## Dan Sanchez

## BAN 502

## Mod 3 Assignment 1

## Dr. Stephen Hill

## June 1st, 2020

library(tidyverse)  
library(MASS)  
library(caret)

bike<-read.csv("hour.csv")  
  
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: Split the data into training and testing sets. Your training set should have 70% of the data. Use a random number (set.seed) of 1234.**

set.seed(1234)  
train.rows = createDataPartition(y=bike$count, p=0.7, list=FALSE)  
train = slice(bike,train.rows)  
test = slice(bike,-train.rows)

**Task 2: How many rows of data are in each set (training and testing)?**

The training set contains 17,379 rows, and the testing set has 5,212 rows. This aligns with the desired 70%-30% split.

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

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   
## -419.31 -61.93 -9.98 52.57 504.24   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -81.2946 6.9356 -11.721 < 2e-16 \*\*\*  
## seasonSummer 28.8486 6.4074 4.502 6.78e-06 \*\*\*  
## seasonFall 19.7865 7.6029 2.602 0.009266 \*\*   
## seasonWinter 62.0339 6.4333 9.643 < 2e-16 \*\*\*  
## mnth2 -0.8013 5.1396 -0.156 0.876114   
## mnth3 2.5584 5.7973 0.441 0.659003   
## mnth4 -1.2250 8.6334 -0.142 0.887166   
## mnth5 -1.5879 9.2279 -0.172 0.863382   
## mnth6 -15.3992 9.4846 -1.624 0.104485   
## mnth7 -38.8277 10.6085 -3.660 0.000253 \*\*\*  
## mnth8 -16.8557 10.3542 -1.628 0.103569   
## mnth9 5.4060 9.2152 0.587 0.557459   
## mnth10 -2.7341 8.5079 -0.321 0.747943   
## mnth11 -12.8043 8.2169 -1.558 0.119193   
## mnth12 -15.3615 6.5409 -2.349 0.018864 \*   
## hr1 -19.7855 6.9722 -2.838 0.004550 \*\*   
## hr2 -28.2440 6.9696 -4.052 5.10e-05 \*\*\*  
## hr3 -40.3146 7.0910 -5.685 1.34e-08 \*\*\*  
## hr4 -40.5469 7.0249 -5.772 8.03e-09 \*\*\*  
## hr5 -26.7454 6.9592 -3.843 0.000122 \*\*\*  
## hr6 32.8518 7.0435 4.664 3.13e-06 \*\*\*  
## hr7 161.3872 6.9925 23.080 < 2e-16 \*\*\*  
## hr8 312.2263 6.9502 44.923 < 2e-16 \*\*\*  
## hr9 164.2556 7.0163 23.411 < 2e-16 \*\*\*  
## hr10 107.1856 6.9552 15.411 < 2e-16 \*\*\*  
## hr11 139.6256 7.0057 19.930 < 2e-16 \*\*\*  
## hr12 179.7448 6.9778 25.760 < 2e-16 \*\*\*  
## hr13 178.6812 7.0201 25.453 < 2e-16 \*\*\*  
## hr14 156.2811 7.0628 22.127 < 2e-16 \*\*\*  
## hr15 168.7543 7.0939 23.788 < 2e-16 \*\*\*  
## hr16 228.1106 7.0881 32.182 < 2e-16 \*\*\*  
## hr17 377.6085 7.0185 53.802 < 2e-16 \*\*\*  
## hr18 347.7287 6.9806 49.813 < 2e-16 \*\*\*  
## hr19 238.7339 7.0128 34.043 < 2e-16 \*\*\*  
## hr20 159.7394 7.0231 22.745 < 2e-16 \*\*\*  
## hr21 108.1070 6.9494 15.556 < 2e-16 \*\*\*  
## hr22 72.3808 6.9874 10.359 < 2e-16 \*\*\*  
## hr23 32.5734 6.9996 4.654 3.30e-06 \*\*\*  
## holidayHoliday -29.0249 6.4088 -4.529 5.98e-06 \*\*\*  
## weekdaySunday -14.0349 3.7638 -3.729 0.000193 \*\*\*  
## weekdayMonday -6.5302 3.8944 -1.677 0.093604 .   
## weekdayTuesday -7.2790 3.8319 -1.900 0.057509 .   
## weekdayWednesday -3.2707 3.7984 -0.861 0.389212   
## weekdayThursday -1.7267 3.8053 -0.454 0.650004   
## weekdayFriday 1.3251 3.7744 0.351 0.725539   
## temp 288.1743 12.1860 23.648 < 2e-16 \*\*\*  
## weathersitMisty -19.6696 2.3717 -8.293 < 2e-16 \*\*\*  
## weathersitLightPrecip -94.1331 3.8166 -24.664 < 2e-16 \*\*\*  
## weathersitHeavyPrecip -80.2490 64.7672 -1.239 0.215356   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 111.9 on 12118 degrees of freedom  
## Multiple R-squared: 0.6217, Adjusted R-squared: 0.6202   
## F-statistic: 414.8 on 48 and 12118 DF, p-value: < 2.2e-16

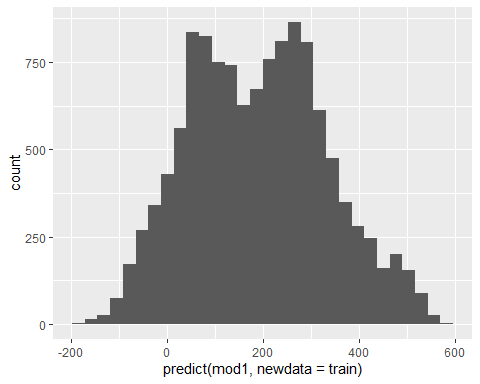
This model has a number of significant predictors of count. The adjusted R-squared value of 0.6202 is respectable in that it does not seem to under-represent or over-fit the data.

**Task 4: Use the predict functions to make predictions (using your model from Task 3) on the training set.**

predict\_train = as.data.frame(predict(mod1, newdata=train))  
ctrl = trainControl(method = "cv", number = 10)  
  
set.seed(1234)  
  
modCV\_train = train(count ~ season + mnth + hr + holiday + weekday + temp + weathersit, train, method="lm",trControl = ctrl, metric="Rsquared")  
summary(modCV\_train)

##   
## Call:  
## lm(formula = .outcome ~ ., data = dat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -419.31 -61.93 -9.98 52.57 504.24   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -81.2946 6.9356 -11.721 < 2e-16 \*\*\*  
## seasonSummer 28.8486 6.4074 4.502 6.78e-06 \*\*\*  
## seasonFall 19.7865 7.6029 2.602 0.009266 \*\*   
## seasonWinter 62.0339 6.4333 9.643 < 2e-16 \*\*\*  
## mnth2 -0.8013 5.1396 -0.156 0.876114   
## mnth3 2.5584 5.7973 0.441 0.659003   
## mnth4 -1.2250 8.6334 -0.142 0.887166   
## mnth5 -1.5879 9.2279 -0.172 0.863382   
## mnth6 -15.3992 9.4846 -1.624 0.104485   
## mnth7 -38.8277 10.6085 -3.660 0.000253 \*\*\*  
## mnth8 -16.8557 10.3542 -1.628 0.103569   
## mnth9 5.4060 9.2152 0.587 0.557459   
## mnth10 -2.7341 8.5079 -0.321 0.747943   
## mnth11 -12.8043 8.2169 -1.558 0.119193   
## mnth12 -15.3615 6.5409 -2.349 0.018864 \*   
## hr1 -19.7855 6.9722 -2.838 0.004550 \*\*   
## hr2 -28.2440 6.9696 -4.052 5.10e-05 \*\*\*  
## hr3 -40.3146 7.0910 -5.685 1.34e-08 \*\*\*  
## hr4 -40.5469 7.0249 -5.772 8.03e-09 \*\*\*  
## hr5 -26.7454 6.9592 -3.843 0.000122 \*\*\*  
## hr6 32.8518 7.0435 4.664 3.13e-06 \*\*\*  
## hr7 161.3872 6.9925 23.080 < 2e-16 \*\*\*  
## hr8 312.2263 6.9502 44.923 < 2e-16 \*\*\*  
## hr9 164.2556 7.0163 23.411 < 2e-16 \*\*\*  
## hr10 107.1856 6.9552 15.411 < 2e-16 \*\*\*  
## hr11 139.6256 7.0057 19.930 < 2e-16 \*\*\*  
## hr12 179.7448 6.9778 25.760 < 2e-16 \*\*\*  
## hr13 178.6812 7.0201 25.453 < 2e-16 \*\*\*  
## hr14 156.2811 7.0628 22.127 < 2e-16 \*\*\*  
## hr15 168.7543 7.0939 23.788 < 2e-16 \*\*\*  
## hr16 228.1106 7.0881 32.182 < 2e-16 \*\*\*  
## hr17 377.6085 7.0185 53.802 < 2e-16 \*\*\*  
## hr18 347.7287 6.9806 49.813 < 2e-16 \*\*\*  
## hr19 238.7339 7.0128 34.043 < 2e-16 \*\*\*  
## hr20 159.7394 7.0231 22.745 < 2e-16 \*\*\*  
## hr21 108.1070 6.9494 15.556 < 2e-16 \*\*\*  
## hr22 72.3808 6.9874 10.359 < 2e-16 \*\*\*  
## hr23 32.5734 6.9996 4.654 3.30e-06 \*\*\*  
## holidayHoliday -29.0249 6.4088 -4.529 5.98e-06 \*\*\*  
## weekdaySunday -14.0349 3.7638 -3.729 0.000193 \*\*\*  
## weekdayMonday -6.5302 3.8944 -1.677 0.093604 .   
## weekdayTuesday -7.2790 3.8319 -1.900 0.057509 .   
## weekdayWednesday -3.2707 3.7984 -0.861 0.389212   
## weekdayThursday -1.7267 3.8053 -0.454 0.650004   
## weekdayFriday 1.3251 3.7744 0.351 0.725539   
## temp 288.1743 12.1860 23.648 < 2e-16 \*\*\*  
## weathersitMisty -19.6696 2.3717 -8.293 < 2e-16 \*\*\*  
## weathersitLightPrecip -94.1331 3.8166 -24.664 < 2e-16 \*\*\*  
## weathersitHeavyPrecip -80.2490 64.7672 -1.239 0.215356   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 111.9 on 12118 degrees of freedom  
## Multiple R-squared: 0.6217, Adjusted R-squared: 0.6202   
## F-statistic: 414.8 on 48 and 12118 DF, p-value: < 2.2e-16

ggplot(predict\_train,aes(x=`predict(mod1, newdata = train)`))+  
 geom\_histogram()



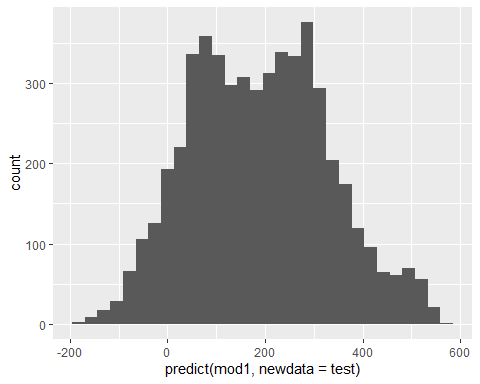
The distribution of predictions seems relatively normal. However, there are quite a few predictions that fall below zero, which would not be possible for ride count.

**Task 5: Use the predict functions to make predictions (using your model from Task 3) on the testing set. As you did in Task 4, comment on the predictions.**

predict\_test = as.data.frame(predict(mod1, newdata=test))  
ctrl = trainControl(method = "cv", number = 10)  
set.seed(1234)  
modCV\_test = train(count ~ season + mnth + hr + holiday + weekday + temp + weathersit, test, method="lm",trControl = ctrl, metric="Rsquared")  
summary(modCV\_test)

##   
## Call:  
## lm(formula = .outcome ~ ., data = dat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -389.05 -62.15 -9.48 51.45 473.83   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -96.072 10.670 -9.004 < 2e-16 \*\*\*  
## seasonSummer 51.104 9.524 5.366 8.41e-08 \*\*\*  
## seasonFall 44.148 11.259 3.921 8.92e-05 \*\*\*  
## seasonWinter 72.308 9.563 7.562 4.69e-14 \*\*\*  
## mnth2 6.895 7.796 0.884 0.376528   
## mnth3 12.047 8.705 1.384 0.166428   
## mnth4 -15.559 12.836 -1.212 0.225509   
## mnth5 -14.898 13.725 -1.085 0.277768   
## mnth6 -23.393 14.059 -1.664 0.096179 .   
## mnth7 -46.424 15.944 -2.912 0.003609 \*\*   
## mnth8 -30.637 15.374 -1.993 0.046333 \*   
## mnth9 3.879 13.658 0.284 0.776437   
## mnth10 -5.872 12.840 -0.457 0.647446   
## mnth11 -29.637 12.317 -2.406 0.016155 \*   
## mnth12 -13.267 9.670 -1.372 0.170134   
## hr1 -13.072 10.776 -1.213 0.225140   
## hr2 -23.518 10.903 -2.157 0.031048 \*   
## hr3 -32.671 10.736 -3.043 0.002353 \*\*   
## hr4 -42.256 10.986 -3.846 0.000121 \*\*\*  
## hr5 -20.861 10.976 -1.901 0.057413 .   
## hr6 34.524 10.604 3.256 0.001138 \*\*   
## hr7 188.581 10.714 17.602 < 2e-16 \*\*\*  
## hr8 306.934 10.827 28.348 < 2e-16 \*\*\*  
## hr9 165.493 10.629 15.569 < 2e-16 \*\*\*  
## hr10 122.819 10.863 11.306 < 2e-16 \*\*\*  
## hr11 137.946 10.798 12.775 < 2e-16 \*\*\*  
## hr12 181.678 11.009 16.502 < 2e-16 \*\*\*  
## hr13 169.313 10.962 15.446 < 2e-16 \*\*\*  
## hr14 169.986 10.937 15.542 < 2e-16 \*\*\*  
## hr15 172.889 10.869 15.907 < 2e-16 \*\*\*  
## hr16 239.579 10.856 22.069 < 2e-16 \*\*\*  
## hr17 401.219 10.973 36.563 < 2e-16 \*\*\*  
## hr18 362.759 11.017 32.927 < 2e-16 \*\*\*  
## hr19 248.514 10.776 23.063 < 2e-16 \*\*\*  
## hr20 164.669 10.661 15.446 < 2e-16 \*\*\*  
## hr21 116.041 10.858 10.687 < 2e-16 \*\*\*  
## hr22 72.658 10.704 6.788 1.27e-11 \*\*\*  
## hr23 34.783 10.641 3.269 0.001087 \*\*   
## holidayHoliday -18.706 9.667 -1.935 0.053045 .   
## weekdaySunday -20.269 5.774 -3.511 0.000451 \*\*\*  
## weekdayMonday -9.862 5.912 -1.668 0.095336 .   
## weekdayTuesday -4.620 5.672 -0.815 0.415328   
## weekdayWednesday -4.953 5.748 -0.862 0.388880   
## weekdayThursday -3.755 5.723 -0.656 0.511747   
## weekdayFriday 3.370 5.764 0.585 0.558861   
## temp 286.865 18.856 15.213 < 2e-16 \*\*\*  
## weathersitMisty -19.155 3.625 -5.284 1.32e-07 \*\*\*  
## weathersitLightPrecip -82.333 5.702 -14.438 < 2e-16 \*\*\*  
## weathersitHeavyPrecip NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 110.1 on 5164 degrees of freedom  
## Multiple R-squared: 0.6334, Adjusted R-squared: 0.6301   
## F-statistic: 189.9 on 47 and 5164 DF, p-value: < 2.2e-16

ggplot(predict\_test,aes(x=`predict(mod1, newdata = test)`))+  
 geom\_histogram()



The predictions for the testing set have a similar distribution to the training set. The issue of negative predictions for count also appears for this data set.

**Task 6: Manually calculate the R squared value on the testing set. Comment on how this value compares to the model’s performance on the training set.**

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

## [1] 0.6289223

The R Squared value for the testing set (.6289) and the training set (.6202) are very close. This shows again that the model performed similarly between both sets of data.

**Task 7: Describe how k-fold cross-validation differs from model validation via a training/testing split.**

With the training/testing split from this exercise, we randomly split the data into two unequal partitions. The best model was built on the larger partition of the data, and then tested on the smaller partition. K-fold cross-validation is more computationally intense. This would randomly split the data into more than two equal partitions. The best model would be generated from all but one of the partitions and tested on that excluded partition. This process would be repeated until all combinations are trained and tested. The next step would be to compare how each combination performed and optimize the model accordingly.