Lab2

Megan Jasek, Rohan Thakur, Charles Kekeh

Friday, March 04, 2016

Question 1

Part 1

$$E(Y|X) = \int_0^x y * \frac{1}{x}\ dy = \frac{y^2}{2x}|_0^x = \frac{x}{2} - 0$$
$$\mathbf{E(Y|X) = \frac{x}{2}}$$

Part 2

$$E(Y) = E(E(Y|X)) = E(\frac{x}{2}) = \frac{1}{2}E(x)$$

We know that,

$$f_X(x) = 1$$

Find E(X) as follows,

$$E(X) = \int_0^1 x * f_X(x)\ dx = \int_0^1 x * 1\ dx = \frac{x^2}{2}|_0^1 = \frac{1}{2} - 0 = \frac{1}{2}$$

Substituting in for E(x) we get,

$$E(Y) = \frac{1}{2} * \frac{1}{2} = \frac{1}{4}$$
$$\mathbf{E(Y) = \frac{1}{4}}$$

Part 3

$$f_{X,Y}(x, y) = f_{Y|X}(y|x) * f_X(x)$$

We know that

$$f_{Y|X}(y|x) = \frac{1}{x}\ and\ f_X(x) = 1$$

Substituting these values in to the equation, we get

$$\mathbf{f_{X,Y}(x, y) = \frac{1}{x}}$$

Part 4

$$f_Y(y) = \int_y^1 f_{Y|X}(y|x) * f_X(x)dx = \int_y^1 \frac{1}{x} * 1 \ dx$$

$$= \log(x)|_y^1 = \log(1) - \log(y) = 0 - \log(y) = \log(\frac{1}{y})$$

$$f_Y(y) = \log(\frac{1}{y})$$

We know that

$$f_{X,Y}(x,y) = f_{X|Y}(x|y) * f_Y(y)$$

Solving for $f_{X|Y}(x|y)$, we get

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$$

Substituting, we get

$$f_{X|Y}(x|y) = \frac{\frac{1}{x}}{\log(\frac{1}{y})}$$

$$\mathbf{f_{X|Y}(x|y) = \frac{1}{x\log(\frac{1}{y})}}$$

Part 5

$$E(X|Y = \frac{1}{2}) = \int_{\frac{1}{2}}^1 x * \frac{1}{x\log(2)} dx = \frac{1}{\log(2)} \int_{\frac{1}{2}}^1 1 \ dx$$

$$= \frac{1}{\log(2)} * (x|_{\frac{1}{2}}^1) = \frac{1}{\log(2)} * (1 - \frac{1}{2})$$

$$= \frac{1}{\log(2)} * \frac{1}{2} = \frac{1}{2\log(2)}$$

$$\mathbf{E(X|Y = \frac{1}{2}) = \frac{1}{2\log(2)}}$$

Question 2

$$Payoff \ function = aA + bB + cC$$

Let us calculate the variance of the payoff.

$$Var(Payoff) = Var(aA + bB + cC)$$
$$= a^2Var(A) + b^2Var(B) + c^2Var(C)$$

Since A, B an C are independent, all covariance terms are 0.
Now, using the relation Var(A)=2Var(B)=3Var(C):

$$= 6a^2Var(C) + \frac{3}{2}b^2Var(C) + c^2Var(C)$$

We can clearly see from this equation, that in order to minimize variance, all the allocation must be in asset C, since any allocation in A or B, leads to a higher variance than the same allocation in C.
Final answer: (a,b,c) = (0,0,1)

Question 3

$y_i, i = 1, \cdots, n$ random uniform variables.

Part 1 - Likelihood Function

$L(\theta)$ being the likelihood function, we know we have:

$$L(\theta) = f(y_1, \cdots, y_n|\theta) = f(y_1|\theta)f(y_2|\theta)\cdots f(y_n|\theta)$$

Where f is the uniform probablity density function with parameter θ.

$$f(y_i, \theta) = \begin{cases} \frac{1}{\theta} & \text{for } 0 \leq y_i \leq \theta \\ 0 & \text{otherwise} \end{cases}$$

Making

$$L(\theta) = \begin{cases} \frac{1}{\theta^n} & \text{for } 0 \leq y_i \leq \theta, i \in 1, \cdots, n \\ 0 & \text{otherwise} \end{cases}$$

Part 2 - MLE

Based on $L(\theta)$ The MLE of θ is a value of θ for which $\theta \geq y_i fori \in 1, \cdots, n$ and which maximizes $1/\theta^n$. $MLE(\theta)$ is the smallest of such values of θ such that $\theta \geq y_i$ for $i \in 1, \cdots, n$. Therefore:

$$\mathbf{MLE}(\theta) = \hat{\theta} = \mathbf{max}(\mathbf{y_1}, \cdots, \mathbf{y_n})$$

Part 3 - Expectation n=1

Taking $\hat{\theta} = max(y_1, \cdots, n)$ and n=1
We have

$$\hat{\theta} = y_1$$

And

$$\mathbf{E}[\hat{\theta}] = \mathbf{E}[\mathbf{y_1}] = \frac{\theta}{\mathbf{2}}$$

Knowing that y_i is from a random uniform distribution over $[0, \theta]$

Part 4 - Bias

Yes, from the above, $\hat{\theta}$ is biased. For any $y_1, \cdots, y_n$, we expect $maxy_1, \cdots, n < \theta$ with probability 1. Hence $\hat{\theta}$ underestimates θ and we have just proven that for n=1, $E(\hat{\theta}) \neq \theta$.

Part 5 - Expectation general case

Taking $\hat{\theta} = max(y_1, \cdots, y_n)$ and assuming $n \geq 1$.

$$E(\hat{\theta}) = E[max(y_1, \cdots, y_n)]$$

Let's define $x = max(y_i), i \in 1, \cdots, n$.

$$CDF(x) = P(max(y_i, \cdots, y_n) < x), i \in 1, \cdots, n$$

$$CDF(x) = P(y_1 < x, y_2 < x, \cdots, y_n < x)$$
$$CDF(x) = \prod P(y_i < x), i \in 1, \cdots, n$$
$$CDF(x) = (\frac{x}{\theta})^n$$

From CDF(x), which is the cumulative distribution of x, we deterne the desnsity probability as

$$PDF(x) = \frac{\delta}{\delta x}(\frac{\theta}{x})^n$$

$$PDF(x) = \frac{nx^{n-1}}{\theta^n}$$

From PDF(x), we can now compute E(x) as:

$$E(x) = \int_{x=0}^{\theta} \frac{n(x^{n-1})}{\theta} x dx$$

and

$$\mathbf{E(x)} = \hat{\theta} = \frac{\mathbf{n}}{\mathbf{n+1}}\theta$$

Part 6 - Expectation general case

From the previous compututation of the general case of $n \geq 1$, we can state that

$$\lim_{\mathbf{n} \to \infty} \hat{\theta} = \theta$$

and $\hat{\theta}$ is a consistent estimator of θ.

Question 4

4.1 Univariate Analysis

- **wage** - The wage variable has a range from \$127 to \$2,404 with a mean of \$579 and median of \$543 with most values occuring between \$250 and \$750. The histogram shows a data distribution that's positevely skewed.
- **logWage** - The logWage variable has a range from \$4.844 to \$7.785 with a mean of \$6.263 and median of \$6.297. The histogram shows a data distribution that's approximately normal.
- **education** - The education variable is an integer and has a range from 2 to 18 with a mean of 12 and median of 12. The histogram shows a data distribution that is slightly negatively skewed. There is a spike at 12 and a smaller spike at 16.
- **experience** - The experience variable is an integer and has a range from 0 to 23 with a mean of 8.788 and median of 8. The histogram shows a data distribution that is slightly positively skewed.
- **experienceSquare** - The experience variable is an integer and has a range from 0 to 529 with a mean of 95.03 and median of 64. The histogram shows a data distribution that is positively skewed. There is a spike at about 50.
- **IQscore** - The IQscore variable is an integer and has a range from 50 to 144 with a mean of 102.3 and median of 103. The histogram shows a data distribution that is approximately normal. There are 316 missing values.

- **dad_education** - The dad_education variable is an integer and has a range from 0 to 18 with a mean of 10.18 and median of 11. The histogram shows a data distribution that has many frequencies at about count 30 and spikes at 8 and 12. These spikes make intuitive sense because these are natural educaiton breakpoints for people. Eight years signifying the end of middle school and 12 years indicating the end of high school. There are 239 missing values.
- **mom_education** - The mom_education variable is an integer and has a range from 0 to 18 with a mean of 10.45 and median of 12. The histogram shows a data distribution that has many frequencies at about count 50 and spikes at 12. This spike makes intuitive sense because 12 years indicates the end of high school which is a natural education break point for people. There are 128 missing values.
- **age** - The age variable is an integer and has a range from 24 to 34 with a mean of 28.01 and median of 27. For the ages between 24 and 28, the fequency is around 105. For the ages between 29 and 34, the frequency is around 65.
- **raceColor** - The raceColor variable is a binary variable with values 0 or 1 and mean 0.238. This means that there are about 24% 1's and 76% 0's.
- **rural** - The rural variable is a binary variable with values 0 or 1 and mean 0.391. This means that there are about 39% 1's and 61% 0's. 39% of the participants live in a rural area and 61% do not.
- **city** - The rural variable is a binary variable with values 0 or 1 and mean 0.712. This means that there are about 71% 1's and 29% 0's. 71% of the participants live in a city and 29% do not.
- **z1** - The z1 variable is a binary variable with values 0 or 1 and mean 0.44. This means that there are about 44% 1's and 56% 0's.
- **z2** - The z2 variable is a binary variable with values 0 or 1 and mean 0.686. This means that there are about 69% 1's and 31% 0's.

```r
# Load the data in to the df dataframe
data = read.csv("WageData2.csv", header = TRUE)
# There was already a logWage variable in the dataset, so set that one
# to logWageOLD
data$logWageOLD = data$logWage
# Create a logWage variable to use for the rest of the problem
data$logWage = log(data$wage)
# Create the experienceSquare variable
data$experienceSquare = data$experience * data$experience
```

```r
# wage variable
summary(data$wage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   127.0   400.0   543.0   578.8   702.5  2404.0
```

```r
print(quantile(data$wage, probs = c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9,
    0.95, 0.99, 1)))
```

```
##      1%      5%     10%     25%     50%     75%     90%     95%     99%
##  187.92  244.90  289.00  400.00  543.00  702.50  914.00 1068.70 1402.23
##    100%
## 2404.00
```

```r
# Plot the histogram of apps at 30 bins
wage.hist <- ggplot(data, aes(wage)) + theme(legend.position = "none") +
    geom_histogram(fill = "Blue", colour = "Black", binwidth = (range(data$wage)[2] -
        range(data$wage)[1])/30) + labs(title = "Distribution of wage",
```

```
    x = "wage ($)", y = "Frequency")

plot(wage.hist)
```


```
# logWage variable
summary(data$logWage)
```

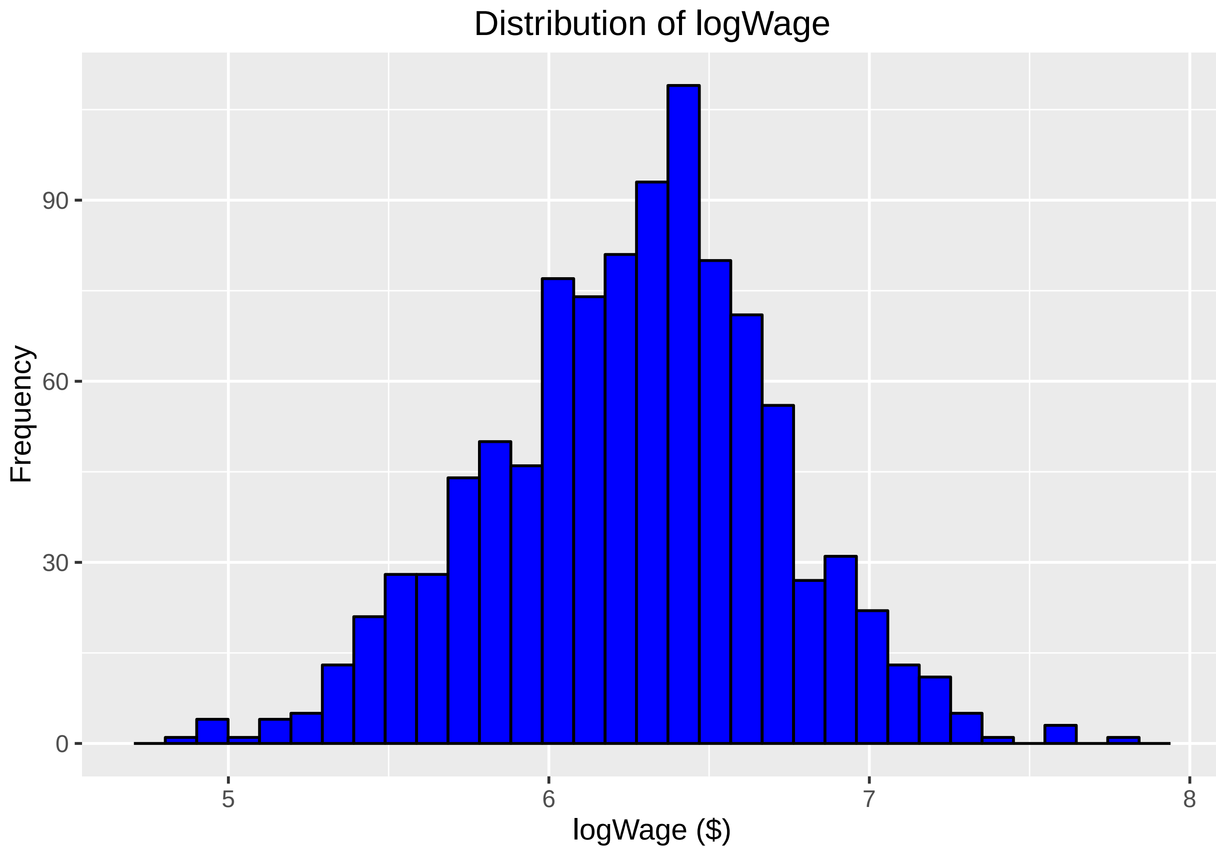
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.844   5.991   6.297   6.263   6.555   7.785
```

```
print(quantile(data$logWage, probs = c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75,
    0.9, 0.95, 0.99, 1)))
```

```
##       1%       5%      10%      25%      50%      75%      90%      95%
## 5.236007 5.500848 5.666427 5.991465 6.297109 6.554645 6.817825 6.974194
##      99%     100%
## 7.245818 7.784889
```

```
# Plot the histogram of apps at 30 bins
logWage.hist <- ggplot(data, aes(logWage)) + theme(legend.position = "none") +
    geom_histogram(fill = "Blue", colour = "Black", binwidth = (range(data$logWage)[2] -
        range(data$logWage)[1])/30) + labs(title = "Distribution of logWage",
    x = "logWage ($)", y = "Frequency")

plot(logWage.hist)
```




```
# education variable
summary(data$education)
```

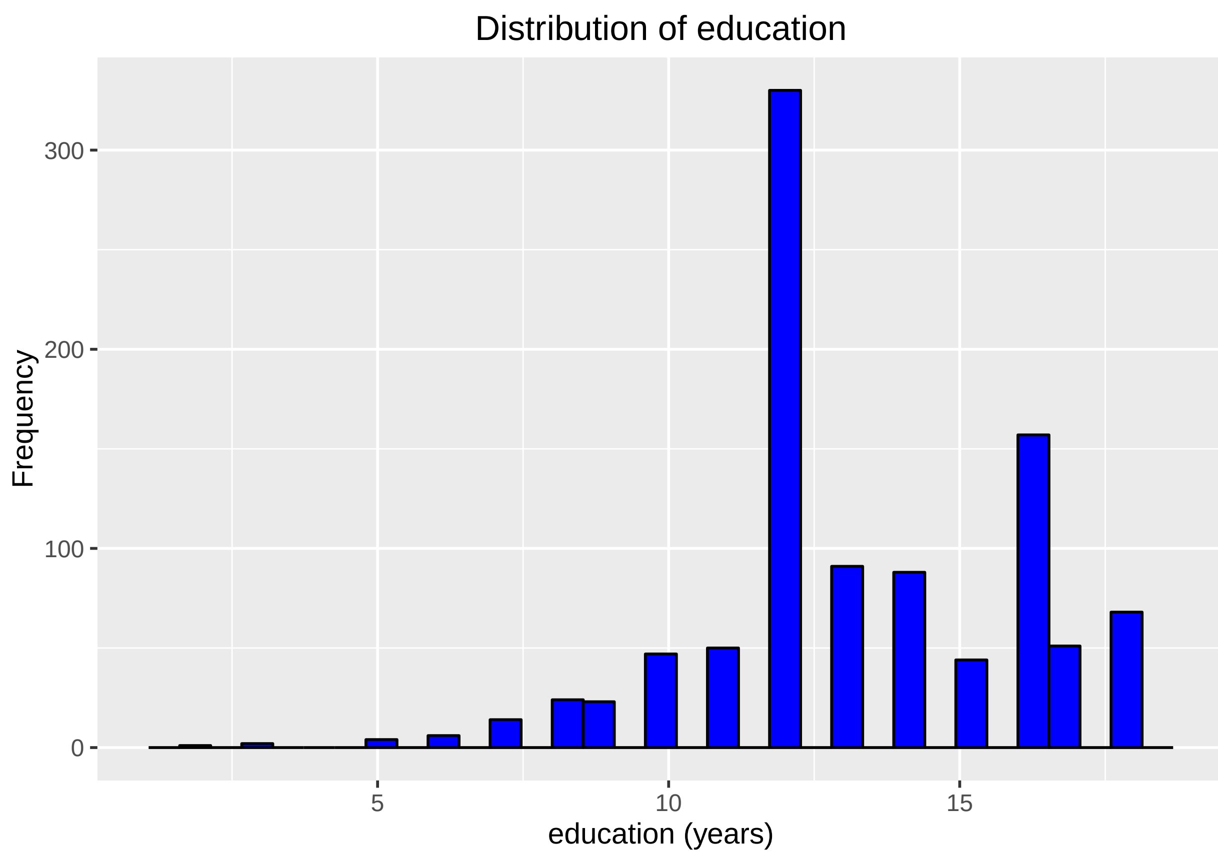
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00   12.00   12.00   13.22   16.00   18.00
```

```
print(quantile(data$education, probs = c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75,
    0.9, 0.95, 0.99, 1)))
```

```
##   1%   5%  10%  25%  50%  75%  90%  95%  99% 100%
##    6    8   10   12   12   16   17   18   18   18
```

```
# Plot the histogram of apps at 30 bins
education.hist <- ggplot(data, aes(education)) + theme(legend.position = "none") +
    geom_histogram(fill = "Blue", colour = "Black", binwidth = (range(data$education)[2] -
        range(data$education)[1])/30) + labs(title = "Distribution of education",
    x = "education (years)", y = "Frequency")

plot(education.hist)
```




```r
# experience variable
summary(data$experience)
```

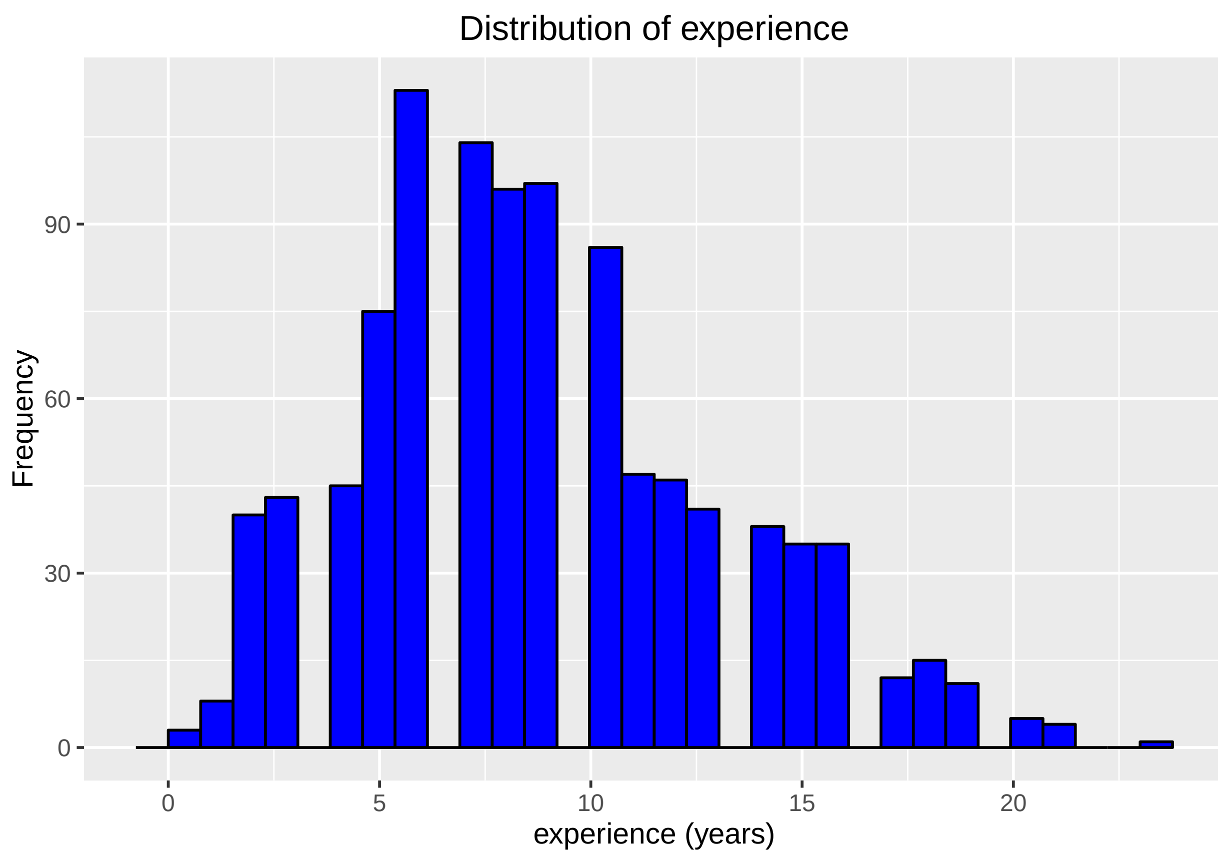
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   6.000   8.000   8.788  11.000  23.000
```

```r
print(quantile(data$experience, probs = c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75,
    0.9, 0.95, 0.99, 1)))
```

```
##    1%    5%   10%   25%   50%   75%   90%   95%   99%  100%
##  1.00  2.00  4.00  6.00  8.00 11.00 15.00 16.00 19.01 23.00
```

```r
# Plot the histogram of apps at 30 bins
experience.hist <- ggplot(data, aes(experience)) + theme(legend.position = "none") +
    geom_histogram(fill = "Blue", colour = "Black", binwidth = (range(data$experience)[2] -
        range(data$experience)[1])/30) + labs(title = "Distribution of experience",
    x = "experience (years)", y = "Frequency")

plot(experience.hist)
```




```
# experienceSquare variable
summary(data$experienceSquare)
```

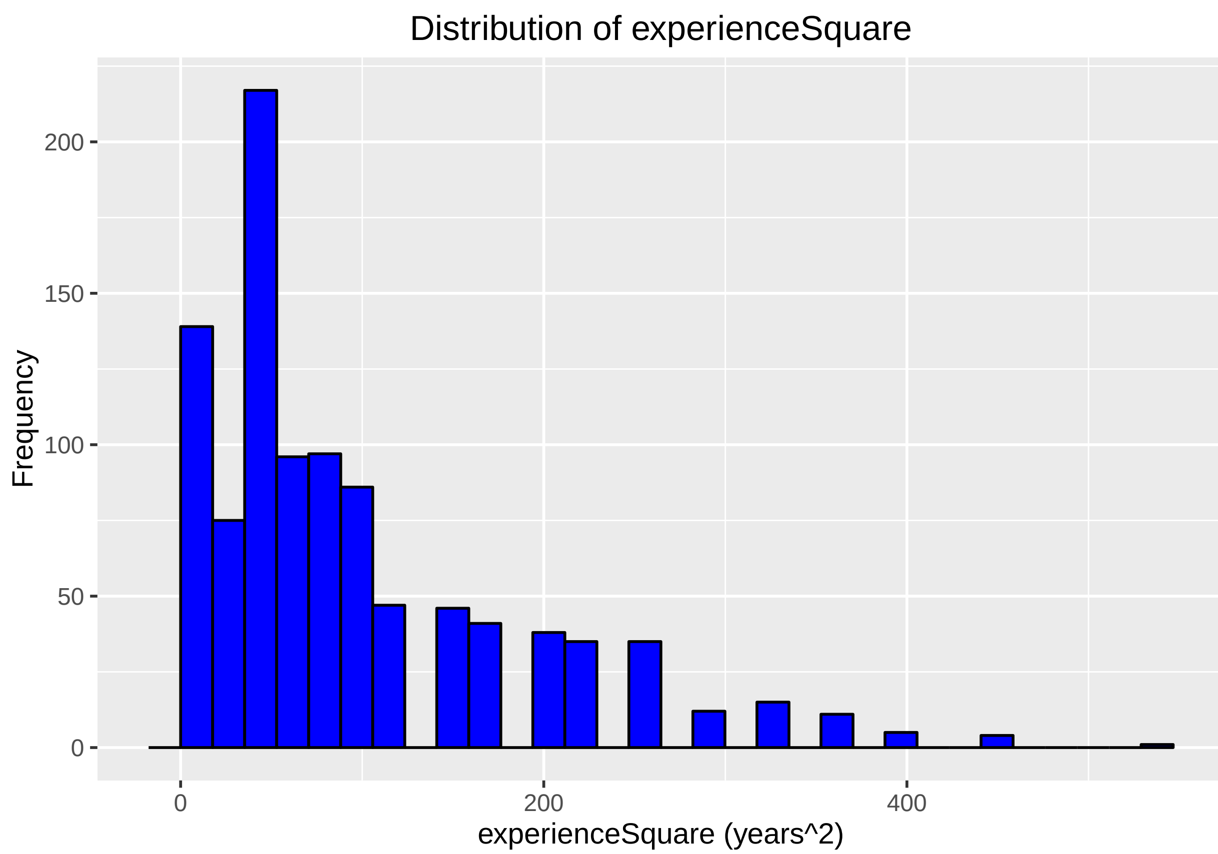
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   36.00   64.00   95.03  121.00  529.00
```

```
print(quantile(data$experienceSquare, probs = c(0.01, 0.05, 0.1, 0.25,
    0.5, 0.75, 0.9, 0.95, 0.99, 1)))
```

```
##     1%     5%    10%    25%    50%    75%    90%    95%    99%   100%
##   1.00   4.00  16.00  36.00  64.00 121.00 225.00 256.00 361.39 529.00
```

```
# Plot the histogram of apps at 30 bins
experienceSquare.hist <- ggplot(data, aes(experienceSquare)) + theme(legend.position = "none") +
    geom_histogram(fill = "Blue", colour = "Black", binwidth = (range(data$experienceSquare)[2] -
        range(data$experienceSquare)[1])/30) + labs(title = "Distribution of experienceSquare",
    x = "experienceSquare (years^2)", y = "Frequency")

plot(experienceSquare.hist)
```




```r
# IQscore variable
summary(data$IQscore)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    50.0    93.0   103.0   102.3   113.0   144.0     316
```

```r
print(quantile(data$IQscore, probs = c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75,
    0.9, 0.95, 0.99, 1), na.rm = TRUE))
```

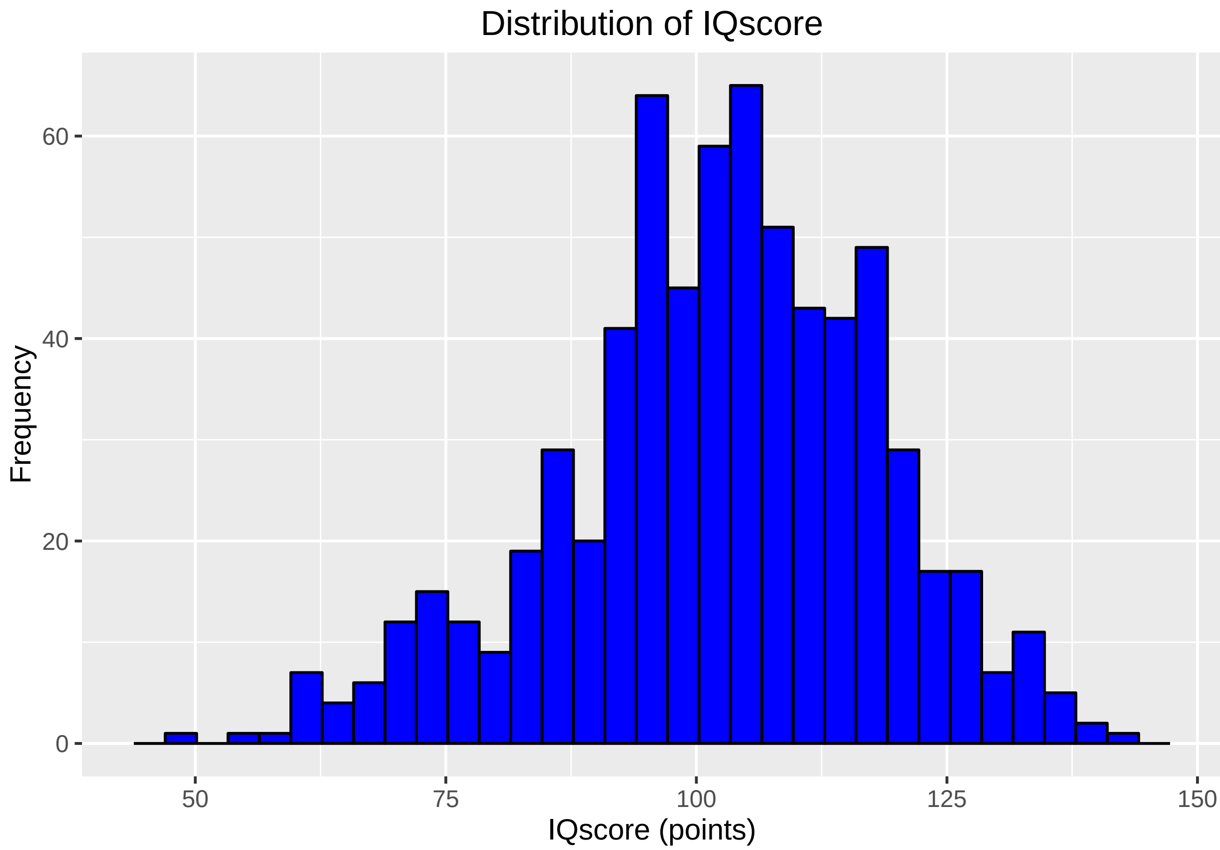
```
##     1%     5%    10%    25%    50%    75%    90%    95%    99%   100%
##  61.83  73.15  82.00  93.00 103.00 113.00 122.00 126.85 135.00 144.00
```

```r
# Plot the histogram of apps at 30 bins
IQscore.hist <- ggplot(data, aes(IQscore)) + theme(legend.position = "none") +
    geom_histogram(fill = "Blue", colour = "Black") + labs(title = "Distribution of IQscore",
    x = "IQscore (points)", y = "Frequency")

plot(IQscore.hist)
```

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

```
## Warning: Removed 316 rows containing non-finite values (stat_bin).
```




```r
# dad_education variable
summary(data$dad_education)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    8.00   11.00   10.18   12.00   18.00     239
```

```r
print(quantile(data$dad_education, probs = c(0.01, 0.05, 0.1, 0.25, 0.5,
    0.75, 0.9, 0.95, 0.99, 1), na.rm = TRUE))
```

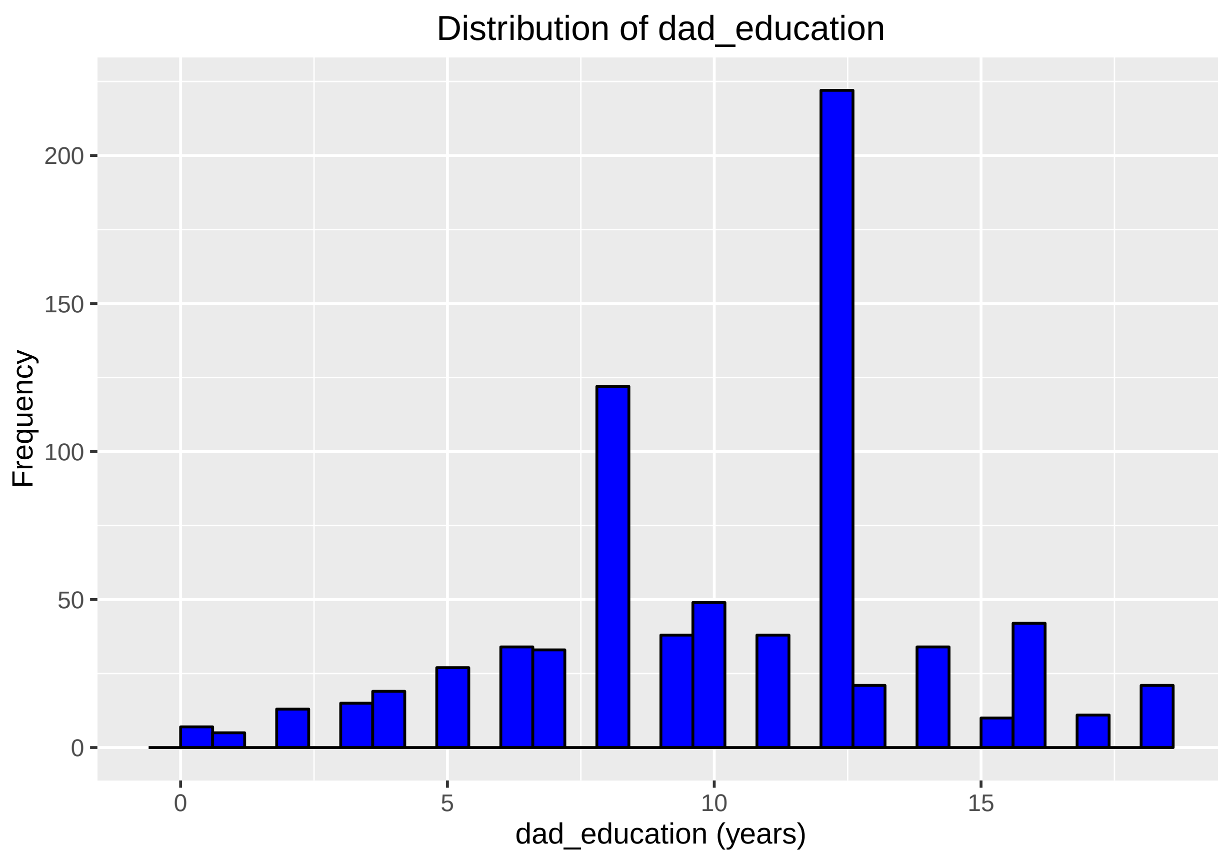
```
##   1%   5%  10%  25%  50%  75%  90%  95%  99% 100%
##    1    3    5    8   11   12   15   16   18   18
```

```r
# Plot the histogram of apps at 30 bins
dad_education.hist <- ggplot(data, aes(dad_education)) + theme(legend.position = "none") +
    geom_histogram(fill = "Blue", colour = "Black") + labs(title = "Distribution of dad_education",
    x = "dad_education (years)", y = "Frequency")

plot(dad_education.hist)
```

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

```
## Warning: Removed 239 rows containing non-finite values (stat_bin).
```




```r
# mom_education variable
summary(data$mom_education)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    8.00   12.00   10.45   12.00   18.00     128
```

```r
print(quantile(data$mom_education, probs = c(0.01, 0.05, 0.1, 0.25, 0.5,
    0.75, 0.9, 0.95, 0.99, 1), na.rm = TRUE))
```

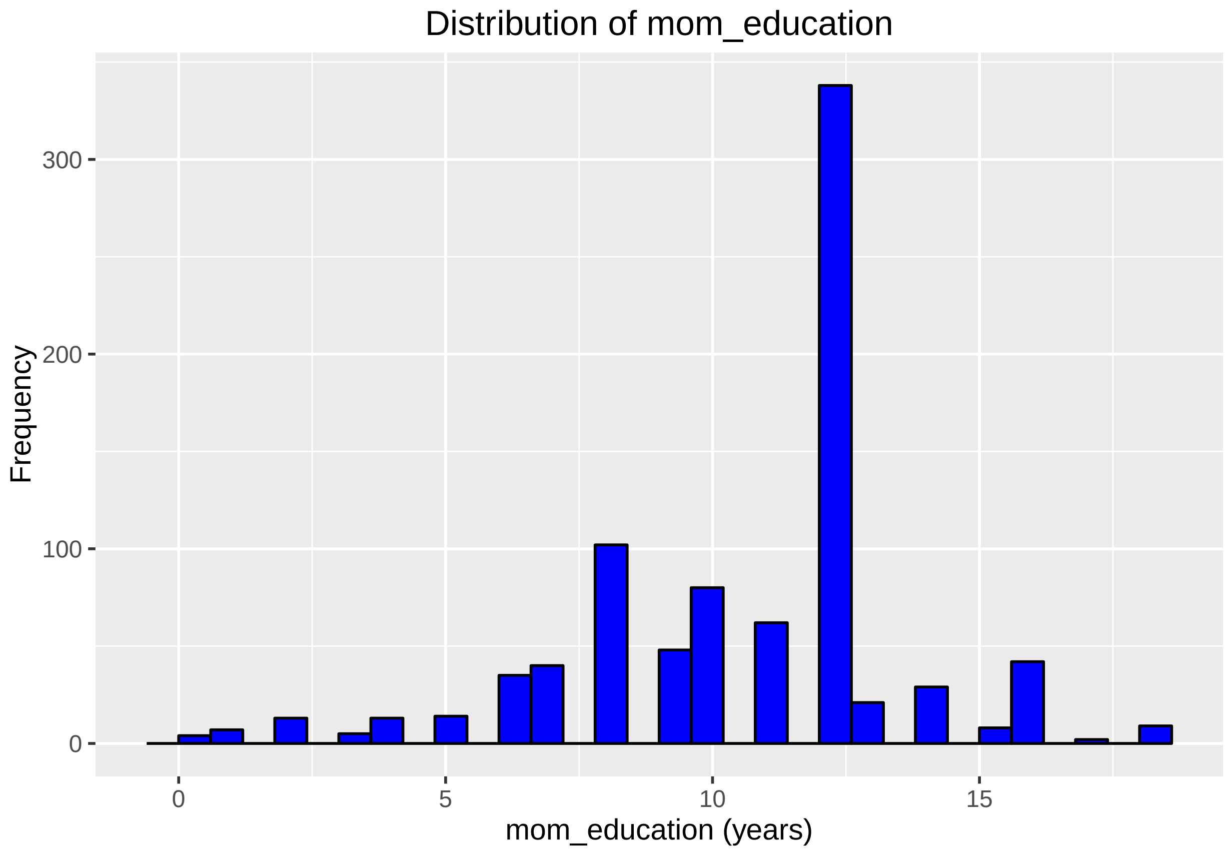
```
##    1%    5%   10%   25%   50%   75%   90%   95%   99%  100%
##  1.00  5.00  6.00  8.00 12.00 12.00 14.00 16.00 17.29 18.00
```

```r
# Plot the histogram of apps at 30 bins
mom_education.hist <- ggplot(data, aes(mom_education)) + theme(legend.position = "none") +
    geom_histogram(fill = "Blue", colour = "Black") + labs(title = "Distribution of mom_education",
    x = "mom_education (years)", y = "Frequency")

plot(mom_education.hist)
```

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

```
## Warning: Removed 128 rows containing non-finite values (stat_bin).
```




```r
# age variable
summary(data$age)
```

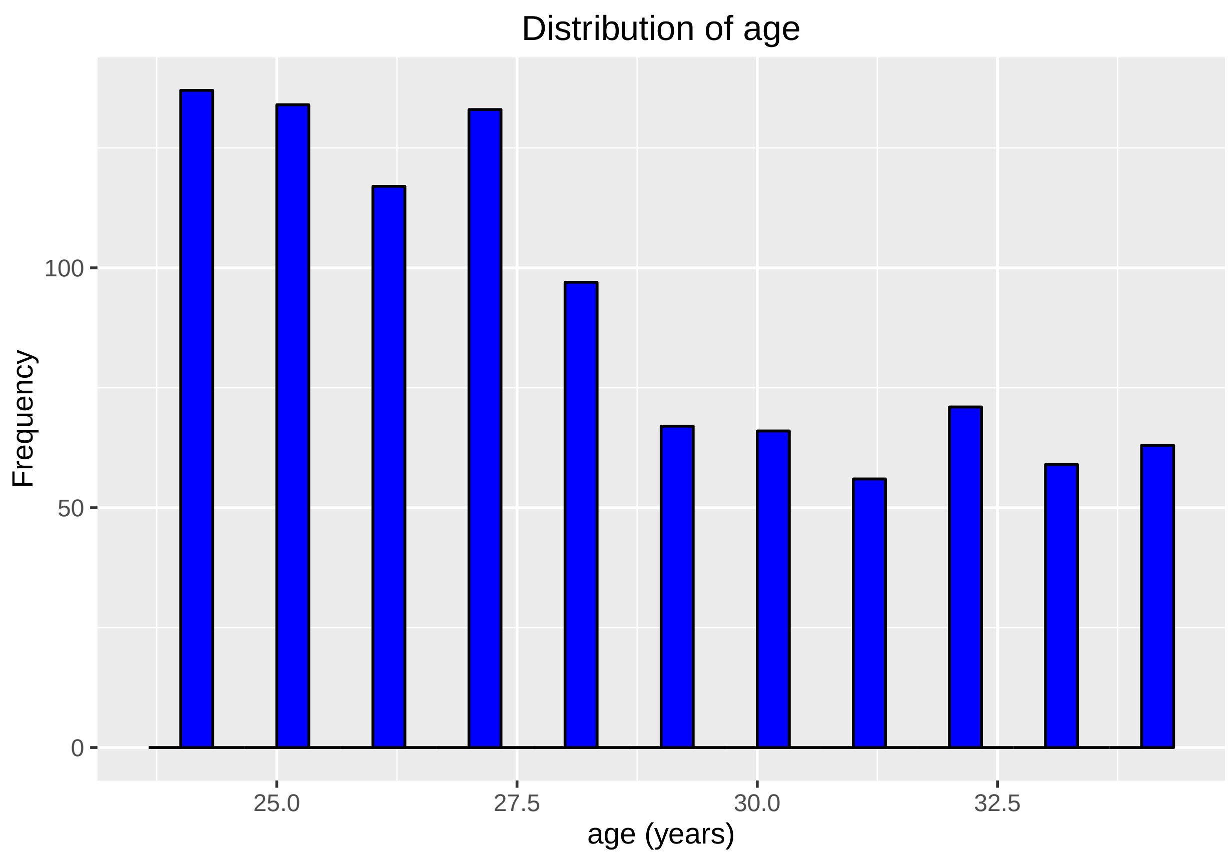
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   24.00   25.00   27.00   28.01   30.00   34.00
```

```r
print(quantile(data$age, probs = c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9,
    0.95, 0.99, 1), na.rm = TRUE))
```

```
##   1%   5%  10%  25%  50%  75%  90%  95%  99% 100%
##   24   24   24   25   27   30   33   34   34   34
```

```r
# Plot the histogram of apps at 30 bins
age.hist <- ggplot(data, aes(age)) + theme(legend.position = "none") +
    geom_histogram(fill = "Blue", colour = "Black", binwidth = (range(data$age)[2] -
        range(data$age)[1])/30) + labs(title = "Distribution of age", x = "age (years)",
    y = "Frequency")

plot(age.hist)
```




```
# raceColor variable
summary(data$raceColor)
```

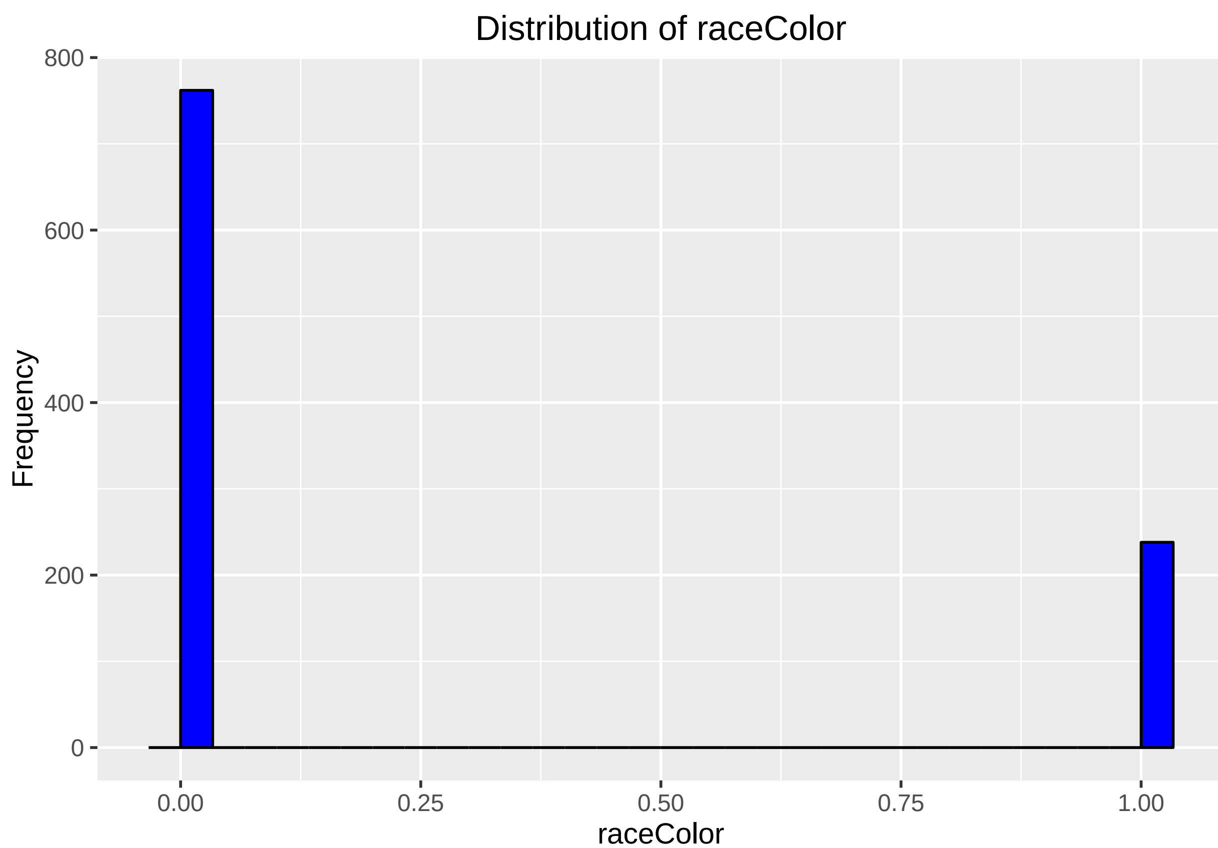
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   0.238   0.000   1.000
```

```
print(quantile(data$raceColor, probs = c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75,
    0.9, 0.95, 0.99, 1), na.rm = TRUE))
```

```
##   1%   5%  10%  25%  50%  75%  90%  95%  99% 100%
##    0    0    0    0    0    0    1    1    1    1
```

```
# Plot the histogram of apps at 30 bins
raceColor.hist <- ggplot(data, aes(raceColor)) + theme(legend.position = "none") +
    geom_histogram(fill = "Blue", colour = "Black", binwidth = (range(data$raceColor)[2] -
        range(data$raceColor)[1])/30) + labs(title = "Distribution of raceColor",
    x = "raceColor", y = "Frequency")

plot(raceColor.hist)
```




```
# rural variable
summary(data$rural)
```

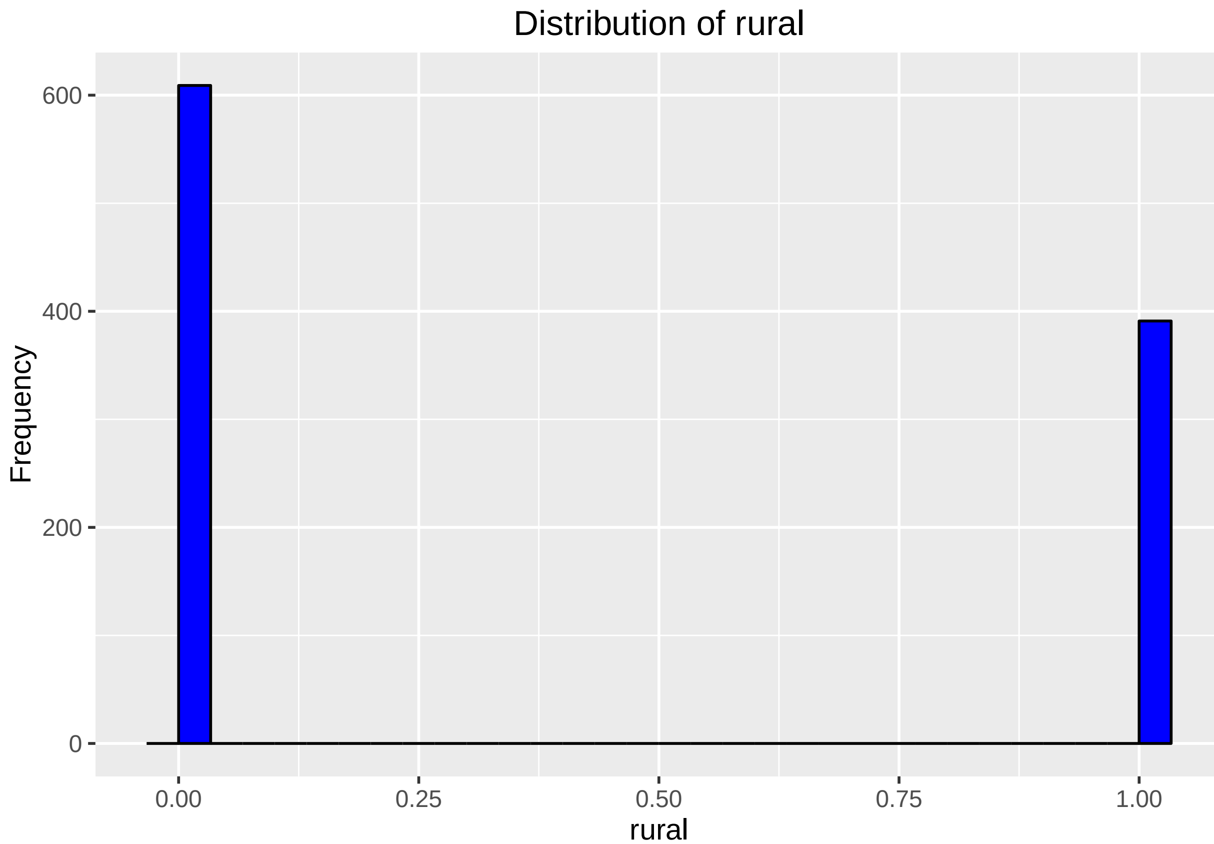
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   0.391   1.000   1.000
```

```
print(quantile(data$rural, probs = c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75,
    0.9, 0.95, 0.99, 1), na.rm = TRUE))
```

```
##   1%   5%  10%  25%  50%  75%  90%  95%  99% 100%
##    0    0    0    0    0    1    1    1    1    1
```

```
# Plot the histogram of apps at 30 bins
rural.hist <- ggplot(data, aes(rural)) + theme(legend.position = "none") +
    geom_histogram(fill = "Blue", colour = "Black", binwidth = (range(data$rural)[2] -
        range(data$rural)[1])/30) + labs(title = "Distribution of rural",
    x = "rural", y = "Frequency")

plot(rural.hist)
```




```r
# city variable
summary(data$city)
```

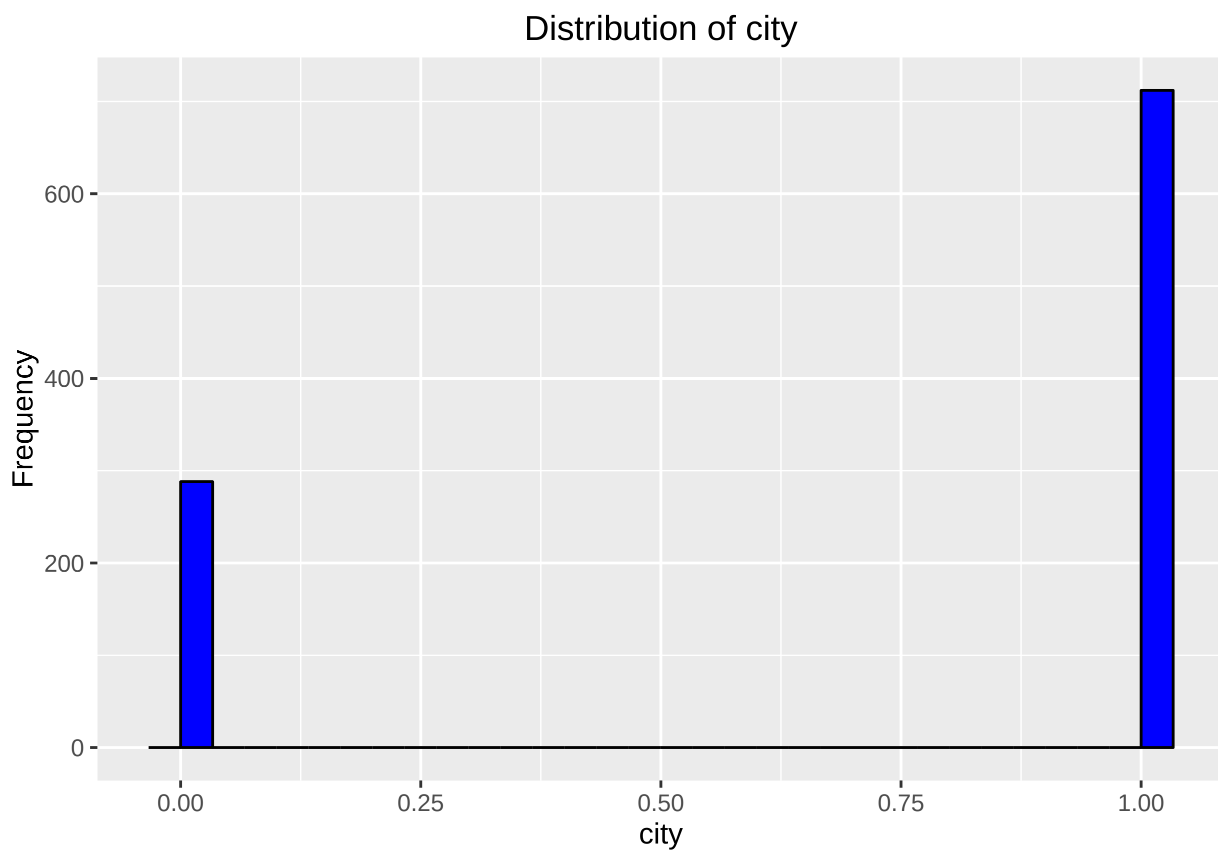
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   0.712   1.000   1.000
```

```r
print(quantile(data$city, probs = c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9,
    0.95, 0.99, 1), na.rm = TRUE))
```

```
##   1%   5%  10%  25%  50%  75%  90%  95%  99% 100%
##    0    0    0    0    1    1    1    1    1    1
```

```r
# Plot the histogram of apps at 30 bins
city.hist <- ggplot(data, aes(city)) + theme(legend.position = "none") +
    geom_histogram(fill = "Blue", colour = "Black", binwidth = (range(data$city)[2] -
        range(data$city)[1])/30) + labs(title = "Distribution of city",
    x = "city", y = "Frequency")

plot(city.hist)
```




```
# z1 variable
summary(data$z1)
```

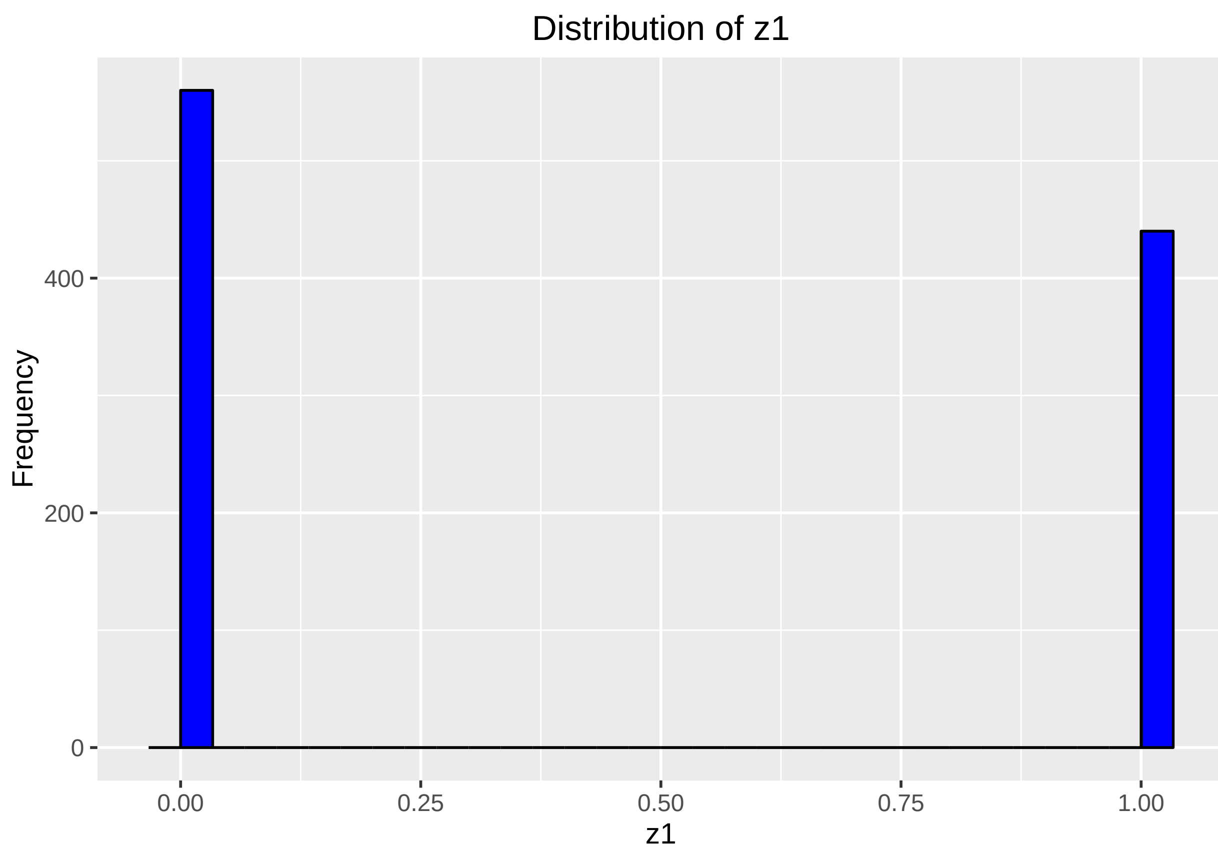
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00    0.44    1.00    1.00
```

```
print(quantile(data$z1, probs = c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9,
    0.95, 0.99, 1), na.rm = TRUE))
```

```
##   1%   5%  10%  25%  50%  75%  90%  95%  99% 100%
##    0    0    0    0    0    1    1    1    1    1
```

```
# Plot the histogram of apps at 30 bins
z1.hist <- ggplot(data, aes(z1)) + theme(legend.position = "none") + geom_histogram(fill = "Blue",
    colour = "Black", binwidth = (range(data$z1)[2] - range(data$z1)[1])/30) +
    labs(title = "Distribution of z1", x = "z1", y = "Frequency")

plot(z1.hist)
```




```r
# z2 variable
summary(data$z2)
```

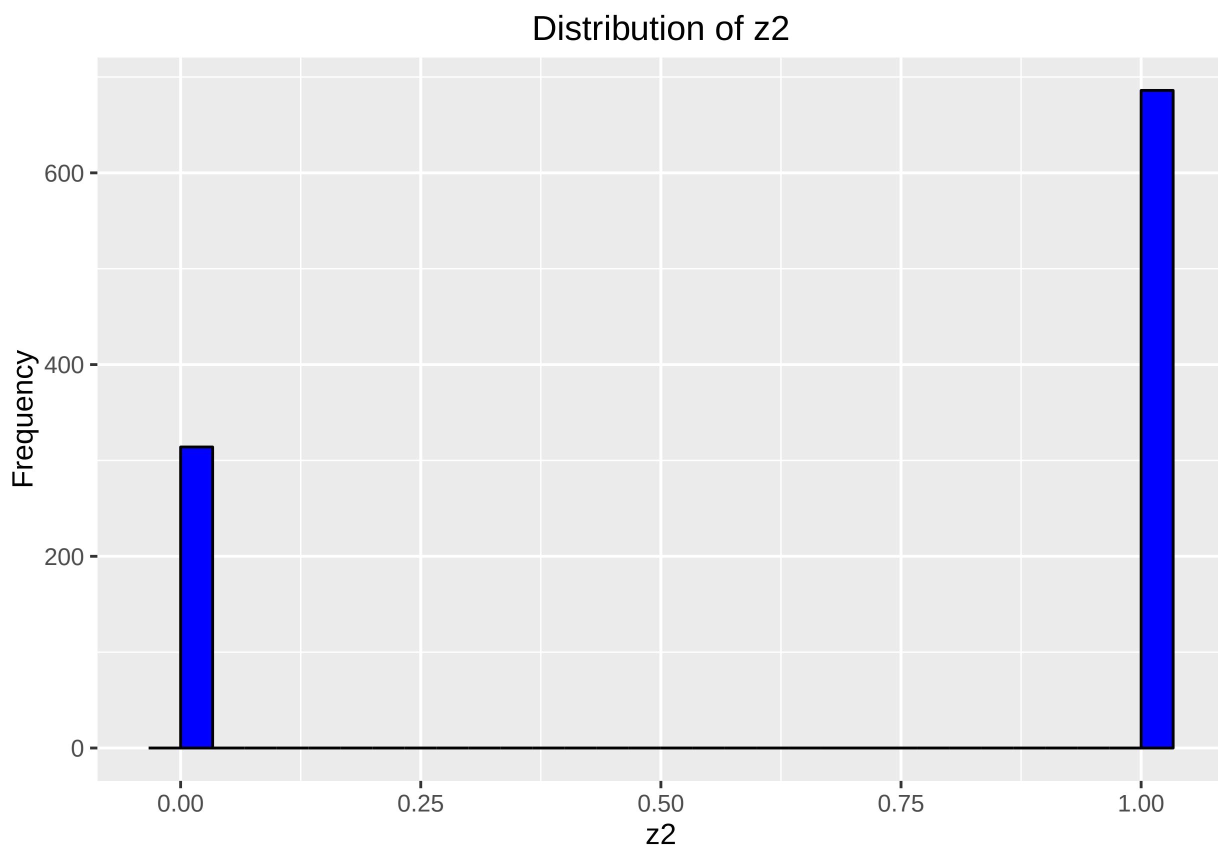
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   0.686   1.000   1.000
```

```r
print(quantile(data$z2, probs = c(0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9,
    0.95, 0.99, 1), na.rm = TRUE))
```

```
##   1%   5%  10%  25%  50%  75%  90%  95%  99% 100%
##    0    0    0    0    1    1    1    1    1    1
```

```r
# Plot the histogram of apps at 30 bins
z2.hist <- ggplot(data, aes(z2)) + theme(legend.position = "none") + geom_histogram(fill = "Blue",
    colour = "Black", binwidth = (range(data$z2)[2] - range(data$z2)[1])/30) +
    labs(title = "Distribution of z2", x = "z2", y = "Frequency")

plot(z2.hist)
```

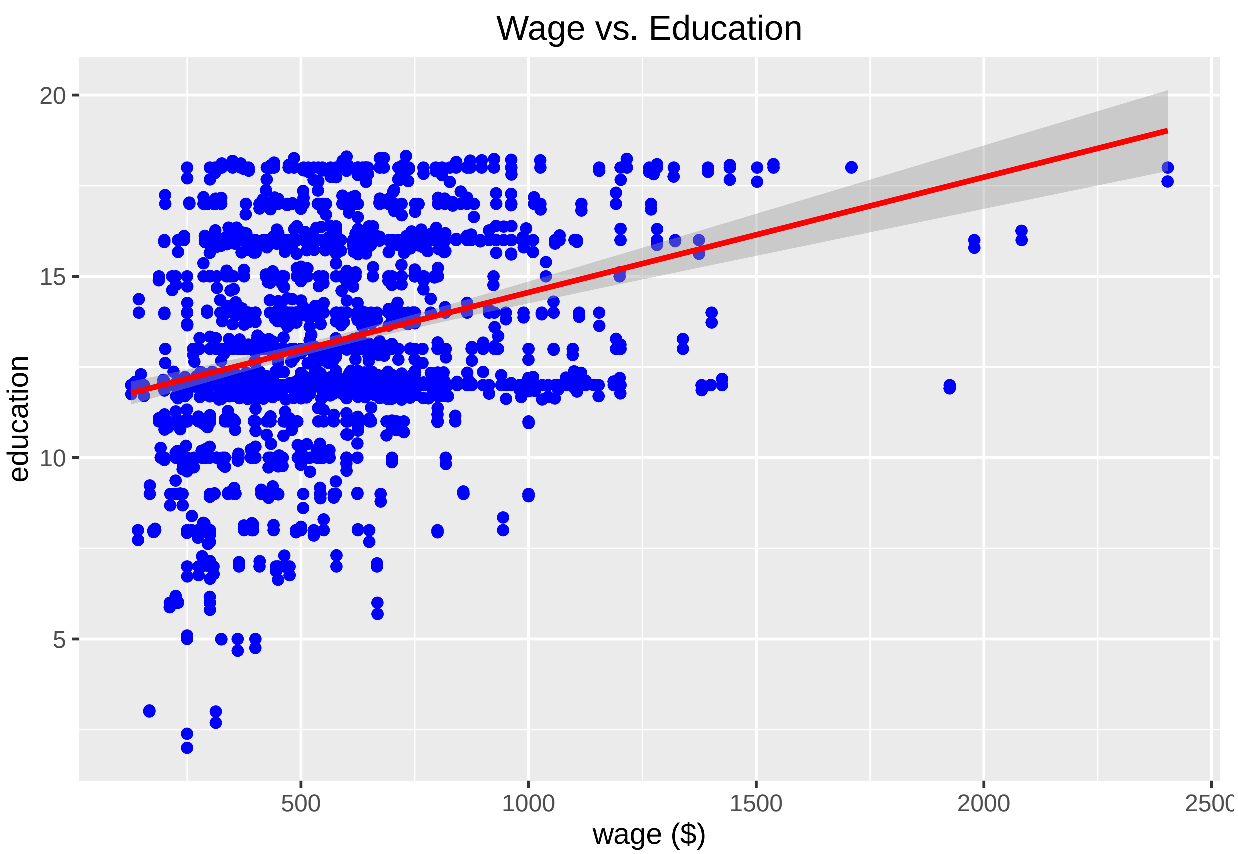


4.2 Bivariate Analysis

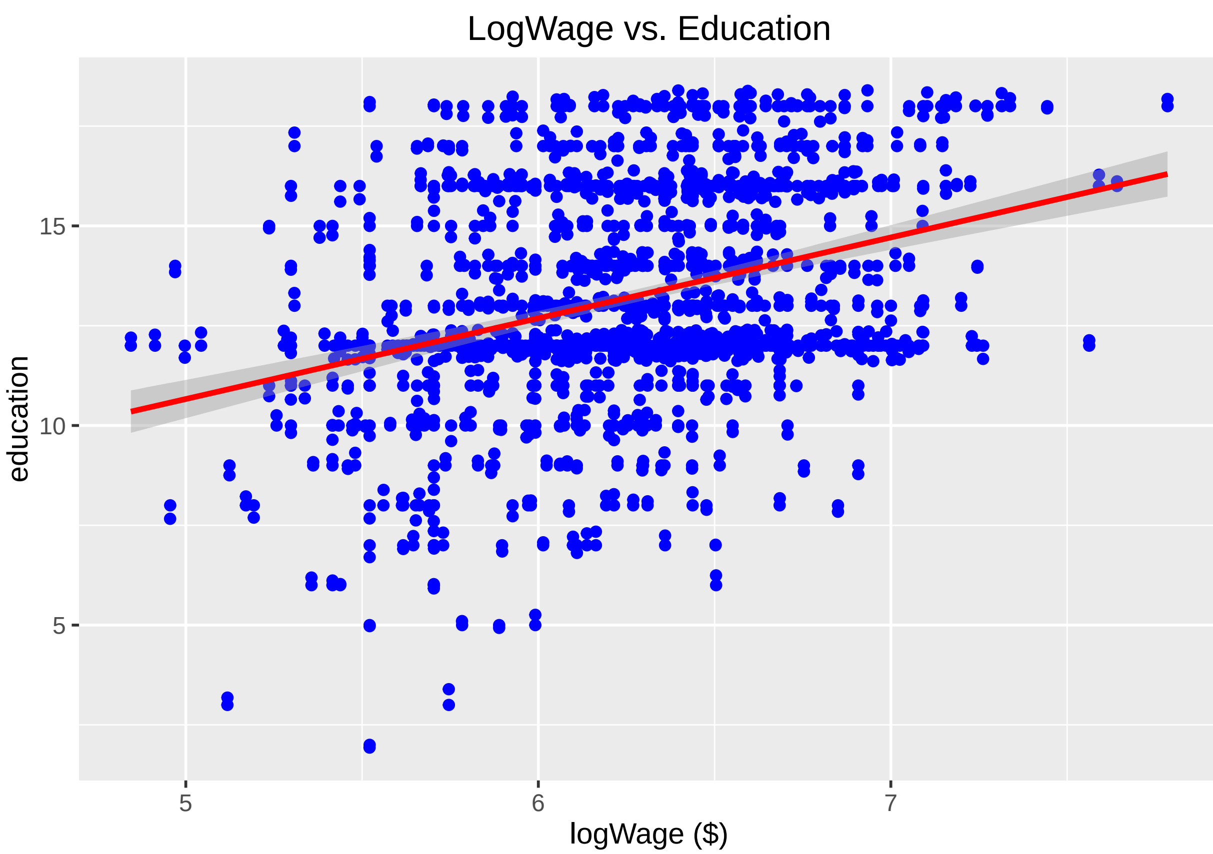
- **wage, logWage vs. education** - Both wage and logWage are weakly correlated with education with a correlation value of about 0.3. The wage vs. education scatterplot shows a possible linear trend.
- **wage, logWage vs. experience** - Both wage and logWage appear uncorrelated with experience with very low correlation values of -0.0060 and -0.0290, respectively. The wage vs. experience scatterplot shows that experience is not affected by wage for the most part. The logWage vs. experience scatterplot shows that experience is not affected by logWage as well.
- **wage, logWage vs. experienceSquare** - Both wage and logWage appear uncorrelated with experienceSquare with very low correlation values of -0.043 and -0.065, respectively. The wage vs. experienceSquare scatterplot shows that experienceSquare is not affected by wage for the most part. The logWage vs. experienceSquare scatterplot shows that experience is not affected by logWage as well.
- **wage, logWage vs. IQscore** - Both wage and logWage are weakly correlated with IQscoare with low correlation values of 0.186 and 0.201, respectively. The wage and logWage vs. IQscore scatterplots show that IQscoare affects wage and logWage slightly. As wage or logWage go up, IQscore increases by a small amount.
- **wage, logWage vs. dad_education** - Both wage and logWage are weakly correlated with dad_education with low correlation values of 0.19 and 0.19, respectively. The wage and logWage vs. dad_education scatterplots show that dad_education affects wage and logWage slightly. As wage or logWage go up, dad_education increases by a small amount.
- **wage, logWage vs. mom_education** - Both wage and logWage are weakly correlated with mom_education with low correlation values of 0.20 and 0.21, respectively. The wage and logWage vs. mom_education scatterplots show that mom_education affects wage and logWage slightly. As wage or logWage go up, mom_education increases by a small amount.

- **wage, logWage vs. age** - Both wage and logWage are weakly correlated with age with low correlation values of 0.26 and 0.25, respectively. The wage and logWage vs. age scatterplots show that age affects wage and logWage slightly. As wage or logWage go up, age increases by a small amount.
- **wage, logWage vs. raceColor** - Both wage and logWage are weakly correlated with raceColor with low correlation values of -0.30 and -0.34, respectively. The wage and logWage vs. raceColor scatterplots show that raceColor affects wage and logWage slightly. As wage or logWage go up, there are fewer people that have the raceColor variable set to 1.
- **wage, logWage vs. rural** - Both wage and logWage are weakly correlated with rural with low correlation values of -0.22 and -0.25, respectively. The wage and logWage vs. rural scatterplots show that rural affects wage and logWage slightly. As wage or logWage go up, there are fewer people that have the rural variable set to 1.
- **wage, logWage vs. city** - Both wage and logWage are weakly correlated with city with low correlation values of 0.22 and 0.24, respectively. The wage and logWage vs. rural scatterplots show that city affects wage and logWage slightly. As wage or logWage go up, there are more people that have the city variable set to 1.
- **wage, logWage vs. z1** - Both wage and logWage are weakly correlated with z1 with low correlation values of 0.101 and 0.087, respectively. The wage and logWage vs. z1 scatterplots show that z1 affects wage and logWage slightly. As wage or logWage go up, there are more people that have the z1 variable set to 1.
- **wage, logWage vs. z2** - Both wage and logWage are weakly correlated with z2 with low correlation values of 0.17 and 0.18, respectively. The wage and logWage vs. z2 scatterplots show that z2 affects wage and logWage slightly. As wage or logWage go up, there are more people that have the z2 variable set to 1. z2 shows a slightly stronger correlation with wage and logWage than z1.

```r
# Scatter plot with wage variable
wage.education.plot = ggplot(data, aes(x = wage, y = education)) + theme(legend.position = "none") +
    geom_point(colour = "Blue") + geom_jitter(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "Wage vs. Education", x = "wage ($)",
    y = "education")
plot(wage.education.plot)
```




```
# Scatter plot with logWage variable
lwage.education.plot = ggplot(data, aes(x = logWage, y = education)) +
    theme(legend.position = "none") + geom_point(colour = "Blue") + geom_jitter(colour = "Blue") +
    geom_smooth(colour = "red", method = "lm") + labs(title = "LogWage vs. Education",
    x = "logWage ($)", y = "education")
plot(lwage.education.plot)
```




```
# Run correlations with wage and logWage variables
cor(data$wage, data$education)
```

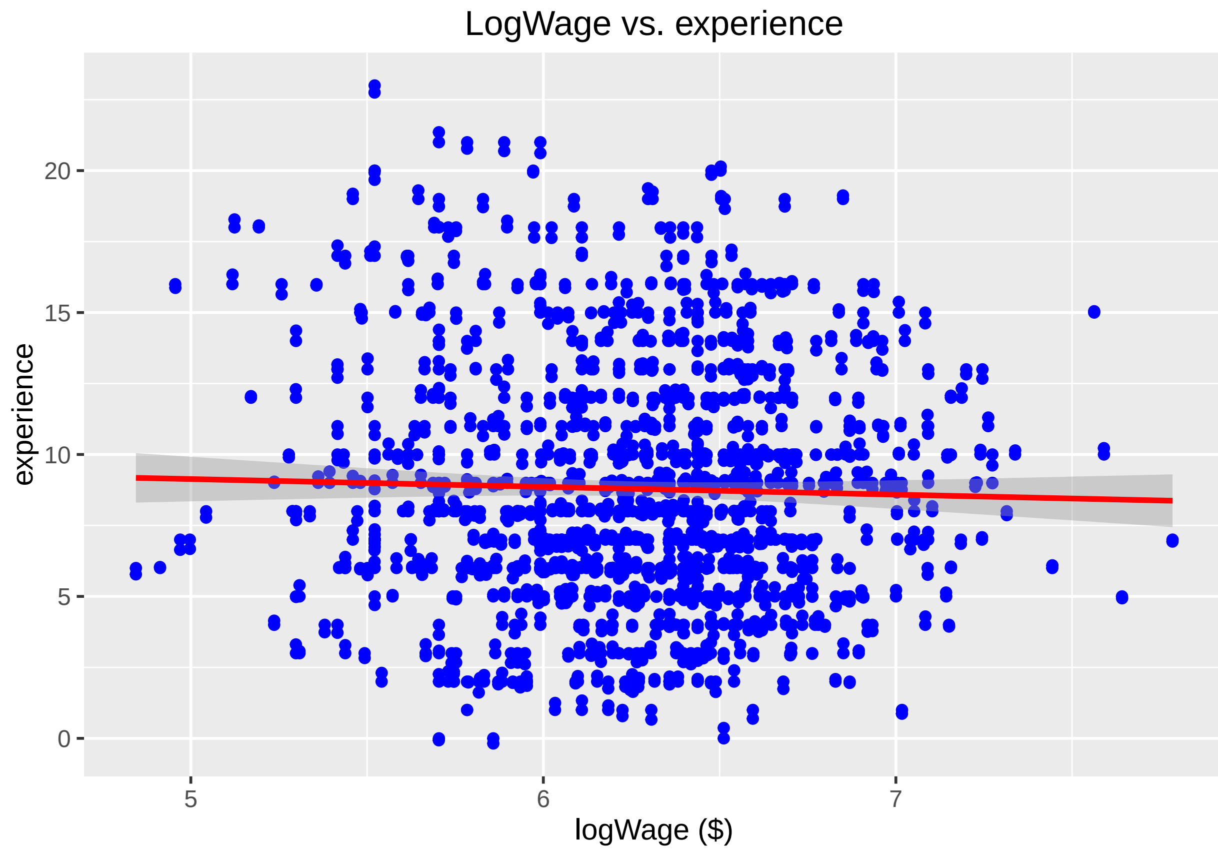
```
## [1] 0.3103986
```

```
cor(data$logWage, data$education)
```

```
## [1] 0.3318494
```

```
# Scatter plot with wage variable
wage.experience.plot = ggplot(data, aes(x = wage, y = experience)) + theme(legend.position = "none") +
    geom_point(colour = "Blue") + geom_jitter(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "Wage vs. experience", x = "wage ($)",
    y = "experience")
plot(wage.experience.plot)
```


```
# Scatter plot with logWage variable
lwage.experience.plot = ggplot(data, aes(x = logWage, y = experience)) +
    theme(legend.position = "none") + geom_point(colour = "Blue") + geom_jitter(colour = "Blue") +
    geom_smooth(colour = "red", method = "lm") + labs(title = "LogWage vs. experience",
    x = "logWage ($)", y = "experience")
plot(lwage.experience.plot)
```




```
# Run correlations with wage and logWage variables
cor(data$wage, data$experience)
```

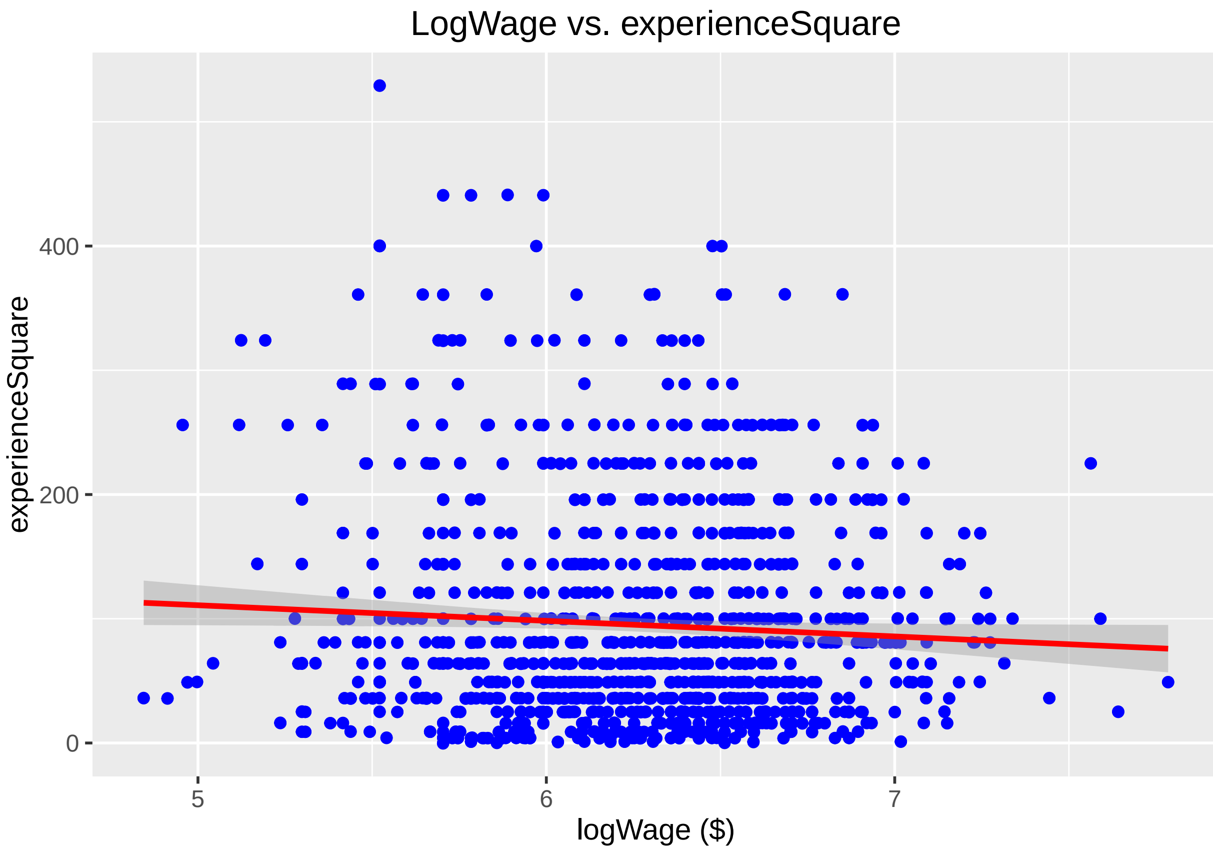
```
## [1] -0.005985988
```

```
cor(data$logWage, data$experience)
```

```
## [1] -0.02905727
```

```
# Scatter plot with wage variable
wage.experienceSquare.plot = ggplot(data, aes(x = wage, y = experienceSquare)) +
    theme(legend.position = "none") + geom_point(colour = "Blue") + geom_jitter(colour = "Blue") +
    geom_smooth(colour = "red", method = "lm") + labs(title = "Wage vs. experienceSquare",
    x = "wage ($)", y = "experienceSquare")
plot(wage.experienceSquare.plot)
```


```
# Scatter plot with logWage variable
lwage.experienceSquare.plot = ggplot(data, aes(x = logWage, y = experienceSquare)) +
    theme(legend.position = "none") + geom_point(colour = "Blue") + geom_jitter(colour = "Blue") +
    geom_smooth(colour = "red", method = "lm") + labs(title = "LogWage vs. experienceSquare",
    x = "logWage ($)", y = "experienceSquare")
plot(lwage.experienceSquare.plot)
```




```
# Run correlations with wage and logWage variables
cor(data$wage, data$experienceSquare)
```

```
## [1] -0.04270455
```

```
cor(data$logWage, data$experienceSquare)
```

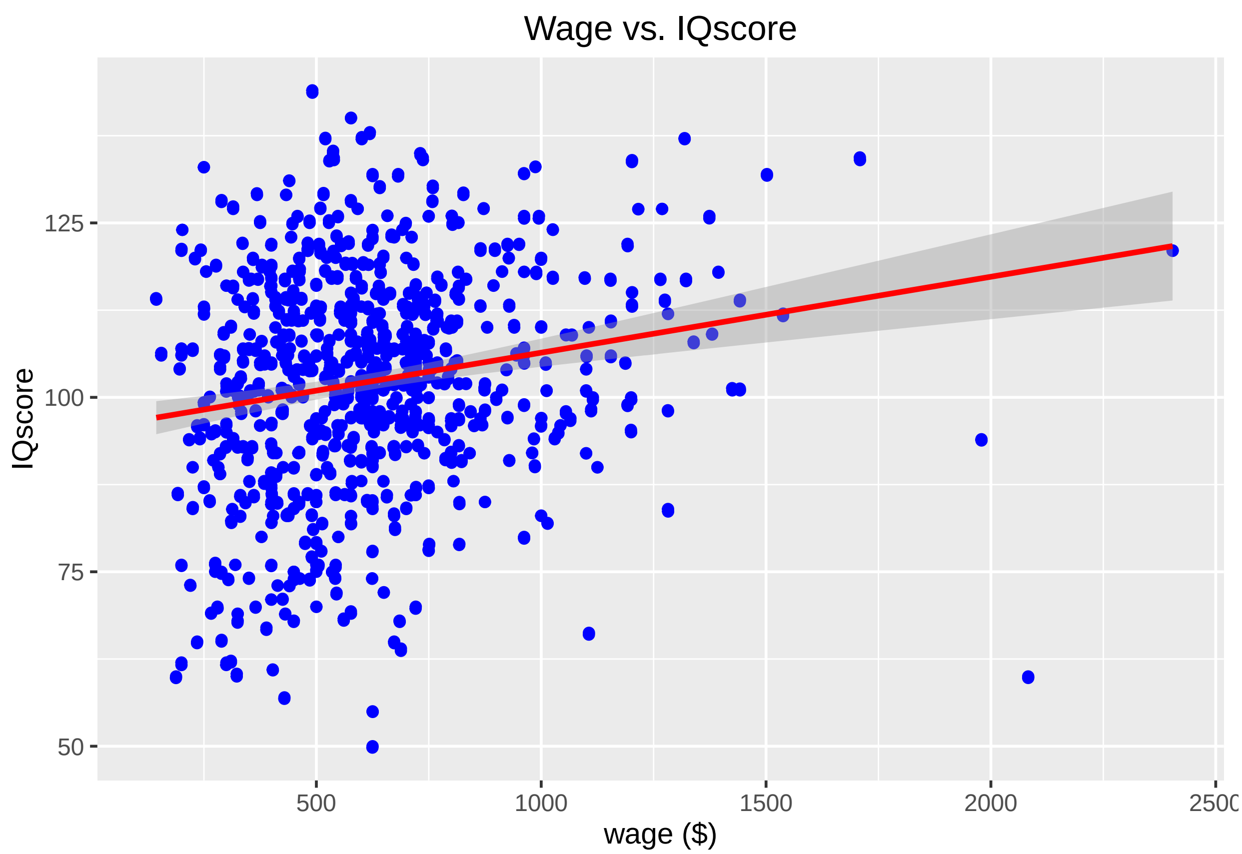
```
## [1] -0.0647476
```

```
# Scatter plot with wage variable
wage.IQscore.plot = ggplot(data, aes(x = wage, y = IQscore)) + theme(legend.position = "none") +
    geom_point(colour = "Blue") + geom_jitter(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "Wage vs. IQscore", x = "wage ($)", y = "IQscore")
plot(wage.IQscore.plot)
```

```
## Warning: Removed 316 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 316 rows containing missing values (geom_point).
```

```
## Warning: Removed 316 rows containing missing values (geom_point).
```

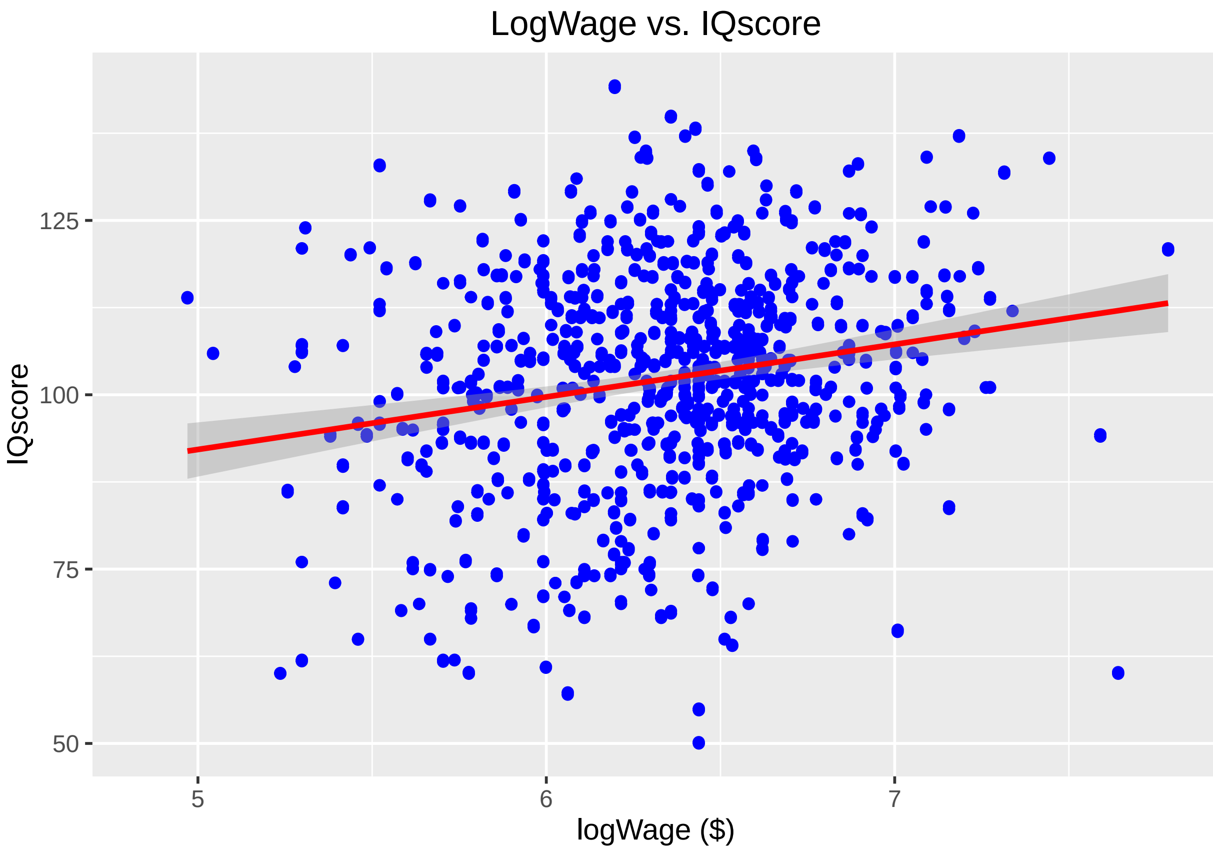



```
# Scatter plot with logWage variable
lwage.IQscore.plot = ggplot(data, aes(x = logWage, y = IQscore)) + theme(legend.position = "none") +
    geom_point(colour = "Blue") + geom_jitter(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "LogWage vs. IQscore", x = "logWage ($)",
    y = "IQscore")
plot(lwage.IQscore.plot)
```

```
## Warning: Removed 316 rows containing non-finite values (stat_smooth).

## Warning: Removed 316 rows containing missing values (geom_point).

## Warning: Removed 316 rows containing missing values (geom_point).
```




```r
# Run correlations with wage and logWage variables
cor(data$wage, data$IQscore, use = "complete.obs")
```

```
## [1] 0.1858557
```

```r
cor(data$logWage, data$IQscore, use = "complete.obs")
```

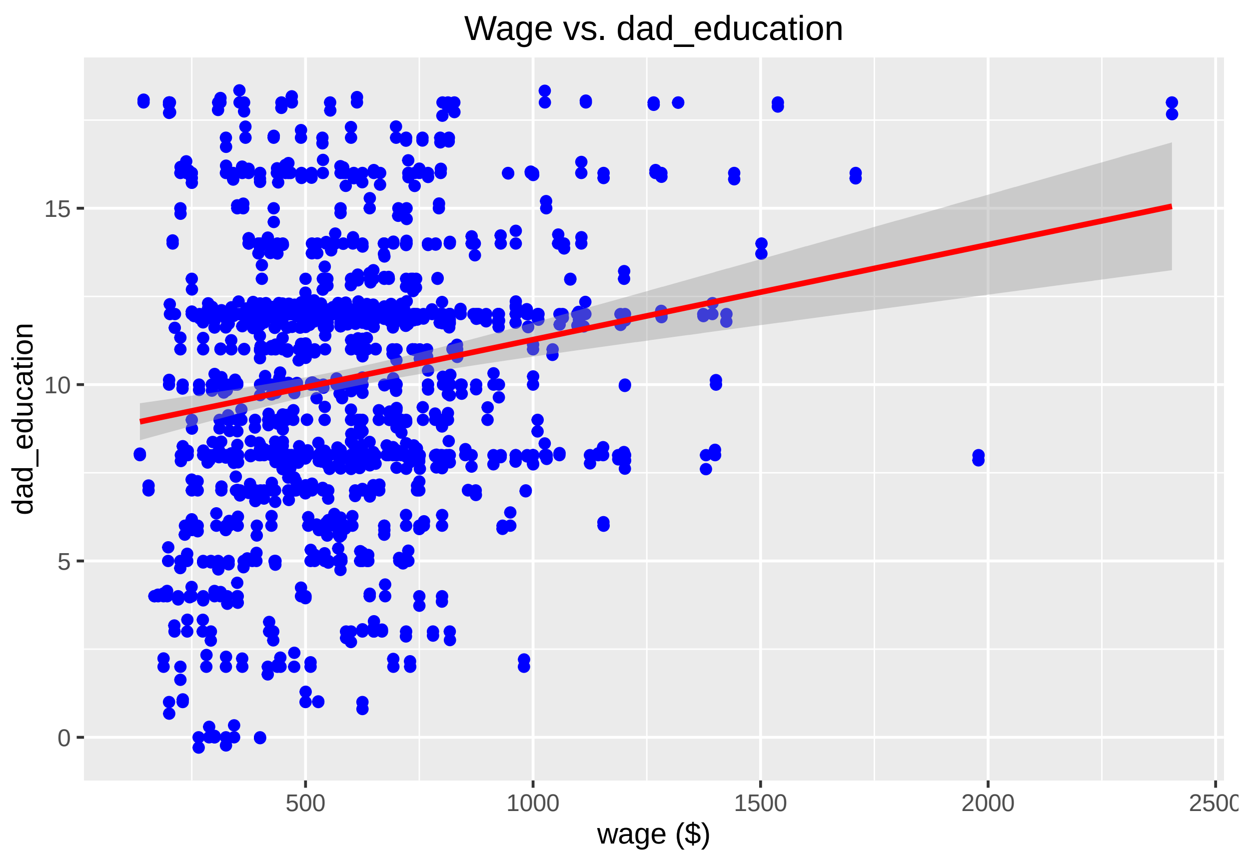
```
## [1] 0.2009578
```

```r
# Scatter plot with wage variable
wage.dad_education.plot = ggplot(data, aes(x = wage, y = dad_education)) +
    theme(legend.position = "none") + geom_point(colour = "Blue") + geom_jitter(colour = "Blue") +
    geom_smooth(colour = "red", method = "lm") + labs(title = "Wage vs. dad_education",
    x = "wage ($)", y = "dad_education")
plot(wage.dad_education.plot)
```

```
## Warning: Removed 239 rows containing non-finite values (stat_smooth).

## Warning: Removed 239 rows containing missing values (geom_point).

## Warning: Removed 239 rows containing missing values (geom_point).
```

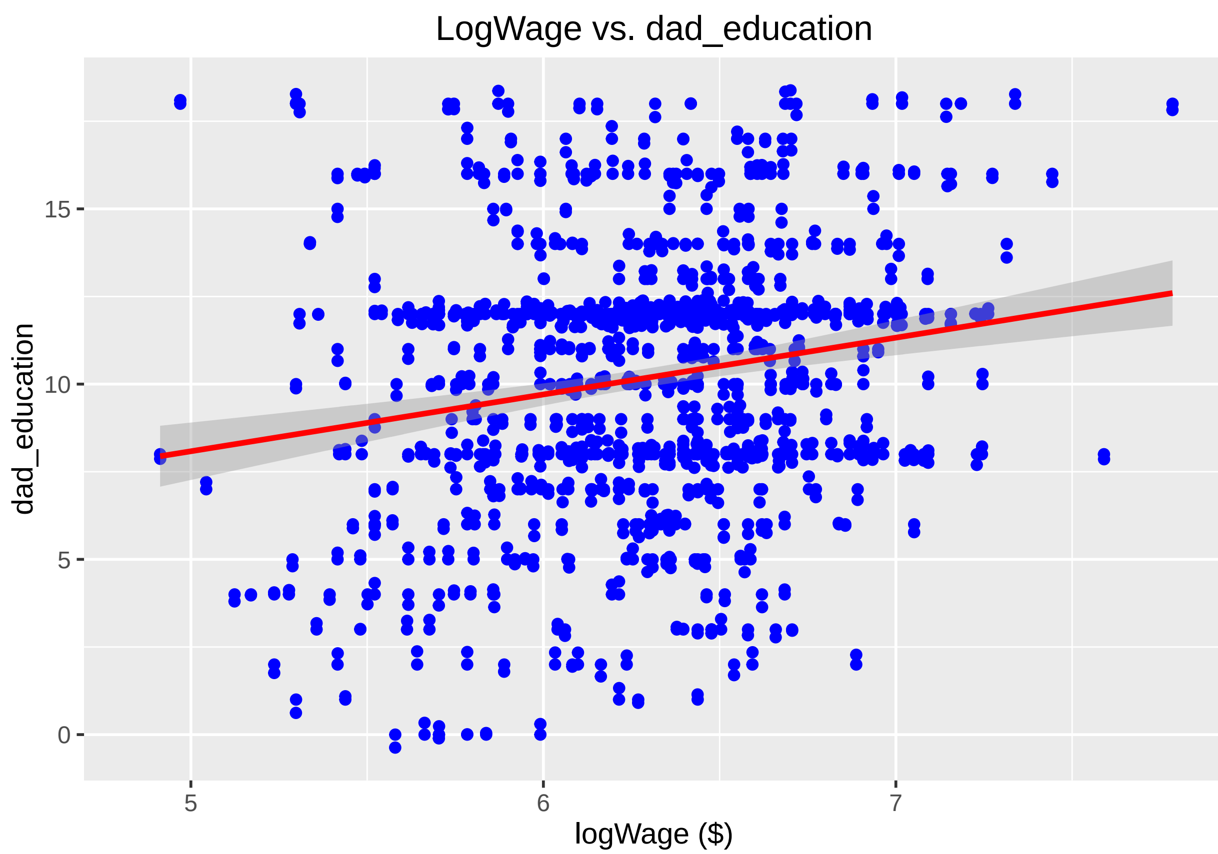



```r
# Scatter plot with logWage variable
lwage.dad_education.plot = ggplot(data, aes(x = logWage, y = dad_education)) +
    theme(legend.position = "none") + geom_point(colour = "Blue") + geom_jitter(colour = "Blue") +
    geom_smooth(colour = "red", method = "lm") + labs(title = "LogWage vs. dad_education",
    x = "logWage ($)", y = "dad_education")
plot(lwage.dad_education.plot)
```

```
## Warning: Removed 239 rows containing non-finite values (stat_smooth).

## Warning: Removed 239 rows containing missing values (geom_point).

## Warning: Removed 239 rows containing missing values (geom_point).
```




```r
# Run correlations with wage and logWage variables
cor(data$wage, data$dad_education, use = "complete.obs")
```

```
## [1] 0.1901681
```

```r
cor(data$logWage, data$dad_education, use = "complete.obs")
```

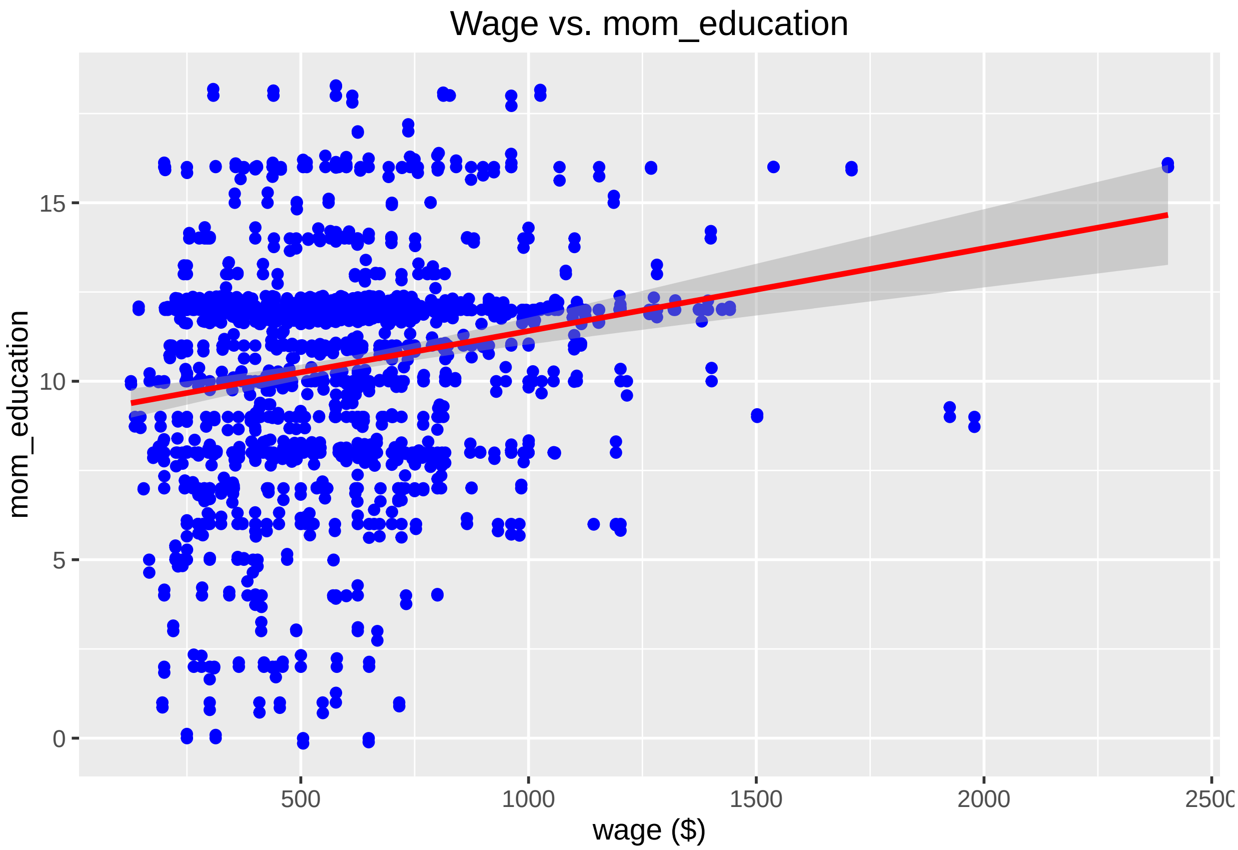
```
## [1] 0.18908
```

```r
# Scatter plot with wage variable
wage.mom_education.plot = ggplot(data, aes(x = wage, y = mom_education)) +
    theme(legend.position = "none") + geom_point(colour = "Blue") + geom_jitter(colour = "Blue") +
    geom_smooth(colour = "red", method = "lm") + labs(title = "Wage vs. mom_education",
    x = "wage ($)", y = "mom_education")
plot(wage.mom_education.plot)
```

```
## Warning: Removed 128 rows containing non-finite values (stat_smooth).

## Warning: Removed 128 rows containing missing values (geom_point).

## Warning: Removed 128 rows containing missing values (geom_point).
```

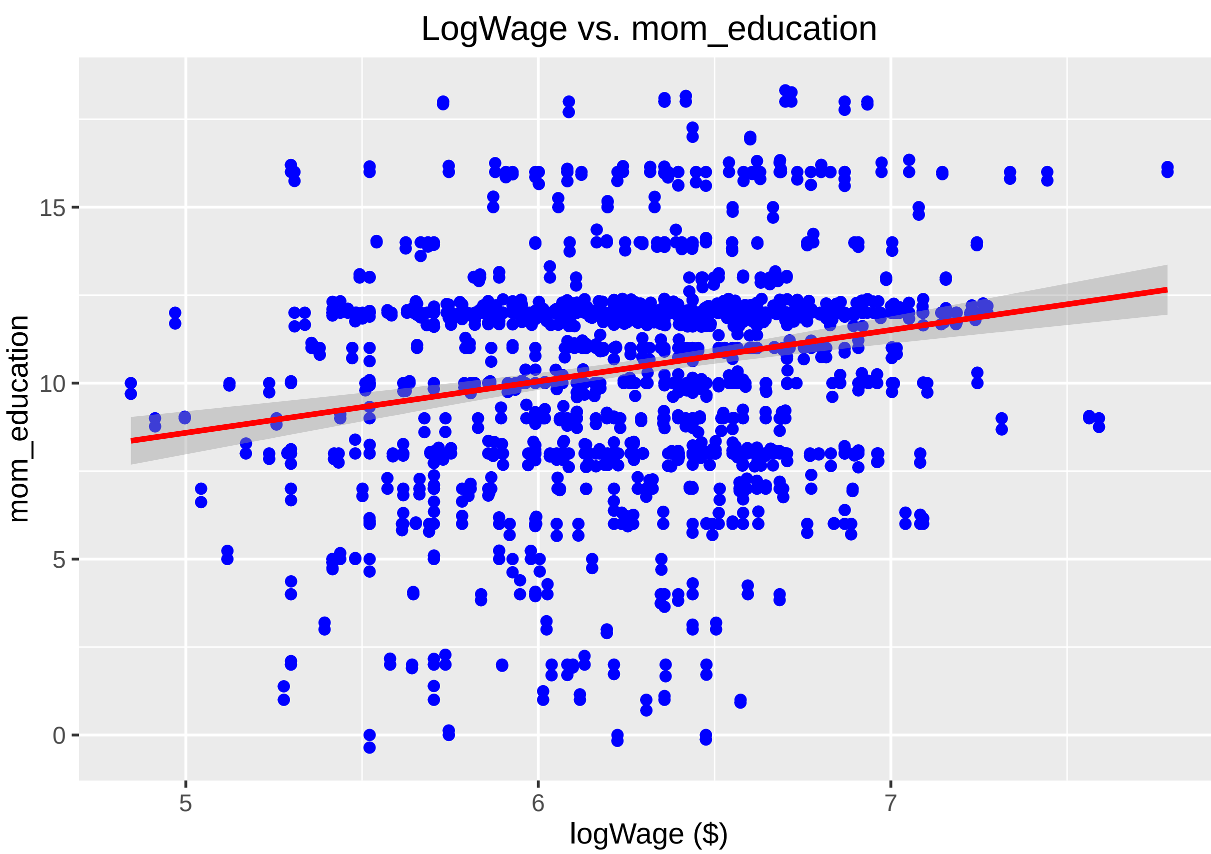



```
# Scatter plot with logWage variable
lwage.mom_education.plot = ggplot(data, aes(x = logWage, y = mom_education)) +
    theme(legend.position = "none") + geom_point(colour = "Blue") + geom_jitter(colour = "Blue") +
    geom_smooth(colour = "red", method = "lm") + labs(title = "LogWage vs. mom_education",
    x = "logWage ($)", y = "mom_education")
plot(lwage.mom_education.plot)
```

```
## Warning: Removed 128 rows containing non-finite values (stat_smooth).

## Warning: Removed 128 rows containing missing values (geom_point).

## Warning: Removed 128 rows containing missing values (geom_point).
```



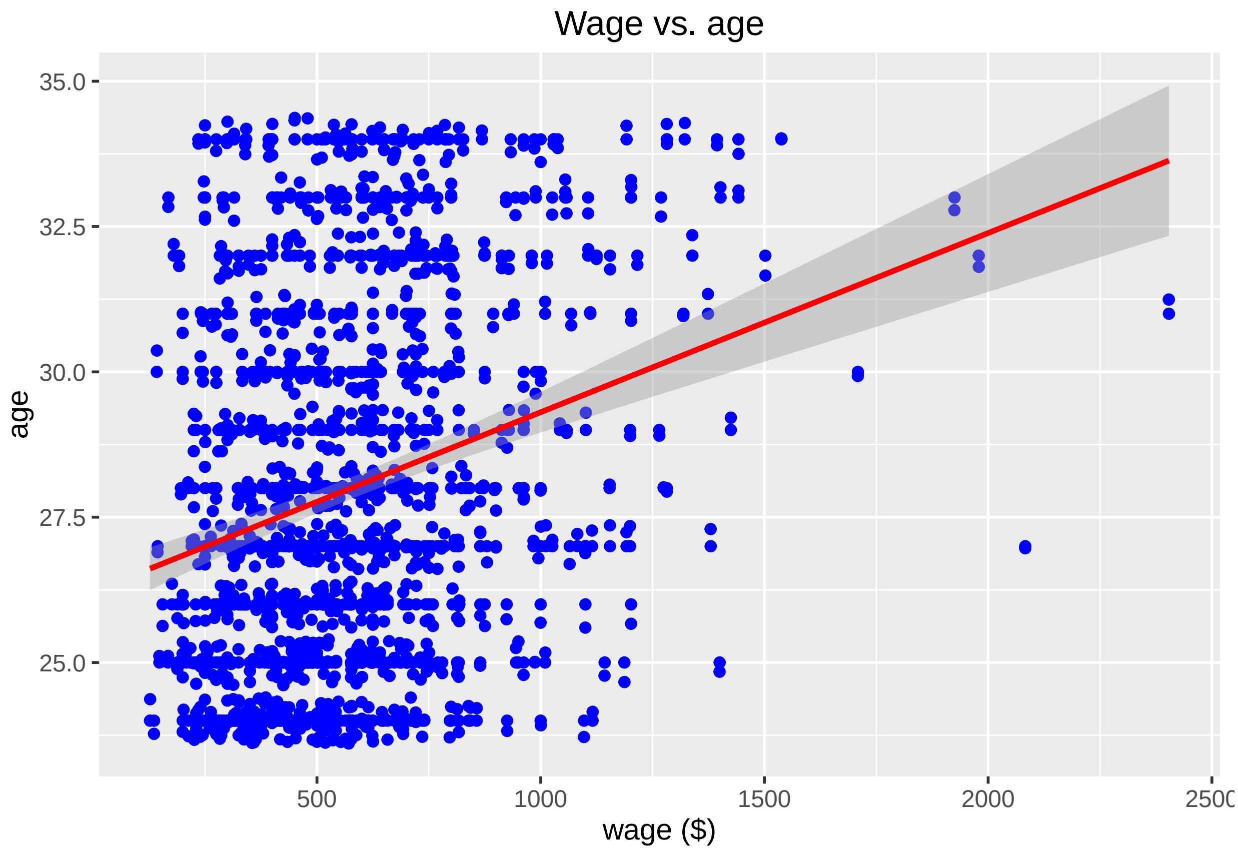

```r
# Run correlations with wage and logWage variables
cor(data$wage, data$mom_education, use = "complete.obs")
```

```
## [1] 0.1983845
```

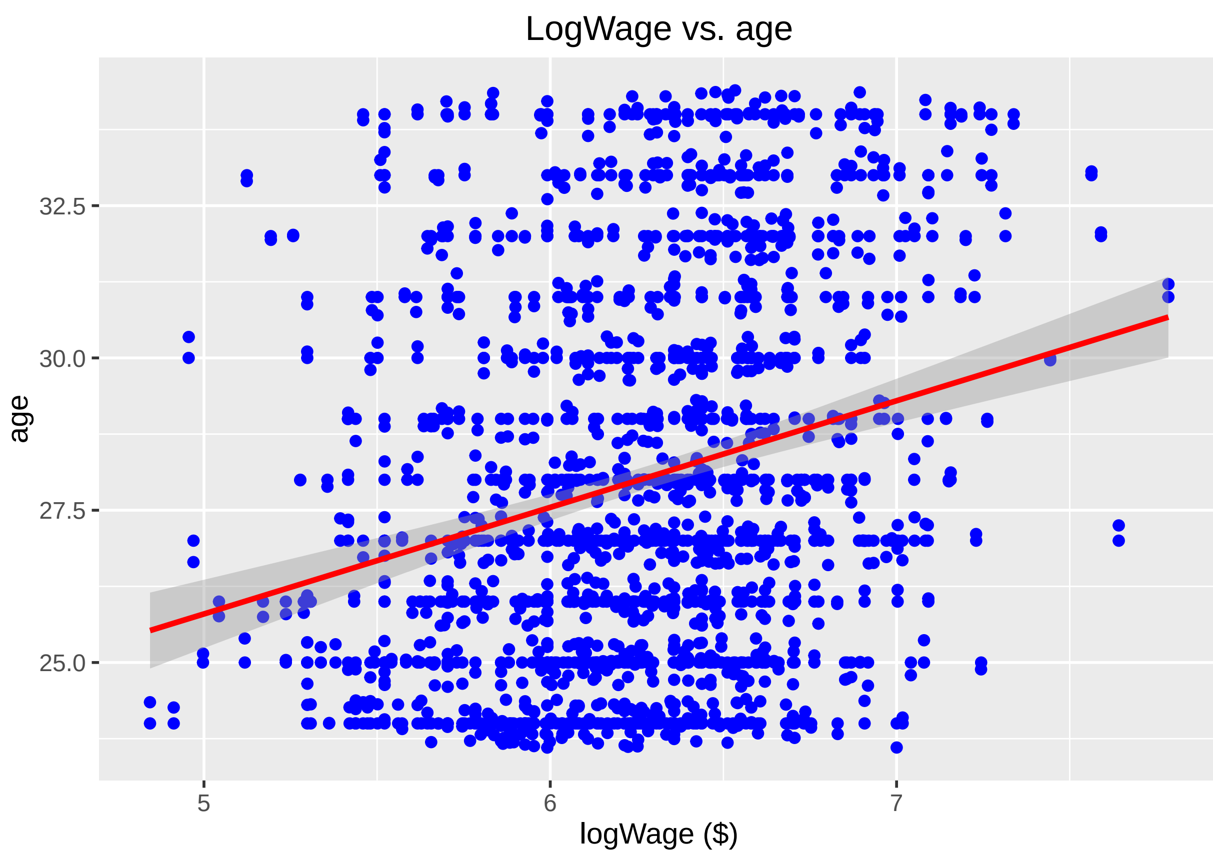
```r
cor(data$logWage, data$mom_education, use = "complete.obs")
```

```
## [1] 0.2104614
```

```r
# Scatter plot with wage variable
wage.age.plot = ggplot(data, aes(x = wage, y = age)) + theme(legend.position = "none") +
    geom_point(colour = "Blue") + geom_jitter(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "Wage vs. age", x = "wage ($)", y = "age")
plot(wage.age.plot)
```




```
# Scatter plot with logWage variable
lwage.age.plot = ggplot(data, aes(x = logWage, y = age)) + theme(legend.position = "none") +
    geom_point(colour = "Blue") + geom_jitter(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "LogWage vs. age", x = "logWage ($)",
    y = "age")
plot(lwage.age.plot)
```



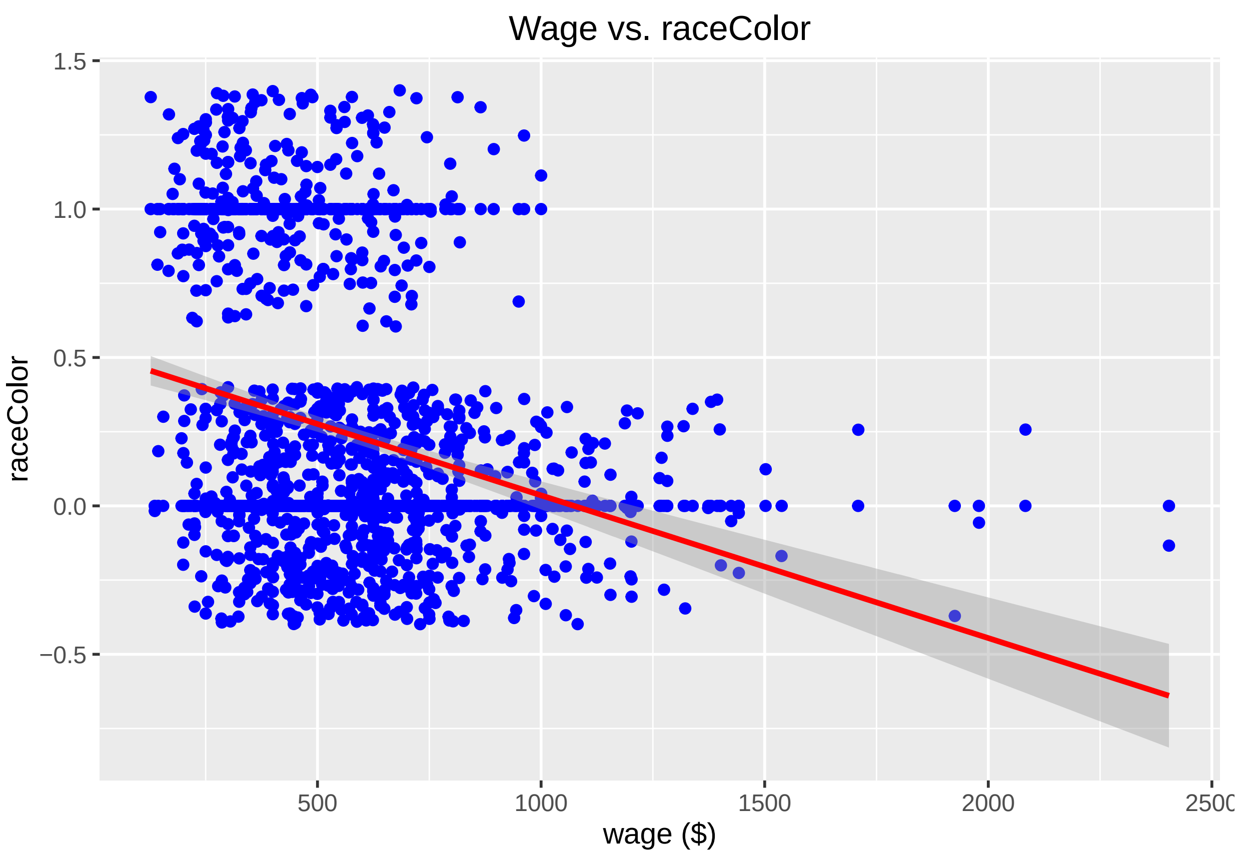

```
# Run correlations with wage and logWage variables
cor(data$wage, data$age)
```

```
## [1] 0.2635783
```

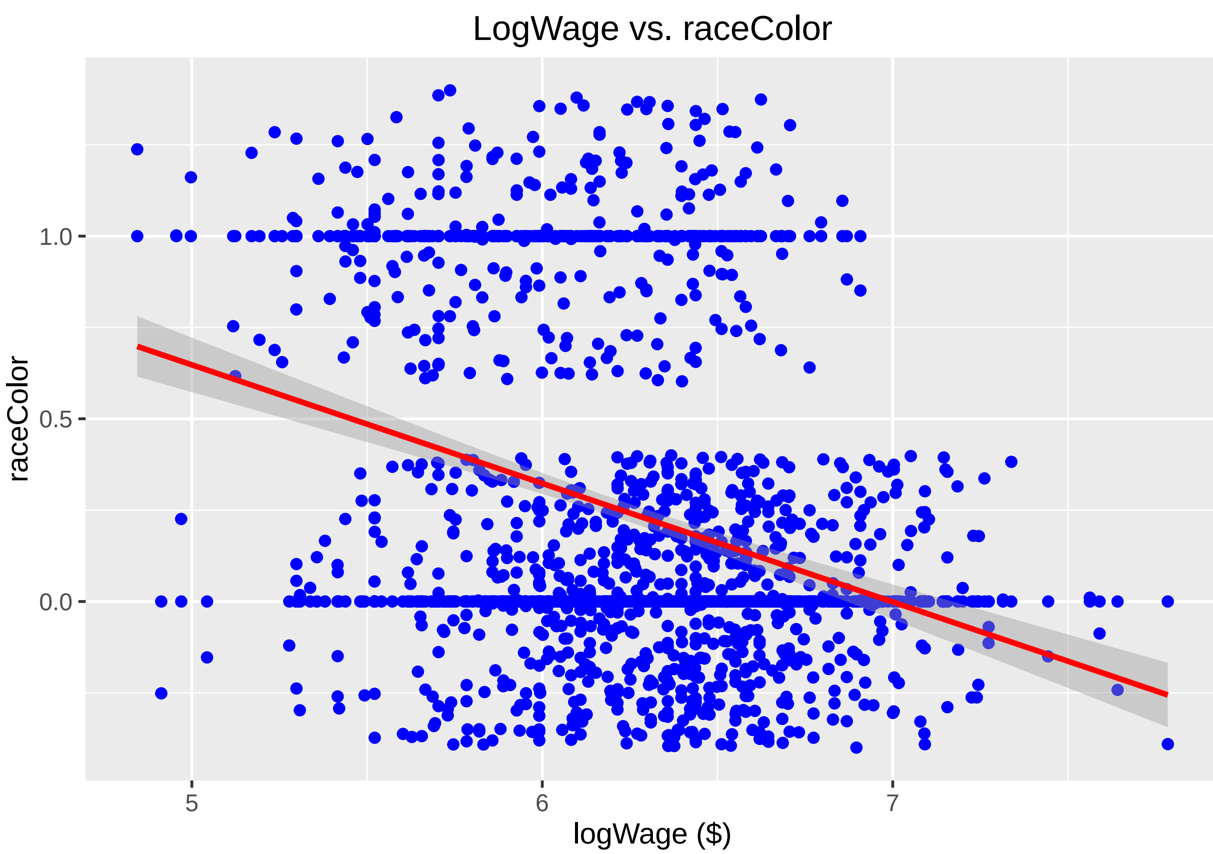
```
cor(data$logWage, data$age)
```

```
## [1] 0.2511202
```

```
# Scatter plot with wage variable
wage.raceColor.plot = ggplot(data, aes(x = wage, y = raceColor)) + theme(legend.position = "none") +
    geom_point(colour = "Blue") + geom_jitter(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "Wage vs. raceColor", x = "wage ($)",
    y = "raceColor")
plot(wage.raceColor.plot)
```




```
# Scatter plot with logWage variable
lwage.raceColor.plot = ggplot(data, aes(x = logWage, y = raceColor)) +
    theme(legend.position = "none") + geom_point(colour = "Blue") + geom_jitter(colour = "Blue") +
    geom_smooth(colour = "red", method = "lm") + labs(title = "LogWage vs. raceColor",
    x = "logWage ($)", y = "raceColor")
plot(lwage.raceColor.plot)
```




```r
# Run correlations with wage and logWage variables
cor(data$wage, data$raceColor)
```

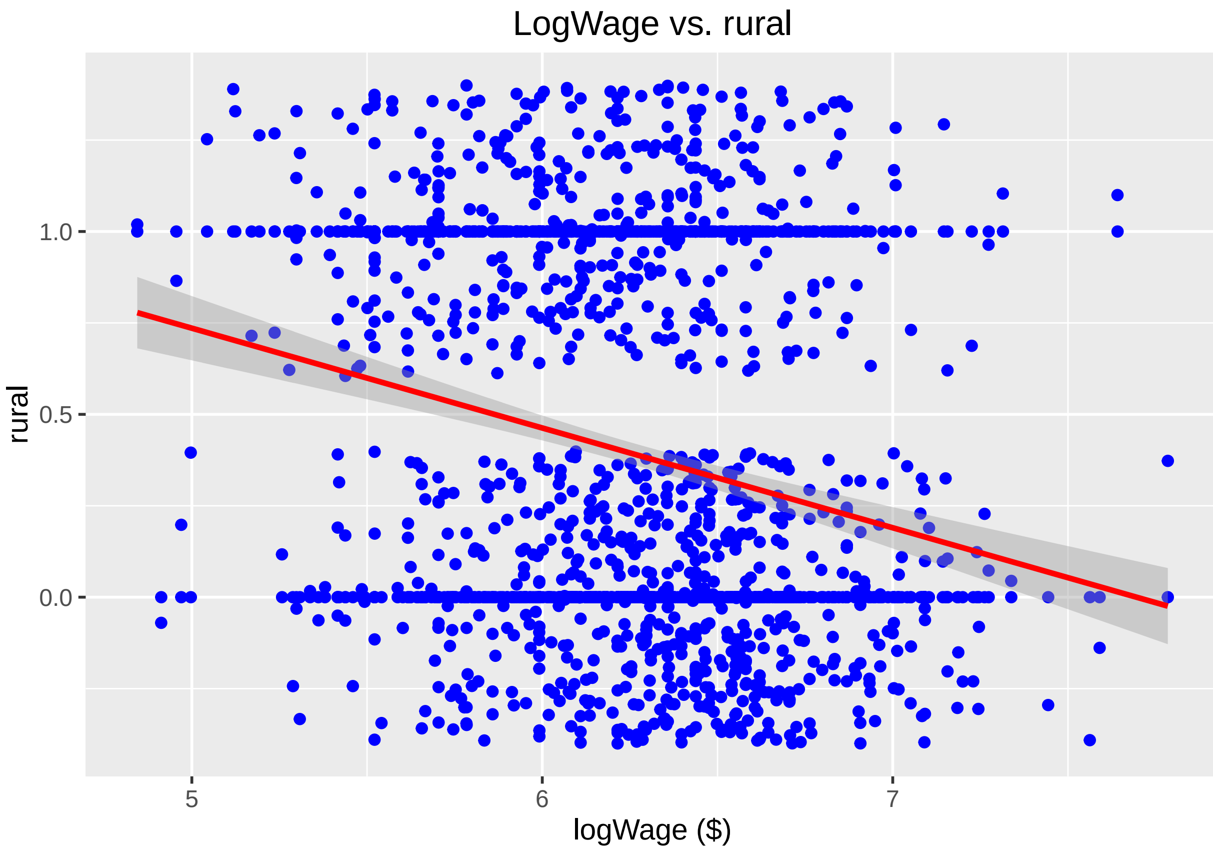
```
## [1] -0.3008475
```

```r
cor(data$logWage, data$raceColor)
```

```
## [1] -0.3407361
```

```r
# Scatter plot with wage variable
wage.rural.plot = ggplot(data, aes(x = wage, y = rural)) + theme(legend.position = "none") +
    geom_point(colour = "Blue") + geom_jitter(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "Wage vs. rural", x = "wage ($)", y = "rural")
plot(wage.rural.plot)
```


```
# Scatter plot with logWage variable
lwage.rural.plot = ggplot(data, aes(x = logWage, y = rural)) + theme(legend.position = "none") +
    geom_point(colour = "Blue") + geom_jitter(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "LogWage vs. rural", x = "logWage ($)",
    y = "rural")
plot(lwage.rural.plot)
```



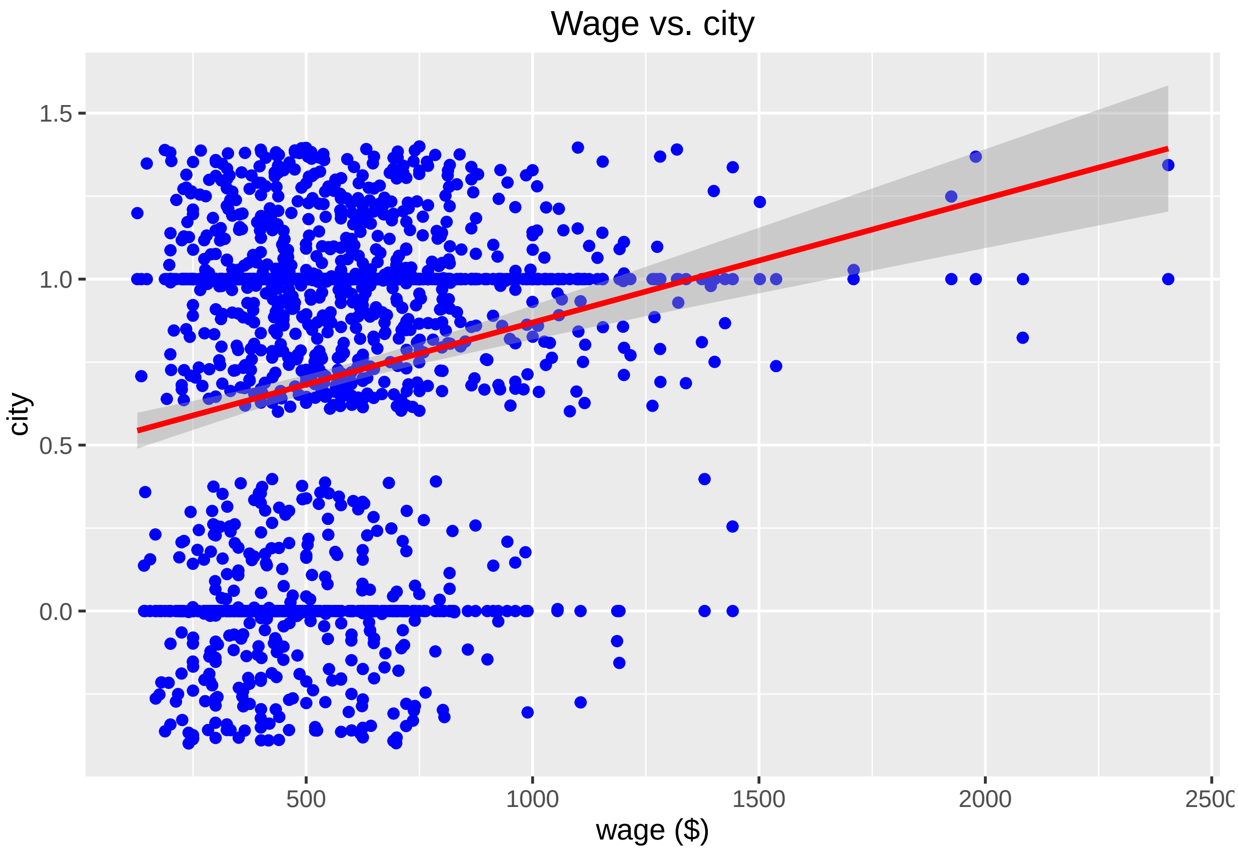

```r
# Run correlations with wage and logWage variables
cor(data$wage, data$rural)
```

```
## [1] -0.2222085
```

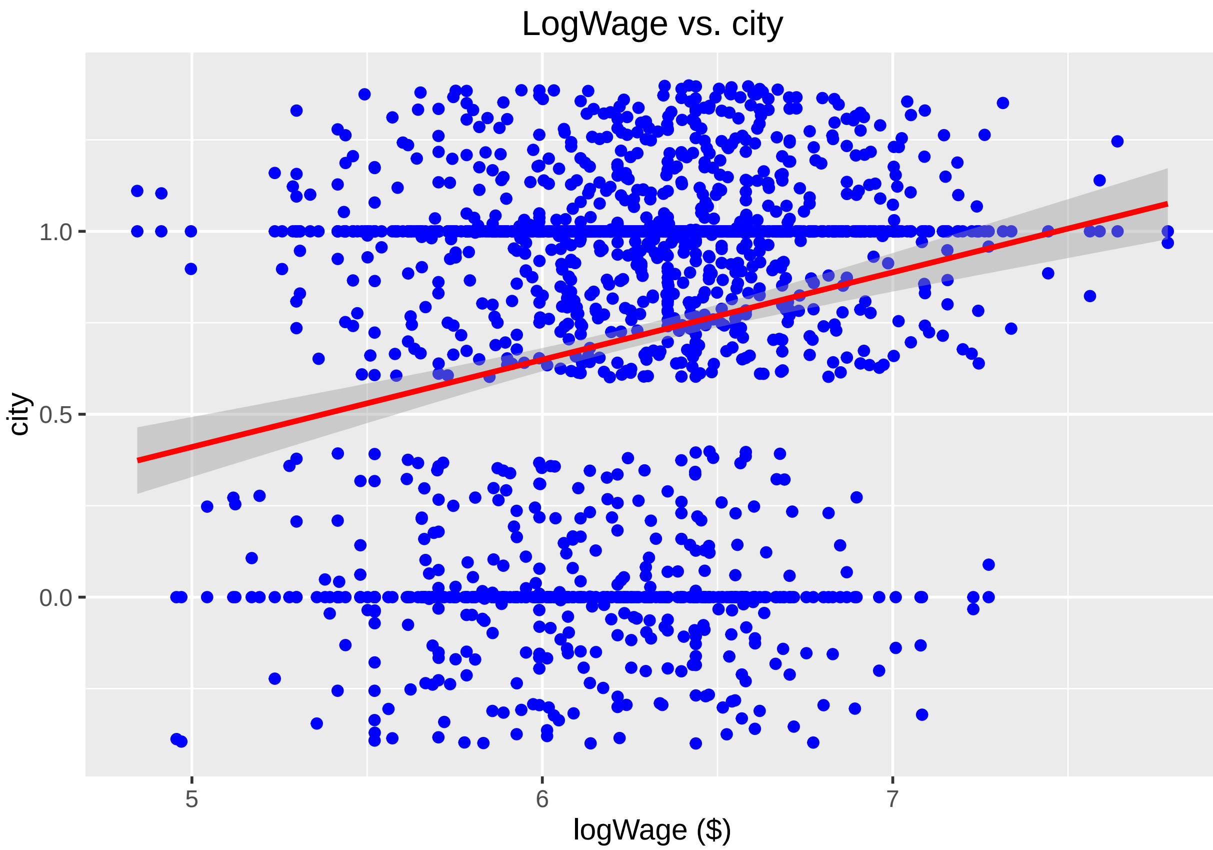
```r
cor(data$logWage, data$rural)
```

```
## [1] -0.2501131
```

```r
# Scatter plot with wage variable
wage.city.plot = ggplot(data, aes(x = wage, y = city)) + theme(legend.position = "none") +
    geom_point(colour = "Blue") + geom_jitter(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "Wage vs. city", x = "wage ($)", y = "city")
plot(wage.city.plot)
```




```
# Scatter plot with logWage variable
lwage.city.plot = ggplot(data, aes(x = logWage, y = city)) + theme(legend.position = "none") +
    geom_point(colour = "Blue") + geom_jitter(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "LogWage vs. city", x = "logWage ($)",
    y = "city")
plot(lwage.city.plot)
```



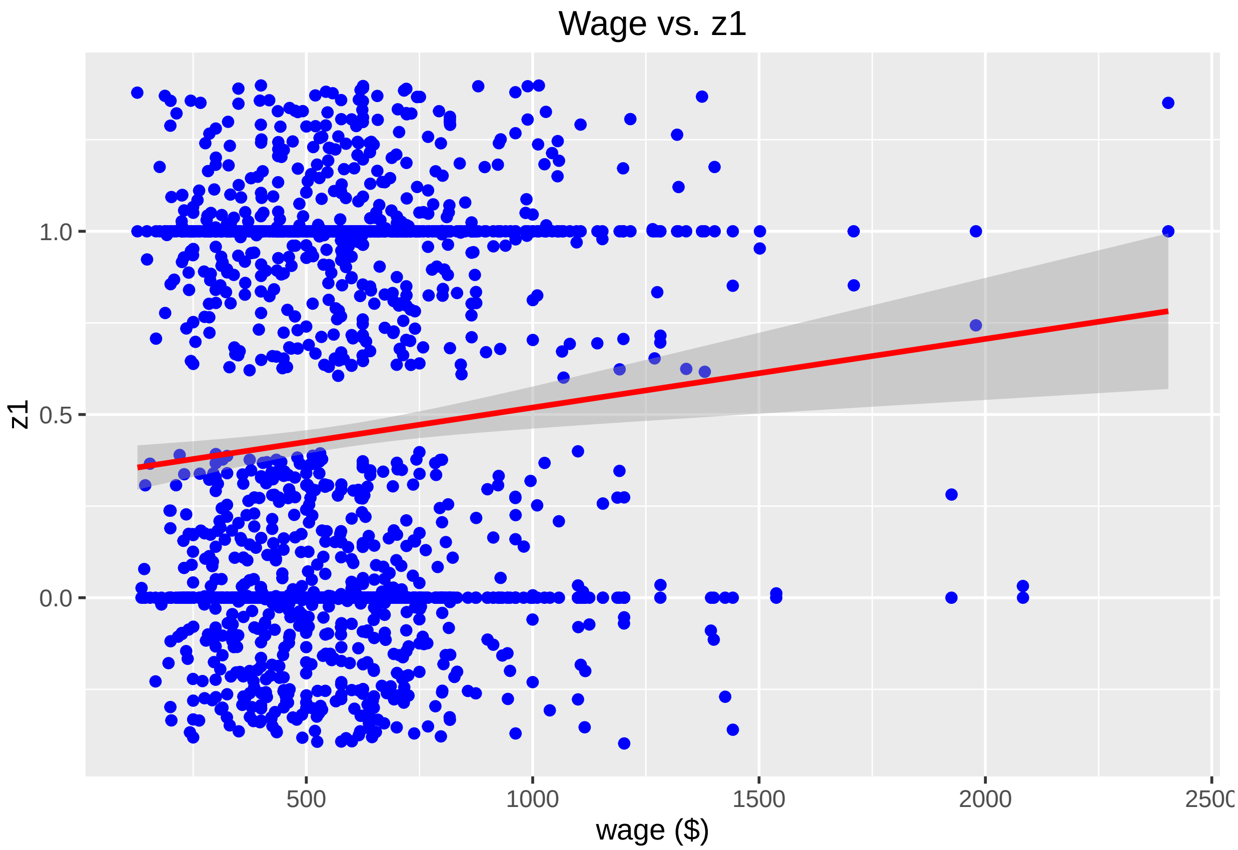

```r
# Run correlations with wage and logWage variables
cor(data$wage, data$city)
```

```
## [1] 0.2196804
```

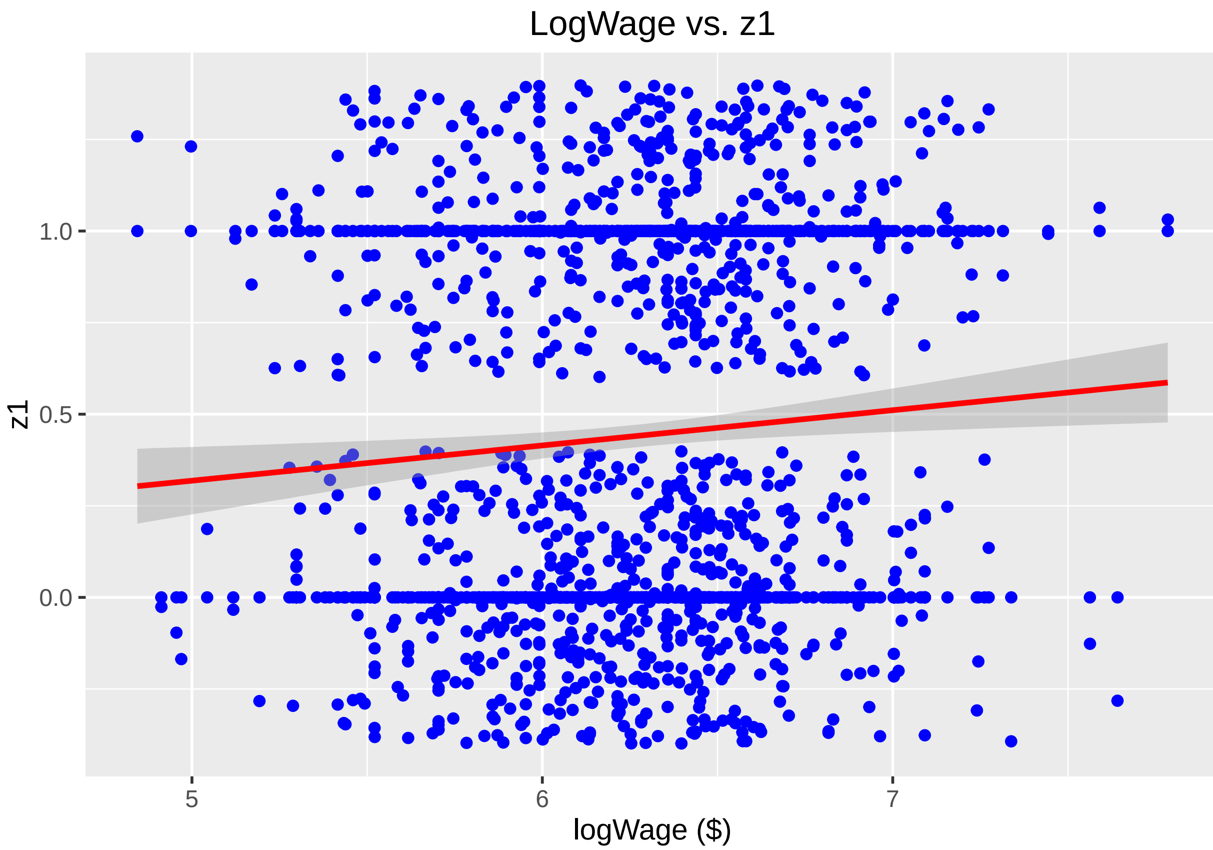
```r
cor(data$logWage, data$city)
```

```
## [1] 0.2358269
```

```r
# Scatter plot with wage variable
wage.z1.plot = ggplot(data, aes(x = wage, y = z1)) + theme(legend.position = "none") +
    geom_point(colour = "Blue") + geom_jitter(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "Wage vs. z1", x = "wage ($)", y = "z1")
plot(wage.z1.plot)
```




```
# Scatter plot with logWage variable
lwage.z1.plot = ggplot(data, aes(x = logWage, y = z1)) + theme(legend.position = "none") +
    geom_point(colour = "Blue") + geom_jitter(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "LogWage vs. z1", x = "logWage ($)",
    y = "z1")
plot(lwage.z1.plot)
```



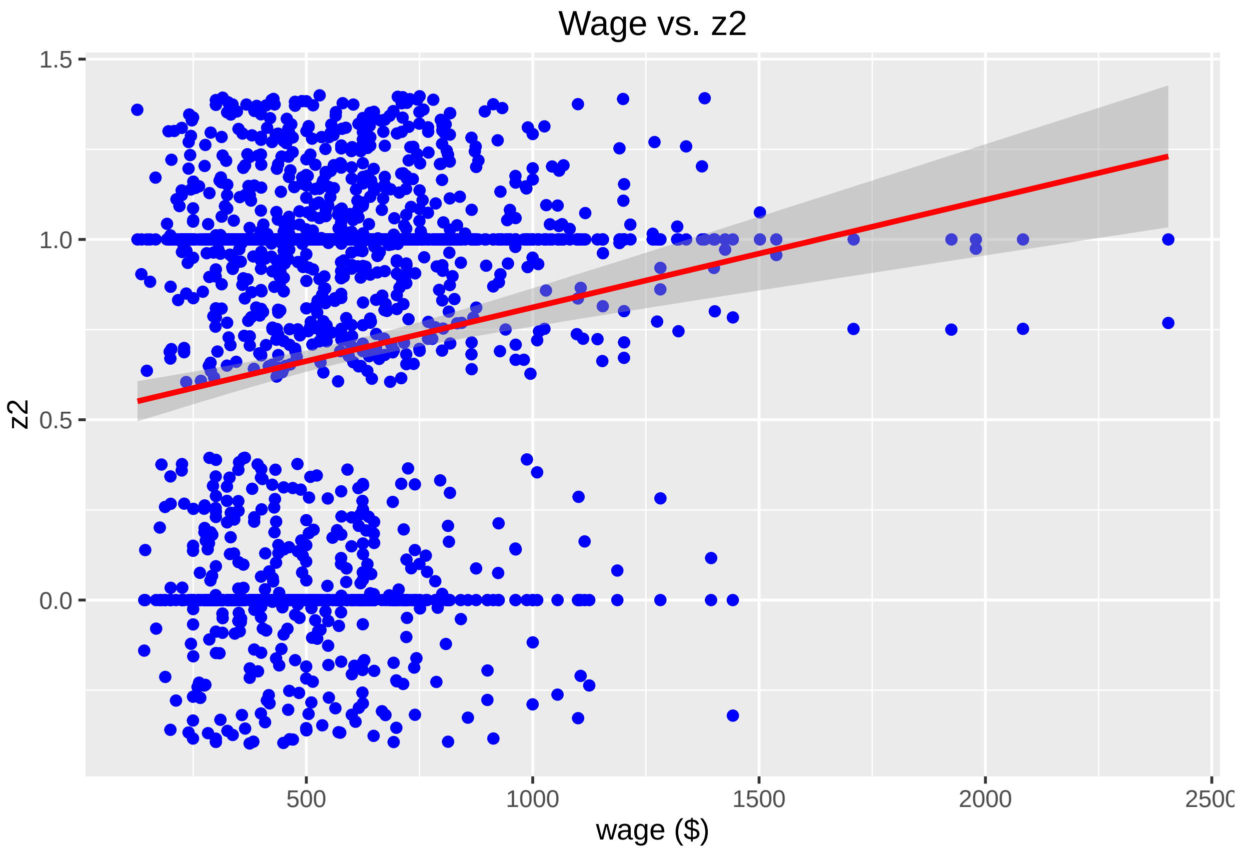

```
# Run correlations with wage and logWage variables
cor(data$wage, data$z1)
```

```
## [1] 0.1005669
```

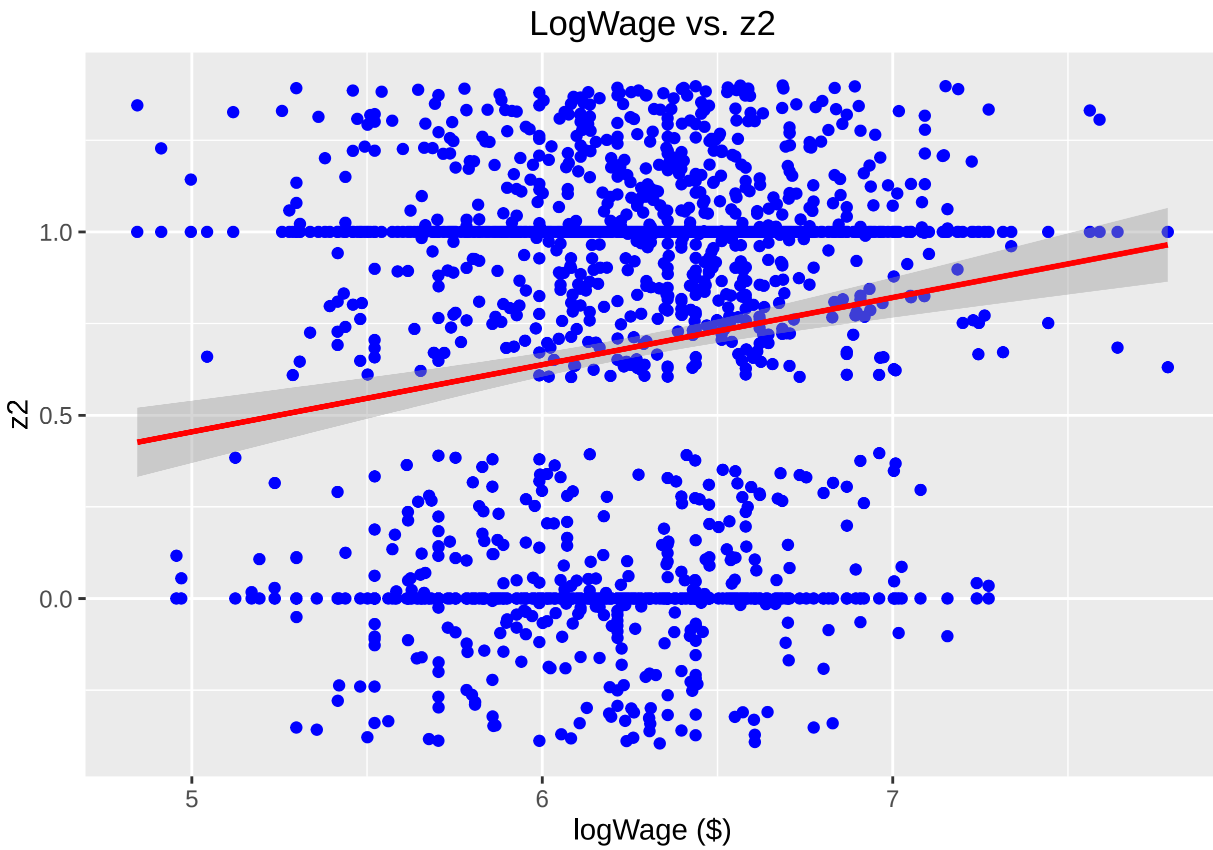
```
cor(data$logWage, data$z1)
```

```
## [1] 0.08668558
```

```
# Scatter plot with wage variable
wage.z2.plot = ggplot(data, aes(x = wage, y = z2)) + theme(legend.position = "none") +
    geom_point(colour = "Blue") + geom_jitter(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "Wage vs. z2", x = "wage ($)", y = "z2")
plot(wage.z2.plot)
```




```r
# Scatter plot with logWage variable
lwage.z2.plot = ggplot(data, aes(x = logWage, y = z2)) + theme(legend.position = "none") +
    geom_point(colour = "Blue") + geom_jitter(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "LogWage vs. z2", x = "logWage ($)",
    y = "z2")
plot(lwage.z2.plot)
```




```
# Run correlations with wage and logWage variables
cor(data$wage, data$z2)
```

```
## [1] 0.1711982
```

```
cor(data$logWage, data$z2)
```

```
## [1] 0.1765267
```

4.3 Regress log(wage) on education, experience, age, and raceColor

Part 1

Report all the estimated coefficients, their standard errors, t-statistics, F-statistic of the regression, R2, adjustedR2, and degrees of freedom
The requested information is shown in the summary information below.

```
OLS.logWage.educ.exper.age.race = lm(logWage ~ education + experience +
    age + raceColor, data = data)
summary(OLS.logWage.educ.exper.age.race)
```

```
##
## Call:
```

```
## lm(formula = logWage ~ education + experience + age + raceColor,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.35396 -0.25550  0.01074  0.24867  1.22932
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.961661   0.113346  43.774   <2e-16 ***
## education    0.079608   0.006376  12.486   <2e-16 ***
## experience   0.035372   0.003988   8.869   <2e-16 ***
## age                NA         NA      NA       NA
## raceColor   -0.260813   0.030453  -8.564   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3917 on 996 degrees of freedom
## Multiple R-squared:  0.236,  Adjusted R-squared:  0.2337
## F-statistic: 102.6 on 3 and 996 DF,  p-value: < 2.2e-16
```

Part 2

Degress of freedom = 996. This value is calculated from the following formula $df = n - k - 1$ where n is the number of observations (n=1000). k is the number of independent varialbes (k=4). Plugging in these values we get, $996 = 1000 - 4 - 1$.

Part 3

The unexpected results from the regression are that the age variable has coeffiecient estimates that are NA. This is because age is a linear combination of the education and experience variables as expressed by the formula $age = education + experience + 6$. To resolve this issue one of these 3 varialbes needs to be removed from the regression. Since the intent is to estimate return to education on race and experience, then the age variable can be removed.

```r
# Create a new variable that represents the linear combination of age
# with education and experience.
data$age.formula = data$education + data$experience + 6
# Show that this new variable isdataeed the same as the age variable to
# subtracting the two variables.
data$age.difference = data$age - data$age.formula
# Now in the summary of the difference variable, all of the values are
# 0 indicating that the age.formula variable is the same as the age
# variable.
summary(data$age.difference)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
```

Part 4 - Interpret the coefficient estimate associated with education

The estimate for the education coefficient is 0.079608. This means that for every unit change in education, there is an 8.00% change in logWage. This value is significant at the 0.1% significance level. This is a small practical effect.

Part 5 - Interpret the coefficient estimate associated with experience

The estimate for the experience coefficient is 0.035372. This means that for every unit change in experience, there is a 3.53% change in logWage. This value is significant at the 0.1% significance level. This is a small practical effect.

Question 4.4

Part 1

See graph below of the estimated effect of experience on wage.

$$\frac{\delta logWage}{\delta experience} = 0.0924 - 2 * (0.00288) * experience$$

Part 2

$$dlogWage10 = 0.0924 - 2 * (0.00288) * 10 = 0.0348$$

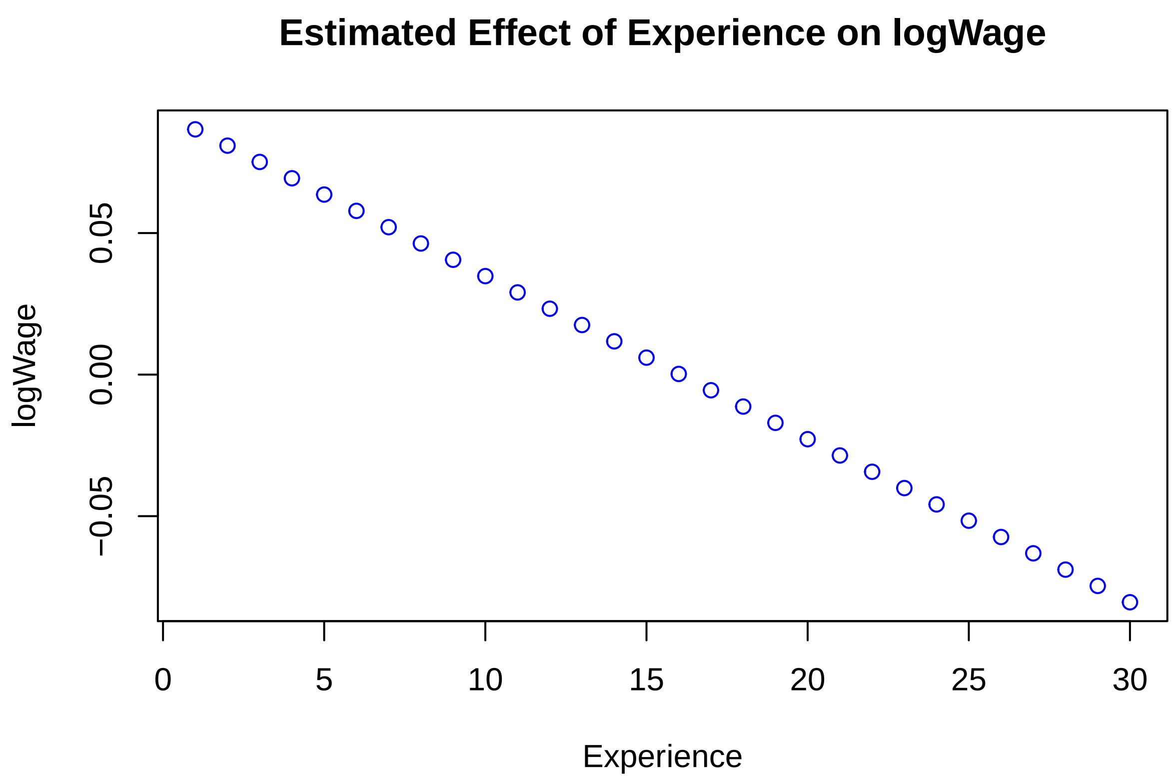
The estimated effect of experience on wage when experience is 10 years is 3.48%.

```r
# Create the model
OLS.logWage.educ.exper.exper2.race = lm(logWage ~ education + experience +
    experienceSquare + raceColor, data = data)
# Print the summary of the model
summary(OLS.logWage.educ.exper.exper2.race)
```

```
##
## Call:
## lm(formula = logWage ~ education + experience + experienceSquare +
##     raceColor, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.38464 -0.25558  0.01909  0.25782  1.24410
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.7355175  0.1197719  39.538  < 2e-16 ***
## education         0.0794641  0.0062917  12.630  < 2e-16 ***
## experience        0.0924930  0.0115147   8.033 2.68e-15 ***
## experienceSquare -0.0028779  0.0005452  -5.279 1.60e-07 ***
## raceColor        -0.2627226  0.0300528  -8.742  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.3865 on 995 degrees of freedom
## Multiple R-squared:  0.2569, Adjusted R-squared:  0.2539
## F-statistic: 85.98 on 4 and 995 DF,  p-value: < 2.2e-16
```

```
# Create a variable dlogWage the represents the line created by the
# change in logWage with respect to a change in experience
dlogWage = 0
for (experience in 1:30) {
    dlogWage[experience] = 0.0924 - 2 * (0.00288) * experience
}
# Graph the line
plot(dlogWage, lty = "dashed", main = "Estimated Effect of Experience on logWage",
    col = "blue", ylab = "logWage", xlab = "Experience")
```




```
# Calculate the value of the effect of experience on wage when
# experience is 10 years.
dlogWage10 = 0.0924 - 2 * (0.00288) * 10
dlogWage10
```

```
## [1] 0.0348
```

Question 4.5

Part 1

The number of observations used in this regression 723 (out of 1,000). The participants with missing mom_education or dad_education values (Group 1) compare to participants that have both a mom_education and a dad_education value (Group 2) as follows.
- **wage** - Group 1 participants have lower median and mean wages than the Group 2 participants. The median and mean for values for Group 1 are $481 and $570 respectively vs. $531 and $597 respectively for Group 2. The standard deviation of Group 1 wages is lower, at 256.9 vs. 268.1 for Group 2. The T-test for difference of means between the 2 groups is significant at the 1% level.
- **education** - Group 1 participants have lower median and mean education than the Group 2 participants. The median and mean for values for Group 1 are 12 and 12.1 respectively vs. 13.7 and 13.7 respectively for Group 2. The standard deviation of Group 1 education is higher, at 2.7 vs. 2.6 for Group 2. The T-test for difference of means between the 2 groups is significant at the 1% level.
- **experience** - Group 1 participants have higher median and mean experience than the Group 2 participants. The median and mean for values for Group 1 are 10 and 10.5 respectively vs. 8 and 8.2 respectively for Group 2. The standard deviation of Group 1 experience is higher, at 4.3 vs. 4.0 for Group 2. The T-test for difference of means between the 2 groups is significant at the 1% level.
- **raceColor** - Group 1 participants have a disproportianately larger number of participants with raceColor=1, at 45%, vs. 16% for Group 2. The T-test for difference of means between the 2 groups is significant at the 1% level.

Part 2

We do not think we can just throw away the participants with the missing values. They are important to the analysis since they represent a disproportional amount of people with lower wages, less education, more experience and higher proportion of raceColor variables equal to 1 than participants without missing values. These differences are statistically significant.

Part 3

This is not a good idea, because averages can be skewed by outliers. Since neither dad_education nor mom_education are evenly distributed, they will inevitably be skewed, hence interfering with our regression, letting outliers have even more influence than they already have on regressions.

Part 4

This is a bad idea because we are introducing multicollinearity into the regression and losing precision of our coefficients.

Part 5

We certainly cannot use the regression models with missing values replaced. Both these techniques lead to highly non-significant coefficients for mom_education and dad_education, meaning that coefficients obtained for these variables cannot be trusted. At the same time, having 277 values missing is not acceptable since our original regression misses a lot of important data for variables for which we do have information. Therefore, we would not elect to go with any of the models from the given choices.

```r
# Part 1 Create the model
OLS.logWage.8var = lm(logWage ~ education + experience + experienceSquare +
    raceColor + dad_education + mom_education + rural + city, data = data)
# Print the model
summary(OLS.logWage.8var)
```

```
##
## Call:
## lm(formula = logWage ~ education + experience + experienceSquare +
##     raceColor + dad_education + mom_education + rural + city,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2961 -0.2240  0.0160  0.2454  1.0404
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.6422296  0.1408825  32.951  < 2e-16 ***
## education         0.0681701  0.0077409   8.806  < 2e-16 ***
## experience        0.0973419  0.0133133   7.312  7.1e-13 ***
## experienceSquare -0.0029568  0.0006678  -4.428  1.1e-05 ***
## raceColor        -0.2130226  0.0425014  -5.012  6.8e-07 ***
## dad_education    -0.0011474  0.0050988  -0.225  0.82202
## mom_education     0.0113176  0.0061886   1.829  0.06785 .
## rural            -0.0919377  0.0314151  -2.927  0.00354 **
## city              0.1782137  0.0323826   5.503  5.2e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3786 on 714 degrees of freedom
##   (277 observations deleted due to missingness)
## Multiple R-squared:  0.2746, Adjusted R-squared:  0.2665
## F-statistic: 33.79 on 8 and 714 DF,  p-value: < 2.2e-16
```

```r
# Creating a dummy variable for rows with missing values
data$missingval = is.na(data$mom_education) | is.na(data$dad_education)
summary(data$missingval)
```

```
##    Mode   FALSE    TRUE    NA's
## logical     723     277       0
```

```r
# Now, we check the variables by the missing value dummy variable.
# Additionally, we check whether there is difference in the two groups
# by running a t-test We do this for wage, education, experience and
# raceColor
by(data$wage, data$missingval, describe)
```

```
## data$missingval: FALSE
##   vars   n   mean     sd median trimmed    mad min  max range skew
## 1    1 723 597.08 268.09    570  569.54 225.36 136 2404  2268 1.39
##   kurtosis   se
```

```
## 1     4.18 9.97
## ------------------------------------------------------------
## data$missingval: TRUE
##   vars   n   mean     sd median trimmed    mad min  max range skew
## 1    1 277 531.03 256.94    481  502.57 213.49 127 2083  1956 1.96
##   kurtosis    se
## 1      7.5 15.44
```

```
t.test(data[data$missingval, c("wage")], data[!data$missingval, c("wage")])
```

```
## 
##  Welch Two Sample t-test
## 
## data:  data[data$missingval, c("wage")] and data[!data$missingval, c("wage")]
## t = -3.5943, df = 519.7, p-value = 0.0003562
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -102.15868  -29.95122
## sample estimates:
## mean of x mean of y 
##  531.0253  597.0802
```

```
by(data$education, data$missingval, describe)
```

```
## data$missingval: FALSE
##   vars   n  mean   sd median trimmed  mad min max range  skew kurtosis  se
## 1    1 723 13.65 2.62     13   13.71 2.97   3  18    15 -0.28      0.1 0.1
## ------------------------------------------------------------
## data$missingval: TRUE
##   vars   n  mean  sd median trimmed  mad min max range  skew kurtosis   se
## 1    1 277 12.09 2.7     12   12.13 1.48   2  18    16 -0.18     0.85 0.16
```

```
t.test(data[data$missingval, c("education")], data[!data$missingval, c("education")])
```

```
## 
##  Welch Two Sample t-test
## 
## data:  data[data$missingval, c("education")] and data[!data$missingval, c("education")]
## t = -8.2548, df = 486.38, p-value = 1.443e-15
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.932803 -1.189596
## sample estimates:
## mean of x mean of y 
##  12.09025  13.65145
```

```
by(data$experience, data$missingval, describe)
```

```
## data$missingval: FALSE
##   vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
## 1    1 723 8.15 4.01      8    7.89 2.97   0  21    21 0.62     0.11 0.15
```

```
## --------------------------------------------------------
## data$missingval: TRUE
##   vars   n  mean   sd median trimmed  mad min max range skew kurtosis   se
## 1    1 277 10.47 4.32     10    10.3 4.45   0  23    23 0.33    -0.51 0.26
```

```r
t.test(data[data$missingval, c("experience")], data[!data$missingval, c("experience")])
```

```
##
##  Welch Two Sample t-test
##
## data:  data[data$missingval, c("experience")] and data[!data$missingval, c("experience")]
## t = 7.7605, df = 468.78, p-value = 5.344e-14
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.732905 2.908047
## sample estimates:
## mean of x mean of y
## 10.465704  8.145228
```

```r
by(data$raceColor, data$missingval, describe)
```

```
## data$missingval: FALSE
##   vars   n mean   sd median trimmed mad min max range skew kurtosis   se
## 1    1 723 0.16 0.36      0    0.07   0   0   1     1 1.87     1.52 0.01
## --------------------------------------------------------
## data$missingval: TRUE
##   vars   n mean  sd median trimmed mad min max range skew kurtosis   se
## 1    1 277 0.45 0.5      0    0.43   0   0   1     1 0.21    -1.96 0.03
```

```r
t.test(data[data$missingval, c("raceColor")], data[!data$missingval, c("raceColor")])
```

```
##
##  Welch Two Sample t-test
##
## data:  data[data$missingval, c("raceColor")] and data[!data$missingval, c("raceColor")]
## t = 8.8244, df = 394.62, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2253732 0.3545809
## sample estimates:
## mean of x mean of y
## 0.4476534 0.1576763
```

```r
# Part 3 Copy the dataset to a new variable
data.avgForNA = data
# Set all of the values with dad_education = NA to the mean of
# dad_education
data.avgForNA$dad_education[is.na(data.avgForNA$dad_education)] = mean(data.avgForNA$dad_education,
    na.rm = TRUE)
# Set all of the values with mom_education = NA to the mean of
# mom_education
```

```r
data.avgForNA$mom_education[is.na(data.avgForNA$mom_education)] = mean(data.avgForNA$mom_education,
    na.rm = TRUE)
# Rerun the regression
OLS.logWage.8var.avgNA = lm(logWage ~ education + experience + experienceSquare +
    raceColor + dad_education + mom_education + rural + city, data = data.avgForNA)

# Part 4 Copy the dataset to a new variable
data.regressForNA = data
# Regress dad_education on the education, experience and raceColor
# variables
m1 = lm(dad_education ~ education + experience + raceColor, data = data)
# Regress mom_education on the education, experience and raceColor
# variables
m2 = lm(mom_education ~ education + experience + raceColor, data = data)

# Set all of the values with dad_education = NA to the value output
# from using the regression coefficients from m1 above.
data.regressForNA$dad_education[is.na(data.regressForNA$dad_education)] = m1$coefficients[1] +
    m1$coefficients[2] * data.regressForNA$education + m1$coefficients[3] *
    data.regressForNA$experience + m1$coefficients[4] * data.regressForNA$raceColor
```

```
## Warning in data.regressForNA$dad_education[is.na(data.regressForNA
## $dad_education)] = m1$coefficients[1] + : number of items to replace is not
## a multiple of replacement length
```

```r
# Set all of the values with mom_education = NA to the value output
# from using the regression coefficients from m2 above.
data.regressForNA$mom_education[is.na(data.regressForNA$mom_education)] = m2$coefficients[1] +
    m2$coefficients[2] * data.regressForNA$education + m2$coefficients[3] *
    data.regressForNA$experience + m2$coefficients[4] * data.regressForNA$raceColor
```

```
## Warning in data.regressForNA$mom_education[is.na(data.regressForNA
## $mom_education)] = m2$coefficients[1] + : number of items to replace is not
## a multiple of replacement length
```

```r
# Rerun the regression
OLS.logWage.8var.regressNA = lm(logWage ~ education + experience + experienceSquare +
    raceColor + dad_education + mom_education + rural + city, data = data.regressForNA)

# Part 5 Print the summaries of the 2 new models
summary(OLS.logWage.8var.avgNA)
```

```
##
## Call:
## lm(formula = logWage ~ education + experience + experienceSquare +
##     raceColor + dad_education + mom_education + rural + city,
##     data = data.avgForNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.30741 -0.23286  0.01943  0.24786  1.28807
##
```

```
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.729e+00  1.226e-01  38.584  < 2e-16 ***
## education          7.097e-02  6.499e-03  10.920  < 2e-16 ***
## experience         8.958e-02  1.124e-02   7.970 4.36e-15 ***
## experienceSquare -2.678e-03  5.318e-04  -5.036 5.65e-07 ***
## raceColor        -2.313e-01  3.099e-02  -7.464 1.84e-13 ***
## dad_education    -3.513e-05  4.416e-03  -0.008 0.993656
## mom_education     3.485e-03  5.009e-03   0.696 0.486742
## rural            -9.529e-02  2.638e-02  -3.612 0.000319 ***
## city              1.671e-01  2.703e-02   6.183 9.21e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3764 on 991 degrees of freedom
## Multiple R-squared:  0.2981, Adjusted R-squared:  0.2925
## F-statistic: 52.62 on 8 and 991 DF,  p-value: < 2.2e-16
```

```r
summary(OLS.logWage.8var.regressNA)
```

```
##
## Call:
## lm(formula = logWage ~ education + experience + experienceSquare +
##     raceColor + dad_education + mom_education + rural + city,
##     data = data.regressForNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.30770 -0.23222  0.02095  0.24785  1.29770
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.7278751  0.1228090  38.498  < 2e-16 ***
## education          0.0710341  0.0064659  10.986  < 2e-16 ***
## experience         0.0896724  0.0112433   7.976 4.16e-15 ***
## experienceSquare -0.0026820  0.0005318  -5.043 5.45e-07 ***
## raceColor        -0.2313406  0.0311112  -7.436 2.24e-13 ***
## dad_education    -0.0003385  0.0041318  -0.082 0.934718
## mom_education     0.0037753  0.0047649   0.792 0.428365
## rural            -0.0952834  0.0263780  -3.612 0.000319 ***
## city              0.1673210  0.0270228   6.192 8.70e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3764 on 991 degrees of freedom
## Multiple R-squared:  0.2982, Adjusted R-squared:  0.2925
## F-statistic: 52.64 on 8 and 991 DF,  p-value: < 2.2e-16
```

Question 4.6

Part 1

The assumptions needed are Cov(z1, education) != 0 and Cov(z1, u) = 0.

Part 2

Suppose z1 is an indicator representing whether or not an individual lives in an area in which there was a recent policy change to promote the importance of education. Yes, z1 could be correlated with other unobservables captured in the error term. Some examples are 1. Income. People with higher incomes might be more educated and thus might place a higher importance on eduction and thus be more likely to live in an area that promotes education, 2. Political party. A particular political party might be more aligned with education and therefore people in that polical party might be more inclined to live in an area that promotes education, and 3. Whether you voted or not. It's possible that people who vote might be more educated and more likely to live in an area that promotes education. These are just a few examples. There could be many more.

Part 3

Using the same specification as that in question 4.5, estimate the equation by 2SLS, using both z1 and z2 as instrument variables.

The coefficient estimate on education goes from 0.0681701 in the original model to 0.0950302, however, in the new model, the education estimate is not significant at the 5% level, so the increase in the coefficient can no longer be used in our interpretation.

However, if we remove mom_education and dad_education from both the TSLS and original models, the education coefficient becomes significant again at the 5% level. The value of the education coefficient now goes from 0.0722416 in the original model to 0.1042749 in the TSLS model. This means that using z1 and z2 as instrumental variables the effect of educaiton on logWage increases from about 7.2% to 10.4% (an increase of about 3 percentage points). This is a 44% increase which is a large practical effect.

```r
# Run the IV TSLS regression with z1 and z2
TSLS.logWage.8var = ivreg(logWage ~ education + experience + experienceSquare +
    raceColor + dad_education + mom_education + rural + city | z1 * z2 +
    experience + experienceSquare + raceColor + dad_education + mom_education +
    rural + city, data = data)
# Print the summary of TSLS the model
summary(TSLS.logWage.8var)
```

```
##
## Call:
## ivreg(formula = logWage ~ education + experience + experienceSquare +
##     raceColor + dad_education + mom_education + rural + city |
##     z1 * z2 + experience + experienceSquare + raceColor + dad_education +
##         mom_education + rural + city, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.31628 -0.23169  0.03689  0.23949  1.03574
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.2815365  0.8256004   5.186 2.80e-07 ***
## education         0.0950302  0.0610647   1.556  0.12010
## experience        0.1069713  0.0255275   4.190 3.13e-05 ***
## experienceSquare -0.0030032  0.0006815  -4.407 1.21e-05 ***
```

```
## raceColor        -0.2001502  0.0517616  -3.867  0.00012 ***
## dad_education    -0.0041758  0.0085477  -0.489  0.62533
## mom_education     0.0071767  0.0112304   0.639  0.52300
## rural            -0.0888567  0.0324316  -2.740  0.00630 **
## city              0.1670192  0.0412727   4.047 5.76e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3818 on 714 degrees of freedom
## Multiple R-Squared: 0.2624,  Adjusted R-squared: 0.2541
## Wald test:    24 on 8 and 714 DF,  p-value: < 2.2e-16
```

```
# Print the summary of the original model
summary(OLS.logWage.8var)
```

```
##
## Call:
## lm(formula = logWage ~ education + experience + experienceSquare +
##     raceColor + dad_education + mom_education + rural + city,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2961 -0.2240  0.0160  0.2454  1.0404
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.6422296  0.1408825  32.951  < 2e-16 ***
## education         0.0681701  0.0077409   8.806  < 2e-16 ***
## experience        0.0973419  0.0133133   7.312  7.1e-13 ***
## experienceSquare -0.0029568  0.0006678  -4.428  1.1e-05 ***
## raceColor        -0.2130226  0.0425014  -5.012  6.8e-07 ***
## dad_education    -0.0011474  0.0050988  -0.225  0.82202
## mom_education     0.0113176  0.0061886   1.829  0.06785 .
## rural            -0.0919377  0.0314151  -2.927  0.00354 **
## city              0.1782137  0.0323826   5.503  5.2e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3786 on 714 degrees of freedom
##   (277 observations deleted due to missingness)
## Multiple R-squared:  0.2746, Adjusted R-squared:  0.2665
## F-statistic: 33.79 on 8 and 714 DF,  p-value: < 2.2e-16
```

Question 5

Part 1

In order to come up with our parsimonious model, we first examined the dataset. We found high correlation between urb and lit, and therefore chose not to use those in order to prevent the negative effects of multicollinearity in our results. Since the research question is concerned with voteshare and absolute_wealth, we choose a simple univariate model with the dependent variable as voteshare and the dependent variable as absolute wealth. However, from examining this variable, it is clear that it is heavily positively skewed, and therefore requires a log transformation. Additionally, some cleanup is required, removing coded values.

Final Parsimonious Model: y = voteshare, x = absolute_wealth

Results from regression: Our model is statistically significant at the 1% level. We can interpret the coefficient as saying a 1% increase in absolute wealth corresponds to a 0.005% increase in votes Answering Research Question: Wealthy candidates fare very slightly better in elections. There is a linear relationship, but with a very small slope, such that it is almost flat.

```
dataset = read.csv("wealthy_candidates.csv")
# Exploring dataset
describe(dataset$absolute_wealth)
```

```
##   vars    n    mean       sd  median trimmed     mad min        max
## 1    1 2497 5034105 31098493 1336629 2168762 1981683   2 1216399232
##        range  skew kurtosis       se
## 1 1216399230 29.33  1028.72 622343.4
```

```
cor(dataset[, c("urb", "lit", "voteshare", "absolute_wealth")], use = "pairwise.complete.obs")
```

```
##                        urb        lit   voteshare absolute_wealth
## urb             1.00000000 0.64682427 0.033492574     0.012277317
## lit             0.64682427 1.00000000 0.037997050     0.019583187
## voteshare       0.03349257 0.03799705 1.000000000     0.001370482
## absolute_wealth 0.01227732 0.01958319 0.001370482     1.000000000
```

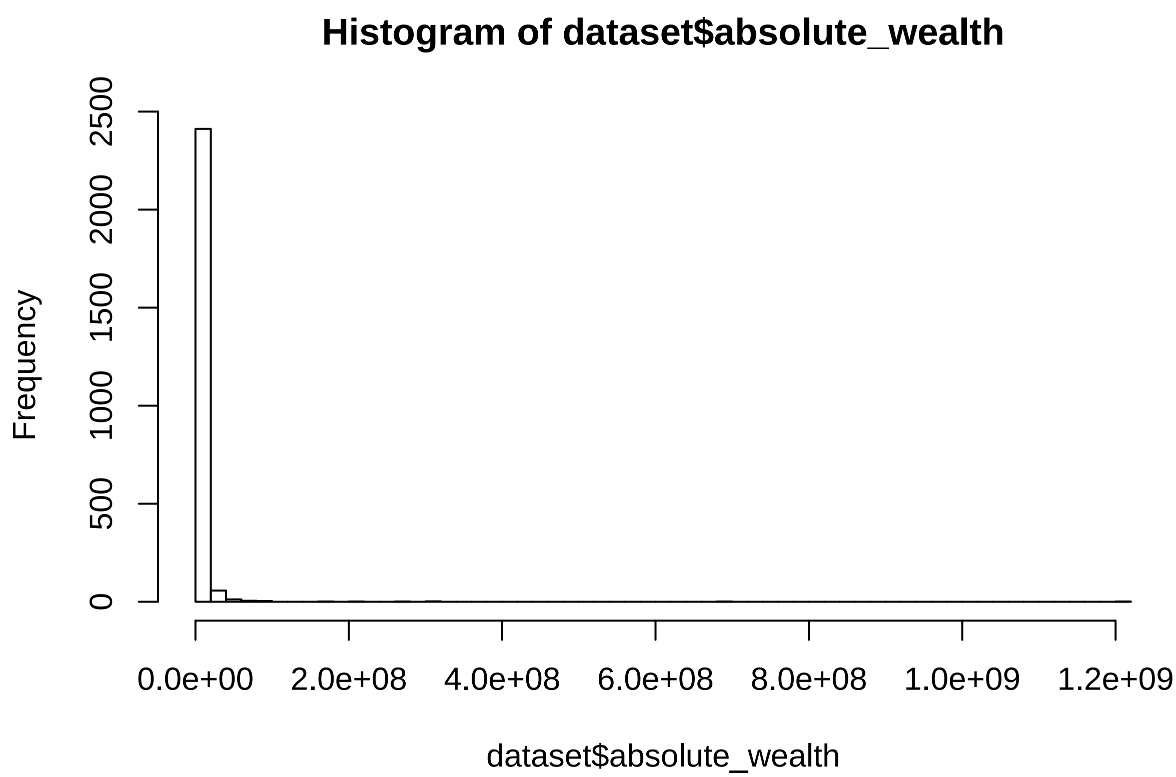
```
# Now, examining abs wealth variable
summary(dataset$absolute_wealth)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
## 2.000e+00 1.875e+05 1.337e+06 5.034e+06 4.092e+06 1.216e+09         1
```

```
print(quantile(dataset$absolute_wealth, probs = c(0.01, 0.05, 0.1, 0.25,
    0.5, 0.75, 0.9, 0.95, 0.99, 1), na.rm = TRUE))
```

```
##         1%         5%        10%        25%        50%        75%
##          2          2          2     187500    1336629    4092001
##        90%        95%        99%       100%
##   10036608   15860393   40552757 1216399232
```

```
hist(dataset$absolute_wealth, breaks = 60)
```




```r
head(dataset[order(dataset$absolute_wealth, decreasing = TRUE), c("absolute_wealth")])
```

```
## [1] 1216399232  699396480  308832992  301821632  268619840  209518016
```

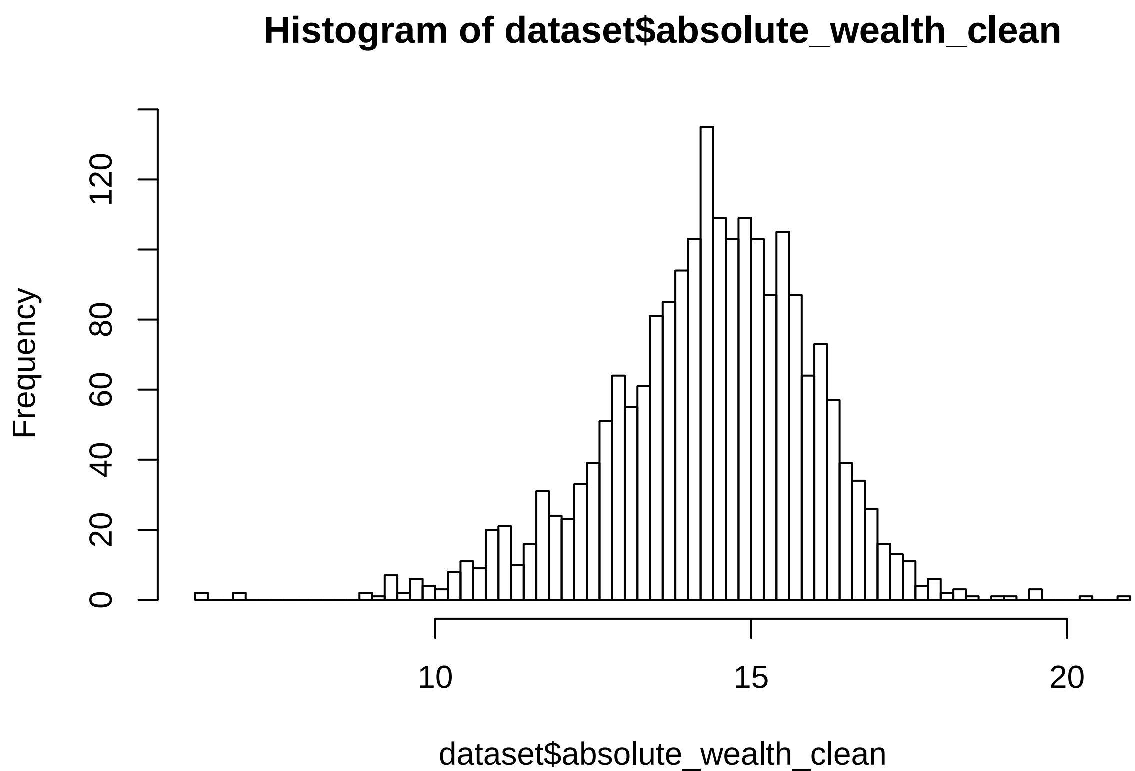
```r
# We can see that this variable is highly skewed. In order continue
# using the variable, we must remove coded values like 2, and transform
# the variable to log.
dataset$absolute_wealth_clean = log(dataset$absolute_wealth)
dataset$absolute_wealth_clean[dataset$absolute_wealth_clean == log(2)] = NA
print(quantile(dataset$absolute_wealth_clean, probs = c(0.01, 0.05, 0.1,
    0.25, 0.5, 0.75, 0.9, 0.95, 0.99, 1), na.rm = TRUE))
```

```
##        1%        5%       10%       25%       50%       75%       90%
##  9.747344 11.278626 12.226369 13.444402 14.442036 15.455854 16.236921
##       95%       99%      100%
## 16.708445 17.659474 20.919161
```

```r
summary(dataset$absolute_wealth_clean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   6.217  13.440  14.440  14.340  15.460  20.920     436
```

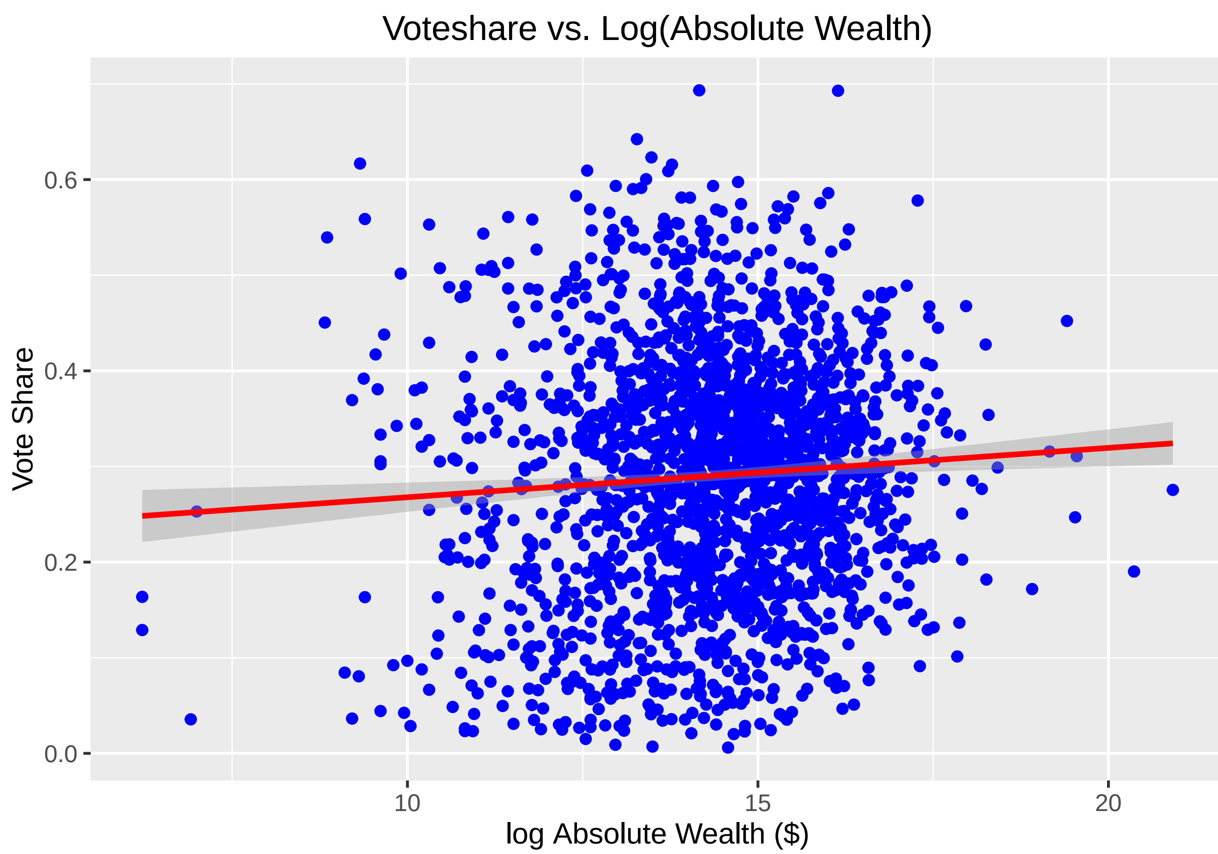
```r
hist(dataset$absolute_wealth_clean, breaks = 60)
```




```r
# Now, to start building the regression model
votes.plot = ggplot(dataset, aes(x = absolute_wealth_clean, y = voteshare)) +
    theme(legend.position = "none") + geom_point(colour = "Blue") + geom_smooth(colour = "red",
    method = "lm") + labs(title = "Voteshare vs. Log(Absolute Wealth)",
    x = "log Absolute Wealth ($)", y = "Vote Share")
plot(votes.plot)
```

```
## Warning: Removed 436 rows containing non-finite values (stat_smooth).

## Warning: Removed 436 rows containing missing values (geom_point).
```




```r
# Does not seem like much of a relation, but we continue on to run the
# regression.
model = lm(voteshare ~ absolute_wealth_clean, data = dataset)
summary(model)
```

```
##
## Call:
## lm(formula = voteshare ~ absolute_wealth_clean, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28540 -0.09048  0.00238  0.08018  0.40401
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.216181   0.024163   8.947  < 2e-16 ***
## absolute_wealth_clean 0.005164   0.001674   3.084  0.00207 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1251 on 2060 degrees of freedom
##   (436 observations deleted due to missingness)
## Multiple R-squared:  0.004597,	Adjusted R-squared:  0.004113
## F-statistic: 9.513 on 1 and 2060 DF,  p-value: 0.002068
```

Part 2

An addition of a quadratic term is absolutely unwarranted, and would only skew the original absolute wealth variable further. For comparison purposes, we will create a new model with the wealth variable without the log, and with the square.

Result: Highly non-significant model and coefficients, cannot reject the null.

```r
# Creating a clean variable without the log
dataset$absolute_wealth_clean2 = dataset$absolute_wealth
dataset$absolute_wealth_clean2[dataset$absolute_wealth_clean2 == 2] = NA
dataset$absolute_wealth_clean2Square = dataset$absolute_wealth_clean2^2
model2 = lm(voteshare ~ absolute_wealth_clean2 + absolute_wealth_clean2Square,
    data = dataset)
summary(model2)
```

```
##
## Call:
## lm(formula = voteshare ~ absolute_wealth_clean2 + absolute_wealth_clean2Square,
##     data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28391 -0.09005  0.00591  0.08069  0.40345
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   2.897e-01  2.944e-03  98.407   <2e-16 ***
## absolute_wealth_clean2        1.054e-10  2.029e-10   0.519    0.603
## absolute_wealth_clean2Square -1.209e-19  2.008e-19  -0.602    0.547
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1254 on 2059 degrees of freedom
##   (436 observations deleted due to missingness)
## Multiple R-squared:  0.0001792,  Adjusted R-squared:  -0.000792
## F-statistic: 0.1845 on 2 and 2059 DF,  p-value: 0.8315
```

Part 3

We run a new model with dummy variables for region 2 and region 3. With this model, we obtain statistical and practical significance of the dummy coefficients, as well as a substantial increase in the R squared value from the original parsimonious model.

Ater testing the difference in models, we obtain a significant wald test as well, showing that the region variables are clearly a good addition to the model.

```r
model3 = lm(voteshare ~ absolute_wealth_clean + region, data = dataset)
summary(model3)
```

```
##
## Call:
## lm(formula = voteshare ~ absolute_wealth_clean + region, data = dataset)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31780 -0.08715  0.00944  0.08108  0.39472
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.087563   0.028181   3.107  0.00191 **
## absolute_wealth_clean 0.012038   0.001832   6.570 6.36e-11 ***
## regionRegion 2        0.040562   0.006914   5.866 5.17e-09 ***
## regionRegion 3        0.060842   0.007222   8.425  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.123 on 2058 degrees of freedom
##   (436 observations deleted due to missingness)
## Multiple R-squared:  0.03936,    Adjusted R-squared:  0.03796
## F-statistic: 28.11 on 3 and 2058 DF,  p-value: < 2.2e-16
```

```r
summary(model)
```

```
##
## Call:
## lm(formula = voteshare ~ absolute_wealth_clean, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28540 -0.09048  0.00238  0.08018  0.40401
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.216181   0.024163   8.947  < 2e-16 ***
## absolute_wealth_clean 0.005164   0.001674   3.084  0.00207 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1251 on 2060 degrees of freedom
##   (436 observations deleted due to missingness)
## Multiple R-squared:  0.004597,    Adjusted R-squared:  0.004113
## F-statistic: 9.513 on 1 and 2060 DF,  p-value: 0.002068
```

```r
waldtest(model, model3, vcov = vcovHC)
```

```
## Wald test
##
## Model 1: voteshare ~ absolute_wealth_clean
## Model 2: voteshare ~ absolute_wealth_clean + region
##   Res.Df Df      F    Pr(>F)
## 1   2060
## 2   2058  2 40.791 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Part 4

In our parsimonious model, our errors are endogenous, leading to the omitted variable bias in our coefficient for log(absolute wealth). This is evident from the fact that when we add region, we see a drastic change in the coefficient for log(absolute wealth). Therefore, we cannot say that we have a causal, unbiased estimate, because we know our coefficient is biased. Causality holds when we have (apart from MLR1-MLR4) 1. Exogeneity of errors (which is volated in this case), and 2. the ability to manipulate x to observe changes in y without affecting the error term. We could theoretically conceive of a situation where we find people following an ideal absolute wealth distribution and have them run for elections, observing the results. However, this is not practical in this case.

Part 5

$$Change\ in\ Voteshare = \beta_0 + \beta_1 * (Change\ in\ log\ absolute\ wealth) + u$$

This model would yield a causal result when the error terms that are endogenous, are also time constant, and would therefore cancel out.

However, in our case, this model does not work for several reasons. 1. We do not have data across time periods. 2. If we assume that we do have data across time, one could argue that the variable absolute wealth is close to being time-constant, and would therefore mostly cancel out, which would mean we would lose our main independent variable. 3. Changes in absolute wealth and its affect on the change in votes does not help answer our original research question. A poorer candidate could have a larger change in wealth than a richer candidate, and we would lose this informaation by doing a difference model.

Question6

Part 1 - Reorganizing the data

With the data loaded, we can see that it includes product data for a period of four years of 2004, 2005, 2006 and 2007. The organization of the data suggests that a transformation to wide form would make the analysis easier. The first step of our analysis is to transform the data using the reshape function such that all product information for the four years of 2004-2007 is represented on a single row.

```
load("retailSales.Rdata")
```

Part 2 - Variable analysis and establishing a population model

The variables in the dataset provided to us are: Year, Product.line, Product.type, Product, Order.method-type, Retailer.country, Revenue, Planned.revenue, Product.cost, Quantity, Unit.cost, Unit.price, Gross.profit and Unit.sale.price. Without additional information about these variables, we want to know which ones would introduce multi-collinearity in our model, should we decide to include them jointly and which ones may require transformations.

We want build a model to predict revenue from the variables available, using data from the first two years. We will build a model predicting revenues in 2005 from 2004 data and will validate that model using 2006 data to predict 2007 data.

Part 2.1 Unit.price and Unit.sale.price

The correlation between these two variables is 0.999275 indicating that only one of these two variables should be part of our model. We choose to drop the Unit.sale.price variable from consideration in our model.

```
cor(retailSales$Unit.price, retailSales$Unit.sale.price, use = "pairwise.complete.obs")
```

```
## [1] 0.999275
```

Part 2.2 Unit.price and Unit.cost

The correlation between these variables has a value of 0.988687. We conclude that adding the two variables to our model will bring little more information than adding a single one. We choose to drop the Unit.cost from consideration in our model.

```
cor(retailSales$Unit.price, retailSales$Unit.cost, use = "pairwise.complete.obs")
```

```
## [1] 0.988687
```

Part 2.3 Gross.profit and Unit.price * Quantity

The correlation value between these terms is 0.9765178. We similarly conclude that incorporating all three variables in out model will add little more information to our model than the two more relevant. Because conceptually Gross.profit is a function of Quantity and Unit.price, we choose to drop Gross.profit from consideration in our model.

```
cor(retailSales$Gross.profit, retailSales$Unit.price * retailSales$Quantity,
    use = "pairwise.complete.obs")
```

```
## [1] 0.9765178
```

Part 2.4 Product, Product.line and Product.type

An analysis of the data confirms that no product belongs to more than one product line or product type as expected. Therefore, we choose to omit the Product.line and Product.type from our prediction model, since they bring no more information than the identification of a product.

Part 2.5 Product.cost and Quantity * Unit.cost

The correlation between these two terms is 0.9998837. Under that observation, we conclude that the Product.cost variable would bring little information in the model beyond that which we would obtain after adding variables Quantity and Product.cost (or Product.Unit.price as an alternative variable that's highly correlated to it) Part 2.5 Population model ——————————— With the above analysis completed, the variables left for consideration when establishing a prediction model are:
Product
Order.method.type
Retailer.country
Unit.price
Quantity

Part 2.6 Revenue

Revenue is the dependent variable in the model. Before stating the population model with this variable, we take a look at a histogram and statistics about this variable.

An analysis of the Revenue variable shows that 59929 out of 84672 values of that variable are NAs. These will be ommited from the model.

A histogram of the Revenue variable indicates a large variance of the revenue numbers across products as is often the case for monetary figures. The result of the very large range is a very positively skewed distribution of revenues with most of the values in the smaller numbers and a long tail of products with very large revenue numbers. In order to adjust the distribution of revenue, we choose to model the log of Revenue.

```
# Remove entries with NA values from Revenue
retailSales.complete <- retailSales[!is.na(retailSales$Revenue), ]

summary(retailSales.complete$Revenue)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##        0    18580    59870   189400   190200 10050000
```

```
stat.desc(retailSales.complete$Revenue)
```

```
##      nbr.val     nbr.null       nbr.na          min          max
## 2.474300e+04 7.600000e+01 0.000000e+00 0.000000e+00 1.005429e+07
##        range          sum       median         mean      SE.mean
## 1.005429e+07 4.686776e+09 5.986727e+04 1.894183e+05 2.484127e+03
## CI.mean.0.95          var      std.dev     coef.var
## 4.869038e+03 1.526863e+11 3.907509e+05 2.062900e+00
```

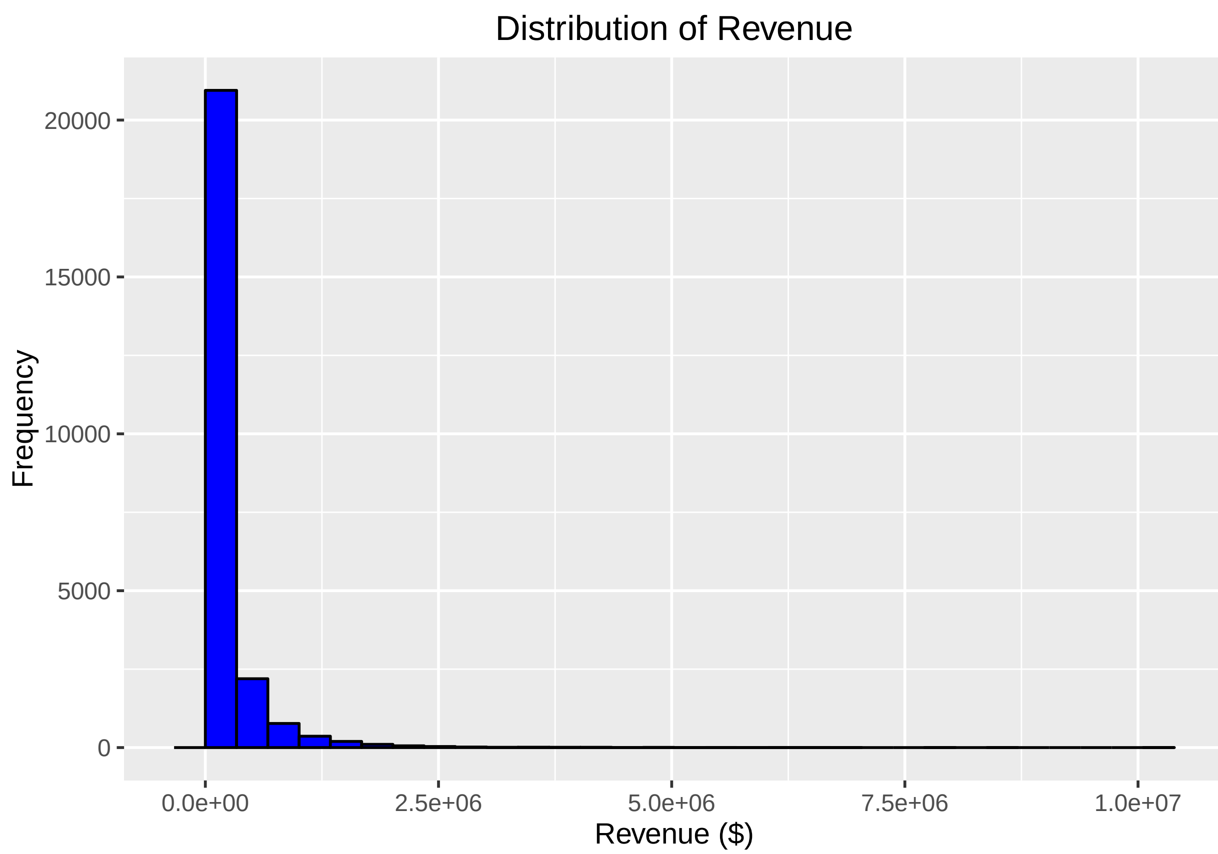
```
print(quantile(retailSales.complete$Revenue, probs = c(0.01, 0.05, 0.1,
    0.25, 0.5, 0.75, 0.9, 0.95, 0.99, 1)))
```

```
##           1%           5%          10%          25%          50%
## 7.789278e+02 3.509748e+03 6.510140e+03 1.857921e+04 5.986727e+04
##          75%          90%          95%          99%         100%
## 1.901930e+05 4.943307e+05 7.924866e+05 1.739111e+06 1.005429e+07
```

```
# Plot the histogram of Revenue
revenue.hist <- ggplot(retailSales.complete, aes(Revenue)) + theme(legend.position = "none") +
    geom_histogram(fill = "Blue", colour = "Black") + labs(title = "Distribution of Revenue",
    x = "Revenue ($)", y = "Frequency")

plot(revenue.hist)
```

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



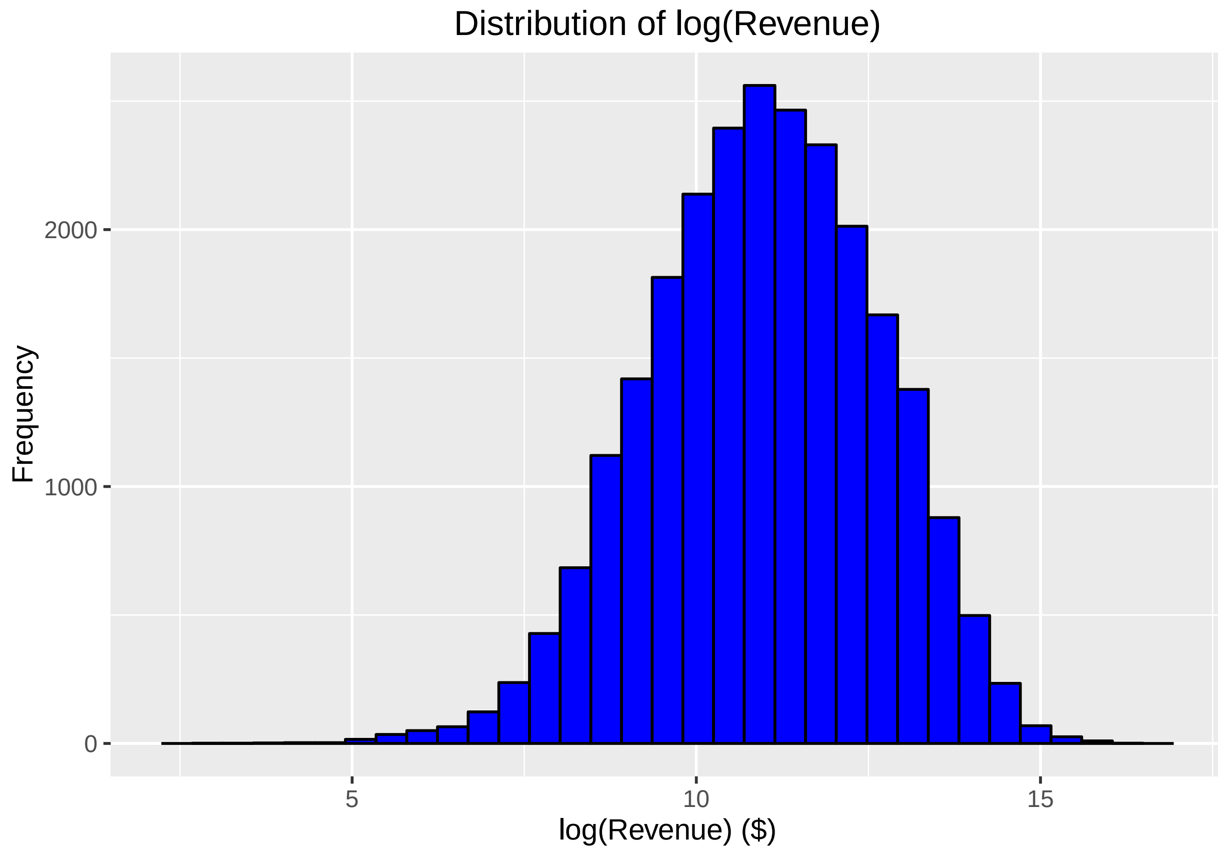

```
retailSales.complete$logRevenue <- log(retailSales.complete$Revenue)

# Plot the histogram of log(Revenue)
log.revenue.hist <- ggplot(retailSales.complete, aes(logRevenue)) + theme(legend.position = "none") +
    geom_histogram(fill = "Blue", colour = "Black") + labs(title = "Distribution of log(Revenue)",
    x = "log(Revenue) ($)", y = "Frequency")

plot(log.revenue.hist)
```

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

```
## Warning: Removed 76 rows containing non-finite values (stat_bin).
```



Part 2.7 Population model

We fit our population model using those variables to predict the revenue of the year 2005 revenue based on the value of those variables in 2004. Therefore our population model becomes:

$$logRevenue.2005 = \beta_0 + \beta 1 * Product + \beta_2 * Order.method.type + \beta 3 * Retailer.country + \beta_4 * Revenue.2004 + \beta_5 * Quantity + u$$

```
# Reshape the data, bringing all 4 years of observations for each
# product into a single record, in wide format.
wideRetailSales <- reshape(retailSales.complete, timevar = "Year", idvar = c("Product",
    "Product.line", "Product.type", "Order.method.type", "Retailer.country"),
    direction = "wide")
```

Part 3: Estimating the population model

```
projected.revenue.model <- lm(Revenue.2005 ~ Product + Retailer.country +
    Order.method.type + Revenue.2004 + Quantity.2004, wideRetailSales)
summary(projected.revenue.model)
```

```
##
## Call:
## lm(formula = Revenue.2005 ~ Product + Retailer.country + Order.method.type +
```

```
##     Revenue.2004 + Quantity.2004, data = wideRetailSales)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1579317   -83605       77    75141  2339374
##
## Coefficients:
##                                          Estimate Std. Error t value
## (Intercept)                            -1.578e+05  3.420e+04  -4.613
## ProductBear Edge                        9.957e+02  4.236e+04   0.024
## ProductBear Survival Edge              -3.943e+03  4.265e+04  -0.092
## ProductBella                            4.425e+04  5.045e+04   0.877
## ProductBlue Steel Max Putter            3.056e+04  4.434e+04   0.689
## ProductBlue Steel Putter                8.765e+03  4.358e+04   0.201
## ProductBugShield Extreme               -8.582e+04  4.327e+04  -1.983
## ProductBugShield Lotion                -2.226e+04  4.196e+04  -0.531
## ProductBugShield Lotion Lite           -5.627e+03  4.387e+04  -0.128
## ProductBugShield Natural               -3.639e+04  4.303e+04  -0.846
## ProductBugShield Spray                 -2.118e+04  4.274e+04  -0.496
## ProductCalamine Relief                 -1.447e+03  4.160e+04  -0.035
## ProductCanyon Mule Carryall             6.192e+04  4.267e+04   1.451
## ProductCanyon Mule Climber Backpack     5.916e+04  4.143e+04   1.428
## ProductCanyon Mule Cooler               3.890e+04  4.237e+04   0.918
## ProductCanyon Mule Extreme Backpack     8.529e+04  4.134e+04   2.063
## ProductCanyon Mule Journey Backpack     1.697e+05  4.278e+04   3.967
## ProductCanyon Mule Weekender Backpack   1.549e+05  4.300e+04   3.603
## ProductCapri                           -8.179e+04  4.862e+04  -1.682
## ProductCat Eye                         -1.348e+04  5.023e+04  -0.268
## ProductCompact Relief Kit              -1.375e+04  4.094e+04  -0.336
## ProductCourse Pro Gloves                1.095e+04  4.354e+04   0.251
## ProductCourse Pro Golf and Tee Set      1.794e+04  4.293e+04   0.418
## ProductCourse Pro Golf Bag              3.658e+04  4.213e+04   0.868
## ProductCourse Pro Putter                2.812e+04  4.438e+04   0.634
## ProductCourse Pro Umbrella              1.666e+04  4.294e+04   0.388
## ProductDante                            5.797e+04  4.946e+04   1.172
## ProductDeluxe Family Relief Kit        -6.938e+04  4.497e+04  -1.543
## ProductDouble Edge                     -1.916e+03  4.186e+04  -0.046
## ProductEdge Extreme                    -1.969e+04  4.218e+04  -0.467
## ProductEverGlow Butane                  1.744e+04  4.184e+04   0.417
## ProductEverGlow Double                  7.051e+03  4.184e+04   0.169
## ProductEverGlow Kerosene               -1.232e+03  4.421e+04  -0.028
## ProductEverGlow Lamp                    1.298e+04  4.238e+04   0.306
## ProductEverGlow Single                  6.362e+03  4.137e+04   0.154
## ProductFairway                         -9.088e+04  5.147e+04  -1.766
## ProductFirefly 2                        1.337e+04  4.209e+04   0.318
## ProductFirefly 4                        1.036e+04  4.160e+04   0.249
## ProductFirefly Extreme                  1.145e+04  4.138e+04   0.277
## ProductFirefly Lite                     7.226e+03  4.138e+04   0.175
## ProductFirefly Mapreader                2.986e+03  4.209e+04   0.071
## ProductFirefly Multi-light             -4.454e+03  4.385e+04  -0.102
## ProductFlicker Lantern                  6.460e+03  4.184e+04   0.154
## ProductGlacier Basic                   -1.524e+04  4.118e+04  -0.370
## ProductGlacier Deluxe                   8.836e+02  4.211e+04   0.021
## ProductGlacier GPS                      8.541e+03  4.191e+04   0.204
```

```
## ProductGlacier GPS Extreme              -1.803e+04  4.151e+04  -0.434
## ProductHailstorm Steel Irons             3.004e+04  4.348e+04   0.691
## ProductHailstorm Steel Woods Set         1.158e+05  4.349e+04   2.663
## ProductHailstorm Titanium Irons          6.538e+04  4.410e+04   1.482
## ProductHailstorm Titanium Woods Set      1.872e+05  4.573e+04   4.094
## ProductHawk Eye                          6.772e+04  4.909e+04   1.380
## ProductHibernator                        7.502e+04  4.206e+04   1.784
## ProductHibernator Camp Cot               6.074e+04  4.140e+04   1.467
## ProductHibernator Extreme                1.209e+05  4.167e+04   2.902
## ProductHibernator Lite                   8.471e+04  4.170e+04   2.031
## ProductHibernator Pad                    2.375e+04  4.320e+04   0.550
## ProductHibernator Pillow                 2.305e+02  4.321e+04   0.005
## ProductHibernator Self - Inflating Mat   4.989e+04  4.063e+04   1.228
## ProductInferno                           2.116e+05  4.898e+04   4.321
## ProductInfinity                          6.474e+05  4.992e+04  12.969
## ProductInsect Bite Relief               -2.043e+03  4.136e+04  -0.049
## ProductKodiak                           -6.685e+04  4.883e+04  -1.369
## ProductLady Hailstorm Steel Irons        2.725e+04  4.468e+04   0.610
## ProductLady Hailstorm Steel Woods Set    7.773e+04  4.400e+04   1.767
## ProductLady Hailstorm Titanium Irons     4.707e+04  4.528e+04   1.040
## ProductLady Hailstorm Titanium Woods Set  1.125e+05  4.492e+04   2.504
## ProductLegend                           -5.945e+04  5.049e+04  -1.177
## ProductLux                               8.803e+04  4.915e+04   1.791
## ProductMax Gizmo                        -3.627e+04  5.045e+04  -0.719
## ProductMaximus                           2.347e+05  4.897e+04   4.792
## ProductMountain Man Analog               5.893e+04  4.185e+04   1.408
## ProductMountain Man Combination          3.417e+03  4.137e+04   0.083
## ProductMountain Man Deluxe              -9.502e+03  4.211e+04  -0.226
## ProductMountain Man Digital              1.018e+04  4.074e+04   0.250
## ProductMountain Man Extreme             -8.796e+03  4.165e+04  -0.211
## ProductOpera Vision                     -6.940e+04  5.045e+04  -1.376
## ProductPocket Gizmo                     -3.443e+04  4.994e+04  -0.689
## ProductPolar Extreme                    -1.091e+04  4.263e+04  -0.256
## ProductPolar Ice                         2.473e+04  4.210e+04   0.587
## ProductPolar Sports                     -4.702e+03  4.354e+04  -0.108
## ProductPolar Sun                         1.423e+05  4.093e+04   3.477
## ProductPolar Wave                       -6.037e+01  4.184e+04  -0.001
## ProductRanger Vision                     5.928e+04  4.861e+04   1.219
## ProductSam                              -2.601e+05  4.921e+04  -5.286
## ProductSeeker 35                         1.040e+04  4.188e+04   0.248
## ProductSeeker 50                         5.581e+03  4.138e+04   0.135
## ProductSeeker Extreme                    4.532e+03  4.187e+04   0.108
## ProductSeeker Mini                      -2.410e+03  4.210e+04  -0.057
## ProductSingle Edge                      -3.101e+04  4.154e+04  -0.747
## ProductStar Dome                         1.329e+05  4.132e+04   3.216
## ProductStar Gazer 2                      1.933e+05  4.348e+04   4.445
## ProductStar Gazer 3                      1.246e+05  4.171e+04   2.988
## ProductStar Gazer 6                      5.703e+04  4.242e+04   1.344
## ProductStar Lite                         2.723e+05  4.239e+04   6.423
## ProductStar Peg                          1.316e+02  4.089e+04   0.003
## ProductSun Blocker                      -1.178e+04  4.187e+04  -0.281
## ProductSun Shelter 15                   -6.227e+03  4.129e+04  -0.151
## ProductSun Shelter 30                   -2.194e+04  4.218e+04  -0.520
## ProductSun Shelter Stick                -1.781e+04  4.336e+04  -0.411
```

```
## ProductSun Shield                      -1.635e+04  4.131e+04  -0.396
## ProductTrail Master                    -2.210e+05  4.987e+04  -4.432
## ProductTrail Scout                     -7.920e+04  4.983e+04  -1.589
## ProductTrail Star                      -3.682e+05  5.071e+04  -7.263
## ProductTrailChef Canteen                5.949e+03  4.162e+04   0.143
## ProductTrailChef Cook Set               5.109e+04  4.143e+04   1.233
## ProductTrailChef Cup                   -6.333e+03  4.194e+04  -0.151
## ProductTrailChef Deluxe Cook Set        5.581e+04  4.175e+04   1.337
## ProductTrailChef Double Flame           5.193e+04  4.121e+04   1.260
## ProductTrailChef Kettle                 2.586e+04  4.284e+04   0.604
## ProductTrailChef Kitchen Kit            1.658e+04  4.236e+04   0.392
## ProductTrailChef Single Flame           5.592e+04  4.144e+04   1.349
## ProductTrailChef Utensils               4.686e+03  4.138e+04   0.113
## ProductTrailChef Water Bag              5.510e+03  4.149e+04   0.133
## ProductTrendi                           1.159e+05  4.884e+04   2.372
## ProductTX                               1.180e+05  5.026e+04   2.348
## ProductVenue                           -9.873e+03  4.945e+04  -0.200
## ProductZone                             5.187e+05  5.367e+04   9.665
## Retailer.countryBelgium                 2.513e+04  1.942e+04   1.294
## Retailer.countryBrazil                 -5.084e+04  2.192e+04  -2.319
## Retailer.countryCanada                  6.488e+04  1.788e+04   3.628
## Retailer.countryChina                  -5.227e+04  2.003e+04  -2.609
## Retailer.countryDenmark                -3.087e+04  2.104e+04  -1.467
## Retailer.countryFinland                 3.182e+04  1.974e+04   1.612
## Retailer.countryFrance                  3.430e+04  1.636e+04   2.097
## Retailer.countryGermany                 3.056e+04  1.726e+04   1.770
## Retailer.countryItaly                   4.932e+04  1.790e+04   2.755
## Retailer.countryJapan                   5.779e+04  1.651e+04   3.499
## Retailer.countryKorea                  -2.253e+04  1.958e+04  -1.151
## Retailer.countryMexico                  5.201e+04  1.896e+04   2.744
## Retailer.countryNetherlands             1.766e+04  1.738e+04   1.016
## Retailer.countrySingapore               5.554e+04  1.844e+04   3.012
## Retailer.countrySpain                   5.291e+02  1.932e+04   0.027
## Retailer.countrySweden                  1.929e+04  2.076e+04   0.929
## Retailer.countryUnited Kingdom         -3.361e+02  1.820e+04  -0.018
## Retailer.countryUnited States           5.007e+04  1.663e+04   3.012
## Order.method.typeFax                    9.661e+04  1.848e+04   5.228
## Order.method.typeMail                   8.684e+04  1.781e+04   4.876
## Order.method.typeSales visit            5.490e+04  1.337e+04   4.106
## Order.method.typeSpecial                6.369e+04  2.628e+04   2.423
## Order.method.typeTelephone              1.543e+04  1.365e+04   1.131
## Order.method.typeWeb                    2.479e+05  1.154e+04  21.481
## Revenue.2004                            9.296e-01  1.528e-02  60.829
## Quantity.2004                           1.839e+00  5.802e-01   3.171
##                                        Pr(>|t|)
## (Intercept)                            4.09e-06 ***
## ProductBear Edge                       0.981249
## ProductBear Survival Edge              0.926351
## ProductBella                           0.380479
## ProductBlue Steel Max Putter           0.490680
## ProductBlue Steel Putter               0.840599
## ProductBugShield Extreme               0.047385 *
## ProductBugShield Lotion                0.595722
## ProductBugShield Lotion Lite           0.897947
```

```
## ProductBugShield Natural                 0.397790
## ProductBugShield Spray                   0.620171
## ProductCalamine Relief                   0.972253
## ProductCanyon Mule Carryall              0.146787
## ProductCanyon Mule Climber Backpack      0.153319
## ProductCanyon Mule Cooler                0.358664
## ProductCanyon Mule Extreme Backpack      0.039178 *
## ProductCanyon Mule Journey Backpack      7.38e-05 ***
## ProductCanyon Mule Weekender Backpack    0.000319 ***
## ProductCapri                             0.092619 .
## ProductCat Eye                           0.788401
## ProductCompact Relief Kit                0.737086
## ProductCourse Pro Gloves                 0.801496
## ProductCourse Pro Golf and Tee Set       0.675979
## ProductCourse Pro Golf Bag               0.385265
## ProductCourse Pro Putter                 0.526409
## ProductCourse Pro Umbrella               0.698033
## ProductDante                             0.241192
## ProductDeluxe Family Relief Kit          0.122913
## ProductDouble Edge                       0.963493
## ProductEdge Extreme                      0.640678
## ProductEverGlow Butane                   0.676779
## ProductEverGlow Double                   0.866190
## ProductEverGlow Kerosene                 0.977776
## ProductEverGlow Lamp                     0.759419
## ProductEverGlow Single                   0.877796
## ProductFairway                           0.077502 .
## ProductFirefly 2                         0.750763
## ProductFirefly 4                         0.803394
## ProductFirefly Extreme                   0.782040
## ProductFirefly Lite                      0.861384
## ProductFirefly Mapreader                 0.943454
## ProductFirefly Multi-light               0.919101
## ProductFlicker Lantern                   0.877283
## ProductGlacier Basic                     0.711395
## ProductGlacier Deluxe                    0.983260
## ProductGlacier GPS                       0.838532
## ProductGlacier GPS Extreme               0.664154
## ProductHailstorm Steel Irons             0.489618
## ProductHailstorm Steel Woods Set         0.007767 **
## ProductHailstorm Titanium Irons          0.138300
## ProductHailstorm Titanium Woods Set      4.32e-05 ***
## ProductHawk Eye                          0.167788
## ProductHibernator                        0.074546 .
## ProductHibernator Camp Cot               0.142468
## ProductHibernator Extreme                0.003724 **
## ProductHibernator Lite                   0.042284 *
## ProductHibernator Pad                    0.582582
## ProductHibernator Pillow                 0.995743
## ProductHibernator Self - Inflating Mat   0.219525
## ProductInferno                           1.59e-05 ***
## ProductInfinity                           < 2e-16 ***
## ProductInsect Bite Relief                0.960608
## ProductKodiak                            0.171120
```

```
## ProductLady Hailstorm Steel Irons        0.542016
## ProductLady Hailstorm Steel Woods Set    0.077336 .
## ProductLady Hailstorm Titanium Irons     0.298584
## ProductLady Hailstorm Titanium Woods Set 0.012305 *
## ProductLegend                            0.239076
## ProductLux                               0.073314 .
## ProductMax Gizmo                         0.472175
## ProductMaximus                           1.71e-06 ***
## ProductMountain Man Analog               0.159131
## ProductMountain Man Combination          0.934185
## ProductMountain Man Deluxe               0.821485
## ProductMountain Man Digital              0.802636
## ProductMountain Man Extreme              0.832744
## ProductOpera Vision                      0.168999
## ProductPocket Gizmo                      0.490598
## ProductPolar Extreme                     0.798103
## ProductPolar Ice                         0.556937
## ProductPolar Sports                      0.914007
## ProductPolar Sun                         0.000512 ***
## ProductPolar Wave                        0.998849
## ProductRanger Vision                     0.222735
## ProductSam                               1.32e-07 ***
## ProductSeeker 35                         0.803928
## ProductSeeker 50                         0.892735
## ProductSeeker Extreme                    0.913819
## ProductSeeker Mini                       0.954354
## ProductSingle Edge                       0.455401
## ProductStar Dome                         0.001311 **
## ProductStar Gazer 2                      9.01e-06 ***
## ProductStar Gazer 3                      0.002822 **
## ProductStar Gazer 6                      0.178924
## ProductStar Lite                         1.48e-10 ***
## ProductStar Peg                          0.997433
## ProductSun Blocker                       0.778464
## ProductSun Shelter 15                    0.880123
## ProductSun Shelter 30                    0.602956
## ProductSun Shelter Stick                 0.681278
## ProductSun Shield                        0.692275
## ProductTrail Master                      9.58e-06 ***
## ProductTrail Scout                       0.112029
## ProductTrail Star                        4.49e-13 ***
## ProductTrailChef Canteen                 0.886343
## ProductTrailChef Cook Set                0.217557
## ProductTrailChef Cup                     0.879985
## ProductTrailChef Deluxe Cook Set         0.181332
## ProductTrailChef Double Flame            0.207658
## ProductTrailChef Kettle                  0.546038
## ProductTrailChef Kitchen Kit             0.695416
## ProductTrailChef Single Flame            0.177249
## ProductTrailChef Utensils                0.909847
## ProductTrailChef Water Bag               0.894349
## ProductTrendi                            0.017738 *
## ProductTX                                0.018905 *
## ProductVenue                             0.841769
```

```
## ProductZone                                     < 2e-16 ***
## Retailer.countryBelgium                        0.195685
## Retailer.countryBrazil                         0.020452 *
## Retailer.countryCanada                         0.000289 ***
## Retailer.countryChina                          0.009103 **
## Retailer.countryDenmark                        0.142421
## Retailer.countryFinland                        0.107082
## Retailer.countryFrance                         0.036096 *
## Retailer.countryGermany                        0.076724 .
## Retailer.countryItaly                          0.005891 **
## Retailer.countryJapan                          0.000471 ***
## Retailer.countryKorea                          0.249924
## Retailer.countryMexico                         0.006099 **
## Retailer.countryNetherlands                    0.309511
## Retailer.countrySingapore                      0.002609 **
## Retailer.countrySpain                          0.978154
## Retailer.countrySweden                         0.352912
## Retailer.countryUnited Kingdom                 0.985271
## Retailer.countryUnited States                  0.002612 **
## Order.method.typeFax                           1.79e-07 ***
## Order.method.typeMail                          1.12e-06 ***
## Order.method.typeSales visit                   4.10e-05 ***
## Order.method.typeSpecial                       0.015421 *
## Order.method.typeTelephone                     0.258116
## Order.method.typeWeb                            < 2e-16 ***
## Revenue.2004                                    < 2e-16 ***
## Quantity.2004                                  0.001532 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 190400 on 4280 degrees of freedom
##   (6475 observations deleted due to missingness)
## Multiple R-squared:  0.7921, Adjusted R-squared:  0.7852
## F-statistic: 114.1 on 143 and 4280 DF,  p-value: < 2.2e-16
```

Part 4: Is the change in the average renevue different from 95 cents when the planned revenue increases by 1

Am I supposed to model average revenue vs planned revenue? Some other relationship?

Part 5: Explain the interaction terms in your model.

I haven't defined any and wouldnt't know which to define.

Part 6: Resons why the OLS coefficients may be biased and/or not consistent

The coefficients may be biased because of omitted variable bias. There are many variables that affect revenue that are in the error term since they don't appear in our model. All of those variables, if correlated any with the independent variables of our mode have the potential to create a bias of the coefficient of the OLS regression.

Part 7: IV Plan to improve forecastign model

I need to think of one. Not sure that I have the right variables in ths current model to begin with.