IST 687 – M403

Brazil Weather, Deforestation, and Firespot Analysis

Final Project Report: Team 1

Lei Cheng
Katie Haugh
Garen Moghoyan
Sandy Spicer
Karl Treen

Table of Contents

Business Questions/Problem Statement

With climate change worsening, our planet is experiencing intensifying droughts, heat waves, and storms. Deforestation in the Brazilian Amazon has been on the rise for over a decade and is widely considered to be a primary contributor to climate change. The goal of our project was to determine if there is a relationship between deforestation, firespots, and weather throughout Brazil. We worked with four separate datasets to identify correlations between the individual variables, focusing on exploring drivers of deforestation and how the climate was changing. To better understand this complicated problem, we challenged ourselves to answer the following business questions:

Business Questions

1. What causes deforestation?
2. How has deforestation changed over the past 20 years?
3. What causes temperature change?
4. How has temperature changed over the past 20 years?
5. Are deforestation and temperature related?

Data Acquisition

Firespots

We collected four separate datasets to understand the relationships between deforestation, firespots, and weather in Brazil. The "firespots" dataset was obtained from Kaggle. This data is broken down by state, month, and year from 1999 to 2019. The original data was extracted from INPE (Instituto Nacional de Pesquisas Espaciais), Brazil's National Institute for Space Research. The dataset shows fire outbreaks that came from satellite images.

Firespots Dataset

year	month	state	state abbrev	latitude	longitude	firespots
1999	1	AMAZONAS	AM	-2.3711133	-59.899933	3
1999	1	MARANHAO	MA	-2.2573947	-45.487831	36
1999	1	MATO GROS	MT	-12.660633	-55.057989	18
1999	1	PARA	PA	-2.4748205	-48.546967	87
1999	1	RONDONIA	RO	-12.8617	-60.5131	1
1999	1	RORAIMA	RR	3.40322467	-60.622853	15
1999	2	AMAPA	AP	-0.155	-52.6831	1
1999	2	AMAZONAS	AM	-2.763167	-63.429781	43
1999	2	MATO GROS	MT	-12.619988	-55.375363	8
1999	2	PARA	PA	-2.1506174	-53.509911	285

Deforestation

The deforestation data is in squared kilometers by year and state, from 1988 to 2020. The original data was extracted from INPE, and we obtained it from Kaggle. According to PRODES (Programa de Monitoramento da Floresta Amazônica Brasileira por Satélite, or Brazilian Amazon Rainforest Monitoring Program by Satellite), it maps primary forest loss using satellite imagery. It has 20-30 meters of spatial resolution and seeks to minimize cloud cover.

Deforestation Dataset

yr	area	st	state abbreviation
1995	7,845.00	PARA	PA
2004	8,870.00	PARA	PA
2003	7,145.00	PARA	PA
2002	7,510.00	PARA	PA
1988	6,990.00	PARA	PA
2005	5,899.00	PARA	PA
2000	6,671.00	PARA	PA
2001	5,237.00	PARA	PA
1996	6,135.00	PARA	PA
1989	5,750.00	PARA	PA

Additionally, we collected shape files representative of yearly deforestation in the Brazilian Amazon from 2005-2019 to use as a visual representation of the growth in forest degradation.

Weather

The main weather data was sourced directly from the Brazilian government's meteorological database, INMET (https://bdmep.inmet.gov.br/). INMET has a network of both conventional and automatic stations throughout Brazil. We decided to focus solely on conventional weather station data due to the newer nature of the automatic stations and their lack of comprehensive data dating back to 1999. We requested the daily data for all conventional weather stations in Brazil and received separate csv files for each station containing all the standard weather data from January 1, 1999, through December 31, 2019. Our final dataset included the geolocation information of each conventional weather station and its associated state and altitude, which was also collected from Kaggle.

Weather Dataset

Nome: BOA VISTA								
Codigo Estacao: 82024								
Latitude: 2.82944443								
Longitude: -60.66194444								
Altitude: 84.18								
Situacao: Operante								
Data Inicial: 1999-01-01								
Data Final: 2019-12-31								
Periodicidade da Medicao: Diaria								
Data Medicao	EVAPORACAO DO PICHE, DIARIA(mm)	INSOLACAO TOTAL, DIARIO(h)	PRECIPITACAO TOTAL, DIARIO(mm)	TEMPERATURA MAXIMA, DIARIA(¬∞C)	TEMPERATURA MEDIA COMPENSADA, DIARIA(¬∞C)	TEMPERATURA MINIMA, DIARIA(¬∞C)	UMIDADE RELATIVA DO AR, MEDIA DIARIA(%)	VENTO, VELOCIDADE MEDIA DIARIA(m/s)
1/1/99	7	0	1,8	32,4	25,68	23,2	92,5	2,666,667
1/2/99	2,4	4	0	32,2	27,56	24	76,25	2,333,333
1/3/99	8,4	5,3	0	32,6	28,04	25,4	71	2,666,667
1/4/99	4,8	6,2	0	32,6	28,28	24,6	72	3,333,333
1/5/99	3,5	4,2	0	33	27,76	24	71,25	2
1/6/99	7	1,4	0	32,2	27,8	24,4	75,75	2
1/7/99	6,1	2,5	0	32,6	28,12	25	78,5	3
1/8/99	5,9	2,4	,9	32,6	28,24	25	79	2,666,667
1/9/99	4,1	0	0	31	27,2	25,2	79,75	2,333,333
1/10/99	6,8	3,4	0	32,8	26,96	23,4	79,5	2,333,333
1/11/99	4	2,3	4,3	32,6	28	24,2	70,25	3,333,333
1/12/99	5,8	4,3	0	32,6	27,76	23,6	72,75	2,333,333
1/13/99	5,8	7,2	0	32,8	27,84	24,4	74,75	2,333,333
1/14/99	6,5	6,3	0	33,4	28,6	24,4	64	3
1/15/99	8,8	4,3	0	33	28,6	24,8	75,5	3,666,667
1/16/99	7,2	,6	0	31	27,2	24,6	80	2,333,333
1/17/99	4,4	1,1	1,8	32,2	26,32	23,6	86,75	1

Data Cleaning, Transformation, Architecture

The deforestation & firespots data was already clean but cleaning the weather dataset was more involved. We received a separate csv file for each weather station, so our first task was to combine all 251 files into a single dataset.

```r
#Turns large list into dataframe
BrazilWeather <- rbind.fill(weather_data)


#Removes weather_data data that is no longer needed
rm(weather_data)
rm(file_names)



#Removes #Creates a list of files which is equivalent to their station name
file_names <- list.files("Data", pattern="*.csv", full.names = FALSE)

#Combines all files data
weather_data <- lapply(file_names, function(x) {
  a <- paste0("Data/", x)
  BrazilWeatherData <- read_csv2(a, skip = 10)
  cbind(station = x, BrazilWeatherData)
})
```

Original BrazilWeather Dataframe

	station	Data Medicao	EVAPORACAO DO PICHE, DIARIA(mm)	INSOLACAO TOTAL, DIARIO(h)	PRECIPITACAO TOTAL, DIARIO(mm)	TEMPERATURA MAXIMA, DIARIA(°C)	TEMPERATURA MEDIA COMPENSADA, DIARIA(°C)	TEMPERATURA MINIMA, DIARIA(°C)	UMIDADE RELATIVA DO AR, MEDIA DIARIA(%)	VENTO, VELOCIDADE MEDIA DIARIA(m/s)	X10	X3
1	82024.csv	1999-01-01	7	0	1.8	32,4	25,68	23,2	92,5	2,666667	NA	NA
2	82024.csv	1999-01-02	2,4	4	0	32,2	27,56	24	76,25	2,333333	NA	NA
3	82024.csv	1999-01-03	8,4	5,3	0	32,6	28,04	25,4	71	2,666667	NA	NA
4	82024.csv	1999-01-04	4,8	6,2	0	32,6	28,28	24,6	72	3,333333	NA	NA
5	82024.csv	1999-01-05	3,5	4,2	0	33	27,76	24	71,25	2	NA	NA
6	82024.csv	1999-01-06	7	1,4	0	32,2	27,8	24,4	75,75	2	NA	NA
7	82024.csv	1999-01-07	6,1	2,5	0	32,6	28,12	25	78,5	3	NA	NA
8	82024.csv	1999-01-08	5,9	2,4	0.9	32,6	28,24	25	79	2,666667	NA	NA
9	82024.csv	1999-01-09	4,1	0	0	31	27,2	25,2	79,75	2,333333	NA	NA
10	82024.csv	1999-01-10	6,8	3,4	0	32,8	26,96	23,4	79,5	2,333333	NA	NA
11	82024.csv	1999-01-11	4	2,3	4.3	32,6	28	24,2	70,25	3,333333	NA	NA
12	82024.csv	1999-01-12	5,8	4,3	0	32,6	27,76	23,6	72,75	2,333333	NA	NA
13	82024.csv	1999-01-13	5,8	7,2	0	32,8	27,84	24,4	74,75	2,333333	NA	NA
14	82024.csv	1999-01-14	6,5	6,3	0	33,4	28,6	24,4	64	3	NA	NA
15	82024.csv	1999-01-15	8,8	4,3	0	33	28,6	24,8	75,5	3,666667	NA	NA
16	82024.csv	1999-01-16	7,2	,6	0	31	27,2	24,6	80	2,333333	NA	NA
17	82024.csv	1999-01-17	4,4	1,1	1.8	32,2	26,32	23,6	86,75	1	NA	NA
18	82024.csv	1999-01-18	2,1	2,4	3.1	34,4	27,88	23,8	79,75	2,666667	NA	NA

Once the weather data was consolidated, we removed unnecessary columns, translated, and renamed the columns from Portuguese into English, and removed the ".csv" from the station codes. Additionally, because the data is in Portuguese, commas are used in instances where we would typically use periods. We substituted those commas for periods to be able to evaluate the data.

```r
#Removes columns not needed
BrazilWeather <- BrazilWeather[, -c(11:12)]
```

```r
colnames(BrazilWeather) <- c('Station', 'Date', 'Evaporation', 'Total_Insolat
ion', 'Precipitation', 'Max_Temperature', 'Avg_Temperature', 'Min_Temperature
', 'Humidity', 'Wind_Speed')
```

```r
#Renames columns

colnames(BrazilWeather) <- c('Station', 'Date', 'Evaporation', 'Total_Insolation', 'P
recipitation', 'Max_Temperature', 'Avg_Temperature', 'Min_Temperature', 'Humidity', '
Wind_Speed')

#Removes unnecessary info from stations column
BrazilWeather$Station <- gsub(".csv", "", BrazilWeather$Station)

#Replaces commas into periods and changes them to numeric
BrazilWeather[-1:-2] <- data.frame(lapply(BrazilWeather[-1:-2], function(x) {
  as.numeric(gsub(",", ".", x))
  }))
```

Cleaned BrazilWeather

	Station	Date	Evaporation	Total_Insolation	Precipitation	Max_Temperature	Avg_Temperature	Min_Temperature	Humidity	Wind_Speed
1	82024	1999-01-01	7.0	0.0	1.8	32.4	25.68	23.2	92.50	2.666667
2	82024	1999-01-02	2.4	4.0	0.0	32.2	27.56	24.0	76.25	2.333333
3	82024	1999-01-03	8.4	5.3	0.0	32.6	28.04	25.4	71.00	2.666667
4	82024	1999-01-04	4.8	6.2	0.0	32.6	28.28	24.6	72.00	3.333333
5	82024	1999-01-05	3.5	4.2	0.0	33.0	27.76	24.0	71.25	2.000000
6	82024	1999-01-06	7.0	1.4	0.0	32.2	27.80	24.4	75.75	2.000000
7	82024	1999-01-07	6.1	2.5	0.0	32.6	28.12	25.0	78.50	3.000000
8	82024	1999-01-08	5.9	2.4	0.9	32.6	28.24	25.0	79.00	2.666667
9	82024	1999-01-09	4.1	0.0	0.0	31.0	27.20	25.2	79.75	2.333333
10	82024	1999-01-10	6.8	3.4	0.0	32.8	26.96	23.4	79.50	2.333333

We then merged the data with the weather station csv using the station code. At this time, we discovered that several of the individual weather station csv files that we originally brought in had completely null data. From our research, that was due to those stations no longer in operation. The merge ended up removing those stations bringing the weather station count down to 191.

```r
#Import Weather Station Data
WeatherStations <- read.csv("Weather Stations/WeatherStations.csv")

#Merge Brazil Weather with Weather Stations
BrazilWeather <- merge(BrazilWeather, WeatherStations, by = "Station", all =
FALSE)
rm(WeatherStations)
```

BrazilWeather Combined with WeatherStations

	Station	Date	Evaporation	Total_Insolation	Precipitation	Max_Temperature	Avg_Temperature	Min_Temperature	Humidity	Wind_Speed	State	Latitude	Longitude	Altitude
1	82024	2014-03-02	8.4	6.2	0.0	35.6	29.10	24.4	60.75	1.666667	RR	2.82	-60.66	83
2	82024	2014-12-25	NA	9.5	0.0	35.1	29.34	25.4	62.50	1.371840	RR	2.82	-60.66	83
3	82024	2014-04-24	2.6	0.5	11.4	30.9	26.22	24.9	91.00	0.666667	RR	2.82	-60.66	83
4	82024	2014-05-23	6.8	10.4	0.0	36.3	29.12	24.4	58.00	1.000000	RR	2.82	-60.66	83
5	82024	2014-05-29	1.7	6.2	0.0	32.6	28.26	25.0	76.25	1.666667	RR	2.82	-60.66	83
6	82024	2014-12-26	NA	9.3	0.0	35.6	26.66	24.9	75.50	1.028880	RR	2.82	-60.66	83
7	82024	2014-05-25	5.2	9.8	7.6	35.9	29.38	25.0	67.75	1.000000	RR	2.82	-60.66	83
8	82024	2014-03-13	6.8	9.5	0.0	37.4	30.72	24.8	54.00	1.666667	RR	2.82	-60.66	83
9	82024	2009-05-26	8.0	5.8	0.0	34.3	29.76	25.5	60.00	3.333333	RR	2.82	-60.66	83
10	82024	2009-05-21	3.5	10.6	0.0	35.4	29.34	24.8	64.75	1.666667	RR	2.82	-60.66	83

Now that the weather data was combined with the geolocation data of the associated weather stations, we investigated a summary of the data. We discovered a considerable number of NAs.

```
> summary(BrazilWeather)
   Station               Date              Evaporation       Total_Insolation Precipitation
 Length:1487980     Min.   :1999-01-01   Min.   : 0.0     Min.   : 0.00    Min.   :  0.0
 Class :character   1st Qu.:2004-04-01   1st Qu.: 2.1     1st Qu.: 4.20    1st Qu.:  0.0
 Mode  :character   Median :2009-07-01   Median : 3.5     Median : 7.60    Median :  0.0
                    Mean   :2009-07-01   Mean   : 4.3     Mean   : 6.73    Mean   :  4.0
                    3rd Qu.:2014-10-01   3rd Qu.: 5.7     3rd Qu.: 9.60    3rd Qu.:  1.8
                    Max.   :2019-12-31   Max.   :99.9     Max.   :27.50    Max.   :295.8
                                         NA's   :223013   NA's   :197251   NA's   :40175
 Max_Temperature Avg_Temperature Min_Temperature    Humidity        Wind_Speed        State
 Min.   : 1.00   Min.   :-2.86   Min.   :-7.20   Min.   :  0.00   Min.   : 0.00   Length:1487980
 1st Qu.:28.40   1st Qu.:22.60   1st Qu.:17.80   1st Qu.: 64.50   1st Qu.: 0.87   Class :character
 Median :31.20   Median :25.38   Median :20.80   Median : 76.00   Median : 1.63   Mode  :character
 Mean   :30.58   Mean   :24.49   Mean   :19.84   Mean   : 73.33   Mean   : 1.86
 3rd Qu.:33.50   3rd Qu.:27.16   3rd Qu.:22.80   3rd Qu.: 84.00   3rd Qu.: 2.67
 Max.   :44.70   Max.   :35.60   Max.   :30.70   Max.   :100.00   Max.   :97.00
 NA's   :97479   NA's   :134382  NA's   :76136   NA's   :120311   NA's   :163338
    Latitude        Longitude         Altitude
 Min.   :-33.51   Min.   :-70.76   Min.   :   1.84
 1st Qu.:-18.23   1st Qu.:-50.86   1st Qu.:  74.04
 Median :-11.20   Median :-45.12   Median : 277.45
 Mean   :-12.76   Mean   :-46.54   Mean   : 366.92
 3rd Qu.: -6.46   3rd Qu.:-41.13   3rd Qu.: 605.34
 Max.   :  2.82   Max.   :-34.86   Max.   :1296.12
```

Before combining the data with the monthly firespot data and the yearly deforestation data, we addressed the NAs in the Brazil Weather data frame. We first identified 156 stations (81.7% of stations) with more than six NAs per row. We then created a data frame that counted those rows by station. We observed that most of the stations with missing rows were missing consecutive days, months, or years of data. Any station having more than one year or a count of 365 rows of missing data was then eliminated from the dataset. This reduced our station count to 116 stations.

```
#returns stations with null values across all columns
nullStations <- BrazilWeather[rowSums(is.na(BrazilWeather)) > 6, ]
nullStationsCount <- data.frame(tapply(nullStations$Date, nullStations$Statio
n, length))
colnames(nullStationsCount) <- "Null Rows By Station"
nullStationsCount$Station <- row.names(nullStationsCount)
BrazilWeather <- merge(BrazilWeather, nullStationsCount, by = "Station", all
= TRUE)
rm(nullStations)
rm(nullStationsCount)

#Shows unique stations before eliminating NULL rows
uniqueStationsBefore <- data.frame(unique(BrazilWeather$Station))

#Removes stations with more than 365 days of missing rows
BrazilWeather <-BrazilWeather[BrazilWeather$`Null Rows By Station` <= 365, ]
#Removes the rows that have blank date
BrazilWeather <- BrazilWeather[!is.na(BrazilWeather$Date), ]

#Shows unique stations after eliminating NULL rows
uniqueStationsAfter <- data.frame(unique(BrazilWeather$Station))
```

The remaining stations still had several NAs in the main weather variables. In rows with both maximum and minimum temperature readings, we began by replacing any NAs in the “Avg_Temperature” column with a calculated value. To replace NAs in remaining columns, we summarized the data by station, year, and month, and replaced NAs with each variable's monthly average. While we still had a few NAs because of several stations missing entire months, we decided to proceed with introducing the firespots data.

```
#Creates Avg Temp for those rows  with NAs by Averaging the Max and Min temp
for the day
BrazilWeather$Avg_Temperature_Calculated <- (BrazilWeather$`Max_Temperature`
+ BrazilWeather$`Min_Temperature`)/2
BrazilWeather$`Avg_Temperature`[is.na(BrazilWeather$`Avg_Temperature`)] <- Br
azilWeather$Avg_Temperature_Calculated[is.na(BrazilWeather$`Avg_Temperature`)
]

#Removes columns no longer needed
BrazilWeather <- BrazilWeather[, -c(15:16)]


#must separate out month from the date to fill NAs
BrazilWeather <- separate(BrazilWeather, Date, sep="-", into = c("Year", "Mon
th", "Day"))

#Replace NAs with Medians by group
#plyr causes problems moving forward
detach(package:plyr)

#Summarizes Each Monthly to Replace NAs
BrazilWeather <- BrazilWeather %>%
  group_by(Station, Year, Month) %>%
  mutate(
    across(c('Evaporation', 'Total_Insolation', 'Precipitation', 'Max_Tempera
ture', 'Avg_Temperature', 'Min_Temperature', 'Humidity', 'Wind_Speed'), funct
ion(x) ifelse(is.na(x), median(x, na.rm = TRUE), x))
  )
```

monthlySummary Table

	State	Month	Year	Evaporation	Total_Insolation	Precipitation	Max_Temperature	Avg_Temperature	Min_Temperature	Humidity	Wind_Speed	Latitude	Longitude	Altitude	Firespots
505	AM	1	1999	1.553763	3.265438	12.835484	30.92545	25.81269	22.31685	88.47133	1.2212665	-3.79	-64.14111	59.50778	3
506	AM	1	2000	1.704301	3.056631	9.007527	31.79677	25.97699	22.12079	89.15502	1.0917563	-3.79	-64.14111	59.50778	7
507	AM	1	2001	1.357706	3.262097	14.044803	31.27670	25.59211	22.06810	88.75806	0.9479092	-3.79	-64.14111	59.50778	3
508	AM	1	2002	1.841935	4.634050	7.320789	32.79606	26.70602	22.99319	86.99373	0.9451613	-3.79	-64.14111	59.50778	17
509	AM	1	2003	2.501971	5.981362	5.545520	33.87097	27.68165	23.95018	82.98208	0.9253286	-3.79	-64.14111	59.50778	379
510	AM	1	2004	1.919355	4.448992	5.468100	33.08931	27.20387	23.71577	85.45817	0.9484466	-3.79	-64.14111	59.50778	277
511	AM	1	2005	1.909319	5.106272	5.235842	33.68817	27.51559	23.46039	83.95072	0.9828899	-3.79	-64.14111	59.50778	225
512	AM	1	2006	1.610215	4.235081	9.513262	32.69176	26.55430	22.97706	87.01658	1.0420550	-3.79	-64.14111	59.50778	79
513	AM	1	2007	1.834677	4.273835	9.459498	32.78756	27.06129	23.35735	86.39695	0.9859020	-3.79	-64.14111	59.50778	106
514	AM	1	2008	1.583154	3.802509	11.575627	32.01075	26.38287	22.73047	87.54570	0.9915173	-3.79	-64.14111	59.50778	45

The Brazil Weather data is daily station data, while the firespots data is monthly state data. Before we could merge the two datasets, we created a monthly summarized table of the mean Brazil Weather data grouped by state, year, and month. This allowed us to merge the firespots data easily. By introducing the firespots data, several new NAs were introduced. We identified three states with entire null rows and removed them from the data frame.

```
#Make Summarized Monthly Table of Data by State
monthlySummary <- BrazilWeather %>% group_by(State, Year, Month) %>% summaris
e_all(list(~ mean(.x, na.rm = TRUE)))

#Removes Unnecessary Columns
monthlySummary <- monthlySummary[, -c(4:5)]
#Write to CSV
write.csv(monthlySummary, "monthlySummary.csv")
```

```r
#Import Firespots Data
Firespots <- read.csv("Firespots/Firespots.csv")
Firespots <- Firespots[,-c(3,5,6)]
colnames(Firespots) <- c("Year","Month","State","Firespots")


#Brazil Months as integer
monthlySummary$Month <- as.integer(monthlySummary$Month)

#Merge Firespots with Monthly Summary
monthlySummary <- merge(monthlySummary, Firespots, by = c("State","Month", "Y
ear"), all = TRUE)
rm(Firespots)

#Clean Monthly Summary
nullStations <- monthlySummary[rowSums(is.na(monthlySummary)) > 7, ]
unique(nullStations$State)

#Remove null stations from monthlySummary
monthlySummary <- monthlySummary[monthlySummary$State != "AP",]
monthlySummary <- monthlySummary[monthlySummary$State != "MS",]
monthlySummary <- monthlySummary[monthlySummary$State != "RO",]
```

The final step in transforming the data was to introduce the deforestation dataset. Because the deforestation data is yearly by state, a yearly summary table was created. Taking the monthly summarized table, we averaged each numeric variable for each year and created a new data frame. We then merged in the deforestation data. And finally, we added an additional column for cumulative deforestation.

```r
#Make Summarized Yearly Table of Data by State
yearlySummary <- monthlySummary %>% group_by(State, Year) %>% summarise_all(l
ist(~ mean(.x, na.rm = TRUE)))

#Removes Unnecessary Columns
yearlySummary <- yearlySummary[, -c(3)]

#Import Deoforestation Data
Deforestation <- read.csv("Deforestation/Deforestation.csv")
Deforestation <- Deforestation[, -c(3)]
colnames(Deforestation) <- c("Year","Deforestation","State")

#Merge Deforestation with Yearly Summary
yearlySummary <- merge(yearlySummary, Deforestation, by = c("State", "Year"),
all = FALSE)
yearlySummary$Deforestation <- as.numeric(gsub(",", "", yearlySummary$Defores
tation))
rm(Deforestation)

#create cumulative deforestation column
yearlySummary <- yearlySummary %>% group_by(State) %>% mutate("cumDeforestati
on" = cumsum(Deforestation))
```

	State	Year	Evaporation	Total_Insolation	Precipitation	Max_Temperature	Avg_Temperature	Min_Temperature	Humidity	Wind_Speed	Latitude	Longitude	Altitude	Firespots	Deforestation	cumDeforestation
1	AC	1999	2.447499	4.524032	6.257196	31.40802	24.74001	20.79586	87.47166	0.7291824	-9.060000	-69.28000	175.00000	69.40000	441	441
2	AC	2000	2.094493	4.543378	5.036988	31.43823	24.70946	21.05904	87.97649	0.8010068	-9.060000	-69.28000	175.00000	53.75000	547	988
3	AC	2001	2.034249	5.092666	5.536310	31.30431	25.30911	21.45769	87.56626	0.8072740	-9.060000	-69.28000	175.00000	138.16667	419	1407
4	AC	2002	2.112231	5.352059	6.161145	31.59053	25.66758	21.56137	84.93653	0.8842496	-9.060000	-69.28000	175.00000	1140.71429	883	2290
5	AC	2003	1.906048	5.077472	5.601683	31.66619	25.49067	21.13239	84.95510	0.9262191	-9.060000	-69.28000	175.00000	1052.30000	1078	3368
6	AC	2004	1.677022	4.558174	6.582030	31.24140	25.32448	21.56987	86.61381	0.9400588	-9.060000	-69.28000	175.00000	727.10000	728	4096
7	AC	2005	1.962091	5.289476	5.470004	32.01201	25.67683	21.44123	83.54843	0.9564184	-9.060000	-69.28000	175.00000	1453.90909	592	4688
8	AC	2006	2.170226	5.079948	5.593507	31.75794	25.52376	21.30403	84.62213	0.8108396	-9.060000	-69.28000	175.00000	619.80000	398	5086
9	AC	2007	1.887846	5.162986	6.068698	31.69373	25.47412	21.03265	83.14415	0.7793576	-9.060000	-69.28000	175.00000	777.18182	184	5270
10	AC	2008	1.930083	4.941962	5.391402	31.14222	25.35464	21.44908	83.47736	0.8064923	-9.060000	-69.28000	175.00000	569.90000	254	5524

Variable Definitions

Table: BrazilWeather	
Column Name	Definition
Station	Weather Station Code
Year	Year of Record
Month	Month of Record
Day	Day of Record
Evaporation_(mm)	Surface Evaporation in Millimeters
Total_Insolation_(h)	Total Incoming Solar Radiation in Hours
Precipitation_(mm)	Precipitation in Millimeters
Max_Temperature_(°C)	Maximum Temperature in ºC
Avg_Temperature_(°C)	Average Temperature in ºC
Min_Temperature_(°C)	Minimum Temperature in ºC
Humidity_(%)	Average Relative Humidity as Percent
Wind_Speed_(m/s)	Average Wind Speed in Meters/Second
State	State Abbreviation
Latitude	Latitude
Longitude	Longitude
Altitude	Altitude in Meters

Table: Firespots	
Column Name	Definition
year	Year of Record
month	Month of Record
state	State Abbreviation
latitude	Latitude
longitude	Longitude
firespots	Number of Forest Fire Outbreaks

Table: Deforestation	
Column Name	Definition
yr	Year of Record
area	Deforested Area in Kilometers
st	State Name

Table: monthlySummary	
Column Name	Definition
State	State Abbreviation
Month	Month of Record
Year	Year of Record
Evaporation_(mm)	Surface Evaporation in Millimeters
Total_Insolation_(h)	Total Incoming Solar Radiation in Hours
Precipitation_(mm)	Precipitation in Millimeters
Max_Temperature_(°C)	Maximum Temperature in ºC
Avg_Temperature_(°C)	Average Temperature in ºC
Min_Temperature_(°C)	Minimum Temperature in ºC
Humidity_(%)	Average Relative Humidity as Percent
Wind_Speed_(m/s)	Average Wind Speed in Meters/Second
Latitude	Latitude
Longitude	Longitude
Altitude	Altitude in Meters
Firespots	Number of Forest Fire Outbreaks

Table: yearlySummary	
Column Name	Definition
State	State Abbreviation
Year	Year of Record
Evaporation_(mm)	Surface Evaporation in Millimeters
Total_Insolation_(h)	Total Incoming Solar Radiation in Hours
Precipitation_(mm)	Precipitation in Millimeters
Max_Temperature_(°C)	Maximum Temperature in ºC
Avg_Temperature_(°C)	Average Temperature in ºC
Min_Temperature_(°C)	Minimum Temperature in ºC
Humidity_(%)	Average Relative Humidity as Percent
Wind_Speed_(m/s)	Average Wind Speed in Meters/Second
Latitude	Latitude
Longitude	Longitude
Altitude	Altitude in Meters
Firespots	Number of Forest Fire Outbreaks
Deforestation	Deforested Area in Kilometers

Table: Table for Significance Testing across Station-Seasons over 21 years	
Column Name	Definition
Station	The specific station
Season	The specific season
slope	The slope of the regression line drawn through the 21 years
Std_err	The standard error of the regression line
degf	The degrees of Freedom for the individual station-season (year count, 21-1)
T_stat	The t-statistic derived after calculating the Slope/Std_Err
P_val	The p-value derived from the T-distribution using the T-stat and Degf
Significance	Whether the p-value was below 0.05

Brazilian State Coordinates from geobr library's read_state() function	
Column Name	Definition
code_state	A 2-digit Code Identifying Each State
abbrev_state	State Abbreviation
name_state	State Name
code_region	A Code Identifying the Region In Which State Is Located
name_region	The Name of the Region In Which State Is Located
geom	Geometric Coordinates Describing State Outline

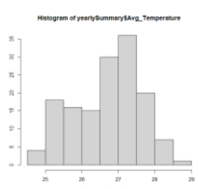
Analysis

Our analysis began by looking at the central tendency and distribution of the data.

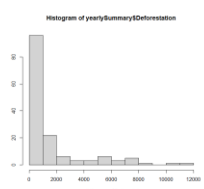
```
> summary(yearlySummary[, c(3,4,5,6,7,8,9,10,13,14,15)])
  Evaporation      Total_Insolation Precipitation    Max_Temperature Avg_Temperature Min_Temperature
 Min.   : 1.238   Min.   :4.250    Min.   :2.665    Min.   :31.14   Min.   :24.71   Min.   :19.33  
 1st Qu.: 2.363   1st Qu.:5.330    1st Qu.:4.452    1st Qu.:32.50   1st Qu.:25.91   1st Qu.:21.58  
 Median : 3.266   Median :6.279    Median :5.359    Median :33.04   Median :26.85   Median :22.37  
 Mean   : 3.591   Mean   :6.114    Mean   :5.380    Mean   :33.01   Mean   :26.68   Mean   :22.23  
 3rd Qu.: 4.485   3rd Qu.:6.776    3rd Qu.:6.266    3rd Qu.:33.50   3rd Qu.:27.34   3rd Qu.:23.05  
 Max.   :12.400   Max.   :7.487    Max.   :8.167    Max.   :34.82   Max.   :28.59   Max.   :24.73  
 NA's   :10                                                                                          
    Humidity        Wind_Speed        Altitude         Firespots       Deforestation    
 Min.   :60.56   Min.   :0.4761   Min.   : 59.51   Min.   :  20.0   Min.   :   23.0  
 1st Qu.:73.48   1st Qu.:0.8212   1st Qu.: 71.50   1st Qu.: 197.6   1st Qu.:  240.5  
 Median :78.37   Median :0.9920   Median :175.00   Median : 629.5   Median :  595.0  
 Mean   :77.57   Mean   :1.0341   Mean   :179.07   Mean   :1199.9   Mean   : 1525.5  
 3rd Qu.:83.92   3rd Qu.:1.2605   3rd Qu.:317.88   3rd Qu.:1390.6   3rd Qu.: 1461.5  
 Max.   :87.98   Max.   :1.5737   Max.   :374.26   Max.   :8904.1   Max.   :11814.0  
```

Descriptive Statistics

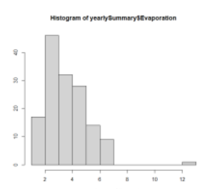
Average Temperature



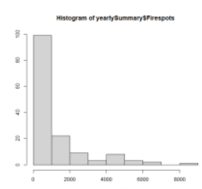
Deforestation



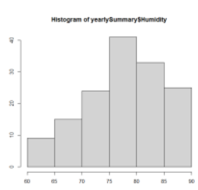
Evaporation



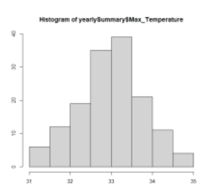
Firespots



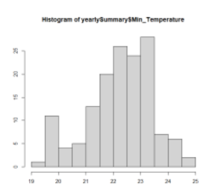
Humidity



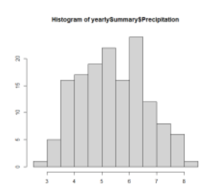
Max Temperature



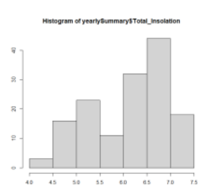
Min Temperature



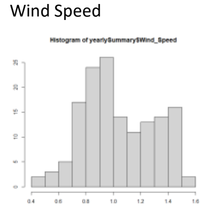
Precipitation



Total Insolation



Wind Speed



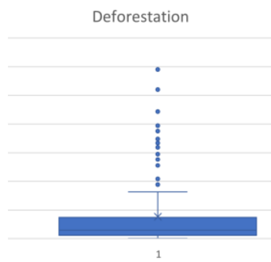
Running descriptive statistics on our various variables showed

- Fairly symmetrical skewness for total insolation, precipitation, maximum temperature, average temperature, wind speed and humidity,
- High positive skewness for evaporation, deforestation and firespots,

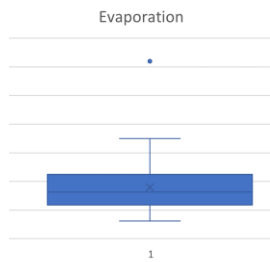
- Moderate negative skewness for minimum temperature.

Kurtosis indicates a peaked distribution for deforestation, firespots and evaporation - indicating heavy outliers in our data. Running boxplots for each, we identify the outliers.

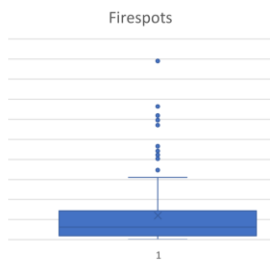
Deforestation



Evaporation

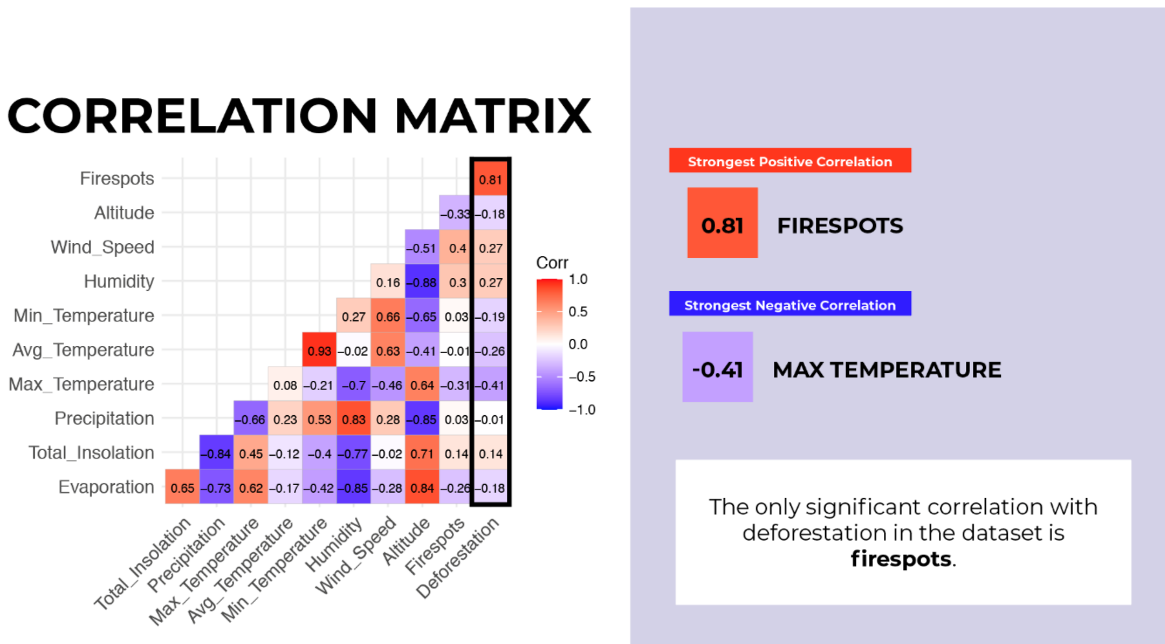


Firespots

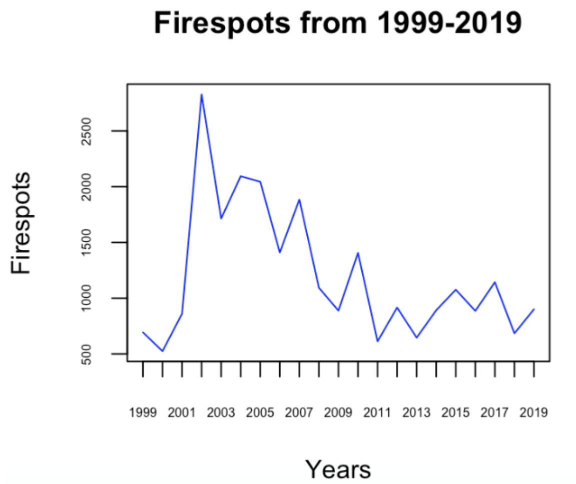


After running the descriptive statistics on the combined deforestation, firespots, and weather dataset, we constructed a correlation matrix to determine which variables to focus on. As evidenced below, firespots is the only variable with strong positive correlation (0.81) to deforestation, while max temperature has the strongest negative correlation (-0.41). This was a very surprising result. Not only were we expecting to see more strong correlations with deforestation, but we were not expecting to see a negative correlation with temperature.

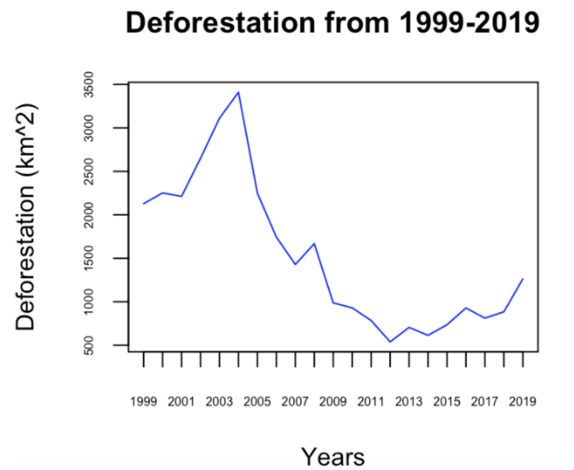
Because of this revelation, we decided to take a multi-pronged approach with our project evaluating not only the relationship between deforestation and fire outbreaks, but also how temperature may influence deforestation, and vice-versa. We explored the correlation between firespots and deforestation in detail before dissecting temperature trends and relationships between other variables.



After observing the above correlation, we wanted to understand how deforestation and firespots changed over time. The following plots were created to show how both trends had progressed through our observed period. The results, for each, showed a sharp increase, and then an unexpected decrease, in activity.



The chart above shows “firespots” (i.e., number of fire outbreaks) over 20 years from 1999 through 2019. We can see a sharp increase in firespots in the early 2000s which steadily decreased over time.



The chart above shows deforestation in km^2 over 20 years from 1999 to 2019. Like firespots, we see a sharp increase in deforestation in the early 2000s which slowly drops off over time. 2014 onward shows a steady increase.

In both trend charts, we can see a large peak in the early 2000s followed by a decrease over time. This was non-intuitive. We expected both wildfires and deforestation increasing in tandem with warming temperatures. The highest firespots occurred in Para in the year 2002 whereas the most deforestation occurred in Mato Grosso in the year 2004. When we look at the top 10 instances of firespot outbreaks and deforestation by state, we see that Para and Mato Grosso take the lead in the early 2000s as shown in the tables below.

TOP 10 DEFORESTATION

Year	State	Deforestation (km^2)
2004	MT	11814
2003	MT	10405
2004	PA	8870
2002	MT	7892
2001	MT	7703
2002	PA	7510
2005	MT	7145
2003	PA	7145
1999	MT	6963
2000	PA	6671

Mato Grosso (MT) Pará (PA)

60% OF TOP SPOTS

40% OF TOP SPOTS

TOP 10 FIRESPOTS

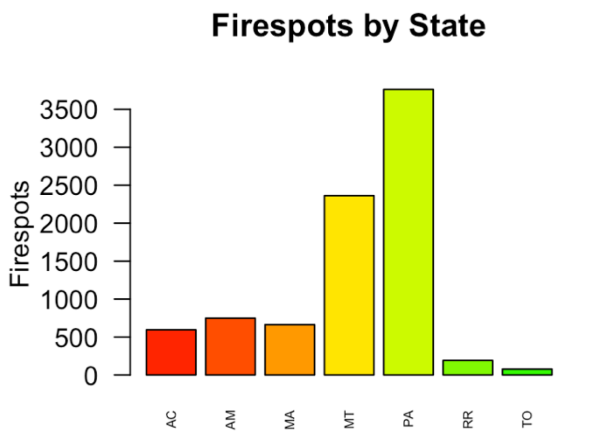
Year	State	Firespots
2002	PA	8904.08333
2002	MT	6640.00000
2004	PA	6184.50000
2005	PA	5956.41667
2004	MT	5868.50000
2007	PA	5707.58333
2010	PA	4766.33333
2006	PA	4653.33333
2005	MT	4457.41667
2003	PA	4420.00000

Mato Grosso (MT) Pará (PA)

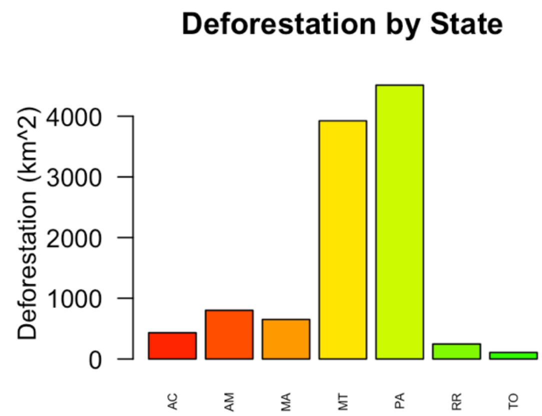
30% OF TOP SPOTS

70% OF TOP SPOTS

We noticed a pattern with certain states being more affected by firespots and deforestation than others and wanted to dig deeper. We then plotted both firespots and deforestation by state to better understand what regions were more affected than others.



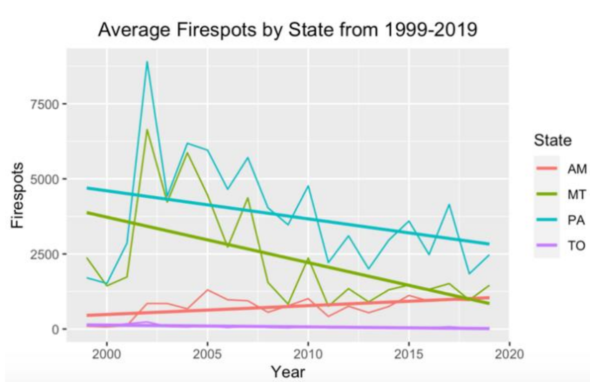
The chart above shows Para and Mato Grosso as the two leading states for firespots. The states with the lowest amount of fire outbreaks are Amapa and Tocantins.



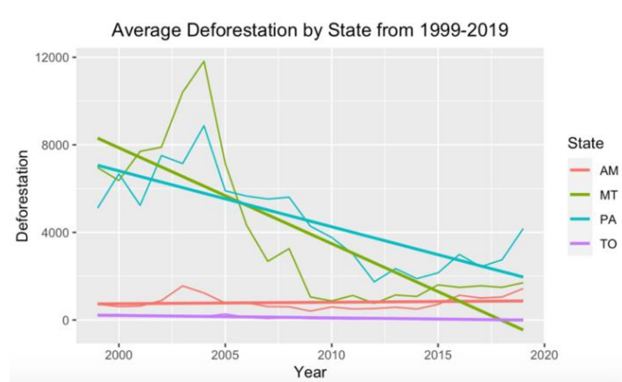
Above, we see a similar pattern with deforestation. Para and Mato Grosso are the highest deforested states whereas Amapa and Tocantins are the lowest.

Here, we see there are certain states that have significant deforestation and firespots. For detailed analysis, we decided to focus on four states with varying levels of firespots and deforestation. These states were: Para, Mato Grosso, Amazonas, and Tocantins. Both Para and Mato Grosso had high levels of deforestation and firespots, so we wanted to make sure to include at least two of the top drivers. Amazonas had moderate deforestation and firespots. We also wanted to see if there were significant differences in deforestation and firespots where the Amazon Rainforest is located. Finally, Tocantins had low deforestation and firespots – possibly because it had and less rainforest at the beginning of the time we were studying.

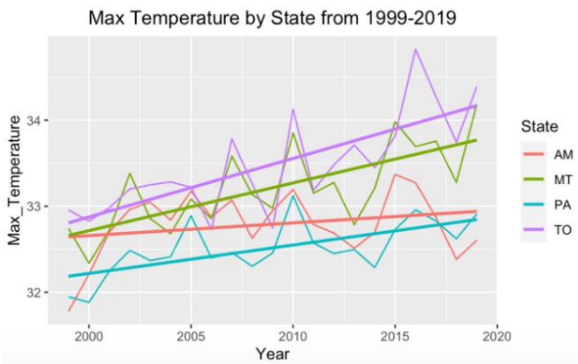
To dig even deeper into our four states, we decided to plot average firespots, deforestation, max temperature, and average temperature by state over the 20 years.



According to the plot above, all but one state shows decreasing trends of firespots

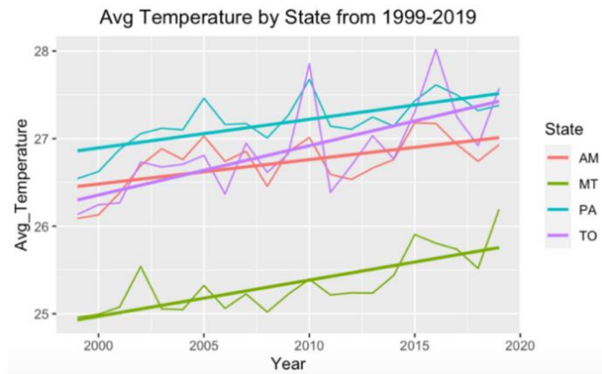


As shown in the plot above, deforestation has a decreasing trend over time for all states over time. Amazonas is the only state that has a slight increase over time since 1999.



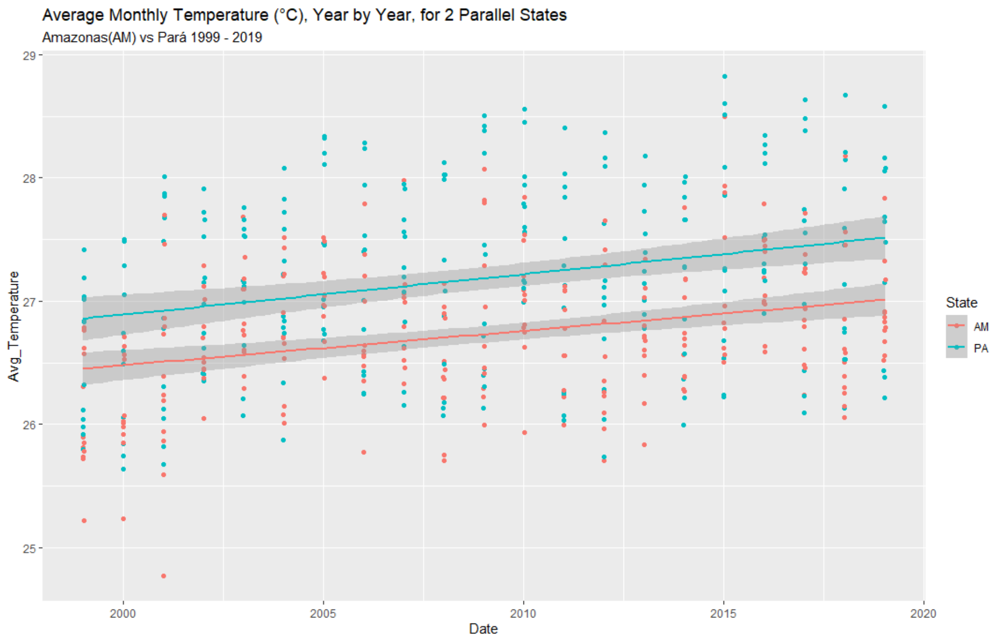
All states show a steadily increasing trends upward in max temperature

except Amazonas. There is a peak in the early 2000s that coincides with the firespots peak.



Similarly, we see a steady increase in average temperature over time for all states. Mato Grosso is roughly 1.5-2 degrees Celsius lower in average temperature compared to the other states. This could be due to the higher altitude of the state.

Drilling down into the average temperatures by state, we find an interesting difference between Pará and Amazonas. These states fall roughly along the same latitude. Both states have approximately the same average altitude for the stations we studied but, as can be seen below, the temperatures measured in Pará are consistently higher than those in its more densely forested neighbor. It is conceivable that this temperature difference is caused by differences in rainforest coverage, but there may be other variables that contribute to this difference. Standard deviation in temperature is also greater in Pará (.755) than Amazonas (.571), which may be the result of having less standing thermal mass (in the form of trees) to moderate temperature fluctuations.



Cumulative Firespots and Deforestation

While the firespots and deforestation, over time, were decreasing, deforestation, in Brazil's rainforest, appears to be cumulative. That is to say that maps show cumulative deforestation increasing over time. To illustrate this, we created animations.

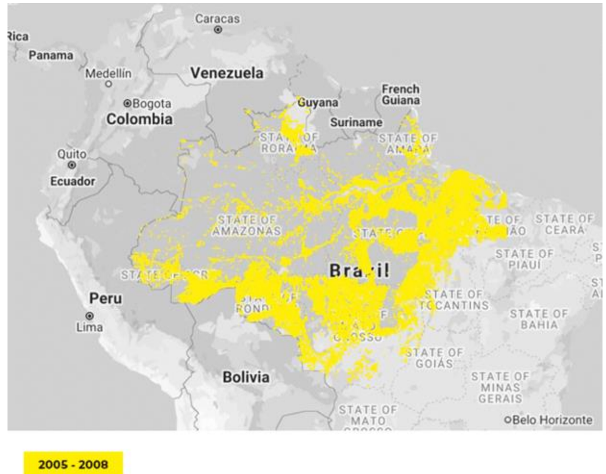
To create the firespots animation (below), data was collected from the firespots dataset. Data was then sliced into seven 3-year buckets. A topographical background map was created using the ggmap() function, and firespots were then layered, sequentially as dots, over the map using the geom_point() function. Each 3-year period was added in a different color, using the custom add.FireDotMap() function below, and snapshots were taken after each layer was added. It is important to note that our data included the number of observed fires but not the area consumed by fire. Dot size was, therefore, rendered using the log of the count of observed firespots and does not represent the actual size of the firespots themselves. After all buckets had been added, the individual snapshots were assembled into the final gif animation using Adobe Photoshop.

```
# adds firespots as dots of varying sizes, depending on the size of the obser
ved fire
add.FireDotMap <- function(saMap, fires, mapTitle="", mapSubTitle="", dotColo
r="#B00000")
{
  saMap <- saMap + geom_point(data=fires, aes(x=lon, y=lat), color=dotColor,
size=log(fires$firespots))
  saMap <- saMap + ggtitle(mapTitle) +
    xlab("") +
    ylab("") +
    labs(subtitle = mapSubTitle)
    theme(plot.title=element_text(hjust=0.5))
  return(saMap)
}
```

A similar process was used to create the deforestation map (below).

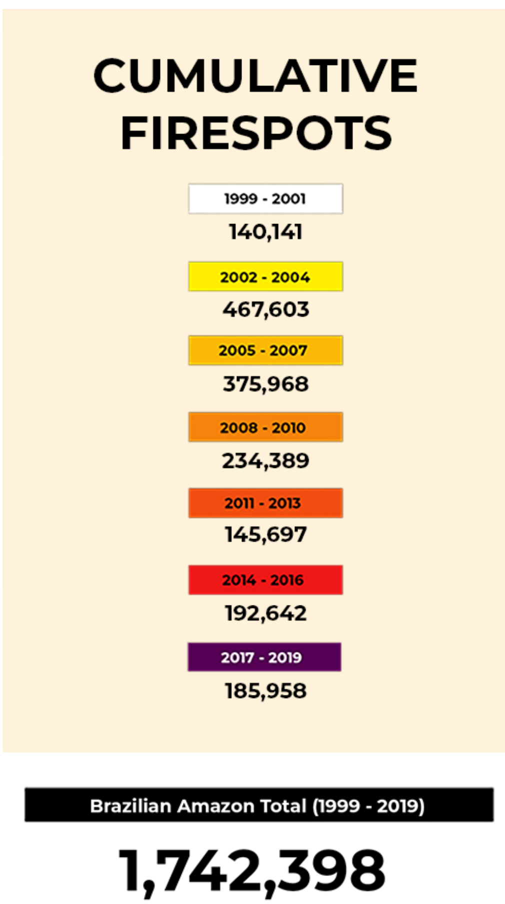
Brazil Firespots (Amazonas, Mato Grosso, Para, Tocantins)

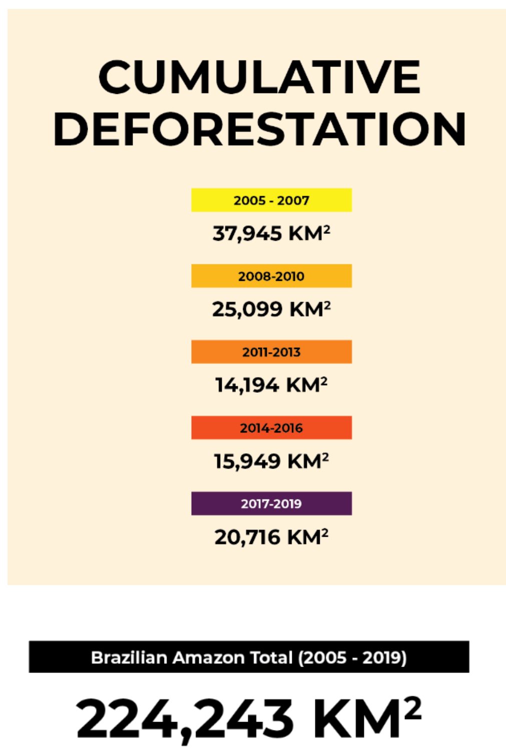
Deforestation, 2005 – 2019



For Office 365, use link to view animation:
[animation of firespots from 1999 – 2019]

For Office 365, use link to view animation:
[deforestation from 2005 – 2019]





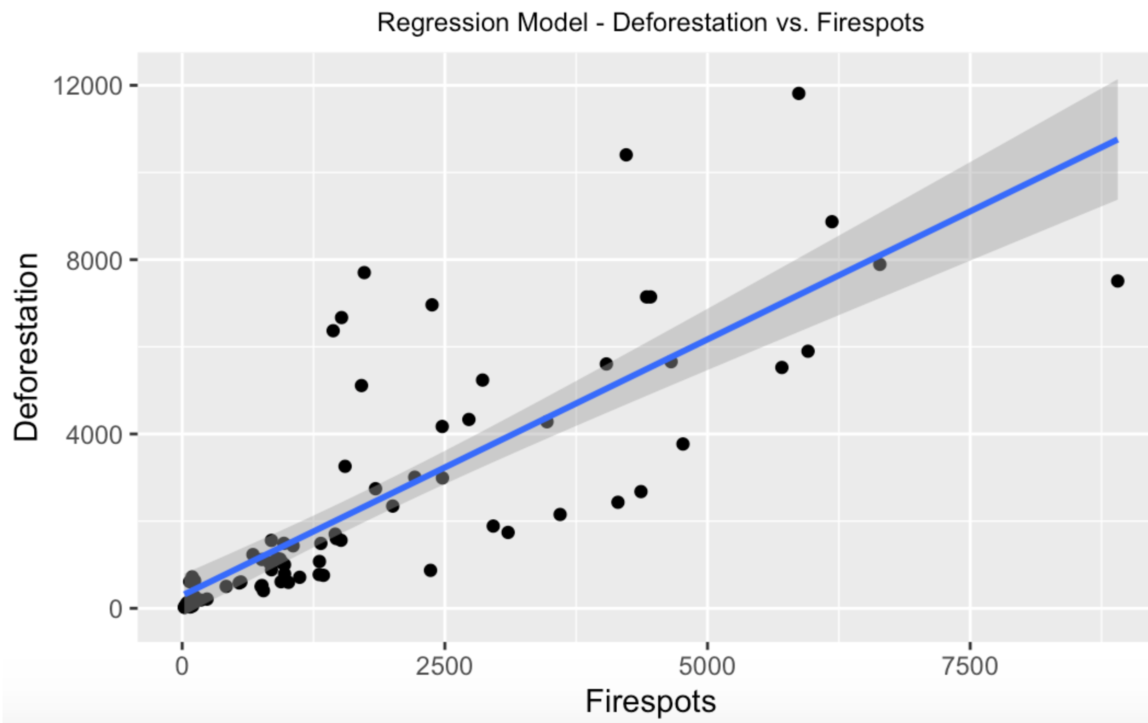
Modeling

To continue investigating variables that influence deforestation, we build a regression model utilizing all continuous variables in our refined yearly summary data as input variables to explain deforestation.

```
#Original Deforestation Regression
deforestationRegression <- lm(formula=Deforestation ~ Total_Insolation + Evap
oration + Precipitation + Max_Temperature + Avg_Temperature + Min_Temperature
+ Humidity + Wind_Speed + Altitude + Firespots, data=yearlySummaryRefined)
summary(deforestationRegression)
```

```
Call:
lm(formula = Deforestation ~ Total_Insolation + Evaporation +
    Precipitation + Max_Temperature + Avg_Temperature + Min_Temperature +
    Humidity + Wind_Speed + Altitude + Firespots, data = yearlySummaryRefined)

Residuals:
    Min      1Q  Median      3Q     Max
-2952.1  -865.3   -80.0   585.9  3963.1

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      13564.4947 38854.9667   0.349    0.728
Total_Insolation   -32.5355   528.5183  -0.062    0.951
Evaporation         96.3011   182.9362   0.526    0.600
Precipitation     -399.1492   406.6424  -0.982    0.330
Max_Temperature   -943.4497   809.0168  -1.166    0.248
Avg_Temperature   -305.8582  1602.9804  -0.191    0.849
Min_Temperature    167.7717  1002.4419   0.167    0.868
Humidity           250.6242   156.6937   1.599    0.114
Wind_Speed        2403.6214  1810.3413   1.328    0.189
Altitude            14.1683     9.0122   1.572    0.120
Firespots            1.0271     0.1289   7.969 2.28e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1447 on 69 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.7697,	Adjusted R-squared:  0.7363
F-statistic: 23.06 on 10 and 69 DF,  p-value: < 2.2e-16
```

While the regression model is significant and shows that 73.6% of deforestation is explained by these input variables, firespots is the only variable actually significant, which is in-line with our previous finding. We plotted the correlation discovering for every 1 km^2 of deforested land, there are 1.17509 firespots.

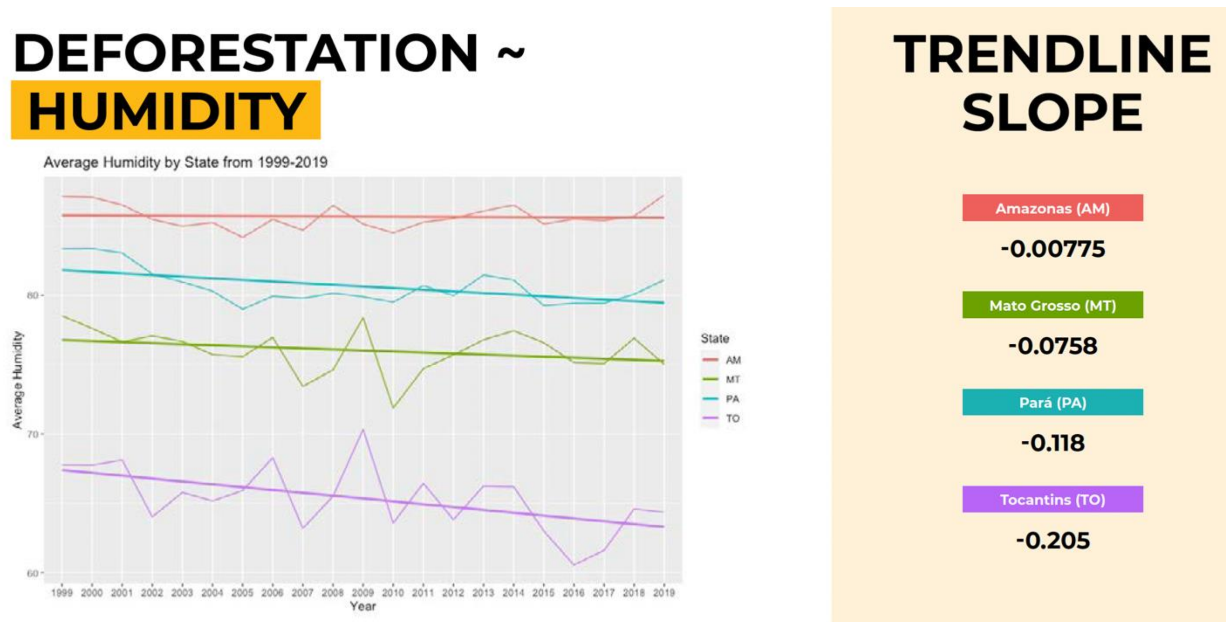


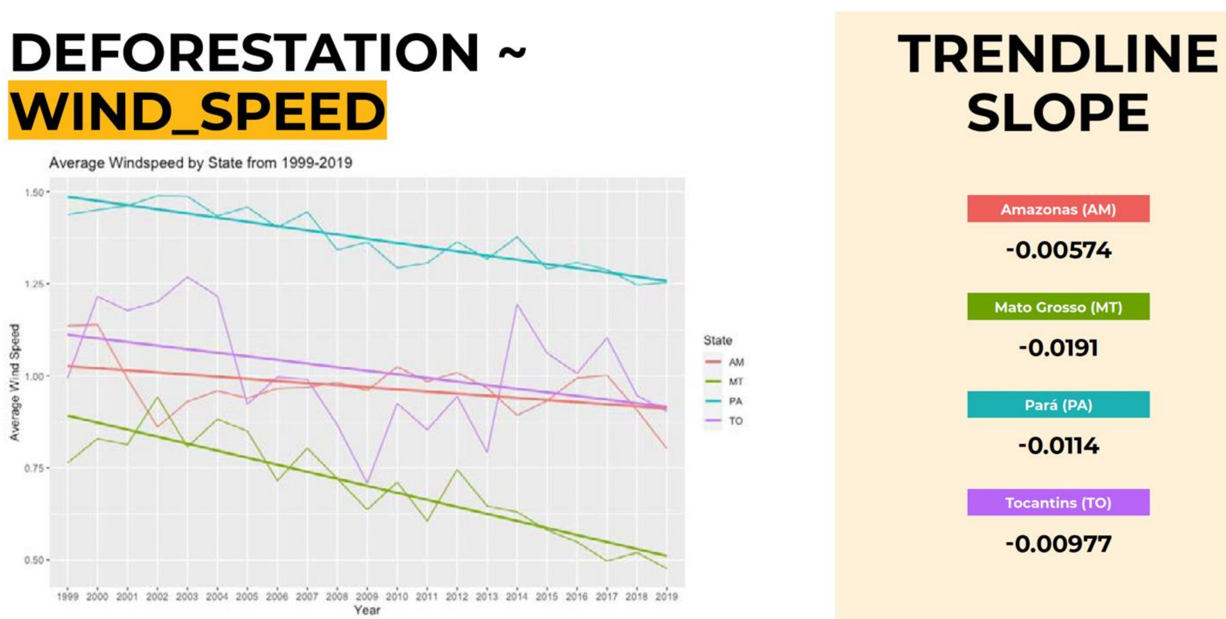
To refine the model, we removed the variable with the highest p-value and reran the regression until all the variables were significant.

```
#Deforestation Regression - All variables significant
deforestationRegression <- lm(formula=Deforestation ~ Humidity + Wind_Speed +
Altitude + Firespots, data=yearlySummaryRefined)
summary(deforestationRegression)
```

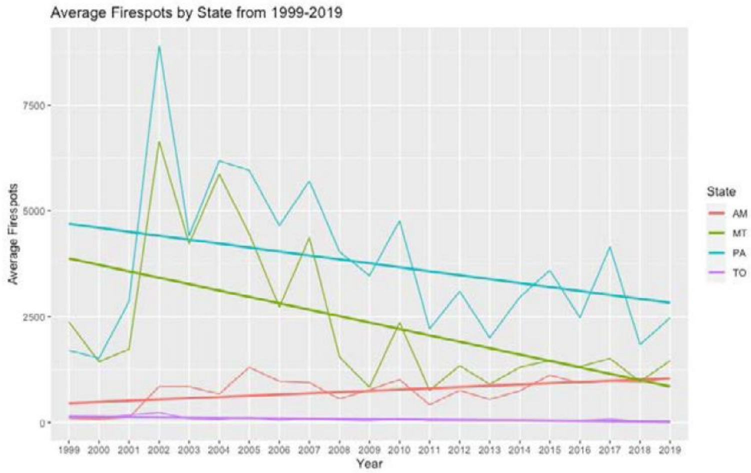
```
Call:
lm(formula = Deforestation ~ Humidity + Wind_Speed + Altitude +
    Firespots, data = yearlySummaryRefined)

Residuals:
    Min      1Q  Median      3Q     Max
-3078.3  -887.5  -101.1   663.3  3952.6

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.661e+04  6.581e+03  -5.563 3.50e-07 ***
Humidity     3.632e+02  6.472e+01   5.612 2.86e-07 ***
Wind_Speed   4.466e+03  1.076e+03   4.152 8.27e-05 ***
Altitude     2.268e+01  3.944e+00   5.749 1.62e-07 ***
Firespots    1.044e+00  9.332e-02  11.184  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1385 on 79 degrees of freedom
Multiple R-squared:  0.7607,	Adjusted R-squared:  0.7486
F-statistic:  62.8 on 4 and 79 DF,  p-value: < 2.2e-16
```

The refined regression model is significant and has improved upon our r-squared value. 74.8% of deforestation in Amazonas, Mato Gross, Para, and Tocantins is explained by Humidity, Wind Speed, Altitude, and Firespots. From the regression, we can observe that the coefficients are all positive therefore as each variable increases, deforestation also increase. To evaluate the effects of each of these variables, we looked at the trendlines of each variable and considered their slope.





DEFORESTATION ~ FIRESPOTS



TRENDLINE SLOPE

Amazonas (AM)

29.4

Mato Grosso (MT)

-151.0

Pará (PA)

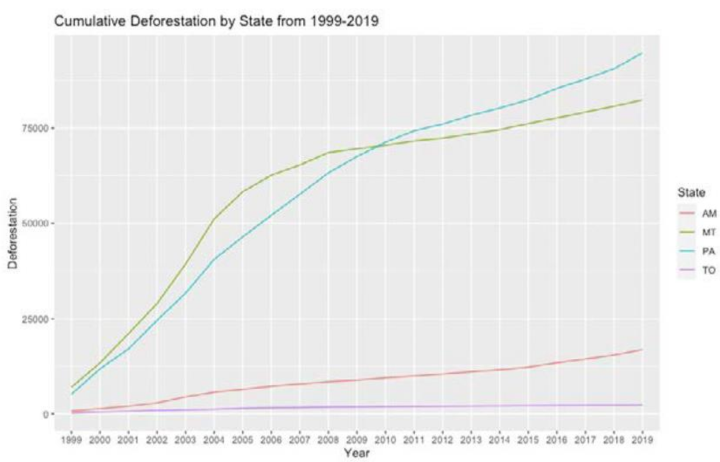
-93.3

Tocantins (TO)

-6.1

In each instance, all but one slope was negative showing that humidity, windspeed, and firespots are decreasing over time. This is consistent with our findings on cumulative deforestation. Deforestation is increasing every year, and repairing those forests take an assertive human effort that is currently not outpacing the rate of destruction. As we look at cumulative deforestation, we can see that is continues to climb, but at a slowing rate. This correlates with the decreasing trends we see with humidity, windspeed, and firespots.

DEFORESTATION CUMULATIVE



21-YEAR TOTALS

Amazonas (AM)

16,847 KM²

Mato Grosso (MT)

82,414 KM²

Pará (PA)

94,762 KM²

Tocantins (TO)

2,260 KM²

Deforestation is **increasing**, but at a **decreasing** rate since 2010

Our two-pronged approach also had us look at maximum temperature as a variable in the climate change discussion. We ran a regression with maximum temperature our dependent variable with deforestation, firespots, and the remaining continuous weather variables.

```
#Original Max_Temp Regression
maxtempRegression <- lm(formula=Max_Temperature ~ Total_Insolation + Evaporat
ion + Precipitation + Deforestation + Avg_Temperature + Min_Temperature + Hum
idity + Wind_Speed + Altitude + Firespots, data=yearlySummaryRefined)
summary(maxtempRegression)
```

```
Call:
lm(formula = Max_Temperature ~ Total_Insolation + Evaporation +
    Precipitation + Deforestation + Avg_Temperature + Min_Temperature +
    Humidity + Wind_Speed + Altitude + Firespots, data = yearlySummaryRefined)

Residuals:
     Min       1Q   Median       3Q      Max
-0.61605 -0.12914 -0.00586  0.12738  0.54133

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       2.568e+01  4.825e+00   5.323 1.20e-06 ***
Total_Insolation -2.110e-01  7.363e-02  -2.866  0.00551 **
Evaporation       2.992e-02  2.677e-02   1.118  0.26757
Precipitation    -9.596e-02  5.922e-02  -1.620  0.10974
Deforestation    -2.049e-05  1.757e-05  -1.166  0.24756
Avg_Temperature   1.095e+00  1.961e-01   5.586 4.29e-07 ***
Min_Temperature  -6.282e-01  1.269e-01  -4.950 5.05e-06 ***
Humidity         -5.420e-02  2.259e-02  -2.399  0.01915 *
Wind_Speed       -1.534e+00  1.972e-01  -7.782 5.00e-11 ***
Altitude         -2.579e-03  1.315e-03  -1.960  0.05402 .
Firespots         5.615e-05  2.544e-05   2.208  0.03060 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2132 on 69 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.8792,    Adjusted R-squared:  0.8617
F-statistic: 50.23 on 10 and 69 DF,  p-value: < 2.2e-16
```

From our regression with maximum temperature, we have a significant equation, and an adjusted r-squared of 0.8617. While most variables are significant, evaporation, precipitation, and deforestation are all not significant. Once again, we refined out model by removing the variable with the highest p-value and rerunning the regression until all variables were significant.

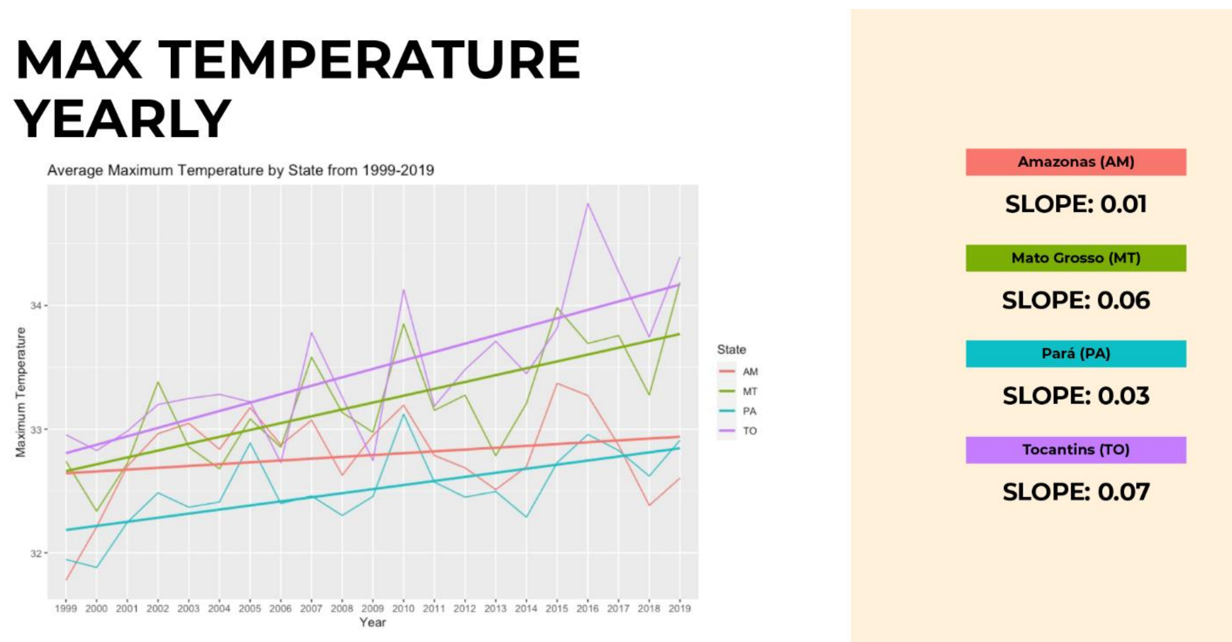
```
#Max Temperature Regression
maxtempRegression <- lm(formula=Max_Temperature ~ Total_Insolation + Avg_Temp
erature + Min_Temperature + Humidity + Wind_Speed + Firespots, data=yearlySum
maryRefined)
summary(maxtempRegression)
```

```
Call:
lm(formula = Max_Temperature ~ Total_Insolation + Avg_Temperature +
    Min_Temperature + Humidity + Wind_Speed + Firespots, data = yearlySummaryRefined)

Residuals:
     Min       1Q   Median       3Q      Max
-0.74562 -0.12062  0.00933  0.13635  0.55816

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       1.514e+01  2.238e+00   6.765 2.31e-09 ***
Total_Insolation -1.819e-01  7.206e-02  -2.525  0.01363 *
Avg_Temperature   1.475e+00  1.533e-01   9.628 7.44e-15 ***
Min_Temperature  -7.760e-01  1.104e-01  -7.032 7.24e-10 ***
Humidity         -2.185e-02  7.396e-03  -2.954  0.00416 **
Wind_Speed       -1.473e+00  1.513e-01  -9.742 4.50e-15 ***
Firespots         4.411e-05  1.834e-05   2.406  0.01855 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2204 on 77 degrees of freedom
Multiple R-squared:  0.8648,	Adjusted R-squared:  0.8543
F-statistic: 82.11 on 6 and 77 DF,  p-value: < 2.2e-16
```

While our adjusted r-squared dropped slightly to 0.8543, we have five significant variables. From our original correlation matrix, we can see that average temperature is correlated with both min temperature and wind speed. In order to meet linear regression assumptions, we should further investigate those assumptions and make the appropriate corrections. We corrected for the correlation between average temperature and min temperature, by removing average temperature. This created an insignificant variable of total insolation, so once again we removed that variable from the equation. What resulted was very similar to our initial regression with deforestation

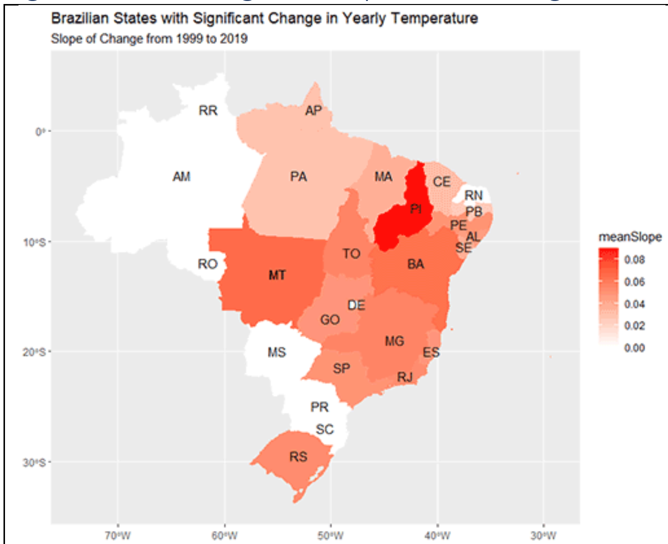
```
#Max Temperature Regression
maxtempRegression <- lm(formula=Max_Temperature ~ Min_Temperature + Humidity
+ Wind_Speed + Firespots, data=yearlySummaryRefined)
summary(maxtempRegression)
```

```
Call:
lm(formula = Max_Temperature ~ Min_Temperature + Humidity + Wind_Speed +
    Firespots, data = yearlySummaryRefined)

Residuals:
     Min       1Q   Median       3Q      Max
-1.19329 -0.20320  0.00558  0.18042  0.77215

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      3.371e+01  7.605e-01  44.322  < 2e-16 ***
Min_Temperature  2.373e-01  4.304e-02   5.513 4.29e-07 ***
Humidity        -5.538e-02  5.174e-03 -10.703  < 2e-16 ***
Wind_Speed      -1.750e+00  2.128e-01  -8.222 3.22e-12 ***
Firespots        6.254e-05  2.324e-05   2.691  0.00869 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3232 on 79 degrees of freedom
Multiple R-squared:  0.7019,    Adjusted R-squared:  0.6868
F-statistic: 46.51 on 4 and 79 DF,  p-value: < 2.2e-16
```



While we have observed an obvious visual increase in max daily temperatures, it is important to statistically prove that a real change had occurred and account for year-to-year variation. Therefore, we decided to dive into significance testing of long-term temperature/weather data.

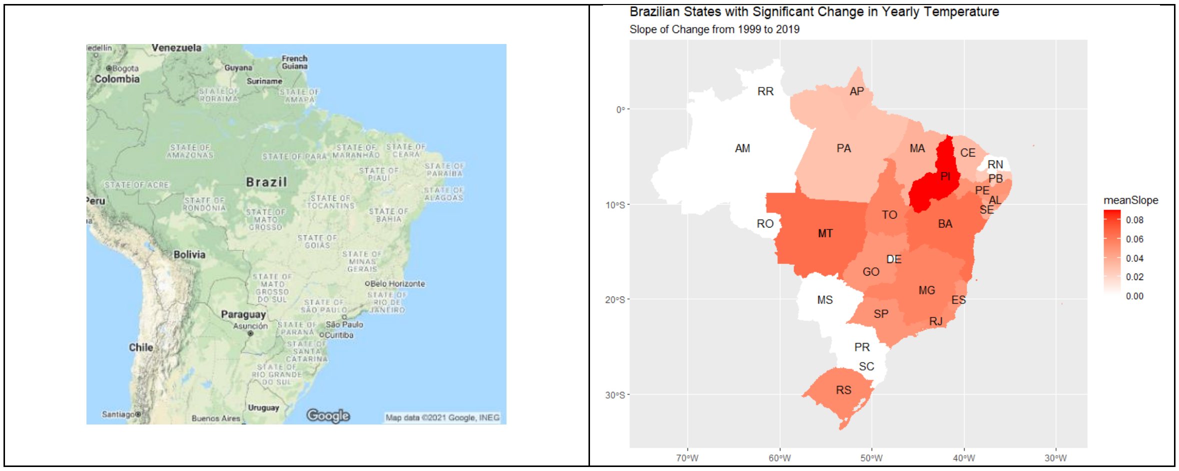
Significance Testing for Temperature Change within Stations



For Office 365, use link to view animation: animation of significant temp change.

Our analysis of temperature change used trends over the last 21 years (whole data). To demonstrate significant changes in average temperature by state, we created an animated map. This animation is based on a series of heat maps consolidated by season. Underlying data included weather station temperature readings for 1999 and 2019. We divided the dataset into four subsets: one set per season. To be included, a weather station must have recorded temperatures for a minimum of 75 days in any given season. After filtering for 75 days per season, we were left with 68 stations eligible for analysis. For each weather station, the change (slope) was calculated between 1999 and 2019 (point-to-point). Slope was then tested for significance with an alpha of .05, the null hypothesis being that the slope was 0 under a T-distribution. Records which rejected that null hypothesis were included in the final datasets. Finally, weather station records were grouped by state taking the mean temperature change for each state. Visualizations were built using ggplot in R and assembled into the animation with Photoshop. States with no significant change are shown in white, while states with significant change are shown from light to dark red to illustrate the increasing significance of the changes we discovered.

When juxtaposed against a google map showing remaining rainforest in green, it is interesting to observe that the most densely forested states appear to demonstrate less significant temperature rise over the 21 year period studied. This contrast may illustrate the mitigating effect of rainforest on local temperature change.



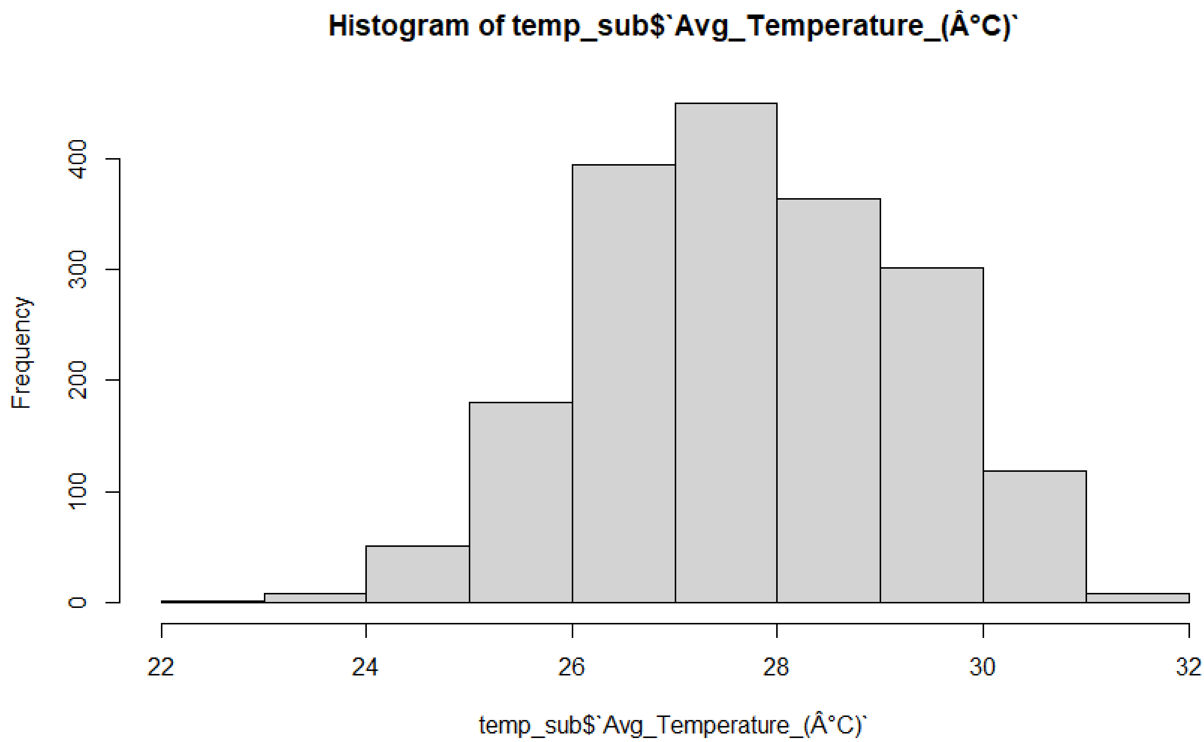
Our temperature change methodology requires a few assumptions:
1) The seasonal trends can be quantified by linear regression
2) The different seasonal record can be treated as independent records
3) Persistence in each of the seasonal records can be characterized by short-term memory (autoregressive process of first order)

Through statistical testing, we found that 58 out of the 68 stations (after the before-mentioned filtering for at least 75 days per season & no seasons) had at least 1 season with a significantly different average temperature.

Example table for results below:

Station	Season	slope	std_err	degf	t_stat	p_val	significant
82098	Winter	0.0302934550403131	0.0077472786778239	20	3.9102059	4.339137e-04	Reject
82181	Summer	0.0039000453443465	0.0169302629001726	20	0.2303594	4.100762e-01	Fail to Reject
82181	Fall	0.0243330384630873	0.0102975626679128	20	2.3629901	1.418481e-02	Reject
82184	Summer	0.0219232386660961	0.0115933086846064	20	1.8910252	3.659811e-02	Reject
82191	Spring	0.0230909215008594	0.00960068615981782	20	2.4051324	1.298334e-02	Reject
82191	Summer	0.022372104475125	0.0141772074124319	20	1.5780332	6.512223e-02	Fail to Reject
82191	Fall	0.0321562284076802	0.00997699432585189	20	3.2230377	2.132882e-03	Reject
82191	Winter	0.033818180320429	0.0106401918726053	20	3.1783431	2.361942e-03	Reject
82198	Spring	0.0267329851411852	0.00745219723801179	20	3.5872622	9.210175e-04	Reject
82263	Fall	0.0371068215893256	0.0107186552373898	20	3.4618915	1.231491e-03	Reject
82376	Spring	0.0325303867888092	0.0130545165159191	20	2.4918875	1.080218e-02	Reject

The T-distribution was used in this analysis because each individual significance test had n=20 and the overall distribution of average temperature looked to be normal:


Histogram of temp_sub$`Avg_Temperature_(Â°C)`
Frequency
0
100
200
300
400
22
24
26
28
30
32
temp_sub$`Avg_Temperature_(Â°C)`

Interpretation

Overall, firespots is the only variable that is strongly correlated with deforestation. In contrast, max temperature shows a moderately negative correlation with deforestation, which goes against our expectations. With the increase in the severity of the climate crisis, we would have expected to see a positive correlation with deforestation and temperature as well as a more significant increase in temperature over time.

There are possible reasons for this counter-intuitive negative correlation. Tocantins, the state with the greatest temperature rise, also had a very low rate of deforestation. However, Tocantins has less rainforest than the other three states we focused on. It is possible that remaining rainforest has continued to mitigate the effects of temperature rise in the other three states, whereas Tocantins is suffering the full brunt of global climate change, state-wide.

Further research needs to be done to understand additional drivers behind deforestation and temperature.

Summary Assessment, Actionable Steps

In conclusion, we sought to establish whether climate played a role in the deforestation phenomenon that has taken place in the Brazilian rain forest over the last two decades.
Our regression model found that 74% of the deforestation is explained by altitude, humidity, windspeed, and firespots.

But the overall increase in deforestation numbers cannot be explained by climate change alone and while trying to understand the phenomenon, we came across literature mentioning human intervention as a major factor. It seems that agriculture (the cultivation of soy products, the need for arable lands, the beef industry) plays an important role in the deforestation process.

We would argue that more research would be required in that area to form a better understanding of all drivers behind climate change and deforestation.

References

- Escobar, Herton. "Illegal deforestation in Brazil soars amid climate of impunity" *Science mag*, August 5, 2020, www.sciencemag.org/news/2020/08/illegal-deforestation-brazil-soars-amid-climate-impunity
- "Brazil: accelerating deforestation of Amazon a direct result of Bolsonaro's policies" *Amnesty International*, December 2, 2020, www.amnesty.org/en/latest/news/2020/12/brazil-accelerating-deforestation-of-amazon-a-direct-result-of-bolsonaros-policies
- Briggs, Helen. "Amazon soya and beef exports 'linked to deforestation'" *British Broadcasting Corporation*, July 18, 2020, www.bbc.com/news/science-environment-53438680
- Asher, Claire. "Illegal soy trade linked to widespread deforestation, carbon emissions" *Mongabay*, April 3, 2010, https://news.mongabay.com/2019/04/brazil-soy-trade-linked-to-widespread-deforestation-carbon-emissions/

Code

```
#Packages
install.packages("readr")
install.packages("tidyverse")
install.packages("stringr")
install.packages("plyr")
install.packages("dplyr")
install.packages("ggmap")
install.packages("tidyr")
install.packages("geobr")
install.packages("ggplot2")
install.packages("sf")
install.packages("sqldf")
install.packages("openxlsx")
install.packages("viridis")
install.packages("knitr")
install.packages("rgdal")
install.packages("ggmap")
install.packages("scales")
install.packages("rmapshaper")



#Libraries
library(readr)
library(tidyverse)
library(stringr)
library(plyr)
library(dplyr)
library(ggmap)
library(tidyr)
library(geobr)
library(ggplot2)
library(sf)
library(sqldf)
library(openxlsx)
library(viridis)
library(knitr)
library(rgdal)# R wrapper around GDAL/OGR
library(ggmap)
library(scales)
library(rmapshaper)


#Creates a list of files which is equivalent to their station name
file_names <- list.files("Data", pattern="*.csv", full.names = FALSE)
view(file_names)
```

```
#Combines all files data
weather_data <- lapply(file_names, function(x) {
  a <- paste0("Data/", x)
  BrazilWeatherData <- read_csv2(a, skip = 10)
  cbind(station = x, BrazilWeatherData)
})

#Turns large list into dataframe
BrazilWeather <- rbind.fill(weather_data)
#Removes weather_data data that is no longer needed
rm(weather_data)
rm(file_names)
rm(BrazilWeather)

#Removes columns not needed
BrazilWeather <- BrazilWeather[, -c(11:12)]

#Renames Columns
colnames(BrazilWeather)
colnames(BrazilWeather) <- c('Station', 'Date', 'Evaporation', 'Total_Insolation', 'Precipitation',
'Max_Temperature', 'Avg_Temperature', 'Min_Temperature', 'Humidity', 'Wind_Speed')

#Removes unnecessary info from stations column
BrazilWeather$Station <- gsub(".csv", "", BrazilWeather$Station)

#Replaces commas into periods and changes them to numeric
BrazilWeather[-1:-2] <- data.frame(lapply(BrazilWeather[-1:-2], function(x) {
  as.numeric(gsub(",", ".", x))
  }))

#Converts Date from character to date
BrazilWeather$Date <- as.Date(BrazilWeather$Date)
str(BrazilWeather)
#Backup to Brazil Weather Station
BackupBW <- BrazilWeather
BrazilWeather <- BackupBW

#Import Weather Station Data
WeatherStations <- read.csv("Weather Stations/WeatherStations.csv")

#Merge Brazil Weather with Weather Stations
BrazilWeather <- merge(BrazilWeather, WeatherStations, by = "Station", all = FALSE)
rm(WeatherStations)

#returns stations with null values across all columns
nullStations <- BrazilWeather[rowSums(is.na(BrazilWeather)) > 6, ]
nullStationsCount <- data.frame(tapply(nullStations$Date, nullStations$Station, length))
colnames(nullStationsCount) <- "Null Rows By Station"
nullStationsCount$Station <- row.names(nullStationsCount)
BrazilWeather <- merge(BrazilWeather, nullStationsCount, by = "Station", all = TRUE)
```

```
rm(nullStations)
rm(nullStationsCount)

#Shows unique stations before eliminating NULL rows
uniqueStationsBefore <- data.frame(unique(BrazilWeather$Station))

#Removes stations with more than 365 days of missing rows
BrazilWeather <-BrazilWeather[BrazilWeather$`Null Rows By Station` <= 365, ]
#Removes the rows that have blank date
BrazilWeather <- BrazilWeather[!is.na(BrazilWeather$Date), ]

#Shows unique stations after eliminating NULL rows
uniqueStationsAfter <- data.frame(unique(BrazilWeather$Station))
rm(uniqueStationsBefore)
rm(uniqueStationsAfter)

#Creates Avg Temp for those rows  with NAs by Averaging the Max and Min temp for the day
BrazilWeather$Avg_Temperature_Calculated <- (BrazilWeather$`Max_Temperature` +
BrazilWeather$`Min_Temperature`)/2
BrazilWeather$`Avg_Temperature`[is.na(BrazilWeather$`Avg_Temperature`)] <-
BrazilWeather$Avg_Temperature_Calculated[is.na(BrazilWeather$`Avg_Temperature`)]

#Removes columns no longer needed
BrazilWeather <- BrazilWeather[, -c(15:16)]


#must separate out month from the date to fill NAs
BrazilWeather <- separate(BrazilWeather, Date, sep="-", into = c("Year", "Month", "Day"))

#Replace NAs with Medians by group
#plyr causes problems moving forward
detach(package:plyr)

#Summarizes Each Monthly to Replace NAs
BrazilWeather <- BrazilWeather %>%
  group_by(Station, Year, Month) %>%
  mutate(
    across(c('Evaporation', 'Total_Insolation', 'Precipitation', 'Max_Temperature', 'Avg_Temperature',
'Min_Temperature', 'Humidity', 'Wind_Speed'), function(x) ifelse(is.na(x), median(x, na.rm = TRUE), x))
  )

#Creates backup to this point
BackupBW <- BrazilWeather

#Make Summarized Monthly Table of Data by State
monthlySummary <- BrazilWeather %>% group_by(State, Year, Month) %>% summarise_all(list(~ mean(.x,
na.rm = TRUE)))
#Removes Unnecessary Columns
monthlySummary <- monthlySummary[, -c(4:5)]
#Write to CSV
```

```
write.csv(monthlySummary, "monthlySummary.csv")

#Import Firespots Data
Firespots <- read.csv("Firespots/Firespots.csv")
Firespots <- Firespots[,-c(3,5,6)]
colnames(Firespots) <- c("Year","Month","State","Firespots")
#Brazil Months as integer
monthlySummary$Month <- as.integer(monthlySummary$Month)

#Merge Firespots with Monthly Summary
monthlySummary <- merge(monthlySummary, Firespots, by = c("State","Month", "Year"), all = TRUE)
rm(Firespots)

#Clean Monthly Summary
nullStations <- monthlySummary[rowSums(is.na(monthlySummary)) > 7, ]
unique(nullStations$State)
#Remove null stations from monthlySummary
monthlySummary <- monthlySummary[monthlySummary$State != "AP",]
monthlySummary <- monthlySummary[monthlySummary$State != "MS",]
monthlySummary <- monthlySummary[monthlySummary$State != "RO",]
#Remove nullStations
rm(nullStations)
#Write to CSV
write.csv(monthlySummary, "monthlySummary.csv")

#Make Summarized Yearly Table of Data by State
yearlySummary <- monthlySummary %>% group_by(State, Year) %>% summarise_all(list(~ mean(.x, na.rm
= TRUE)))
#Removes Unnecessary Columns
yearlySummary <- yearlySummary[, -c(3)]

#Import Deoforestation Data
Deforestation <- read.csv("Deforestation/Deforestation.csv")
Deforestation <- Deforestation[, -c(3)]
colnames(Deforestation) <- c("Year","Deforestation","State")

#Merge Deforestation with Yearly Summary
yearlySummary <- merge(yearlySummary, Deforestation, by = c("State", "Year"), all = FALSE)
yearlySummary$Deforestation <- as.numeric(gsub(",", "", yearlySummary$Deforestation))
rm(Deforestation)

#create cumulative deforestation column
yearlySummary <- yearlySummary %>% group_by(State) %>% mutate("cumDeforestation" =
cumsum(Deforestation))


#Write to CSV
write.csv(yearlySummary, "yearlySummary.csv")

#Start of Summary Analysis
```

```
install.packages("ggcorrplot")
library(ggcorrplot)


#Run Correlations for Monthly Summary All
corrMonth <- round(cor(monthlySummary[,c(4,5,6,7,8,9,10,11,14,15)], use = "complete.obs"), 2)
ggcorrplot(corrMonth, hc.order = FALSE, type = 'lower', lab = TRUE) +
  ggtitle("Monthly Summary Correlation Matrix")



#Run Correlations for Yearly Summary All
corrYear <- round(cor(yearlySummary[,c(3,4,5,6,7,8,9,10,13,14,15)], use = "complete.obs"), 2)
ggcorrplot(corrYear, hc.order = FALSE, type = 'lower', lab = TRUE) +
  ggtitle("Yearly Summary Correlation Matrix")

#Run Correlations for Yearly Summary All with cumulative deforestation
corrYear <- round(cor(yearlySummary[,c(3,4,5,6,7,8,9,10,13,14,16)], use = "complete.obs"), 2)
ggcorrplot(corrYear, hc.order = FALSE, type = 'lower', lab = TRUE) +
  ggtitle("Yearly Summary Correlation Matrix")

#Reduce yearly and monthly summaries to only Mato Grosso, Para, Amazonas, and Tocantins
monthlySummaryRefined <- monthlySummary[monthlySummary$State %in% c('PA', 'MT', 'AM', 'TO'), ]
yearlySummaryRefined <- yearlySummary[yearlySummary$State %in% c('PA', 'MT', 'AM', 'TO'), ]

#Run Correlations for Monthly Summary Refined by states we are focusing on
corrMonthRefined <- round(cor(monthlySummaryRefined[,c(4,5,6,7,8,9,10,11,14,15)], use =
"complete.obs"), 2)
ggcorrplot(corrMonthRefined, hc.order = FALSE, type = 'lower', lab = TRUE) +
  ggtitle("Monthly Summary Correlation Matrix Refined")

#Run Correlations for Yearly Summary Refined
corrYearRefined <- round(cor(yearlySummaryRefined[,c(3,4,5,6,7,8,9,10,13,14,15)], use = "complete.obs"),
2)
ggcorrplot(corrYearRefined, hc.order = FALSE, type = 'lower', lab = TRUE) +
  ggtitle("Yearly Summary Correlation Matrix Refined")

#Run Correlations for Yearly Summary Refined with Cumulative Deforestation
corrYearRefined <- round(cor(yearlySummaryRefined[,c(3,4,5,6,7,8,9,10,13,14,16)], use = "complete.obs"),
2)
ggcorrplot(corrYearRefined, hc.order = FALSE, type = 'lower', lab = TRUE) +
  ggtitle("Yearly Summary Correlation Matrix Refined")

#Regression Analysis -----------------------------------------------------------

#Original Deforestation Regression
deforestationRegression <- lm(formula=Deforestation ~ Total_Insolation + Evaporation + Precipitation +
Max_Temperature + Avg_Temperature + Min_Temperature + Humidity + Wind_Speed + Altitude +
Firespots, data=yearlySummaryRefined)
summary(deforestationRegression)
```

```
#Minus Total_Insolation
deforestationRegression <- lm(formula=Deforestation ~ Evaporation + Precipitation + Max_Temperature
+ Avg_Temperature + Min_Temperature + Humidity + Wind_Speed + Altitude + Firespots,
data=yearlySummaryRefined)
summary(deforestationRegression)
#Minus Min_temp
deforestationRegression <- lm(formula=Deforestation ~ Evaporation + Precipitation + Max_Temperature
+ Avg_Temperature + Humidity + Wind_Speed + Altitude + Firespots, data=yearlySummaryRefined)
summary(deforestationRegression)
#Minus Avg_Temp
deforestationRegression <- lm(formula=Deforestation ~ Evaporation + Precipitation + Max_Temperature
+ Humidity + Wind_Speed + Altitude + Firespots, data=yearlySummaryRefined)
summary(deforestationRegression)
#Minus Evapoaration
deforestationRegression <- lm(formula=Deforestation ~ Precipitation + Max_Temperature + Humidity +
Wind_Speed + Altitude + Firespots, data=yearlySummaryRefined)
summary(deforestationRegression)
#Minus Precipitation
deforestationRegression <- lm(formula=Deforestation ~ Max_Temperature + Humidity + Wind_Speed +
Altitude + Firespots, data=yearlySummaryRefined)
summary(deforestationRegression)
#Minus Max_Temp - All variables significant
deforestationRegression <- lm(formula=Deforestation ~ Humidity + Wind_Speed + Altitude + Firespots,
data=yearlySummaryRefined)
summary(deforestationRegression)



#Cumulative Deforestation Regression
summary(lm(formula=cumDeforestation ~ Total_Insolation + Evaporation + Precipitation +
Max_Temperature + Avg_Temperature + Min_Temperature + Humidity + Wind_Speed + Altitude +
Firespots, data=yearlySummary))

#Minus Average Temp
summary(lm(formula=cumDeforestation ~ Total_Insolation + Evaporation + Precipitation +
Max_Temperature + Min_Temperature + Humidity + Wind_Speed + Altitude + Firespots,
data=yearlySummary))

#Minus Max Temp
summary(lm(formula=cumDeforestation ~ Total_Insolation + Evaporation + Precipitation +
Min_Temperature + Humidity + Wind_Speed + Altitude + Firespots, data=yearlySummary))

#Cumulative Deforestation Regression with Max and Average Temp
summary(lm(formula=cumDeforestation ~ Max_Temperature, data=yearlySummary))
summary(lm(formula=cumDeforestation ~ Avg_Temperature, data=yearlySummary))



#Original Max_Temp Regression
```

```
maxtempRegression <- lm(formula=Max_Temperature ~ Total_Insolation + Evaporation + Precipitation +
Deforestation + Avg_Temperature + Min_Temperature + Humidity + Wind_Speed + Altitude + Firespots,
data=yearlySummaryRefined)
summary(maxtempRegression)
#Minus Evaporation
maxtempRegression <- lm(formula=Max_Temperature ~ Total_Insolation + Precipitation + Deforestation
+ Avg_Temperature + Min_Temperature + Humidity + Wind_Speed + Altitude + Firespots,
data=yearlySummaryRefined)
summary(maxtempRegression)
#Minus Deforestation
maxtempRegression <- lm(formula=Max_Temperature ~ Total_Insolation + Precipitation +
Avg_Temperature + Min_Temperature + Humidity + Wind_Speed + Altitude + Firespots,
data=yearlySummaryRefined)
summary(maxtempRegression)
#Minus Precipitation
maxtempRegression <- lm(formula=Max_Temperature ~ Total_Insolation + Avg_Temperature +
Min_Temperature + Humidity + Wind_Speed + Altitude + Firespots, data=yearlySummaryRefined)
summary(maxtempRegression)
#Minus Altitude
maxtempRegression <- lm(formula=Max_Temperature ~ Total_Insolation + Avg_Temperature +
Min_Temperature + Humidity + Wind_Speed + Firespots, data=yearlySummaryRefined)
summary(maxtempRegression)


#Trend Charts ------------------------------------------
install.packages("ggpmisc")
library(ggpmisc)

#Humidity Trend Chart
humidityplot <- ggplot(yearlySummaryRefined, aes(x=Year, y=Humidity, group=State, color = State)) +
geom_line() + xlab("Year") + ylab("Average Humidity") + ggtitle("Average Humidity by State from 1999-
2019") + scale_fill_discrete(name = "State") + geom_smooth(method = "lm", se = FALSE)
humidityplot
summary(humidityplot)

#With Regression equation:
HumidityPlot <- ggplot(yearlySummaryRefined, aes(x=Year, y=Humidity, group=State, color = State)) +
geom_line()+ geom_smooth(method = "lm", se = FALSE) + stat_poly_eq(formula = y~x, aes(label =
paste(..eq.label.., ..rr.label.., sep = "~~~")), parse = TRUE)

#Wind Trend Chart
windplot <- ggplot(yearlySummaryRefined, aes(x=Year, y=Wind_Speed, group=State, color = State)) +
geom_line() + xlab("Year") + ylab("Average Wind Speed") + ggtitle("Average Windspeed by State from
1999-2019") + scale_fill_discrete(name = "State") + geom_smooth(method = "lm", se = FALSE)
windplot

#With Regression Equation
#WindSpeed
```

```
WindSpeedPlot <- ggplot(yearlySummaryRefined, aes(x=Year, y=Wind_Speed, group=State, color = State)) + geom_line()+ geom_smooth(method = "lm", se = FALSE) + stat_poly_eq(formula = y~x, aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")), parse = TRUE)

#Firespots Trend Chart
firespotsplot <- ggplot(yearlySummaryRefined, aes(x=Year, y=Firespots, group=State, color = State)) + geom_line() + xlab("Year") + ylab("Average Firespots") + ggtitle("Average Firespots by State from 1999-2019") + scale_fill_discrete(name = "State") + geom_smooth(method = "lm", se = FALSE)
firespotsplot

#With regression equation
firespotsplot <- ggplot(yearlySummaryRefined, aes(x=Year, y=Firespots, group=State, color = State)) + geom_line()+ geom_smooth(method = "lm", se = FALSE) + stat_poly_eq(formula = y~x, aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")), parse = TRUE)

#Deforestation Trend Chart
deforestationplot <- ggplot(yearlySummaryRefined, aes(x=Year, y=Deforestation, group=State, color = State)) + geom_line() + xlab("Year") + ylab("Deforestation") + ggtitle("Deforestation by State from 1999-2019") + scale_fill_discrete(name = "State") + geom_smooth(method = "lm", se = FALSE)
deforestationplot

#With regression equation
deforestationplot <- ggplot(yearlySummaryRefined, aes(x=Year, y=Deforestation, group=State, color = State)) + geom_line()+ geom_smooth(method = "lm", se = FALSE) + stat_poly_eq(formula = y~x, aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")), parse = TRUE)

#Deforestation Cumulative Trend Chart
cumdeforestationplot <- ggplot(yearlySummaryRefined, aes(x=Year, y=cumDeforestation, group=State, color = State)) + geom_line() + xlab("Year") + ylab("Deforestation") + ggtitle("Cumulative Deforestation by State from 1999-2019") + scale_fill_discrete(name = "State")
cumdeforestationplot

#With regression equation
cumdeforestationplot <- ggplot(yearlySummaryRefined, aes(x=Year, y=cumDeforestation, group=State, color = State)) + geom_line()+ geom_smooth(method = "lm", se = FALSE) + stat_poly_eq(formula = y~x, aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")), parse = TRUE)


#Altitude Trend Chart
altitudeplot <- ggplot(yearlySummaryRefined, aes(x=State, y=Altitude, fill = State)) + geom_col() + xlab("Year") + ylab("Altitude") + ggtitle("Average Altitude by State") + scale_fill_discrete(name = "State")
altitudeplot



# Create a dataframe to store max temperature by year and state
dfMaxTemp <- aggregate(x=yearlySummary[,7], by=list(yearlySummary$Year,yearlySummary$State), FUN="mean")
colnames(dfMaxTemp) <- c("Year","State","Max_Temperature")
dfMaxTemp
```

```
# Create a dataframe to store max temperature by state
dfMaxTempState <- aggregate(dfMaxTemp$Max_Temperature, by=list(State=dfMaxTemp$State),
FUN="mean")
colnames(dfMaxTempState) <- c("State","Max_Temperature")
dfMaxTempState

# Create a dataframe to store firespots by year and state
dfFirespots <- aggregate(yearlySummaryRefined$Firespots, by=list(yearlySummaryRefined$Year,
yearlySummaryRefined$State), FUN="mean")
colnames(dfFirespots) <- c("Year","State","Firespots")
dfFirespots

# Create a dataframe to store deforestation by year and state
dfDeforestation <- aggregate(yearlySummaryRefined$Deforestation, by=list(yearlySummaryRefined$Year,
yearlySummaryRefined$State), FUN="mean")
colnames(dfDeforestation) <- c("Year","State","Deforestation")
dfDeforestation

# Create a dataframe to store firespots by year and state for all states
dfFirespotsAll <- aggregate(yearlySummary$Firespots, by=list(yearlySummary$Year,
yearlySummary$State), FUN="mean")
colnames(dfFirespotsAll) <- c("Year","State","Firespots")
dfFirespotsAll

# Create a dataframe to store deforestation by year and state for all states
dfDeforestationAll <- aggregate(yearlySummary$Deforestation, by=list(yearlySummary$Year,
yearlySummary$State), FUN="mean")
colnames(dfDeforestationAll) <- c("Year","State","Deforestation")
dfDeforestationAll

# Create a dataframe to store max temperature by year and state for refined states
dfMaxTempRefined <- aggregate(x=yearlySummaryRefined[,7],
by=list(yearlySummaryRefined$Year,yearlySummaryRefined$State), FUN="mean")
colnames(dfMaxTempRefined) <- c("Year","State","Max_Temperature")
dfMaxTempRefined

# Create a dataframe to store avg temperature by year and state
dfAvgTemp <- aggregate(x=yearlySummary[,8], by=list(yearlySummary$Year,yearlySummary$State),
FUN="mean")
colnames(dfAvgTemp) <- c("Year","State","Avg_Temperature")
dfAvgTemp

# Create a dataframe to store avg temperature by state
dfAvgTempState <- aggregate(dfAvgTemp$Avg_Temperature, by=list(State=dfAvgTemp$State),
FUN="mean")
colnames(dfAvgTempState) <- c("State","Avg_Temperature")
dfAvgTempState

# Create a dataframe to store avg temperature by year and state for refined states
```

```
dfAvgTempRefined <- aggregate(x=yearlySummaryRefined[,8],
by=list(yearlySummaryRefined$Year,yearlySummaryRefined$State), FUN="mean")
colnames(dfAvgTempRefined) <- c("Year","State","Avg_Temperature")
dfAvgTempRefined

# Create a plot of max temp by year and state for all states
g <- ggplot(dfMaxTemp, aes(x=Year, y=Max_Temperature, group=State, color = State)) + geom_line()
g + xlab("Year") + ylab("Max Temperature") + ggtitle("Max Temperature by State from 1999-2019") +
scale_fill_discrete(name = "State") + theme(plot.title = element_text(hjust = 0.5))

# Create a plot of max temp by year and state for refined states
g <- ggplot(dfMaxTempRefined, aes(x=Year, y=Max_Temperature, group=State, color = State)) +
geom_line()
g + geom_smooth(method='lm', formula= y~x, se=FALSE) +  ggtitle("Max Temperature by State from
1999-2019") + theme(plot.title = element_text(hjust = 0.5))

# Create a plot of avg temp by year and state for all states
g <- ggplot(dfAvgTemp, aes(x=Year, y=Avg_Temperature, group=State, color = State)) + geom_line()
g + xlab("Year") + ylab("Avg Temperature") + ggtitle("Avg Temperature by State from 1999-2019") +
scale_fill_discrete(name = "State") + theme(plot.title = element_text(hjust = 0.5))

# Create a plot of avg temp by year and state for refined states
g <- ggplot(dfAvgTempRefined, aes(x=Year, y=Avg_Temperature, group=State, color = State)) +
geom_line()
g + geom_smooth(method='lm', formula= y~x, se=FALSE) +  ggtitle("Avg Temperature by State from
1999-2019") + theme(plot.title = element_text(hjust = 0.5))

# Create a plot of firespots by year and state for refined states
g <- ggplot(dfFirespots, aes(x=Year, y=Firespots, group=State, color = State)) + geom_line()
g + geom_smooth(method='lm', formula= y~x, se=FALSE) +  ggtitle("Average Firespots by State from
1999-2019") + theme(plot.title = element_text(hjust = 0.5))

# Create a plot of deforestation by year and state for refined states
g <- ggplot(dfDeforestation, aes(x=Year, y=Deforestation, group=State, color = State)) + geom_line()
g + geom_smooth(method='lm', formula= y~x, se=FALSE) +  ggtitle("Average Deforestation by State
from 1999-2019") + theme(plot.title = element_text(hjust = 0.5))

# Create a scatter plot of firespots overtime for entire dataset
plot(dfYearlyFirespots$Year, dfYearlyFirespots$AverageFirespots,
    main="Firespots from 1999-2019",
    xlab="Years",
    ylab="Firespots",
    col="blue",
    type="l",
    xaxp = c(1999, 2019, 20),
    cex.axis=0.5)

# Create plot of average firespots by state for all states
g <- ggplot(dfYearlyFirespotsState, aes(x=Year, y=AverageFirespots, group=State, color = State)) +
geom_line()
```

```
g + xlab("Year") + ylab("Average Firespots") + ggtitle("Average Firespots by State from 1999-2019") +
scale_fill_discrete(name = "State")

# Create a dataframe to store firespots by year
dfYearlyFirespots <- aggregate(x=yearlySummary[,15], by=list(yearlySummary$Year), FUN="mean")
colnames(dfYearlyFirespots) <- c("Year","AverageFirespots")
dfYearlyFirespots

# Create a dataframe to store firespots by state
dfStatesFirespots <- aggregate(x=yearlySummary[,15], by=list(yearlySummary$State), FUN="mean")
colnames(dfStatesFirespots) <- c("State","AverageFirespots")
dfStatesFirespots[order(-dfStatesFirespots$AverageFirespots),]

# Create a dataframe to store firespots by year and state
dfYearlyFirespotsState <- aggregate(x=yearlySummary[,15],
by=list(yearlySummary$Year,yearlySummary$State), FUN="mean")
colnames(dfYearlyFirespotsState) <- c("Year","State","AverageFirespots")
dfYearlyFirespotsState

# Create a scatter plot of firespots overtime for entire dataset
plot(dfYearlyFirespots$Year, dfYearlyFirespots$AverageFirespots,
    main="Firespots from 1999-2019",
    xlab="Years",
    ylab="Firespots",
    col="blue",
    type="l",
    xaxp = c(1999, 2019, 20),
    cex.axis=0.5)

# Create a bar plot of firespots by state
barplot(dfStatesFirespots$AverageFirespots,names.arg=dfStatesFirespots$State,
        main="Firespots by State",
        ylab="Firespots",
        col=rainbow(20),
        las=2,
        cex.names=0.5)

# Find highest firespots
maxFirespot = max(dfFirespotsAll$Firespots)
maxFirespot

# Find the index with highest firespots
index = as.numeric(rownames(dfFirespotsAll)[which.max(dfFirespotsAll$Firespots)])
index

# Find the year and state with highest firespots
paste("The highest firespots occurred in", dfFirespotsAll$State[index], "in the year",
dfFirespotsAll$Year[index])

# Order data frame by highest firespots in descending order
```

```
dfFirespotsAll[order(-dfFirespotsAll$Firespots),]

# Create a dataframe to store deforestation by year
dfYearlyDeforestation <- aggregate(x=yearlySummary[,16], by=list(yearlySummary$Year), FUN="mean")
colnames(dfYearlyDeforestation) <- c("Year","AverageDeforestation")
dfYearlyDeforestation$Year <- as.numeric(dfYearlyDeforestation$Year)

# Create a dataframe to store deforestation by state
dfStatesDeforestation <- aggregate(x=yearlySummary[,16], by=list(yearlySummary$State), FUN="mean")
colnames(dfStatesDeforestation) <- c("State","AverageDeforestation")
dfStatesDeforestation[order(-dfStatesDeforestation$AverageDeforestation),]

# Create a dataframe to store deforestation by year and state
dfYearlyDeforestationState <- aggregate(x=yearlySummary[,16],
by=list(yearlySummary$Year,yearlySummary$State), FUN="mean")
colnames(dfYearlyDeforestationState) <- c("Year","State","AverageDeforestation")
dfYearlyDeforestationState

# Create a scatter plot of deforestation overtime for entire dataset
plot(dfYearlyDeforestation$Year, dfYearlyDeforestation$AverageDeforestation,
    main="Deforestation from 1999-2019",
    xlab="Years",
    ylab="Deforestation (km^2)",
    col="blue",
    type="l",
    xaxp = c(1999, 2019, 20),
    cex.axis=0.5)

# Create a bar plot of deforestation by state
barplot(dfStatesDeforestation$AverageDeforestation,names.arg=dfStatesDeforestation$State,
        main="Deforestation by State",
        ylab="Deforestation (km^2)",
        col=rainbow(20),
        las=2,
        cex.names=0.5)


# Find highest deforestation
maxDeforestation = max(dfDeforestationAll$Deforestation)
maxDeforestation

# Find the index with highest deforestation
index2 = as.numeric(rownames(dfDeforestationAll)[which.max(dfDeforestationAll$Deforestation)])
index2

# Find the year and state with highest deforestation
paste("The highest deforestation occurred in", dfDeforestationAll$State[index2], "in the year",
dfDeforestationAll$Year[index2])

# Order data frame by highest deforestation in descending order
```

```
dfDeforestationAll[order(-dfDeforestationAll$Deforestation),]

# Correlation plot - Deforestation vs. Firespots
corrFD <- ggplot(yearlySummaryRefined, aes(x = Firespots, y = Deforestation)) + geom_point() +
stat_smooth(method="lm")
corrFD <- corrFD + ggtitle("Regression Model - Deforestation vs. Firespots") + theme(plot.title =
element_text(size = 9, hjust = 0.5))
corrFD

# ---------------------------- Description -------------------------------------

# The next series of functions and inline code creates two collections of maps.

# The first collection of maps is a series of heat maps which show significant
# temperature change between the years 1999 and 2019, using the mean slope of
# change to indicate the severity of the rise in temperature.

# The second collection of maps shows firespots around the stations that
# reported them. This colleciton focuses on 4 states, AMAZONAS, MATO GROSSO,
# PARA, and TOCANTINS, some of which experienced significant loss of forest
# due to man-made firespots. and some of which did not.

# -------------------------- End Description -----------------------------------

# register the google key so we can use ggmap
register_google(key="AIzaSyAPDi4PuY9YRPMjivKmTXQnaIi7XvwbB3A")

# from https://cran.r-project.org/web/packages/geobr/vignettes/intro_to_geobr.html
# Remove plot axis
no_axis <- theme(axis.title=element_blank(),
                 axis.text=element_blank(),
                 axis.ticks=element_blank())

# uncomment to load states (takes awhile)
#region <- read_state()

# local working directory (where all data files are found)
wd <- "D:\\Karl\\1 Syracuse Masters Program\\2021 - Spring\\IST 687 - ISchool\\Final Project\\DataSets\\"

# Function from IST687 class homework 7 to remove axis formats from the heatmaps
ditch_the_axes <- theme(
  axis.text = element_blank(),
  axis.line = element_blank(),
  axis.ticks = element_blank(),
  panel.border = element_blank(),
  panel.grid = element_blank()
)

# purpose:  gets the names and abbreviations of the Brazillian states we are most interested in
# returns:  a data frame with the Brizillian states we are most interested in
```

```
getStateNames <- function()
{
  # create a dataframe to contain states and abbreviations (to merge below)
  stateNames <- c("AMAZONAS", "MATO GROSSO", "PARA", "TOCANTINS")
  stateAbb <- c("AM", "MT", "PA", "TO")
  states <- data.frame(stateNames, stateAbb)
  colnames(states) <- c("stateName", "state")

  return(states)
}

# purpose: loads all station codes with name, state, lat, lon, and altitude
# returns:  the stations
# lastMod:  3/17/2021
getStations <- function()
{
  df <- setwd(wd)
  df <- read.csv("WeatherStations.csv")
  # rename columns
  colnames(df) <- c("station", "name", "state", "lat", "lon", "altitude")

  return(df)
}

# purpose: loads station codes for the states we are studying
# returns:  the stations
# lastMod:  3/7/2021
mungeStations <- function(showAllStates = FALSE)
{
  df <- getStations()

  if(showAllStates == FALSE)
    {
      states <- getStateNames()

      df1 <- df[df$state==c("AM"),] # amazonas state
      df2 <- df[df$state==c("MT"),] # Mato Grosso state
      df3 <- df[df$state==c("PA"),] # para state
      df4 <- df[df$state==c("TO"),] # Tocantins state

      df <- rbind(df1, df2, df3, df4)

      df <- merge(df, states, by="state")
  }

  return(df)
}

# purpose: loads all firespots from 1999 - 2019
# returns:  the a data frame containing the fires data
```

```
# lastMod:  3/10/2021
getFirespots <- function()
{
  # get firespots
  fires <- setwd(wd)
  fires <- read.csv("inpe_brazilian_amazon_fires_1999_2019.csv")
  colnames(fires)[3] <- c("stateName")
  stateNames <- getStateNames()
  fires <- merge(fires, stateNames, by="stateName")
  colnames(fires) <- c("stateName", "year", "month", "lat", "lon", "firespots", "state")

  return(fires)
}

# purpose: loads all significance results for temps from 1999-2019
# returns:  returns sig data as well as lat, long, and state for stations
# lastMod:  3/17/2021
getTempSignificance <- function()
{
  # get the file
  temps <- setwd(wd)
  temps <- read.xlsx("Significance_results.xlsx")

  # rename columns so they match the stations table
  colnames(temps) <- tolower(colnames(temps))

  # merge with the stations table
  temps <- merge(temps, getStations(), by="station")

  return(temps)
}

# Purpose: gets monthly summary of temperatures
# returns: what you'd expect. ;)
getTemperatures <- function()
{
  df <- setwd(wd)
  df <- read.csv("monthlySummary.csv")

  return(df)
}

# purpose:    draws a topographical map of South America
# returns:    the topographical map
# parameters
#        zoomFactor: how close in we want to be. smaller numbers are farther away
#        centerLongitude: the mid point of longitude for our map
#        centerLatitude: the mid point of latitude for our map
# lastMod:    3/10/2021
```

```
draw.southAmerica <- function(topTitle="", topSubTitle="", color_Or_bW="bw", zoomFactor,
centerLongitude=-55.509545, centerLatitude=-11.860846)
{
  # create the topographical map
  thisMap <- ggmap(get_googlemap(center = c(lon = centerLongitude, lat = centerLatitude), zoom =
zoomFactor, scale = 2, maptype='terrain', color=color_Or_bW))

  # get rid of the axis
  thisMap <- thisMap

  # add a label
  thisMap <- thisMap + labs(title=topTitle, size=16) +
    labs(subtitle=topSubTitle, size=8) +
    xlab("") +
    ylab("") +
    ditch_the_axes

  # show the map
  return(thisMap)
}

# purpose:    adds firespots as dots of varying sizes, depending on the size of the observed fire
# returns:    the topographical map with an added geom_point layer
# parameters
#         saMap:    the map to be altered - a map of south america created with ggmap
#         fires:    the fires that occurred with lat, lon, and "firespots" data, which is the size of the fire
#         mapTitle: the title of the map
#         mapSubTitle: the subtitle of the map
#         dotColor: the color of the dots to be plotted
# lastMod:    3/10/2021
add.FireDotMap <- function(saMap, fires, mapTitle="", mapSubTitle="", dotColor="#B00000")
{
  saMap <- saMap + geom_point(data=fires, aes(x=lon, y=lat), color=dotColor, size=log(fires$firespots))
  saMap <- saMap + ggtitle(mapTitle) +
    xlab("") +
    ylab("") +
    labs(subtitle = mapSubTitle)
    theme(plot.title=element_text(hjust=0.5))

  return(saMap)
}

# Purpose:      Get a blank map of South America (non-topographical)
# parameters
#       region:     the data with lat and long
#       backColor: the background color for the map
# adapted from: https://www.datanovia.com/en/blog/how-to-create-a-map-using-ggplot2/
draw.blank.southAmerica <- function(region=NULL, backColor="#000000", outlineColor = "#FFFFFF")
{
  if(is.null(region) | length(region$code_state) == 0){
```

```
    region <- read_state() # function from the geobr library
  }

  #thisLat <- geocode("Ilha do Bananal, Tocantins", key=MyKey)[2] + 10

  return(
    ggplot() +
      geom_sf(data=region, fill=backColor, color=outlineColor, size=.15, show.legend = FALSE) +
      labs(subtitle="States", size=8) +
      xlab("") +
      ylab("") +
      coord_sf(ylim = c(-30, 3))
  )
}

# purpose:  create data frame with all state abbreviations and the x/y/locations where
#           they should appear on a map
# returns:  a data frame of state abbreviations with x and y coords where they should go on the map
# usage:    tagCoords <- stateTags
#           myMap <- myMap + annotate(geom="text", x=tagCoords$x, y=tagCoords$y, label =
tagCoords$state, size = 4)
stateTags <- function()
{
  xCoords <- c(-64, -61.5, -51.5, -61.5, -53, -45, -42, -39.5, -36.5, -36.5, -38, -36.5, -37.5, -42, -48, -55, -55, -
50, -47.5, -44, -55, -49, -51, -50.5, -53)
  yCoords <- c(-4, 2, 2, -12, -4, -4, -7, -4.5, -5.75, -7.25, -8.5, -9.5, -10.5, -12, -11, -13, -13, -17, -15.7, -19, -
20, -21.5, -25, -27, -29.5)
  stateTags <- c('AM', 'RR', 'AP', 'RO', 'PA', 'MA', 'PI', 'CE', 'RN', 'PB', 'PE', 'AL', 'SE', 'BA', 'TO', 'MT', 'MT', 'GO',
'DE', 'MG', 'MS', 'SP', 'PR', 'SC', 'RS')

  stateCoordinates <- data.frame(stateTags, yCoords, xCoords)
  colnames(stateCoordinates) <- c("state","y","x")

  return(stateCoordinates)
}

# Purpose:      Get a heat map of South America (non-topographical)
# returns:      heat map of brazil where state color is shown by slope of change
#               for states where significant change occurred during the given season
# parameters
#       sa:     the data with multi-polygons of states
#       season: the season in which the changes occurred
# adapted from: https://www.datanovia.com/en/blog/how-to-create-a-map-using-ggplot2/
draw.southAmerica.heatmap <- function(sa, season="Yearly", lowColor="#FFFFFF", highColor="Red")
{
  # ----- vars for inline run ----- #
  #season<-"Winter"
  #lowColor <- "#FFFFFF"
  #highColor <- "Red"
```

```
# ----- end vars for inline run ----- #

# shape files for all states in Brazil
stateShapes <- sa[,c(2, 3, 6)]
colnames( stateShapes)[1] <- "state" # make sure state col is named to match our data

# get stations/seasons with significant temp change
significantStations <- getTempSignificance()
significantStations <- significantStations[,c(1, 2, 3, 8)][significantStations$significant=="Reject",]

# if we selected a specific season, remove the remaining seasons
if(tolower(season) %in% c("winter", "spring", "summer", "fall")){
  significantStations <- significantStations[significantStations$season == season,]
}

# get station codes by state
stations <- getStations()[,c(1,3)]

# merge stations with temp change to add state to stations
significantStations <- merge(significantStations, stations, by="station")
significantStations <- significantStations[order(significantStations$state),] # sort by state

# get mean slope by state - this turns significantStations into a vector
significantStations <- tapply(significantStations$slope, significantStations$state, mean)
# turn it back into a data frame with 2 columns
significantStations <- data.frame(rownames(significantStations), significantStations)
colnames(significantStations) <- c("state", "meanSlope") # reset the col names

# merge the stateShapes with significant stations
mapData <- merge(stateShapes, significantStations, by="state")
mapData <- mapData[,c(1,3,4)]

# create df containing states with no change (to add to the mapData)
statesWithNoChange <- merge(stateShapes, getStations(), by="state")
# eliminate extra columns (even though geom column isn't mentioned, it will be included as "geometry")
statesWithNoChange <- statesWithNoChange[,1]
# RO doesn't have a station, but needs to be on the map
fakeROStation <- getStations()[1,]
fakeROStation[,3] <- "RO"
RO <- merge(stateShapes[stateShapes$state == "RO", c(1, 3)], fakeROStation, by="state")
# have to merge RO with something so that geom column will be renamed "geometry"
# add RO to statesWithNoChange
statesWithNoChange <- rbind(statesWithNoChange, RO[1])
# add zero for all mean slopes in statesWIthNoChange
statesWithNoChange$meanSlope <- rep(0,times=length(statesWithNoChange$state))
# remove unnecessary columns
statesWithNoChange <- statesWithNoChange[,c(1,3,2)]

# remove states with change
for(i in mapData$state)
```

```
  {
    statesWithNoChange <- statesWithNoChange[statesWithNoChange$state != i,]
  }

  lowColor <- "#FFFFFF"
  highColor <- "Red"

  # add states that had no change
  mapData <- rbind(mapData, statesWithNoChange)

  # create the
  returnMap <- ggplot(mapData, aes(fill=meanSlope, color=meanSlope)) +
    labs(title=paste("Brazillian States with Significant Change in", season,"Temperature") , size=16) +
    labs(subtitle="Slope of Change from 1999 to 2019", size=8) +
    xlab("") +
    ylab("")
  returnMap <- returnMap + geom_sf()
  returnMap <- returnMap + coord_sf()
  # add color scale
  returnMap <- returnMap + scale_colour_gradient(
    low = lowColor,
    high = highColor
  )
  # add color gradient
  returnMap <- returnMap + scale_fill_gradient(
    low = lowColor,
    high = highColor
  )

  # add state labels
  tags <- stateTags()
  returnMap <- returnMap + annotate(geom="text", x=tags$x, y=tags$y, label = tags$state, size = 4)

  return(returnMap)
}

#---------------add temp significance to map ----------------------
sa <- read_state() # function from the geobr library

# draw yearly heat map map
draw.southAmerica.heatmap(sa)

# draw heat maps by season
draw.southAmerica.heatmap(sa, "Winter")
draw.southAmerica.heatmap(sa, "Spring")
draw.southAmerica.heatmap(sa, "Summer")
draw.southAmerica.heatmap(sa, "Fall")

# draw the empty map of south america
draw.blank.southAmerica(sa, backColor = "#FFFFFF")
```

```
# example of how to read one state's data if we need in future
# paraState <- read_state(code_state="PA", year="2018")

#--------------- end add temp changes to map -----------------------

#--------------- map the firespots --------------------------------

# store stations in variable and translate headers
stations <- mungeStations()
allStations <- mungeStations(TRUE)

# load all firespots from 1999 to 2019
fireSpots <- getFirespots()

# get firespots in 3-year buckets to output in separate maps below
fireSpots1 <- fireSpots[fireSpots$year  %in% c(1999, 2000, 2001),]
fireSpots2 <- fireSpots[fireSpots$year  %in% c(2002, 2003, 2004),]
fireSpots3 <- fireSpots[fireSpots$year  %in% c(2005, 2006, 2007),]
fireSpots4 <- fireSpots[fireSpots$year  %in% c(2008, 2009, 2010),]
fireSpots5 <- fireSpots[fireSpots$year  %in% c(2011, 2012, 2013),]
fireSpots6 <- fireSpots[fireSpots$year  %in% c(2014, 2015, 2016),]
fireSpots7 <- fireSpots[fireSpots$year  %in% c(2017, 2018, 2019),]

# central spot on map (several options)
#centerOfMap <- geocode("Alta Floresta, Mato Grosso", key=MyKey)
#centerOfMap <- geocode("Ilha do Bananal, Tocantins", key=MyKey)
centerOfMap <- geocode("Sinop, Mato Grosso", key=MyKey)
centerLat <- centerOfMap$lat
centerLon <- centerOfMap$lon

# zoom factor goes from 1 - 15 with 1 being the world and 15 being extremely close
zoomFact = 4

reds <- c("#FFFFFF","#FFEE00", "#FBB806", "#F6830C", "#F24D11", "#ED1717", "#550055")

# ___________________________________________________________________________
# NOTE: the compiler appears to get overwhelmed if you try to render all of your
#       plots at once. To avoid this, step through the lines below.
# ___________________________________________________________________________

# create the topographical map of South America
thisMap <- draw.southAmerica(topTitle="Brazil Firespots (Amazonas, Mato Grosso, Para, Tocantins)",
color="bw", topSubTitle="", zoomFactor=zoomFact, centerLon, centerLat)
thisMap

# add firespots to the map, putting each map to the screen with each addition
thisMap <- add.FireDotMap(thisMap, fireSpots1, mapTitle="Brazil Firespots (Amazonas, Mato Grosso,
Para, Tocantins)", mapSubTitle = "1999 - 2001", reds[1])
thisMap
```

```
thisMap <- add.FireDotMap(thisMap, fireSpots2, mapTitle="Brazil Firespots (Amazonas, Mato Grosso, Para, Tocantins)", mapSubTitle = "1999 - 2004", reds[2])
thisMap
thisMap <- add.FireDotMap(thisMap, fireSpots3, mapTitle="Brazil Firespots (Amazonas, Mato Grosso, Para, Tocantins)", mapSubTitle = "1999 - 2007", reds[3])
thisMap
thisMap <- add.FireDotMap(thisMap, fireSpots4, mapTitle="Brazil Firespots (Amazonas, Mato Grosso, Para, Tocantins)", mapSubTitle = "1999 - 2010", reds[4])
thisMap
thisMap <- add.FireDotMap(thisMap, fireSpots5, mapTitle="Brazil Firespots (Amazonas, Mato Grosso, Para, Tocantins)", mapSubTitle = "1999 - 2013", reds[5])
thisMap
thisMap <- add.FireDotMap(thisMap, fireSpots6, mapTitle="Brazil Firespots (Amazonas, Mato Grosso, Para, Tocantins)", mapSubTitle = "1999 - 2016", reds[6])
thisMap
thisMap <- add.FireDotMap(thisMap, fireSpots7, mapTitle="Brazil Firespots (Amazonas, Mato Grosso, Para, Tocantins)", mapSubTitle = "1999 - 2019", reds[7])
thisMap

#--------------- end map the firespots ---------------------------------

#--------------- begin plotting monthly temperatures for AM and PA -----

# get monthly summary of temperatures
temps <- getTemperatures()

# here we are constructing a Date column from Months and Years
months <- temps$Month
years <- temps$Year
days <- rep.int(1, length(temps$Year))
df1 <- data.frame(months, days, years)
temps$Date <- as.Date(paste(df1$year, '01', df1$month, sep='-'))

# use SQL different method to grab just PA and AM from the temps df
temps2 <- sqldf("SELECT * FROM temps WHERE state=='AM' OR state='PA' ORDER BY date")

# focus on most important columns
temps2 <- temps2[,c(2, 3, 4, 9, 17)]

AM <- temps2[temps2$State=="AM",]
PA <- temps2[temps2$State=="PA",]

# plot AM against PA to see 20-year trend for 2 states
ggplot(temps2, aes(x=Date, y=Avg_Temperature, color=State)) +
  geom_point() +
  geom_smooth(method = lm) +
  labs(title = "Average Monthly Temperature (°C), Year by Year, for 2 Parallel States",
      subtitle = "Amazonas(AM) vs Pará 1999 - 2019")

#--------------- end plotting monthly temperatures for AM and PA -----
```

```r
#--------------------deforestation maps--------------------------------
#Adds API key
register_google(key = "AIzaSyAV2RlEAaIOziz4LNzdco2DsXMHUKhZERk", write = TRUE)

#Create Map ---------------------------------------------------------------
#Colors
l1 <- "#ffee00"
l2 <- "#fbb806"
l3 <- "#f6830c"
l4 <- "#f24d11"
l5 <- "#550055"

#save values of center of map
center <- c(lon = -55.509545, lat = -11.860846)

#Function To remove axis (courtesy of professor Awaysheh)
ditch_the_axes <- theme(
  axis.text = element_blank(),
  axis.line = element_blank(),
  axis.ticks = element_blank(),
  panel.border = element_blank(),
  panel.grid = element_blank(),
  axis.title = element_blank()
)

#create base map
basemap <- get_map(source = "google", maptype ="roadmap", location = center, zoom = 4, color='bw')
basemap <- ggmap(basemap) +
  coord_fixed() +
  ditch_the_axes


# First read in the shapefile, using the path to the shapefile and the shapefile name minus the
# Next the shapefile has to be converted to a dataframe for use in ggplot2
# extension as arguments
shapefile2019 <- readOGR("Shapefiles/2019", "yearly_deforestation")
deforestation2019shp <- fortify(shapefile2019)
rm(shapefile2019)

shapefile2007 <- readOGR("Shapefiles/2007", "PDigital2007_AMZ_pol")
deforestation2007shp <- fortify(shapefile2007)
rm(shapefile2007)

shapefile2006 <- readOGR("Shapefiles/2006", "PDigital2006_AMZ_pol")
deforestation2006shp <- fortify(shapefile2006)
rm(shapefile2006)
```

```
shapefile2005 <- readOGR("Shapefiles/2005", "PDigital2005_AMZ_pol")
deforestation2005shp <- fortify(shapefile2005)
rm(shapefile2005)

shapefile2017to2019 <- readOGR("Shapefiles/2017-2019", "yearly_deforestation")
deforestation2017to2019shp <- fortify(shapefile2017to2019)
rm(shapefile2017to2019)

shapefile2014to2016 <- readOGR("Shapefiles/2014-2016", "yearly_deforestation")
deforestation2014to2016shp <- fortify(shapefile2014to2016)
rm(shapefile2014to2016)

shapefile2011to2013 <- readOGR("Shapefiles/2011-2013", "yearly_deforestation")
deforestation2011to2013shp <- fortify(shapefile2011to2013)
rm(shapefile2011to2013)

shapefile2008to2010 <- readOGR("Shapefiles/2008-2010", "yearly_deforestation")
deforestation2008to2010shp <- fortify(shapefile2008to2010)
rm(shapefile2008to2010)




#Creates geoms by year
geom2017to2019 <- geom_polygon(data = deforestation2017to2019shp, aes(x = long, y = lat, group = group), fill = l5, colour = l5)
geom2014to2016 <- geom_polygon(data = deforestation2014to2016shp, aes(x = long, y = lat, group = group), fill = l4,  colour = l4)
geom2011to2013 <- geom_polygon(data = deforestation2011to2013shp, aes(x = long, y = lat, group = group), fill = l3, colour = l3)
geom2008to2010 <- geom_polygon(data = deforestation2008to2010shp, aes(x = long, y = lat, group = group), fill = l2, colour = l2)

geom2019 <- geom_polygon(data = deforestation2019shp, aes(x = long, y = lat, group = group), fill = l5, colour = l5)
geom2007 <- geom_polygon(data = deforestation2007shp, aes(x = long, y = lat, group = group), fill = l1, colour = l1)
geom2006 <- geom_polygon(data = deforestation2006shp, aes(x = long, y = lat, group = group), fill = l1, colour = l1)
geom2005 <- geom_polygon(data = deforestation2005shp, aes(x = long, y = lat, group = group), fill = l1, colour = l1)



#Create one map for 2017 through 2019
deforestation2017to2019 <- ggmap(basemap) + geom2017to2019
#Create one map for 2014 through 2016
deforestation2014to2016 <- ggmap(basemap) + geom2014to2016
```

```
#Create one map for 2011 through 2013
deforestation2011to2013 <- ggmap(basemap) + geom2011to2013
#Create one map for 2008 through 2010
deforestation2008to2010 <- ggmap(basemap) + geom2008to2010
#Create one map for 2005 through 2007
deforestation2005to2007 <- ggmap(basemap) + geom2005 + geom2006 + geom2007

#Creates map to compare 2005 and 2019
deforestation2005to2007 <- ggmap(basemap) + geom2005 + geom2019


#Creates map to compare all
deforestationl5 <- basemap + geom2017to2019 + geom2014to2016 + geom2011to2013 +
geom2008to2010 + geom2005 + geom2006 + geom2007

#Creates map to compare all
deforestationl4 <- basemap + geom2014to2016 + geom2011to2013 + geom2008to2010 + geom2005 +
geom2006 + geom2007

#Creates map to compare all
deforestationl3 <- basemap + geom2011to2013 + geom2008to2010 + geom2005 + geom2006 +
geom2007

#Creates map to compare all
deforestationl2 <- basemap + geom2008to2010 + geom2005 + geom2006 + geom2007

#Creates map to compare all
deforestationl1 <- basemap + geom2005 + geom2006 + geom2007

#-------------------------------Temperature Significance Testing-------------------------#
#-------------------------------Summarized by station, year, season----------------------#

#create seasons
seasons <- data.frame(cbind(c("Spring","Spring","Spring",
                              "Summer","Summer","Summer",
                              "Fall","Fall","Fall",
                              "Winter","Winter","Winter"),
                            c("October", "November","December",
                              "January", "February", "March",
                              "April","May","June",
                              "July","August","September")))

#combine seasons with weather data
colnames(seasons)<-c("Season","Month")
BrazilWeather$Month <- months(BrazilWeather$Date)

df_weather <- merge(x = BrazilWeather, y = seasons, by = "Month", all.x = TRUE)
df_weather$Year <- year(df_weather$Date)

#summarize weather data
```

```
min_temp <- df_weather[,c(2,12,9,13)]
min_summ <- min_temp %>% group_by(Station,Year,Season) %>% summarise_all(list(
  mean_min = ~ mean(., na.rm = TRUE),
  median_min = ~ median(.,na.rm= TRUE),
  s_dev_min = ~ sd(., na.rm = TRUE),
  sum_min = ~ sum(.,na.rm = TRUE)))

avg_temp <- df_weather[,c(2,12,8,13)]
avg_summ <- avg_temp %>% group_by(Station,Year,Season) %>% summarise_all(list(
  mean_av = ~ mean(., na.rm = TRUE),
  median_av = ~ median(.,na.rm= TRUE),
  s_dev_av = ~ sd(., na.rm = TRUE),
  sum_av = ~ sum(.,na.rm = TRUE)))

max_temp <- df_weather[,c(2,12,7,13)]
max_summ <- max_temp %>% group_by(Station,Year,Season) %>% summarise_all(list(
  mean_max = ~ mean(., na.rm = TRUE),
  median_max = ~ median(.,na.rm= TRUE),
  s_dev_max = ~ sd(., na.rm = TRUE),
  sum_max = ~ sum(.,na.rm = TRUE)))

#re-merge weather data to get all temperature aggregations
temp_by_stationSeason <- merge(merge(min_summ,avg_summ,by =
c("Season","Year","Station")),max_summ,by =  c("Season","Year","Station"))

#check for distributions quickly
temp_sub <- df_weather[df_weather$Station=="82024"& df_weather$Season=="Winter",]
hist(temp_sub$`Avg_Temperature_(Â°C)`)

#preparing the dataset for linear reg
temp_by_stationSeason <- temp_by_stationSeason[temp_by_stationSeason$mean_av!="NaN",]
temp_by_stationSeason$Year <- temp_by_stationSeason$Year-1998
temp_by_stationSeason$n_av <- temp_by_stationSeason$sum_av/temp_by_stationSeason$mean_av
temp_by_stationSeason <- temp_by_stationSeason[temp_by_stationSeason$n_av>74,]

#Begin forloop to obtain trend line & std deviation
#init variables
stn <- c()
seas <- c()
slope <- c()
std_err <- c()
degf <- c()

#looping over st = stations, ss = seasons (inner)
for(st in unique(temp_by_stationSeason$Station)){

  temp_stn <- temp_by_stationSeason[temp_by_stationSeason$Station==st,c(1,2,3,8,9,10)]

  for(ss in c("Spring","Summer","Fall","Winter")){
```

```
    temp_seas <- temp_stn[temp_stn$Season==ss,]

    if(nrow(temp_seas)<2){next}

    stn<-append(stn,st)
    seas <- append(seas,ss)

    #begin model
    mod <- lm(mean_av ~ Year, data = temp_seas)

    #getting model parameters/results
    std_err <- append(std_err,coef(summary(mod))[2,2])
    slope <- append(slope,coef(summary(mod))[2,1])
    degf <- append(degf, nrow(temp_seas)-1)
  }
}

#combine vectors to create new dataframe
df_lin <- data.frame(cbind(stn,seas,as.numeric(slope),as.numeric(std_err),degf))
colnames(df_lin) <- c("Station","Season","slope","std_err","degf")
df_lin$degf <- as.numeric(df_lin$degf)

#more cleaning with degf>19 (need all years present)
df_lin <- df_lin[df_lin$degf>19,]
df_lin$t_stat <- as.numeric(df_lin$slope)/as.numeric(df_lin$std_err)

#get p-values
df_lin$p_val <- pt(as.numeric(df_lin$t_stat),as.numeric(df_lin$degf),lower.tail = FALSE)

#ifelse to get reject or not reject
df_lin$significant <- ifelse(df_lin$p_val < 0.05,"Reject", "Fail to Reject")
```