Assignment 1

Applied Forecasting in Complex Systems 2022

Yanchao MURONG (14090759)

University of Amsterdam
November, 7, 2022

Exercise 1

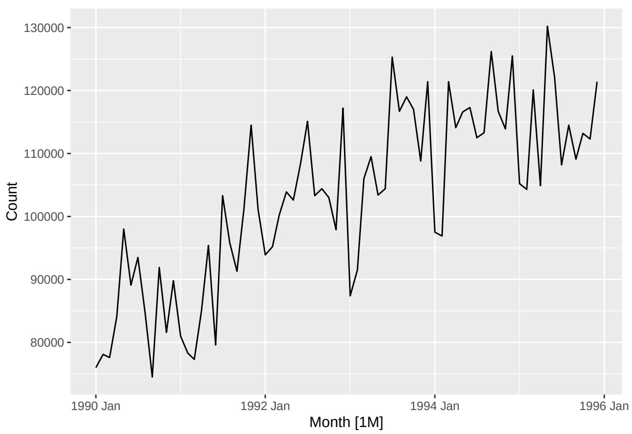
1.1)

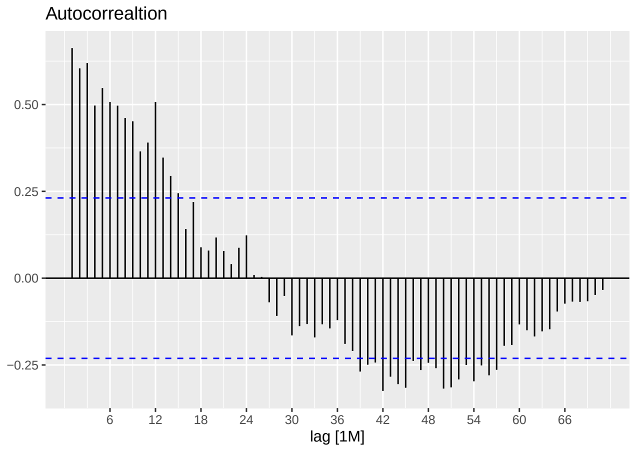
```
pigs <- aus_livestock %>%
  filter(State == "Victoria" & Animal == "Pigs" & between(year(Month), 1990, 1995))
pigs
```

```
## # A tsibble: 72 x 4 [1M]
## # Key:       Animal, State [1]
##       Month Animal State    Count
##       <mth> <fct>  <fct>    <dbl>
##  1 1990 Jan Pigs   Victoria 76000
##  2 1990 Feb Pigs   Victoria 78100
##  3 1990 Mar Pigs   Victoria 77600
##  4 1990 Apr Pigs   Victoria 84100
##  5 1990 May Pigs   Victoria 98000
##  6 1990 Jun Pigs   Victoria 89100
##  7 1990 Jul Pigs   Victoria 93500
##  8 1990 Aug Pigs   Victoria 84700
##  9 1990 Sep Pigs   Victoria 74500
## 10 1990 Oct Pigs   Victoria 91900
## # ... with 62 more rows
```

1.2)

```
par(mar = c(4, 4, .1, .1))
pigs %>% autoplot(Count)
pigs %>% ACF(Count, lag_max = 100) %>%
  autoplot() +
  labs(title = "Autocorrealtion", y = "")
```

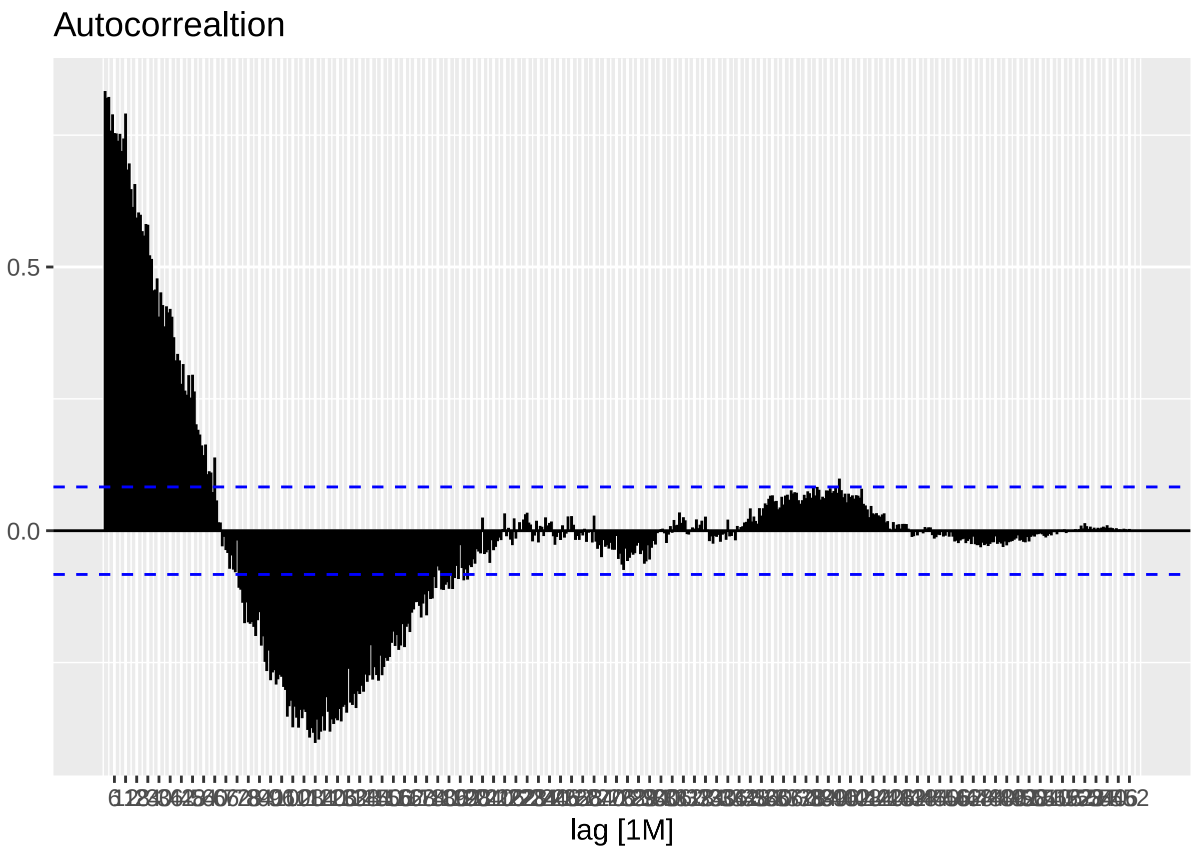




For white noise series, we expect each autocorrelation to be close to zero and less than 5% of splikes should be outside the blue bounds. However, from the above figure on the right, the values slowly decrease as the lags increase and we observe a lot of autocorrealtion coefficients lie outside the bounds. As a matter of fact, the ACF figure actually demontrates trends and seasonality caracteristics.

1.3)

```
all_pigs <- aus_livestock %>%
  filter(State == "Victoria" & Animal == "Pigs")
all_pigs %>% ACF(Count, lag_max = 1000) %>%
  autoplot() +
  labs(title = "Autocorrealtion", y = "")
```

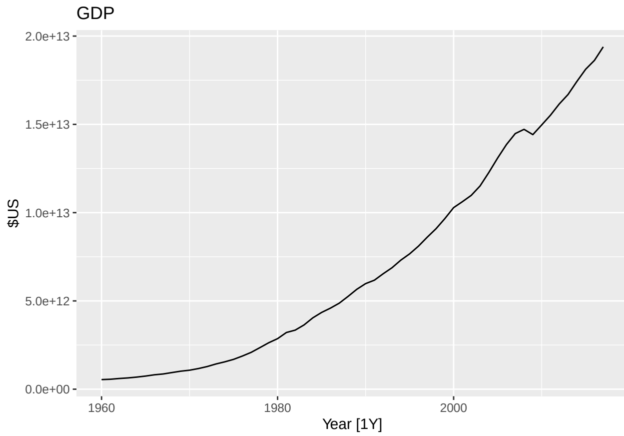


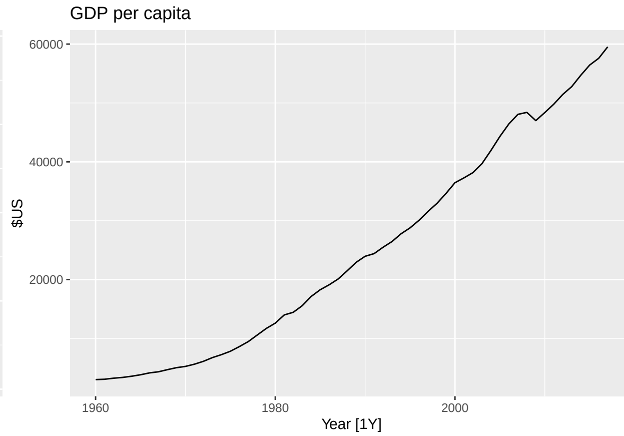
Now, if we include the whole pigs data from 1972 to 2018 and draw the ACF plot, interestingly, when the lags become larger than 240, the data seems to demonstrate a white noise pattern, which might indicate that the pigs slaughtering pattern has been tremendously changed after 20 years and would be difficult to do predictions based the data 20 years ago.

Exercise 2

2.1)

```
par(mar = c(4, 4, .1, .1))

global_economy %>%
  filter(Country=="United States") %>%
  autoplot(GDP) +
  labs(title= "GDP", y = "$US")

global_economy %>%
  filter(Country=="United States") %>%
  autoplot(GDP/Population) +
  labs(title= "GDP per capita", y = "$US")
```

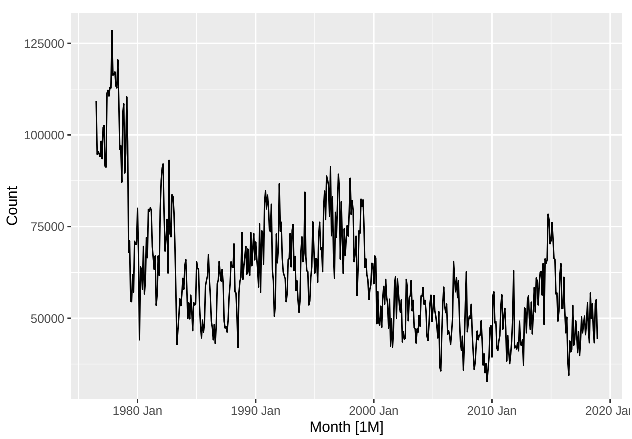


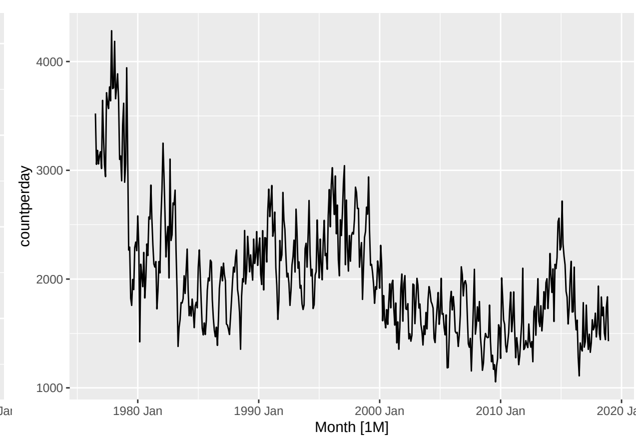


We have done population adjustments and removed the population effect from the increase of GDP.

2.2)

```
par(mar = c(4, 4, .1, .1))

aus_livestock %>%
  filter(State == "Victoria" & Animal == "Bulls, bullocks and steers") %>%
  autoplot(Count)

aus_livestock %>%
  filter(State == "Victoria" & Animal == "Bulls, bullocks and steers") %>%
  mutate(daysofmonth = days_in_month(as_date(Month)))  %>%
  mutate(countperday = Count/daysofmonth)  %>%
  autoplot(countperday)
```



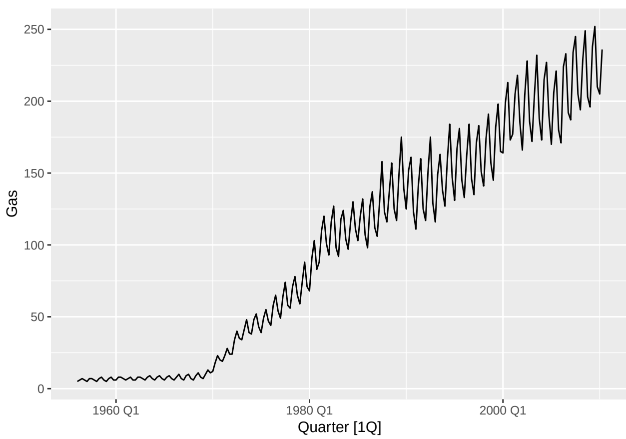


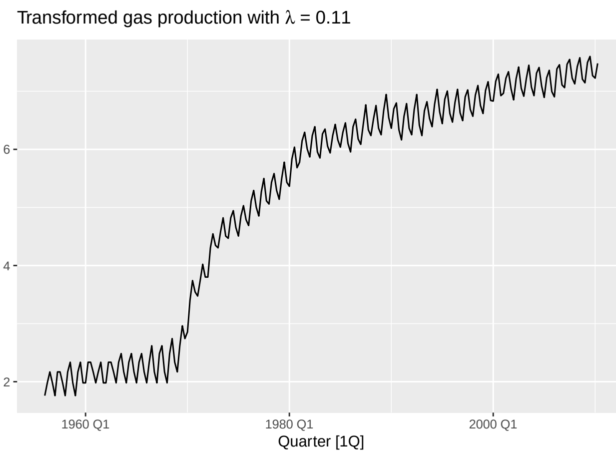
We have done calender adjustments by calculating the counts per day in each month rather than the total counts in the month, which remove the calendar variation.

2.3)

```
par(mar = c(4, 4, .1, .1))

aus_production %>%
```

```
  autoplot(Gas)

lambda <- aus_production %>%
  features(Gas, features = guerrero) %>%
  pull(lambda_guerrero)

aus_production %>%
  autoplot(box_cox(Gas, lambda)) +
  labs(y = "",
       title = latex2exp::TeX(paste0(
         "Transformed gas production with $\\lambda$ = ",
         round(lambda,2))))
```





We applied the box cox transformation to make a constant variation with the level of the series.

Exercise 3

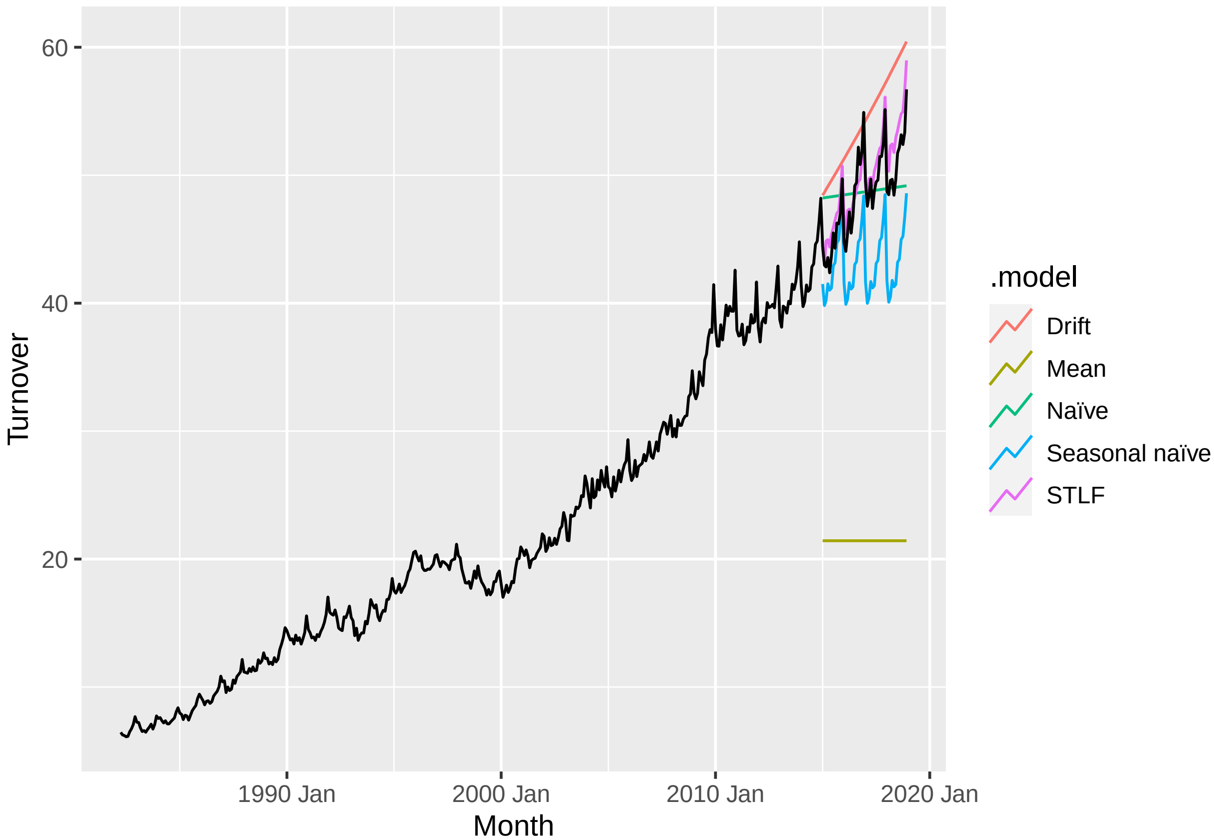
3.1) We took the aus_retail data until 2014 and leave the last 4 years as test data. A calendar transformation has also been done to remove calendar variation effect. We have calculated the lambda of boxcox transformation as we also observed increasing variation along with the timeline.

```
aus_train = aus_retail %>%
  filter(Industry == "Takeaway food services") %>%
  filter_index(.~"2014-12") %>%
  mutate(daysofmonth = days_in_month(as_date(Month)))  %>%
  mutate(Turnover = Turnover/daysofmonth)  %>%
  summarise(Turnover = sum(Turnover))

aus_all = aus_retail %>%
  filter(Industry == "Takeaway food services") %>%
  mutate(daysofmonth = days_in_month(as_date(Month)))  %>%
  mutate(Turnover = Turnover/daysofmonth)  %>%
  summarise(Turnover = sum(Turnover))
```

```
lambda <- aus_train %>%
  features(Turnover, features = guerrero) %>%
  pull(lambda_guerrero)
```

3.2) Mean, Naive, Seasonal Naive, Drift and STLF(decomposition model) are benchmarked here. Note we have applied box-cox transformation when we train the model.

```
aus_fit <- aus_train %>%
  model(
    Mean = MEAN(box_cox(Turnover,lambda)),
    `Naïve` = NAIVE(box_cox(Turnover,lambda)),
    `Seasonal naïve` = SNAIVE(box_cox(Turnover,lambda)),
    Drift = RW(box_cox(Turnover,lambda) ~ drift()),
    STLF = decomposition_model(
      STL(box_cox(Turnover,lambda) ~ trend(window = 15) + season(window = 5), robust = TRUE),
      RW(season_adjust ~ drift())
    )
  )

aus_fc <- aus_fit %>%
  forecast(h = 48)

aus_fc %>%
  autoplot(aus_all,level = NULL)
```



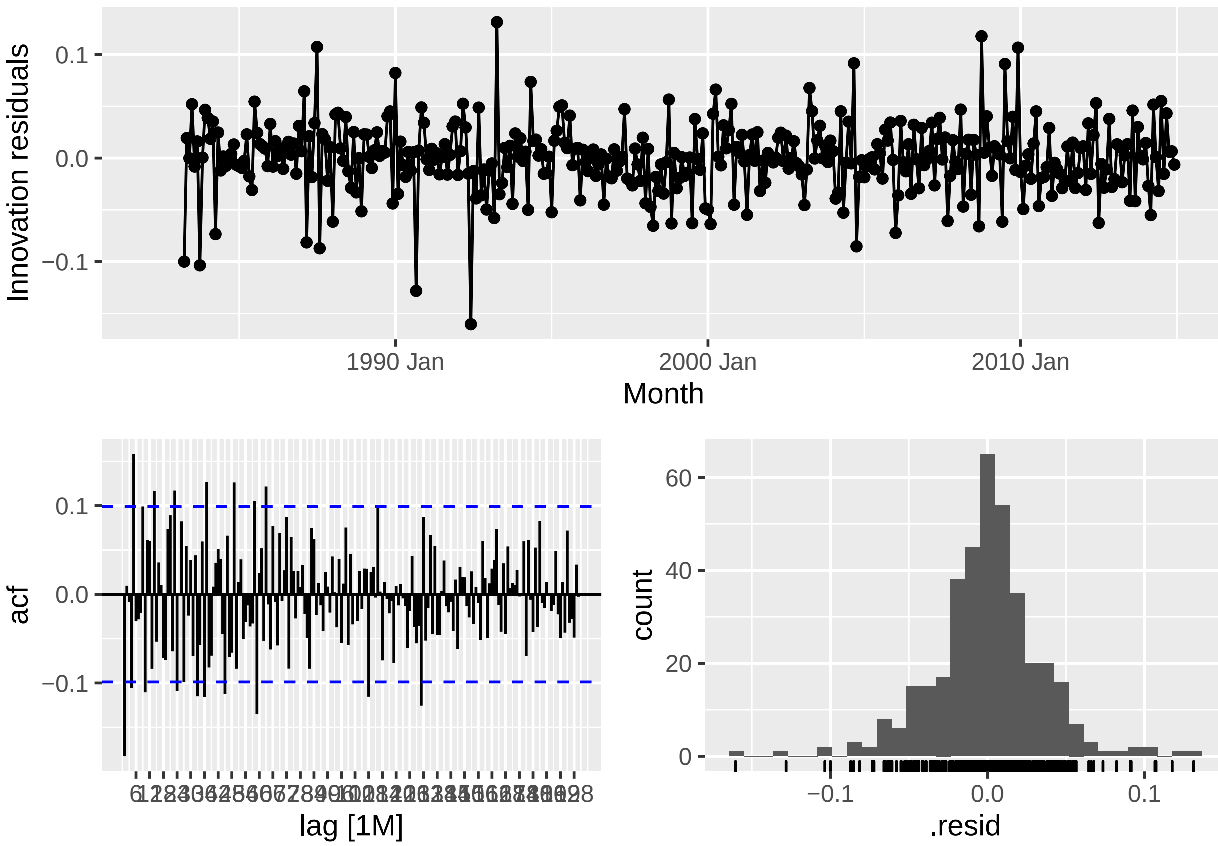
3.3)

```
accuracy(aus_fc, aus_all)
```

```
## # A tibble: 4 x 10
##   .model   .type     ME  RMSE   MAE    MPE  MAPE  MASE RMSSE
##   <chr>    <chr>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 Drift    Test  -2.14   3.25  2.77 -4.75   5.90  2.01  1.88
## 2 Mean     Test  27.2   27.5  27.2  55.7   55.7  19.8  15.9
## 3 Naïve    Test   0.469  3.49  2.88  0.461  5.92  2.10  2.02
## 4 Seasona~ Test   5.78   6.30  5.78 11.7   11.7   4.21  3.65
## # ... with 1 more variable: ACF1 <dbl>
```

As we are comparing forecast methods applied to a single time series, we consider using MAE and RMSE. Clearly, STLF(decomposition model) turns out to be the best forecasting method.

```
aus_train %>%
  model(STLF = decomposition_model(
      STL(box_cox(Turnover,lambda) ~ trend(window = 15) + season(window = 5), robust = TRUE),
      RW(season_adjust ~ drift())
    )) %>%
  gg_tsresiduals(type = "innovation",lag_max=200)
```



By producing residual diagnostic graphs of STLF models, we see that

1) innovation residuals shows a relatively constant variance along the time axis.
2) The innovation residuals have zero mean indicating that it is not biased.
3) The ACF graph also show a white noise pattern indicating that the innovation residuals are uncorrelated.

4) The innovation residuals are normally distributed.