Experimental Project for Testing and Practice

Tom Thorpe

April 17, 2018

Objective

View different plots of the cleaned Forest Cover data from the previous section to learn more about the data.

Include required libraries.

```
progStart=Sys.time()
print(paste("R script started at",progStart))
```

```
## [1] "R script started at 2018-05-24 09:02:25"
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
library(ggplot2)
```

Point to data. The forestcover_clean_full.csv is the cleaned data to be graphed.

```
infile="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcover_clean_full2.csv"
infile="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcover_clean_full_sample2.csv"
out2file="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcover_graph.csv"
#out1file="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcoversmall_clean_full.csv"
#out2file="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcoversmall_clean.csv"

alphaVal<-0.01 # large data
alphaVal<-0.1  # small data
```

Load the data.

```
startTime=Sys.time()
print(paste("Data load started at",startTime))
```

```
## [1] "Data load started at 2018-05-24 09:02:26"
```

```
forestcover <- read.csv(infile,header=TRUE,sep=",") %>% tbl_df()

# Shorten some names
forestcover$ClimateName <- as.character(forestcover$ClimateName)
forestcover$ClimateName[forestcover$ClimateZone == 1] <- "MonLowDry" # was "Mont_LowDry"
forestcover$ClimateName[forestcover$ClimateZone == 2] <- "MonLow" # was "Montane_Low"
forestcover$ClimateName[forestcover$ClimateZone == 3] <- "MonDry" # was "Montane_Dry"
forestcover$ClimateName[forestcover$ClimateZone == 4] <- "Montane" # was "Montane"
```

```
forestcover$ClimateName[forestcover$ClimateZone == 5] <- "M&MDry" # was "Mon&Mon_Dry"
forestcover$ClimateName[forestcover$ClimateZone == 6] <- "MonSubAlp" # was "Mon_SubAlp"
forestcover$ClimateName[forestcover$ClimateZone == 7] <- "SubAlpine" # was "SubAlpine"
forestcover$ClimateName[forestcover$ClimateZone == 8] <- "Alpine" # was "Alpine"
forestcover$ClimateName <- as.factor(forestcover$ClimateName)

endTime=Sys.time()
  print(paste("Data load completed at",endTime))
```

```
## [1] "Data load completed at 2018-05-24 09:02:27"
```

```
  print(paste("Elapsed time=",endTime-startTime,"seconds."))
```

```
## [1] "Elapsed time= 1.10402512550354 seconds."
```

Data Overview

The forest cover data has a row for each sample representing a 30 meter by 30 meter square area of land.

```
#glimpse(forestcover)
```

List Data Ranges for Non-Binary Data

List Data Ranges for Non-Binary Data.

```
myranges <- function(name,x) { c(name, min = min(x), mean = mean(x), max = max(x)) }

forestDataRanges <- data.frame("Data"=character(), "min"=double(), "mean"=double(), "max"=double(),
                               stringsAsFactors=FALSE)
forestDataRanges[nrow(forestDataRanges)+1,] <- myranges("Elev",forestcover$Elev)
forestDataRanges[nrow(forestDataRanges)+1,] <- myranges("Aspect",forestcover$Aspect)
forestDataRanges[nrow(forestDataRanges)+1,] <- myranges("Slope",forestcover$Slope)
forestDataRanges[nrow(forestDataRanges)+1,] <- myranges("H2OHD",forestcover$H2OHD)
forestDataRanges[nrow(forestDataRanges)+1,] <- myranges("H2OVD",forestcover$H2OVD)
forestDataRanges[nrow(forestDataRanges)+1,] <- myranges("RoadHD",forestcover$RoadHD)
forestDataRanges[nrow(forestDataRanges)+1,] <- myranges("FirePtHD",forestcover$FirePtHD)
forestDataRanges[nrow(forestDataRanges)+1,] <- myranges("Shade9AM",forestcover$Shade9AM)
forestDataRanges[nrow(forestDataRanges)+1,] <- myranges("Shade12P",forestcover$Shade12PM)
forestDataRanges[nrow(forestDataRanges)+1,] <- myranges("Shade3PM",forestcover$Shade3PM)
forestDataRanges[nrow(forestDataRanges)+1,] <- myranges("RWwild",forestcover$RWwild)
forestDataRanges[nrow(forestDataRanges)+1,] <- myranges("NEwild",forestcover$NEwild)
forestDataRanges[nrow(forestDataRanges)+1,] <- myranges("CMwild",forestcover$CMwild)
forestDataRanges[nrow(forestDataRanges)+1,] <- myranges("CPwild",forestcover$CPwild)
forestDataRanges
```

```
##        Data  min               mean  max
## 1      Elev 1880   2960.07332185886 3857
## 2    Aspect    0   155.431755593804  360
## 3     Slope    0   14.0944061962134   57
## 4     H2OHD    0   268.901118760757 1307
## 5     H2OVD -159   46.6323580034423  508
## 6    RoadHD    0    2356.4578313253 6944
## 7  FirePtHD    0   1981.99277108434 7107
## 8  Shade9AM    0   212.124010327022  254
## 9  Shade12P  110   223.240791738382  254
## 10 Shade3PM    0   142.524354561102  249
## 11   RWwild    0  0.449569707401033    1
## 12   NEwild    0 0.0503442340791738    1
## 13   CMwild    0  0.437263339070568    1
## 14   CPwild    0 0.0628227194492255    1
```

Data distributions

Now check some basic distributions.

Elevation - Figure 1

```r
# plot 1
  jpeg(filename="ExpFigure01.jpg")
  plot(table(forestcover$Elev))
  dev.off()
```

```
## pdf
##   2
```

The distribution of the elevation seems reasonable for Colorado's high country.

Elevation Histogram 2 - Figure 2

```r
# Figure 32
  g <- ggplot(forestcover,aes(Elev)) +
        geom_histogram(bins=100) # +
       # facet_grid(. ~ factor(CovName)) +
  ggsave("ExpFigure02.jpg")
```

```
## Saving 6.5 x 4.5 in image
```

shapiro.test(forestcover$Elev) # Does myVec follow a normal distribution?

Error Message: "Sample size must be between 3 and 5000"

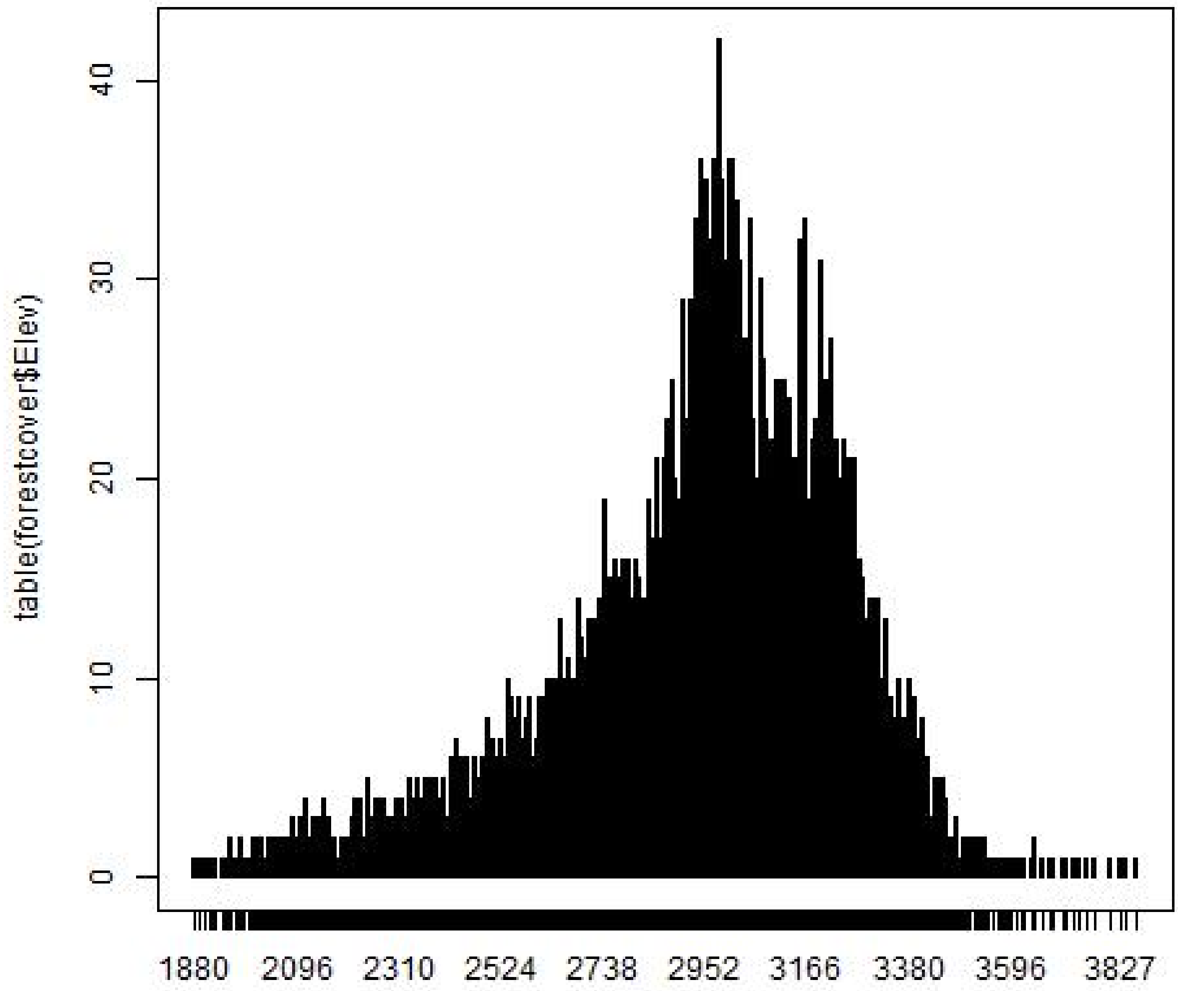

Figure 1: Elevation Histogram 1

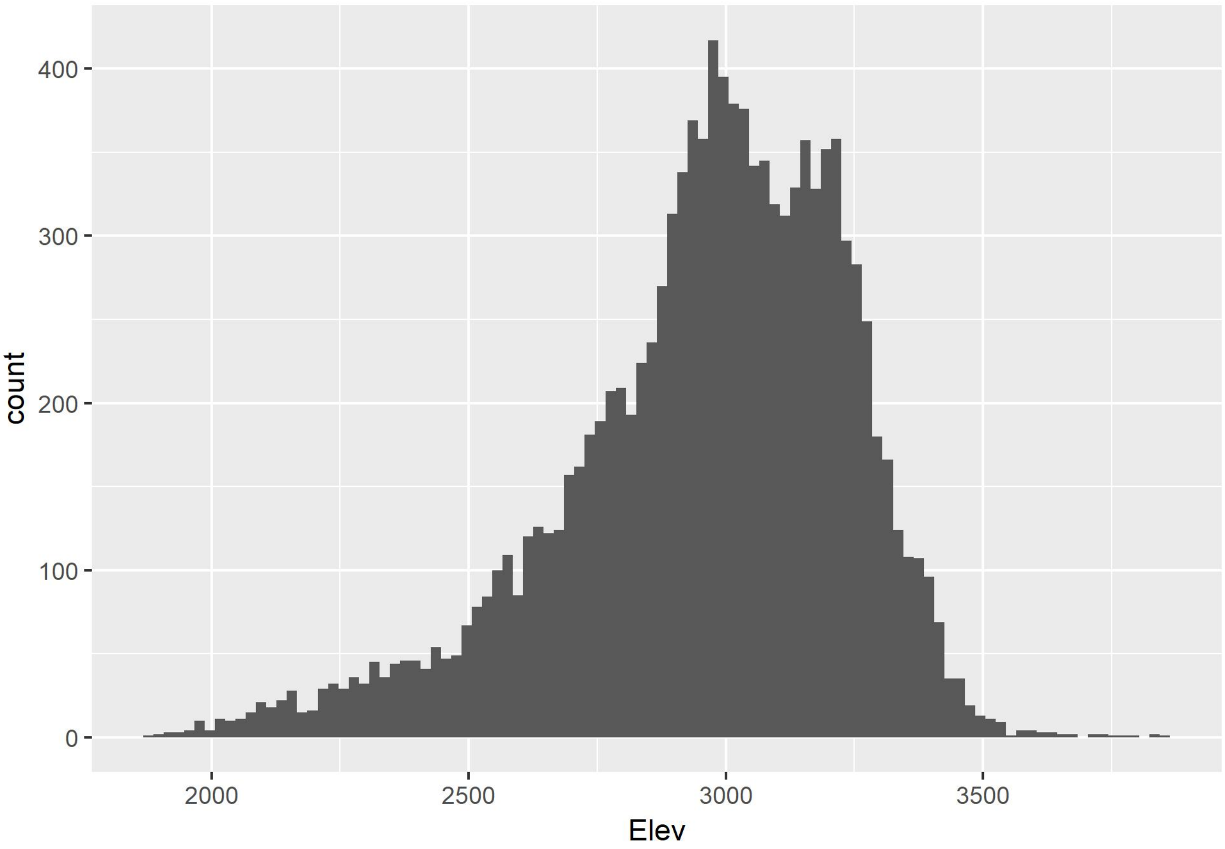

Figure 2: Elevation Histogram 2

Aspect - Figure 3

```
# plot 1
  jpeg(filename="ExpFigure03.jpg")
  plot(table(forestcover$Aspect))
  dev.off()
```

```
## pdf
##   2
```

The distribution of the Aspect follows the full 360 degrees of the compass.

Aspect Histogram 2 - Figure 2

```
# Figure 32
  g <- ggplot(forestcover,aes(Aspect)) +
        geom_histogram(bins=200) # +
       # facet_grid(. ~ factor(CovName)) +
  ggsave("ExpFigure04.jpg")
```

```
## Saving 6.5 x 4.5 in image
```

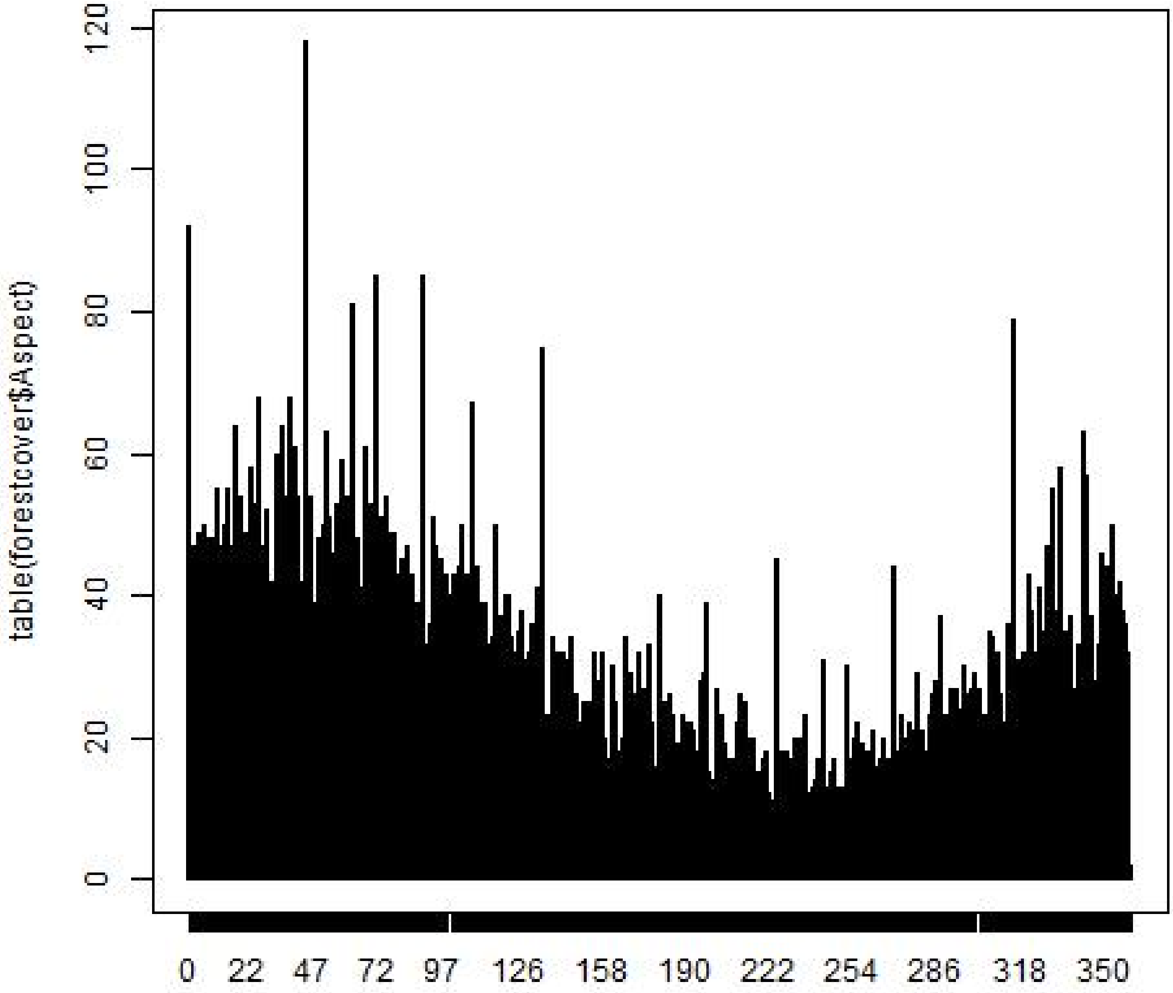

Figure 3: Aspect Histogram 1

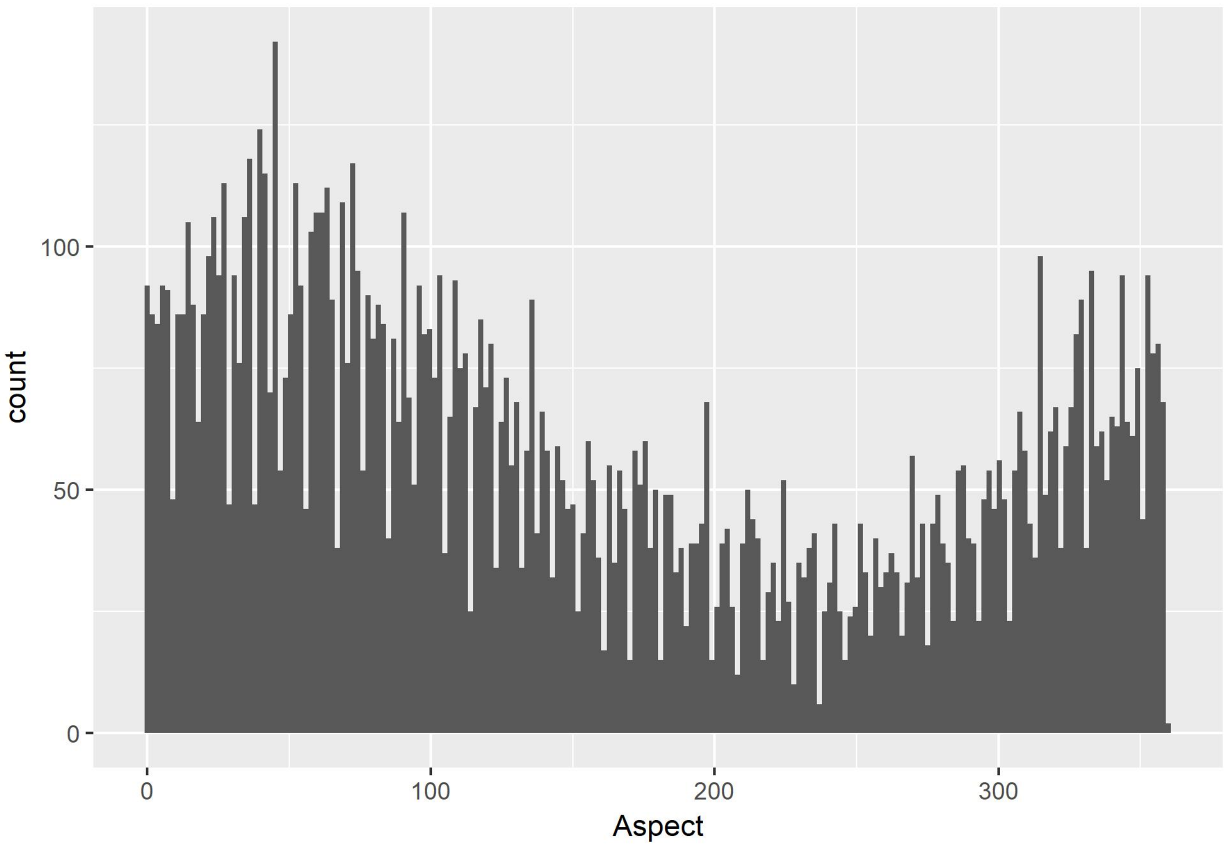

Figure 4: Aspect Histogram 2

Slope - Figure 5

```
# plot 1
  jpeg(filename="ExpFigure05.jpg")
  plot(table(forestcover$Slope))
  dev.off()
```

```
## pdf
##   2
```

The distribution of the slope seems reasonable.

Slope Histogram 2 - Figure 6

```
# Figure 32
  g <- ggplot(forestcover,aes(Slope)) +
        geom_histogram(bins=60) # +
       # facet_grid(. ~ factor(CovName)) +
  ggsave("ExpFigure06.jpg")
```

```
## Saving 6.5 x 4.5 in image
```

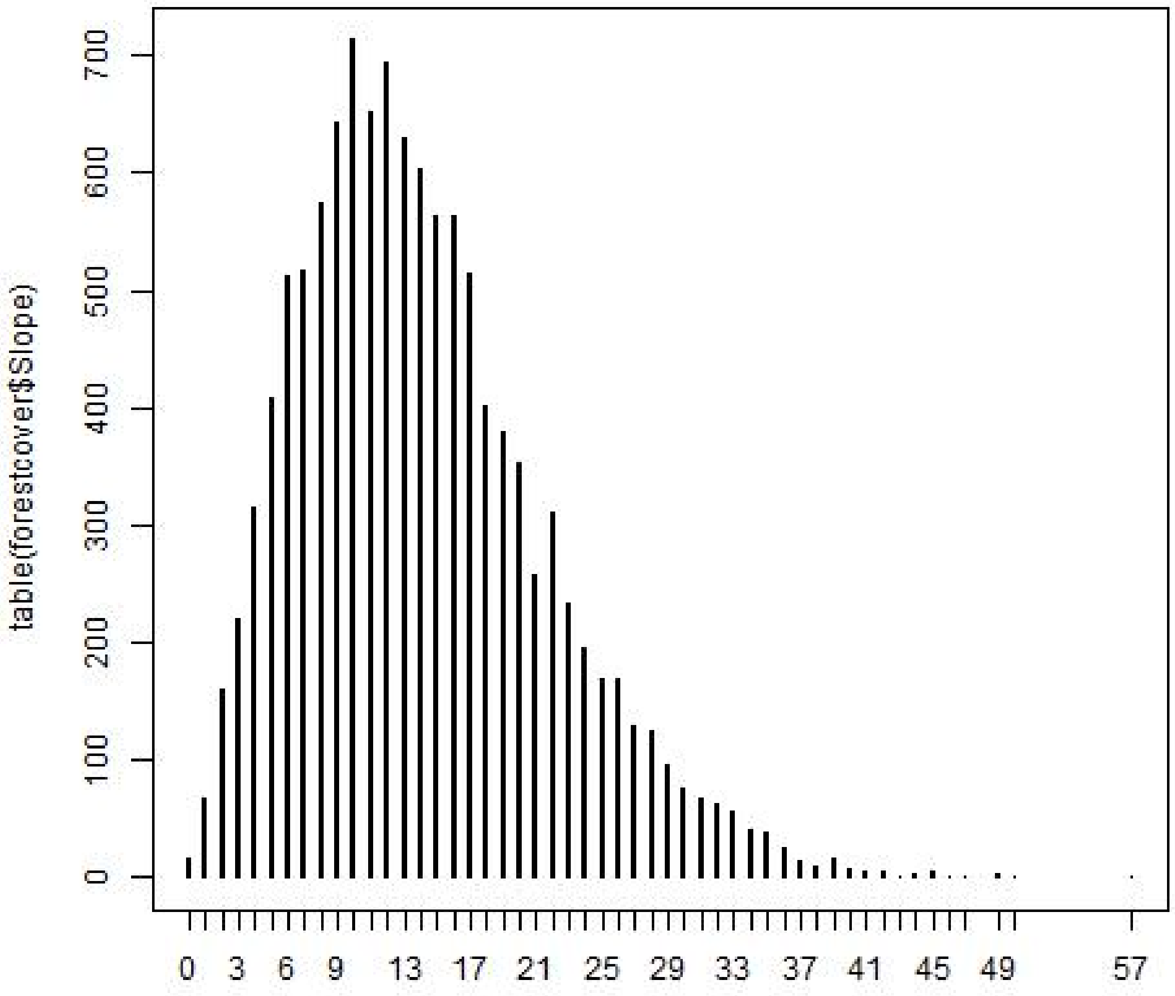

Figure 5: Slope Histogram 1

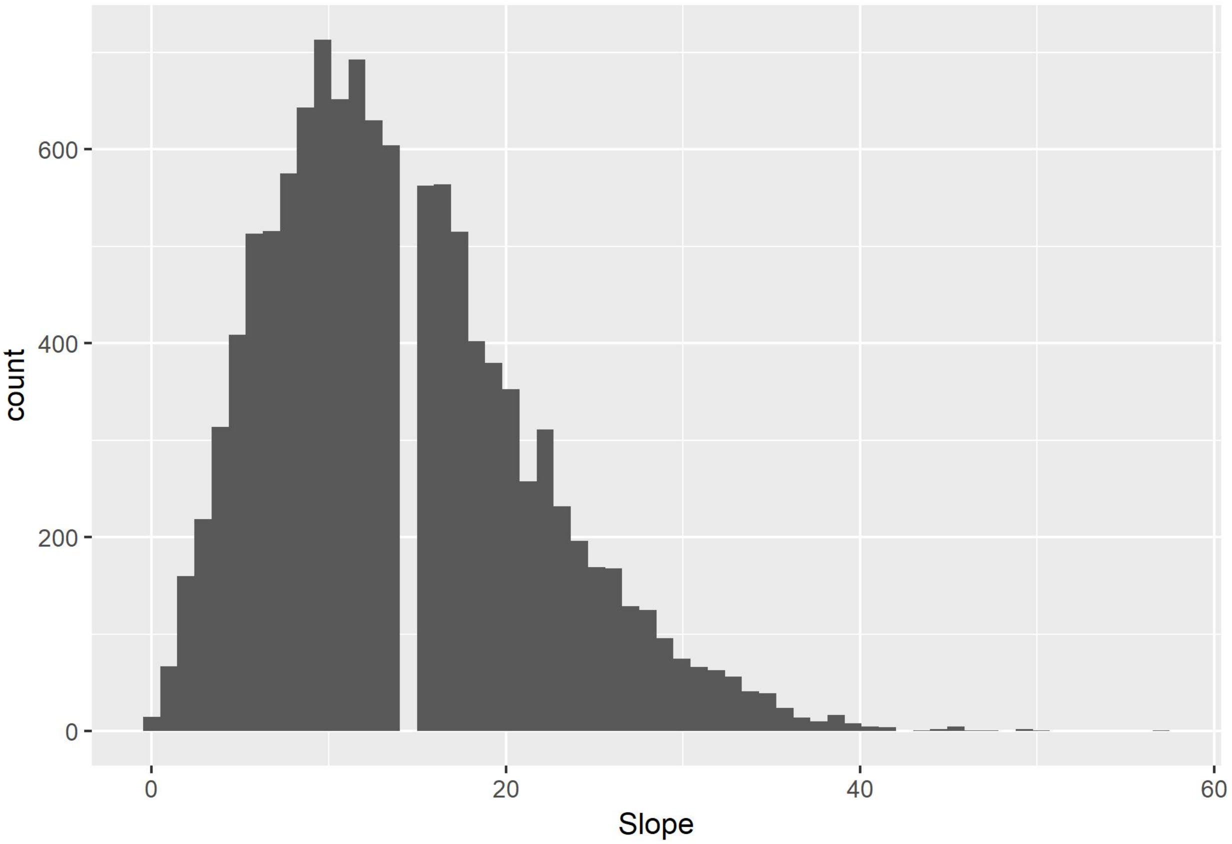

Figure 6: Slope Histogram 2

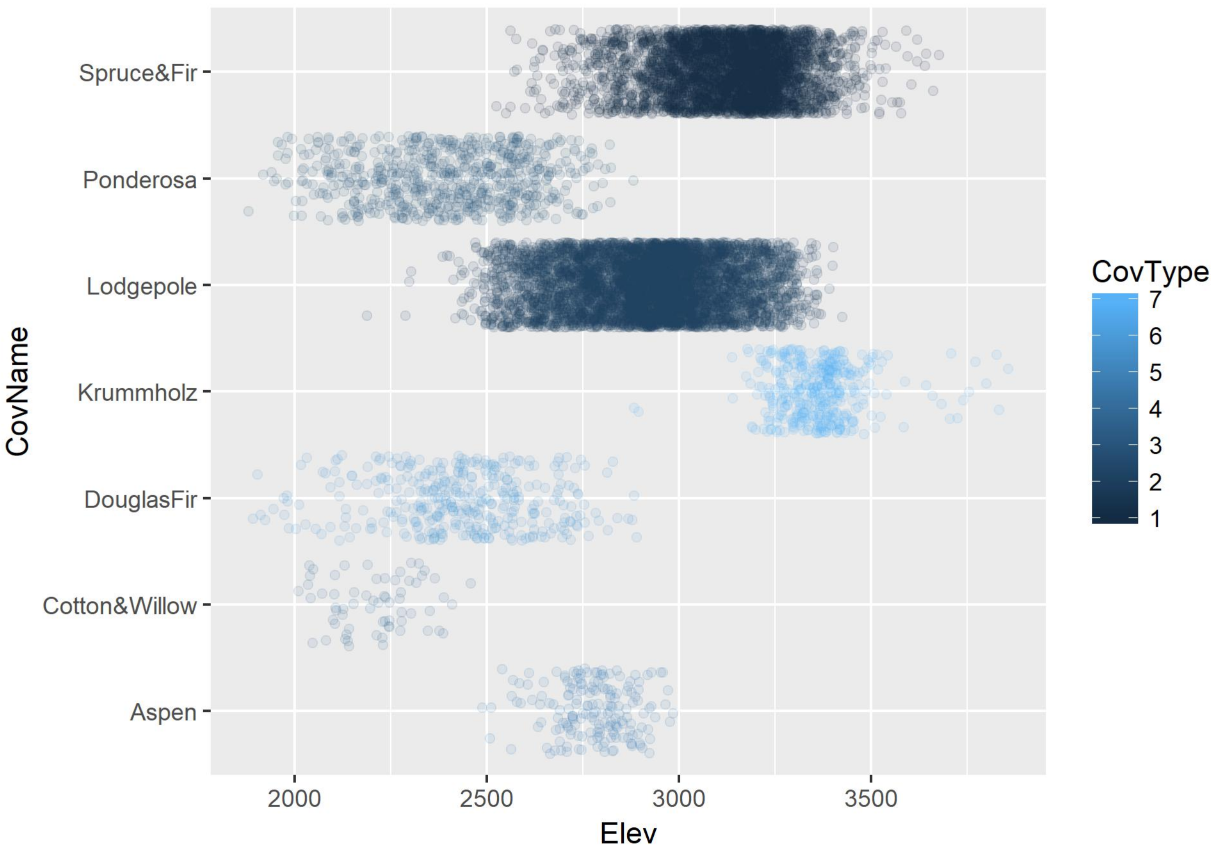

Figure 7: Plot 11

Tree Type vs Elevation - Plot 11

```
# plot 11
  g <- ggplot(forestcover,aes(Elev, CovName,col=CovType)) +
        geom_jitter(alpha=alphaVal)
  ggsave("plot11.jpg")
```

```
## Saving 6.5 x 4.5 in image
```

Elevation vs Tree Type shows that trees reside in a range of elevations and will help in determining tree type, but more information will be needed where there is overlap in elevation.

This graph looks a little strange. The next graph reverses the axes.

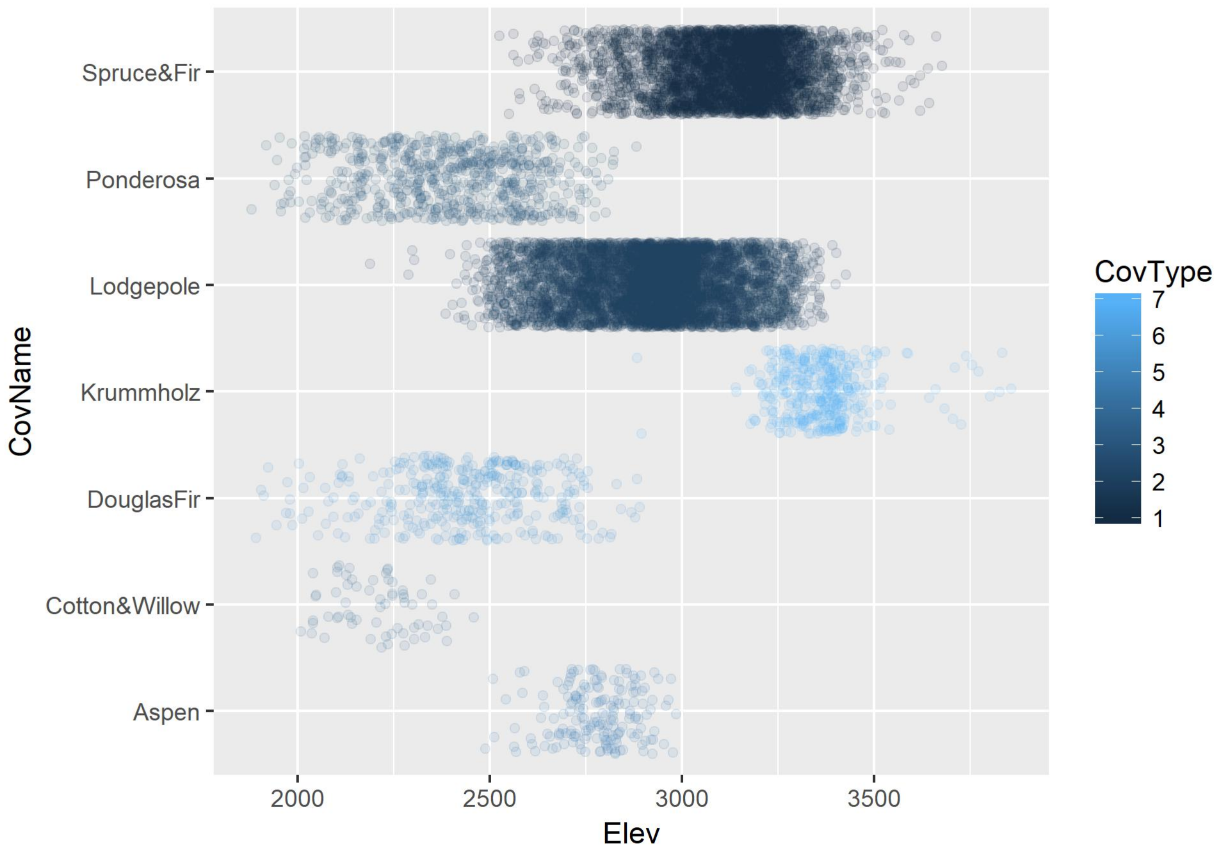

Figure 8: Plot 13

13

```
# plot 13
  g <- ggplot(forestcover,aes(Elev, CovName,col=CovType)) +
        geom_jitter(alpha=alphaVal)
  ggsave("plot13.jpg")
```

```
## Saving 6.5 x 4.5 in image
```

blah blah blah blah

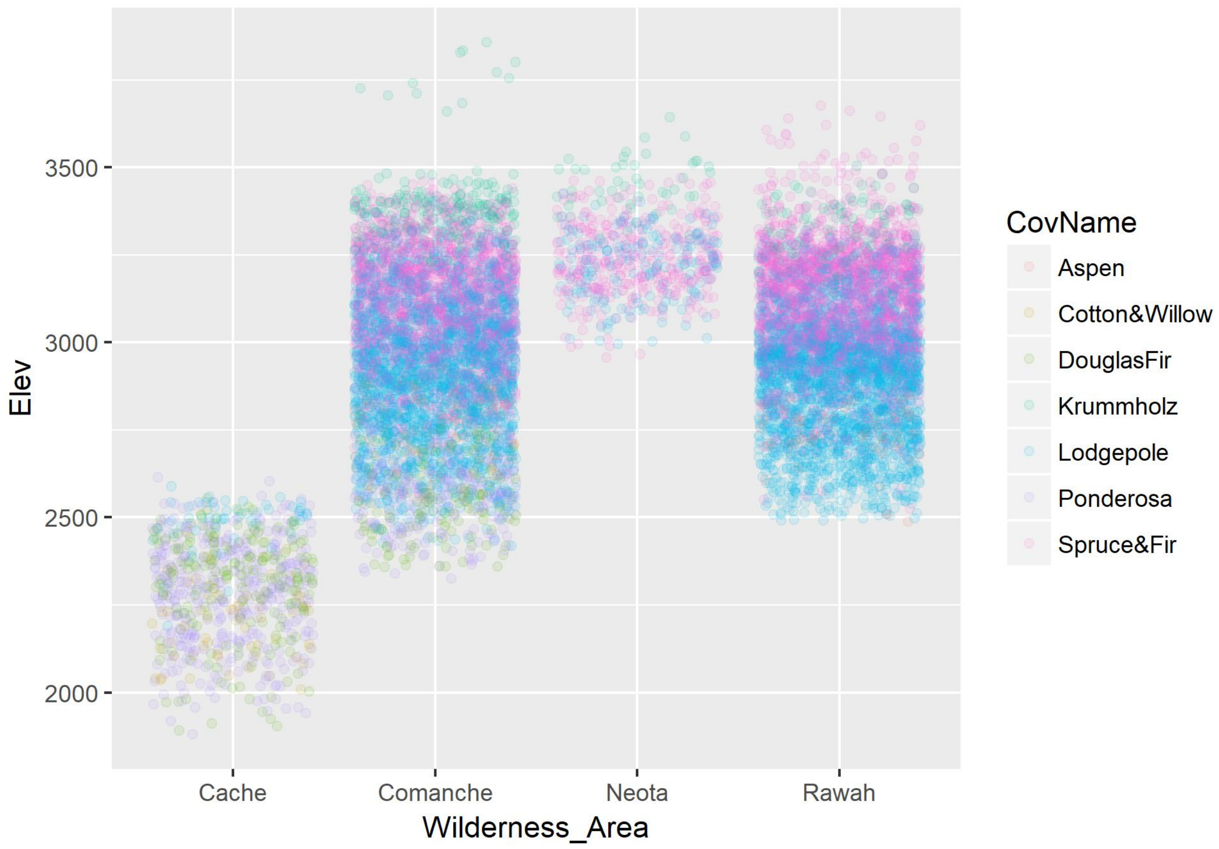

Figure 9: Elevation vs Wilderness Area with Tree Type

Elevation vs Wilderness Area with Tree Type - Figure 32

```
# Figure 32
  g <- ggplot(forestcover,aes(Wilderness_Area,Elev,col=CovName)) +
       geom_jitter(alpha=0.1) # +
      # facet_grid(. ~ factor(CovName)) +
  ggsave("Figure32.jpg")
```

```
## Saving 6.5 x 4.5 in image
```

Elevation vs Wilderness area shows the wilderness area should be able to help classifying tree type.