Project2_DATA607_Chunjie_Nan

Chunjie Nan

10/2/2021

Task

Choose any three of the "wide" datasets identified in the Week 6 Discussion items. (You may use your own dataset; please don't use my Sample Post dataset, since that was used in your Week 6 assignment!) For each of the three chosen datasets:

Create a .CSV file (or optionally, a MySQL database!) that includes all of the information included in the dataset. You're encouraged to use a "wide" structure similar to how the information appears in the discussion item, so that you can practice tidying and transformations as described below.

Read the information from your .CSV file into R, and use tidyr and dplyr as needed to tidy and transform your data. [Most of your grade will be based on this step!]

Perform the analysis requested in the discussion item. Your code should be in an R Markdown file, posted to rpubs.com, and should include narrative descriptions of your data cleanup work, analysis, and conclusions.

1. West Nile virus

Web Scraping

```
library(RCurl)
library(XML)
url<-'https://www.cdc.gov/westnile/statsmaps/cumMapsData.html#four'


web<- getURL(url)
virus<- htmlParse(web)

virus<-readHTMLTable(virus)
data<-virus$`West Nile virus disease cases reported to CDC by state of residence, 1999-2018`
# data is successfully imported.
```

Subseting Data for East Coast States only

My interest in the analysis is East Coast State only. Therefore, restrict the data for the East Coast States.

```
library(dplyr)
```

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

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

## The following objects are masked from 'package:base':
```

```
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:RCurl':
##
##     complete
```

```
# Convert Characters to numeric
virus_data <- as.data.frame(sapply(data, as.numeric),na.omit=T)
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
# Recover the lost state variable
virus_data$State<-data$State


# Subset the data the East Coast States only
east_virus<- virus_data%>%
  filter(State %in% c('Connecticut','Delaware','Florida','Georgia', 'Maine','Maryland','Massachusetts',
```

The tedious part for this data is, there are many “0”s, I assume they are not missing values, they are the real value of 0 or 0 reports in virus case.

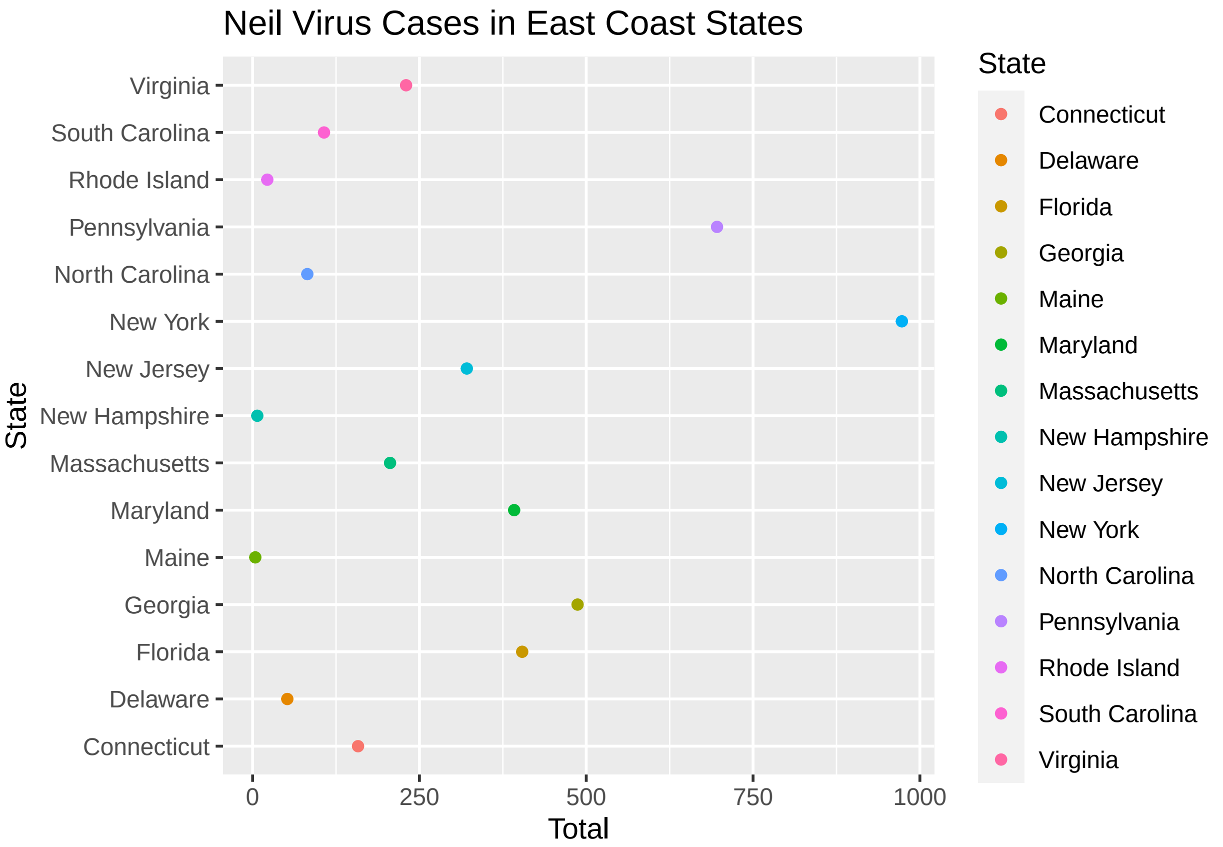
Analysis

```
east_virus%>%
  group_by(State)%>%
  summarise(Total, percent = Total/sum(east_virus$Total))%>%
  arrange(desc(percent))
```

```
## # A tibble: 15 x 3
##    State          Total  percent
##    <chr>          <dbl>    <dbl>
##  1 New York         973 0.235
##  2 Pennsylvania     696 0.168
##  3 Georgia          487 0.118
##  4 Florida          404 0.0976
##  5 Maryland         392 0.0947
##  6 New Jersey       321 0.0775
##  7 Virginia         230 0.0555
##  8 Massachusetts    206 0.0497
##  9 Connecticut      158 0.0382
## 10 South Carolina   107 0.0258
## 11 North Carolina    82 0.0198
## 12 Delaware          52 0.0126
## 13 Rhode Island      22 0.00531
## 14 New Hampshire      7 0.00169
## 15 Maine              4 0.000966
```

```
east_virus<-east_virus%>%
  mutate(percent = Total/sum(east_virus$Total))
```

```
library(ggplot2)
ggplot(data = east_virus, mapping = aes(x = State, y = Total)) +
  geom_point(mapping = aes(color = State)) +
  coord_flip()+ggtitle("Neil Virus Cases in East Coast States")
```



According to the summary and plot, New York has the highest number of Nile Virus report of 973 cases, then Pennsylvania follows the next of 696 cases. These two states share almost 40 percent of the Nile Virus cases among the East Coast States. Rhode Island, New Hampshire, and Maine listed the lowest number of cases in the Nile Virus report.

2. GDP Annual Growth

Import the Excel data

linked phrase

This is the GDP growth(annual%) data which is provided by The World Bank.

```
library(readxl)
gdp<-read_excel('GDP.xls')
```

```
## New names:
## * `` -> ...3
## * `` -> ...4
## * `` -> ...5
## * `` -> ...6
## * `` -> ...7
## * ...
```

```
# rename the variables.
colnames(gdp)<-gdp[3,]

# delete the top 3 rows
```

```
gdp<-gdp[-c(1:3),]
```

Tidy data

The variables include Country Code, Indicator Name, Indicator Code are not necessary, also I'm going to see the year from 2000 to 2020 Which is included the period of the 2008 recession and 2020 pandemic. Let's see how does the pandemic has impacted the annual GDP growth for the top 10 GDP countries of the year 2019.

linked phrase According to the Investopedia, below is the list of Top 10 highest Nominal GDP countries in 2019.

1.United States 2.China 3.Japan 4.Germany 5.India 6.United Kingdom 7.France 8.Italy 9.Brazil 10.Canada

```
library(dplyr)
library(tidyr)
# subset the year and countries.
gdp<-gdp[,-c(2:44)]
gdp<-gdp%>%
  filter(`Country Name` %in% c('United States','China','Japan','Germany','India','United Kingdom','Franc
gdp_growth<- as.data.frame(sapply(gdp, as.numeric),na.omit=T)
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
gdp_growth[,1] <-gdp[,1]

names(which(sapply(gdp_growth, anyNA)))
```

```
## [1] "2020"
```

There is NA in 2020 Japan. I have to remove Japan in this case because there is no way to estimate the external shock of a pandemic to measure the GDP growth for Japan since most of the countries' GDP growth turned negative in 2020.

```
library(dplyr)
gdp_growth<-gdp_growth%>%
  filter(`Country Name`!= 'Japan')

#Convert to Time Series
ts<-ts(gdp_growth[,c(2:21)])

#Add a column for prediction
gdp_growth<-gdp_growth%>%
  mutate('2020_prediction')
```

Time Series Modeling

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
#Brazil
ts_model_Brazil<-ts[1,]%>%
  auto.arima(stepwise = FALSE, approximation =FALSE) %>% forecast(h=1)
gdp_growth[1,23]<-ts_model_Brazil$fitted[1]
```

```
#Canada
ts_model_Canana<-ts[2,]%>%
  auto.arima(stepwise = FALSE, approximation =FALSE) %>% forecast(h=1)
gdp_growth[2,23]<-ts_model_Canana$fitted[1]


#China
ts_model_China<-ts[3,]%>%
  auto.arima(stepwise = FALSE, approximation =FALSE) %>% forecast(h=1)
gdp_growth[3,23]<-ts_model_China$fitted[1]


#Germany
ts_model_Germany<-ts[4,]%>%
  auto.arima(stepwise = FALSE, approximation =FALSE) %>% forecast(h=1)
gdp_growth[4,23]<-ts_model_Germany$fitted[1]


#France
ts_model_France<-ts[5,]%>%
  auto.arima(stepwise = FALSE, approximation =FALSE) %>% forecast(h=1)
gdp_growth[5,23]<-ts_model_France$fitted[1]


#United Kingdom
ts_model_UK<-ts[6,]%>%
  auto.arima(stepwise = FALSE, approximation =FALSE) %>% forecast(h=1)
gdp_growth[6,23]<-ts_model_UK$fitted[1]


#India
ts_model_India<-ts[7,]%>%
  auto.arima(stepwise = FALSE, approximation =FALSE) %>% forecast(h=1)
gdp_growth[7,23]<-ts_model_India$fitted[1]


#Italy
ts_model_Italy<-ts[8,]%>%
  auto.arima(stepwise = FALSE, approximation =FALSE) %>% forecast(h=1)
gdp_growth[8,23]<-ts_model_Italy$fitted[1]

#United States
ts_model_US<-ts[9,]%>%
  auto.arima(stepwise = FALSE, approximation =FALSE) %>% forecast(h=1)
gdp_growth[9,23]<-ts_model_US$fitted[1]
```

Calculate the difference

```
gdp_growth[,23]<-as.data.frame(sapply(gdp_growth[,23], as.numeric),na.omit=T)

gdp_growth<-gdp_growth%>%
  mutate(difference = gdp_growth$`2020`- gdp_growth$`"2020_prediction"`)
```

The difference tells us how much has the country been impacted by the pandemic. According to the analysis, the Covid- 19 impacted the GDP growth of India and the United Kingdom the most, more than 10 % of their GDP growth. The United States and China have been affected the least among the top GDP countries in 2020, However, it is about 6 % of their GDP annual growth. In this analysis, I didn't separate the training set and test set to testify my model accuracy since it is not the main object for this assignment.

3.TAP Aid NYC

Import the TAP data, and subset CUNY only

```
tuition<-read.csv('https://raw.githubusercontent.com/nancunjie4560/Data607/master/Tuition_Assistance_Pr

cuny<-subset(tuition, TAP.Sector.Group =='2-CUNY CC'|TAP.Sector.Group =='1-CUNY SR' )
which(is.na(cuny)) # No missing value
```

```
## integer(0)
```

```
max(cuny$TAP.Fall.Headcount)
```

```
## [1] 9529
```

```
arrange<-arrange(cuny,desc(TAP.Fall.Headcount))
arrange[c(1:10),]
```

```
##    Academic.Year TAP.College.Code Federal.School.Code Level TAP.Level.of.Study
## 1           2015             1404                2691     U     2 yr Undergrad
## 2           2017             1404                2691     U     2 yr Undergrad
## 3           2016             1404                2691     U     2 yr Undergrad
## 4           2014             1404                2691     U     2 yr Undergrad
## 5           2018             1404                2691     U     2 yr Undergrad
## 6           2012             1404                2691     U     2 yr Undergrad
## 7           2013             1404                2691     U     2 yr Undergrad
## 8           2019             1404                2691     U     2 yr Undergrad
## 9           2011             1404                2691     U     2 yr Undergrad
## 10          2018             1413                2689     U     4 yr Undergrad
##       TAP.College.Name Sector.Type TAP.Sector.Group TAP.Fall.Headcount
## 1    CUNY MANHATTAN CC      PUBLIC        2-CUNY CC               9529
## 2    CUNY MANHATTAN CC      PUBLIC        2-CUNY CC               9365
## 3    CUNY MANHATTAN CC      PUBLIC        2-CUNY CC               9250
## 4    CUNY MANHATTAN CC      PUBLIC        2-CUNY CC               8604
## 5    CUNY MANHATTAN CC      PUBLIC        2-CUNY CC               8509
## 6    CUNY MANHATTAN CC      PUBLIC        2-CUNY CC               8139
## 7    CUNY MANHATTAN CC      PUBLIC        2-CUNY CC               8116
## 8    CUNY MANHATTAN CC      PUBLIC        2-CUNY CC               7790
## 9    CUNY MANHATTAN CC      PUBLIC        2-CUNY CC               7740
## 10 CUNY HUNTER COLLEGE      PUBLIC        1-CUNY SR               7355
```

According to the sorting, the CUNY Manhattan Community College receives the most TAP aid every year among the CUNY schools.

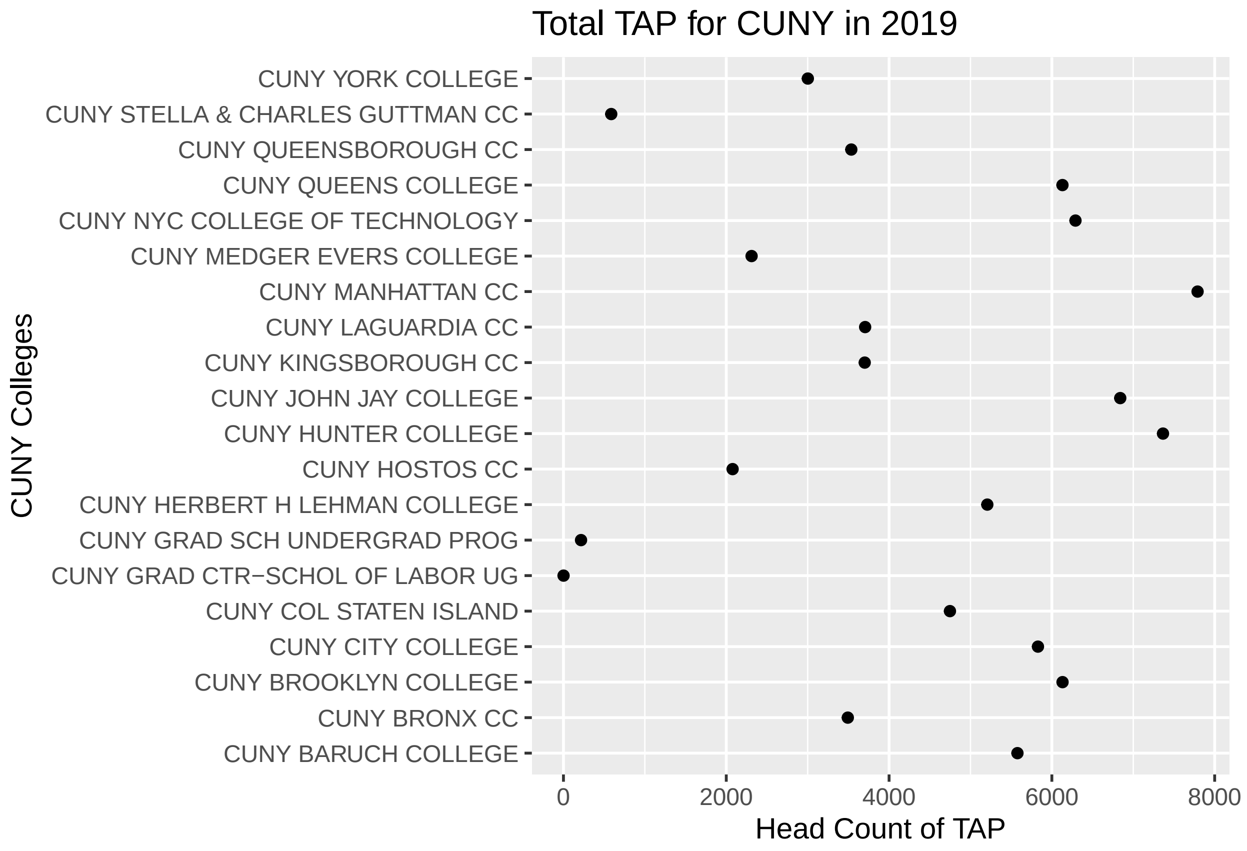
```
cuny%>%
  filter(TAP.Sector.Group=='1-CUNY SR')%>%
  group_by(TAP.College.Name)%>%
  arrange(desc(TAP.Fall.Headcount))
```

```
## # A tibble: 656 x 9
## # Groups:   TAP.College.Name [24]
##    Academic.Year TAP.College.Code Federal.School.Code Level TAP.Level.of.Study
##            <int>            <int>               <int> <chr> <chr>
##  1          2018             1413                2689 U     4 yr Undergrad
##  2          2019             1413                2689 U     4 yr Undergrad
##  3          2017             1413                2689 U     4 yr Undergrad
##  4          2019             1414                2693 U     4 yr Undergrad
##  5          2018             1414                2693 U     4 yr Undergrad
##  6          2016             1413                2689 U     4 yr Undergrad
##  7          2015             1413                2689 U     4 yr Undergrad
##  8          2017             1414                2693 U     4 yr Undergrad
##  9          2017             1416                2690 U     4 yr Undergrad
## 10          2013             1414                2693 U     4 yr Undergrad
## # ... with 646 more rows, and 4 more variables: TAP.College.Name <chr>,
## #   Sector.Type <chr>, TAP.Sector.Group <chr>, TAP.Fall.Headcount <int>
```

Year 2019

```
CUNY<-cuny%>%
  filter(Academic.Year == 2019)%>%
  group_by(TAP.College.Name) %>%
  summarise(SUM=sum(TAP.Fall.Headcount))%>%
  arrange(desc(SUM))

ggplot(CUNY, aes(x=TAP.College.Name, y=SUM))+geom_point()+coord_flip()+ggtitle("Total TAP for CUNY in 20
```



```
cuny%>%
  filter(Academic.Year == 2019)%>%
summarise(TOTAL=sum(TAP.Fall.Headcount))
```

```
##   TOTAL
## 1 84538
```

In 2019, Among the 4 Year College, CUNY Hunter College and CUNY John Jay College receives the most of the TAP aid. 7365 students in Hunter College received TAP Aid and 6840 students in John Jay College received it. CUNY Graduate Center - School of Labor received only 1 TAP which is the least. In total, 84538 CUN Students received the TAP Aid in the year 2019.