Lab 5

Brandon Leslie

```r
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(fivethirtyeight)
```

```
Some larger datasets need to be installed separately, like senators and
house_district_forecast. To install these, we recommend you install the
fivethirtyeightdata package by running:
install.packages('fivethirtyeightdata', repos =
'https://fivethirtyeightdata.github.io/drat/', type = 'source')
```

```r
library(tidymodels)
```

```
-- Attaching packages -------------------------------------- tidymodels 1.2.0 --
v broom        1.0.7     v rsample      1.2.1
v dials        1.3.0     v tune         1.2.1
v infer        1.0.7     v workflows    1.1.4
v modeldata    1.4.0     v workflowsets 1.1.0
v parsnip      1.2.1     v yardstick    1.3.1
```

```
v recipes      1.1.0
-- Conflicts ----------------------------------------- tidymodels_conflicts() --
x scales::discard() masks purrr::discard()
x dplyr::filter()   masks stats::filter()
x recipes::fixed()  masks stringr::fixed()
x dplyr::lag()      masks stats::lag()
x yardstick::spec() masks readr::spec()
x recipes::step()   masks stats::step()
* Search for functions across packages at https://www.tidymodels.org/find/
```

Lab 5 - Regression Exercises 1

Show "manually" that $b_1 = r\frac{S_{yy}}{S_{xx}}$ using the movie_scores data set with linear regression of audience critics from the fivethirtyeight package.

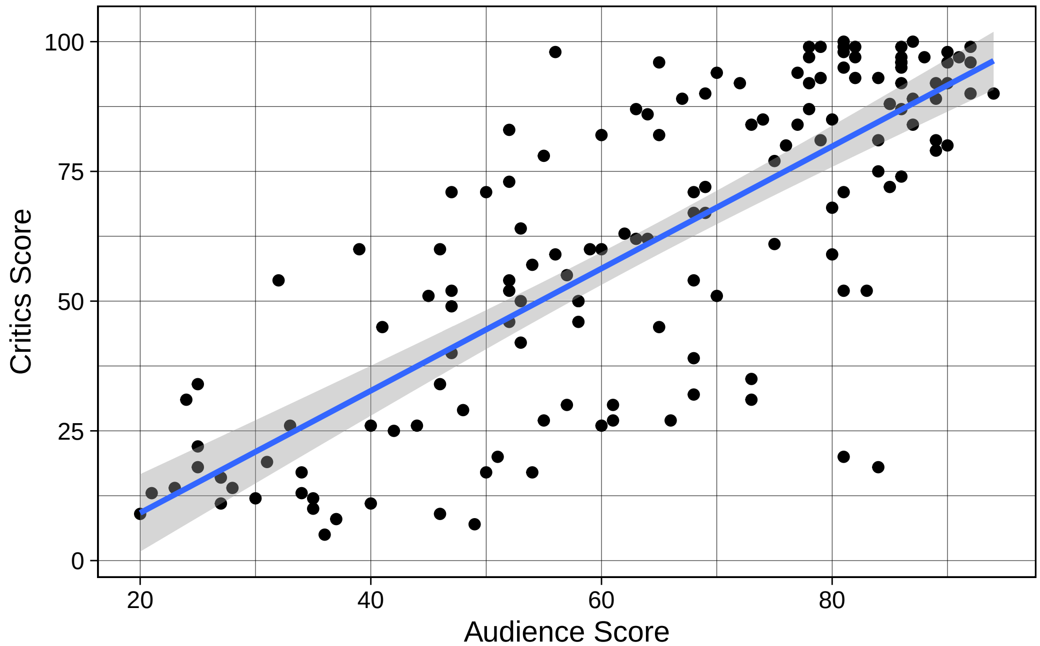
```
#
movie_scores <- fandango %>%
  group_by(rottentomatoes, rottentomatoes_user) %>%
  rename( audience = rottentomatoes_user,
             critics = rottentomatoes
             ) %>%
  select(audience, critics)

aud_crit <- linear_reg() %>%
  fit(audience ~ critics, data = movie_scores)
tidy(aud_crit)
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   32.3      2.34        13.8 4.03e-28
2 critics        0.519    0.0345      15.0 2.70e-31
```

```
movie_scores %>%
  ggplot(aes(x = audience, y = critics)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(
    x = "Audience Score",
```

```
    y = "Critics Score",
    Title = "Audience vs. Critics"
     ) + theme_linedraw()
```




```
cov_aud_crit <- cov(movie_scores$critics, movie_scores$audience)
var_aud_crit <- var(movie_scores$critics)
b1 <- (cov_aud_crit / var_aud_crit)
b1
```

```
[1] 0.5186777
```

Using the above example, show "manually" that $b_1 = \frac{S_{yy}}{S_{xx}}$. Here, S_{xy} is the "unnormalized" covariance of X and Y ; that is, just the numerator, without dividing by the product of the sample standard deviations of X and Y.

```
x1 <- movie_scores$critics
y1 <- movie_scores$audience
sxy <- sum((x1 - mean(x1)) * (y1 - mean(y1)))
sxx <- sum((x1 - mean(x1))^2)
b1.1 <- sxy / sxx
b1.1
```

```
[1] 0.5186777
```

What happens if you scale the predictors; that is subtracting the mean and dividing by standard deviation. What changes- coefficients, predicted values, R^2 values, p-values? Try using a data set of your choice, and scale manually using mutate to create new columns.

Orignal Dataset

```r
mod <- linear_reg() %>% fit(mpg ~ wt, data = mtcars)
  tidy(mod)
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)    37.3      1.88      19.9  8.24e-19
2 wt             -5.34     0.559     -9.56 1.29e-10
```

```r
preds <- predict(mod, new_data = mtcars)

results <- bind_cols(preds, mtcars %>% select(mpg))

results %>% rsq(truth = mpg, .pred)
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rsq     standard       0.753
```

```r
summary(results)
```

```
     .pred             mpg
 Min.   : 8.297   Min.   :10.40
 1st Qu.:17.992   1st Qu.:15.43
 Median :19.515   Median :19.20
 Mean   :20.091   Mean   :20.09
 3rd Qu.:23.490   3rd Qu.:22.80
 Max.   :29.199   Max.   :33.90
```

```r
glance(mod)
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
1     0.753         0.745  3.05      91.4 1.29e-10     1  -80.0  166.  170.
# i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Scaled dataset

```r
mtcars_mod <- mtcars %>%
  mutate(wt_scaled = (wt - mean(wt))/ sd(wt))
  #scaled

#When take the z-score, forcing mean = 0, SD = 1
mod2 <- linear_reg() %>% fit(mpg ~ wt_scaled, data = mtcars_mod)

tidy(mod2)
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)    20.1      0.538     37.3  1.06e-26
2 wt_scaled      -5.23     0.547     -9.56 1.29e-10
```

```r
preds2 <- predict(mod2, new_data = mtcars_mod)

results2 <- bind_cols(preds2, mtcars_mod %>% select(mpg))

results2 %>%  rsq(truth = mpg, .pred)
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rsq     standard       0.753
```

```r
summary(results2)
```

```
     .pred             mpg       
 Min.   : 8.297   Min.   :10.40  
 1st Qu.:17.992   1st Qu.:15.43  
 Median :19.515   Median :19.20  
 Mean   :20.091   Mean   :20.09  
 3rd Qu.:23.490   3rd Qu.:22.80  
 Max.   :29.199   Max.   :33.90  
```

```r
glance(mod2)
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
1     0.753         0.745  3.05      91.4 1.29e-10     1  -80.0  166.  170.
# i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

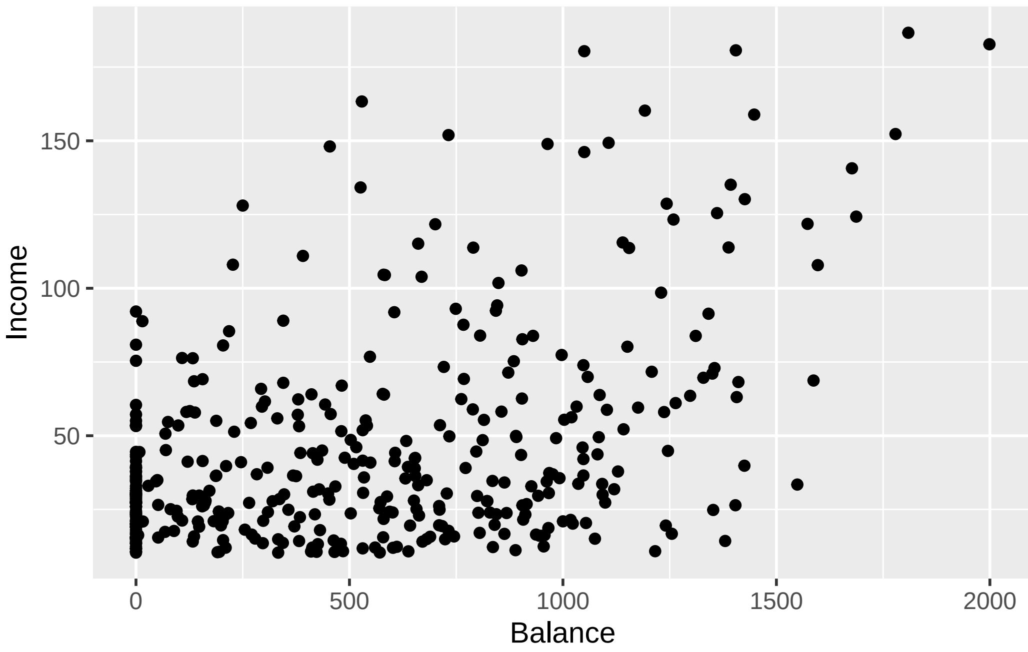
When you scale the predictors to a linear regression model, little changes. While the **coefficients** change, other important statistics like the r^2 value, **P-Value**, and **predicted values**, stay the same. In the above example, we used the mpg data-set and kept two versions: A control version, and a manipulated one. The manipulated one had a slight change in coefficients, like the intercept and the wt/wt_scaled variables. However, every other aspect is roughly the same. Summary statistics like standard error, mean, quartile(s), are the same.

With the Credit data in ISLR package, complete the following and comment on the results.

```r
library(ISLR)
```

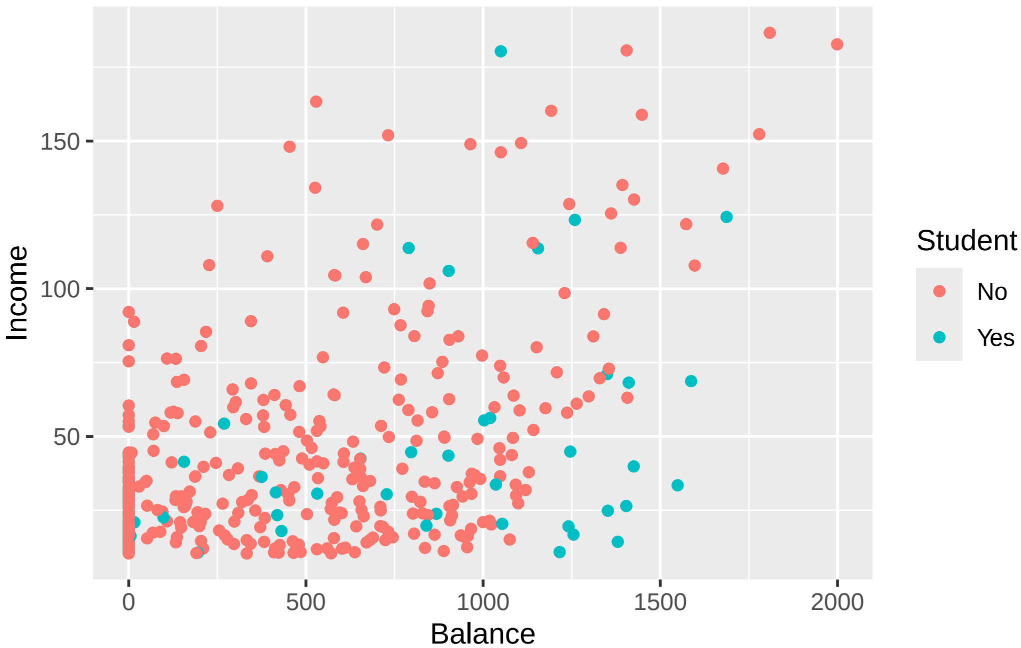
Plot a scatterplot of Balance vs Income

```r
Credit %>%
  ggplot(aes(x = Balance, y = Income)) +
  geom_point()
```



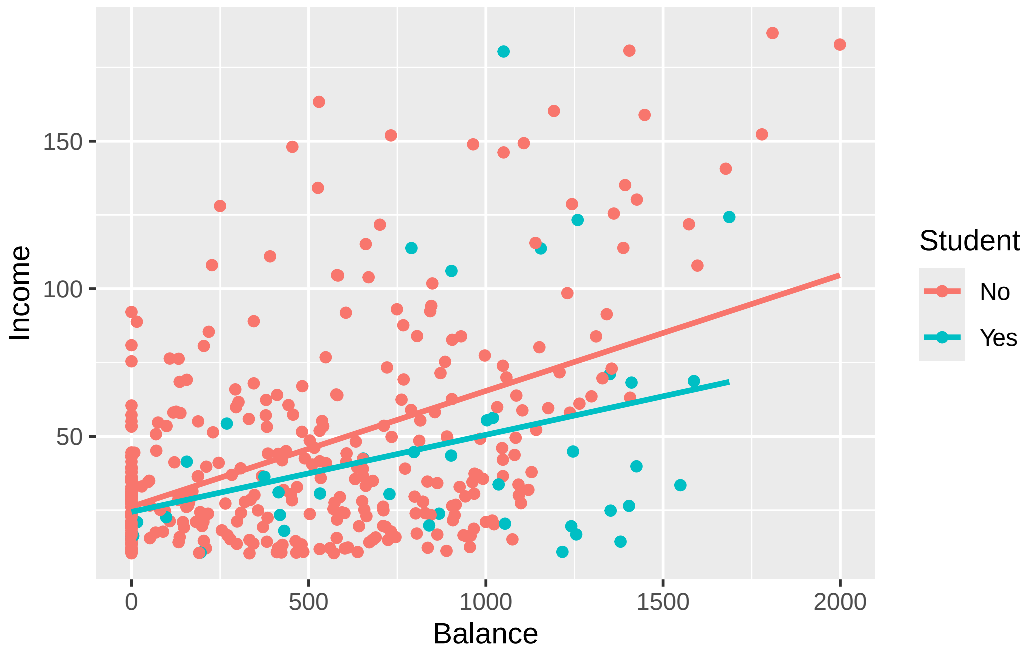
Now use color to indicating Student

```
Credit %>%
  ggplot(aes(x = Balance, y = Income)) +
  geom_point(aes(color = Student))
```



Add the regression line separately for students and non students. (ggplot should do this automatically.) What can you say about the slopes and intercepts?

```
Credit %>%
  ggplot(aes(x = Balance, y = Income, color = Student)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, )
```



The slopes for both lines are both increasing, however, the Student status of *"no"* has a steeper slope then the student status of *"yes"*. For the intercepts, they're roughly the same.

Now fit the following models, and interpret. Which corresponds to your scatterplot? Which gives an adjustment on the intercept for Student ? An adjustment on the slope for Student ? An adjustment on both?

```
bimod1 <- linear_reg() %>%
  fit(Balance ~ Income, data = Credit)
tidy(bimod1)
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   247.      33.2        7.43 6.90e-13
2 Income          6.05     0.579     10.4  1.03e-22
```

```
bimod2 <- linear_reg() %>%
  fit(Balance ~ Income + Student, data = Credit)
tidy(bimod2)
```

```
# A tibble: 3 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   211.      32.5        6.51 2.34e-10
2 Income          5.98     0.557     10.8  7.82e-24
3 StudentYes    383.      65.3        5.86 9.78e- 9
```

```
bimod3 <- linear_reg() %>%
  fit(Balance ~ Income + Student + Student:Income, data = Credit)
tidy(bimod3)
```

```
# A tibble: 4 x 5
  term              estimate std.error statistic  p.value
  <chr>                <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)         201.      33.7        5.95 5.79e- 9
2 Income                6.22     0.592     10.5  6.34e-23
3 StudentYes          477.     104.         4.57 6.59e- 6
4 Income:StudentYes    -2.00     1.73      -1.15 2.49e- 1
```

- The model that represents the scatter plot is **bimod1**.
- The model that gives an adjustment on the intercept for Student is **bimod2**.
- The model that gives an adjustment on the slope & intercept for student is **bimod3**.

With the Credit data, we might want to simply compare the mean Balance for students and non students. On the other hand, we might study the relationship by fitting the linear model Balance ~ Student . Compare the results of these two types of comparisons

Mean Balance for Students & non-students

```
Mean.Bal <- Credit %>%
  group_by(Student) %>%
  summarize( "Mean Balance" = mean(Balance, na.rm = TRUE))
Mean.Bal
```

```
# A tibble: 2 x 2
  Student `Mean Balance`
  <fct>            <dbl>
1 No                480.
2 Yes               877.
```

Linear Model

```r
Mean.Bal.lm <- linear_reg() %>%
  fit(Balance ~ Student, data = Credit)
tidy(Mean.Bal.lm)
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)     480.      23.4     20.5  2.90e-64
2 StudentYes      396.      74.1      5.35 1.49e- 7
```

Comparison

For the basic comparison of means, we can see that the mean *balance* for students (Yes) is higher then those who aren't students (No). However for the linear model, we can see that both the Intercept (No), and StudentYes (Yes), have extremely low p-values. This mean that they both are statistically significant. But what is interesting is that the Intercept has a higher estimate then StudentYes, with more "valuable" statistics then it's counterpart, StudentYes. the intercept has a lower p-value, higher Statistic, and lower standard error. this indicates that there isn't enough data compared to it's counterpart, which seems to be true. the intercept (StudentNo), makes up 90% of observances in the dataset, while StudentYes makes up 10%.

Show by example using any data set you like that a prediction interval is larger than a confidence interval. Intuitively, this is because a CI models the uncertainty in the mean for a given set of predictors; that is, the average over the population that has those same predictors. On the other hand, once you have the mean there is the additional uncertainty of an individual observation. This would remain even if there were no uncertainty in the mean.

Confidence Interval

```r
  t.test(Income ~ Student, data = Credit)
```

```

	Welch Two Sample t-test

data:  Income by Student
```

```
t = -0.36182, df = 46.39, p-value = 0.7191
alternative hypothesis: true difference in means between group No and group Yes is not equal
95 percent confidence interval:
 -15.11559  10.50856
sample estimates:
 mean in group No mean in group Yes
         44.98853          47.29205
```

Prediction Interval

```r
stud.inc <- lm(Income ~ Student, data = Credit)
predict(stud.inc, new_data = Credit, interval = "prediction")
```

```
        fit       lwr      upr
1   44.98853 -24.46956 114.4466
2   47.29205 -22.93145 117.5155
3   44.98853 -24.46956 114.4466
4   44.98853 -24.46956 114.4466
5   44.98853 -24.46956 114.4466
6   44.98853 -24.46956 114.4466
7   44.98853 -24.46956 114.4466
8   44.98853 -24.46956 114.4466
9   44.98853 -24.46956 114.4466
10  47.29205 -22.93145 117.5155
11  44.98853 -24.46956 114.4466
12  44.98853 -24.46956 114.4466
13  44.98853 -24.46956 114.4466
14  44.98853 -24.46956 114.4466
15  44.98853 -24.46956 114.4466
16  44.98853 -24.46956 114.4466
17  44.98853 -24.46956 114.4466
18  44.98853 -24.46956 114.4466
19  44.98853 -24.46956 114.4466
20  44.98853 -24.46956 114.4466
21  44.98853 -24.46956 114.4466
22  44.98853 -24.46956 114.4466
23  44.98853 -24.46956 114.4466
24  44.98853 -24.46956 114.4466
25  44.98853 -24.46956 114.4466
26  44.98853 -24.46956 114.4466
```

27	47.29205	-22.93145	117.5155
28	44.98853	-24.46956	114.4466
29	44.98853	-24.46956	114.4466
30	44.98853	-24.46956	114.4466
31	44.98853	-24.46956	114.4466
32	44.98853	-24.46956	114.4466
33	44.98853	-24.46956	114.4466
34	44.98853	-24.46956	114.4466
35	44.98853	-24.46956	114.4466
36	47.29205	-22.93145	117.5155
37	44.98853	-24.46956	114.4466
38	44.98853	-24.46956	114.4466
39	44.98853	-24.46956	114.4466
40	44.98853	-24.46956	114.4466
41	44.98853	-24.46956	114.4466
42	47.29205	-22.93145	117.5155
43	44.98853	-24.46956	114.4466
44	44.98853	-24.46956	114.4466
45	44.98853	-24.46956	114.4466
46	44.98853	-24.46956	114.4466
47	47.29205	-22.93145	117.5155
48	47.29205	-22.93145	117.5155
49	44.98853	-24.46956	114.4466
50	47.29205	-22.93145	117.5155
51	44.98853	-24.46956	114.4466
52	44.98853	-24.46956	114.4466
53	44.98853	-24.46956	114.4466
54	44.98853	-24.46956	114.4466
55	44.98853	-24.46956	114.4466
56	44.98853	-24.46956	114.4466
57	44.98853	-24.46956	114.4466
58	44.98853	-24.46956	114.4466
59	44.98853	-24.46956	114.4466
60	44.98853	-24.46956	114.4466
61	44.98853	-24.46956	114.4466
62	44.98853	-24.46956	114.4466
63	44.98853	-24.46956	114.4466
64	44.98853	-24.46956	114.4466
65	44.98853	-24.46956	114.4466
66	44.98853	-24.46956	114.4466
67	44.98853	-24.46956	114.4466
68	44.98853	-24.46956	114.4466
69	44.98853	-24.46956	114.4466

70	44.98853	-24.46956	114.4466
71	44.98853	-24.46956	114.4466
72	44.98853	-24.46956	114.4466
73	44.98853	-24.46956	114.4466
74	44.98853	-24.46956	114.4466
75	44.98853	-24.46956	114.4466
76	44.98853	-24.46956	114.4466
77	47.29205	-22.93145	117.5155
78	44.98853	-24.46956	114.4466
79	44.98853	-24.46956	114.4466
80	44.98853	-24.46956	114.4466
81	44.98853	-24.46956	114.4466
82	44.98853	-24.46956	114.4466
83	44.98853	-24.46956	114.4466
84	44.98853	-24.46956	114.4466
85	44.98853	-24.46956	114.4466
86	44.98853	-24.46956	114.4466
87	44.98853	-24.46956	114.4466
88	44.98853	-24.46956	114.4466
89	44.98853	-24.46956	114.4466
90	44.98853	-24.46956	114.4466
91	44.98853	-24.46956	114.4466
92	44.98853	-24.46956	114.4466
93	44.98853	-24.46956	114.4466
94	44.98853	-24.46956	114.4466
95	44.98853	-24.46956	114.4466
96	44.98853	-24.46956	114.4466
97	47.29205	-22.93145	117.5155
98	44.98853	-24.46956	114.4466
99	47.29205	-22.93145	117.5155
100	44.98853	-24.46956	114.4466
101	44.98853	-24.46956	114.4466
102	47.29205	-22.93145	117.5155
103	47.29205	-22.93145	117.5155
104	44.98853	-24.46956	114.4466
105	44.98853	-24.46956	114.4466
106	44.98853	-24.46956	114.4466
107	44.98853	-24.46956	114.4466
108	44.98853	-24.46956	114.4466
109	44.98853	-24.46956	114.4466
110	44.98853	-24.46956	114.4466
111	44.98853	-24.46956	114.4466
112	44.98853	-24.46956	114.4466

113	44.98853	-24.46956	114.4466
114	44.98853	-24.46956	114.4466
115	44.98853	-24.46956	114.4466
116	44.98853	-24.46956	114.4466
117	44.98853	-24.46956	114.4466
118	44.98853	-24.46956	114.4466
119	44.98853	-24.46956	114.4466
120	44.98853	-24.46956	114.4466
121	44.98853	-24.46956	114.4466
122	44.98853	-24.46956	114.4466
123	44.98853	-24.46956	114.4466
124	44.98853	-24.46956	114.4466
125	44.98853	-24.46956	114.4466
126	44.98853	-24.46956	114.4466
127	47.29205	-22.93145	117.5155
128	44.98853	-24.46956	114.4466
129	47.29205	-22.93145	117.5155
130	44.98853	-24.46956	114.4466
131	47.29205	-22.93145	117.5155
132	44.98853	-24.46956	114.4466
133	44.98853	-24.46956	114.4466
134	44.98853	-24.46956	114.4466
135	44.98853	-24.46956	114.4466
136	44.98853	-24.46956	114.4466
137	44.98853	-24.46956	114.4466
138	44.98853	-24.46956	114.4466
139	44.98853	-24.46956	114.4466
140	44.98853	-24.46956	114.4466
141	47.29205	-22.93145	117.5155
142	44.98853	-24.46956	114.4466
143	44.98853	-24.46956	114.4466
144	44.98853	-24.46956	114.4466
145	44.98853	-24.46956	114.4466
146	44.98853	-24.46956	114.4466
147	44.98853	-24.46956	114.4466
148	44.98853	-24.46956	114.4466
149	44.98853	-24.46956	114.4466
150	44.98853	-24.46956	114.4466
151	44.98853	-24.46956	114.4466
152	44.98853	-24.46956	114.4466
153	47.29205	-22.93145	117.5155
154	44.98853	-24.46956	114.4466
155	44.98853	-24.46956	114.4466

156	44.98853	-24.46956	114.4466
157	44.98853	-24.46956	114.4466
158	47.29205	-22.93145	117.5155
159	44.98853	-24.46956	114.4466
160	44.98853	-24.46956	114.4466
161	44.98853	-24.46956	114.4466
162	44.98853	-24.46956	114.4466
163	44.98853	-24.46956	114.4466
164	44.98853	-24.46956	114.4466
165	44.98853	-24.46956	114.4466
166	44.98853	-24.46956	114.4466
167	44.98853	-24.46956	114.4466
168	44.98853	-24.46956	114.4466
169	44.98853	-24.46956	114.4466
170	44.98853	-24.46956	114.4466
171	44.98853	-24.46956	114.4466
172	44.98853	-24.46956	114.4466
173	44.98853	-24.46956	114.4466
174	44.98853	-24.46956	114.4466
175	44.98853	-24.46956	114.4466
176	44.98853	-24.46956	114.4466
177	44.98853	-24.46956	114.4466
178	44.98853	-24.46956	114.4466
179	44.98853	-24.46956	114.4466
180	44.98853	-24.46956	114.4466
181	44.98853	-24.46956	114.4466
182	44.98853	-24.46956	114.4466
183	44.98853	-24.46956	114.4466
184	44.98853	-24.46956	114.4466
185	44.98853	-24.46956	114.4466
186	44.98853	-24.46956	114.4466
187	44.98853	-24.46956	114.4466
188	44.98853	-24.46956	114.4466
189	44.98853	-24.46956	114.4466
190	44.98853	-24.46956	114.4466
191	44.98853	-24.46956	114.4466
192	47.29205	-22.93145	117.5155
193	44.98853	-24.46956	114.4466
194	44.98853	-24.46956	114.4466
195	44.98853	-24.46956	114.4466
196	44.98853	-24.46956	114.4466
197	44.98853	-24.46956	114.4466
198	44.98853	-24.46956	114.4466

```
199 44.98853 -24.46956 114.4466
200 44.98853 -24.46956 114.4466
201 44.98853 -24.46956 114.4466
202 44.98853 -24.46956 114.4466
203 44.98853 -24.46956 114.4466
204 47.29205 -22.93145 117.5155
205 44.98853 -24.46956 114.4466
206 44.98853 -24.46956 114.4466
207 44.98853 -24.46956 114.4466
208 47.29205 -22.93145 117.5155
209 44.98853 -24.46956 114.4466
210 44.98853 -24.46956 114.4466
211 44.98853 -24.46956 114.4466
212 44.98853 -24.46956 114.4466
213 44.98853 -24.46956 114.4466
214 44.98853 -24.46956 114.4466
215 44.98853 -24.46956 114.4466
216 44.98853 -24.46956 114.4466
217 44.98853 -24.46956 114.4466
218 44.98853 -24.46956 114.4466
219 47.29205 -22.93145 117.5155
220 44.98853 -24.46956 114.4466
221 47.29205 -22.93145 117.5155
222 44.98853 -24.46956 114.4466
223 47.29205 -22.93145 117.5155
224 44.98853 -24.46956 114.4466
225 44.98853 -24.46956 114.4466
226 44.98853 -24.46956 114.4466
227 44.98853 -24.46956 114.4466
228 44.98853 -24.46956 114.4466
229 44.98853 -24.46956 114.4466
230 44.98853 -24.46956 114.4466
231 44.98853 -24.46956 114.4466
232 44.98853 -24.46956 114.4466
233 44.98853 -24.46956 114.4466
234 44.98853 -24.46956 114.4466
235 44.98853 -24.46956 114.4466
236 44.98853 -24.46956 114.4466
237 44.98853 -24.46956 114.4466
238 44.98853 -24.46956 114.4466
239 44.98853 -24.46956 114.4466
240 44.98853 -24.46956 114.4466
241 44.98853 -24.46956 114.4466
```

242 44.98853 -24.46956 114.4466
243 47.29205 -22.93145 117.5155
244 44.98853 -24.46956 114.4466
245 44.98853 -24.46956 114.4466
246 44.98853 -24.46956 114.4466
247 44.98853 -24.46956 114.4466
248 44.98853 -24.46956 114.4466
249 47.29205 -22.93145 117.5155
250 47.29205 -22.93145 117.5155
251 44.98853 -24.46956 114.4466
252 44.98853 -24.46956 114.4466
253 44.98853 -24.46956 114.4466
254 44.98853 -24.46956 114.4466
255 44.98853 -24.46956 114.4466
256 44.98853 -24.46956 114.4466
257 44.98853 -24.46956 114.4466
258 44.98853 -24.46956 114.4466
259 44.98853 -24.46956 114.4466
260 44.98853 -24.46956 114.4466
261 44.98853 -24.46956 114.4466
262 47.29205 -22.93145 117.5155
263 44.98853 -24.46956 114.4466
264 44.98853 -24.46956 114.4466
265 44.98853 -24.46956 114.4466
266 44.98853 -24.46956 114.4466
267 44.98853 -24.46956 114.4466
268 44.98853 -24.46956 114.4466
269 44.98853 -24.46956 114.4466
270 44.98853 -24.46956 114.4466
271 44.98853 -24.46956 114.4466
272 44.98853 -24.46956 114.4466
273 47.29205 -22.93145 117.5155
274 47.29205 -22.93145 117.5155
275 44.98853 -24.46956 114.4466
276 44.98853 -24.46956 114.4466
277 44.98853 -24.46956 114.4466
278 44.98853 -24.46956 114.4466
279 44.98853 -24.46956 114.4466
280 47.29205 -22.93145 117.5155
281 44.98853 -24.46956 114.4466
282 44.98853 -24.46956 114.4466
283 44.98853 -24.46956 114.4466
284 44.98853 -24.46956 114.4466

285	44.98853	-24.46956	114.4466
286	44.98853	-24.46956	114.4466
287	44.98853	-24.46956	114.4466
288	44.98853	-24.46956	114.4466
289	44.98853	-24.46956	114.4466
290	44.98853	-24.46956	114.4466
291	44.98853	-24.46956	114.4466
292	44.98853	-24.46956	114.4466
293	44.98853	-24.46956	114.4466
294	44.98853	-24.46956	114.4466
295	44.98853	-24.46956	114.4466
296	44.98853	-24.46956	114.4466
297	44.98853	-24.46956	114.4466
298	44.98853	-24.46956	114.4466
299	44.98853	-24.46956	114.4466
300	44.98853	-24.46956	114.4466
301	44.98853	-24.46956	114.4466
302	44.98853	-24.46956	114.4466
303	44.98853	-24.46956	114.4466
304	44.98853	-24.46956	114.4466
305	44.98853	-24.46956	114.4466
306	44.98853	-24.46956	114.4466
307	44.98853	-24.46956	114.4466
308	44.98853	-24.46956	114.4466
309	44.98853	-24.46956	114.4466
310	44.98853	-24.46956	114.4466
311	47.29205	-22.93145	117.5155
312	44.98853	-24.46956	114.4466
313	44.98853	-24.46956	114.4466
314	44.98853	-24.46956	114.4466
315	44.98853	-24.46956	114.4466
316	44.98853	-24.46956	114.4466
317	44.98853	-24.46956	114.4466
318	44.98853	-24.46956	114.4466
319	44.98853	-24.46956	114.4466
320	44.98853	-24.46956	114.4466
321	47.29205	-22.93145	117.5155
322	44.98853	-24.46956	114.4466
323	44.98853	-24.46956	114.4466
324	44.98853	-24.46956	114.4466
325	47.29205	-22.93145	117.5155
326	44.98853	-24.46956	114.4466
327	44.98853	-24.46956	114.4466

328	44.98853	-24.46956	114.4466
329	44.98853	-24.46956	114.4466
330	44.98853	-24.46956	114.4466
331	47.29205	-22.93145	117.5155
332	44.98853	-24.46956	114.4466
333	44.98853	-24.46956	114.4466
334	44.98853	-24.46956	114.4466
335	44.98853	-24.46956	114.4466
336	44.98853	-24.46956	114.4466
337	44.98853	-24.46956	114.4466
338	44.98853	-24.46956	114.4466
339	44.98853	-24.46956	114.4466
340	44.98853	-24.46956	114.4466
341	44.98853	-24.46956	114.4466
342	44.98853	-24.46956	114.4466
343	44.98853	-24.46956	114.4466
344	44.98853	-24.46956	114.4466
345	44.98853	-24.46956	114.4466
346	44.98853	-24.46956	114.4466
347	44.98853	-24.46956	114.4466
348	44.98853	-24.46956	114.4466
349	44.98853	-24.46956	114.4466
350	44.98853	-24.46956	114.4466
351	44.98853	-24.46956	114.4466
352	44.98853	-24.46956	114.4466
353	44.98853	-24.46956	114.4466
354	44.98853	-24.46956	114.4466
355	44.98853	-24.46956	114.4466
356	44.98853	-24.46956	114.4466
357	44.98853	-24.46956	114.4466
358	44.98853	-24.46956	114.4466
359	44.98853	-24.46956	114.4466
360	44.98853	-24.46956	114.4466
361	44.98853	-24.46956	114.4466
362	44.98853	-24.46956	114.4466
363	44.98853	-24.46956	114.4466
364	44.98853	-24.46956	114.4466
365	44.98853	-24.46956	114.4466
366	47.29205	-22.93145	117.5155
367	44.98853	-24.46956	114.4466
368	44.98853	-24.46956	114.4466
369	44.98853	-24.46956	114.4466
370	44.98853	-24.46956	114.4466

```
371 44.98853 -24.46956 114.4466
372 44.98853 -24.46956 114.4466
373 47.29205 -22.93145 117.5155
374 47.29205 -22.93145 117.5155
375 44.98853 -24.46956 114.4466
376 44.98853 -24.46956 114.4466
377 44.98853 -24.46956 114.4466
378 44.98853 -24.46956 114.4466
379 44.98853 -24.46956 114.4466
380 44.98853 -24.46956 114.4466
381 44.98853 -24.46956 114.4466
382 44.98853 -24.46956 114.4466
383 47.29205 -22.93145 117.5155
384 44.98853 -24.46956 114.4466
385 44.98853 -24.46956 114.4466
386 44.98853 -24.46956 114.4466
387 44.98853 -24.46956 114.4466
388 44.98853 -24.46956 114.4466
389 44.98853 -24.46956 114.4466
390 44.98853 -24.46956 114.4466
391 44.98853 -24.46956 114.4466
392 44.98853 -24.46956 114.4466
393 44.98853 -24.46956 114.4466
394 44.98853 -24.46956 114.4466
395 44.98853 -24.46956 114.4466
396 44.98853 -24.46956 114.4466
397 44.98853 -24.46956 114.4466
398 44.98853 -24.46956 114.4466
399 44.98853 -24.46956 114.4466
400 44.98853 -24.46956 114.4466
```

While there is only one mean interval for the confidence Interval (t.test), the Prediction interval is larger„ ranging from around **-25 to 120**, while the Confidence Interval ranges from roughly **-15 - 10**. I apologize for the long PDF as I don't know hpow to only show the first couple rows for the prediction intervals.