Project2

Seung Min Song & Ted Kim

2022-10-04

GitHub: https://github.com/seung-m1nsong/607
Rpub:https://rpubs.com/seungm1nsong/953863

1. Joyce Aldrich's Real GDP by County Analysis: Comparing all industries GDP grow rate for each county

Read csv file into a data frame **real_dgp_by_county**. The first three rows are headers, therefore set the header option to *False*.

```
real_gdp_by_county <- read.csv('./gdp_by_county.csv', header = FALSE, sep = ',')
head(real_gdp_by_county)
```

```
##      V1          V2     V3       V4                                      V5
## 1  FIPS Countyname Postal LineCode                           IndustryName
## 2
## 3
## 4 01001    Autauga     AL        1                         All Industries
## 5 01001    Autauga     AL        2     Private goods-producing industries
## 6 01001    Autauga     AL        3  Private services-providing industries
##                                         V6      V7      V8      V9
## 1 Real Gross domestic product (GDP) by county
## 2         (thousands of chained 2012 dollars)
## 3                                       2012    2013    2014    2015
## 4                                    1383941 1322416 1312668 1412939
## 5                                     286396  299115  310672  325250
## 6                                     948490  880098  861153  946148
```

Reset the header in the data frame. Use first row value for column one trough five and use third row value for column six through nine.

```
header <- real_gdp_by_county[1, ] %>%
            select(c(1:5))
unlist(c(header, real_gdp_by_county[3, ] %>%
                            select(c(6:9))))
```

```
##             V1             V2             V3             V4             V5
##         "FIPS"   "Countyname"       "Postal"     "LineCode" "IndustryName"
##             V6             V7             V8             V9
##         "2012"         "2013"         "2014"         "2015"
```

```
header <- unlist(c(header, real_gdp_by_county[3, ] %>%
                            select(c(6:9))))
colnames(real_gdp_by_county) = header
head(real_gdp_by_county)
```

```
##    FIPS Countyname Postal LineCode                            IndustryName
## 1  FIPS Countyname Postal LineCode                            IndustryName
## 2
## 3
## 4 01001    Autauga     AL        1                           All Industries
## 5 01001    Autauga     AL        2      Private goods-producing industries
## 6 01001    Autauga     AL        3   Private services-providing industries
##                                        2012    2013    2014    2015
## 1 Real Gross domestic product (GDP) by county
## 2         (thousands of chained 2012 dollars)
## 3                                      2012    2013    2014    2015
## 4                                   1383941 1322416 1312668 1412939
## 5                                    286396  299115  310672  325250
## 6                                    948490  880098  861153  946148
```

Use *slice( )* function to remove unnecessary rows from the data frame. Remove rows one through three which overlap the header and the last four rows which is not relevant to current data.

```
real_gdp_by_county <- real_gdp_by_county %>%
                        slice(4:(n() - 4))

head(real_gdp_by_county)
```

```
##    FIPS Countyname Postal LineCode                          IndustryName
## 1 01001    Autauga     AL        1                         All Industries
## 2 01001    Autauga     AL        2    Private goods-producing industries
## 3 01001    Autauga     AL        3 Private services-providing industries
## 4 01001    Autauga     AL        4 Government and government enterprises
## 5 01003    Baldwin     AL        1                         All Industries
## 6 01003    Baldwin     AL        2    Private goods-producing industries
##      2012    2013    2014    2015
## 1 1383941 1322416 1312668 1412939
## 2  286396  299115  310672  325250
## 3  948490  880098  861153  946148
## 4  149055  143062  140893  141294
## 5 5599194 6218819 6247887 5981958
## 6  681871  675300  667273  681451
```

```
tail(real_gdp_by_county)
```

```
##        FIPS Countyname Postal LineCode                          IndustryName
## 12447 56043   Washakie     WY        3 Private services-providing industries
## 12448 56043   Washakie     WY        4 Government and government enterprises
## 12449 56045     Weston     WY        1                         All Industries
## 12450 56045     Weston     WY        2    Private goods-producing industries
## 12451 56045     Weston     WY        3 Private services-providing industries
## 12452 56045     Weston     WY        4 Government and government enterprises
##         2012   2013   2014   2015
## 12447 167938 179501 192289 195478
## 12448  62263  62283  61337  59740
## 12449 332472 311082 317811 388824
## 12450 181482 158661 162602 239734
## 12451  96240  95353  99420 104625
## 12452  54750  56535  55338  53722
```

Normalize data frame vy three data frame, *df_county*, *df_industry_name*, and *df_real_gdf_wider*. Remove duplicated row and use *unique()* function to make it unique.

```
df_county <- unique(real_gdp_by_county %>%
                      select(c(1:3))) #FIPS, CountyName, Postal
head(df_county)
```

```
##     FIPS Countyname Postal
## 1  01001    Autauga     AL
## 5  01003    Baldwin     AL
## 9  01005    Barbour     AL
## 13 01007       Bibb     AL
## 17 01009     Blount     AL
## 21 01011    Bullock     AL
```

```
df_industry_name <- unique(real_gdp_by_county %>%
                      select(c(4:5))) #LineCode, IndustryName
head(df_industry_name)
```

```
##   LineCode                            IndustryName
## 1        1                           All Industries
## 2        2     Private goods-producing industries
## 3        3   Private services-providing industries
## 4        4   Government and government enterprises
```

```
df_real_gdf_wider <- real_gdp_by_county %>%
                        select(c(1, 4, 6:9)) #FIPS, LineCode, '2012', '2013', '2014', '2015'
head(df_real_gdf_wider)
```

```
##    FIPS LineCode    2012    2013    2014    2015
## 1 01001        1 1383941 1322416 1312668 1412939
## 2 01001        2  286396  299115  310672  325250
## 3 01001        3  948490  880098  861153  946148
## 4 01001        4  149055  143062  140893  141294
## 5 01003        1 5599194 6218819 6247887 5981958
## 6 01003        2  681871  675300  667273  681451
```

Transfer into a tidy data frame. After that, Use *pivot_longer()* function to pivot from column *2012* to last column *2015*. Did not use *values_drop_na = TRUE* since there are no cells with NA value. Change all values with *D* to *0* after the pivot.

(D) Not shown to avoid disclosure of confidential information, but the estimates for this item are included in the totals

```
df_real_gdf_longer <- df_real_gdf_wider %>%
                        pivot_longer(
                          cols = colnames(df_real_gdf_wider)[-c(1,2)],
                          names_to = 'Year',
                          values_to = 'RealGDP'
                        ) %>%
                        mutate(
                          RealGDP = ifelse(RealGDP == '(D)', 0, RealGDP)
                        )
head(df_real_gdf_longer)
```

```
## # A tibble: 6 x 4
##   FIPS  LineCode Year  RealGDP
##   <chr> <chr>    <chr> <chr>
## 1 01001 1        2012  1383941
## 2 01001 1        2013  1322416
## 3 01001 1        2014  1312668
## 4 01001 1        2015  1412939
## 5 01001 2        2012  286396
## 6 01001 2        2013  299115
```

```
tail(df_real_gdf_longer)
```

```
## # A tibble: 6 x 4
##   FIPS  LineCode Year  RealGDP
##   <chr> <chr>    <chr> <chr>
## 1 56045 3        2014  99420
## 2 56045 3        2015  104625
```

```
## 3 56045 4        2012  54750
## 4 56045 4        2013  56535
## 5 56045 4        2014  55338
## 6 56045 4        2015  53722
```

Convert *Year* and *RealGDP* data type; Character to double.

```
df_real_gdf_longer$Year = as.numeric(df_real_gdf_longer$Year)
df_real_gdf_longer$RealGDP = as.numeric(df_real_gdf_longer$RealGDP)
```

Use *merge( )* to match *FIPS* and *Countyname*. Calculate *LineCode 1* (all industries) growth rate for each county. Replace *NA* value with *0*.

```
df_real_gdf_longer <- merge(x = df_real_gdf_longer, y = df_county, by = 'FIPS') %>%
  group_by(FIPS, LineCode) %>%
    mutate(
      Prev_GDP = ifelse(is.na(lag(RealGDP)), 0, lag(RealGDP)),
      Diff_growth = ifelse(is.na(lag(RealGDP)), 0, RealGDP - lag(RealGDP)),
      Rate_growth = ifelse(is.na(lag(RealGDP)), 0, round((RealGDP / lag(RealGDP) - 1) * 100, digits = 2)
      )
df_real_gdf_longer
```

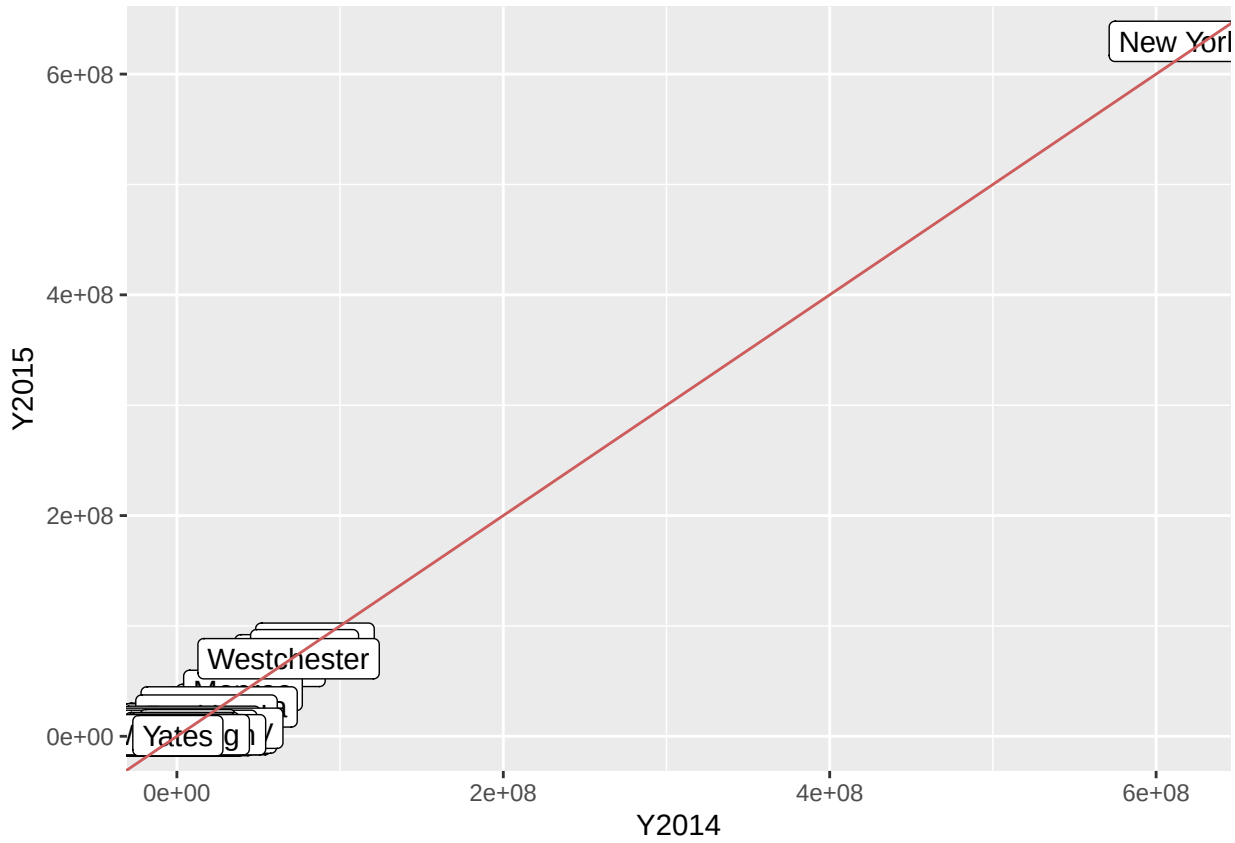
```
## # A tibble: 49,808 x 9
## # Groups:   FIPS, LineCode [12,452]
##    FIPS  LineCode  Year RealGDP Countyname Postal Prev_GDP Diff_growth Rate_gr~1
##    <chr> <chr>    <dbl>   <dbl> <chr>      <chr>     <dbl>       <dbl>     <dbl>
##  1 01001 1         2012 1383941 Autauga    AL            0           0      0
##  2 01001 1         2013 1322416 Autauga    AL      1383941      -61525     -4.45
##  3 01001 1         2014 1312668 Autauga    AL      1322416       -9748     -0.74
##  4 01001 1         2015 1412939 Autauga    AL      1312668      100271      7.64
##  5 01001 2         2012  286396 Autauga    AL            0           0      0
##  6 01001 2         2013  299115 Autauga    AL       286396       12719      4.44
##  7 01001 2         2014  310672 Autauga    AL       299115       11557      3.86
##  8 01001 2         2015  325250 Autauga    AL       310672       14578      4.69
##  9 01001 3         2012  948490 Autauga    AL            0           0      0
```

```
## 10 01001 3         2013  880098 Autauga    AL       948490      -68392     -7.21
## # ... with 49,798 more rows, and abbreviated variable name 1: Rate_growth
```

Use *ggplot() + geom_point + geom_label* to compare 2014 and 2015's all industries' summarized GDP growth rate. *geom_abline* displays the red line in the graph and any blue dots above the red line indicate a positive growth rate and blue dots below the red line indicate a negative growth rate. Most of the county's GDP increased compared to 2014. Most of the county falls in between 0e + 00 and 2e+08.

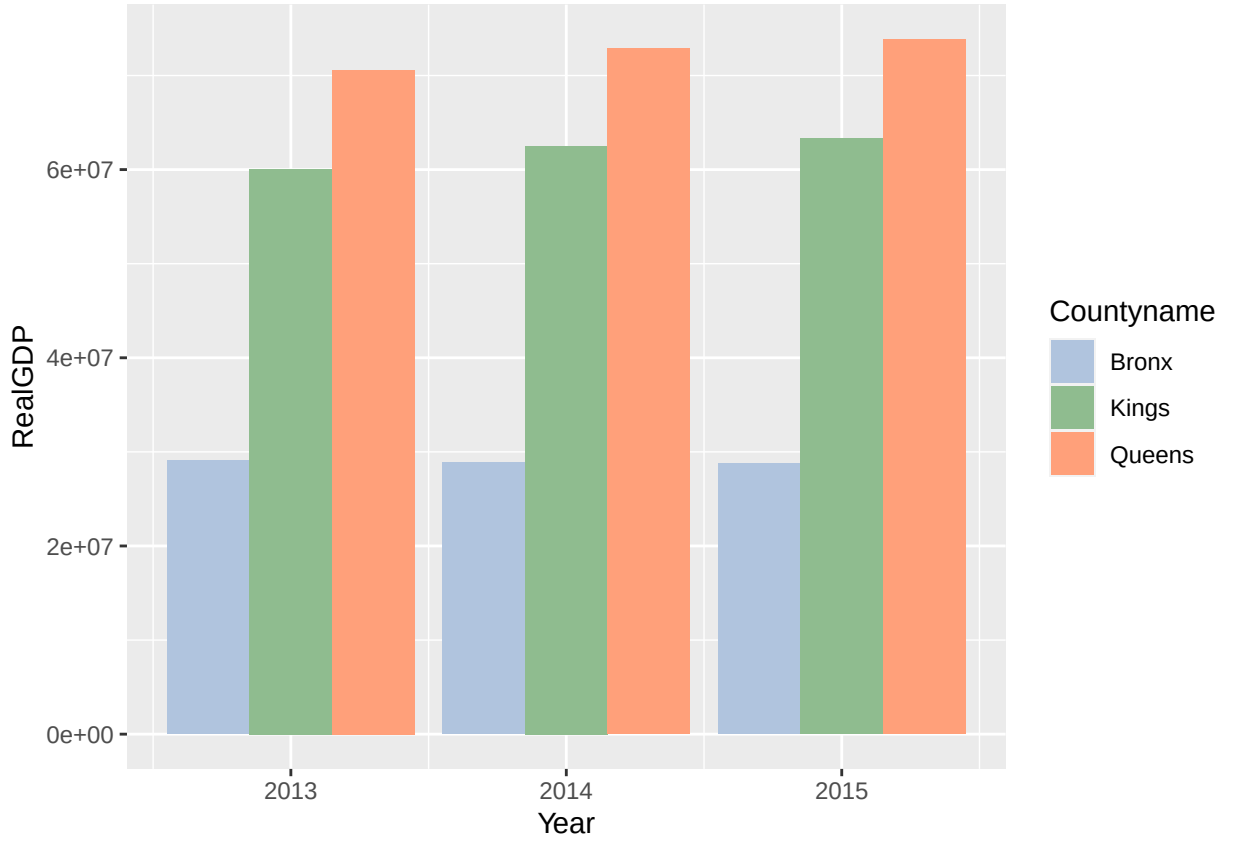
```
df_real_gdf_longer %>%
  filter(LineCode == '1', Postal == 'NY', Year == 2015) %>%
  ggplot(aes(x = Prev_GDP, y = RealGDP, label = Countyname), color = 'SteelBlue') +
  geom_point() +
  geom_label(
    nudge_x = 0.25, nudge_y = 0.25,
    check_overlap = T
  )+
  geom_abline(intercept = 0, slope = 1, size = 0.5, color = 'IndianRed') +
  labs(
    x = 'Y2014',
    y = 'Y2015'
  )
```

```
## Warning: Ignoring unknown parameters: check_overlap
```



Use 2013, 2014, and 2015 Queens, Brooklyn, and the Bronx's all industry's GDP growth rate and compare. Do not have to use 2012 because it has nothing to compare with.

```
counties <- c('Bronx', 'Queens', 'Kings')
df_real_gdf_longer %>%
  filter(LineCode == '1', Year != 2012, Postal == 'NY', Countyname %in% counties) %>%
  ggplot(aes(x = Year, y = RealGDP, fill = Countyname)) +
    geom_bar(stat = 'identity', position = 'dodge') +
      scale_color_manual(values = c('SteelBlue', 'OliveDrab', 'Coral')) +
        scale_fill_manual(values = c('LightSteelBlue', 'DarkSeaGreen', 'LightSalmon'))
```



2. Jawaid Hakim's Pharmaceutical Drug Spending by Countries Analysis: An interesting analysis would be to plot the growth in spend by country over time, and comparison of growth in spend between countries.

Read csv file into a data frame **drug_spend_wider**. Only the first row is head, therefore set the header option to *True*.

```
drug_spend_wider <- read.csv('./drug_spending.csv', header = TRUE, sep = ',')
head(drug_spend_wider)
```

```
##   LOCATION TIME PC_HEALTHXP PC_GDP USD_CAP FLAG_CODES TOTAL_SPEND
## 1      AUS 1971      15.992  0.727  35.720                462.11
## 2      AUS 1972      15.091  0.686  36.056                475.11
## 3      AUS 1973      15.117  0.681  39.871                533.47
## 4      AUS 1974      14.771  0.755  47.559                652.65
## 5      AUS 1975      11.849  0.682  47.561                660.76
## 6      AUS 1976      10.920  0.630  46.908                658.26
```

Transfer into a tidy data frame. Remove column 6 named FLAG_CODES in *drug_spend_wider* data frame. Use *pivot_longer()* function to pivot from column *PC_HEALTHXP* to last column *TOTAL SPEND*. Did not use *values_drop_na = TRUE* since there are no cells with NA value.

```
drug_spend_longer <- drug_spend_wider %>%
                         select(-c(6)) %>%
                         pivot_longer(
                           cols = colnames(drug_spend_wider)[-c(1,2, 6)],
                           names_to = 'Measure',
                           values_to = 'Value'
                         )
head(drug_spend_longer)
```

```
## # A tibble: 6 x 4
##   LOCATION  TIME Measure       Value
##   <chr>    <int> <chr>         <dbl>
## 1 AUS       1971 PC_HEALTHXP  16.0
## 2 AUS       1971 PC_GDP        0.727
## 3 AUS       1971 USD_CAP      35.7
## 4 AUS       1971 TOTAL_SPEND 462.
## 5 AUS       1972 PC_HEALTHXP  15.1
## 6 AUS       1972 PC_GDP        0.686
```

```
tail(drug_spend_longer)
```

```
## # A tibble: 6 x 4
##   LOCATION  TIME Measure       Value
##   <chr>    <int> <chr>         <dbl>
## 1 SVN       2011 USD_CAP      483.
## 2 SVN       2011 TOTAL_SPEND  991.
## 3 SVN       2012 PC_HEALTHXP   20.2
## 4 SVN       2012 PC_GDP         1.77
## 5 SVN       2012 USD_CAP      510.
## 6 SVN       2012 TOTAL_SPEND 1048.
```

Calculate the total spending rate for each country. Compare with data from 10 years ago.

```
drug_spend_longer <- drug_spend_longer %>%
  group_by(LOCATION) %>%
  filter(Measure =='TOTAL_SPEND')  %>%
    mutate(
      Prev_spend = ifelse(is.na(lag(Value, n = 10)), 0, lag(Value, n = 10)),
      Diff_spend = Value - lag(Value),
      Rate_percent = round((Value / lag(Value) - 1) * 100, digits = 2)
      )
drug_spend_longer
```

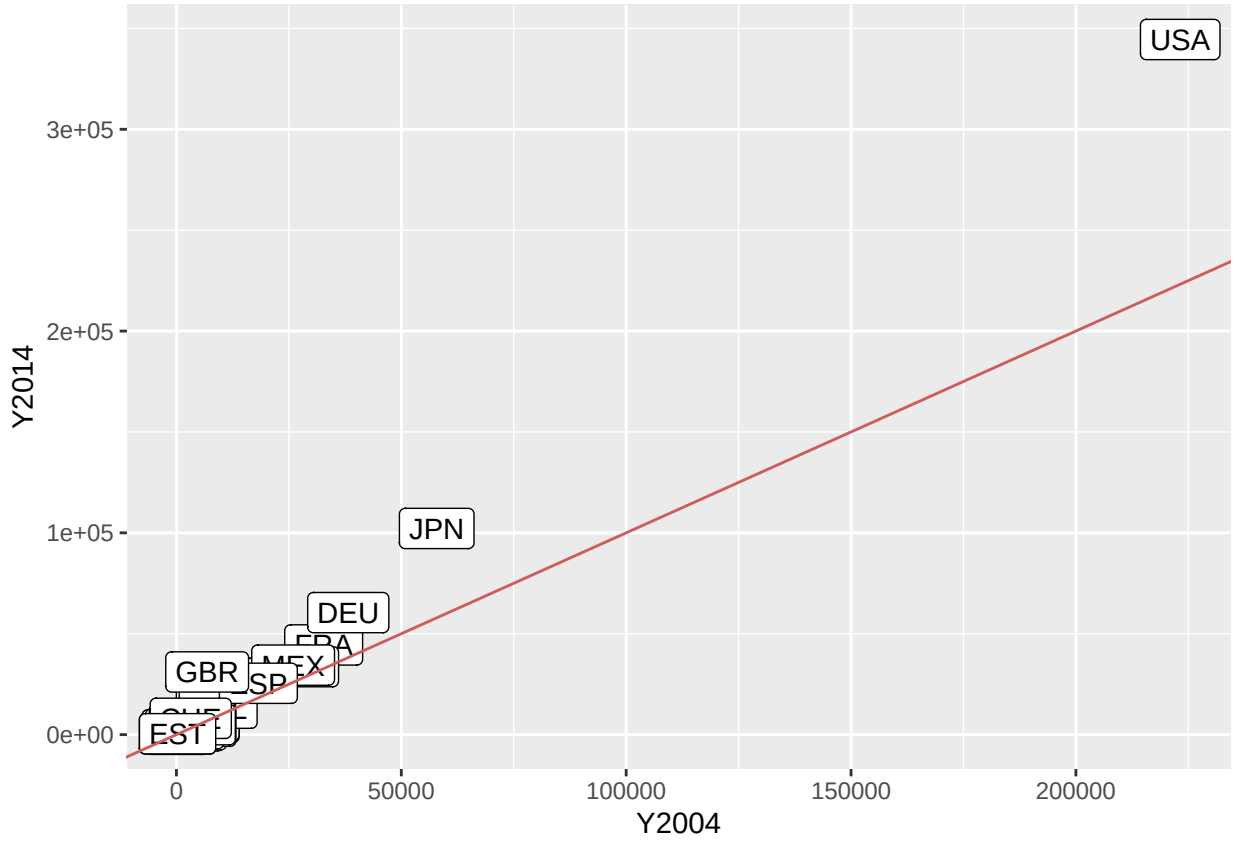
```
## # A tibble: 1,000 x 7
## # Groups:   LOCATION [32]
##    LOCATION  TIME Measure     Value Prev_spend Diff_spend Rate_percent
##    <chr>    <int> <chr>       <dbl>      <dbl>      <dbl>        <dbl>
##  1 AUS       1971 TOTAL_SPEND  462.          0      NA           NA
##  2 AUS       1972 TOTAL_SPEND  475.          0      13            2.81
##  3 AUS       1973 TOTAL_SPEND  533.          0      58.4         12.3
##  4 AUS       1974 TOTAL_SPEND  653.          0     119.          22.3
##  5 AUS       1975 TOTAL_SPEND  661.          0       8.11         1.24
##  6 AUS       1976 TOTAL_SPEND  658.          0      -2.5         -0.38
##  7 AUS       1977 TOTAL_SPEND  676.          0      18.0          2.73
##  8 AUS       1978 TOTAL_SPEND  729.          0      53.1          7.86
##  9 AUS       1979 TOTAL_SPEND  722.          0      -7.07        -0.97
## 10 AUS       1980 TOTAL_SPEND  837.          0     115.          15.9
## # ... with 990 more rows
```

Use *ggplot() + geom_point* to Compare 2004 and 2014. *geom_abline* displays the red line in the graph. Any blue dots above the red line indicate a positive spending rate and blue dots below the red line indicate a negative spending rate. Most of the nation spent more in 2014 compare to 2004.

```
drug_spend_longer %>%
  filter(TIME == 2014) %>%
  ggplot(aes(x = Prev_spend, y = Value, label = LOCATION), color = 'SteelBlue') +
```

```
  geom_point() +
  geom_label(
    nudge_x = 0.25, nudge_y = 0.25,
    check_overlap = T
  ) +
  geom_abline(intercept = 0, slope = 1, size = 0.5, color = 'IndianRed') +
  labs(
    x = 'Y2004',
    y = 'Y2014'
  )
```

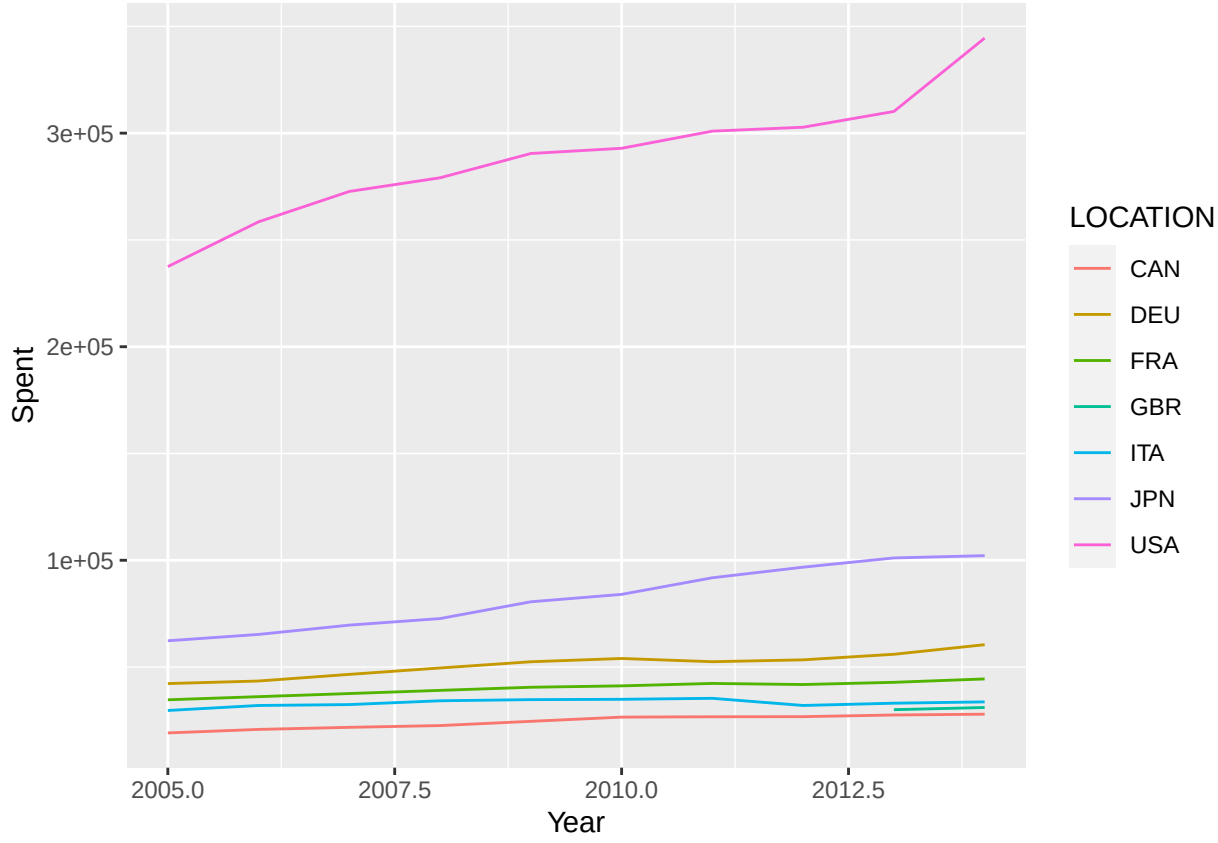
```
## Warning: Ignoring unknown parameters: check_overlap
```



Compare drug spending for G7 country in a 10-year cycle. Overall, drug spending in the US is significantly higher than other g7 countries, and drug spending in g7 countries has also steadily increased.

```
#DEU: Germany, GBR: the United Kingdom
G7 <- c('CAN', 'FRA', 'DEU', 'ITA', 'JPN', 'GBR', 'USA')
```

```
drug_spend_longer %>%
  filter(LOCATION %in% G7, TIME > 2004, TIME < 2015) %>%
  ggplot(aes(x = TIME, y = Value, color = LOCATION)) +
    geom_line() +
    labs(
      x = 'Year',
      y = 'Spent'
    )
```



3. Benjamin Inbar's Global GDP Dataset 1960-2021 Analysis: One interesting analysis would be to get the % change year on year, per country, or per region.

Read csv file into a data frame **global_gdp**. Only the first row is head, therefore set the header option to *True*.

```r
global_gdp <- read.csv('./global_gdp_1960_2021.csv', header = TRUE, sep = ',')
head(global_gdp)
```

```
##                   Country.Name Country.Code    Indicator.Name Indicator.Code
## 1                        Aruba          ABW GDP (current US$) NY.GDP.MKTP.CD
## 2 Africa Eastern and Southern          AFE GDP (current US$) NY.GDP.MKTP.CD
## 3                  Afghanistan          AFG GDP (current US$) NY.GDP.MKTP.CD
## 4  Africa Western and Central          AFW GDP (current US$) NY.GDP.MKTP.CD
## 5                       Angola          AGO GDP (current US$) NY.GDP.MKTP.CD
## 6                      Albania          ALB GDP (current US$) NY.GDP.MKTP.CD
##         X1960       X1961       X1962       X1963       X1964       X1965
## 1          NA          NA          NA          NA          NA          NA
## 2 21290586003 21808473825 23707015394 28210036878 26118787467 29682172751
## 3   537777811   548888896   546666678   751111191   800000044  1006666638
## 4 10404135069 11127894641 11943187848 12676330765 13838369295 14862225760
## 5          NA          NA          NA          NA          NA          NA
## 6          NA          NA          NA          NA          NA          NA
##         X1966       X1967       X1968       X1969       X1970       X1971
## 1          NA          NA          NA          NA          NA          NA
## 2 32239121547 33514552047 36521482937 41828336213 44862605393 49478916698
## 3  1399999967  1673333418  1373333367  1408888922  1748886596  1831108971
## 4 15832591204 14426038230 14880349280 16882092549 23504605476 20832817218
## 5          NA          NA          NA          NA          NA          NA
## 6          NA          NA          NA          NA          NA          NA
##         X1972       X1973       X1974       X1975       X1976        X1977
## 1          NA          NA          NA          NA          NA           NA
## 2 53514844534 69600788111 86057777551 91649152687 91124551926 103416000000
## 3  1595555476  1733333264  2155555498  2366666616  2555555567   2953333418
## 4 25264953766 31273819026 44214484997 51444731784 62129390375  65315008068
## 5          NA          NA          NA          NA          NA           NA
## 6          NA          NA          NA          NA          NA           NA
##          X1978       X1979        X1980        X1981        X1982        X1983
## 1           NA          NA           NA           NA           NA           NA
## 2 115345000000 1.34671e+11 170654000000 174387000000 167266000000 174918000000
## 3   3300000109 3.69794e+09   3641723322   3478787909           NA           NA
## 4  71199708192 8.86284e+10 112031000000 211003000000 187164000000 138115000000
```

```
## 5           NA          NA   5930503401   5550483036   5550483036   5784341596
## 6           NA          NA           NA           NA           NA           NA
##          X1984       X1985        X1986        X1987        X1988        X1989
## 1           NA          NA    405586592    487709497    596648045    695530726
## 2 160134000000 1.36297e+11 152518000000 186145000000 204140000000 217539000000
## 3           NA          NA           NA           NA           NA           NA
## 4 114263000000 1.16507e+11 107498000000 110322000000 108943000000 101769000000
## 5   6131475065 7.55356e+09   7072063346   8083872012   8769250550  10201099039
## 6   1857338012 1.89705e+09   2097326250   2080796250   2051236250   2253090000
##          X1990        X1991        X1992        X1993        X1994        X1995
## 1    764804469    872067039    958659218   1083240223   1245810056   1320670391
## 2 253224000000 273403000000 238255000000 236527000000 240120000000 269637000000
## 3           NA          NA           NA           NA           NA           NA
## 4 121802000000 117457000000 118282000000  98826369836  86281743753 108221000000
## 5  11228764963  10603784541   8307810974   5768720422   4438321017   5538749260
## 6   2028553750   1099559028    652174991   1185315468   1880951520   2392764853
##          X1996        X1997        X1998        X1999        X2000        X2001
## 1   1379888268   1531843575   1665363128   1722905028   1873184358   1896648045
## 2 268414000000 282185000000 265814000000 262172000000 283925000000 258819000000
## 3           NA          NA           NA           NA           NA           NA
## 4 125763000000 127064000000 130107000000 137521000000 140410000000 148013000000
## 5   7526446606   7648377413   6506229607   6152922943   9129594819   8936063723
## 6   3199641336   2258513974   2545964541   3212121651   3480355258   3922100794
##          X2002        X2003        X2004        X2005        X2006        X2007
## 1   1962011173   2044134078   2254748603   2359776536   2469832402   2677653631
## 2 264870000000 352659000000 438834000000 512211000000 575921000000 661179000000
## 3   4055179566   4515558808   5226778809   6209137625   6971285595   9747879532
## 4 176938000000 204645000000 254093000000 310558000000 393305000000 461791000000
## 5  15285594828  17812704825  23552047248  36970918699  52381006892  65266452081
## 6   4348068242   5611496257   7184685782   8052073539   8896072919  10677324144
##          X2008        X2009        X2010        X2011        X2012        X2013
## 1   2843016760   2553631285   2453631285   2637988827   2615083799   2727932961
## 2 708287000000 719217000000 860478000000 964418000000 973043000000 983937000000
## 3  10109305183  12416161049  15856678596  17805113119  19907317066  20146404996
## 4 566481000000 507044000000 591596000000 670983000000 727570000000 820793000000
## 5  88538610805  70307166934  81699556137 109437000000 124998000000 133402000000
```

```
## 6  12881353508  12044208086  11926922829  12890764531  12319830437  12776220507
##          X2014        X2015        X2016        X2017        X2018        X2019
## 1 2.791061e+09   2963128492   2983798883 3.092179e+09   3202234637   3310055866
## 2 1.003680e+12 924253000000 882355000000 1.020650e+12 991022000000 997534000000
## 3 2.049713e+10  19134211764  18116562465 1.875347e+10  18053228579  18799450743
## 4 8.649900e+11 760734000000 690546000000 6.837490e+11 741690000000 794543000000
## 5 1.372440e+11  87219290029  49840494026 6.897276e+10  77792940077  69309104807
## 6 1.322815e+10  11386850130  11861199831 1.301969e+10  15156432310  15401830754
##          X2020        X2021
## 1   2496648045           NA
## 2 921646000000 1.082100e+12
## 3  20116137326           NA
## 4 784446000000 8.358080e+11
## 5  53619071176 7.254699e+10
## 6  15131866271 1.826004e+10
```

Remove column 3 and 4 cause it is all the same value. All column 3 values are GDP *(current US$)* and column 4 values are *NY.GDP.MKTP.CD*.

```
global_gdp <- global_gdp %>%
                select(-c(3:4))
head(global_gdp)
```

```
##                  Country.Name Country.Code       X1960       X1961       X1962
## 1                       Aruba          ABW          NA          NA          NA
## 2 Africa Eastern and Southern          AFE 21290586003 21808473825 23707015394
## 3                 Afghanistan          AFG   537777811   548888896   546666678
## 4  Africa Western and Central          AFW 10404135069 11127894641 11943187848
## 5                      Angola          AGO          NA          NA          NA
## 6                     Albania          ALB          NA          NA          NA
##         X1963       X1964       X1965       X1966       X1967       X1968
## 1          NA          NA          NA          NA          NA          NA
## 2 28210036878 26118787467 29682172751 32239121547 33514552047 36521482937
## 3   751111191   800000044  1006666638  1399999967  1673333418  1373333367
## 4 12676330765 13838369295 14862225760 15832591204 14426038230 14880349280
## 5          NA          NA          NA          NA          NA          NA
```

```
## 6          NA          NA          NA          NA          NA          NA
##         X1969       X1970       X1971       X1972       X1973       X1974
## 1          NA          NA          NA          NA          NA          NA
## 2 41828336213 44862605393 49478916698 53514844534 69600788111 86057777551
## 3  1408888922  1748886596  1831108971  1595555476  1733333264  2155555498
## 4 16882092549 23504605476 20832817218 25264953766 31273819026 44214484997
## 5          NA          NA          NA          NA          NA          NA
## 6          NA          NA          NA          NA          NA          NA
##         X1975       X1976        X1977        X1978       X1979        X1980
## 1          NA          NA           NA           NA          NA           NA
## 2 91649152687 91124551926 103416000000 115345000000 1.34671e+11 170654000000
## 3  2366666616  2555555567   2953333418   3300000109 3.69794e+09   3641723322
## 4 51444731784 62129390375  65315008068  71199708192 8.86284e+10 112031000000
## 5          NA          NA           NA           NA          NA   5930503401
## 6          NA          NA           NA           NA          NA           NA
##          X1981        X1982        X1983        X1984       X1985        X1986
## 1           NA           NA           NA           NA          NA    405586592
## 2 174387000000 167266000000 174918000000 160134000000 1.36297e+11 152518000000
## 3   3478787909           NA           NA           NA          NA           NA
## 4 211003000000 187164000000 138115000000 114263000000 1.16507e+11 107498000000
## 5   5550483036   5550483036   5784341596   6131475065 7.55356e+09   7072063346
## 6           NA           NA           NA   1857338012 1.89705e+09   2097326250
##          X1987        X1988        X1989        X1990        X1991        X1992
## 1    487709497    596648045    695530726    764804469    872067039    958659218
## 2 186145000000 204140000000 217539000000 253224000000 273403000000 238255000000
## 3           NA           NA           NA           NA           NA           NA
## 4 110322000000 108943000000 101769000000 121802000000 117457000000 118282000000
## 5   8083872012   8769250550  10201099039  11228764963  10603784541   8307810974
## 6   2080796250   2051236250   2253090000   2028553750   1099559028    652174991
##          X1993        X1994        X1995        X1996        X1997        X1998
## 1   1083240223   1245810056   1320670391   1379888268   1531843575   1665363128
## 2 236527000000 240120000000 269637000000 268414000000 282185000000 265814000000
## 3           NA           NA           NA           NA           NA           NA
## 4  98826369836  86281743753 108221000000 125763000000 127064000000 130107000000
## 5   5768720422   4438321017   5538749260   7526446606   7648377413   6506229607
## 6   1185315468   1880951520   2392764853   3199641336   2258513974   2545964541
```

```
##          X1999        X2000        X2001        X2002        X2003        X2004
## 1   1722905028   1873184358   1896648045   1962011173   2044134078   2254748603
## 2 262172000000 283925000000 258819000000 264870000000 352659000000 438834000000
## 3           NA           NA           NA   4055179566   4515558808   5226778809
## 4 137521000000 140410000000 148013000000 176938000000 204645000000 254093000000
## 5   6152922943   9129594819   8936063723  15285594828  17812704825  23552047248
## 6   3212121651   3480355258   3922100794   4348068242   5611496257   7184685782
##          X2005        X2006        X2007        X2008        X2009        X2010
## 1   2359776536   2469832402   2677653631   2843016760   2553631285   2453631285
## 2 512211000000 575921000000 661179000000 708287000000 719217000000 860478000000
## 3   6209137625   6971285595   9747879532  10109305183  12416161049  15856678596
## 4 310558000000 393305000000 461791000000 566481000000 507044000000 591596000000
## 5  36970918699  52381006892  65266452081  88538610805  70307166934  81699556137
## 6   8052073539   8896072919  10677324144  12881353508  12044208086  11926922829
##          X2011        X2012        X2013        X2014        X2015        X2016
## 1   2637988827   2615083799   2727932961 2.791061e+09   2963128492   2983798883
## 2 964418000000 973043000000 983937000000 1.003680e+12 924253000000 882355000000
## 3  17805113119  19907317066  20146404996 2.049713e+10  19134211764  18116562465
## 4 670983000000 727570000000 820793000000 8.649900e+11 760734000000 690546000000
## 5 109437000000 124998000000 133402000000 1.372440e+11  87219290029  49840494026
## 6  12890764531  12319830437  12776220507 1.322815e+10  11386850130  11861199831
##          X2017        X2018        X2019        X2020        X2021
## 1 3.092179e+09   3202234637   3310055866   2496648045           NA
## 2 1.020650e+12 991022000000 997534000000 921646000000 1.082100e+12
## 3 1.875347e+10  18053228579  18799450743  20116137326           NA
## 4 6.837490e+11 741690000000 794543000000 784446000000 8.358080e+11
## 5 6.897276e+10  77792940077  69309104807  53619071176 7.254699e+10
## 6 1.301969e+10  15156432310  15401830754  15131866271 1.826004e+10
```

Change . in the header to *_* and remove X in front of years.

```
colnames(global_gdp) <- colnames(global_gdp) %>%
                        str_replace_all('\\.', '_') %>%
                        str_replace_all('X(?=\\d)', '')
head(global_gdp)
```

```
##                   Country_Name Country_Code        1960        1961        1962
## 1                        Aruba          ABW          NA          NA          NA
## 2 Africa Eastern and Southern          AFE 21290586003 21808473825 23707015394
## 3                  Afghanistan          AFG   537777811   548888896   546666678
## 4  Africa Western and Central          AFW 10404135069 11127894641 11943187848
## 5                       Angola          AGO          NA          NA          NA
## 6                      Albania          ALB          NA          NA          NA
##          1963        1964        1965        1966        1967        1968
## 1          NA          NA          NA          NA          NA          NA
## 2 28210036878 26118787467 29682172751 32239121547 33514552047 36521482937
## 3   751111191   800000044  1006666638  1399999967  1673333418  1373333367
## 4 12676330765 13838369295 14862225760 15832591204 14426038230 14880349280
## 5          NA          NA          NA          NA          NA          NA
## 6          NA          NA          NA          NA          NA          NA
##          1969        1970        1971        1972        1973        1974
## 1          NA          NA          NA          NA          NA          NA
## 2 41828336213 44862605393 49478916698 53514844534 69600788111 86057777551
## 3  1408888922  1748886596  1831108971  1595555476  1733333264  2155555498
## 4 16882092549 23504605476 20832817218 25264953766 31273819026 44214484997
## 5          NA          NA          NA          NA          NA          NA
## 6          NA          NA          NA          NA          NA          NA
##          1975        1976         1977         1978        1979         1980
## 1          NA          NA           NA           NA          NA           NA
## 2 91649152687 91124551926 103416000000 115345000000 1.34671e+11 170654000000
## 3  2366666616  2555555567   2953333418   3300000109 3.69794e+09   3641723322
## 4 51444731784 62129390375  65315008068  71199708192 8.86284e+10 112031000000
## 5          NA          NA           NA           NA          NA   5930503401
## 6          NA          NA           NA           NA          NA           NA
##          1981         1982         1983         1984        1985         1986
## 1          NA           NA           NA           NA          NA    405586592
## 2 174387000000 167266000000 174918000000 160134000000 1.36297e+11 152518000000
## 3   3478787909           NA           NA           NA          NA           NA
## 4 211003000000 187164000000 138115000000 114263000000 1.16507e+11 107498000000
## 5   5550483036   5550483036   5784341596   6131475065 7.55356e+09   7072063346
## 6          NA           NA           NA   1857338012 1.89705e+09   2097326250
##          1987         1988         1989         1990         1991         1992
```

```
## 1    487709497    596648045    695530726    764804469    872067039    958659218
## 2 186145000000 204140000000 217539000000 253224000000 273403000000 238255000000
## 3           NA           NA           NA           NA           NA           NA
## 4 110322000000 108943000000 101769000000 121802000000 117457000000 118282000000
## 5   8083872012   8769250550  10201099039  11228764963  10603784541   8307810974
## 6   2080796250   2051236250   2253090000   2028553750   1099559028    652174991
##           1993         1994         1995         1996         1997         1998
## 1   1083240223   1245810056   1320670391   1379888268   1531843575   1665363128
## 2 236527000000 240120000000 269637000000 268414000000 282185000000 265814000000
## 3           NA           NA           NA           NA           NA           NA
## 4  98826369836  86281743753 108221000000 125763000000 127064000000 130107000000
## 5   5768720422   4438321017   5538749260   7526446606   7648377413   6506229607
## 6   1185315468   1880951520   2392764853   3199641336   2258513974   2545964541
##           1999         2000         2001         2002         2003         2004
## 1   1722905028   1873184358   1896648045   1962011173   2044134078   2254748603
## 2 262172000000 283925000000 258819000000 264870000000 352659000000 438834000000
## 3           NA           NA           NA   4055179566   4515558808   5226778809
## 4 137521000000 140410000000 148013000000 176938000000 204645000000 254093000000
## 5   6152922943   9129594819   8936063723  15285594828  17812704825  23552047248
## 6   3212121651   3480355258   3922100794   4348068242   5611496257   7184685782
##           2005         2006         2007         2008         2009         2010
## 1   2359776536   2469832402   2677653631   2843016760   2553631285   2453631285
## 2 512211000000 575921000000 661179000000 708287000000 719217000000 860478000000
## 3   6209137625   6971285595   9747879532  10109305183  12416161049  15856678596
## 4 310558000000 393305000000 461791000000 566481000000 507044000000 591596000000
## 5  36970918699  52381006892  65266452081  88538610805  70307166934  81699556137
## 6   8052073539   8896072919  10677324144  12881353508  12044208086  11926922829
##           2011         2012         2013         2014         2015         2016
## 1   2637988827   2615083799   2727932961 2.791061e+09   2963128492   2983798883
## 2 964418000000 973043000000 983937000000 1.003680e+12 924253000000 882355000000
## 3  17805113119  19907317066  20146404996 2.049713e+10  19134211764  18116562465
## 4 670983000000 727570000000 820793000000 8.649900e+11 760734000000 690546000000
## 5 109437000000 124998000000 133402000000 1.372440e+11  87219290029  49840494026
## 6  12890764531  12319830437  12776220507 1.322815e+10  11386850130  11861199831
##           2017         2018         2019         2020         2021
## 1 3.092179e+09   3202234637   3310055866   2496648045           NA
```

```
## 2 1.020650e+12 991022000000 997534000000 921646000000 1.082100e+12
## 3 1.875347e+10  18053228579  18799450743  20116137326           NA
## 4 6.837490e+11 741690000000 794543000000 784446000000 8.358080e+11
## 5 6.897276e+10  77792940077  69309104807  53619071176 7.254699e+10
## 6 1.301969e+10  15156432310  15401830754  15131866271 1.826004e+10
```

```
tail(global_gdp)
```

```
##     Country_Name Country_Code       1960       1961       1962        1963
## 261        Samoa          WSM         NA         NA         NA          NA
## 262       Kosovo          XKX         NA         NA         NA          NA
## 263  Yemen, Rep.          YEM         NA         NA         NA          NA
## 264 South Africa          ZAF 8748596504 9225996313 9813996079 10854195663
## 265       Zambia          ZMB  713000000  696285714  693142857   718714286
## 266     Zimbabwe          ZWE 1052990400 1096646600 1117601600  1159511700
##            1964        1965        1966        1967        1968        1969
## 261          NA          NA          NA          NA          NA          NA
## 262          NA          NA          NA          NA          NA          NA
## 263          NA          NA          NA          NA          NA          NA
## 264 11955995223 13068994778 14211394321 15821393678 17124793157 19256992305
## 265   839428571  1082857143  1264285714  1368000000  1605857143  1965714286
## 266  1217138000  1311435800  1281749500  1397002000  1479599900  1747998800
##            1970        1971        1972        1973        1974        1975
## 261          NA          NA          NA          NA          NA          NA
## 262          NA          NA          NA          NA          NA          NA
## 263          NA          NA          NA          NA          NA          NA
## 264 21218391522 23411079378 24515911652 33262767311 41389185875 42906919870
## 265  1825285714  1687000000  1910714286  2268714286  3121833333  2618666667
## 266  1884206300  2178716300  2677729400  3309353600  3982161400  4371300700
##            1976        1977        1978        1979        1980        1981
## 261          NA          NA          NA          NA          NA          NA
## 262          NA          NA          NA          NA          NA          NA
## 263          NA          NA          NA          NA          NA          NA
## 264 41150449966 45328399963 51607399958 63038687893 89411894561 93141478235
## 265  2746714286  2483000000  2813375000  3325500000  3829500000  3872666667
## 266  4318372000  4364382100  4351600500  5177459400  6678868200  8011373800
##            1982        1983        1984        1985        1986        1987
```

```
## 261   121221652   111862824   109200934    95572173   100947849   111713922
## 262          NA          NA          NA          NA          NA          NA
## 263          NA          NA          NA          NA          NA          NA
## 264 85904070614 96204110959 84870134619 64459376104 73354782109 96535747615
## 265  3994777778  3216307692  2739444444  2281258065  1661948718  2269894737
## 266  8539700700  7764067000  6352125900  5637259300  6217523700  6741215100
##             1988         1989         1990         1991         1992
## 261    133016065    122888610    125766270    125597205    132303041
## 262           NA           NA           NA           NA           NA
## 263           NA           NA   5647119229   5930370370   6463649985
## 264 103977000000 108056000000 126048000000 135204000000 146957000000
## 265   3713614458   3998637681   3285217391   3378882353   3181921788
## 266   7814784100   8286322700   8783816700   8641481700   6751472200
##             1993         1994         1995         1996         1997
## 261    133122897    221098107    224865731    249908971    285475592
## 262           NA           NA           NA           NA           NA
## 263   5368270615   4167356037   4258788725   5785685311   6838557384
## 264 147197000000 153513000000 171735000000 163237000000 168977000000
## 265   3273237853   3656647744   3807067122   3597220962   4303281932
## 266   6563813300   6890675000   7111270700   8553146600   8529571600
##             1998         1999         2000         2001         2002
## 261    269481523    255410025    258856139    266299604    281793615
## 262           NA           NA           NA           NA           NA
## 263   6325141676   7641102523   9652436180   9861560095  10694628092
## 264 152983000000 151517000000 151753000000 135430000000 129088000000
## 265   3537683046   3404311977   3600683040   4094480988   4193845678
## 266   6401968200   6858013100   6689957600   6777384700   6342116400
##             2003         2004         2005         2006         2007
## 261    333428714    407749577    476801791    499923741    573548460
## 262           NA           NA           NA           NA           NA
## 263  11777966673  13872791659  16746344766  19061978586  21650532264
## 264 197020000000 255807000000 288868000000 303861000000 333075000000
## 265   4901839731   6221077675   8331870169  12756858899  14056957976
## 266   5727591800   5805598400   5755215200   5443896500   5291950100
##             2008         2009         2010         2011         2012
## 261    641346175    628006123    663161528    737401684    760549578
```

```
## 262   5181776769   5015894693   5344014318   6341737194   6163785173
## 263  26910851362  25130274124  30906749533  32726417212  35401341663
## 264 316132000000 329753000000 417365000000 458202000000 434401000000
## 265  17910858638  15328342304  20265559484  23459515276  25503060420
## 266   4415702800   9665793300  12041655200  14101920300  17114849900
##             2013         2014         2015         2016         2017
## 261    770059565    756805950    788307315    799493898    832025556
## 262   6735731173   7074657898   6295820482   6682832632   7180813376
## 263  40415235702  43228585321  42444495590  31317828584  26842231205
## 264 400886000000 381199000000 346710000000 323586000000 381449000000
## 265  28037239463  27141023558  21251216799  20958412538  25873601261
## 266  19091020000  19495519600  19963120600  20548678100  17584890937
##             2018         2019         2020         2021
## 261    821286939    852007105    807147528    788389972
## 262   7878508503   7899879086   7716925356   9007159196
## 263  21606161066  21887614217  18840511908  21061691630
## 264 404842000000 387935000000 335442000000 419946000000
## 265  26311590297  23308667781  18110631358  21203059080
## 266  18115543791  19284289739  18051170799  26217726717
```

Transfer into a tidy data frame. Use *pivot_longer()* function to pivot from column 3 *1960* to last column *2021*. Exclude cell with NA value while pivoting.

```
global_gdp_longer <- global_gdp %>%
                        pivot_longer(
                          cols = colnames(global_gdp)[-c(1,2)],
                          names_to = 'Year',
                          values_to = 'GDP',
                          values_drop_na = TRUE
                        )
head(global_gdp_longer)
```

```
## # A tibble: 6 x 4
##   Country_Name Country_Code Year         GDP
##   <chr>        <chr>        <chr>      <dbl>
## 1 Aruba        ABW          1986  405586592.
```

```
## 2 Aruba        ABW          1987  487709497.
## 3 Aruba        ABW          1988  596648045.
## 4 Aruba        ABW          1989  695530726.
## 5 Aruba        ABW          1990  764804469.
## 6 Aruba        ABW          1991  872067039.
```

```
tail(global_gdp_longer)
```

```
## # A tibble: 6 x 4
##   Country_Name Country_Code Year          GDP
##   <chr>        <chr>        <chr>       <dbl>
## 1 Zimbabwe     ZWE          2016  20548678100
## 2 Zimbabwe     ZWE          2017  17584890937
## 3 Zimbabwe     ZWE          2018  18115543791
## 4 Zimbabwe     ZWE          2019  19284289739
## 5 Zimbabwe     ZWE          2020  18051170799
## 6 Zimbabwe     ZWE          2021  26217726717
```

Convert *Year*data type; Character to double.

```
global_gdp_longer$Year = as.numeric(global_gdp_longer$Year)
```

Calculate the growth rate for each county.

```
global_gdp_longer <- global_gdp_longer %>%
  group_by(Country_Name) %>%
    mutate(
      Prev_10_GDP = ifelse(is.na(lag(GDP, n = 10)), 0, lag(GDP, n = 10)),
      Diff_growth = ifelse(is.na(lag(GDP)), 0, GDP - lag(GDP)),
      Rate_percent = ifelse(is.na(lag(GDP)), 0, round((GDP / lag(GDP) - 1) * 100, digits = 2))
      )
global_gdp_longer
```

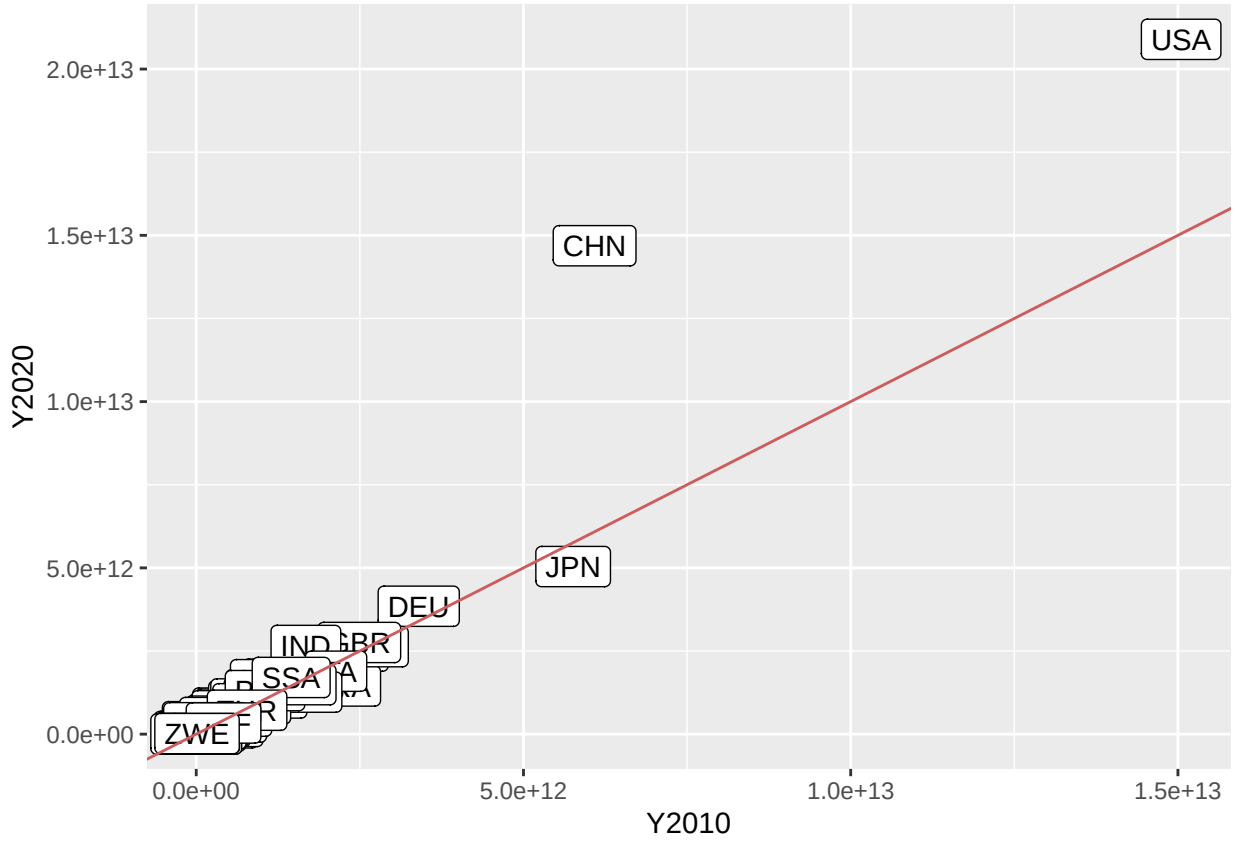
```
## # A tibble: 13,118 x 7
## # Groups:   Country_Name [262]
```

```
##    Country_Name Country_Code  Year         GDP Prev_10_GDP Diff_growth Rate_pe~1
##    <chr>        <chr>        <dbl>       <dbl>       <dbl>       <dbl>     <dbl>
##  1 Aruba        ABW           1986  405586592.           0           0       0
##  2 Aruba        ABW           1987  487709497.           0    82122905      20.2
##  3 Aruba        ABW           1988  596648045.           0   108938548.     22.3
##  4 Aruba        ABW           1989  695530726.           0    98882682.     16.6
##  5 Aruba        ABW           1990  764804469.           0    69273743       9.96
##  6 Aruba        ABW           1991  872067039.           0   107262570.     14.0
##  7 Aruba        ABW           1992  958659218.           0    86592179.      9.93
##  8 Aruba        ABW           1993 1083240223           0   124581005.     13
##  9 Aruba        ABW           1994 1245810056           0   162569833      15.0
## 10 Aruba        ABW           1995 1320670391           0    74860335       6.01
## # ... with 13,108 more rows, and abbreviated variable name 1: Rate_percent
```

Compare 2010 and 2020. The graph shows the GDP of developed countries continued to rise over time. Especially, USA and China's GDP grew strongly. However, underdeveloped countries did not grow.

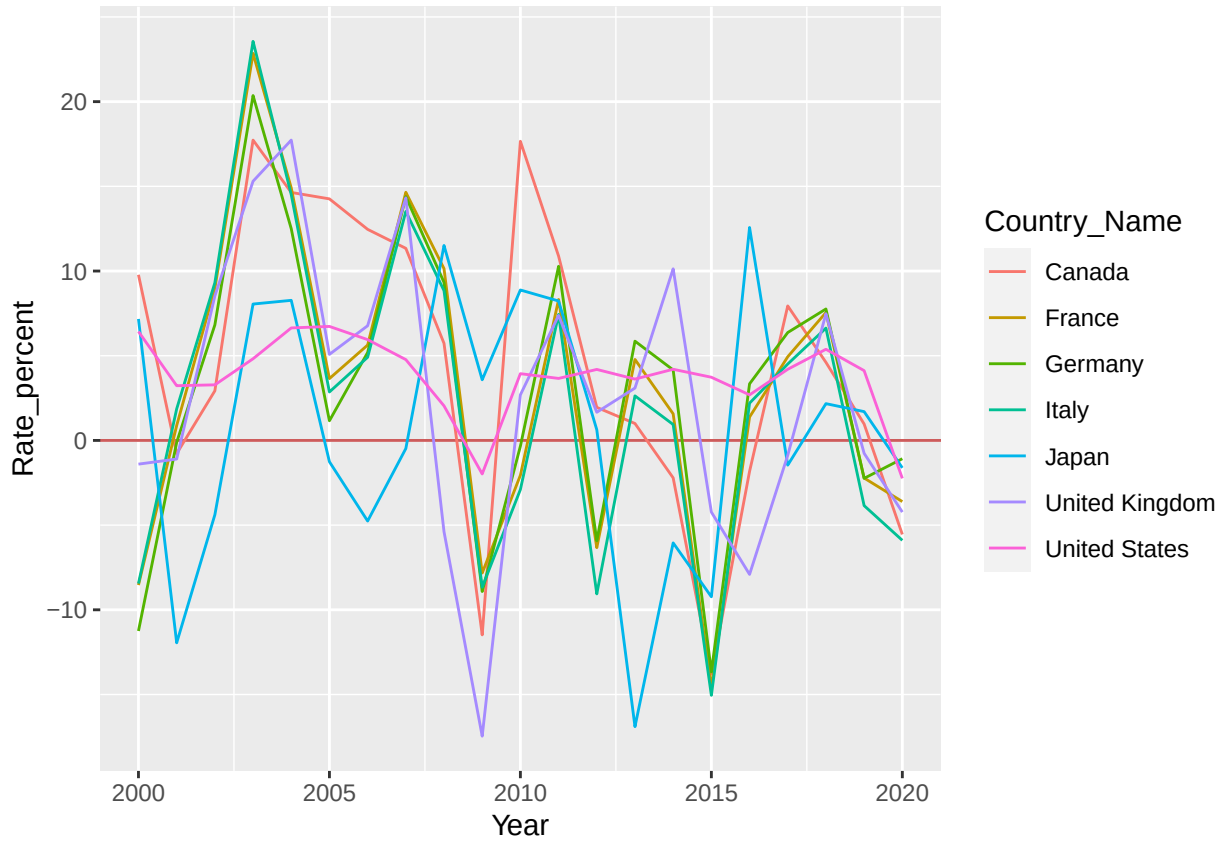
```
exclude <- c('WLD', 'OED', 'PST', 'EAP', 'EAR', 'EAS', 'ECA', 'ECS', 'FCS', 'HPC',
             'UMC', 'EUU', 'EMU', 'LMC', 'TLA', 'TEA', 'HIC', 'NAC', 'IBT', 'MIC',
             'LTE', 'LMY', 'LCN', 'LDC', 'TEC', 'TSA', 'TMN', 'TSS', 'IBD', 'LAC',
             'SAS', 'MEA', 'SSF')
global_gdp_longer %>%
  filter(Year == 2020, !(Country_Code %in% exclude)) %>%
  ggplot(aes(x = Prev_10_GDP, y = GDP, label = Country_Code), color = 'SteelBlue') +
  geom_point() +
  geom_label(
    nudge_x = 0.25, nudge_y = 0.25,
    check_overlap = T
  ) +
  geom_abline(intercept = 0, slope = 1, size = 0.5, color = 'IndianRed') +
  labs(
    x = 'Y2010',
    y = 'Y2020'
  )
```

```
## Warning: Ignoring unknown parameters: check_overlap
```

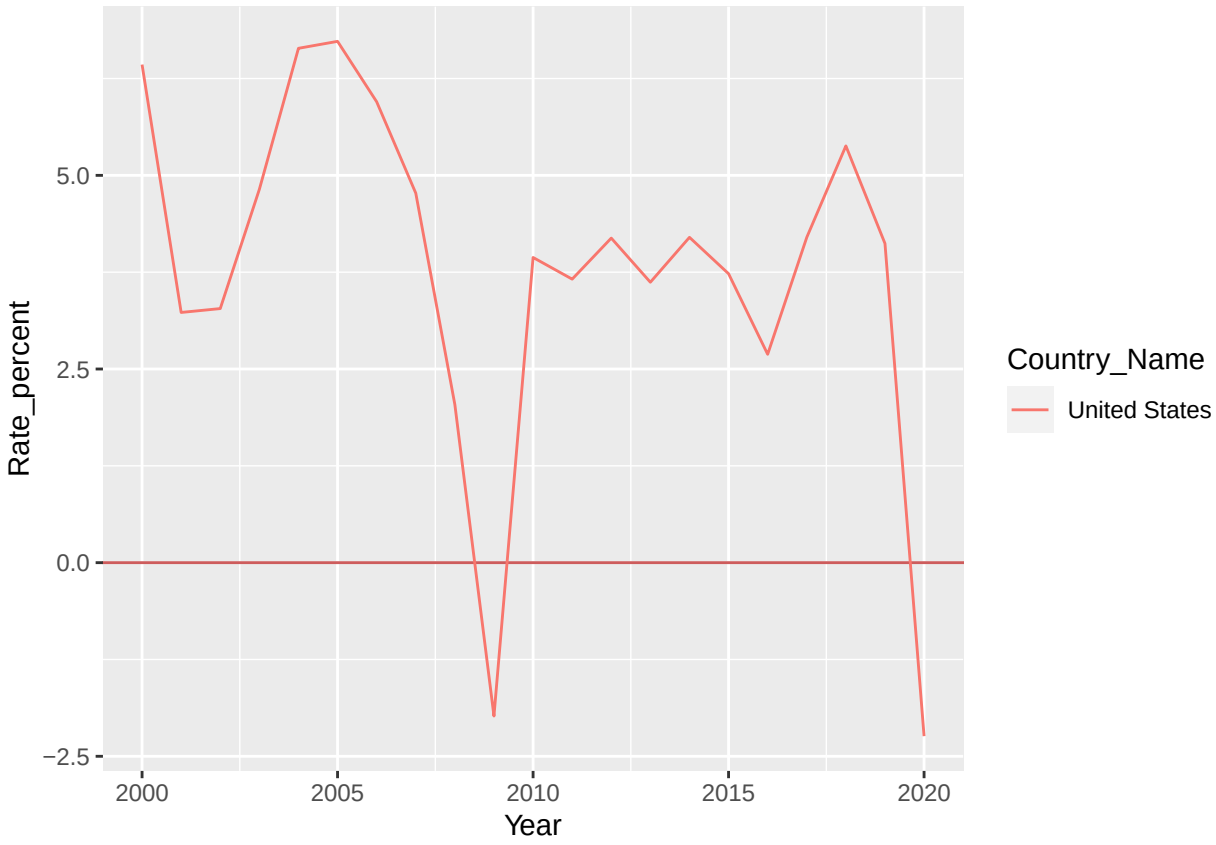


Compare GDP growth for G7 country in a 10-year cycle. In 2008 there was the subprime mortgage crisis, and in 2020 there was the corona virus pandemic.

```
#DEU: Germany, GBR: the United Kingdom
G7 <- c('CAN', 'FRA', 'DEU', 'ITA', 'JPN', 'GBR', 'USA')
global_gdp_longer %>%
  filter(Country_Code %in% G7, Year > 1999, Year < 2021) %>%
  ggplot(aes(x = Year, y = Rate_percent, color = Country_Name)) +
    geom_abline(intercept = 0, slope = 0, size = 0.5, color = 'IndianRed')+
      geom_line() +
      labs(
        x = 'Year',
        y = 'Rate_percent'
      )
```




```
#DEU: Germany, GBR: the United Kingdom
G7 <- c('USA')
global_gdp_longer %>%
  filter(Country_Code %in% G7, Year > 1999, Year < 2021) %>%
  ggplot(aes(x = Year, y = Rate_percent, color = Country_Name)) +
    geom_abline(intercept = 0, slope = 0, size = 0.5, color = 'IndianRed')+
      geom_line() +
      labs(
        x = 'Year',
        y = 'Rate_percent'
      )
```


Rate_percent
5.0
2.5
0.0
−2.5
2000
2005
2010
2015
2020
Year
Country_Name
United States