### Model Validation Assignment

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.7  
## v tidyr 0.8.2 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.3.0

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

Read- in dataset

bike = read\_csv("hour.csv")

## Parsed with column specification:  
## cols(  
## instant = col\_integer(),  
## dteday = col\_character(),  
## season = col\_integer(),  
## yr = col\_integer(),  
## mnth = col\_integer(),  
## hr = col\_integer(),  
## holiday = col\_integer(),  
## weekday = col\_integer(),  
## workingday = col\_integer(),  
## weathersit = col\_integer(),  
## temp = col\_double(),  
## atemp = col\_double(),  
## hum = col\_double(),  
## windspeed = col\_double(),  
## casual = col\_integer(),  
## registered = col\_integer(),  
## count = col\_integer()  
## )

bike = bike %>% drop\_na()   
str(bike)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 17379 obs. of 17 variables:  
## $ instant : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ dteday : chr "1/1/2011" "1/1/2011" "1/1/2011" "1/1/2011" ...  
## $ season : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ yr : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ mnth : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ hr : int 0 1 2 3 4 5 6 7 8 9 ...  
## $ holiday : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ weekday : int 6 6 6 6 6 6 6 6 6 6 ...  
## $ workingday: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ weathersit: int 1 1 1 1 1 2 1 1 1 1 ...  
## $ temp : num 0.24 0.22 0.22 0.24 0.24 0.24 0.22 0.2 0.24 0.32 ...  
## $ atemp : num 0.288 0.273 0.273 0.288 0.288 ...  
## $ hum : num 0.81 0.8 0.8 0.75 0.75 0.75 0.8 0.86 0.75 0.76 ...  
## $ windspeed : num 0 0 0 0 0 0.0896 0 0 0 0 ...  
## $ casual : int 3 8 5 3 0 0 2 1 1 8 ...  
## $ registered: int 13 32 27 10 1 1 0 2 7 6 ...  
## $ count : int 16 40 32 13 1 1 2 3 8 14 ...

Use the same code from Module 2 Assignment 1, convert variables

bike = bike %>% mutate(season = as\_factor(as.character(season))) %>%  
mutate(season = fct\_recode(season,  
"Spring" = "1",  
"Summer" = "2",  
"Fall" = "3",  
"Winter" = "4"))  
  
bike = bike %>% mutate(yr = as\_factor(as.character(yr)))  
bike = bike %>% mutate(mnth = as\_factor(as.character(mnth)))  
bike = bike %>% mutate(hr = as\_factor(as.character(hr)))  
  
bike = bike %>% mutate(holiday = as\_factor(as.character(holiday))) %>%  
mutate(holiday = fct\_recode(holiday,  
"NotHoliday" = "0",  
" Holiday" = "1"))  
  
bike = bike %>% mutate(workingday = as\_factor(as.character(workingday))) %>%  
mutate(workingday = fct\_recode(workingday,  
"NotWorkingDay" = "0",  
" Workingday" = "1"))  
  
bike = bike %>% mutate(weathersit = as\_factor(as.character(weathersit))) %>%  
mutate(weathersit = fct\_recode(weathersit,  
"NoPrecip" = "1",  
"Misty" = "2",  
"LightPrecip" = "3",  
"HeavyPrecip" = "4"))  
  
bike = bike %>% mutate(weekday = as\_factor(as.character(weekday))) %>%  
mutate(weekday = fct\_recode(weekday,  
"Sunday" = "0",  
"Monday" = "1",  
"Tuesday" = "2",  
"Wednesday" = "3",  
"Thursday" = "4",  
"Friday" = "5",  
"Saturday" = "6"))

# Task 1

set.seed(1234)  
train.row = createDataPartition(y= bike$hr, p=0.7, list = FALSE)   
train = bike[train.row,]  
test = bike[-train.row,]

# Task 2

Training = 12,173 observations (rows) in the dataset

Testing = 5,206 observations (rows) in the dataset

# Task 3

allmod2 = lm(count ~ season + mnth + hr + holiday + weekday + temp + weathersit, train)  
summary(allmod2)

##   
## Call:  
## lm(formula = count ~ season + mnth + hr + holiday + weekday +   
## temp + weathersit, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -417.22 -62.65 -9.94 51.99 499.15   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -84.1865 6.9931 -12.039 < 2e-16 \*\*\*  
## seasonSummer 42.4128 6.4425 6.583 4.79e-11 \*\*\*  
## seasonFall 23.9324 7.5395 3.174 0.001506 \*\*   
## seasonWinter 61.5058 6.3884 9.628 < 2e-16 \*\*\*  
## mnth2 -0.2310 5.1586 -0.045 0.964280   
## mnth3 4.0991 5.8417 0.702 0.482880   
## mnth4 -12.0175 8.6493 -1.389 0.164734   
## mnth5 -11.9380 9.2574 -1.290 0.197224   
## mnth6 -23.6688 9.4495 -2.505 0.012266 \*   
## mnth7 -43.3200 10.5833 -4.093 4.28e-05 \*\*\*  
## mnth8 -23.8876 10.2690 -2.326 0.020026 \*   
## mnth9 1.9119 9.1307 0.209 0.834149   
## mnth10 -0.9806 8.4727 -0.116 0.907860   
## mnth11 -17.3828 8.1577 -2.131 0.033122 \*   
## mnth12 -16.1398 6.4900 -2.487 0.012901 \*   
## hr1 -20.8260 7.0129 -2.970 0.002987 \*\*   
## hr2 -28.5894 7.0331 -4.065 4.83e-05 \*\*\*  
## hr3 -40.8110 7.0827 -5.762 8.51e-09 \*\*\*  
## hr4 -41.9050 7.0861 -5.914 3.44e-09 \*\*\*  
## hr5 -25.7089 7.0407 -3.651 0.000262 \*\*\*  
## hr6 30.0505 7.0199 4.281 1.88e-05 \*\*\*  
## hr7 166.3557 7.0105 23.730 < 2e-16 \*\*\*  
## hr8 311.5327 7.0040 44.479 < 2e-16 \*\*\*  
## hr9 164.2248 7.0051 23.443 < 2e-16 \*\*\*  
## hr10 114.0057 7.0141 16.254 < 2e-16 \*\*\*  
## hr11 136.4484 7.0346 19.397 < 2e-16 \*\*\*  
## hr12 179.1662 7.0505 25.412 < 2e-16 \*\*\*  
## hr13 172.4599 7.0831 24.348 < 2e-16 \*\*\*  
## hr14 153.0996 7.0908 21.591 < 2e-16 \*\*\*  
## hr15 167.7927 7.0997 23.634 < 2e-16 \*\*\*  
## hr16 225.0846 7.0975 31.713 < 2e-16 \*\*\*  
## hr17 377.2997 7.0812 53.282 < 2e-16 \*\*\*  
## hr18 355.1635 7.0498 50.379 < 2e-16 \*\*\*  
## hr19 238.8482 7.0302 33.974 < 2e-16 \*\*\*  
## hr20 160.6658 7.0183 22.892 < 2e-16 \*\*\*  
## hr21 106.9804 7.0093 15.263 < 2e-16 \*\*\*  
## hr22 71.1685 7.0013 10.165 < 2e-16 \*\*\*  
## hr23 32.9363 6.9989 4.706 2.55e-06 \*\*\*  
## holiday Holiday -25.7196 6.3098 -4.076 4.61e-05 \*\*\*  
## weekdaySunday -16.9637 3.7619 -4.509 6.56e-06 \*\*\*  
## weekdayMonday -11.2807 3.9086 -2.886 0.003907 \*\*   
## weekdayTuesday -7.7171 3.8125 -2.024 0.042978 \*   
## weekdayWednesday -3.9687 3.7917 -1.047 0.295261   
## weekdayThursday -3.0008 3.8002 -0.790 0.429745   
## weekdayFriday 1.6326 3.7772 0.432 0.665578   
## temp 295.6620 12.2370 24.161 < 2e-16 \*\*\*  
## weathersitMisty -19.3402 2.3791 -8.129 4.74e-16 \*\*\*  
## weathersitLightPrecip -92.5690 3.7639 -24.594 < 2e-16 \*\*\*  
## weathersitHeavyPrecip 86.6199 111.8612 0.774 0.438738   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 111.7 on 12124 degrees of freedom  
## Multiple R-squared: 0.6235, Adjusted R-squared: 0.622   
## F-statistic: 418.2 on 48 and 12124 DF, p-value: < 2.2e-16

In the above code the adjusted Rsquared is 0.622.

# Task 4

predict\_train = predict(allmod2, newdata = train)  
head(predict\_train,6)

## 1 2 3 4 5 6   
## -13.22764 -39.96691 -54.03862 -55.13261 -58.27672 10.90959

It appears that these predictions have a lower values. It appears that they are not correlated with other factors in the trianing dataset.

# Task 5

predict\_test = predict(allmod2, newdata = test)  
head(predict\_test,6)

## 1 2 3 4 5 6   
## -47.7303 141.3016 174.6501 142.1707 158.7002 403.8643

It appears that many of these numbers have high values, they seem to align with the original dataset information.

# Task 6

SSE = sum((test$count - predict\_test)^2)   
SST = sum((test$count - mean(test$count))^2)   
1- SSE/SST

## [1] 0.6240518

In the linear regression model the adjusted r squared was returned to have a similar value as above. The lm model produced a 0.622 while this manually calculated model produced a 0.624.

# Task 7

A K-fold is used to split the data. With a train test split the data is always trained around 70%-80%, where as the test is completed with 20%-30% of the data.The main difference between the two is with K-fold datasets the data is divided into subsets, those methods are repeated on all subsets. One of the subsets is for testing the others are for training. Then ensure validity within the K-fold test.