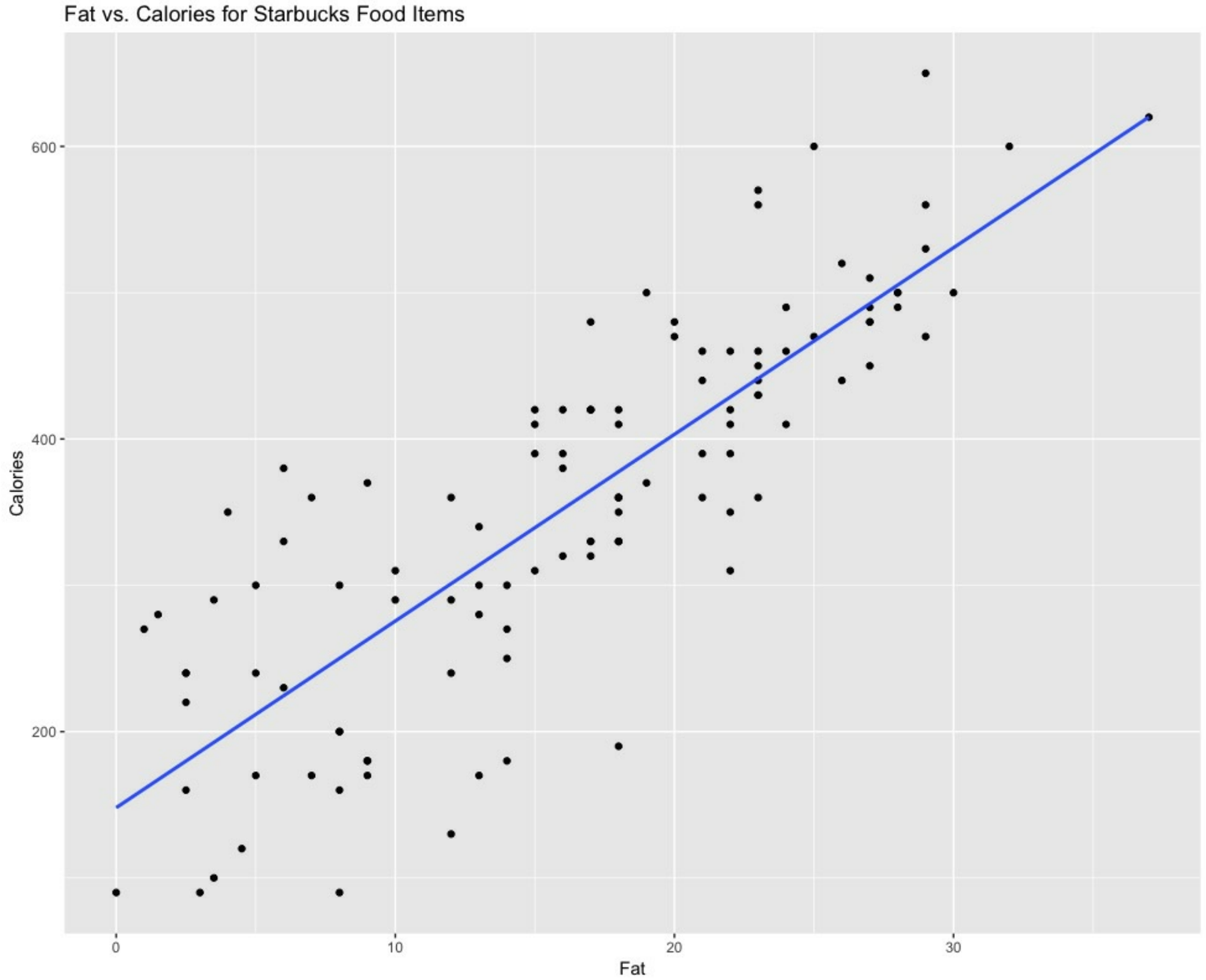
INFERENCE FOR LINEAR REGRESSION

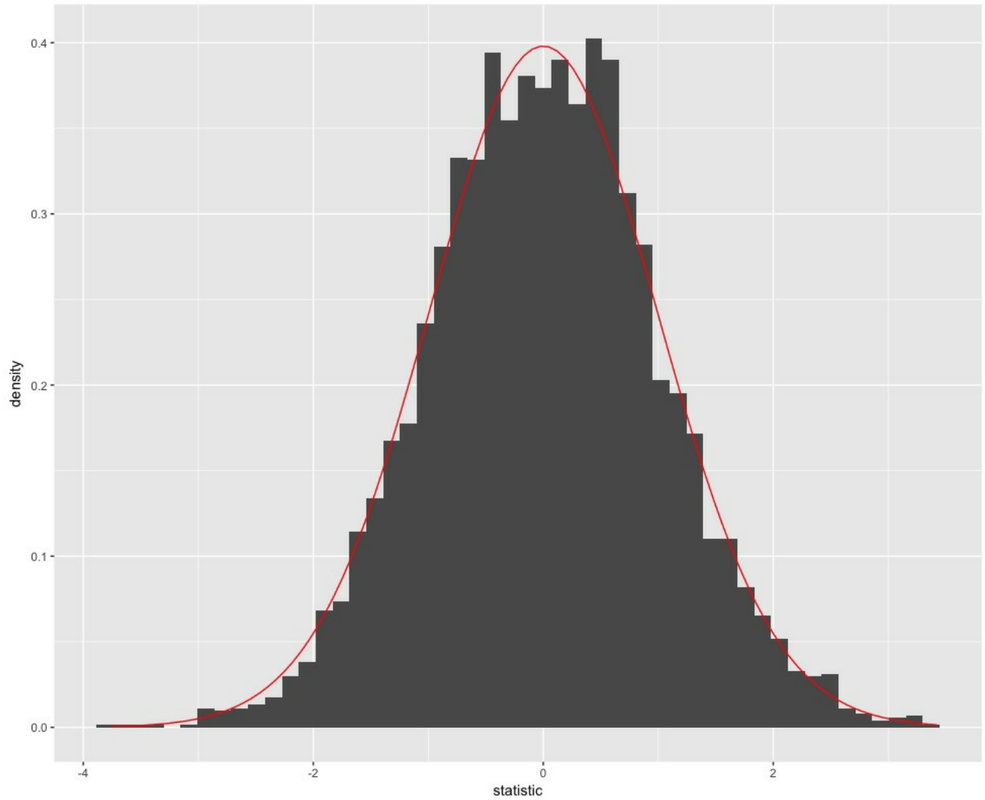
Mathematical approximation

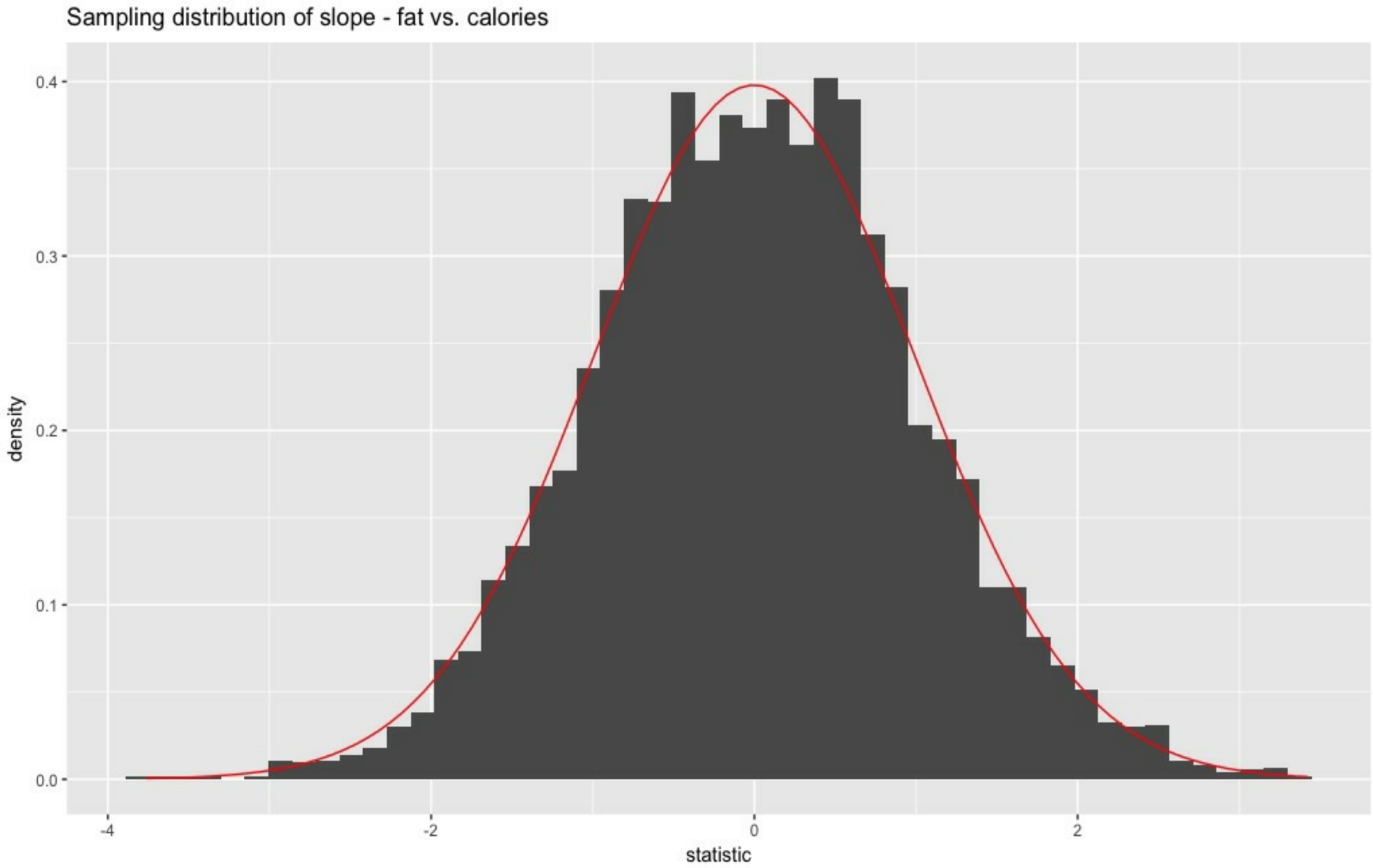
Jo Hardin
Professor, Pomona College

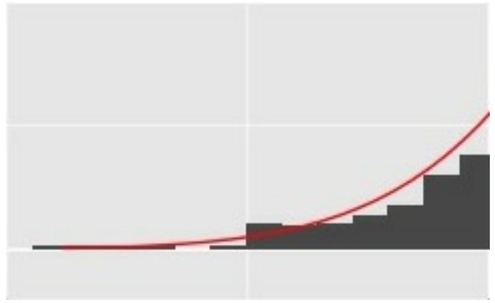

Fat vs. Calories for Starbucks Food Items
Calories
600
400
200
0
10
20
30
Fat

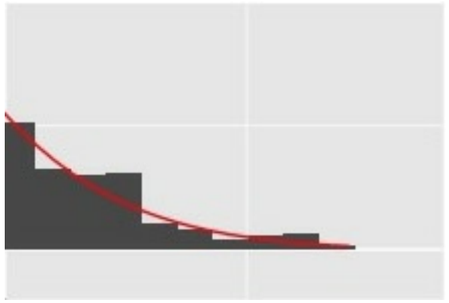
Sampling distribution of slope: good t fit

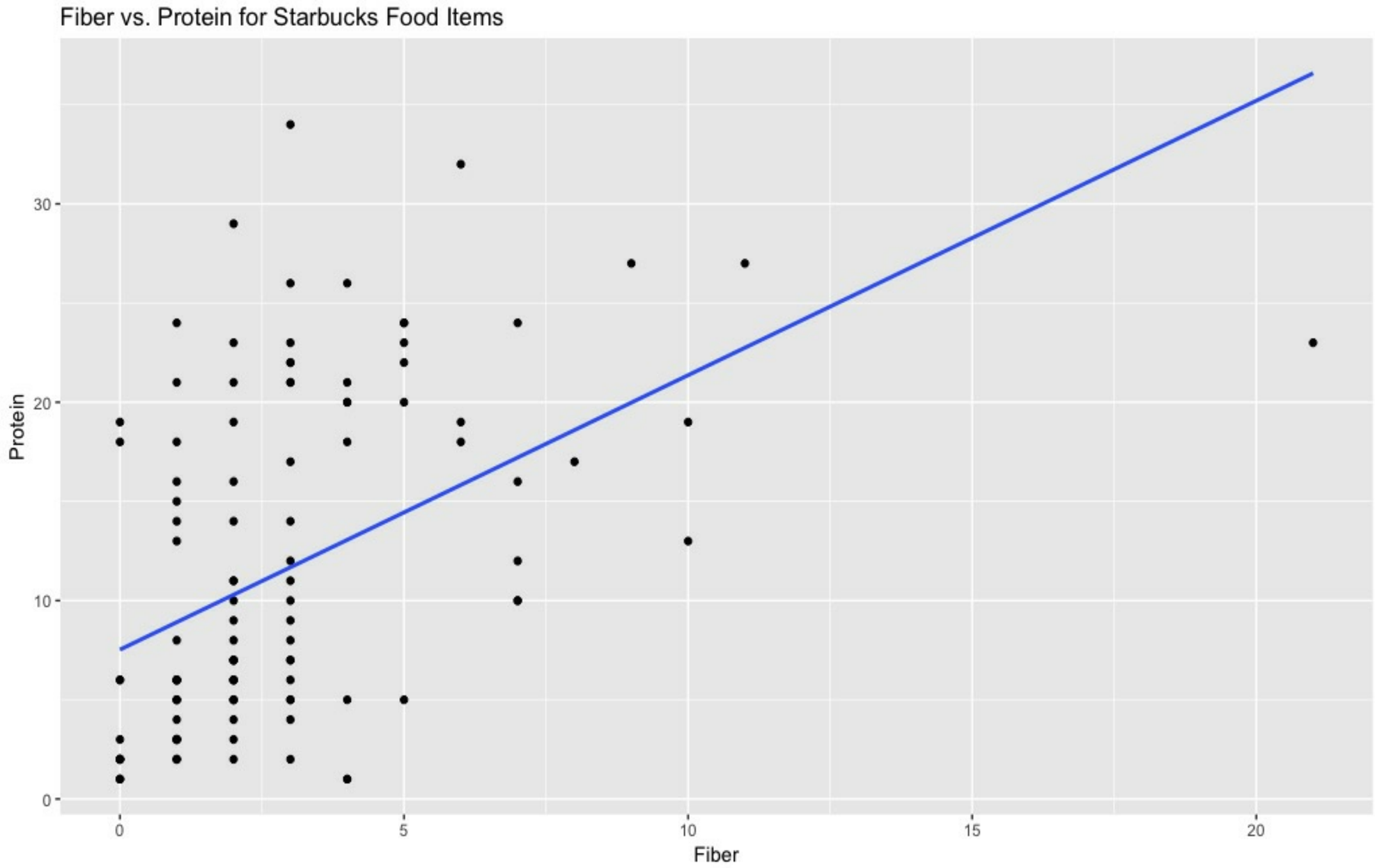
```
ggplot(starFatCal, aes(x = statistic)) +
  geom_histogram(aes(y = ..density..), bins = 50) +
  stat_function(fun = dt,
                color = "red",
                args = list(df = nrow(starbucks) - 2))
```




Sampling distribution of slope - fat vs. calories
density
0.4
0.3
0.2
0.1
0.0
-4
-2
0
2
statistic

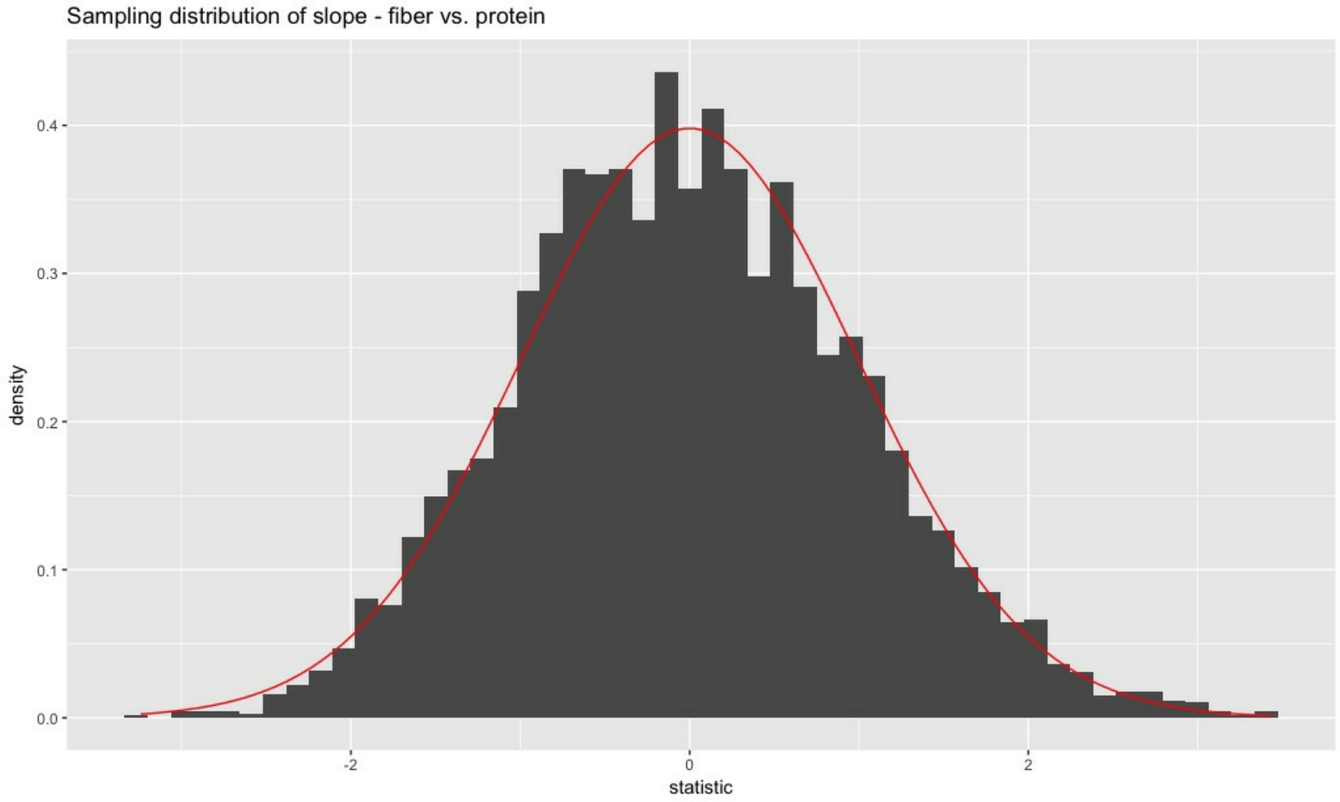




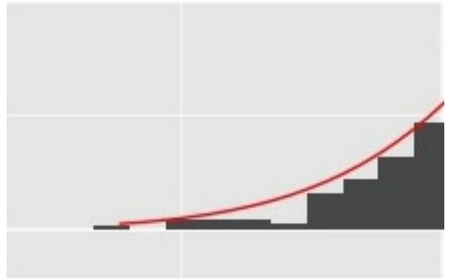

Fiber vs. Protein for Starbucks Food Items
Protein
0
10
20
30
0
5
10
15
20
Fiber

Sampling distribution of slope: poor t fit

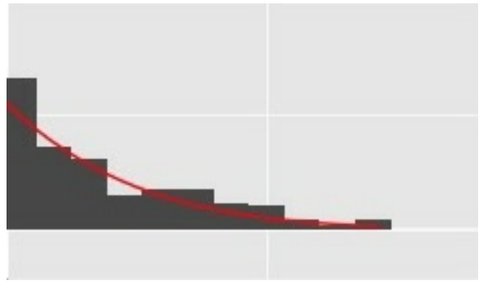
```
ggplot(starProFib, aes(x=statistic)) + geom_histogram(aes(y=..density..), bins=50
  stat_function(fun=dt, color="red", args=list(df=nrow(starbucks) - 2))
```



Sampling distribution of slope: poor t fit



Sampling distribution of slope: poor t fit



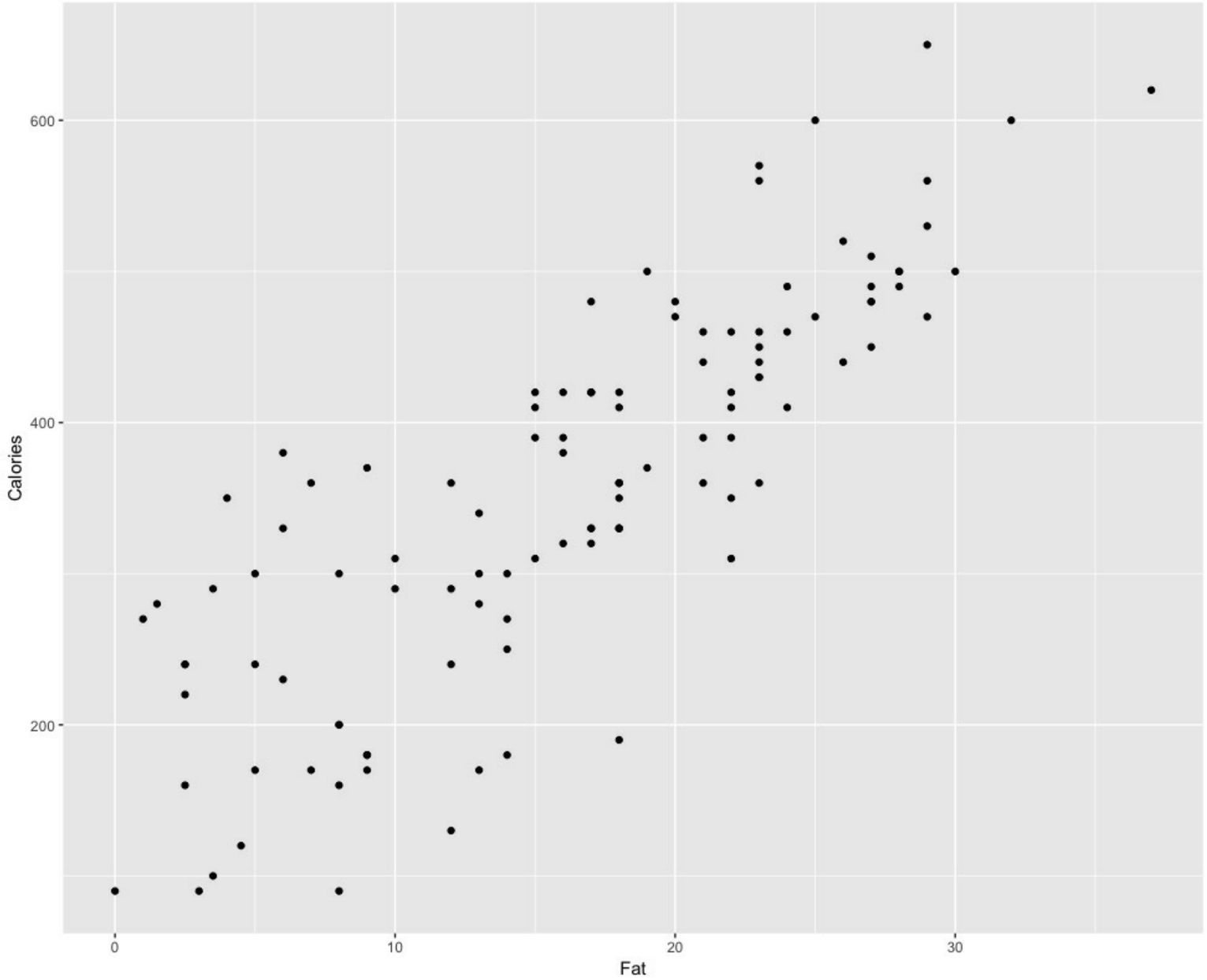
INFERENCE FOR LINEAR REGRESSION

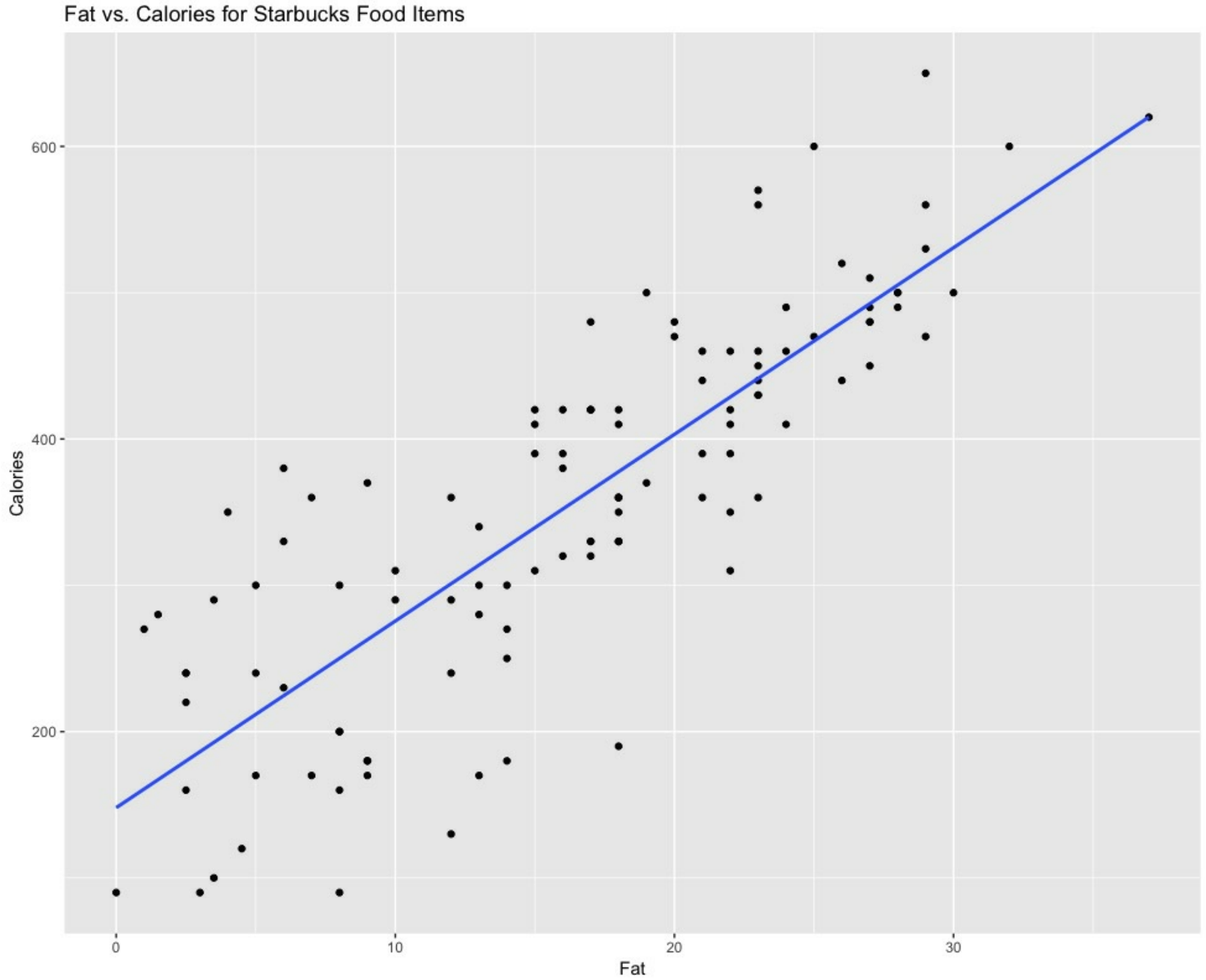
Let's practice!

INFERENCE FOR LINEAR REGRESSION

Intervals in regression

Jo Hardin
Professor, Pomona College


600
400
200
Calories
0
10
20
30
Fat


Fat vs. Calories for Starbucks Food Items
Calories
600
400
200
0
10
20
30
Fat

Confidence Interval for slope and intercept parameters

```
alpha = 0.05
crit_val <- qt((1-alpha/2), df = nrow(starbucks) - 2)

lm(Calories ~ Fat, data=starbucks) %>%
  tidy(conf.int = TRUE, conf.level=1-alpha)

# term estimate  std.error statistic      p.value
# 1 (Intercept) 147.9833 14.9719851  9.884013 6.630009e-17
# 2         Fat  12.7586  0.8171655 15.613236 8.937367e-30
# conf.low conf.high
# 1 118.31530 177.65128
# 2  11.13933  14.37787

lm(Calories ~ Fat, data=starbucks) %>% tidy() %>%
   mutate(lower = estimate - crit_val*std.error,
          upper = estimate + crit_val*std.error)

# term estimate  std.error statistic      p.value
# 1 (Intercept) 147.9833 14.9719851  9.884013 6.630009e-17
# 2         Fat  12.7586  0.8171655 15.613236 8.937367e-30
# lower     upper
# 1 118.31530 177.65128
# 2  11.13933  14.37787
```

Confidence Interval for intercept parameter

```
tidy_mod <- lm(Calories ~ Fat,
               data = starbucks) %>%
  tidy(conf.int = TRUE,
       conf.level=1-alpha)

tidy_mod

# term estimate  std.error
# 1 (Intercept) 147.9833 14.9719851
# 2         Fat  12.7586  0.8171655
# statistic      p.value  conf.low
# 1  9.884013 6.630009e-17 118.31530
# 2 15.613236 8.937367e-30  11.13933
# conf.high
# 1 177.65128
# 2  14.37787
```

```
tidy_mod %>%
  filter(term == "(Intercept)") %>%
  select(conf.low, conf.high)

#   conf.low conf.high
# 1 118.3153  177.6513
```

Confidence Interval for slope parameter

```
tidy_mod <- lm(Calories ~ Fat,
               data = starbucks) %>%
  tidy(conf.int = TRUE,
       conf.level=1-alpha)

tidy_mod

# term estimate  std.error
# 1 (Intercept) 147.9833 14.9719851
# 2         Fat  12.7586  0.8171655
# statistic      p.value  conf.low
# 1  9.884013 6.630009e-17 118.31530
# 2 15.613236 8.937367e-30  11.13933
# conf.high
# 1 177.65128
# 2  14.37787
```

```
tidy_mod %>%
  filter(term == "Fat") %>%
  select(conf.low, conf.high)

#   conf.low conf.high
# 1 11.13933  14.37787
```

Bootstrap interval for slope

```
BS_slope <- starbucks %>%
   specify(Calories ~ Fat) %>%
   generate(reps = 1000, type = "bootstrap") %>%
   calculate(stat = "slope")

BS_slope %>%
   summarize(low=quantile(stat, alpha / 2),
             high=quantile(stat, 1 - alpha / 2))

#  A tibble: 1 x 2
#        low     high
#      <dbl>    <dbl>
# 1 11.16712 14.34817
```

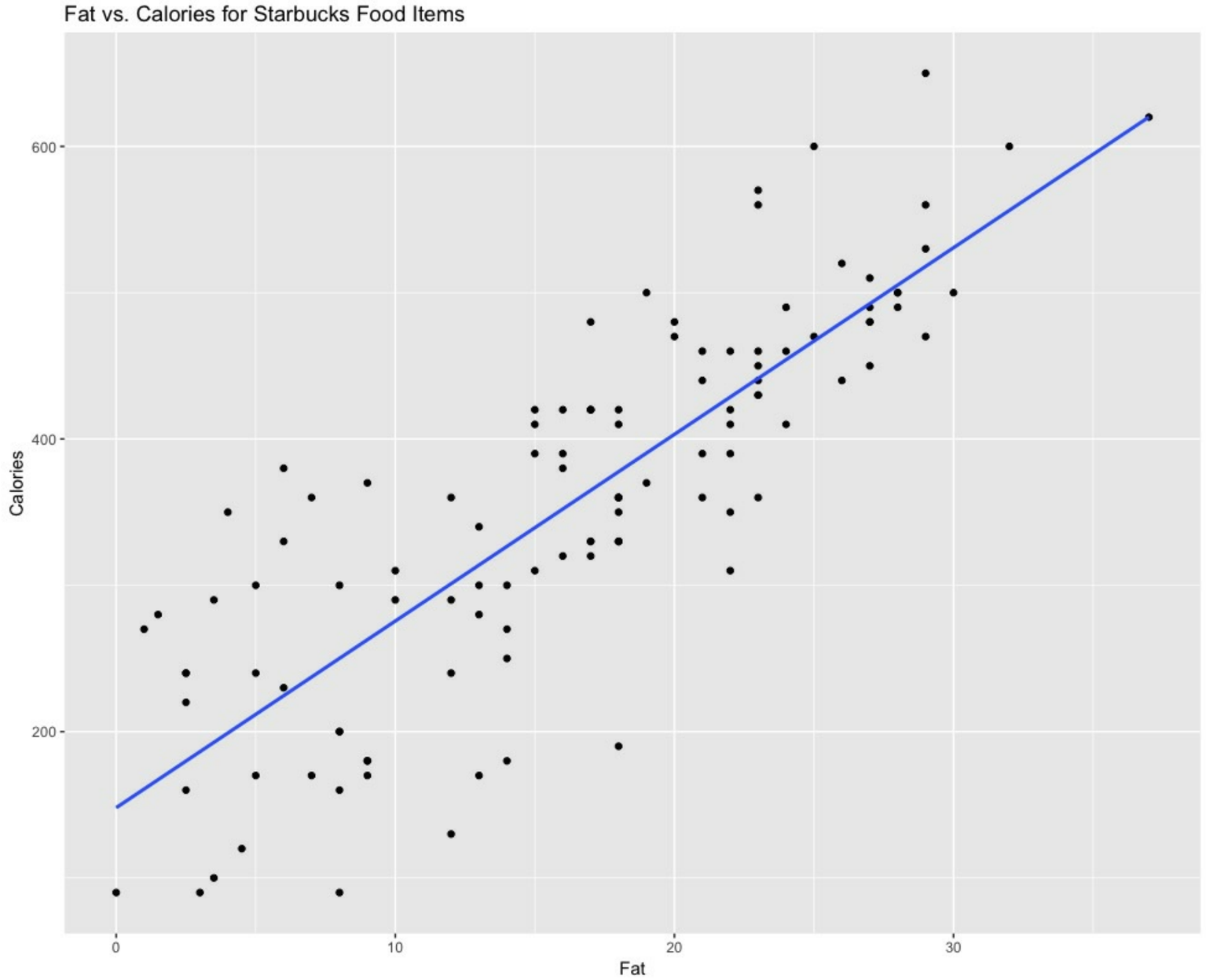
INFERENCE FOR LINEAR REGRESSION

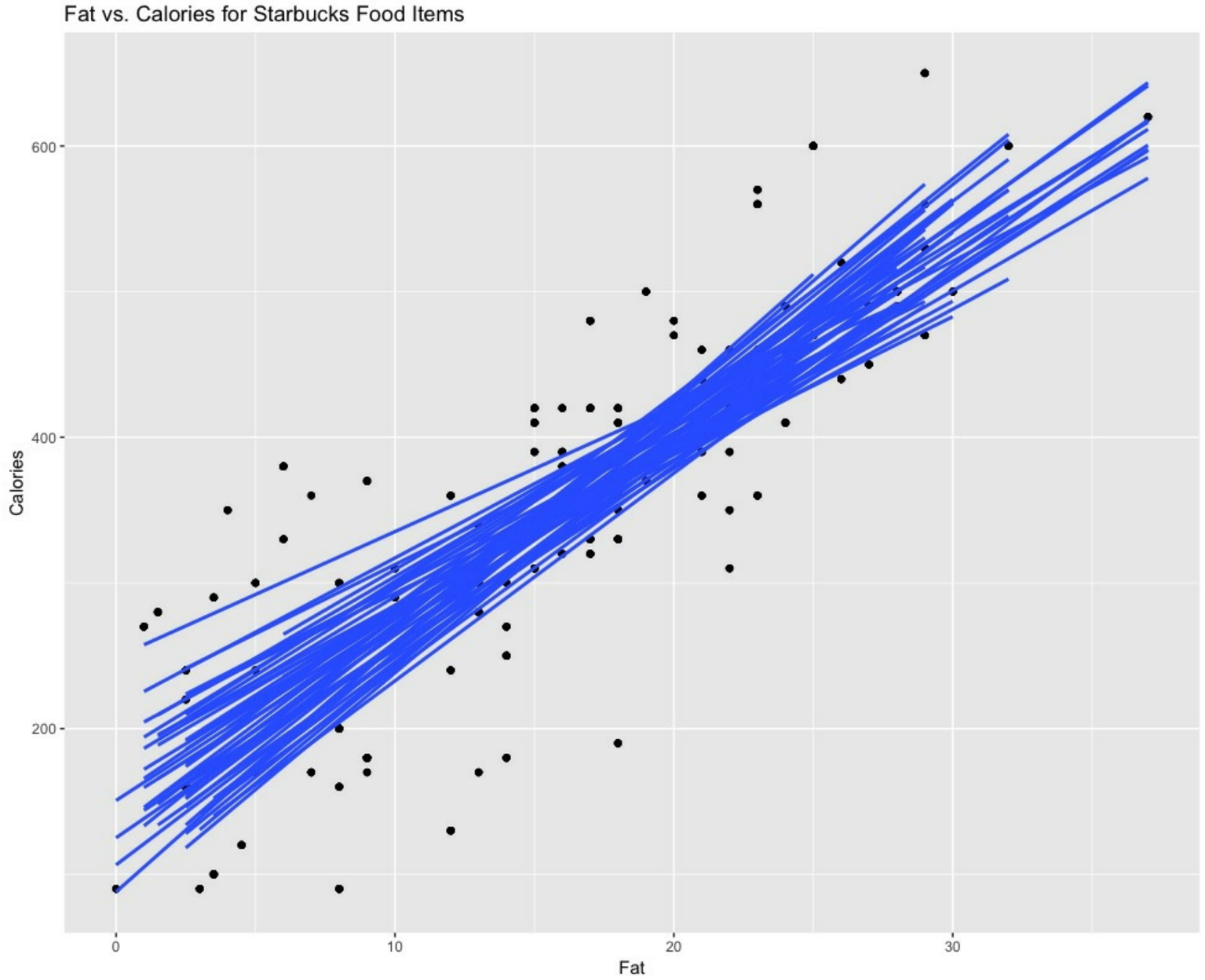
Let's practice!

INFERENCE FOR LINEAR REGRESSION

Different types of Intervals

Jo Hardin
Professor, Pomona College


Fat vs. Calories for Starbucks Food Items
Calories
600
400
200
0
10
20
30
Fat


Fat vs. Calories for Starbucks Food Items
Calories
600
400
200
0
10
20
30
Fat

Predicting average calories at specific fat

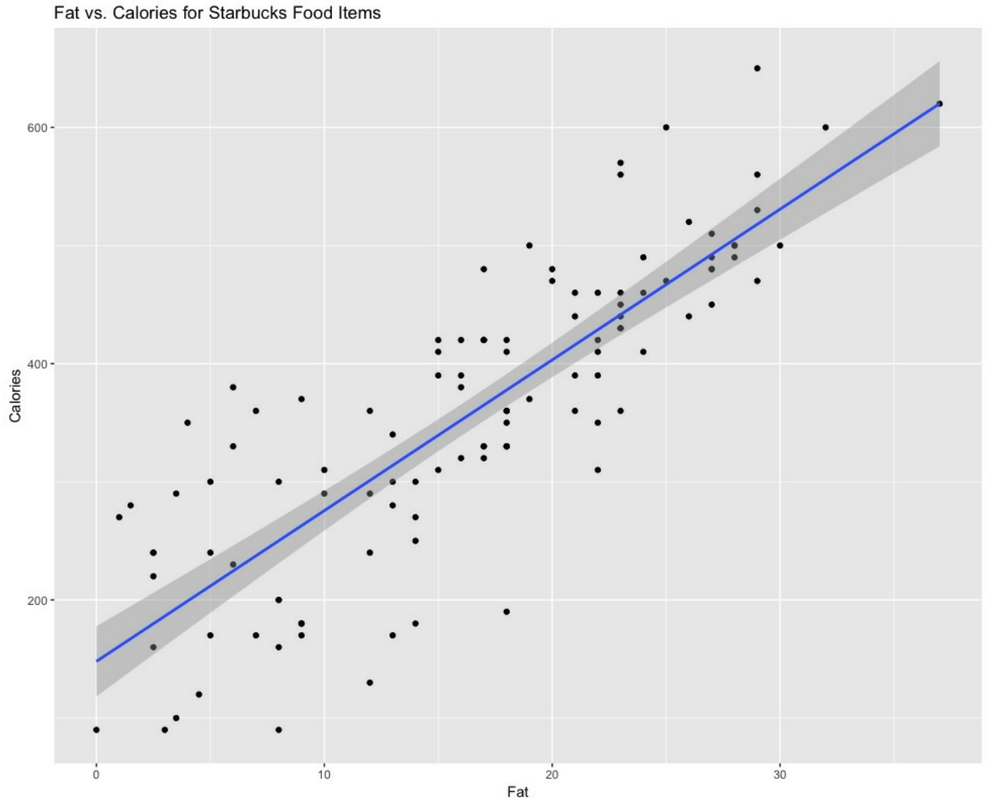
```
library(broom)
alpha <- .05
crit_val <- qt((1-alpha/2), df = nrow(starbucks) - 2)

newfood <- data.frame(Fat = c(0,10,20,30))

augment(lm(Calories ~ Fat, data=starbucks), newdata = newfood) %>%
   mutate(lowMean = .fitted - crit_val*.se.fit,
          upMean = .fitted + crit_val*.se.fit)

#   Fat  .fitted   .se.fit  lowMean   upMean
# 1   0 147.9833 14.971985 118.3153 177.6513

# 2  10 275.5693  8.516206 258.6938 292.4447

# 3  20 403.1552  7.378555 388.5341 417.7763

# 4  30 530.7412 13.035040 504.9114 556.5710
```

Creating CI for average response

```
predMeans <- augment(lm(Calories ~ Fat, data = starbucks)) %>%
   select(Calories, Fat, .fitted, .se.fit) %>%
   mutate(lowMean = .fitted - crit_val*.se.fit,
          upMean =  .fitted + crit_val*.se.fit)

head(predMeans)
#   Calories Fat  .fitted   .se.fit  lowMean   upMean
# 1      300   5 211.7763 11.473843 189.0401 234.5125
# 2      380   6 224.5349 10.823741 203.0869 245.9828
# 3      410  22 428.6724  8.176354 412.4704 444.8744
# 4      460  23 441.4310  8.663769 424.2632 458.5989
# 5      420  22 428.6724  8.176354 412.4704 444.8744
# 6      380  16 352.1209  6.756473 338.7324 365.5093
```

Plotting CI for average response

```
ggplot(predMeans, aes(x = Fat, y = Calories)) +
   geom_point() +
   stat_smooth(method = "lm", se = FALSE) +
   geom_ribbon(aes(ymin = lowMean, ymax = upMean), alpha=.2)
```

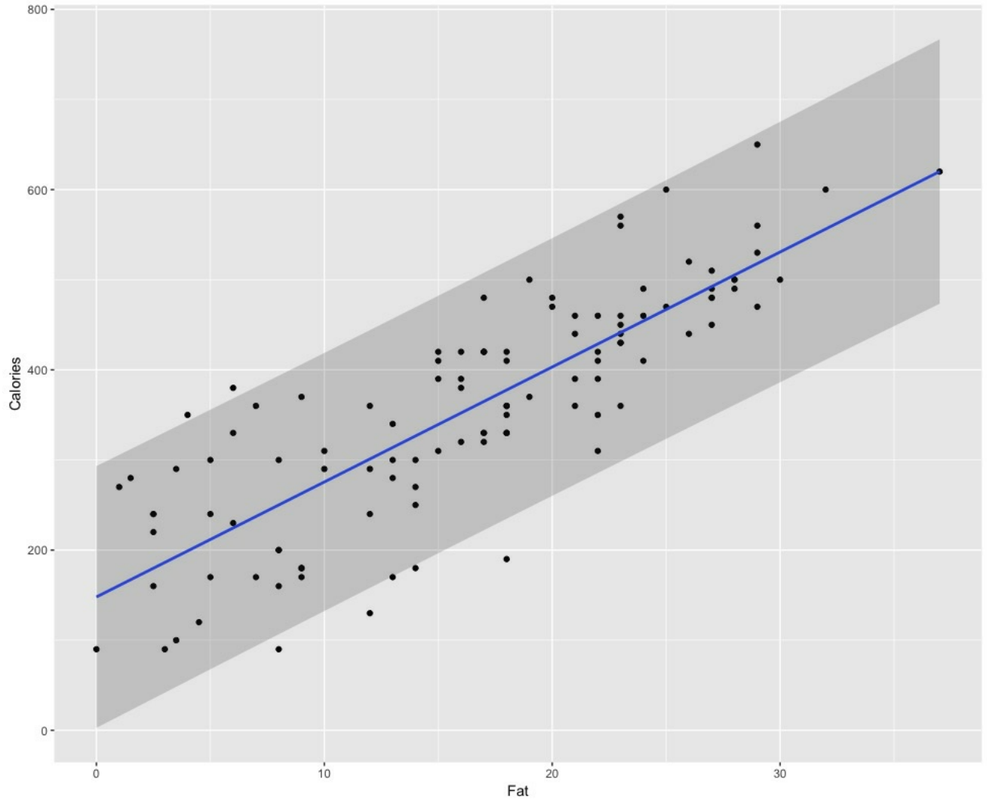


Prediction intervals

```
alpha <- .05
crit_val <- qt((1-alpha/2), df = nrow(twins) - 2)

FatCal_lm <- lm(Calories ~ Fat, data = starbucks)
FatCal_gl <- glance(FatCal_lm)
FatCal_sig <- pull(FatCal_gl, sigma)

FatCal_pred <- augment(FatCal_lm) %>%
  mutate(.se.pred = sqrt(FatCal_sig^2 + .se.fit^2))

predResp <- FatCal_pred %>%
  mutate(lowResp = .fitted - crit_val*.se.pred,
         upResp = .fitted + crit_val*.se.pred)

predResp
# A tibble: 113 x 12
#    Calories   Fat  .fitted   .se.fit     .resid        .hat   .sigma
#       <int> <dbl>    <dbl>     <dbl>      <dbl>       <dbl>    <dbl>
#  1      300     5 211.7763 11.473843  88.223722 0.025567957 71.57640
#  2      380     6 224.5349 10.823741 155.465125 0.022752704 70.50502
#  3      410    22 428.6724  8.176354 -18.672436 0.012983674 72.05959
# ... with 103 more rows, and 5 more variables: .cooksd <dbl>,
#   .std.resid <dbl>, .se.pred <dbl>, lowResp <dbl>, upResp <dbl>
```

Plotting prediction intervals

```
ggplot(predResp, aes(x = Fat, y = Calories)) +
  geom_point() +
  stat_smooth(method = "lm", se = FALSE) +
  geom_ribbon(aes(ymin = lowResp, ymax = upResp), alpha = .2)
```



INFERENCE FOR LINEAR REGRESSION

Let's practice!