Lab 1: Multiple Linear Regression

Srishti Saha (ss1078)

06 September, 2019

Lab 1: Multiple Linear Regression

```
#importing general libraries
library(tidyr)
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(moderndive)
```

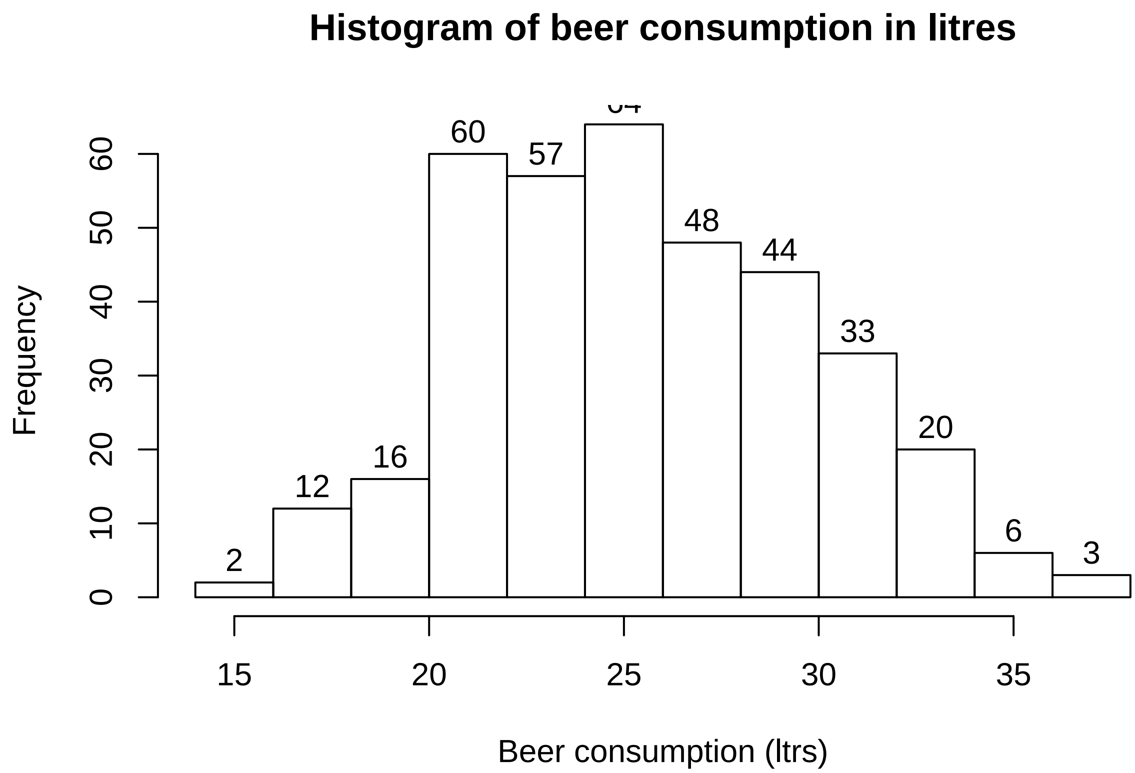
Exercise 1: Import data and investigate response variable

The data on beer is first imported and transformed.

```
#import dataset
beer <- read.csv("consumo_cerveja.csv",stringsAsFactors = FALSE, sep = ",",dec=",")
# rename the variables
beer$date <- beer$Data
beer$temp_median_c <- beer$Temperatura.Media..C.
beer$temp_min_c <- beer$Temperatura.Minima..C.
beer$temp_max_c <- beer$Temperatura.Maxima..C.
beer$precip_mm <- beer$Precipitacao..mm.
beer$weekend <- factor(beer$Final.de.Semana)
beer$beer_cons_liters <- as.numeric(beer$Consumo.de.cerveja..litros.)
beer <- beer[ , 8:ncol(beer)]
```

Treating beer_cons_liters as the response variable. KIt depicts the beer consumption in litres.

```
#plot histogram of beer_cons_liters
hist(beer$beer_cons_liters ,main='Histogram of beer consumption in litres', xlab= 'Beer consumption (lt
```

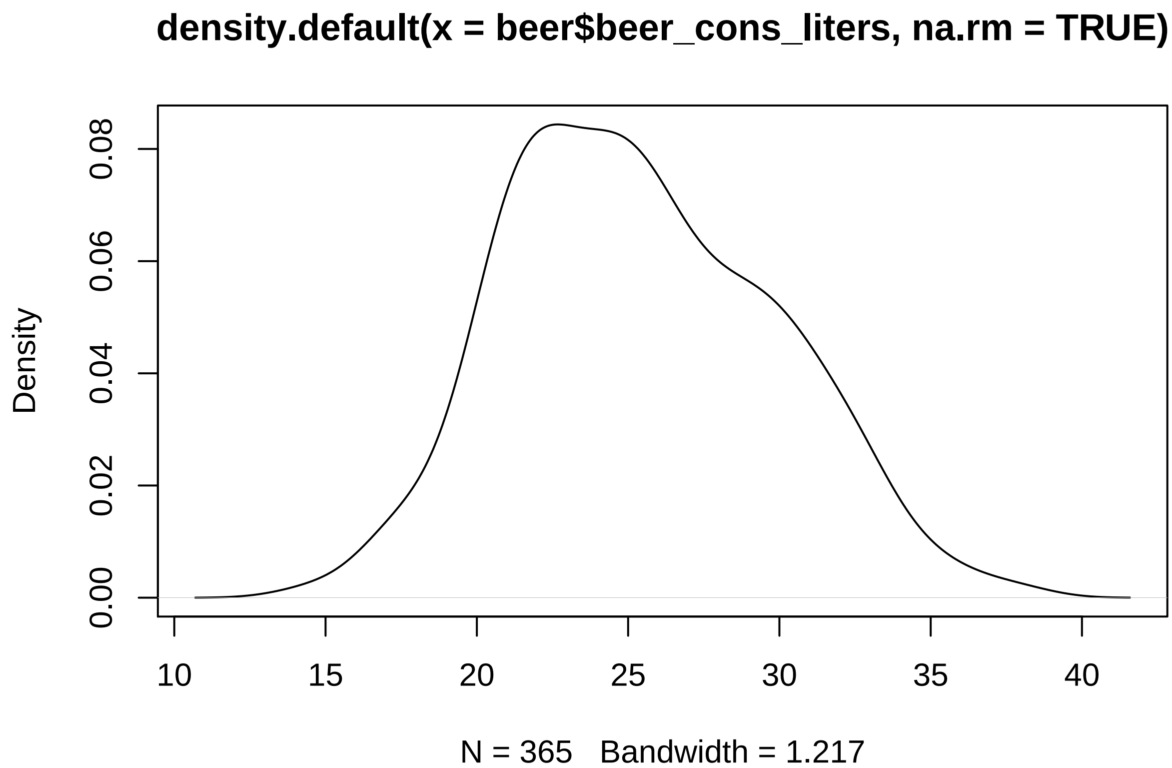



```r
#plotting the density plot to observe better
#plot density plot of beer_cons_liters

sum(is.na(beer$beer_cons_liters)) # 576 missing values
```

```
## [1] 576
```

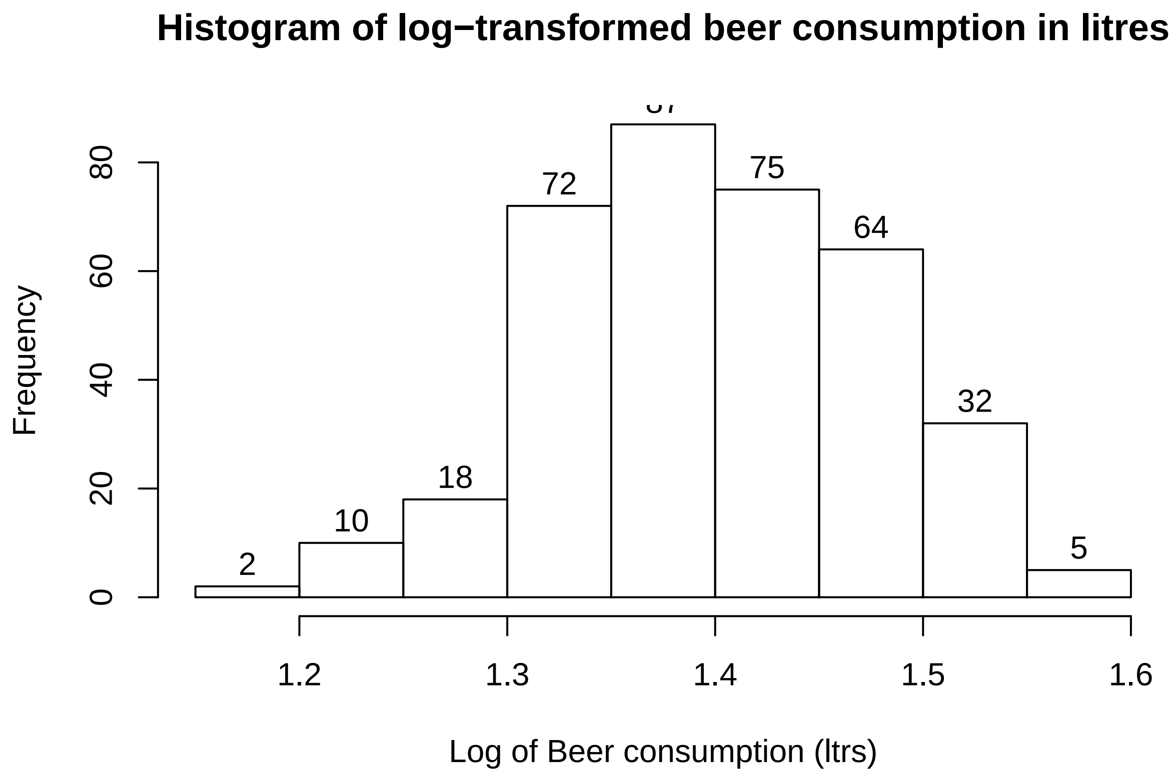
```r
density_beer<-density(beer$beer_cons_liters,na.rm=TRUE)
plot(density_beer) # plots the results
```



The distribution looks centres but has a slight right skew. This might fail the assumption of normality. To centre the distribution, let's look at log transformation of the variable

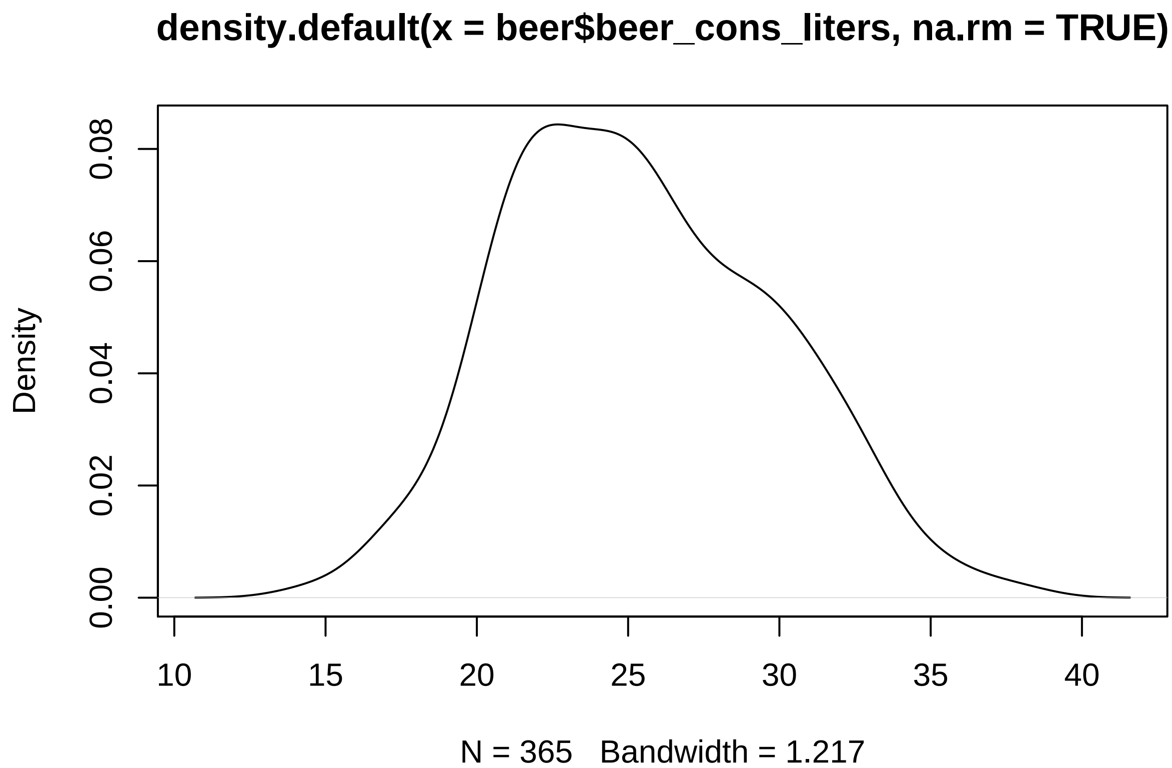
```
#log transformation of beer_cons_liters
beer$log.beer_cons_liters<- log10(beer$beer_cons_liters)

#plot histogram of beer_cons_liters
hist(beer$log.beer_cons_liters ,main='Histogram of log-transformed beer consumption in litres', xlab= 'l
```




```r
#plot density plot of beer_cons_liters

density_beer_log<-density(beer$beer_cons_liters,na.rm=TRUE)
plot(density_beer_log) # plots the results
```

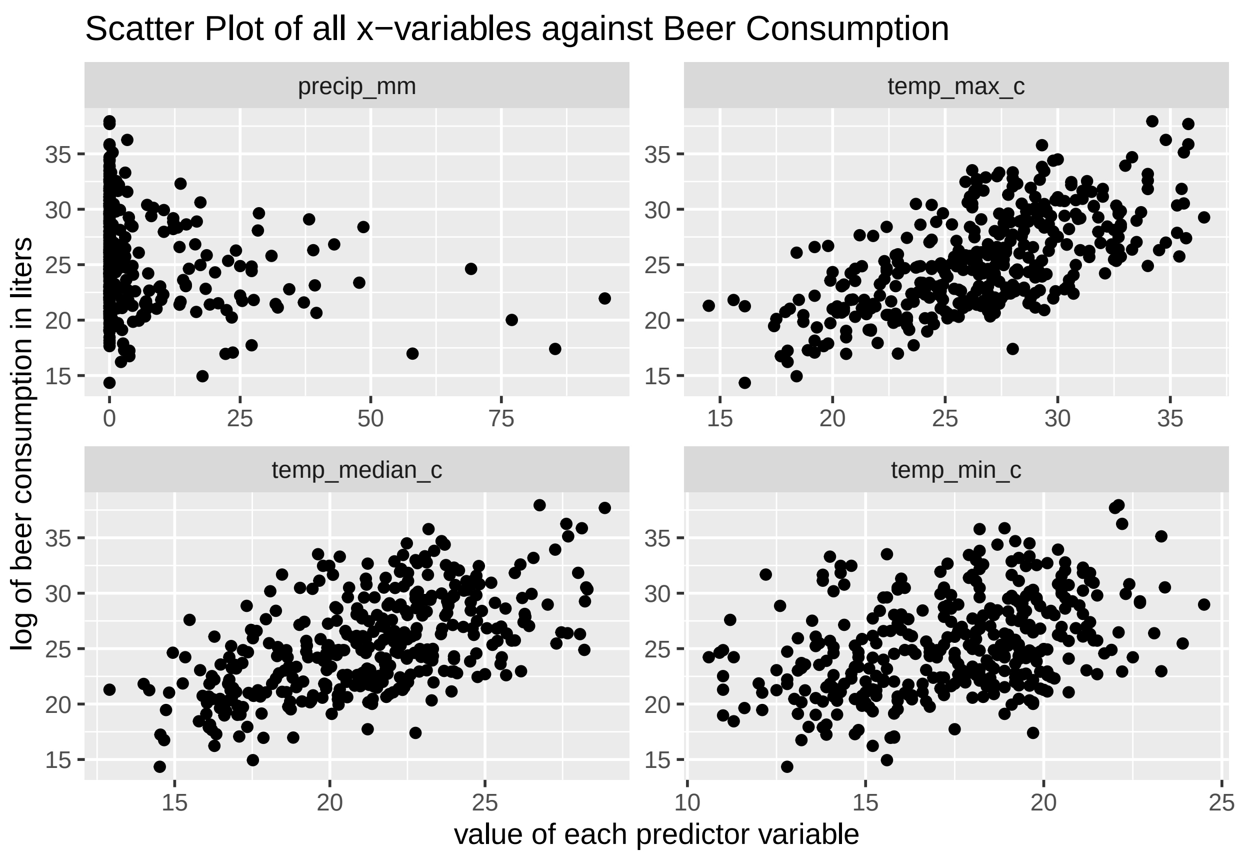


There seems to be a slight left skew in this plot. HOwever, since there is no significant improvement, we will consider beer_cons_liters.

Exercise 2: Investigating response variable versus all predictor variables

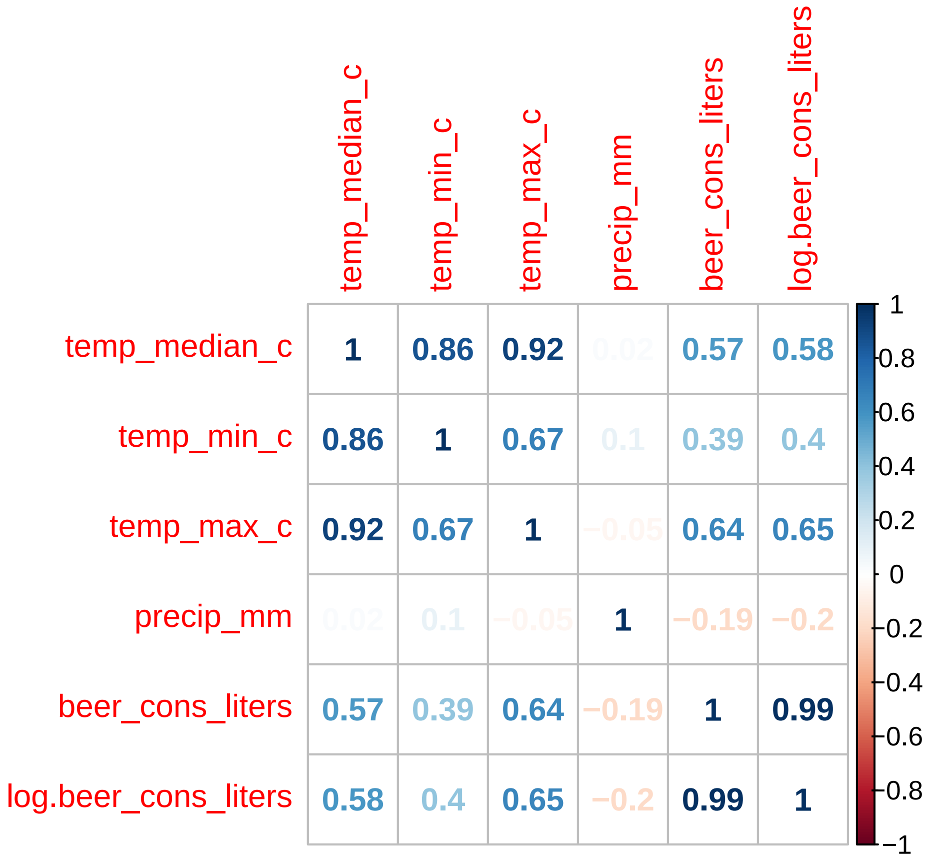
```
# scatter plots of variables versus beer_cons_liters
beer %>%
  gather(-log.beer_cons_liters,-date, -weekend,-beer_cons_liters, key = "var", value = "value") %>%
  ggplot(aes(x = value, y = beer_cons_liters)) +
    geom_point() +
    facet_wrap(~ var, scales = "free") + labs(title="Scatter Plot of all x-variables against Beer Consu
```

```
## Warning: Removed 2304 rows containing missing values (geom_point).
```



The relationship between the minimum, maximum and median temperatures and beer consumption (log-transformed) is linear. However, the relationship between precipitation and beer consumption does not seem to be completely linear.

```
#Create correlation matrix
corrplot_beer <- cor(beer[c(2:5,7:8)],use="complete.obs")
corrplot(corrplot_beer, method = "number")
```



We see here that maximum temperature has a high correlation with our response variable (positive correlation of 0.65). Precipitation has a negative correlation with beer consumption(-0.2).

Exercise 3: Does it make sense to include all three temperature variables?

From the correlation matrix, we see that temp_max_c and temp_median_c have a high correlation among themselves. Moreover, they have a similar correlation with the response variable (log.beer_cons_liters). Furthermore, from the scatter plots we see that these variables have a similar trend. Although minimum temperature has a correlation with the other variables, it might be because of the sclae. It has a similar trend with the response variable as the other temperature variables. Thus, we can choose one of these variables.

Hence, temperature variables selected: * temp_median_c

Exercise 4: Regression

```r
# regression model for predicting log.Rate votes from Age
lm_model_beers <- lm(beer_cons_liters~weekend+temp_median_c+precip_mm,data=beer);
summary(lm_model_beers)
```

```
##
## Call:
## lm(formula = beer_cons_liters ~ weekend + temp_median_c + precip_mm,
##     data = beer)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4802 -2.0347 -0.1904  1.8908  6.5165
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.47348    0.91957   7.040 9.77e-12 ***
## weekend1       5.22787    0.29855  17.511  < 2e-16 ***
## temp_median_c  0.83971    0.04245  19.782  < 2e-16 ***
## precip_mm     -0.07420    0.01086  -6.835 3.51e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.571 on 361 degrees of freedom
##   (576 observations deleted due to missingness)
## Multiple R-squared:  0.6612, Adjusted R-squared:  0.6584
## F-statistic: 234.8 on 3 and 361 DF,  p-value: < 2.2e-16
```

From the above model summary, we see that the intercept is 6.4735 which means that for a weekday (weekend=0) with 0 as median temperature and 0 as precip_mm will see beer consumption of 6.47 liters.

The estimate of weekend1 is 5.23 which implies that for all other variables as constant (same for male and female), a weekend (=1) will see a beer consumption of 5,23 liters more than a weekday (weekend=0)

According to the p-values all variables are significant in this model as the p-value is very low (of the order (e^{-16})). However, if we look at the t-values, median temperature (temp_median_c) seems to be the most significant predictor variable.

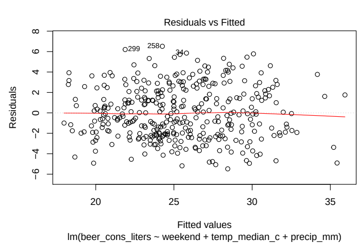
The R-squared value is 0.6612 while the adjusted-R-squared is 0.6584. This means 65.84% of the variability in beer_cons_liters (or log(beer_cons_liters)) is explained by this model.

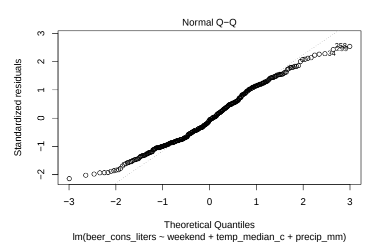
Exercise 5: Most significant covariate

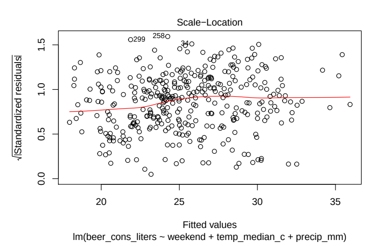
According to the absolute t-values of the significant variables, **temp_median_c is the most significant (t-value=19.782)** and weekend flag is the 2nd most (next) significant variable.

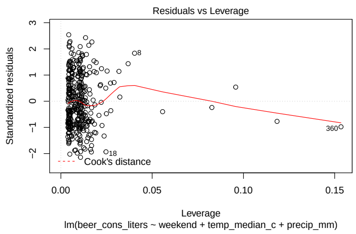
Exercise 6: Potential limitations of the model

```r
#residual versus fitted
plot(lm_model_beers)
```









1. If we look at the adjusted R-squared values, only ~66% of the variation is explained by the model. There might be other significant variables that have not been mtaken into account to fit the best model.
2. We lose a lot of data points (576 points) due to missing values in the response variable. This might lead to erroneous training of the data.
3. The model is susceptible to outliers. The leverage plot shows that the 360th observation has a high leverage on the model. This might be a problem which needs to be investigated further.

Problem 7: Compute in-sample root mean squared error (RMSE) for the regression model

```
#function for rmse
rmse <- function(error)
{
    sqrt(mean(error^2))
}

#apply the function on the residuals from the model
rmse(lm_model_beers$residuals)
```

```
## [1] 2.557158
```

The in-sample rmse of the model is 2.55.

Exercise 8: K-fold validation

```
#K-fold
set.seed(10) # use whatever number you want
# Now randomly re-shuffle the data
Data <- beer[sample(nrow(beer)),]
Data<- na.omit(Data)
# Define the number of folds you want
K <- 10
# Define a matrix to save your results into
RMSE <- matrix(0,nrow=10,ncol=1)
# Split the row indexes into k equal parts
kth_fold <- cut(seq(1,nrow(Data)),breaks=K,labels=FALSE)
# Now write the for loop for the k-fold cross validation
for(k in 1:10){
```

```r
# Split your data into the training and test datasets
test_index <- which(kth_fold==k)
train <- Data[-test_index,]
test <- Data[test_index,]
# Now that you've split the data,
model<-lm(beer_cons_liters~weekend+temp_median_c+precip_mm,data=train)
pred_test<- get_regression_points(model,newdata=test)
RMSE[k,] <- rmse(pred_test$residual) # write your code for computing RMSE for each k here
# You should consider using your code for question 7 above
}

#Calculate the average of all values in the RSME matrix here.
paste0("RMSE after K-fold is= ",mean(RMSE))
```

```
## [1] "RMSE after K-fold is= 2.58499464524515"
```

The average RMSE of the model above after K-fold cross validation (with k=10) is 2.585.

Exercise 9: Regression Model with interaction Terms

```r
#revise the regression model with interaction terms
lm_model_beers_it <- lm(beer_cons_liters~(weekend+temp_median_c+precip_mm )^2,data=beer);
summary(lm_model_beers_it)
```

```
##
## Call:
## lm(formula = beer_cons_liters ~ (weekend + temp_median_c + precip_mm)^2,
##     data = beer)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4592 -2.0254 -0.1756  2.0274  6.5737
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.2622667  1.1426702   5.480 8.02e-08 ***
## weekend1                5.3725915  1.9991143   2.687  0.00754 **
## temp_median_c           0.8469778  0.0529214  16.004  < 2e-16 ***
## precip_mm              -0.0198878  0.1120946  -0.177  0.85928
## weekend1:temp_median_c  0.0008413  0.0937850   0.009  0.99285
## weekend1:precip_mm     -0.0309290  0.0233426  -1.325  0.18602
## temp_median_c:precip_mm -0.0020273  0.0051361  -0.395  0.69329
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.575 on 358 degrees of freedom
##   (576 observations deleted due to missingness)
## Multiple R-squared:  0.6631, Adjusted R-squared:  0.6575
## F-statistic: 117.4 on 6 and 358 DF,  p-value: < 2.2e-16
```

If we look at the p-values of the interaction terms, they are very high. This implies that we can accept the null hypothesis that $\beta_j = 0$. Thus, these interaction terms are not significant.

Exercise 10:

```r
#K-fold
set.seed(10) # use whatever number you want
# Now randomly re-shuffle the data
Data <- beer[sample(nrow(beer)),]
Data<- na.omit(Data)
# Define the number of folds you want
K <- 10
# Define a matrix to save your results into
RMSE <- matrix(0,nrow=10,ncol=1)
# Split the row indexes into k equal parts
kth_fold <- cut(seq(1,nrow(Data)),breaks=K,labels=FALSE)
# Now write the for loop for the k-fold cross validation
for(k in 1:10){
# Split your data into the training and test datasets
test_index <- which(kth_fold==k)
train <- Data[-test_index,]
test <- Data[test_index,]
# Now that you've split the data,
model_it<-lm(beer_cons_liters~(weekend+temp_median_c+precip_mm )^2,data=train)
pred_test<- get_regression_points(model_it,newdata=test)
RMSE[k,] <- rmse(pred_test$residual) # write your code for computing RMSE for each k here
# You should consider using your code for question 7 above
}

#Calculate the average of all values in the RSME matrix here.
paste0("RMSE after K-fold for the model with interaction terms is= ",mean(RMSE))
```

```
## [1] "RMSE after K-fold for the model with interaction terms is= 2.61230547896688"
```

The RMSE after including interaction terms is 2.61 which is higher than the one obtained in Question 8 (2.58). As a result, we can infer that the root mean squared error is higher for this model i.e. the model with interaction tems has a higher average residual value (hhigher errors). Thus, we should not consider the interaction terms.