Rough Draft Report V1.0

Ryan Erickson

2023-06-02

This is a rough draft of the code and their results.

Data Creation

Note: Monitor data was processed and exported from another document.

This first section is dedicated to the loading, processing, and creation of all necessary variables: - Variables Added: Monitor distance to nearest road, total distance of roads within a 500m radius of monitor, average NDVI within a 500m radius of monitor, elevation at the monitor location, average monthly precipitation, average monthly maximum relative humidity, and average monthly temperature.

Libraries

```
# This is for downloading and processing the .nc files from GridMET: https://www.climatologylab.org/gri
# I used the second option '2. Create "wget script" -> might take this out
library(raster)
```

```
## Loading required package: sp
```

```
library(rasterVis)
```

```
## Loading required package: lattice
```

```
library(ncdf4)
library(lattice)
library(stringr)
library(raster)
library(sf)
```

```
## Linking to GEOS 3.9.3, GDAL 3.5.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
library(dplyr)
```

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

```
## The following objects are masked from 'package:raster':
##
##     intersect, select, union

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

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

Special R Document

Here I'm using 'rsource()' to load the R script where I created/cleaned the gauge locations. The results of it are the gauge locations and their resulting mda8 average.

```
source("ozone_krige.R") #need change for mac when applicable
```

```
# Necessary Folders
path_to_cropped_data = "../final_data/"  #need change for mac when applicable
co_data_list = list.files(path_to_cropped_data)
# Random Forest Variables
# Spatial Variables:
```

Some pathing jargon:

Dist to nearest Road

```
getting_folders = grep("co_roads_2019", co_data_list, value = T)
road_shp_file = "co_roads_2019.shp"
path_to_roads = paste0(path_to_cropped_data,getting_folders,"/")
# C = County
# I = Interstate
# M = Common Name
# O = Other
# S = State recognized
# U = U.S.
roads_shp = st_read(paste0(path_to_roads,road_shp_file))
```

```
## Reading layer 'co_roads_2019' from data source
##   'C:\Users\RErickson\Documents\GitHub\ozone_data\final_data\co_roads_2019\co_roads_2019.shp'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 2944 features and 4 fields
## Geometry type: LINESTRING
## Dimension:     XY
## Bounding box:  xmin: -109.0602 ymin: 36.99251 xmax: -102.0417 ymax: 41.00307
## Geodetic CRS:  NAD83
```

```
roads_transformed = st_transform(roads_shp,crs=CRS(prg))
roads_projected = as(roads_transformed, "Spatial")
dist2road = o3_projected$dist2road[1]=round(rgeos::gDistance(roads_projected,o3_projected[1,]),2)
for(i in 2:nrow(o3_projected)){
  dist2road=c(dist2road,round(rgeos::gDistance(roads_projected,o3_projected[i,]),2))
}
o3_projected$dist2road=dist2road
```

Sum of Roads in 500m Buffer

```
add_road_buffer = read_csv("../final_data/roads_in_500m_buffer.csv")[,c(2,4)] %>%
  filter(site_name != "Aspen Park" & site_name != "Evergreen" & site_name !="Welch")
```

```
## Rows: 11 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): site_name
## dbl (3): OID_, FREQUENCY, SUM_SHAPE_Length_1
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
names(add_road_buffer) = c("site_name","road_length")
add_road_buffer$road_length=ifelse(is.na(add_road_buffer$road_length),0,add_road_buffer$road_length)
o3_projected = merge(x=o3_projected,y=add_road_buffer,by="site_name")
```

Elevation

```
elevation_to_add = raster("../final_data/elevation.tiff")
elevation_projected = raster::projectRaster(elevation_to_add, crs=prg)
o3_projected$elev=round(raster::extract(elevation_projected,o3_projected),2)
```

Temporary Data Frame Creation:

```
year_o3 = as.data.frame(o3_projected) %>%
  dplyr::select(c("site_name","elev","dist2road","road_length","lat","long"),everything())

# use to create dataframe of specific months, ex below is summer
summer_o3 = year_o3 %>%
  dplyr::select(contains(c("site_name","lat","long","elev","dist2road","road_length","Apr","May","Jun",
  pivot_longer(cols = contains(c("Apr","May","Jun","Jul","Aug","Sep","Oct")), names_to = "date", values
# preview data
#summer_o3
```

Spatio-Temporal Variables:

```
new_path = "../final_data/Monthly_Averages/"
monthly_path = list.files(paste0(new_path))
max_rh_files=grep("rmax_",monthly_path, value = T)
max_temp_files=grep("tmmx_",monthly_path, value = T)
max_precip_files=grep("pr_",monthly_path, value = T)
renameing_convention = c(paste0(month.abb,".",2017),
                         paste0(month.abb,".",2018),
                         paste0(month.abb,".",2019),
                         paste0(month.abb,".",2020),
                         paste0(month.abb,".",2021),
                         paste0(month.abb,".",2022))
coordinates(summer_o3) = c('long', 'lat')
proj4string(summer_o3) = CRS(SRS_string = prg)
AE=summer_o3[which(summer_o3$site_name=="Aurora East"),]
BR=summer_o3[which(summer_o3$site_name=="Boulder Reservoir"),]
DC=summer_o3[which(summer_o3$site_name=="Denver - Camp"),]
HR=summer_o3[which(summer_o3$site_name=="Highland Reservoir"),]
LA=summer_o3[which(summer_o3$site_name=="La Casa"),]
NREL=summer_o3[which(summer_o3$site_name=="National Renewable Energy Labs - Nrel"),]
RF=summer_o3[which(summer_o3$site_name=="Rocky Flats-N"),]
WY=summer_o3[which(summer_o3$site_name=="Welby"),]

ndvi_sort_2018=summer_o3[grep(".2018",summer_o3$date),c("site_name","date")]
ndvi_sort_2019=summer_o3[grep(".2019",summer_o3$date),c("site_name","date")]
ndvi_sort_2020=summer_o3[grep(".2020",summer_o3$date),c("site_name","date")]
ndvi_sort_2021=summer_o3[grep(".2021",summer_o3$date),c("site_name","date")]
ndvi_sort_2022=summer_o3[grep(".2022",summer_o3$date),c("site_name","date")]
```

Average Monthly Precipitation

```
max_precip_to_add = stack(paste0(new_path,max_precip_files)) # create stack of raster bricks (each "sta
#plots
# plot(max_precip_to_add$tmmx_2017_monthly_avg_1.1, main="Precipitation for Jan 2017")
max_precip_projected = raster::projectRaster(max_precip_to_add, crs=prg)
names(max_precip_projected) = renameing_convention
#plots: after running names() (above code)
# plot(max_precip_projected$Jan.2017, main="Precipitation for Jan 2017")
summer_max_precip =max_precip_projected[[c(grep("2018", names(max_precip_projected)),
                                           grep("2019", names(max_precip_projected)),
                                           grep("2020", names(max_precip_projected)),
                                           grep("2021", names(max_precip_projected)),
                                           grep("2022", names(max_precip_projected)))]]
summer_max_precip =summer_max_precip[[c(grep("Apr", names(summer_max_precip)),
                                        grep("May", names(summer_max_precip)),
                                        grep("Jun", names(summer_max_precip)),
                                        grep("Jul", names(summer_max_precip)),
                                        grep("Aug", names(summer_max_precip)),
                                        grep("Sep", names(summer_max_precip)),
                                        grep("Oct", names(summer_max_precip)))]]
```

Average Monthly Temperatures

- rounded K to F formula just in case: 1.8*(K-273) + 32

```
max_temperature_to_add = stack(paste0(new_path,max_temp_files)) # create stack of raster bricks (each "
#plots
# plot(max_temperature_to_add$tmmx_2017_monthly_avg_1.1, main="Temperature for Jan 2017")

max_temperature_projected = raster::projectRaster(max_temperature_to_add, crs=prg)
names(max_temperature_projected) = renameing_convention
#plots: after running names() (above code)
# plot(max_temperature_projected$Jan.2017, main="Temperature for Jan 2017")
summer_max_temps =max_temperature_projected[[c(grep("2018", names(max_temperature_projected)),
                                grep("2019", names(max_temperature_projected)),
                                grep("2020", names(max_temperature_projected)),
                                grep("2021", names(max_temperature_projected)),
                                grep("2022", names(max_temperature_projected)))]]
summer_max_temps =summer_max_temps[[c(grep("Apr", names(summer_max_temps)),
            grep("May", names(summer_max_temps)),
            grep("Jun", names(summer_max_temps)),
            grep("Jul", names(summer_max_temps)),
            grep("Aug", names(summer_max_temps)),
            grep("Sep", names(summer_max_temps)),
            grep("Oct", names(summer_max_temps)))]]
```

Average Monthly Relative Humidity

```
max_rh_to_add = stack(paste0(new_path,max_rh_files)) # create stack of raster bricks (each "stack" is a
#plots
# plot(max_rh_to_add$tmmx_2017_monthly_avg_1.1, main="Relative Humidity for Jan 2017")
max_rh_projected = raster::projectRaster(max_rh_to_add, crs=prg)
names(max_rh_projected) = renameing_convention
#plots: after running names() (above code)
# plot(max_rh_projected$Jan.2017, main="Relative Humidity for Jan 2017")
summer_max_rh =max_rh_projected[[c(grep("2018", names(max_rh_projected)),
                                        grep("2019", names(max_rh_projected)),
                                        grep("2020", names(max_rh_projected)),
                                        grep("2021", names(max_rh_projected)),
                                        grep("2022", names(max_rh_projected)))]]
summer_max_rh =summer_max_rh[[c(grep("Apr", names(summer_max_rh)),
                                    grep("May", names(summer_max_rh)),
                                    grep("Jun", names(summer_max_rh)),
                                    grep("Jul", names(summer_max_rh)),
                                    grep("Aug", names(summer_max_rh)),
                                    grep("Sep", names(summer_max_rh)),
                                    grep("Oct", names(summer_max_rh)))]]
```

Exposure Assignment

```
for(i in 1:nrow(AE)) {
  AE$tmax[i] = raster::extract(summer_max_temps[[i]],AE[i,])
  AE$rhmax[i] = raster::extract(summer_max_rh[[i]],AE[i,])
  AE$pmax[i] = raster::extract(summer_max_precip[[i]],AE[i,])
}
for(i in 1:nrow(BR)) {
  BR$tmax[i] = raster::extract(summer_max_temps[[i]],BR[i,])
  BR$rhmax[i] = raster::extract(summer_max_rh[[i]],BR[i,])
  BR$pmax[i] = raster::extract(summer_max_precip[[i]],BR[i,])
}
for(i in 1:nrow(DC)) {
  DC$tmax[i] = raster::extract(summer_max_temps[[i]],DC[i,])
  DC$rhmax[i] = raster::extract(summer_max_rh[[i]],DC[i,])
  DC$pmax[i] = raster::extract(summer_max_precip[[i]],DC[i,])
}
for(i in 1:nrow(HR)) {
  HR$tmax[i] = raster::extract(summer_max_temps[[i]],HR[i,])
  HR$rhmax[i] = raster::extract(summer_max_rh[[i]],HR[i,])
  HR$pmax[i] = raster::extract(summer_max_precip[[i]],HR[i,])
}
for(i in 1:nrow(LA)) {
  LA$tmax[i] = raster::extract(summer_max_temps[[i]],LA[i,])
  LA$rhmax[i] = raster::extract(summer_max_rh[[i]],LA[i,])
  LA$pmax[i] = raster::extract(summer_max_precip[[i]],LA[i,])
}
for(i in 1:nrow(NREL)) {
  NREL$tmax[i] = raster::extract(summer_max_temps[[i]],NREL[i,])
  NREL$rhmax[i] = raster::extract(summer_max_rh[[i]],NREL[i,])
  NREL$pmax[i] = raster::extract(summer_max_precip[[i]],NREL[i,])
}
for(i in 1:nrow(RF)) {
  RF$tmax[i] = raster::extract(summer_max_temps[[i]],RF[i,])
  RF$rhmax[i] = raster::extract(summer_max_rh[[i]],RF[i,])
  RF$pmax[i] = raster::extract(summer_max_precip[[i]],RF[i,])
}
for(i in 1:nrow(WY)) {
  WY$tmax[i] = raster::extract(summer_max_temps[[i]],WY[i,])
  WY$rhmax[i] = raster::extract(summer_max_rh[[i]],WY[i,])
  WY$pmax[i] = raster::extract(summer_max_precip[[i]],WY[i,])
}

rbind(AE,BR,DC,HR,LA,NREL,RF,WY)
```

```
## class       : SpatialPointsDataFrame
## features    : 272
## extent      : 481219.8, 536954.6, 4379800, 4435552  (xmin, xmax, ymin, ymax)
## crs         : +proj=utm +zone=13 +datum=WGS84 +units=m +no_defs
## variables   : 9
## names       :   site_name,    elev, dist2road,      road_length,     date,             mda8,
## min values  : Aurora East, 1592.47,     68.83,                0, Apr.2018, 18.4914838709677, 286.2353
## max values  :       Welby, 1793.14,  11202.39, 3251.26803095015, Sep.2022, 62.7419677419355, 307.0416
```

```
ggs = as.data.frame(rbind(AE,BR,DC,HR,LA,NREL,RF,WY))
ggs %>%
  group_by(site_name) %>%
  slice_head(n=1)
```

```
## # A tibble: 8 x 11
## # Groups:   site_name [8]
##   site_name       lat   long  elev dist2road road_length date   mda8  tmax rhmax
##   <chr>         <dbl>  <dbl> <dbl>     <dbl>       <dbl> <chr> <dbl> <dbl> <dbl>
## 1 Aurora East  4.39e6 5.37e5 1793.   11202.           0  Apr.~  45.4  290.  64.1
## 2 Boulder Res~ 4.44e6 4.81e5 1594.     545.           0  Apr.~  43.2  290.  66.0
## 3 Denver - Ca~ 4.40e6 5.01e5 1609.    1175.           0  Apr.~  37.6  291.  62.7
## 4 Highland Re~ 4.38e6 5.04e5 1746.     281.        3251. Apr.~  45.9  290.  62.5
## 5 La Casa      4.40e6 5.00e5 1609.     413         1232. Apr.~  37.2  291.  62.7
## 6 National Re~ 4.40e6 4.85e5 1767.    1431.           0  Apr.~  43.7  289.  63.7
## 7 Rocky Flats~ 4.42e6 4.84e5 1728.      68.8       2405. Apr.~  47.4  290.  61.6
## 8 Welby        4.41e6 5.04e5 1592.     818.           0  Apr.~  34.4  291.  64.1
## # i 1 more variable: pmax <dbl>
```

Average NDVI Within 500m Buffer

```
ndvi_folder = "NDVIs"
ndvi_path = list.dirs(paste0("../final_data/",ndvi_folder))[-1]
ndvi_files = paste0(ndvi_path,"/den_CO_NDVI_",2018:2022,".tif")
ndvi_2018_projected = raster(paste0(ndvi_path[1],"/co_ndvi_",2018,"projected.tif"))
ndvi_2019_projected = raster(paste0(ndvi_path[2],"/co_ndvi_",2019,"projected.tif"))
ndvi_2020_projected = raster(paste0(ndvi_path[3],"/co_ndvi_",2020,"projected.tif"))
ndvi_2021_projected = raster(paste0(ndvi_path[4],"/co_ndvi_",2021,"projected.tif"))
ndvi_2022_projected = raster(paste0(ndvi_path[5],"/co_ndvi_",2022,"projected.tif"))

# Replacing values less than 0 - need new value
values(ndvi_2018_projected)=ifelse(values(ndvi_2018_projected)<0,0,values(ndvi_2018_projected))
values(ndvi_2019_projected)=ifelse(values(ndvi_2019_projected)<0,0,values(ndvi_2019_projected))
values(ndvi_2020_projected)=ifelse(values(ndvi_2020_projected)<0,0,values(ndvi_2020_projected))
values(ndvi_2021_projected)=ifelse(values(ndvi_2021_projected)<0,0,values(ndvi_2021_projected))
values(ndvi_2022_projected)=ifelse(values(ndvi_2022_projected)<0,0,values(ndvi_2022_projected))

ndvi_sort_2018$ndvi = raster::extract(ndvi_2018_projected,ndvi_sort_2018)
ndvi_sort_2019$ndvi = raster::extract(ndvi_2019_projected,ndvi_sort_2019)
ndvi_sort_2020$ndvi = raster::extract(ndvi_2020_projected,ndvi_sort_2020)
ndvi_sort_2021$ndvi = raster::extract(ndvi_2021_projected,ndvi_sort_2021)
ndvi_sort_2022$ndvi = raster::extract(ndvi_2022_projected,ndvi_sort_2022)
ndvi_to_final_dataframe = as.data.frame(rbind(ndvi_sort_2018,ndvi_sort_2019,ndvi_sort_2020,ndvi_sort_202
  select(-lat,-long)
#plot
# plot(ndvi_2018_projected)
```

Final Dataframe before Dummy Variable Add-On

```
merge(ndvi_to_final_dataframe, ggs, by=c("site_name", "date")) %>%
  group_by(site_name) %>%
  slice_head(n=1)
```

```
## # A tibble: 8 x 12
## # Groups:   site_name [8]
##   site_name   date    ndvi    lat   long  elev dist2road road_length  mda8  tmax
##   <chr>       <chr>  <dbl>  <dbl>  <dbl> <dbl>     <dbl>       <dbl> <dbl> <dbl>
## 1 Aurora East Apr.~ 0.115  4.39e6 5.37e5 1793.   11202.           0   45.4  290.
## 2 Boulder Re~ Apr.~ 0.134  4.44e6 4.81e5 1594.     545.           0   43.2  290.
## 3 Denver - C~ Apr.~ 0.0736 4.40e6 5.01e5 1609.    1175.           0   37.6  291.
## 4 Highland R~ Apr.~ 0.0837 4.38e6 5.04e5 1746.     281.        3251.  45.9  290.
## 5 La Casa     Apr.~ 0.180  4.40e6 5.00e5 1609.     413         1232.  37.2  291.
## 6 National R~ Apr.~ 0.130  4.40e6 4.85e5 1767.    1431.           0   43.7  289.
## 7 Rocky Flat~ Apr.~ 0.153  4.42e6 4.84e5 1728.      68.8       2405.  47.4  290.
## 8 Welby       Apr.~ 0.134  4.41e6 5.04e5 1592.     818.           0   34.4  291.
## # i 2 more variables: rhmax <dbl>, pmax <dbl>
```

Dummy Variables:

```
# monthly dummy variable -
# make a 1 for each month of interest and a 0 for other months
rough_variables = merge(ndvi_to_final_dataframe, ggs, by=c("site_name", "date"))
rough_variables$apr_dummy = ifelse(str_detect(rough_variables$date, "Apr."),1,0)
rough_variables$may_dummy = ifelse(str_detect(rough_variables$date, "May."),1,0)
rough_variables$jun_dummy = ifelse(str_detect(rough_variables$date, "Jun."),1,0)
rough_variables$jul_dummy = ifelse(str_detect(rough_variables$date, "Jul."),1,0)
rough_variables$aug_dummy = ifelse(str_detect(rough_variables$date, "Aug."),1,0)
rough_variables$sep_dummy = ifelse(str_detect(rough_variables$date, "Sep."),1,0)
rough_variables$oct_dummy = ifelse(str_detect(rough_variables$date, "Oct."),1,0)
# yearly dummy variable -
# make a 1 for each year of interest and a 0 for other years
rough_variables$yr_2018_dummy = ifelse(str_detect(rough_variables$date, ".2018"),1,0)
rough_variables$yr_2019_dummy = ifelse(str_detect(rough_variables$date, ".2019"),1,0)
rough_variables$yr_2020_dummy = ifelse(str_detect(rough_variables$date, ".2020"),1,0)
rough_variables$yr_2021_dummy = ifelse(str_detect(rough_variables$date, ".2021"),1,0)
rough_variables$yr_2022_dummy = ifelse(str_detect(rough_variables$date, ".2022"),1,0)
```

Data Inspection

Various plots and statistic metrics will go here. For now here's a glimpse of the values generated.

Date, NDVI, Lat, Long, Elevation, Distance to nearest road:

```
rough_variables[,1:8] %>%
  group_by(site_name) %>%
  slice_head(n=1)
```

```
## # A tibble: 8 x 8
## # Groups:   site_name [8]
##   site_name               date    ndvi    lat   long  elev dist2road road_length
##   <chr>                   <chr>  <dbl>  <dbl>  <dbl> <dbl>     <dbl>       <dbl>
## 1 Aurora East             Apr.~ 0.115  4.39e6 5.37e5 1793.   11202.           0
## 2 Boulder Reservoir       Apr.~ 0.134  4.44e6 4.81e5 1594.     545.           0
## 3 Denver - Camp           Apr.~ 0.0736 4.40e6 5.01e5 1609.    1175.           0
## 4 Highland Reservoir      Apr.~ 0.0837 4.38e6 5.04e5 1746.     281.        3251.
## 5 La Casa                 Apr.~ 0.180  4.40e6 5.00e5 1609.     413         1232.
## 6 National Renewable Ene~ Apr.~ 0.130  4.40e6 4.85e5 1767.    1431.           0
## 7 Rocky Flats-N           Apr.~ 0.153  4.42e6 4.84e5 1728.      68.8       2405.
## 8 Welby                   Apr.~ 0.134  4.41e6 5.04e5 1592.     818.           0
```

Monthly Average Ozone Value at Monitor Location , Monthly Average Maximum Temperature, Relative Humidity,and Precipitation

```
rough_variables[,c(1,9:12)] %>%
  group_by(site_name) %>%
  slice_head(n=1)
```

```
## # A tibble: 8 x 5
## # Groups:   site_name [8]
##   site_name                              mda8  tmax rhmax  pmax
##   <chr>                                 <dbl> <dbl> <dbl> <dbl>
## 1 Aurora East                            45.4  290.  64.1 0.703
## 2 Boulder Reservoir                      43.2  290.  66.0 0.900
## 3 Denver - Camp                          37.6  291.  62.7 0.904
## 4 Highland Reservoir                     45.9  290.  62.5 1.02
## 5 La Casa                                37.2  291.  62.7 0.904
## 6 National Renewable Energy Labs - Nrel  43.7  289.  63.7 0.915
## 7 Rocky Flats-N                          47.4  290.  61.6 1.05
## 8 Welby                                  34.4  291.  64.1 0.803
```

Dummy Variables - Repeated for each monitor

```
rough_variables[,c(2,17:22)] %>%
  group_by(date) %>%
  slice_head(n=1)
```

```
## # A tibble: 34 x 7
## # Groups:   date [34]
##    date  aug_dummy sep_dummy oct_dummy yr_2018_dummy yr_2019_dummy yr_2020_dummy
##    <chr>     <dbl>     <dbl>     <dbl>         <dbl>         <dbl>         <dbl>
##  1 Apr.~         0         0         0             1             0             0
```

```
##  2 Apr.~         0         0         0         0         1         0
##  3 Apr.~         0         0         0         0         0         1
##  4 Apr.~         0         0         0         0         0         0
##  5 Apr.~         0         0         0         0         0         0
##  6 Aug.~         1         0         0         1         0         0
##  7 Aug.~         1         0         0         0         1         0
##  8 Aug.~         1         0         0         0         0         1
##  9 Aug.~         1         0         0         0         0         0
## 10 Aug.~         1         0         0         0         0         0
## # i 24 more rows
```

Graphs and Statistics

```
rough_variables %>%
  group_by(site_name) %>%
  slice_head(n=1) %>%
  select(site_name) %>%
  kableExtra::kable()
```

Monitor Locations

site_name
Aurora East
Boulder Reservoir
Denver - Camp
Highland Reservoir
La Casa
National Renewable Energy Labs - Nrel
Rocky Flats-N
Welby

Dates

```
rough_variables %>%
  separate(date, c("Months","Years")) %>%
  dplyr::select(Months) %>%
  unique() %>%
  kableExtra::kable()
```

Months

	Months
1	Apr
6	Aug
11	Jul
16	Jun
21	May
26	Oct
30	Sep

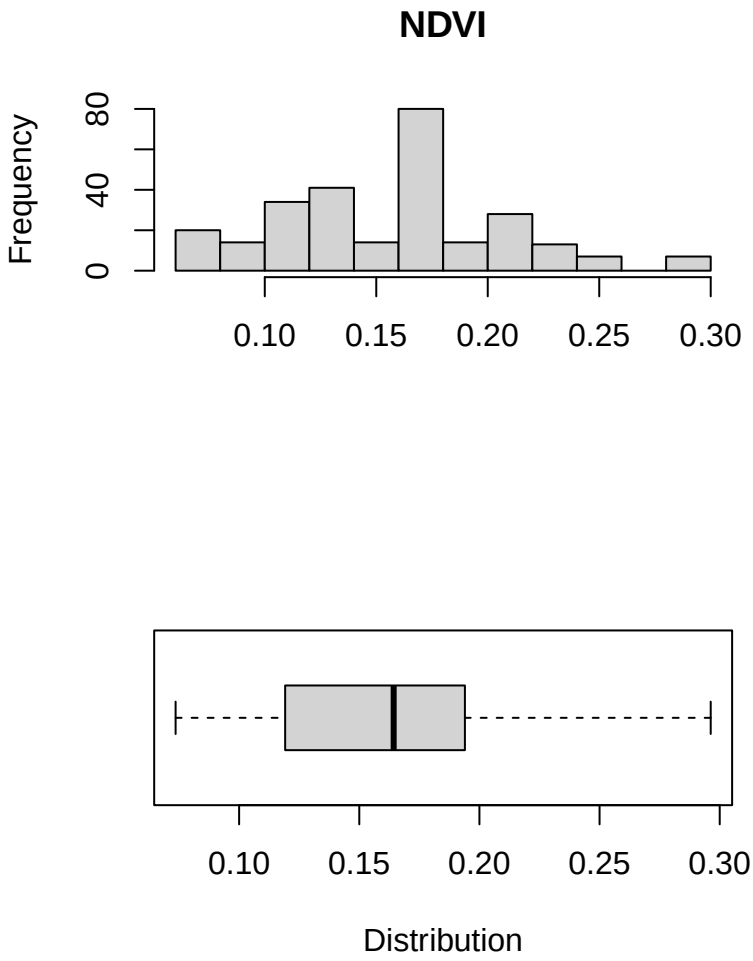
```
rough_variables %>%
  separate(date, c("Months","Years")) %>%
  dplyr::select(Years) %>%
  unique() %>%
  kableExtra::kable()
```

Years

Years
2018
2019
2020
2021
2022

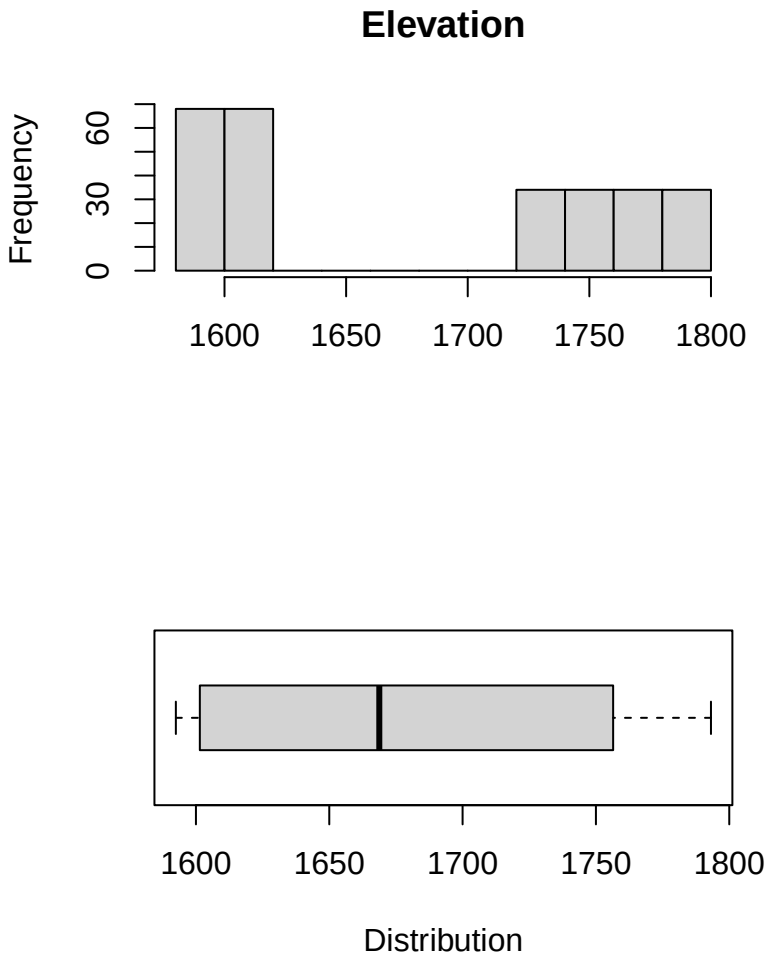
Variable Distribution

```
par(mfrow=c(2,1))
hist(rough_variables$ndvi,xlab=NULL, main="NDVI")
boxplot(rough_variables$ndvi,xlab="Distribution", main=NULL, horizontal =T)
```



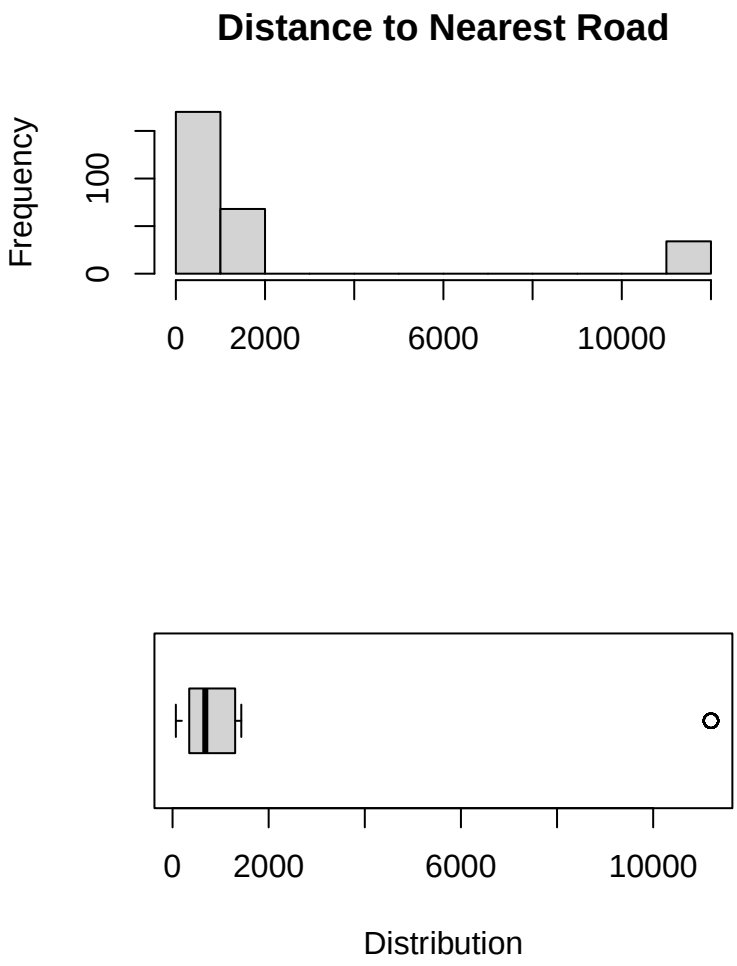
NDVI ##### Elevation

```
par(mfrow=c(2,1))
hist(rough_variables$elev,xlab=NULL, main="Elevation")
boxplot(rough_variables$elev,xlab="Distribution", main=NULL, horizontal =T)
```

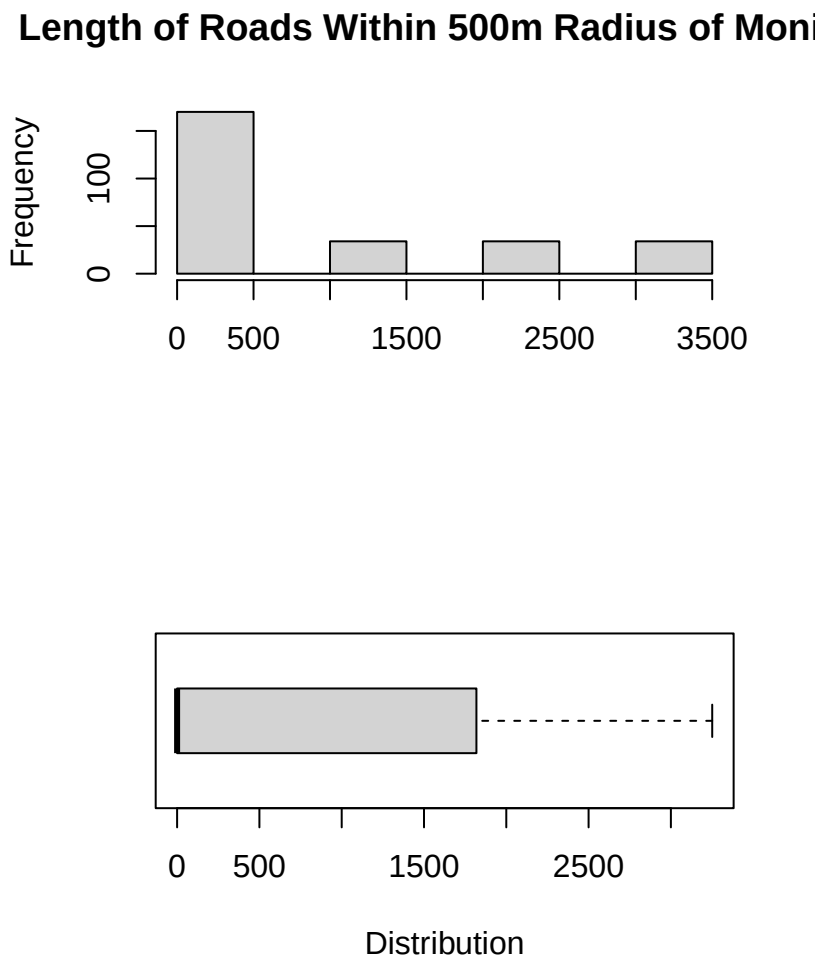


Distance to Roads

```
par(mfrow=c(2,1))
hist(rough_variables$dist2road,xlab=NULL, main="Distance to Nearest Road")
boxplot(rough_variables$dist2road,xlab="Distribution", main=NULL, horizontal =T)
```

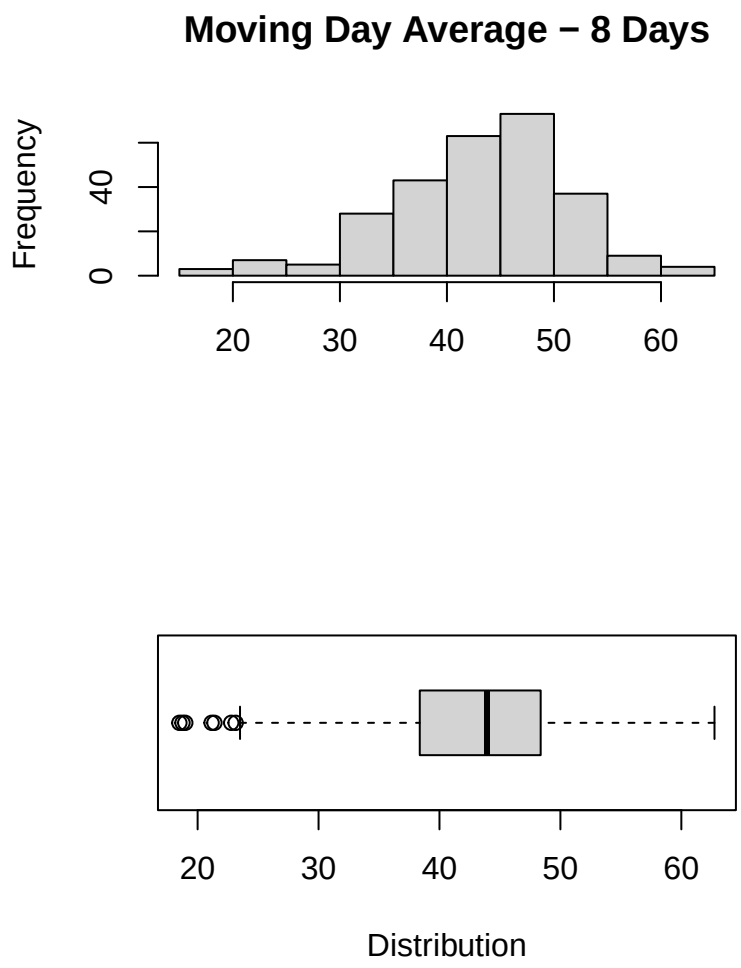



```
par(mfrow=c(2,1))
hist(rough_variables$road_length,xlab=NULL, main="Length of Roads Within 500m Radius of Monitor")
boxplot(rough_variables$road_length,xlab="Distribution", main=NULL, horizontal =T)
```



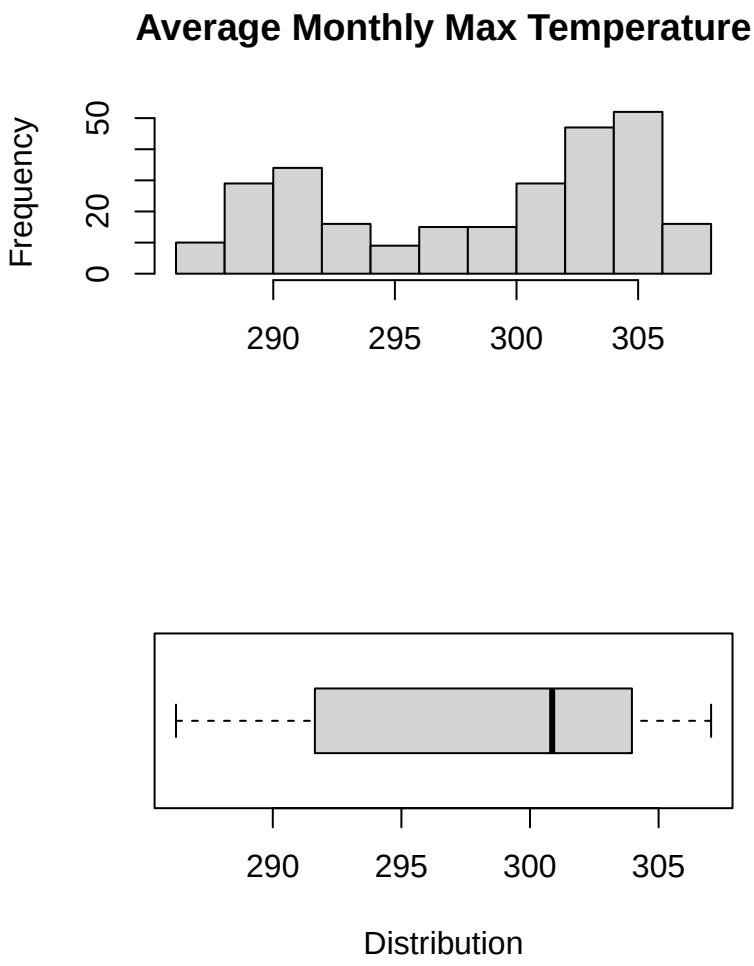
Road Length

```
par(mfrow=c(2,1))
hist(rough_variables$mda8,xlab=NULL, main="Moving Day Average - 8 Days")
boxplot(rough_variables$mda8,xlab="Distribution", main=NULL, horizontal =T)
```



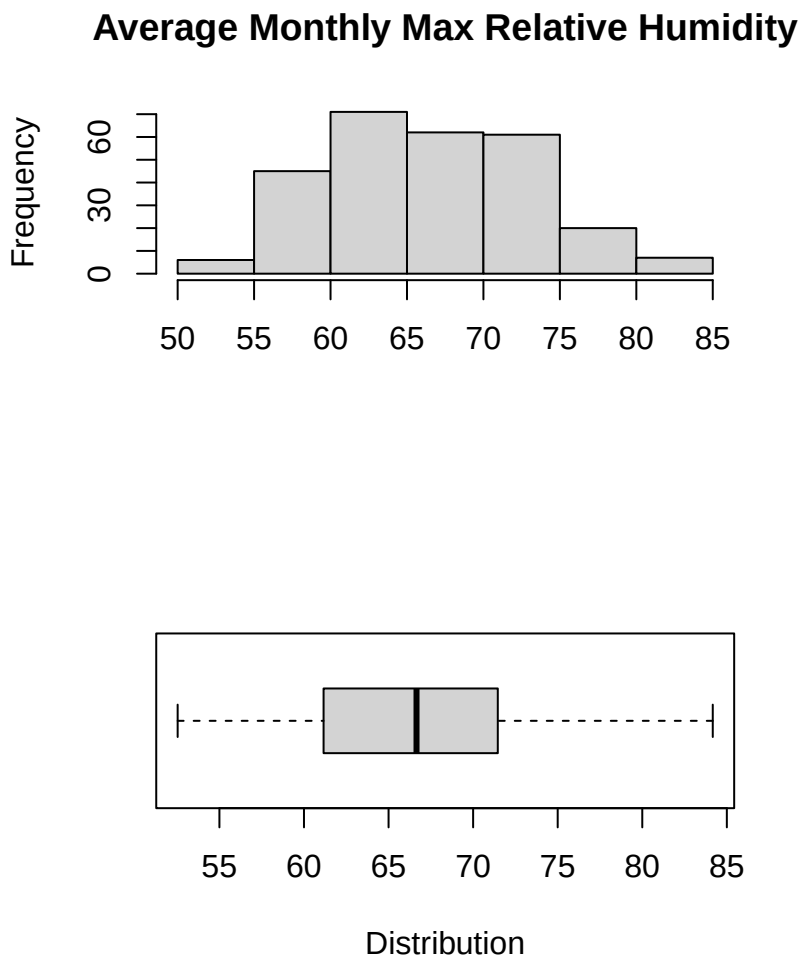
MDA8

```
par(mfrow=c(2,1))
hist(rough_variables$tmax,xlab=NULL, main="Average Monthly Max Temperature")
boxplot(rough_variables$tmax,xlab="Distribution", main=NULL, horizontal =T)
```



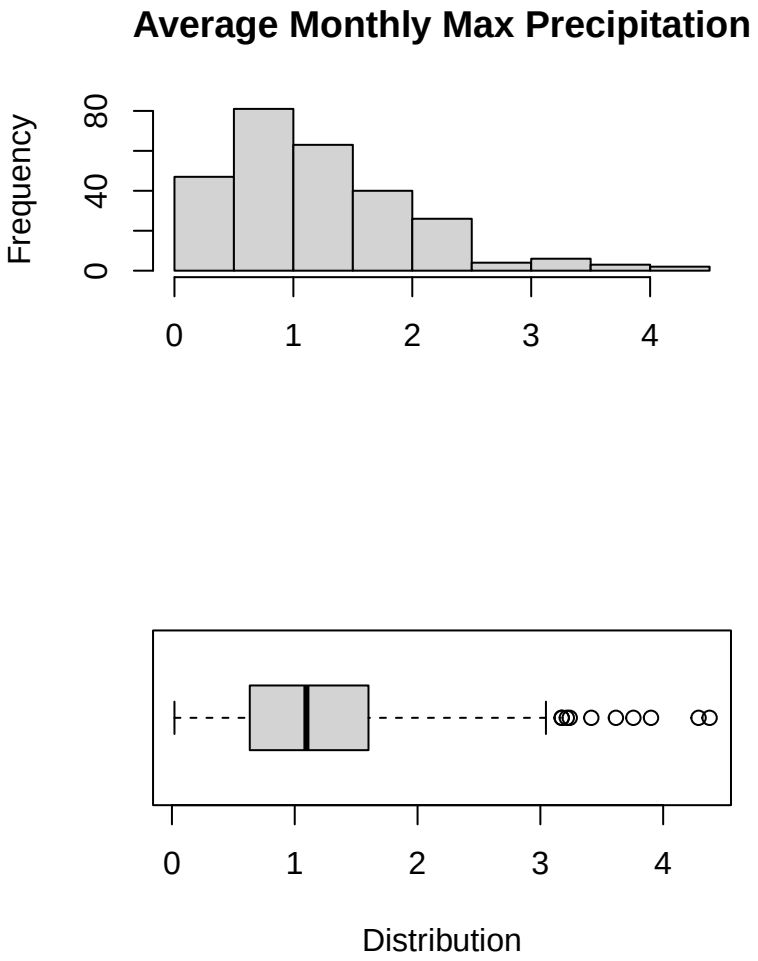
Maximum Temperature

```
par(mfrow=c(2,1))
hist(rough_variables$rhmax,xlab=NULL, main="Average Monthly Max Relative Humidity")
boxplot(rough_variables$rhmax,xlab="Distribution", main=NULL, horizontal =T)
```



Maximum Relative Humidity

```
par(mfrow=c(2,1))
hist(rough_variables$pmax,xlab=NULL, main="Average Monthly Max Precipitation")
boxplot(rough_variables$pmax,xlab="Distribution", main=NULL, horizontal =T)
```



Maximum Precipitation

Still Need:

- RF Model, "Leave one out" cross validation