### Model Validation Assignment 1

## Engel, Alec

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

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

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

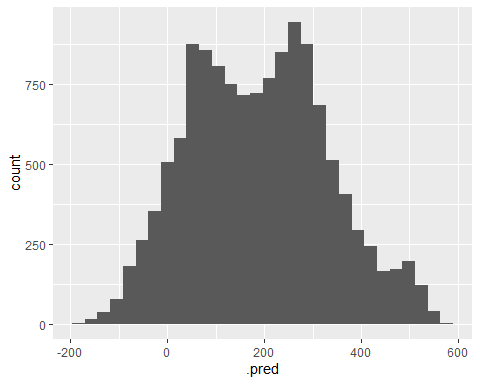
**Training = 13,036 rows**  
**Test = 4,343 rows**

bike\_recipe = recipe(count ~ season + mnth + hr + holiday + weekday + temp + weathersit,train)  
  
lm\_model =   
 linear\_reg() %>%  
 set\_engine("lm")  
  
lm\_wflow =   
 workflow() %>%  
 add\_model(lm\_model) %>%  
 add\_recipe(bike\_recipe)  
  
lm\_fit = fit(lm\_wflow, train)  
  
#summary(lm\_fit$fit$fit$fit)

**With this training model I see an estimated intercept of -.98.973 and overall p-value less than .05. The adjusted R-squared value on this training set looks decent at .6229. This training model looks to be a relatively accurate representation of the full data models I’ve previously viewed.**

predict\_train = predict(lm\_fit,train)  
ggplot(predict\_train,aes(.pred)) +  
 geom\_histogram()

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



**With this histogram, we can view it as somewhat of a normal distribution although there is a slight dip in count of bike rentals when our prediction is around 200. There is a steady incline from roughly negative 200 to positive 100, then said dip, then back up and an additional decline from 300 onwards to 600.**

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 112.   
## 2 rsq standard 0.623  
## 3 mae standard 81.6

**With this testing set, I can see that the R-squared value is .623. This R-squared value is extremely close to that of the training set leading me to believe that the training model generalizes very well.**