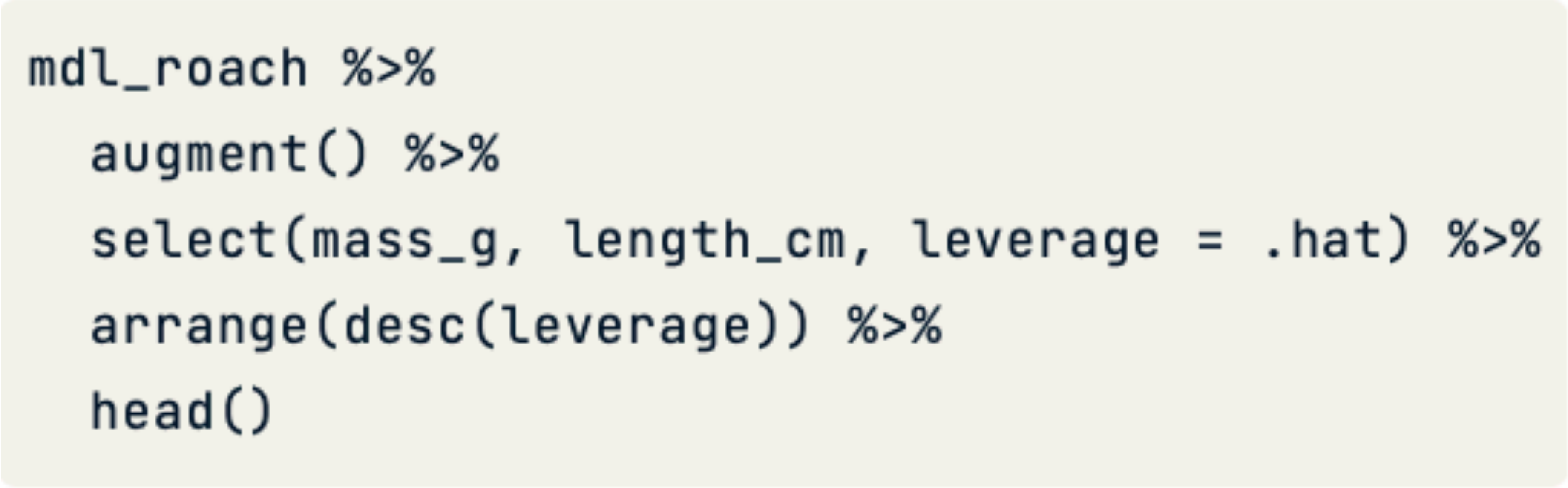
```r
library(broom)
augment(mdl_roach)
```

```
# A tibble: 20 x 9
   mass_g length_cm .fitted .se.fit   .resid   .hat .sigma  .cooksd .std.resid
    <dbl>     <dbl>   <dbl>   <dbl>    <dbl>  <dbl>  <dbl>    <dbl>      <dbl>
 1     40      12.9   -28.6    21.4   68.6   0.314    33.8 1.07           2.17
 2     69      16.5    55.4    13.5   13.6   0.126    39.1 0.0104         0.381
 3     78      17.5    78.7    11.7   -0.711 0.0935   39.3 0.0000197     -0.0196
 4     87      18.2    95.0    10.5   -8.03  0.0763   39.2 0.00198       -0.219
 5    120      18.6   104.      9.98  15.6   0.0684   39.1 0.00661        0.424
...
```

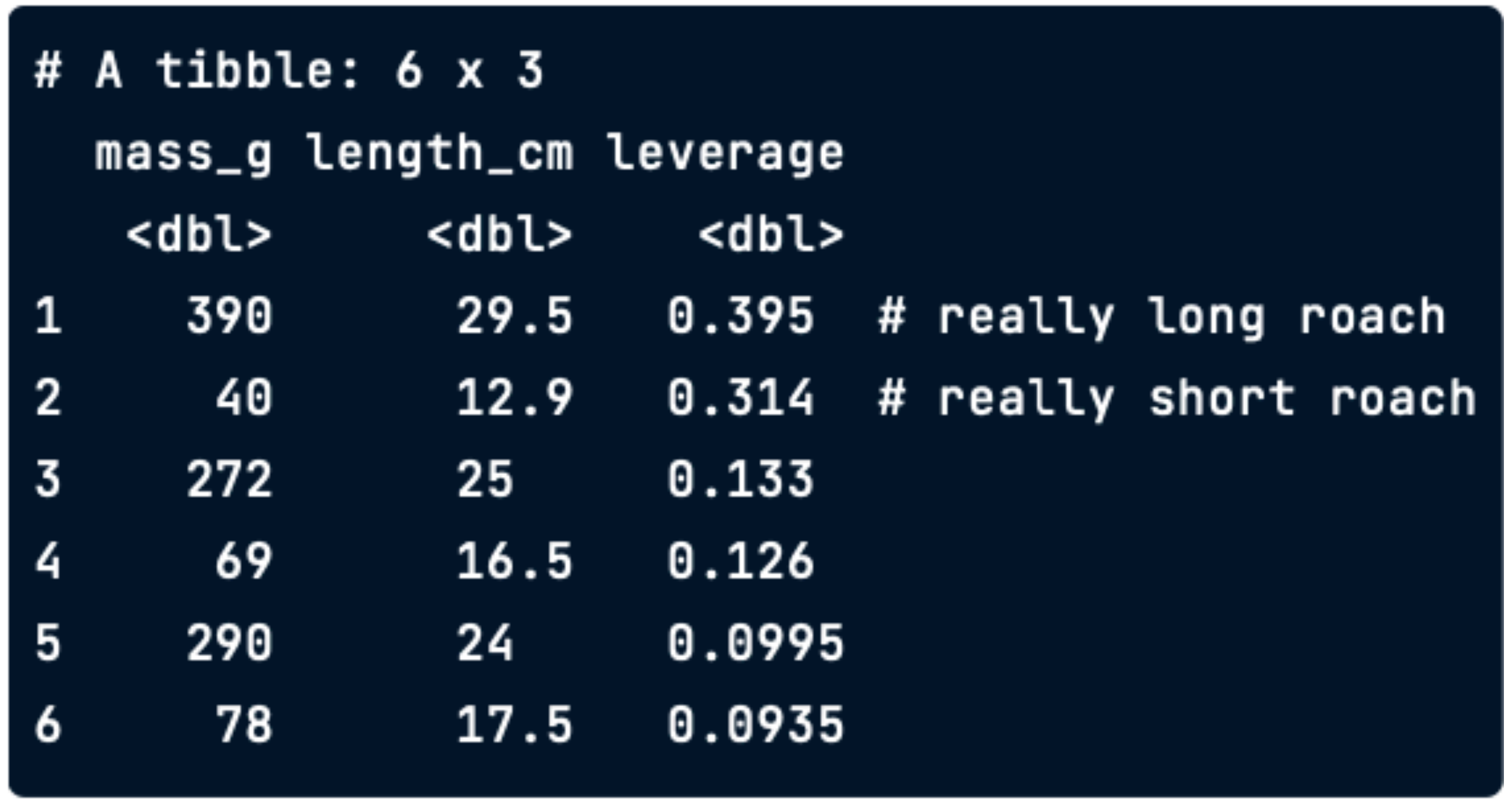
Outliers, **leverage**, and influence

Find the values with the most leverage

Sort the .hat to find the highly leveraged values *You are finding outliers programmatically*



```
mdl_roach %>%
  augment() %>%
  select(mass_g, length_cm, leverage = .hat) %>%
  arrange(desc(leverage)) %>%
  head()
```



```
# A tibble: 6 x 3
  mass_g length_cm leverage
   <dbl>     <dbl>    <dbl>
1    390      29.5   0.395  # really long roach
2     40      12.9   0.314  # really short roach
3    272      25     0.133
4     69      16.5   0.126
5    290      24     0.0995
6     78      17.5   0.0935
```

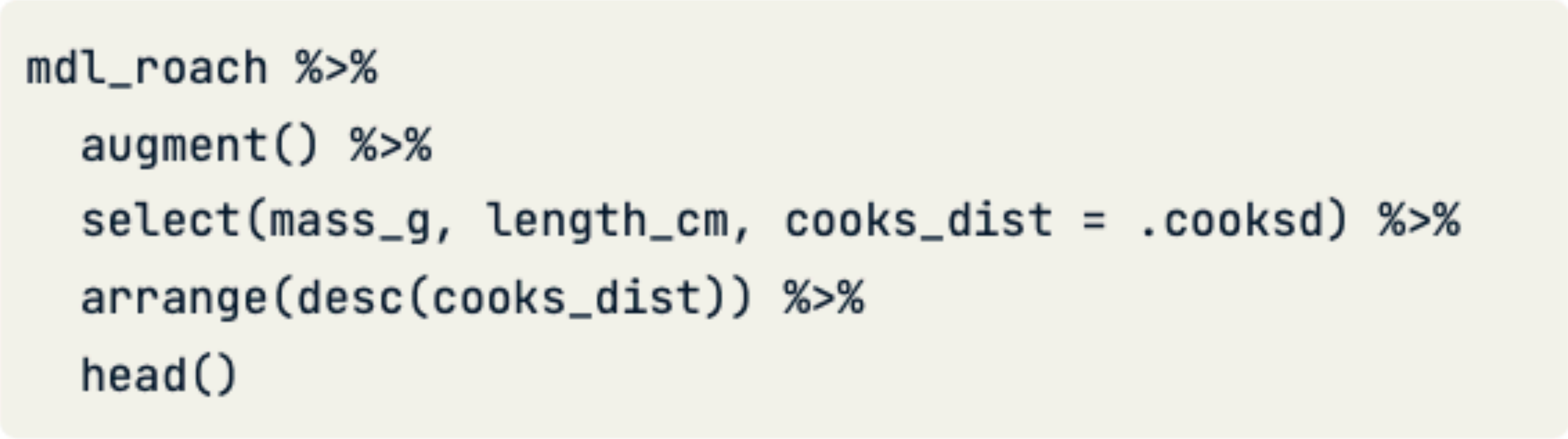
Programmatic approach:

1. augment
2. select the columns of interest
3. arrange to get the top values

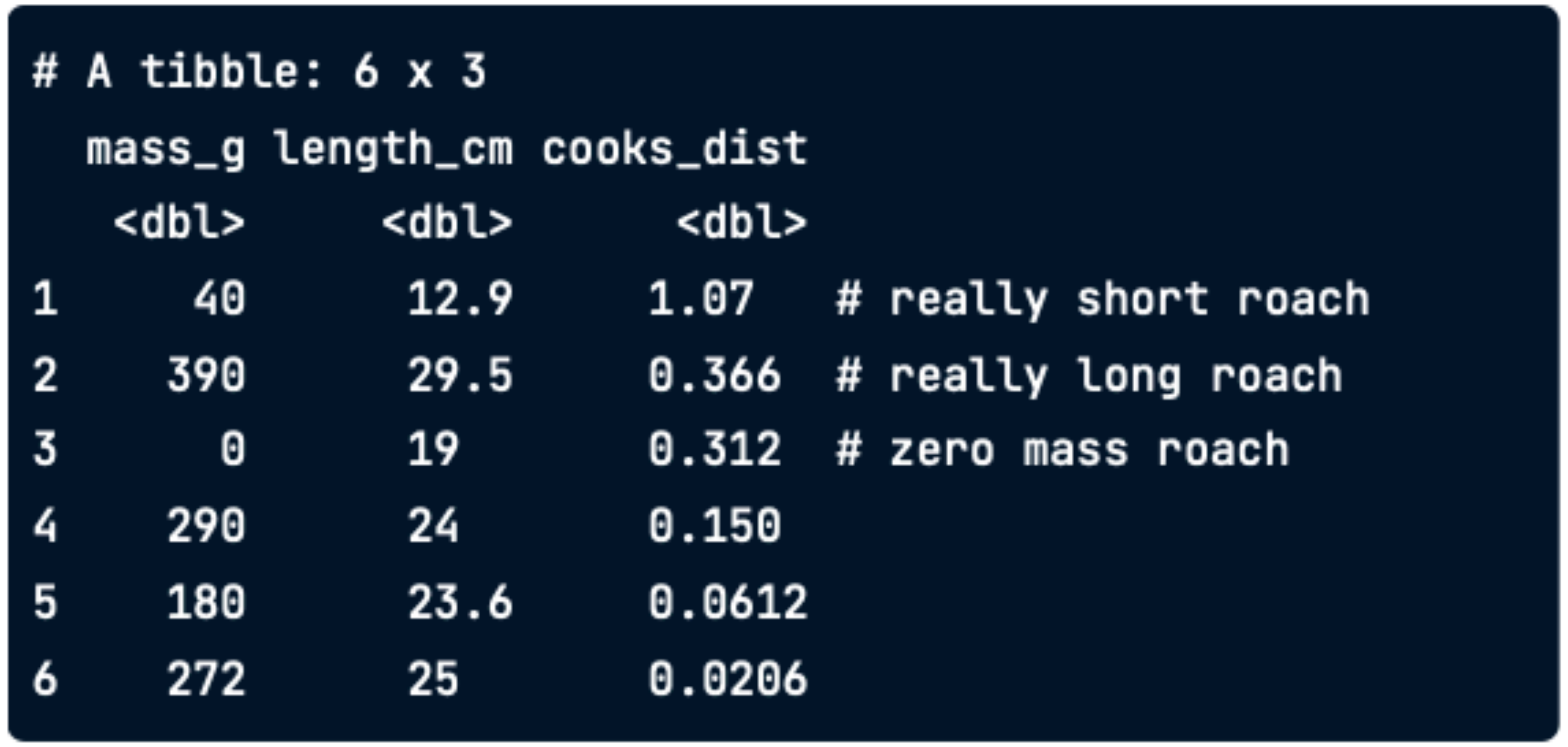
Outliers, leverage, and **influence**

Find the values with the most leverage

Sort the .hat to find the highly leveraged values *You are finding outliers programmatically*



```
mdl_roach %>%
  augment() %>%
  select(mass_g, length_cm, cooks_dist = .cooksd) %>%
  arrange(desc(cooks_dist)) %>%
  head()
```



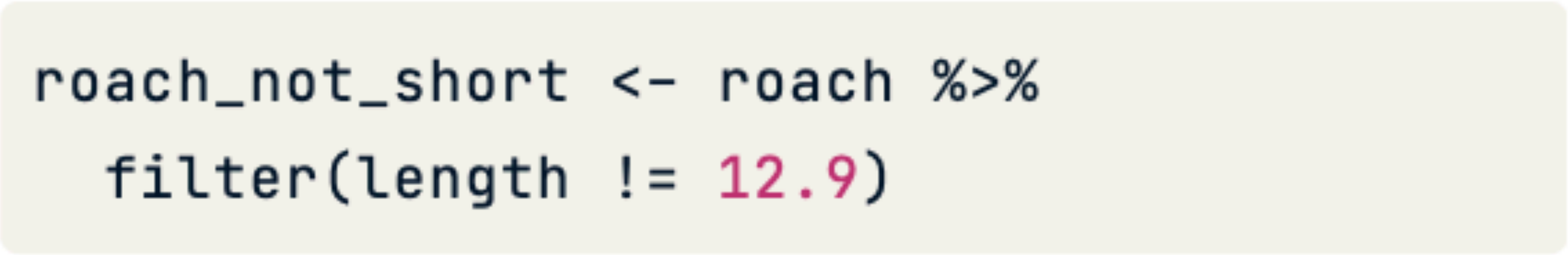
```
# A tibble: 6 x 3
  mass_g length_cm cooks_dist
   <dbl>     <dbl>      <dbl>
1     40      12.9     1.07   # really short roach
2    390      29.5     0.366  # really long roach
3      0      19       0.312  # zero mass roach
4    290      24       0.150
5    180      23.6     0.0612
6    272      25       0.0206
```

Same Programmatic approach:

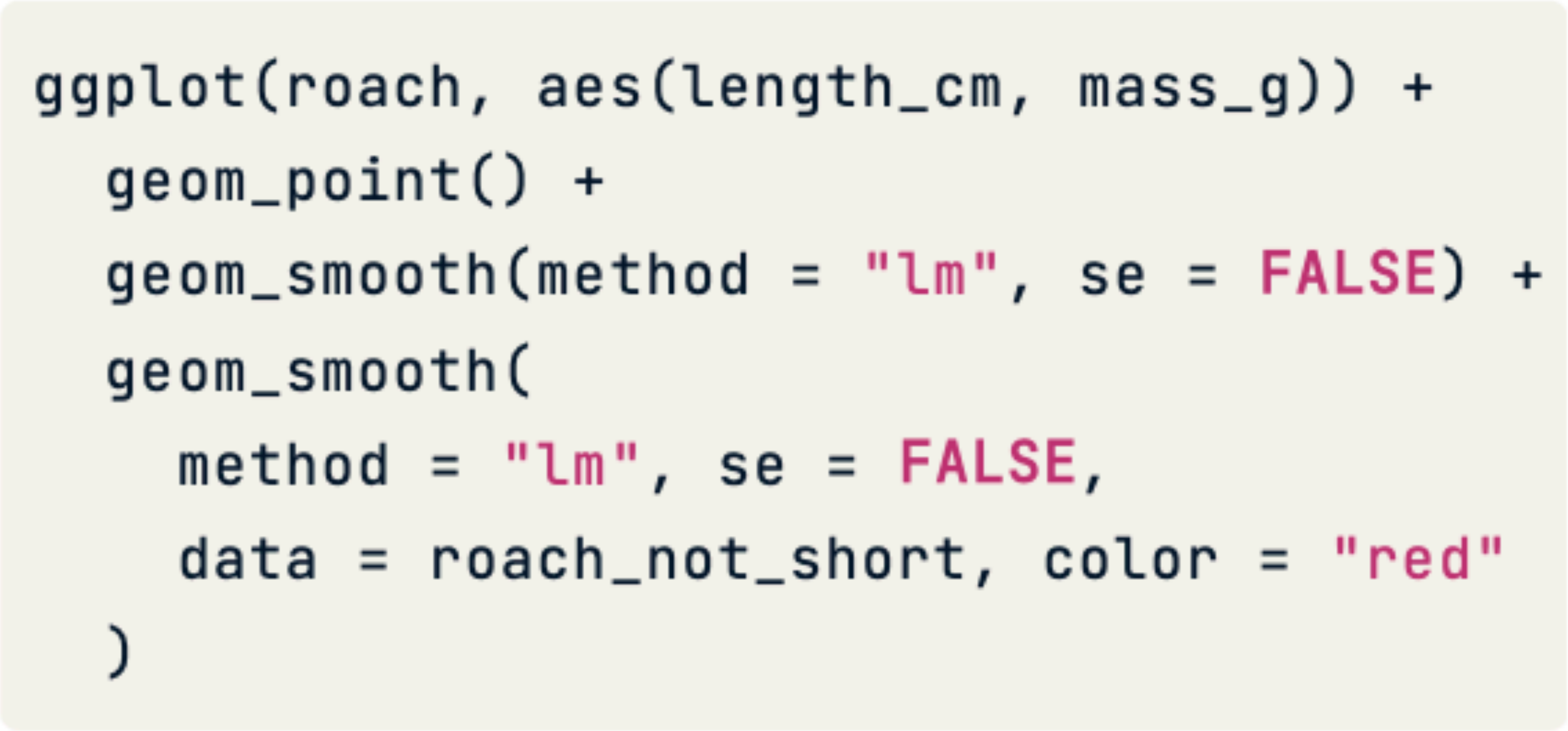
1. augment
2. select the columns of interest
3. arrange to get the top values

See how **influence** works (by removing the most influential points)

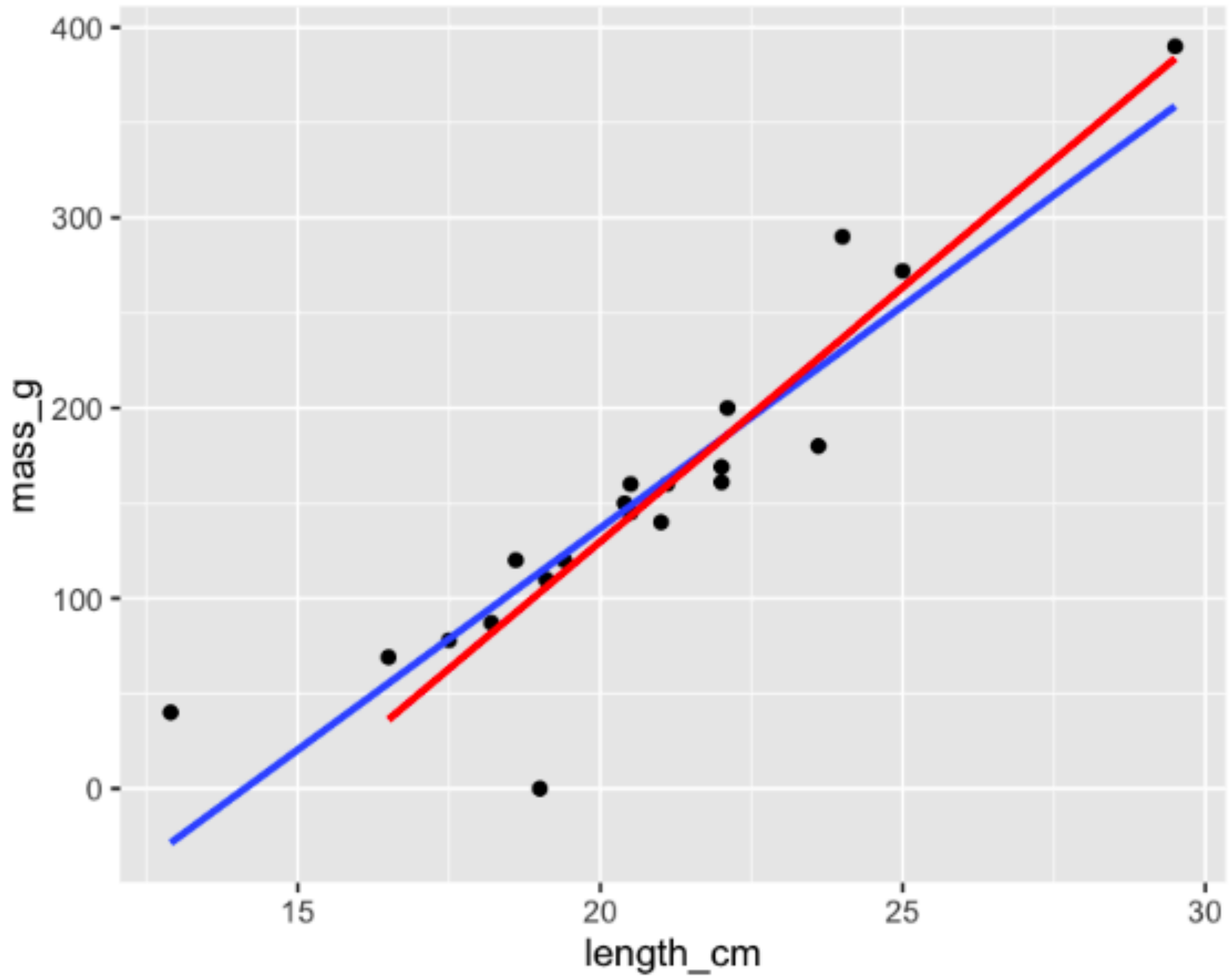
Draw the usual plot, but add another regression line using the dataset without one of the influential points.



```
roach_not_short <- roach %>%
  filter(length != 12.9)
```



```
ggplot(roach, aes(length_cm, mass_g)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  geom_smooth(
    method = "lm", se = FALSE,
    data = roach_not_short, color = "red"
  )
```



Slope of the line change by having one less data point.

Influence measures how much the model would change if you left the observation out of the dataset when modeling.

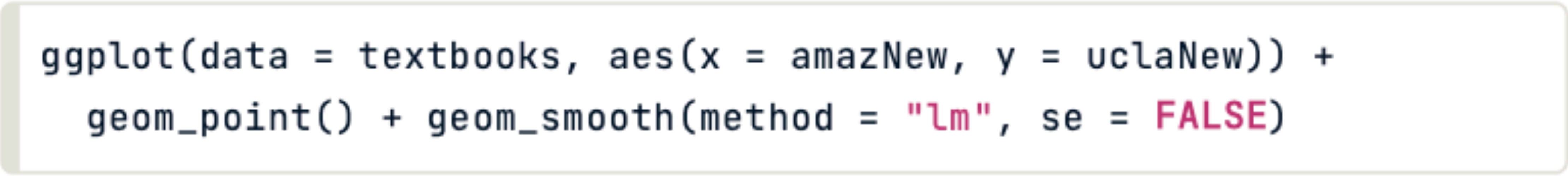
Lab - part 3

Visualizing model fit & Extracting outliers

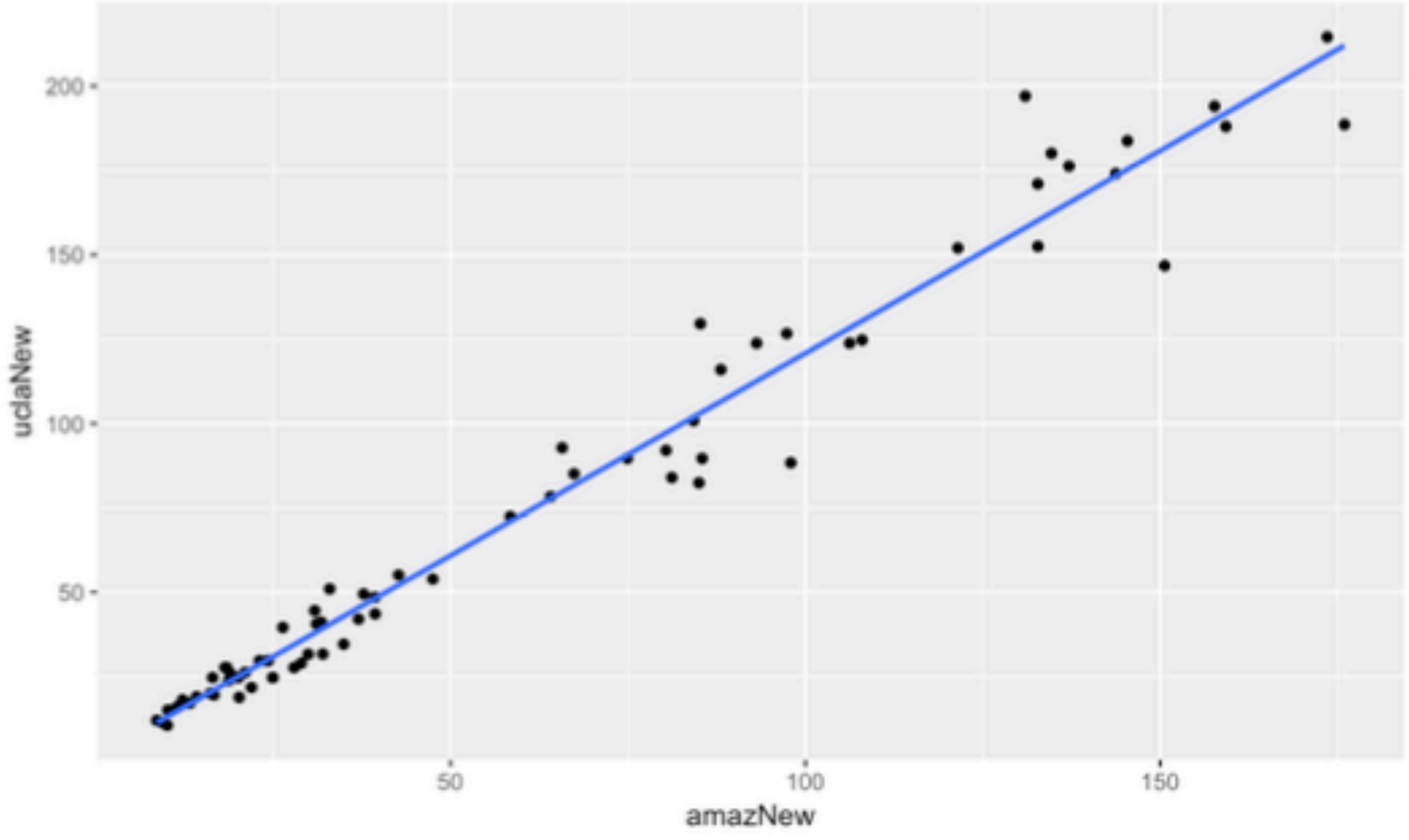
Quality of model fit *(brief review from last class)*

How well does the model work?

How well does our textbook model fit?

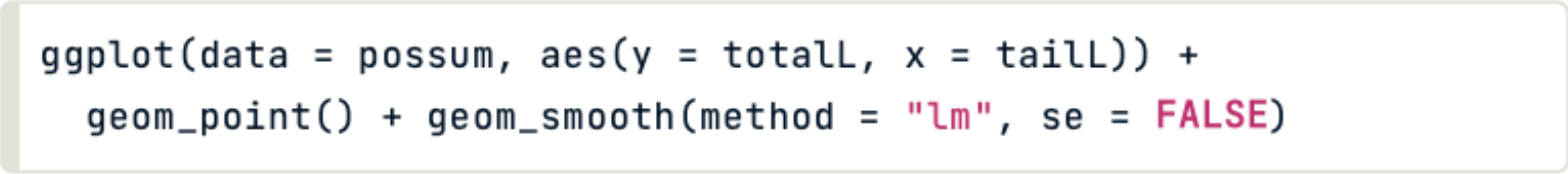


```r
ggplot(data = textbooks, aes(x = amazNew, y = uclaNew)) +
  geom_point() + geom_smooth(method = "lm", se = FALSE)
```

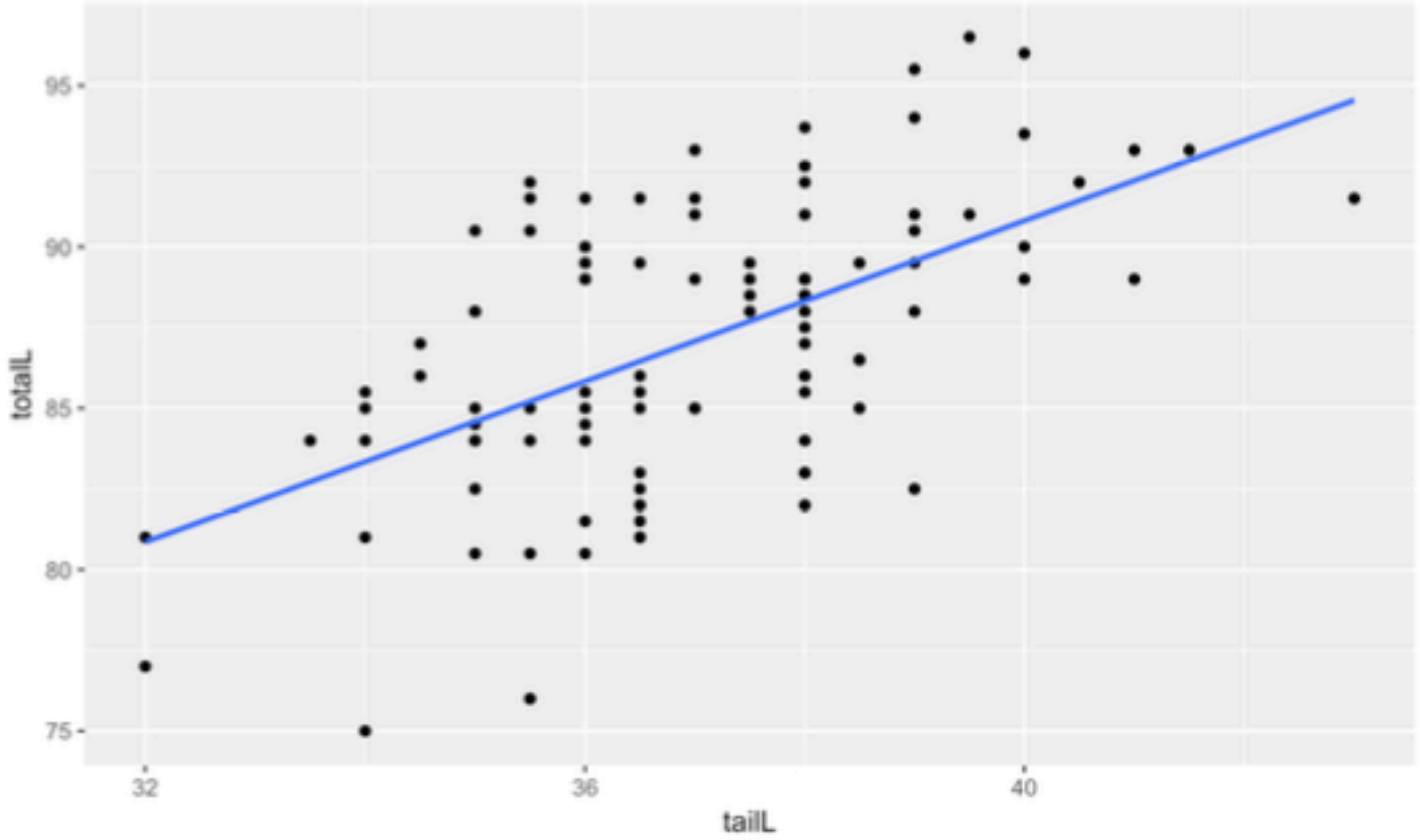


Residual Std. error: 10.5

How well does our possum model fit?



```r
ggplot(data = possum, aes(y = totalL, x = tailL)) +
  geom_point() + geom_smooth(method = "lm", se = FALSE)
```

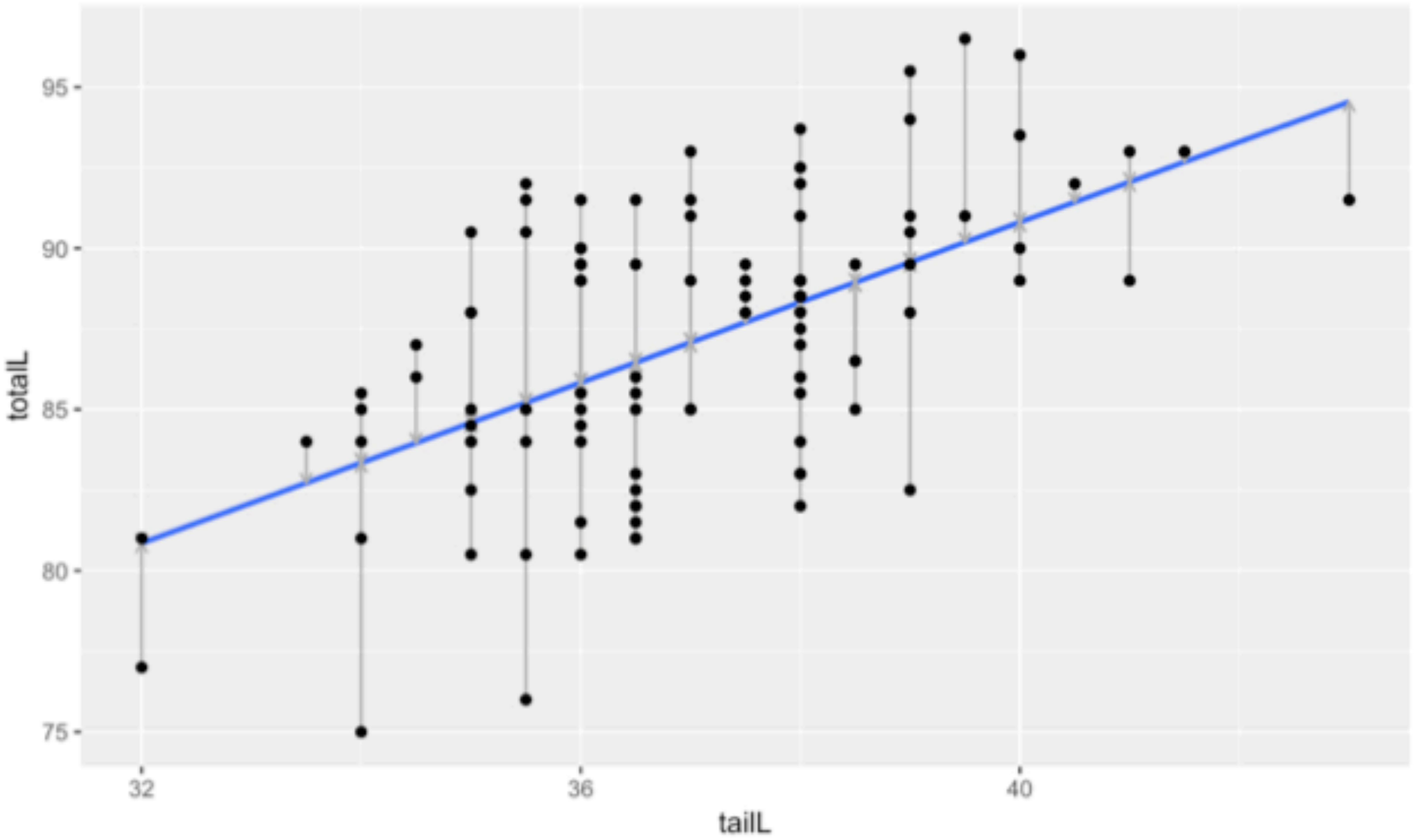


Residual Std. error: 3.57

Sum of squared deviations

Can we quantify our intuition of the model fit?

SS is the vertical distance between that point and the fitted line



To minimize SS, make the grey lines shorter, collectively and after squaring them

RMSE

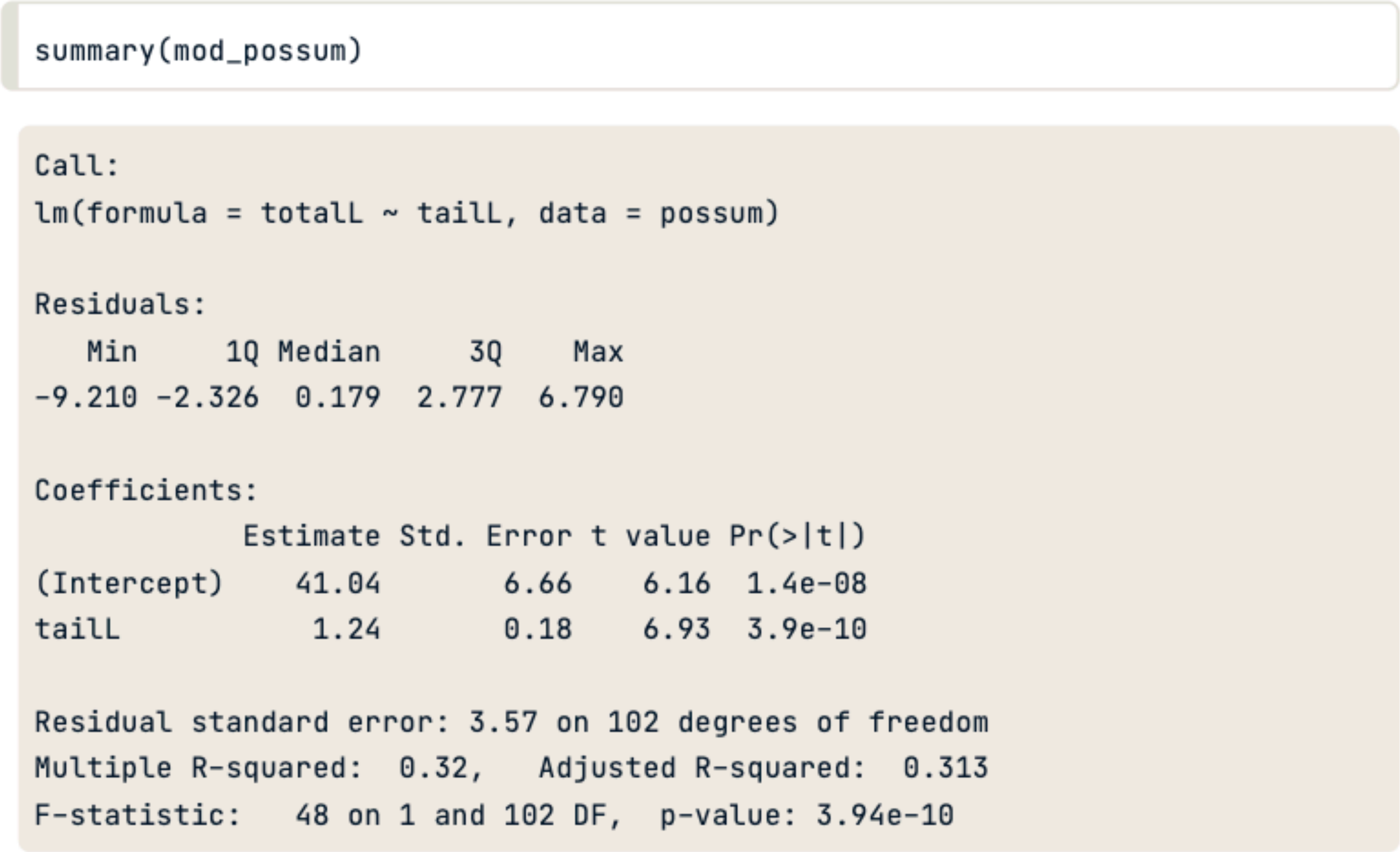
Model Accuracy

Divide by the number of degrees of freedom.

$$RMSE = \sqrt{\frac{\sum_i e_i^2}{d.f}} = \sqrt{\frac{SSE}{n-2}}$$

RMSE

Residual standard error for possum data



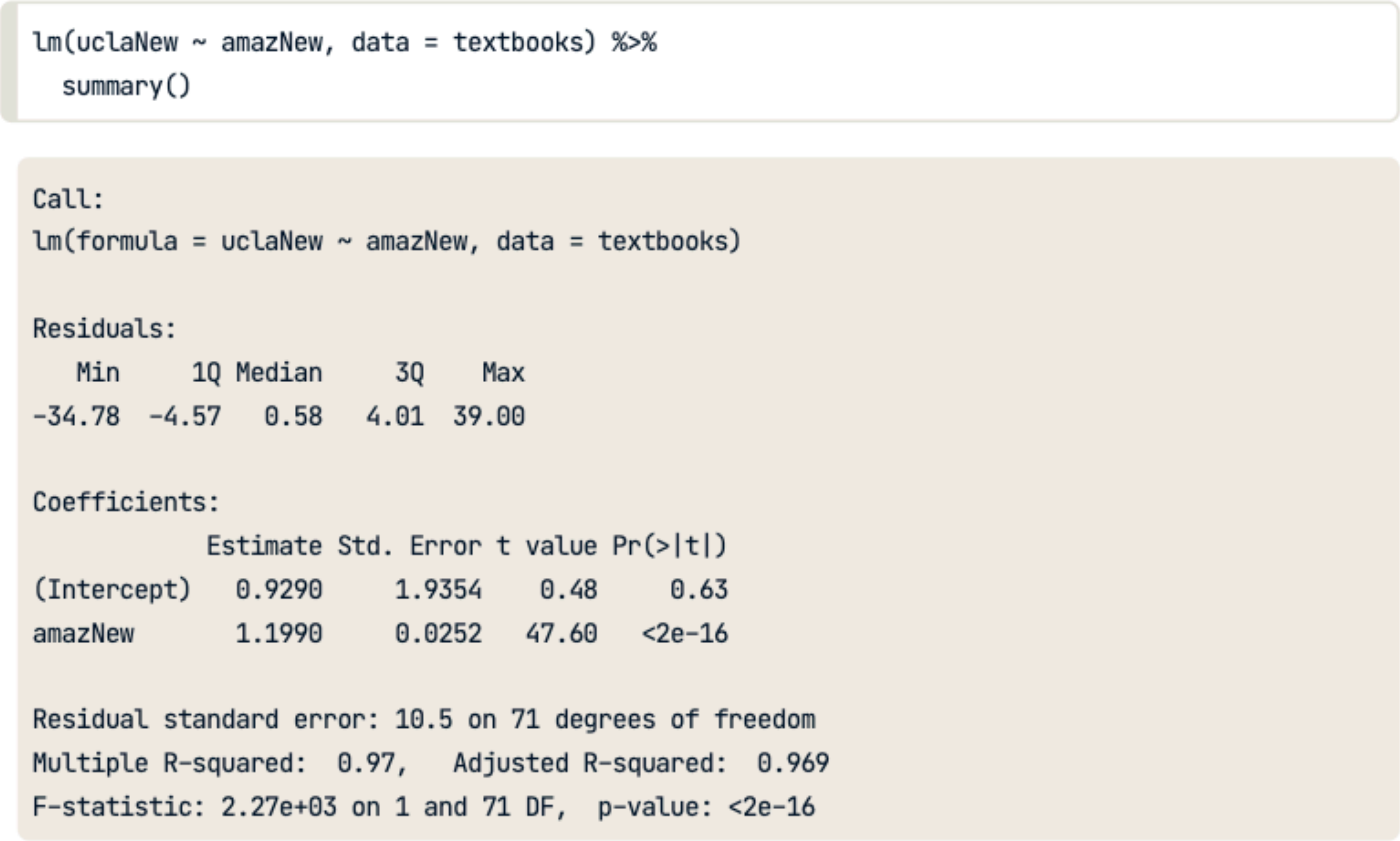
```
summary(mod_possum)
```

```
Call:
lm(formula = totalL ~ tailL, data = possum)

Residuals:
   Min     1Q Median     3Q    Max 
-9.210 -2.326  0.179  2.777  6.790 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    41.04       6.66    6.16  1.4e-08
tailL           1.24       0.18    6.93  3.9e-10

Residual standard error: 3.57 on 102 degrees of freedom
Multiple R-squared:  0.32,   Adjusted R-squared:  0.313 
F-statistic:   48 on 1 and 102 DF,  p-value: 3.94e-10
```

RMSE value interpretation: Our model makes a predicted body length that is typically within 3.57 centimeters of the truth

RMSE

Residual standard error for the textbooks data



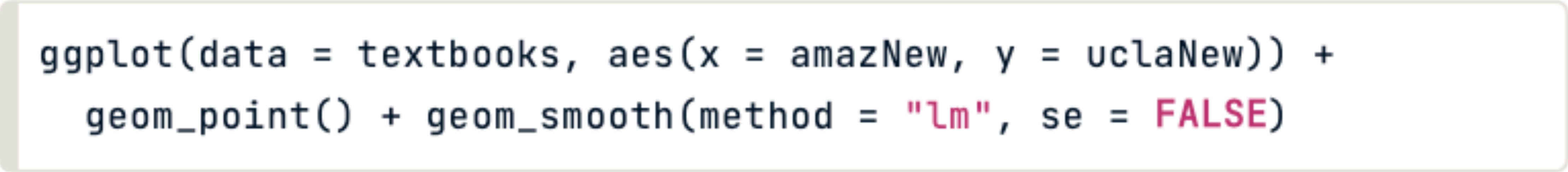
```
lm(uclaNew ~ amazNew, data = textbooks) %>%
  summary()
```

```
Call:
lm(formula = uclaNew ~ amazNew, data = textbooks)

Residuals:
   Min     1Q Median     3Q    Max
-34.78  -4.57   0.58   4.01  39.00

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.9290     1.9354    0.48     0.63
amazNew       1.1990     0.0252   47.60   <2e-16

Residual standard error: 10.5 on 71 degrees of freedom
Multiple R-squared:  0.97,   Adjusted R-squared:  0.969
F-statistic: 2.27e+03 on 1 and 71 DF,  p-value: <2e-16
```

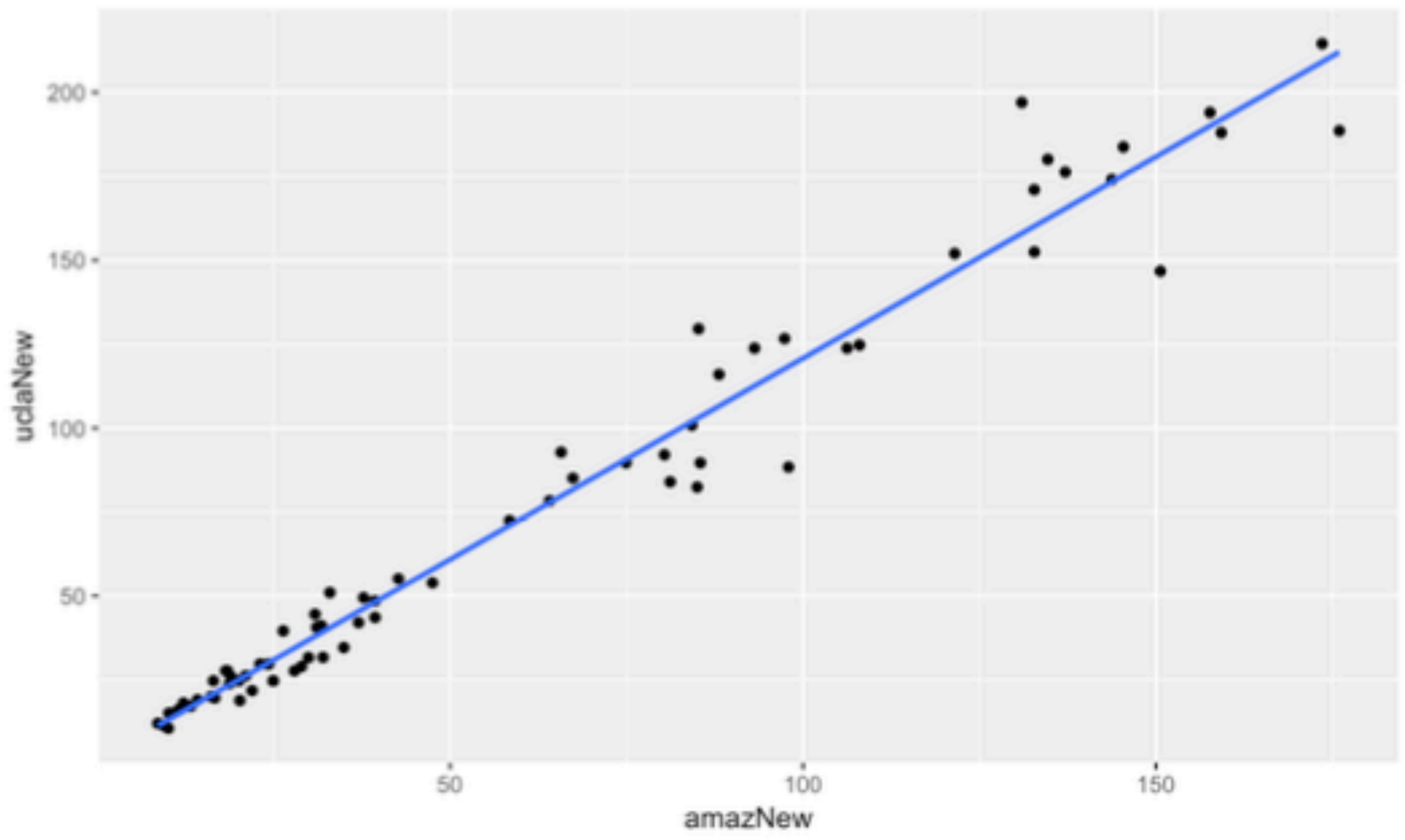
RMSE value interpretation?

Quality of model fit — why we need null model

How well does our textbook model fit?

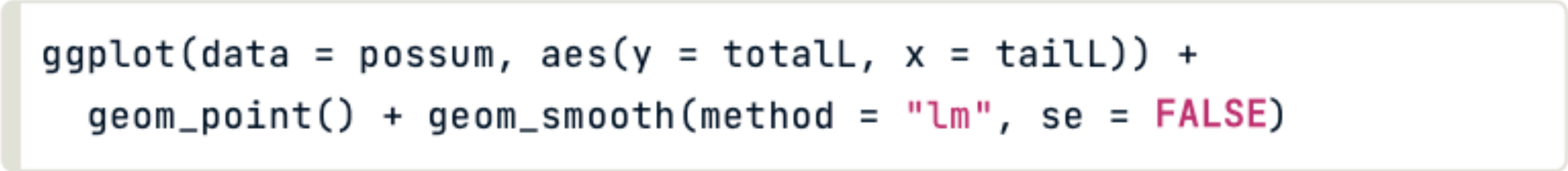


```
ggplot(data = textbooks, aes(x = amazNew, y = uclaNew)) +
  geom_point() + geom_smooth(method = "lm", se = FALSE)
```

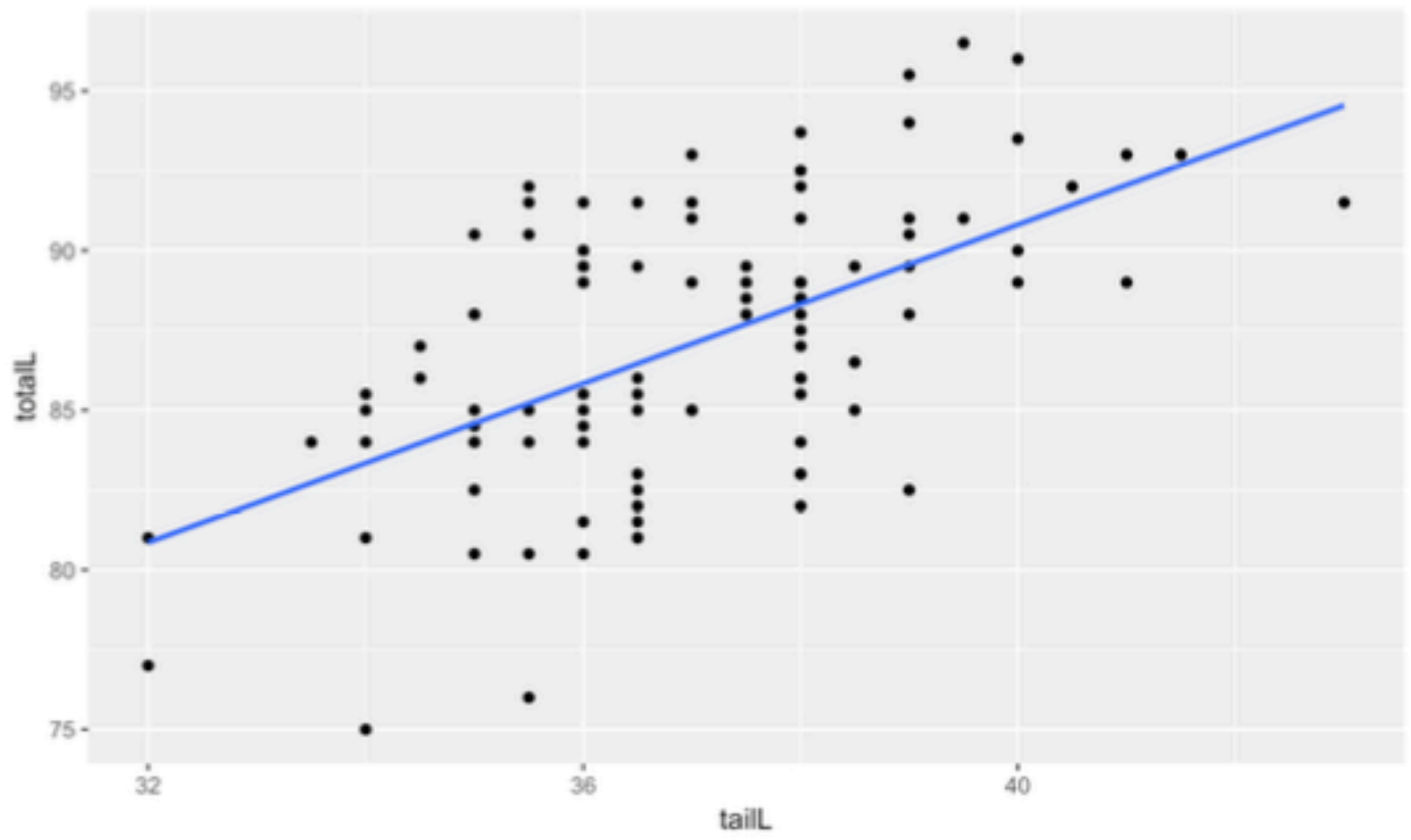


Residual Std. error: 10.5

How well does our possum model fit?



```
ggplot(data = possum, aes(y = totalL, x = tailL)) +
  geom_point() + geom_smooth(method = "lm", se = FALSE)
```



Residual Std. error: 3.57

We need a way to compare the quality of the fit that was unitles.

Null model

Benchmark model

Intuition: If you had to predict a response variable (say body length of a possum), but you didn't have any information about that particular possum, what would be your prediction be?

A sensible choice: would be the average length of all possums.

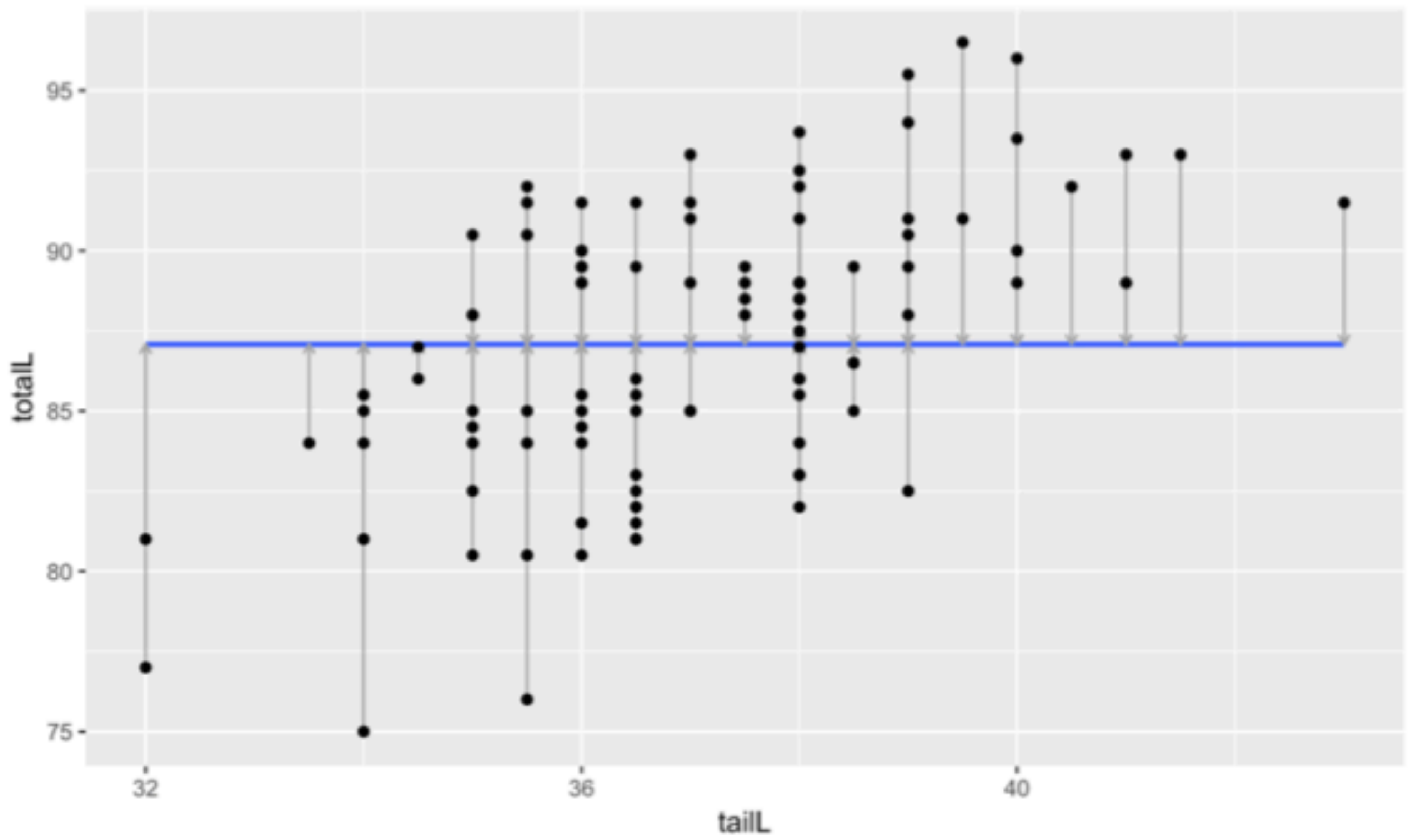
For all observations: predicted value = average value

$$\hat{y} = \bar{y}$$

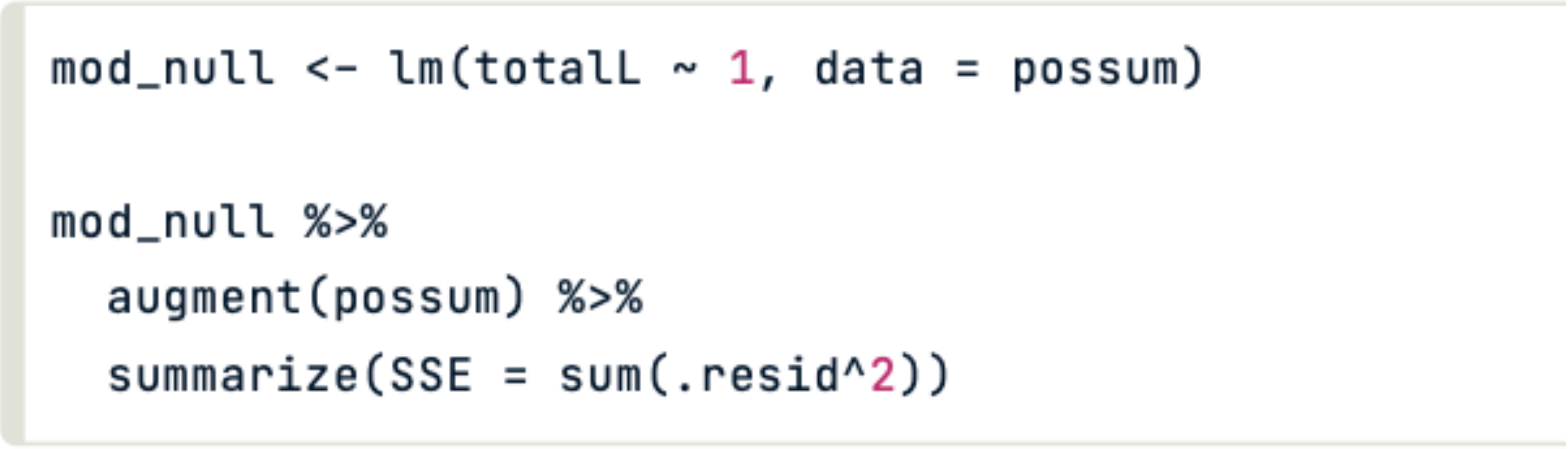
Called a **null model**, since it doesn't require any insight to make and yet there is no reasonable model that could be worse that this model

Null model

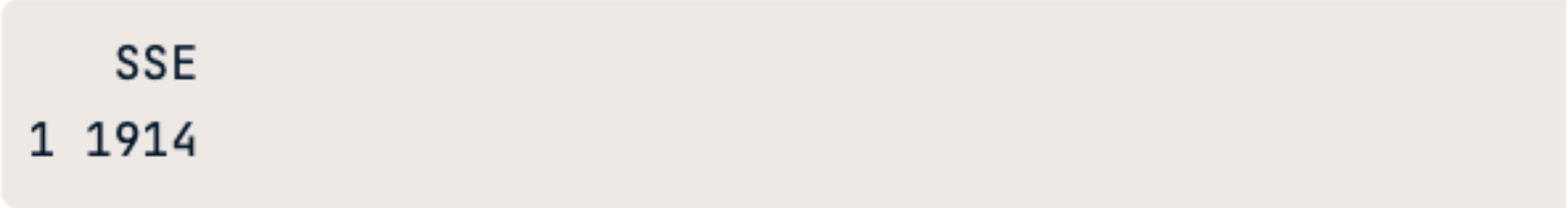
Visualizing null model



Computing null model



```
mod_null <- lm(totalL ~ 1, data = possum)

mod_null %>%
  augment(possum) %>%
  summarize(SSE = sum(.resid^2))
```

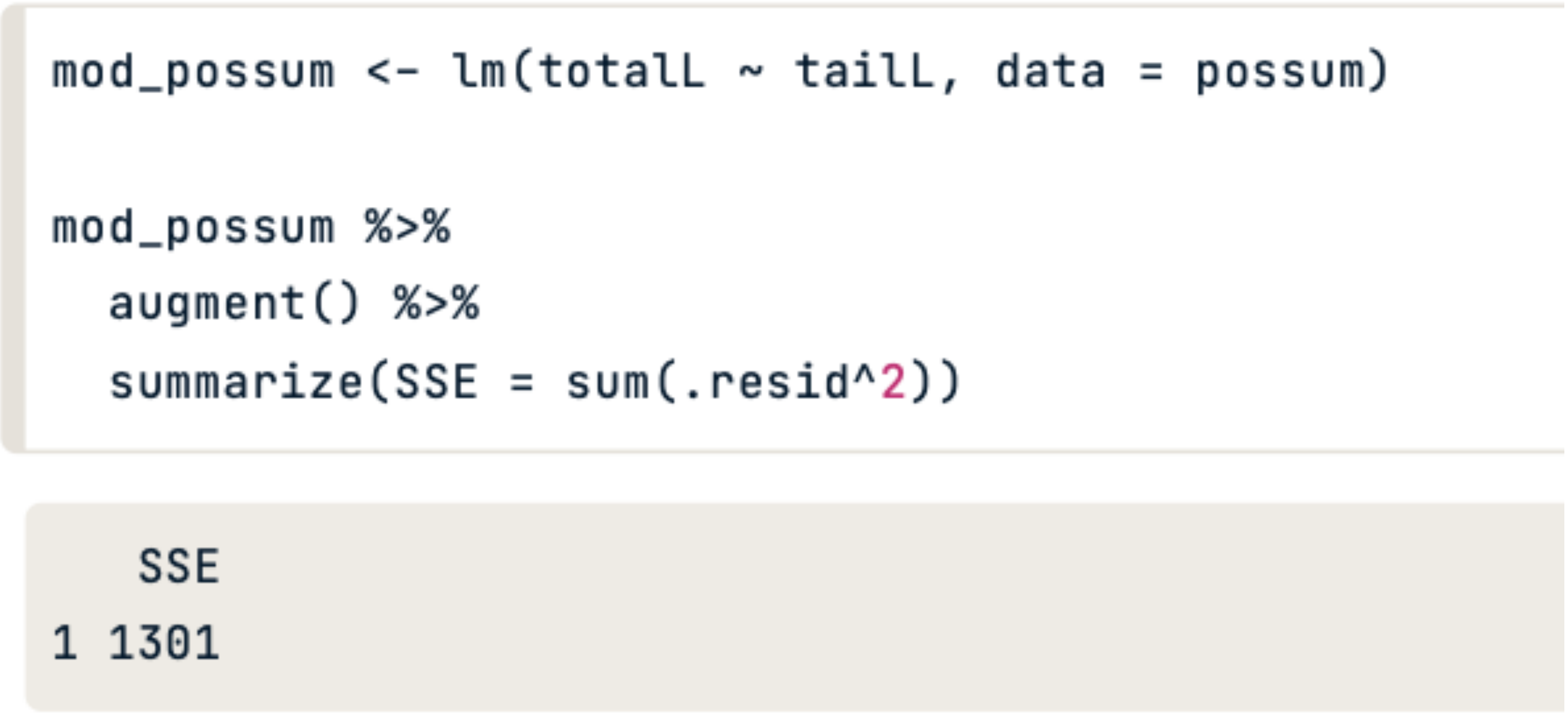


```
   SSE
1 1914
```

Null vs. Linear model

Compare SSE of null vs. linear model

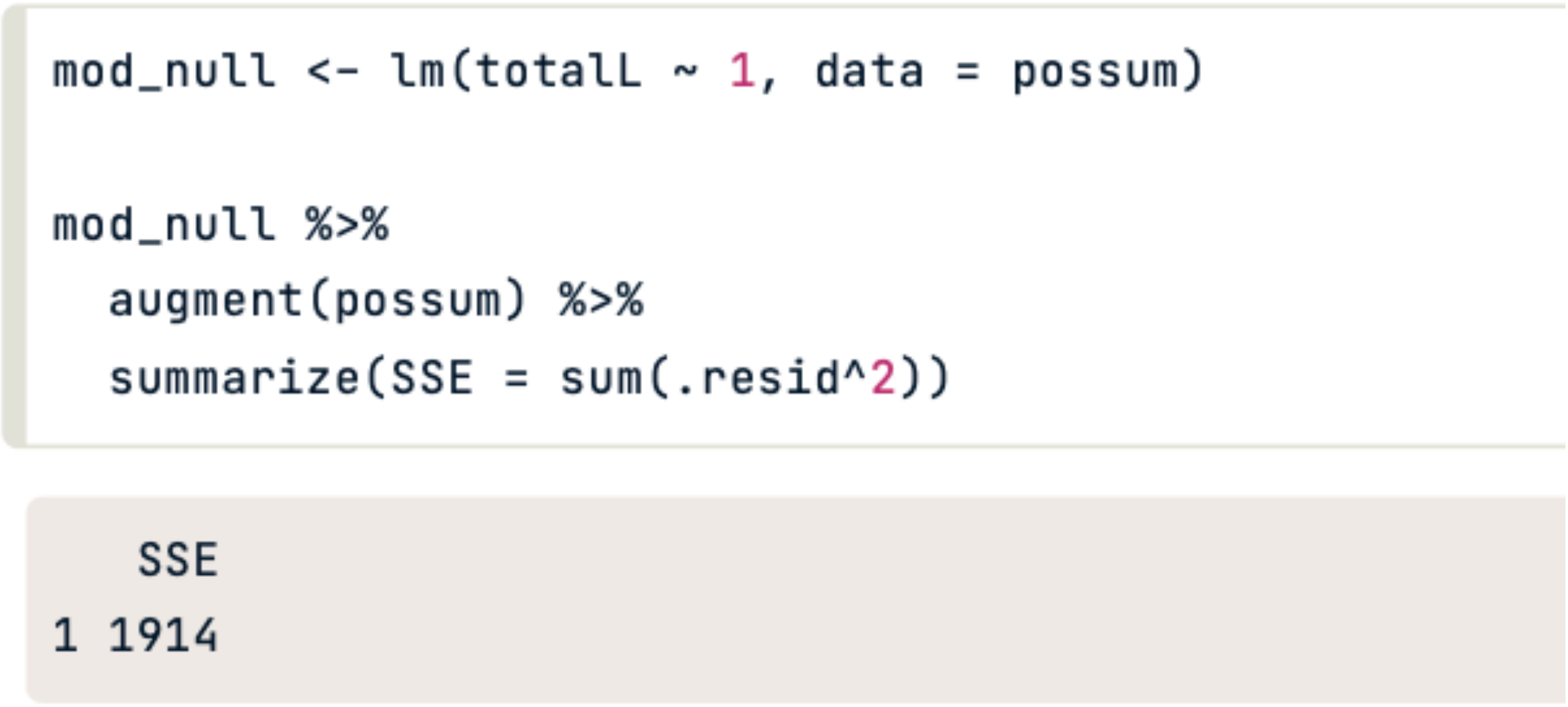
Computing linear model



```
mod_possum <- lm(totalL ~ tailL, data = possum)

mod_possum %>%
  augment() %>%
  summarize(SSE = sum(.resid^2))
```

```
   SSE
1 1301
```

Computing null model



```
mod_null <- lm(totalL ~ 1, data = possum)

mod_null %>%
  augment(possum) %>%
  summarize(SSE = sum(.resid^2))
```

```
   SSE
1 1914
```

SSE of null model is also called the SST (total sum of squares)

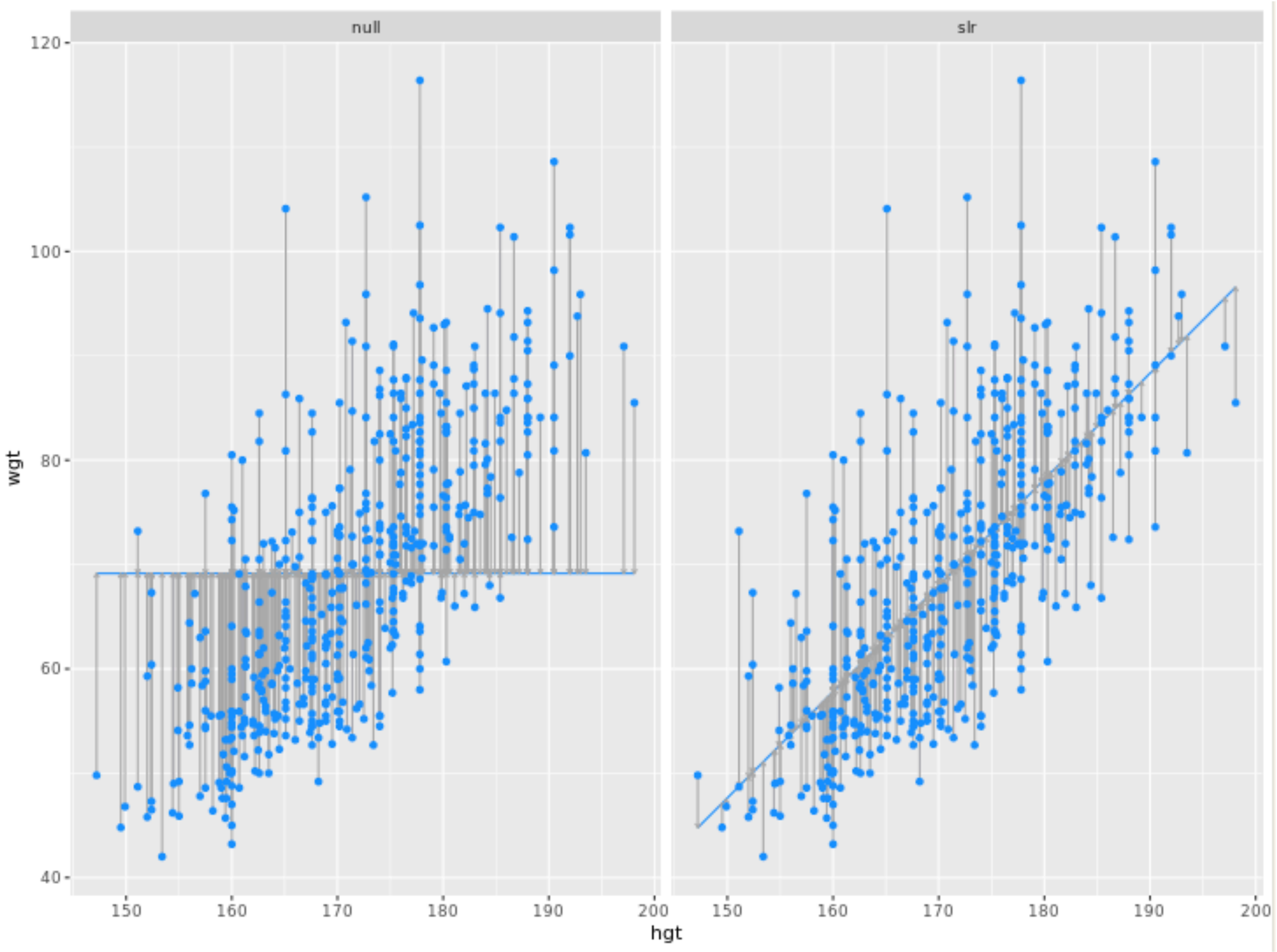
Null vs. Linear model

Visualizing

if you squared the lengths of the grey arrows on the left and summed them up, you would get a larger value than if you performed the same operation on the grey arrows on the right.

NULL model

Linear model



Coefficient of determination R^2

Quantification of the variability explained by our regression model

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{Var(e)}{Var(y)}$$

SSE of null model is also called the SST (total sum of squares)

Ratio of SSE of linear by null explains the **variability in the response variable**.

By building a regression model, we hope to explain some of that variability. The portion of the SST variability that is not explained by our model is the SSE

R2 is the measure of the quality of of the fit of a regression model

Connection to correlation

For a simple linear regression

$$r_{x,y}^2 = R^2$$

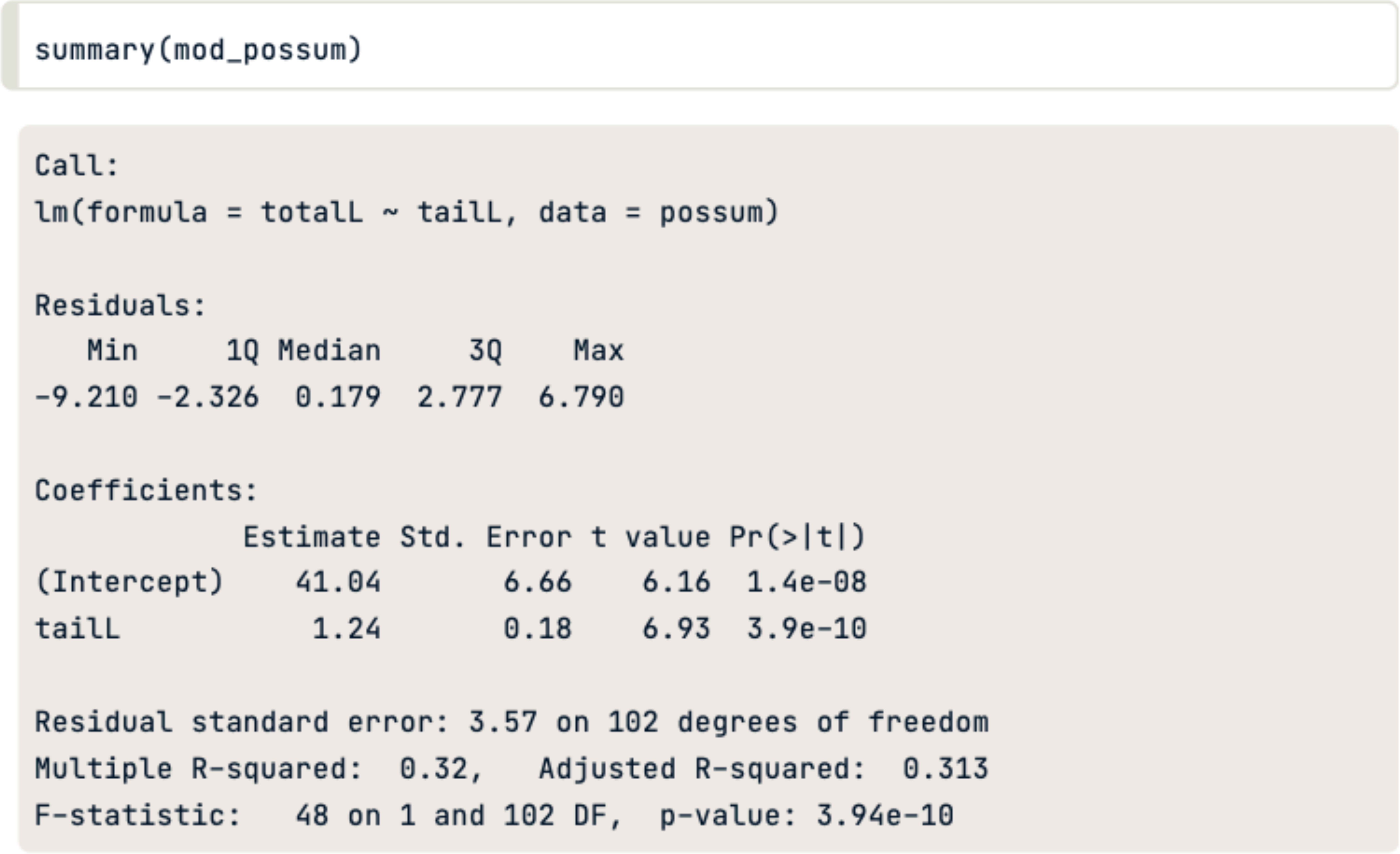
Then why do we need both quantities?

Correlation can only be applicable to a bivariate quantity (single response and single explanatory variable, or simple linear regression)

Regression is a much more generic framework.

R2 via the summary function

R2 interpretation: Our model based on tail length explains about 32% variability in body length for these possums.



```
summary(mod_possum)
```

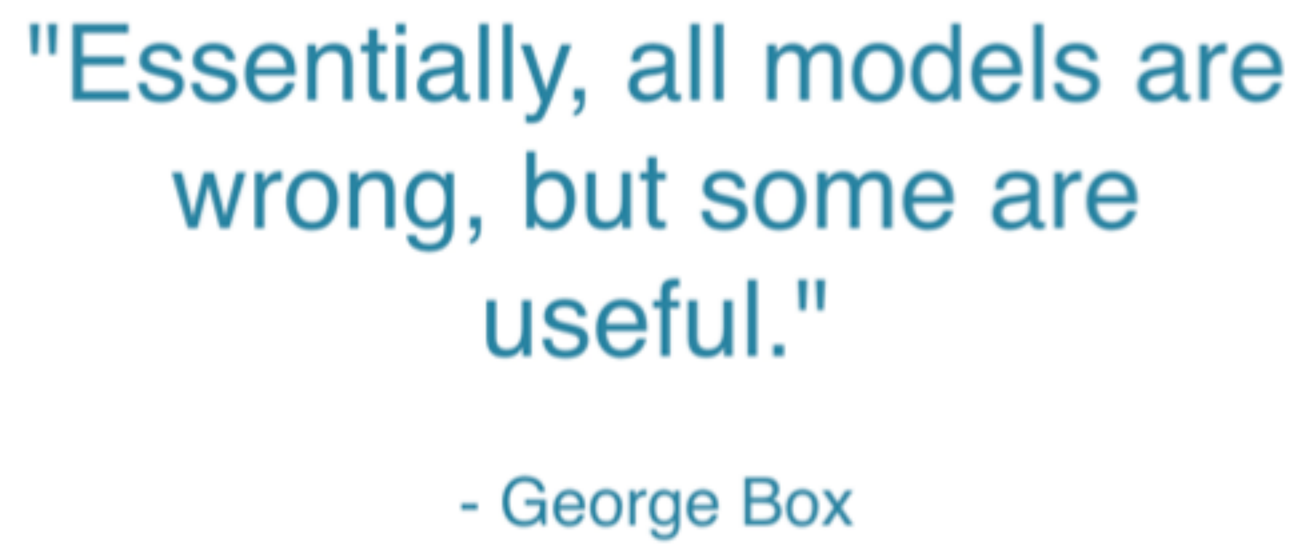
```
Call:
lm(formula = totalL ~ tailL, data = possum)

Residuals:
   Min      1Q Median      3Q     Max
-9.210 -2.326  0.179  2.777  6.790

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    41.04       6.66    6.16  1.4e-08
tailL           1.24       0.18    6.93  3.9e-10

Residual standard error: 3.57 on 102 degrees of freedom
Multiple R-squared:  0.32,	Adjusted R-squared:  0.313
F-statistic:   48 on 1 and 102 DF,  p-value: 3.94e-10
```

Over reliance on R2

While high R2 is better,

A model with high R2 doesn’t alone mean that the model fit is good. There could be **overfitting**.

A model with a low R2 can still be useful and provide statistically significant insight. Determinig what is low also depends on the context of the domain (e.g. human behavior modeling quite hard and low R2 might still be good).



"Essentially, all models are wrong, but some are useful."

- George Box

R-squared vs. adjusted R-squared

Two common measures of how well a model fits to data

Coefficient of determination

$$R^2 = 1 - \frac{SSE}{SST},$$

As SSE gets smaller, R2 gets higher (generally denoting a higher model fit)

Issue with R2 is that the SSE can only decrease as new variables are added to the model, while the SST does not change. So you can increase R2 by adding ***any*** additional variable—even random noise

More of an issue in multiple linear regression.

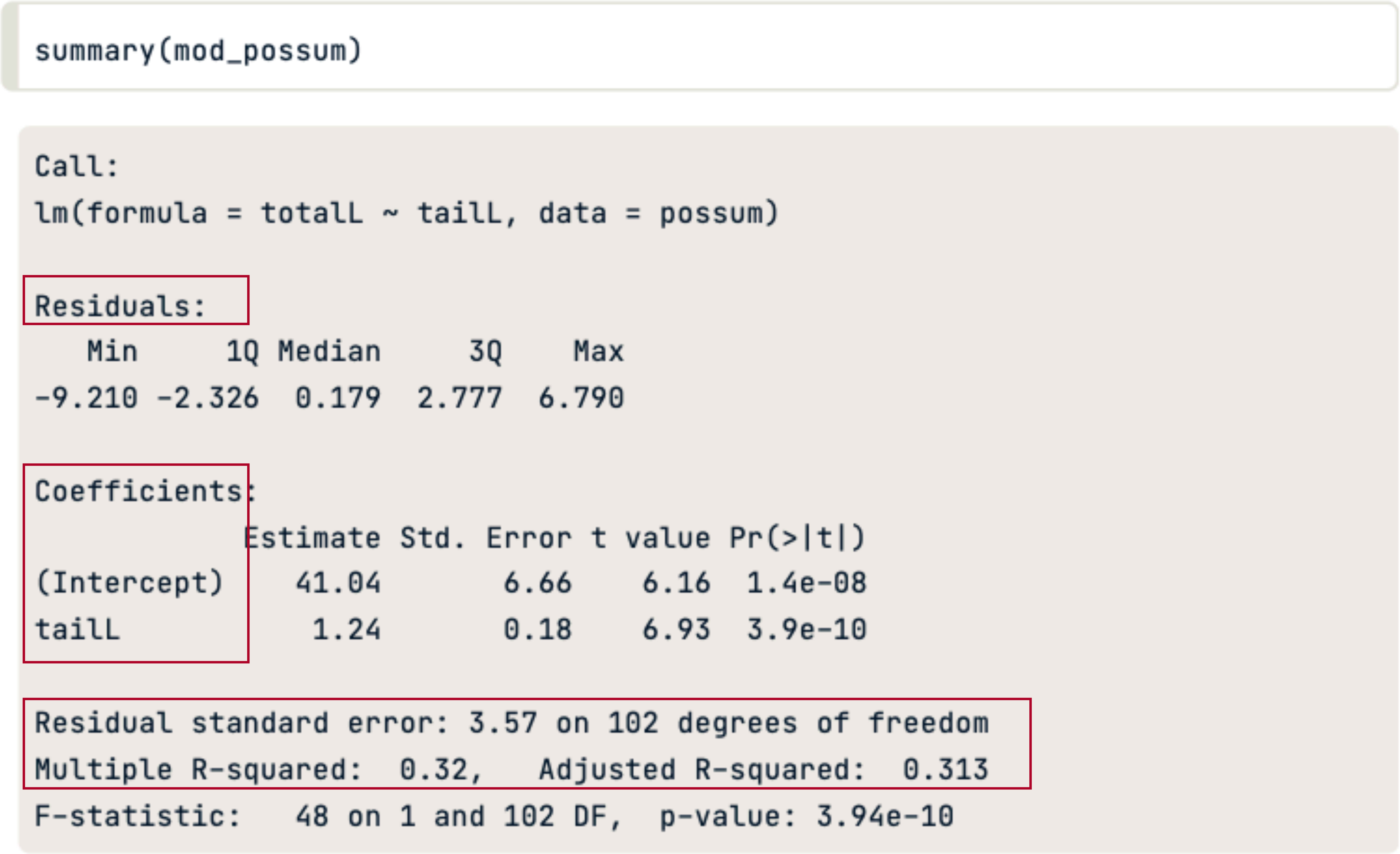
$$R^2_{adj} = 1 - \frac{SSE}{SST} \cdot \frac{n-1}{n-p-1},$$

Adjusted R2 penalizes a model for each additional explanatory variable (where *p* is the number of explanatory variables)

Revisiting summary

We have covered these!!

Still some left



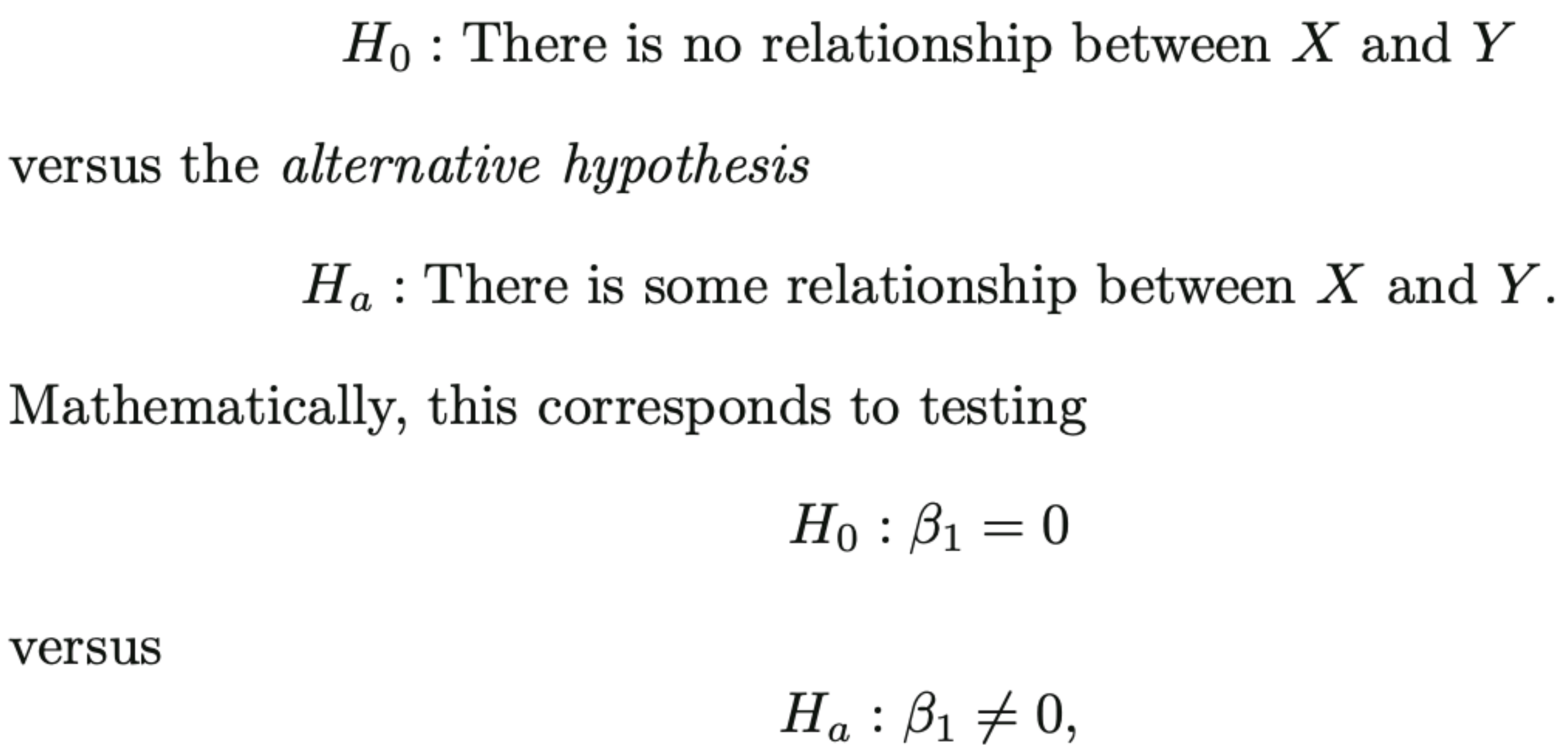
```
summary(mod_possum)
```

```
Call:
lm(formula = totalL ~ tailL, data = possum)

Residuals:
   Min     1Q Median     3Q    Max
-9.210 -2.326  0.179  2.777  6.790

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    41.04       6.66    6.16  1.4e-08
tailL           1.24       0.18    6.93  3.9e-10

Residual standard error: 3.57 on 102 degrees of freedom
Multiple R-squared:  0.32,   Adjusted R-squared:  0.313
F-statistic:   48 on 1 and 102 DF,  p-value: 3.94e-10
```

Coefficients in simple linear regression

t-statistic, hypothesis, p-value

Regression key question: Is there a relation between the response and the explanatory variable?



$$H_0 : \text{There is no relationship between } X \text{ and } Y$$

versus the *alternative hypothesis*

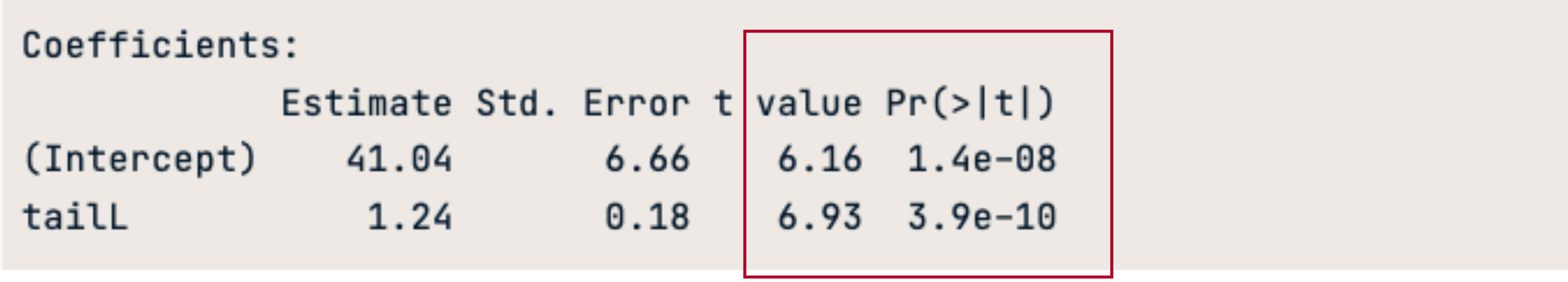
$$H_a : \text{There is some relationship between } X \text{ and } Y.$$

Mathematically, this corresponds to testing

$$H_0 : \beta_1 = 0$$

versus

$$H_a : \beta_1 \neq 0,$$



```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    41.04       6.66    6.16  1.4e-08
tailL           1.24       0.18    6.93  3.9e-10
```

we compute a t-statistic to determine whether we can reject H0. Check p-value

James et al. (2013). An *Introduction to Statistical Learning with Applications in R*, Chapter 3 Springer

Break time

Back at 11:47am

Lab - part 4

Assessing model fit

Multiple regression

Allows for multiple predictors and is an extension of a simple linear regression is multiple regression

Captures the relationship between the response variables and the p predictors ($p - 1$ predictors and the intercept)

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$$

The βi parameters are estimated using the same least squares approach

We interpret βi as the average effect on Y of a one unit increase in Xi, holding all other predictors fixed.

Multiple regression

Allows for multiple predictors and is an extension of a simple linear regression is multiple regression

Captures the relationship between the response variables and the p predictors ($p - 1$ predictors and the intercept)

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$$

Questions for multi-regression

- Is at least one of the predictors X1,X2,…,Xp useful in predicting the response?
- Do all the predictors help to explain Y, or is only a subset of the predictors useful?
- How well does the model fit the data?
- Given a set of predictor values, what response value should we predict, and how accurate is our prediction?

Regression model - *recall simple linear regression*

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot X$$

Expected value or estimate

$$e = Y - \hat{Y}$$

Difference between observed and expected value

Positive or negative residuals depending on observed and expected value

$$Y = \beta_0 + \beta_1 \cdot X + \epsilon \,, \qquad \epsilon \sim N(0, \sigma_\epsilon)$$

Actual observed value

Intercept

Slope

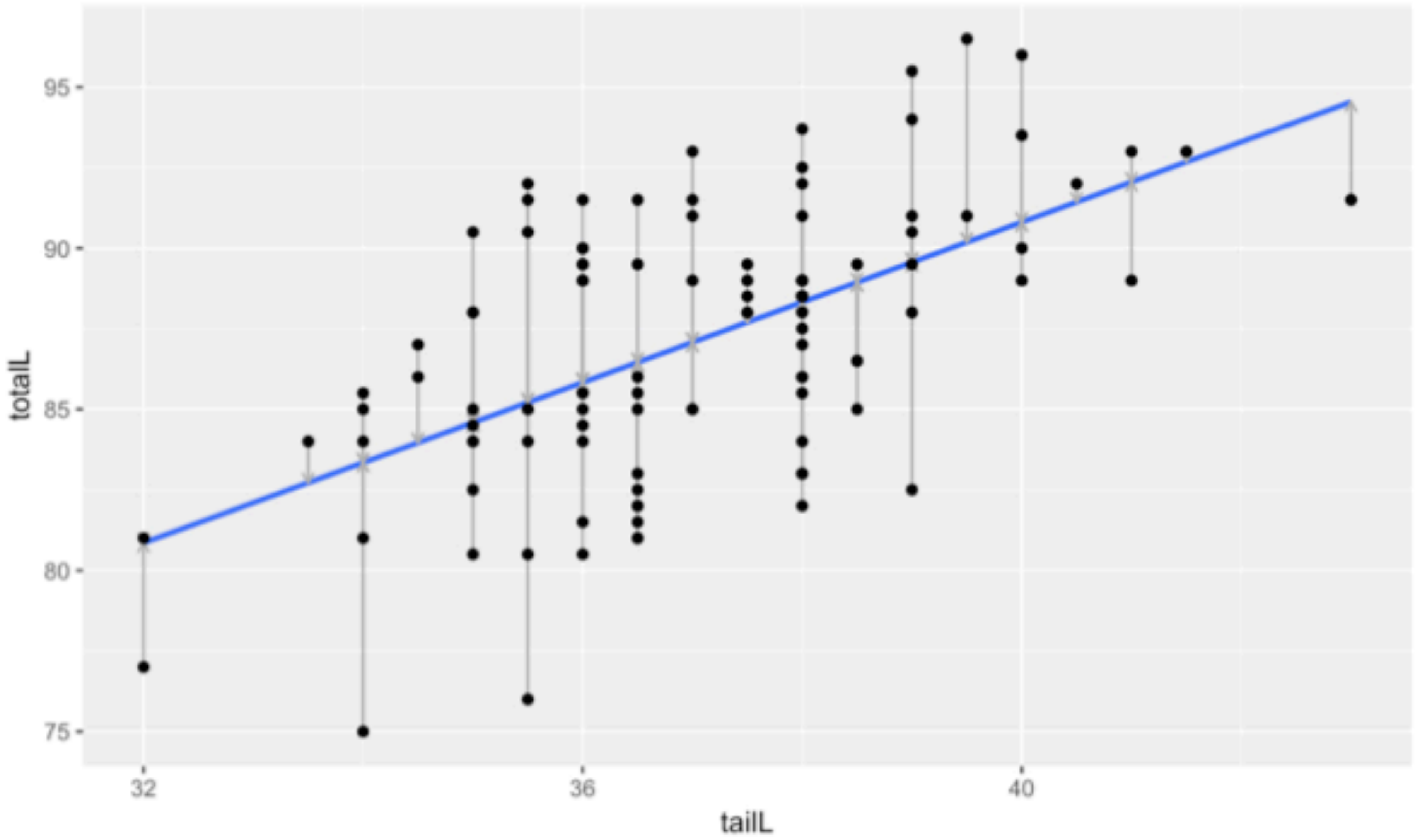
Noise

Residuals are the realization are the noise term. Epsilon is an unknown true quantity, while e is an estimate of that quantity.

Sum of squared deviations *same principle of least square distances hold*

Simple linear

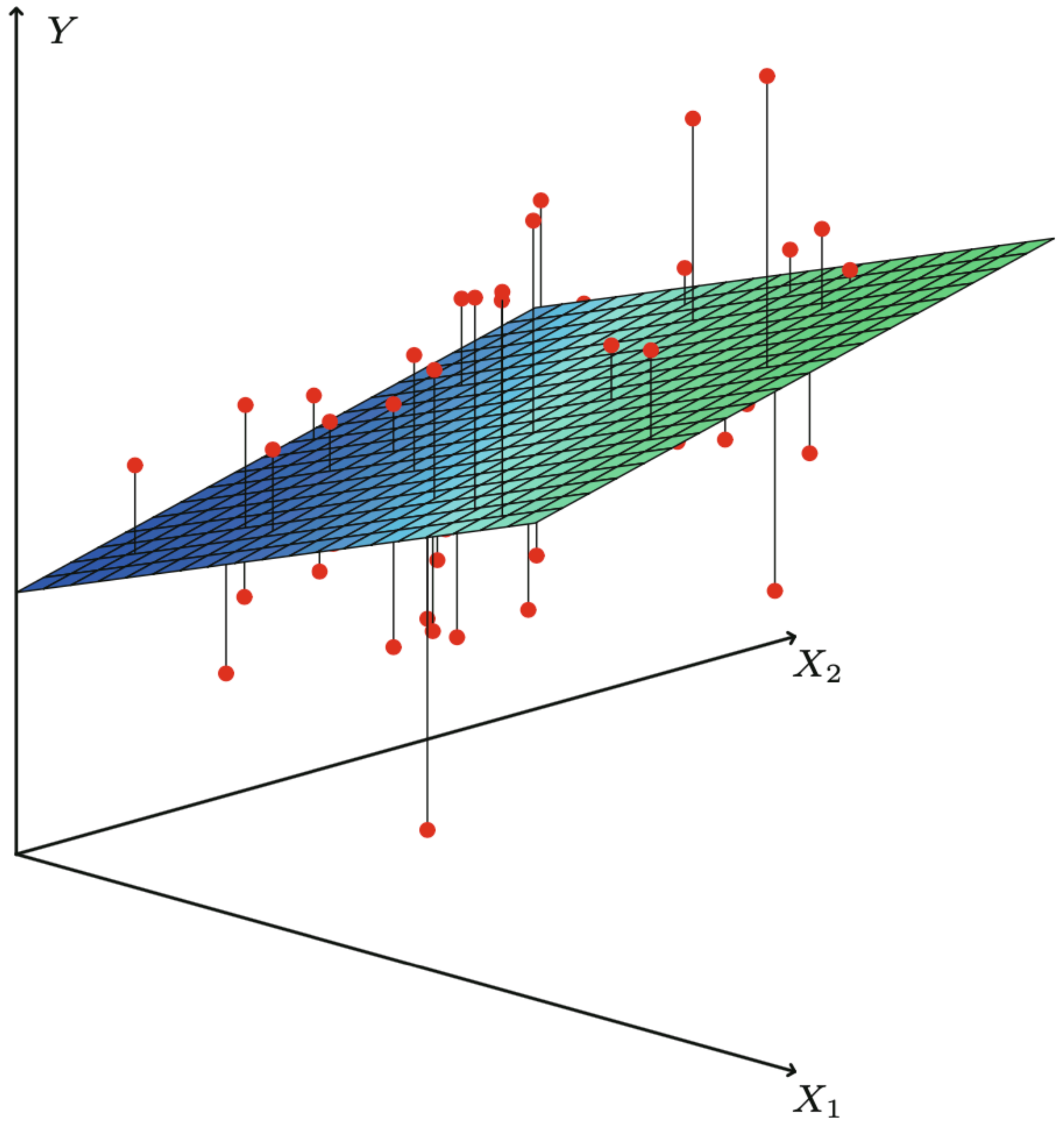
SS is the vertical distance between that point and the fitted line



To **minimize** SS, make the grey lines shorter, collectively and after squaring them

Multiple linear

With 2 predictors and 1 response, the least squares distance now becomes a plane.



The plan is chosen to **minimize** the SS vertical distances between each each point and the plane

Multiple linear regression: Model fit

Model fit concept carries forward from simple linear regression: **R-squared** vs. **adjusted R-squared**

Coefficient of determination

$$R^2 = 1 - \frac{SSE}{SST} ,$$

As SSE gets smaller, R2 gets higher (generally denoting a higher model fit)

Issue with R2 is that the SSE can only decrease as new variables are added to the model, while the SST does not change. So you can increase R2 by adding ***any*** additional variable—even random noise

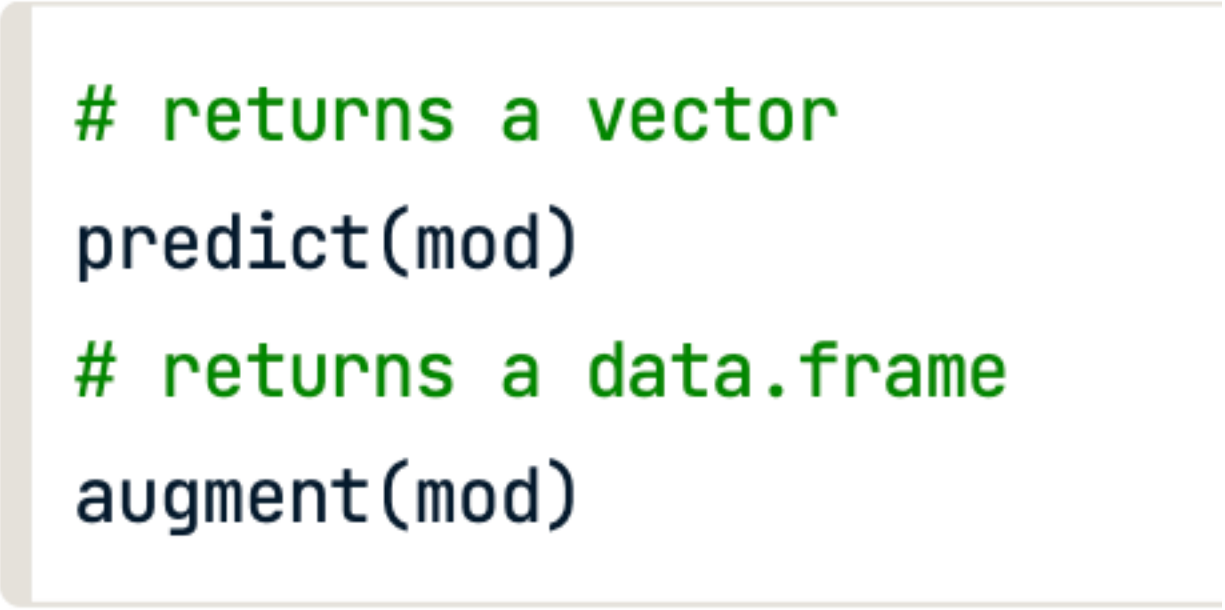
More of an issue in multiple linear regression.

$$R^2_{adj} = 1 - \frac{SSE}{SST} \cdot \frac{n-1}{n-p-1} ,$$

Adjusted R2 penalizes a model for each additional explanatory variable (where *p* is the number of explanatory variables)

Multiple regression: fitted values

Fitted values: same concept as simple regression



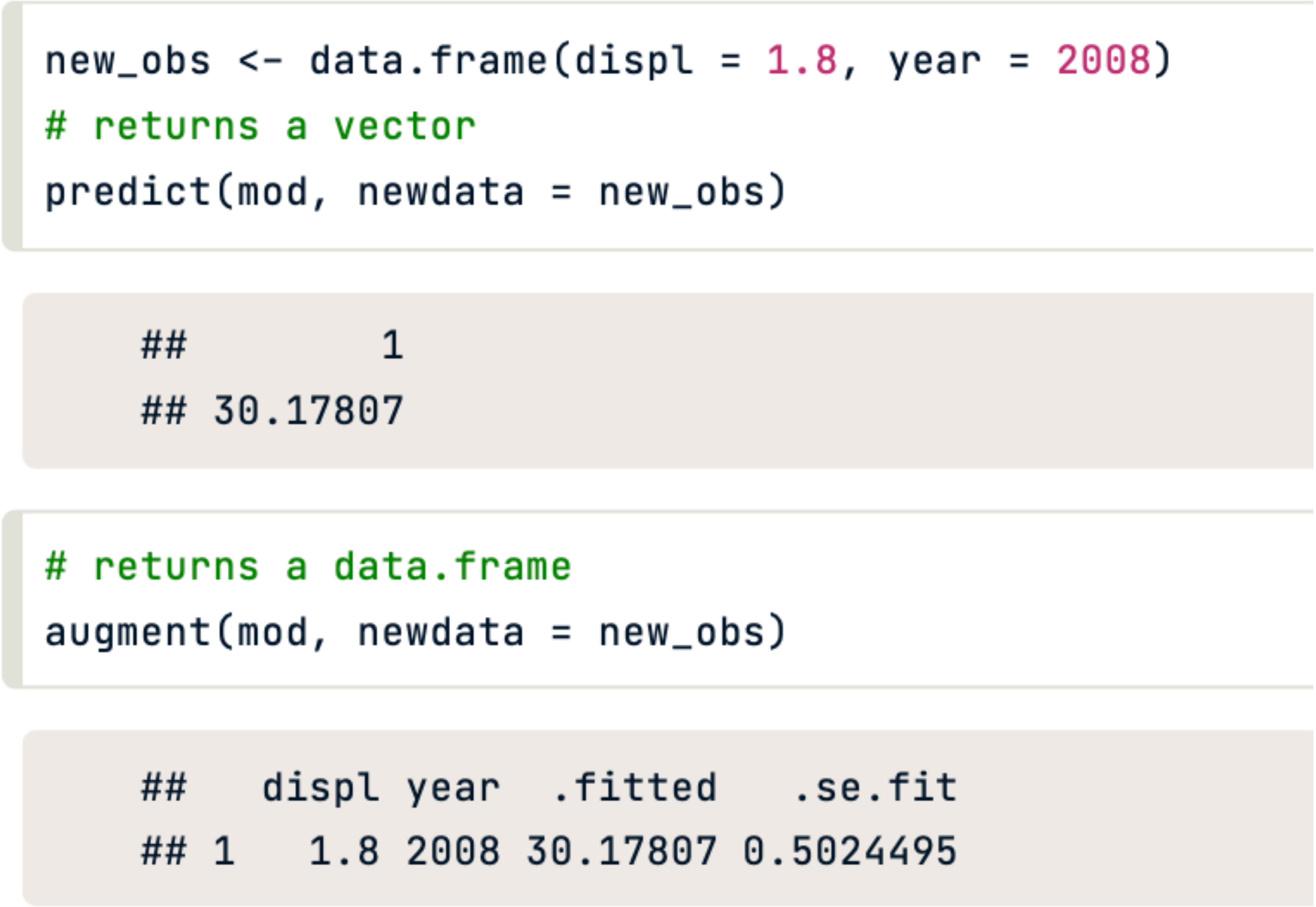
```
# returns a vector
predict(mod)
# returns a data.frame
augment(mod)
```

Two different ways

Multiple regression: Prediction

Prediction: same concept as simple regression. Making out-of-sample dataset

Example: Predicting the fuel economy of a car which is not there in the dataset.



```r
new_obs <- data.frame(displ = 1.8, year = 2008)
# returns a vector
predict(mod, newdata = new_obs)
```

```
##        1
## 30.17807
```

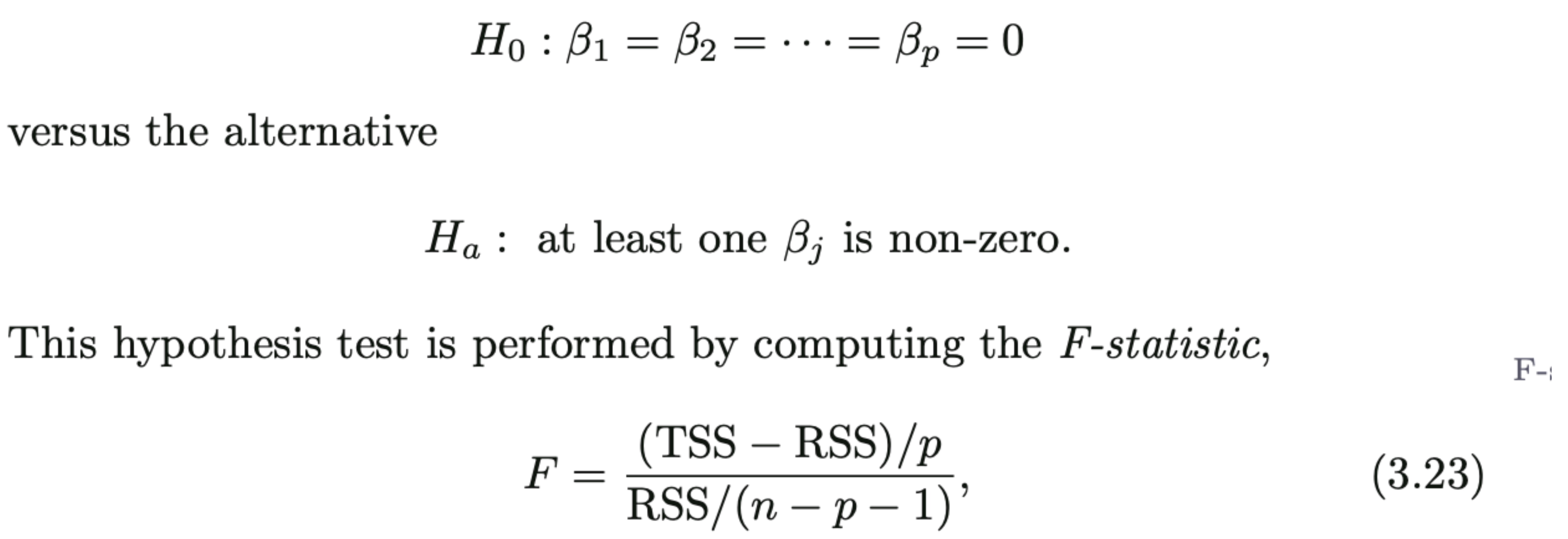
```r
# returns a data.frame
augment(mod, newdata = new_obs)
```

```
##   displ year  .fitted   .se.fit
## 1   1.8 2008 30.17807 0.5024495
```

Coefficients

F-statistic, hypothesis, p-value

Regression key question: Is there a relation between the response and explanatory variables?



$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$$

versus the alternative

$$H_a : \text{ at least one } \beta_j \text{ is non-zero.}$$

This hypothesis test is performed by computing the *F-statistic*,

F-

$$F = \frac{(\text{TSS} - \text{RSS})/p}{\text{RSS}/(n-p-1)}, \qquad (3.23)$$

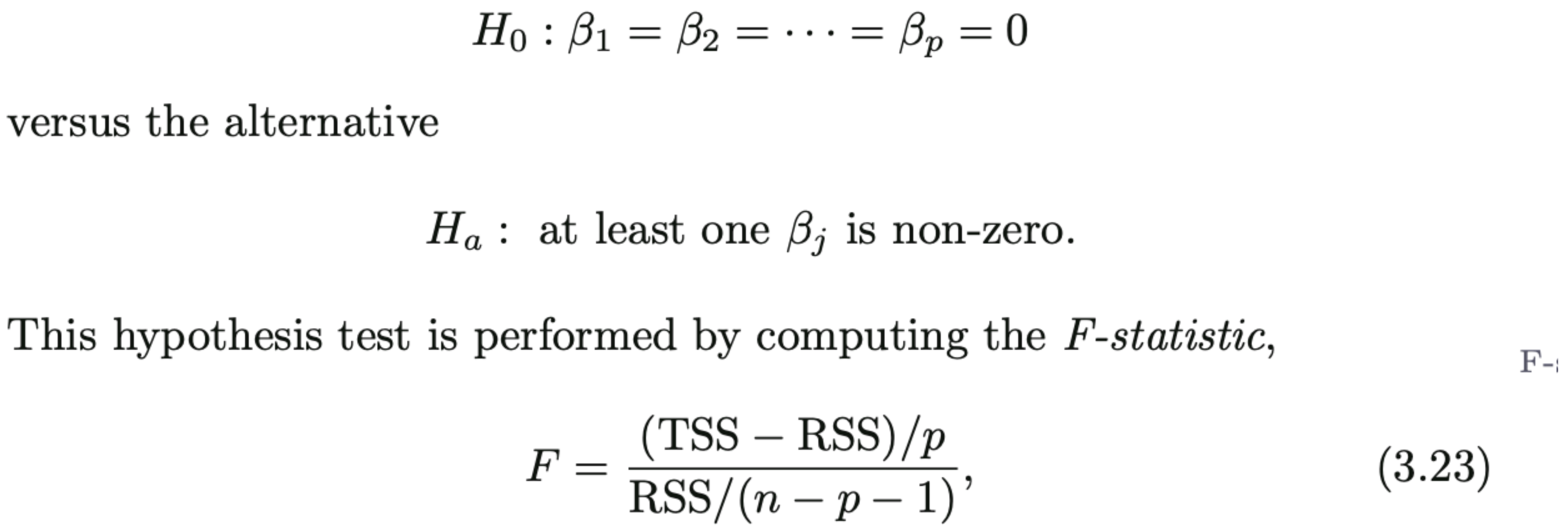
When there is no relationship between the response and predictors, one would expect the F-statistic to take on a value close to 1.

James et al. (2013). An *Introduction to Statistical Learning with Applications in R*, Chapter 3 Springer

Coefficients

F-statistic, hypothesis, p-value

Regression key question: Is there a relation between the response and explanatory variables?



$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$$

versus the alternative

$$H_a : \text{ at least one } \beta_j \text{ is non-zero.}$$

This hypothesis test is performed by computing the *F-statistic*,

F-

$$F = \frac{(\text{TSS} - \text{RSS})/p}{\text{RSS}/(n-p-1)}, \qquad (3.23)$$

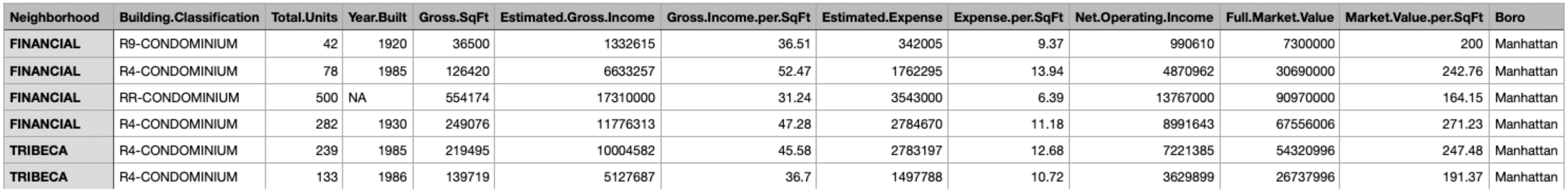
We interpret βi as the average effect on Y of a one unit increase in Xi, holding all other predictors fixed.

When interpreting coefficients, always include the phrase: **holding all other predictors fixed** or **controlling for all other predictors.**

James et al. (2013). An *Introduction to Statistical Learning with Applications in R*, Chapter 3 Springer

Multiple regression

Example: New York City condo evaluations data for fiscal year 2011–2012,



Neighborhood	Building.Classification	Total.Units	Year.Built	Gross.SqFt	Estimated.Gross.Income	Gross.Income.per.SqFt	Estimated.Expense	Expense.per.SqFt	Net.Operating.Income	Full.Market.Value	Market.Value.per.SqFt	Boro
FINANCIAL	R9-CONDOMINIUM	42	1920	36500	1332615	36.51	342005	9.37	990610	7300000	200	Manhattan
FINANCIAL	R4-CONDOMINIUM	78	1985	126420	6633257	52.47	1762295	13.94	4870962	30690000	242.76	Manhattan
FINANCIAL	RR-CONDOMINIUM	500	NA	554174	17310000	31.24	3543000	6.39	13767000	90970000	164.15	Manhattan
FINANCIAL	R4-CONDOMINIUM	282	1930	249076	11776313	47.28	2784670	11.18	8991643	67556006	271.23	Manhattan
TRIBECA	R4-CONDOMINIUM	239	1985	219495	10004582	45.58	2783197	12.68	7221385	54320996	247.48	Manhattan
TRIBECA	R4-CONDOMINIUM	133	1986	139719	5127687	36.7	1497788	10.72	3629899	26737996	191.37	Manhattan

Fitting a multiple regression model:

Multiple predictors separated on the right side of the formula using + signs

```
> house1 <- lm(ValuePerSqFt ~ Units + SqFt + Boro, data = housing)
> summary(house1)
```

Multiple regression

```
> house1 <- lm(ValuePerSqFt ~ Units + SqFt + Boro, data = housing)
> summary(house1)


Residuals:
     Min       1Q   Median       3Q      Max 
-168.458  -22.680    1.493   26.290  261.761 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)        4.430e+01  5.342e+00   8.293  < 2e-16 ***
Units             -1.532e-01  2.421e-02  -6.330 2.88e-10 ***
SqFt               2.070e-04  2.129e-05   9.723  < 2e-16 ***
BoroBrooklyn       3.258e+01  5.561e+00   5.858 5.28e-09 ***
BoroManhattan      1.274e+02  5.459e+00  23.343  < 2e-16 ***
BoroQueens         3.011e+01  5.711e+00   5.272 1.46e-07 ***
BoroStaten Island -7.114e+00  1.001e+01  -0.711    0.477    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 43.2 on 2613 degrees of freedom
Multiple R-squared:  0.6034, Adjusted R-squared:  0.6025 
F-statistic:  662.6 on 6 and 2613 DF,  p-value: < 2.2e-16
```

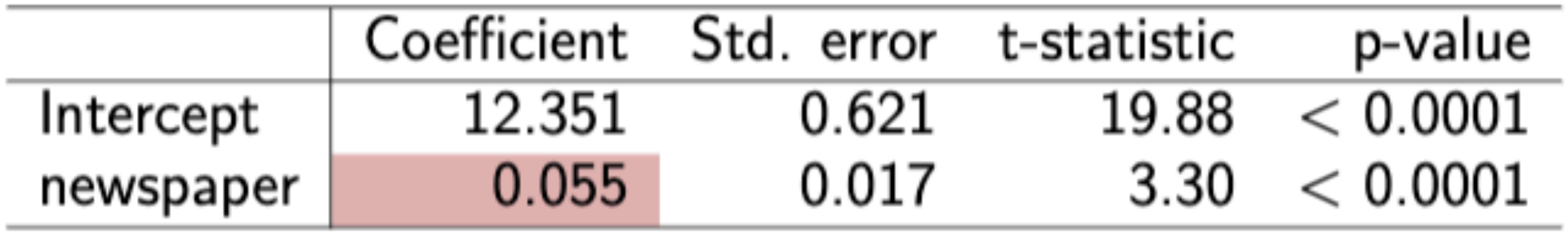
Summary shows (in addition to what you'd expect from simple linear reg), you see:

coefficient estimates, standard errors and p-values for **each variable**

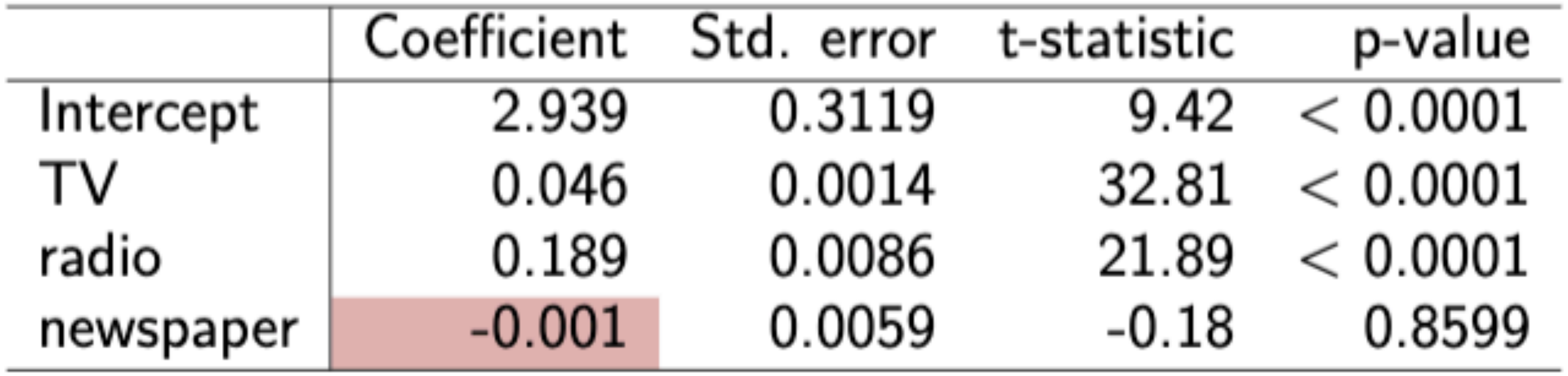
We interpret βi as the average effect on Y of a one unit increase in Xi, holding all other predictors fixed.

Simple vs. Multiple

Coefficients for the same parameter can be different



	Coefficient	Std. error	t-statistic	p-value
Intercept	12.351	0.621	19.88	< 0.0001
newspaper	0.055	0.017	3.30	< 0.0001



	Coefficient	Std. error	t-statistic	p-value
Intercept	2.939	0.3119	9.42	< 0.0001
TV	0.046	0.0014	32.81	< 0.0001
radio	0.189	0.0086	21.89	< 0.0001
newspaper	-0.001	0.0059	-0.18	0.8599

In the **simple regression** case, the slope term represents the average effect of a $1,000 increase in newspaper advertising, ignoring other predictors such as TV and radio.

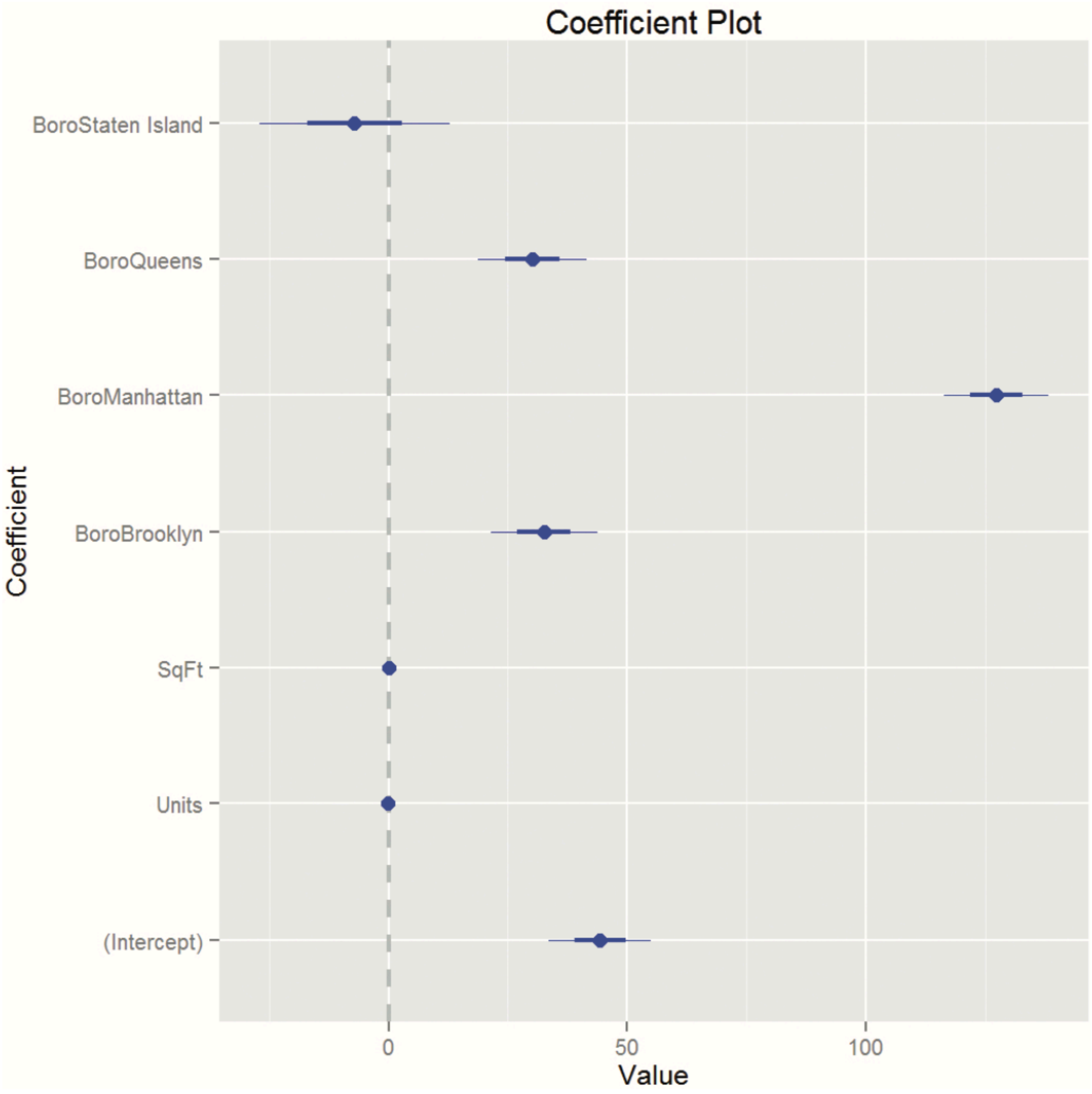
In the **multiple regression** setting, the coefficient for newspaper represents the average effect of increasing newspaper spending by $1,000 while holding TV and radio fixed.

Assessing Coefficients

Table format

```
> coef(house1)
        (Intercept)           Units           SqFt
       4.430325e+01   -1.532405e-01   2.069727e-04
       BoroBrooklyn   BoroManhattan     BoroQueens
       3.257554e+01    1.274259e+02   3.011000e+01
BoroStaten Island
      -7.113688e+00
```

Check Ladner book Ch. 16 for code



Visual format

Figure 16.9 Coefficient plot for condo value regression.

Lab - part 5

Multiple regression