# Module Validation Assignment

## Drew Rabil

### Module 3 - Assignment 1

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.1.0 ✓ dplyr 1.0.5  
## ✓ tidyr 1.1.3 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.1

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

library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 0.1.3 ──

## ✓ broom 0.7.6 ✓ rsample 0.1.0   
## ✓ dials 0.0.9 ✓ tune 0.1.5   
## ✓ infer 0.5.4 ✓ workflows 0.2.2   
## ✓ modeldata 0.1.0 ✓ workflowsets 0.0.2   
## ✓ parsnip 0.1.5 ✓ yardstick 0.0.8   
## ✓ recipes 0.1.16

## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## x scales::discard() masks purrr::discard()  
## x dplyr::filter() masks stats::filter()  
## x recipes::fixed() masks stringr::fixed()  
## x dplyr::lag() masks stats::lag()  
## x yardstick::spec() masks readr::spec()  
## x recipes::step() masks stats::step()  
## • Use tidymodels\_prefer() to resolve common conflicts.

## Read-in Dataset

bike <- read\_csv("bike\_cleaned-2.csv")

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## instant = col\_double(),  
## dteday = col\_character(),  
## season = col\_character(),  
## mnth = col\_character(),  
## hr = col\_double(),  
## holiday = col\_character(),  
## weekday = col\_character(),  
## workingday = col\_character(),  
## weathersit = col\_character(),  
## temp = col\_double(),  
## atemp = col\_double(),  
## hum = col\_double(),  
## windspeed = col\_double(),  
## casual = col\_double(),  
## registered = col\_double(),  
## count = col\_double()  
## )

bike <- bike %>% mutate(dteday = mdy(dteday)) #convert "dteday" to a date variable  
bike <- bike %>% mutate\_if(is.character,as\_factor) #convert remaining character variables to factors  
bike <- bike %>% mutate(hr = as\_factor(hr)) #convert the"hr" variable into a factor

## Task 1

set.seed(1234)  
bike\_split = initial\_split(bike, prop = 0.70, strata = count)  
train = training(bike\_split)  
test = testing(bike\_split)

## Task 2

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

The training set has 12,163 rows of observations (or 70% of the bike data). The testing set has 5,216 rows of observations (or the remianing 30% of data from the bike dataframe).

## Task 3

bike\_recipe = recipe(count ~ season + mnth + hr + holiday + weekday + temp + weathersit, train)  
  
lm\_model = #name of model type  
 linear\_reg() %>% #doing a linear regression  
 set\_engine("lm") #specify the specific type of linear tool we are using  
  
lm\_wflow =  
 workflow() %>%  
 add\_model(lm\_model) %>%  
 add\_recipe(bike\_recipe)  
  
lm\_fit = fit(lm\_wflow, train)  
  
summary(lm\_fit$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -427.33 -62.08 -9.82 51.84 503.54   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -81.6699 6.9466 -11.757 < 2e-16 \*\*\*  
## seasonSpring 27.4972 6.3951 4.300 1.72e-05 \*\*\*  
## seasonSummer 18.7645 7.5881 2.473 0.01342 \*   
## seasonFall 62.5367 6.4533 9.691 < 2e-16 \*\*\*  
## mnthFeb -0.5997 5.1373 -0.117 0.90707   
## mnthMar 3.0778 5.7904 0.532 0.59506   
## mnthApr -1.3130 8.6231 -0.152 0.87898   
## mnthMay -2.6894 9.2230 -0.292 0.77060   
## mnthJun -15.8125 9.4879 -1.667 0.09562 .   
## mnthJul -40.2300 10.6077 -3.793 0.00015 \*\*\*  
## mnthAug -16.4993 10.3574 -1.593 0.11119   
## mnthSep 3.9859 9.2187 0.432 0.66548   
## mnthOct -3.0817 8.5334 -0.361 0.71800   
## mnthNov -14.7632 8.2403 -1.792 0.07322 .   
## mnthDec -16.2734 6.5606 -2.480 0.01313 \*   
## hr1 -20.7836 6.9908 -2.973 0.00295 \*\*   
## hr2 -29.0673 6.9980 -4.154 3.29e-05 \*\*\*  
## hr3 -41.4592 7.0968 -5.842 5.29e-09 \*\*\*  
## hr4 -41.2506 7.0386 -5.861 4.73e-09 \*\*\*  
## hr5 -27.2665 6.9794 -3.907 9.41e-05 \*\*\*  
## hr6 31.8318 7.0125 4.539 5.70e-06 \*\*\*  
## hr7 164.5446 7.0278 23.413 < 2e-16 \*\*\*  
## hr8 305.3583 6.9782 43.759 < 2e-16 \*\*\*  
## hr9 163.9524 7.0096 23.390 < 2e-16 \*\*\*  
## hr10 105.9395 6.9986 15.137 < 2e-16 \*\*\*  
## hr11 138.1987 6.9861 19.782 < 2e-16 \*\*\*  
## hr12 179.5246 6.9799 25.720 < 2e-16 \*\*\*  
## hr13 177.5739 7.0533 25.176 < 2e-16 \*\*\*  
## hr14 152.0364 7.1106 21.382 < 2e-16 \*\*\*  
## hr15 170.3496 7.0967 24.004 < 2e-16 \*\*\*  
## hr16 229.1493 7.1110 32.225 < 2e-16 \*\*\*  
## hr17 384.6252 7.0221 54.774 < 2e-16 \*\*\*  
## hr18 342.3854 7.0387 48.643 < 2e-16 \*\*\*  
## hr19 236.7980 7.0437 33.618 < 2e-16 \*\*\*  
## hr20 158.1195 7.0488 22.432 < 2e-16 \*\*\*  
## hr21 107.9022 6.9453 15.536 < 2e-16 \*\*\*  
## hr22 72.0674 6.9890 10.312 < 2e-16 \*\*\*  
## hr23 31.3404 7.0004 4.477 7.64e-06 \*\*\*  
## holidayHoliday -25.5839 6.3712 -4.016 5.97e-05 \*\*\*  
## weekdaySunday -12.8572 3.7603 -3.419 0.00063 \*\*\*  
## weekdayMonday -8.6638 3.8974 -2.223 0.02623 \*   
## weekdayTuesday -6.7687 3.8295 -1.768 0.07716 .   
## weekdayWednesday -3.6852 3.8010 -0.970 0.33231   
## weekdayThursday -3.1739 3.8047 -0.834 0.40418   
## weekdayFriday 0.5683 3.7761 0.151 0.88036   
## temp 293.4586 12.1953 24.063 < 2e-16 \*\*\*  
## weathersitMisty -19.7902 2.3715 -8.345 < 2e-16 \*\*\*  
## weathersitLightPrecip -92.1438 3.8276 -24.073 < 2e-16 \*\*\*  
## weathersitHeavyPrecip -78.2430 64.7522 -1.208 0.22694   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 111.8 on 12114 degrees of freedom  
## Multiple R-squared: 0.6224, Adjusted R-squared: 0.6209   
## F-statistic: 416.1 on 48 and 12114 DF, p-value: < 2.2e-16

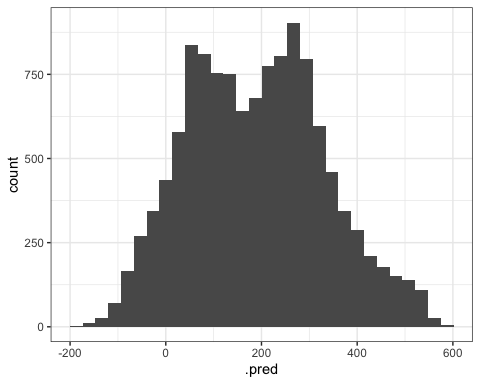
The model for the training set has an adjusted R-squared value of 0.6209. In other words, this means that using the seven variables (“season”, “mnth”, “hr”, “holiday”, and “weekday”, “temp”, and “weathersit”) can predict the variable count roughly 62% of the time. The summary of the model shows that only certain days of the week, seasons, and months are of significance while every hour of the day is of significance.

The next step is to see if this model will be generalizable for outside data the model has not seen.

## Task 4

predict\_train <- predict(lm\_fit,train)  
  
ggplot(predict\_train,aes(x=.pred)) +  
 geom\_histogram() +  
 theme\_bw()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



The histogram shows the distribution of the predictions to be close to a normal distribution. Instead of being a normal bell curve, there is a bimodal shape at the top with two separate peaks for the predictions.

## Task 5

lm\_fit %>% predict(test) %>% bind\_cols(test) %>% metrics(truth = count, estimate = .pred)

## # A tibble: 3 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 rmse standard 110.   
## 2 rsq standard 0.627  
## 3 mae standard 80.1

The R-squared value of the model on the testing set is 0.6271. This R-squared is very close to the initial adjusted r-square value of 0.6209 for the model on the training set. The similarity in R-squared values tells us this model is a good fit for data the model has not seen before.