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(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()

library(lubridate)

##   
## Attaching package: 'lubridate'

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

bike\_cleaned <- read\_csv("C:/Users/Karibu/Desktop/UNCW/Spring 1 2021/BAN 502/Module 3/bike\_cleaned.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\_cleaned

bike = bike %>% mutate(dteday = mdy(dteday))  
bike <- mutate\_if(bike, is.character, as.factor)  
bike$hr <- as.factor(bike$hr)

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

There are 4343 rows of data in testing and 13036 in training.

bike2 = train %>% dplyr::select(season, mnth, hr, holiday,weekday, temp,weathersit)

bike\_recipe = recipe(count ~ temp, train)  
lm\_model =  
linear\_reg()%>%  
set\_engine("lm")  
  
lm\_flow=  
 workflow()%>%  
 add\_model(lm\_model)%>%  
 add\_recipe(bike\_recipe)  
  
lm\_fit= fit(lm\_flow, train)

summary(lm\_fit$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -290.93 -109.61 -32.50 75.73 745.37   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.9633 4.0000 -0.241 0.81   
## temp 381.9134 7.4991 50.928 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 165.2 on 13034 degrees of freedom  
## Multiple R-squared: 0.166, Adjusted R-squared: 0.1659   
## F-statistic: 2594 on 1 and 13034 DF, p-value: < 2.2e-16

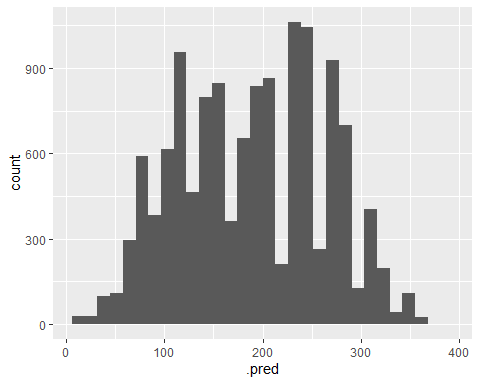
predict\_train <- predict(lm\_fit,train)  
predict\_train

## # A tibble: 13,036 x 1  
## .pred  
## <dbl>  
## 1 90.7  
## 2 83.1  
## 3 83.1  
## 4 90.7  
## 5 90.7  
## 6 83.1  
## 7 75.4  
## 8 90.7  
## 9 121.   
## 10 137.   
## # ... with 13,026 more rows

Task 4: Use the predict functions to make predictions (using your model from Task 3) on thetrainingset.Hint: Be sure to store the predictions in an object, perhaps named “predict\_train” or similar.Develop a histogram of the predictions (Hint: The predictions are likely stored in a variable called “.pred” inyour predictions object). Comment on the distribution of the predictions.

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

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



bike\_recipe\_2 = recipe(count ~ temp, test)  
lm\_model =  
linear\_reg()%>%  
set\_engine("lm")  
  
lm\_flow=  
 workflow()%>%  
 add\_model(lm\_model)%>%  
 add\_recipe(bike\_recipe)  
  
lm\_fit= fit(lm\_flow, test)

summary(lm\_fit$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -292.72 -111.55 -33.82 79.85 723.82   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.732 7.078 0.386 0.7   
## temp 379.469 13.312 28.505 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 167.9 on 4341 degrees of freedom  
## Multiple R-squared: 0.1577, Adjusted R-squared: 0.1575   
## F-statistic: 812.5 on 1 and 4341 DF, p-value: < 2.2e-16

The R squared of the test model is smaller than the fitting model, hence the model is not overfitting.