Lab 1: Multiple Linear Regression

Sebastián Soriano Pérez [ss1072]

9/6/2019

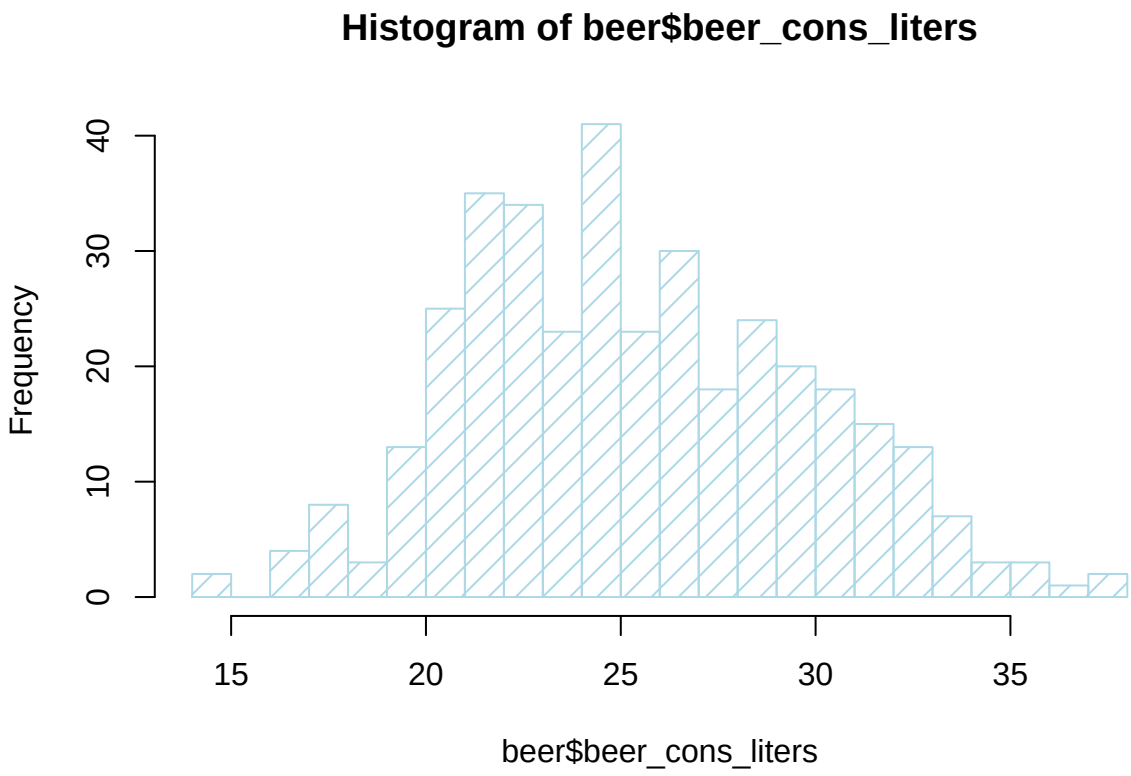
Beer Consumption

```
beer <- read.csv("consumo_cerveja.csv",stringsAsFactors = FALSE, sep = ",",dec=",")
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)]
```

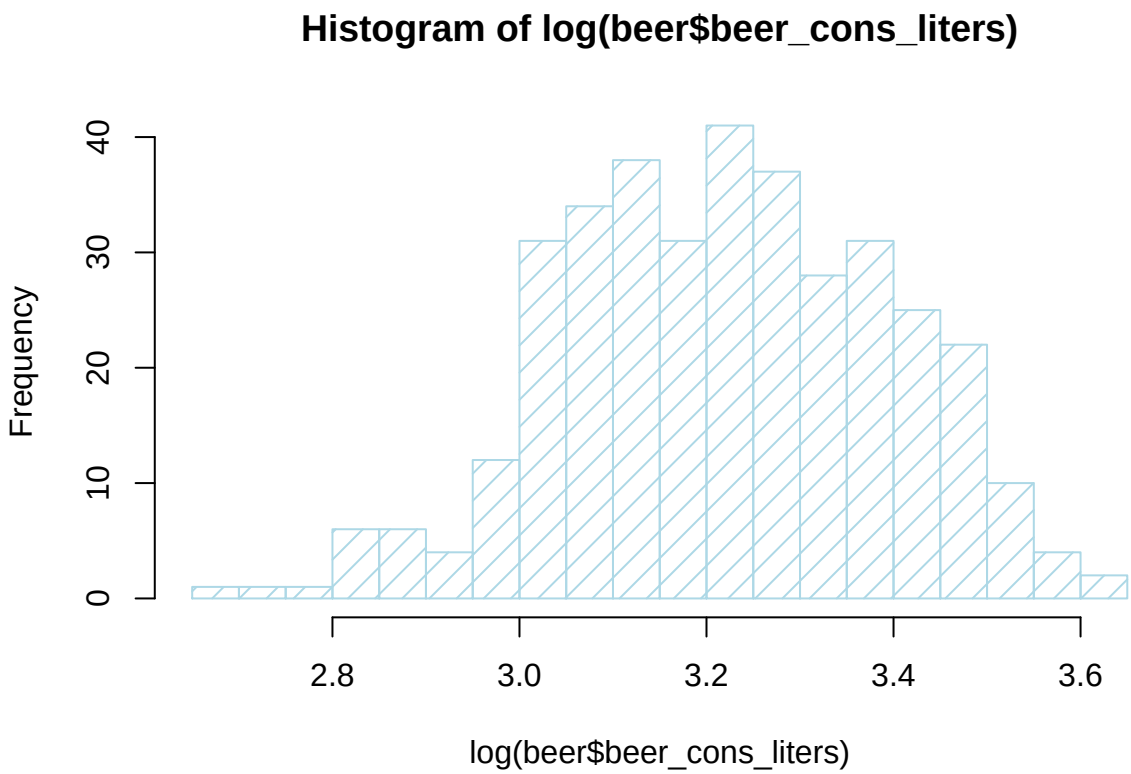
Exercise 1

Make a histogram of beer_cons_liters. Describe the distribution. Is the normality assumption a plausible one here? If you think the histogram does not look normal enough, make a histogram of log(beer_cons_liters). Does that look more "normal" than beer_cons_liters?

```
hist(beer$beer_cons_liters, breaks = 20, col = "lightblue", density = 10)
```



```r
hist(log(beer$beer_cons_liters), breaks = 20, col = "lightblue", density = 10)
```

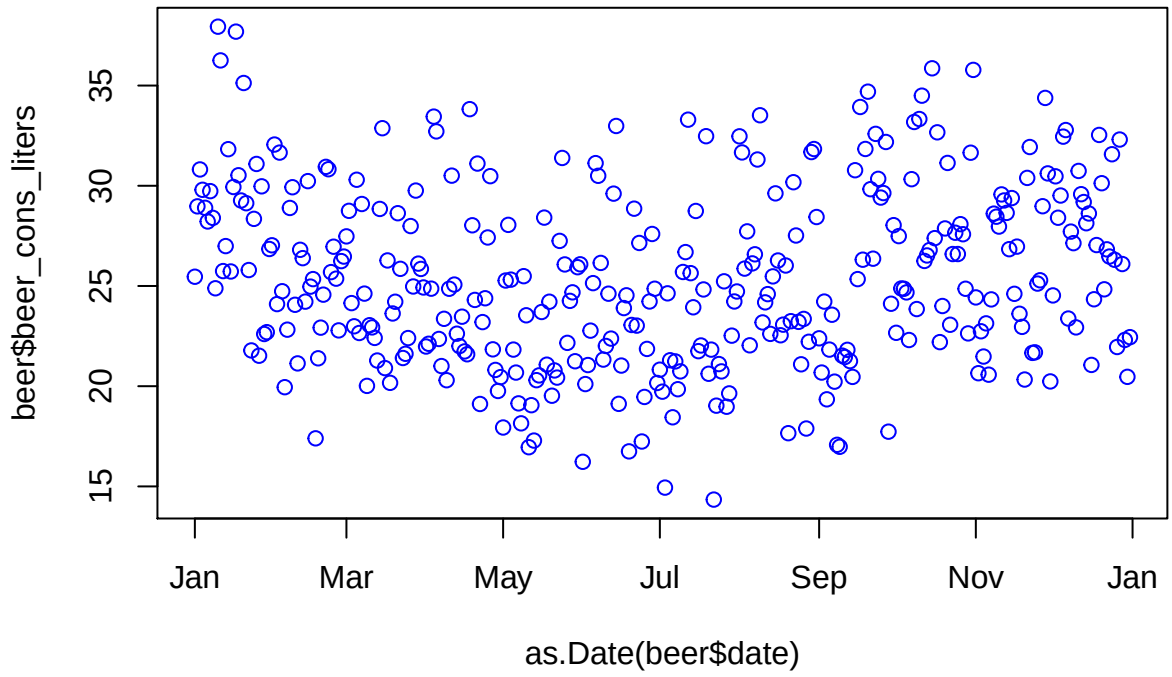


The first variable seems to adjust better to a normal distribution. Although the logarithmic transformation corrects some of the big drops in the frequency of the categories of (23, 24) and (25, 26) liters on the histogram for beer_cons_liters, it skews the data a bit too much to the right. I will use beer_cons_liters as the response variable.

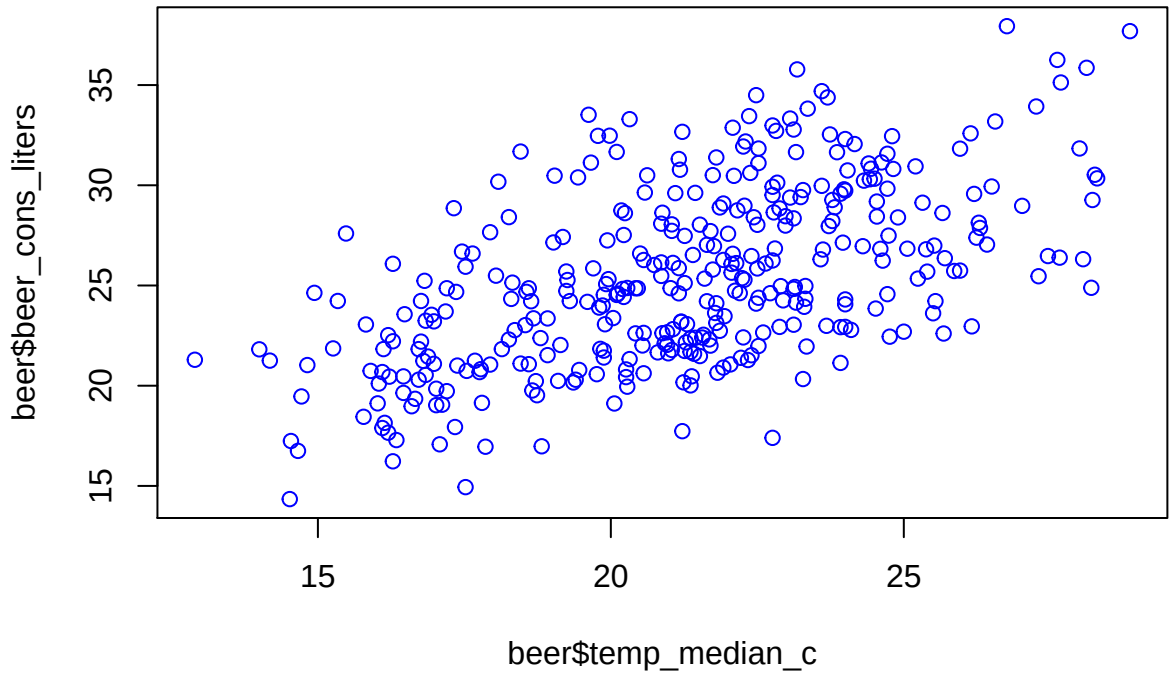
Exercise 2

Make exploratory plots of beer_cons_liters (or log(beer_cons_liters)) versus each potential predictor. Are all the relationships linear? If any one of them is nonlinear, describe the relationship.

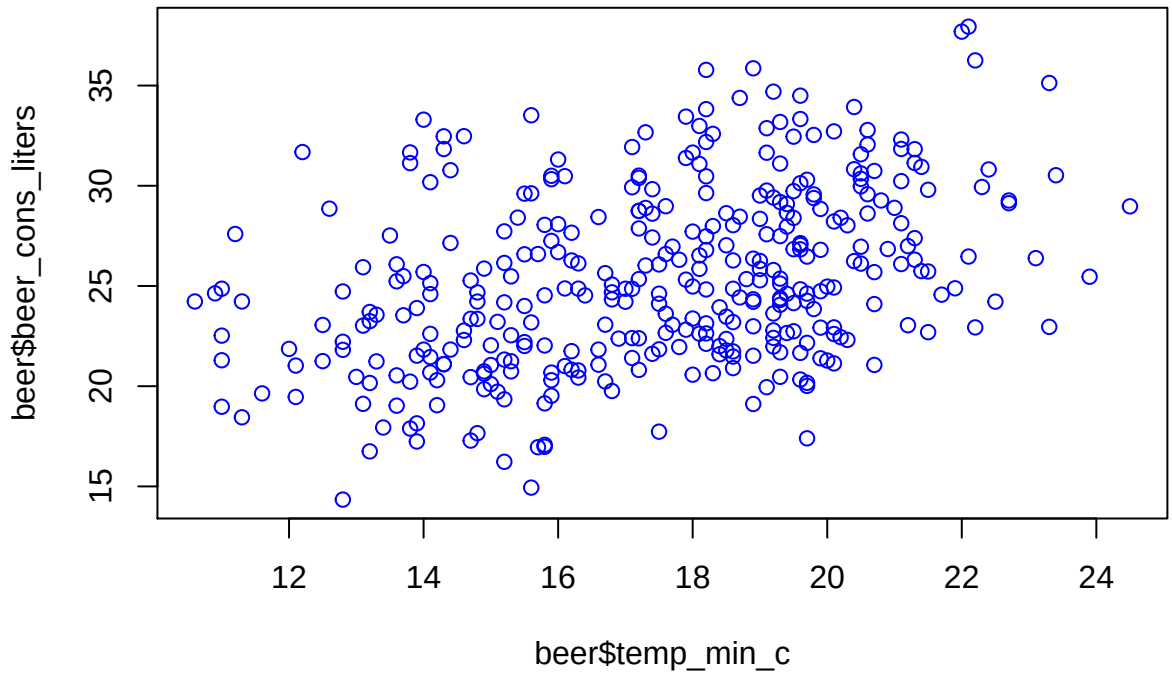
```r
plot(beer$beer_cons_liters ~ as.Date(beer$date), pch = 1, col = 'blue')
```



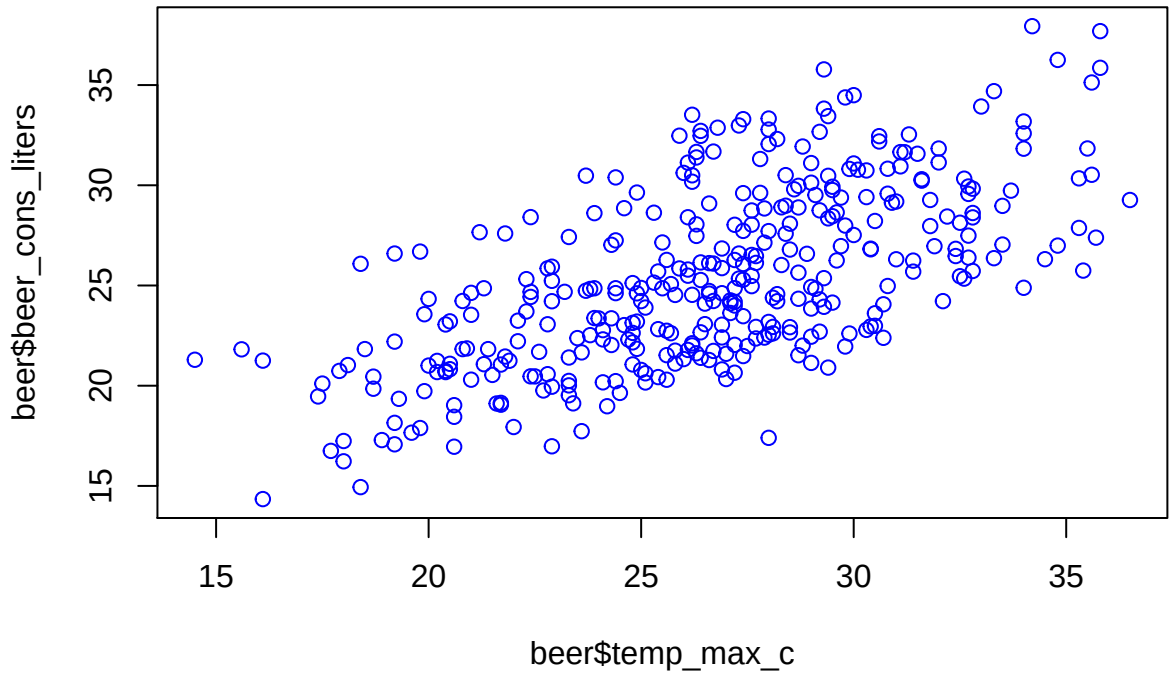

```
plot(beer$beer_cons_liters ~ beer$temp_median_c, pch = 1, col = 'blue')
```



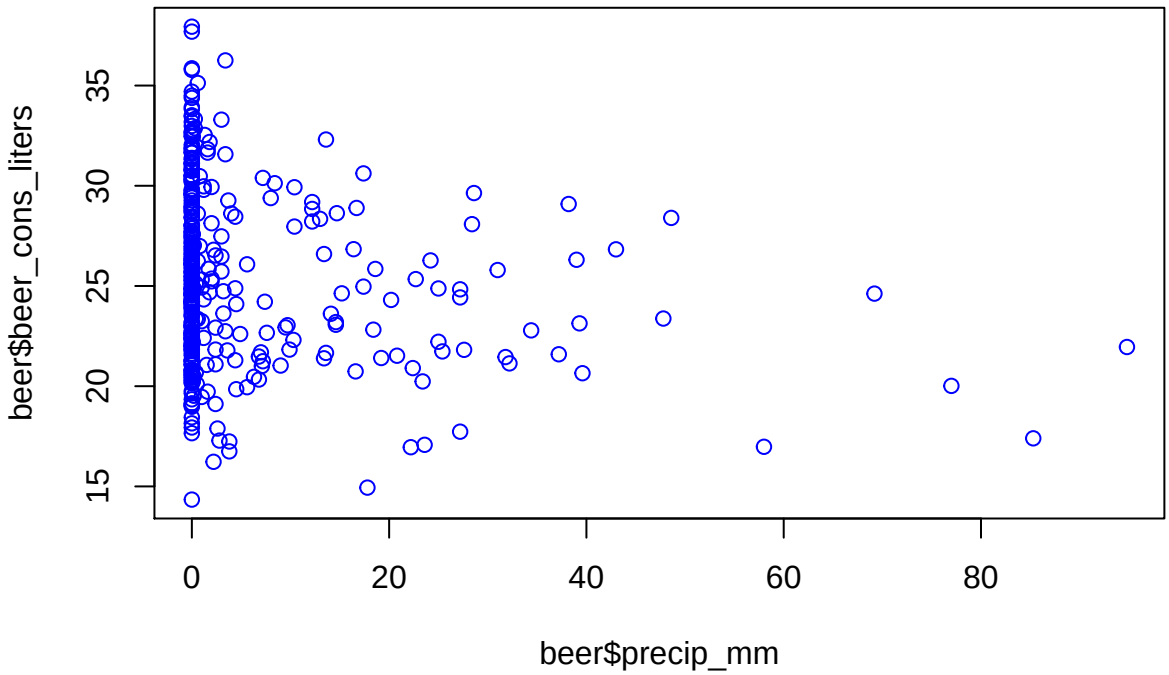

```
plot(beer$beer_cons_liters ~ beer$temp_min_c, pch = 1, col = 'blue')
```



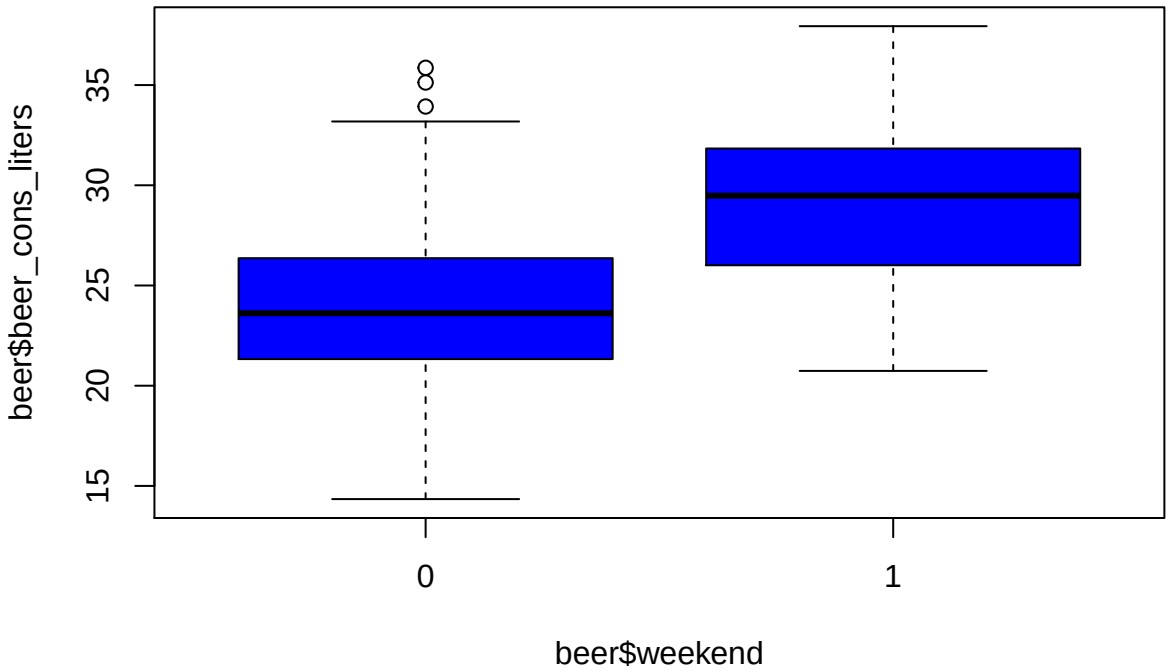

```
plot(beer$beer_cons_liters ~ beer$temp_max_c, pch = 1, col = 'blue')
```




```
plot(beer$beer_cons_liters ~ beer$precip_mm, pch = 1, col = 'blue')
```




```
plot(beer$beer_cons_liters ~ beer$weekend, pch = 1, col = 'blue')
```



Most of the relationships appear to be linear. The exceptions are date, temp_min_c, and precip_mm. date does not seem to predict beer_cons_liters at all, as beer consumption remains evenly spread throughout the year for the data that was captured. temp_min_c just barely seems to predict beer_cons_liters, but the data points are all over the place it is hard to say it could be used as a predictor variable. precip_mm does not seem to predict beer_cons_liters at all, as most data points are at precip_mm = 0 and beer_cons_liters seems unaffected by this value.

Exercise 3

Does it make sense to include all three of temp_median_c, temp_min_c and temp_max_c as predictors in

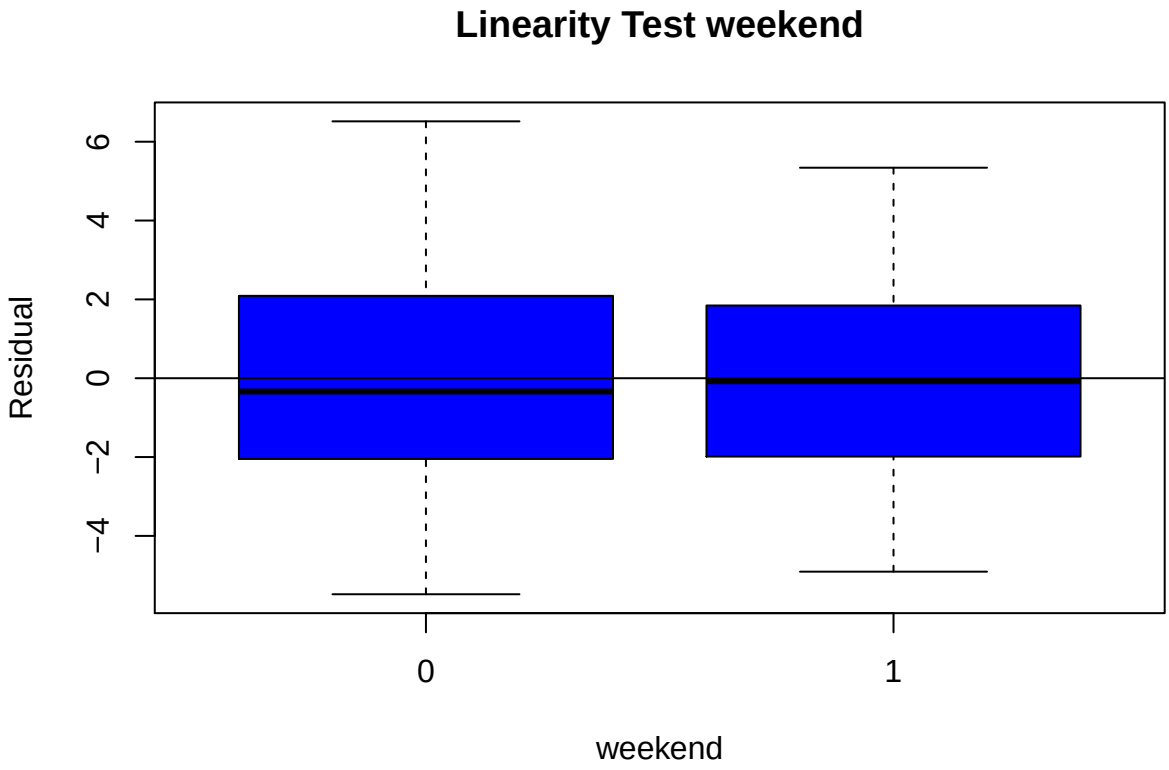
a MLR model for predicting beer_cons_liters (or log(beer_cons_liters))? Justify your response in one or two sentences.
I believe it makes sense to include either temp_max_c or temp_median_c, since they both seem to have a linear correlation with beer_cons_liters. Including both would probably be problematic as they are highly correlated (they both indicate a feature of temperature in the same places). The minimun temperature temp_min_c does not seem to influence or predict the response variable very well, so I don't believe it makes sense to include it.

Exercise 4
Fit a linear model for beer_cons_liters (or log(beer_cons_liters)) using weekend, precip_mm, and temp_median_c as your predictors. Interpret all the parameters of the fitted regression model in context of the data. What percent of the variability in beer_cons_liters (or log(beer_cons_liters)) is explained by your model?

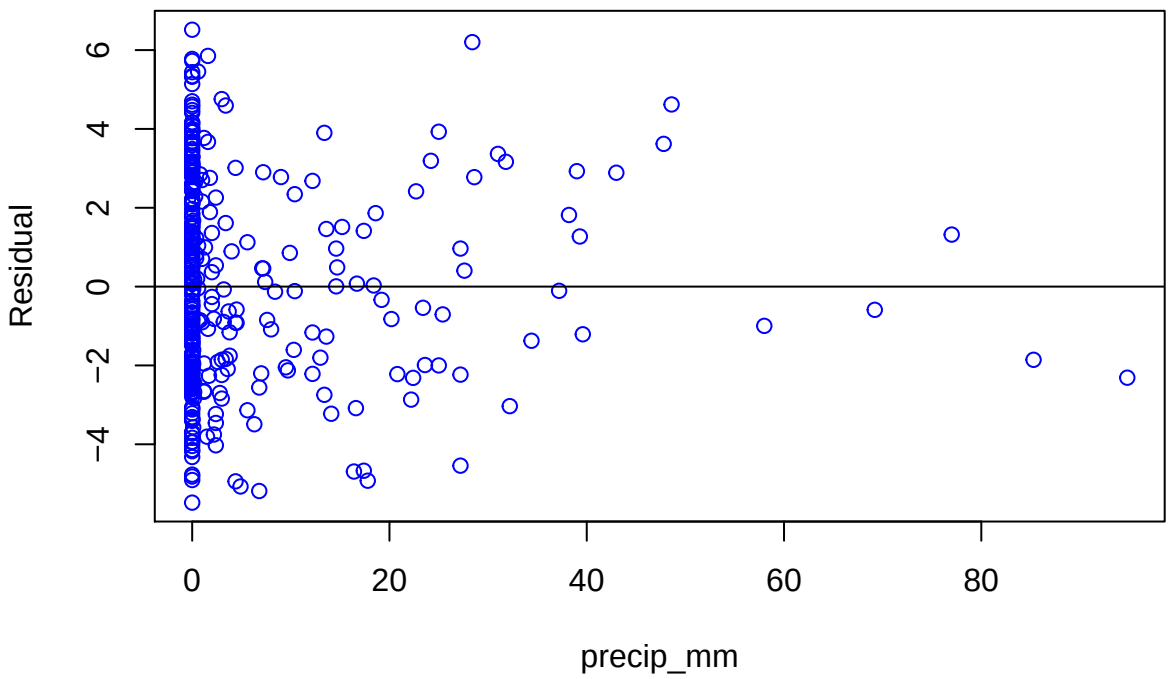
```
lm_beer <- lm(beer_cons_liters ~ weekend + precip_mm + temp_median_c, beer)
summary(lm_beer)
```

```
##
## Call:
## lm(formula = beer_cons_liters ~ weekend + precip_mm + temp_median_c,
##     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 ***
## precip_mm     -0.07420    0.01086  -6.835 3.51e-11 ***
## temp_median_c  0.83971    0.04245  19.782  < 2e-16 ***
## ---
## 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
```

```
plot(y = lm_beer$residual, x = beer[1:365,]$weekend, xlab = "weekend", ylab = "Residual",
     main = "Linearity Test weekend", col = 'blue')
abline(0,0)
```

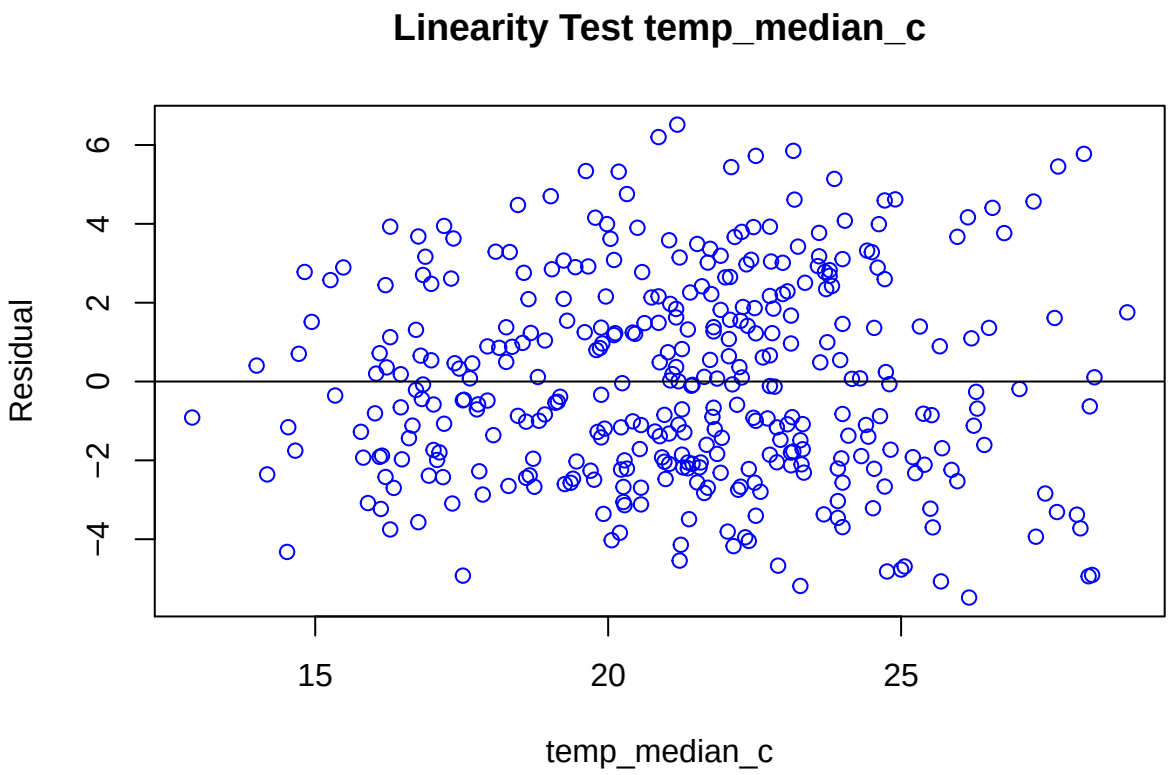



```
plot(y = lm_beer$residual, x = beer[1:365,]$precip_mm, xlab = "precip_mm",
     ylab = "Residual", main = "Linearity Test precip_mm", col = 'blue')
abline(0,0)
```

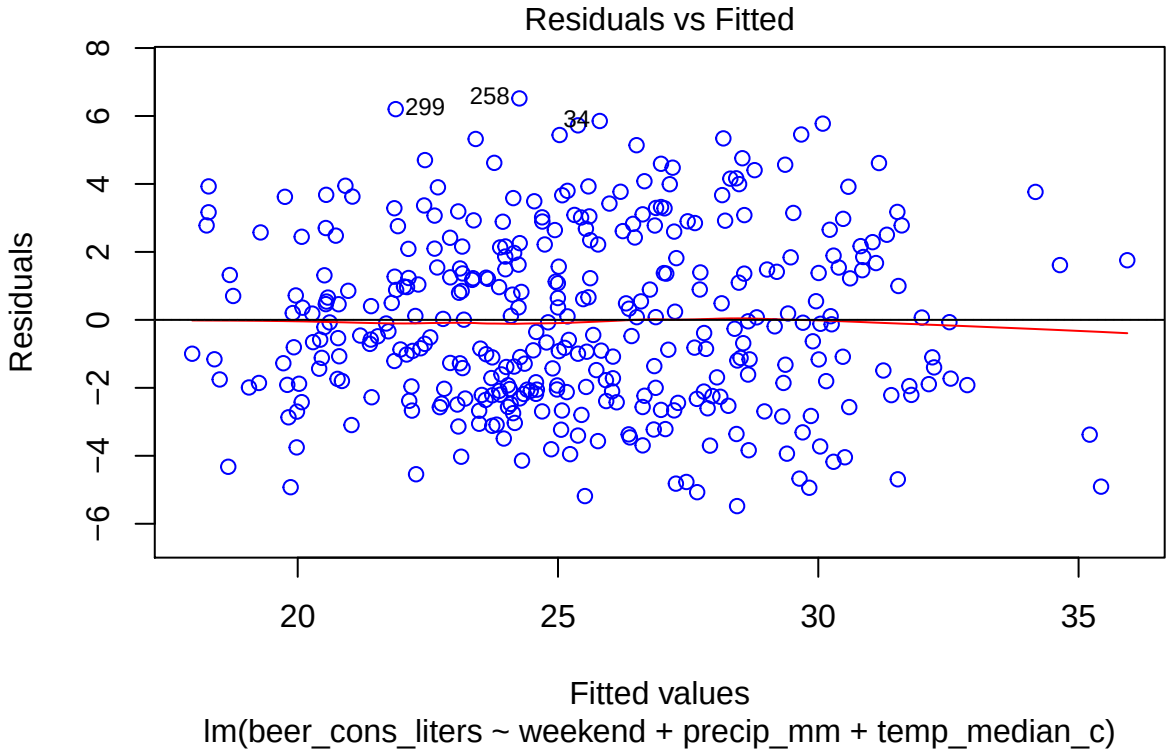



```
plot(y = lm_beer$residual, x = beer[1:365,]$temp_median_c, xlab = "temp_median_c",
     ylab = "Residual", main = "Linearity Test temp_median_c", col = 'blue')
```

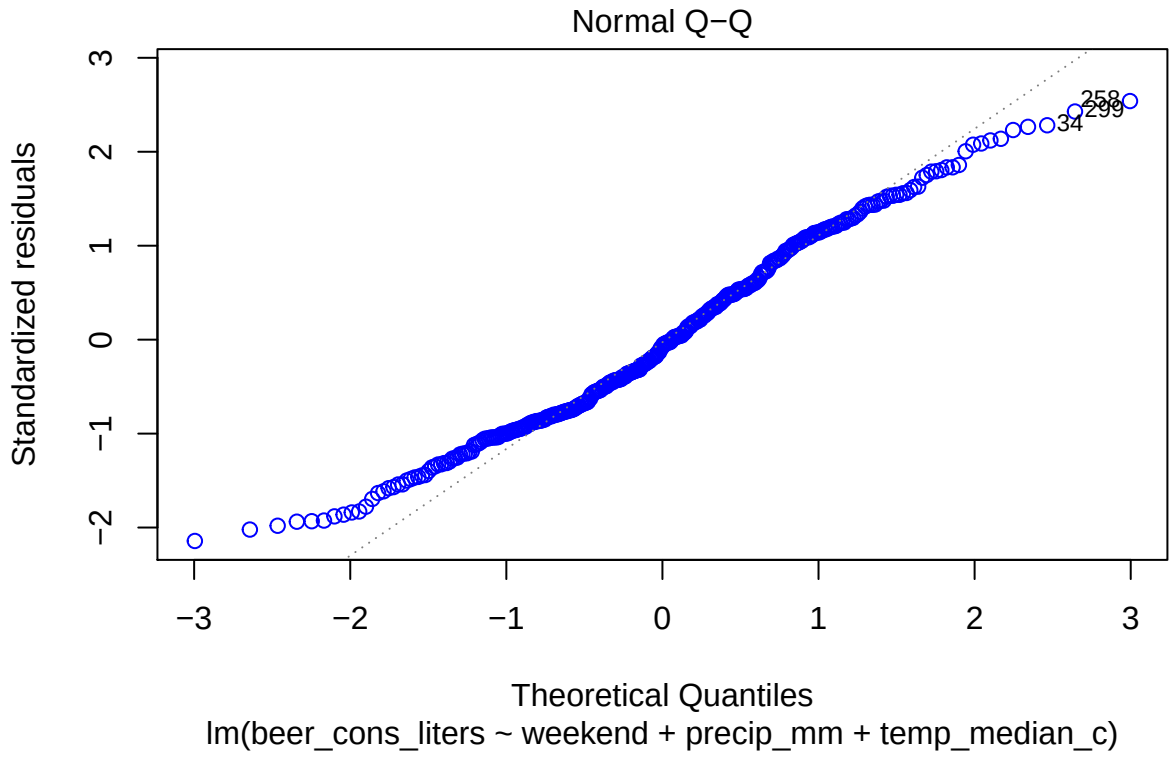
```
abline(0,0)
```




```
plot(lm_beer, which = 1, col = 'blue')
abline(0,0)
```



```r
plot(lm_beer, which = 2, col = 'blue')
```



Model:

$$\hat{beer\_cons\_liters}_i = \hat{\beta}_0 + \hat{\beta}_1 \cdot weekend_i + \hat{\beta}_2 \cdot precip\_mm_i + \hat{\beta}_3 \cdot temp\_median\_c_i$$

where:

$$\hat{\beta}_0 = 6.47348$$
$$\hat{\beta}_1 = 5.22787$$
$$\hat{\beta}_2 = -0.07420$$
$$\hat{\beta}_3 = 0.83971$$

All four coefficients have a p-value less than 0.05, which means we can reject the null hypothesis of them being equal to zero. beta_0 indicates that when weekend = 0, precip_mm = 0, and temp_median_c = 0, we can expect beer_cons_liters to be 6.47348. For values of weekend = 1, beer_cons_liters will increase by 5.22787. For every unit increase in precip_mm, beer_cons_liters will decrease by 0.07420 liters. Finally, for every unit increase in temp_median_c, beer_cons_liters will increase by 0.83971 liters. This model explains R^2 = 0.6584 or 65.84% of the variability in beer_cons_liters. Most models assumptions seem to be met (the linearity plot for precip_mm has most points to the left side, which may be explained by the fact that most geographical locations have a value of 0 for this variable, but otherwise the points seem to be equally spread along the y-axis). The normality Q-Q plot seems to depart from the diagonal at the tails, which may raise some concerns.

Exercise 5

Which of the variables appears to be the best covariate for explaining or predicting beer consumption? Why?
The variable temp_median_c has the largest t-value, which means it has the strongest linear association with the response variable, followed by weekend. temp_median_c appears to be the best covariate for explaining or predicting beer consumption.

Exercise 6
Are there any potential limitations of the model you have fit? If yes, what are two potential limitations?
First, it cannot be guaranteed that the model assumptions were met (see Exercise 4) as the Q-Q plot seems like the data is violating the normality assumption, so we cannot be certain this is a good model. Second, this model only explains association or correlation between the predictor variables and cannot be used as proof of causation or to extrapolate predictions outside the dataset range.

Exercise 7
Compute the in-sample root mean squared error (RMSE) for the regression model in question 4. Refer back to the class notes for details on how to compute in-sample (or within-sample) RMSE.

```
RMSE <- sqrt(sum(residuals(lm_beer) ^ 2) / df.residual(lm_beer)); RMSE
```

```
## [1] 2.571286
```

In-sample RMSE: $RMSE = 2.571286$

Exercise 8
Write a code for doing k-fold cross validation. Refer back to the class notes for details on k k-fold cross validation. Let k=10 and use average RMSE as the metric for quantifying predictive error. What is the average RMSE for the model in question 4 above?

```
# First set a seed to ensure your results are reproducible
set.seed(1) # use whatever number you want
# Now randomly re-shuffle the data
beer <- beer[1:365,]
beer <- beer[sample(nrow(beer)),]
# Define the number of folds you want
K <- 10
# Define a matrix to save your results into
RSME <- matrix(0, nrow = K, ncol = 1)
# Split the row indexes into k equal parts
kth_fold <- cut(seq(1, nrow(beer)), breaks = K, labels = FALSE)
# Now write the for loop for the k-fold cross validation
for(k in 1:K){
  # Split your data into the training and test datasets
  test_index <- which(kth_fold == k)
  train <- beer[-test_index,]
  test <- beer[test_index,]
  lm_train <- lm(beer_cons_liters ~ weekend + precip_mm + temp_median_c, train)
  predicted_test_values <- predict(lm_train, test)
  RSME[k,] <- sqrt(mean((test$beer_cons_liters - predicted_test_values) ^ 2))
  # You should consider using your code for question 7 above
}
mean(RSME) #Calculate the average of all values in the RSME matrix here.
```

```
## [1] 2.580269
```

Average MSE:

$$Avg.MSE = \frac{1}{10}\sum_{k=1}^{10} MSE_{test}^{(k)} = 2.583551$$

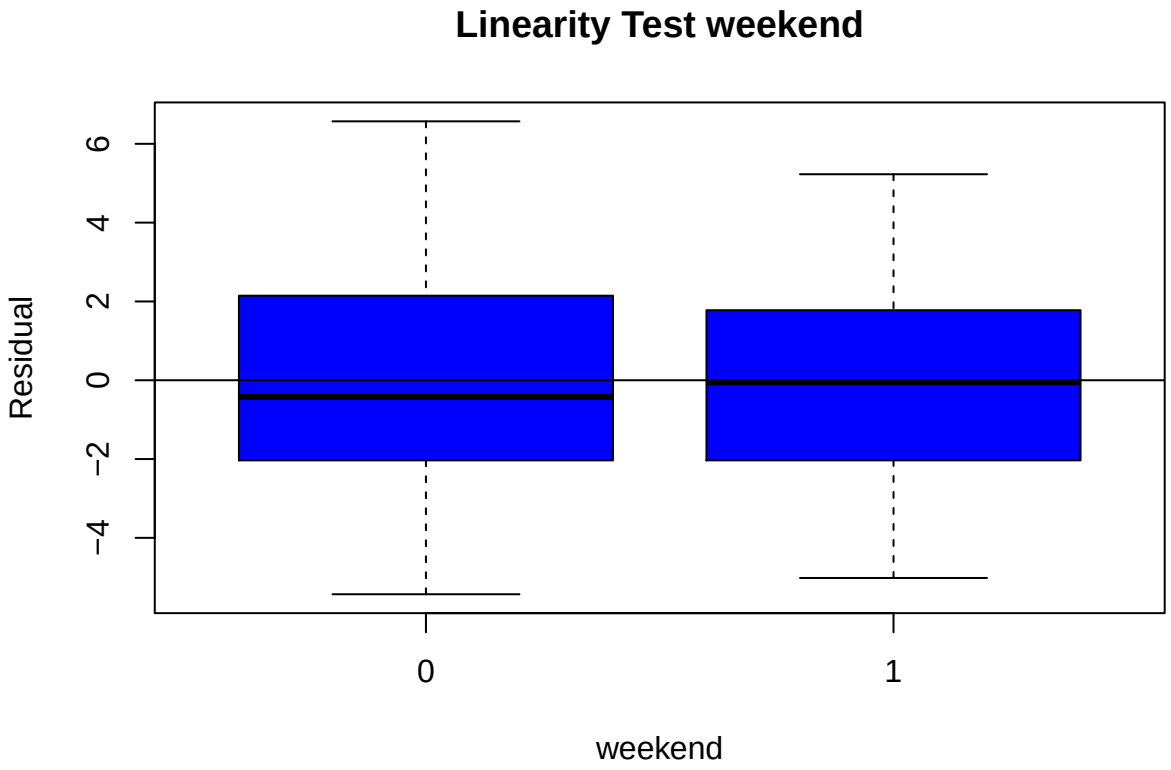
Exercise 9
Extend the model in question 4 to include interaction terms between weekend and the other two predictors.

Are the interaction terms significant?

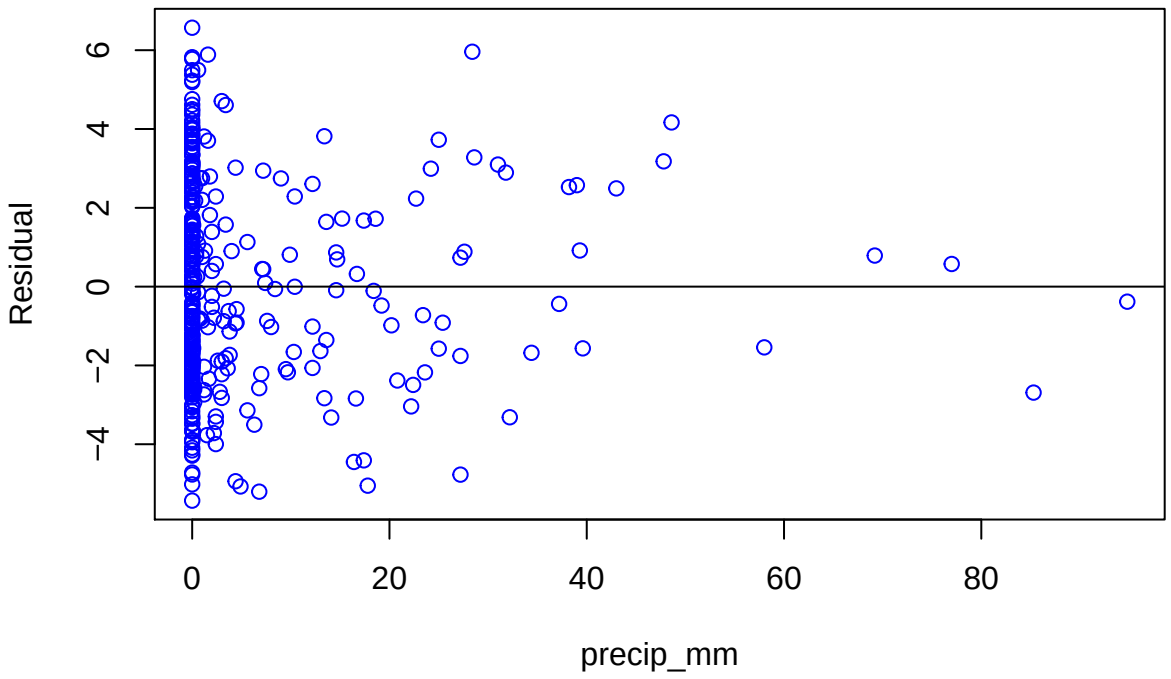
```
lm_beer2 <- lm(beer_cons_liters ~ weekend + precip_mm
               + temp_median_c + precip_mm:weekend + temp_median_c:weekend, beer)
summary(lm_beer2)
```

```
##
## Call:
## lm(formula = beer_cons_liters ~ weekend + precip_mm + temp_median_c +
##     precip_mm:weekend + temp_median_c:weekend, data = beer)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4311 -2.0341 -0.1772  2.0234  6.5705
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              6.398984   1.087629   5.883 9.21e-09 ***
## weekend1                 5.419563   1.993222   2.719  0.00687 **
## precip_mm               -0.063824   0.013217  -4.829 2.04e-06 ***
## temp_median_c            0.840677   0.050397  16.681  < 2e-16 ***
## weekend1:precip_mm      -0.031905   0.023184  -1.376  0.16962
## weekend1:temp_median_c -0.001192   0.093533  -0.013  0.98984
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.572 on 359 degrees of freedom
## Multiple R-squared:  0.663,  Adjusted R-squared:  0.6583
## F-statistic: 141.2 on 5 and 359 DF,  p-value: < 2.2e-16
```

```
plot(y = lm_beer2$residual, x = beer[1:365,]$weekend, xlab = "weekend", ylab = "Residual",
     main = "Linearity Test weekend", col = 'blue')
abline(0,0)
```

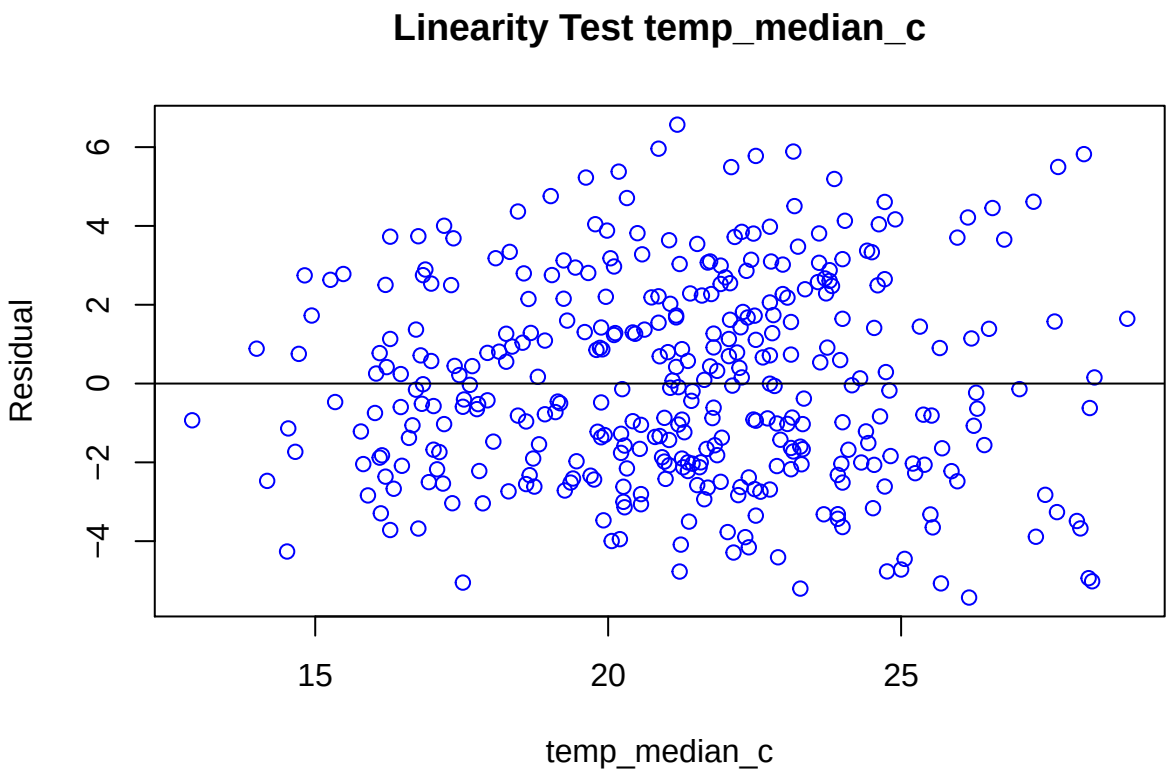



```
plot(y = lm_beer2$residual, x = beer[1:365,]$precip_mm, xlab = "precip_mm",
     ylab = "Residual", main = "Linearity Test precip_mm", col = 'blue')
abline(0,0)
```

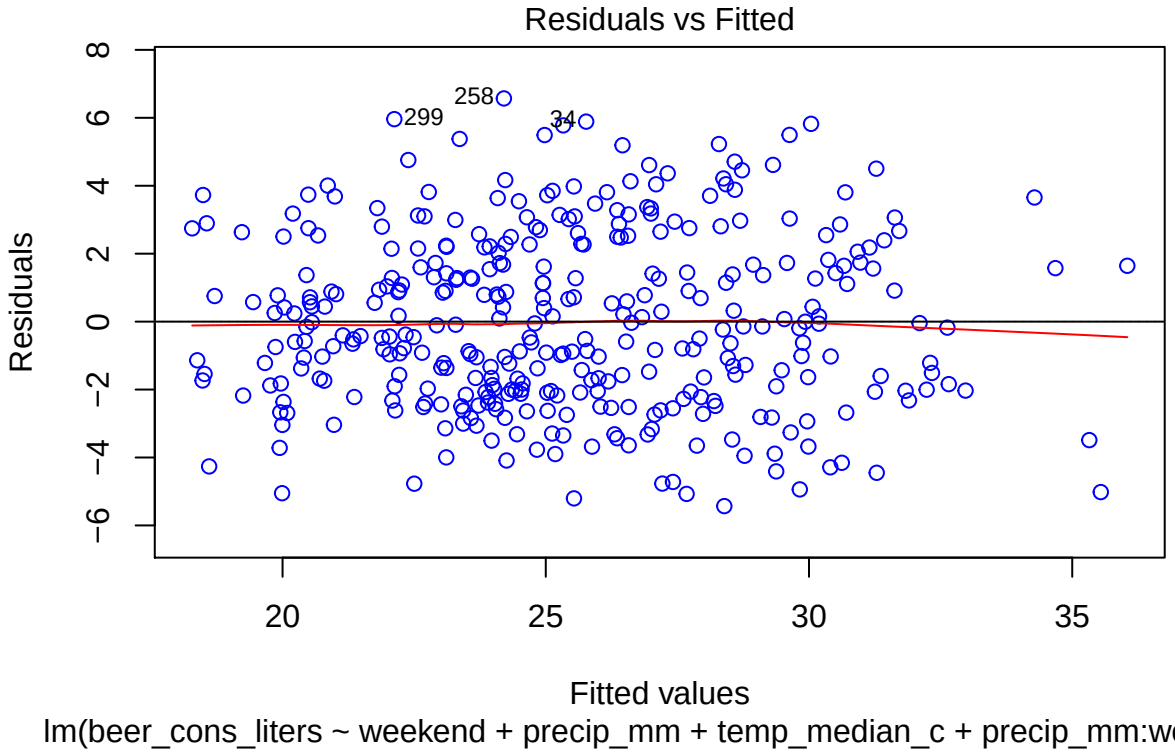



```
plot(y = lm_beer2$residual, x = beer[1:365,]$temp_median_c, xlab = "temp_median_c",
     ylab = "Residual", main = "Linearity Test temp_median_c", col = 'blue')
```

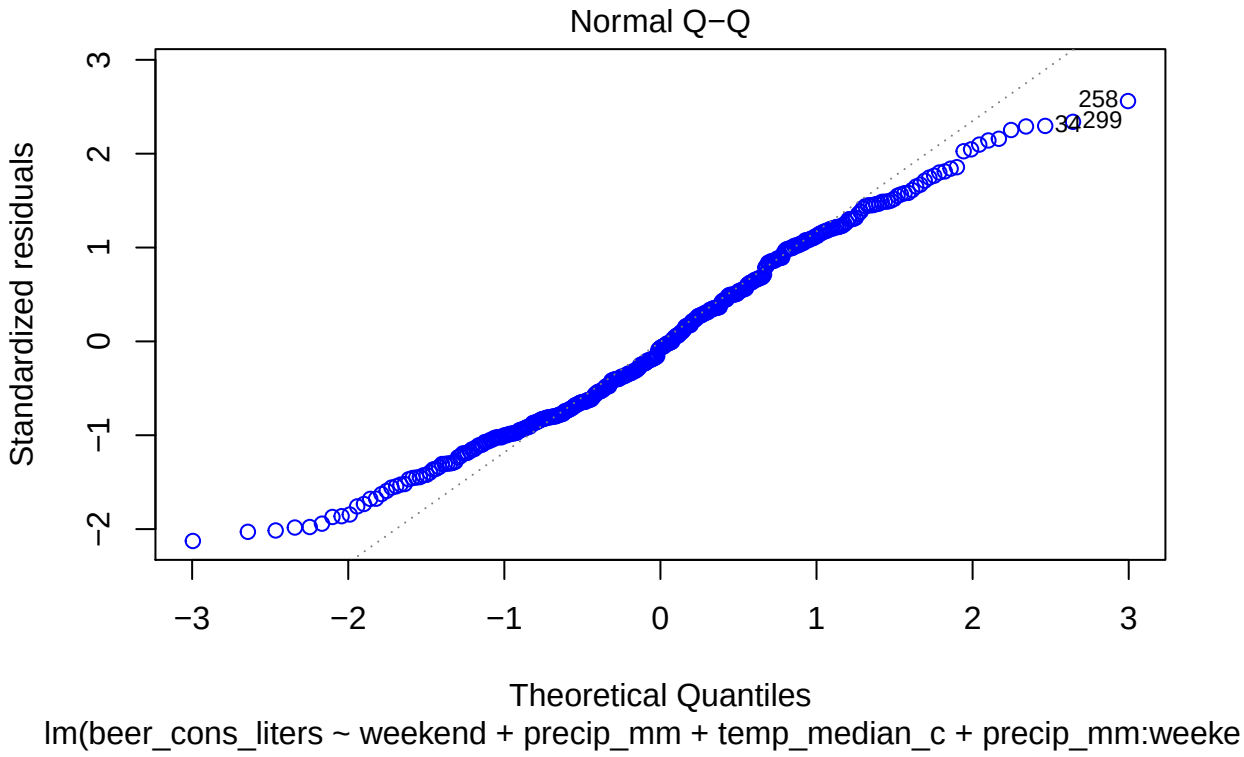
```
abline(0,0)
```




```
plot(lm_beer2, which = 1, col = 'blue')
abline(0,0)
```



```r
plot(lm_beer2, which = 2, col = 'blue')
```



The interaction terms do not seem to be significant. We cannot reject the null hypotheses that they are equal to zero as their p-values are greater than 0.05 in both cases (0.16962 for weekend:precip_mm, and 0.98984$ for weekend:temp_median_c). The Q-Q plot suggests the normality assumption is not met.

Exercise 10

Use your code for the k-fold cross validation from question 8 to compute the average RMSE for the new model in question 9. Is the new RMSE model lower or higher? What can you infer from that?

```r
# First set a seed to ensure your results are reproducible
set.seed(1) # use whatever number you want
# Now randomly re-shuffle the data
beer <- beer[1:365,]
beer <- beer[sample(nrow(beer)),]
# Define the number of folds you want
K <- 10
# Define a matrix to save your results into
RSME <- matrix(0, nrow = K, ncol = 1)
# Split the row indexes into k equal parts
kth_fold <- cut(seq(1, nrow(beer)), breaks = K, labels = FALSE)
# Now write the for loop for the k-fold cross validation
for(k in 1:K){
  # Split your data into the training and test datasets
  test_index <- which(kth_fold == k)
  train <- beer[-test_index,]
  test <- beer[test_index,]
  lm_train <- lm(beer_cons_liters ~ weekend + precip_mm + temp_median_c
                 + precip_mm:weekend + temp_median_c:weekend, train)
  predicted_test_values <- predict(lm_train, test)
```

```
  RSME[k,] <- sqrt(mean((test$beer_cons_liters - predicted_test_values) ^ 2))
  # You should consider using your code for question 7 above
}
mean(RSME) #Calculate the average of all values in the RSME matrix here.
```

```
## [1] 2.582553
```

Average MSE:

$$Avg.MSE = \frac{1}{10}\sum_{k=1}^{10} MSE_{test}^{(k)} = 2.584816$$

The new average MSE value is slightly higher than the previous one (by 0.0490%), which suggests this new model is slightly worse than the previous one.