## Module 3 Assignment 1

### Underwood, Katie

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.4 v dplyr 1.0.2  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.0

## -- 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 0.1.2 --

## v broom 0.7.2 v recipes 0.1.15  
## v dials 0.0.9 v rsample 0.0.8   
## v infer 0.5.4 v tune 0.1.2   
## v modeldata 0.1.0 v workflows 0.2.1   
## v parsnip 0.1.5 v yardstick 0.0.7

## -- 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()

bike = bike\_cleaned\_2 <- 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))  
bike = bike%>% mutate(season = as.factor(season))  
bike = bike%>% mutate(mnth = as.factor(mnth))  
bike = bike%>% mutate(holiday = as.factor(holiday))  
bike = bike%>% mutate(weekday = as.factor(weekday))  
bike = bike%>% mutate(workingday = as.factor(workingday))  
bike = bike%>% mutate(weathersit = as.factor(weathersit))  
bike = bike%>% mutate(hr = as.factor(hr))

### Task 1

set.seed(1234)  
bike\_split = initial\_split(bike, prop = .7, strata = count)  
train = training(bike\_split)  
testing = testing(bike\_split)

### Task 2

There are 5212 rows in testing set and 12167 rows in training set.

### Task 3

The model has an R-squared value of .62 which is decent. The p value overall is very low (which is good) and all variables besides weathersit have a low p value.

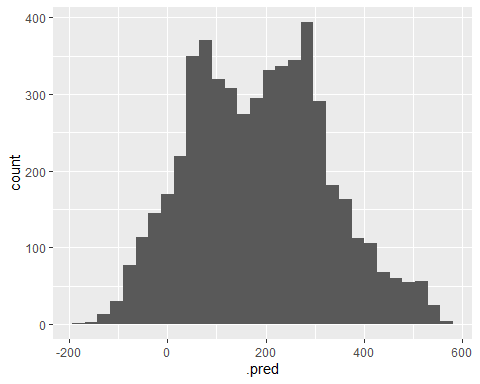
count\_recipe = recipe(count ~ season + mnth + hr + holiday + weekday + temp + weathersit, train) %>%  
 step\_dummy(all\_nominal()) %>%  
 step\_center(all\_predictors()) %>%  
 step\_scale(all\_predictors())  
  
lm\_model = linear\_reg() %>%   
 set\_engine("lm")  
  
lm\_workflow = workflow() %>%   
 add\_model(lm\_model) %>%  
 add\_recipe(count\_recipe)  
  
lm\_fit = fit(lm\_workflow, train)  
  
#summary(lm\_fit$fit$fit$fit)

### Task 4

The data looks fairly normally distributed with an average just under 200. This means the predicted count of bikes would be 200 on average.

predict\_train = lm\_fit %>% predict(testing)  
  
ggplot(data=predict\_train, aes(x=.pred)) +  
 geom\_histogram()

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



### Task 5

The R squared value for the testing set is .61 which is nearly identical to the R squared value for the training set (.62). This indicates that this model is a good fit for the data.