Capstone Data Wrangle Assignment

Tom Thorpe

March 28, 2018

Objective

Do the initial data cleanup of capstone project forest coverage data file.

Include required libraries.

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

```
## [1] "R script started at 2018-04-02 21:31:57"
```

```
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(knitr)
```

Point to data. The forestcover.csv is the raw data that is to be cleaned. The AlternatenateCoding02.csv is the file containing data to expand soil types into their individual components. The forest_clean_full.csv file contains the cleaned data with the orignial soil type columns included. The forest_clean.csv file is the cleaned data with unneeded columns removed.

```
cleanMethod <- 2
infile="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcover.csv"
#infile="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestsmall.csv"
transformfile="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/AlternateCoding02.csv"
out1file="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcover_clean_full.csv"
out2file="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcover_clean.csv"
#out1file="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcoversmall_clean_full.csv"
#out2file="C:/Users/Tom/git/datasciencefoundation/ForestCoverage/forestcoversmall_clean.csv"
```

Load the data.

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

```
## [1] "Data load started at 2018-04-02 21:31:58"
```

```
forestcover <- read.csv(infile,header=TRUE,sep=",") %>% tbl_df()
xform <- read.csv(transformfile,header=TRUE,sep=",") %>% tbl_df()
endTime=Sys.time()
  print(paste("Data load completed at",endTime))
```

```
## [1] "Data load completed at 2018-04-02 21:32:18"
```

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

```
## [1] "Elapsed time= 20.1791088581085 seconds."
```

Quick peek at the data

```
glimpse(forestcover)
```

```
## Observations: 581,012
## Variables: 55
## $ Elev      <int> 2596, 2590, 2804, 2785, 2595, 2579, 2606, 2605, 2617...
## $ Aspect    <int> 51, 56, 139, 155, 45, 132, 45, 49, 45, 59, 201, 151,...
## $ Slope     <int> 3, 2, 9, 18, 2, 6, 7, 4, 9, 10, 4, 11, 22, 7, 4, 7, ...
## $ H2OHD     <int> 258, 212, 268, 242, 153, 300, 270, 234, 240, 247, 18...
## $ H2OVD     <int> 0, -6, 65, 118, -1, -15, 5, 7, 56, 11, 51, 26, 69, 4...
## $ RoadHD    <int> 510, 390, 3180, 3090, 391, 67, 633, 573, 666, 636, 7...
## $ Shade9AM  <int> 221, 220, 234, 238, 220, 230, 222, 222, 223, 228, 21...
## $ Shade12PM <int> 232, 235, 238, 238, 234, 237, 225, 230, 221, 219, 24...
## $ Shade3PM  <int> 148, 151, 135, 122, 150, 140, 138, 144, 133, 124, 16...
## $ FirePtHD  <int> 6279, 6225, 6121, 6211, 6172, 6031, 6256, 6228, 6244...
## $ RWwild    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ NEwild    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ CMwild    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ CPwild    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST01      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST02      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST03      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST04      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST05      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST06      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST07      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST08      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST09      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST10      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST11      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST12      <int> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST13      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST14      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST15      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST16      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0...
## $ ST17      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST18      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1...
## $ ST19      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST20      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST21      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST22      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST23      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST24      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST25      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST26      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST27      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST28      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST29      <int> 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0...
```

```
## $ ST30      <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0...
## $ ST31      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST32      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST33      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST34      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST35      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST36      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST37      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST38      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST39      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST40      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ CovType   <int> 5, 5, 2, 2, 5, 2, 5, 5, 5, 5, 5, 2, 2, 5, 5, 5, 5, 5...
```

The forest cover data has a row for each sample representing a 30 meter by 30 meter square area of land. Each cell sample is described by elevation, slope and direction the cell faces, distance to water, roads, and fire and binary columns for wilderness area and soil type. One of 4 possible winderness areas and one of 40 possible aggregated soil types are set in each row. The predicted variable is the coverage type indicating 1 of 7 possible trees found in the cell sample.

The data is described in detail here: https://archive.ics.uci.edu/ml/machine-learning-databases/covtype/covtype.info. The data names have been abbreviated but can be related to the data descriptions easily.

```
#apply(forestcover,2,table)
glimpse(xform)
```

```
## Observations: 40
## Variables: 62
## $ ST                      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,...
## $ USFS_Code               <int> 2702, 2703, 2704, 2705, 2706, 2717, 35...
## $ Description             <fct> Cathedral family - Rock outcrop comple...
## $ Montane_low             <int> 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,...
## $ Montane                 <int> 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,...
## $ Subalpine               <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Alpine                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Dry                     <int> 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,...
## $ Non_Dry                 <int> 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,...
## $ Alluviam                <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Glacial                 <int> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,...
## $ Sed_mix                 <int> 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,...
## $ Ign_Meta                <int> 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1,...
## $ Aquolis_cmplx           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Argiborolis_Pachic      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Borohemists_cmplx       <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bross                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bullwark                <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,...
## $ Bullwark_Cmplx          <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,...
## $ Catamount               <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1,...
## $ Catamount_cmplx         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cathedral               <int> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Como                    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquepts_cmplx        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquepts_Typic        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls              <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls_cmplx        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls_Typic        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls_Typic_cmplx  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
```

```
## $ Cryoborolis_cmplx       <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryorthents             <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryorthents_cmplx       <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryumbrepts             <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryumbrepts_cmplx       <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Gateview                <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Gothic                  <int> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,...
## $ Granile                 <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Haploborolis            <int> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Legault                 <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,...
## $ Legault_cmplx           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Leighcan                <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Leighcan_cmplx          <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Leighcan_warm           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Moran                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Ratake                  <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Ratake_cmplx            <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rogert                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Supervisor_Limber_cmplx <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,...
## $ Troutville              <int> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,...
## $ Unspecified             <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Vanet                   <int> 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0,...
## $ Wetmore                 <int> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,...
## $ Bouldery_ext            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rock_Land               <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,...
## $ Rock_Land_cmplx         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,...
## $ Rock_Outcrop            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rock_Outcrop_cmplx      <int> 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0,...
## $ Rubbly                  <int> 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1,...
## $ Stony                   <int> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0,...
## $ Stony_extreme           <int> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Stony_very              <int> 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,...
## $ Till_Substratum         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
```

The xform data is used to break out the aggregated soil types into their individual components. For example, soil type 33 is described as:

Leighcan - Catamount families - Rock outcrop complex, extremely stony in the subalpine climate zone and the igneous and metamorphic geologic zone.

The xform data for soil type 33 will have binary column values set to 1 for the 'Leighcan' soil family, the 'Catamount' soil family, the 'Rock outcrop complex' rock density, the 'extremely stony' rock density, the 'subalpine' climate and 'igneous and metamorphic' geologic columns.

The columns in the xform file (except the first three descriptive columns) will be copied to the forest cover data frame and values set according to the soil type in the cell sample. I am interested to see if breaking out the soil type into it's components will provide greater predictability than with a single soil type indicator.

Two clean methods are discussed. Two methods are discussed because the first method took over 14 hours to run against the 581012 rows. I did not want to wait 14 hours for each test run to complete. I was able to use more efficient methods to reduce the run time to 6 minutes with most of that time spent reading and writing the data files.

The data clean up steps are described next.

Check Data for valid Ranges and Missing Values

First, the non-binary data is checked for valid ranges.

```
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 1859  2959.36530054457 3858
## 2    Aspect    0  155.656807432549  360
## 3     Slope    0  14.1037035379648   66
## 4     H2OHD    0  269.428216628916 1397
## 5     H2OVD -173   46.418855376481  601
## 6    RoadHD    0  2350.14661142971 7117
## 7  FirePtHD    0    1980.291226343 7173
## 8  Shade9AM    0  212.146048618617  254
## 9  Shade12P    0  223.318716308785  254
## 10 Shade3PM    0   142.52826275533  254
## 11   RWwild    0 0.448865083681576    1
## 12   NEwild    0 0.051434393781884    1
## 13   CMwild    0 0.436073609495157    1
## 14   CPwild    0 0.063626913041383    1
```

The results show all the data values have reasonable values and there is no missing data. The elevation ranges from 1859 meters (6099 feet) to 3858 meters (12657 feet). These are valid ranges for elevation in the Colorado wilderness areas being sampled, but the rule of thumb for timberline (the maximum elevation for where trees are found) is 11500 feet. It might be interesting to see how accurate predictions are if samples above 11800 feet are removed.

The Aspect which is the compass heading that the terrain faces, ranges from 0 to 360 degrees and is a valid data range. The Slope is the steepness of the terrain with 0 degrees being flat and 90 degrees being vertical. The maximum Slope was found to be 66 degrees which seems logical since trees are not usually seen on near-vertical cliffs. (It's a different story in New Zealand!)

The horizontal distance to the nearest water features, range from 0 to 1397 meters which seems reasonable. The vertical distance to nearest water features, range from -173 to 601 meters which seems reasonable and can be negative since the nearest water may be below the forest cover data sample.

The horizontal distance to the nearest raod ranges from 0 to 7117 meters which is reasonable. The horizontal distance to the nearest fire features range from 0 to 7173 meters which is reasonable. The amount of shade

present in a cell sample at 9AM, 12PM and 3PM ranges from 0 (full sun) to 254 (fully shaded).

Check Soil Type encoding

Check the binary data to ensure multiple columns have not been selected. Starting with Soil Type, check that there is no more than one ST___ column set to 1 in each row.

First, find the location/index of the "ST01" column in the forest coverage data frame. There are 40 soil types in the forest coverage data frame, so the last column index is found by adding 39 to the index of the "ST01" column.

```
firstIndex<-grep("^ST01$", colnames(forestcover))
lastIndex<-firstIndex+39
print(paste("first=",firstIndex,", last=",lastIndex))
```

```
## [1] "first= 15 , last= 54"
```

Check each row to be sure there is only one ST___ column value set to 1.

```
if(cleanMethod==1) {
rowschecked<-0
errorCnt<-0
system.time({
  for(i in 1:nrow(forestcover)) {
    rowschecked<-rowschecked+1
    # find Soil type in this row
    soilIndex<-0    # index variable that will be used to find the data in the xform data frame
    soilTypeCnt<-0  # error check var: There should be exactly 1 soil type in the forest cover data fra

    # check each "ST__" column in the forest coverage data frame for a "1" indicating the soil type
    for (j in firstIndex:lastIndex) { # first & last indicies of the soil types were calculated above
      if(forestcover[i,j]==1) {       # checking if a soil type was found. If "1", we have found one
        soilIndex=j-firstIndex+1      # calculate the soil index to be used by the xform data frame
        soilTypeCnt<-soilTypeCnt+1    # count how many soil types we find. Should only be one
      }
    }

    # report any errors
    if(soilIndex==0) {
      print(paste("Soil type missing in row",i))
      errorCnt <- errorCnt + 1
    }
    if(soilTypeCnt > 1) {
      print(paste("Too many soil types in row",i))
      errorCnt <- errorCnt + 1
    }
  }
})
print(paste(errorCnt,"errors found in ST__ columns"))
print(paste(rowschecked,"Rows Checked"))
}
```

The above method took 76 minutes to run against the ~580,000 rows. The more efficient method shown below took a second to run. I was surprised that referencing individual cells is as expensive as setting values

in individual cells. The code below was verified by altering test data to have a row with no columns selected and several rows with multiple columns selected.

```
if(cleanMethod==2) {
  system.time({
    STcols=c("ST01","ST02","ST03","ST04","ST05","ST06","ST07","ST08","ST09","ST10",
             "ST11","ST12","ST13","ST14","ST15","ST16","ST17","ST18","ST19","ST20",
             "ST21","ST22","ST23","ST24","ST25","ST26","ST27","ST28","ST29","ST30",
             "ST31","ST32","ST33","ST34","ST35","ST36","ST37","ST38","ST39","ST40"
             )
    forestcover <- mutate(forestcover,STsum=rowSums(forestcover[,STcols]))
    myranges("STsum",forestcover$STsum)

    # print the row numbers where the STsum is not 1
    which(forestcover$STsum!=1)
  })
}
```

```
##    user  system elapsed
##    0.25    0.03    0.28
```

The soil type data is clean. There are no rows where the Soil Type columns do not have exactly one column set to 1.

Using a the same method to check the soil type, the Wilderness indicators will be checked to ensure only one column has a value of 1.

```
system.time({
    Wildcols=c("RWwild","NEwild","CMwild","CPwild")
    forestcover <- mutate(forestcover,Wildsum=rowSums(forestcover[,Wildcols]))
    myranges("wildsum",forestcover$Wildsum)

    # print the row numbers where the STsum is not 1
    which(forestcover$Wildsum!=1)
  })
```

```
##    user  system elapsed
##    0.06    0.00    0.06
```

The data is clean. There are no rows where the Wilderness columns do not have exactly one column set to 1.

Expand Soil type

The soil type needs to be expanded into columns that comprise the different components of each soil type. A particular soil type represents the climate zone, geologic zone, one or more soil families and one or more rock densities for a given sample/row.

The xform data frame drives the conversion of soil type to additional columns noted above. Each row in the xform data frame corresponds to one of the 40 possible soil types in the forest coverage data frame. The first 3 columns identify the soil type number that corresponds to the soil type column in the forest coverage data frame, a US Forest Service soil code and a description of the soil families and rock densities for the soil type. The remaining columns are the columns that will be added to the forest coverage data frame and the values for each column. The columns added to the forest column data frame correspond to the individual components in the description of each soil type.

I created the xform data frame to reflect what I think would be a good way to break out the soil type. I may want to change the way the data is broken out and therefore want later changes to data encoding to be easy

to implement by using the xform data frame to be the source of column names and values with minimum hard coding involved.

The xform data frame is sorted by ST, the soil type column, so that it can be indexed by the soil type index that will be calculated from the forest coverage soil type columns data.

```
xform<-arrange(xform,ST)
xform
```

```
## # A tibble: 40 x 62
##       ST USFS_Code Description  Montane_low Montane Subalpine Alpine   Dry
##    <int>     <int> <fct>              <int>   <int>     <int>  <int> <int>
##  1     1      2702 Cathedral f~           1       0         0      0     0
##  2     2      2703 Vanet - Rat~           1       0         0      0     0
##  3     3      2704 Haploboroli~           1       0         0      0     0
##  4     4      2705 Ratake fami~           1       0         0      0     0
##  5     5      2706 Vanet famil~           1       0         0      0     0
##  6     6      2717 Vanet - Wet~           1       0         0      0     0
##  7     7      3501 Gothic fami~           0       1         0      0     1
##  8     8      3502 Supervisor ~           0       1         0      0     1
##  9     9      4201 Troutville ~           0       1         0      0     0
## 10    10      4703 Bullwark - ~           0       1         0      0     0
## # ... with 30 more rows, and 54 more variables: Non_Dry <int>,
## #   Alluviam <int>, Glacial <int>, Sed_mix <int>, Ign_Meta <int>,
## #   Aquolis_cmplx <int>, Argiborolis_Pachic <int>,
## #   Borohemists_cmplx <int>, Bross <int>, Bullwark <int>,
## #   Bullwark_Cmplx <int>, Catamount <int>, Catamount_cmplx <int>,
## #   Cathedral <int>, Como <int>, Cryaquepts_cmplx <int>,
## #   Cryaquepts_Typic <int>, Cryaquolls <int>, Cryaquolls_cmplx <int>,
## #   Cryaquolls_Typic <int>, Cryaquolls_Typic_cmplx <int>,
## #   Cryoborolis_cmplx <int>, Cryorthents <int>, Cryorthents_cmplx <int>,
## #   Cryumbrepts <int>, Cryumbrepts_cmplx <int>, Gateview <int>,
## #   Gothic <int>, Granile <int>, Haploborolis <int>, Legault <int>,
## #   Legault_cmplx <int>, Leighcan <int>, Leighcan_cmplx <int>,
## #   Leighcan_warm <int>, Moran <int>, Ratake <int>, Ratake_cmplx <int>,
## #   Rogert <int>, Supervisor_Limber_cmplx <int>, Troutville <int>,
## #   Unspecified <int>, Vanet <int>, Wetmore <int>, Bouldery_ext <int>,
## #   Rock_Land <int>, Rock_Land_cmplx <int>, Rock_Outcrop <int>,
## #   Rock_Outcrop_cmplx <int>, Rubbly <int>, Stony <int>,
## #   Stony_extreme <int>, Stony_very <int>, Till_Substratum <int>
```

Create a new column to retain the soil type as a number.

```
forestcover <- mutate(forestcover,SoilType = 0)
glimpse(forestcover)
```

```
## Observations: 581,012
## Variables: 58
## $ Elev      <int> 2596, 2590, 2804, 2785, 2595, 2579, 2606, 2605, 2617...
## $ Aspect    <int> 51, 56, 139, 155, 45, 132, 45, 49, 45, 59, 201, 151,...
## $ Slope     <int> 3, 2, 9, 18, 2, 6, 7, 4, 9, 10, 4, 11, 22, 7, 4, 7, ...
## $ H2OHD     <int> 258, 212, 268, 242, 153, 300, 270, 234, 240, 247, 18...
## $ H2OVD     <int> 0, -6, 65, 118, -1, -15, 5, 7, 56, 11, 51, 26, 69, 4...
## $ RoadHD    <int> 510, 390, 3180, 3090, 391, 67, 633, 573, 666, 636, 7...
## $ Shade9AM  <int> 221, 220, 234, 238, 220, 230, 222, 222, 223, 228, 21...
## $ Shade12PM <int> 232, 235, 238, 238, 234, 237, 225, 230, 221, 219, 24...
## $ Shade3PM  <int> 148, 151, 135, 122, 150, 140, 138, 144, 133, 124, 16...
```

```
## $ FirePtHD  <int> 6279, 6225, 6121, 6211, 6172, 6031, 6256, 6228, 6244...
## $ RWwild    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ NEwild    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ CMwild    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ CPwild    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST01      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST02      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST03      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST04      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST05      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST06      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST07      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST08      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST09      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST10      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST11      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST12      <int> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST13      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST14      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST15      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST16      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0...
## $ ST17      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST18      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1...
## $ ST19      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST20      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST21      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST22      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST23      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST24      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST25      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST26      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST27      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST28      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST29      <int> 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0...
## $ ST30      <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0...
## $ ST31      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST32      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST33      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST34      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST35      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST36      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST37      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST38      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST39      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ ST40      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ CovType   <int> 5, 5, 2, 2, 5, 2, 5, 5, 5, 5, 5, 2, 2, 5, 5, 5, 5, 5...
## $ STsum     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ Wildsum   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ SoilType  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
```

```
table(forestcover$SoilType)
```

```
##
##      0
## 581012
```

We want to add the same soil data column names in the xform data frame to the forest coverage data frame. Unfortunately I was not able to find a way to use a variable name to assign the name of the new column when using the *mutate()* function. We can use the *colnames* function to change the column name after each mutate operation adds a column to the forest coverage data frame.

Start by getting the current forest coverage column names and creating an empty vector to collect the xform data frame column names.

```
xformcnames=c()
forestcnames=colnames(forestcover)
```

Next iterate through the column names in the xform data frame. The first three columns in the xform data frame are not to be added to the coverage data frame. These column names are skipped by checking for the column names as shown in the first *if* statement below. This is the only hard coding required for method 1 and the only requirement of column layout in the xform data frame.

For every other xform data frame column, a column is added to the forest coverage data frame. The xform column name is added to the xform data frame column names and the forest service column names vectors. After the column is added to the forest coverage data frame, the forest service column names vector will be used to reset the new column name. The xform column name vector will be used later to index both the xform and forest coverage data frames when setting values in the forest coverage data frame.

```
startTime=Sys.time()
print(paste("Column name creation started at",startTime))
```

```
## [1] "Column name creation started at 2018-04-02 21:32:21"
```

```
for(colname in colnames(xform)){
  if (colname!="ST" & colname !="USFS_Code" & colname != "Description")
  {
    #print(colname)
    forestcover <- mutate(forestcover,colname = 0) # add column named "colname" to forestcover
    forestcnames<-c(forestcnames,colname)          # add the actual column name to forest colnames vect
    colnames(forestcover) <- forestcnames          # set the forest cover column names
    xformcnames=c(xformcnames,colname)             # add the column name to xform column names vector
  }
}
endTime=Sys.time()
print(paste("Column name creation completed at",endTime))
```

```
## [1] "Column name creation completed at 2018-04-02 21:32:21"
```

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

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

```
print(xformcnames)
```

```
##  [1] "Montane_low"             "Montane"
##  [3] "Subalpine"               "Alpine"
##  [5] "Dry"                     "Non_Dry"
##  [7] "Alluviam"                "Glacial"
##  [9] "Sed_mix"                 "Ign_Meta"
## [11] "Aquolis_cmplx"           "Argiborolis_Pachic"
## [13] "Borohemists_cmplx"       "Bross"
## [15] "Bullwark"                "Bullwark_Cmplx"
## [17] "Catamount"               "Catamount_cmplx"
## [19] "Cathedral"               "Como"
## [21] "Cryaquepts_cmplx"        "Cryaquepts_Typic"
```

```
## [23] "Cryaquolls"               "Cryaquolls_cmplx"
## [25] "Cryaquolls_Typic"         "Cryaquolls_Typic_cmplx"
## [27] "Cryoborolis_cmplx"        "Cryorthents"
## [29] "Cryorthents_cmplx"        "Cryumbrepts"
## [31] "Cryumbrepts_cmplx"        "Gateview"
## [33] "Gothic"                   "Granile"
## [35] "Haploborolis"             "Legault"
## [37] "Legault_cmplx"            "Leighcan"
## [39] "Leighcan_cmplx"           "Leighcan_warm"
## [41] "Moran"                    "Ratake"
## [43] "Ratake_cmplx"             "Rogert"
## [45] "Supervisor_Limber_cmplx" "Troutville"
## [47] "Unspecified"              "Vanet"
## [49] "Wetmore"                  "Bouldery_ext"
## [51] "Rock_Land"                "Rock_Land_cmplx"
## [53] "Rock_Outcrop"             "Rock_Outcrop_cmplx"
## [55] "Rubbly"                   "Stony"
## [57] "Stony_extreme"            "Stony_very"
## [59] "Till_Substratum"
```

```
glimpse(forestcover)
```

```
## Observations: 581,012
## Variables: 117
## $ Elev                   <int> 2596, 2590, 2804, 2785, 2595, 2579, 26...
## $ Aspect                 <int> 51, 56, 139, 155, 45, 132, 45, 49, 45,...
## $ Slope                  <int> 3, 2, 9, 18, 2, 6, 7, 4, 9, 10, 4, 11,...
## $ H2OHD                  <int> 258, 212, 268, 242, 153, 300, 270, 234...
## $ H2OVD                  <int> 0, -6, 65, 118, -1, -15, 5, 7, 56, 11,...
## $ RoadHD                 <int> 510, 390, 3180, 3090, 391, 67, 633, 57...
## $ Shade9AM               <int> 221, 220, 234, 238, 220, 230, 222, 222...
## $ Shade12PM              <int> 232, 235, 238, 238, 234, 237, 225, 230...
## $ Shade3PM               <int> 148, 151, 135, 122, 150, 140, 138, 144...
## $ FirePtHD               <int> 6279, 6225, 6121, 6211, 6172, 6031, 62...
## $ RWwild                 <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ NEwild                 <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CMwild                 <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CPwild                 <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST01                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST02                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST03                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST04                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST05                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST06                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST07                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST08                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST09                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST10                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST11                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST12                   <int> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST13                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST14                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST15                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST16                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST17                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
```

```
## $ ST18                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ ST19                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST20                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST21                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST22                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST23                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST24                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST25                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST26                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST27                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST28                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST29                   <int> 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,...
## $ ST30                   <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1,...
## $ ST31                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST32                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST33                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST34                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST35                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST36                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST37                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST38                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST39                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST40                   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CovType                <int> 5, 5, 2, 2, 5, 2, 5, 5, 5, 5, 5, 2, 2,...
## $ STsum                  <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ Wildsum                <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ SoilType               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Montane_low            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Montane                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Subalpine              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Alpine                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Dry                    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Non_Dry                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Alluviam               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Glacial                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Sed_mix                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Ign_Meta               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Aquolis_cmplx          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Argiborolis_Pachic     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Borohemists_cmplx      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bross                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bullwark               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bullwark_Cmplx         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Catamount              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Catamount_cmplx        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cathedral              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Como                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquepts_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquepts_Typic       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls_Typic       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls_Typic_cmplx <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryoborolis_cmplx      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
```

```
## $ Cryorthents             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryorthents_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryumbrepts             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryumbrepts_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Gateview                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Gothic                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Granile                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Haploborolis            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Legault                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Legault_cmplx           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Leighcan                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Leighcan_cmplx          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Leighcan_warm           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Moran                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Ratake                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Ratake_cmplx            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rogert                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Supervisor_Limber_cmplx <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Troutville              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Unspecified             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Vanet                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Wetmore                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bouldery_ext            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rock_Land               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rock_Land_cmplx         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rock_Outcrop            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rock_Outcrop_cmplx      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rubbly                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Stony                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Stony_extreme           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Stony_very              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Till_Substratum         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
```

The new columns have been added to the forest coverage data frame and the column values need to be populated based on the values in the xform data frame.

Two methods of expanding the soil types are shown below. The first method updates values in the forest cover table cell by cell. This method took 14.6 hours to run against the 581012 rows of forest coverage data. A second method was explored to try to see if the time could be reduced. By using a column update, the total time was reduced to 6 minutes with most of that time used to read and write the data to disk. This shows me that individual cell updates in R are very expensive and it is important to take advantage of R packages and functions that are designed to be more efficient.

The advantage of the first method is that no hard coding is required when column names change in the xform data frame. In the second method, a column select and assignment must be hard coded for each column name in the xform data frame that is to be replicated in the forest coverage data frame. This disadvantage was overcome by writing a script to create column select and assignment statements by reading the xform data frame and copying the generated code into the R script.

The first method is discussed next. In the first method, the column index of the first soil type in the forest coverage data frame is found. The soil type columns are labeled "ST01", "ST02", ..., "ST40", sequentially in order with no other data columns between them. A non-zero value in one of the columns indicates the soil type. There can only be one soil type set per row. Finding the column index with the non zero value and subtracting the index of "ST01" gives the row index into the xform data frame. The soil data from the corresponding soil type in the xform data frame are then copied to the forest coverage data frame cell by cell.

All the information needed to proceed with the data update for clean method 1 is now ready. The code comments describe the details to update the forest coverage data frame.

```
if (cleanMethod == 1) {
  system.time({
  startTime=Sys.time()
  print(paste("Starting soil type expansion at",startTime))
  errorCnt<- 0
  totalCnt<-0
  reportCnt<-0
  # go through every row in the forest coverage data frame
  for(i in 1:nrow(forestcover)) {
    reportCnt<- reportCnt + 1
    totalCnt <- totalCnt+1
    # find Soil type in this row
    soilIndex<-0    # index variable that will be used to find the data in the xform data frame
    soilTypeCnt<-0  # error check var: There should be exactly 1 soil type in the forest cover data fra

    # check each "ST__" column in the forest coverage data frame for a "1" indicating the soil type
    for (j in firstIndex:lastIndex) { # first and last indicies of the soil types were calculated above
      if(forestcover[i,j]==1) {       # checking if a soil type was found. If "1", we have found one
        soilIndex=j-firstIndex+1      # calculate the soil index to be used by the xform data frame
        soilTypeCnt<-soilTypeCnt+1    # count how many soil types we find. Should only be one
      }
    }

    # do some error checking to be sure there are no errors in the forest coverage data frame
    if(soilIndex==0) {
      print(paste("Soil type mising in row",i))
      errorCnt <- errorCnt + 1
    }
    if(soilTypeCnt > 1) {
      print(paste("Too many soil types in row",i))
      errorCnt <- errorCnt + 1
    }
    #print(paste("row:",i,"Elev=",forestcover[i,1]))

    if(soilIndex > 0){
      forestcover[i,"SoilType"] <- soilIndex # save the original soil type code

      # copy the soil type data from the xform data frame to the forest coverage data frame
      for (colname in xformcnames) {                          # use xform column names for indexing
        if (!is.na(xform[soilIndex,colname])) {               # don't copy NA values
          forestcover[i,colname] <- xform[soilIndex,colname]
          # current forest cover row get the soil data from the xform data frame
          # NOTE: individual cell updates in a data frame are very expensive
          #       probably due to memory management inefficiencies
        }
      }
    }

    if (reportCnt > 49999) {
      curTime <- Sys.time()
      print(paste(totalCnt," rows processed at",curTime,", elapsed time",round(curTime-startTime),"secs
```

```
      reportCnt <- 0
    }
  }
  endTime=Sys.time()
  print(paste("Forest soil type expansion completed at",endTime))
  print(paste("Elapsed time=",round(endTime-startTime),"seconds."))
  print(paste("Total rows processed:",totalCnt))
  print(paste(errorCnt,"errors were found."))
  })
}
```

Clean method 1 is not executed for the creation of the final report. The output from clean method 1 was saved from previous runs and compared with the output from clean method 2. The files were identical. In addition to manually checking a few rows for each of the soil types were being generated correctly, getting the same results from two different methods helped to verify that output was being generated correctly.

In clean method 2, each row in the xform data frame is processed. Each row represents a soil type. If a data column in the xform soil type is set to '1', the same data column should be set to '1' in the forest cover data frame, for the same soil type.

This code is much simpler and much faster taking only 6 seconds in a typical execution. There are total of 220 cells set to 1 in the xform data frame meaning there are 220 column update calls made by the script to set the data values in forest cover data. Adding in the 40 calls to set the soil type, there are a total of 260 column update calls made in clean method 2. With 40 rows in the xform data, there is an average of 5.5 cell updates per row in clean method 1. Adding in the update of the soil type in method one brings the average to 6.5 updates per row. With 581000 rows in the forest cover data, about 3.7 million cells are individually updated using method 1. The only disadvantage of clean method 2 is having to hard code the column update statements when the xform column names change, which should not be too often.

```
if (cleanMethod == 2) {
  startTime=Sys.time()
  print(paste("Starting soil type expansion at",startTime))

  # start by setting the Soil Type column in the forest cover data
  forestcover$SoilType[forestcover$ST01 == 1] <- 1
  forestcover$SoilType[forestcover$ST02 == 1] <- 2
  forestcover$SoilType[forestcover$ST03 == 1] <- 3
  forestcover$SoilType[forestcover$ST04 == 1] <- 4
  forestcover$SoilType[forestcover$ST05 == 1] <- 5
  forestcover$SoilType[forestcover$ST06 == 1] <- 6
  forestcover$SoilType[forestcover$ST07 == 1] <- 7
  forestcover$SoilType[forestcover$ST08 == 1] <- 8
  forestcover$SoilType[forestcover$ST09 == 1] <- 9
  forestcover$SoilType[forestcover$ST10 == 1] <- 10
  forestcover$SoilType[forestcover$ST11 == 1] <- 11
  forestcover$SoilType[forestcover$ST12 == 1] <- 12
  forestcover$SoilType[forestcover$ST13 == 1] <- 13
  forestcover$SoilType[forestcover$ST14 == 1] <- 14
  forestcover$SoilType[forestcover$ST15 == 1] <- 15
  forestcover$SoilType[forestcover$ST16 == 1] <- 16
  forestcover$SoilType[forestcover$ST17 == 1] <- 17
  forestcover$SoilType[forestcover$ST18 == 1] <- 18
  forestcover$SoilType[forestcover$ST19 == 1] <- 19
  forestcover$SoilType[forestcover$ST20 == 1] <- 20
  forestcover$SoilType[forestcover$ST21 == 1] <- 21
```

```
forestcover$SoilType[forestcover$ST22 == 1] <- 22
forestcover$SoilType[forestcover$ST23 == 1] <- 23
forestcover$SoilType[forestcover$ST24 == 1] <- 24
forestcover$SoilType[forestcover$ST25 == 1] <- 25
forestcover$SoilType[forestcover$ST26 == 1] <- 26
forestcover$SoilType[forestcover$ST27 == 1] <- 27
forestcover$SoilType[forestcover$ST28 == 1] <- 28
forestcover$SoilType[forestcover$ST29 == 1] <- 29
forestcover$SoilType[forestcover$ST30 == 1] <- 30
forestcover$SoilType[forestcover$ST31 == 1] <- 31
forestcover$SoilType[forestcover$ST32 == 1] <- 32
forestcover$SoilType[forestcover$ST33 == 1] <- 33
forestcover$SoilType[forestcover$ST34 == 1] <- 34
forestcover$SoilType[forestcover$ST35 == 1] <- 35
forestcover$SoilType[forestcover$ST36 == 1] <- 36
forestcover$SoilType[forestcover$ST37 == 1] <- 37
forestcover$SoilType[forestcover$ST38 == 1] <- 38
forestcover$SoilType[forestcover$ST39 == 1] <- 39
forestcover$SoilType[forestcover$ST40 == 1] <- 40
table(forestcover$SoilType)

errorCnt<- 0
totalCnt<-0
reportCnt<-0
# go through every row/soil type in the xform data frame to determine which columns
# in the forest coverage data frame must be set.
for(ndx in 1:nrow(xform)) {
  # if a property is set for the current soiltype (the ndx variable),
  #                                    set the same property in forest cover for that soil type
  if (xform[ndx,"Montane_low"] == 1) { forestcover$Montane_low[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Montane"] == 1) { forestcover$Montane[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Subalpine"] == 1) { forestcover$Subalpine[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Alpine"] == 1) { forestcover$Alpine[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Dry"] == 1) { forestcover$Dry[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Non_Dry"] == 1) { forestcover$Non_Dry[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Alluviam"] == 1) { forestcover$Alluviam[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Glacial"] == 1) { forestcover$Glacial[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Sed_mix"] == 1) { forestcover$Sed_mix[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Ign_Meta"] == 1) { forestcover$Ign_Meta[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Aquolis_cmplx"] == 1) { forestcover$Aquolis_cmplx[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Argiborolis_Pachic"] == 1) { forestcover$Argiborolis_Pachic[forestcover$SoilType == n
  if (xform[ndx,"Borohemists_cmplx"] == 1) { forestcover$Borohemists_cmplx[forestcover$SoilType == ndx
  if (xform[ndx,"Bross"] == 1) { forestcover$Bross[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Bullwark"] == 1) { forestcover$Bullwark[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Bullwark_Cmplx"] == 1) { forestcover$Bullwark_Cmplx[forestcover$SoilType == ndx] <-
  if (xform[ndx,"Catamount"] == 1) { forestcover$Catamount[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Catamount_cmplx"] == 1) { forestcover$Catamount_cmplx[forestcover$SoilType == ndx] <
  if (xform[ndx,"Cathedral"] == 1) { forestcover$Cathedral[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Como"] == 1) { forestcover$Como[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Cryaquepts_cmplx"] == 1) { forestcover$Cryaquepts_cmplx[forestcover$SoilType == ndx]
  if (xform[ndx,"Cryaquepts_Typic"] == 1) { forestcover$Cryaquepts_Typic[forestcover$SoilType == ndx]
  if (xform[ndx,"Cryaquolls"] == 1) { forestcover$Cryaquolls[forestcover$SoilType == ndx] <- 1 }
  if (xform[ndx,"Cryaquolls_cmplx"] == 1) { forestcover$Cryaquolls_cmplx[forestcover$SoilType == ndx]
```

```
    if (xform[ndx,"Cryaquolls_Typic"] == 1) { forestcover$Cryaquolls_Typic[forestcover$SoilType == ndx]
    if (xform[ndx,"Cryaquolls_Typic_cmplx"] == 1) { forestcover$Cryaquolls_Typic_cmplx[forestcover$Soil
    if (xform[ndx,"Cryoborolis_cmplx"] == 1) { forestcover$Cryoborolis_cmplx[forestcover$SoilType == ndx
    if (xform[ndx,"Cryorthents"] == 1) { forestcover$Cryorthents[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Cryorthents_cmplx"] == 1) { forestcover$Cryorthents_cmplx[forestcover$SoilType == ndx
    if (xform[ndx,"Cryumbrepts"] == 1) { forestcover$Cryumbrepts[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Cryumbrepts_cmplx"] == 1) { forestcover$Cryumbrepts_cmplx[forestcover$SoilType == ndx
    if (xform[ndx,"Gateview"] == 1) { forestcover$Gateview[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Gothic"] == 1) { forestcover$Gothic[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Granile"] == 1) { forestcover$Granile[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Haploborolis"] == 1) { forestcover$Haploborolis[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Legault"] == 1) { forestcover$Legault[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Legault_cmplx"] == 1) { forestcover$Legault_cmplx[forestcover$SoilType == ndx] <- 1
    if (xform[ndx,"Leighcan"] == 1) { forestcover$Leighcan[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Leighcan_cmplx"] == 1) { forestcover$Leighcan_cmplx[forestcover$SoilType == ndx] <-
    if (xform[ndx,"Leighcan_warm"] == 1) { forestcover$Leighcan_warm[forestcover$SoilType == ndx] <- 1
    if (xform[ndx,"Moran"] == 1) { forestcover$Moran[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Ratake"] == 1) { forestcover$Ratake[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Ratake_cmplx"] == 1) { forestcover$Ratake_cmplx[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Rogert"] == 1) { forestcover$Rogert[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Supervisor_Limber_cmplx"] == 1) { forestcover$Supervisor_Limber_cmplx[forestcover$So
    if (xform[ndx,"Troutville"] == 1) { forestcover$Troutville[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Unspecified"] == 1) { forestcover$Unspecified[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Vanet"] == 1) { forestcover$Vanet[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Wetmore"] == 1) { forestcover$Wetmore[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Bouldery_ext"] == 1) { forestcover$Bouldery_ext[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Rock_Land"] == 1) { forestcover$Rock_Land[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Rock_Land_cmplx"] == 1) { forestcover$Rock_Land_cmplx[forestcover$SoilType == ndx] <
    if (xform[ndx,"Rock_Outcrop"] == 1) { forestcover$Rock_Outcrop[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Rock_Outcrop_cmplx"] == 1) { forestcover$Rock_Outcrop_cmplx[forestcover$SoilType ==
    if (xform[ndx,"Rubbly"] == 1) { forestcover$Rubbly[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Stony"] == 1) { forestcover$Stony[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Stony_extreme"] == 1) { forestcover$Stony_extreme[forestcover$SoilType == ndx] <- 1
    if (xform[ndx,"Stony_very"] == 1) { forestcover$Stony_very[forestcover$SoilType == ndx] <- 1 }
    if (xform[ndx,"Till_Substratum"] == 1) { forestcover$Till_Substratum[forestcover$SoilType == ndx] <
  }
  endTime=Sys.time()
  print(paste("Forest coverage soil type expansion completed at",endTime))
  print(paste("Elapsed time=",round(endTime-startTime),"seconds."))
}
```

```
## [1] "Starting soil type expansion at 2018-04-02 21:32:22"
## [1] "Forest coverage soil type expansion completed at 2018-04-02 21:32:29"
## [1] "Elapsed time= 7 seconds."
```

```
glimpse(forestcover)
```

```
## Observations: 581,012
## Variables: 117
## $ Elev                    <int> 2596, 2590, 2804, 2785, 2595, 2579, 26...
## $ Aspect                  <int> 51, 56, 139, 155, 45, 132, 45, 49, 45,...
## $ Slope                   <int> 3, 2, 9, 18, 2, 6, 7, 4, 9, 10, 4, 11,...
## $ H2OHD                   <int> 258, 212, 268, 242, 153, 300, 270, 234...
## $ H2OVD                   <int> 0, -6, 65, 118, -1, -15, 5, 7, 56, 11,...
```

```
## $ RoadHD                <int> 510, 390, 3180, 3090, 391, 67, 633, 57...
## $ Shade9AM              <int> 221, 220, 234, 238, 220, 230, 222, 222...
## $ Shade12PM             <int> 232, 235, 238, 238, 234, 237, 225, 230...
## $ Shade3PM              <int> 148, 151, 135, 122, 150, 140, 138, 144...
## $ FirePtHD              <int> 6279, 6225, 6121, 6211, 6172, 6031, 62...
## $ RWwild                <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ NEwild                <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CMwild                <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CPwild                <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST01                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST02                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST03                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST04                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST05                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST06                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST07                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST08                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST09                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST10                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST11                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST12                  <int> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST13                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST14                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST15                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST16                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST17                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST18                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ ST19                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST20                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST21                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST22                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST23                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST24                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST25                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST26                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST27                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST28                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST29                  <int> 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,...
## $ ST30                  <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1,...
## $ ST31                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST32                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST33                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST34                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST35                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST36                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST37                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST38                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST39                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST40                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CovType               <int> 5, 5, 2, 2, 5, 2, 5, 5, 5, 5, 5, 2, 2,...
## $ STsum                 <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ Wildsum               <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ SoilType              <dbl> 29, 29, 12, 30, 29, 29, 29, 29, 29, 29...
## $ Montane_low           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
```

```
## $ Montane                 <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ Subalpine               <dbl> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ Alpine                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Dry                     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Non_Dry                 <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ Alluviam                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Glacial                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Sed_mix                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Ign_Meta                <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ Aquolis_cmplx           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Argiborolis_Pachic      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Borohemists_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bross                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bullwark                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bullwark_Cmplx          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Catamount               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Catamount_cmplx         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cathedral               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Como                    <dbl> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,...
## $ Cryaquepts_cmplx        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquepts_Typic        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls_cmplx        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls_Typic        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls_Typic_cmplx  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryoborolis_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryorthents             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryorthents_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryumbrepts             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryumbrepts_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Gateview                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Gothic                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Granile                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Haploborolis            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Legault                 <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Legault_cmplx           <dbl> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,...
## $ Leighcan                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Leighcan_cmplx          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Leighcan_warm           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Moran                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Ratake                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Ratake_cmplx            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rogert                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ Supervisor_Limber_cmplx <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Troutville              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Unspecified             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Vanet                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Wetmore                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bouldery_ext            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rock_Land               <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1,...
## $ Rock_Land_cmplx         <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rock_Outcrop            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rock_Outcrop_cmplx      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rubbly                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
```

```
## $ Stony                    <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Stony_extreme            <dbl> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,...
## $ Stony_very               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ Till_Substratum          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
```

Data Scaling

Some scaling of the data may be required but is unknown at this time, so only a note is made here to remember to update this section as needed.

Expand CovType variable into binary columns.

Create the column names.

```
system.time({
forestcover <- mutate(forestcover,Spruce_Fir=0)
forestcover <- mutate(forestcover,LodgepolePine=0)
forestcover <- mutate(forestcover,PonderosaPine=0)
forestcover <- mutate(forestcover,Cottonwood_Willow=0)
forestcover <- mutate(forestcover,Aspen=0)
forestcover <- mutate(forestcover,DouglasFir=0)
forestcover <- mutate(forestcover,Krummholz=0)
})
```

```
##    user  system elapsed
##    0.11    0.00    0.11
```

Populate the columns.

```
system.time({
forestcover$Spruce_Fir[forestcover$CovType == 1] <- 1
forestcover$LodgepolePine[forestcover$CovType == 2] <- 1
forestcover$PonderosaPine[forestcover$CovType == 3] <- 1
forestcover$Cottonwood_Willow[forestcover$CovType == 4] <- 1
forestcover$Aspen[forestcover$CovType == 5] <- 1
forestcover$DouglasFir[forestcover$CovType == 6] <- 1
forestcover$Krummholz[forestcover$CovType == 7] <- 1
glimpse(forestcover)
table(forestcover$CovType)
table(forestcover$Spruce_Fir)
table(forestcover$LodgepolePine)
table(forestcover$PonderosaPine)
table(forestcover$Cottonwood_Willow)
table(forestcover$Aspen)
table(forestcover$DouglasFir)
table(forestcover$Krummholz)
})
```

```
## Observations: 581,012
## Variables: 124
## $ Elev                     <int> 2596, 2590, 2804, 2785, 2595, 2579, 26...
## $ Aspect                   <int> 51, 56, 139, 155, 45, 132, 45, 49, 45,...
## $ Slope                    <int> 3, 2, 9, 18, 2, 6, 7, 4, 9, 10, 4, 11,...
## $ H2OHD                    <int> 258, 212, 268, 242, 153, 300, 270, 234...
```

```
## $ H2OVD                 <int> 0, -6, 65, 118, -1, -15, 5, 7, 56, 11,...
## $ RoadHD                <int> 510, 390, 3180, 3090, 391, 67, 633, 57...
## $ Shade9AM              <int> 221, 220, 234, 238, 220, 230, 222, 222...
## $ Shade12PM             <int> 232, 235, 238, 238, 234, 237, 225, 230...
## $ Shade3PM              <int> 148, 151, 135, 122, 150, 140, 138, 144...
## $ FirePtHD              <int> 6279, 6225, 6121, 6211, 6172, 6031, 62...
## $ RWwild                <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ NEwild                <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CMwild                <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CPwild                <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST01                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST02                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST03                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST04                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST05                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST06                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST07                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST08                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST09                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST10                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST11                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST12                  <int> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST13                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST14                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST15                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST16                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST17                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST18                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ ST19                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST20                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST21                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST22                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST23                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST24                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST25                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST26                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST27                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST28                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST29                  <int> 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,...
## $ ST30                  <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1,...
## $ ST31                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST32                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST33                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST34                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST35                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST36                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST37                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST38                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST39                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ ST40                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CovType               <int> 5, 5, 2, 2, 5, 2, 5, 5, 5, 5, 5, 2, 2,...
## $ STsum                 <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ Wildsum               <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ SoilType              <dbl> 29, 29, 12, 30, 29, 29, 29, 29, 29, 29...
```

```
## $ Montane_low             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Montane                 <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ Subalpine               <dbl> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ Alpine                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Dry                     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Non_Dry                 <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ Alluviam                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Glacial                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Sed_mix                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Ign_Meta                <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ Aquolis_cmplx           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Argiborolis_Pachic      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Borohemists_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bross                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bullwark                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bullwark_Cmplx          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Catamount               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Catamount_cmplx         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cathedral               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Como                    <dbl> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,...
## $ Cryaquepts_cmplx        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquepts_Typic        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls_cmplx        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls_Typic        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls_Typic_cmplx  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryoborolis_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryorthents             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryorthents_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryumbrepts             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryumbrepts_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Gateview                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Gothic                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Granile                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Haploborolis            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Legault                 <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Legault_cmplx           <dbl> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,...
## $ Leighcan                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Leighcan_cmplx          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Leighcan_warm           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Moran                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Ratake                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Ratake_cmplx            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rogert                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ Supervisor_Limber_cmplx <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Troutville              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Unspecified             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Vanet                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Wetmore                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bouldery_ext            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rock_Land               <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1,...
## $ Rock_Land_cmplx         <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rock_Outcrop            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rock_Outcrop_cmplx      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
```

```
## $ Rubbly                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Stony                   <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Stony_extreme           <dbl> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,...
## $ Stony_very              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ Till_Substratum         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Spruce_Fir              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ LodgepolePine           <dbl> 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1,...
## $ PonderosaPine           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cottonwood_Willow       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Aspen                   <dbl> 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0,...
## $ DouglasFir              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Krummholz               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
```

```
##    user  system elapsed
##    5.84    0.17    6.12
```

Save current forest cover data with ST___ columns

```
system.time({
startTime=Sys.time()
print(paste("Data save with ST__ columns started at",startTime))

write.csv(forestcover, file=out1file,row.names=FALSE)

endTime=Sys.time()
  print(paste("Data save with ST__ columns completed at",endTime))
  print(paste("Elapsed time=",round(endTime-startTime),"seconds."))
})
```

```
## [1] "Data save with ST__ columns started at 2018-04-02 21:32:35"
## [1] "Data save with ST__ columns completed at 2018-04-02 21:34:23"
## [1] "Elapsed time= 2 seconds."
```

```
##    user  system elapsed
##  104.23    1.17  107.32
```

Remove Unneeded columns

The binary soil type columns are no longer needed and are removed. Save the cleaned data before removing columns in a CSV file.

```
startTime=Sys.time()
print(paste("Forest coverage ST__ column deletion started at",startTime))
```

```
## [1] "Forest coverage ST__ column deletion started at 2018-04-02 21:34:23"
```

```
forestcover <- forestcover %>% select(-ST01,-ST02,-ST03,-ST04,-ST05,-ST06,-ST07,-ST08,-ST09,-ST10,
                                      -ST11,-ST12,-ST13,-ST14,-ST15,-ST16,-ST17,-ST18,-ST19,-ST20,
                                      -ST21,-ST22,-ST23,-ST24,-ST25,-ST26,-ST27,-ST28,-ST29,-ST30,
                                      -ST31,-ST32,-ST33,-ST34,-ST35,-ST36,-ST37,-ST38,-ST39,-ST40,
                                      -STsum, -Wildsum
                                      )
endTime=Sys.time()
print(paste("Forest coverage ST__ column deletion completed at",endTime))
```

```
## [1] "Forest coverage ST__ column deletion completed at 2018-04-02 21:34:23"
```

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

```
## [1] "Elapsed time= 0 seconds."
```

```
glimpse(forestcover)
```

```
## Observations: 581,012
## Variables: 82
## $ Elev                    <int> 2596, 2590, 2804, 2785, 2595, 2579, 26...
## $ Aspect                  <int> 51, 56, 139, 155, 45, 132, 45, 49, 45,...
## $ Slope                   <int> 3, 2, 9, 18, 2, 6, 7, 4, 9, 10, 4, 11,...
## $ H2OHD                   <int> 258, 212, 268, 242, 153, 300, 270, 234...
## $ H2OVD                   <int> 0, -6, 65, 118, -1, -15, 5, 7, 56, 11,...
## $ RoadHD                  <int> 510, 390, 3180, 3090, 391, 67, 633, 57...
## $ Shade9AM                <int> 221, 220, 234, 238, 220, 230, 222, 222...
## $ Shade12PM               <int> 232, 235, 238, 238, 234, 237, 225, 230...
## $ Shade3PM                <int> 148, 151, 135, 122, 150, 140, 138, 144...
## $ FirePtHD                <int> 6279, 6225, 6121, 6211, 6172, 6031, 62...
## $ RWwild                  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ NEwild                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CMwild                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CPwild                  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CovType                 <int> 5, 5, 2, 2, 5, 2, 5, 5, 5, 5, 5, 2, 2,...
## $ SoilType                <dbl> 29, 29, 12, 30, 29, 29, 29, 29, 29, 29...
## $ Montane_low             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Montane                 <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ Subalpine               <dbl> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ Alpine                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Dry                     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Non_Dry                 <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ Alluviam                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Glacial                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Sed_mix                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Ign_Meta                <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ Aquolis_cmplx           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Argiborolis_Pachic      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Borohemists_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bross                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bullwark                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bullwark_Cmplx          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Catamount               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Catamount_cmplx         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cathedral               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Como                    <dbl> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,...
## $ Cryaquepts_cmplx        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquepts_Typic        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls_cmplx        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls_Typic        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryaquolls_Typic_cmplx  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryoborolis_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryorthents             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryorthents_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
```

```
## $ Cryumbrepts              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cryumbrepts_cmplx        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Gateview                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Gothic                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Granile                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Haploborolis             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Legault                  <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Legault_cmplx            <dbl> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,...
## $ Leighcan                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Leighcan_cmplx           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Leighcan_warm            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Moran                    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Ratake                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Ratake_cmplx             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rogert                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ Supervisor_Limber_cmplx <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Troutville               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Unspecified              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Vanet                    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Wetmore                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Bouldery_ext             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rock_Land                <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1,...
## $ Rock_Land_cmplx          <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rock_Outcrop             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rock_Outcrop_cmplx       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Rubbly                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Stony                    <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Stony_extreme            <dbl> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,...
## $ Stony_very               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ Till_Substratum          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Spruce_Fir               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ LodgepolePine            <dbl> 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1,...
## $ PonderosaPine            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Cottonwood_Willow        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Aspen                    <dbl> 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0,...
## $ DouglasFir               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Krummholz                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
```

Save the cleaned data without ST___ columns in a CSV file.

```
startTime=Sys.time()
print(paste("Data save without ST__ columns started at",startTime))
```

```
## [1] "Data save without ST__ columns started at 2018-04-02 21:34:23"
```

```
write.csv(forestcover, file=out2file,row.names=FALSE)

endTime=Sys.time()
  print(paste("Data save without ST__ columns completed at",endTime))
```

```
## [1] "Data save without ST__ columns completed at 2018-04-02 21:36:15"
```

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

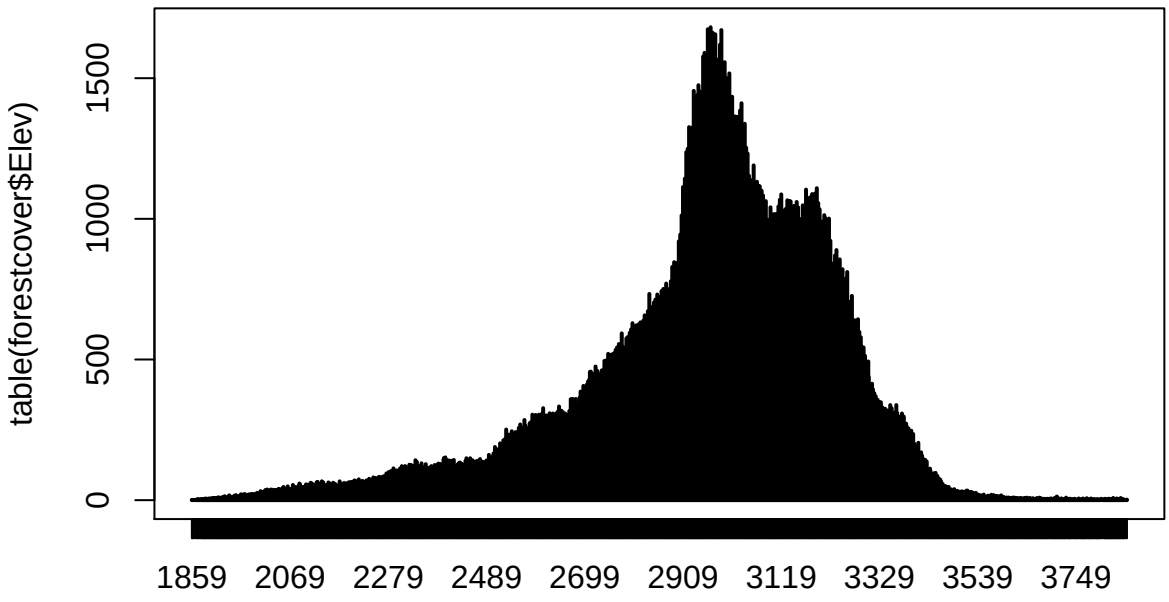
```
## [1] "Elapsed time= 2 seconds."
```

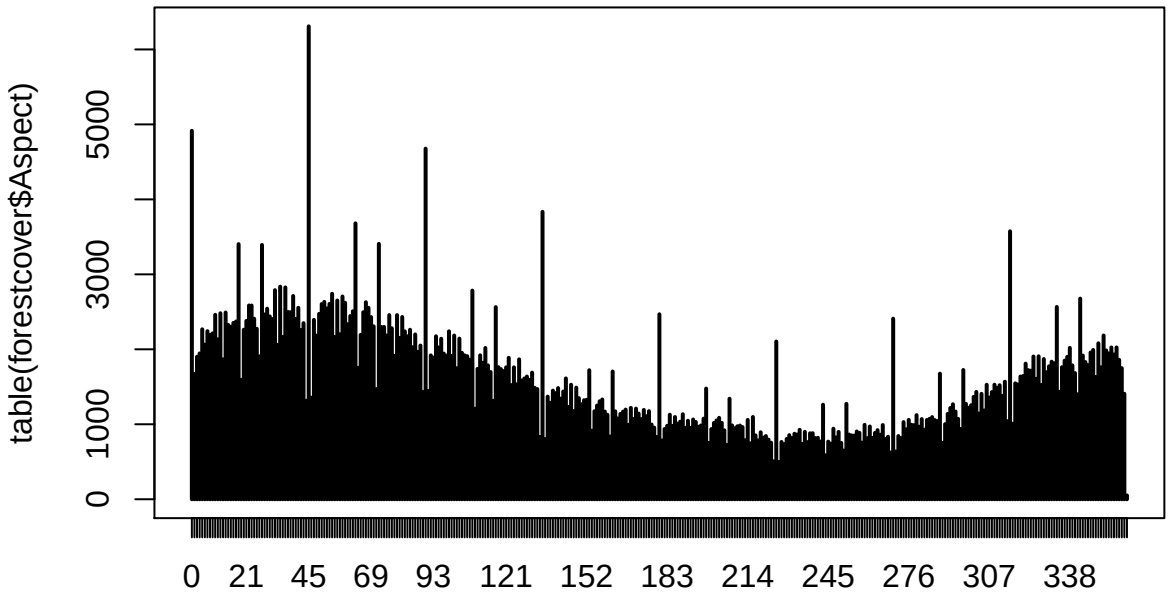
Data distributions

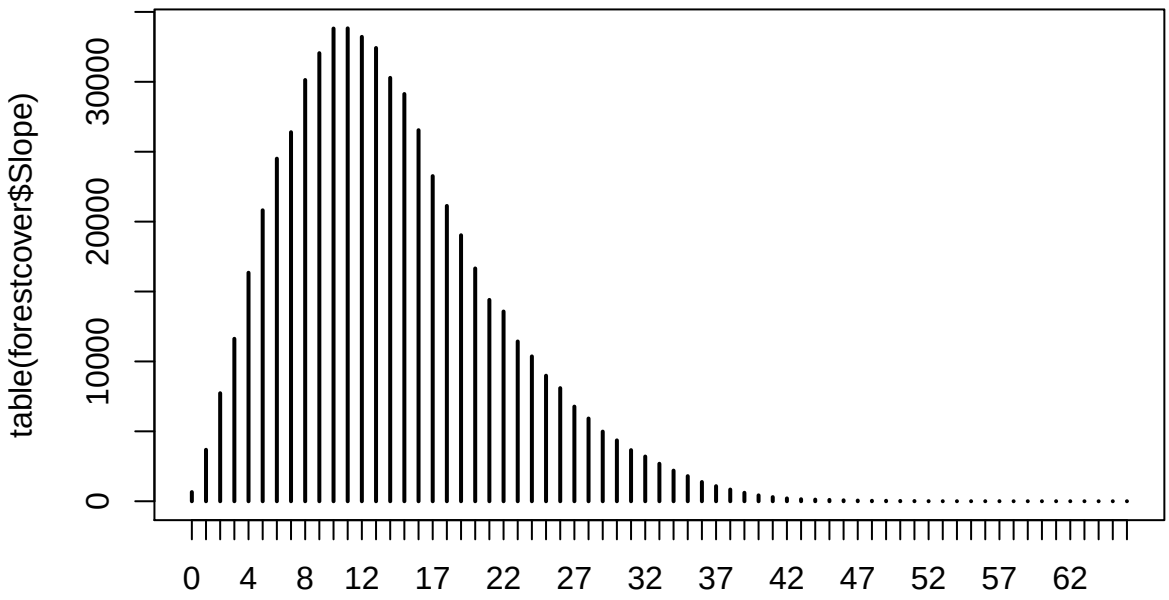
Now check some data plots for fun. No comments for them yet.

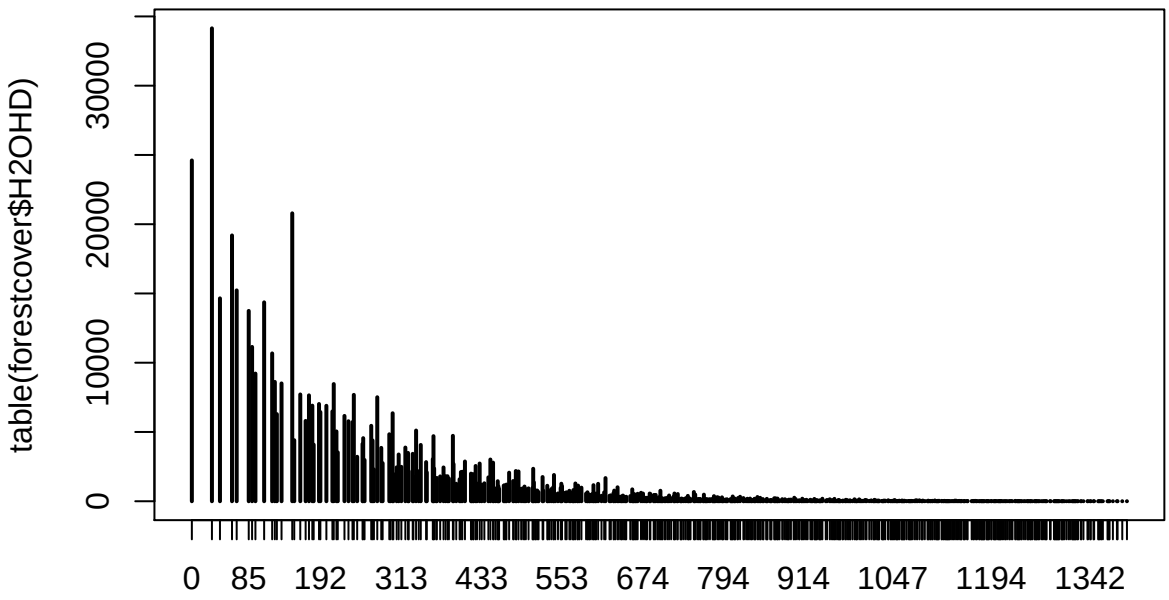
```
test=TRUE
if(test) {
  startTime=Sys.time()
  print(paste("Plot creation started at",startTime))

  plot(table(forestcover$Elev))
  plot(table(forestcover$Aspect))
  plot(table(forestcover$Slope))
  plot(table(forestcover$H2OHD))
  plot(table(forestcover$H2OVD))
  plot(table(forestcover$RoadHD))
  plot(table(forestcover$Shade9AM))
  plot(table(forestcover$Shade12PM))
  plot(table(forestcover$Shade3PM))
  plot(table(forestcover$FirePtHD))

  endTime=Sys.time()
  print(paste("Plots completed at",endTime))
  print(paste("Elapsed time=",round(endTime-startTime),"seconds."))
}
```

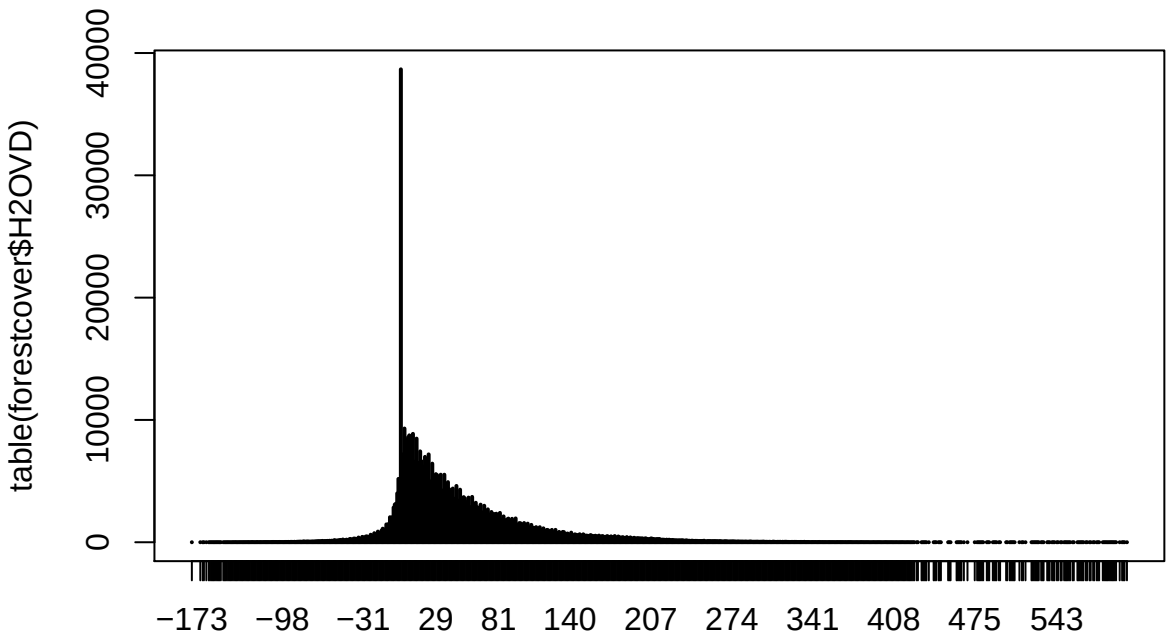
```
## [1] "Plot creation started at 2018-04-02 21:36:15"
```

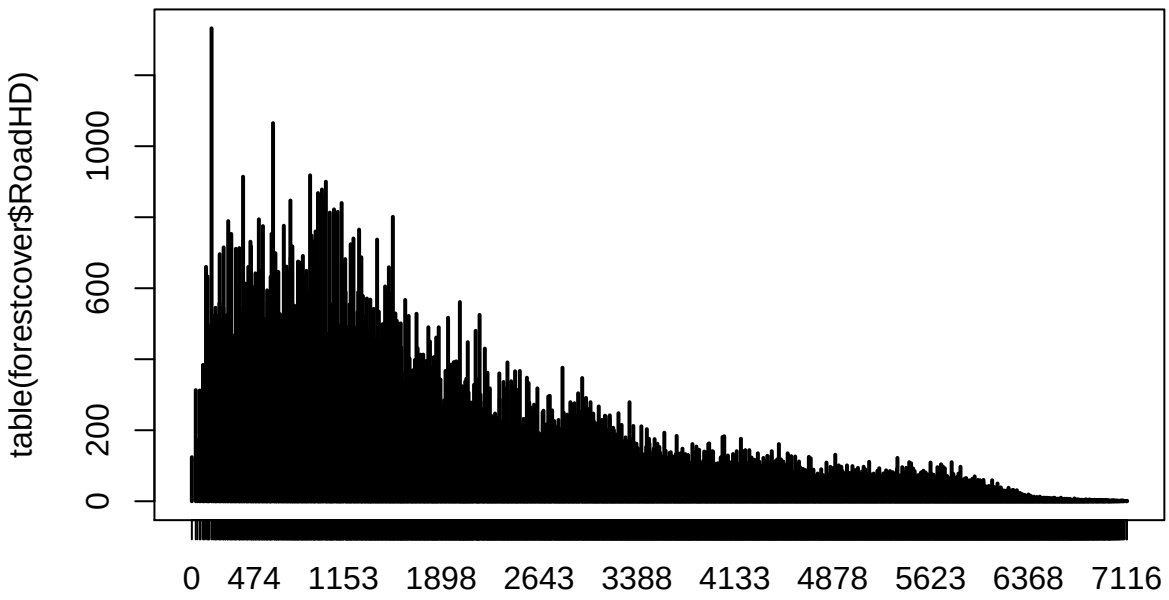


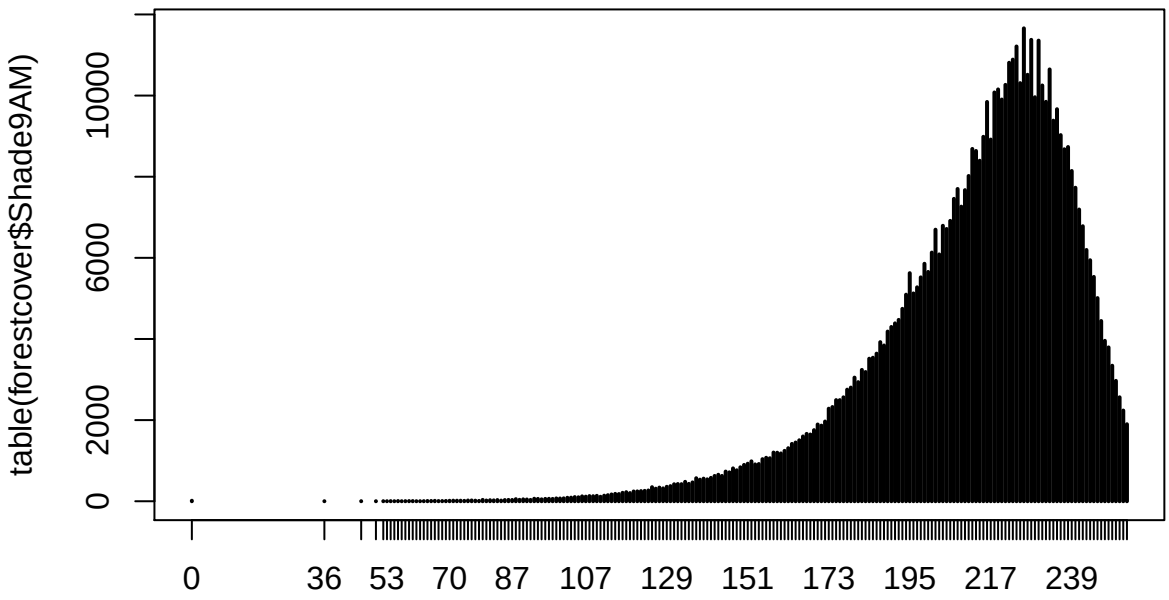

table(forestcover$Aspect)
0
1000
3000
5000
0
21
45
69
93
121
152
183
214
245
276
307
338

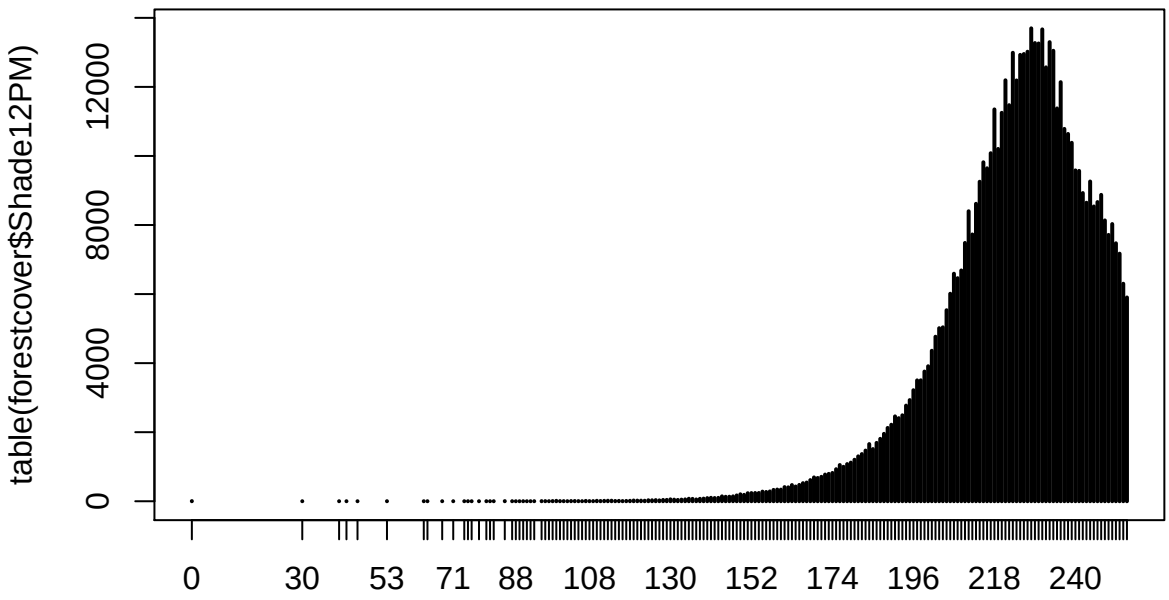

table(forestcover$Slope)
0
10000
20000
30000
0
4
8
12
17
22
27
32
37
42
47
52
57
62

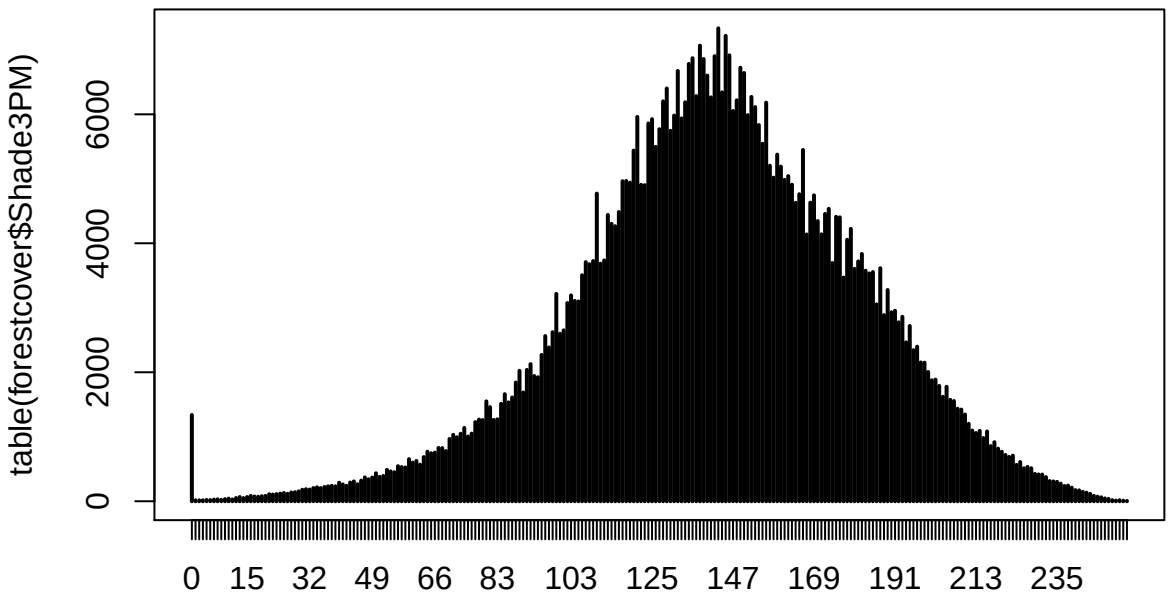

table(forestcover$H2OHD)
0
10000
20000
30000
0
85
192
313
433
553
674
794
914
1047
1194
1342

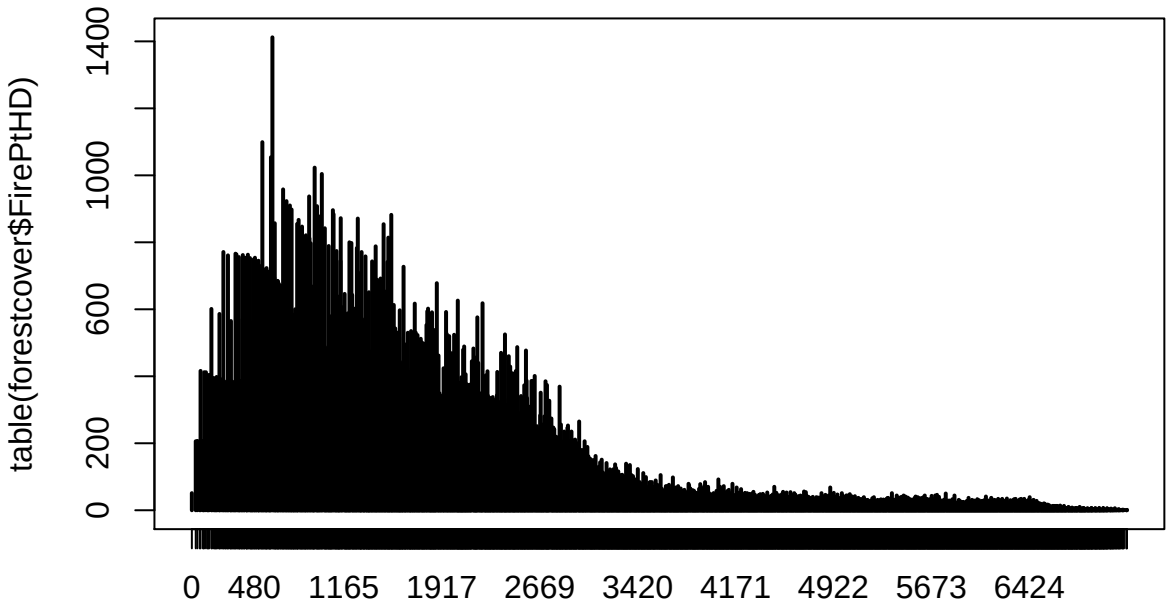

table(forestcover$H2OVD)
0
10000
20000
30000
40000
−173
−98
−31
29
81
140
207
274
341
408
475
543


table(forestcover$RoadHD)
0
200
600
1000
0
474
1153
1898
2643
3388
4133
4878
5623
6368
7116


table(forestcover$Shade9AM)
0
2000
6000
10000
0
36
53
70
87
107
129
151
173
195
217
239


table(forestcover$Shade12PM)
0
4000
8000
12000
0
30
53
71
88
108
130
152
174
196
218
240


table(forestcover$Shade3PM)
0
2000
4000
6000
0 15 32 49 66 83 103 125 147 169 191 213 235




```
## [1] "Plots completed at 2018-04-02 21:36:21"
## [1] "Elapsed time= 5 seconds."
```

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

```
## [1] "R script  started  at 2018-04-02 21:31:57"
```

```
print(paste("R script completed at",progEnd))
```

```
## [1] "R script completed at 2018-04-02 21:36:21"
```

```
#print(paste("Elapsed time=",progEnd-progStart,"seconds."))
```

That concludes the current data wrangling exercise on my capstone data.