R Task Rabbit

First, read the data from .csv file.

```
rabbit_data <- read.table("sample.csv", header = TRUE, sep = ",")
head(rabbit_data)
```

```
##                      recommendation_id          created_at  tasker_id
## 1 0-0-70cf97d7-37af-4834-901c-ce3ad4893b8c 2017-09-01 00:32:25 1009185352
## 2 0-0-70cf97d7-37af-4834-901c-ce3ad4893b8c 2017-09-01 00:32:25 1006892359
## 3 0-0-70cf97d7-37af-4834-901c-ce3ad4893b8c 2017-09-01 00:32:25 1012023956
## 4 0-0-70cf97d7-37af-4834-901c-ce3ad4893b8c 2017-09-01 00:32:25 1009733517
## 5 0-0-70cf97d7-37af-4834-901c-ce3ad4893b8c 2017-09-01 00:32:25 1013579273
## 6 0-0-70cf97d7-37af-4834-901c-ce3ad4893b8c 2017-09-01 00:32:25 1012043028
##   position hourly_rate num_completed_tasks hired          category
## 1        1          38                 151     0 Furniture Assembly
## 2        2          40                 193     0 Furniture Assembly
## 3        3          28                   0     0 Furniture Assembly
## 4        4          43                 303     0 Furniture Assembly
## 5        5          29                  39     0 Furniture Assembly
## 6        6          28                   2     0 Furniture Assembly
```

Following are questions and answers about the data in the CSV sample file:

1. How many recommendation sets are in this data sample?

```
length(unique(rabbit_data$recommendation_id))
```

```
## [1] 2100
```

Answer : Total no of recommendation sets are 2100 in this data set.

2. Each recommendation set shows from 1 to 15 Taskers, what is:

- average number of Taskers shown

```
rabbit_data$tasker_id <- as.factor(rabbit_data$tasker_id)
total_no_shown <- as.numeric(length(rabbit_data$recommendation_id))
tolal_tasker <- as.numeric(length(unique(rabbit_data$tasker_id)))
avg_no_tasker <- total_no_shown/tolal_tasker
avg_no_tasker
```

```
## [1] 36.14458
```

```
  Answer : Average number of Taskers shown 36.14458
```

- median number of Taskers shown

```
tasker_data <- table(rabbit_data$tasker_id)
nrow(tasker_data)
```

```
## [1] 830
```

```
sort(tasker_data)[416]
```

```
## 1009820249
##         16
```

```
   Answer: - Median  number of Taskers shown is 16
```

```
count the frequency of the tasker appears as per recommendation set. Arrange the frequency and as th
```

3. How many total unique Taskers are there in this data sample?

```
unique_tsker <- length(unique(rabbit_data$tasker_id))
unique_tsker
```

```
## [1] 830
```

Answer: Total unique 830 Taskers are there in this data sample

4. Which Tasker has been shown the most?

```
tail(names(sort(table(rabbit_data$tasker_id))), 1)
```

```
## [1] "1014508755"
```

```
length(rabbit_data$tasker_id[rabbit_data$tasker_id == "1014508755"])
```

```
## [1] 608
```

```
#sort(table(rabbit_data$tasker_id[rabbit_data$tasker_id == "1014508755"]))
```

Answer: Tasker with “1014508755” appear most with count 608.

Which Tasker has been shown the least?

```
head(names(sort(table(rabbit_data$tasker_id))), 1)
```

```
## [1] "1006690425"
```

```
length(rabbit_data$tasker_id[rabbit_data$tasker_id == "1006690425"])
```

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

```
#sort(table(rabbit_data$tasker_id[rabbit_data$tasker_id == "1006690425"]))
```

```
Answer: Tasker with "1006690425" appear most with count 1.
```

5. Which Tasker has been hired the most?

```
most_hired <- tail(names(sort(table(rabbit_data$tasker_id[rabbit_data$hired == 1]))), 1)
most_hired
```

```
## [1] "1012043028"
```

```
length(rabbit_data$tasker_id[rabbit_data$hired == 1 & rabbit_data$tasker_id == most_hired])
```

```
## [1] 59
```

```
Answer: 1012043028 Tasker has been hired the most with count 59
```

Which Tasker has been hired the least?

```
least_hired <- head(names(sort(table(rabbit_data$tasker_id[rabbit_data$hired == 1]))), 1)
least_hired
```

```
## [1] "1006646767"
```

```
length(rabbit_data$tasker_id[rabbit_data$hired == 1 & rabbit_data$tasker_id == least_hired])
```

```
## [1] 0
```

Answer: 1006720321 Tasker has been hired the least with count 1

6. If we define the “Tasker conversion rate” as the number of times a Tasker has been hired, out of the number of times the Tasker has been shown, how many Taskers have a conversion rate of 100%

```
get_lenght_hired <- function(id)
{
  get_length_hired <- length(rabbit_data$tasker_id[rabbit_data$hired == 1 & rabbit_data$tasker_id == id]
  if(get_length_hired)
  {
    return(get_length_hired)
  }
  else
  {
    return(0)
  }
}

get_lenght_shown<- function(id)
{
  get_lenght_shown <- length(rabbit_data$recommendation_id[rabbit_data$tasker_id == id])
  if(get_lenght_shown)
  {
    return(get_lenght_shown)
  }
  else
  {
    return(0)
  }
}
```

```
unique_id <- unique(rabbit_data$tasker_id)
tasker_id <-NULL
appear_cnt <-NULL
hired_cnt <- NULL

for (i in 1:length(unique_id))
{
  tasker_id[i] <- unique_id[i]
  appear_cnt[i] <- get_lenght_shown(unique_id[i])
  hired_cnt[i] <- get_lenght_hired(unique_id[i])
}

task_conversion <- data.frame(id = tasker_id,AppearCount = appear_cnt ,HiredCount = hired_cnt)
length(task_conversion$id[task_conversion$AppearCount == task_conversion$HiredCount])
```

```
## [1] 6
```

```
task_conversion$id[task_conversion$AppearCount == task_conversion$HiredCount]
```

```
## [1] 487  50 176 554 111 776
```

```
Answer: There are 6 Taskers have a conversion rate of 100%
```

7. Would it be possible for all Taskers to have a conversion rate of 100% Please explain your reasoning.

Answer: Not possible practically, Only possible if tasker showed once and get hired. and takser only get one chance to get hired and show.

8. For each category, what is the average position of the Tasker who is hired?

```
table(rabbit_data$category[rabbit_data$hired == 1])
```

```
##
## Furniture Assembly           Mounting        Moving Help
##                572                562                571
```

Answer: The above table shows, each category, with an average position of the Tasker who is hired.

9. For each category, what is the average hourly rate and average number of completed tasks for the Taskers who are hired?

```
aggregate(cbind(num_completed_tasks,hourly_rate) ~ category, rabbit_data, mean)
```

```
##             category num_completed_tasks hourly_rate
## 1 Furniture Assembly            185.8338     39.4197
## 2           Mounting            220.2708     50.4749
## 3        Moving Help            257.6025     82.5530
```

```
    Answer: The above table shows, average hourly rate and average number of completed tasks for the Tas
```

10. Based on the previous, how would you approach the question of:

    How can we use market data to suggest hourly rates to Taskers that would maximize their opportunity to be hired?

    Please describe in detail, with code and formulas that support your model.

Answer:
To get suggestions about hourly rates to taskers form the market data. We need to consider only the data set who have hired for the different roles. Secondly, the data is not in the categorical format so we will use the logistic regression model to predict the hourly rates form given data. In this linear regression model hourly rate is dependent variable and need to figure out independent the variable to predict the value.

Step 1: Prepare the data for the logistic regression model. Step 1.1: Subset the data with hired tasker. Using subset function. And get the structure of the data using the str function.

```
data_hired <-  subset(rabbit_data, rabbit_data$hired == 1)
data_hired$position <- as.factor(data_hired$position)

str(data_hired)
```

```
## 'data.frame':    1705 obs. of  8 variables:
##  $ recommendation_id  : Factor w/ 2100 levels "0-0-00033225-3f89-47dd-b4f1-5d1feb359a76",..: 1322 195
##  $ created_at         : Factor w/ 2093 levels "2017-09-01 00:32:25",..: 1 2 3 4 7 8 9 10 11 12 ...
##  $ tasker_id          : Factor w/ 830 levels "1006646767","1006648538",..: 363 363 487 231 222 222 24
##  $ position           : Factor w/ 15 levels "1","2","3","4",..: 13 10 3 4 11 9 2 2 2 2 ...
##  $ hourly_rate        : int  50 50 32 95 35 35 42 42 42 34 ...
##  $ num_completed_tasks: int  914 914 0 1053 59 59 353 353 353 75 ...
##  $ hired              : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ category           : Factor w/ 3 levels "Furniture Assembly",..: 1 1 3 3 1 1 1 1 1 3 ...
```

Step 2: Check Missing value using missmap function

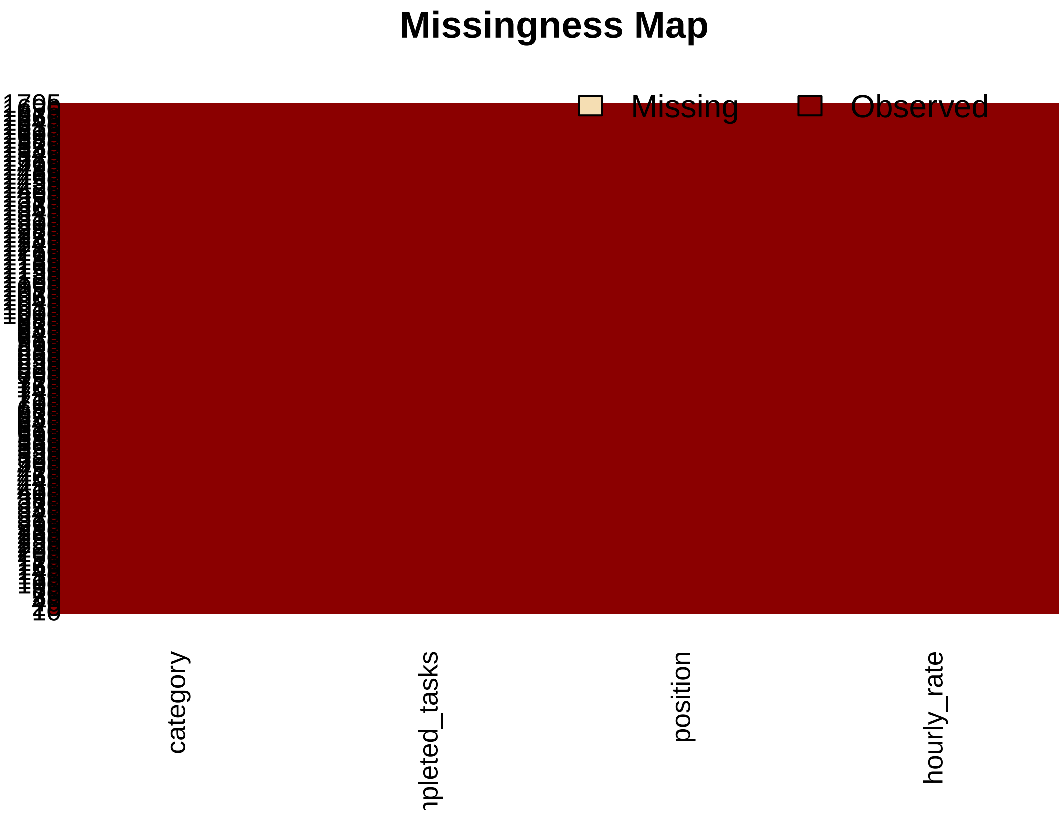
```
data_hired <- subset(data_hired,select = c("hourly_rate","position","num_completed_tasks","category"))
library(Amelia)
```

```
## Loading required package: Rcpp
```

```
## Warning: package 'Rcpp' was built under R version 3.4.2
```

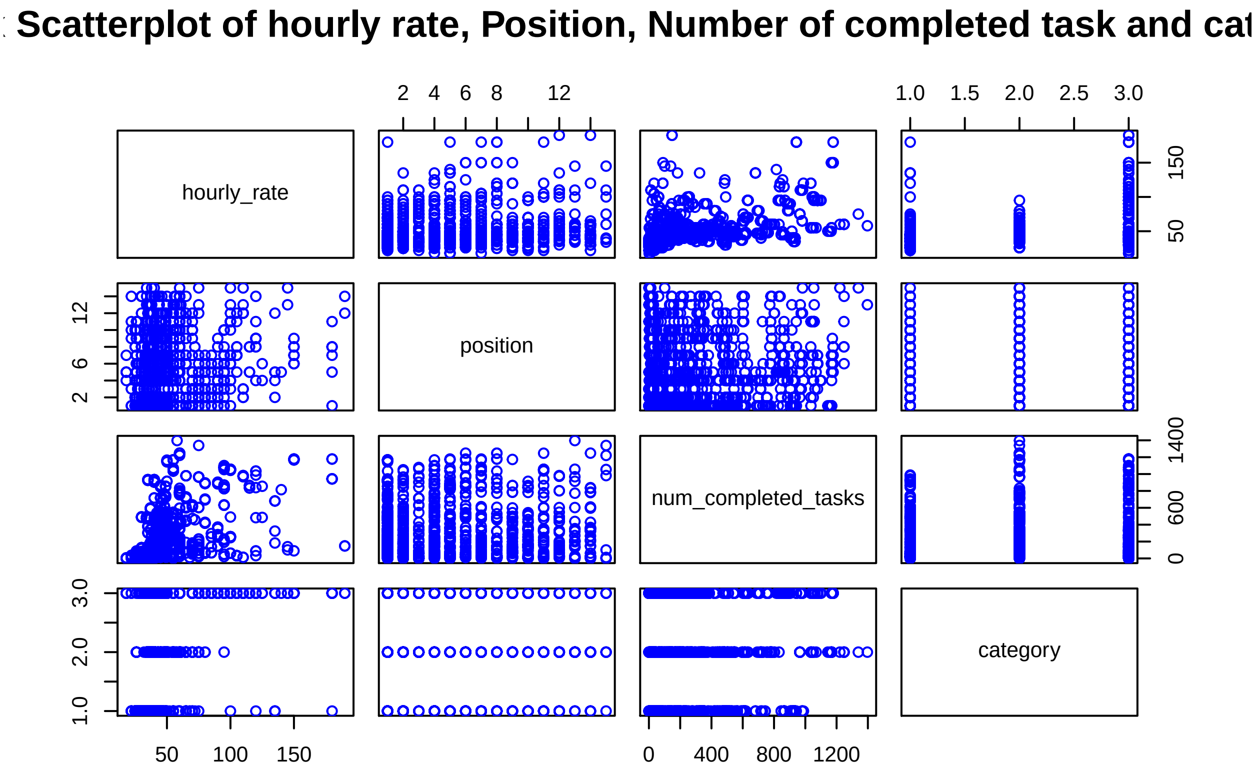
```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.4, built: 2015-12-05)
## ## Copyright (C) 2005-2017 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
missmap(data_hired)
```

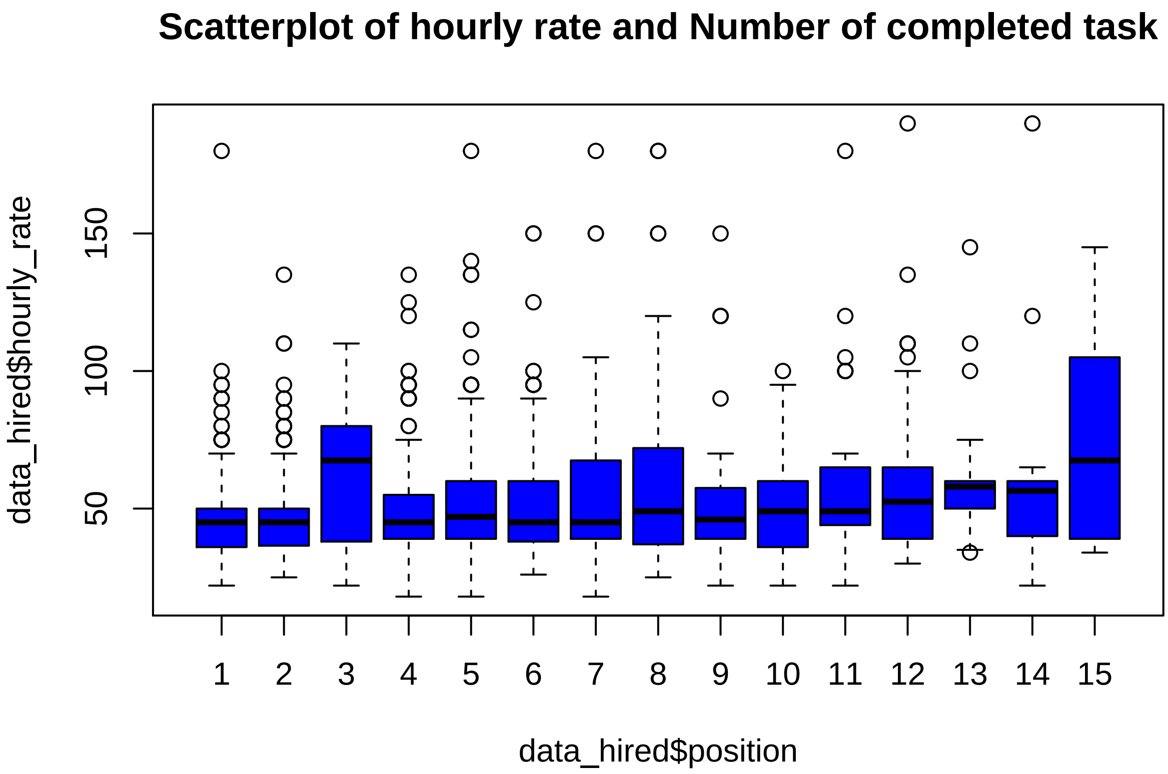


Step 3: Find the correlation between dependent and independent variable by visualizing data

```
plot(data_hired, col="blue", main="Matrix Scatterplot of hourly rate, Position, Number of completed tas
```

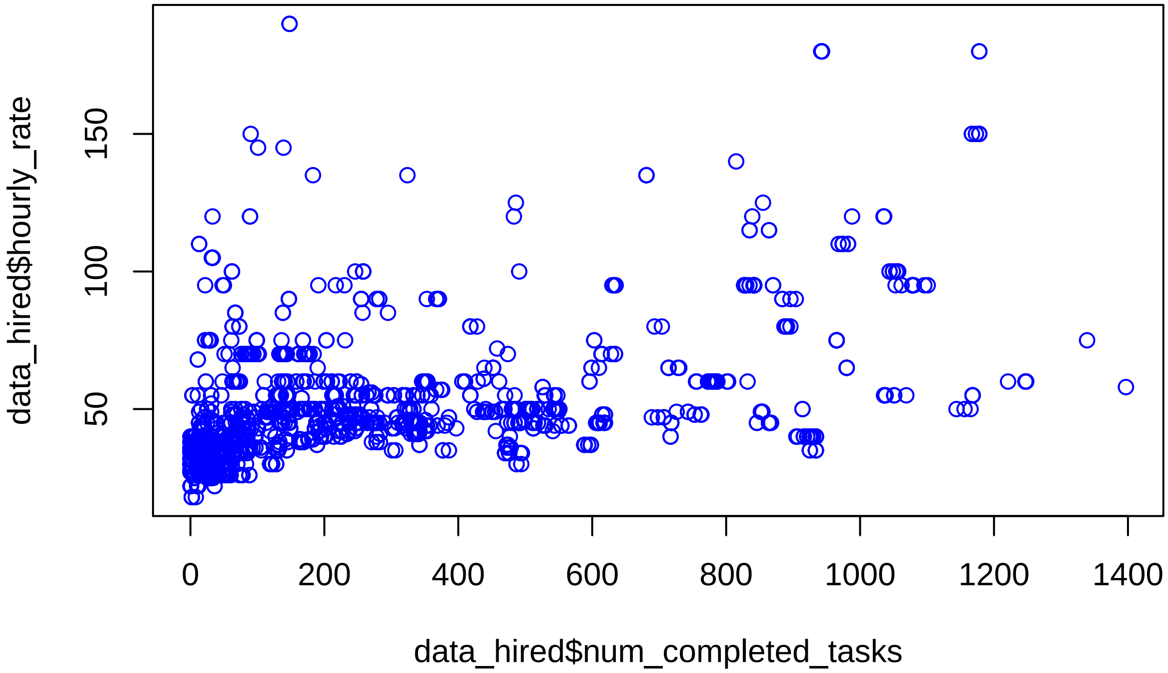


```
plot(data_hired$hourly_rate ~ data_hired$position, col="blue", main="Scatterplot of hourly rate and Numb
```



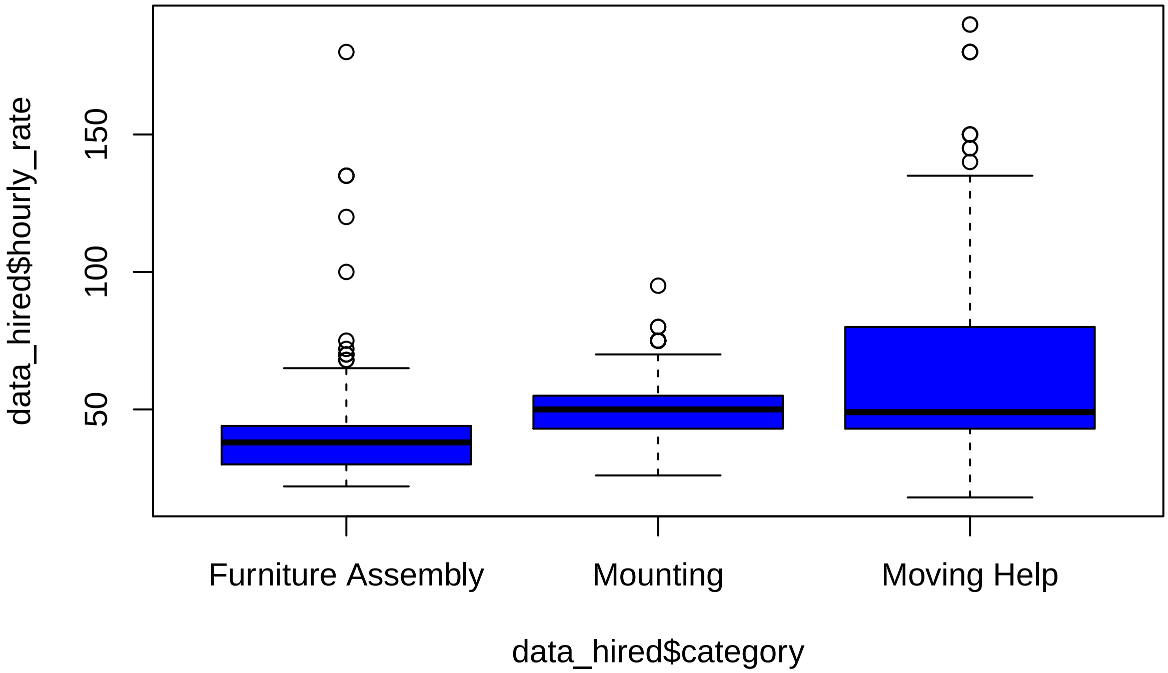
```
plot(data_hired$hourly_rate ~ data_hired$num_completed_tasks, col="blue", main="Bar char to display rela
```

Bar char to display relation between the Hourly rate and Position



```
plot(data_hired$hourly_rate ~ data_hired$category, col="blue", main="Bar chart to display relation betw
```

Bar chart to display relation between Hourly Rate and Category



Form above second graphs we can not conculed the conclued the postion have a positive influence on the hourly rate. In the third graph can see the positive increment in hourly rate as per the frequency of task submissions. and finally category plays a positive influence on hourly rate.

Step 4: Create sample data set to bult the model

```
set.seed(100)
data_rows_sample <- sample(1:nrow(data_hired),round(0.7*nrow(data_hired)))
data_hired_sample <- data_hired[data_rows_sample,]
data_hired_test <- data_hired[-data_rows_sample,]
```

Step 5: Build linear regression model to predict the hourly rate from the position, category, and a number of completed task.

```
model <- lm(hourly_rate~category+position+num_completed_tasks,data_hired_sample)
summary(model)
```

```
##
## Call:
## lm(formula = hourly_rate ~ category + position + num_completed_tasks,
##     data = data_hired_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.296 -10.341  -0.495   6.511 128.185
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          27.552991   1.097778  25.099  < 2e-16 ***
## categoryMounting     10.596508   1.212386   8.740  < 2e-16 ***
## categoryMoving Help  22.919228   1.213564  18.886  < 2e-16 ***
## position2             1.651625   1.497640   1.103 0.270332
## position3            12.372231   2.586835   4.783 1.95e-06 ***
## position4             0.802252   1.697114   0.473 0.636504
## position5             2.265486   2.079467   1.089 0.276176
## position6             5.711062   2.543759   2.245 0.024945 *
## position7             9.125585   2.361604   3.864 0.000118 ***
## position8            15.526949   3.029250   5.126 3.46e-07 ***
## position9             6.892842   2.983051   2.311 0.021024 *
## position10            7.070930   2.733395   2.587 0.009804 **
## position11            6.807191   3.170070   2.147 0.031971 *
## position12           13.124561   3.265362   4.019 6.21e-05 ***
## position13            7.775567   3.601056   2.159 0.031033 *
## position14            6.694506   3.595881   1.862 0.062893 .
## position15           -3.462736   8.442998  -0.410 0.681784
## num_completed_tasks   0.031410   0.001711  18.354  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.73 on 1176 degrees of freedom
## Multiple R-squared:  0.4364, Adjusted R-squared:  0.4282
## F-statistic: 53.56 on 17 and 1176 DF,  p-value: < 2.2e-16
```

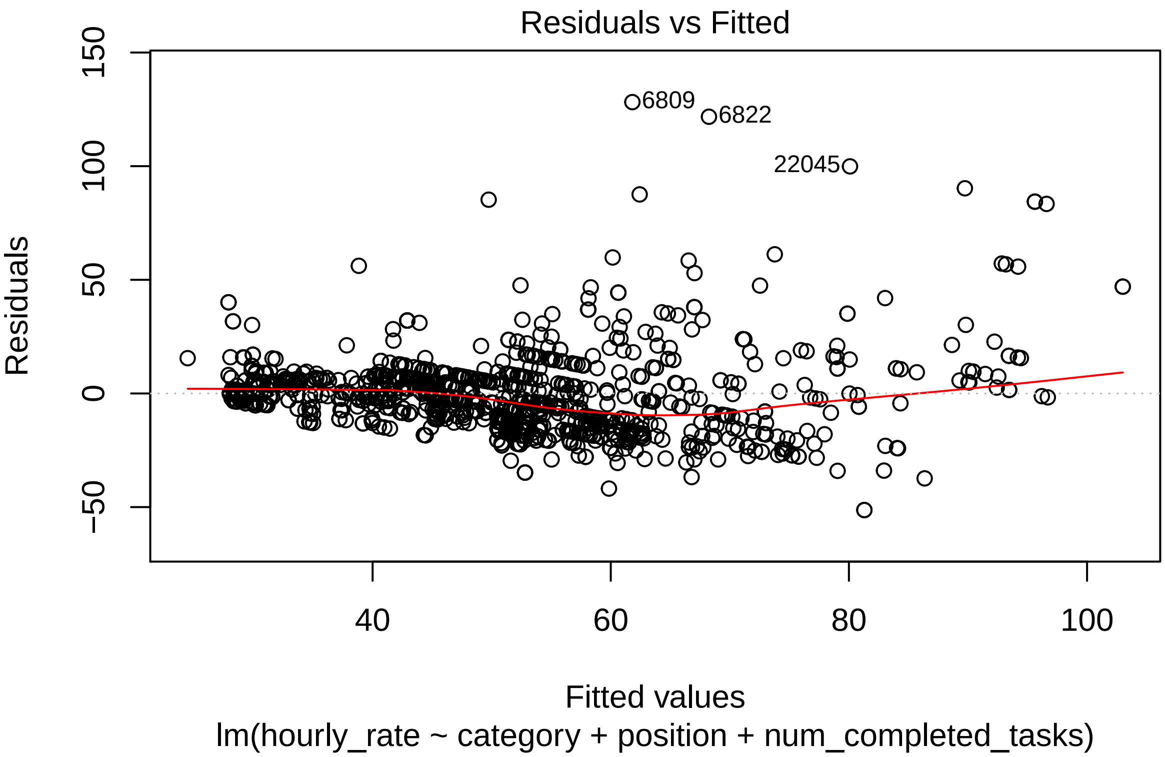
Step 6: Analysis

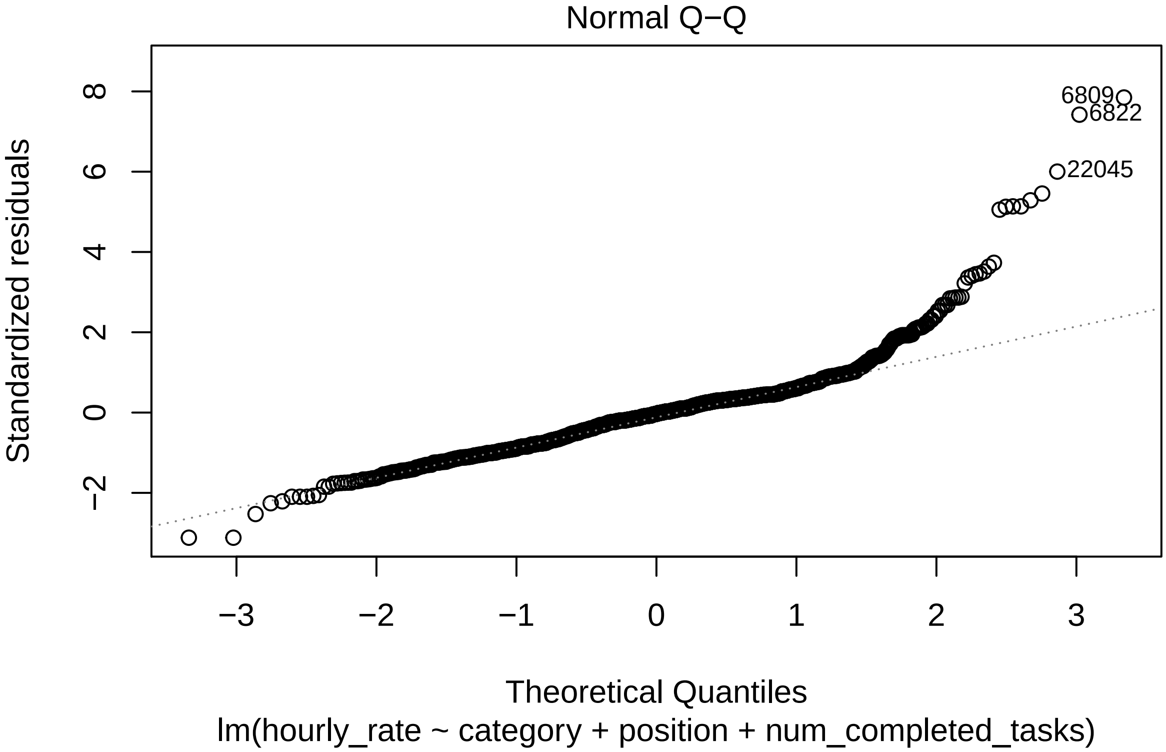
The output shows the F-statistic: 23.05 (p-value: < 2.2e-16) clearly shows reject the null hypothesis that the variables category, position, num_completed_tasks, collectively have no effect on hourly_rate. In addition, the output also shows that R squre is 0.4269. Here our main goal is not to produce precise predictions.So it would be not get problem having low r squre.

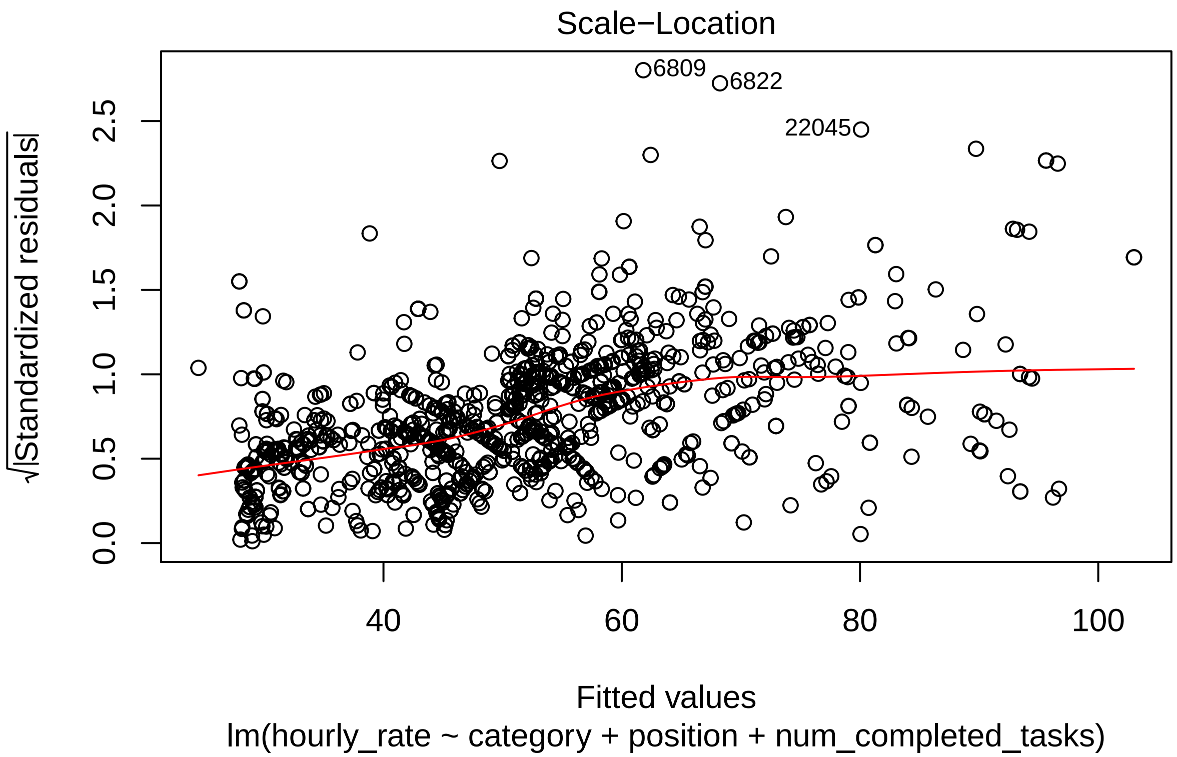
Example:

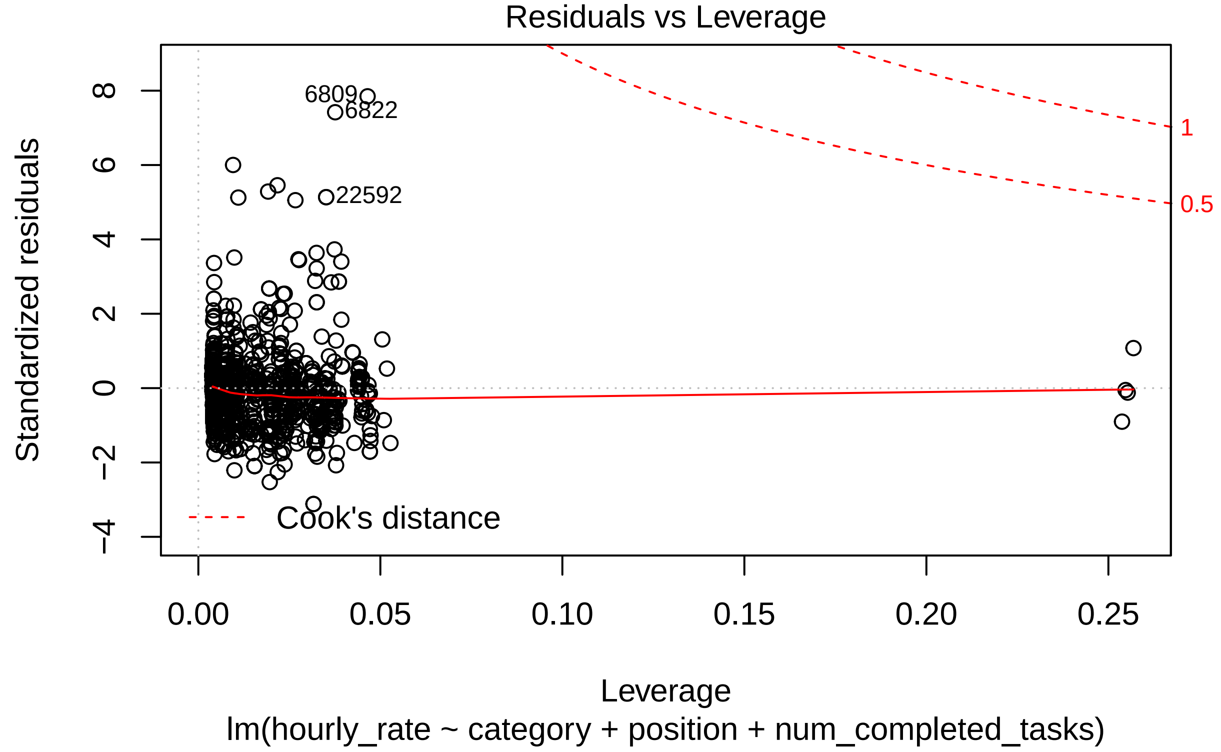
Visulise and Apply above model on test data

```
plot(model)
```









```r
suggest_hourly_rates <- predict.lm(model,data_hired_test)

#class(data_hired_test)
```

The above liner regression model we can use for suggest hourly rates to Taskers that would maximize their opportunity to be hired.