Exploratory Data Analysis Using R

Name: Oyeleke Folashade

Candidate No.: 399411

Module: Software Analytics

1. Display descriptive statistics on the dataset.

```
#Loading necessary packages
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✓ tibble  3.1.6     ✓ purrr   0.3.4
## ✓ tidyr   1.2.0     ✓ stringr 1.4.0
## ✓ readr   2.1.2     ✓ forcats 0.5.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(moments)
#Reading the file, wage.csv, into the wage dataframe
wage = read.csv('~/Downloads/wage.csv')

#displaying the first five rows of the wage dataframe
head(wage, 5)
```

	married <dbl>	hourly_wage <dbl>	years_in_education <dbl>	years_in_employment <dbl> ▸
1	1	3.24	12	2
2	0	3.00	11	0

3	1	6.00	8	28
4	1	5.30	12	2
5	1	8.75	16	8

5 rows | 1-5 of 8 columns

```
#Displaying the structure of the dataframe
str(wage)
```

```
## 'data.frame':    525 obs. of  7 variables:
##  $ married            : num  1 0 1 1 1 0 0 0 1 0 ...
##  $ hourly_wage        : num  3.24 3 6 5.3 8.75 ...
##  $ years_in_education : num  12 11 8 12 16 18 12 12 17 16 ...
##  $ years_in_employment: num  2 0 28 2 8 7 3 4 21 2 ...
##  $ num_dependents     : num  3 2 0 1 0 0 0 2 0 0 ...
##  $ gender             : chr  "female" "male" "male" "male" ...
##  $ race               : chr  "white" "white" "white" "white" ...
```

The structure of the wage dataframe shows that the dataframe has 7 variables of which ***married***, ***hourly_wage***, ***years_in_education***, ***years_in_employment*** and ***num_dependents*** are numeric variables, while ***gender*** and ***race*** are character variables.

```
#Displaying the descriptive Statistics of the Wage data frame
summary(wage)
```

```
##     married        hourly_wage     years_in_education years_in_employment
##  Min.   :0.0000   Min.   : 0.530   Min.   : 0.00      Min.   : 0.000     
##  1st Qu.:0.0000   1st Qu.: 3.350   1st Qu.:12.00      1st Qu.: 0.000     
##  Median :1.0000   Median : 4.670   Median :12.00      Median : 2.000     
##  Mean   :0.6092   Mean   : 5.918   Mean   :12.56      Mean   : 5.152     
##  3rd Qu.:1.0000   3rd Qu.: 6.880   3rd Qu.:14.00      3rd Qu.: 7.000     
##  Max.   :1.0000   Max.   :24.980   Max.   :18.00      Max.   :44.000     
##  NA's   :3        NA's   :8        NA's   :3          NA's   :6          
##  num_dependents     gender               race          
##  Min.   :0.000   Length:525         Length:525        
##  1st Qu.:0.000   Class :character   Class :character  
##  Median :1.000   Mode  :character   Mode  :character  
##  Mean   :1.044                                        
##  3rd Qu.:2.000                                        
##  Max.   :6.000                                        
##  NA's   :5                                            
```

The summary table displays the **minimum**, **1st** quartile, **median**, **mean**, **3rd** quartile and the **maximum** for all numeric variables of the dataframe.

```r
#creating a function that takes a vector and returns the value with the highest frequency
mode_value = function(x) {
  #Checks for the unique values in the vector
   uniq_val = unique(x)
  #the which.max function returns the index of the unique value
  #the value with the highest frequency is returned
  return( uniq_val[ which.max( tabulate( match(x, uniq_val)))])
}

#Displaying the counts of the categories for the married variable
table(wage$married)
```

```
## 
##   0   1 
## 204 318
```

```r
#Assigning the category of the married variable with the highest frequency to mode_m using the mode_value function
mode_m = mode_value(wage$married)
#Displaying the mode of the married variable
if (mode_m == 1){
  cat('The mode of the married variable is', mode_m,'which represents married. \n')
}else{
cat('The mode of the married variable is', mode_m,'which represents not married. \n')
}
```

```
## The mode of the married variable is 1 which represents married.
```

```r
#Displaying the counts of the categories for the gender variable
table(wage$gender)
```

```
## 
##        female   male 
##      4    249    272
```

```r
#Assigning the category of the gender variable with the highest frequency to mode_g using the mode_value function
mode_g = mode_value(wage$gender)
#Displaying the mode of the gender variable
cat('The mode of the gender variable is', mode_g, '\n')
```

```
## The mode of the gender variable is male
```

```r
#Displaying the counts of the categories for the race variable
table(wage$race)
```

```
## 
##          nonwhite    white 
##       10       54      461
```

```
#Assigning the category of the race variable with the highest frequency to mode_r using the
mode_value function
mode_r = mode_value(wage$race)
#Displaying the mode of the race variable
cat('The mode of the race variable is', mode_r)
```

```
## The mode of the race variable is white
```

I used the table function to create a contingency table that dispalys the counts for each category of the variables. I also created a function that returns the mode. I used the function to get the mode for each variable.

For the first table, ***1*** represents ***married*** and ***0*** represents ***not married***.

2. Check if any records in the data have any missing values; handle the missing data as appropriate.

```
 #Re-coding Empty spaces to missing values for the entire dataframe
#select observations with empty spaces and recode to missing values
wage[wage == ''] = NA

#Using the sapply, sum and is.na function to iterate over the wage dataframe and get the tot
al number of missing values for each variable
sapply(wage, function(x) sum(is.na(x)))
```

```
##             married         hourly_wage  years_in_education years_in_employment
##                   3                   8                   3                   6
##      num_dependents              gender                race
##                   5                   4                  10
```

The table shows the number of missing values for each variable in the wage dataframe.

```
#Replacing the missing values for the categorical variables gender and race using the mode

#Replace NA values in the married column with the category with the highest frequency
wage$married = replace_na(wage$married, mode_m)

#Replace NA values in the gender column with the category with the highest frequency
wage$gender = replace_na(wage$gender, mode_g)

#Replace NA values in the race column with the category with the highest frequency
wage$race = replace_na(wage$race, mode_r)
```

I replaced the missing values of categorical variables with their mode.

```
#Replacing the missing values for the Continuous variables using the mean

#Replace NA values in the hourly_wage column with the mean
wage$hourly_wage = replace_na(wage$hourly_wage, mean(wage$hourly_wage, na.rm = TRUE))

#Replace NA values in the years_in_education column with the mean
wage$years_in_education = replace_na(wage$years_in_education, mean(wage$years_in_education,
na.rm = TRUE))

#Replace NA values in the years_in_employment column with the mean
wage$years_in_employment = replace_na(wage$years_in_employment, mean(wage$years_in_employmen
t, na.rm = TRUE))

#Replace NA values in the num_dependents column with the mean
wage$num_dependents = replace_na(wage$num_dependents, mean(wage$num_dependents, na.rm = TRUE
))

#Using the sapply, sum and is.na function to iterate over the wage dataframe to check the mi
ssing values for each variable
sapply(wage, function(x) sum(is.na(x)))
```

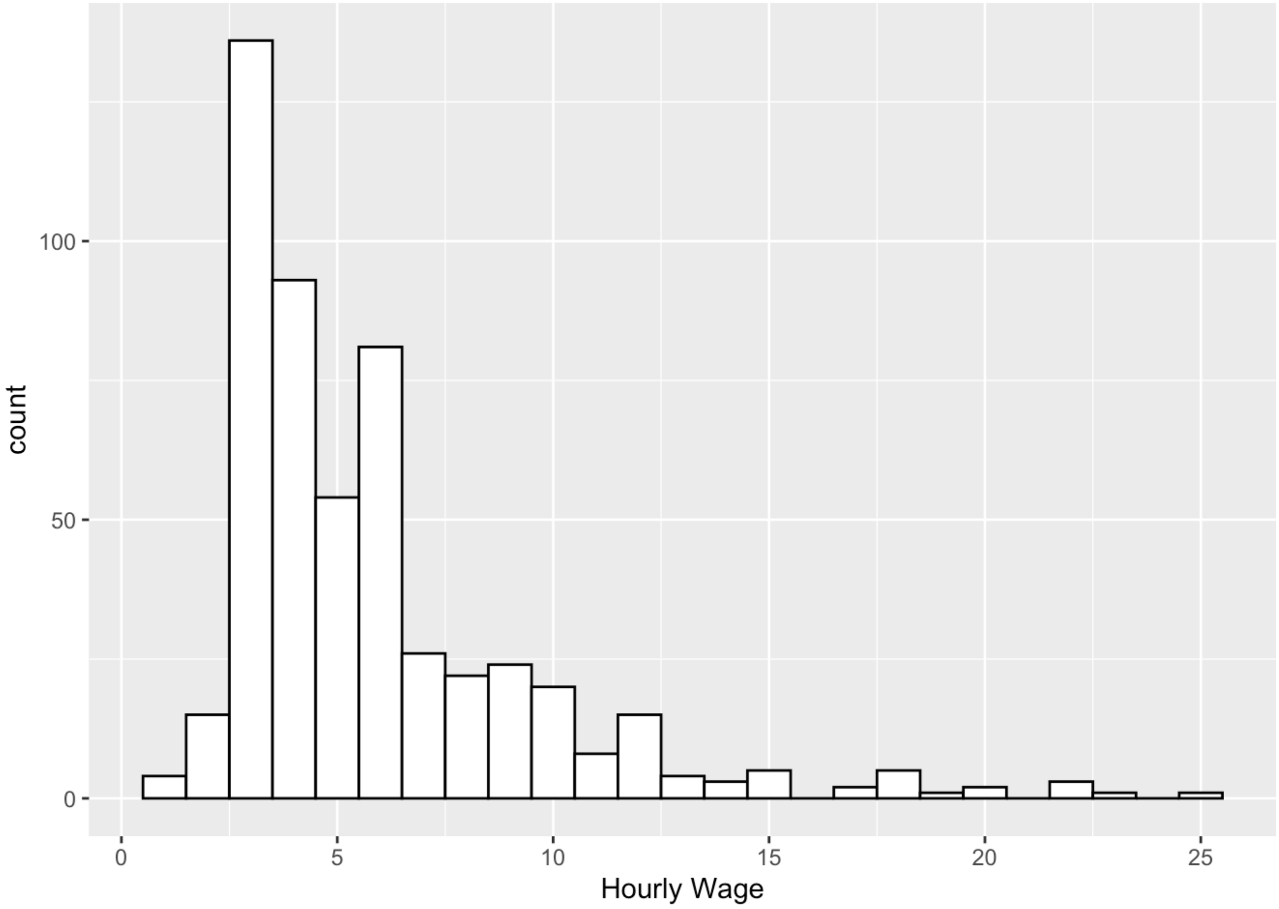
```
##             married         hourly_wage  years_in_education years_in_employment
##                   0                   0                   0                   0
##      num_dependents              gender                race
##                   0                   0                   0
```

I replaced the missing values of continuous variables with their mean.

The table shows that there are no more missing values in the wage dataframe.

3. Build a graph visualizing the distribution of one or more individual continuous variables of the dataset

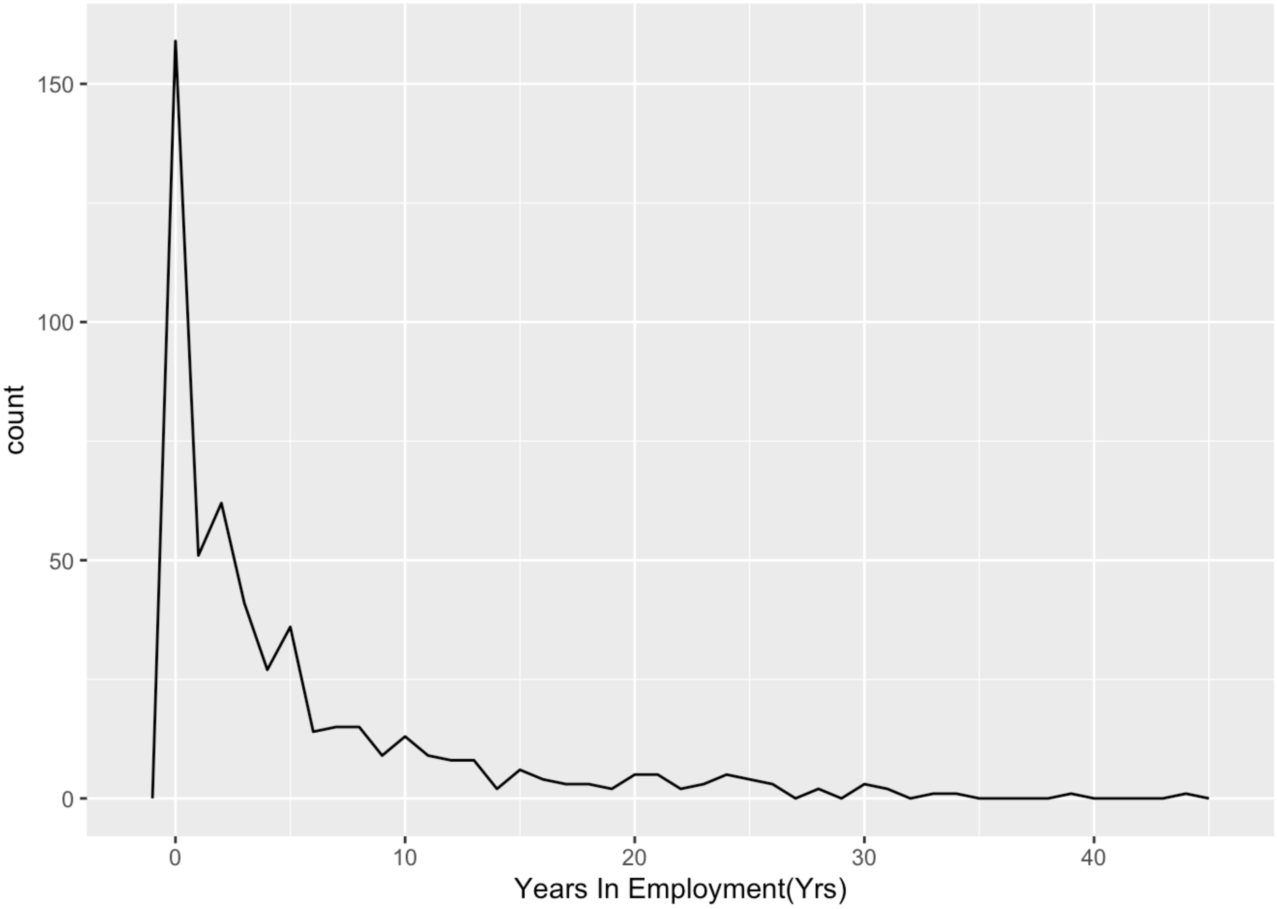
```
#Using ggplot to plot a histogram that shows the distribution of hourly wages
ggplot(wage, aes(x = hourly_wage)) +
  geom_histogram(binwidth = 1, color = "black", fill = 'white') +
  scale_x_continuous(name="Hourly Wage")
```



This **histogram** shows the **Hourly wage** of the population**.**

The histogram is **right-skewed**. Majority of the population’s hourly wage is between 2.5 - 8.

```
#Using ggplot to plot a frequency polygon that shows the distribution of years in employment
ggplot(wage, aes(x = years_in_employment)) +
  geom_freqpoly(binwidth = 1, color = "black") +
  scale_x_continuous(name="Years In Employment(Yrs)")
```

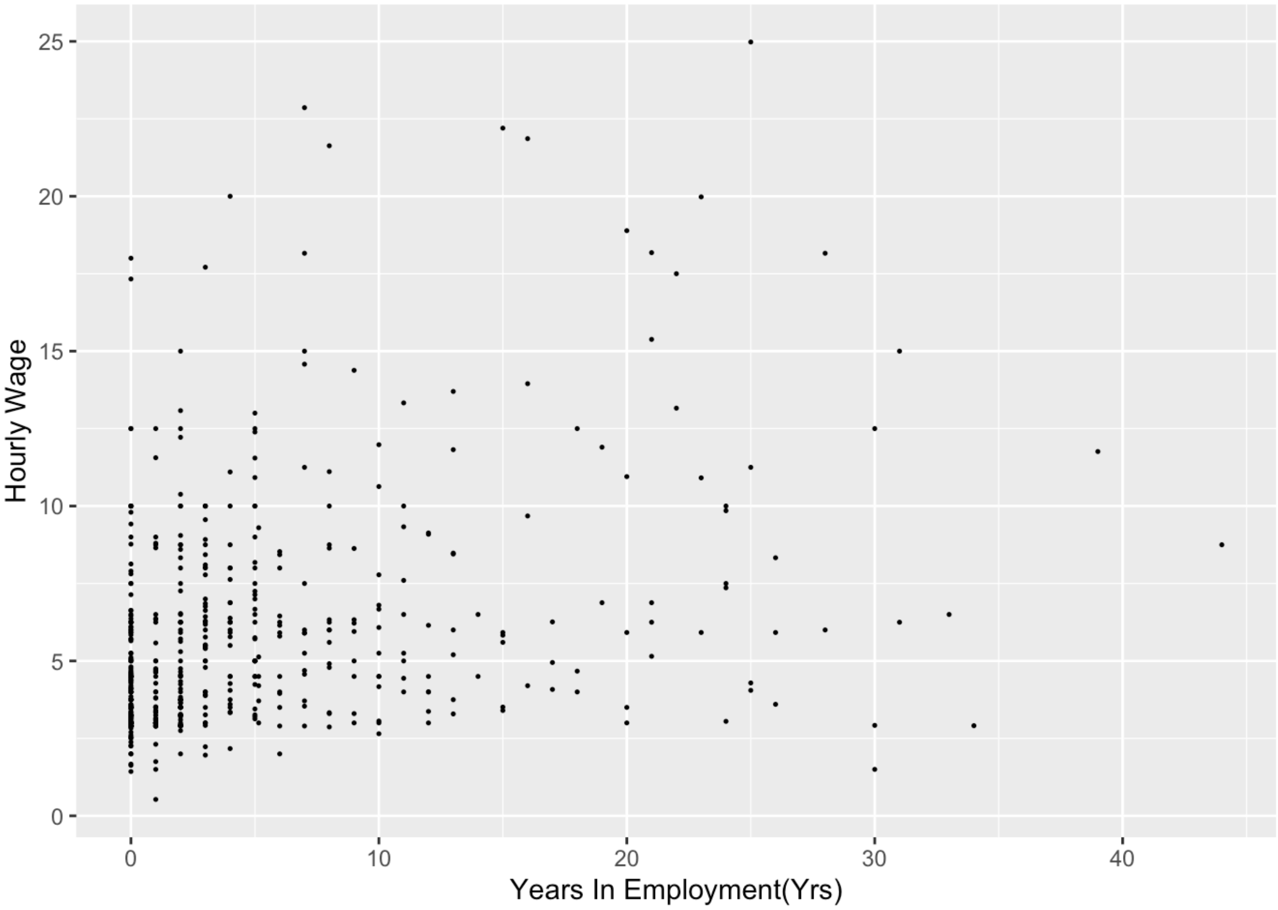


The **frequency polygon** shows the **Years in employment** of the population.

The frequency polygon is **right-skewed.** Majority of the population have only been in employment for less than 5 years.

4. Build a graph visualizing the relationship in a pair of continuous variables. Determine the correlation between them.

```
#using ggplot to plot a scatter plot that shows the relationship between the hourly_wage and the Years_in_employment
ggplot(wage, aes(years_in_employment, hourly_wage)) +
  geom_point(size = 0.25) +
  scale_x_continuous(name="Years In Employment(Yrs)") +
  scale_y_continuous(name="Hourly Wage")
```

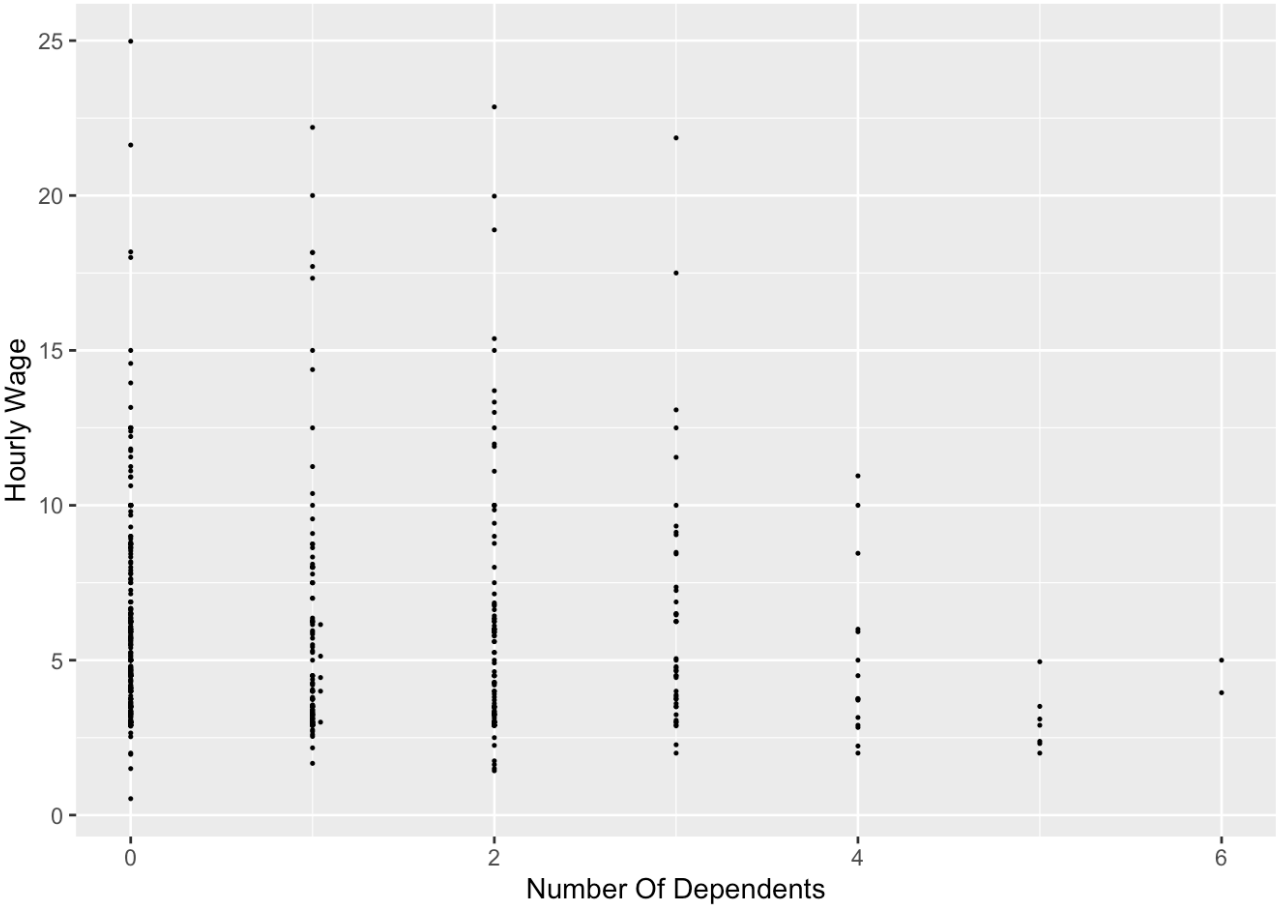


```
#The scatter plot of Hourly wage and Years in employment shows that there is a weak positive
linear relationship between the two variables.
#Calculating the pearson correlation coefficient
cor_1 = cor(wage$hourly_wage, wage$years_in_education, method = 'pearson')
cat('The pearsons coefficient of correlation between hourly wage and years in employment is'
, cor_1)
```

```
## The pearsons coefficient of correlation between hourly wage and years in employment is 0.
3886195
```

The Pearsons coefficient of correlation of ***Hourly wage*** and ***Years in employment*** shows that there is a weak positive linear relationship between the two variables.

```
#using ggplot to plot a scatter plot that shows the relationship between the Years_in_educat
ion and the Years_in_employment
ggplot(wage, aes(num_dependents, hourly_wage)) +
  geom_point(size = 0.25) +
  scale_x_continuous(name="Number Of Dependents") +
  scale_y_continuous(name="Hourly Wage")
```



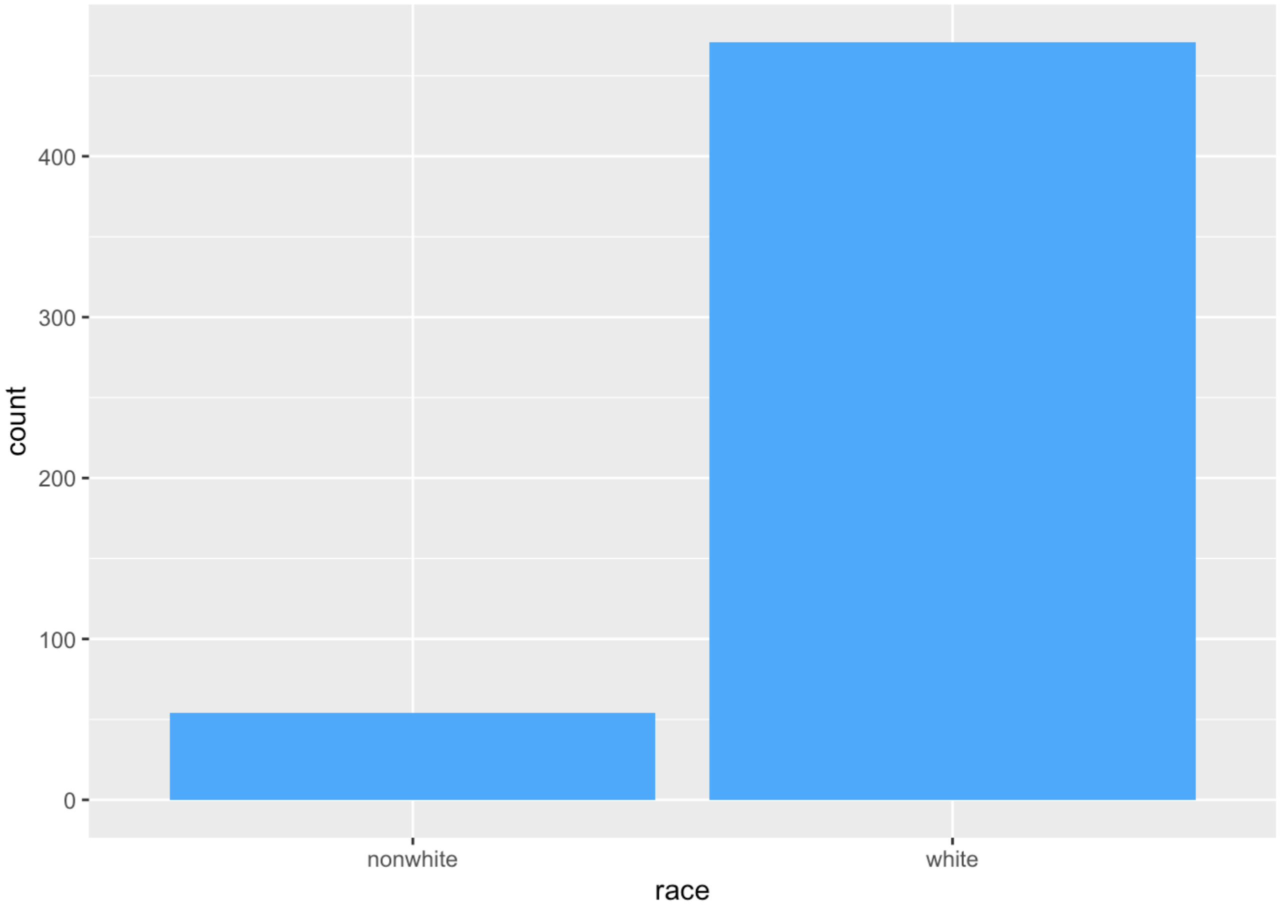
```
#The scatter plot of Number of Dependents and Hourly Wage shows that there is no linear rela
tionship between the two variables.
#Calculating the pearson correlation coefficient
cor_2 = cor(wage$num_dependents, wage$hourly_wage, method = 'pearson')
cat('The pearsons coefficient of correlation between number of dependents and hourly wage is
', cor_2)
```

```
## The pearsons coefficient of correlation between number of dependents and hourly wage is -
0.04730779
```

The Pearsons coefficient of correlation of ***Number of Dependents*** and ***Hourly Wage*** shows that there is no linear relationship between the two variables.

5. Display unique values of a categorical variable.

```
#Using a bar plot to visualize the number of each categories in the Race variable
#using ggplot to plot the barchart
ggplot(wage, aes(race)) +
  geom_bar(fill = '#51ADFC')
```



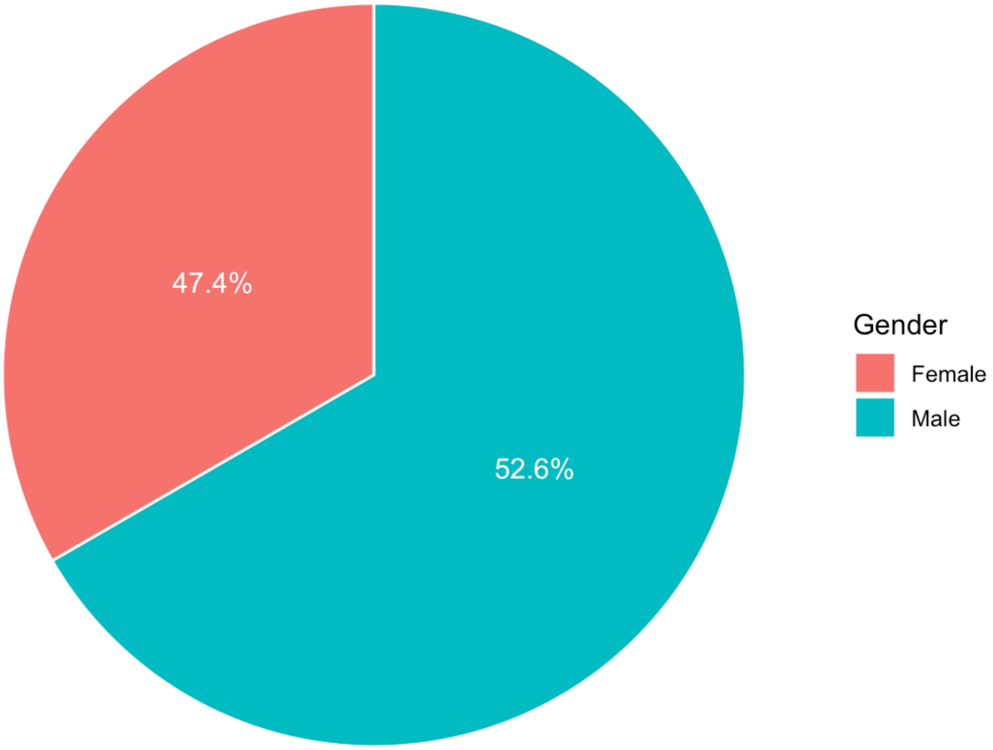
The barplot shows that majority of the population is white.

```
#Using a pie chart to visualize the proportion of each categories in the gender variable

#creating a dataframe that has the gender categories and its respective proportions
gender_prop = data.frame(prop.table(table(wage$gender)))

#creating a dataframe that has the gender categories and percentages
gender_per = data.frame(prop = scales::percent(gender_prop[,2]), Gender = gender_prop[,1])

#Using ggplot to plot a pie chart showing the percentage of the population that is male or f
emale
ggplot(gender_per, aes(x = "", y = prop, fill = Gender)) +
  geom_col(color = "white") +
  geom_text(aes(label = prop),
            position = position_stack(vjust = 0.5), color = 'white') +
  coord_polar(theta = "y") +
  scale_fill_discrete(labels = c("Female", "Male")) +
  theme_void()
```



The pie chart shows that 52.6% of the population is male and 47.4% is female.

6. Build a contingency table of two potentially related categorical variables. Conduct a statistical test of the independence between them.

```
#building a contingency table of the race and married variables
cont_table1 = table(wage$race, wage$married)
cont_table1
```

```
## 
##               0   1
##   nonwhite   26  28
##   white     178 293
```

```
#Using the chi square test to test for independence of variables
#the null hypothesis of the test is that there is no relationship between the race and marri
ed variables
#the alternative hypothesis is that there is a relationship between the race and married var
iables
chi1 = chisq.test(cont_table1)
if(chi1$p.value < 0.05){
  cat('The p_value of the test,', chi1$p.value, 'is less than the significance level of 5%')
}else{
cat('The p_value of the test,', chi1$p.value, 'is greater than the significance level of 5%'
)}
```

```
## The p_value of the test, 0.183038 is greater than the significance level of 5%
```

The chi square test shows us that the p_value is greater than the 5% significance level, Therefore we accept the null hypothesis that there is no relationship between the race and married variables.

```
#building a contingency table of the gender and married variables
cont_table2 = table(wage$gender, wage$married)
cont_table2
```

```
##
##            0   1
##   female 118 131
##   male    86 190
```

```
#Using the chi square test to test for independence of variables
#the null hypothesis of the test is that there is no relationship between the gender and mar
ried variables
#the alternative hypothesis is that there is a relationship between the gender and married v
ariables
chi2 = chisq.test(cont_table2)
if(chi2$p.value < 0.05){
  cat('The p_value of the test,', chi2$p.value, 'is less than the significance level of 5%')
}else{
cat('The p_value of the test,', chi2$p.value, 'is greater than the significance level of 5%'
)}
```

```
## The p_value of the test, 0.0001992053 is less than the significance level of 5%
```

The chi square test shows us that the p_value is less than the 5% significance level, Therefore we reject the null hypothesis. This means there is a relationship between the gender and married variables.

7. Retrieve one or more subset of rows based on two or more criteria and present descriptive statistics on the subset(s).

```
#Sub-setting the wage dataframe by the married females
mar_fem = subset(wage, married == 1 & gender == 'female')

#displaying the descriptive statistics of the numerical variables
summary(mar_fem)
```

```
##     married   hourly_wage     years_in_education years_in_employment
##  Min.   :1   Min.   : 1.430   Min.   : 0.00      Min.   : 0.000     
##  1st Qu.:1   1st Qu.: 3.250   1st Qu.:12.00      1st Qu.: 0.000     
##  Median :1   Median : 4.000   Median :12.00      Median : 3.000     
##  Mean   :1   Mean   : 4.606   Mean   :12.46      Mean   : 4.553     
##  3rd Qu.:1   3rd Qu.: 5.815   3rd Qu.:14.00      3rd Qu.: 5.576     
##  Max.   :1   Max.   :15.000   Max.   :18.00      Max.   :34.000     
##  num_dependents    gender              race          
##  Min.   :0.00   Length:131         Length:131        
##  1st Qu.:0.00   Class :character   Class :character  
##  Median :1.00   Mode  :character   Mode  :character  
##  Mean   :1.13                                        
##  3rd Qu.:2.00                                        
##  Max.   :5.00                                        
```

```
#for the categorical variables,I used the mode_value function to get the mode.
#mode of gender variable in the mar_fem dataframe
mode_g1 = mode_value(mar_fem$gender)
cat('The mode of the gender variable is', mode_g1, '\n')
```

```
## The mode of the gender variable is female
```

```
#mode of race variable in the mar_fem dataframe
mode_r1 = mode_value(mar_fem$race)
cat('The mode of the race variable is', mode_r1, '\n')
```

```
## The mode of the race variable is white
```

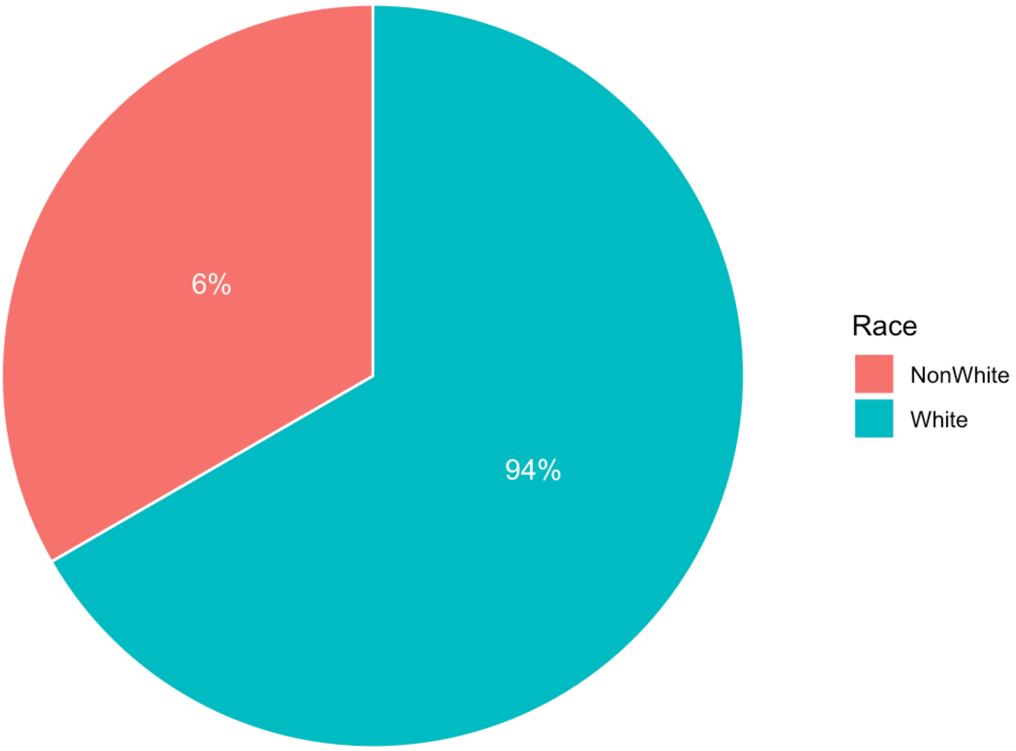
```
#mode of married variable in the mar_fem dataframe
mode_m1 = mode_value(mar_fem$married)
if (mode_m1 == 1){
  cat('The mode of the married variable is', mode_m1,'which represents married')
}else{
cat('The mode of the married variable is', mode_m1,'which represents not married')
}
```

```
## The mode of the married variable is 1 which represents married
```

```
#Using a pie chart to visualize the proportion of each categories in the race variable for the mar_fem subset

#creating a dataframe that has the race categories and its respective proportions
race_prop = data.frame(prop.table(table(mar_fem$race)))
#creating a dataframe that has the race categories and percentages
race_per = data.frame(prop = scales::percent(race_prop[,2]), Race = race_prop[,1])

#Using ggplot to plot a pie chart showing the percentage of the population that is male and female
ggplot(race_per, aes(x = "", y = prop, fill = Race)) +
  geom_col(color = "white") +
  geom_text(aes(label = prop),
            position = position_stack(vjust = 0.5), color = 'white') +
  coord_polar(theta = "y") +
  scale_fill_discrete(labels = c("NonWhite", "White")) +
  theme_void()
```



The pie chart shows that of the married women in the population, 94% are white and 6% are nonwhite.

```
#Subsetting the wage dataframe by the nonwhite single males with no dependents
nonw_male = subset(wage, gender == 'male' & race == 'nonwhite' & married == 0 & num_dependen
ts == 0)

#displaying the descriptive statistics of the numerical variables
summary(nonw_male)
```

```
##     married   hourly_wage     years_in_education years_in_employment
##  Min.   :0   Min.   : 2.920   Min.   : 3.00      Min.   : 0.000     
##  1st Qu.:0   1st Qu.: 3.000   1st Qu.: 8.50      1st Qu.: 0.000     
##  Median :0   Median : 4.000   Median :12.00      Median : 0.000     
##  Mean   :0   Mean   : 5.274   Mean   :10.71      Mean   : 7.143     
##  3rd Qu.:0   3rd Qu.: 7.000   3rd Qu.:13.50      3rd Qu.:10.000     
##  Max.   :0   Max.   :10.000   Max.   :16.00      Max.   :30.000     
##  num_dependents    gender              race          
##  Min.   :0      Length:7           Length:7          
##  1st Qu.:0      Class :character   Class :character  
##  Median :0      Mode  :character   Mode  :character  
##  Mean   :0                                           
##  3rd Qu.:0                                           
##  Max.   :0                                           
```

```
#for the categorical variables,I used the mode_value function to get the mode.
#mode of gender variable in the nonw_male dataframe
mode_g2 = mode_value(nonw_male$gender)
cat('The mode of the gender variable is', mode_g2, '\n')
```

```
## The mode of the gender variable is male
```

```
#mode of race variable in the nonw_male dataframe
mode_r2 = mode_value(nonw_male$race)
cat('The mode of the race variable is', mode_r2, '\n')
```

```
## The mode of the race variable is nonwhite
```

```
#mode of married variable in the nonw_male dataframe
mode_m2 = mode_value(nonw_male$married)
if (mode_m2 == 1){
  cat('The mode of the married variable is', mode_m2,'which represents married')
}else{
cat('The mode of the married variable is', mode_m2,'which represents not married')
}
```

```
## The mode of the married variable is 0 which represents not married
```

The summary table shows the **minimum**, **1st** quartile, **median**, **mean**, **3rd** quartile and the **maximum** for all numeric variables in the dataframe. I used the mode_value function to get the mode of the continuous variables.

8. Conduct a statistical test of the significance of the difference between the means of two subsets of the data.

```
#conducting a statistical test for the hourly wage of the married male and married female su
bset
#Creating the subset for married male
subset_m = subset(wage, married == 1 & gender == 'male')
#Creating the subset for married female
subset_f = subset(wage, married == 1 & gender == 'female')

#using the two sample t-test to test if the means of the hourly wage of the two subsets are
different
#the null hypothesis is that the means are equal, i.e., the difference in the means is equal
to zero.
ttest = t.test(subset_m$hourly_wage, subset_f$hourly_wage)
ttest
```

```
## 
##  Welch Two Sample t-test
## 
## data:  subset_m$hourly_wage and subset_f$hourly_wage
## t = 9.2527, df = 284.26, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  2.649892 4.081973
## sample estimates:
## mean of x mean of y 
##  7.972251  4.606319
```

```
if(ttest$p.value < 0.05){
  cat('The p_value of the test,', ttest$p.value, 'is less than the significance level of 5%'
)
}else{
cat('The p_value of the test,', ttest$p.value, 'is greater than the significance level of 5%
')}
```

```
## The p_value of the test, 5.456763e-18 is less than the significance level of 5%
```

Since, the p_value of the test is less than the significance level of 5%, we reject the null hypothesis that the means are equal, i.e., the difference in the means is equal to zero. Hence, there is a significant difference between the means of the married male and married female population.

From the output of the hypothesis test, it can be seen that the mean of hourly wage for the married male (7.97) is significantly higher than that of the married female (4.6) .

```
#conducting a statistical test for the hourly wage of new workers ( year of employment = 0 )
and existing workers ( year of employment not = 0 )

#Creating the subset for new workers
subset_n = subset(wage, years_in_employment == 0)
#Creating the subset for existing workers
subset_e = subset(wage, years_in_employment != 0)

#using the two sample t-test to test if the means of the hourly wage of the two subsets are
different
#the null hypothesis is that the means are equal, i.e., the difference in the means is equal
to zero.
ttest2 = t.test(subset_n$hourly_wage, subset_e$hourly_wage)
ttest2
```

```
## 
##  Welch Two Sample t-test
## 
## data:  subset_n$hourly_wage and subset_e$hourly_wage
## t = -6.663, df = 450.9, p-value = 7.839e-11
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.474742 -1.347413
## sample estimates:
## mean of x mean of y 
##  4.585443  6.496520
```

```
if(ttest2$p.value < 0.05){
  cat('The p_value of the test,', ttest2$p.value, 'is less than the significance level of 5%
')
}else{
cat('The p_value of the test,', ttest2$p.value, 'is greater than the significance level of 5
%')}
```

```
## The p_value of the test, 7.838758e-11 is less than the significance level of 5%
```

Since, the p_value of the test is less than the significance level of 5%, we reject the null hypothesis that the means are equal. Hence, there is a significant difference between the means of hourly wage of new workers and existing workers in the population.

From the output of the hypothesis test, it can be seen that the mean of the hourly wage for new workers (4.58) is significantly lower than that of the existing workers (6.49).

9. Create one or more tables that group the data by a certain categorical variable and displays summarized information for each group (e.g. the mean or sum within the group).

```
#creating a table that shows the mean of hourly_wage, mean of years_in_employment and mean of years_in_education for each gender category
wage_gen_mean = summarise(group_by(wage, gender),  Mean_hw = mean(hourly_wage), Mean_Yem = mean(years_in_employment), Mean_Yed = mean(years_in_education))
wage_gen_mean
```

gender	Mean_hw	Mean_Yem	Mean_Yed
<chr>	<dbl>	<dbl>	<dbl>
female	4.638419	3.725948	12.32129
male	7.071904	6.438957	12.77055

2 rows

This table shows the means of the hourly_wage, years_in_employment and years_in_education for each gender category.

```
#creating a table that shows the mean of hourly_wage, mean of years_in_employment and mean of years_in_education for each race category
wage_race_mean = summarise(group_by(wage, race),  Mean_hw = mean(hourly_wage), Mean_Yem = mean(years_in_employment), Mean_Yed = mean(years_in_education))
wage_race_mean
```

race	Mean_hw	Mean_Yem	Mean_Yed
<chr>	<dbl>	<dbl>	<dbl>
nonwhite	5.645615	5.447263	11.87037
white	5.948936	5.118389	12.63625

2 rows

This table shows the means of the hourly_wage, years_in_employment and years_in_education for each race category.

```
#creating a table that shows the mean of hourly_wage, mean of years_in_employment and mean of years_in_education for each married category
wage_marr_mean = aggregate(wage[,c("hourly_wage", "years_in_education", "years_in_employment")], list(wage$married), mean)
#Renaming the first row from 0,1 to notmarried and married
wage_marr_mean[,1] = c('Not Married', 'Married')
#Renaming the columns
colnames(wage_marr_mean) = c("Married", "Mean of Hourly Wage", "Mean of Years in Education", "Mean of Years in Employment")
wage_marr_mean
```

Married	Mean of Hourly Wage	Mean of Years in Education ▸
<chr>	<dbl>	<dbl>
Not Married	4.846355	12.33333
Married	6.598615	12.69991

2 rows | 1-3 of 4 columns

The table above shows the means of the hourly_wage, years_in_employment and years_in_education for each married category.

10. Implement a linear regression model and interpret its output.

```
#Since all categories in the wage dataframe contain only two categories, there is no need to
created dummy variables as R does this already
#using the lm function to create a model that estimates the hourly wage
#For this model the dependent variable, y, is the hourly wage, while the other variables are
the independent variables
model = lm(hourly_wage ~ married + years_in_education + years_in_employment + num_dependents
+ gender + race, data = wage)
summary(model)
```

```
##
## Call:
## lm(formula = hourly_wage ~ married + years_in_education + years_in_employment +
##     num_dependents + gender + race, data = wage)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6122 -1.7614 -0.5708  1.0549 15.0633
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -2.76796    0.75951  -3.644 0.000295 ***
## married              0.70305    0.28278   2.486 0.013225 *
## years_in_education   0.52137    0.04919  10.598  < 2e-16 ***
## years_in_employment  0.15311    0.01892   8.092 4.20e-15 ***
## num_dependents       0.09566    0.10849   0.882 0.378342
## gendermale           1.67318    0.26814   6.240 9.11e-10 ***
## racewhite           -0.06650    0.43138  -0.154 0.877551
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.977 on 518 degrees of freedom
## Multiple R-squared:   0.35,  Adjusted R-squared:  0.3424
## F-statistic: 46.48 on 6 and 518 DF,  p-value: < 2.2e-16
```

```
#The race and the number of dependents variables are not significant because their p_value i
s greater than the level of significance 5%.

#I removed the independent variables that were insignificant to the model to try and improve
it.
model = lm(hourly_wage ~ married + years_in_education + years_in_employment + gender, data =
wage)
summary(model)
```

```
##
## Call:
## lm(formula = hourly_wage ~ married + years_in_education + years_in_employment +
##     gender, data = wage)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5179 -1.7540 -0.5703  0.9878 14.9642
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -2.61066    0.63069  -4.139 4.06e-05 ***
## married              0.74471    0.27708   2.688  0.00743 **
## years_in_education   0.51059    0.04768  10.709  < 2e-16 ***
## years_in_employment  0.15197    0.01884   8.064 5.11e-15 ***
## gendermale           1.67094    0.26775   6.241 9.04e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.973 on 520 degrees of freedom
## Multiple R-squared:  0.3489, Adjusted R-squared:  0.3439
## F-statistic: 69.67 on 4 and 520 DF,  p-value: < 2.2e-16
```

1. **Coefficients on the variables.** The estimated coefficients are specified in the second summary table.The equation for the model is:

$$Hourlywage = -2.61066 + 0.74471 * married + 0.51059 * yearsineducation + 0.15197 * yearsinemployment$$

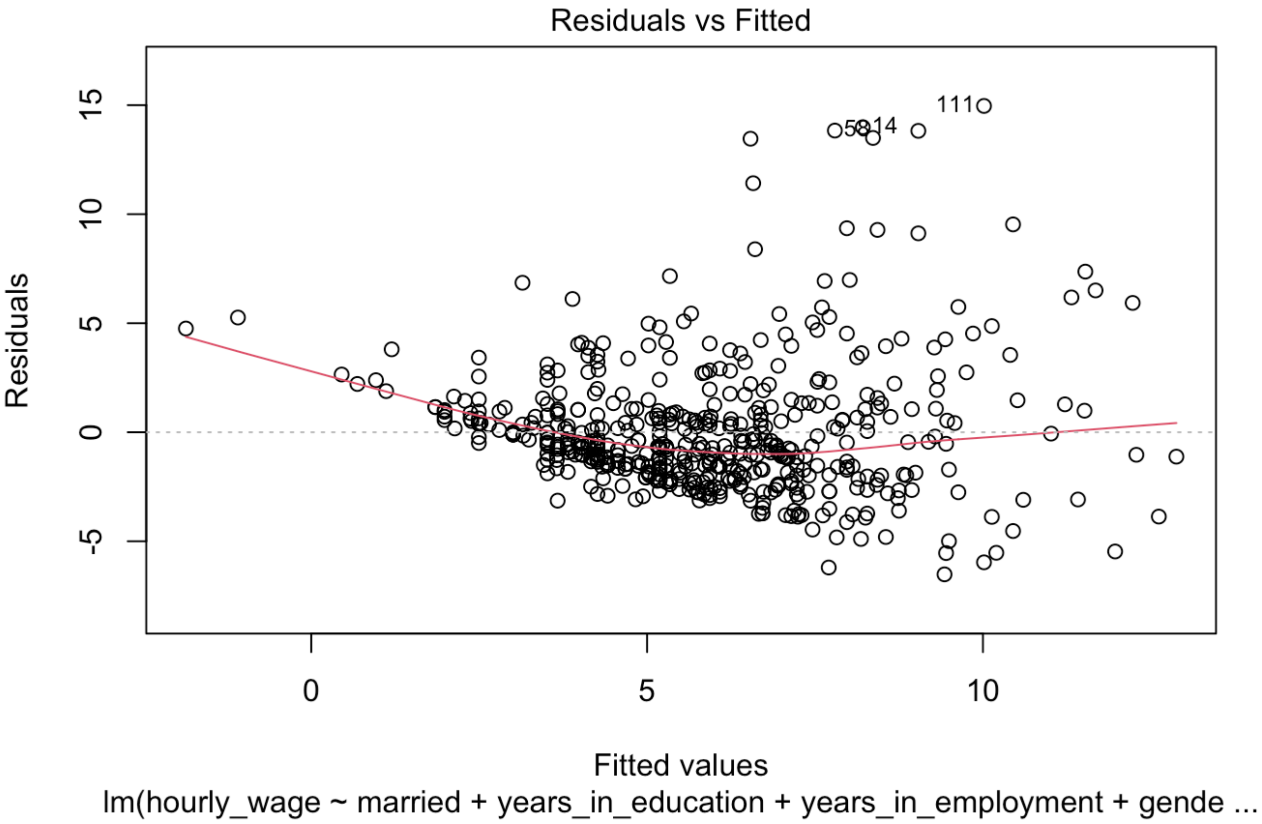
The coefficients of the married, years_in_education, years_in_employment and gender variables shows that they have a positive effect on the the hourly wage. For the gender variable if the value is 1,i.e., if the person is male, it increases the hourly wage. If the person is female, the hourly wage doesn't change.

2. **Significance of the variables.** From the summary, we can see that the married, years_in_education, years_in_employment and gender variables are significant because the p_value is less than the level of significance 5%.

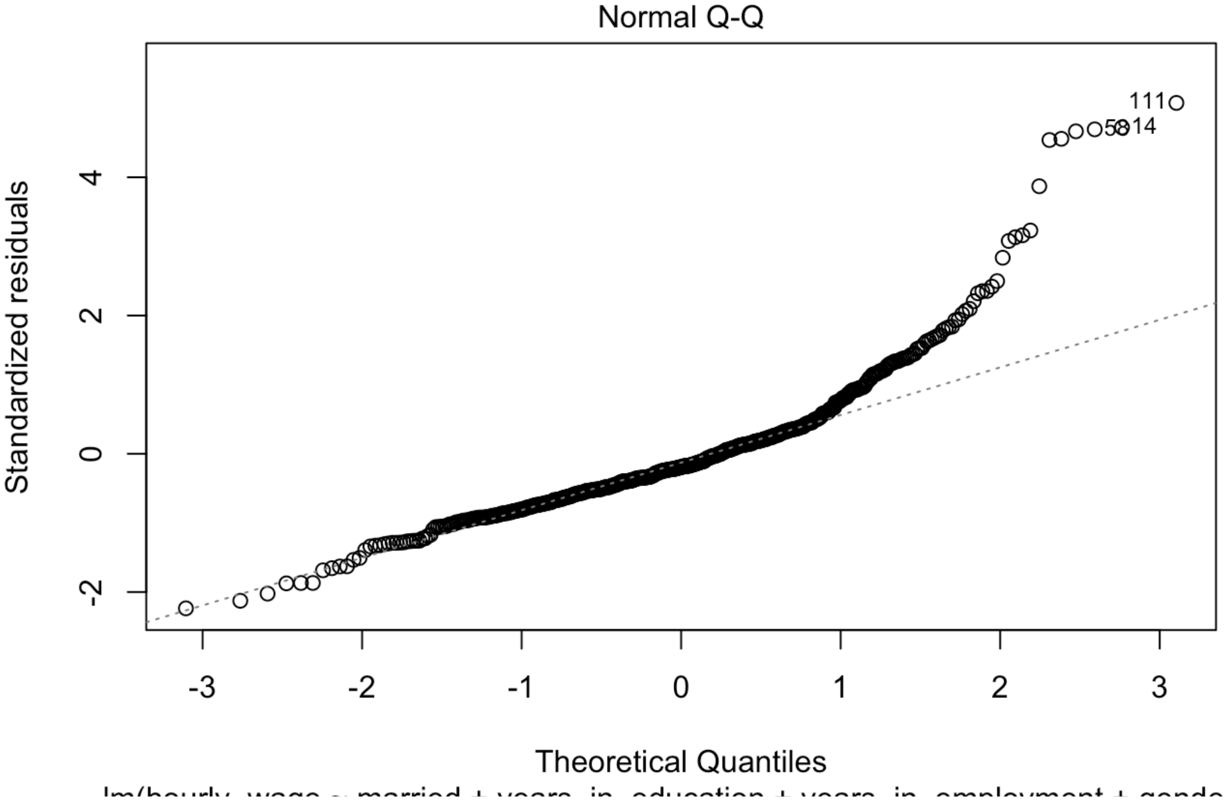
3. **Quality of the model.** It can be seen that the removal of the insignificant variables improved the Adjusted R^2 by a little bit, but the model is still not accurate.

####Checking The Classical Linear Regression Method Assupmtions

```
#Linearity of the data
#This is checked by inspecting the plot of the residuals vs fitted values
plot(model, 1)
```



```
#Normality of residuals
#It can be checked by using the Q-Q plot of residuals
plot(model, 2)
```



lm(hourly_wage ~ married + years_in_education + years_in_employment + gende ...

The residual plot and the Q-Q plot show that the residuals are not equally distributed around 0 and are not normally distributed.

```
#Using the Jarque Bera test to confirm normality of residuals
jarque.test(model$residuals)
```

```
##
##  Jarque-Bera Normality Test
##
## data:  model$residuals
## JB = 883.57, p-value < 2.2e-16
## alternative hypothesis: greater
```

The Jarque-Bera test shows that the residuals are not normally distributed since the p value is lower than the 5% significance level.

Based on the violation of the assumptions of the classical linear regression method, we can conclude that this model is not reliable. The addition of other independent variables affecting the hourly wage that the model didn’t take into account might improve the model.