STA2202 - Time Series Analysis - Assignment 1

Luis Correia - Student No. 1006508566

May 15th 2020

Submission instructions:

- Submit *a single PDF file* with your answers to both Theory & Practice parts to A1 on Quercus - the deadline is 11:59PM on Thursday, May 21.
- Your answers to the Theory part can be handwritten (PDF scan/photo is OK).
- Your answers to the Practice part should be in the form of a report combining code, output, and commentary. You can compile your report with RMarkdown (recommended) or another editor (e.g. Word/LaTex).

Theory

Question 1

1. In this course we work with (weakly) stationary time series. This class of models is closed under linear tranformations, i.e. whenever you take a (non-exploding) linear combination of stationary series, you always end up with a stationary series. For this question you have to prove this result. Consider two *independent* zero-mean stationary series, $\{X_t\}$ and $\{Y_t\}$, with autocovariance functions (ACVFs) $\gamma_X(h)$ and $\gamma_Y(h)$, respectively.

(a) [4 marks] Find the ACVF of the linear combination $Z_t = aX_t + bY_t,\ a, b \in \mathbb{R}$ in terms of the ACVFs of $\{X_t\}, \{Y_t\}$, and show that it is stationary (i.e. only depends on h).

(b) [6 marks] Find the ACVF of the linear filter $V_t = \sum_{j=0}^{p} a_j X_{t-j},\ a_j \in \mathbb{R}$ in terms of the ACVF of $\{X_t\}$, and show that it is stationary.

{*Solution.*}

item (a)

Considering $\{X_t\}$ and $\{Y_t\}$ independent, with ACVF given respectively by $\gamma_X(h)$ and $\gamma_Y(h)$, in order to find the ACVF of $Z_t = aX_t + bY_t$, with $a, b \in \mathbb{R}$ we must start with the definition of ACVF of $\gamma_Z(h)$.

$$
\begin{aligned}
\gamma_Z(h) &= Cov(Z_{t+h}, Z_t) \\
&= Cov(aX_{t+h} + bY_{t+h}, aX_t + bY_t) \\
&= Cov(aX_{t+h}, aX_t) + Cov(aX_{t+h}, bY_t) + Cov(bY_{t+h}, aX_t) + Cov(bY_{t+h}, bY_t) \\
&= a^2 Cov(X_{t+h}, X_t) + ab Cov(X_{t+h}, Y_t) + ba Cov(Y_{t+h}, X_t) + b^2 Cov(Y_{t+h}, Y_t)
\end{aligned}
$$

As we have $\{X_t\}$ and $\{Y_t\}$ independent, $cov(X_{t+h}, Y_t) = cov(Y_{t+h}, X_t) = 0$ and we can rewrite $\gamma_Z(h)$ as follows:

$$\begin{aligned}\gamma_Z(h) &= a^2 Cov(X_{t+h}, X_t) + b^2 Cov(Y_{t+h}, Y_t) \\ &= a^2\gamma_X(h) + b^2\gamma_Y(h)\end{aligned}$$

$$\implies \gamma_Z(h) = a^2\gamma_X(h) + b^2\gamma_Y(h) \tag{1}$$

Then (1) shows that $\gamma_Z(h)$ depends only on h.

In order to show that $\{Z_t\}$ is stationary, we need to demonstrate 02 additional conditions:

- $E(Z_t)$ is constant;
- $Var(Z_t)$ is finite.

$E(Z_t)$ is constant

Due to the assumption that $\{X_t\}$ and $\{Y_t\}$ are zero-mean we have

$$\begin{aligned}E(Z_t) &= E(aX_t + bY_t) \\ &= aE(X_t) + bE(Y_t) \\ &= a.0 + b.0 = 0\end{aligned}$$

$$\implies E(Z_t) = 0 \tag{2}$$

$Var(Z_t)$ is finite

As $\{X_t\}$ and $\{Y_t\}$ are *independent* and *stationary*, we have:

- $Var(X_t) < \infty, \forall t \in \mathbb{Z}$
- $Var(Y_t) < \infty, \forall t \in \mathbb{Z}$
- $Cov(X_t, Y_t) = 0, \forall t \in \mathbb{Z}$

Then we have:

$$\begin{aligned}Var(Z_t) &= Var(aX_t + bY_t) \\ &= Var(aX_t) + Var(bY_t) + 2Cov(X_t, Y_t) \\ &= a^2 Var(X_t) + b^2 Var(Y_t) < \infty\end{aligned}$$

$$\implies Var(Z_t) < \infty, \forall t \in \mathbb{Z} \tag{3}$$

Then, by (1), (2) and (3) implies that $\{Z_t\}$ is also *stationary*.

item (b)

Let the linear filter given by $V_t = \sum_{j=0}^{p} a_j X_{t-j},\ \forall a_j \in \mathbb{R}$, its ACVF can be written as follows:

$$\gamma_V(h) = Cov(V_t, X_{t+h}),\ \forall h \in \mathbb{Z} \tag{4}$$

We can then, develop further the equation (4) as follows:

$$\begin{aligned}\gamma_V(h) &= Cov(V_t, V_{t+h}) \\ &= Cov(\sum_{i=0}^{p} a_i X_{t-i}, \sum_{j=0}^{p} a_j X_{t+h-j}) \\ &= \sum_{i=0}^{p} a_i Cov(X_{t-i}, \sum_{j=0}^{p} a_j X_{t+h-j}) \\ &= \sum_{i=0}^{p} a_i \Bigg[\sum_{j=0}^{p} a_j Cov(X_{t-i}, X_{t+h-j}) \Bigg]\end{aligned}$$

Note that $Cov(X_{t-i}, X_{t+h-j}) = \gamma_X(h-j+i)$ so we can rewrite this expression as function of γ_X in the following way:

$$\implies \gamma_V(h) = \sum_{i=0}^{p} a_i \Bigg[\sum_{j=0}^{p} a_j \gamma_X(h-j+i) \Bigg] \tag{5}$$

The expression in (5) can be expanded as follows:

$$\begin{gathered}\gamma_V(h) = \Bigg[a_0^2 \gamma_X(h) + a_0 a_1 \gamma_X(h-1) + \dots + a_0 a_p \gamma_X(h-p) \Bigg] + \\ \Bigg[a_1 a_0 \gamma_X(h+1) + a_1^2 \gamma_X(h) + \dots + a_1 a_p \gamma_X(h-p+1) \Bigg] + \dots \\ \Bigg[a_p a_0 \gamma_X(h+p) + a_p a_1 \gamma_X(h+p-1) + \dots + a_p^2 \gamma_X(h) \Bigg]\end{gathered}$$

By grouping the common terms, this expression can be simplified to

$$\implies \gamma_V(h) = \sum_{j=0}^{p} a_j^2 \gamma_X(h) + 2 \sum_{i=0}^{p} \sum_{j=i+1}^{p} a_i a_j \gamma_X(h-j+i) \tag{6}$$

And we have that (6) does not depend on t, but only on h.

To prove $\{V_t\}$ is stationary we need to verify two additional conditions, which are $E(V_t)$ is constant and $Var(V_t)$ is finite (i.e., non-explosive).

$E(V_t)$ is constant

Note that $E(V_t)$ can be written as follows:

$$\begin{aligned}E(V_t) &= E\Bigg(\sum_{j=0}^{p} a_j X_{t-j} \Bigg) \\ &= \sum_{j=0}^{p} a_j E(X_{t-j})\end{aligned}$$

Considering that $\{X_t\}$ is stationary and *zero-mean*, we have:

$$\implies E(V_t) = 0 \tag{7}$$

$Var(V_t)$ **is finite**

Recall from (6) that we calculated the ACVF of $\gamma_V(h)$ so, in order to calculate the $Var(V_t)$ we just have to set $h = 0$, so, in this sense:

$$\begin{aligned} Var(V_t) &= \gamma_V(0) \\ &= \sum_{j=0}^{p} a_j^2 \gamma_X(0) + 2 \sum_{i=0}^{p} \sum_{j=i+1}^{p} a_i a_j \gamma_X(0 - j + i) \end{aligned}$$

Using the fact that $\{X_t\}$ is stationary, $\gamma_X(.)$ is finite then, we have:

$$\implies Var(V_t) = \gamma_X(0) \sum_{j=0}^{p} a_j^2 + 2 \sum_{i=0}^{p} \sum_{j=i+1}^{p} a_i a_j \gamma_X(i - j) < \infty \tag{8}$$

From (6), (7) and (8) we conclude that V_t is stationary.

Question 2

2. [10 marks] Consider the random walk (RW) series $X_t = X_{t-1} + W_t,\ \forall t \geq 1$, where $X_0 = 0$ and $W_t \sim WN(0,1)$. Although the series is *not stationary*, assume we treat it as such and calculate the *sample* ACVF $\hat{\gamma}(h)$, based on a sample of size n, as:

$$\hat{\gamma}(h) = \frac{1}{n}\sum_{t=1}^{n-h}(X_{t+h}X_t),\quad \forall h = 0,1,\ldots,n-1$$

Show that the *expected value* of the sample auto-covariances are given by

$$\mathbb{E}[\hat{\gamma}(h)] = \frac{(n-h)(n-h+1)}{2n}$$

(*Hint*: the ACVF of X is $\gamma(s,t) = \min(s,t),\ \forall s,t \geq 1$, and the arithmetic series formula is $\sum_{i=1}^{n} i = n(n+1)/2$.)
(*Note*: this illustrates the behavior of the sample ACF of a RW series: it is in fact a quadratic in h, but it behaves very close to linear for the small values of h that appear in the ACF plot.)

{*Solution.*}

Considering a *Random-Walk(RW)* $\{X_t\}$ which is given by $X_t = X_{t-1} + W_t, \forall t \geq 1$, where $X_0 = 0$, $W_t \sim WN(0,1)$ and its sample ACVF $\hat{\gamma}(h)$ calculated as:

$$\hat{\gamma}(h) = \frac{1}{n}\sum_{t=1}^{n-h}\Big(X_{t+h}X_t\Big),\quad \forall h = 0,1,\ldots,n-1 \tag{9}$$

Calculating the *expected value* of $\hat{\gamma}(h)$ using (9), we have:

$$\begin{aligned}
E\big[\hat{\gamma}(h)\big] &= E\left[\frac{1}{n}\sum_{t=1}^{n-h}\big(X_{t+h}X_t\big)\right] \\
&= \frac{1}{n}\sum_{t=1}^{n-h} E\big(X_{t+h}X_t\big) \\
&= \frac{1}{n}\sum_{t=1}^{n-h} E\Big[\Big(\sum_{i=1}^{t+h} W_i\Big)\Big(\sum_{j=1}^{t} W_j\Big)\Big] \\
&= \frac{1}{n}\sum_{t=1}^{n-h}\Big[E\Big(W_1\sum_{j=1}^{t}W_j + W_2\sum_{j=1}^{t}W_j + \cdots + W_{t+h}\sum_{j=1}^{t}W_j\Big)\Big] \\
&= \frac{1}{n}\sum_{t=1}^{n-h}\Big[E\Big(W_1\sum_{j=1}^{t}W_j\Big) + E\Big(W_2\sum_{j=1}^{t}W_j\Big) + \cdots + E\Big(W_{t+h}\sum_{j=1}^{t}W_j\Big)\Big]
\end{aligned}$$

$$\implies E\big[\hat{\gamma}(h)\big] = \frac{1}{n}\sum_{t=1}^{n-h}\Big[E\Big(W_1\sum_{j=1}^{t}W_j\Big) + E\Big(W_2\sum_{j=1}^{t}W_j\Big) + \cdots + E\Big(W_{t+h}\sum_{j=1}^{t}W_j\Big)\Big] \tag{10}$$

Considering that $W_t \sim WN(0,1)$, we have that:

$$E\big(W_iW_j\big) = \begin{cases} Cov(W_i,W_j) = 0, & \text{if } i \neq j \\ Cov(W_i,W_j) = Var(W_i) = 1, & \text{otherwise} \end{cases} \tag{11}$$

From (10) and (11) we have that:

$$E\Big(W_i \sum_{j=1}^{t} W_j\Big) = \begin{cases} \sum_{j=1}^{t} E\Big(W_i W_j\Big) = 0, & \text{if } i \neq j \ or \ i > t \\ Var(W_i) = 1, & \text{otherwise} \end{cases} \tag{12}$$

Then, applying (12) in (10) we have:

$$\begin{aligned} E\big[\hat{\gamma}(h)\big] &= \frac{1}{n} \sum_{t=1}^{n-h} \Big[E\Big(W_1 W_1\Big) + E\Big(W_2 W_2\Big) + \cdots + E\Big(W_t W_t\Big)\Big] \\ &= \frac{1}{n} \sum_{t=1}^{n-h} \Big[Var\big(W_1\big) + Var\big(W_2\big) + \cdots + Var\big(W_t\big)\Big] \\ &= \frac{1}{n} \sum_{t=1}^{n-h} \big[1 + 1 + \cdots + 1\big] \\ &= \frac{1}{n} \sum_{t=1}^{n-h} t \\ &= \frac{1}{n} \frac{(n-h)(n-h+1)}{2} \end{aligned}$$

$$\implies E\big[\hat{\gamma}(h)\big] = \frac{(n-h)(n-h+1)}{2n}. \tag{13}$$

This concludes the demonstration.

Practice

You will work with Statistics Canada's open socio-economic series data. The data are organized by topic in tables, and we will focus on monthly employment numbers by industry (table 14-10-0355-01); see also this brief tutorial. An easy way to access these data directly through R is with the `cansim` library, using "vectors" to identify individual series. **You will be working with employment data for diferent industries and over different time periods, based on the last two digits of your student #, according to the scheme described in the following tables:**

last digit of student #	Industry	Unadjusted	Seasonally adjusted	Trend-cycle
1	Accommodation and food services	v2057828	v2057619	v123355122
2	Agriculture	v2057814	v2057605	v123355108
3	Construction	v2057817	v2057608	v123355111
4	Educational services	v2057825	v2057616	v123355119
5	Forestry, fishing, mining, quarrying, oil and gas	v2057815	v2057606	v123355109
6	Goods-producing sector	v2057813	v2057604	v123355107
7	Information, culture and recreation	v2057827	v2057618	v123355121
8	Manufacturing	v2057818	v2057609	v123355112
9	Public administration	v2057830	v2057621	v123355124
0	Services-producing sector	v2057819	v2057610	v123355113

2nd to last digit of student #	Time period
odd	Jan 1980 to Dec 1999
even	Jan 2000 to Dec 2019

E.g., if your student ID ends in 42, you should use the Agriculture industry data (last digit = 2) over Jan 2000 to Dec 2019 (next-to-last digit = 4 is even). **Beware to use the right data, otherwise you will lose marks**. The following starter code downloads the data for student # ending in 42.

```r
library(cansim)
library(tidyverse)
# unadjusted (raw) series
ua = get_cansim_vector( "v2057605", start_time = "2000-01-01", end_time = "2019-12-01") %>%
  pull(VALUE) %>% ts( start = c(2000,1), frequency = 12)
plot(ua)
```

Question 1

1. [3 marks] Plot the unadjusted series, its ACF & PACF, and comment on the following characteristics: trend, seasonality, stationarity.

{*Solution.*}

For this question, considering the last 2-digit of my Student number is “66”, we have the following code which leads to the folowing plots:

```
# Libraries needed for this assignment
library(cansim)
library(tidyverse)
library(ggplot2)
library(forecast)
library(grid)
```

```
# unadjusted (raw) series
ua = get_cansim_vector( "v2057814",
                        start_time = "2000-01-01", end_time = "2019-12-01") %>%
  pull(VALUE) %>% ts( start = c(2000,1), frequency = 12)
```

```
autoplot(ua)
```

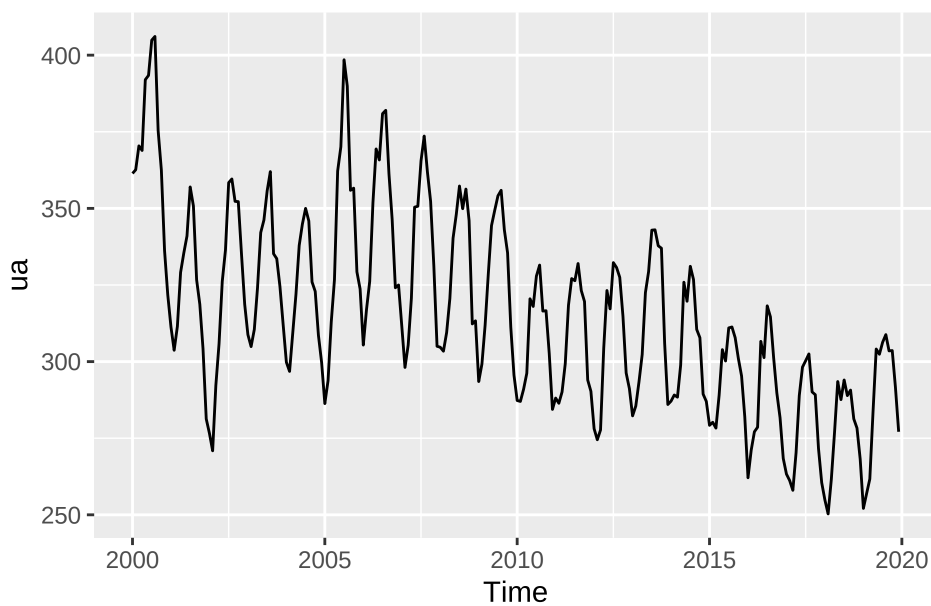


Figure 1: Unadjusted Series - Employment Goods-producing sector (Jan2000-Dec2019)

```
# Plotting the ACF/PACF for unnadjusted series
p1 <- ggAcf(ua, lag.max = 24)+
  ggtitle("ACF Unnadjusted")
p2 <- ggPacf(ua, lag.max = 24)+
  ggtitle("PACF Unnadjusted")

gridExtra::grid.arrange(p1, p2, nrow = 1)
```

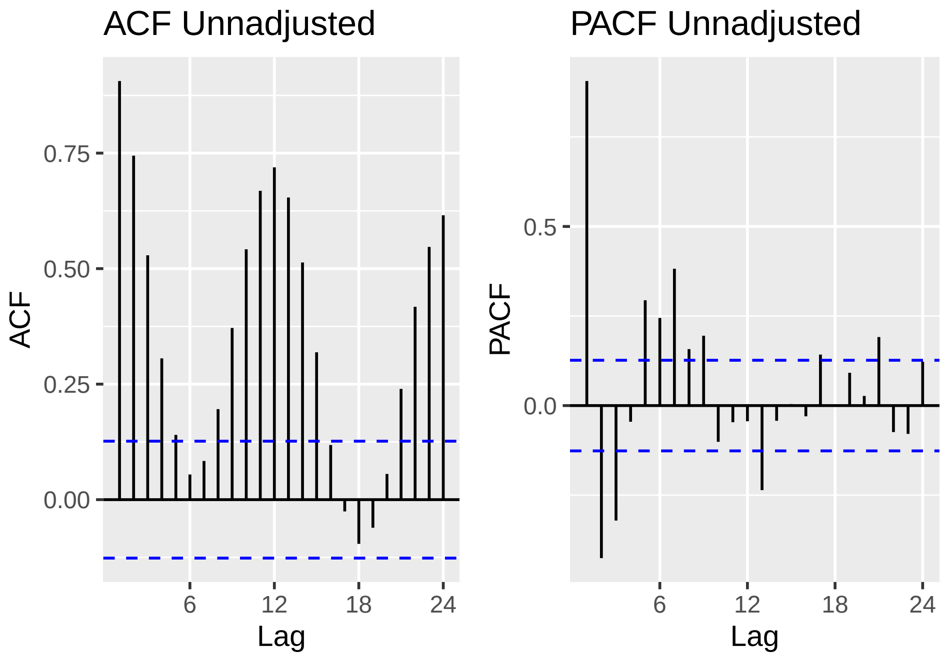


Figure 2: ACF/PACF - Employment Goods-producing sector (Jan2000-Dec2019)

Trend: Analysing the graphs we can see two major similar behaviours with increasing trends, one from Jan-2000 to 2008, when we have an abrupt fall of the activity probably due to the 2008 sub-prime crisis in US, which was affected most of the worldwide economies. Canada, which has close relationship with american economy was affected so we can understand, specially the employment on goods-producing industry would be seriously affected as well. On the other hand, after from 2010 to 2020 we can observe a positive trend reflecting the market has recovered his force and demonstrated a positive growth in activity, reflecting on demand for employees.

```r
# Plotting the Monthplot to identify differences on months
ggmonthplot(ua, ylab="Employment")
```

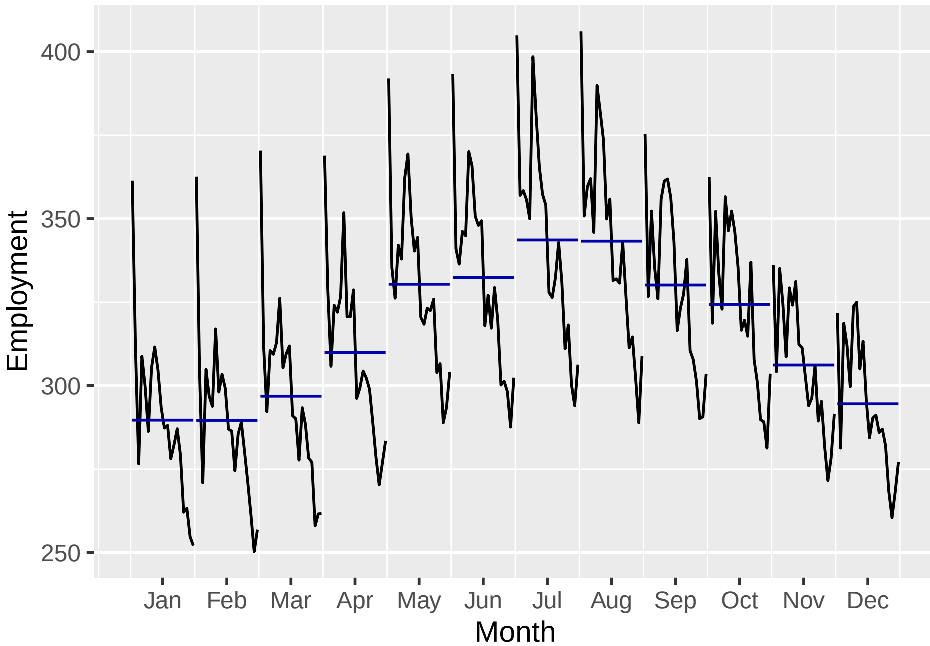


Figure 3: Monthplot - Employment Goods-producing sector (Jan2000-Dec2019)

Seasonality: Regarding the seasonality, we can clearly observe an anual behavior, i.e., the cycle is 12-month duration which reflects the fiscal year where the producing/sale of goods are mostly consumed. This can be observed through the `monthplot()` of the series when we see a clear pattern during the months from Mar-Aug (positive trend) and Sep-Dec (negative trend) in demand from goods industry and Jan-Feb a lower level of activity of this industry. This reflects fairly well the cycle of the economy where, during Spring-Summer, we have naturally more demand when compared to Fall-Winter seasons which naturally have direct impact over employment in this industry.

Stationarity: Visually from the series plot we do not observe explosive values which can suggest we have a well behaviored variance around a common mean, around levels $3,800 - 3,900$. In this sense we can consider the series is *weakly stationary*, as we will see further in this assignment.

Question 2

2. [5 marks] Perform a classical *multiplicative* decomposition of the unadjusted series (X_{ua}) into trend (T), seasonal (S), and remainder (R) components (i.e. $X_{ua} = T \times S \times R$):

a. First, apply a *12-point MA* to the raw (unadjusted) series to get an estimate of the trend.
b. Then, use the *detrended* data to estimate seasonality: find the seasonal pattern by calculating sample means for each month, and then center the pattern at 1 (i.e divide the pattern by its mean, so that its new mean is 1).
c. Finally, calculate the *remainder* component by removing both trend and seasonality from the raw series. Create a time-series plot of all components like the one below.
(*Hint*: you results should perfectly match those of the `decompose` function, which uses the above process)

{*Solution.*}

item (a)

Multiplicative decomposition

Consideting the cycle is annual, i.e., $m = 12$, I did the following steps to decompose the series:

Step 1: Calculated the $\hat{T}_t$ series using the $2 \times 12 - MA$

```
m <- 12
THat <- ua %>%
  stats::filter(c(.5, rep(1,(m-1)), .5)/m)
autoplot(THat, ylab="Employment - Trend")
```

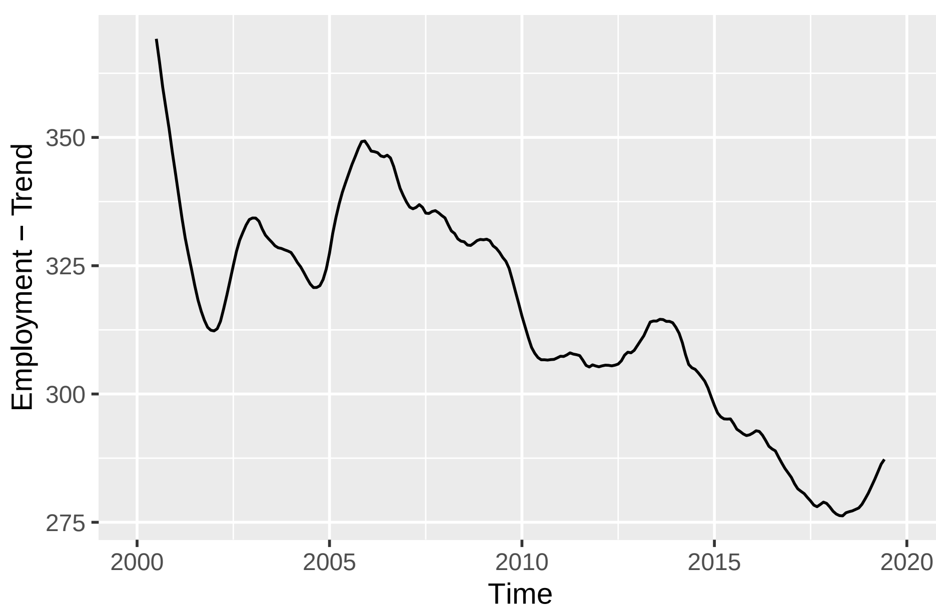


Figure 4: Trend Series - Employment Goods-producing sector (Jan2000-Dec2019)

item (b)

Step 2: Calculated the detrended series by dividing the original series by the one calculated in previous step, i.e., $y_t/\hat{T}_t$.

```r
Det_ua <- ua/THat
autoplot(Det_ua, ylab="Employment - Detrended")
```

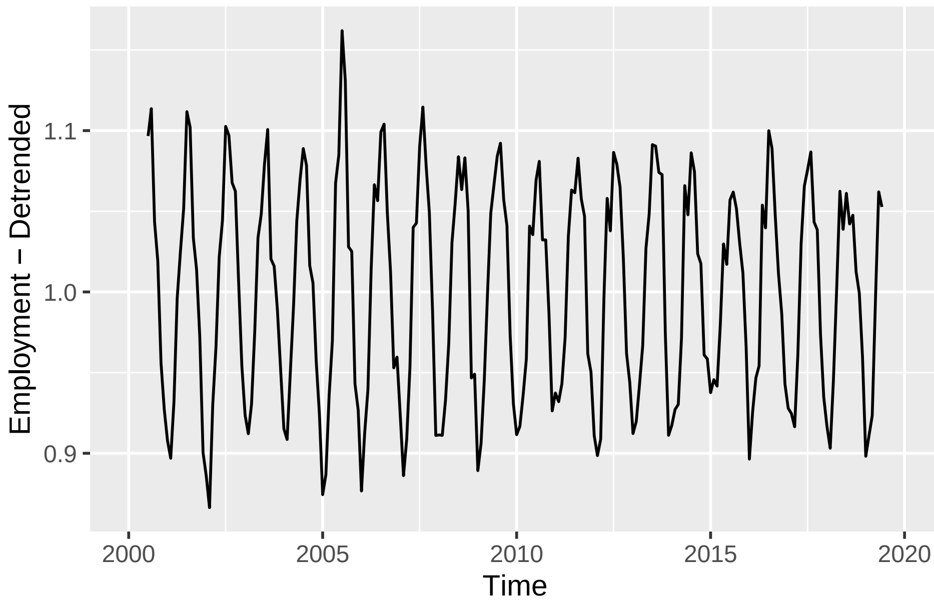


Figure 5: Detrended Series - Employment Goods-producing sector (Jan2000-Dec2019)

Step 3: To calculate the seasonal component for each season, I will average the detrended series for each of the 12 seasons we have in our cycle, by Calculated the detrended series by dividing the original series by the one calculated in previous step, i.e., $\hat{S}_t = y_t/\hat{T}_t$.

```r
# Set initial values to calculate seasonality
n <- length(ua)
nper <- n/m

# Vector containing the indexes to be used to calculate averages of each period
v <- array(dim=c(m, nper))
MPer <- vector()

# Calculate the indexes for whole series
for (i in 1:m)
  v[i,] <- (0:(nper-1)*m)+1+(i-1)

# Calculate the seasonality
for (i in 1:m)
  MPer[i] <- mean(Det_ua[v[i,1:nper]], na.rm = TRUE)

# Replicate to all series
SHat <- ts(rep(MPer, nper), start=head(time(ua), 1), frequency=frequency(ua))
# SHat <- ua/rep(MPer, nper)
autoplot(SHat, ylab="Employment - Seasonal")
```

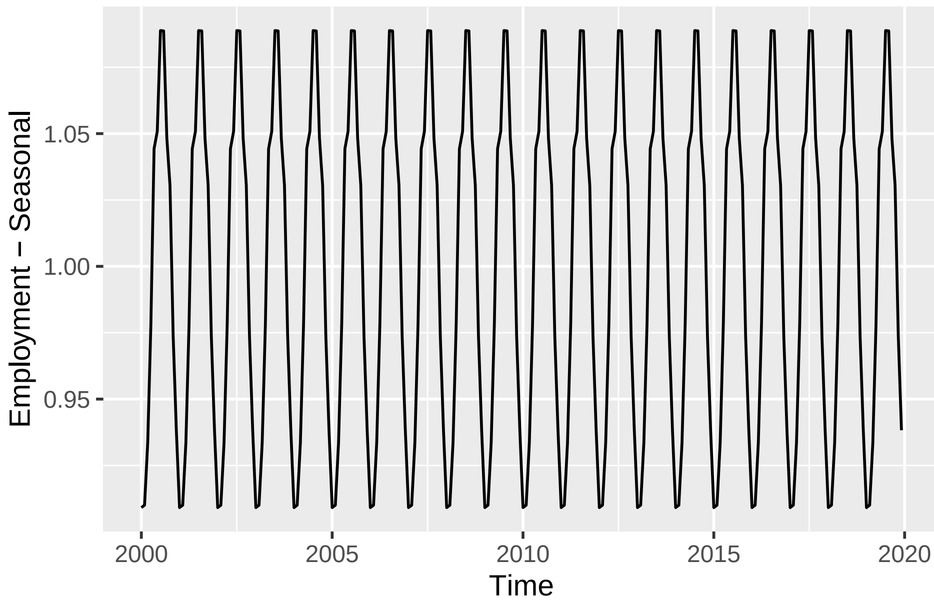


Figure 6: Seasonal Series - Employment Goods-producing sector (Jan2000-Dec2019)

item (c)

This leads us to the following *detrended-deseasonal* series:

```
Dets_ua <- Det_ua/SHat
autoplot(Dets_ua, ylab="Employment\nDetrended,Deseasonal")
```

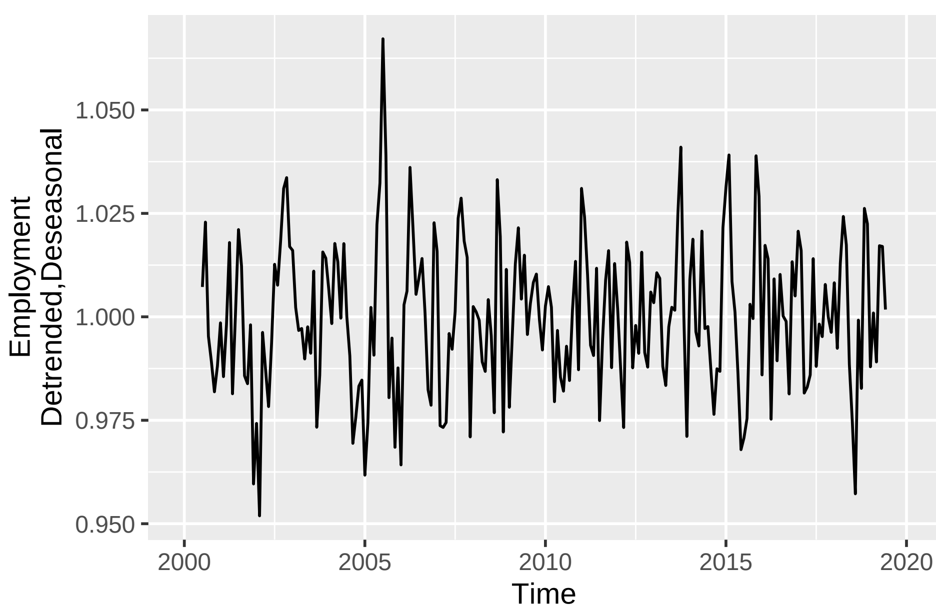


Figure 7: Detrended-deseasonal Series - Employment Goods-producing sector (Jan2000-Dec2019)

This series is equal to the `remainder` series, as we removed the `trend` and `seasonal` aspects from the original series, as we will see in the next step.

Step 4: The remainder component will be now calculated by dividing out the estimated seasonal and trend-cycle components, i.e., $\hat{R}_t = y_t/(\hat{S}_t\hat{T}_t)$.

```
# Set initial values to calculate seasonality
RHat <- ua/(SHat*THat)
autoplot(RHat, ylab="Employment - Random")
```

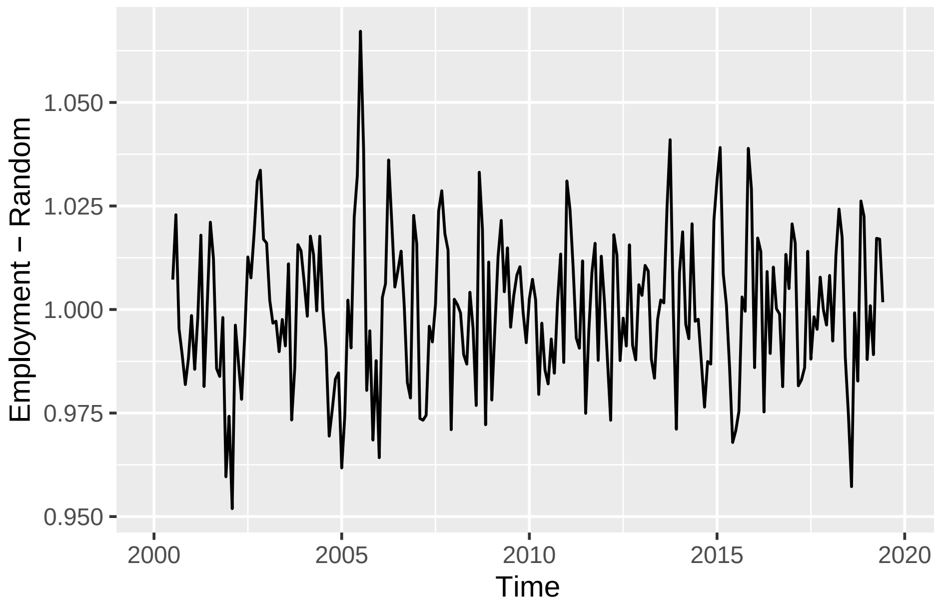


Figure 8: Random Series - Employment Goods-producing sector (Jan2000-Dec2019)

Which is the exactly the *detrended-deseasonal* series obtained.

In order to do check if the results are the same, I will plot the decomposed series using the `decompose()` function against the results obtained in the decomposition above.

```
Dec_ua <- decompose(ua, type = "multiplicative")
autoplot(Dec_ua)
```

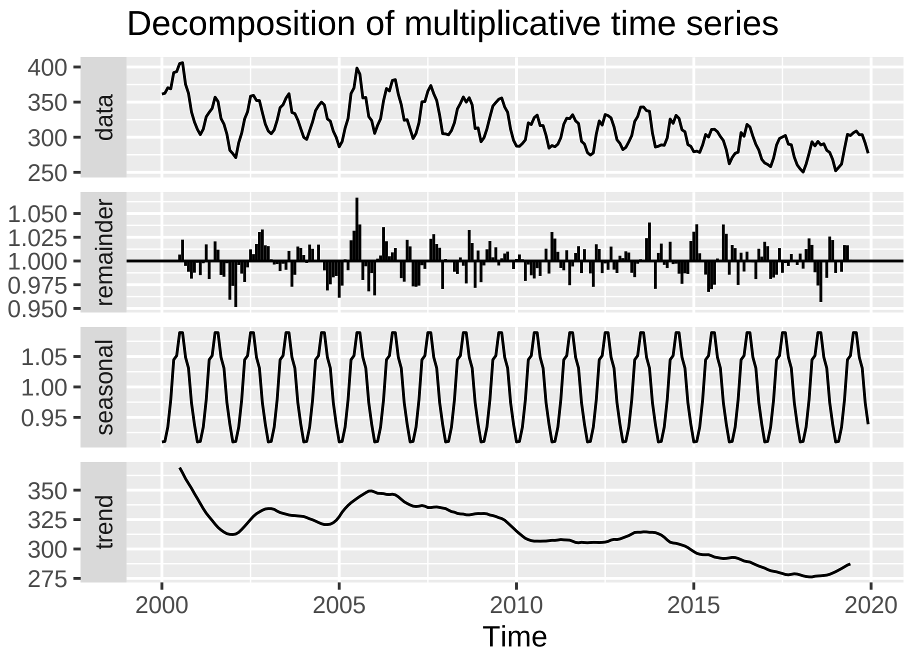


Figure 9: R-Decomposition - Employment Goods-producing sector (Jan2000-Dec2019)

```
# Plot the composed series obtained
p1 <- autoplot(ua, ylab="observed", xlab=NULL)
p2 <- autoplot(THat, ylab="trend")
p3 <- autoplot(SHat, ylab="seasonal", xlab=NULL)
p4 <- autoplot(RHat, ylab="random")

grid.newpage()
grid.draw(rbind(ggplotGrob(p1), ggplotGrob(p2), size = "last"))
```

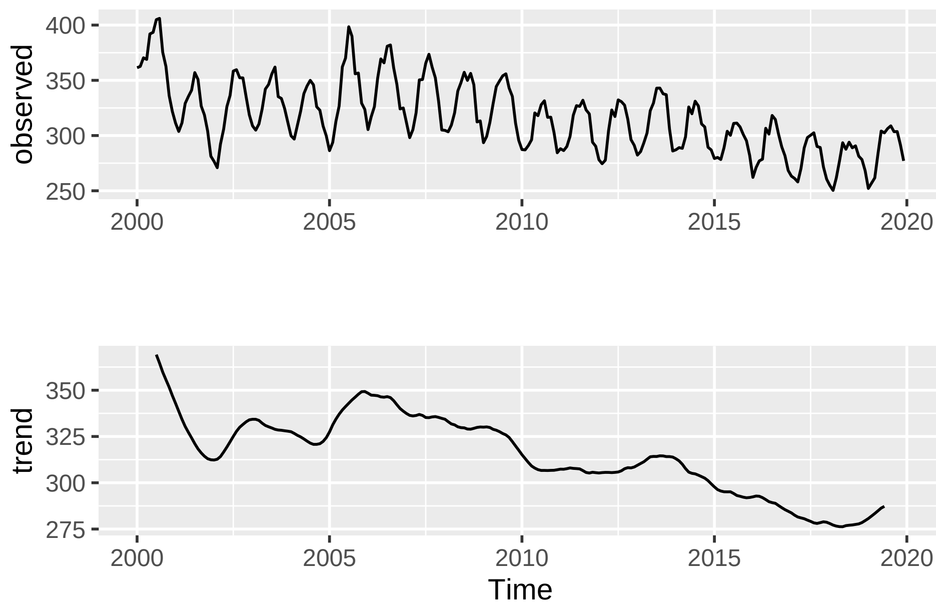


Figure 10: Our Decomposition - Employment Goods-producing sector (Jan2000-Dec2019)

```
grid.newpage()
grid.draw(rbind(ggplotGrob(p3), ggplotGrob(p4), size = "last"))
```

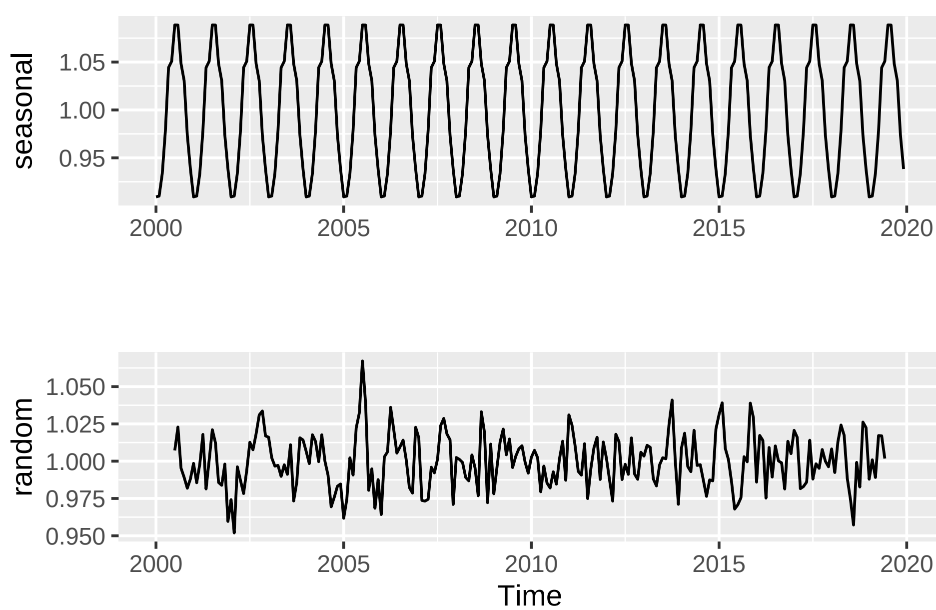


Figure 11: Our Decomposition - Employment Goods-producing sector (Jan2000-Dec2019)

As the graphs are identical, this concludes question 2.

Question 3

3. [2 marks] Statistics Canada (StatCan) does their own seasonal adjustment using a more sophisticated method (namely, X-12-ARIMA). Download the corresponding *seasonally adjusted* series for your industry and time period, and plot them on the same plot with your own seasonally adjusted data ($X_{sa} = X_{ua}/S = T \times R$) from the previous part. The two versions should be close, but not identical. Report the mean absolute error (MAE) between the two versions (StaCan's and yours) of seasonally adjusted data.

{*Solution.*}

First pulling the appropriate data referred to Goods-Producing in the period from Jan-2000 to Dec-2019.

```
# Pull adjusted series
ad = get_cansim_vector( "v2057605",
                        start_time = "2000-01-01", end_time = "2019-12-01") %>%
  pull(VALUE) %>% ts( start = c(2000,1), frequency = 12)
```

... and now plotting both graphs in the same view...

```
# Compare both plots in the same graph
autoplot(ad, series="StatCan")+
  autolayer(ua/SHat, series="MyExerc") +
  xlab("Year") + ylab("Employment") +
  scale_colour_manual(values=c("gray","blue"),
             breaks=c("StatCan","MyExerc"))
```

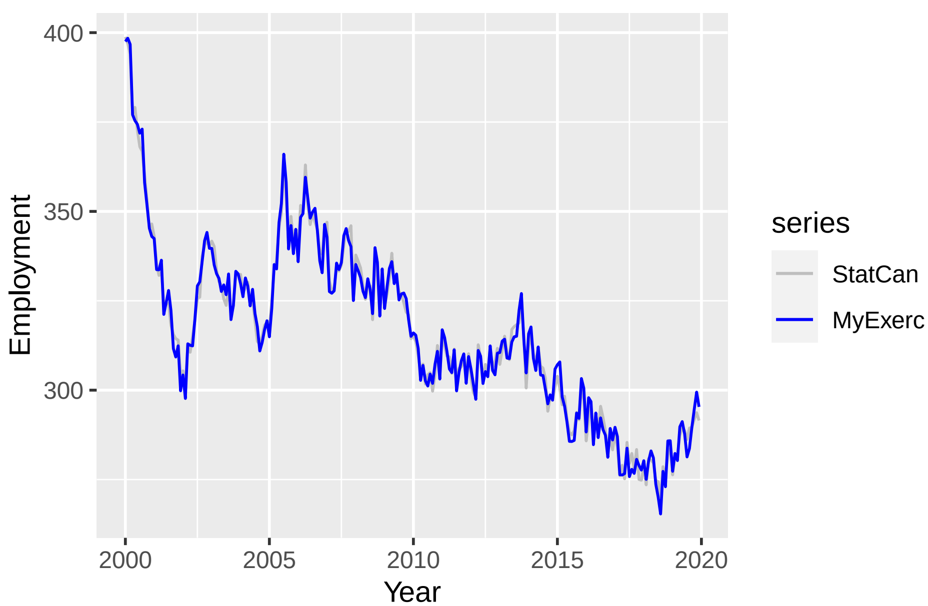


Figure 12: Compare of Decompositions - StatCan site vs. My Exercise

As expected, both adjustments are very close, one to another.

As per definition, the *Mean Absolute Error* is a measure of errors between paired observations expressing the same phenomenon. In this sense, we have for our case:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |\texttt{ad}_i - \texttt{ua}_i/\texttt{SHat}_i|}{n}$$

where $\texttt{ad}_i$ is ith term of the adjusted series obtained from the website; and $\texttt{ua}_i/\texttt{SHat}_i$ is the ith term of the de-seasonal series obtained in previous exercise.

```
# Creating the function to calculate the Mean Absolute Error (to be used further on next questions)
MAE <- function (y, x) {
  return(sum(abs(y-x))/length(y))
}

cat("\nThe MAE (Mean Absolute Error) is ", MAE(ad, ua/SHat),"\n")
```

```
##
## The MAE (Mean Absolute Error) is  2.175066
```

Question 4

4. [5 marks] The library `seasonal` contains R functions for performing seasonal adjustments/decompositions using various methods. Use the following three methods described in FPP for performing seasonal adjustments (you don't need to know their details):

a. X11

b. SEATS

c. STL

Create seasonaly adjusted versions of your raw series based on each method, and plot them together with StaCan's version. Note that the first two methods (X11 & SEATS) are *multiplicative* by default, and you must use the `forecast` library function `seasadj`, `seasonal`, `trendcycle`, and `remainder` to extract the various components. The last method (STL) however is only *additive*, so you need to take a logarithmic transformation of the data to do the *multiplicative* decomposition, and then transform them back to the original scale for making comparisons.
Which method gives a seasonal adjustment that is closest to StaCan's, based on MAE?

{*Solution.*}

I will divide this answer in 02 part as follows:

- *Part-One* - Give a brief definition and characteristic of each adjustment included a descriptio from the textbook from Hyndman[1], and then plot the adjustment of the original series;
- *Part-Two* - Comparing the adjustments of each methodology against the one used by StatCan by plotting simultaneous graphs with all series adjusted and, finally, elaborate a table with the MAE calculated for each adjustment against StatCan to identify the one that most approximates that adjustment.

Part-One

Including the library `seasonal` we will need for this exercise.

```r
library(seasonal)
library(devtools)
```

For this question we will use the *unadjusted* series (i.e., $\texttt{ua}_i$) to perform the required decompositions.

(a) *X11 Decomposition*: Another popular method for decomposing quarterly and monthly data is the X11 method which originated in the US Census Bureau and Statistics Canada.

This method is based on classical decomposition, but includes many extra steps and features in order to overcome the drawbacks of classical decomposition that were discussed in the previous section. In particular, trend-cycle estimates are available for all observations including the end points, and the seasonal component is allowed to vary slowly over time. X11 also has some sophisticated methods for handling trading day variation, holiday effects and the effects of known predictors. It handles both additive and multiplicative decomposition. The process is entirely automatic and tends to be highly robust to outliers and level shifts in the time series.

```r
# X11 decomposition
fitx11 <- ua %>%
  seas(x11="")
autoplot(fitx11)
```

[1] Hyndman, R. J. and Athanasopoulos, G., [Forecasting: Principles and Practice](https://otexts.com/fpp2/)

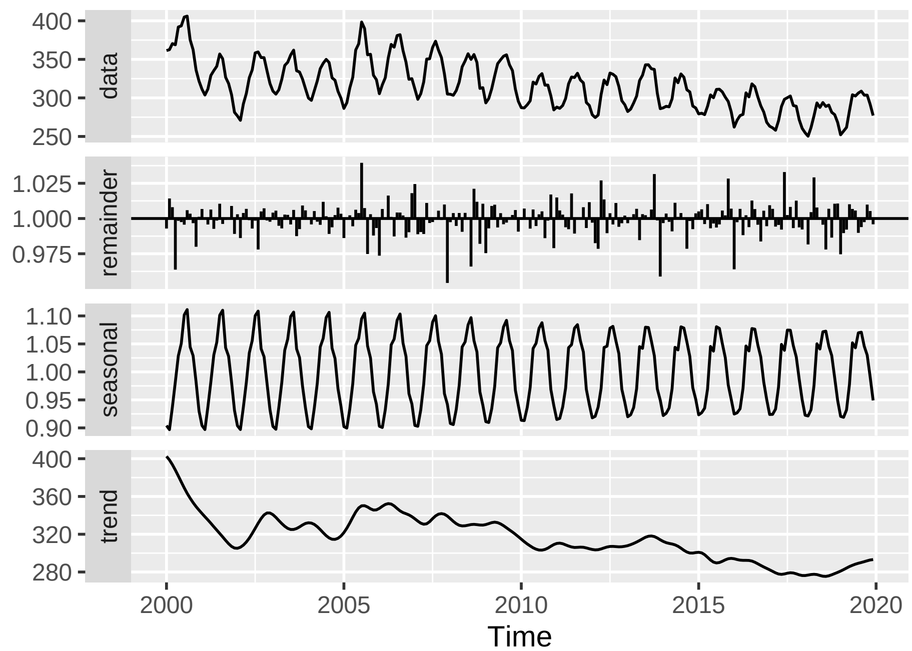


Figure 13: X11 decomposition of Goods Producing from Jan-2000 to Dec-2019

(b) *SEATS Decomposition*: "SEATS" stands for "Seasonal Extraction in ARIMA Time Series". This procedure was developed at the Bank of Spain, and is now widely used by government agencies around the world. The procedure works only with quarterly and monthly data. So seasonality of other kinds, such as daily data, or hourly data, or weekly data, require an alternative approach.

```
# SEATS decomposition
fitSeats <- ua %>% seas()
autoplot(fitSeats)
```

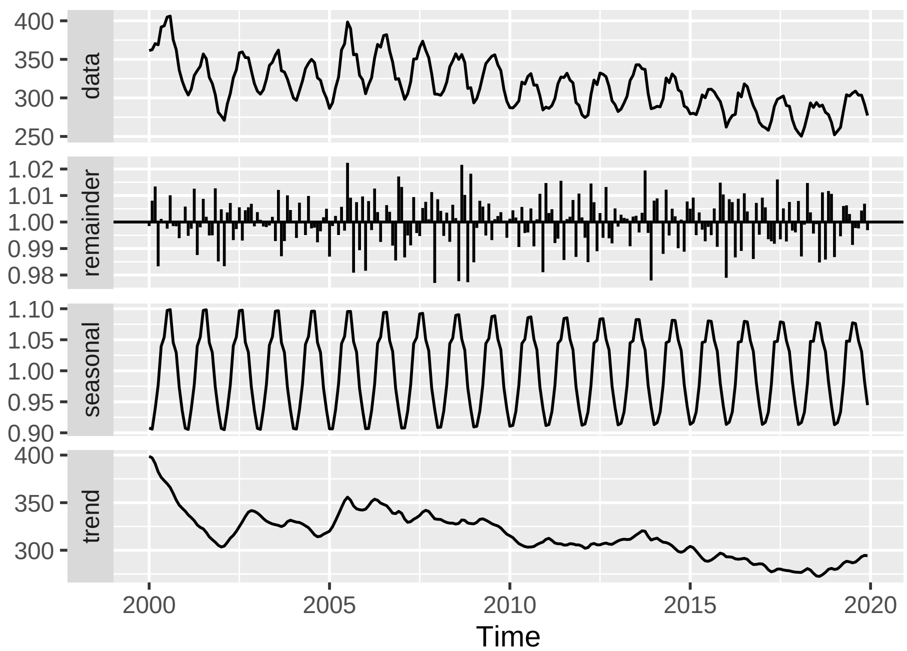


Figure 14: SEATS decomposition of Goods Producing from Jan-2000 to Dec-2019

(c) *STL Decomposition*: STL is a versatile and robust method for decomposing time series. STL is an acronym for "Seasonal and Trend decomposition using Loess", while Loess is a method for estimating nonlinear relationships. The STL method was developed by Cleveland, Cleveland, McRae, & Terpenning (1990).

```
# STL decomposition
fitSTL <- ua %>%
  stl(s.window="periodic", robust=TRUE)
autoplot(fitSTL)
```

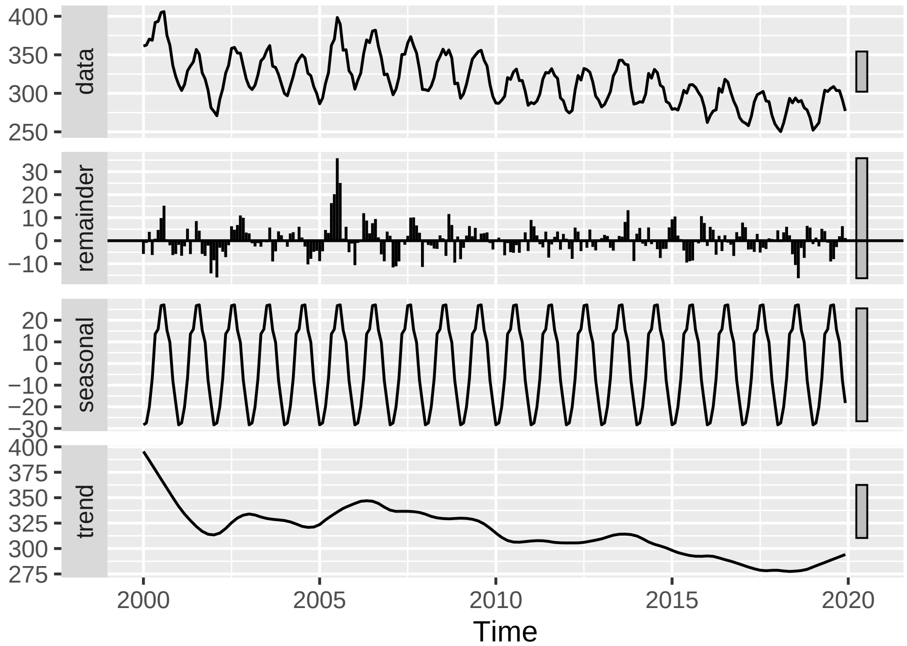


Figure 15: STL decomposition of Goods Producing from Jan-2000 to Dec-2019

Part-Two

Now comparing the X11, SEATS and STL adjustment against StatCan in the same plot we have the following:

```
# Plot X11 adjustment against the StatCan Adjustment
autoplot(ad, series="StatCan") +
  autolayer(seasadj(fitx11), series="X11 Adjusted") +
  xlab("Year") + ylab("Employment") +
  scale_colour_manual(values=c("gray","blue"),
             breaks=c("StatCan","X11 Adjusted"))
```

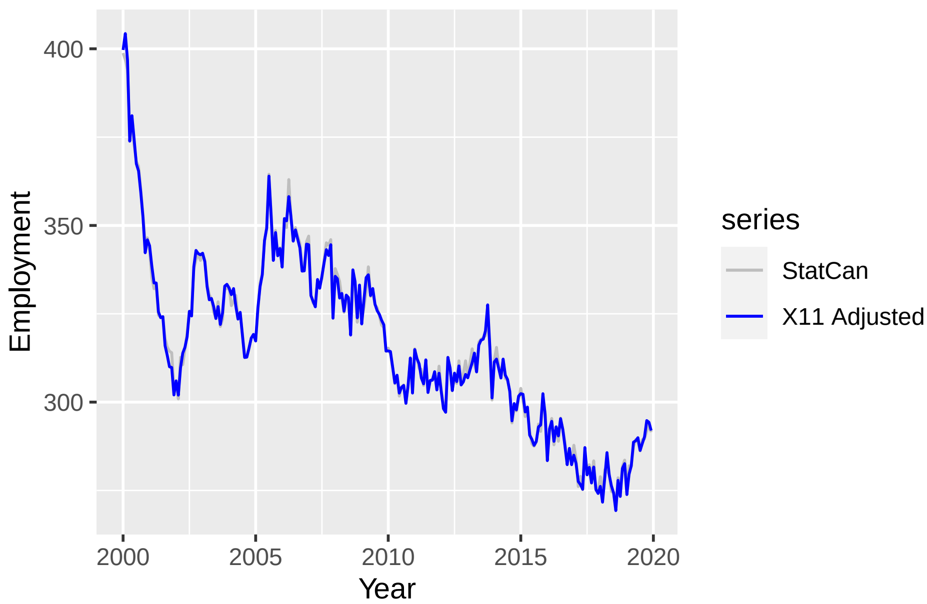


Figure 16: X11 against StatCan Adjustment - Goods Producing from Jan-2000 to Dec-2019

```
# Calculate the mean absolute error for X11 method over our series
MAEX11 <- MAE(ad, seasadj(fitx11))
```

```
# Plot SEATS adjustment against the StatCan Adjustment
autoplot(ad, series="StatCan") +
  autolayer(seasadj(fitSeats), series="SEATS Adjusted") +
  xlab("Year") + ylab("Employment") +
  scale_colour_manual(values=c("gray","blue"),
             breaks=c("StatCan","SEATS Adjusted"))
```

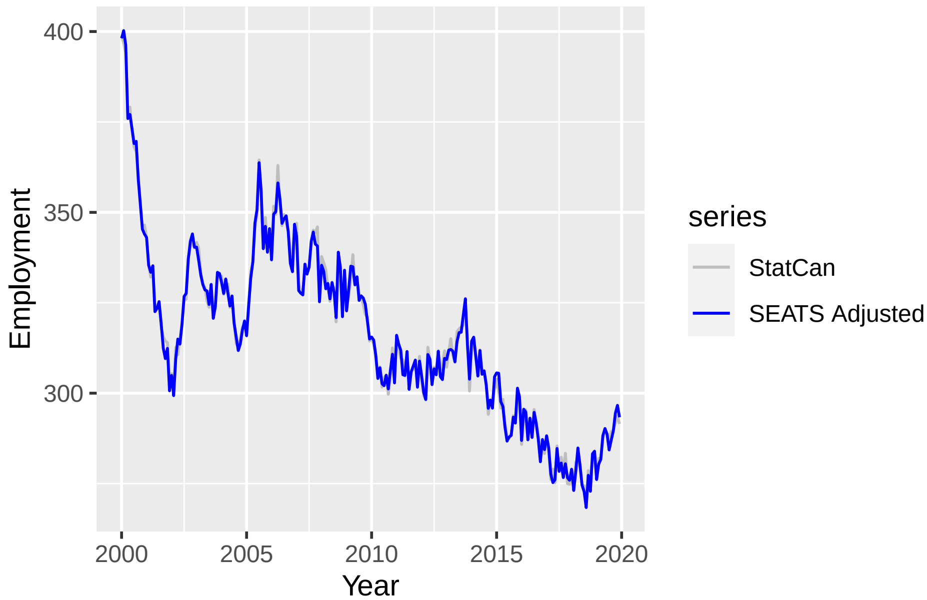


Figure 17: SEATS against StatCan Adjustment - Goods Producing from Jan-2000 to Dec-2019

```
# Calculate the mean absolute error for SEATS method over our series
MAESeats <- MAE(ad, seasadj(fitSeats))
```

```
# As STL is an additive method we need to use another decomposition of the series

# First get the additive decomposition of our series
yt <- ua
St <- fitSTL$time.series[,"seasonal"]
Tt <- fitSTL$time.series[,"trend"]
Rt <- fitSTL$time.series[,"remainder"]

# Calculating de-seasonal series obtained from STL method
dyS <- yt-St

# Plot STL adjustment against the StatCan Adjustment
autoplot(ad, series="StatCan") +
  autolayer(dyS, series="STL Adjusted") +
  xlab("Year") + ylab("Employment") +
  scale_colour_manual(values=c("gray","blue"),
             breaks=c("StatCan","STL Adjusted"))
```

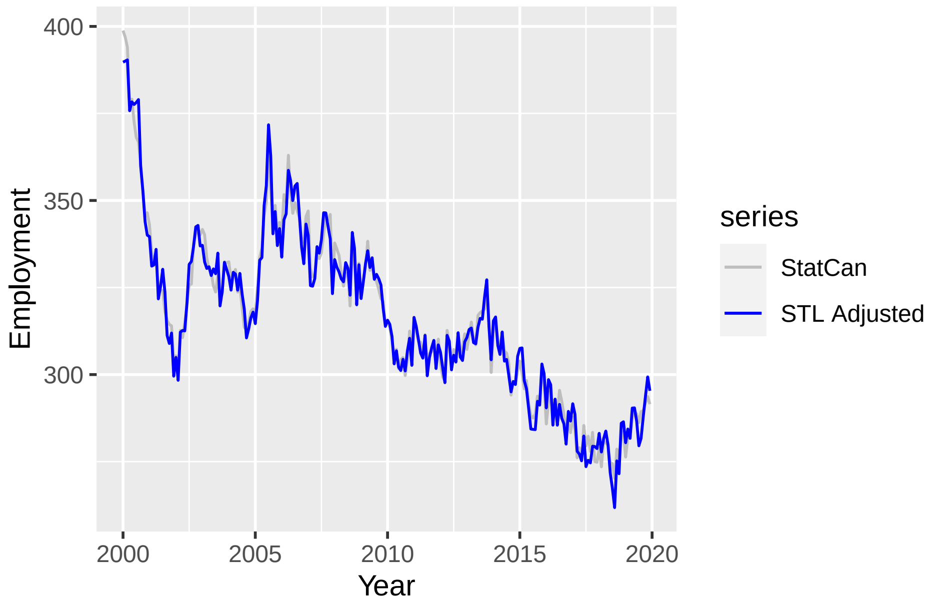


Figure 18: STL against StatCan Adjustment - Goods Producing from Jan-2000 to Dec-2019

```r
# Calculate the mean absolute error for SEATS method over our series
MAESTL <- MAE(ad, dyS)
```

```r
prtTable <- data.frame (
  Method = c("X11","SEATS","STL"),
  MAE = c(MAEX11, MAESeats, MAESTL)
)
# Format Intervals for printing
kableExtra::kable(prtTable, "latex", booktabs = TRUE, caption = "MAE for each de-seasonal Method")
```

Table 3: MAE for each de-seasonal Method

Method	MAE
X11	1.189743
SEATS	1.517850
STL	2.933545

Comparing the methods, the *X11*-method presented the smallest MAE(Mean Absolute error) when compared with *SEATS* and *STL*, for that reason it can be considered the closest to StaCan's.

Question 5

5. [5 marks] Using StatCan's data (unadjusted, and/or seasonally adjusted, and/or trend-cycle), calculate the *remainder* series (R). Plot R and its sample ACF and PACF, and answer the following questions:

a. Based on these plots, can you identify any remaining seasonality in your series?
b. Comment on the stationarity of the series and propose any further pre-processing.
c. Comment on the (partial) autocorrelations of the series, and propose an appropriate ARMA(p, q) model (i.e. appropriate orders p & q).

{*Solution.*}

item (a)

Analyze ACF/PACF and discuss about remaining seasonality on this series

For this question we will use the remainder calculated in question 2, which is stored in `RHat`$_i$ and plot the `ACF()` and `PACF()`, as requested.

```
# Excluding both head() and tail() which are "NA"s due to 12-MA calculation
adjRHat <- tail(head(RHat,-m/2),-m/2)
autoplot(adjRHat)
```

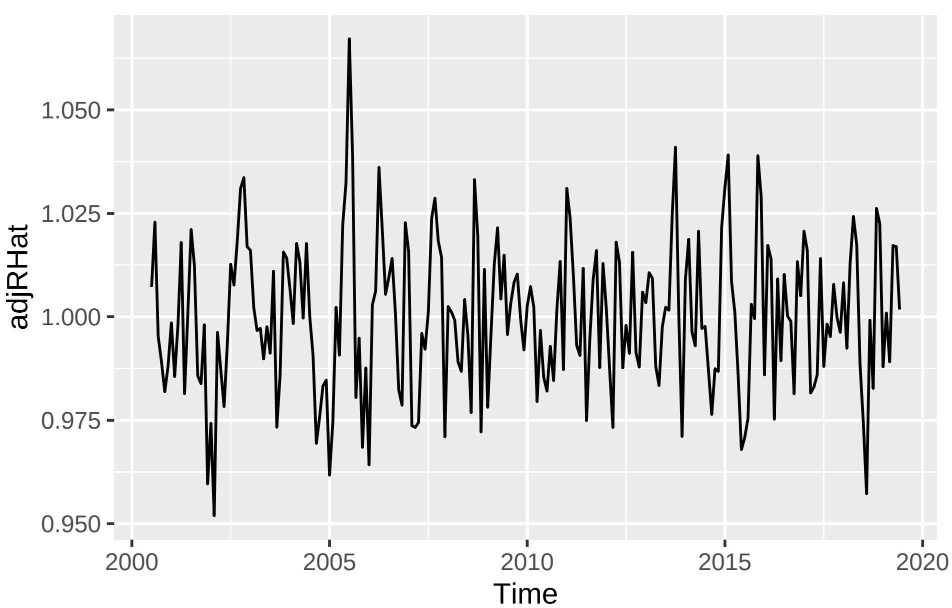


Figure 19: Remainder Series - Employment Goods-producing sector (Jan2000-Dec2019)

For the ACF and PACF plots we will use `lag.max=60` to visualize a broad period and identify if there are other possible dependencies and/or patters that might indicate some additional seasonality/trend not recovered by our process in Question 2.

```
# Plotting the ACF/PACF for Remainder
p1 <- ggAcf(adjRHat, lag.max = 60)+
  ggtitle("ACF RHat")
p2 <- ggPacf(adjRHat, lag.max = 60)+
  ggtitle("PACF RHat")
```

```r
gridExtra::grid.arrange(p1, p2, nrow = 1)
```

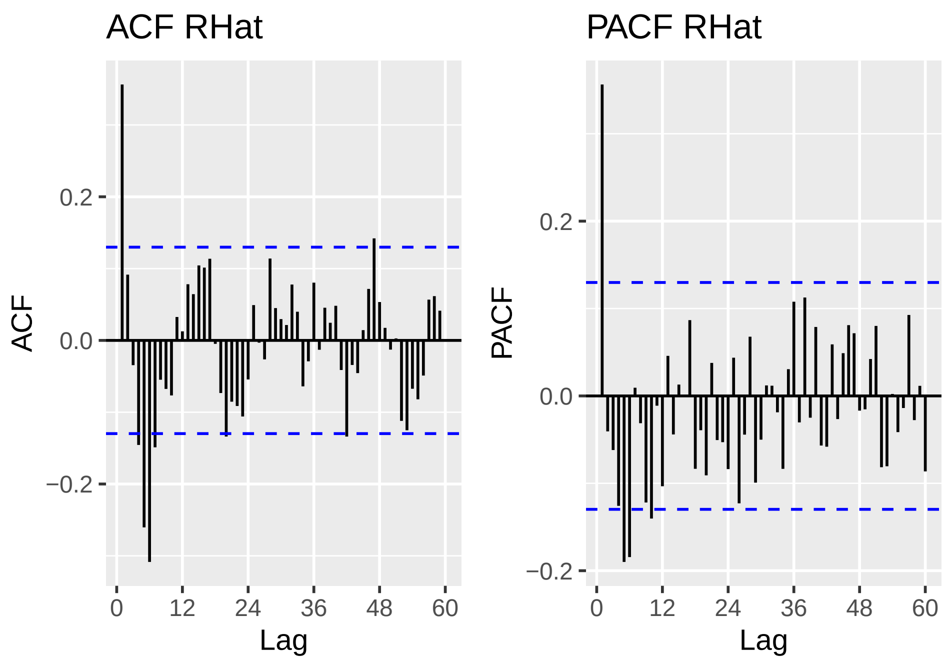


Figure 20: ACF/PACF - Remainder Series for Employment Goods-producing sector (Jan2000-Dec2019)

An additional plot that can help in evaluating *seasonal* patterns is the period-plot, in our case the `monthplot`, as follows.

```r
ggmonthplot(adjRHat)
```

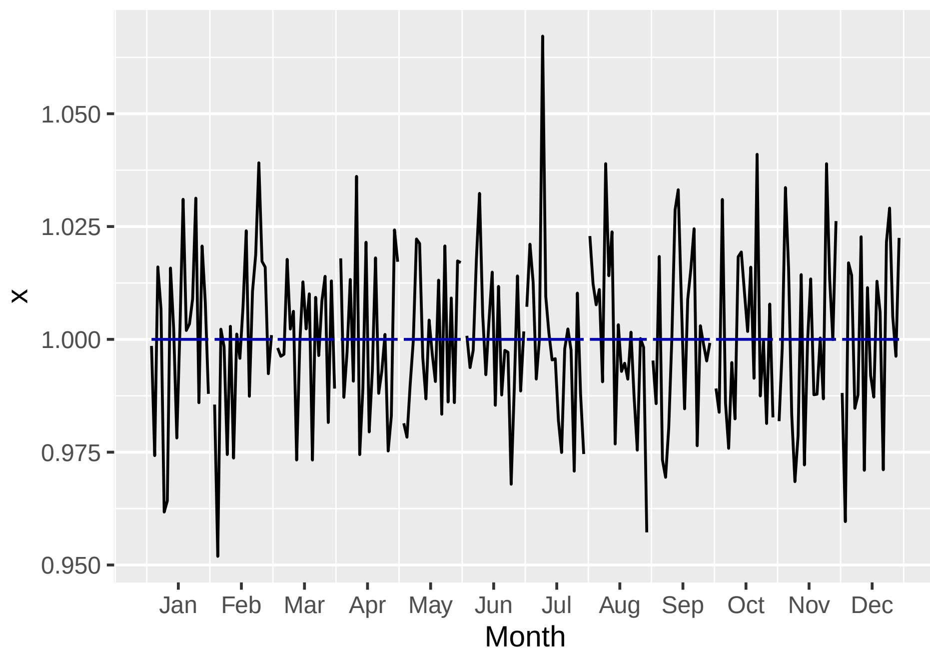


Figure 21: Monthplot of Remainder Series - Employment Goods-producing sector (Jan2000-Dec2019)

The `monthplot` doesn't show evidences of additional hidden seasonality and both ACF/PACF graphs shows an oscilating/not decreasing pattern - specially after `lag=72`, probably affected by 2008 sub-prime crisis - which might suggest there is an additional hidden seasonality, probably with higher order, or *different behaviour* which could be obscured in the first decomposition.

In order to check this, we will perform another decomposition and analyze if there are additional factorization series and verify if *there is additional seasonal and trend components* that can be included on the original decomposition, as follows:

$$y_t = \hat{T}_t \times \hat{S}_t \times \hat{R}_t = \hat{T}_t \times \hat{S}_t \times (\hat{T}^{\star}_t \times \hat{S}^{\star}_t \times \hat{R}^{\star}_t)$$

which leads to the following *2nd-order decomposition* of the series:

$$y_t = (\hat{T}_t \times \hat{T}^{\star}_t) \times (\hat{S}_t \times \hat{S}^{\star}_t) \times \hat{R}^{\star}_t$$

In R, we have then:

```
# Performing an additional multiplicative decomposition of the remainder

# Calculate the 2nd order-trend
THatStar <- RHat %>%
  stats::filter(c(.5, rep(1,(m-1)), .5)/m)

Det_uaStar <- RHat/THatStar

# Calculate the 2nd order-seasonality
for (i in 1:m)
  MPer[i] <- mean(Det_uaStar[v[i,1:nper]], na.rm = TRUE)

# Replicate to all series
```

```
SHatStar <- ts(rep(MPer, nper), start=head(time(RHat), 1), frequency=frequency(RHat))

Dets_uaStar <- Det_uaStar/SHatStar

# Calculate the 2nd order-remainder
RHatStar <- RHat/(SHatStar*THatStar)
```

```
# Calculating the new seasonality and trend of 2nd order
s2OrderTHat <- tail(head((THatStar*THat),-m),-m)
s2OrderSHat <- tail(head((SHatStar*SHat),-m),-m)

# Comparing the newly decomposed series - 2nd order
autoplot(THat, series="1stDec") +
  autolayer(s2OrderTHat, series="2ndDec") +
  xlab("Year") + ylab("Employment") +
  scale_colour_manual(values=c("gray","blue"),
             breaks=c("1stDec","2ndDec"))
```

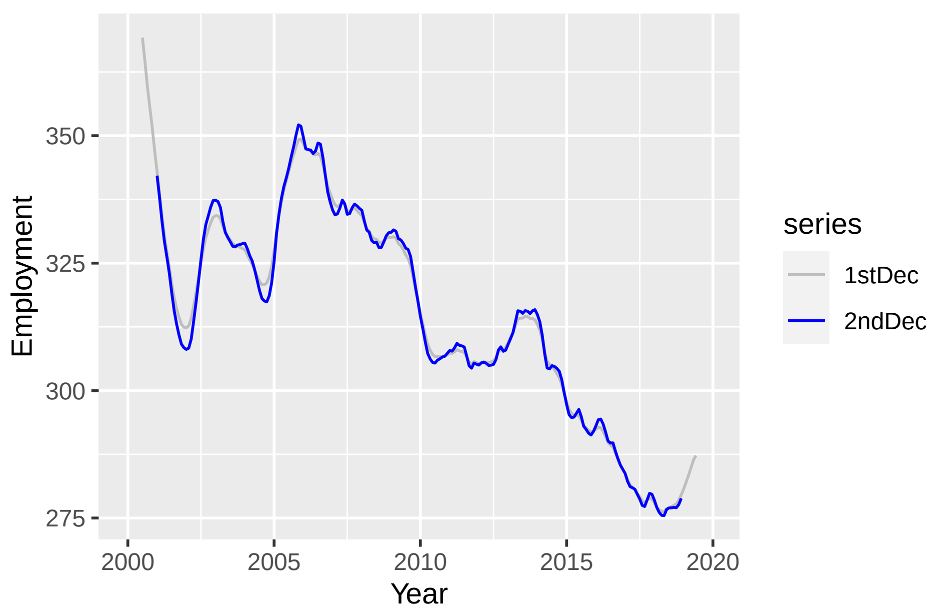


Figure 22: Trend of 2nd Order decomposition - Employment Goods-producing sector (Jan2000-Dec2019)

```
# Comparing the newly decomposed series - 2nd order
autoplot(SHat, series="1stDec") +
  autolayer(s2OrderSHat, series="2ndDec") +
  xlab("Year") + ylab("Value") +
  scale_colour_manual(values=c("gray","blue"),
             breaks=c("1stDec","2ndDec"))
```

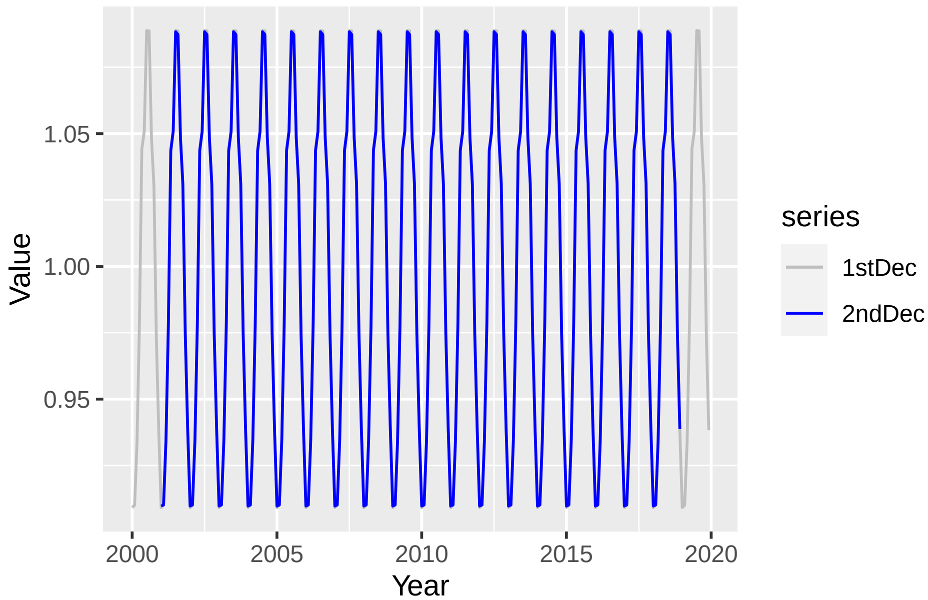


Figure 23: Seasonality of 2nd Order decomposition - Employment Goods-producing sector (Jan2000-Dec2019)

```
# Plotting the ACF/PACF for 2nd-order Remainder
p1 <- ggAcf(RHatStar, lag.max = 60)+
  ggtitle("ACF 2nd-Order RHat")
p2 <- ggPacf(RHatStar, lag.max = 60)+
  ggtitle("PACF 2nd-Order RHat")

gridExtra::grid.arrange(p1, p2, nrow = 1)
```

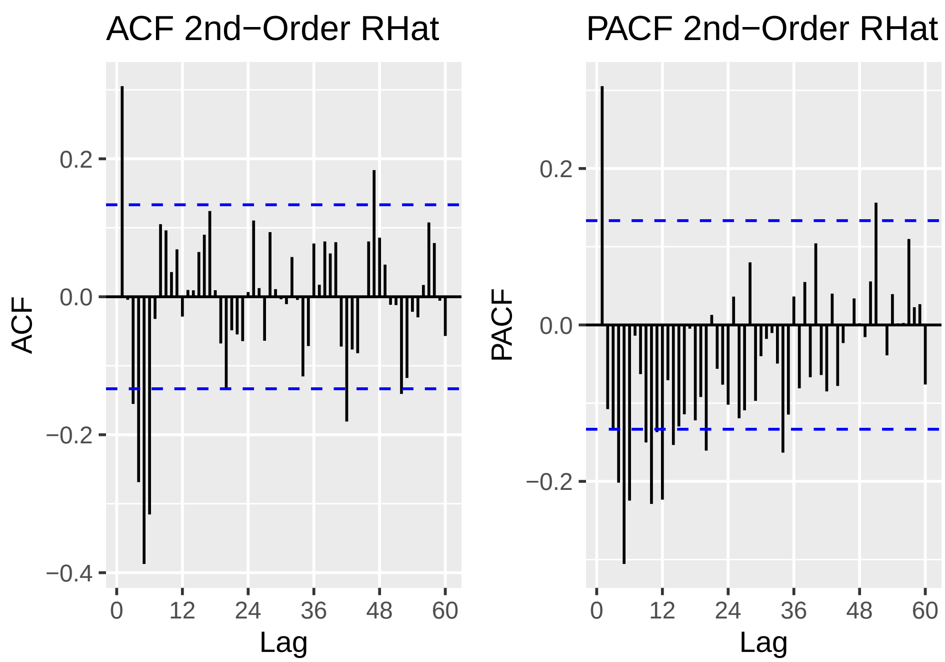


Figure 24: ACF/PACF of 2nd Order decomposition - Employment Goods-producing sector (Jan2000-Dec2019)

We can observe a small gain in in incorporating the new 2nd-Order trend $\hat{T}^*_t$ but in terms of seasonality, $\hat{S}^*_t$ doesn't aggregate significant information. Note that by doing this additional decomposition, information is lost on head and tails of the series due to calculating the detrended series, where we need to evaluate new 12-MA over the original remainder.

item (b)

Stationarity of the series

By analysing both ACF and PACF, whch have a regularity of behaviour around the 95% C.I. and the remainder series which seems to be randomly distributed around 1.0 (remember this is a multiplicative decomposition), we have no evidences that the weakly-stationarity is violated.

Nevertheless, additional processing could be done by splitting the series over some periods of know external influences, such as the crisis of 2008 and now, the COVID-19 pandemic, when incorporating the data from 2020. This could help understand in a more pure way what drivers can be considered when analysing the pattern of the employment in Goods Producing industry in Canada.

item (c)

Partial Autocorrelations and ARMA(p,q) model

The joint begaviour of the ACF/PACF plots from the seasonally-adjusted series we cand identify its ACF with $lags = 4$ with alternate positive/negative correlations outside the 95% C.I. interval, and the PACF, after $lag = 3$ the sample partial autocorrelations seems fall into the 95% C.I. which can suggest an `ARMA(4,3)` could be a fair model to begin testing and modelling the series.

```
# Checking the goodness of fit of the proposed model and check for inconsistencies
#astsa::sarima(adjRHat, 4,0,3,0,0,0,0)
chk_fit <- arima(adjRHat, order=c(4,0,3))
summary(chk_fit)
```

```
##
## Call:
## arima(x = adjRHat, order = c(4, 0, 3))
##
## Coefficients:
##          ar1     ar2      ar3      ar4      ma1      ma2     ma3  intercept
##       0.5787  0.7403  -0.5837  -0.0208  -0.2996  -0.9077  0.3003     1.0001
## s.e.  0.3010  0.1452   0.2522   0.1472   0.2945   0.0738  0.2805     0.0004
##
## sigma^2 estimated as 0.0002493:  log likelihood = 621.72,  aic = -1225.43
##
## Training set error measures:
##                         ME       RMSE         MAE         MPE     MAPE      MASE
## Training set -0.0003615998 0.01578788 0.01242421 -0.06116345 1.243805 0.7617133
##                     ACF1
## Training set 0.001581522
```

```
tsdiag(chk_fit)
```

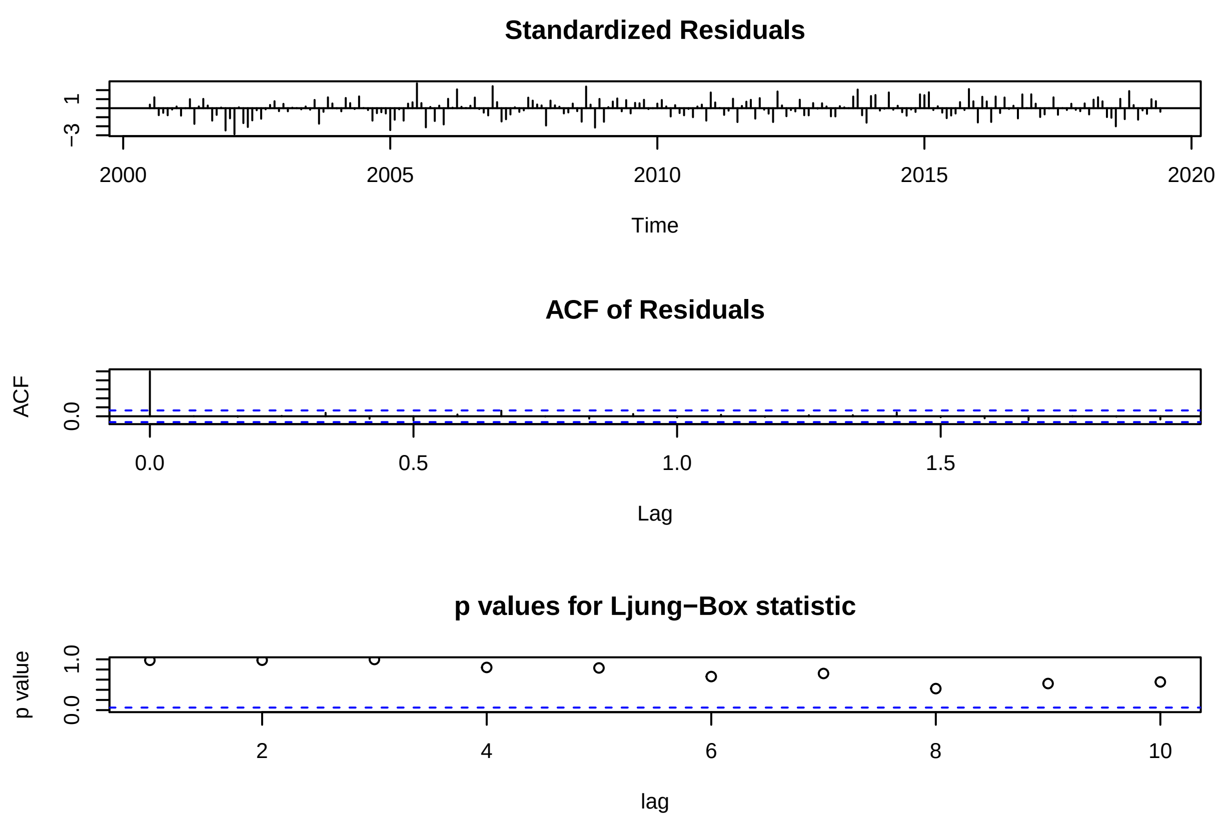


Figure 25: Check ARMA(4,3) - Employment Goods-producing sector (Jan2000-Dec2019)

```
qqnorm(chk_fit$residuals)
```

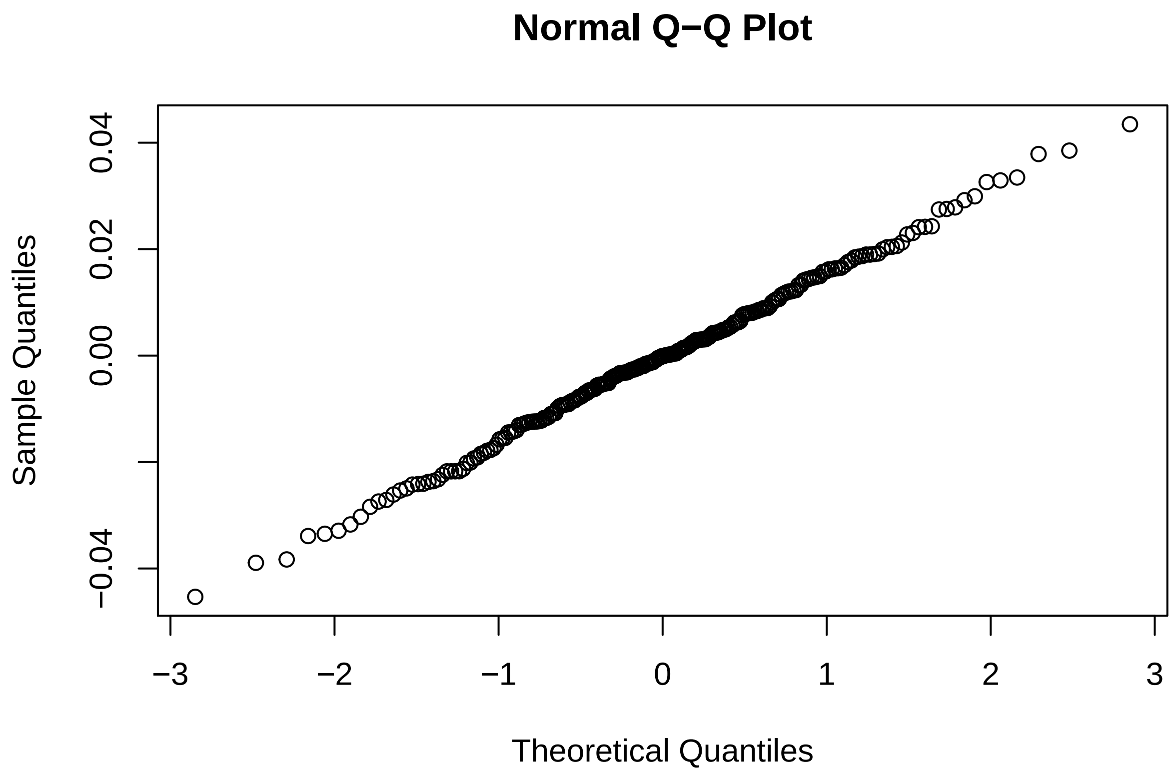


Figure 26: Check ARMA(4,3) - Employment Goods-producing sector (Jan2000-Dec2019)

After running a simulation using `arima()` and diagnostics with `tsdiag()`[2], we can then see that residuals seems to be uncorrelated and that there are not further structures in residuals that may require our attention, so we can see this could be a valid starting model for further analysis.

Now, if we were thinking about *proposing a model for the original series*, my approach would be a little different:

(1) Verify the ACF/PACF plots for de-seasonal series

```
# Plotting the ACF/PACF for de-seasonal series
Des_ua <- (ua/SHat)
# autoplot(Des_ua)
p1 <- ggAcf(Des_ua, lag.max = 24)+
  ggtitle("ACF De-Seasonal")
p2 <- ggPacf(Des_ua, lag.max = 24)+
  ggtitle("PACF De-Seasonal")

gridExtra::grid.arrange(p1, p2, nrow = 1)
```

[2]We know that the recommended option to check for model consistency would be use the `sarima()` function but in this case, we are just checking a preliminar model, with no major issue in our approach.

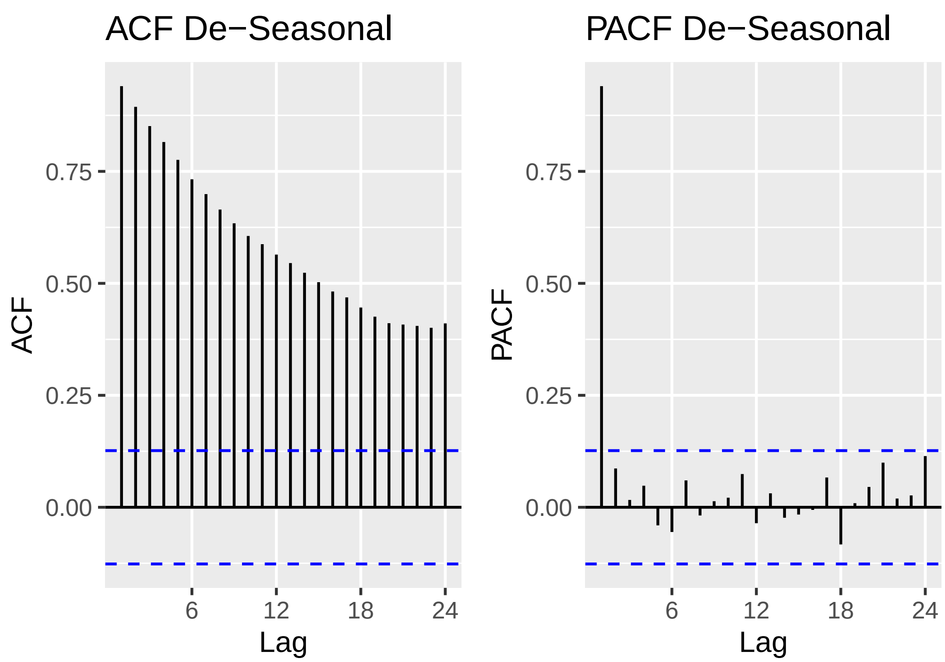


Figure 27: ACF/PACF - Employment Goods-producing sector (Jan2000-Dec2019)

As we can see, the ACF plot seems like a *Random-Walk* ACF and PACF has `lag=1` significant then I would propose an ARMA(1,1) model in this case

(2) Checking the proposed model ARMA(1,1):

```
# Checking the goodness of fit of the proposed model and check for inconsistencies
chk_fit2 <- arima(Des_ua, order=c(1,0,1))
summary(chk_fit2)
```

```
##
## Call:
## arima(x = Des_ua, order = c(1, 0, 1))
##
## Coefficients:
##          ar1      ma1  intercept
##       0.9905  -0.1983   331.4273
## s.e.  0.0098   0.0646    27.5958
##
## sigma^2 estimated as 42.36:  log likelihood = -791.87,  aic = 1591.74
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.6593883 6.508435 5.152577 -0.2301335 1.629731 0.9824986
##                    ACF1
## Training set -0.0159851
```

```
tsdiag(chk_fit2)
```

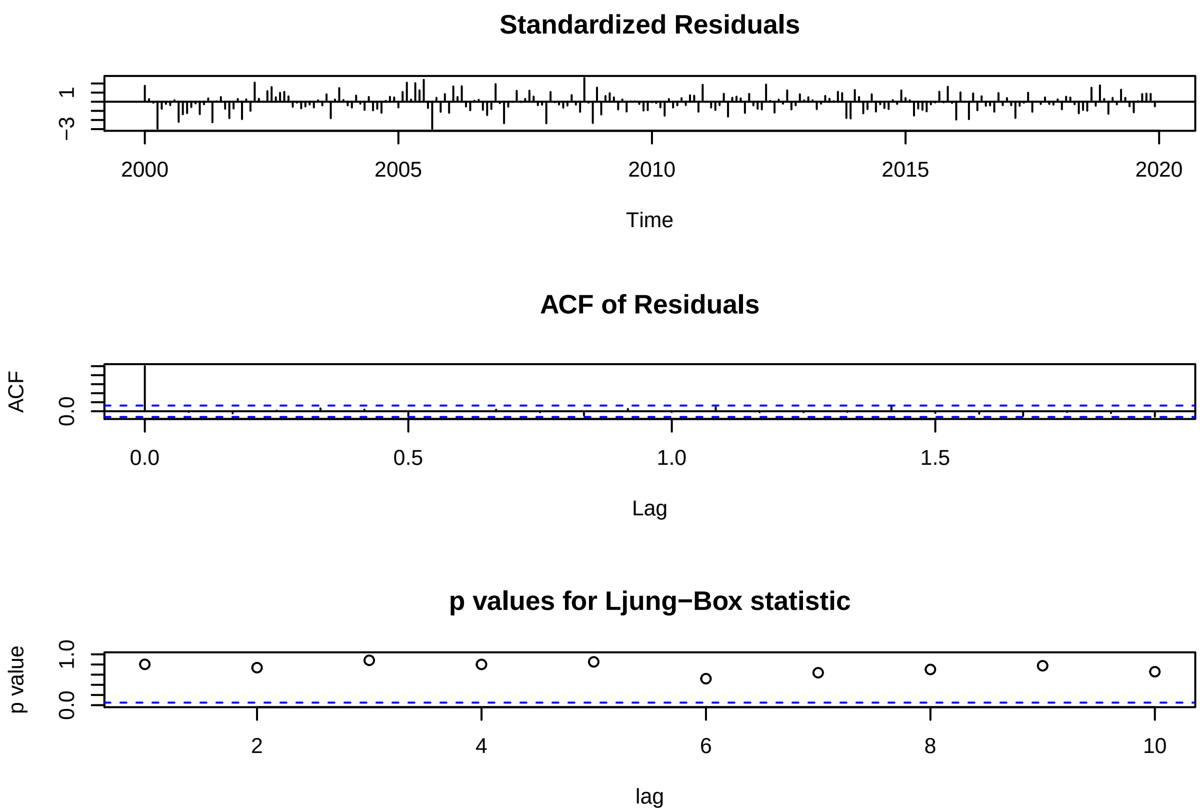


Figure 28: Check ARMA(1, 1) - Employment Goods-producing sector (Jan2000-Dec2019)

```
qqnorm(chk_fit2$residuals)
```

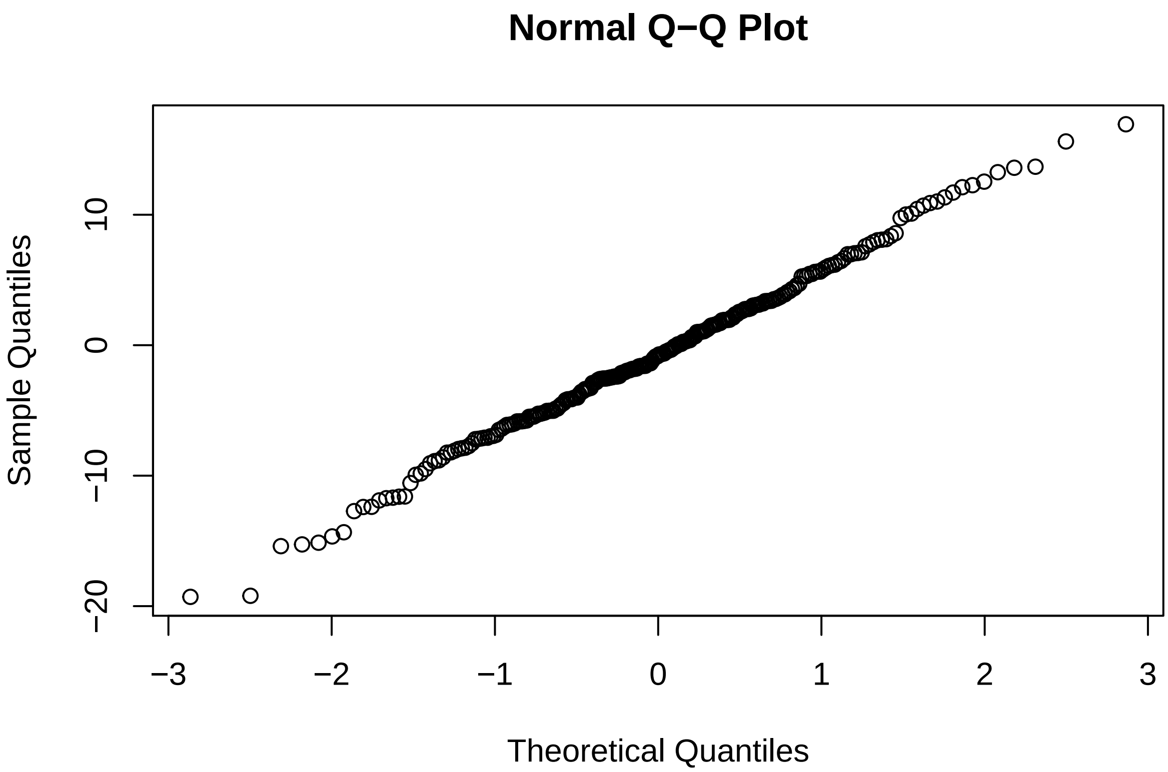


Figure 29: Check ARMA(1, 1) - Employment Goods-producing sector (Jan2000-Dec2019)

Question 6

6. [10 marks; **STA2202 (grad) students ONLY**] Download employment data *up to April 2020* (the most recent month) for *all* of the above industries, and use them to answer the following question: *Which industry's employment was hit hardest by the COVID-19 pandemic?*.
You need to back up your answer with valid arguments based on time series techniques, to account for things like seasonality (e.g., you can't simply rank last month's differences in employment numbers). Clearly explain your reasoning and the methods & metrics used for making comparisons.

last digit of student #	Industry	Unadjusted	Seasonally adjusted	Trend-cycle
1	Accommodation and food services	v2057828	v2057619	v123355122
2	Agriculture	v2057814	v2057605	v123355108
3	Construction	v2057817	v2057608	v123355111
4	Educational services	v2057825	v2057616	v123355119
5	Forestry, fishing, mining, quarrying, oil and gas	v2057815	v2057606	v123355109
6	Goods-producing sector	v2057813	v2057604	v123355107
7	Information, culture and recreation	v2057827	v2057618	v123355121
8	Manufacturing	v2057818	v2057609	v123355112
9	Public administration	v2057830	v2057621	v123355124
0	Services-producing sector	v2057819	v2057610	v123355113

2nd to last digit of student #	Time period
odd	Jan 1980 to Dec 1999
even	Jan 2000 to Dec 2019

{*Solution.*}

We will download the following database organized into a `data-frame` as follows:

```
DBMaster  <- data.frame (
  Industry = c("Accommodation and food services", "Agriculture", "Construction",
               "Educational services", "Forestry, fishing, mining, quarrying, oil and gas",
               "Goods-producing sector", "Information, culture and recreation",
               "Manufacturing", "Public administration",
               "Services-producing sector"),
  Unadj = c("v2057828", "v2057814", "v2057817", "v2057825", "v2057815", "v2057813",
            "v2057827", "v2057818", "v2057830", "v2057819" ),
  SeasonAdj = c("v2057619", "v2057605", "v2057608", "v2057616", "v2057606", "v2057604",
                "v2057618", "v2057609", "v2057621", "v2057610"),
  TrendCycle = c("v123355122", "v123355108", "v123355111", "v123355119", "v123355109",
                 "v123355107", "v123355121", "v123355112", "v123355124", "v123355113")
)

kableExtra::kable(DBMaster, "latex", booktabs = TRUE,
                  caption = "Employment Data per Industry in Canada")
```

Table 6: Employment Data per Industry in Canada

Industry	Unadj	SeasonAdj	TrendCycle
Accommodation and food services	v2057828	v2057619	v123355122
Agriculture	v2057814	v2057605	v123355108
Construction	v2057817	v2057608	v123355111
Educational services	v2057825	v2057616	v123355119
Forestry, fishing, mining, quarrying, oil and gas	v2057815	v2057606	v123355109
Goods-producing sector	v2057813	v2057604	v123355107
Information, culture and recreation	v2057827	v2057618	v123355121
Manufacturing	v2057818	v2057609	v123355112
Public administration	v2057830	v2057621	v123355124
Services-producing sector	v2057819	v2057610	v123355113

Now reading all databases of the de-seasonal series in a single structure. The reason to work with the adjusted (de-seasonal) series is to eliminate seasonal aspects that might interfere in the comparisons between industries and to have a better visualization of trend in order to compare the data.

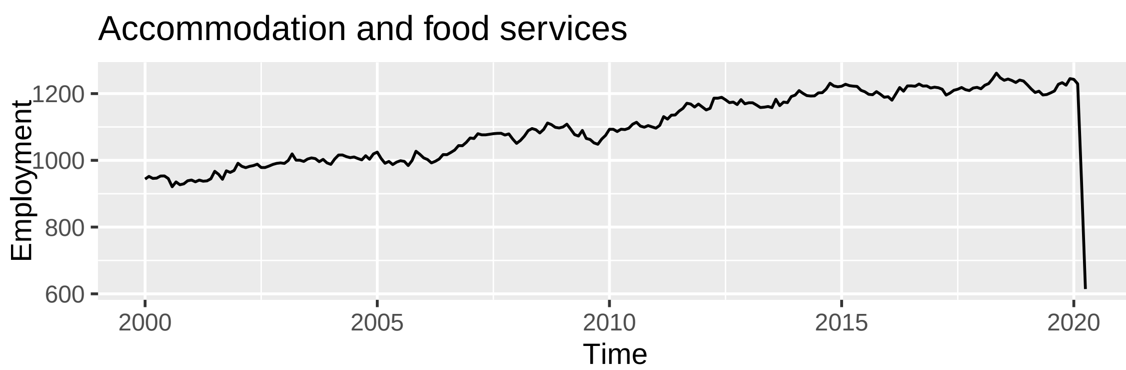
```
# 1 - Create data-types to store databases (unadjusted/raw version)
NullSeries <- data.frame (
  Series = ts(rep(0,244), start=c(2000,1), frequency = 12)
)

# Creating the data-frame to accomodate all series in a single structure
dt.uaD = data.frame (
  dtU = c(NullSeries, NullSeries, NullSeries, NullSeries, NullSeries,
          NullSeries, NullSeries, NullSeries, NullSeries, NullSeries )
)

# Renaming columns to get it more user friendly
colnames(dt.uaD) = c(paste0("Series_",1:10))

# Load entire database with the most recent data for all industries
for (i in 1:nrow(DBMaster)) {
  uaDTmp <- get_cansim_vector( DBMaster$Unadj[i],
                               start_time = "2000-01-01", end_time = "2020-04-01") %>%
  pull(VALUE) %>% ts( start = c(2000,1), frequency = 12)
  dt.uaD[,paste0("Series_",i)] <- uaDTmp
}


# 2 - Crea <- e data-types to store databases (de-seasonal)
# Creating the data-frame to accomodate all series in a single structure
dt.SeaAdj = data.frame (
  dtS = c(NullSeries, NullSeries, NullSeries, NullSeries, NullSeries,
          NullSeries, NullSeries, NullSeries, NullSeries, NullSeries )
)

# Renaming columns to get it more user friendly
colnames(dt.SeaAdj) = c(paste0("Series_",1:10))

# Load entire database with the most recent data for all industries
```

```
for (i in 1:nrow(DBMaster)) {
  SeaTmp <- get_cansim_vector( DBMaster$SeasonAdj[i],
                               start_time = "2000-01-01", end_time = "2020-04-01") %>%
  pull(VALUE) %>% ts( start = c(2000,1), frequency = 12)
  dt.SeaAdj[,paste0("Series_",i)] <- SeaTmp
}

# 3 - Create data-types to store databases (trend cycle)
# Creating the data-frame to accomodate all series in a single structure
dt.Trend = data.frame (
  dtT = c(NullSeries, NullSeries, NullSeries, NullSeries, NullSeries,
          NullSeries, NullSeries, NullSeries, NullSeries, NullSeries )
)

# Renaming columns to get it more user friendly
colnames(dt.Trend) = c(paste0("Series_",1:10))

# Load entire database with the most recent data for all industries
for (i in 1:nrow(DBMaster)) {
  TrendTmp <- get_cansim_vector( DBMaster$TrendCycle[i],
                                 start_time = "2000-01-01", end_time = "2020-04-01") %>%
  pull(VALUE) %>% ts( start = c(2000,1), frequency = 12)
  dt.Trend[,paste0("Series_",i)] <- TrendTmp
}
```

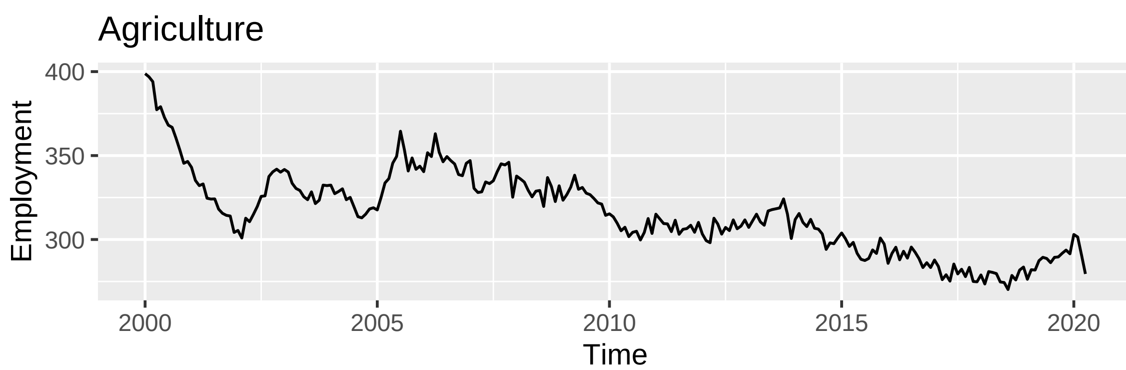
After downloading all databases, I decided to work with the *seasonal-adjusted* series to avoid particularities of each industry to affect the comparisons between series.

The strategy here will be as follows:

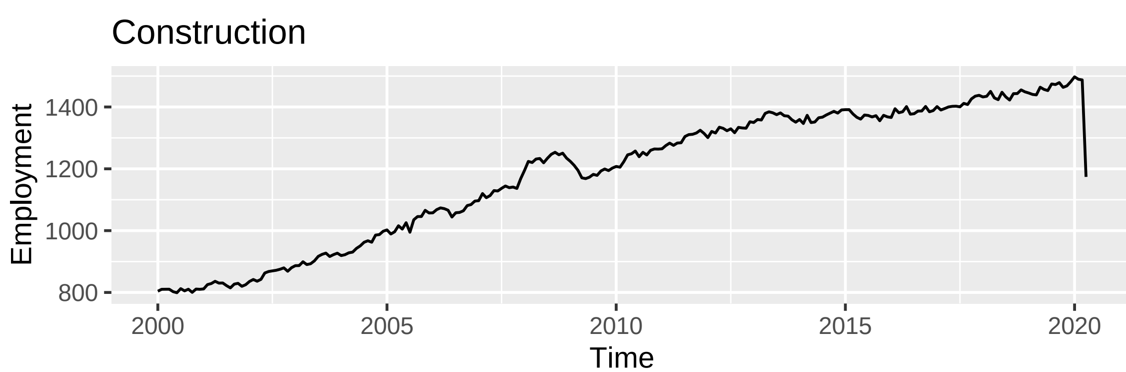
- Visualize each de-seasonal graph to verify the probable candidates and if there is additional aspects to be considered to analyze the data;
- Proceed with a *standardization* of each series in order to enable comparisons;
- Verify the changes during the last months where COVID has been active in Canada to identify which industry presented the most significant impact.



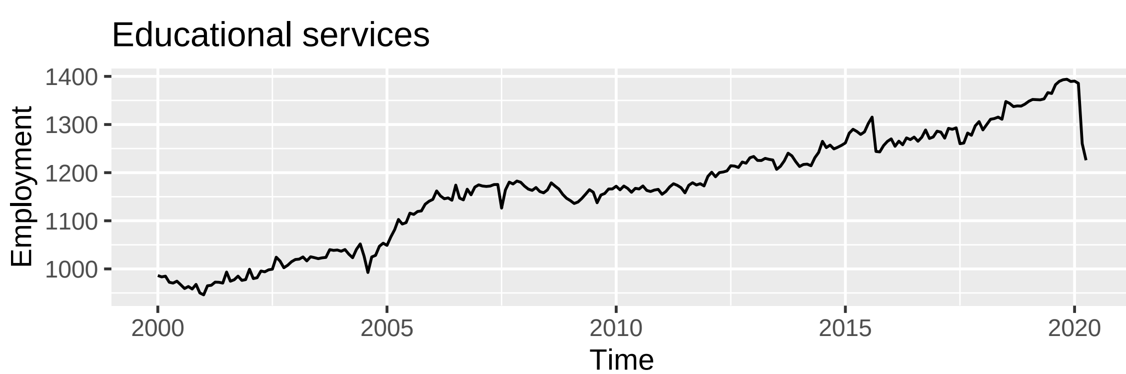
Accommodation and food services
Employment
1200
1000
800
600
2000
2005
2010
2015
2020
Time



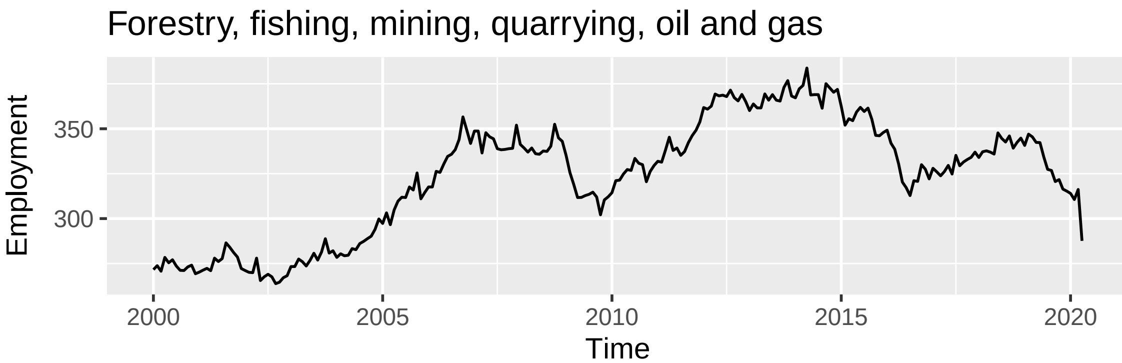
Agriculture
Employment
400
350
300
2000
2005
2010
2015
2020
Time

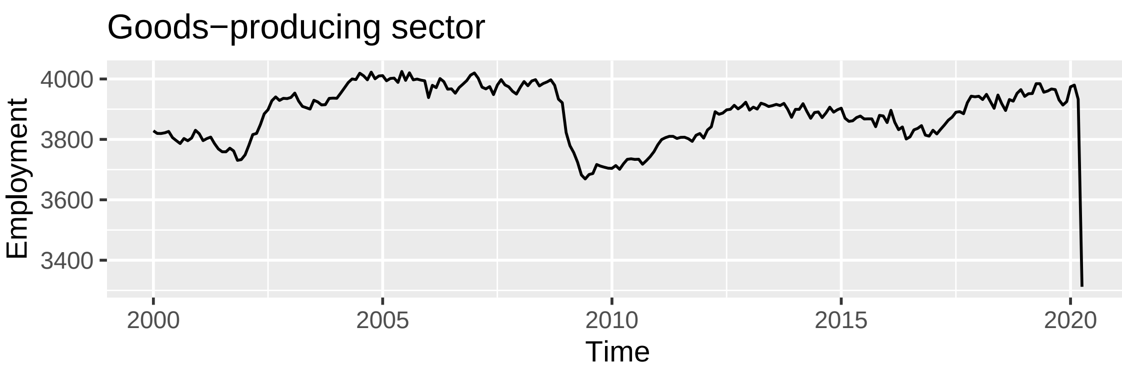


Construction
Employment
1400
1200
1000
800
2000
2005
2010
2015
2020
Time

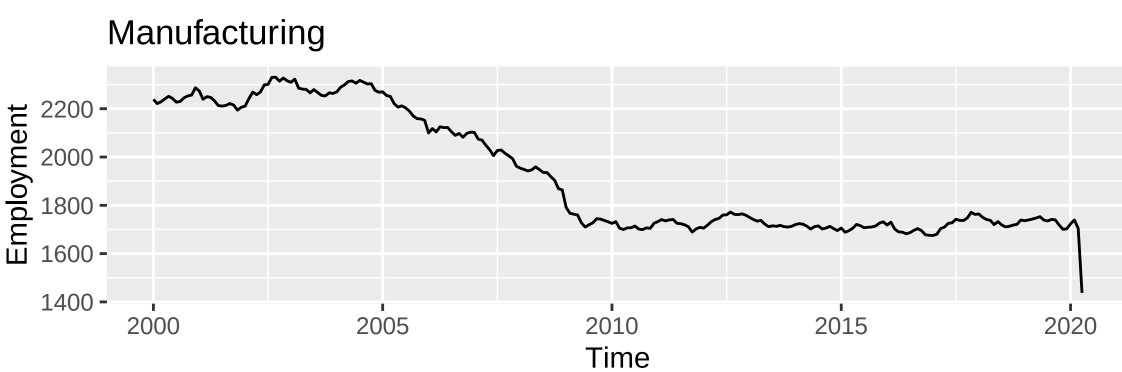


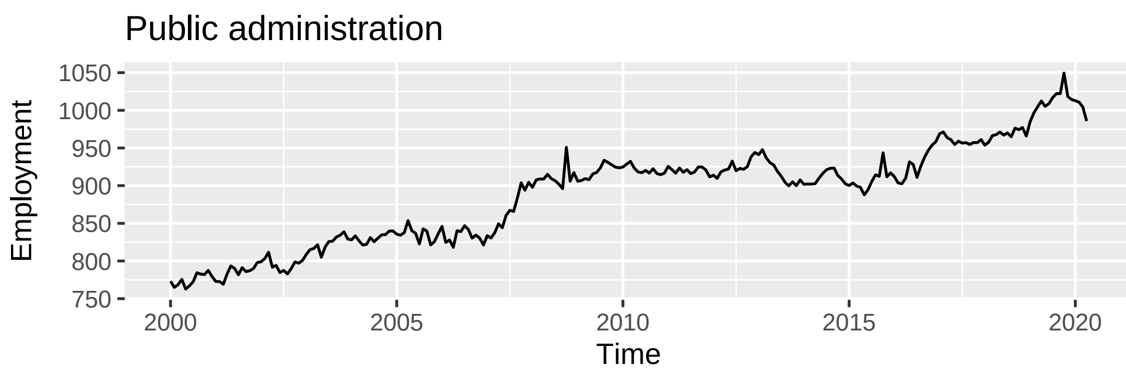
Educational services
Employment
1400
1300
1200
1100
1000
2000
2005
2010
2015
2020
Time

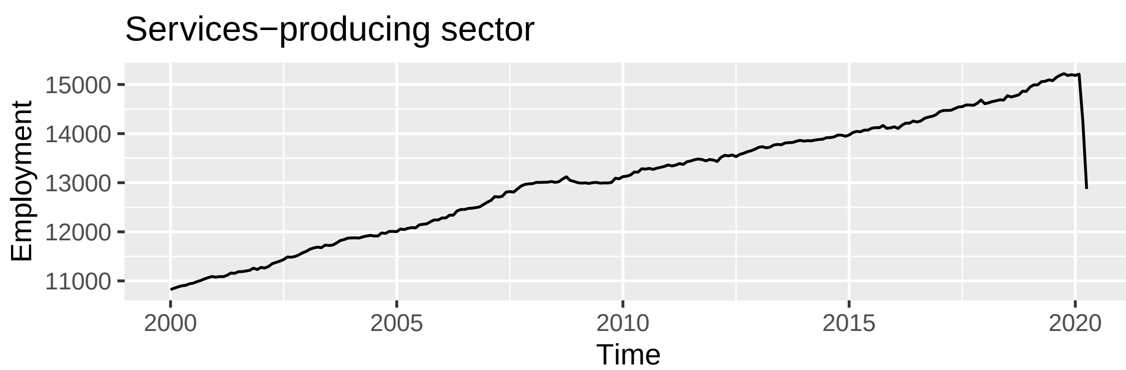

Forestry, fishing, mining, quarrying, oil and gas
Employment
350
300
2000
2005
2010
2015
2020
Time


Goods−producing sector
Employment
4000
3800
3600
3400
2000
2005
2010
2015
2020
Time


Information, culture and recreation
Employment
800
750
700
650
600
2000
2005
2010
2015
2020
Time


Manufacturing
Employment
2200
2000
1800
1600
1400
2000
2005
2010
2015
2020
Time





Following our strategy, we will now create *stardardized* versions of each series in order to enable one-to-one comparisons.

Important to mention that we are assuming here that COVID has been in place since *Feb/2020* in Canada, and the most recent update is *Apr/2020* we will use `lag=3` in order to put in evidence the differences of level of employment on each industry.

```
## Create versions of Standardized Series bases on Seasonally adjusted ones
dt.StdS = data.frame (
  dtS = c(NullSeries, NullSeries, NullSeries, NullSeries, NullSeries,
          NullSeries, NullSeries, NullSeries, NullSeries, NullSeries )
)

# Renaming columns to get it more user friendly
colnames(dt.StdS) = c(paste0("Series_",1:10))

# This procedure standardize the Time-Series
for (i in 1:nrow(DBMaster)) {
  r <- range(dt.SeaAdj[,paste0("Series_",i)])
  dt.StdS[,paste0("Series_",i)] <- (dt.SeaAdj[,paste0("Series_",i)]-
                                    min(dt.SeaAdj[,paste0("Series_",i)]))/(r[2]-r[1])
}
```

```
# Set initial variables
lg <- 3

CmpInd <- data.frame (
  IxInd = 1:nrow(DBMaster),
  Industry = DBMaster$Industry,
  ImpactStd1 = rep(0, nrow(DBMaster)),
  ImpactStd2 = rep(0, nrow(DBMaster)),
```

```
  ImpactStd3 = rep(0, nrow(DBMaster))
)

for (i in 1:nrow(DBMaster)) {
  CmpInd$ImpactStd1[i] <- head(tail(diff(dt.StdS[,paste0("Series_",i)], lag=lg),1),1)
  CmpInd$ImpactStd2[i] <- head(tail(diff(dt.StdS[,paste0("Series_",i)], lag=lg),2),1)
  CmpInd$ImpactStd3[i] <- head(tail(diff(dt.StdS[,paste0("Series_",i)], lag=lg),3),1)
}

TmpTb <- CmpInd %>%
  arrange(ImpactStd1)

colnames(TmpTb) <- c("Rank", "Industry", "Apr/20", "Mar/20", "Feb/20")

kableExtra::kable(TmpTb, "latex", booktabs = TRUE,
                  caption = "COVID-19 Impact Assessment over Industries in Canada ")
```

Table 7: COVID-19 Impact Assessment over Industries in Canada

Rank	Industry	Apr/20	Mar/20	Feb/20
1	Accommodation and food services	-0.9710974	-0.4789799	0.0063369
6	Goods-producing sector	-0.9287418	0.0105204	0.0927199
7	Information, culture and recreation	-0.8184424	-0.4290492	0.0931677
10	Services-producing sector	-0.5258585	-0.2168297	0.0059131
3	Construction	-0.4634880	0.0081615	0.0306415
4	Educational services	-0.3671875	-0.2883929	-0.0189732
8	Manufacturing	-0.3198925	0.0022401	0.0432348
5	Forestry, fishing, mining, quarrying, oil and gas	-0.2200000	0.0075000	-0.0483333
2	Agriculture	-0.1827372	-0.0062208	0.0606532
9	Public administration	-0.0938590	-0.0352408	-0.0258200

In this table we present the impact on each industry during the last 03-months, period when COVID-19 has been mostly active in every country other than China.

As we can see, the industry **Accommodation and food services** as the most affected by COVID-19 during Mar-Apr/20.

Another approach is to use the *percentual variation* of employment on each industry and then compare which one had the most significant impact during the period of Mar-Apr/20.

```
# Calculate the Percentual Variation
dt.VarPct = data.frame (
  dtS = c(NullSeries, NullSeries, NullSeries, NullSeries, NullSeries,
          NullSeries, NullSeries, NullSeries, NullSeries, NullSeries )
)

# Renaming columns to get it more user friendly
colnames(dt.VarPct) = c(paste0("Series_",1:10))

# This procedure calculates the percentual variation of each Time-Series
for (i in 1:nrow(DBMaster)) {
  dt.VarPct[,paste0("Series_",i)] <-
    (dt.SeaAdj[,paste0("Series_",i)]/c(NA,head(dt.SeaAdj[,paste0("Series_",i)],-1))-1)
```

```
}

# Get the last 03 months to analyze pct variation
CmpInd$ImpactStd1[1:10] <- as.matrix(dt.VarPct %>%
  filter(row_number() == nrow(dt.SeaAdj)))
CmpInd$ImpactStd2[1:10] <- as.matrix(dt.VarPct %>%
  filter(row_number() == (nrow(dt.SeaAdj)-1)))
CmpInd$ImpactStd3[1:10] <- as.matrix(dt.VarPct %>%
  filter(row_number() == (nrow(dt.SeaAdj)-2)))

# Elaborate a temporary table to manipulate data
TmpTb <- CmpInd %>%
  arrange(ImpactStd1)

colnames(TmpTb) <- c("Rank", "Industry", "Apr/20", "Mar/20", "Feb/20")

kableExtra::kable(TmpTb, "latex", booktabs = TRUE,
                  caption = "COVID-19 Pct. Variation in Employment over Industries in Canada ")
```

Table 8: COVID-19 Pct. Variation in Employment over Industries in Canada

Rank	Industry	Apr/20	Mar/20	Feb/20
1	Accommodation and food services	-0.3428877	-0.2394664	-0.0107025
3	Construction	-0.2108235	-0.0014768	-0.0052751
6	Goods-producing sector	-0.1577837	-0.0118599	0.0014091
8	Manufacturing	-0.1566845	-0.0198367	0.0092851
7	Information, culture and recreation	-0.1214815	-0.1331707	0.0188408
10	Services-producing sector	-0.0963983	-0.0633481	0.0016200
5	Forestry, fishing, mining, quarrying, oil and gas	-0.0904491	0.0180296	-0.0108280
2	Agriculture	-0.0385277	-0.0358209	-0.0049505
4	Educational services	-0.0272973	-0.0905023	-0.0033806
9	Public administration	-0.0184208	-0.0065288	-0.0017774

This confirms the industry of **Accommodation and food services** was most affected by COVID-19 with -34.3% decrease in employment in Apr/20 preceeded by -23.9% decrease in Mar/20.

Here a comparison of *1st* and *2nd* places on each approach in the last 5 years to illustrate what we got numerically.

```
# Plot Industries with 1st vs. 2nd and 1st vs. last  impact

p1 <- autoplot(window(dt.StdS[,paste0("Series_",1)],2015,c(2020,4)), series = "1st") +
  autolayer(window(dt.StdS[,paste0("Series_",6)],2015,c(2020,4)), series="2nd") +
  ylab("Employment\nStandardized") +
  scale_color_manual(name = "",
                     values=c("1st" = "black","2nd" = "red"),
                     labels = c("1st" = DBMaster$Industry[1], "2nd" = DBMaster$Industry[6]))
p2 <- autoplot(window(dt.VarPct[,paste0("Series_",1)],2015, c(2020,4)), series = "1st") +
  autolayer(window(dt.VarPct[,paste0("Series_",3)],2015,c(2020,4)), series="2nd") +
  ylab("Employment\nPct.Variation") +
  scale_color_manual(name = "",
                     values=c("1st" = "black","2nd" = "red"),
```

```
                        labels = c("1st" = DBMaster$Industry[1], "2nd" = DBMaster$Industry[3]))
grid.newpage()
grid.draw(rbind(ggplotGrob(p1), ggplotGrob(p2), size = "last"))
```

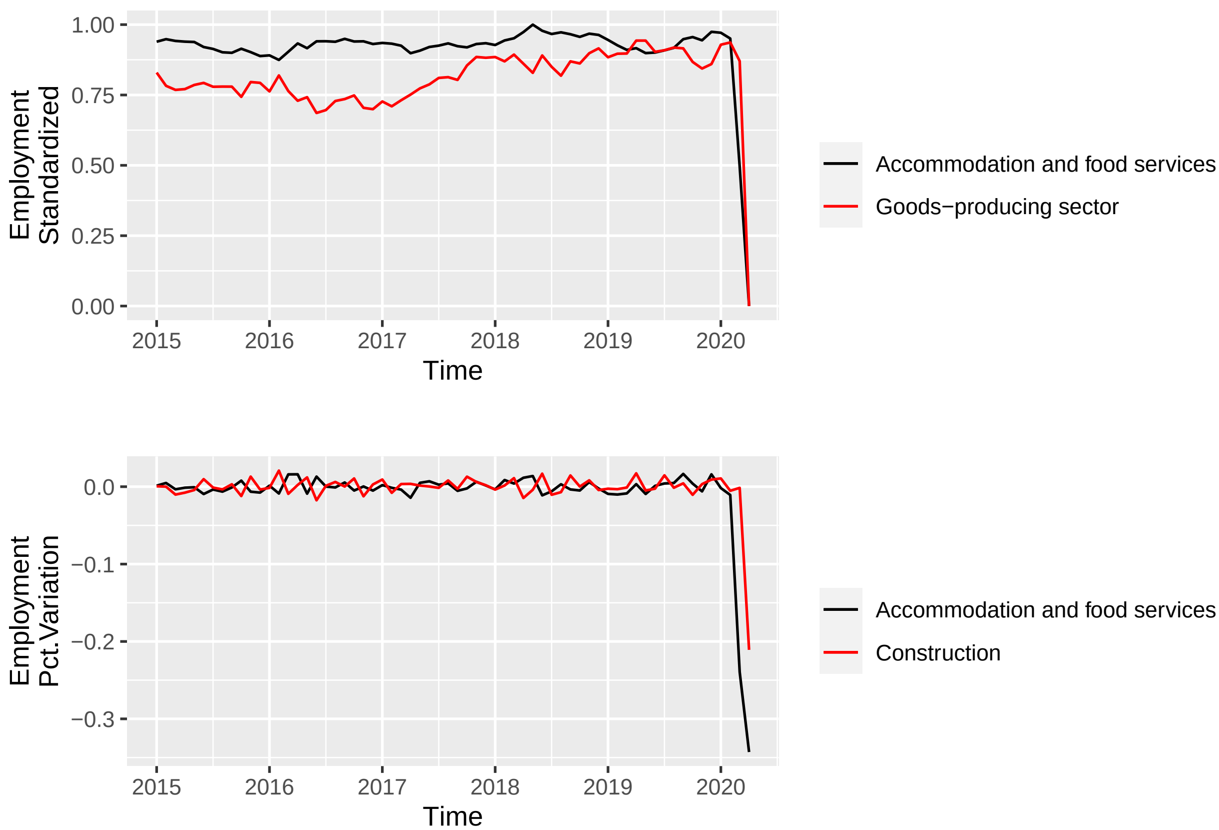


Figure 30: Industry Compare - COVID-19 Impact

This concludes the Question 6.