Jiaming Ren 217218863 Assignment 1

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Warning: package 'tibble' was built under R version 4.0.5
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
## Warning: package 'purrr' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
## Warning: package 'stringr' was built under R version 4.0.5
```

```
## Warning: package 'forcats' was built under R version 4.0.5
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(stringr)
library(tidyr)
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.0.5
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(dplyr)
library(naniar)
```

```
## Warning: package 'naniar' was built under R version 4.0.5
```

Q1

read the World_development_Indicators.csv

```
ds_worldDI = read_csv("World_Development_Indicators.csv")
```

```
## Rows: 12157 Columns: 64
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): Country Name, Country Code, Series Name, Series Code
## dbl (60): 1960.00, 1961.00, 1962.00, 1963.00, 1964.00, 1965.00, 1966.00, 196...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

display head of dataset

```
##display first 5 records
head(ds_worldDI)
```

```
## # A tibble: 6 x 64
##   `Country Name` `Country Code` `Series Name`  `Series Code` `1960.00` `1961.00`
##   <chr>          <chr>          <chr>          <chr>             <dbl>     <dbl>
## 1 Afghanistan    AFG            Mobile cellul~ IT.CEL.SETS.~      0        NA
## 2 Afghanistan    AFG            CO2 emissions~ EN.ATM.CO2E.~      0.05      0.05
## 3 Afghanistan    AFG            Exports of go~ NE.EXP.GNFS.~      4.13      4.45
## 4 Afghanistan    AFG            Imports of go~ NE.IMP.GNFS.~      7.02      8.1
## 5 Afghanistan    AFG            Fertility rat~ SP.DYN.TFRT.~      7.45      7.45
## 6 Afghanistan    AFG            Gross capital~ NE.GDI.TOTL.~     16.1      16.6
## # ... with 58 more variables: 1962.00 <dbl>, 1963.00 <dbl>, 1964.00 <dbl>,
## #   1965.00 <dbl>, 1966.00 <dbl>, 1967.00 <dbl>, 1968.00 <dbl>, 1969.00 <dbl>,
## #   1970.00 <dbl>, 1971.00 <dbl>, 1972.00 <dbl>, 1973.00 <dbl>, 1974.00 <dbl>,
## #   1975.00 <dbl>, 1976.00 <dbl>, 1977.00 <dbl>, 1978.00 <dbl>, 1979.00 <dbl>,
## #   1980.00 <dbl>, 1981.00 <dbl>, 1982.00 <dbl>, 1983.00 <dbl>, 1984.00 <dbl>,
## #   1985.00 <dbl>, 1986.00 <dbl>, 1987.00 <dbl>, 1988.00 <dbl>, 1989.00 <dbl>,
## #   1990.00 <dbl>, 1991.00 <dbl>, 1992.00 <dbl>, 1993.00 <dbl>, ...
```

columns and rows of dataset

```
#number of columns
print("number of columns")
```

```
## [1] "number of columns"
```

```
ncol(ds_worldDI)
```

```
## [1] 64
```

```
#number of rows
print("number of rows")
```

```
## [1] "number of rows"
```

```
nrow(ds_worldDI)
```

```
## [1] 12157
```

```
#the structure of dataset
print("the structure of dataset")
```

```
## [1] "the structure of dataset"
```

```
str(ds_worldDI)
```

```
## spec_tbl_df [12,157 x 64] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Country Name: chr [1:12157] "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ Country Code: chr [1:12157] "AFG" "AFG" "AFG" "AFG" ...
##  $ Series Name : chr [1:12157] "Mobile cellular subscriptions (per 100 people)" "CO2 emissions (me
tric tons per capita)" "Exports of goods and services (% of GDP)" "Imports of goods and services (% o
f GDP)" ...
##  $ Series Code : chr [1:12157] "IT.CEL.SETS.P2" "EN.ATM.CO2E.PC" "NE.EXP.GNFS.ZS" "NE.IMP.GNFS.ZS"
...
##  $ 1960.00     : num [1:12157] 0 0.05 4.13 7.02 7.45 ...
##  $ 1961.00     : num [1:12157] NA 0.05 4.45 8.1 7.45 ...
##  $ 1962.00     : num [1:12157] NA 0.07 4.88 9.35 7.45 ...
##  $ 1963.00     : num [1:12157] NA 0.07 9.17 16.86 7.45 ...
##  $ 1964.00     : num [1:12157] NA 0.09 8.89 18.06 7.45 ...
##  $ 1965.00     : num [1:12157] 0 0.1 11.26 21.41 7.45 ...
##  $ 1966.00     : num [1:12157] NA 0.11 8.57 18.57 7.45 ...
##  $ 1967.00     : num [1:12157] NA 0.12 6.77 14.21 7.45 ...
##  $ 1968.00     : num [1:12157] NA 0.12 8.9 15.21 7.45 ...
##  $ 1969.00     : num [1:12157] NA 0.09 10.09 14.98 7.45 ...
##  $ 1970.00     : num [1:12157] 0 0.15 9.78 11.94 7.45 ...
##  $ 1971.00     : num [1:12157] NA 0.17 10.92 16.14 7.45 ...
##  $ 1972.00     : num [1:12157] NA 0.13 14.76 18.11 7.45 ...
##  $ 1973.00     : num [1:12157] NA 0.14 12.95 14.74 7.45 ...
##  $ 1974.00     : num [1:12157] NA 0.15 14.02 14.85 7.45 ...
##  $ 1975.00     : num [1:12157] 0 0.17 12.68 14.27 7.45 ...
##  $ 1976.00     : num [1:12157] 0 0.15 13.22 14.87 7.45 ...
##  $ 1977.00     : num [1:12157] 0 0.18 11.66 14.82 7.45 ...
##  $ 1978.00     : num [1:12157] 0 0.16 10.84 13.87 7.45 ...
##  $ 1979.00     : num [1:12157] 0 0.17 NA NA 7.45 ...
##  $ 1980.00     : num [1:12157] 0 0.13 NA NA 7.45 ...
##  $ 1981.00     : num [1:12157] 0 0.15 NA NA 7.45 ...
##  $ 1982.00     : num [1:12157] 0 0.16 NA NA 7.45 ...
##  $ 1983.00     : num [1:12157] 0 0.2 NA NA 7.45 ...
##  $ 1984.00     : num [1:12157] 0 0.23 NA NA 7.46 ...
##  $ 1985.00     : num [1:12157] 0 0.29 NA NA 7.46 ...
##  $ 1986.00     : num [1:12157] 0 0.27 NA NA 7.46 ...
##  $ 1987.00     : num [1:12157] 0 0.27 NA NA 7.46 ...
##  $ 1988.00     : num [1:12157] 0 0.25 NA NA 7.46 ...
##  $ 1989.00     : num [1:12157] 0 0.23 NA NA 7.46 ...
##  $ 1990.00     : num [1:12157] 0 0.24 NA NA 7.47 ...
##  $ 1991.00     : num [1:12157] 0 0.21 NA NA 7.48 ...
##  $ 1992.00     : num [1:12157] 0 0.1 NA NA 7.5 ...
##  $ 1993.00     : num [1:12157] 0 0.09 NA NA 7.54 ...
##  $ 1994.00     : num [1:12157] 0 0.08 NA NA 7.57 ...
##  $ 1995.00     : num [1:12157] 0 0.07 NA NA 7.61 ...
##  $ 1996.00     : num [1:12157] 0 0.06 NA NA 7.63 ...
##  $ 1997.00     : num [1:12157] 0 0.06 NA NA 7.63 ...
##  $ 1998.00     : num [1:12157] 0 0.05 NA NA 7.61 ...
##  $ 1999.00     : num [1:12157] 0 0.04 NA NA 7.56 ...
##  $ 2000.00     : num [1:12157] 0 0.04 NA NA 7.49 ...
##  $ 2001.00     : num [1:12157] 0 0.04 NA NA 7.39 ...
##  $ 2002.00     : num [1:12157] 0.11 0.05 NA NA 7.27 ...
##  $ 2003.00     : num [1:12157] 0.84 0.06 NA NA 7.15 ...
##  $ 2004.00     : num [1:12157] 2.43 0.05 NA NA 7.02 ...
##  $ 2005.00     : num [1:12157] 4.68 0.06 NA NA 6.88 ...
##  $ 2006.00     : num [1:12157] 9.53 0.07 NA NA 6.72 ...
##  $ 2007.00     : num [1:12157] 17.23 0.09 NA NA 6.56 ...
```

```
##  $ 2008.00     : num [1:12157] 28.49 0.16 NA NA 6.37 ...
##  $ 2009.00     : num [1:12157] 36.98 0.21 NA NA 6.18 ...
##  $ 2010.00     : num [1:12157] 35 0.3 NA NA 5.98 ...
##  $ 2011.00     : num [1:12157] 45.81 0.41 NA NA 5.77 ...
##  $ 2012.00     : num [1:12157] 49.23 0.34 NA NA 5.56 ...
##  $ 2013.00     : num [1:12157] 52.08 0.26 NA NA 5.36 ...
##  $ 2014.00     : num [1:12157] 55.16 0.23 NA NA 5.16 ...
##  $ 2015.00     : num [1:12157] 57.27 0.23 NA NA 4.98 ...
##  $ 2016.00     : num [1:12157] 61.05 0.21 NA NA 4.8 ...
##  $ 2017.00     : num [1:12157] 65.93 0.2 NA NA 4.63 ...
##  $ 2018.00     : num [1:12157] 59.12 0.2 NA NA 4.47 ...
##  $ 2019.00     : num [1:12157] 59.36 NA NA NA 4.32 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `Country Name` = col_character(),
##   ..   `Country Code` = col_character(),
##   ..   `Series Name` = col_character(),
##   ..   `Series Code` = col_character(),
##   ..   `1960.00` = col_double(),
##   ..   `1961.00` = col_double(),
##   ..   `1962.00` = col_double(),
##   ..   `1963.00` = col_double(),
##   ..   `1964.00` = col_double(),
##   ..   `1965.00` = col_double(),
##   ..   `1966.00` = col_double(),
##   ..   `1967.00` = col_double(),
##   ..   `1968.00` = col_double(),
##   ..   `1969.00` = col_double(),
##   ..   `1970.00` = col_double(),
##   ..   `1971.00` = col_double(),
##   ..   `1972.00` = col_double(),
##   ..   `1973.00` = col_double(),
##   ..   `1974.00` = col_double(),
##   ..   `1975.00` = col_double(),
##   ..   `1976.00` = col_double(),
##   ..   `1977.00` = col_double(),
##   ..   `1978.00` = col_double(),
##   ..   `1979.00` = col_double(),
##   ..   `1980.00` = col_double(),
##   ..   `1981.00` = col_double(),
##   ..   `1982.00` = col_double(),
##   ..   `1983.00` = col_double(),
##   ..   `1984.00` = col_double(),
##   ..   `1985.00` = col_double(),
##   ..   `1986.00` = col_double(),
##   ..   `1987.00` = col_double(),
##   ..   `1988.00` = col_double(),
##   ..   `1989.00` = col_double(),
##   ..   `1990.00` = col_double(),
##   ..   `1991.00` = col_double(),
##   ..   `1992.00` = col_double(),
##   ..   `1993.00` = col_double(),
##   ..   `1994.00` = col_double(),
##   ..   `1995.00` = col_double(),
##   ..   `1996.00` = col_double(),
##   ..   `1997.00` = col_double(),
```

```
##   ..   `1998.00` = col_double(),
##   ..   `1999.00` = col_double(),
##   ..   `2000.00` = col_double(),
##   ..   `2001.00` = col_double(),
##   ..   `2002.00` = col_double(),
##   ..   `2003.00` = col_double(),
##   ..   `2004.00` = col_double(),
##   ..   `2005.00` = col_double(),
##   ..   `2006.00` = col_double(),
##   ..   `2007.00` = col_double(),
##   ..   `2008.00` = col_double(),
##   ..   `2009.00` = col_double(),
##   ..   `2010.00` = col_double(),
##   ..   `2011.00` = col_double(),
##   ..   `2012.00` = col_double(),
##   ..   `2013.00` = col_double(),
##   ..   `2014.00` = col_double(),
##   ..   `2015.00` = col_double(),
##   ..   `2016.00` = col_double(),
##   ..   `2017.00` = col_double(),
##   ..   `2018.00` = col_double(),
##   ..   `2019.00` = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Q2

1.Select unique value of series name and code

2.count unique series

3.print first 9 rows

4.find how many contain the keyword fertility

```
#Total unique values for the Columns Series Name and code
unique_series <- unique(na.omit(ds_worldDI[c("Series Name","Series Code")]))

unique_series
```

```
## # A tibble: 55 x 2
##    `Series Name`                                                  `Series Code`
##    <chr>                                                          <chr>
##  1 Mobile cellular subscriptions (per 100 people)                 IT.CEL.SETS.~
##  2 CO2 emissions (metric tons per capita)                         EN.ATM.CO2E.~
##  3 Exports of goods and services (% of GDP)                       NE.EXP.GNFS.~
##  4 Imports of goods and services (% of GDP)                       NE.IMP.GNFS.~
##  5 Fertility rate, total (births per woman)                       SP.DYN.TFRT.~
##  6 Gross capital formation (% of GDP)                             NE.GDI.TOTL.~
##  7 Life expectancy at birth, total (years)                        SP.DYN.LE00.~
##  8 Adolescent fertility rate (births per 1,000 women ages 15-19)  SP.ADO.TFRT
##  9 Population, total                                              SP.POP.TOTL
## 10 Net official development assistance and official aid received ~ DT.ODA.ALLD.~
## # ... with 45 more rows
```

```
#get the length of unique series
len_unique_series <- unique_series[1] %>% count()

len_unique_series
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    55
```

```
#print the first 9 unique series
unique_series %>% head(9)
```

```
## # A tibble: 9 x 2
##   `Series Name`                                                  `Series Code`
##   <chr>                                                          <chr>
## 1 Mobile cellular subscriptions (per 100 people)                 IT.CEL.SETS.P2
## 2 CO2 emissions (metric tons per capita)                         EN.ATM.CO2E.PC
## 3 Exports of goods and services (% of GDP)                       NE.EXP.GNFS.ZS
## 4 Imports of goods and services (% of GDP)                       NE.IMP.GNFS.ZS
## 5 Fertility rate, total (births per woman)                       SP.DYN.TFRT.IN
## 6 Gross capital formation (% of GDP)                             NE.GDI.TOTL.ZS
## 7 Life expectancy at birth, total (years)                        SP.DYN.LE00.IN
## 8 Adolescent fertility rate (births per 1,000 women ages 15-19)  SP.ADO.TFRT
## 9 Population, total                                              SP.POP.TOTL
```

```
#get series that contain the keyword Fertility
fertility <- ds_worldDI[str_detect(ds_worldDI$"Series Name", regex("fertility",ignore_case = TRUE)),
 ]
fertility %>% length()
```

```
## [1] 64
```

```
#get unique series that contain the keyword Fertility
unique_fertility <- unique_series %>% filter(`Series Name` == regex("Fertility", ignore_case = T))
#row numbers of rows that contain fertility
unique_fertility %>% length()
```

```
## [1] 2
```

(1) top 5 countries

1. get all rows for GDP (current US

) ### 2. sort the dataset and get top 5 countries with highest GDP (current US) in 2019

(2) bottom 5 countries

3. sort the dataset and get bottom 5 countries with hightest GDP (current US$) in 2019

4. only keep the GDP data related to the 10 countries

```
#select country name and 2019.00 columns
gdp_2019 <- ds_worldDI %>% filter(ds_worldDI$"Series Name" == "GDP (current US$)") %>% select("Countr
y Name", "2019.00")

#sort the rows by the column 2019.00 and select bottom 5 countries
bot <- gdp_2019 %>% arrange(gdp_2019$"2019.00") %>% head(5)
bot
```

```
## # A tibble: 5 x 2
##   `Country Name`    `2019.00`
##   <chr>                 <dbl>
## 1 Tuvalu            47271463.
## 2 Nauru            118223430.
## 3 Kiribati         194647202.
## 4 Marshall Islands 239462200
## 5 Palau            268354900
```

```
#sort the rows by the column 2019.00 and select top 5 countries
top <- gdp_2019 %>% arrange(desc(gdp_2019$"2019.00")) %>% head(5)
top
```

```
## # A tibble: 5 x 2
##   `Country Name` `2019.00`
##   <chr>              <dbl>
## 1 United States    2.14e13
## 2 China            1.43e13
## 3 Japan            5.06e12
## 4 Germany          3.86e12
## 5 India            2.87e12
```

```
#combined two top and bot datasets
filtered_countries <- rbind(top,bot)$"Country Name"

#filter and keep the GDP data related to the 10 coutries
filtered_ds_worldDI <- ds_worldDI %>% filter(ds_worldDI$"Country Name" %in% filtered_countries)
filtered_ds_worldDI
```

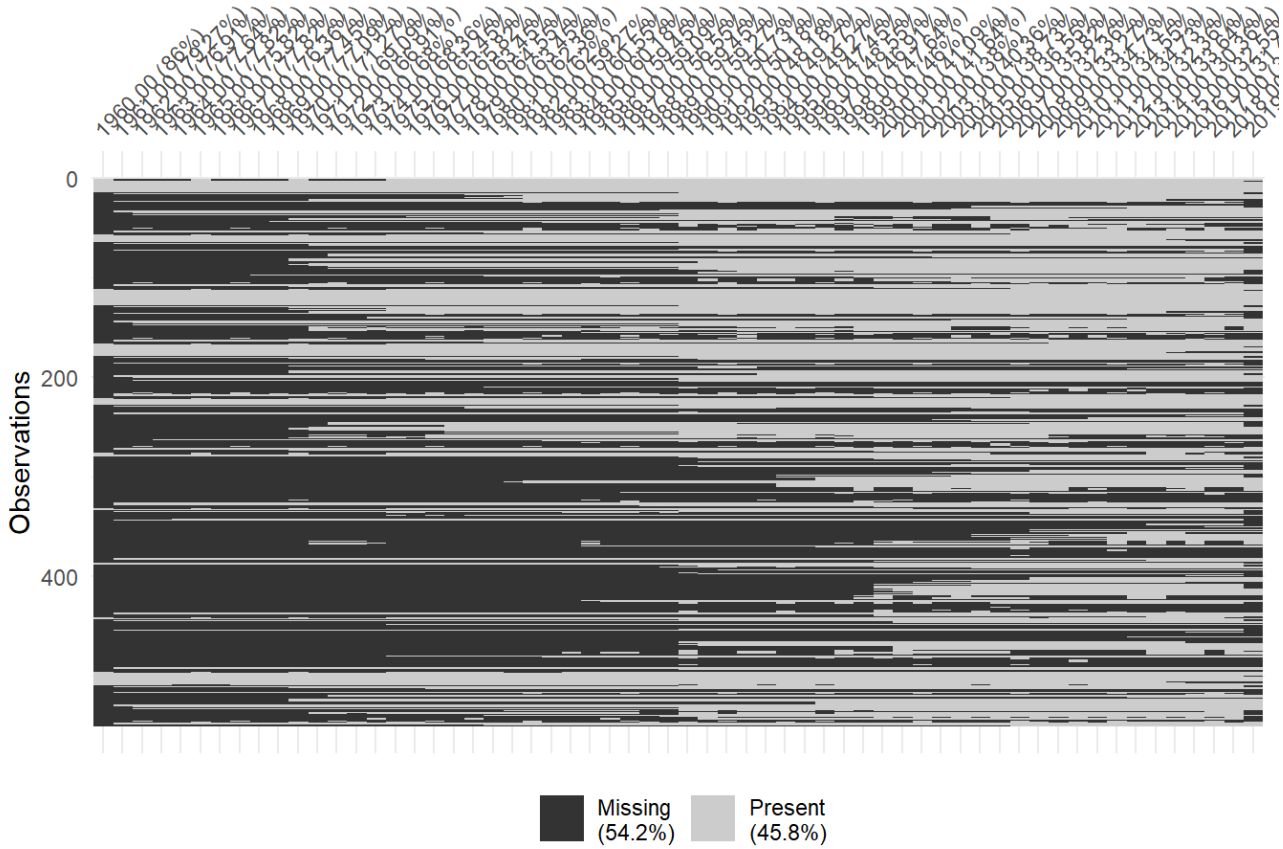
```
## # A tibble: 550 x 64
##    `Country Name` `Country Code` `Series Name` `Series Code` `1960.00` `1961.00`
##    <chr>          <chr>          <chr>         <chr>             <dbl>     <dbl>
##  1 China          CHN            Mobile cellu~ IT.CEL.SETS.~      0        NA
##  2 China          CHN            CO2 emission~ EN.ATM.CO2E.~      1.17      0.84
##  3 China          CHN            Exports of g~ NE.EXP.GNFS.~      4.31      3.87
##  4 China          CHN            Imports of g~ NE.IMP.GNFS.~      4.43      3.49
##  5 China          CHN            Fertility ra~ SP.DYN.TFRT.~      5.76      5.91
##  6 China          CHN            Merchandise ~ TG.VAL.TOTL.~      8.74      7.37
##  7 China          CHN            Agriculture,~ NV.AGR.TOTL.~     23.2      35.8
##  8 China          CHN            Gross capita~ NE.GDI.TOTL.~     39.6      22.8
##  9 China          CHN            Life expecta~ SP.DYN.LE00.~     43.7      44.0
## 10 China          CHN            Industry (in~ NV.IND.TOTL.~     44.4      31.9
## # ... with 540 more rows, and 58 more variables: 1962.00 <dbl>, 1963.00 <dbl>,
## #   1964.00 <dbl>, 1965.00 <dbl>, 1966.00 <dbl>, 1967.00 <dbl>, 1968.00 <dbl>,
## #   1969.00 <dbl>, 1970.00 <dbl>, 1971.00 <dbl>, 1972.00 <dbl>, 1973.00 <dbl>,
## #   1974.00 <dbl>, 1975.00 <dbl>, 1976.00 <dbl>, 1977.00 <dbl>, 1978.00 <dbl>,
## #   1979.00 <dbl>, 1980.00 <dbl>, 1981.00 <dbl>, 1982.00 <dbl>, 1983.00 <dbl>,
## #   1984.00 <dbl>, 1985.00 <dbl>, 1986.00 <dbl>, 1987.00 <dbl>, 1988.00 <dbl>,
## #   1989.00 <dbl>, 1990.00 <dbl>, 1991.00 <dbl>, 1992.00 <dbl>, ...
```

Q4

1. count how many columns contain missing values

2. list each of these columns with the corresponding missingess percentages

```
#visualize each of year columns with the corresponding missing percentages
vis_miss(filtered_ds_worldDI[,c(-1,-2,-3,-4)])
```




```
#convert the table from wide to long. group by year and value
year_value <- filtered_ds_worldDI %>% gather(Year, Value, -"Country Name", -"Country Code", -"Series
 Name",  -"Series Code")

#print into a table about how much missing value for each year in percentage
year_value %>%
  group_by(Year) %>%
  summarise(Percentage = mean(is.na(Value)*100), Na_rows= is.na(Value) %>% sum(), total_rows = Value
 %>% length())
```

```
## # A tibble: 60 x 4
##    Year    Percentage Na_rows total_rows
##    <chr>        <dbl>   <int>      <int>
##  1 1960.00       86       473        550
##  2 1961.00       79.3     436        550
##  3 1962.00       76.9     423        550
##  4 1963.00       77.6     427        550
##  5 1964.00       77.8     428        550
##  6 1965.00       75.8     417        550
##  7 1966.00       77.8     428        550
##  8 1967.00       76.4     420        550
##  9 1968.00       77.5     426        550
## 10 1969.00       77.1     424        550
## # ... with 50 more rows
```

#Q5 ## 1. display statistical infromation. min max and mean of the gdp from 2010 - 2019

```
#statistical data only for min and max and mean
year_value %>%
  group_by(Year) %>%
  summarise(Min =Value %>% min(na.rm = T), Max = Value %>% max(na.rm=T), Mean = Value %>%  mean(na.rm
=T)) %>%
  filter(Year == c("2010.00","2011.00","2012.00","2013.00","2014.00","2015.00","2016.00","2017.00","2
018.00","2019.00"))
```

```
## # A tibble: 10 x 4
##    Year            Min     Max          Mean
##    <chr>         <dbl>   <dbl>         <dbl>
##  1 2010.00   -9356673. 1.52e13 252134097548.
##  2 2011.00 -850717035. 1.58e13 318359814084.
##  3 2012.00 -180529999. 1.67e13 326319472077.
##  4 2013.00 -656549988. 1.72e13 346474574019.
##  5 2014.00 -947070007. 1.81e13 345086048560.
##  6 2015.00 -306290008. 1.87e13 362886246482.
##  7 2016.00 -791429993. 1.91e13 368060134705.
##  8 2017.00 -989940002. 2   e13 393349428082.
##  9 2018.00 -705450012. 2.16e13 442713503162.
## 10 2019.00 -589979980. 2.34e13 594736978737.
```

```
#all statistical from 2010 - 2019
summary(filtered_ds_worldDI[55:length(colnames(filtered_ds_worldDI))])
```

```
##     2010.00              2011.00              2012.00
##  Min.   :-9.357e+06   Min.   :-8.507e+08   Min.   :-1.805e+08
##  1st Qu.: 9.000e+00   1st Qu.: 9.000e+00   1st Qu.: 9.000e+00
##  Median : 4.900e+01   Median : 5.600e+01   Median : 6.000e+01
##  Mean   : 2.521e+11   Mean   : 3.184e+11   Mean   : 3.263e+11
##  3rd Qu.: 4.355e+03   3rd Qu.: 4.200e+03   3rd Qu.: 6.972e+03
##  Max.   : 1.520e+13   Max.   : 1.580e+13   Max.   : 1.670e+13
##  NA's   :190          NA's   :191          NA's   :178
##     2013.00              2014.00              2015.00
##  Min.   :-6.565e+08   Min.   :-9.471e+08   Min.   :-3.063e+08
##  1st Qu.: 9.000e+00   1st Qu.: 1.000e+01   1st Qu.: 8.000e+00
##  Median : 5.900e+01   Median : 6.700e+01   Median : 4.700e+01
##  Mean   : 3.465e+11   Mean   : 3.451e+11   Mean   : 3.629e+11
##  3rd Qu.: 6.906e+03   3rd Qu.: 4.345e+03   3rd Qu.: 4.295e+03
##  Max.   : 1.720e+13   Max.   : 1.810e+13   Max.   : 1.870e+13
##  NA's   :185          NA's   :167          NA's   :179
##     2016.00              2017.00              2018.00
##  Min.   :-7.914e+08   Min.   :-9.899e+08   Min.   :-7.055e+08
##  1st Qu.: 8.000e+00   1st Qu.: 1.000e+01   1st Qu.: 8.000e+00
##  Median : 5.300e+01   Median : 4.200e+01   Median : 4.500e+01
##  Mean   : 3.681e+11   Mean   : 3.933e+11   Mean   : 4.427e+11
##  3rd Qu.: 2.920e+03   3rd Qu.: 5.708e+03   3rd Qu.: 6.340e+03
##  Max.   : 1.910e+13   Max.   : 2.000e+13   Max.   : 2.160e+13
##  NA's   :173          NA's   :178          NA's   :195
##     2019.00
##  Min.   :-5.900e+08
##  1st Qu.: 8.000e+00
##  Median : 5.100e+01
##  Mean   : 5.947e+11
##  3rd Qu.: 1.984e+04
##  Max.   : 2.340e+13
##  NA's   :274
```

Q6

1.get all china value from the year_value table

2.print max gdp for china

3.print min gdp for china

```
#select china and gdp rows
China_df<-year_value %>%
  filter(`Country Name` == "China") %>%
  filter(`Series Name` == "GDP (current US$)")

print("max gdp:")
```

```
## [1] "max gdp:"
```

```
China_df$Value %>% max()
```

```
## [1] 1.43e+13
```

```
print("min gdp:")
```

```
## [1] "min gdp:"
```

```
China_df$Value %>% min()
```

```
## [1] 47209359006
```

Q7

1.filter all GDP information for 10 selected countries (top5 and bot5).

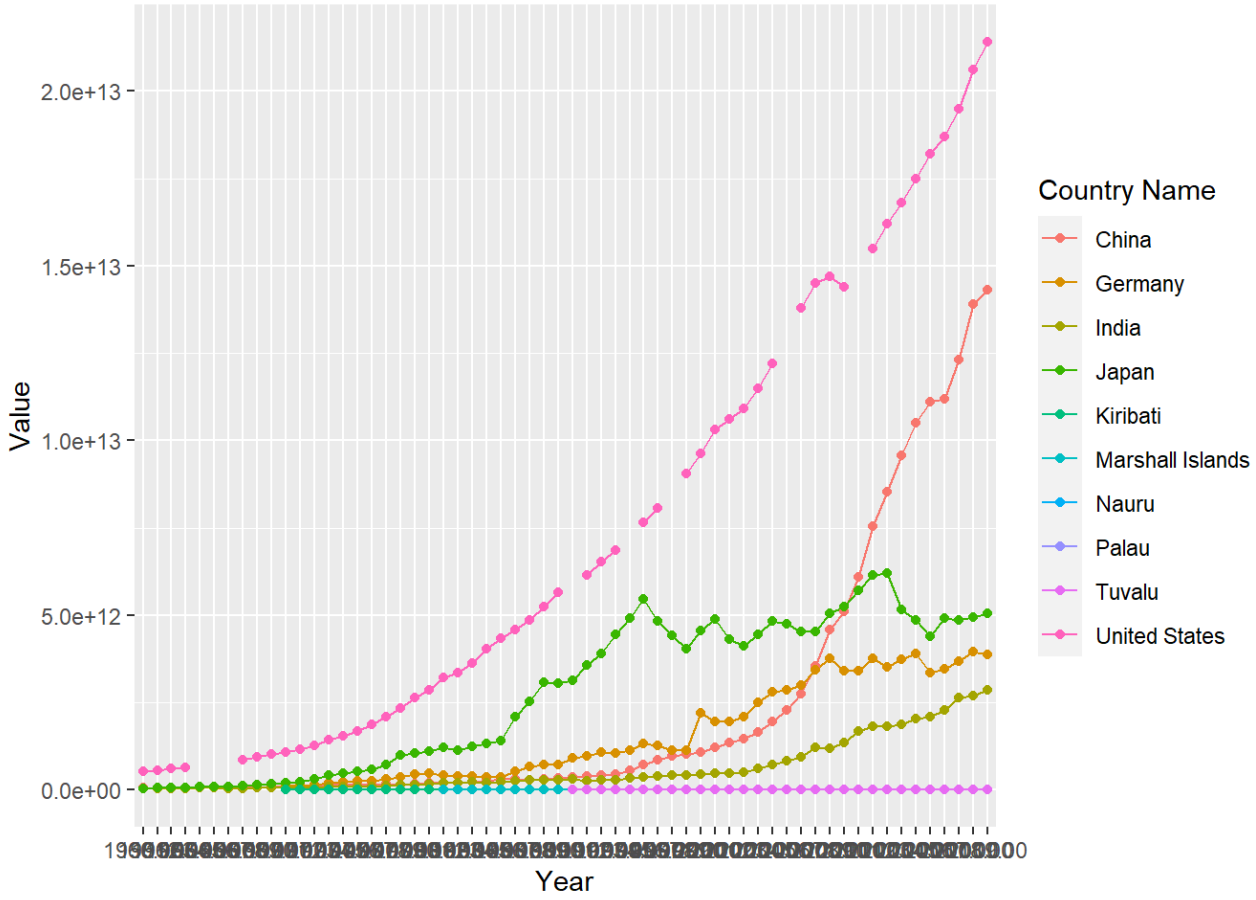
2.plot GDP for 10 selected countries over all years.

3.answer question what is one advantage and one idsadvantage of including all these countries in one plot

```
#filter to get GDP (current US$) series for each Country in each year
ten_gdp <- year_value %>%
  filter(`Series Name` == "GDP (current US$)")

#plot all ten countries in one graph
ten_gdp %>%
  ggplot(aes(Year, Value, group=`Country Name`, color = `Country Name`))+
  geom_line() +
  geom_point()
```

```
## Warning: Removed 161 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 169 rows containing missing values (geom_point).
```




```
#the advantage is that it is easier to compare each country in one graph
#the disadvantage is that the top 5 and bot 5 countries has dramastic different of GDP. From the grap
h, you can see the bot countries are stacked into one line.
```

Q8

1.create a new column and name it as Period

2.assign Period factors

3.create a table that shows mean of each period for each country

4.plot graph for each country

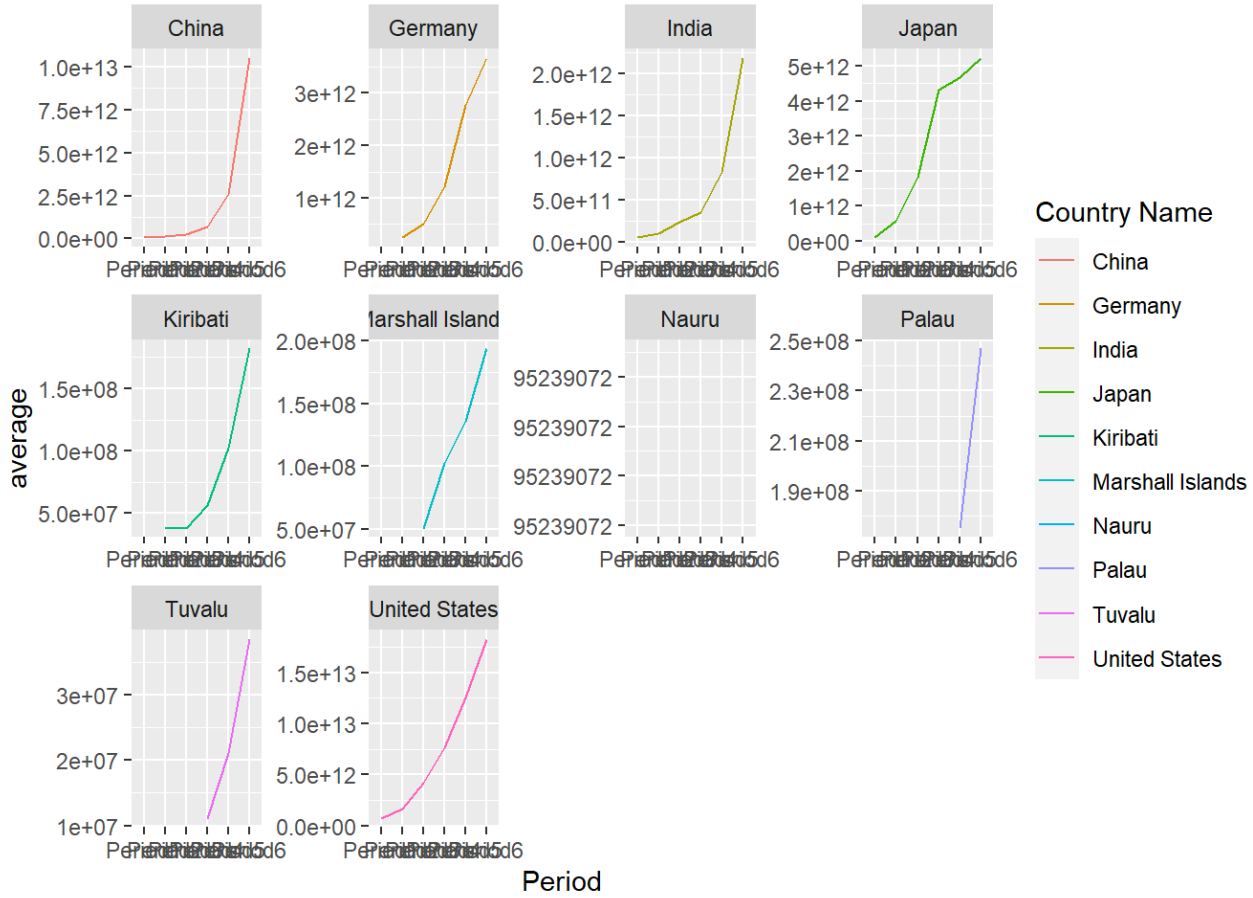
```
#add a Period columns
ten_gdp$Period<-''

#label Periods
ten_gdp$Period[ten_gdp$Year <1970 &ten_gdp$Year >=1960]<- "Period1"
ten_gdp$Period[ten_gdp$Year <1980 &ten_gdp$Year >=1970]<- "Period2"
ten_gdp$Period[ten_gdp$Year <1990 &ten_gdp$Year >=1980]<- "Period3"
ten_gdp$Period[ten_gdp$Year <2000 &ten_gdp$Year >=1990]<- "Period4"
ten_gdp$Period[ten_gdp$Year <2010 &ten_gdp$Year >=2000]<- "Period5"
ten_gdp$Period[ten_gdp$Year <2020 &ten_gdp$Year >=2010]<- "Period6"

#create a table that shows mean of each period for each country
average_period <- ten_gdp %>% group_by(`Country Name`, Period) %>% summarise(average = Value %>% mean
(na.rm = T), .groups = 'drop')
average_period
```

```
## # A tibble: 60 x 3
##    `Country Name` Period    average
##    <chr>          <chr>       <dbl>
##  1 China          Period1   6.38e10
##  2 China          Period2   1.41e11
##  3 China          Period3   2.63e11
##  4 China          Period4   6.86e11
##  5 China          Period5   2.59e12
##  6 China          Period6   1.05e13
##  7 Germany        Period1 NaN
##  8 Germany        Period2   2.48e11
##  9 Germany        Period3   5.06e11
## 10 Germany        Period4   1.22e12
## # ... with 50 more rows
```

```
#plot graph for each country
average_period %>%
  ggplot(aes(Period, average,color=`Country Name`,group=`Country Name`))+
  geom_line()+
  facet_wrap(~`Country Name`,scale="free")
```

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```



Q9

1.subset America with GDP

2.swap wide table to long table

3.assign mean value to all NA

```
#subset America with GDP
America <- filtered_ds_worldDI %>% filter(`Country Name` == "United States") %>% filter(`Series Name`
== "GDP (current US$)")

#swap wide table to long table
America <- America %>% gather(Year,Value, -"Country Name", -"Country Code", -"Series Name", -"Series
 Code")

#assign mean value to NA
America$Value[is.na(America$Value)] <- America$Value %>% mean(na.rm=T)

America
```

```
## # A tibble: 60 x 6
##    `Country Name` `Country Code` `Series Name`     `Series Code`  Year      Value
##    <chr>          <chr>          <chr>             <chr>          <chr>     <dbl>
##  1 United States  USA            GDP (current US$) NY.GDP.MKTP.CD 1960.00 5.43e11
##  2 United States  USA            GDP (current US$) NY.GDP.MKTP.CD 1961.00 5.63e11
##  3 United States  USA            GDP (current US$) NY.GDP.MKTP.CD 1962.00 6.05e11
##  4 United States  USA            GDP (current US$) NY.GDP.MKTP.CD 1963.00 6.39e11
##  5 United States  USA            GDP (current US$) NY.GDP.MKTP.CD 1964.00 7.60e12
##  6 United States  USA            GDP (current US$) NY.GDP.MKTP.CD 1965.00 7.60e12
##  7 United States  USA            GDP (current US$) NY.GDP.MKTP.CD 1966.00 7.60e12
##  8 United States  USA            GDP (current US$) NY.GDP.MKTP.CD 1967.00 8.62e11
##  9 United States  USA            GDP (current US$) NY.GDP.MKTP.CD 1968.00 9.43e11
## 10 United States  USA            GDP (current US$) NY.GDP.MKTP.CD 1969.00 1.02e12
## # ... with 50 more rows
```