**Project Summary**

**Team Name:** SparkMagic

**Project Title: Potential Impact of Temperature Change and Sunlight on Death Rates Minnesota**

**Team Members:**

Messac Che Neba

Carl Adams

Linda Reber

Karrin Connors

**Question:**

After considering several options for researching, wanting to combine aspects of finance, environment and healthcare. After discussing several alternatives, the group considered anecdotal observation that a higher number of deaths occur in adults at an average age of 85 in certain months of the year. Contemplating using data to prove or disprove this observation, the group found resources available on the website of the Center for Disease Control. The CDC data focused on the number of deaths grouped by State, County and 10-year age spans. The group decided to review 10 years of historical data.

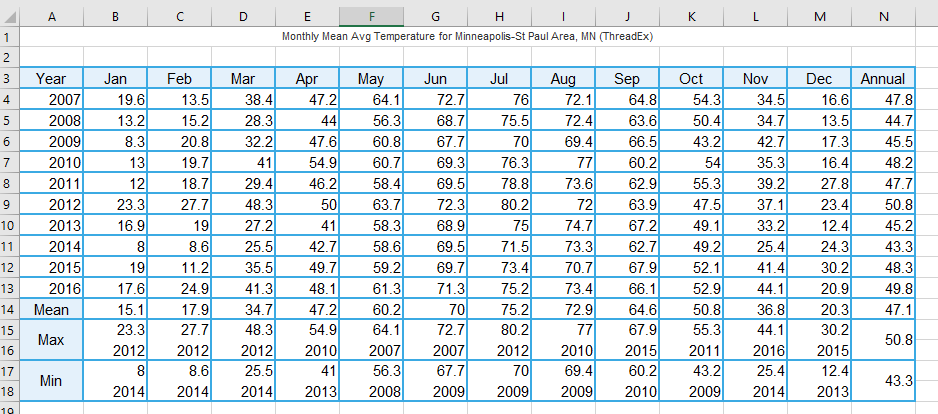
After considering the options of drawing weather data from multiple different sources, the group decided to use data available from the National Weather Service. Given the project timeline, the group determined that a sharp focus on temperature would provide the most likely indicator of “weather”.

The group was able to create a master CSV file of number of deaths in the state of Minnesota, by county, month, year and age group. The group then created a master CSV of temperatures in the state of Minnesota by county, month and year. Each CSV was read into a Pandas Data frame, then the two files were merged within Jupyter Notebook using a Pandas data frame by County, Year and Month. From there, two team members completed the data analysis and visualization using Maplotlib.

Several perspectives were considered including a statewide view, several more heavily populated counties as well as several more sparsely populated counties. Scatterplots were initially developed by age group to visually assess if there appeared to be a correlation between temperature and death. Following the initial work, linear regression analysis was completed to analyze the strength of the correlation between the two data points.

Given the pValue of the data, the team concluded that temperature is a statistically significant factor for death, but the R-squared value would indicate that there are most likely additional significant variables that contribute to mortality.

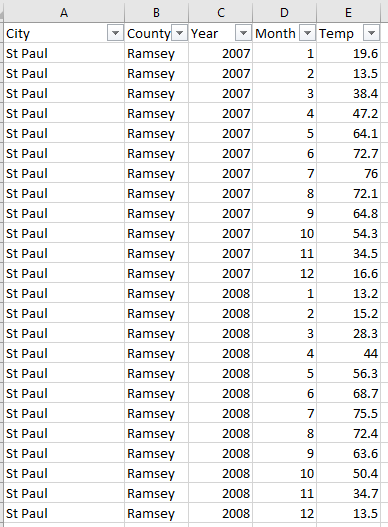
Weather Data – in its original form



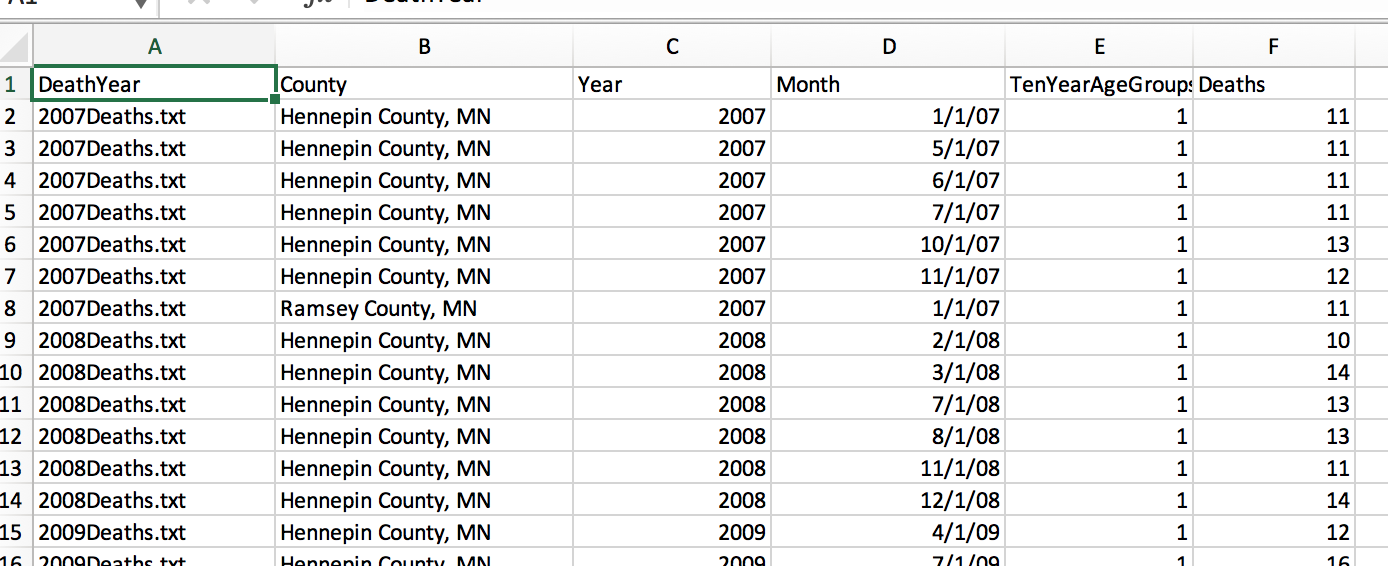
Macro used to transform weather data:

Sub CreateCSV()Dim LastRow  As Long  
Dim City As String  
Dim Month As Integer  
Dim Year As Integer  
Dim RowCounter As IntegerSheets.Add.Name = "CombineCSV"For Each ws In Worksheets' Initialize the variables  
City = ws.Name  
'LastRow = ws.Cells(Rows.Count, 1).End(xlUp).Row 'Determine the Last Row  
Sheets("CombineCSV").Select  
Range("A1") = "City"  
Range("B1") = "Year"  
Range("C1") = "Month"  
Range("D1") = "Temp"  
RowCounter = ActiveSheet.Cells(Rows.Count, 1).End(xlUp).Row + 1 'Determine the Last RowFor Year = 2007 To 2016  
 For Month = 1 To 12  
   Range("A" & RowCounter) = City  
   Range("B" & RowCounter) = Year  
   Range("C" & RowCounter) = Month  
   Range("D" & RowCounter) = ws.Cells(4 + Year - 2007, Month + 1)  
   RowCounter = RowCounter + 1  
  Next Month  
 Next Year  
Next wsEnd Sub

Data Clean-up image

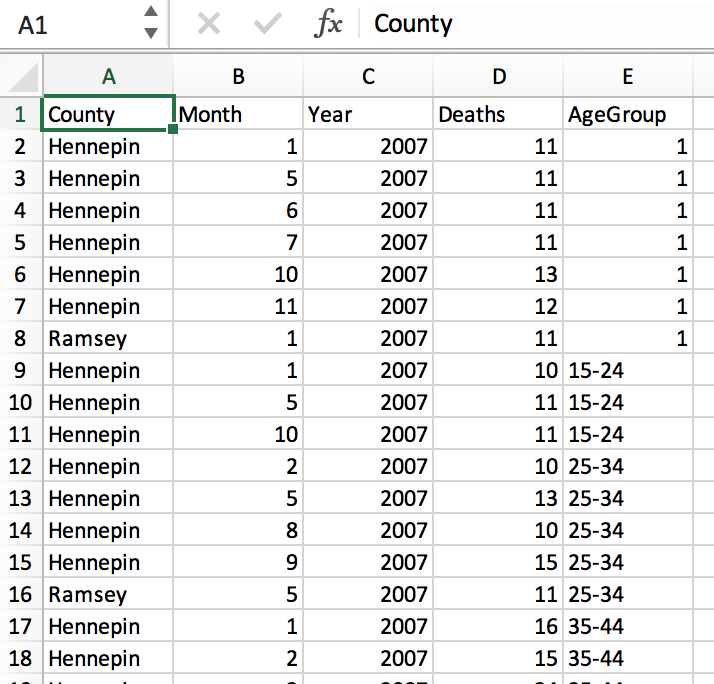


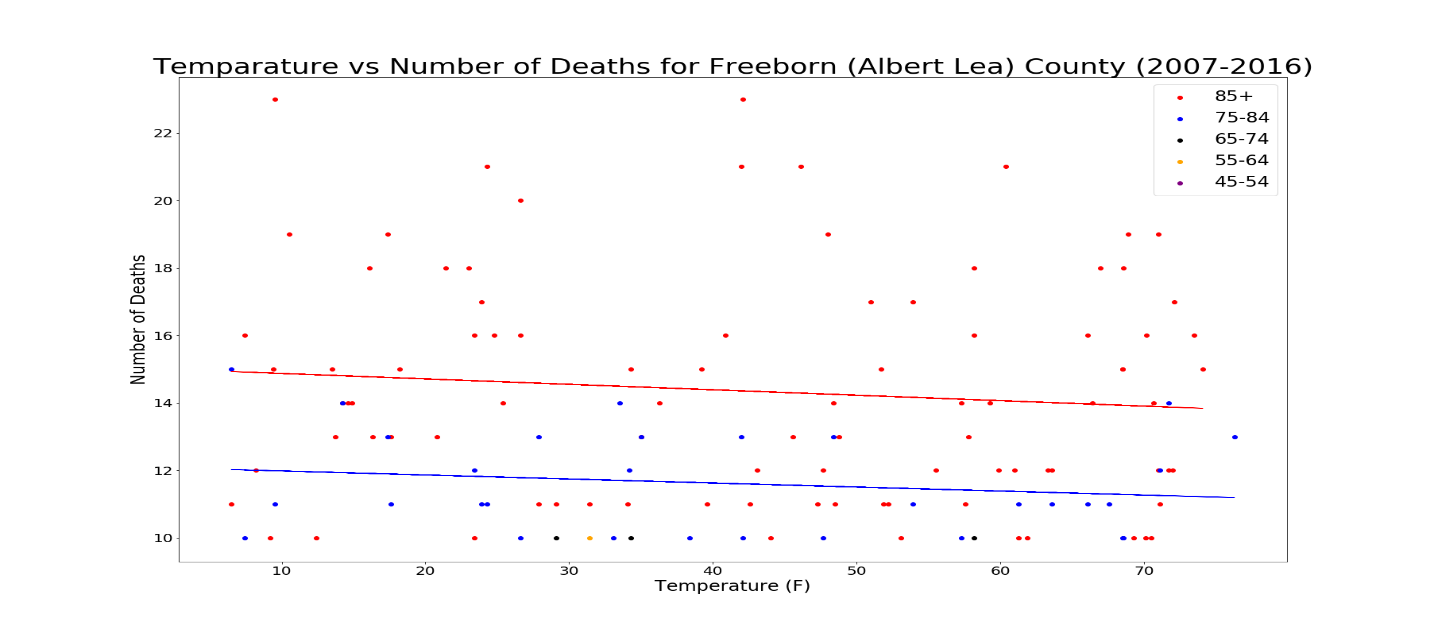
Oirginal Death Data

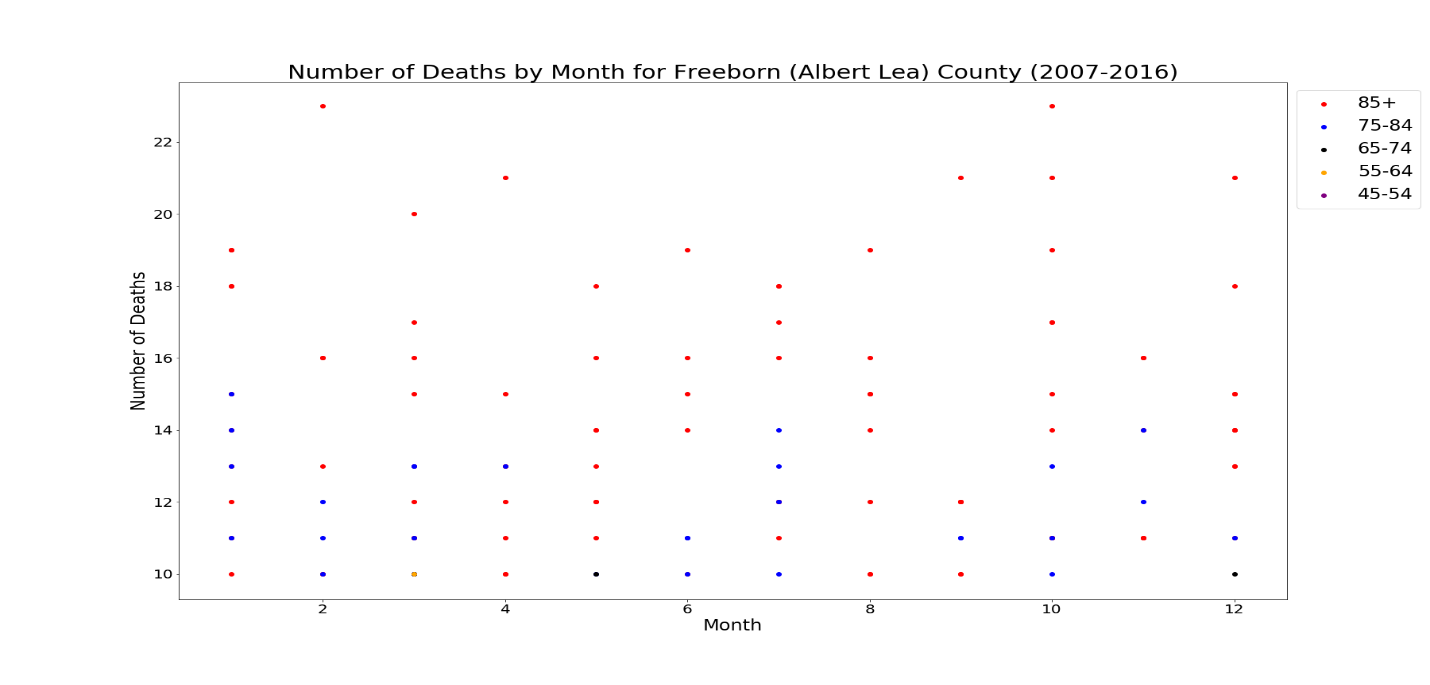


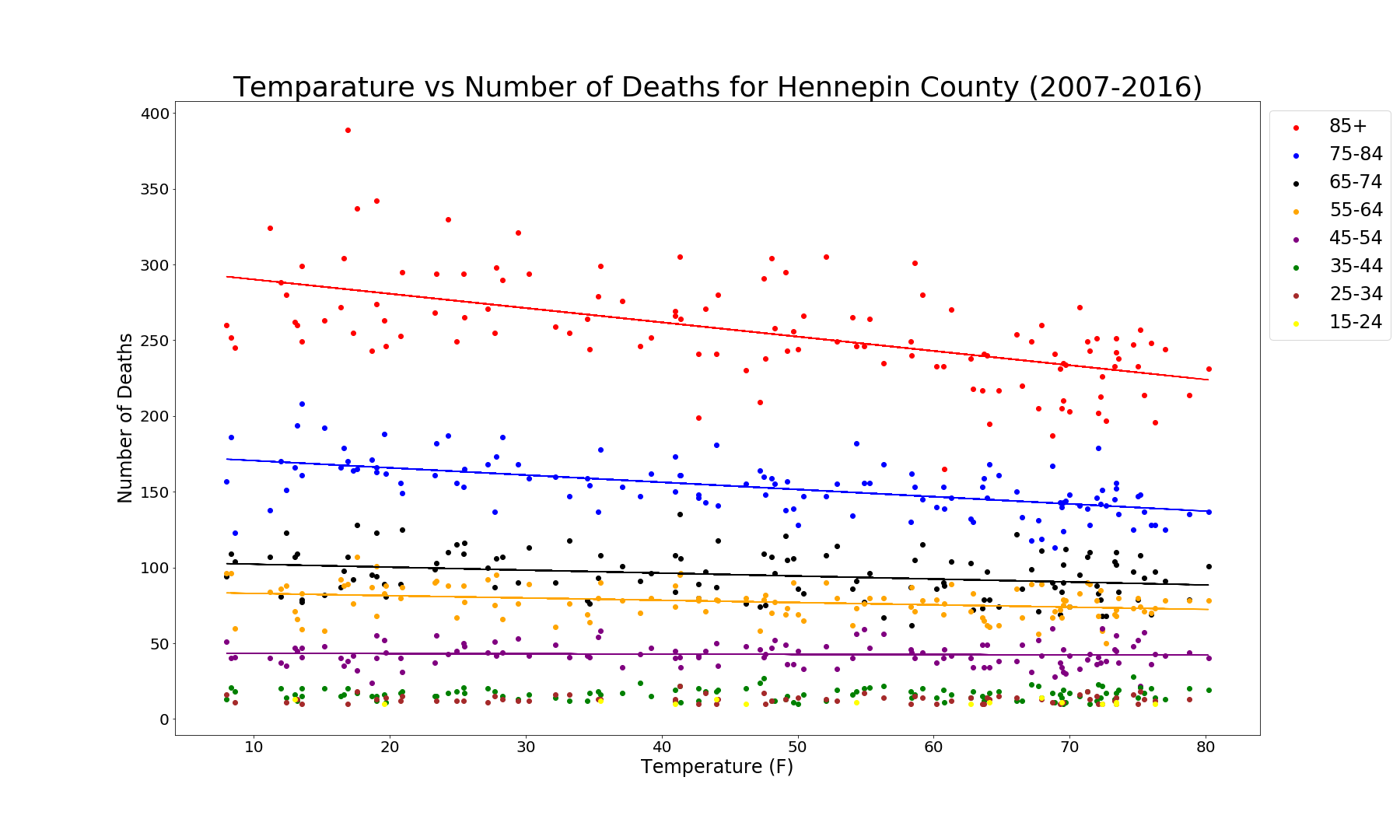
Top of Form

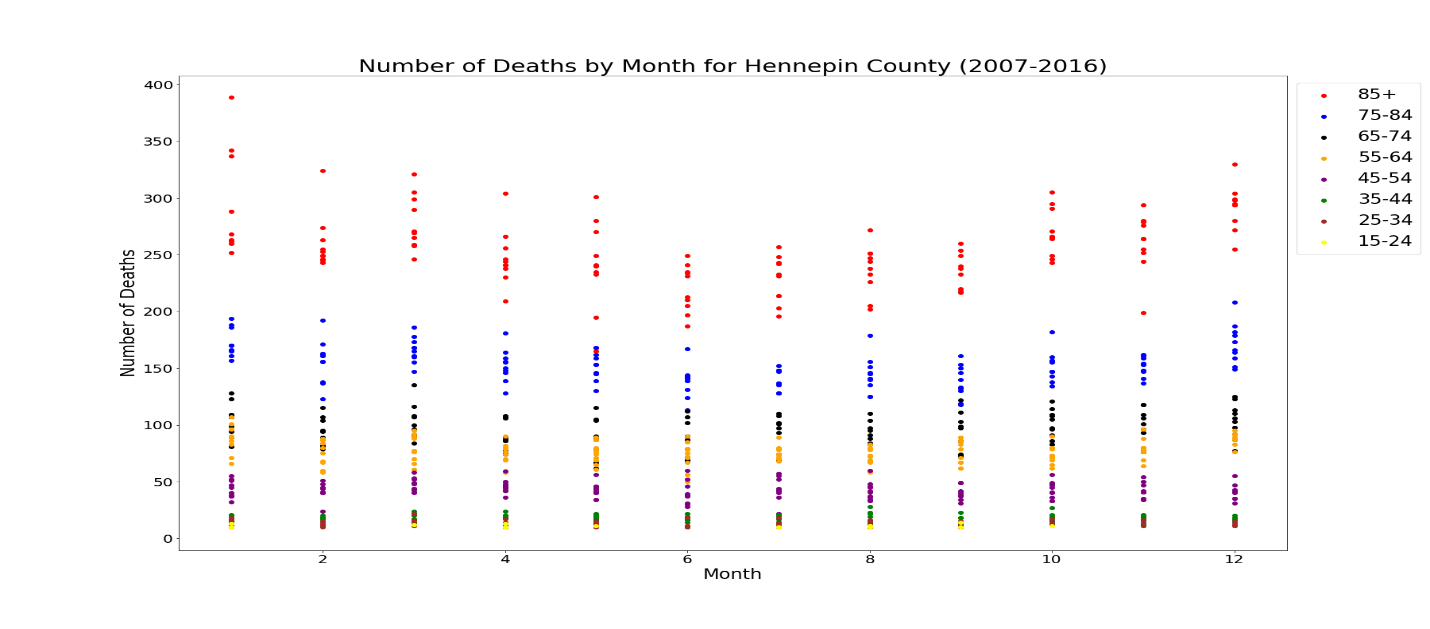
Converted Data:

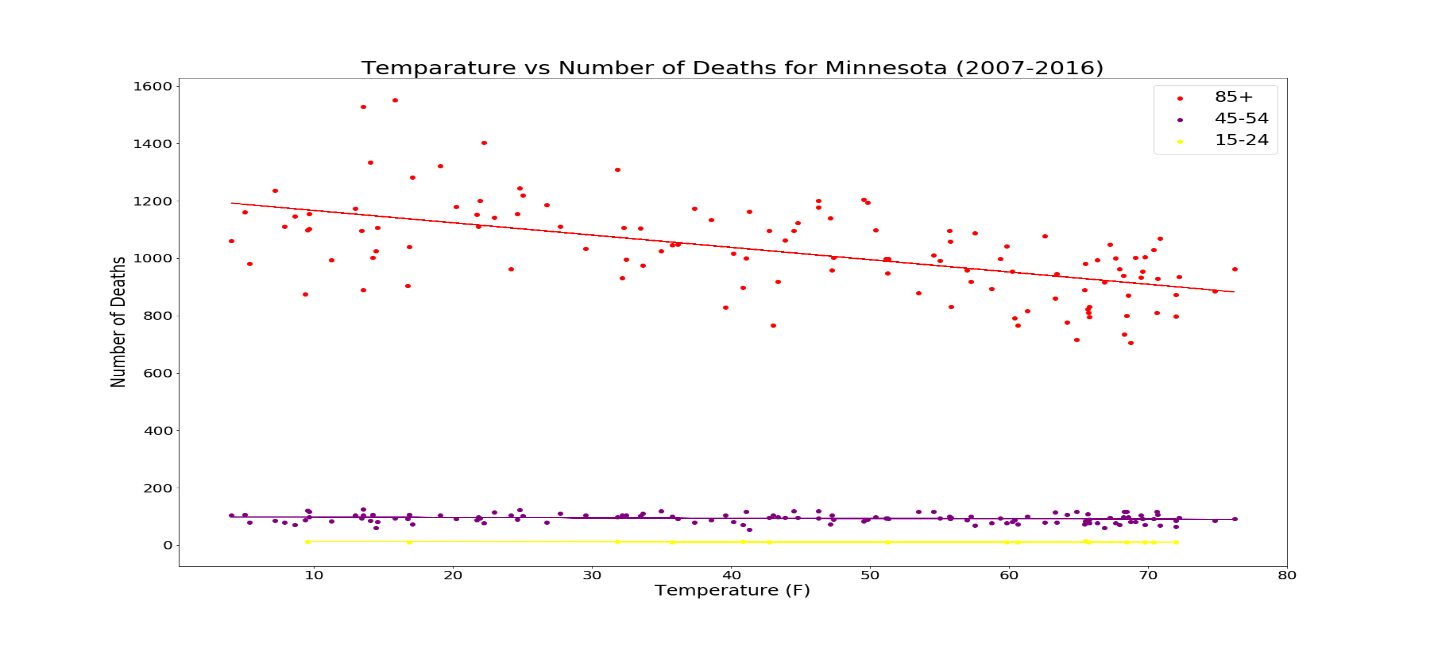


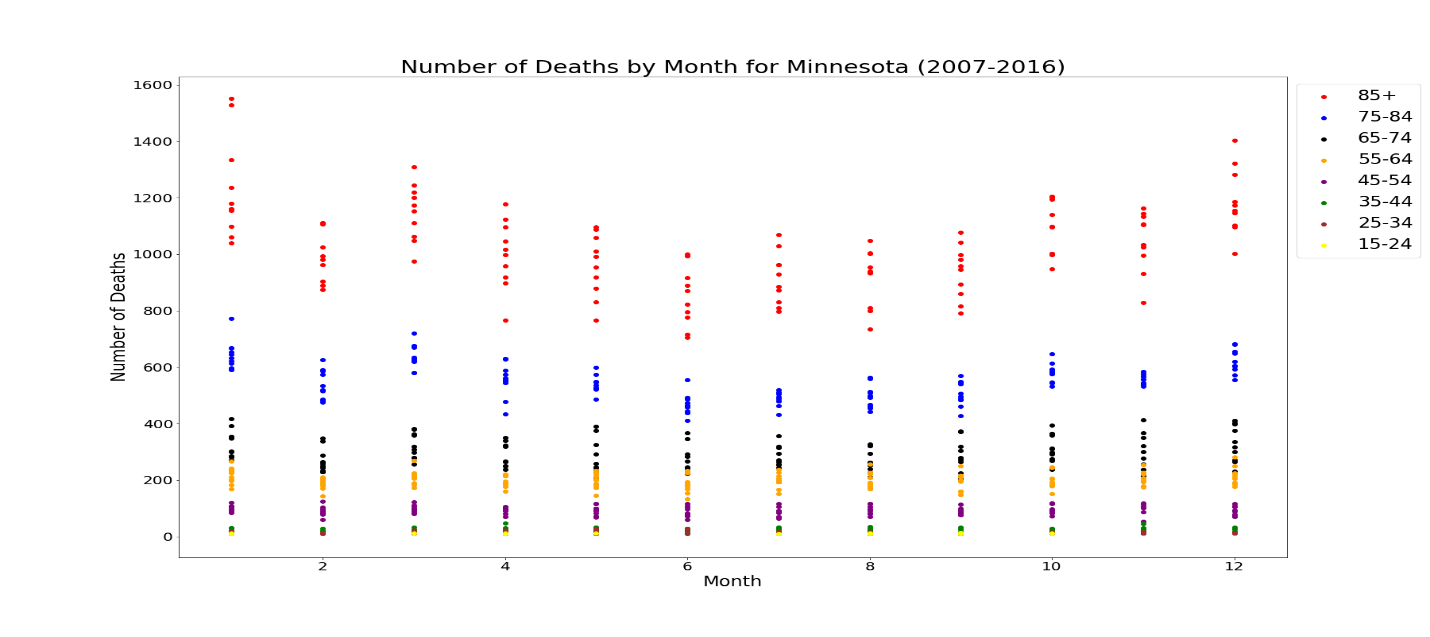


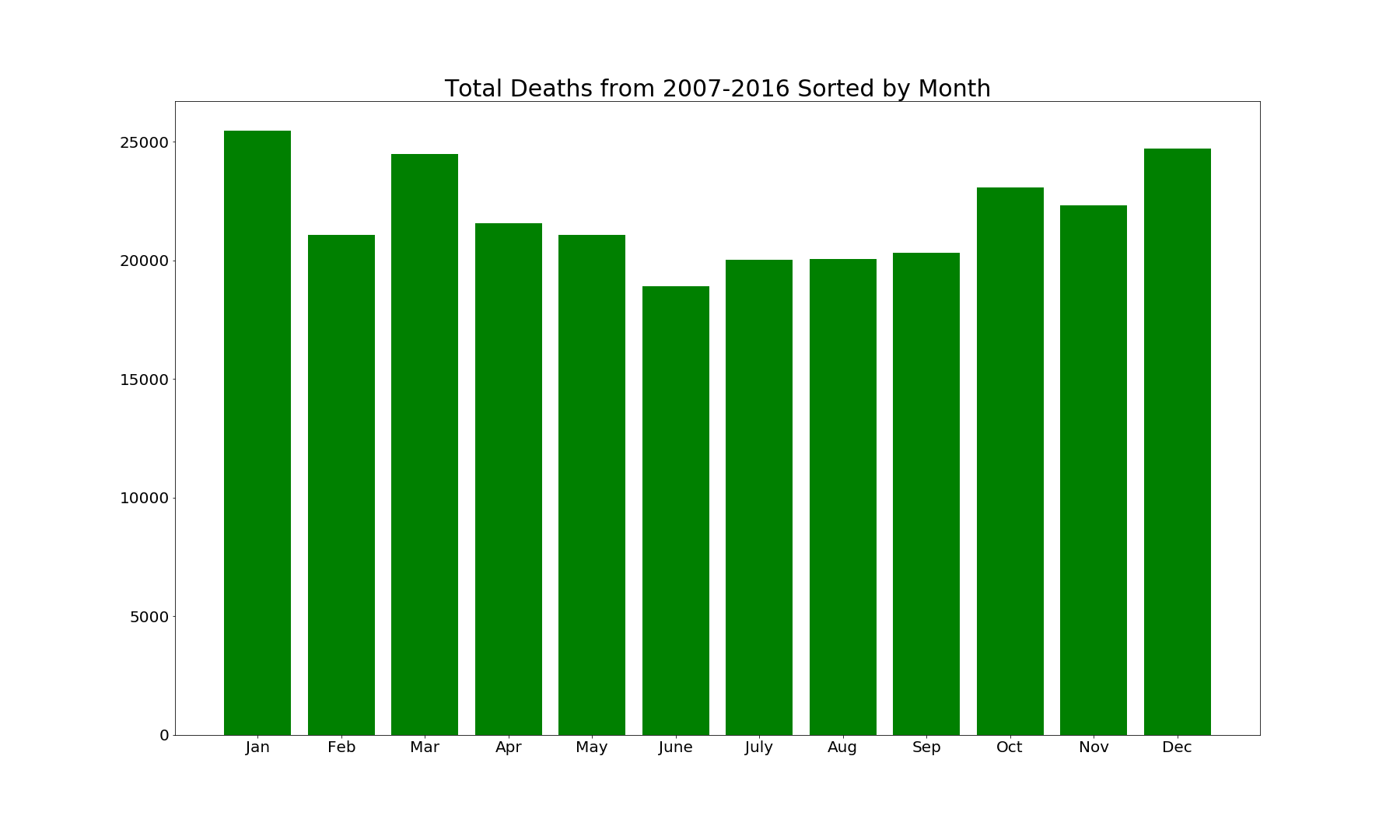












Python File/Jupyter Notebook

# coding: utf-8

# In[56]:

get\_ipython().run\_line\_magic('matplotlib', 'inline')

import numpy as np

import pandas as pd

# The statistical module used to run chi square test

import scipy.stats as stats

from scipy.stats import linregress

import matplotlib.pyplot as plt

import matplotlib

# In[57]:

weather\_df = pd.read\_csv("../Resources/weatherdatacsv.csv")

weather\_df["Temp"]=pd.to\_numeric(weather\_df["Temp"], errors='coerce')

weather\_df = weather\_df.dropna()

# In[3]:

death\_df = pd.read\_csv("../Resources/masterdeathfilenew.csv")

death\_df.head()

# In[4]:

left\_key=["County", "Year", "Month"]

right\_key = ["County", "Year", "Month"]

result\_df=death\_df.merge(weather\_df, left\_on=left\_key, right\_on=right\_key, how='inner')

result\_df.head()

#result\_df.to\_csv("resultsfile.csv")

# In[89]:

#Counties: Hennepin, St Louis (Duluth), Freeborn (Albert Lea), Clay(Moorhead)

group1=result\_df.loc[(result\_df["AgeGroup"]=="85+") & (result\_df["County"]=="Hennepin"),:]

group2 = result\_df.loc[(result\_df["AgeGroup"]=="75-84") & (result\_df["County"]=="Hennepin"),:]

group3 = result\_df.loc[(result\_df["AgeGroup"]=="65-74") & (result\_df["County"]=="Hennepin"),:]

group4 = result\_df.loc[(result\_df["AgeGroup"]=="55-64") & (result\_df["County"]=="Hennepin"),:]

group5 = result\_df.loc[(result\_df["AgeGroup"]=="45-54") & (result\_df["County"]=="Hennepin"),:]

group6 = result\_df.loc[(result\_df["AgeGroup"]=="35-44") & (result\_df["County"]=="Hennepin"),:]

group7 = result\_df.loc[(result\_df["AgeGroup"]=="25-34") & (result\_df["County"]=="Hennepin"),:]

group8 = result\_df.loc[(result\_df["AgeGroup"]=="15-24") & (result\_df["County"]=="Hennepin"),:]

group1.head()

# In[90]:

plt.figure(figsize=(25,15))

plt.scatter(x=group1["Temp"], y=group1["Deaths"], color = "red", label = "85+")

plt.scatter(x=group2["Temp"], y=group2["Deaths"], color = "blue",label = "75-84")

plt.scatter(x=group3["Temp"], y=group3["Deaths"], color = "black",label = "65-74")

plt.scatter(x=group4["Temp"], y=group4["Deaths"], color = "orange",label = "55-64")

plt.scatter(x=group5["Temp"], y=group5["Deaths"], color = "purple",label = "45-54")

plt.scatter(x=group6["Temp"], y=group6["Deaths"], color = "green",label = "35-44")

plt.scatter(x=group7["Temp"], y=group7["Deaths"], color = "brown",label = "25-34")

plt.scatter(x=group8["Temp"], y=group8["Deaths"], color = "yellow",label = "15-24")

(slope1, intercept1, residual1, pvalue1, stderr1) = linregress(group1["Temp"], group1["Deaths"])

fit1 = slope1 \* group1["Temp"] + intercept1

plt.plot(group1["Temp"], fit1, 'red', label="\_nolegend\_")

(slope2, intercept2, residual2, pvalue2, stderr2) = linregress(group2["Temp"], group2["Deaths"])

fit2 = slope2 \* group2["Temp"] + intercept2

plt.plot(group2["Temp"], fit2, 'blue', label="\_nolegend\_")

(slope3, intercept3, residual3, pvalue3, stderr3) = linregress(group3["Temp"], group3["Deaths"])

fit3 = slope3 \* group3["Temp"] + intercept3

plt.plot(group3["Temp"], fit3, 'black', label="\_nolegend\_")

(slope4, intercept4, residual4, pvalue4, stderr4) = linregress(group4["Temp"], group4["Deaths"])

fit4 = slope4 \* group4["Temp"] + intercept4

plt.plot(group4["Temp"], fit4, 'orange', label="\_nolegend\_")

(slope5, intercept5, residual5, pvalue5, stderr5) = linregress(group5["Temp"], group5["Deaths"])

fit5 = slope5 \* group5["Temp"] + intercept5

plt.plot(group5["Temp"], fit5, 'purple', label="\_nolegend\_")

matplotlib.rc('xtick', labelsize=20)

plt.xlabel("Temperature (F)", fontsize=24)

matplotlib.rc('ytick', labelsize=20)

plt.ylabel("Number of Deaths", fontsize=24)

plt.title("Temparature vs Number of Deaths for Hennepin County (2007-2016)", fontsize=36)

plt.legend(bbox\_to\_anchor=(1, 1), fontsize=24)

#plt.savefig("../Images/FreebornByAge.png")

plt.show()

# In[91]:

print(f"P-value for 85+: {pvalue1} R-squared: {residual1\*\*2}")

print(f"P-value for 75-84: {pvalue2} R-squared: {residual2\*\*2}")

print(f"P-value for 65-74: {pvalue3} R-squared: {residual3\*\*2}")

print(f"P-value for 55-64: {pvalue4} R-squared: {residual4\*\*2}")

print(f"P-value for 45-54: {pvalue5} R-squared: {residual5\*\*2}")

# In[84]:

plt.figure(figsize=(25,15))

plt.scatter(x=group1["Month"], y=group1["Deaths"], color = "red", label = "85+")

plt.scatter(x=group2["Month"], y=group2["Deaths"], color = "blue",label = "75-84")

plt.scatter(x=group3["Month"], y=group3["Deaths"], color = "black",label = "65-74")

plt.scatter(x=group4["Month"], y=group4["Deaths"], color = "orange",label = "55-64")

plt.scatter(x=group5["Month"], y=group5["Deaths"], color = "purple",label = "45-54")

#plt.scatter(x=group6["Month"], y=group6["Deaths"], color = "green",label = "35-44")

#plt.scatter(x=group7["Month"], y=group7["Deaths"], color = "brown",label = "25-34")

#plt.scatter(x=group8["Month"], y=group8["Deaths"], color = "yellow",label = "15-24")

matplotlib.rc('xtick', labelsize=20)

plt.xlabel("Month", fontsize=24)

matplotlib.rc('ytick', labelsize=20)

plt.ylabel("Number of Deaths", fontsize=24)

plt.title("Number of Deaths by Month for Hennepin County (2007-2016)", fontsize=30)

plt.legend(bbox\_to\_anchor=(1, 1), fontsize=24)

#plt.savefig("../Images/FreebornByMonth.png")

plt.show()

# In[85]:

mn\_temp= weather\_df.groupby(["Year", "Month"], as\_index=False)["Temp"].mean()

mn\_death= death\_df.groupby(["Year", "Month","AgeGroup"], as\_index=False)["Deaths"].sum()

left\_key=["Year", "Month"]

right\_key = ["Year", "Month"]

mn\_df=mn\_death.merge(mn\_temp, on = ["Year", "Month"], how='inner')

mn\_df.head()

# In[86]:

group10=mn\_df.loc[(mn\_df["AgeGroup"]=="85+"),:]

group20 = mn\_df.loc[(mn\_df["AgeGroup"]=="75-84"),:]

group30 = mn\_df.loc[(mn\_df["AgeGroup"]=="65-74"),:]

group40 = mn\_df.loc[(mn\_df["AgeGroup"]=="55-64"),:]

group50 = mn\_df.loc[(mn\_df["AgeGroup"]=="45-54"),:]

group60 = mn\_df.loc[(mn\_df["AgeGroup"]=="35-44"),:]

group70 = mn\_df.loc[(mn\_df["AgeGroup"]=="25-34"),:]

group80 = mn\_df.loc[(mn\_df["AgeGroup"]=="15-24"),:]

# In[93]:

plt.figure(figsize=(25,15))

plt.scatter(x=group10["Temp"], y=group10["Deaths"], color = "red", label = "85+")

#plt.scatter(x=group20["Temp"], y=group20["Deaths"], color = "blue",label = "75-84")

#plt.scatter(x=group30["Temp"], y=group30["Deaths"], color = "black",label = "65-74")

#plt.scatter(x=group40["Temp"], y=group40["Deaths"], color = "orange",label = "55-64")

plt.scatter(x=group50["Temp"], y=group50["Deaths"], color = "purple",label = "45-54")

#plt.scatter(x=group60["Temp"], y=group60["Deaths"], color = "green",label = "35-44")

#plt.scatter(x=group70["Temp"], y=group70["Deaths"], color = "brown",label = "25-34")

plt.scatter(x=group80["Temp"], y=group80["Deaths"], color = "yellow",label = "15-24")

(slope10, intercept10, residual10, pvalue10, stderr10) = linregress(group10["Temp"], group10["Deaths"])

fit10 = slope10 \* group10["Temp"] + intercept10

plt.plot(group10["Temp"], fit10, 'red', label="\_nolegend\_")

#(slope20, intercept20, residual20, pvalue20, stderr20) = linregress(group20["Temp"], group20["Deaths"])

#fit20 = slope20 \* group20["Temp"] + intercept20

#plt.plot(group20["Temp"], fit20, 'blue', label="\_nolegend\_")

#(slope30, intercept30, residual30, pvalue30, stderr30) = linregress(group30["Temp"], group30["Deaths"])

#fit30 = slope30 \* group30["Temp"] + intercept30

#plt.plot(group30["Temp"], fit30, 'black', label="\_nolegend\_")

#(slope40, intercept40, residual40, pvalue40, stderr40) = linregress(group40["Temp"], group40["Deaths"])

#fit40 = slope40 \* group40["Temp"] + intercept40

#plt.plot(group40["Temp"], fit40, 'orange', label="\_nolegend\_")

(slope50, intercept50, residual50, pvalue50, stderr50) = linregress(group50["Temp"], group50["Deaths"])

fit50 = slope50 \* group50["Temp"] + intercept50

plt.plot(group50["Temp"], fit50, 'purple', label="\_nolegend\_")

(slope80, intercept80, residual80, pvalue80, stderr80) = linregress(group80["Temp"], group80["Deaths"])

fit80 = slope80 \* group80["Temp"] + intercept80

plt.plot(group80["Temp"], fit80, 'yellow', label="\_nolegend\_")

matplotlib.rc('xtick', labelsize=20)

plt.xlabel("Temperature (F)", fontsize=24)

matplotlib.rc('ytick', labelsize=20)

plt.ylabel("Number of Deaths", fontsize=24)

plt.title("Temparature vs Number of Deaths for Minnesota (2007-2016)", fontsize=30)

plt.legend(bbox\_to\_anchor=(1, 1), fontsize=24)

plt.savefig("../Images/MinnesotaByAgeLimited.png")

plt.show()

# In[94]:

print(f"P-value for 85+: {pvalue10} R-squared: {residual10\*\*2}")

#print(f"P-value for 75-84: {pvalue20} R-squared: {residual20\*\*2}")

#print(f"P-value for 65-74: {pvalue30} R-squared: {residual30\*\*2}")

#print(f"P-value for 55-64: {pvalue40} R-squared: {residual40\*\*2}")

print(f"P-value for 45-54: {pvalue50} R-squared: {residual50\*\*2}")

print(f"P-value for 15-24: {pvalue80} R-squared: {residual80\*\*2}")

# In[68]:

plt.figure(figsize=(25,15))

plt.scatter(x=group10["Month"], y=group10["Deaths"], color = "red", label = "85+")

plt.scatter(x=group20["Month"], y=group20["Deaths"], color = "blue",label = "75-84")

plt.scatter(x=group30["Month"], y=group30["Deaths"], color = "black",label = "65-74")

plt.scatter(x=group40["Month"], y=group40["Deaths"], color = "orange",label = "55-64")

plt.scatter(x=group50["Month"], y=group50["Deaths"], color = "purple",label = "45-54")

plt.scatter(x=group60["Month"], y=group60["Deaths"], color = "green",label = "35-44")

plt.scatter(x=group70["Month"], y=group70["Deaths"], color = "brown",label = "25-34")

plt.scatter(x=group80["Month"], y=group80["Deaths"], color = "yellow",label = "15-24")

matplotlib.rc('xtick', labelsize=20)

plt.xlabel("Month", fontsize=24)

matplotlib.rc('ytick', labelsize=20)

plt.ylabel("Number of Deaths", fontsize=24)

plt.title("Number of Deaths by Month for Minnesota (2007-2016)", fontsize=30)

plt.legend(bbox\_to\_anchor=(1, 1), fontsize=24)

plt.savefig("../Images/MinnesotaByMonth.png")

plt.show()

# In[42]:

AgeGroup\_df = mn\_death.set\_index("AgeGroup")

AgeGroup\_df.head()

# In[43]:

# create monthly death list that aggregates 10 years of data for each month

monthly\_death\_list = []

jan\_deaths = AgeGroup\_df.loc[AgeGroup\_df["Month"] == 1,:]

jan\_total\_deaths = jan\_deaths["Deaths"].sum()

monthly\_death\_list.append(jan\_total\_deaths)

feb\_deaths = AgeGroup\_df.loc[AgeGroup\_df["Month"] == 2,:]

feb\_total\_deaths = feb\_deaths["Deaths"].sum()

monthly\_death\_list.append(feb\_total\_deaths)

mar\_deaths = AgeGroup\_df.loc[AgeGroup\_df["Month"] == 3,:]

mar\_total\_deaths = mar\_deaths["Deaths"].sum()

monthly\_death\_list.append(mar\_total\_deaths)

apr\_deaths = AgeGroup\_df.loc[AgeGroup\_df["Month"] == 4,:]

apr\_total\_deaths = apr\_deaths["Deaths"].sum()

monthly\_death\_list.append(apr\_total\_deaths)

may\_deaths = AgeGroup\_df.loc[AgeGroup\_df["Month"] == 5,:]

may\_total\_deaths = may\_deaths["Deaths"].sum()

monthly\_death\_list.append(may\_total\_deaths)

june\_deaths = AgeGroup\_df.loc[AgeGroup\_df["Month"] == 6,:]

june\_total\_deaths = june\_deaths["Deaths"].sum()

monthly\_death\_list.append(june\_total\_deaths)

july\_deaths = AgeGroup\_df.loc[AgeGroup\_df["Month"] == 7,:]

july\_total\_deaths = july\_deaths["Deaths"].sum()

monthly\_death\_list.append(july\_total\_deaths)

aug\_deaths = AgeGroup\_df.loc[AgeGroup\_df["Month"] == 8,:]

aug\_total\_deaths = aug\_deaths["Deaths"].sum()

monthly\_death\_list.append(aug\_total\_deaths)

sep\_deaths = AgeGroup\_df.loc[AgeGroup\_df["Month"] == 9,:]

sep\_total\_deaths = sep\_deaths["Deaths"].sum()

monthly\_death\_list.append(sep\_total\_deaths)

oct\_deaths = AgeGroup\_df.loc[AgeGroup\_df["Month"] == 10,:]

oct\_total\_deaths = oct\_deaths["Deaths"].sum()

monthly\_death\_list.append(oct\_total\_deaths)

nov\_deaths = AgeGroup\_df.loc[AgeGroup\_df["Month"] == 11,:]

nov\_total\_deaths = nov\_deaths["Deaths"].sum()

monthly\_death\_list.append(nov\_total\_deaths)

dec\_deaths = AgeGroup\_df.loc[AgeGroup\_df["Month"] == 12,:]

dec\_total\_deaths = dec\_deaths["Deaths"].sum()

monthly\_death\_list.append(dec\_total\_deaths)

#monthly\_death\_list

# In[44]:

month\_list = ["Jan", "Feb", "Mar", "Apr", "May", "June", "July","Aug", "Sep", "Oct", "Nov","Dec"]

#month\_list

# In[17]:

deadly\_month\_df = pd.DataFrame({"Month": month\_list, "Total Deaths": monthly\_death\_list})

deadly\_month\_df

# In[59]:

plt.figure(figsize=(25,15))

x\_axis = deadly\_month\_df["Month"]

y\_axis = deadly\_month\_df["Total Deaths"]

plt.bar(x\_axis, y\_axis, color="g", align="center")

tick\_locations = [value for value in x\_axis]

plt.xticks(tick\_locations, deadly\_month\_df["Month"], fontsize=20)

plt.title("Total Deaths from 2007-2016 Sorted by Month", fontsize=30)

matplotlib.rc('ytick', labelsize=20)

#plt.xlabel("Month", fontsize=24)

#plt.ylabel("Total Deaths", fontsize=24)

#plt.grid()

plt.savefig("../Images/deadliestmonth.png")

plt.show()