Lab One: Cross Validation

Derek Wales

06SEP19

Lab report

Load data here

```
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)]
```

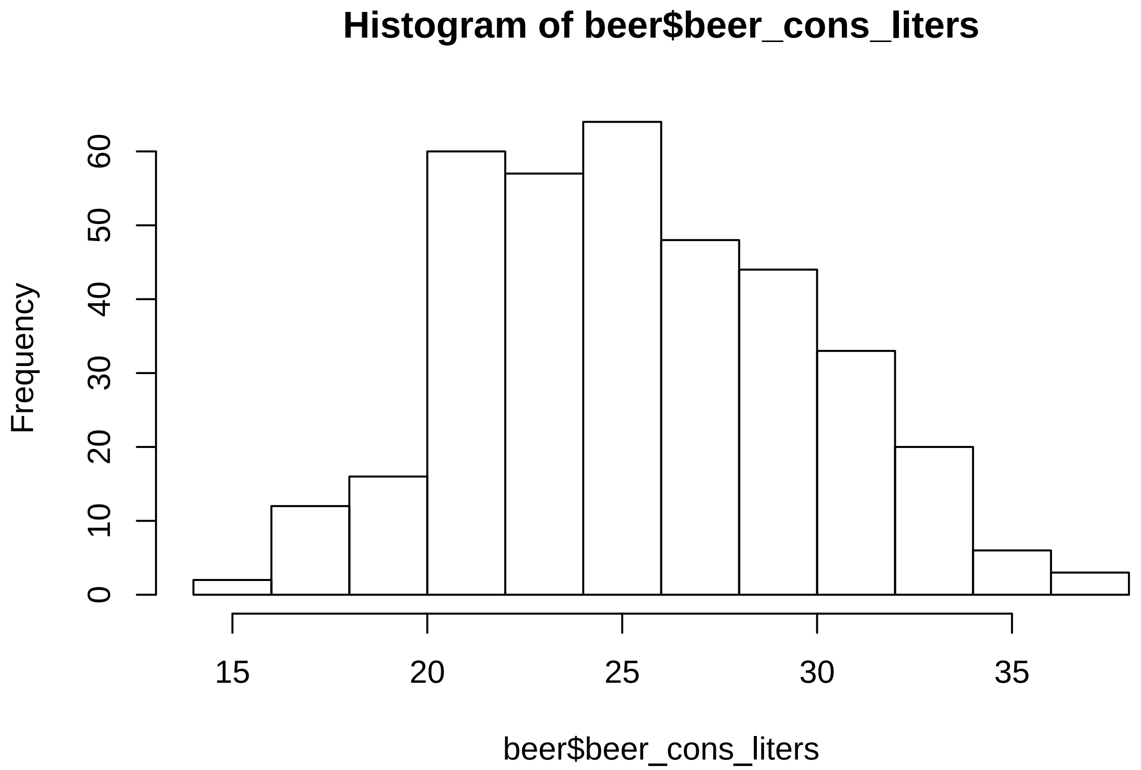
Question 1: Make a histogram of beer__cons__liters. Describe the distribution. Is the normality assumption

a plausible one here? If you think the histogram does not look normal enough,

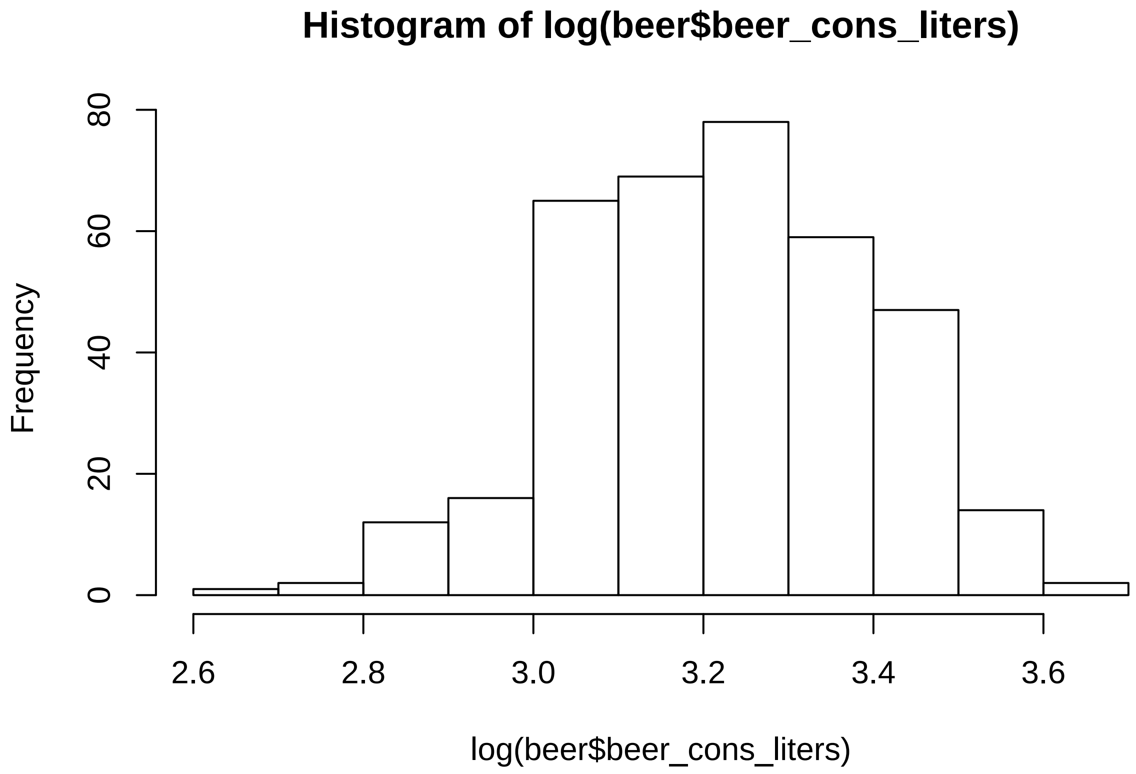
make a histogram of log(beer__cons__liters). Does that look more normal than beer__cons__liters?

The log distribution looks closer to a normal distribution than the unscaled.

```
hist(beer$beer_cons_liters)
```



```
hist(log(beer$beer_cons_liters))
```



Question 2: Make exploratory plots of beer__cons__liters (or log(beer__cons__liters)) versus each potential

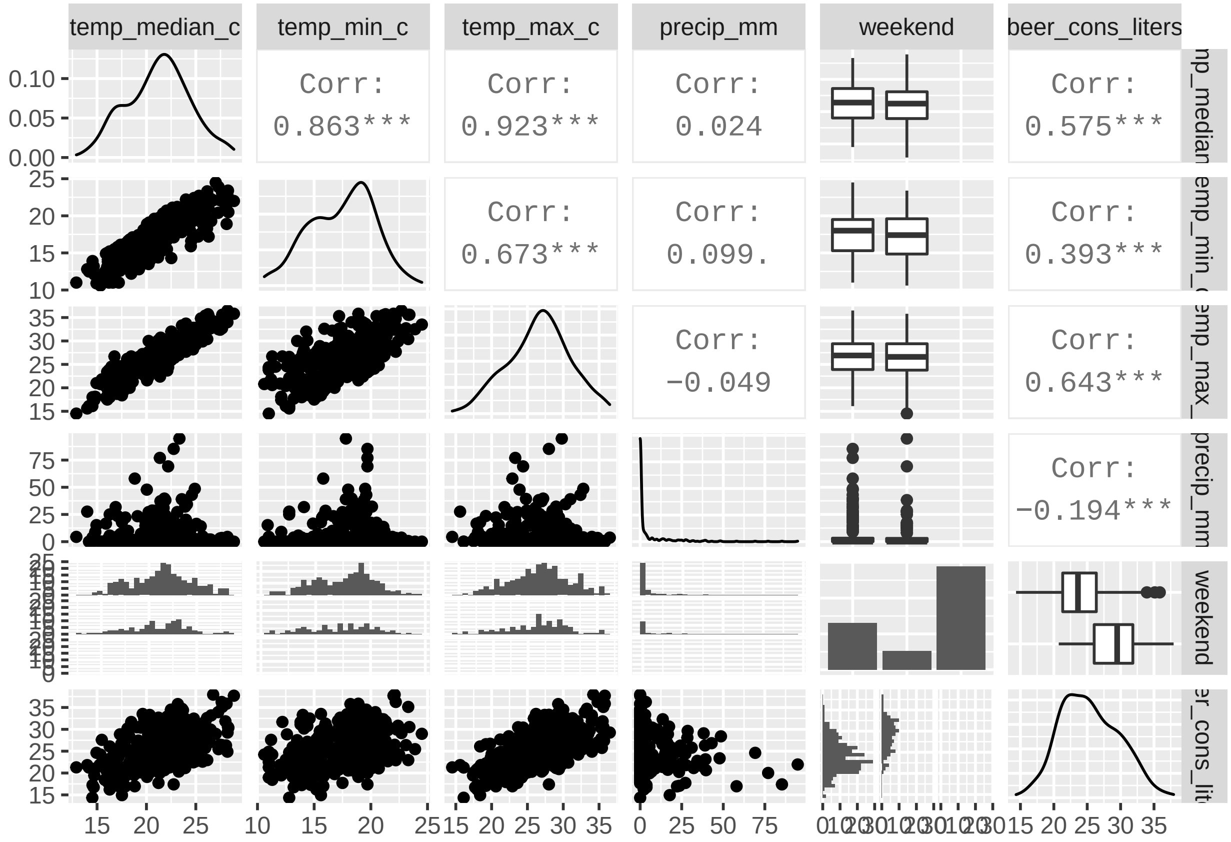
predictor. Are all the relationships linear? If any one of them is nonlinear, describe

the distribution.

The most meaningful predictors for Beer consumption in San Paulo seem to be rising temperatures and whether or not its a weekend.

```
ggpairs(beer,columns = 2:7)
```

```
## Warning: Removed 576 rows containing non-finite values (stat_density).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm =
## na.rm, : Removed 576 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm =
## na.rm, : Removed 576 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm =
## na.rm, : Removed 576 rows containing missing values

## Warning: Removed 576 rows containing non-finite values (stat_boxplot).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm =
## na.rm, : Removed 576 rows containing missing values

## Warning: Removed 576 rows containing missing values (geom_point).
## Warning: Removed 576 rows containing non-finite values (stat_density).
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm =
## na.rm, : Removed 576 rows containing missing values
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm =
## na.rm, : Removed 576 rows containing missing values
## Warning: Removed 576 rows containing non-finite values (stat_boxplot).
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm =
## na.rm, : Removed 576 rows containing missing values
## Warning: Removed 576 rows containing missing values (geom_point).
## Warning: Removed 576 rows containing missing values (geom_point).
## Warning: Removed 576 rows containing non-finite values (stat_density).
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm =
## na.rm, : Removed 576 rows containing missing values
## Warning: Removed 576 rows containing non-finite values (stat_boxplot).
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm =
## na.rm, : Removed 576 rows containing missing values
## Warning: Removed 576 rows containing missing values (geom_point).
## Warning: Removed 576 rows containing missing values (geom_point).
## Warning: Removed 576 rows containing missing values (geom_point).
## Warning: Removed 576 rows containing non-finite values (stat_density).
## Warning: Removed 576 rows containing non-finite values (stat_boxplot).
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm =
## na.rm, : Removed 576 rows containing missing values
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 576 rows containing non-finite values (stat_bin).
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 576 rows containing non-finite values (stat_bin).
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 576 rows containing non-finite values (stat_bin).
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 576 rows containing non-finite values (stat_bin).
## Warning: Removed 576 rows containing missing values (stat_boxplot).
## Warning: Removed 576 rows containing missing values (geom_point).
## Warning: Removed 576 rows containing missing values (geom_point).
## Warning: Removed 576 rows containing missing values (geom_point).
## Warning: Removed 576 rows containing missing values (geom_point).
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 576 rows containing non-finite values (stat_bin).

## Warning: Removed 576 rows containing non-finite values (stat_density).
```



Question 3: Does it make sense to include all three of temp_median_c, temp_min_c and temp_max_c as

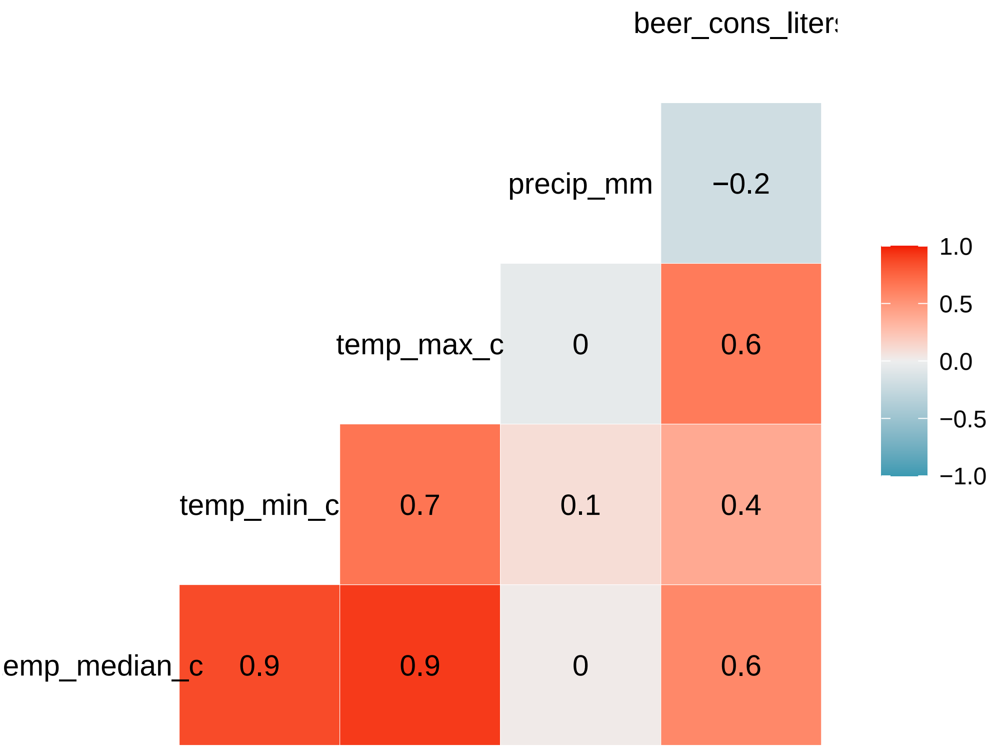
predictors in a MLR model for predicting beer_cons_liters (or log(beer_cons_liters))? Justify

your response in one or two sentences.

No because all of these are correlated which will violate one of the assumptions for using a Linear Model.

```r
ggcorr(beer, label = TRUE)
```

```
## Warning in ggcorr(beer, label = TRUE): data in column(s) 'date', 'weekend'
## are not numeric and were ignored
```



Question 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?

The Adjusted R-Squared value is 0.6554 which means that our model matches real life 65.54% percent of the time.

```
lm_beer_consumption <- lm(log(beer_cons_liters) ~ weekend + precip_mm + temp_median_c, data = beer)
summary(lm_beer_consumption)
```

```
##
## Call:
## lm(formula = log(beer_cons_liters) ~ weekend + precip_mm + temp_median_c,
##     data = beer)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.295118 -0.078081 -0.003897  0.074038  0.255047
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.4609607  0.0366579  67.133  < 2e-16 ***
## weekend1       0.2022884  0.0119016  16.997  < 2e-16 ***
## precip_mm     -0.0029986  0.0004328  -6.929 1.96e-11 ***
## temp_median_c  0.0337657  0.0016921  19.955  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1025 on 361 degrees of freedom
##   (576 observations deleted due to missingness)
## Multiple R-squared:  0.6583, Adjusted R-squared:  0.6554
## F-statistic: 231.8 on 3 and 361 DF,  p-value: < 2.2e-16
```

Question 5: Which of the variables appears to be the best covariate for explaining or predicting beer

consumption? Why?

The variable with the highest t value (aka varies with the results) is the temp_median_c.

```r
# Enter your code for question 5 here
```

Question 6: Are there any potential limitations of the model you have fit? If yes, what are two

potential limitations?

It is not a time series, it does not account for Temperature and percipitation which are often effected by the previous day. Additionally it doesn't account for holidays.

```r
# Enter your code for question 6 here
```

Question 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.

See response below.

```r
y_hat <- exp(predict.lm(lm_beer_consumption))
y <- na.omit(beer$beer_cons_liters)
RMSE <- (sqrt((1/length(y))*(sum((y-y_hat)^2))))
print(RMSE)
```

```
## [1] 2.59066
```

Question 8: Write a code for doing k-fold cross validation. Refer back to the class notes for details on

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?

The new RMSE is 2.562191.

```r
# Suppose your data is stored in the object "Data"
# First set a seed to ensure your results are reproducible
set.seed(123) # use whatever number you want
# Now randomly re-shuffle the data
Data <- 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(Data)),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 <- Data[-test_index,]
  test <- Data[test_index,]
  lm2 = lm(beer_cons_liters ~ weekend + precip_mm + temp_median_c, data = train, na.action = na.omit)
  pred1 = predict(lm2, test)
  # Now that you've split the data,
  RSME[k,] <- mean((test$beer_cons_liters - pred1)^2, na.rm = T)^(1/2)
  # You should consider using your code for question 7 above
}
#Calculate the average of all values in the RSME matrix here.
mean(RSME)
```

```
## [1] 2.562191
```

Question 9: Extend the model in question 4 to include interaction terms between weekend and the

other two predictors. Are the interaction terms significant?

The p values were not significant. Additionally, it did not effect the R squared.

```
lm_beer_consumption_2 <- lm(log(beer_cons_liters) ~ weekend + precip_mm + temp_median_c + weekend:precip
summary(lm_beer_consumption_2)
```

```
##
## Call:
## lm(formula = log(beer_cons_liters) ~ weekend + precip_mm + temp_median_c +
##     weekend:precip_mm + weekend:temp_median_c, data = beer)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29210 -0.07896 -0.00836  0.07904  0.25207
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.4258223  0.0433032  56.019  < 2e-16 ***
## weekend1               0.3202883  0.0793588   4.036 6.65e-05 ***
## precip_mm             -0.0027580  0.0005262  -5.241 2.73e-07 ***
## temp_median_c          0.0353547  0.0020065  17.620  < 2e-16 ***
## weekend1:precip_mm    -0.0007148  0.0009230  -0.774    0.439
## weekend1:temp_median_c -0.0054221  0.0037240  -1.456    0.146
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1024 on 359 degrees of freedom
##   (576 observations deleted due to missingness)
## Multiple R-squared:  0.6609, Adjusted R-squared:  0.6562
## F-statistic:   140 on 5 and 359 DF,  p-value: < 2.2e-16
```

Question 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?

2.559797 for Question 10 vs 2.562191 for Question 8, so it does make the model sightly more accurate but it is not meaningful.

```r
# Suppose your data is stored in the object "Data"
# First set a seed to ensure your results are reproducible
set.seed(123) # use whatever number you want
# Now randomly re-shuffle the data
Data <- 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(Data)),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 <- Data[-test_index,]
  test <- Data[test_index,]
  lm2 = lm(beer_cons_liters ~ weekend + precip_mm + temp_median_c + weekend:precip_mm + weekend:temp_med
  pred1 = predict(lm2, test)
  # Now that you've split the data,
  RSME[k,] <- mean((test$beer_cons_liters - pred1)^2, na.rm = T)^(1/2)
  # You should consider using your code for question 7 above
}
#Calculate the average of all values in the RSME matrix here.
mean(RSME)
```

```
## [1] 2.559797
```