Final Project

Tyler Welsh, Kim Dang, Anastasia Zimina

Contents

Problem A

The objective for this problem is to analyze the 100k Movie Lens data. We are interested in finding various confidence intervals, hypothesis tests, and linear regression models. These evaluations can be useful to find any correlations between age and gender in regards to movie ratings, and see if there is any striking differences in the average movie ratings between men and women.

Below is a list of explanations for various terms and phrases that will be used throughout the paper.

- Confidence Intervals are useful for determining how 'certain' we are that a value falls into a certain range. In the following problems below, we will be using a 95% confidence interval, which is saying, "We are 95% sure the values fall within the given interval".
- Hypothesis tests help us determine whether certain claims, such as "The average ratings between men and woman are equal", are true or not.
- Linear Regression Models are ideal for estimating data given another set of data, such as "can we estimate the average rating of users dependent on their gender and age?".
- Sample Size and Sample Population. The sample population is the total population of which we are sampling from, while the sample size is the size of our sample extracted from that population.
- Sample Mean is the average value of the data we're interested in the given sample of data extracted from the sample population.

The data set used for Problem A is a subset of the full Movie Lens 100k data, where we are only interested in the UserID, Age, Gender, and Rating. In most cases, Rating is reduced to the Average Mean Rating per User. All movie ratings are between 1 to 5 stars. The data can be viewed by running the mergeUserData() function (Appendix A.1) and viewing the output file "u.merged".

Before we go through the various findings, it is important to note the difference in sample sizes. The number of men in the study is roughly triple the number of women in the study. This difference will have an affect on both the mean and standard deviation when finding differences of the means as well as the related confidence interval. The data found is still valid, but the intervals will generally be smaller with bigger sample sizes.

a) An approximate 95% confidence interval for the mean ratings by men is:

$$(3.556,\ 3.621)$$

This interval tells us that we are 95% sure that the average male movie ratings fall between 3.556 and 3.621.

There was a Sample Size of 670 Men, a Sample Mean of 3.588, and a Standard Deviation of 0.430

The interval was found using the confIntMen() function in Appendix A.2.

b) An approximate 95% confidence interval for the population mean rating by women is:

$$(3.530,\ 3.644)$$

The interval found tells us that we are 95% sure that the average movie ratings for women fall between the values 3.530 and 3.644.

There was a Sample Size of 273 Women, a Sample Mean of 3.587, and a Standard Deviation of 0.481

The interval was found using the confIntFemale() function in Appendix A.2

c) An approximate 95% confidence interval for the difference between the two means in a) and b) is:

$$(-0.064,\ 0.067)$$

This interval found tells us that we are 95% sure that the difference between the average movie ratings between men and women is between -0.064 and 0.067. This is different than the previous intervals in that this interval doesn't give an average rating interval, but a difference in averages between two sets. When we have such a small interval, we can reason that the difference between the two networks is very small, and therefor not very significant.

There was a Sample Size of 670 Men and 273 Women, a Sample Mean of 3.588 for Male and 3.587 for Female, and a Standard Deviation of 0.430 for Male and 0.481 for Female.

The interval was found using the confIntDiff() function in Appendix A.2

d) The following is a significance test of the hypothesis that the male and female population means are equal. First we will go through the derivation and then evaluate the results.

We want to test if $H_0 = c$ where H_0 is the Male mean ratings and c is the Female mean ratings. We will use the equation:

$$Z = \frac{\bar{X} - c}{\sigma/\sqrt{n}}$$

Where $\bar{X}$ is the sample mean ratings of men, c is the female mean ratings,and $\sigma/\sqrt{n}$ is the standard error, with σ being the standard deviation of female mean ratings, and n being the sample population of females.

We will reject the hypothesis if the value of Z is less than -1.96 or greater than 1.96. This 1.96 is generally called the 5% level cutoff for hypothesis testing.

Running the hypoth() function in A.4 we found

$$Z = -0.086$$

Since Z lies within -1.96 and 1.96, we do **NOT** reject the hypothesis that the male and female populations means are equal.

e) Histogram plots for the average ratings of Men and Woman were created. The Male histogram can be found on page 7 and the Female histogram can be found on page 8. From the graphs, we can see there is a large occurrence of mean ratings between 3.25 and 3.75 for Men and Women. What's interesting is that there are zero occurrences of 5 star ratings among men, but some for women.

The histograms were created using the histo() function in Appendix A.3

f) An approximate 95% confidence interval for the difference between the number of ratings between men and women is:

$$(48505.93,\ 48534.07)$$

This interval shows that the difference in the number of ratings of Men and the number of ratings of Women falls between 48,505.93 and 48,534.07. At first, this seems like an unusually high interval, but the provided data supports the results:

Men had a mean number of ratings of 74260 while women had a mean number of ratings of 25740

Now we can see why the interval is so large.

We must stress that the number of ratings by male and female users are independent of each other. This is an important distinction that was left out earlier, but when calculating confidence intervals, the data must be independent for our mathematical work to hold. In previous problems, the mean rating of users have been independent, while the rating per movie was not independent. Because we are not working with actual rating values but the number of ratings per user in this situation, we need not worry about accounting for dependent variables.

The interval was found using the confIntPopRat() function in Appendix A.2

g) An approximate 95% confidence interval for the proportion of users who are male is:

$$(0.683,\ 0.739)$$

Instead of a raw numeric value like previous intervals, this confidence interval shows us a percentage value. This is the percentage of the users who are male, which is somewhere between 68.3% and 73.9%.

The interval was found using the confIntPropMale() function in Appendix A.2

h) Here, we will use a Linear Regression Model to fit the data into a suitable model and make various calculations such as estimations of data given other data. For this problem, we will fit the MovieLens data into a linear function to predict a movie's rating based on a user's age and gender. The function we have produced is:

$$m_{W;H}(t) = \beta_0 + \beta_1 t_1 + \beta_2 t_2$$

Which translates to:

$$\text{mean rating} = \beta_0 + \beta_1 \text{ age} + \beta_2 \text{ gender}$$

The β values come from our data set: β_0 is our y-intercept using the Mean Rating data, β_1 using the Age data, and β_2 using the Gender data. β_1 and β_2 end up being our slope values for the linear function.

Below we can see the data produced by running the linear model function lm(). To see the full code used, see Appendix A.5. We are only really interested in the Estimate and Std. Error columns of the (Intercept), A$Gen, and A$Age rows

```
Call:
lm(formula = A$Mean ~ A$Gen + A$Age)

Residuals:
     Min        1Q    Median        3Q       Max
-2.06903  -0.25972   0.03078   0.27967   1.34615

Coefficients:
```

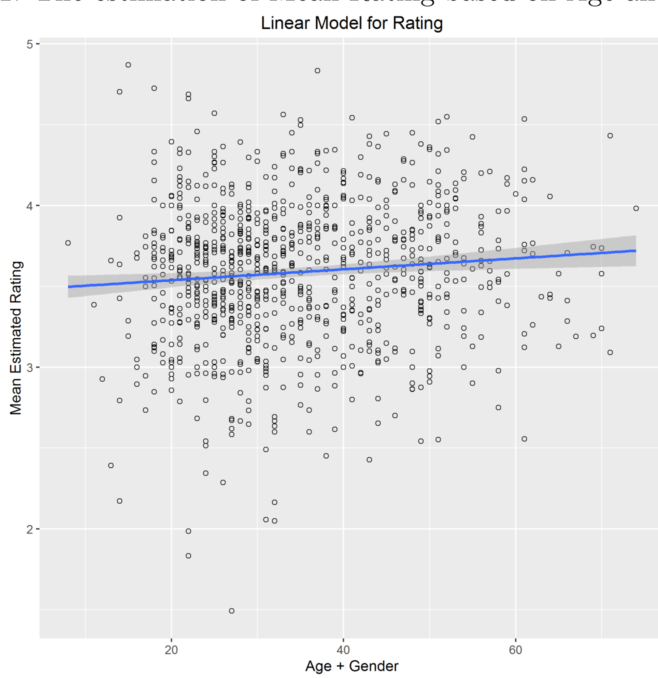
```
                Estimate  Std. Error  t value  Pr(>|t|)
(Intercept) 3.4725821   0.0482655   71.947   < 2e-16 ***
A$Gen       0.0002862   0.0318670    0.009   0.99284
A$Age       0.0033891   0.0011860    2.858   0.00436 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4438 on 940 degrees of freedom
Multiple R-squared:  0.008615,      Adjusted R-squared:  0.006505
F-statistic: 4.084 on 2 and 940 DF,  p-value: 0.01714
```

Using this data, we can form a 95% confidence interval for the estimation of mean rating from age and gender:

$$(3.375,\ 3.565)$$

From this, we can say that the estimate mean rating from age and gender will fall somewhere between 3.375 and 3.565, which is not far off of previous data found. See Figure 1 for a visualization of the data.

Figure 1: The estimation of Mean Rating based on Age and Gender



We can also find a confidence interval for the coefficient β_{age}. From the output above, we know the estimate of β_{age} is 0.0034, with a Standard Error of 0.0011. We can use these to find the interval:

$$(0.0011,\ 0.0058)$$

From this, we can ascertain that the coefficient β_{age} must fall somewhere between 0.0011 and 0.0058.

Just as we did with d), we will test a hypothesis: $H_0 : \beta_{age} = 0$ The hypothesis that the above coefficient for β_{age} is 0.

For our hypothesis test, we can use the Standard Error we found in the summary above. Using the equation:

$$Z = \frac{\hat{\theta} - c}{s.e(\hat{\theta})}$$

Where $\hat{\theta} = 0.0033891$, $c = \beta_{age} = 0$ and $s.e(\hat{\theta}) = 0.0011860$

$$Z = \frac{0.0033891 - 0}{0.0011860} = 2.857588$$

This is larger than 1.98, and so we reject the hypothesis that $\beta_{age} = 0$.

Next, with a similar call to lm() (see Appendix A.6 for the output of the lm() call), we can find a confidence interval for the mean population rating among women who are 28 years old:

$$(3.294,\ 3.846)$$

This was found by fixing the β values to β_{gender} = Female and $\beta_{age} = 28$. From the output we found the estimated mean rating among woman who are 28 is 3.57 with a standard error of 0.1407.

So now we are 95% sure that the mean rating among woman who are 28 years old will fall between 3.294 and 3.846. This is somewhat of a broad interval, as there is a small sample size of females who are age 28 and therefor is not a very good representation of the full population.

Figure 2: Histogram of the frequency of average movie ratings for men

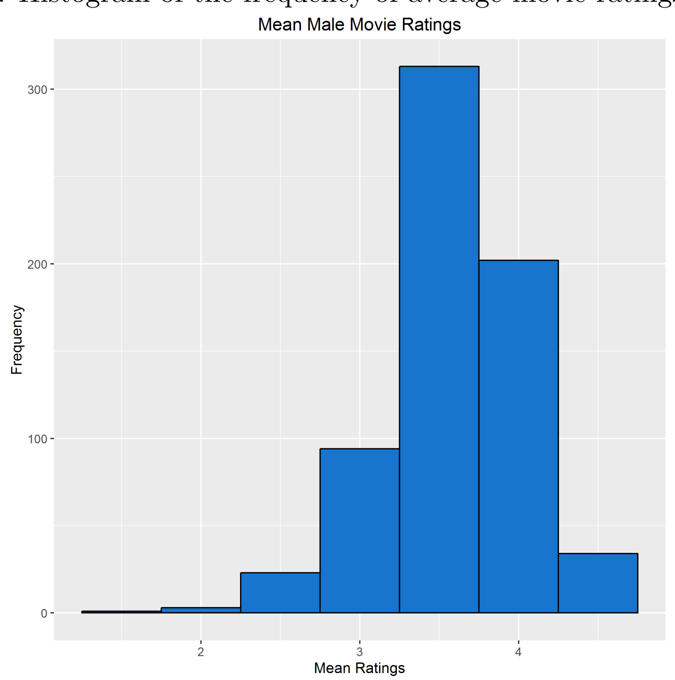
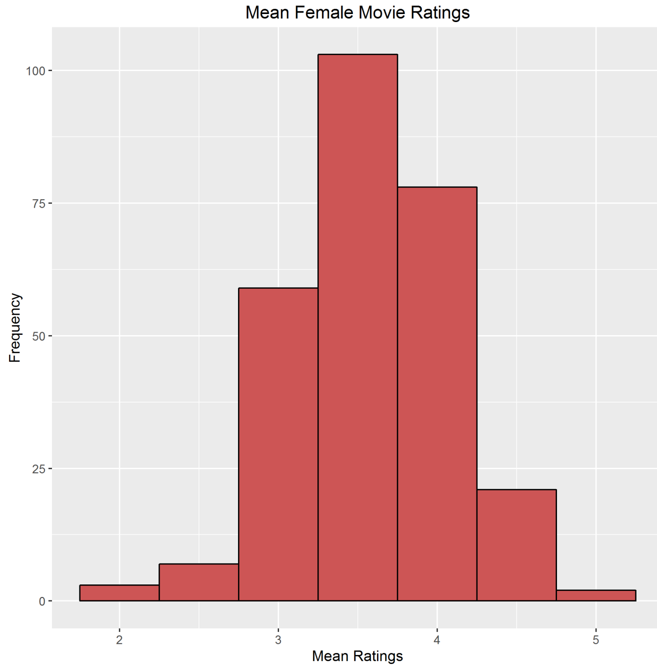


Figure 3: Histogram of the frequency of average movie ratings for women



Problem B

Introduction to Modeling Wordbank Data

A study from Stanford utilizes questionnaires to collect data on a child's vocabulary development in various language, including factors such as their age, ethnicity, order of birth, gender, and mother's education. This data is open for public use as an online database called WordBank. One thing we could learn from this data is whether or not the mentioned factors have a strong effect on a child's English vocabulary size. We can explore this question by creating a regression model. By fitting the data into a linear function, factors included, we can see if there is a relationship among the variables included in the data.

We will assume that our data is linear enough to implement linear regression function. Then, we identify which of the variables we deem independent and dependent. Since we are determining vocabulary size by the other variables, it will serve as our dependent variable for most of our analysis. The rest are independent variables which we will control to evaluate our dependent variable.

Vocabulary and Age

We will first examine the relationship between vocabulary size and the child's age. We start out with the following regression function:

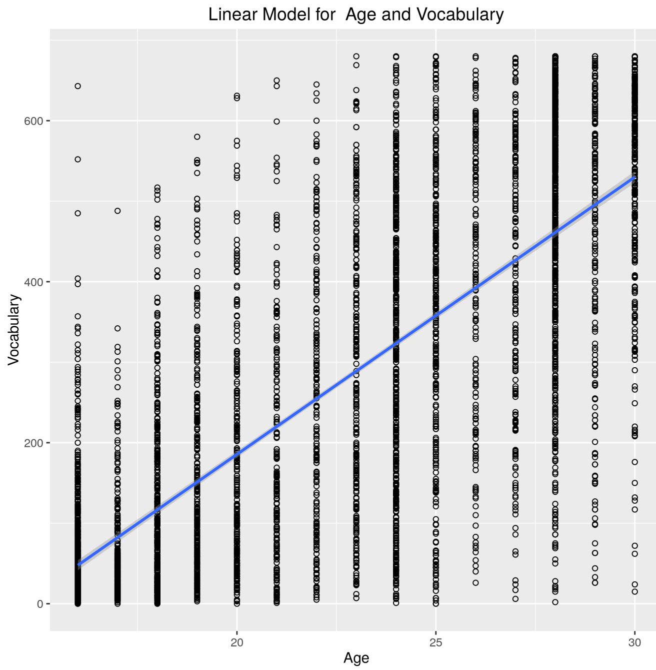
$$m_{S;A}(t) = \beta_0 + \beta_1 t_1$$

In more colloquial terms:

$$\text{vocabulary size} = \beta_0 + \beta_1 \text{ age}$$

The β can be considered constants that we will determine from our code later. What is important to note here is that we believe that for an increase in age, there will be an increase in vocabulary size.

Figure 4: Age and Vocabulary: Linear Regression Model



Now that we've determined the variables for our model, we can begin discussing the most important topic of verifying our regression function. Since we are using the regression function to create a best fit line that encompasses all our data, we naturally would want to know how well our data fits the function. We can use the built in R function lm() for linear models to perform a regression analysis as well as tests for correlation.

A call to the function AgeVocab(), referenced from A.8 gives us interesting information, presented below:

```
Average age 22.65378
Average vocabulary 275.4396
Standard deviation: age 4.27693
Standard deviation: vocabulary 204.9773
Sample size: age 2741
Sample size: vocabulary 2741
Standard error: age 0.1601127
95% Confidence Interval 22.49366 , 22.81389
Standard error: vocabulary 7.673603
95% Confidence Interval 267.766 , 283.1132
```

A large standard deviation on both ends of this regression function is telling, in that the two might not have a strong relationship.

We could do more tests on the various variables to see if we are good at predicting with this model, but we can easily see these details through more telling parts of the readout. The results of the call to lm() can be found in Appendix A.7. At the bottom of our results, we see the terms "R-squared" and "Adjusted R-squared". These numbers use a thing called residuals, which in simple terms are the distances each point of data is from the regression function. The quantity R^2 is an estimate of the correlation between vocabulary size and all the variables we used to predict it. The closer R^2 is to 1, a proportion of 100% correlation, the better our data fits the model and the better our model can predict future cases.

In the case of this regression function, R^2 only slightly above 0.5. Normally we would prefer a number higher than 0.5 and closer to 1.0, but we recognize that there many factors that must be taken into account that we didn't include in this regression model. We might have to think twice about whether or not there is a relationship between age and vocabulary.

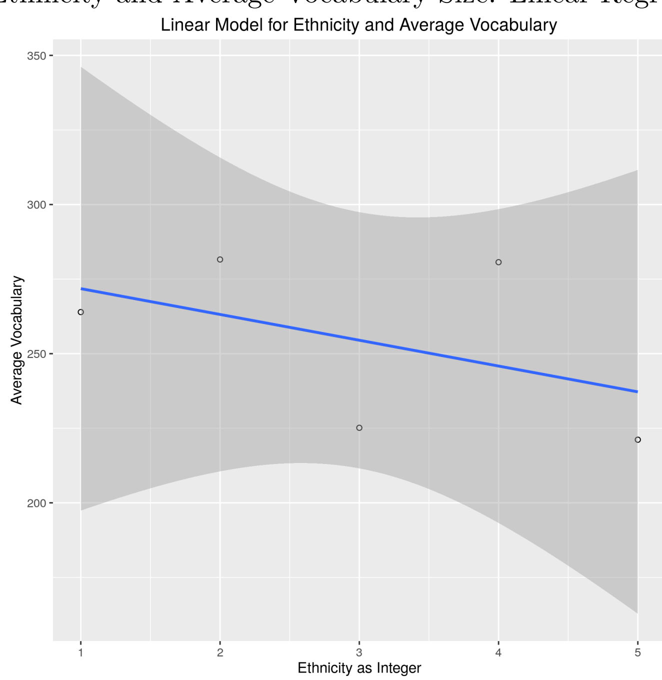
We plotted our data in Figure 4 and noted that there is a positive correlation between age and vocabulary size, as we had predicted earlier when we first formulated our regression function. Using the constants in our results, we have our complete regression function for age as a predictor of vocabulary size:

$$m_{S;A}(t) = 1.793 + 1.645t_1$$

After looking at what our tests provides, we conclude that age is not a good indicator for vocabulary size. This could be caused by a variety of factors, including variables that we will be addressing separately. If we have to make a full regression model that encompasses all vocabulary development, age shouldn't be considered as a highly critical factor.

Vocabulary and Ethnicity

Figure 5: Ethnicity and Average Vocabulary Size: Linear Regression model



We repeat the same process as we had done for age and vocabulary. But this time, we want to determine how one's ethnicity influences one's vocabulary. First, we find the average, standard deviation, standard error, and 95 percent confidence interval for vocabulary size for all ethnicities:

```
Ethnicity:  Asian
Mean:  263.9437
Standard Deviation:  193.038
Sample Size: 71
Standard error:  44.9016
```

```
95% Confidence Interval:  219.0421  ,  308.8453
Ethnicity:  Black
Mean:  281.5746
Standard Deviation:  191.9908
Sample Size: 228
Standard error:  24.92075
95% Confidence Interval:  256.6538  ,  306.4953
Ethnicity:  Other
Mean:  225.1515
Standard Deviation:  174.9849
Sample Size: 99
Standard error:  34.46919
95% Confidence Interval:  190.6823  ,  259.6207
Ethnicity:  White
Mean:  280.6689
Standard Deviation:  208.1625
Sample Size: 2211
Standard error:  8.676731
95% Confidence Interval:  271.9922  ,  289.3457
Ethnicity:  Hispanic
Mean:  221.1515
Standard Deviation:  188.7241
Sample Size: 132
Standard error:  32.195
95% Confidence Interval:  188.9565  ,  253.3465
```

As we can see from the statistics above, some groups are better represented than others. Majority of the sample size classified themselves as "White".That results in smallest standard error and, as a result, confidence interval. Furthermore, even for the most represented group, the standard deviation is very big, which implies that predicting child's vocabulary based on ethnicity might not be the best idea. Huge standard deviation is typical for all the groups, which further proves that ethnicity is not the leading factor in developing child's vocabulary. Nevertheless, we will proceed to the linear regression model, and check if our prediction holds. For convenience, we will assign each ethnicity a number: Asian would be a "1", Black - "2", Other - "3", White -"4" and Hispanic - "5".

We refer to the lm() readout that can be referenced in Appendix A.7. We have low values for R^2 and adjusted R^2, which is not surprising since ethnicity is not a variable with quantitative value.

Our final regression model, constants included, is as follows:

$$m_{E;V}(t) = -479922 + 0.02285t_1$$

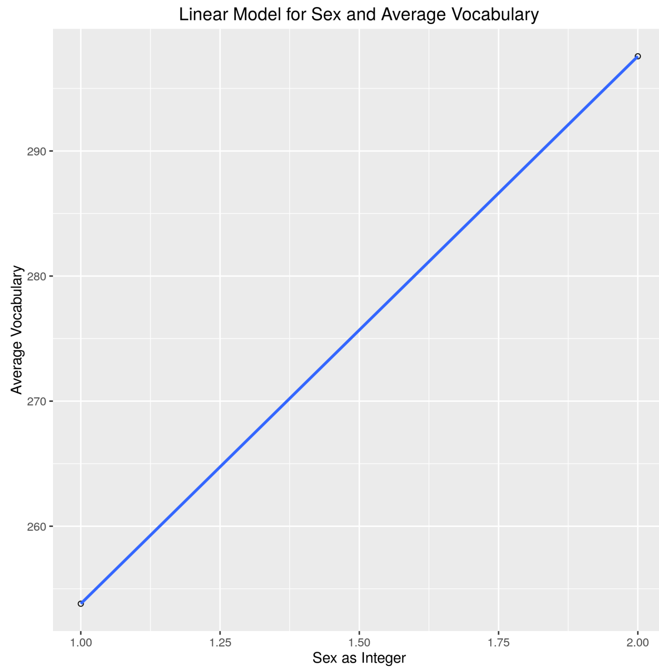
The results of the test prove our previous hypothesis: although there is some correlation between ethnicity and vocabulary size, R-squared value is ridiculously small to claim that there is a strong relationship between the two. Overall, one is not likely to accurately predict the child's vocabulary only knowing child's ethnicity.

Vocabulary and Sex

How does sex determine the extent of vocabulary development? We put this question to the test by using this regression model.

$$m_{S;V}(t) = \beta_0 + \beta_1 t_1$$

Figure 6: Sex and Vocabulary: Linear Regression Model



In other words:

$$\text{vocabulary size} = \beta_0 + \beta_1 \text{ sex}$$

A call to the function (referenced in Appendex A.8 gives us the following information.

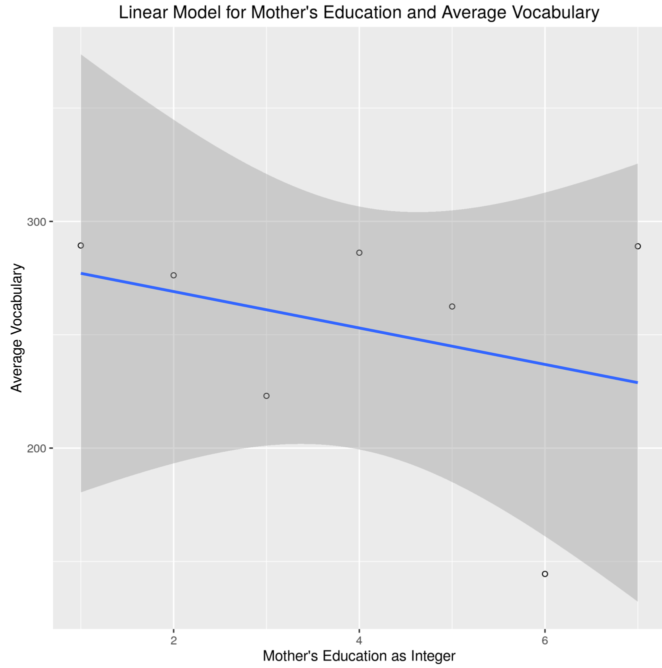
```
Sex:   Female
Mean:   297.5697
Standard Deviation:   208.9149
Sample Size:  1355
Standard error:   11.12367
95% Confidence Interval:   286.4461   ,   308.6934
Sex:   Male
Mean:   253.8045
Standard Deviation:   198.7591
Sample Size:  1386
Standard error:   10.4639
95% Confidence Interval:   243.3406   ,   264.2684
```

We can see that for the sample size we have, the standard deviation, and therefore the confidence interval, is fairly large. This suggests that we might not have a strong relationship between gender and vocabulary development.

Since we only had two choices, we end up with a very dichotomous variable. This is because we are dealing with a dummy variable, a qualitative, categorical value that is usually used to adjust regression models. Since there is a great difference in male and female vocabulary development, we should consider this variable when making a more encompassing regression model.

Vocabulary and Mother's Education

Figure 7: Mother's Education and Average Vocabulary Size: Linear Regression Model



We wonder if a mother's education level can affect her child's vocabulary development. This relationship is modeled as this linear regression function.

$$m_{M;V}(t) = \beta_0 + \beta_1 t_1$$

In other words:

$$\text{vocabulary size} = \beta_0 + \beta_1 \text{ education}$$

A call to the function education() (see A.8 gives us basic information about our data as follows.

```
Eductation:  Graduate
Mean:  289.3767
Standard Deviation:  214.425
Sample Size: 576
Standard error:  17.51105
95% Confidence Interval:  271.8657  ,  306.8878
Eductation:  College
Mean:  276.229
Standard Deviation:  204.778
Sample Size: 847
Standard error:  13.79081
95% Confidence Interval:  262.4382  ,  290.0199
Eductation:  Some Secondary
Mean:  223.0469
Standard Deviation:  185.5626
Sample Size: 128
Standard error:  32.14648
95% Confidence Interval:  190.9004  ,  255.1934
```

```
Eductation:  Secondary
Mean:  286.1986
Standard Deviation:  198.4454
Sample Size: 433
Standard error:  18.69155
95% Confidence Interval:  267.5071  ,  304.8902
Eductation:  Some College
Mean:  262.463
Standard Deviation:  202.3929
Sample Size: 594
Standard error:  16.27609
95% Confidence Interval:  246.1869  ,  278.7391
Eductation:  Primary
Mean:  144.5
Standard Deviation:  173.1869
Sample Size: 8
Standard error:  120.0102
95% Confidence Interval:  24.48978  ,  264.5102
Eductation:  Some Graduate
Mean:  289.0323
Standard Deviation:  206.1282
Sample Size: 155
Standard error:  32.45037
95% Confidence Interval:  256.5819  ,  321.4826
```

The large standard deviations suggest that the relationship between educationa and vocabulary is not as strong as we thought.

The results of the call to lm() can be found in Appendix A.7. The R^2 is 0.1063 and the Adjusted R^2 is -0.07246. This is a very low number, and makes us question whether or not there is a relationship between education and vocabulary.

Our plot in Figure /reffig:momV shows that there is a (positve/negative) relationship between a mother's education and average vocabulary. However, the data is very widely spread throughout the graph. Using the constants from our lm() readout, we have our final regression model:

$$m_{M;V}(t) = 7.34497 + -0.01322t_1$$

Our tests bring us to the conclusion that there is not a strong relationship between a mother's education and vocabulary development.

Vocabulary and Order of Birth

Our last variable for evaluation is the order of birth as an indicator for vocabulary development. The regression function we made for this relationship is as follows:

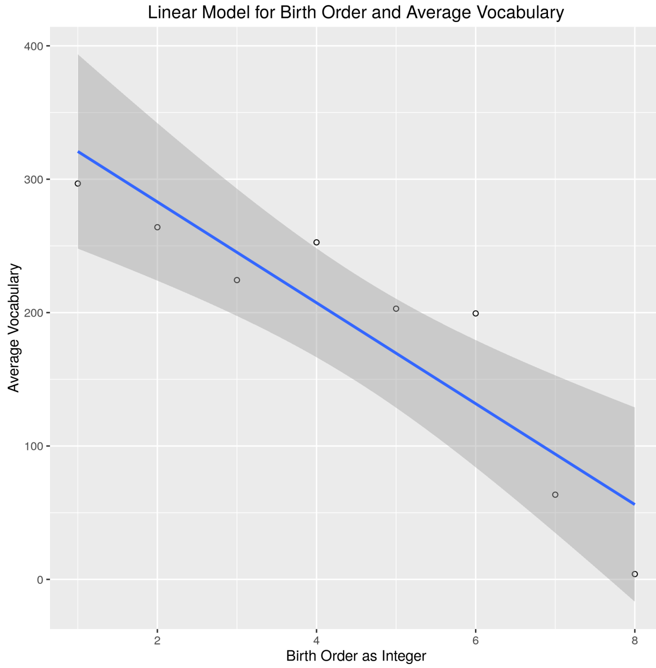
$$m_{B;V}(t) = \beta_0 + \beta_1 t_1$$

In other words:

$$\text{vocabulary size} = \beta_0 + \beta_1 \text{ order}$$

A call to the function BirthOrderVocab() (see Appendix A.8 for reference), gives us the following:

Figure 8: Order of Birth and Average Vocabulary Size: Linear Regression Model




```
Birth order:  First
Mean:  296.7671
Standard Deviation:  205.6228
Sample Size: 1417
Standard error:  10.70618
95% Confidence Interval:  286.0609  ,  307.4733
Birth order:  Second
Mean:  264.0164
Standard Deviation:  204.0315
Sample Size: 917
Standard error:  13.20568
95% Confidence Interval:  250.8107  ,  277.222
Birth order:  Third
Mean:  224.3915
Standard Deviation:  193.3608
Sample Size: 281
Standard error:  22.60806
95% Confidence Interval:  201.7834  ,  246.9995
Birth order:  Fourth
Mean:  252.6703
Standard Deviation:  207.8089
Sample Size: 91
Standard error:  42.69643
95% Confidence Interval:  209.9739  ,  295.3668
Birth order:  Fifth
Mean:  202.95
Standard Deviation:  151.392
```

```
Sample Size: 20
Standard error:  66.34923
95% Confidence Interval:  136.6008  ,  269.2992
Birth order:  Sixth
Mean:  199.4
Standard Deviation:  193.8552
Sample Size: 10
Standard error:  120.1505
95% Confidence Interval:  79.24954  ,  319.5505
Birth order:  Eighth
Mean:  4
Standard Deviation:  0
Sample Size: 1
Standard error:  0
95% Confidence Interval:  4  ,  4
Birth order:  Seventh
Mean:  63.5
Standard Deviation:  104.0336
Sample Size: 4
Standard error:  101.9511
95% Confidence Interval:  −38.4511  ,  165.4511
```

Most notably, most people don't have more than three siblings, so the standard deviations for the lower ranked children have disproportionately high standard deviations. However, for the more common ranks, the standard deviation is very low, suggesting a strong relationship between birth order and vocabulary.

The results of this model's call to lm() can be found in Appendix A.7. We see that there is an R^2 of 0.8248 and an adjusted R^2 of 0.7956. Compared to the other variables we've evaluated, this is a fairly high number.

Our plot in Figure /reffig:birthFig shows that there is a negative relationship between order of birth and average vocabulary. Using the constants from our lm() readout, we have our final regression model:

$$m_{B;V}(t) = 8.610472 + -0.021811t_1$$

Given the results provided by our tests, we can conclude that order of birth is a strong indicator of vocabulary size. If we should make a general regression model for vocabulary development, we should look to this variable as a critical factor.

Conclusion As a result of the tests, we can conclude that order of birth and age have the strongest relationship with vocabulary size while ethnicity and mother's education have the least influence. Gender has some influence on the the vocabulary size, but not as much as birth order or age.

A Appendix

A.1 Problem A General Functions

Code that is indented is for readability and where code would wraparound.

- The readData() function reads in the data from u.data.

```
readData <- function()
{
    #Extracts the data from the file u.data with the following header
    colNames <- c("UserID", "MovieID", "Rating", "Timestamp")
    uData <- read.table("u.data", col.names = colNames)

    return(uData)
}
```

- The readUser() function reads in u.user and returns the data in an appropriate data frame.

```
readUser <- function()
{
    #Extracts the data from the file u.user with the following header
    colNames <- c("UserID", "Age", "Gender", "Occupation", "ZipCode")
    uUser <- read.table("u.user", sep = "|", col.names = colNames)

    return(uUser)
}
```

- mergeDataUser() merges the two data frames returned by readUser() and readData() into one single data frame

```
mergeDataUser <- function()
{
    #Get the data frames for u.user and u.data
    data <- readData()
    user <- readUser()
    n <- intersect(names(data),names(user))
    #Merge them together where they intersect on headers
    merged <- merge(data, user, by=intersect(names(data), names(user)))
    #Write out to a file
    write.table(merged, file = "u.merged", sep = "|", row.names = FALSE)
    return(merged)
}
```

- getData() parses the information provided by mergeDataUser() into a useable data frame that is relevant to the problem. The data frame has the columns:
  UserID, Gender, MeanRatings

```
getData <- function()
{
    #Get merged data to find the mean rating
    userRatings <- mergeDataUser()
    #Read in users to subset it for UserID and Gender
    users <- readUser()
```

```
    #Split the data from userRatings and find the mean of the ratings
    # per user
    ratings <- split(userRatings$Rating, userRatings$UserID)
    Mean <- sapply(ratings, mean)
    #Subset the user data frame to only UserID and Gender, and
    # append on the Mean ratings per UserID
    A <- subset(users, select=c("UserID", "Gender"))
    A$Mean <- c(Mean)

    return(A)
}
```

- getDataNum() is similar to getData(), but instead of appending the Mean ratings per user ID, it returns a data frame with the full non-aggregated user ratings

```
getDataNum <- function()
{
    #Get merged data for all ratings
    userRatings <- mergeDataUser()
    #Get user data
    users <- readUser()
    #Subset the data by UserID, Gender, and Ratings
    A <- subset(userRatings, select=c("UserID", "Gender", "Rating"))

    return(A)
}
```

A.2 Problem A Confidence Interval Functions

- confIntMen() finds the approximate 95% confidence interval for the population mean rating by men. Utilizes the function getData() in A.1 and equation (10.14) from "From Algorithms to Z-Scores"

```
confIntMen <- function()
{
    #Get the data for Mean User Ratings
    A <- getData()
    #Get a subset of only males
    men <- A[A$Gender=='M',]

    #Xbar - sample mean
    sampleMean <- mean(men$Mean)
    #n - sample population
    samplePop <- nrow(men)
    #sigma - standard deviation
    stdd <- sd(men$Mean)
    #1.96*s.e(theta) = the standard error applied to 95%
    # interval
    #Uses the equation sigma / sqrt(n)
    error <- qnorm(0.975)*stdd/sqrt(samplePop)
    #Find the left and right ends of the interval
    # Xbar -+ error
    left <- sampleMean - error
    right <- sampleMean + error
```

```
    cat("Sample Mean: ", sampleMean, "\n")
    cat("Sample Population: ", samplePop, "\n")
    cat("St Dev: ", stdd, " Error: ", error, "\n")
    cat("Interval: ", left, ", ", right, "\n")

    return(1)
}
```

- confIntFemale() finds the approximate 95% confidence interval for the population mean rating by females. Utilizes the function getData() in A.1 and equation (10.14) from "From Algorithms to Z-Scores"

```
confIntFemale <- function()
{
    #Get the data for Mean User Ratings
    A <- getData()
    #Get a subset of only females
    female <- A[A$Gender=='F',]

    #Xbar - sample mean
    sampleMean <- mean(female$Mean)
    #n - sample population
    samplePop <- nrow(female)
    #sigma - standard deviation
    stdd <- sd(female$Mean)
    #1.96*s.e(theta) = the standard error applied to 95%
    # interval
    #Uses the equation sigma / sqrt(n)
    error <- qnorm(0.975)*stdd/sqrt(samplePop)

    #Find the left and right ends of the interval
    # Xbar -+ error
    left <- sampleMean - error
    right <- sampleMean + error

    cat("Sample Mean: ", sampleMean, "\n")
    cat("Sample Population: ", samplePop, "\n")
    cat("St Dev: ", stdd, " Error: ", error, "\n")
    cat("Interval: ", left, ", ", right, "\n")

    return(1)
}
```

- confIntDiff() finds the approximate 95% confidence interval for the population mean difference of the ratings of men and woman . Utilizes the function getData() in A.1 and equation (10.20) from "From Algorithms to Z-Scores"

```
confIntDiff <- function()
{
    #Get the data for Mean User Ratings
    A <- getData()
    #Get a subset of only males
    men <- A[A$Gender=='M',]
    #Get a subset of only females
```

```
    female <- A[A$Gender=='F',]

    #Xbar - sample mean for Males
    sampleMeanMen <- mean(men$Mean)
    #Ybar - sample mean for Females
    sampleMeanFemale <- mean(female$Mean)
    #Xbar - Ybar ; the difference of the sample means
    sampleMeanDiff <- abs(sampleMeanMen - sampleMeanFemale)
    #n_male - sample population of males
    samplePopM <- nrow(men)
    #n_female - sample population of females
    samplePopF <- nrow(female)
    #s_1 - standard deviation of male data
    stddM <- sd(men$Mean)
    #s_2 - standard deviation of female data
    stddF <- sd(female$Mean)
    #1.96*s.e(theta) = the standard error applied to 95%
    # interval
    #Uses the equation sqrt( ((s_1)^2 / n_male) + ((s_2)^2 / n_female) )
    error <- qnorm(0.975)* sqrt( (stddM^2 / samplePopM) +
        (stddF^2 / samplePopF) )
    #Apply the error to the intveral
    # Xbar - Ybar +- error
    left <- sampleMeanDiff - error
    right <- sampleMeanDiff + error

    cat("Sample Mean Male: ", sampleMeanMen, " Sample Mean Female: "
        , sampleMeanFemale, "\n")
    cat("Sample Population Male: ", samplePopM, " Sample Population Female: "
        , samplePopF, "\n")
    cat("St Dev Male: ", stddM, " St Dev Female: ", stddF, "\n")
    cat("Error: ", error, "\n")
    cat("Interval: ", left, ", ", right, "\n")

    return(error)
}
```

- confIntPopRat() finds the approximate 95% confidence interval for the difference bteween the population mean number of ratings by men and woman. Utilizes the functions mergeDataUser(), readUser(), and getDataNum() in A.1 and equation (10.20) from "From Algorithms to Z-Scores"

```
confIntPopRat <- function()
{
    #Get the full data to extra ratings
    A <- mergeDataUser()
    #Split ratings by UserID
    ratings <- split(A$Rating, A$UserID)
    #Find the number of ratings per user
    l <- sapply(ratings, length)
    #Get user data to parse out
    users <- readUser()
    #Parse the data by UserID and Gender
    TotRatPerUser <- subset(users, select=c("UserID", "Gender"))
    #Append on the number of ratings per userID
```

```
    TotRatPerUser$NumR <- c(1)
    #Get the full Data Set
    NonMeanA <- getDataNum()

    #Xbar - total number of male ratings
    m <- nrow(NonMeanA[NonMeanA$Gender == 'M',])
    #Ybar - total number of female ratings
    f <- nrow(NonMeanA[NonMeanA$Gender == 'F',])
    #Xbar - Ybar ; difference in female and male ratings
    sampleMeanDiff <- abs(m-f)
    #get the total number of ratings per User of males
    men <- TotRatPerUser[TotRatPerUser$Gender=='M',]
    #get the total number of ratings per User of females
    female <- TotRatPerUser[TotRatPerUser$Gender=='F',]
    #n_1 - sample population of men
    samplePopM <- nrow(men)
    #n_2 - sample  population of female
    samplePopF <- nrow(female)
    #s_1 - standard deviation of the number of ratings by men
    stddM <- sd(men$NumR)
    #s_2 - standard deviation of the number of ratings be women
    stddF <- sd(female$NumR)

    #1.96*s.e(theta) = the standard error applied to 95%
    # interval
    #Uses the equation sqrt( ((s_1)^2 / n_1) + ((s_2)^2 / n_2) )
    error <- qnorm(0.975) * sqrt( (stddM^2 / samplePopM) +
        (stddF^2 / samplePopF) )
    #Apply the error to the intveral
    # Xbar - Ybar +- error
    left <- sampleMeanDiff - error
    right <- sampleMeanDiff + error

    cat("Sample Mean Male: ", m, " Sample Mean Female: ", f, "\n")
    cat("Sample Population Male: ", samplePopM, " Sample Population Female: "
        , samplePopF, "\n")
    cat("St Dev Male: ", stddM, " St Dev Female: ", stddF, "\n")
    cat("Error: ", error, "\n")
    cat("Interval: ", left, ", ", right, "\n")
    return(error)
}
```

- confIntPropMale() finds an approximate 95% confidence interval for the population proportiong of users who are male. Utilizes the function getData() in A.1

```
confIntPropMale <- function()
{
    #Get the data mean ratings
    A <- getData()
    #Split the data into men and female frames
    men <- A[A$Gender=='M',]
    female <- A[A$Gender=='F',]
    #sample populations of men and women
    samplePopM <- nrow(men)
    samplePopF <- nrow(female)
    #n is the total sample population
```

```
    n <- samplePopM + samplePopF
    #p is the sample population of men divided by the total
    p <- samplePopM / n
    #will use
    #1.96*s.e(theta) = the standard error applied to 95%
    # interval
    #Uses the equation sqrt( p*(1-p) / n ) where p = # men and n = totalPop
    error <- qnorm(0.975) * sqrt( ( p * (1 - p) ) / n)
    #Apply the error to the intveral
    # p +- error
    left <- p - error
    right <- p + error

    cat("Sample Mean: ", p, "\n")
    cat("Sample Population: ", n, "\n")
    cat("Error: ", error, "\n")
    cat("Interval: ", left, ", ", right, "\n")
}
```

A.3 Problem A Histogram Functions

- histo() produces two historgrams showing the average user rating for Males (Figure 2) and Females (Figure 3). Utilizes the function getData() in A.1 and the ggplot2 library.

```
histo <- function()
{
    #Get the data of mean ratings
    A <- getData()
    #Split into female and male subsets
    men <- A[A$Gender=='M',]
    female <- A[A$Gender=='F',]

    #Produce histograms based off the average ratings per females and males
    malePlot <- ggplot() + aes(men$Mean) +
        geom_histogram(binwidth = 0.5, colour = "black",
        fill = "dodgerblue3") + labs(title = "Mean Male Movie Ratings",
        x = "Mean Ratings", y = "Frequency")

    femalePlot <- ggplot() + aes(female$Mean) +
        geom_histogram(binwidth = 0.5, colour = "black",
        fill = "indianred3") + labs(title = "Mean Female Movie Ratings",
        x = "Mean Ratings",y = "Frequency")

    #Save to the respective files maleHistogram.png and femaleHistrogram.png
    ggsave(malePlot, file="maleHistogram.png")
    ggsave(femalePlot, file="femaleHistogram.png")
}
```

A.4 Problem A Hypothesis Testing Functions

- hypothe() finds Z for use in the hypothesis test that the female and male population means are equal. Utilizes the function getData() in A.1 and equation (11.6) from "From Algorithms to Z-Scores"

```
hypothe <- function()
{
    #Get data of mean ratings
    A <- getData()
    #Seperate into male and female groups
    men <- A[A$Gender=='M',]
    female <- A[A$Gender=='F',]
    #mu0 - Hypothesis mean
    sampleMeanMen <- mean(men$Mean)
    #Xbar - True mean
    sampleMeanFemale <- mean(female$Mean)
    xbar <- sampleMeanFemale
    mu0 <- sampleMeanMen
    #sigma - standard deviation of ratings of men
    sigma <- sd(men$Mean)
    #Number of men
    n <- nrow(men)
    #Uses equation 11.6
    z <- (xbar - mu0)/(sigma/sqrt(n))

    return(z)
}
```

A.5 Problem A Linear Model Functions

- linearMod() is used for A.h, returning both the estimated ratings from age and gender, and also the estimated ratings from women of age 28. The estimation of ratings from age and gender are plotted and can be seen in Figure 1. Utilizies the functions mergeDataUser() and readUser() in A.1, and the ggplot2 library.

```
linearMod <- function()
{
    #Get the raw data of the merged Users and Data
    userRatings <- mergeDataUser()
    #Get the raw data of users because we need age
    users <- readUser()
    #Split the ratings by UserID
    ratings <- split(userRatings$Rating, userRatings$UserID)
    #Create the data frame A consisting of UserID, Gender, and Age
    A <- subset(users, select=c("UserID", "Gender", "Age"))
    #Find the mean rating per User
    Mean <- sapply(ratings, mean)
    #Append the last value onto our data frame
    A$Mean <- c(Mean)
    #Convert gender to binary 0 and 1 indicator variables
    A$Gen <- as.numeric(A$Gender == 'M')

    #Data frame B is used for A.h,
    # finding the mean average ratings for females
    # of age 28
    B <- A[A$Gender=='F',]
    B <- B[B$Age==28,]

    #Call lm (linear model) to estimate mean ratings from gender and age
```

```
    sum <- summary(lm(A$Mean ~ A$Gen + A$Age))
    #Call lm (linear model) to estimate mean ratings for women of age 28
    womanAge <- summary(lm(B$Mean ~ B$Gen + B$Age))
    #Save both to a file for easy copy-pasting
    capture.output(sum, file = "reg.data")
    capture.output(womanAge, file = "woman.dat")

    #Plot the scatter plot and linear model for estimating
    # mean ratings from gender and age using GGplot
    plot <- ggplot() + aes(x=A$Gen+A$Age, y=A$Mean) + geom_point(shape = 1)
        + geom_smooth(method=lm, se=TRUE) +
        labs(title = "Linear Model for Rating",
        y = "Mean Estimated Rating", x = "Age + Gender")
    ggsave(plot, file="regression.png")

    return(1)
}
```

A.6 Problem A Linear Model Output

```
Call:
lm(formula = B$Mean ~ B$Gen + B$Age)

Residuals:
    Min      1Q  Median      3Q     Max
-0.7277 -0.1991  0.1031  0.1596  0.8220

Coefficients: (2 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.5702     0.1407   25.37 1.11e-09 ***
B$Gen             NA         NA      NA       NA
B$Age             NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.445 on 9 degrees of freedom
```

A.7 Problem B Linear Model Outputs

Linear Model for Age and Vocabulary Size

Finds the linear regression model for child's age and vocabulary size.

```
Call:
lm(formula = dataB$age ~ dataB$vocab)

Residuals:
     Min       1Q   Median       3Q      Max
-12.5689  -2.2638  -0.3796   1.9465  11.8189

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.793e+01  6.634e-02  270.32   <2e-16 ***
dataB$vocab 1.654e-02  1.939e-04   85.32   <2e-16 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.097 on 5496 degrees of freedom
Multiple R-squared:  0.5698,     Adjusted R-squared:  0.5697
F-statistic:  7279 on 1 and 5496 DF,  p-value: < 2.2e-16
```

Linear Model For Birth Order and Mean Vocabulary Size

Finds the linear regression model for child's birth order and average vocabulary size.

```
Call:
lm(formula = births ~ means)

Residuals:
    Min      1Q  Median      3Q     Max
-1.1378 -0.7503 -0.3744  0.8371  1.7386

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.610472   0.866818   9.933 6.02e-05 ***
means       -0.021811   0.004104  -5.315   0.0018 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.107 on 6 degrees of freedom
Multiple R-squared:  0.8248,     Adjusted R-squared:  0.7956
F-statistic: 28.25 on 1 and 6 DF,  p-value: 0.001804
```

Linear Model for Mother's Education and Child's Vocabulary Size

Finds the linear regression model for mother's education level and child's average vocabulary size.

```
Call:
lm(formula = int_edu ~ means)

Residuals:
      1       2       3       4       5       6       7
-2.5187 -1.6926 -1.3958  0.4393  1.1254  0.5657  3.4767

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.34497    4.41942   1.662    0.157
means       -0.01322    0.01715  -0.771    0.475

Residual standard error: 2.237 on 5 degrees of freedom
Multiple R-squared:  0.1063,     Adjusted R-squared:  -0.07246
F-statistic: 0.5946 on 1 and 5 DF,  p-value: 0.4755
```

Linear Model for Ethnicity and Vocabulary Size

Finds the linear regression model for child's ethnicity and average vocabulary size.

```
Call:
lm(formula = int_eth ~ means)

Residuals:
      1       2       3       4       5
-1.7653 -0.3271 -0.7293  1.6503  1.1713

Coefficients:
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.32427    7.01585   1.329    0.276
means       -0.02485    0.02742  -0.906    0.432

Residual standard error: 1.618 on 3 degrees of freedom
Multiple R-squared:  0.2149,	Adjusted R-squared:  -0.04676
F-statistic: 0.8213 on 1 and 3 DF,  p-value: 0.4316
```

Linear Model for Sex and Vocabulary Size
Finds the linear regression model for child's gender and average vocabulary size.

```
Call:
lm(formula = int_sex ~ means)

Residuals:
ALL 2 residuals are 0: no residual degrees of freedom!

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.79922         NA      NA       NA
means        0.02285         NA      NA       NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,	Adjusted R-squared:    NaN
F-statistic:   NaN on 1 and 0 DF,  p-value: NA
```

A.8 Problem B Functions

- readDataB() reads in a select amount of data from the set of data vocabulary_norms_data.csv and organizes this into a large matrix. The function omits rows with missing information, or in other rows containing at least one "NA"

```
readDataB <- function()
{
  bank <- read.csv(file="vocabulary_norms_data.csv", head=TRUE, sep=",")[,
        c('age', 'birth_order', 'ethnicity', 'sex', 'mom_ed', 'vocab')]
  clean <- bank[complete.cases(bank), ]
  return(clean)
}
```

- AgeVocab() finds the mean, standard deviation, standard error, and 95% confidence interval for age and children's vocabulary. After, it finds and plots the linear regression model for age and vocabulary size.

```
AgeVocab <-function()
{
  dataB <- readDataB()
  #Mean, standard deviation and sample size for age and vocabulary
  mean_age <- mean(dataB$age)
  mean_vocab <- mean(dataB$vocab)
  cat("Average age", mean_age, "\n")
  cat("Average vocabulary", mean_vocab, "\n")
  sd_age <- sd(dataB$age)
  sd_vocab <- sd(dataB$vocab)
```

```
    cat("Standard deviation: age", sd_age, "\n")
    cat("Standard deviation: vocabulary", sd_vocab, "\n")
    size_age <- length(dataB$age)
    size_vocab <- length(dataB$vocab)
    cat("Sample size: age", size_age, "\n")
    cat("Sample size: vocabulary", size_vocab, "\n")
    # COnfidence interval for age
    error_age  <- qnorm(0.975)*sd_age/sqrt(size_age)
    left_age <- mean_age - error_age
    right_age<- mean_age + error_age
    cat("Standard error: age", error_age, "\n")
    cat("95\% Confidence Interval", left_age, " , ",
      right_age, "\n")
    # Confidence interval for vocabulary
    error_vocab  <- qnorm(0.975)*sd_vocab/sqrt(size_vocab)
    left_vocab <- mean_vocab - error_vocab
    right_vocab<- mean_vocab + error_vocab
    cat("Standard error: vocabulary", error_vocab, "\n")
    cat("95\% Confidence Interval", left_vocab, " , ",
      right_vocab, "\n")

    #linear model to find correlation between age and vocabulary
    sum <- summary(lm(dataB$age ~ dataB$vocab))
    capture.output(sum, file = "age_vocab.data")
    AgeVocab <- ggplot() + aes(x = dataB$age, y = dataB$vocab)
      + geom_point(shape = 1) + geom_smooth(method = lm, se=TRUE)
      + labs(title = "Linear Model for  Age and Vocabulary",
      x = "Age", y = "Vocabulary")
    ggsave(AgeVocab, file="AgeVocab.png")
}
```

- BirthOrderVocab() is designed to explore the relationship between the birth order and child's vocabulary size. The function finds the mean, standard deviation, standard error, and 95% confidence interval for vocabulary for each birth order, then finds the linear regression model and plots it.

```
BirthOrderVocab <- function()
{
    dataB <- readDataB()
    m <- length(dataB$age)
    #extract unique values
    n <- length(unique(dataB$birth_order))
    orders <- (unique(dataB$birth_order))
    birth <- as.vector(orders)
    #cat("orders" , orders)
    means <- vector( , n)
    sds <- vector( ,n )
    error <- vector( ,n)
    ls <- vector( ,n)
    rs <- vector( ,n)
    sizes <- vector( ,n)
    for(i in 1:n)
    #first <- dataB[dataB$birth_order == 'First', ]
    {
    order <- dataB[(dataB$birth_order == birth[i]), ]
```

```
cat("Birth order: ",  birth[i], "\n")
if(length(order$vocab) > 1)
{
means[i] <- mean(order$vocab)
cat("Mean: ", means[i], "\n")
sds[i] <- sd(order$vocab)
cat("Standard Deviation: ", sds[i], "\n")
sizes[i] <- length(order$vocab)
cat("Sample Size:", sizes[i], "\n")
error[i] <-  qnorm(0.975)*sds[i]/sqrt(sizes[i])
cat("Standard error: ", error[i], "\n")
ls[i] <- means[i] - error[i]
rs[i] <- means[i] + error[i]
cat("95\% Confidence Interval: ", ls[i], " , ", rs[i], "\n")
}
else
{
means[i] <- order$vocab
sizes[i] <-1
sds[i] <- 0
error[i] <- 0
ls[i] <- means[i]
rs[i] <- means[i]
cat("Mean: ", means[i], "\n")
cat("Standard Deviation: ", sds[i], "\n")
cat("Sample Size:", sizes[i], "\n")
cat("Standard error: ", error[i], "\n")
cat("95% Confidence Interval: ", ls[i], " , ", rs[i], "\n")
}
}
births<- vector( , n)
for(j in 1:n)
{
if(birth[j] == 'First')
 births[j] = 1
if(birth[j] == 'Second')
 births[j] = 2
if(birth[j] == 'Third')
 births[j] = 3
if(birth[j] == 'Fourth')
 births[j] = 4
if(birth[j] == 'Fifth')
 births[j] = 5
if(birth[j] == 'Sixth')
 births[j] = 6
if(birth[j] == 'Seventh')
 births[j] = 7
if(birth[j] == 'Eighth')
 births[j] = 8
}
sum <- summary(lm(births ~ means))
capture.output(sum, file = "birth_mean_vocab.data")

means_order <- ggplot() + aes(x = births, y = means)
    + geom_point(shape = 1) + geom_smooth(method = lm, se=TRUE)
```

```
        + labs(title = "Linear Model for Birth Order
            and Average Vocabulary",
        x = "Birth Order as Integer", y = "Average Vocabulary")
    ggsave(means_order, file="means_orders.png")
}
```

- ethnicity_vocab() is designed to explore the relationship between the ethnicity of the child and child's vocabulary size. The function finds the mean, standard deviation, standard error, and 95% confidence interval for vocabulary for each ethnicity, then finds the linear regression model and plots it.

```
ethnicity_vocab <- function()
{
    dataB <- readDataB()
    #extract unique values
    n <- length(unique(dataB$ethnicity))
    eth <- (unique(dataB$ethnicity))
    ethn <- as.vector(eth)
    #cat("orders" , orders)
    means <- vector( , n)
    sds <- vector( ,n )
    error <- vector( ,n)
    ls <- vector( ,n)
    rs <- vector( ,n)
    sizes <- vector( ,n)
    for(i in 1:n)
    {
    order <- dataB[(dataB$ethnicity == ethn[i]), ]
    cat("Ethnicity: ", ethn[i], "\n")
    if(length(order$vocab) > 1)
    {
    means[i] <- mean(order$vocab)
    cat("Mean: ", means[i], "\n")
    sds[i] <- sd(order$vocab)
    cat("Standard Deviation: ", sds[i], "\n")
    sizes[i] <- length(order$vocab)
    cat("Sample Size:", sizes[i], "\n")
    error[i] <- qnorm(0.975)*sds[i]/sqrt(sizes[i])
    cat("Standard error: ", error[i], "\n")
    ls[i] <- means[i] - error[i]
    rs[i] <- means[i] + error[i]
    cat("95\% Confidence Interval: ", ls[i], " , ", rs[i], "\n")
    }
    else
    {
    means[i] <- order$vocab
    sizes[i] <-1
    sds[i] <- 0
    error[i] <- 0
    ls[i] <- means[i]
    rs[i] <- means[i]
    cat("Mean: ", means[i], "\n")
    cat("Standard Deviation: ", sds[i], "\n")
    cat("Sample Size:", sizes[i], "\n")
    cat("Standard error: ", error[i], "\n")
```

```
    cat("95\% Confidence Interval: ", ls[i], " , ", rs[i], "\n")
    }
    int_eth <- vector( ,n)
    for(j in 1:n)
    {
    if(ethn[j] == 'Asian')
     int_eth[j] = 1
    if(ethn[j] == 'Black')
     int_eth[j] = 2
    if(ethn[j] == 'Other')
     int_eth[j] = 3
    if(ethn[j] == 'White')
     int_eth[j] = 4
    if(ethn[j] == 'Hispanic')
     int_eth[j] = 5
    }
    sum <- summary(lm(int_eth ~ means))
    capture.output(sum, file = "ethnicity_mean_vocab.data")

    means_order <- ggplot() + aes(x =int_eth , y = means) + geom_point(shape = 1)
        + geom_smooth(method = lm, se=TRUE)
        + labs(title = "Linear Model for Ethnicity
             and Average Vocabulary",
        x = "Ethnicity as Integer", y = "Average Vocabulary")
    ggsave(means_order, file="means_ethn.png")

    }
}
```

- education() is designed to explore the relationship between mother's education level and child's vocabulary size. The function finds the mean, standard deviation, standard error, and 95% confidence interval for vocabulary for each education level, then finds the linear regression model and plots it.

```
education <- function()
{
    dataB <- readDataB()
    #extract unique values
    n <- length(unique(dataB$mom_ed))
    ed <- (unique(dataB$mom_ed))
    edu <- as.vector(ed)
    #cat("orders" , orders)
    means <- vector( , n)
    sds <- vector( ,n )
    error <- vector( ,n)
    ls <- vector( ,n)
    rs <- vector( ,n)
    sizes <- vector( ,n)
    for(i in 1:n)
    {
    order <- dataB[(dataB$mom_ed == edu[i]), ]
    cat("Eductation: ", edu[i], "\n")
    if(length(order$vocab) > 1)
    {
    means[i] <- mean(order$vocab)
```

```
    cat("Mean: ", means[i], "\n")
    sds[i] <- sd(order$vocab)
    cat("Standard Deviation: ", sds[i], "\n")
    sizes[i] <- length(order$vocab)
    cat("Sample Size:", sizes[i], "\n")
    error[i] <-  qnorm(0.975)*sds[i]/sqrt(sizes[i])
    cat("Standard error: ", error[i], "\n")
    ls[i] <- means[i] - error[i]
    rs[i] <- means[i] + error[i]
    cat("95\% Confidence Interval: ", ls[i], " , ", rs[i], "\n")
    }
    else
    {
    means[i] <- order$vocab
    sizes[i] <-1
    sds[i] <- 0
    error[i] <- 0
    ls[i] <- means[i]
    rs[i] <- means[i]
    cat("Mean: ", means[i], "\n")
    cat("Standard Deviation: ", sds[i], "\n")
    cat("Sample Size:", sizes[i], "\n")
    cat("Standard error: ", error[i], "\n")
    cat("95\% Confidence Interval: ", ls[i], " , ", rs[i], "\n")
    }
    }
    int_edu <- vector( , n)
    for(j in 1:n)
    {
    if(edu[j] == "Graduate")
    int_edu[j] = 1
    if(edu[j] == "College")
    int_edu[j] = 2
    if(edu[j] == "Some Secondary")
    int_edu[j] = 3
    if(edu[j] == "Secondary")
    int_edu[j] = 4
    if(edu[j] == "Some College")
    int_edu[j] = 5
    if(edu[j] == "Primary")
    int_edu[j] = 6
    if(edu[j] == "Some Graduate")
    int_edu[j] = 7
    }
    sum <- summary(lm(int_edu ~ means))
    capture.output(sum, file = "edu_mean_vocab.data")

    means_order <- ggplot() + aes(x =int_edu, y = means)
        + geom_point(shape = 1) + geom_smooth(method = lm, se=TRUE)
        + labs(title = "Linear Model for Mother's
            Education and Average Vocabulary",
        x = "Mother's Education as Integer", y = "Average Vocabulary")
    ggsave(means_order, file="means_edu.png")
}
```

- gender_vocab() is designed to explore the relationship between the child's genfer and child's vocabulary size. The function finds the mean, standard deviation, standard error, and 95% confidence interval for vocabulary for each gender, then finds the linear regression model and plots it.

```
gender_vocab <- function()
{
    dataB <- readDataB()
    #extract unique values
    n <- length(unique(dataB$sex))
    sex <- (unique(dataB$sex))
    gender <- as.vector(sex)
    #cat("orders" , orders)
    means <- vector( , n)
    sds <- vector( ,n )
    error <- vector( ,n)
    ls <- vector( ,n)
    rs <- vector( ,n)
    sizes <- vector( ,n)
    for(i in 1:n)
    {
    order <- dataB[(dataB$sex == gender[i]), ]
    cat("Sex: ",  gender[i], "\n")
    if(length(order$vocab) > 1)
    {
    means[i] <- mean(order$vocab)
    cat("Mean: ", means[i], "\n")
    sds[i] <- sd(order$vocab)
    cat("Standard Deviation: ", sds[i], "\n")
    sizes[i] <- length(order$vocab)
    cat("Sample Size:", sizes[i], "\n")
    error[i] <-  qnorm(0.975)*sds[i]/sqrt(sizes[i])
    cat("Standard error: ", error[i], "\n")
    ls[i] <- means[i] - error[i]
    rs[i] <- means[i] + error[i]
    cat("95% Confidence Interval: ", ls[i], " , ", rs[i], "\n")
    }
    else
    {
    means[i] <- order$vocab
    sizes[i] <-1
    sds[i] <- 0
    error[i] <- 0
    ls[i] <- means[i]
    rs[i] <- means[i]
    cat("Mean: ", means[i], "\n")
    cat("Standard Deviation: ", sds[i], "\n")
    cat("Sample Size:", sizes[i], "\n")
    cat("Standard error: ", error[i], "\n")
    cat("95% Confidence Interval: ", ls[i], " , ", rs[i], "\n")
    }
    }

    int_sex <- vector( , n)
    for(j in 1:n)
    {
```

```
    if(gender[j] == 'Male')
     int_sex[j] = 1
    if(gender[j] == 'Female')
     int_sex[j] = 2
    }

    sum <- summary(lm(int_sex ~ means))
    capture.output(sum, file = "sex_mean_vocab.data")
}
```

A.9 Who Did What

- Tyler:
  - Problem A: Full Writeup, Math, Code, Problem A Appendix
  - Problem B: Appendix
  - Latex: General Structure
- Anastasia:
  - Problem B: Writeup, Code, Appendix
- Kim:
  - Problem A: Writeup Proofreading
  - Problem B: Writeup, Code