Data Visualization Project

mutwiri_ian@yahoo.com

03 April,2022

These are my data visualization practice projects and in the following sections I demonstrate my skills by walking through my steps in creating the visualizations herein. The three visualization are on the demographics in Kenya,particularly births rates in Kenya disaggregated at the county level,the public debt trends, the migration trends in sample OECD countries and key macro rates of the US economy. Let's get into it!

For this project the required packages: the `Tidyverse` meta-package which loads a collection of other packages which together to manipulate,transform and visualize data,`patchwork` for arrangement of plots and `rKenyaCensus`, which I use to access the Kenya census data for 2019.

```
library(tidyverse)
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

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

```
library(patchwork)
```

The data on Kenya birth rates are provided by the Kenya National Bureau of Statistics and a format which has been pre-processed and easier to work with has been provided by the `rKenyaCensus` package. First I load the data,do some wrangling and then generate a plot using the powerful `ggplot2` package. The number of counties cannot all fit in one screen so I prefer to create two side by side plots.

```
births1 <- rKenyaCensus::V4_T2.40[-c(1:3),]%>%
  select(-c(2,5,6))%>%
  pivot_longer(cols = c(2,3),values_to = 'Count',
                                         names_to = "Status")%>%
  select(-Percent_Notified) %>%
  group_by(County) %>%
  mutate(
    Status=case_when(Status=="Notified"~"Notified",TRUE~"Not Notified"),
    pct=Count/sum(Count)) %>%
  arrange(desc(Count))
```

```
counties <- distinct(births1,County) %>% pull(County)
set1 <- counties[1:23]
set2 <- counties[-(1:23)]
births1
```

```
## # A tibble: 94 x 4
## # Groups:   County [47]
##    County       Status    Count   pct
##    <chr>        <chr>     <dbl> <dbl>
##  1 NAIROBI CITY Notified 484632 0.963
##  2 KIAMBU       Notified 250515 0.970
##  3 NAKURU       Notified 204887 0.915
##  4 KAKAMEGA     Notified 150147 0.897
##  5 BUNGOMA      Notified 140356 0.881
##  6 MERU         Notified 137214 0.953
##  7 KILIFI       Notified 131765 0.924
##  8 MOMBASA      Notified 122981 0.939
##  9 MACHAKOS     Notified 118122 0.935
## 10 KAJIADO      Notified 113508 0.896
## # ... with 84 more rows
```

```
#Generate first plot
birthsA<- births1%>%
  filter(County%in%set1) %>%
  ggplot(aes(reorder(County,Count),Count,fill=Status))+
  geom_bar(stat = 'identity',position = position_dodge(width = 1))+
  labs(title = 'Kenya Birth Numbers across counties',
       subtitle = 'First 23 counties ordered by notification rate',
       caption = "Chart by @mutwiriian\n   Source: 2019 Kenya Population and Housing Census Results")+
  xlab('County')+ylab('Number of births')+
  geom_text(aes(label=Count),position=position_dodge(1),
            size=3.2,hjust=0,vjust=.4)+
  geom_text(aes(label=paste(",",round(pct*100,2),"%")),
            position = position_dodge(1),size=3.2,hjust=-.8,vjust=.4)+
  scale_fill_brewer(palette = 'Dark2',type = 'qual')+
  scale_y_continuous(labels = scales::comma,expand = c(0,0),limits = c(0,530000))+
  theme(legend.position = c(.8,.6),
        plot.caption = element_text(face = 'bold.italic',size = 10,vjust = 5,hjust = .05))+
  coord_flip()
#Generate second plot
birthsB<- births1%>%
  filter(County%in%set2) %>%
  ggplot(aes(reorder(County,Count),Count,fill=Status))+
  geom_bar(stat = 'identity',position = position_dodge(width = 1))+
  labs(title = 'Kenya Birth Numbers across counties',
       subtitle = 'Next 24 counties ordered by notification rate',
       caption = "Chart by @mutwiriian\n   Source: 2019 Kenya Population and Housing Census Results")+
  xlab('County')+ylab('Number of births')+
  geom_text(aes(label=Count),position=position_dodge(1),
            size=3.2,hjust=0,vjust=.4)+
  geom_text(aes(label=paste(",",round(pct*100,2),"%")),
            position = position_dodge(1),size=3.2,hjust=-.8,vjust=.4)+
  scale_fill_brewer(palette = 'Dark2',type = 'qual')+
```

```r
  scale_y_continuous(labels = scales::comma,expand = c(0,0),limits = c(0,200000))+
  theme(legend.position = c(.8,.6),
        plot.caption = element_text(face = 'bold.italic',size = 10,vjust = 5,hjust = .05))+
  coord_flip()

birthsA/birthsB
```

Kenya Birth Numbers across counties

First 23 counties ordered by notification rate

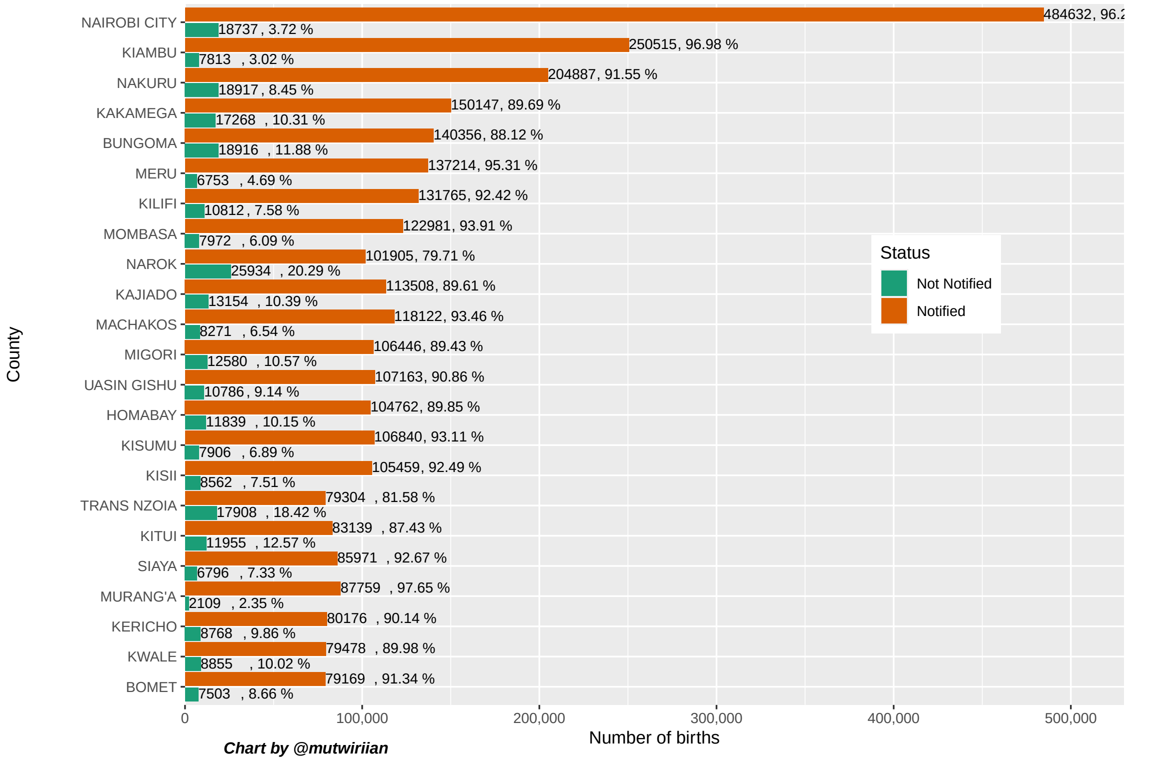


Chart by @mutwiriian
Source: 2019 Kenya Population and Housing Census Results

Kenya Birth Numbers across counties

Next 24 counties ordered by notification rate

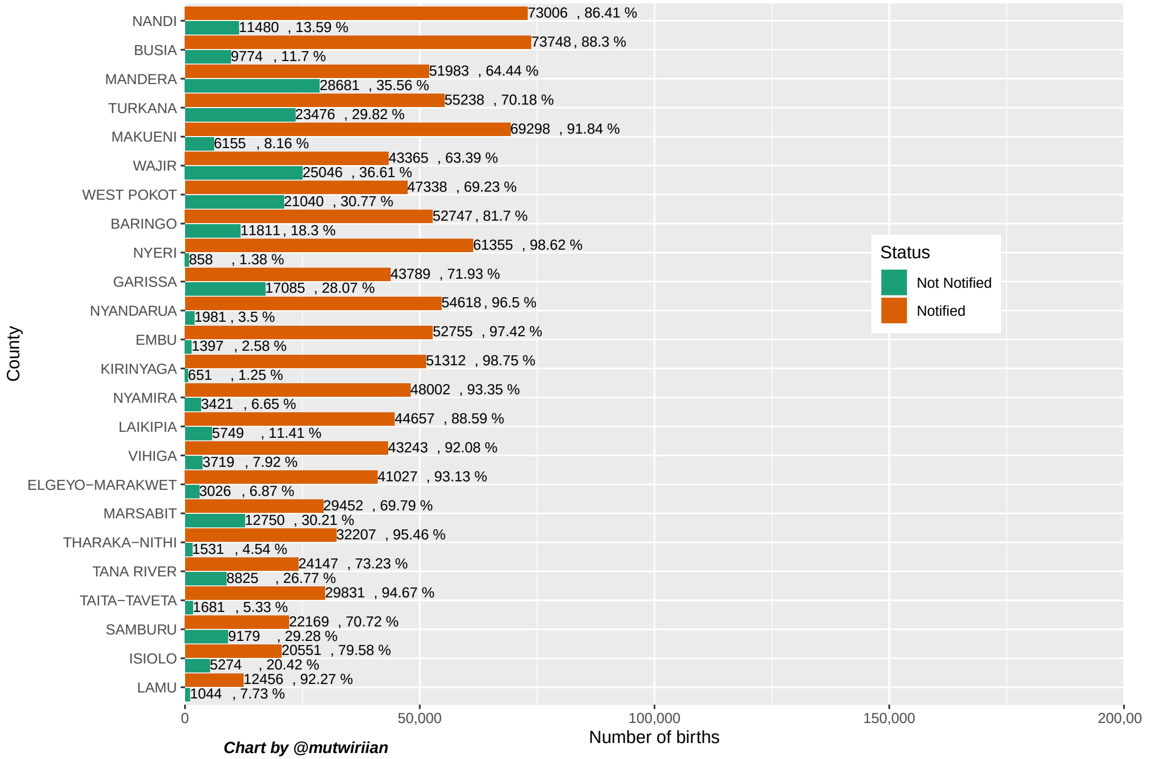


Chart by @mutwiriian
Source: 2019 Kenya Population and Housing Census Results

Next is my visualization of the Kenya debt levels from 2000 to 2020. There are a few errors and inconsistencies which requires pre-processing steps so that it is in a format that can be visualized easily. First, I rename the first column and since the data is in text format inc which values are separated by the big mark comma, I use a for loop to remove the commas and then transform the data from text to numeric type.

```
debt <- read.csv("E:/Workspace/cbkdebt.csv",sep=",",header = T)
colnames(debt)[1] <- "Year"
for(i in 3:5){
  debt[,i] <- as.numeric(lapply(debt[,i],gsub,pattern=',',replacement=''))
}
clean_debt <- debt%>%
  filter(Month=="December"|Month=="June"&Year=="2020")%>%
  group_by(Year,Month)%>%
  select(-2)%>%
  pivot_longer(cols=c(Domestic.Debt,External.Debt,Total),names_to="Type",
             values_to="Amount") %>%
  mutate(Amount=Amount/1000000,
         Type=case_when(
           Type=="Domestic.Debt"~"Domestic",
           Type=="External.Debt"~"External",
           TRUE~'Total'
         ))
```

```
## Adding missing grouping variables: ‘Month‘
```

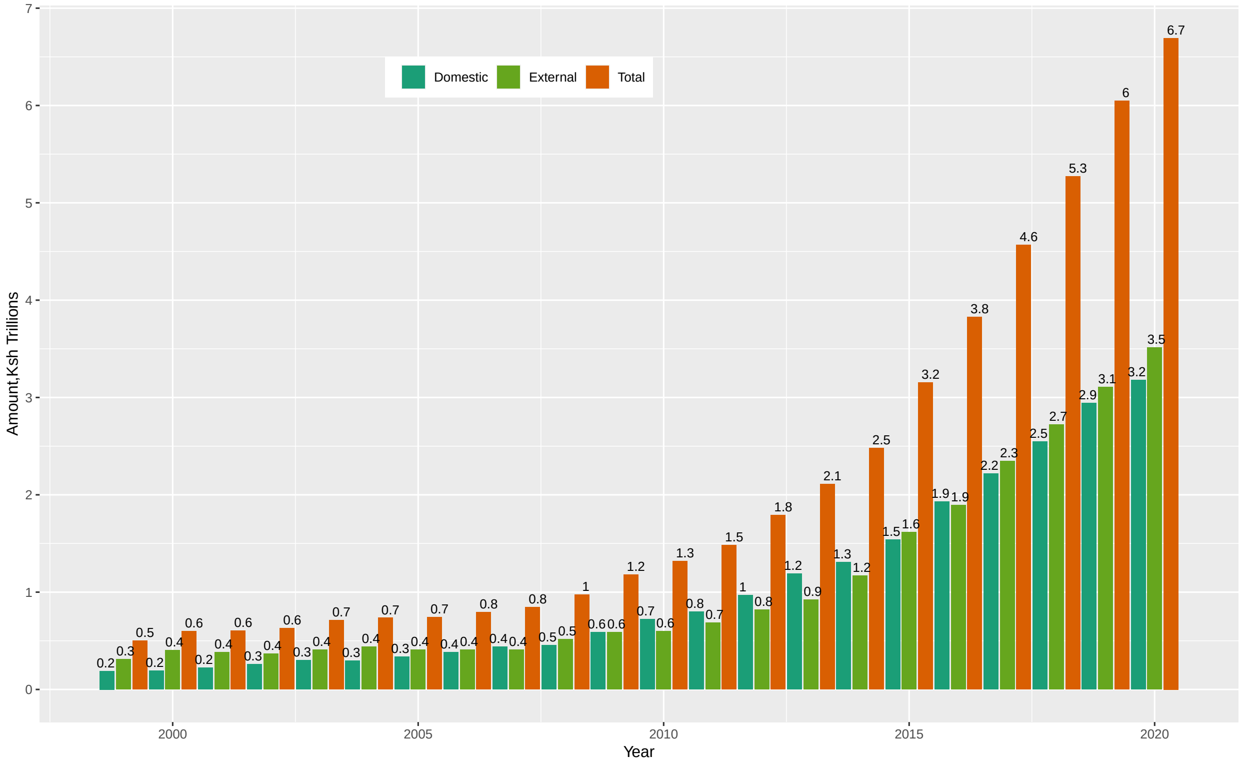
```
glimpse(clean_debt)
```

```
## Rows: 66
## Columns: 4
## Groups: Year, Month [22]
## $ Month  <chr> "June", "June", "June", "December", "December", "December", "De~
## $ Year   <int> 2020, 2020, 2020, 2019, 2019, 2019, 2018, 2018, 2018, 2017, 201~
## $ Type   <chr> "Domestic", "External", "Total", "Domestic", "External", "Total~
## $ Amount <dbl> 3.1775259, 3.5158108, 6.6933366, 2.9421035, 3.1068230, 6.048926~
```

The data is in a `Tidy` format and I proceed to create the visualization

```
ggplot(clean_debt,aes(x=Year,y=Amount,fill=Type))+
  geom_bar(stat="identity",position=position_dodge(1))+
  geom_text(aes(label=round(Amount,1)),vjust=-.4,hjust=.4,size=3,
            color="black",position = position_dodge(1.2))+
  scale_y_continuous(labels = paste(seq(0,7,1)),
                     breaks =seq(0,7,1))+
  labs(title="Kenya Debt Composition,09/`99-06/`20",
       y="Amount,Ksh Trillions",
       caption ="Compiled by @mutwiriian\nSource:Central Bank of Kenya")+
        scale_fill_manual(values = c("#1B9E77","#66A61E","#D95F02"),
                          labels=c("Domestic","External","Total"))+
  theme(legend.direction = "horizontal",legend.position = c(0.4,.9),
        legend.title =element_blank(),
        plot.caption = element_text(size = 10,
        margin =margin(t=5),hjust = .1))
```

Kenya Debt Composition,09/`99−06/`20



Compiled by @mutwiriian
Source:Central Bank of Kenya

In this final plot, I use data from the OECD International Migration Database.I use the `innerjoin` function to select countries for both emigration and immigration data is available and also remove countries which have atleast one missing entry since this will cause errors especially with scatterplots and line plots which require values on both axis to be of the same length.

```
migrationA <- readxl::read_xlsx("E:/Workspace/inflowsOECD.xlsx",sheet = 1,
                               skip = 2,na = "..")
```

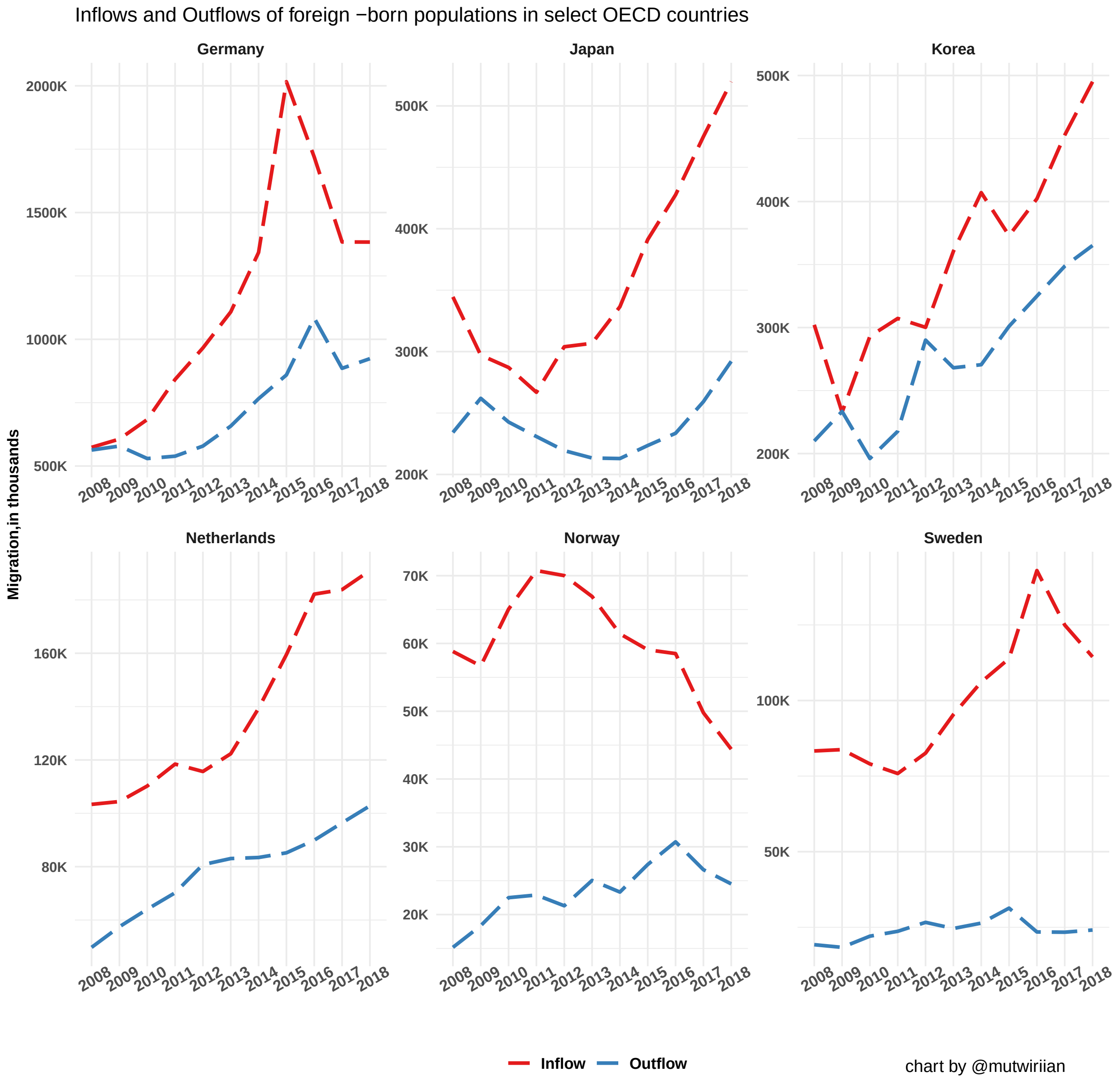
```
## New names:
## * ‘‘ -> ...1
```

```
migrationA <- na.omit(migrationA)
colnames(migrationA)[1] <- 'Country'

migrationB <- readxl::read_xlsx("E:/Workspace/outflowsOECD.xlsx",sheet = 1,
                               skip = 2,na = "..")
```

```
## New names:
## * ‘‘ -> ...1
```

```
colnames(migrationB)[1] <- 'Country'
migrationB <- na.omit(migrationB)

migration <- migrationA %>%
  inner_join(migrationB,by = "Country") %>%
  pivot_longer(
    -Country,names_to = "Year",values_to = "Total"
  ) %>%
  separate("Year",into = c("Year","Type"),sep = 5)%>%
  mutate(
    Year=str_remove(string = Year,pattern = "\\.$"),
    Type=case_when(Type=="x"~"Inflow",TRUE~'Outflow'))
```

```
selected <- c("Sweden","Norway","Japan",'Korea',"Germany","Netherlands")
migrationplot <- migration %>%
  filter(Country%in%selected) %>%
  ggplot(aes(Year,Total,group=Type))+
  geom_line(aes(color=Type),linetype=5,size=1.1)+
  scale_y_continuous(label=scales::number_format(big.mark = "",suffix = 'K'))+
  scale_color_brewer(name="",type = 'qual',palette = "Set1")+
  labs(x="",y="Migration,in thousands",
       title = "Inflows and Outflows of foreign -born populations in select OECD countries",
       caption = 'chart by @mutwiriian')+
  theme_minimal()+
  theme(
    legend.position = 'bottom',
    legend.text = element_text(face = 'bold',size = 10),
    axis.text = element_text(face = "bold"),
    axis.title.y.left = element_text(face = 'bold',size = 10),
    axis.text.x.bottom = element_text(face = 'bold',size = 10,angle = 30),
    strip.text.x = element_text(face = 'bold',size = 10),
    plot.caption = element_text(size = 11,vjust = 10,hjust = .95)
    )+
  facet_wrap(~Country,nrow = 2,scales = "free")
migrationplot
```



Now lets get into my personal favorite! I downloaded the consumption,treasury ill rate and inflation rate data from the St.Louis Federal Reserve Bank of the United States. After some pre=processing I join all these data into a single dataset which then produces the highly customized plot in the `Economist` magazine style.

```
consumption <- read_csv('E:/Workspace/MacroEcon/realconsumption.csv')
```

```
## Rows: 91 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): DPCERL1A225NBEA
## date (1): DATE
##
## 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.
```

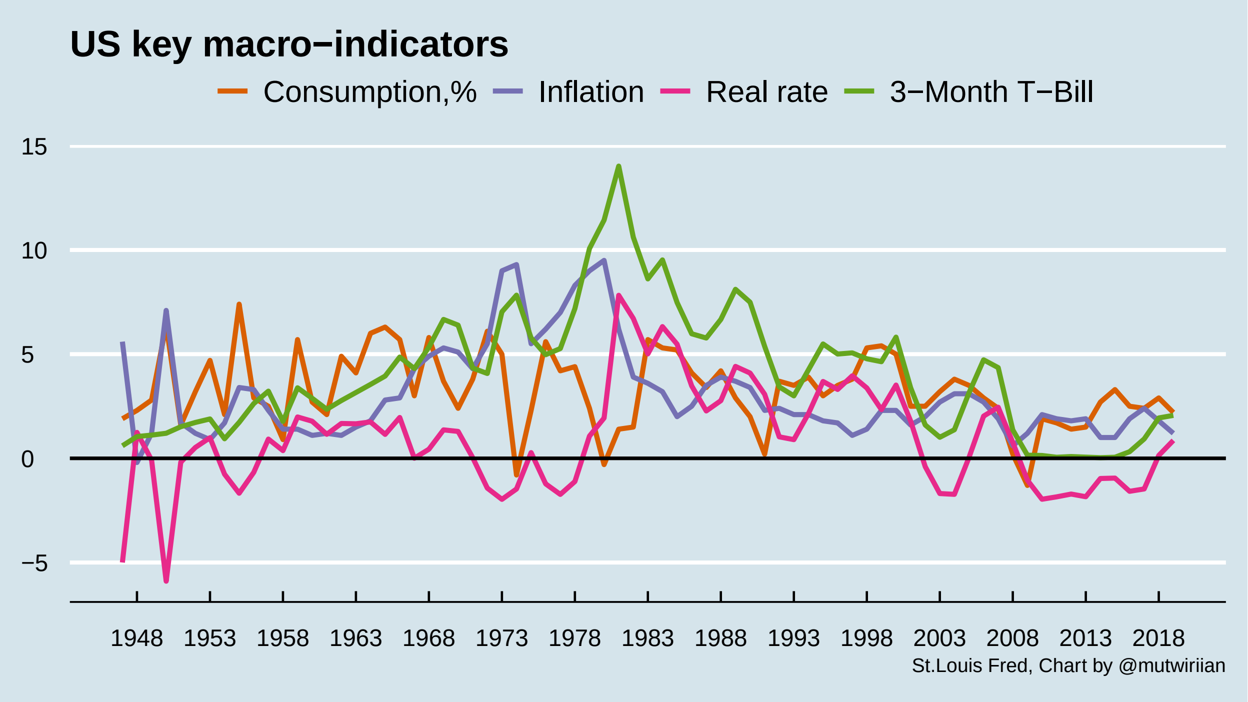
```
colnames(consumption) <- c("Date",'cons_Growth')
consumption$cons_Growth <- as.double(consumption$cons_Growth)

tbill <- read_csv('E:/Workspace/MacroEcon/TB3MS.csv')
```

```
## Rows: 88 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): TB3MS
## date (1): DATE
##
## 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.
```

```
colnames(tbill) <- c('Date','trate')
tbill$trate <- as.double(tbill$trate)
tbill<- tbill %>%
  mutate(trate=round(trate,digits = 3))

deflator <- read_csv('E:/Workspace/MacroEcon/usdeflator.csv')
```

```
## Rows: 91 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): A191RI1A225NBEA
## date (1): DATE
##
## 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.
```

```
colnames(deflator) <- c('Date',"deflator")
deflator <- deflator %>% mutate(deflator=c(deflator[2:91],NA))

rates<- consumption %>%
  inner_join(tbill,by = 'Date') %>%
  inner_join(deflator,by = 'Date') %>%
  filter(Date<'2020-01-01'&Date>='1947-01-01')%>%
  mutate(real_rate=trate-deflator) %>%
  pivot_longer(cols = c(2:5),names_to = 'measure',values_to = 'rate')

p <- rates %>%
  ggplot(aes(Date,rate))+
  geom_line(aes(color=measure),size=1.1)+
  geom_hline(yintercept = 0,size=.8)+
  labs(
    x=NULL,
    y=NULL,
    title = 'US key macro-indicators',
    caption = 'St.Louis Fred, Chart by @mutwiriian'
  )+
  scale_x_date(date_breaks = '5 years',date_labels  = '%Y')+
  scale_color_manual(values = c('#D95F02','#7570B3','#E7298A','#66A61E'),
                     name='',
                     labels=c('Consumption,%','Inflation','Real rate','3-Month T-Bill'))+
  theme_bw()+
  theme(
    axis.text  = element_text(size = 10),
    plot.caption = element_text(size = 12,face = 'bold',vjust = .9,hjust = .01),
    legend.text = element_text(size = 12),
    legend.direction = 'horizontal',
    legend.position = c(.6,.13),
    legend.background = element_blank()
  )+
  #guides(color=guide_legend(nrow = 2))+
  ggthemes::theme_economist()
p
```



```
ggsave('usrates.png',width = 2006,height = 1159,units = 'px',scale = 1.2)
```