## Module 3 Model Validation Assignment 1

### Nicholas Barrett, 02/09/19

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)

## Warning: package 'caret' was built under R version 3.5.2

## Loading required package: lattice

##   
## Attaching package: 'caret'

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

library(GGally)

## Warning: package 'GGally' was built under R version 3.5.2

##   
## Attaching package: 'GGally'

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

bike <- read\_csv("hour.csv")

## Parsed with column specification:  
## cols(  
## instant = col\_integer(),  
## dteday = col\_date(format = ""),  
## 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 %>% 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)))  
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 : Date, format: "2011-01-01" "2011-01-01" ...  
## $ season : Factor w/ 4 levels "Spring","Summer",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ yr : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ mnth : Factor w/ 12 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ hr : Factor w/ 24 levels "0","1","2","3",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ 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 ...

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, "Saturday" = "6", "Sunday" = "0", "Monday" = "1", "Tuesday" = "2", "Wednesday" = "3", "Thursday" = "4", "Friday" = "5"))  
  
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 : Date, format: "2011-01-01" "2011-01-01" ...  
## $ season : Factor w/ 4 levels "Spring","Summer",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ yr : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ mnth : Factor w/ 12 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ hr : Factor w/ 24 levels "0","1","2","3",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ holiday : Factor w/ 2 levels "NotHoliday","Holiday": 1 1 1 1 1 1 1 1 1 1 ...  
## $ weekday : Factor w/ 7 levels "Saturday","Sunday",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ workingday: Factor w/ 2 levels "NotWorkingDay",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ weathersit: Factor w/ 4 levels "NoPrecip","Misty",..: 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 ...

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 : Date, format: "2011-01-01" "2011-01-01" ...  
## $ season : Factor w/ 4 levels "Spring","Summer",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ yr : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ mnth : Factor w/ 12 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ hr : Factor w/ 24 levels "0","1","2","3",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ holiday : Factor w/ 2 levels "NotHoliday","Holiday": 1 1 1 1 1 1 1 1 1 1 ...  
## $ weekday : Factor w/ 7 levels "Saturday","Sunday",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ workingday: Factor w/ 2 levels "NotWorkingDay",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ weathersit: Factor w/ 4 levels "NoPrecip","Misty",..: 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 ...

### Task 1 - Split the data into traininig and testing sets. Use random number of 1234

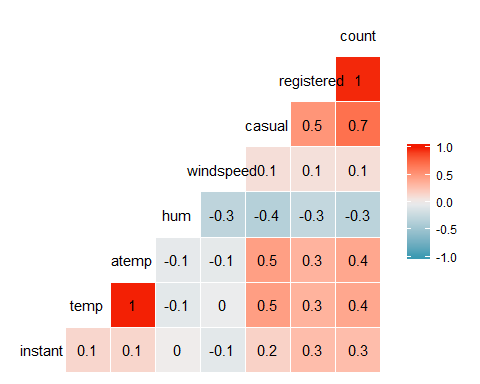
ctrl <- trainControl(method = "cv", number = 10)  
  
set.seed(1234)  
modCV <- train(count ~ casual, bike, method = "lm", trControl = ctrl, metric = "Rsquared")  
summary(modCV)

##   
## Call:  
## lm(formula = .outcome ~ ., data = dat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -387.85 -84.50 -41.07 31.81 668.50   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 98.30242 1.22189 80.45 <2e-16 \*\*\*  
## casual 2.55522 0.02008 127.27 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 130.5 on 17377 degrees of freedom  
## Multiple R-squared: 0.4824, Adjusted R-squared: 0.4824   
## F-statistic: 1.62e+04 on 1 and 17377 DF, p-value: < 2.2e-16

train.rows <- createDataPartition(y= bike$count, p= 0.7, list = FALSE)  
train <- bike[train.rows,]  
test <- bike[-train.rows,]

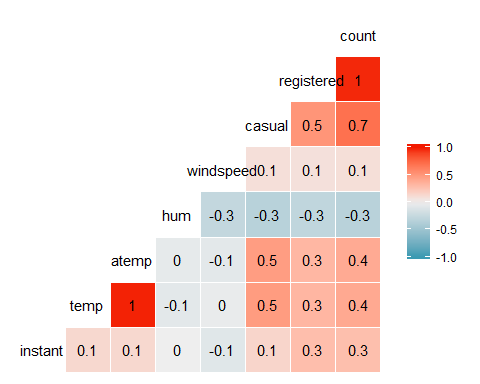
ggcorr(train, label = TRUE)

## Warning in ggcorr(train, label = TRUE): data in column(s) 'dteday',  
## 'season', 'yr', 'mnth', 'hr', 'holiday', 'weekday', 'workingday',  
## 'weathersit' are not numeric and were ignored



ggcorr(test, label = TRUE)

## Warning in ggcorr(test, label = TRUE): data in column(s) 'dteday',  
## 'season', 'yr', 'mnth', 'hr', 'holiday', 'weekday', 'workingday',  
## 'weathersit' are not numeric and were ignored



### Task 2 - How many rows of data are in each set - training and testing?

str(train)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 12167 obs. of 17 variables:  
## $ instant : int 1 4 6 9 10 11 12 13 15 16 ...  
## $ dteday : Date, format: "2011-01-01" "2011-01-01" ...  
## $ season : Factor w/ 4 levels "Spring","Summer",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ yr : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ mnth : Factor w/ 12 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ hr : Factor w/ 24 levels "0","1","2","3",..: 1 4 6 9 10 11 12 13 15 16 ...  
## $ holiday : Factor w/ 2 levels "NotHoliday","Holiday": 1 1 1 1 1 1 1 1 1 1 ...  
## $ weekday : Factor w/ 7 levels "Saturday","Sunday",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ workingday: Factor w/ 2 levels "NotWorkingDay",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ weathersit: Factor w/ 4 levels "NoPrecip","Misty",..: 1 1 2 1 1 1 1 1 2 2 ...  
## $ temp : num 0.24 0.24 0.24 0.24 0.32 0.38 0.36 0.42 0.46 0.44 ...  
## $ atemp : num 0.288 0.288 0.258 0.288 0.348 ...  
## $ hum : num 0.81 0.75 0.75 0.75 0.76 0.76 0.81 0.77 0.72 0.77 ...  
## $ windspeed : num 0 0 0.0896 0 0 ...  
## $ casual : int 3 3 0 1 8 12 26 29 35 40 ...  
## $ registered: int 13 10 1 7 6 24 30 55 71 70 ...  
## $ count : int 16 13 1 8 14 36 56 84 106 110 ...

str(test)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 5212 obs. of 17 variables:  
## $ instant : int 2 3 5 7 8 14 19 21 25 26 ...  
## $ dteday : Date, format: "2011-01-01" "2011-01-01" ...  
## $ season : Factor w/ 4 levels "Spring","Summer",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ yr : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ mnth : Factor w/ 12 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ hr : Factor w/ 24 levels "0","1","2","3",..: 2 3 5 7 8 14 19 21 1 2 ...  
## $ holiday : Factor w/ 2 levels "NotHoliday","Holiday": 1 1 1 1 1 1 1 1 1 1 ...  
## $ weekday : Factor w/ 7 levels "Saturday","Sunday",..: 1 1 1 1 1 1 1 1 2 2 ...  
## $ workingday: Factor w/ 2 levels "NotWorkingDay",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ weathersit: Factor w/ 4 levels "NoPrecip","Misty",..: 1 1 1 1 1 2 3 2 2 2 ...  
## $ temp : num 0.22 0.22 0.24 0.22 0.2 0.46 0.42 0.4 0.46 0.44 ...  
## $ atemp : num 0.273 0.273 0.288 0.273 0.258 ...  
## $ hum : num 0.8 0.8 0.75 0.8 0.86 0.72 0.88 0.87 0.88 0.94 ...  
## $ windspeed : num 0 0 0 0 0 ...  
## $ casual : int 8 5 0 2 1 47 9 11 4 1 ...  
## $ registered: int 32 27 1 0 2 47 26 25 13 16 ...  
## $ count : int 40 32 1 2 3 94 35 36 17 17 ...

The training set has 12,167 rows of data and 17 variables.

The testing set has 5,212 rows of data and 17 variables.

### Task 3 - Build a linear regression model (using the training set) to predict “count” using variables - season, mnth, hr, holiday, weekday, temp and weathersit. Comment on the quality of the model. Note the Adjusted R-squared value.

The model appears to be a good model. There are significant relationships in the variables in some of the levels within the variables. The overall linear model with these variables has a strong R-squared value of 0.6223

mod1 <- lm(count ~ season + mnth + hr + holiday + weekday + temp + weathersit, train)  
summary(mod1)

##   
## Call:  
## lm(formula = count ~ season + mnth + hr + holiday + weekday +   
## temp + weathersit, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -421.47 -62.42 -9.78 51.87 500.36   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -87.0705 6.9033 -12.613 < 2e-16 \*\*\*  
## seasonSummer 34.3246 6.3139 5.436 5.54e-08 \*\*\*  
## seasonFall 20.7690 7.4582 2.785 0.00537 \*\*   
## seasonWinter 66.8606 6.3498 10.530 < 2e-16 \*\*\*  
## mnth2 0.1953 5.1790 0.038 0.96992   
## mnth3 8.8250 5.7887 1.525 0.12740   
## mnth4 -3.9143 8.5485 -0.458 0.64704   
## mnth5 -3.1760 9.1526 -0.347 0.72860   
## mnth6 -12.1022 9.4004 -1.287 0.19797   
## mnth7 -32.3425 10.5194 -3.075 0.00211 \*\*   
## mnth8 -10.4347 10.2509 -1.018 0.30873   
## mnth9 10.1809 9.0944 1.119 0.26296   
## mnth10 -3.5025 8.4696 -0.414 0.67922   
## mnth11 -16.2326 8.1501 -1.992 0.04643 \*   
## mnth12 -16.1179 6.5034 -2.478 0.01321 \*   
## hr1 -16.1012 6.9804 -2.307 0.02109 \*   
## hr2 -27.4902 6.8954 -3.987 6.74e-05 \*\*\*  
## hr3 -37.1271 7.0389 -5.275 1.35e-07 \*\*\*  
## hr4 -40.7703 7.0862 -5.753 8.96e-09 \*\*\*  
## hr5 -20.8171 6.8897 -3.021 0.00252 \*\*   
## hr6 37.9933 6.9745 5.447 5.21e-08 \*\*\*  
## hr7 168.1327 6.9311 24.258 < 2e-16 \*\*\*  
## hr8 317.1462 6.9078 45.911 < 2e-16 \*\*\*  
## hr9 167.5306 6.9515 24.100 < 2e-16 \*\*\*  
## hr10 111.9697 6.9330 16.150 < 2e-16 \*\*\*  
## hr11 143.7975 6.9399 20.720 < 2e-16 \*\*\*  
## hr12 187.6260 7.0152 26.746 < 2e-16 \*\*\*  
## hr13 177.5887 6.9741 25.464 < 2e-16 \*\*\*  
## hr14 167.1369 7.0500 23.707 < 2e-16 \*\*\*  
## hr15 169.5986 7.0837 23.942 < 2e-16 \*\*\*  
## hr16 234.1743 7.0265 33.327 < 2e-16 \*\*\*  
## hr17 377.4241 7.0405 53.608 < 2e-16 \*\*\*  
## hr18 355.9896 7.0505 50.492 < 2e-16 \*\*\*  
## hr19 245.7974 7.0704 34.764 < 2e-16 \*\*\*  
## hr20 160.9709 6.9122 23.288 < 2e-16 \*\*\*  
## hr21 113.0973 6.9739 16.217 < 2e-16 \*\*\*  
## hr22 74.9889 6.8726 10.911 < 2e-16 \*\*\*  
## hr23 35.7396 6.9379 5.151 2.63e-07 \*\*\*  
## holidayHoliday -33.5339 6.4191 -5.224 1.78e-07 \*\*\*  
## weekdaySunday -17.9320 3.7804 -4.743 2.13e-06 \*\*\*  
## weekdayMonday -10.5093 3.9080 -2.689 0.00717 \*\*   
## weekdayTuesday -8.1391 3.8139 -2.134 0.03286 \*   
## weekdayWednesday -3.9877 3.7940 -1.051 0.29325   
## weekdayThursday -3.5232 3.7950 -0.928 0.35323   
## weekdayFriday -0.3563 3.7805 -0.094 0.92492   
## temp 287.5534 12.2548 23.465 < 2e-16 \*\*\*  
## weathersitMisty -19.1011 2.3713 -8.055 8.67e-16 \*\*\*  
## weathersitLightPrecip -90.8396 3.8369 -23.675 < 2e-16 \*\*\*  
## weathersitHeavyPrecip -151.9205 111.7956 -1.359 0.17420   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 111.6 on 12118 degrees of freedom  
## Multiple R-squared: 0.6238, Adjusted R-squared: 0.6223   
## F-statistic: 418.6 on 48 and 12118 DF, p-value: < 2.2e-16

### Task 4 - Use the predict functions to make predictions using the mod1. Use “head” to display the first six predictions.

The first 6 predictons in the testing set range from -57 to 299 on the fitted lines with a upper value range from 161.15 - 518.20 and a lower value range from 79.97 to -277.10 using the variables to determine the number of bike rides.

predict\_train <- predict(mod1, interval = "prediction")

## Warning in predict.lm(mod1, interval = "prediction"): predictions on current data refer to \_future\_ responses

head(predict\_train)

## fit lwr upr  
## 1 -18.05766 -237.16187 201.0466  
## 2 -55.18473 -274.31709 163.9476  
## 3 -57.97587 -277.10229 161.1505  
## 4 299.08852 79.97305 518.2040  
## 5 172.47726 -46.65502 391.6095  
## 6 134.16955 -84.97111 353.3102

#### Task 5 - Use the predict functions to make prediction using the model from Task 3 on the training set. Use the “head” funiton to display the first six predicitons.

The first 6 predicitons in the training set range from -53 to 204 on the fitted lines with a upper value range from 164.0357 - 422.1641 and a lower range between -13.93566 and -271.72673 using the variables to determine the number of bike rides.

mod2 <- lm(count ~ season + mnth + hr + holiday + weekday + temp + weathersit, test)  
summary(mod2)

##   
## Call:  
## lm(formula = count ~ season + mnth + hr + holiday + weekday +   
## temp + weathersit, data = test)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -410.28 -62.29 -9.17 51.61 441.22   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -81.2588 10.7905 -7.531 5.93e-14 \*\*\*  
## seasonSummer 38.0492 9.8498 3.863 0.000113 \*\*\*  
## seasonFall 41.5297 11.7666 3.529 0.000420 \*\*\*  
## seasonWinter 61.4329 9.8616 6.230 5.05e-10 \*\*\*  
## mnth2 3.9366 7.6550 0.514 0.607095   
## mnth3 -4.2944 8.7089 -0.493 0.621959   
## mnth4 -10.2516 13.1142 -0.782 0.434419   
## mnth5 -12.5769 13.9927 -0.899 0.368793   
## mnth6 -31.2641 14.3428 -2.180 0.029320 \*   
## mnth7 -63.3355 16.2468 -3.898 9.81e-05 \*\*\*  
## mnth8 -46.4837 15.7505 -2.951 0.003179 \*\*   
## mnth9 -9.9654 14.0626 -0.709 0.478578   
## mnth10 -5.5380 12.9587 -0.427 0.669137   
## mnth11 -22.9823 12.5329 -1.834 0.066747 .   
## mnth12 -12.6231 9.7865 -1.290 0.197162   
## hr1 -21.5605 10.7744 -2.001 0.045437 \*   
## hr2 -25.9739 11.2020 -2.319 0.020451 \*   
## hr3 -39.9004 10.9228 -3.653 0.000262 \*\*\*  
## hr4 -42.3093 10.7914 -3.921 8.95e-05 \*\*\*  
## hr5 -35.6483 11.2763 -3.161 0.001580 \*\*   
## hr6 23.1091 10.8523 2.129 0.033267 \*   
## hr7 172.6013 10.9485 15.765 < 2e-16 \*\*\*  
## hr8 294.4587 10.9972 26.776 < 2e-16 \*\*\*  
## hr9 157.6936 10.8536 14.529 < 2e-16 \*\*\*  
## hr10 111.0896 10.9589 10.137 < 2e-16 \*\*\*  
## hr11 127.1225 11.0364 11.518 < 2e-16 \*\*\*  
## hr12 163.6303 10.8943 15.020 < 2e-16 \*\*\*  
## hr13 171.4018 11.1414 15.384 < 2e-16 \*\*\*  
## hr14 143.8544 10.9818 13.099 < 2e-16 \*\*\*  
## hr15 169.1190 10.9214 15.485 < 2e-16 \*\*\*  
## hr16 223.7834 11.0625 20.229 < 2e-16 \*\*\*  
## hr17 398.9907 10.9233 36.526 < 2e-16 \*\*\*  
## hr18 342.8788 10.8085 31.723 < 2e-16 \*\*\*  
## hr19 232.3915 10.6249 21.872 < 2e-16 \*\*\*  
## hr20 161.5315 11.0579 14.608 < 2e-16 \*\*\*  
## hr21 103.6331 10.8026 9.593 < 2e-16 \*\*\*  
## hr22 64.6227 11.1300 5.806 6.77e-09 \*\*\*  
## hr23 27.7363 10.8590 2.554 0.010671 \*   
## holidayHoliday -10.9021 9.6395 -1.131 0.258113   
## weekdaySunday -11.2664 5.7012 -1.976 0.048191 \*   
## weekdayMonday -1.6451 5.8685 -0.280 0.779233   
## weekdayTuesday -3.1729 5.7334 -0.553 0.580014   
## weekdayWednesday -3.8183 5.7696 -0.662 0.508136   
## weekdayThursday -0.4456 5.7512 -0.077 0.938241   
## weekdayFriday 5.4904 5.7512 0.955 0.339799   
## temp 290.5107 18.5993 15.619 < 2e-16 \*\*\*  
## weathersitMisty -19.6637 3.6201 -5.432 5.83e-08 \*\*\*  
## weathersitLightPrecip -89.6984 5.6349 -15.918 < 2e-16 \*\*\*  
## weathersitHeavyPrecip -40.7355 78.6821 -0.518 0.604674   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 110.7 on 5163 degrees of freedom  
## Multiple R-squared: 0.6291, Adjusted R-squared: 0.6257   
## F-statistic: 182.5 on 48 and 5163 DF, p-value: < 2.2e-16

predict\_test <- predict(mod2, interval = "prediction")

## Warning in predict.lm(mod2, interval = "prediction"): predictions on current data refer to \_future\_ responses

head(predict\_test)

## fit lwr upr  
## 1 -38.90690 -256.79879 178.9850  
## 2 -43.32036 -261.28216 174.6414  
## 3 -53.84554 -271.72673 164.0357  
## 4 5.76263 -212.13921 223.6645  
## 5 149.44465 -68.47888 367.3682  
## 6 204.11420 -13.93566 422.1641

### Task 6 - Manually calculate the R-squared value on the testing set.

The value computed manually, 0.6234 is very similar to the value generated with the training set as 0.6223. This would indicate that the model is not likely overfitted.

test\_preds <- predict(mod1, newdata = test)

SSE <- sum((test$count - test\_preds)^2)  
SST <- sum((test$count - mean(test$count))^2)  
1 - SSE/SST

## [1] 0.6234064

### Task 7 - Compare k-fold cross-validation and train/test split.

Two ways to validate a linear modeal is split the dat using two methods such as: Train/Test Splitting and K-Fold Cross-Validation. These two methods help to prevent over-fitting. For Train/Test method is when you randomly split data into two groups to evaluate performance on the training set and compare the modeled results against the test set (“the one hidden behind you back”. You would use this method along with K-fold cross-validate on very large dataset.

In K-fold Cross- Validate the data is split into “k” folds or partitions typically in groups of 3, 5 or 10. Each fold of data removed ones fold to help test the values. K-Fold is preferred because the model can be run multiple (k) times to get the best view of model validity. K-fold is used with large dataset and can be combined with train-test for the best accuracy.