

Final Assignment

Efthymios Ioannis Kavour

4/4/2022

Assignment :

- 1) Present the plots of the Data assigned to you.
- 2) In each one of the series find a suitable trend : Apply the polynomial form of trend or the moving average.
- 3) Present the trend-cycle of the series by one of the following options:
 - With substitution or division of the trend
 - Apply the moving average plot to show the cycle of the series.
- 4) Present the cycle of the series in a plot and find the leader of them.
- 5) (Optional) Check the stationary of the series used.

Solutions:

1) Present the plots of the data assigned to you.

The datasets that we have to analyse about Croatia, are Unemployment Rate, Inflation rate, GDP per capita and GINI Index of the mentioned country.

Unemployment rate is calculated by expressing the number of able to work but unemployed people as a percentage of the total number of people in labor force of a country. The labor force of a country is described as the number of people employed and the number of people unemployed.

Inflation is the rate at which the value of a currency is falling and, consequently, the general level of prices for goods and services is rising. Inflation is sometimes classified into three types: 1. Demand-Pull inflation 2. Cost-Push inflation 3. Built-In inflation

Moving forward, GDP can be calculated by adding up all of the money spent by consumers, businesses and government in a given period. It may also be calculated by adding up all of the money received by all the participants in the economy. In either case, the number is an estimate of 'nominal GDP'.

Finally, the Gini Index measures the area between the Lorenz curve and a hypothetical line of absolute equality, expressed as a percentage of the maximum area under the line. Thus a Gini index of 0 represents perfect equality, while an index of 100 implies perfect inequality.

- A Lorenz curve is a graphical representation of the distribution of income or wealth within a population. Lorenz curves graph percentiles of the population against cumulative income or wealth of people at or below that percentile.

Now, that we have an idea what each of the variables represent, we need to load some libraries in order to begin our analysis.

```
library(readxl)
library(dplyr)
library(tidyr)
library(tsibble)
library(ggplot2)
library(fpp3)
library(forecast)
library(fable)
library(fabletools)
library(zoo)
library(slides)
```

Apart from the libraries, as a basic step, we need to load the time series data for the aforementioned variables.

There are some actions needed to bring the data to the correct form, meaning, we have to pivot the excel files and keep only the range of years for which we have data.

For the Unemployment data we have the following transformations

```
unemp_pivot <- unemp %>%
  select(`1991`[YR1991]:ncol(unemp)) %>%
  pivot_longer(cols = everything(), names_to = "Year", values_to = 'Unemployment') %>%
  separate(col = Year, into = 'Year', sep = " ", remove = TRUE) %>%
  drop_na('Unemployment')
```

```
## Warning: Expected 1 pieces. Additional pieces discarded in 180 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

Let us now check if there are any missing values in our datasheet and if everything filled with a value, hopefully (!), we will proceed by transforming the data to a time series object.

```
unemp_pivot %>%
  summarise(sum(is.na("Unemployment")))
```

```
## # A tibble: 1 x 1
##   'sum(is.na("Unemployment"))'
##                               <int>
## 1                               0
```

```
unemp_ts <- tsibble(Year = 1991:2020,
                    Unemployment = unemp_pivot$Unemployment,
                    index = Year)
unemp_ts
```

```
## # A tsibble: 30 x 2 [1Y]
##   Year Unemployment
##   <int>      <dbl>
## 1  1991         11.1
## 2  1992         11.0
```

```
## 3 1993      10.7
## 4 1994      10.6
## 5 1995      10.2
## 6 1996       9.68
## 7 1997       9.71
## 8 1998      11.4
## 9 1999      13.6
## 10 2000     16.1
## # ... with 20 more rows
```

Same procedure is going to be followed for the rest of the data sets! Hence, for inflation,

```
infl_pivot <- infl %>%
  select(`1986` [YR1986]` : ncol(infl)) %>%
  drop_na(everything()) %>%
  pivot_longer(col = everything(), names_to = "Year", values_to = "Inflation") %>%
  separate(col = "Year", into = "Year", sep = " ")
```

```
## Warning: Expected 1 pieces. Additional pieces discarded in 35 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
infl_pivot %>%
  summarise(sum(is.na("Inflation")))
```

```
## # A tibble: 1 x 1
##   `sum(is.na("Inflation"))`
##               <int>
## 1                      0
```

```
infl_ts <- tsibble(Year = 1986:2020,
                  Inflation = infl_pivot$Inflation,
                  index = Year)

infl_ts
```

```
## # A tsibble: 35 x 2 [1Y]
##   Year Inflation
##   <int>    <dbl>
## 1 1986      50
## 2 1987    133.
## 3 1988    186.
## 4 1989   1400
## 5 1990    500
## 6 1991    122.
## 7 1992   625.
## 8 1993   1500
## 9 1994    107.
## 10 1995     3.95
## # ... with 25 more rows
```

GDP per capita,

```
gdp_pivot <- gdp %>%
  select(`1995` [YR1995]:ncol(gdp)) %>%
  drop_na(everything()) %>%
  pivot_longer(col = everything(), names_to = 'Year', values_to = 'Gdp per Capita ($)') %>%
  separate(col = 'Year', into = 'Year', sep = ' ')
```

```
## Warning: Expected 1 pieces. Additional pieces discarded in 26 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
gdp_pivot %>%
  summarise(sum(is.na(`Gdp per Capita ($)`)))
```

```
## # A tibble: 1 x 1
##   `sum(is.na(`Gdp per Capita ($)`)`)'
##                                     <int>
## 1                                     0
```

```
gdp_ts <- tsibble(Year = 1995:2020,
                  GDP = gdp_pivot$`Gdp per Capita ($)`,
                  index = Year)
gdp_ts
```

```
## # A tsibble: 26 x 2 [1Y]
##   Year   GDP
##   <int> <dbl>
## 1  1995 4917.
## 2  1996 5283.
## 3  1997 5312.
## 4  1998 5691.
## 5  1999 5247.
## 6  2000 4888.
## 7  2001 5413.
## 8  2002 6293.
## 9  2003 8130.
## 10 2004 9747.
## # ... with 16 more rows
```

and finally GINI,

```
gini_pivot <- gini %>%
  select(X, X.1) %>%
  rename(Year = X, `Gini Index` = X.1) %>%
  slice(2:43)

gini_pivot %>%
  summarise(sum(is.na(`Gini Index`)))
```

```
##   sum(is.na(`Gini Index`))
## 1                          0
```

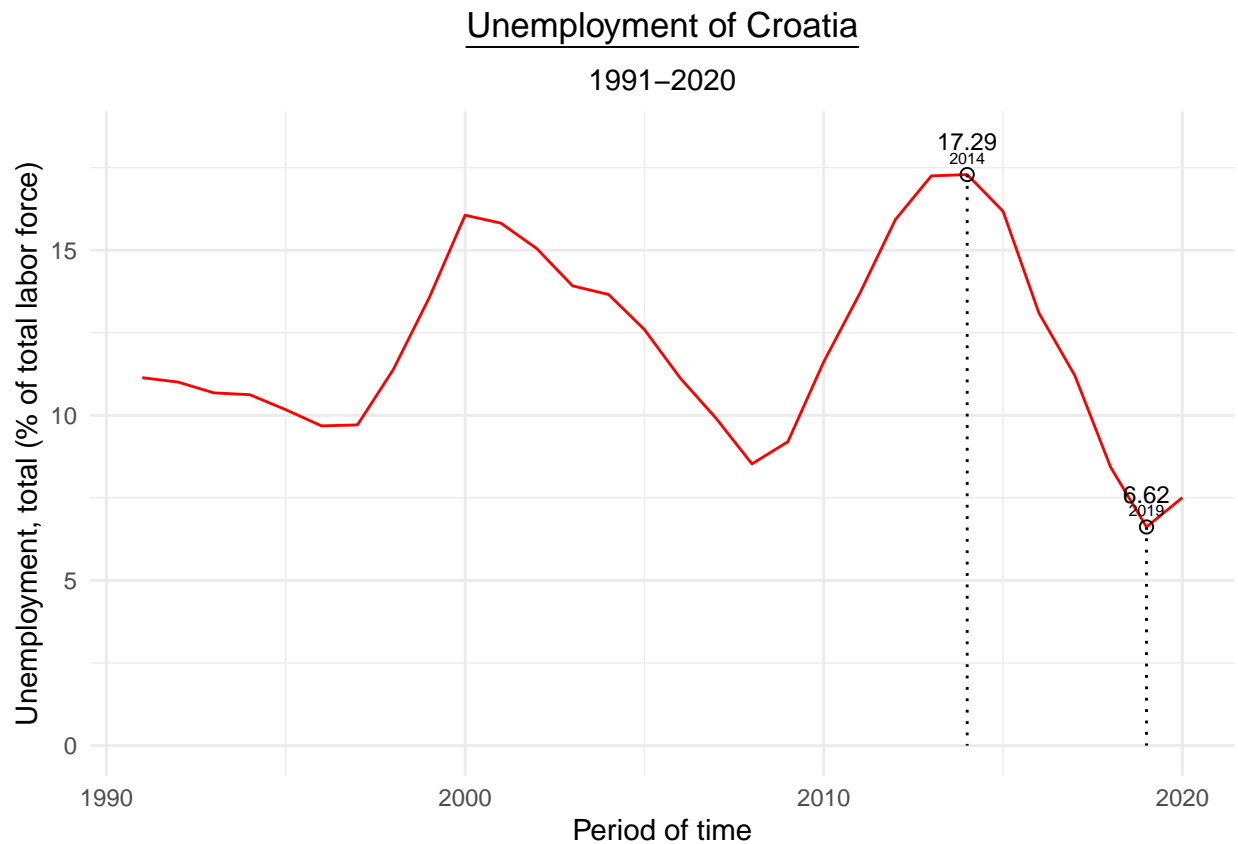
```
gini_ts <- tsibble(Year = 1980:2021,
                  GINI = as.double(gini_pivot$`Gini Index`),
                  index = Year)
gini_ts$GINI
```

```
## [1] 0.0418 0.0418 0.0418 0.0418 0.0431 0.0428 0.0434 0.0409 0.0411 0.0461
## [11] 0.0494 0.0531 0.0578 0.0634 0.0682 0.0736 0.0730 0.0772 0.0733 0.0772
## [21] 0.0741 0.0754 0.0789 0.0775 0.0834 0.0929 0.0968 0.1016 0.1009 0.0889
## [31] 0.0921 0.1015 0.0986 0.1063 0.1078 0.1126 0.1018 0.1053 0.1039 0.1020
## [41] 0.1024 0.1024
```

Initially, what we need is to see the raw data, meaning the plot of the time series data. As a result, separately, we have:

```
autoplot(unemp_ts, Unemployment, col = 'red') +
  labs(title = expression(underline('Unemployment of Croatia'))),
  subtitle = '1991-2020',
  x = 'Period of time',
  y = 'Unemployment, total (% of total labor force)' +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.subtitle = element_text(hjust = 0.5)) +
  geom_point(data = unemp_ts[which.min(unemp_ts$Unemployment)], size = 2, shape = 1) +
  annotate(geom = 'text',
    x = unemp_ts$Year[unemp_ts$Unemployment == min(unemp_ts$Unemployment)],
    y = min(unemp_ts$Unemployment) + 1,
    label = round(min(unemp_ts$Unemployment), 2),
    size = 3) +
  annotate(geom = 'text',
    x = unemp_ts$Year[unemp_ts$Unemployment == min(unemp_ts$Unemployment)],
    y = min(unemp_ts$Unemployment) + 0.5,
    label = unemp_ts$Year[unemp_ts$Unemployment == min(unemp_ts$Unemployment)],
    size = 2) +
  annotate(geom = 'segment',
    x = unemp_ts$Year[unemp_ts$Unemployment == min(unemp_ts$Unemployment)],
    xend = unemp_ts$Year[unemp_ts$Unemployment == min(unemp_ts$Unemployment)],
    y = min(unemp_ts$Unemployment),
    yend = 0,
    colour = 'black', linetype = 'dotted') +
  geom_point(data = unemp_ts[which.max(unemp_ts$Unemployment)], size = 2, shape = 1) +
  annotate(geom = 'text',
    x = unemp_ts$Year[unemp_ts$Unemployment == max(unemp_ts$Unemployment)],
    y = max(unemp_ts$Unemployment) + 1,
    label = round(max(unemp_ts$Unemployment), 2),
    size = 3) +
  annotate(geom = 'text',
    x = unemp_ts$Year[unemp_ts$Unemployment == max(unemp_ts$Unemployment)],
    y = max(unemp_ts$Unemployment) + 0.5,
    label = unemp_ts$Year[unemp_ts$Unemployment == max(unemp_ts$Unemployment)],
    size = 2) +
  annotate(geom = 'segment',
    x = unemp_ts$Year[unemp_ts$Unemployment == max(unemp_ts$Unemployment)],
    xend = unemp_ts$Year[unemp_ts$Unemployment == max(unemp_ts$Unemployment)],
```

```
y = max(unemp_ts$Unemployment),
yend = 0,
colour = 'black', linetype = 'dotted')
```



Moving forward, similarly for the rest of the variables we have the following:

```
autoplot(infl_ts, Infaltion, col = 'red') +
  labs(title = expression(underline('Inflation rate for consumer prices of Croatia')),
        subtitle = '1991-2020',
        x = 'Period of time',
        y = 'Inflation rate') +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.subtitle = element_text(hjust = 0.5)) +
  geom_point(data = infl_ts[which.min(infl_ts$Infaltion),], size = 2, shape = 1) +
  annotate(geom = 'text',
          x = infl_ts$Year[infl_ts$Infaltion == min(infl_ts$Infaltion)],
          y = min(infl_ts$Infaltion) + 130,
          label = round(min(infl_ts$Infaltion),2),
          size = 3) +
  annotate(geom = 'text',
          x = infl_ts$Year[infl_ts$Infaltion == min(infl_ts$Infaltion)],
          y = min(infl_ts$Infaltion) + 80,
          label = infl_ts$Year[infl_ts$Infaltion == min(infl_ts$Infaltion)],
          size = 2) +
  annotate(geom = 'segment',
```

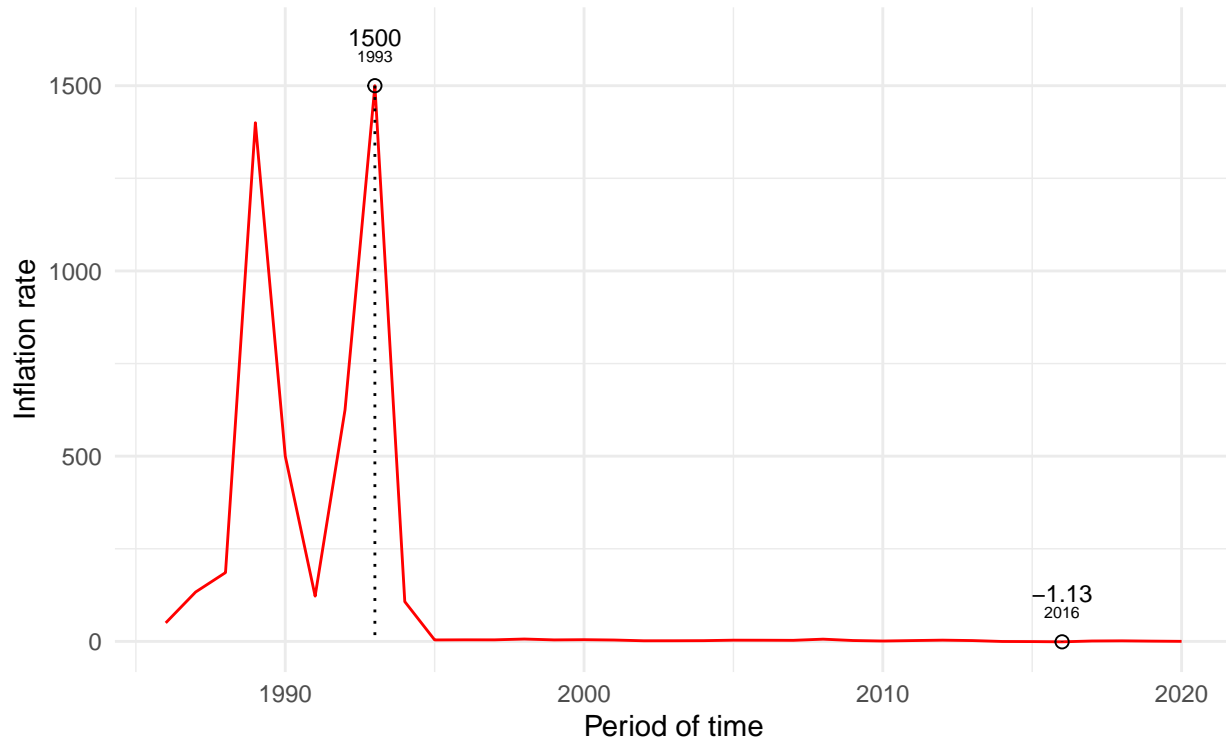
```

x = infl_ts$Year[infl_ts$Infaltion == min(infl_ts$Infaltion)],
xend = infl_ts$Year[infl_ts$Infaltion == min(infl_ts$Infaltion)],
y = min(infl_ts$Infaltion),
yend = 0,
colour = 'black', linetype = 'dotted') +
geom_point(data = infl_ts[which.max(infl_ts$Infaltion),], size = 2, shape = 1) +
annotate(geom = 'text',
x = infl_ts$Year[infl_ts$Infaltion == max(infl_ts$Infaltion)],
y = max(infl_ts$Infaltion) + 130,
label = round(max(infl_ts$Infaltion),2),
size = 3) +
annotate(geom = 'text',
x = infl_ts$Year[infl_ts$Infaltion == max(infl_ts$Infaltion)],
y = max(infl_ts$Infaltion) + 80,
label = infl_ts$Year[infl_ts$Infaltion == max(infl_ts$Infaltion)],
size = 2) +
annotate(geom = 'segment',
x = infl_ts$Year[infl_ts$Infaltion == max(infl_ts$Infaltion)],
xend = infl_ts$Year[infl_ts$Infaltion == max(infl_ts$Infaltion)],
y = max(infl_ts$Infaltion),
yend = 0,
colour = 'black', linetype = 'dotted')

```

Inflation rate for consumer prices of Croatia

1991–2020



```

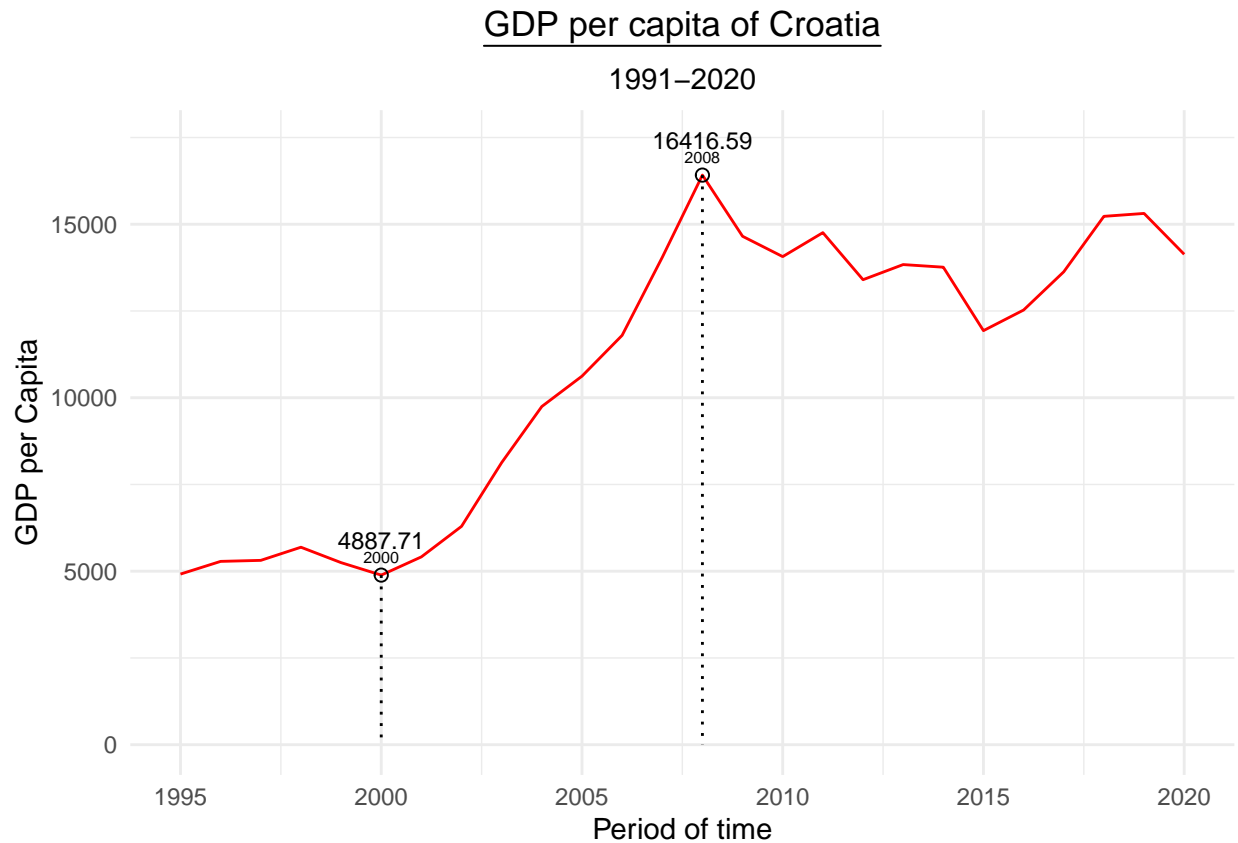
autoplot(gdp_ts, GDP, col = 'red') +
  labs(title = expression(underline('GDP per capita of Croatia'))),

```

```

    subtitle = '1991-2020',
    x = 'Period of time',
    y = 'GDP per Capita') +
theme_minimal() +
theme(plot.title = element_text(hjust = 0.5)) +
theme(plot.subtitle = element_text(hjust = 0.5)) +
geom_point(data = gdp_ts[which.min(gdp_ts$GDP),], size = 2, shape = 1) +
annotate(geom = 'text',
         x = gdp_ts$Year[gdp_ts$GDP == min(gdp_ts$GDP)],
         y = min(gdp_ts$GDP) + 1000,
         label = round(min(gdp_ts$GDP),2),
         size = 3) +
annotate(geom = 'text',
         x = gdp_ts$Year[gdp_ts$GDP == min(gdp_ts$GDP)],
         y = min(gdp_ts$GDP) + 500,
         label = gdp_ts$Year[gdp_ts$GDP == min(gdp_ts$GDP)],
         size = 2) +
annotate(geom = 'segment',
         x = gdp_ts$Year[gdp_ts$GDP == min(gdp_ts$GDP)],
         xend = gdp_ts$Year[gdp_ts$GDP == min(gdp_ts$GDP)],
         y = min(gdp_ts$GDP),
         yend = 0,
         colour = 'black', linetype = 'dotted') +
geom_point(data = gdp_ts[which.max(gdp_ts$GDP),], size = 2, shape = 1) +
annotate(geom = 'text',
         x = gdp_ts$Year[gdp_ts$GDP == max(gdp_ts$GDP)],
         y = max(gdp_ts$GDP) + 1000,
         label = round(max(gdp_ts$GDP),2),
         size = 3) +
annotate(geom = 'text',
         x = gdp_ts$Year[gdp_ts$GDP == max(gdp_ts$GDP)],
         y = max(gdp_ts$GDP) + 500,
         label = gdp_ts$Year[gdp_ts$GDP == max(gdp_ts$GDP)],
         size = 2) +
annotate(geom = 'segment',
         x = gdp_ts$Year[gdp_ts$GDP == max(gdp_ts$GDP)],
         xend = gdp_ts$Year[gdp_ts$GDP == max(gdp_ts$GDP)],
         y = max(gdp_ts$GDP),
         yend = 0,
         colour = 'black', linetype = 'dotted')

```

```

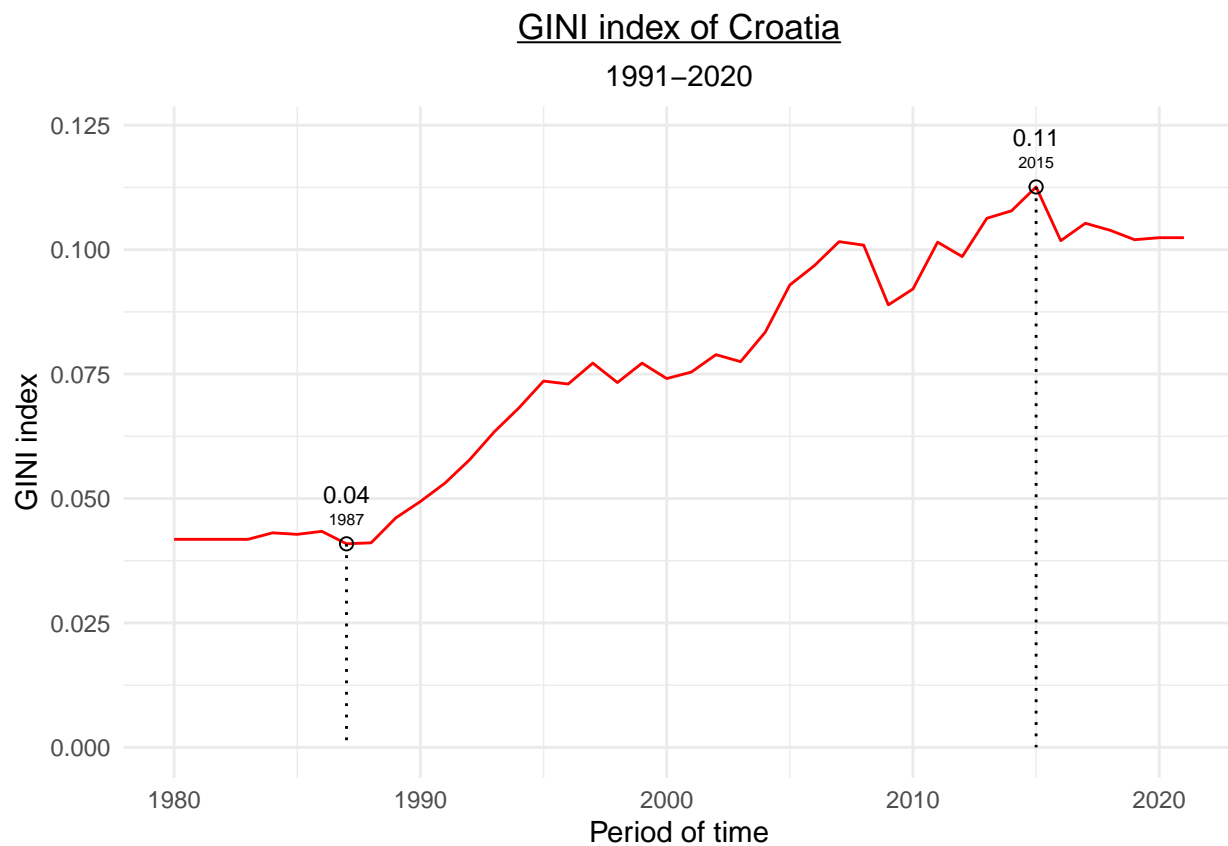
autoplot(gini_ts, GINI, col = 'red') +
  labs(title = expression(underline('GINI index of Croatia')),
       subtitle = '1991-2020',
       x = 'Period of time',
       y = 'GINI index') +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.subtitle = element_text(hjust = 0.5)) +
  geom_point(data = gini_ts[which.min(gini_ts$GINI),], size = 2, shape = 1) +
  annotate(geom = 'text',
         x = gini_ts$Year[gini_ts$GINI == min(gini_ts$GINI)],
         y = min(gini_ts$GINI) + 0.01,
         label = round(min(gini_ts$GINI), 2),
         size = 3) +
  annotate(geom = 'text',
         x = gini_ts$Year[gini_ts$GINI == min(gini_ts$GINI)],
         y = min(gini_ts$GINI) + 0.005,
         label = gini_ts$Year[gini_ts$GINI == min(gini_ts$GINI)],
         size = 2) +
  annotate(geom = 'segment',
         x = gini_ts$Year[gini_ts$GINI == min(gini_ts$GINI)],
         xend = gini_ts$Year[gini_ts$GINI == min(gini_ts$GINI)],
         y = min(gini_ts$GINI),
         yend = 0,
         colour = 'black', linetype = 'dotted') +
  geom_point(data = gini_ts[which.max(gini_ts$GINI),], size = 2, shape = 1) +

```

```

annotate(geom = 'text',
  x = gini_ts$Year[gini_ts$GINI == max(gini_ts$GINI)],
  y = max(gini_ts$GINI) + 0.01,
  label = round(max(gini_ts$GINI),2),
  size = 3) +
annotate(geom = 'text',
  x = gini_ts$Year[gini_ts$GINI == max(gini_ts$GINI)],
  y = max(gini_ts$GINI) + 0.005,
  label = gini_ts$Year[gini_ts$GINI == max(gini_ts$GINI)],
  size = 2) +
annotate(geom = 'segment',
  x = gini_ts$Year[gini_ts$GINI == max(gini_ts$GINI)],
  xend = gini_ts$Year[gini_ts$GINI == max(gini_ts$GINI)],
  y = max(gini_ts$GINI),
  yend = 0,
  colour = 'black', linetype = 'dotted')

```



2) In each one of the series find a suitable trend : Apply the polynomial form of trend or the moving average.

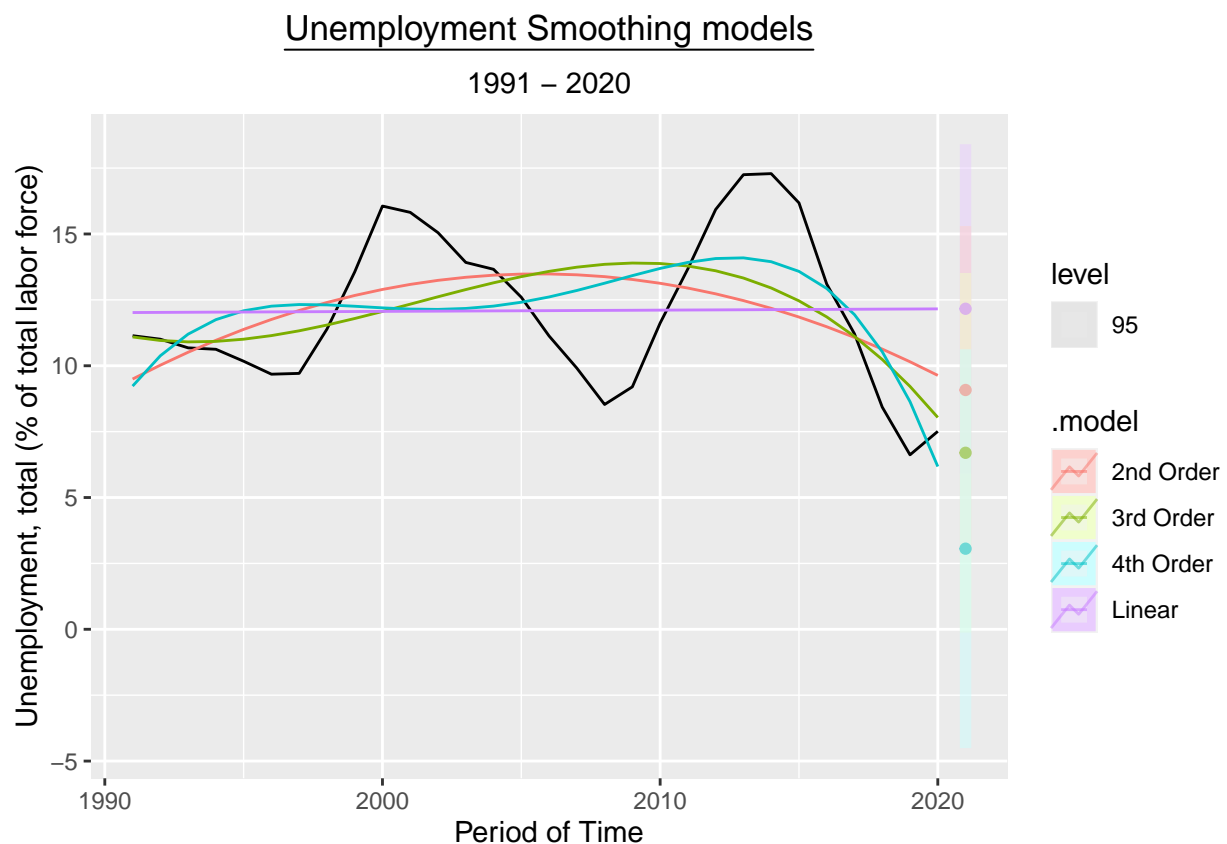
Now, as for the second part of this assignment we have to construct and plot the polynomial form of the trend or the moving average. Here in one plot, we are going to construct the first, second, up until fifth order polynomial trend, and plot them. What we expect is that the linear form is going to be a straight line, the second is going to give us back some small curve, and as we move forward the polynomial is going to

fit better and better to our raw data. Apart from the requested information, here a small forecast for two years is going to be provided with 95% statistical significance.

```
fit_unem <- unemp_ts %>%
  model(
    Linear = TSLM(Unemployment ~ trend()),
    `2nd Order` = TSLM(Unemployment ~ trend() + I(trend()^2)),
    `3rd Order` = TSLM(Unemployment ~ trend() + I(trend()^2) + I(trend()^3)),
    `4th Order` = TSLM(Unemployment ~ trend() + I(trend()^2) + I(trend()^3) + I(trend()^4))

fore_unem1 <- fit_unem %>%
  forecast(h = '1 years')

unemp_ts %>%
  autoplot(Unemployment) +
  geom_line(data = fitted(fit_unem),
    aes(y = .fitted, color = .model)) +
  autolayer(fore_unem1, alpha = 0.5, level = 95) +
  labs(title = expression(underline('Unemployment Smoothing models')),
    subtitle = '1991 - 2020',
    x = 'Period of Time',
    y = 'Unemployment, total (% of total labor force)' +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.subtitle = element_text(hjust = 0.5))
```



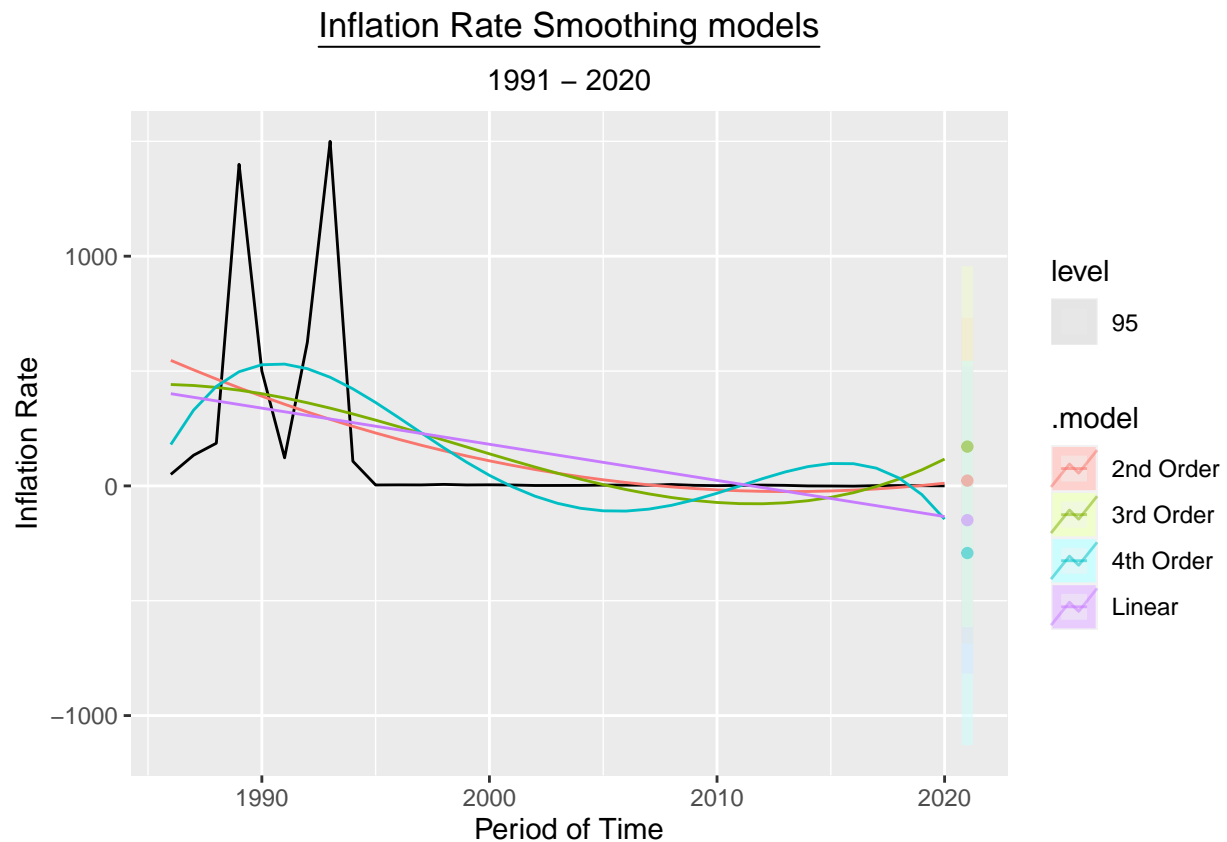
What we have to do now is repeat the process we just completed for the variable of Unemployment, for the

rest of the variables that we analyze. As a result, we have the following inflation rate of Croatia :

```
fit_inf <- infl_ts %>%
  model(
    Linear = TSLM(Infaltion ~ trend()),
    `2nd Order` = TSLM(Infaltion ~ trend() + I(trend()^2)),
    `3rd Order` = TSLM(Infaltion ~ trend() + I(trend()^2) + I(trend()^3)),
    `4th Order` = TSLM(Infaltion ~ trend() + I(trend()^2) + I(trend()^3) + I(trend()^4)))

fore_infl1 <- fit_inf %>%
  forecast(h = '1 years')

infl_ts %>%
  autoplot(Infaltion) +
  geom_line(data = fitted(fit_inf),
    aes(y = .fitted, color = .model)) +
  autolayer(fore_infl1, alpha = 0.5, level = 95) +
  labs(title = expression(underline('Inflation Rate Smoothing models')),
    subtitle = '1991 - 2020',
    x = 'Period of Time',
    y = 'Inflation Rate') +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.subtitle = element_text(hjust = 0.5))
```



Furthermore, for GDP per capita, we have the following :

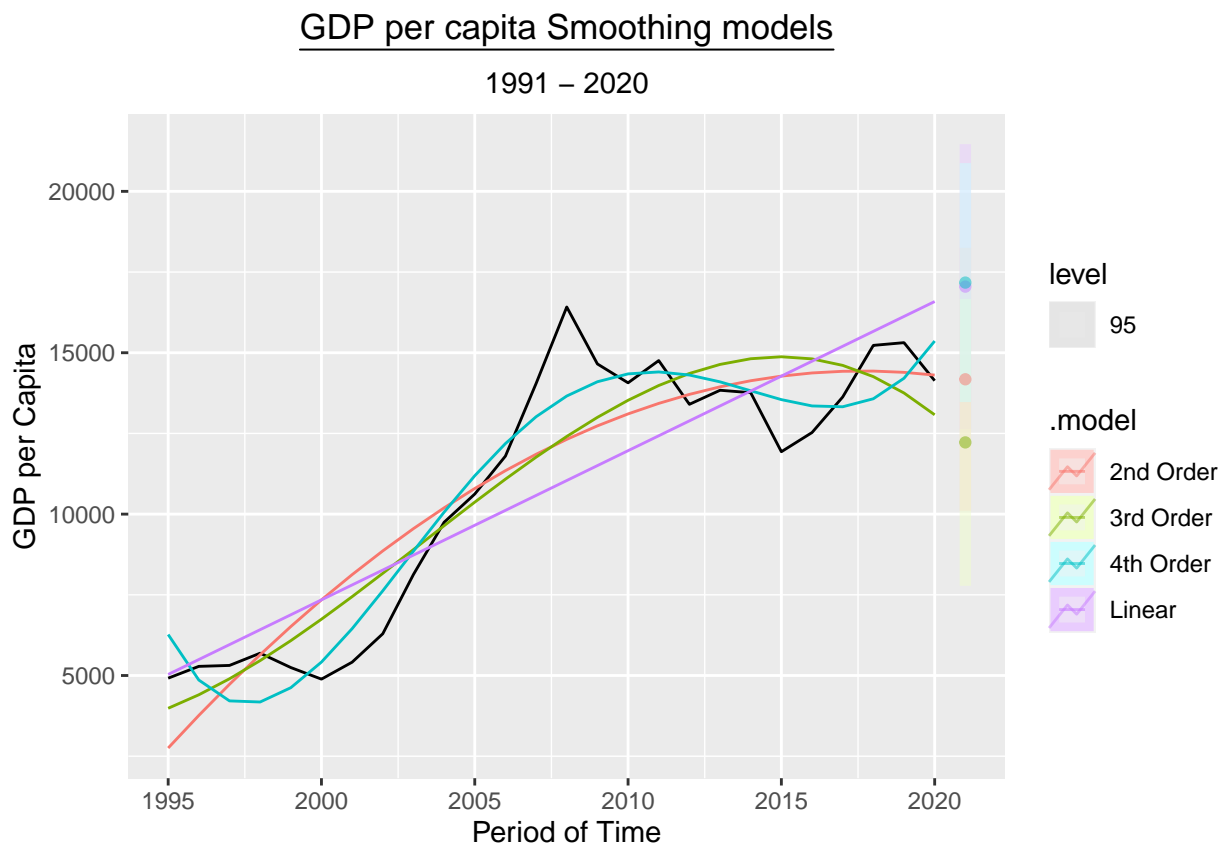
```

fit_gdp <- gdp_ts %>%
  model(
    Linear = TSLM(GDP ~ trend()),
    `2nd Order` = TSLM(GDP ~ trend() + I(trend()^2)),
    `3rd Order` = TSLM(GDP ~ trend() + I(trend()^2) + I(trend()^3)),
    `4th Order` = TSLM(GDP ~ trend() + I(trend()^2) + I(trend()^3) + I(trend()^4))

fore_gdp1 <- fit_gdp %>%
  forecast(h = '1 years')

gdp_ts %>%
  autoplot(GDP) +
  geom_line(data = fitted(fit_gdp),
            aes(y = .fitted, color = .model)) +
  autolayer(fore_gdp1, alpha = 0.5, level = 95) +
  labs(title = expression(underline('GDP per capita Smoothing models')),
        subtitle = '1991 - 2020',
        x = 'Period of Time',
        y = 'GDP per Capita') +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.subtitle = element_text(hjust = 0.5))

```



Finally, we have the GINI index variable as follows :

```

fit_gini <- gini_ts %>%
  model(

```

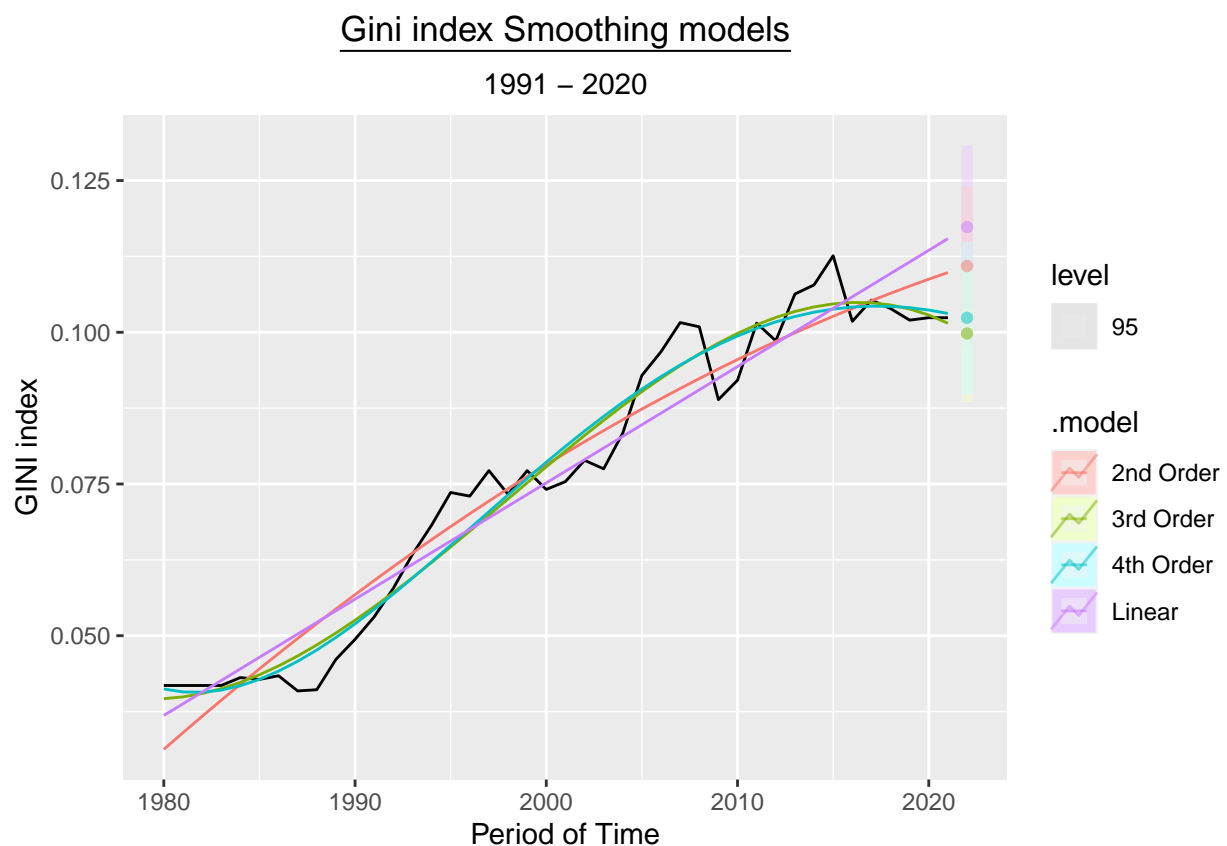
```

Linear = TSLM(GINI ~ trend()),
`2nd Order` = TSLM(GINI ~ trend() + I(trend()^2)),
`3rd Order` = TSLM(GINI ~ trend() + I(trend()^2) + I(trend()^3)),
`4th Order` = TSLM(GINI ~ trend() + I(trend()^2) + I(trend()^3) + I(trend()^4))

fore_gini1 <- fit_gini %>%
  forecast(h = '1 years')

gini_ts %>%
  autoplot(GINI) +
  geom_line(data = fitted(fit_gini),
            aes(y = .fitted, color = .model)) +
  autolayer(fore_gini1, alpha = 0.5, level = 95) +
  labs(title = expression(underline('Gini index Smoothing models')),
        subtitle = '1991 - 2020',
        x = 'Period of Time',
        y = 'GINI index') +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.subtitle = element_text(hjust = 0.5))

```



3) Present the trend-cycle of the series by one of the following options:

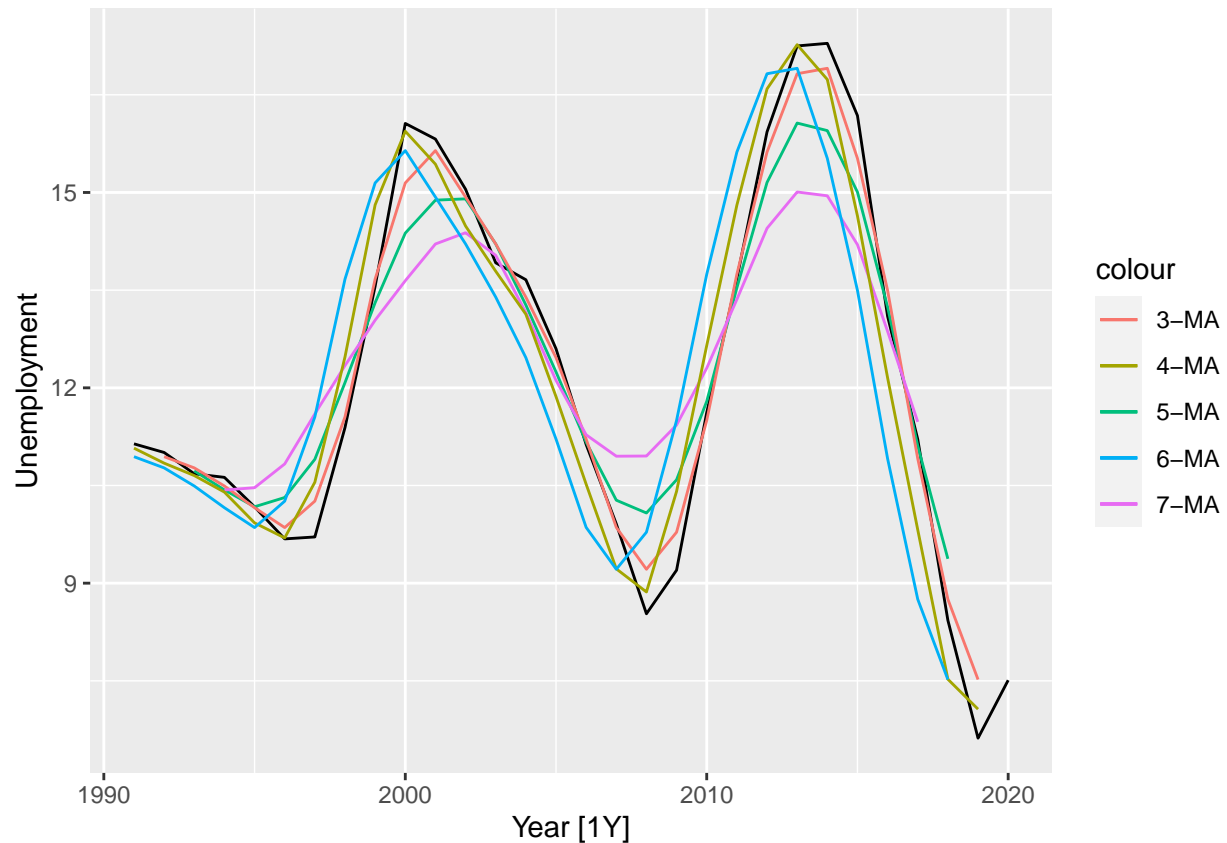
- With substitution or division of the trend
- Apply the moving average plot to show the cycle of the series.

As for this third question here, we are going to apply the moving average for each one of the data provided and present our findings with the help of the autoplot function. We choose to continue with the exact same turn of each dataset. Hence for the unemployment we have 30 different data points. We are going to compute the moving average of the moving average in order to have symmetry around the mean. Initially, we are going to present, a plot with raw data, as well as the moving average with different steps. Later on, we are going to select one of them and proceed by calculating and presenting the moving average of the moving average as described earlier.

```
colors <- c('Raw' = 'black', '3-MA' = 'red', '4-MA' = 'green', '5-MA' = 'blue', '6-MA' = 'yellow', '7-MA' = 'purple')

unemp_ts %>%
  mutate(
    `3-MA` = slide_dbl(Unemployment, mean,
                      .before = 1, .after = 1, .complete = TRUE),
    `5-MA` = slide_dbl(Unemployment, mean,
                      .before = 2, .after = 2, .complete = TRUE),
    `7-MA` = slide_dbl(Unemployment, mean,
                      .before = 3, .after = 3, .complete = TRUE),
    `4-MA` = slide_dbl(Unemployment, mean,
                      .start = 2, .after = 1, .complete = TRUE),
    `6-MA` = slide_dbl(Unemployment, mean,
                      .start = 3, .after = 2, .complete = TRUE),
  ) %>%
  autoplot() +
  geom_line(aes(y = `5-MA`, color = '5-MA')) +
  geom_line(aes(y = `3-MA`, color = '3-MA')) +
  geom_line(aes(y = `7-MA`, color = '7-MA')) +
  geom_line(aes(y = `4-MA`, color = '4-MA')) +
  geom_line(aes(y = `6-MA`, color = '6-MA'))
```

```
## Plot variable not specified, automatically selected '.vars = Unemployment'
```



```
labs(title = expression(underline('Unemployment of Croatia')),
      subtitle = '1991-2020',
      x = 'Period of time',
      y = 'Unemployment, total (% of total labor force)',
      color = "Legend") +
scale_color_manual(values = colors) +
theme(plot.title = element_text(hjust = 0.5)) +
theme(plot.subtitle = element_text(hjust = 0.5))
```

```
## NULL
```

```
colors <- c('Raw' = 'black', '3-MA' = 'red', '4-MA' = 'green', '5-MA' = 'blue', '6-MA' = 'yellow', '7-MA' = 'magenta')

infl_ts %>%
  mutate(
    `3-MA` = slide_dbl(Inflation, mean,
                      .before = 1, .after = 1, .complete = TRUE),
    `5-MA` = slide_dbl(Inflation, mean,
                      .before = 2, .after = 2, .complete = TRUE),
    `7-MA` = slide_dbl(Inflation, mean,
                      .before = 3, .after = 3, .complete = TRUE),
    `4-MA` = slide_dbl(Inflation, mean,
                      .start = 2, .after = 1, .complete = TRUE),
    `6-MA` = slide_dbl(Inflation, mean,
                      .start = 3, .after = 2, .complete = TRUE)
```

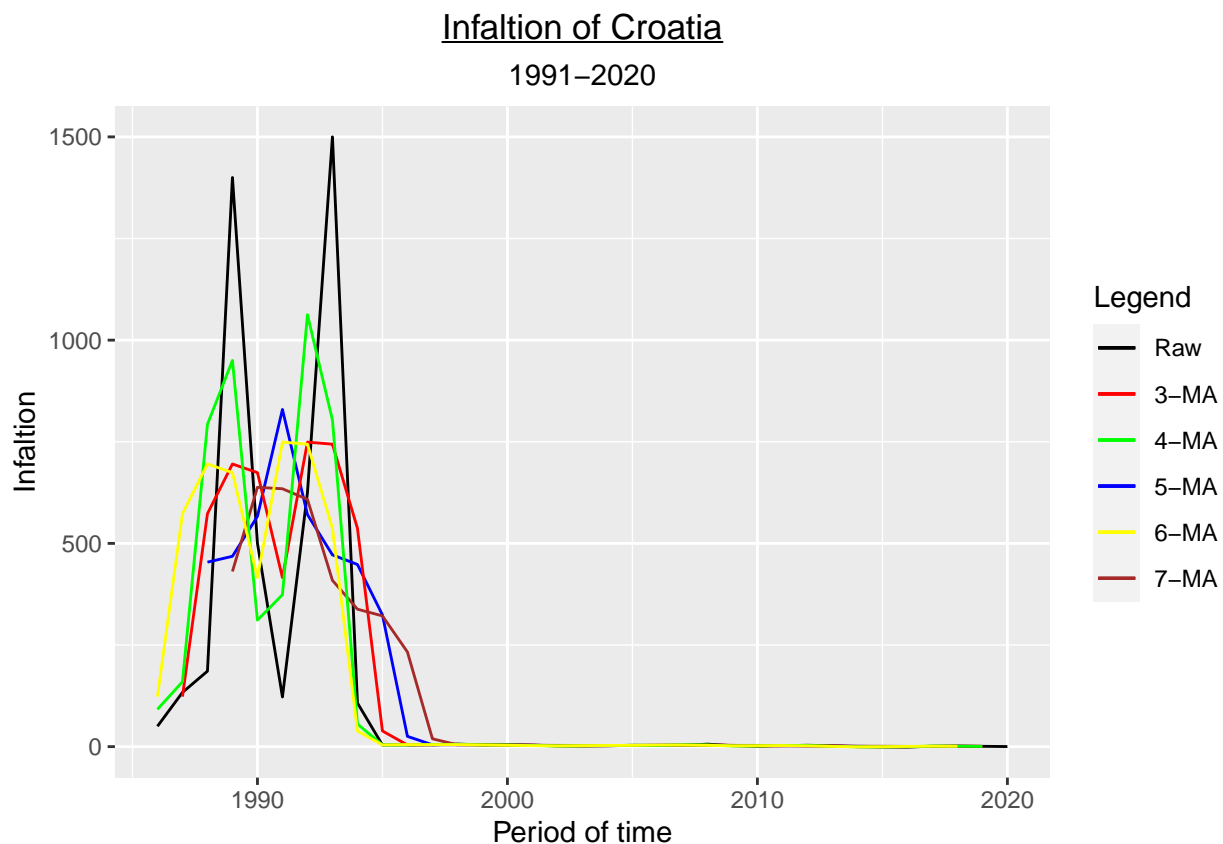


```

) %>%
  autoplot() +
  geom_line(aes(y = `5-MA`, color = '5-MA')) +
  geom_line(aes(y = `3-MA`, color = '3-MA')) +
  geom_line(aes(y = `7-MA`, color = '7-MA')) +
  geom_line(aes(y = `4-MA`, color = '4-MA')) +
  geom_line(aes(y = `6-MA`, color = '6-MA')) +
  labs(title = expression(underline('Infaltion of Croatia')),
       subtitle = '1991-2020',
       x = 'Period of time',
       y = 'Infaltion',
       color = "Legend") +
  scale_color_manual(values = colors) +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.subtitle = element_text(hjust = 0.5))

```

Plot variable not specified, automatically selected '.vars = Infaltion'



```

colors <- c('Raw' = 'black', '3-MA' = 'red', '4-MA' = 'green', '5-MA' = 'blue', '6-MA' = 'yellow', '7-MA' = 'brown')

gdp_ts %>%
  mutate(
    `3-MA` = slide_dbl(GDP, mean,
                      .before = 1, .after = 1, .complete = TRUE),

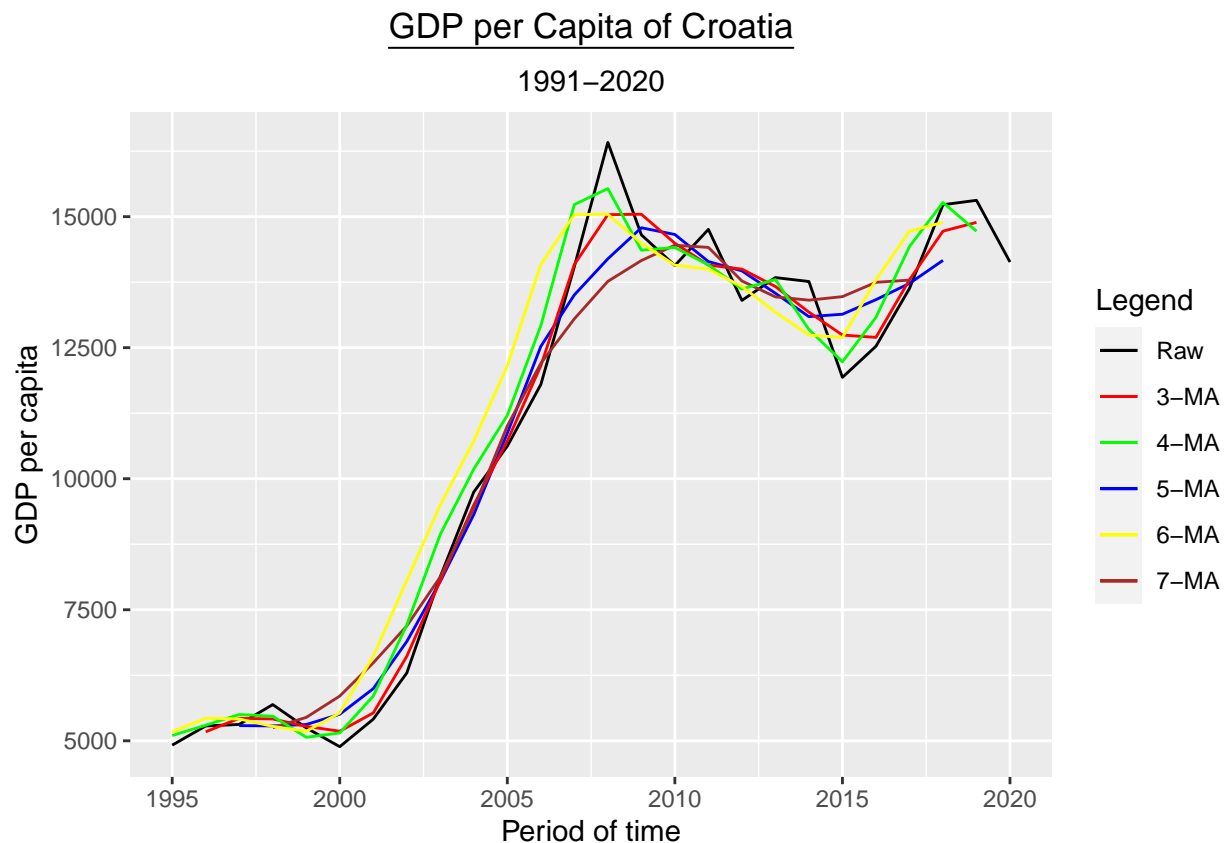
```

```

`5-MA` = slide_dbl(GDP, mean,
  .before = 2, .after = 2, .complete = TRUE),
`7-MA` = slide_dbl(GDP, mean,
  .before = 3, .after = 3, .complete = TRUE),
`4-MA` = slide_dbl(GDP, mean,
  .start = 2, .after = 1, .complete = TRUE),
`6-MA` = slide_dbl(GDP, mean,
  .start = 3, .after = 2, .complete = TRUE)
) %>%
autoplot() +
geom_line(aes(y = `5-MA`, color = '5-MA')) +
geom_line(aes(y = `3-MA`, color = '3-MA')) +
geom_line(aes(y = `7-MA`, color = '7-MA')) +
geom_line(aes(y = `4-MA`, color = '4-MA')) +
geom_line(aes(y = `6-MA`, color = '6-MA')) +
labs(title = expression(underline('GDP per Capita of Croatia')),
  subtitle = '1991-2020',
  x = 'Period of time',
  y = 'GDP per capita',
  color = "Legend") +
scale_color_manual(values = colors) +
theme(plot.title = element_text(hjust = 0.5)) +
theme(plot.subtitle = element_text(hjust = 0.5))

```

Plot variable not specified, automatically selected '.vars = GDP'



```

colors <- c('Raw' = 'black', '3-MA' = 'red', '4-MA' = 'green', '5-MA' = 'blue', '6-MA' = 'yellow', '7-MA' = 'purple')

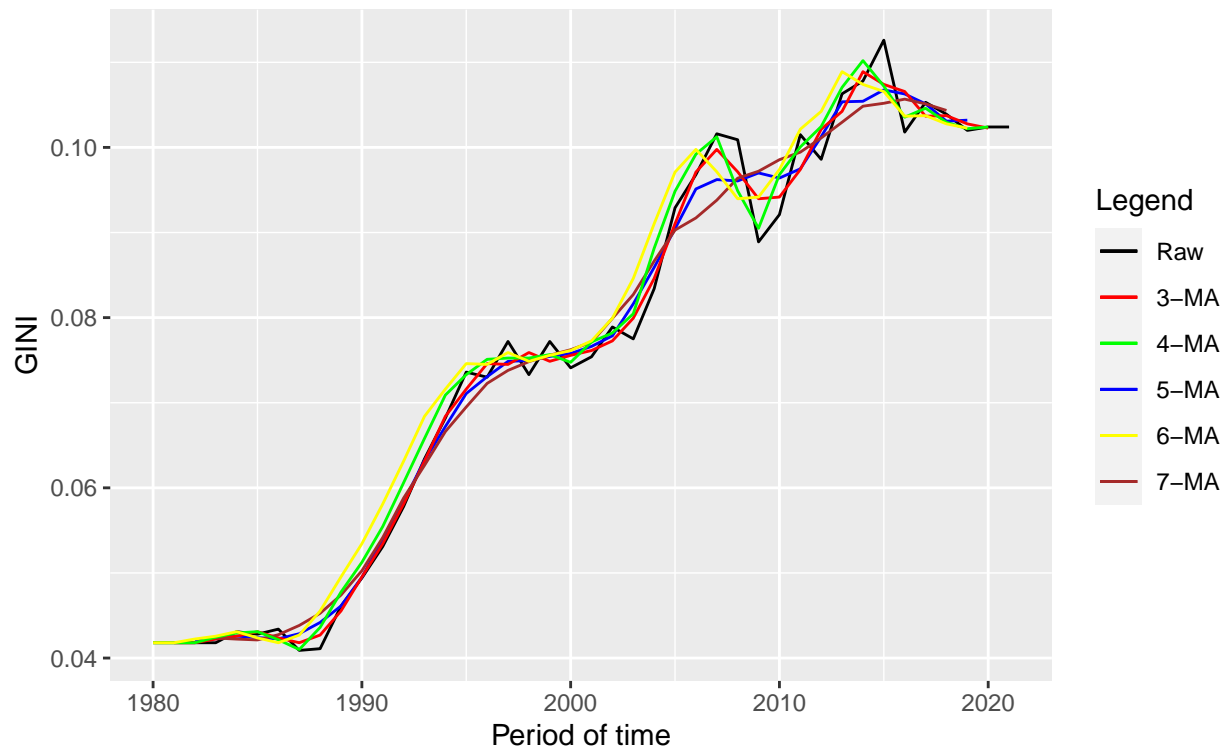
gini_ts %>%
  mutate(
    `3-MA` = slide_dbl(GINI, mean,
                       .before = 1, .after = 1, .complete = TRUE),
    `5-MA` = slide_dbl(GINI, mean,
                       .before = 2, .after = 2, .complete = TRUE),
    `7-MA` = slide_dbl(GINI, mean,
                       .before = 3, .after = 3, .complete = TRUE),
    `4-MA` = slide_dbl(GINI, mean,
                       .start = 2, .after = 1, .complete = TRUE),
    `6-MA` = slide_dbl(GINI, mean,
                       .start = 3, .after = 2, .complete = TRUE)
  ) %>%
  autoplot() +
  geom_line(aes(y = `5-MA`, color = '5-MA')) +
  geom_line(aes(y = `3-MA`, color = '3-MA')) +
  geom_line(aes(y = `7-MA`, color = '7-MA')) +
  geom_line(aes(y = `4-MA`, color = '4-MA')) +
  geom_line(aes(y = `6-MA`, color = '6-MA')) +
  labs(title = expression(underline('GINI index of Croatia')),
       subtitle = '1991-2020',
       x = 'Period of time',
       y = 'GINI',
       color = "Legend") +
  scale_color_manual(values = colors) +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme(plot.subtitle = element_text(hjust = 0.5))

```

```
## Plot variable not specified, automatically selected '.vars = GINI'
```

GINI index of Croatia

1991–2020



```
unemp_ts %>%
  mutate(
    Unemp_7MA = slide_dbl(Unemployment, mean,
                          .before = 3, .after = 3, .complete = TRUE))
```

```
## # A tibble: 30 x 3 [1Y]
##   Year Unemployment Unemp_7MA
##   <int>      <dbl>      <dbl>
## 1 1991      11.1        NA
## 2 1992      11.0        NA
## 3 1993      10.7        NA
## 4 1994      10.6      10.4
## 5 1995      10.2      10.5
## 6 1996       9.68     10.8
## 7 1997       9.71     11.6
## 8 1998      11.4     12.3
## 9 1999      13.6     13.0
## 10 2000      16.1     13.6
## # ... with 20 more rows
```

```
infl_ts %>%
  mutate(
    Infl_7MA = slide_dbl(Inflation, mean,
                          .before = 3, .after = 3, .complete = TRUE))
```

```
## # A tsibble: 35 x 3 [1Y]
##   Year Infaltion Infl_7MA
##   <int>   <dbl>   <dbl>
## 1  1986     50      NA
## 2  1987    133.     NA
## 3  1988    186.     NA
## 4  1989   1400    431.
## 5  1990    500    638.
## 6  1991    122.    634.
## 7  1992    625.    608.
## 8  1993   1500    409.
## 9  1994    107.    338.
## 10 1995     3.95    322.
## # ... with 25 more rows
```

```
gdp_ts %>%
  mutate(
    GDP_7MA = slide_dbl(GDP, mean,
                        .before = 3, .after = 3, .complete = TRUE))
```

```
## # A tsibble: 26 x 3 [1Y]
##   Year  GDP GDP_7MA
##   <int> <dbl> <dbl>
## 1  1995 4917.  5301.
## 2  1996 5283.  5290.
## 3  1997 5312.  5223.
## 4  1998 5691.  5250.
## 5  1999 5247.  5447.
## 6  2000 4888.  5853.
## 7  2001 5413.  6487.
## 8  2002 6293.  7191.
## 9  2003 8130.  8127.
## 10 2004 9747.  9436.
## # ... with 16 more rows
```

```
gini_ts %>%
  mutate(
    Gini_7MA = slide_dbl(GINI, mean,
                        .before = 3, .after = 3, .complete = TRUE)
  )
```

```
## # A tsibble: 42 x 3 [1Y]
##   Year  GINI Gini_7MA
##   <int> <dbl>   <dbl>
## 1  1980 0.0418   NA
## 2  1981 0.0418   NA
## 3  1982 0.0418   NA
## 4  1983 0.0418  0.0424
## 5  1984 0.0431  0.0422
## 6  1985 0.0428  0.0421
## 7  1986 0.0434  0.0427
## 8  1987 0.0409  0.0438
## 9  1988 0.0411  0.0453
```

```
## 10 1989 0.0461 0.0474
## # ... with 32 more rows
```