library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.6 v dplyr 1.0.7  
## v tidyr 1.1.4 v stringr 1.4.0  
## v readr 2.1.1 v forcats 0.5.1

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

library(tidymodels)

## Registered S3 method overwritten by 'tune':  
## method from   
## required\_pkgs.model\_spec parsnip

## -- Attaching packages -------------------------------------- tidymodels 0.1.4 --

## v broom 0.7.11 v rsample 0.1.1   
## v dials 0.0.10 v tune 0.1.6   
## v infer 1.0.0 v workflows 0.2.4   
## v modeldata 0.1.1 v workflowsets 0.1.0   
## v parsnip 0.1.7 v yardstick 0.0.9   
## v recipes 0.1.17

## -- 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.

library(lubridate)

##   
## Attaching package: 'lubridate'

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

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

## Rows: 17379 Columns: 16

## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (7): dteday, season, mnth, holiday, weekday, workingday, weathersit  
## dbl (9): instant, hr, temp, atemp, hum, windspeed, casual, registered, count

##   
## i Use `spec()` to retrieve the full column specification for this data.  
## i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

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))

bike2 = bike %>% dplyr::select("season", "hr", "holiday", "weekday", "temp", "weathersit", "count")

set.seed(1234)  
bike\_split = initial\_split(bike2, prop = 0.70, strata = count)  
train = bike\_split %>%  
 training()  
test = bike\_split %>%  
 testing()

How many rows of data rare in each set? (Training and Testing) In the training set, there are 12163 rows of 7 variables. In the testing set, there are 5216 rows of 7 variables.

bike\_recipe = recipe(count ~., train) %>%  
 step\_dummy(all\_nominal())   
   
  
lm\_model = #give the model type a name  
 linear\_reg() %>% #specify that we are doing linear regression  
 set\_engine("lm") #specify the specify type of linear tool we want to use  
  
  
  
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   
## -413.14 -62.51 -9.67 51.32 525.76   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -80.1601 6.3363 -12.651 < 2e-16 \*\*\*  
## temp 280.1483 9.2458 30.300 < 2e-16 \*\*\*  
## season\_Spring 27.9009 3.6586 7.626 2.60e-14 \*\*\*  
## season\_Summer 6.2334 4.7245 1.319 0.187070   
## season\_Fall 56.1343 3.1287 17.942 < 2e-16 \*\*\*  
## hr\_X1 -21.1023 7.0140 -3.009 0.002630 \*\*   
## hr\_X2 -29.7472 7.0204 -4.237 2.28e-05 \*\*\*  
## hr\_X3 -41.9009 7.1179 -5.887 4.05e-09 \*\*\*  
## hr\_X4 -41.9038 7.0575 -5.937 2.97e-09 \*\*\*  
## hr\_X5 -28.5854 6.9958 -4.086 4.42e-05 \*\*\*  
## hr\_X6 30.8661 7.0306 4.390 1.14e-05 \*\*\*  
## hr\_X7 163.7009 7.0469 23.230 < 2e-16 \*\*\*  
## hr\_X8 304.7114 7.0003 43.528 < 2e-16 \*\*\*  
## hr\_X9 163.3760 7.0318 23.234 < 2e-16 \*\*\*  
## hr\_X10 106.8102 7.0167 15.222 < 2e-16 \*\*\*  
## hr\_X11 138.5361 6.9991 19.793 < 2e-16 \*\*\*  
## hr\_X12 180.1528 6.9808 25.807 < 2e-16 \*\*\*  
## hr\_X13 178.4438 7.0457 25.327 < 2e-16 \*\*\*  
## hr\_X14 152.4613 7.0921 21.497 < 2e-16 \*\*\*  
## hr\_X15 171.3886 7.0741 24.228 < 2e-16 \*\*\*  
## hr\_X16 229.7513 7.0882 32.413 < 2e-16 \*\*\*  
## hr\_X17 385.6743 7.0119 55.003 < 2e-16 \*\*\*  
## hr\_X18 343.3196 7.0386 48.777 < 2e-16 \*\*\*  
## hr\_X19 237.4914 7.0551 33.662 < 2e-16 \*\*\*  
## hr\_X20 158.5944 7.0642 22.450 < 2e-16 \*\*\*  
## hr\_X21 108.1126 6.9664 15.519 < 2e-16 \*\*\*  
## hr\_X22 71.9522 7.0103 10.264 < 2e-16 \*\*\*  
## hr\_X23 31.1293 7.0241 4.432 9.43e-06 \*\*\*  
## holiday\_Holiday -26.1995 6.3233 -4.143 3.45e-05 \*\*\*  
## weekday\_Sunday -12.4317 3.7697 -3.298 0.000977 \*\*\*  
## weekday\_Monday -8.2923 3.9067 -2.123 0.033808 \*   
## weekday\_Tuesday -6.3700 3.8360 -1.661 0.096818 .   
## weekday\_Wednesday -3.0513 3.8093 -0.801 0.423147   
## weekday\_Thursday -1.8988 3.8110 -0.498 0.618310   
## weekday\_Friday 0.9383 3.7877 0.248 0.804342   
## weathersit\_Misty -18.5374 2.3695 -7.823 5.56e-15 \*\*\*  
## weathersit\_LightPrecip -90.3413 3.8277 -23.602 < 2e-16 \*\*\*  
## weathersit\_HeavyPrecip -78.0093 64.9218 -1.202 0.229546   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 112.2 on 12125 degrees of freedom  
## Multiple R-squared: 0.6194, Adjusted R-squared: 0.6183   
## F-statistic: 533.4 on 37 and 12125 DF, p-value: < 2.2e-16

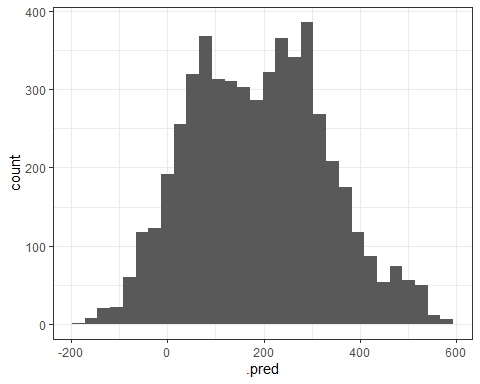
The adjusted R-squared value being .6183 is not surprising given the number of variables (alongside contemplating how all of these variables would interact with one another naturally). The p-value being sub .05 is not alarming and relatively reasonable to expect with this model. There could also be some multicollinearity looking at some of the variables(weekday, holiday, weathersit, for example). It would still be in our best interest to test the model for over-fitting and overall quality & performance as well.

predict\_train <- predict(lm\_fit, test) %>%  
 bind\_cols(test)  
predict\_train

## # A tibble: 5,216 x 8  
## .pred season hr holiday weekday temp weathersit count  
## <dbl> <fct> <fct> <fct> <fct> <dbl> <fct> <dbl>  
## 1 -12.9 Winter 0 NotHoliday Saturday 0.24 NoPrecip 16  
## 2 140. Winter 7 NotHoliday Saturday 0.2 NoPrecip 3  
## 3 172. Winter 20 NotHoliday Saturday 0.4 Misty 36  
## 4 17.7 Winter 0 NotHoliday Sunday 0.46 Misty 17  
## 5 -23.2 Winter 2 NotHoliday Sunday 0.42 Misty 9  
## 6 170. Winter 12 NotHoliday Sunday 0.36 Misty 93  
## 7 168. Winter 13 NotHoliday Sunday 0.36 Misty 75  
## 8 70.4 Winter 14 NotHoliday Sunday 0.36 LightPrecip 59  
## 9 83.7 Winter 15 NotHoliday Sunday 0.34 LightPrecip 74  
## 10 -53.5 Winter 1 NotHoliday Monday 0.2 NoPrecip 2  
## # ... with 5,206 more rows

ggplot(predict\_train, mapping = aes( x= .pred)) +   
 geom\_histogram() + theme\_bw()

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



Looking at the predictions in a histogram format shows that the distribution has some sort of bi modal distribution, as it has two very obvious peaks. It does seem to have a relatively normal distribution beyond that fact. There does not appear to be any blatant skewness one way or the other, nor any clusters/outliers that would cause serious concern with this predictive model.

bike\_recipe2 = recipe(count ~., test) %>%  
 step\_dummy(all\_nominal())  
  
lm\_model2 = #give the model type a name  
 linear\_reg() %>% #specify that we are doing linear regression  
 set\_engine("lm") #specify the specify type of linear tool we want to use  
  
  
  
lm\_wflow2 =   
 workflow() %>%  
 add\_model(lm\_model) %>%  
 add\_recipe(bike\_recipe2)  
  
lm\_fit2 = fit(lm\_wflow2, train)

summary(lm\_fit2$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -413.14 -62.51 -9.67 51.32 525.76   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -80.1601 6.3363 -12.651 < 2e-16 \*\*\*  
## temp 280.1483 9.2458 30.300 < 2e-16 \*\*\*  
## season\_Spring 27.9009 3.6586 7.626 2.60e-14 \*\*\*  
## season\_Summer 6.2334 4.7245 1.319 0.187070   
## season\_Fall 56.1343 3.1287 17.942 < 2e-16 \*\*\*  
## hr\_X1 -21.1023 7.0140 -3.009 0.002630 \*\*   
## hr\_X2 -29.7472 7.0204 -4.237 2.28e-05 \*\*\*  
## hr\_X3 -41.9009 7.1179 -5.887 4.05e-09 \*\*\*  
## hr\_X4 -41.9038 7.0575 -5.937 2.97e-09 \*\*\*  
## hr\_X5 -28.5854 6.9958 -4.086 4.42e-05 \*\*\*  
## hr\_X6 30.8661 7.0306 4.390 1.14e-05 \*\*\*  
## hr\_X7 163.7009 7.0469 23.230 < 2e-16 \*\*\*  
## hr\_X8 304.7114 7.0003 43.528 < 2e-16 \*\*\*  
## hr\_X9 163.3760 7.0318 23.234 < 2e-16 \*\*\*  
## hr\_X10 106.8102 7.0167 15.222 < 2e-16 \*\*\*  
## hr\_X11 138.5361 6.9991 19.793 < 2e-16 \*\*\*  
## hr\_X12 180.1528 6.9808 25.807 < 2e-16 \*\*\*  
## hr\_X13 178.4438 7.0457 25.327 < 2e-16 \*\*\*  
## hr\_X14 152.4613 7.0921 21.497 < 2e-16 \*\*\*  
## hr\_X15 171.3886 7.0741 24.228 < 2e-16 \*\*\*  
## hr\_X16 229.7513 7.0882 32.413 < 2e-16 \*\*\*  
## hr\_X17 385.6743 7.0119 55.003 < 2e-16 \*\*\*  
## hr\_X18 343.3196 7.0386 48.777 < 2e-16 \*\*\*  
## hr\_X19 237.4914 7.0551 33.662 < 2e-16 \*\*\*  
## hr\_X20 158.5944 7.0642 22.450 < 2e-16 \*\*\*  
## hr\_X21 108.1126 6.9664 15.519 < 2e-16 \*\*\*  
## hr\_X22 71.9522 7.0103 10.264 < 2e-16 \*\*\*  
## hr\_X23 31.1293 7.0241 4.432 9.43e-06 \*\*\*  
## holiday\_Holiday -26.1995 6.3233 -4.143 3.45e-05 \*\*\*  
## weekday\_Sunday -12.4317 3.7697 -3.298 0.000977 \*\*\*  
## weekday\_Monday -8.2923 3.9067 -2.123 0.033808 \*   
## weekday\_Tuesday -6.3700 3.8360 -1.661 0.096818 .   
## weekday\_Wednesday -3.0513 3.8093 -0.801 0.423147   
## weekday\_Thursday -1.8988 3.8110 -0.498 0.618310   
## weekday\_Friday 0.9383 3.7877 0.248 0.804342   
## weathersit\_Misty -18.5374 2.3695 -7.823 5.56e-15 \*\*\*  
## weathersit\_LightPrecip -90.3413 3.8277 -23.602 < 2e-16 \*\*\*  
## weathersit\_HeavyPrecip -78.0093 64.9218 -1.202 0.229546   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 112.2 on 12125 degrees of freedom  
## Multiple R-squared: 0.6194, Adjusted R-squared: 0.6183   
## F-statistic: 533.4 on 37 and 12125 DF, p-value: < 2.2e-16

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

## # A tibble: 3 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 rmse standard 112.   
## 2 rsq standard 0.619  
## 3 mae standard 80.9

lm\_fit2 %>% predict(test) %>% bind\_cols(test) %>% metrics(truth = count, estimate = .pred)

## # A tibble: 3 x 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 rmse standard 111.   
## 2 rsq standard 0.624  
## 3 mae standard 80.3

Looking at the R-squared value of the testing set versus the training set, there is a slight difference. The testing set has an rsq of 0.623 whereas the training set has an rsq of .619. A slight difference of only .04. Test has the same number of variables, but less than half of the rows. It does not appear that these models are over-fitted, but