Wednesday

Lucy Eckert

```
data.path <- "C:/Users/leckert/Documents/NCSU/ST558/Project_2"
day <- read_csv(paste0(data.path,"/day.csv"))
```

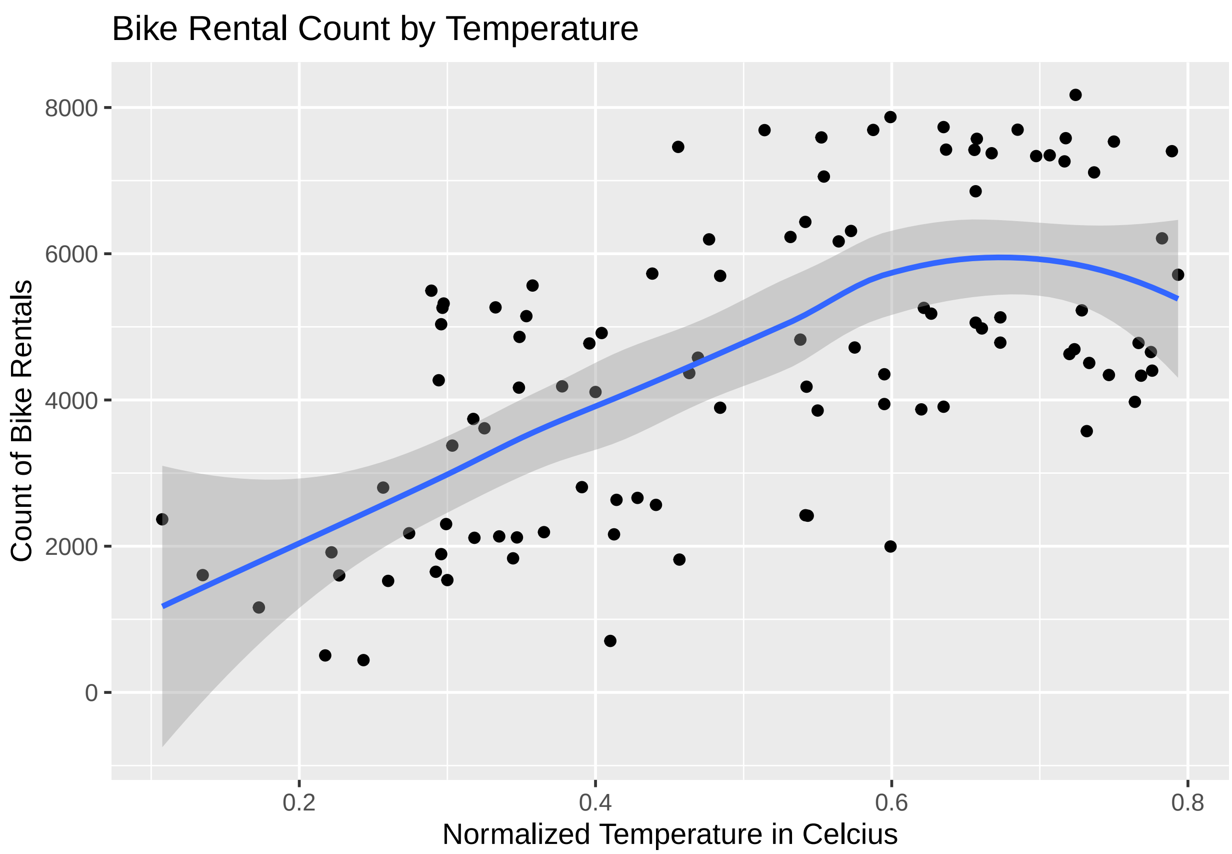
```
## Parsed with column specification:
## cols(
##   instant = col_double(),
##   dteday = col_date(format = ""),
##   season = col_double(),
##   yr = col_double(),
##   mnth = col_double(),
##   holiday = col_double(),
##   weekday = col_double(),
##   workingday = col_double(),
##   weathersit = col_double(),
##   temp = col_double(),
##   atemp = col_double(),
##   hum = col_double(),
##   windspeed = col_double(),
##   casual = col_double(),
##   registered = col_double(),
##   cnt = col_double()
## )
```

```
byday <- day %>% select(-c(casual,registered, instant, dteday))
#Filter out Monday data, and remove unused variables
ByDay <- day %>% filter(weekday==3) %>% select(-c(casual,registered, instant, dteday))
```

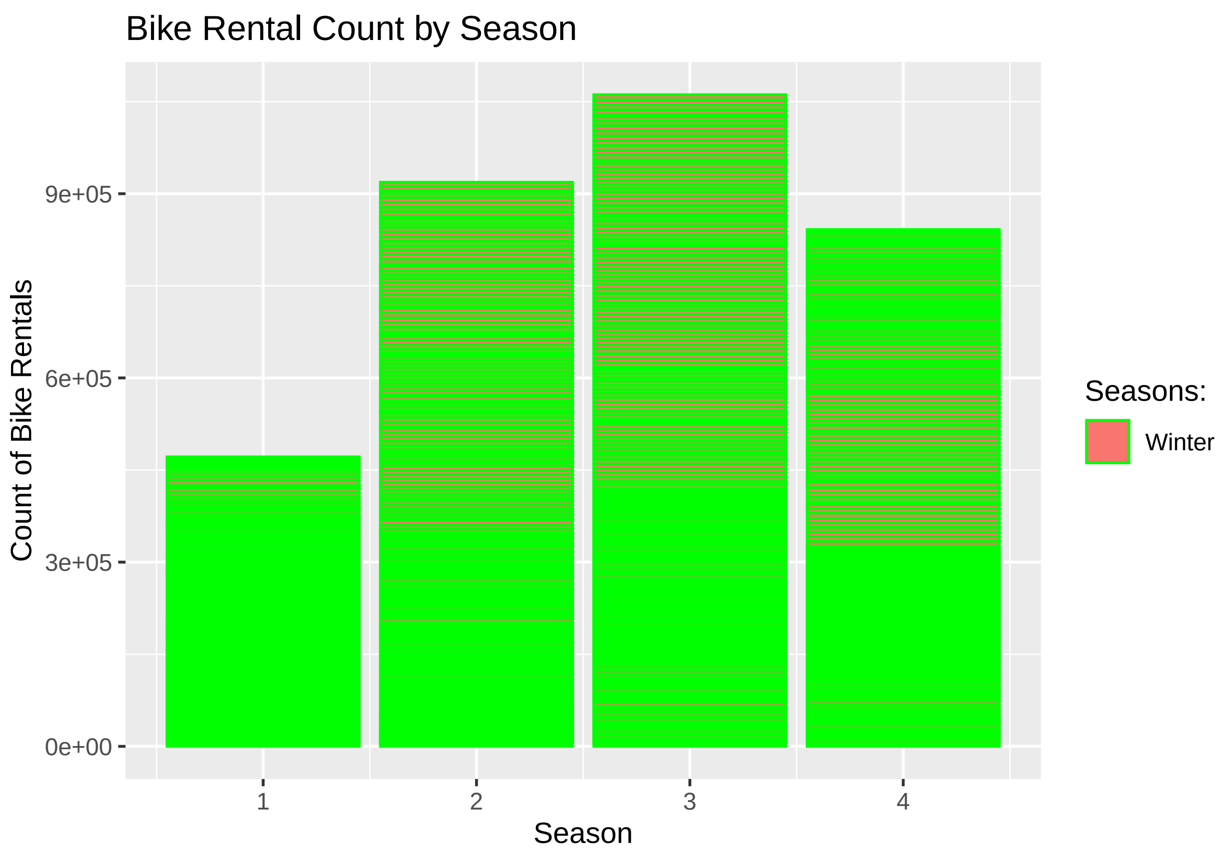
Review Data by Summaries and Plots

```
#Rentals by Temperature
a <- ggplot(ByDay, aes(temp, cnt))
a + geom_jitter() +geom_smooth() +labs(title = "Bike Rental Count by Temperature",
                                       x = "Normalized Temperature in Celcius",
                                       y = "Count of Bike Rentals")
```

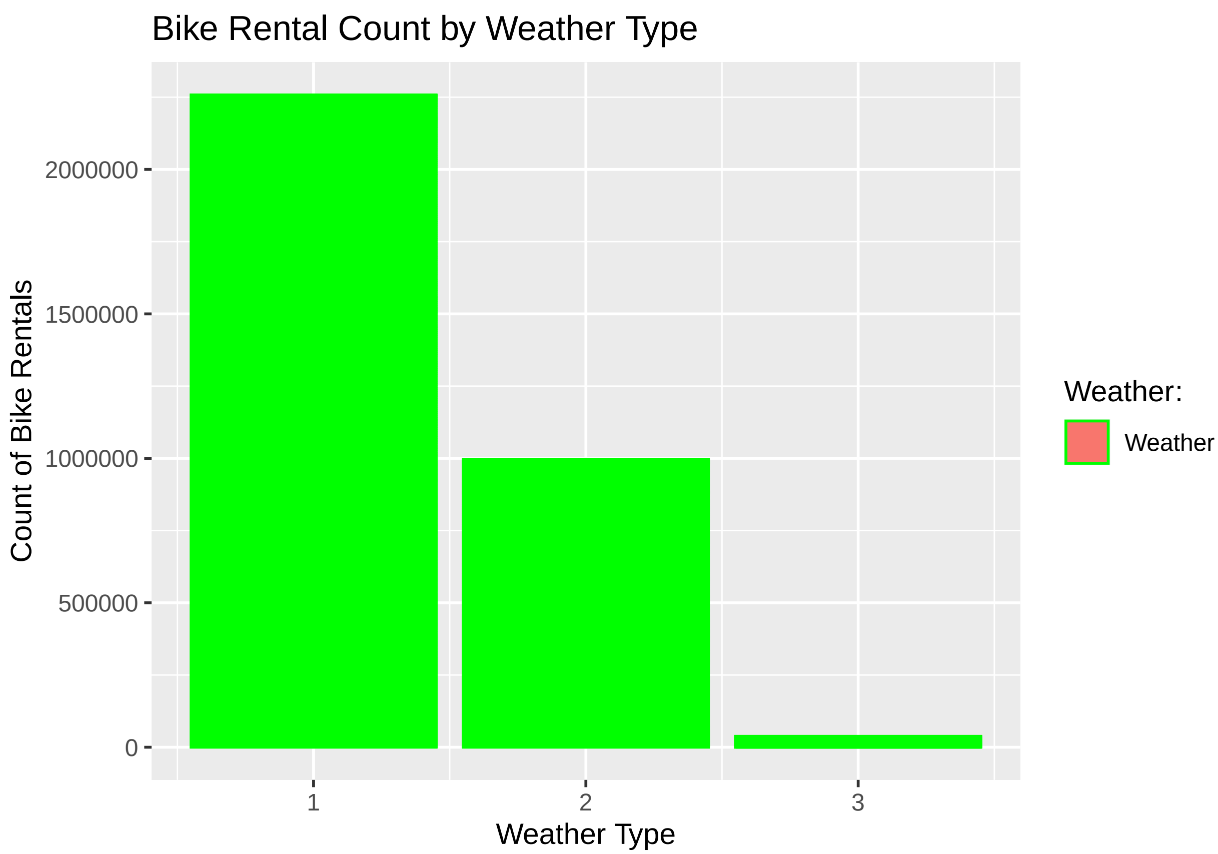
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```r
#Rentals by Season
b <- ggplot(day, aes(x = season, y = cnt))
b + geom_bar(stat = "identity", aes(y=cnt, fill="Season"), colour="green") + labs(title = "Bike Rental C
       labels = c("Winter", "Spring", "Summer", "Fall"))
```




```
#Rentals by Weather Type
c <- ggplot(day, aes(x = weathersit, y = cnt))
c + geom_bar(stat = "identity", aes(y=cnt, fill="Weather"), colour="green") +
  labs(title = "Bike Rental Count by Weather Type", x = "Weather Type", y = "Count of Bike Rentals") +
```



Review Summary Stats for Continuous Variables

```
summary_data <- ByDay %>% select(temp:windspeed)
kable(apply(summary_data, 2, summary), caption = paste("Summary Stats for Continuous Variables"),
      digit = 2)
```

Table 1: Summary Stats for Continuous Variables

	temp	atemp	hum	windspeed
Min.	0.11	0.12	0.36	0.06
1st Qu.	0.35	0.34	0.54	0.13
Median	0.53	0.51	0.63	0.18
Mean	0.50	0.48	0.65	0.19
3rd Qu.	0.66	0.61	0.74	0.24
Max.	0.79	0.75	0.97	0.42

Create train and test data sets for Monday data. Clean data.

```
set.seed(1)
trainIndex <- createDataPartition(ByDay$cnt, p = 0.7, list = FALSE)
Train <- ByDay[trainIndex, ]
Test <-  ByDay[-trainIndex, ]
```

Build Models forTrain Data

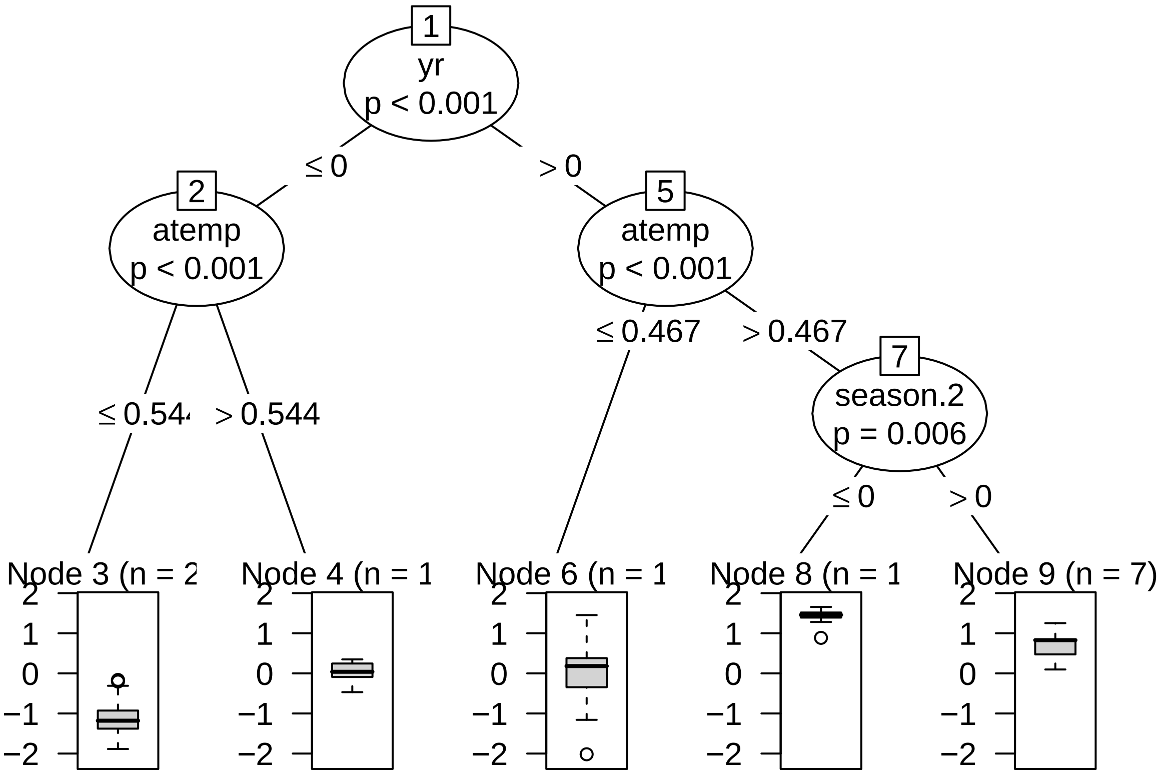
Model 1: Non-Ensemble Tree

While doing some research on model building, I discovered the concept of of using dummy variables as a way to create "switches" for some of the variables. It really helped me break down which were more useful for the model.

```
ByDay      <- day %>% filter(weekday==3) %>% select(-c(casual,registered, instant, dteday))
trainIndex   <- createDataPartition(ByDay$cnt, p = 0.7, list = FALSE)
Train <- ByDay[trainIndex, ] %>% select(-c(workingday, weekday)) %>%
  mutate(mnth=as.factor(mnth), season=as.factor(season), weathersit = as.factor(weathersit))
dmy          <- dummyVars(" ~ .", data = Train, fullRank = T)
Train.trf <- data.frame(predict(dmy, newdata = Train)) %>% mutate(y = scale(cnt)) %>% select(-cnt)
fitControl <- trainControl(method = "LOOCV")
model      <- train(y ~., data = Train.trf, method = "ctree",
                    trControl = fitControl)
print(model)
```

```
## Conditional Inference Tree
##
## 76 samples
## 22 predictors
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 75, 75, 75, 75, 75, 75, ...
## Resampling results across tuning parameters:
##
##   mincriterion  RMSE       Rsquared   MAE
##   0.01          0.6300109  0.6093195  0.4795661
##   0.50          0.6357537  0.6029646  0.4846904
##   0.99          0.6283244  0.6062013  0.4896874
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mincriterion = 0.99.
```

```
plot(model$finalModel)
```



Model 2: Boosted Tree

I selected this model after trying many combinations of the n.trees, shrinkage, and interaction depth. I selected it for the most favorable RMSE.

```
set.seed(1)
boostFit8 <- gbm(cnt ~., data = Train, distribution = "gaussian", n.trees = 100,
                 shrinkage = .1, interaction.depth = 2)
boostPred <- predict(boostFit8, newdata = dplyr::select(Test, -cnt), n.trees = 100)
boostRMSE <- sqrt(mean((boostPred-Test$cnt)^2))
#Print RMSE
boostRMSE
```

```
## [1] 1130.976
```