Data processing using UNIX

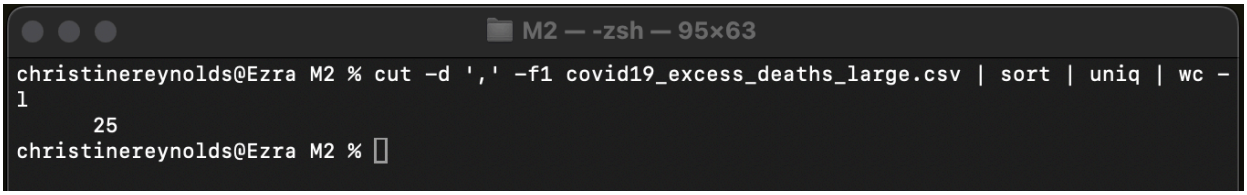
Chris Reynolds | Info 287: FA.24

How many unique countries does the data contain? 24

cut -d ',' -f1 covid19_excess_deaths_large.csv | sort | uniq | wc -l

There are 4 parts to this code delineate via a pipe (|):

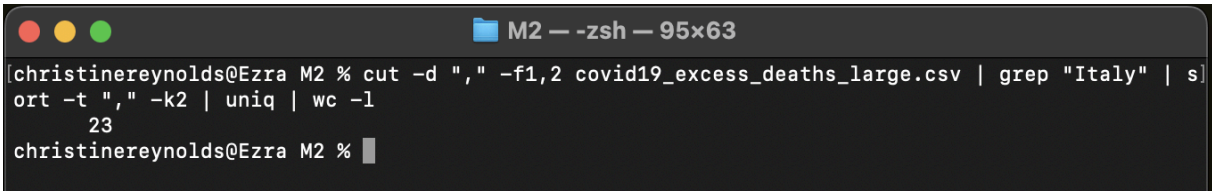
- via the “cut” command, we tell the system to look at only the first column of data in the Covid data set csv, which contains the information regarding which country the data is coming from;
- the system sorts the country data alphabetically;
- the system looks for unique items within the set, because it is sorted alphabetically it only needs the “unqi” command to compare each instance to the instances above and blow the one it is looking at, at a time;
- we are asking for a line count;
- we manually subtract 1 from the number presented as the header of “country” is in this count.



How many unique regions in Italy does the data contain? 22

cut -d "," -f1,2 covid19_excess_deaths_large.csv | grep "Italy" | sort -t "," -k2 | uniq | wc -l

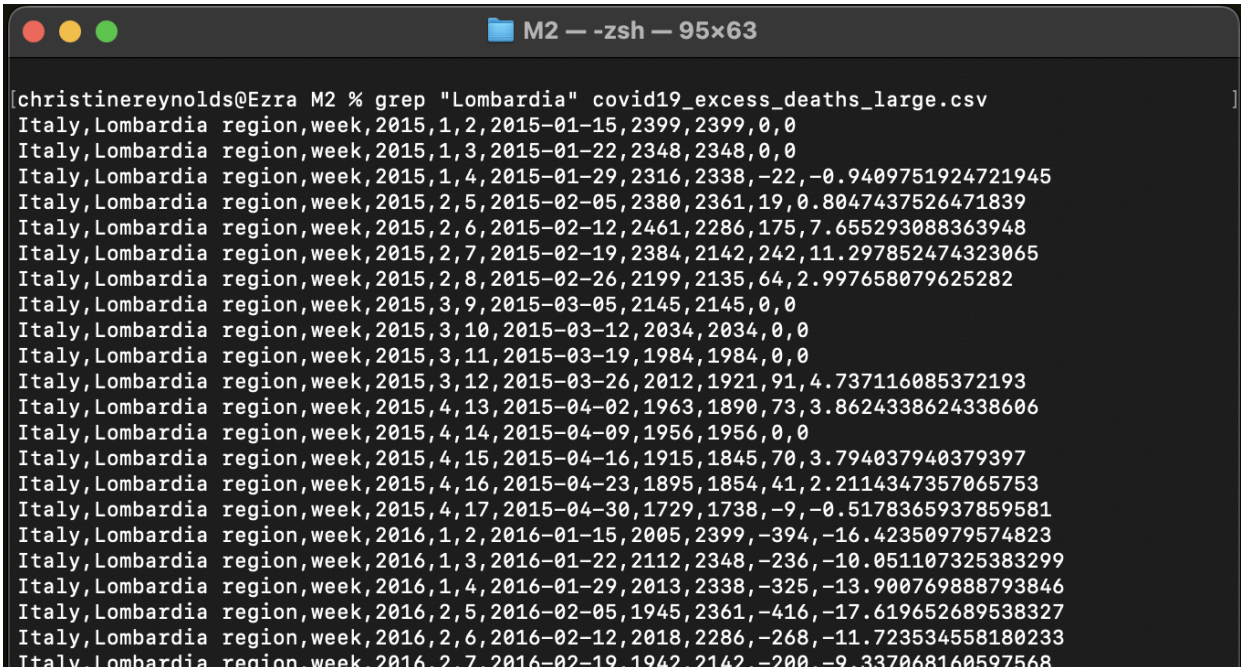
- Here we are telling the system to “cut” and only look at the first 2 columns of data in the file;
- “grep” tells the system to look for instances of the word Italy anywhere within the data (because of the nature of this data set, I know it will only be seeing/counting in the first column so that doesn’t need to be specified);
- this time the system sorts the 2nd column alphabetically;
- looking for unique instances within the regions column;
- again we’re asking for a line count;
- additionally we need to subtract 1 from the number presented as the header “region” isn’t apart of what we want to count.



Find all records which pertain to the "Lombardia" in Italy.

grep "Lombardia" covid19_excess_deaths_large.csv

Written this way, the command retrieves all rows of data that contain Lombardia in any cell and writes them in the terminal.



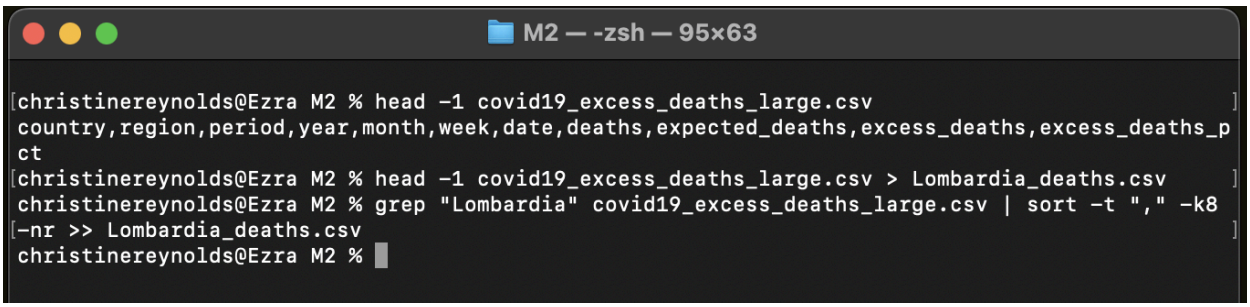
Copy the records obtained (in step 3) to a new file where the records are organized by the number of deaths in descending order.

head -1 covid19_excess_deaths_large.csv > Lombardia_deaths.csv

grep "Lombardia" covid19_excess_deaths_large.csv | sort -t "," -k8 -nr >> Lombardia_deaths.csv

This is done in two separate code chunks.

- The system sets the new file "Lombardia_deaths.csv" with only the first row of information from the Covid file in it, essentially setting up a table header;
- here we're using the same "grep" code to find the instances of "Lombardia";
- this time we're asking the system to sort the 8th column (the number of deaths) in reverse numerical order (from largest to smallest);
- ">>" appends that data to the new csv.



Lombardia_deaths

country	region	period	year	month	week	date	deaths	expected_deaths	excess_deaths	excess_deaths_pct
Italy	Lombardia region	week	2020	3	12	2020-03-25	7830	1921	5909	307.6002082248830
Italy	Lombardia region	week	2020	4	13	2020-04-01	6949	1890	5059	267.67195767195800
Italy	Lombardia region	week	2020	3	11	2020-03-18	6105	1984	4121	207.71169354838700
Italy	Lombardia region	week	2020	4	14	2020-04-08	5269	1956	3313	169.37627811860900
Italy	Lombardia region	week	2020	4	15	2020-04-15	4207	1845	2362	128.02168021680200
Italy	Lombardia region	week	2020	3	10	2020-03-11	3875	2034	1841	90.51130776794490
Italy	Lombardia region	week	2020	4	16	2020-04-22	3310	1854	1456	78.53290183387270
Italy	Lombardia region	week	2017	1	2	2017-01-15	2914	2399	515	21.467278032513500
Italy	Lombardia region	week	2017	1	3	2017-01-22	2722	2348	374	15.928449744463400
Italy	Lombardia region	week	2020	4	17	2020-04-29	2627	1738	889	51.15074798619100
Italy	Lombardia region	week	2017	1	4	2017-01-29	2585	2338	247	10.564585115483300
Italy	Lombardia region	week	2018	1	2	2018-01-15	2553	2399	154	6.419341392246780
Italy	Lombardia region	week	2015	2	6	2015-02-12	2461	2286	175	7.655293088363950
Italy	Lombardia region	week	2020	3	9	2020-03-04	2419	2145	274	12.773892773892800
Italy	Lombardia region	week	2019	1	4	2019-01-29	2416	2338	78	3.3361847733105200
Italy	Lombardia region	week	2019	2	5	2019-02-05	2412	2361	51	2.1601016518424400
Italy	Lombardia region	week	2018	1	3	2018-01-22	2407	2348	59	2.512776831345830
Italy	Lombardia region	week	2015	1	2	2015-01-15	2399	2399	0	0
Italy	Lombardia region	week	2019	2	7	2019-02-19	2389	2142	247	11.53127917833800
Italy	Lombardia region	week	2019	2	6	2019-02-12	2386	2286	100	4.374453193350830
Italy	Lombardia region	week	2015	2	7	2015-02-19	2384	2142	242	11.297852474323100
Italy	Lombardia region	week	2015	2	5	2015-02-05	2380	2361	19	0.8047437526471840
Italy	Lombardia region	week	2017	2	5	2017-02-05	2361	2361	0	0
Italy	Lombardia region	week	2015	1	3	2015-01-22	2348	2348	0	0
Italy	Lombardia region	week	2018	1	4	2018-01-29	2338	2338	0	0
Italy	Lombardia region	week	2019	2	8	2019-02-26	2326	2135	191	8.946135831381720
Italy	Lombardia region	week	2015	1	4	2015-01-29	2316	2338	-22	-0.9409751924721950
Italy	Lombardia region	week	2018	2	5	2018-02-05	2301	2361	-60	-2.541296060991100
Italy	Lombardia region	week	2019	1	2	2019-01-15	2300	2399	-99	-4.1267194664443600
Italy	Lombardia region	week	2017	2	6	2017-02-12	2286	2286	0	0
Italy	Lombardia region	week	2019	3	9	2019-03-05	2269	2145	124	5.780885780885780
Italy	Lombardia region	week	2019	1	3	2019-01-22	2267	2348	-81	-3.4497444633730800
Italy	Lombardia region	week	2015	2	8	2015-02-26	2199	2135	64	2.997658079625280
Italy	Lombardia region	week	2020	1	4	2020-01-29	2165	2338	-173	-7.399486740804110
Italy	Lombardia region	week	2018	3	9	2018-03-05	2161	2145	16	0.745920745920742
Italy	Lombardia region	week	2015	3	9	2015-03-05	2145	2145	0	0
Italy	Lombardia region	week	2018	2	7	2018-02-19	2142	2142	0	0
Italy	Lombardia region	week	2018	2	8	2018-02-26	2135	2135	0	0
Italy	Lombardia region	week	2020	1	3	2020-01-22	2129	2348	-219	-9.327086882453150
Italy	Lombardia region	week	2018	3	10	2018-03-12	2118	2034	84	4.129793510324480
Italy	Lombardia region	week	2016	1	3	2016-01-22	2112	2348	-236	-10.051107325383300
Italy	Lombardia region	week	2020	2	6	2020-02-12	2108	2286	-178	-7.786526684164470
Italy	Lombardia region	week	2020	1	2	2020-01-15	2108	2399	-291	-12.130054189245500
Italy	Lombardia region	week	2020	2	8	2020-02-26	2105	2135	-30	-1.405152248243600
Italy	Lombardia region	week	2020	2	5	2020-02-05	2099	2361	-262	-11.096992799661200
Italy	Lombardia region	week	2017	2	7	2017-02-19	2079	2142	-63	-2.941176470588230
Italy	Lombardia region	week	2020	2	7	2020-02-19	2076	2142	-66	-3.081232492997200
Italy	Lombardia region	week	2019	3	11	2019-03-19	2057	1984	73	3.679435483870980
Italy	Lombardia region	week	2017	3	9	2017-03-05	2056	2145	-89	-4.149184149184150
Italy	Lombardia region	week	2019	3	10	2019-03-12	2051	2034	17	0.8357915437561450
Italy	Lombardia region	week	2018	2	6	2018-02-12	2051	2286	-235	-10.27996500437450
Italy	Lombardia region	week	2015	3	10	2015-03-12	2034	2034	0	0
Italy	Lombardia region	week	2016	2	6	2016-02-12	2018	2286	-268	-11.723534558180200
Italy	Lombardia region	week	2016	1	4	2016-01-29	2013	2338	-325	-13.900769888793800
Italy	Lombardia region	week	2015	3	12	2015-03-26	2012	1921	91	4.737116085372190
Italy	Lombardia region	week	2016	1	2	2016-01-15	2005	2399	-394	-16.42350979574820
Italy	Lombardia region	week	2018	3	11	2018-03-19	2004	1984	20	1.008064516129030
Italy	Lombardia region	week	2019	4	14	2019-04-09	1998	1956	42	2.147239263803700
Italy	Lombardia region	week	2017	2	8	2017-02-26	1994	2135	-141	-6.604215456674480
Italy	Lombardia region	week	2016	2	8	2016-02-26	1989	2135	-146	-6.838407494145200
Italy	Lombardia region	week	2015	3	11	2015-03-19	1984	1984	0	0
Italy	Lombardia region	week	2018	4	14	2018-04-09	1969	1956	13	0.6646216768916130
Italy	Lombardia region	week	2015	4	13	2015-04-02	1963	1890	73	3.8624338624338600
Italy	Lombardia region	week	2015	4	14	2015-04-09	1956	1956	0	0
Italy	Lombardia region	week	2019	3	12	2019-03-26	1953	1921	32	1.6657990629880300
Italy	Lombardia region	week	2016	3	10	2016-03-11	1950	2034	-84	-4.12979351032449
Italy	Lombardia region	week	2017	3	11	2017-03-19	1945	1984	-39	-1.9657258064516200
Italy	Lombardia region	week	2016	2	5	2016-02-05	1945	2361	-416	-17.619652689538300
Italy	Lombardia region	week	2016	2	7	2016-02-19	1942	2142	-200	-9.337068160597570
Italy	Lombardia region	week	2018	4	16	2018-04-23	1930	1854	76	4.09924487594391
Italy	Lombardia region	week	2018	3	12	2018-03-26	1921	1921	0	0
Italy	Lombardia region	week	2016	3	12	2016-03-25	1921	1921	0	0
Italy	Lombardia region	week	2018	4	13	2018-04-02	1917	1890	27	1.4285714285714200
Italy	Lombardia region	week	2016	3	9	2016-03-04	1916	2145	-229	-10.675990675990700
Italy	Lombardia region	week	2015	4	15	2015-04-16	1915	1845	70	3.794037940379400
Italy	Lombardia region	week	2015	4	16	2015-04-23	1895	1854	41	2.2114347357065800
Italy	Lombardia region	week	2019	4	13	2019-04-02	1890	1890	0	0
Italy	Lombardia region	week	2016	4	13	2016-04-01	1881	1890	-9	-0.4761904761904820
Italy	Lombardia region	week	2016	3	11	2016-03-18	1881	1984	-103	-5.1915322580645100
Italy	Lombardia region	week	2017	3	10	2017-03-12	1877	2034	-157	-7.718780727630280
Italy	Lombardia region	week	2016	4	14	2016-04-08	1874	1956	-82	-4.192229038854810
Italy	Lombardia region	week	2017	3	12	2017-03-26	1862	1921	-59	-3.0713170223841800
Italy	Lombardia region	week	2019	4	16	2019-04-23	1854	1854	0	0
Italy	Lombardia region	week	2018	4	15	2018-04-16	1847	1845	2	0.1084010840108500
Italy	Lombardia region	week	2017	4	15	2017-04-16	1845	1845	0	0
Italy	Lombardia region	week	2017	4	13	2017-04-02	1836	1890	-54	-2.857142857142860
Italy	Lombardia region	week	2017	4	17	2017-04-30	1824	1738	86	4.948216340621410
Italy	Lombardia region	week	2016	4	16	2016-04-22	1792	1854	-62	-3.3441208198489700
Italy	Lombardia region	week	2016	4	15	2016-04-15	1792	1845	-53	-2.8726287262872600
Italy	Lombardia region	week	2017	4	14	2017-04-09	1780	1956	-176	-8.997955010224940
Italy	Lombardia region	week	2017	4	16	2017-04-23	1776	1854	-78	-4.207119741100330
Italy	Lombardia region	week	2019	4	15	2019-04-16	1775	1845	-70	-3.794037940379400
Italy	Lombardia region	week	2019	4	17	2019-04-30	1762	1738	24	1.3808975834292300
Italy	Lombardia region	week	2018	4	17	2018-04-30	1738	1738	0	0
Italy	Lombardia region	week	2015	4	17	2015-04-30	1729	1738	-9	-0.5178365937859580
Italy	Lombardia region	week	2016	4	17	2016-04-29	1728	1738	-10	-0.5753739930955110

Analyzing COVID-19 Data using R

Chris Reynolds | Info 287: FA.24

Load the CSV file into an R dataframe; view the dataframe and display the category of each variable (column)

```r
#set directory
setwd("/Users/christinereynolds/Desktop/Problem Solving/M5")
#read data from CSV
coviddata <- read.csv("covid19_excess_deaths_large_1.csv", header=TRUE, sep="," )
#show table in RStudio
View(coviddata)
#write column titles and data types in console
str (coviddata)
#clean the data
coviddata2 <- na.omit(coviddata)
```

Explain:

- Start by setting up the directory & pointing the code to the file/data we're starting with
- Asking the system to show us all of the column titles and types of data represented in the columns to better understand the data we're working with
- Asking the system to clean the data by telling it to omit any rows with blank/missing data

Visual:

```
> #write column titles and data types in console
> str (coviddata)
'data.frame':   53947 obs. of  11 variables:
 $ country          : chr  "Italy" "Italy" "Italy" "Italy" ...
 $ region           : chr  "Bergamo city" "Bergamo city" "Bergamo city" "Bergamo city"
...
 $ period           : chr  "week" "week" "week" "week" ...
 $ year             : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
 $ month            : int  1 1 1 2 2 2 2 3 3 3 ...
 $ week             : int  2 3 4 5 6 7 8 9 10 11 ...
 $ date             : chr  "2015-01-15" "2015-01-22" "2015-01-29" "2015-02-05" ...
 $ deaths           : num  39 31 41 31 38 35 24 46 30 25 ...
 $ expected_deaths  : num  36 31 39 31 35 32 24 33 30 27 ...
 $ excess_deaths    : num  3 0 2 0 3 3 0 13 0 -2 ...
 $ excess_deaths_pct: num  8.33 0 5.13 0 8.57 ...
```

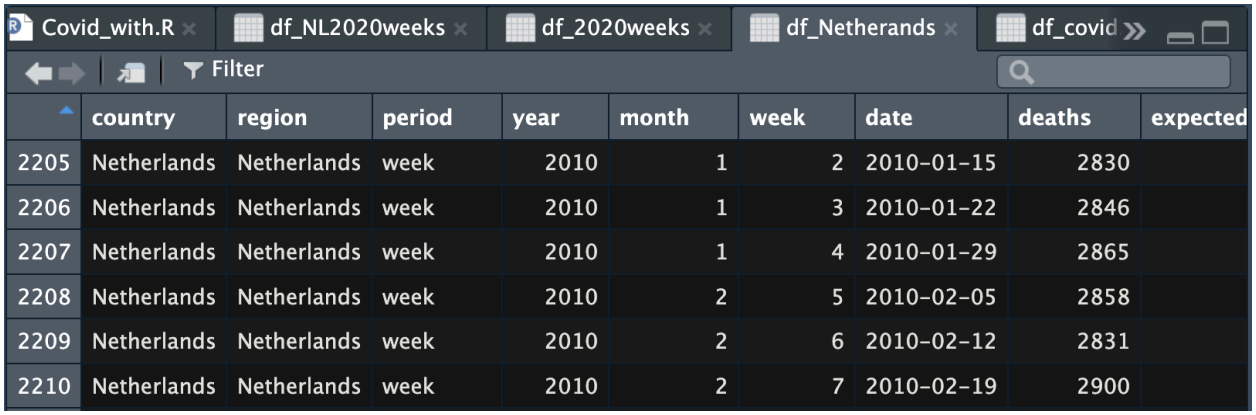
Pick one country and use R to filter the data using that country

```r
#filtering for Netherlands (data frame) Opt 1
df_Netherands <- coviddata2[coviddata2$country == 'Netherlands', ]
#filtering for Netherlands (view w/tidyverse) Opt 2
library(tidyverse)
coviddata2 %>%
  filter(country == 'Netherlands')
```

Explain:
I have two options here as I was having trouble with what was shown in lecture, so the second option of using tidyverse was actually the one I was first successful with. Opt 1 basically reads:

- create data frame: df_Netherands f
- rom the coviddata2 (cleaned data set)
- where the column of country is Netherlands
- include all other columns [,]

Visual:



Covid_with.R | df_NL2020weeks | df_2020weeks | df_Netherands | df_covid

Filter

	country	region	period	year	month	week	date	deaths	expected
2205	Netherlands	Netherlands	week	2010	1	2	2010-01-15	2830	
2206	Netherlands	Netherlands	week	2010	1	3	2010-01-22	2846	
2207	Netherlands	Netherlands	week	2010	1	4	2010-01-29	2865	
2208	Netherlands	Netherlands	week	2010	2	5	2010-02-05	2858	
2209	Netherlands	Netherlands	week	2010	2	6	2010-02-12	2831	
2210	Netherlands	Netherlands	week	2010	2	7	2010-02-19	2900	

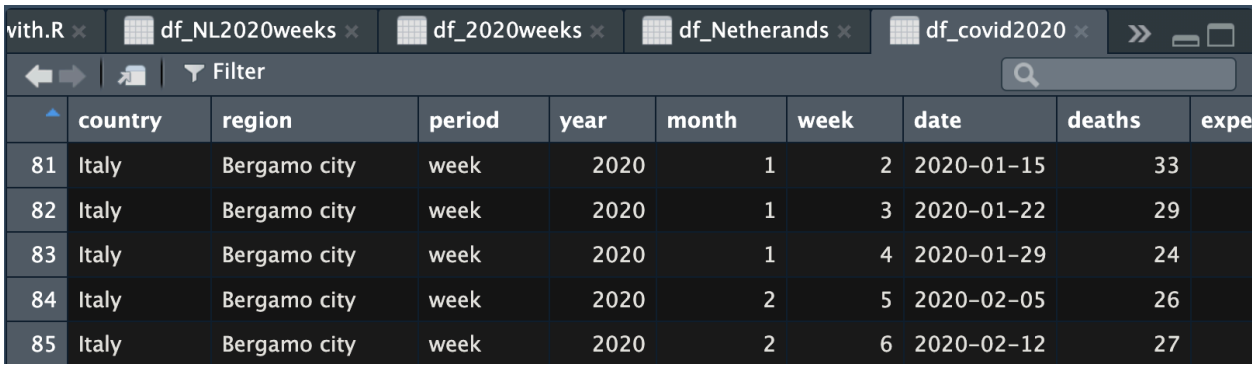
Filter the dataframe for the year 2020

```
19  #filtering for the year 2020 (data frame) Opt 1
20  df_covid2020 <- coviddata2[coviddata2$year == 2020, ]
21  #filtering for the year 2020 Opt 2
22  coviddata2 %>%
23    filter(year == 2020)
```

Explain:
Again I first figured out the tidyverse before understanding the first option shown in class. Opt one reads:

- create data frame df_covid2020
- from the cleaned data set
- where the column year is 2020
- include all other columns

Visual:



vith.R | df_NL2020weeks | df_2020weeks | df_Netherands | df_covid2020

Filter

	country	region	period	year	month	week	date	deaths	expe
81	Italy	Bergamo city	week	2020	1	2	2020-01-15	33	
82	Italy	Bergamo city	week	2020	1	3	2020-01-22	29	
83	Italy	Bergamo city	week	2020	1	4	2020-01-29	24	
84	Italy	Bergamo city	week	2020	2	5	2020-02-05	26	
85	Italy	Bergamo city	week	2020	2	6	2020-02-12	27	

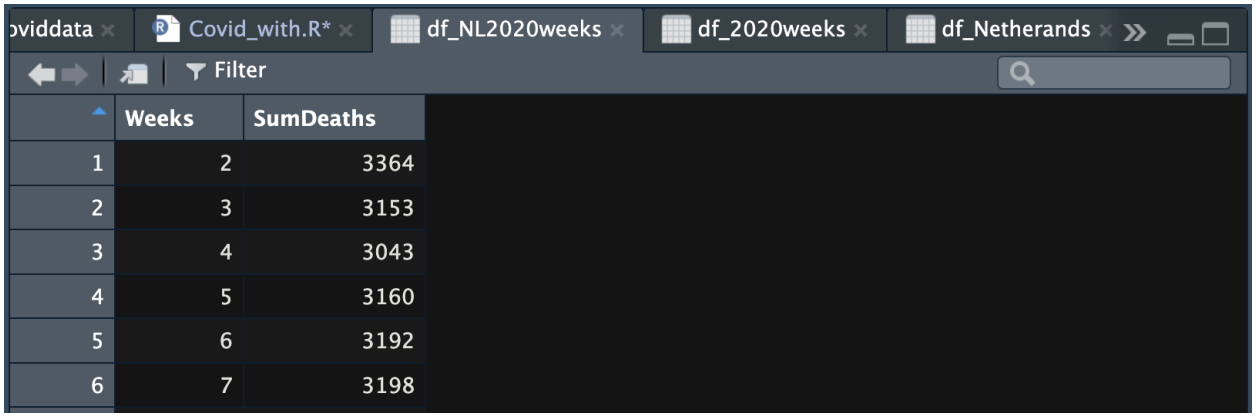
For the new dataframe (generated in the last question), sum the number of deaths for each week

```
#sum weekly deaths for df_covid2020
df_2020weeks <- aggregate(list(deaths=df_covid2020$deaths), by=list
                          (Weeks=df_covid2020$week), FUN=sum)
View(df_2020weeks)

#sum weekly deaths for Netherlands in 2020
df_NL2020 <- df_Netherands[df_Netherands$year == 2020, ]
df_NL2020weeks <- aggregate(list(SumDeaths=df_NL2020$deaths),
                            by=list(Weeks=df_NL2020$week), FUN=sum)

#Scatterplot Sum Deaths by weeks of 2020
library(ggplot2)
ggplot(df_NL2020weeks, aes(x=Category,y=x)) +
  geom_point() + ylim(0,5100)
```

Explain:
I realized here that these were supposed to be stacking on each other at this point, but figured I'd get a little extra practice in so I did both the larger set from 2020 and then do the data frames for the Netherlands. That second chunk of code reads:

- Create a data frame from the previous Netherlands dataset and filter it by year
- Then create another new set where the deaths are added together for the whole country by week

Visual:



	Weeks	SumDeaths
1	2	3364
2	3	3153
3	4	3043
4	5	3160
5	6	3192
6	7	3198

For the country you have picked, in the year 2020, is there a relationship between the week number and number of deaths? Report correlation values for these and comment on them

Answer: According to linear regression, no. -0.01161864

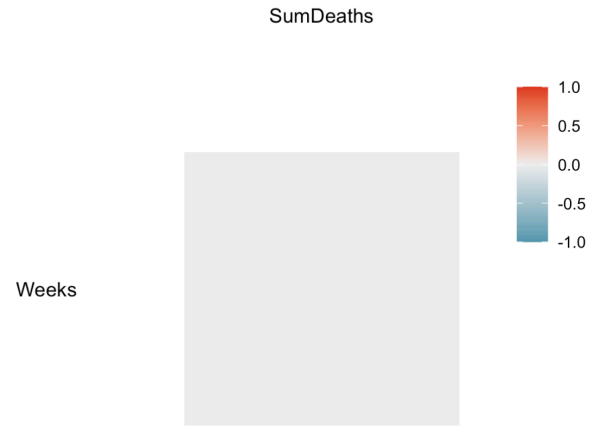
```
40  #Correlation
41  #Compute Pearson correlation
42  cor(df_NL2020weeks$Weeks, df_NL2020weeks$SumDeaths)
43
44  #Correlation Matrix
45  cor(df_NL2020weeks, method = c("kendall"))
46  cor(df_NL2020weeks, method = c("pearson"))
```

Explain:
There is a super weak negative correlation: as the weeks go on there is the slightest decrease in deaths. I tried out a Kendall and Pearson correlation matrix, there is a “stronger” correlation shown with Kendall (which is typically used for ordered data); I thought Peasrson might work as the scatter plot showed a bit of a curve, but isn’t normalized.
Visual:

```
> #Compute Pearson correlation
> cor(df_NL2020weeks$Weeks, df_NL2020weeks$SumDeaths)
[1] -0.01161864
>
> #Correlation Matrix
> cor(df_NL2020weeks, method = c("kendall"))
              Weeks  SumDeaths
Weeks      1.0000000 -0.1301522
SumDeaths -0.1301522  1.0000000
> cor(df_NL2020weeks, method = c("pearson"))
               Weeks   SumDeaths
Weeks      1.00000000 -0.01161864
SumDeaths -0.01161864  1.00000000
```

Generate visualization for the correlation matrix obtained in the last step

```
48  library(GGally)
49  ggcorr(df_NL2020weeks)
```



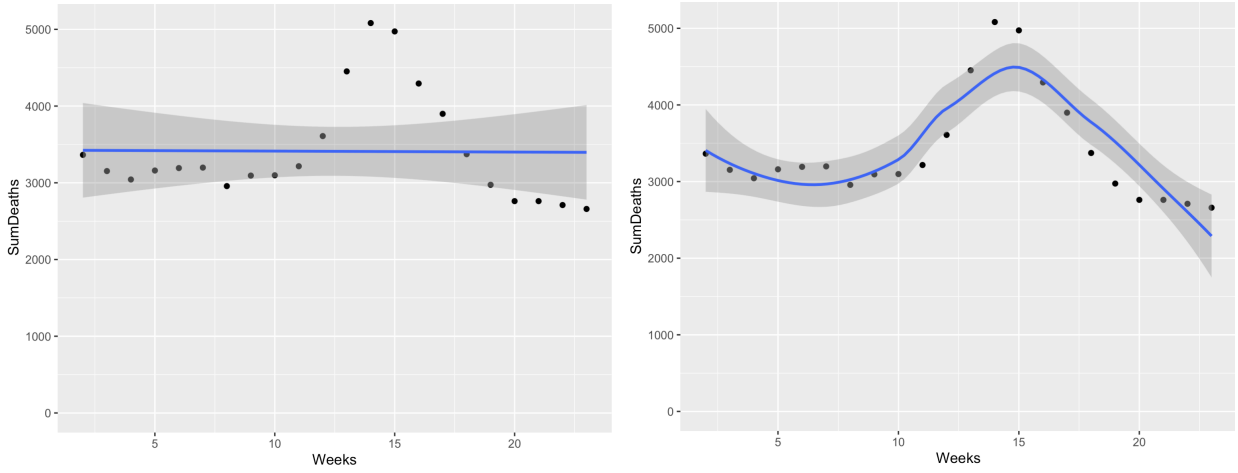
Explain: Using the library GGally you can create a visualization of the correlation matrix. These are more interesting when there are more values to look at, this visualization basically says there isn’t a correlation between the deaths and the weeks.

Perform regression using Week as the predictor and Number of deaths as the response/outcome variable. Show the regression graph with the regression line

```
50
51  #linear regression
52  ggplot(df_NL2020weeks, aes(x=Weeks,y=SumDeaths)) +
53    geom_point()+ ylim(0,5100) + stat_smooth(method="lm")
54
```

```
58  #curve regression???
59  ggplot(df_NL2020weeks, aes(x=Weeks,y=SumDeaths)) +
60    geom_point()+ ylim(0,5100) + geom_smooth()
61
```

Explain:
I found this interesting as this is essentially the same visual that I got from the scatter plot only this time it came with the correlation line. I tried out both codes given to us (linear and curved). I can clearly see why the curved version could be overfitting.
Visual:



Write the line equation

Answer: SumDeath = -1.251*Weeks+3425.732

```
55  #line equation
56  lm(SumDeaths~Weeks, df_NL2020weeks)
```

Explain:
Lm extracts the coefficients from the data frame in y~x form because we're using x (the week) to predict y (the number of deaths in the country)
Visual

```
Call:
lm(formula = SumDeaths ~ Weeks, data = df_NL2020weeks)

Coefficients:
(Intercept)        Weeks
   3425.732       -1.251
```

Analyzing COVID-19 Data using Python

Chris Reynolds | Info 287: FA.24

Load the CSV file into a Python dataframe. Write appropriate commands to view the top and bottom few rows of the dataframe. Print the category of each variable.

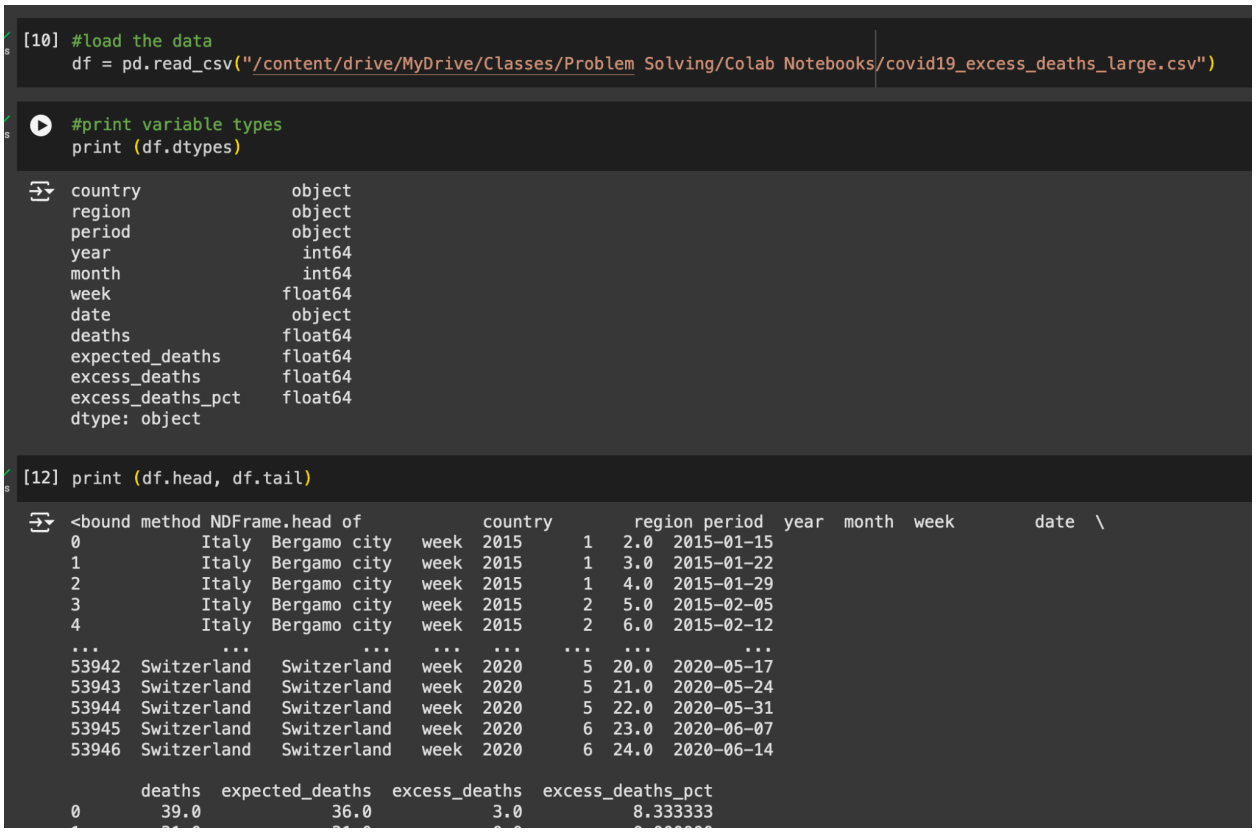
Code:

```
df = pd.read_csv("/content/drive/MyDrive/Classes/PS/Colab Notebooks/covid10_excess_deaths_large.csv")
print (df.head, df.tail)
print(df.dtypes)
```

Explain:

- First line of code tells the Google Colab environment where to find the csv and to read it in as a dataframe named “df”.
- The second line asks for the first 5 and last 5 rows of the dataframe to be printed, so we can double check that the read is complete.
- The third line gives us more information about what variables are in the dataset and what type of data they are.

Visual



```
[10] #load the data
     df = pd.read_csv("/content/drive/MyDrive/Classes/Problem Solving/Colab Notebooks/covid19_excess_deaths_large.csv")

     #print variable types
     print (df.dtypes)

     country              object
     region               object
     period               object
     year                  int64
     month                 int64
     week                float64
     date                 object
     deaths              float64
     expected_deaths     float64
     excess_deaths       float64
     excess_deaths_pct   float64
     dtype: object

[12] print (df.head, df.tail)

     <bound method NDFrame.head of            country        region period  year  month  week        date  \
     0            Italy  Bergamo city   week  2015      1   2.0  2015-01-15
     1            Italy  Bergamo city   week  2015      1   3.0  2015-01-22
     2            Italy  Bergamo city   week  2015      1   4.0  2015-01-29
     3            Italy  Bergamo city   week  2015      2   5.0  2015-02-05
     4            Italy  Bergamo city   week  2015      2   6.0  2015-02-12
     ...            ...          ...    ...   ...    ...   ...         ...
     53942  Switzerland   Switzerland   week  2020      5  20.0  2020-05-17
     53943  Switzerland   Switzerland   week  2020      5  21.0  2020-05-24
     53944  Switzerland   Switzerland   week  2020      5  22.0  2020-05-31
     53945  Switzerland   Switzerland   week  2020      6  23.0  2020-06-07
     53946  Switzerland   Switzerland   week  2020      6  24.0  2020-06-14

           deaths  expected_deaths  excess_deaths  excess_deaths_pct
     0       39.0             36.0            3.0           8.333333
```

Pick one of the two regions: "New York City", "Lombardia region"

Use Python to filter the data using that region.

Answer: New York City

Code:

```
df_NYC = df[df["region"]].isin(["New York City"])]
```

Explain:

- This codes creates a new dataframe from the larger set by telling the computer to find the “region” column and return with all instances of “New York City”

Visual:

Check the type and shape of df_NYC using:

Code:

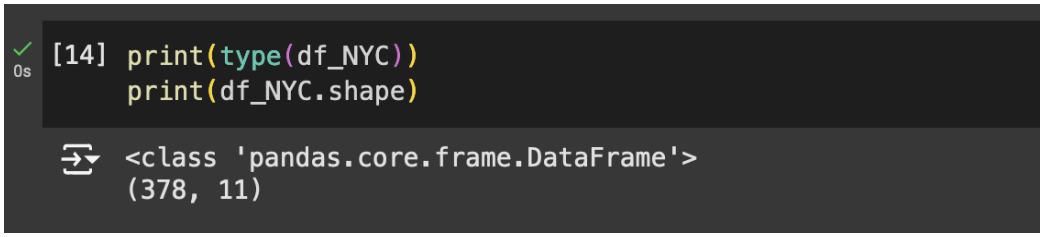
```
print(type(df_NYC))
print(df_NYC.shape)
```

Report your findings: ‘pandas.core.frame.DataFrame’ - (378, 11)

Explain:

- We asked what type of data is our dataframe, and what dimensions is it.
- Python told us that it is a dataframe that was created with pandas, with 378 rows and 11 columns.

Visual:



For the new dataframe, df_NYC (generated in the last question), find out the correlation between the following pairs:

Code format:

```
dataframe_name["indep_verible"].corr(dataframe_name["depent_veriable"])
```

- year and excess_deaths
- year and expected_deaths
- month and excess_deaths
- month and expected_deaths
- week and excess_deaths
- week and expected_deaths
- expected_deaths and excess_deaths

Show your code & report correlation values for these pairs and comment on them:

```
df_NYC["year"].corr(df_NYC["excess_deaths"])
      0.23396172917730804
df_NYC["year"].corr(df_NYC["expected_deaths"])
      0.06610680808556683
```

```
df_NYC["month"].corr(df_NYC["excess_deaths"])
      -0.0869475912348743
df_NYC["month"].corr(df_NYC["expected_deaths"])
      -0.4350711961277345
df_NYC["week"].corr(df_NYC["excess_deaths"])
      -0.08899497249801491
df_NYC["week"].corr(df_NYC["expected_deaths"])
      -0.44655513477878506
df_NYC["expected_deaths"].corr(df_NYC["excess_deaths"])
      -0.011660100774849353
```

Comment:

- As the numbers get closer to 1 or -1 the correlation is stronger. The positive and negative indicates the direction of the correlation.

For the variable excess_deaths (and NOT expected_deaths), pick the two most strongly related variables (obtained in the previous step) and explain (in a few lines) why there might be a strong relationship between them

Year: `0.23396172917730804`
Week: `-0.08899497249801491`
These have the strongest correlation with excess deaths. Year has the strongest relationship which is due to COVID 19. Both week and month seemed pretty weak to me as they were both less than .01; if I had to guess I'd say this could be due to how many data points there are at the week and month level over this many years.

Should excess_deaths be your predictor or outcome? Why?

Outcome as it is the dependent variable.

Based on your results obtained in the previous steps (Q5), pick one outcome and two predictor variables (Important: For regression, the variables that show the strongest correlation with the outcome are often picked as predictors - if all the predictors show weak correlations, pick those with higher correlation than others). State which variables you have picked as predictors (two variables) and the outcome (one variable).

Year: Predictor Variable
Week: Predictor Variable
Excess_deaths: Outcome Variable

Perform regression using one predictor (one of the two in the previous step) and one outcome variable. Show your code.

Code:

```
import numpy as np
from sklearn.linear_model import LinearRegression
x1 = np.array(df_NYC['year']).reshape((-1,1))
y1 = np.array(df_NYC['excess_deaths'])
```

```
mode4 = LinearRegression()
mode4.fit(x1, y1)
r2 = mode4.score(x1,y1)
intercept = mode4.intercept_
slope = mode4.coef_

pred4 = mode4.predict(x1)
```

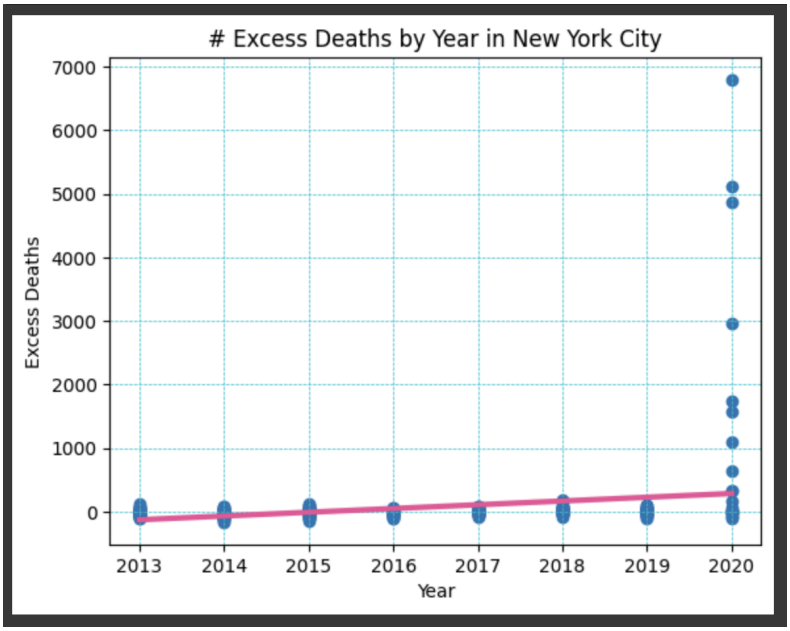
Visual - see ***Appendix: Code***

Show the regression graph with the regression line. Write the line equation.

Code:

```
import matplotlib.pyplot as plt
plt.scatter(df_NYC['year'], df_NYC['excess_deaths'])
plt.plot(x1, pred4, color='#EE5397', linewidth=3)
plt.grid(color = '#00C4D4', linestyle = '--', linewidth = 0.5)
plt.xlabel('Year')
plt.ylabel('Excess Deaths')
plt.title('# Excess Deaths by Year in New York City')
plt.show()
```

Explain:



- These lines use matplot Library and call for it via plt
- .scatter creates plots the data points from the dataframe as a scatter plot where x is years (independent variable) and y is excess deaths (dependent variable)
- The .plot created the the regression line
- I chose to add a dotted line grid to help with understanding the numbers being represented, and added x and y axis labels as well as a graph title.

Line Equation

Answer: y = -119013 + 59x (y = a + bx)

```
print("Intercept:", intercept)
print("Slop:", slope)
print("R2:", r2)
```

Intercept: -119013
Slop: [59]
R2: 0.054738

```
Start coding or generate with AI.
```

Appendix: Code

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
```

```
import pandas as pd
```

```
#load the data
df = pd.read_csv("/content/drive/MyDrive/Classes/Problem Solving/Colab Notebooks/cc
```

```
print (df.head, df.tail)
```

```
<bound method NDFrame.head of            country        region period  year  r
0            Italy  Bergamo city   week  2015      1   2.0  2015-01-15
1            Italy  Bergamo city   week  2015      1   3.0  2015-01-22
2            Italy  Bergamo city   week  2015      1   4.0  2015-01-29
3            Italy  Bergamo city   week  2015      2   5.0  2015-02-05
4            Italy  Bergamo city   week  2015      2   6.0  2015-02-12
...            ...           ...    ...   ...    ...   ...         ...
53942  Switzerland   Switzerland   week  2020      5  20.0  2020-05-17
53943  Switzerland   Switzerland   week  2020      5  21.0  2020-05-24
53944  Switzerland   Switzerland   week  2020      5  22.0  2020-05-31
53945  Switzerland   Switzerland   week  2020      6  23.0  2020-06-07
53946  Switzerland   Switzerland   week  2020      6  24.0  2020-06-14

       deaths  expected_deaths  excess_deaths  excess_deaths_pct
0        39.0             36.0            3.0           8.333333
1        31.0             31.0            0.0           0.000000
2        41.0             39.0            2.0           5.128205
3        31.0             31.0            0.0           0.000000
4        38.0             35.0            3.0           8.571429
...       ...              ...            ...                ...
53942  1127.3           1141.0          -13.7          -1.200701
53943  1180.0           1198.0          -18.0          -1.502504
53944  1072.2           1171.0          -98.8          -8.437233
53945  1148.8           1155.0           -6.2          -0.536797
53946  1088.7           1111.0          -22.3          -2.007201

[53947 rows x 11 columns]> <bound method NDFrame.tail of            country        region period  year
0            Italy  Bergamo city   week  2015      1   2.0  2015-01-15
1            Italy  Bergamo city   week  2015      1   3.0  2015-01-22
2            Italy  Bergamo city   week  2015      1   4.0  2015-01-29
3            Italy  Bergamo city   week  2015      2   5.0  2015-02-05
```

```
4            Italy  Bergamo city   week  2015      2   6.0  2015-02-12
...            ...           ...    ...   ...    ...   ...         ...
53942  Switzerland   Switzerland   week  2020      5  20.0  2020-05-17
53943  Switzerland   Switzerland   week  2020      5  21.0  2020-05-24
53944  Switzerland   Switzerland   week  2020      5  22.0  2020-05-31
53945  Switzerland   Switzerland   week  2020      6  23.0  2020-06-07
53946  Switzerland   Switzerland   week  2020      6  24.0  2020-06-14

       deaths  expected_deaths  excess_deaths  excess_deaths_pct
0        39.0             36.0            3.0           8.333333
1        31.0             31.0            0.0           0.000000
2        41.0             39.0            2.0           5.128205
3        31.0             31.0            0.0           0.000000
4        38.0             35.0            3.0           8.571429
...       ...              ...            ...                ...
53942  1127.3           1141.0          -13.7          -1.200701
53943  1180.0           1198.0          -18.0          -1.502504
53944  1072.2           1171.0          -98.8          -8.437233
53945  1148.8           1155.0           -6.2          -0.536797
53946  1088.7           1111.0          -22.3          -2.007201

[53947 rows x 11 columns]>
```

```
#print variable types
print (df.dtypes)
```

```
country              object
region               object
period               object
year                  int64
month                 int64
week                float64
date                 object
deaths              float64
expected_deaths     float64
excess_deaths       float64
excess_deaths_pct   float64
dtype: object
```

```
df_NYC = df[df["region"].isin(["New York City"])]
```

```
print(type(df_NYC))
print(df_NYC.shape)
```

```
<class 'pandas.core.frame.DataFrame'>
(378, 11)
```

```
df_NYC["year"].corr(df_NYC["excess_deaths"])
```

```
0.23396172917730804
```

```
df_NYC["year"].corr(df_NYC["expected_deaths"])
```

```
0.06610680808556683
```

```
df_NYC["month"].corr(df_NYC["excess_deaths"])
```

```
-0.0869475912348743
```

```
df_NYC["month"].corr(df_NYC["expected_deaths"])
```

```
-0.4350711961277345
```

```
df_NYC["week"].corr(df_NYC["excess_deaths"])
```

```
-0.08899497249801491
```

```
df_NYC["week"].corr(df_NYC["expected_deaths"])
```

```
-0.44655513477878506
```

```
df_NYC["expected_deaths"].corr(df_NYC["excess_deaths"])
```

```
-0.011660100774849353
```

```
import numpy as np
from sklearn.linear_model import LinearRegression
```

```
import matplotlib.pyplot as plt
```

```
np.corrcoef(df_NYC['excess_deaths'], df_NYC['deaths'])
```

```
array([[1.        , 0.99174825],
       [0.99174825, 1.        ]])
```

```
#Regression using scikitlearn - y = a +bx
#y is outcoming x is predictor a is y-intercept b is the slope
```

```
x1 = np.array(df_NYC['year']).reshape((-1,1))
y1 = np.array(df_NYC['excess_deaths'])
mode4 = LinearRegression()
mode4.fit(x1, y1)
r2 = mode4.score(x1,y1)
intercept = mode4.intercept_
```

```
slope = mode4.coef_


print("Intercept:", intercept)
print("Slop:", slope)
print("R2:", r2)
```

```
Intercept: -119013.32539623718
Slop: [59.05833713]
R2: 0.05473809071963642
```

```
pred4 = mode4.predict(x1)


plt.scatter(df_NYC['year'], df_NYC['excess_deaths'])
plt.plot(x1, pred4, color='#EE5397', linewidth=3)
plt.grid(color = '#00C4D4', linestyle = '--', linewidth = 0.5)
plt.xlabel('Year')
plt.ylabel('Excess Deaths')
plt.title('# Excess Deaths by Year in New York City')
plt.show()
```