Lab 11 - Choosing a Model

Sanaz Saadatifar

4/17

Lab report

Load data

```
data <- read_csv("https://dyurovsky.github.io/85309/data/lab11/animal_game.csv")
```

```
## Rows: 1303 Columns: 8
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): subj_group, trial_target, appearance
## dbl (4): subj, trial, avg_known, length
## lgl (1): known
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Exercise 1:

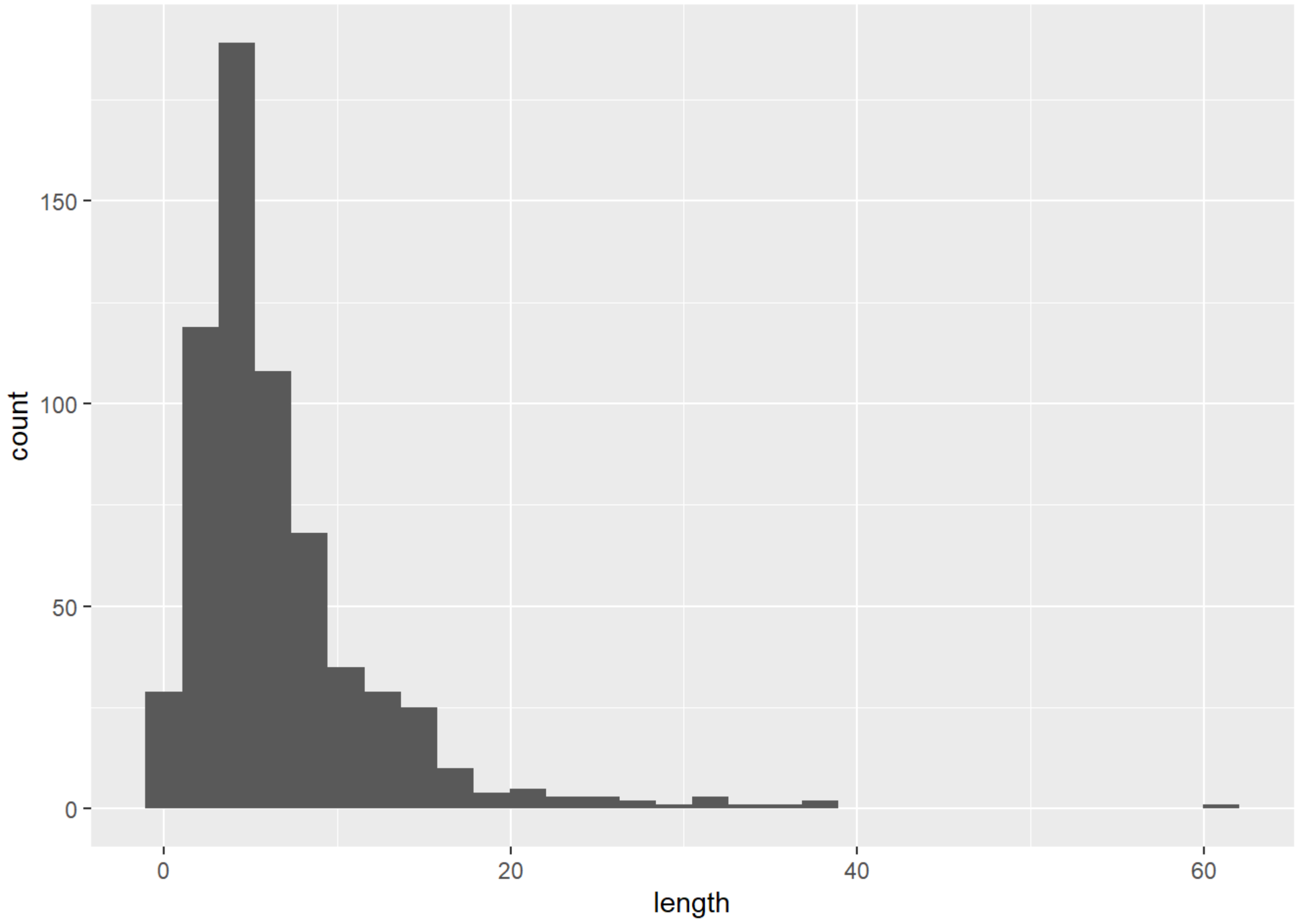
```
train_data <- data %>%
  filter(subj_group == "train")

train_data
```

```
## # A tibble: 638 x 8
##    subj_group  subj trial trial_target appearance avg_known known length
##    <chr>      <dbl> <dbl> <chr>        <chr>          <dbl> <lgl>  <dbl>
##  1 train          1    20 swan         first          0.122 FALSE     38
##  2 train          1    34 swan         second         0.122 FALSE      9
##  3 train          2    17 swan         first          0.122 FALSE     19
##  4 train          2    29 swan         second         0.122 FALSE     11
##  5 train          3    11 swan         first          0.122 FALSE      3
##  6 train          3    25 swan         second         0.122 FALSE      3
##  7 train          4     8 swan         first          0.122 FALSE      4
##  8 train          4    32 swan         second         0.122 FALSE     10
##  9 train          5     5 swan         first          0.122 FALSE     10
## 10 train          5    17 swan         second         0.122 FALSE      4
## # ... with 628 more rows
```

Exercise 2:

```
ggplot(train_data, aes(x = length)) +
  geom_histogram()
```

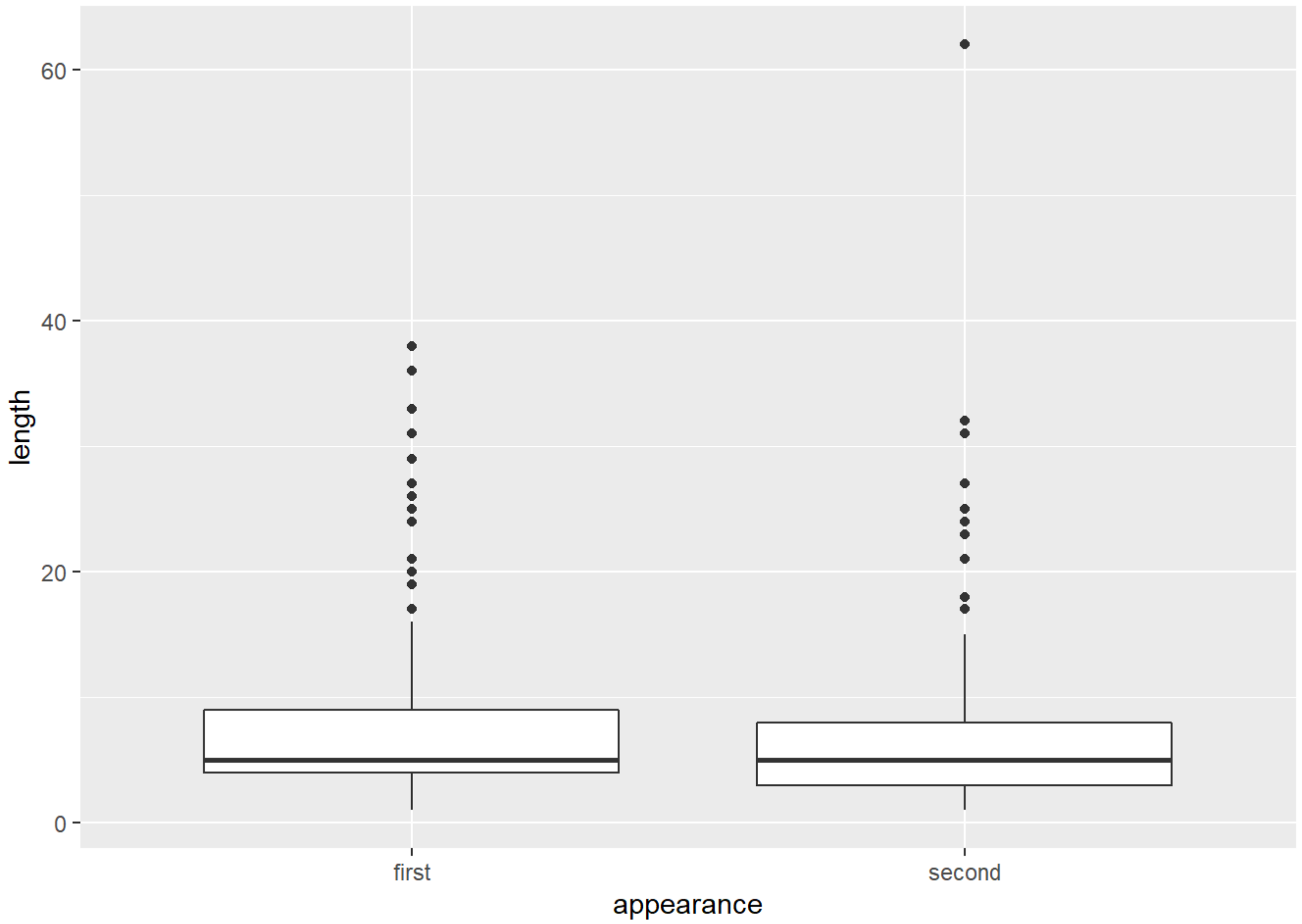
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



the distribution is right is right-skewed. this makes sense because there is a hard lower bound on number of words, you can’t say fewer than 0 words.

Exercise 3:

```
ggplot(train_data, aes(x = appearance , y = length)) +
  geom_boxplot()
```



It

looks like median length is above the same on 1st and 2nd appearance, but there is more mass in the left tail of the distribution on 2nd trials. parents are more likely to use particularly short sentences the second time

Exercise 4:

H0: the average length on the first appearance is the same as the second appearance. Difference in length between first and second appearance is 0 HA: the average length of the first appearance is different from the average length on the second appearance

```
parent_lengths <- train_data %>%
  group_by(subj, appearance) %>%
  summarise(length = mean(length)) %>%
  summarise(diff = first(length) - last(length)) %>%
  summarise(diff = mean(diff)) %>%
  pull(diff)
```

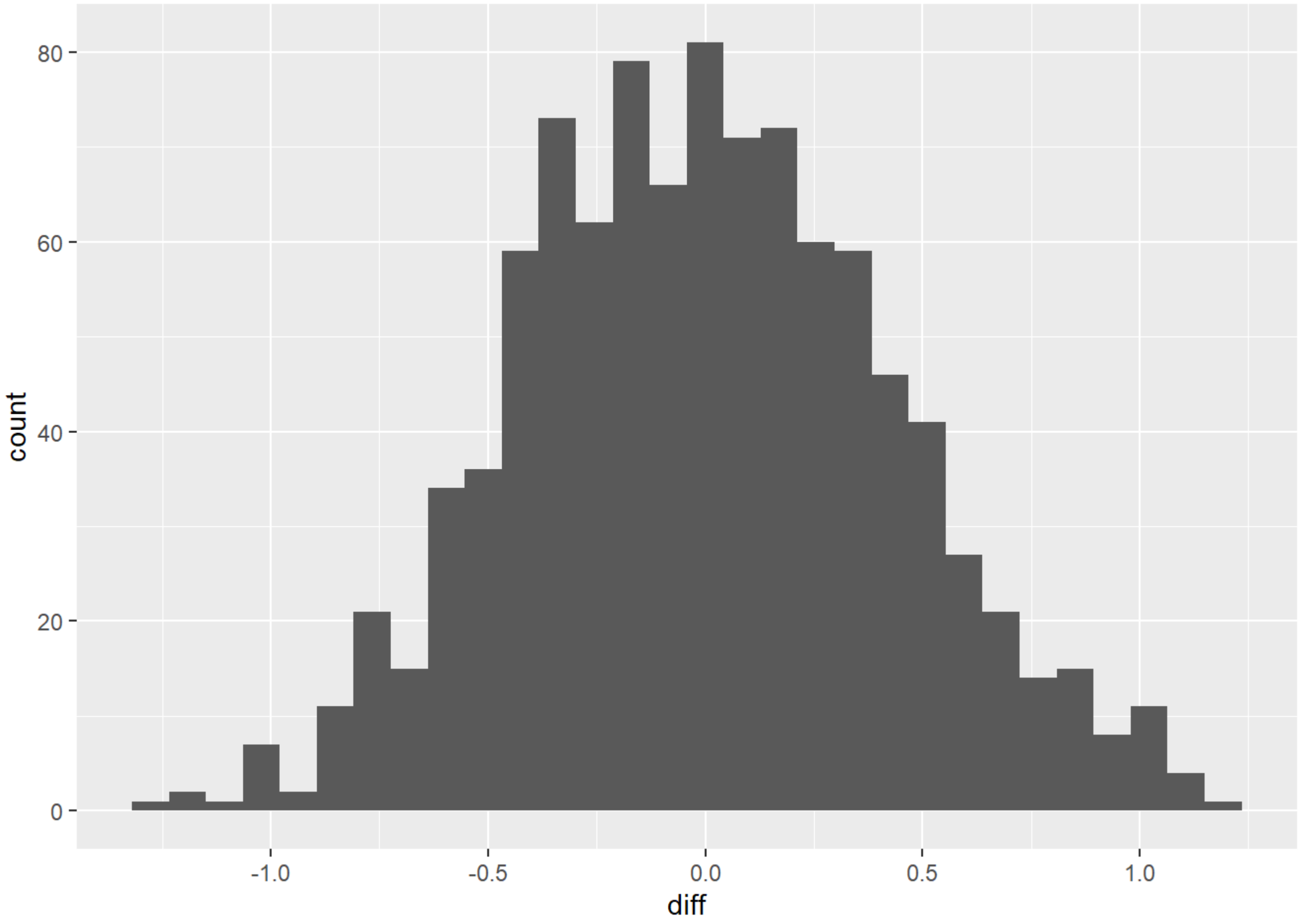
```
## `summarise()` has grouped output by 'subj'. You can override using the `.groups` argument.
```

```r
simulate_null <- function() {
  train_data %>%
    group_by(subj) %>%
    mutate(appearance = sample(appearance)) %>%
    group_by(subj, appearance) %>%
    summarise(length = mean(length),
              .groups = "drop_last") %>%
    summarise(diff = first(length) - last(length)) %>%
    summarise(diff = mean(diff)) %>%
    pull(diff)

}

simulate_null()
```

```
## [1] -0.01291667
```

```r
null_data <- tibble(id = 1:1000,
                    diff = replicate(1000, simulate_null()))



ggplot(null_data, aes(x = diff)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
mean(parent_lengths > pull(null_data, diff))
```

```
## [1] 0.974
```

Our data is in the 98.3 percentile of the null hypothesis distribution. So we reject the null hypothesis that parents produce sentences of the same length on 1st and 2nd appearances.

Exercise 5:

```
parent_lengths_individual <- train_data %>%
  group_by(subj, appearance) %>%
  summarise(length = mean(length))
```

```
## `summarise()` has grouped output by 'subj'. You can override using the `.groups` argument.
```

```
parent_lengths_individual
```

```
## # A tibble: 40 x 3
## # Groups:   subj [20]
##     subj appearance length
##    <dbl> <chr>       <dbl>
##  1     1 first        9.12
##  2     1 second       7.06
##  3     2 first       10.3
##  4     2 second      10.2
##  5     3 first        4
##  6     3 second       3.31
##  7     4 first        4.5
##  8     4 second       4.75
##  9     5 first        4.44
## 10     5 second       4.06
## # ... with 30 more rows
```

```
t.test(length~appearance, data = parent_lengths_individual, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  length by appearance
## t = 2.752, df = 19, p-value = 0.01268
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2115562 1.5555271
## sample estimates:
## mean of the differences
##               0.8835417
```

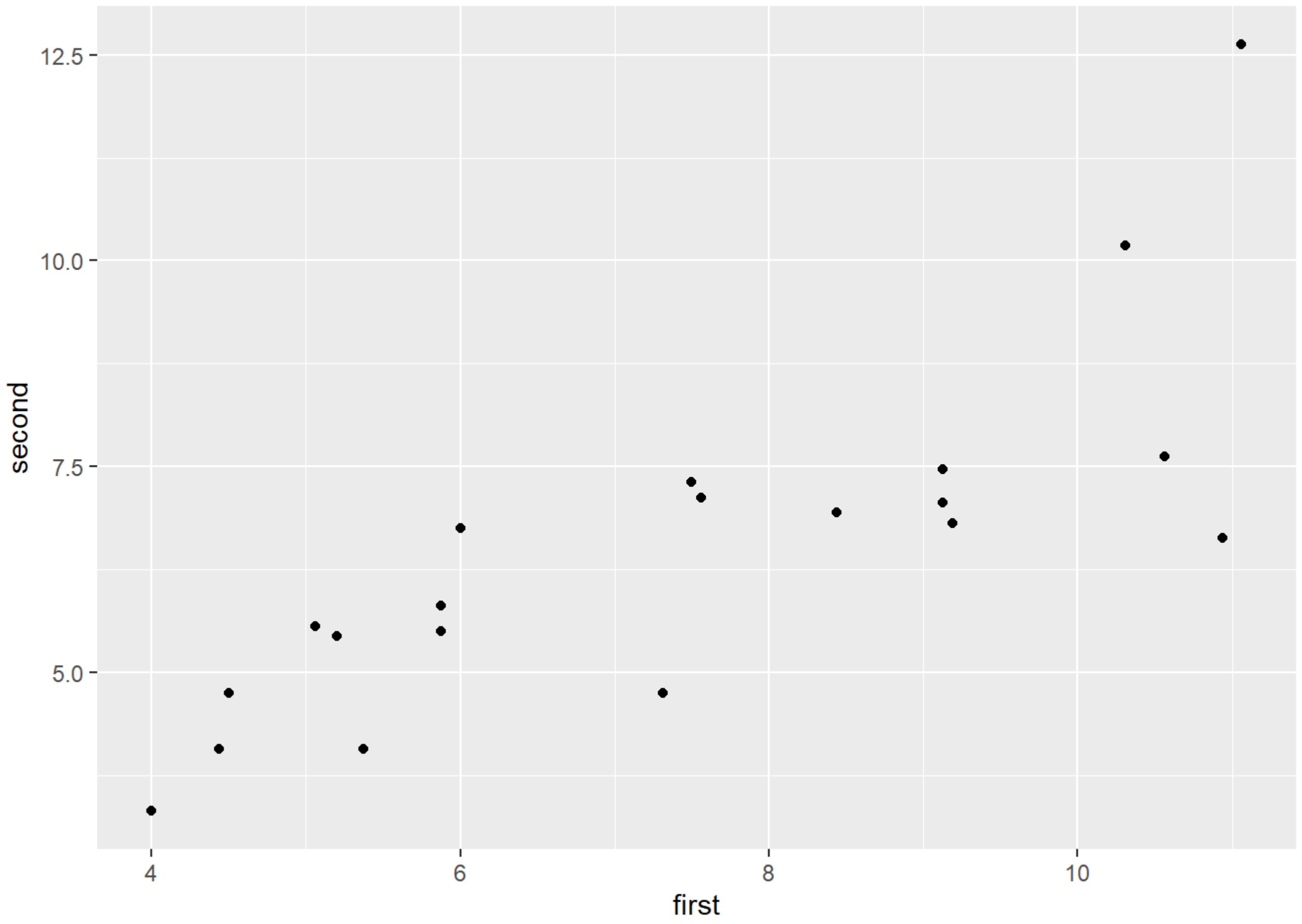
We need to use a paired T test because our parents are the natural pairing unit, they have a first and second appearance that are probably dependent on each other. We got the same result from our T test and our simulation: first appearance sentences are longer on average

Exercise 6:

```
parent_lengths2 <- train_data %>%
  group_by(subj, appearance) %>%
  summarise(length = mean(length),
            .groups = "drop_last") %>%
  pivot_wider(names_from = "appearance", values_from = "length")

parent_lengths2
```

```
## # A tibble: 20 x 3
## # Groups:   subj [20]
##     subj first second
##    <dbl> <dbl>  <dbl>
##  1     1  9.12   7.06
##  2     2 10.3   10.2 
##  3     3  4      3.31
##  4     4  4.5    4.75
##  5     5  4.44   4.06
##  6     6  7.5    7.31
##  7     7 11.1   12.6 
##  8     8  8.44   6.94
##  9     9 10.6    7.62
## 10    10  5.88   5.81
## 11    11  5.2    5.44
## 12    12  5.06   5.56
## 13    13  7.31   4.75
## 14    14  5.38   4.06
## 15    15  6      6.75
## 16    16  5.88   5.5 
## 17    17  9.19   6.81
## 18    18  9.12   7.47
## 19    19  7.56   7.12
## 20    20 10.9    6.62
```

```
ggplot(parent_lengths2, aes(x = first, y = second)) +
  geom_point()
```

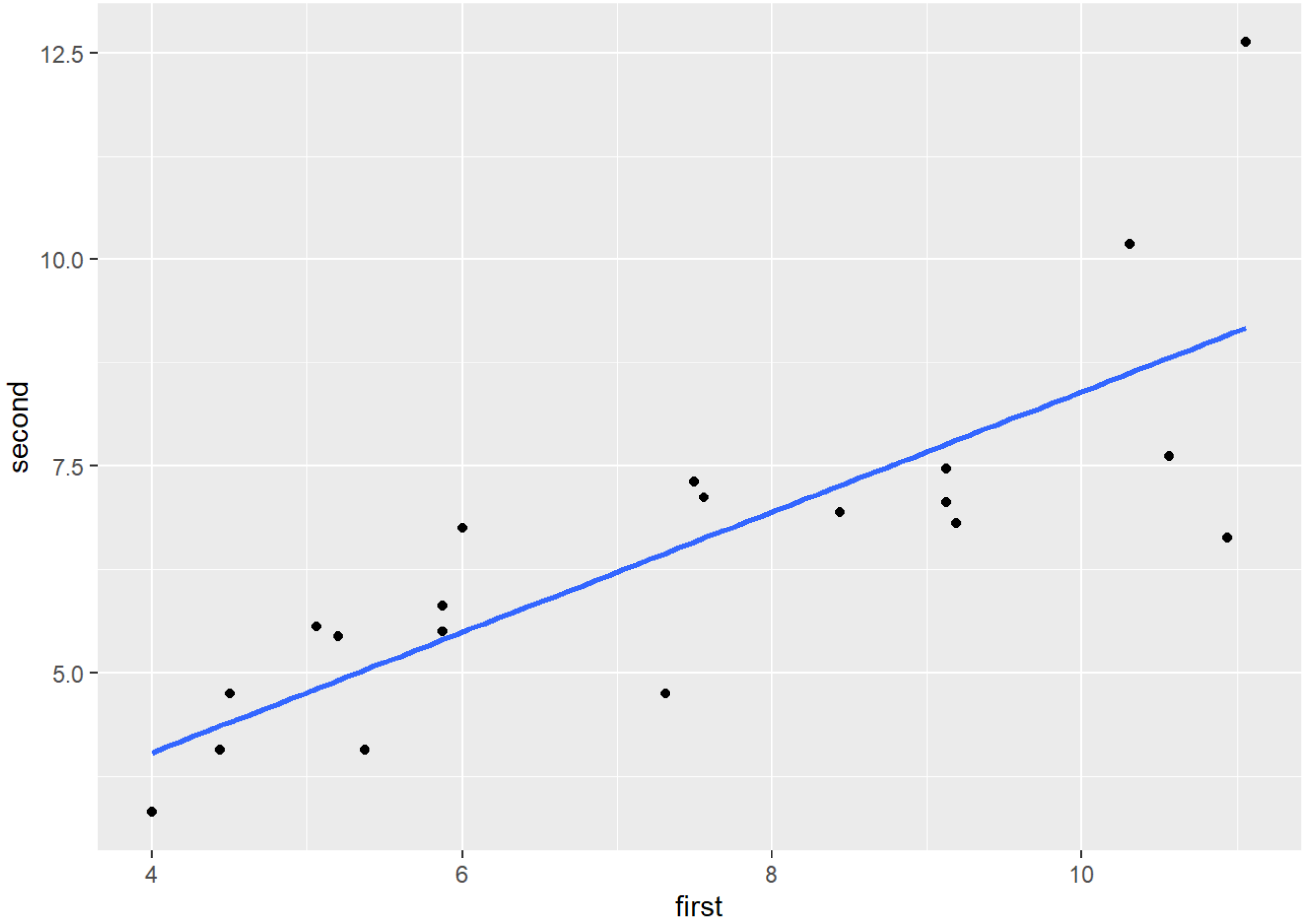


```
m1 <- lm(second ~ first, data = parent_lengths2)
summary(m1)
```

```
##
## Call:
## lm(formula = second ~ first, data = parent_lengths2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4577 -0.7854 -0.0953  0.5795  3.4514
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.1251     0.9979   1.127    0.274
## first         0.7275     0.1293   5.627 2.44e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.321 on 18 degrees of freedom
## Multiple R-squared:  0.6376, Adjusted R-squared:  0.6175
## F-statistic: 31.67 on 1 and 18 DF,  p-value: 2.439e-05
```

```
ggplot(parent_lengths2, aes(x = first, y = second)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



With the linear regression method we can also get the similar result because there is a positive moderate relationship between the length of 1st and 2nd appearances the slope is positive 0.7275. the Pvalue is not exactly the same as the P value of the T test but it corresponds because the P value is small enough to reject the null hypothesis.

Exercise 7:

```
transformed_train <- train_data %>%
  mutate(transformed_length = log(length))


transformed_train_individual <- transformed_train %>%
  group_by(subj, appearance) %>%
  summarise(transformed_length = mean(transformed_length))
```

```
## `summarise()` has grouped output by 'subj'. You can override using the `.groups` argument.
```

```
transformed_train_individual
```

```
## # A tibble: 40 x 3
## # Groups:   subj [20]
##     subj appearance transformed_length
##    <dbl> <chr>                   <dbl>
##  1     1 first                    1.95
##  2     1 second                   1.85
##  3     2 first                    1.97
##  4     2 second                   2.10
##  5     3 first                    1.31
##  6     3 second                   1.18
##  7     4 first                    1.47
##  8     4 second                   1.36
##  9     5 first                    1.24
## 10     5 second                   1.08
## # ... with 30 more rows
```

```
t.test(transformed_length~appearance, data = transformed_train_individual, paired = TRUE)
```

```
## 
##  Paired t-test
## 
## data:  transformed_length by appearance
## t = 3.6302, df = 19, p-value = 0.001782
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.05567129 0.20727181
## sample estimates:
## mean of the differences 
##               0.1314715
```

The P value is changed it is very smaller in the second T test with the log length data. the smaller P value is better because it shows that we can trust the model now better. the reason for this change is it transformation of the length two log length. the length itself had a right skewed distribution but the lug length probably meets the normality requirement more.

Exercise 8:

```
full_model <- lm(transformed_length  ~ trial + appearance + avg_known + known , data = transform
         ed_train)
summary(full_model)
```

```
##
## Call:
## lm(formula = transformed_length ~ trial + appearance + avg_known +
##     known, data = transformed_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9490 -0.3983 -0.0223  0.4233  2.3342
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.115577   0.082781  25.556  < 2e-16 ***
## trial            -0.004849   0.003224  -1.504 0.133036
## appearancesecond -0.070825   0.067407  -1.051 0.293791
## avg_known        -0.451588   0.117051  -3.858 0.000126 ***
## knownTRUE         0.022273   0.078323   0.284 0.776212
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6666 on 633 degrees of freedom
## Multiple R-squared:  0.05374,    Adjusted R-squared:  0.04776
## F-statistic: 8.987 on 4 and 633 DF,  p-value: 4.606e-07
```

Overall the r ^2 and adjusted r ^2 are small and other than average knowns P value, other P values are not small enough so this model is not very appropriate to make predictions. however based on the data we have average known seems to be the most relevant data to make predictions about the transformed length due to the smallest P value. in terms of relationships trial has not that strong but negative slope.

Exercise 9:

```
step_model <- step(full_model)
```

```
## Start:  AIC=-512.54
## transformed_length ~ trial + appearance + avg_known + known
## 
##              Df Sum of Sq    RSS     AIC
## - known       1    0.0359 281.30 -514.46
## - appearance  1    0.4906 281.76 -513.43
## <none>                    281.27 -512.54
## - trial       1    1.0053 282.27 -512.27
## - avg_known   1    6.6138 287.88 -499.72
## 
## Step:  AIC=-514.46
## transformed_length ~ trial + appearance + avg_known
## 
##              Df Sum of Sq    RSS     AIC
## - appearance  1    0.4837 281.79 -515.37
## <none>                    281.30 -514.46
## - trial       1    1.0244 282.33 -514.14
## - avg_known   1   12.2191 293.52 -489.33
## 
## Step:  AIC=-515.37
## transformed_length ~ trial + avg_known
## 
##             Df Sum of Sq    RSS     AIC
## <none>                   281.79 -515.37
## - trial      1    3.4005 285.19 -509.71
## - avg_known  1   12.2597 294.05 -490.20
```

Appearance and known are deleted and trial and average known are still in the model. my first interpretation based on the previous questions result was so just have average known as a criteria to make predictions based on that. however based on the results of this question apparently trial is also important in making predictions besides average known

Exercise 10:

```
predicted_train <- transformed_train %>%
  mutate(predicted_full = predict(full_model),
         predicted_step = predict(step_model))

predicted_train %>%
  summarise(cor = cor(predicted_full, transformed_length )) %>%
  pull()
```

```
## [1] 0.2318105
```

```
predicted_train %>%
  summarise(cor = cor(predicted_step, transformed_length )) %>%
  pull()
```

```
## [1] 0.2280083
```

I expected a high correlation between the predicted step and transform length compared to the correlation of predicted full and transform length but apparently it didn't happen

More Practice

Exercise 11:

```
test_data <- data %>%
  filter(subj_group == "test")

transformed_test  <- test_data %>%
  mutate(transformed_length = log(length))

predicted_test <- transformed_test %>%
  mutate(predicted_full = predict(full_model, newdata = .),
         predicted_step = predict(step_model, newdata = .))
```

Exercise 12:

```
predicted_test %>%
  summarise(cor = cor(predicted_full, transformed_length )) %>%
  pull()
```

```
## [1] 0.1756766
```

```
predicted_test %>%
  summarise(cor = cor(predicted_step, transformed_length )) %>%
  pull()
```

```
## [1] 0.1583336
```

Full model predicts better because it has a higher correlation compared to the step model

Exercise 13:

```
transformed_data_full  <- data %>%
  mutate(transformed_length = log(length))



data_full_model <- lm(transformed_length  ~ trial + appearance + avg_known + known , data = tran
         sformed_data_full)
summary(data_full_model)
```

```
##
## Call:
## lm(formula = transformed_length ~ trial + appearance + avg_known +
##     known, data = transformed_data_full)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.00423 -0.40442 -0.01895  0.43518  2.45785
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.005913   0.058652  34.200  < 2e-16 ***
## trial             0.001479   0.002264   0.653 0.513688
## appearancesecond -0.165245   0.047178  -3.503 0.000476 ***
## avg_known        -0.292818   0.078558  -3.727 0.000202 ***
## knownTRUE        -0.118477   0.052863  -2.241 0.025182 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6801 on 1298 degrees of freedom
## Multiple R-squared:  0.0516, Adjusted R-squared:  0.04868
## F-statistic: 17.66 on 4 and 1298 DF,  p-value: 4.03e-14
```

```
step_model <- step(data_full_model)
```

```
## Start:  AIC=-999.65
## transformed_length ~ trial + appearance + avg_known + known
##
##              Df Sum of Sq    RSS      AIC
## - trial       1    0.1974 600.57 -1001.22
## <none>                    600.38  -999.65
## - known       1    2.3233 602.70  -996.62
## - appearance  1    5.6746 606.05  -989.39
## - avg_known   1    6.4263 606.80  -987.78
##
## Step:  AIC=-1001.22
## transformed_length ~ appearance + avg_known + known
##
##              Df Sum of Sq    RSS      AIC
## <none>                    600.57 -1001.22
## - known       1    2.3512 602.93  -998.13
## - avg_known   1    6.4126 606.99  -989.38
## - appearance  1    7.0103 607.58  -988.10
```

Yes the result is different in the training data set trial and average known were left at the end but in the full data set trial is deleted and average known is left besides two other variables which are known and appearance. Of course the main reason is that the data set is changed, and data set got larger so we have more data to do analysis based on. And Because of the increased number of data in the larger data set (full data), the, It is more trustable in terms of variables (known, avg-known, Appearance) that are left for predictions.