Assignment 1

Christopher Bovolos (st.number 13979582)

15 November 2021

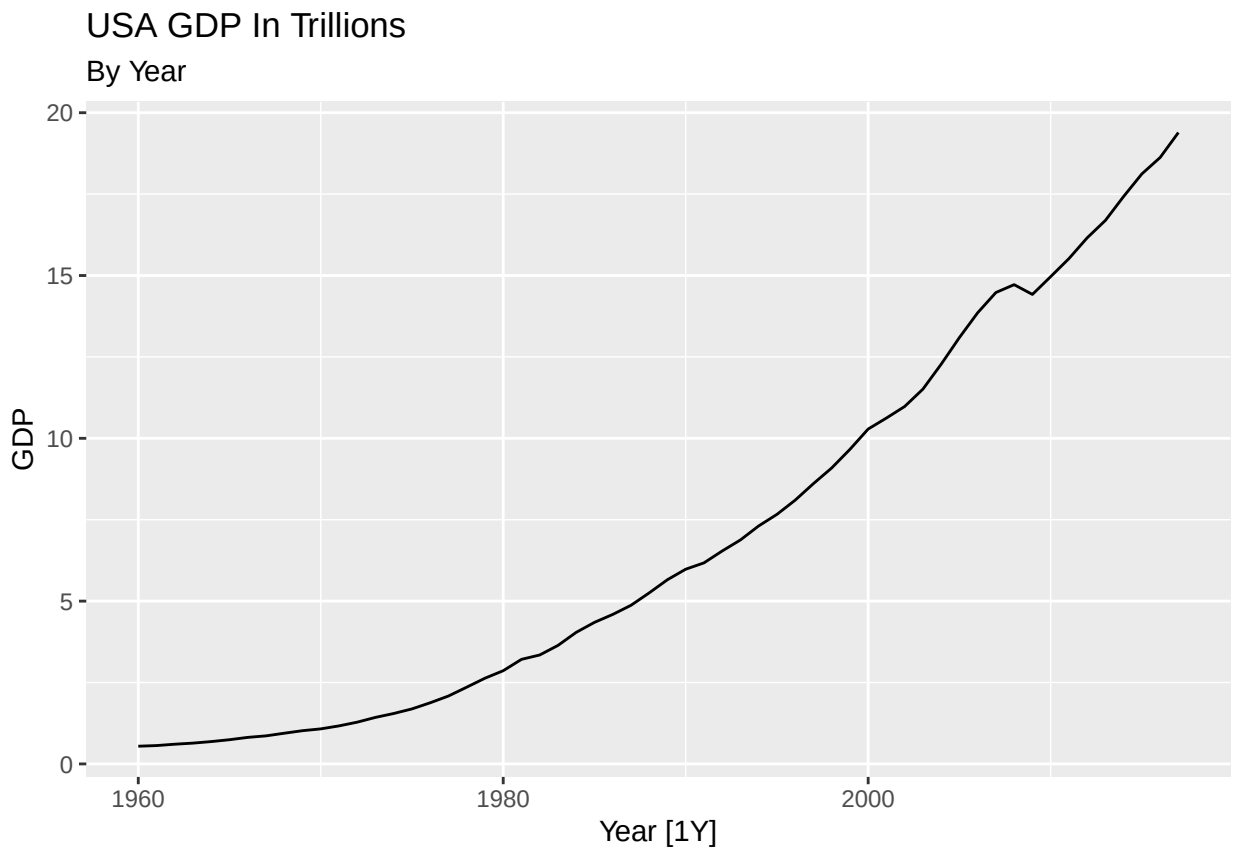
Exercises

Exercise 1

Below we can see the plot of US GDP in Trillions. As we can see there are not any obvious indicators for seasonality and/or cyclicity, since it resembles a line that steadily increases. The trend is an increasing one.

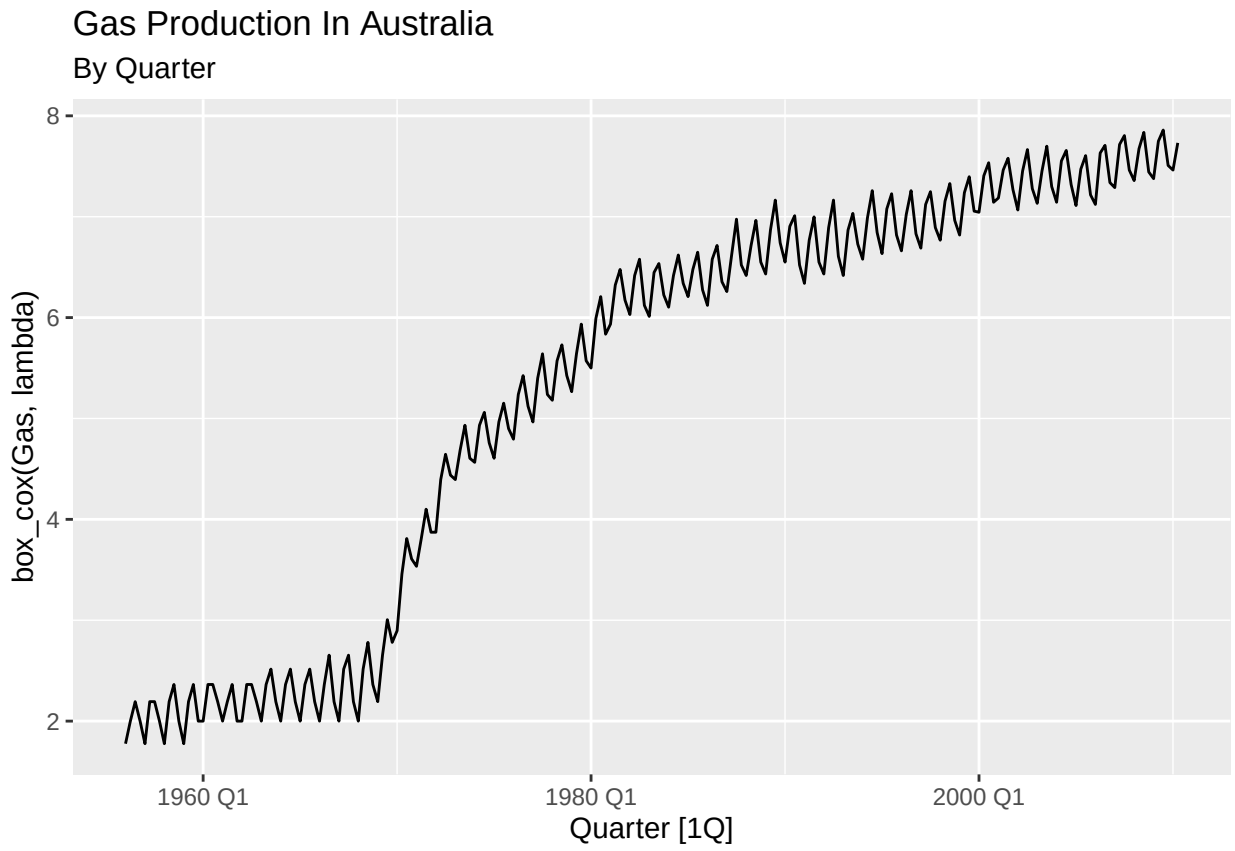
```
USD_economy = global_economy %>% filter(Code == "USA") %>% mutate(GDP = GDP/1000000000000)
autoplot(USD_economy) + labs(title = "USA GDP In Trillions", subtitle = "By Year")
```

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



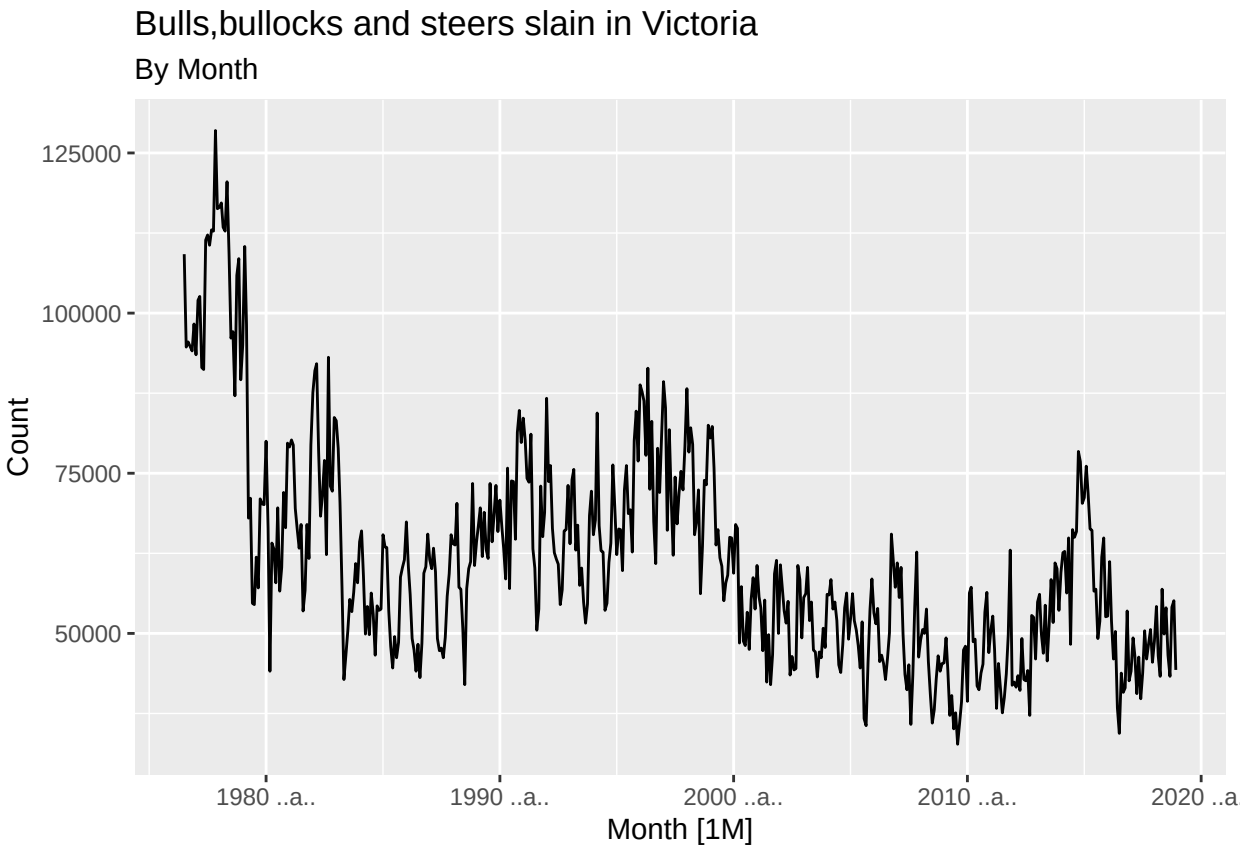
Below we can see the gas production in Australia by each quarter. We have applied a Box-Cox transformation because the data displayed increasing variation, by using the Guerrero feature which selects an appropriate Lambda for us in order to make our dataset express less variance. There is apparent seasonality in this plot and we could discern 2 cycles, the first one ending at approximately 1968. The trend does not change direction until 1970 where it becomes an increasing one.

```
lambda <- aus_production %>%
  features(Gas, features = guerrero) %>%
  pull(lambda_guerrero)
aus_production %>% autoplot(box_cox(Gas,lambda)) +
  labs(title = "Gas Production In Australia", subtitle = "By Quarter")
```



The plot for bulls,bullocks and steers slain expresses obvious seasonality probable correlated to consumer holidays and we could argue for the existanse of 4 cycles, first one ending in 1978, second one ending in 2000, third one ending in 2014 and the last one ending with the last measurement of our dataset. Apart from the first cycle where the trend is downwards, the trend is pretty flat without any noticeable increases or decreases.

```
livestockCountVictoria = aus_livestock %>% filter(State == "Victoria",
                                                  Animal =="Bulls, bullocks and steers")
autoplot(livestockCountVictoria, Count) +
  labs(title="Bulls,bullocks and steers slain in Victoria", subtitle = "By Month")
```



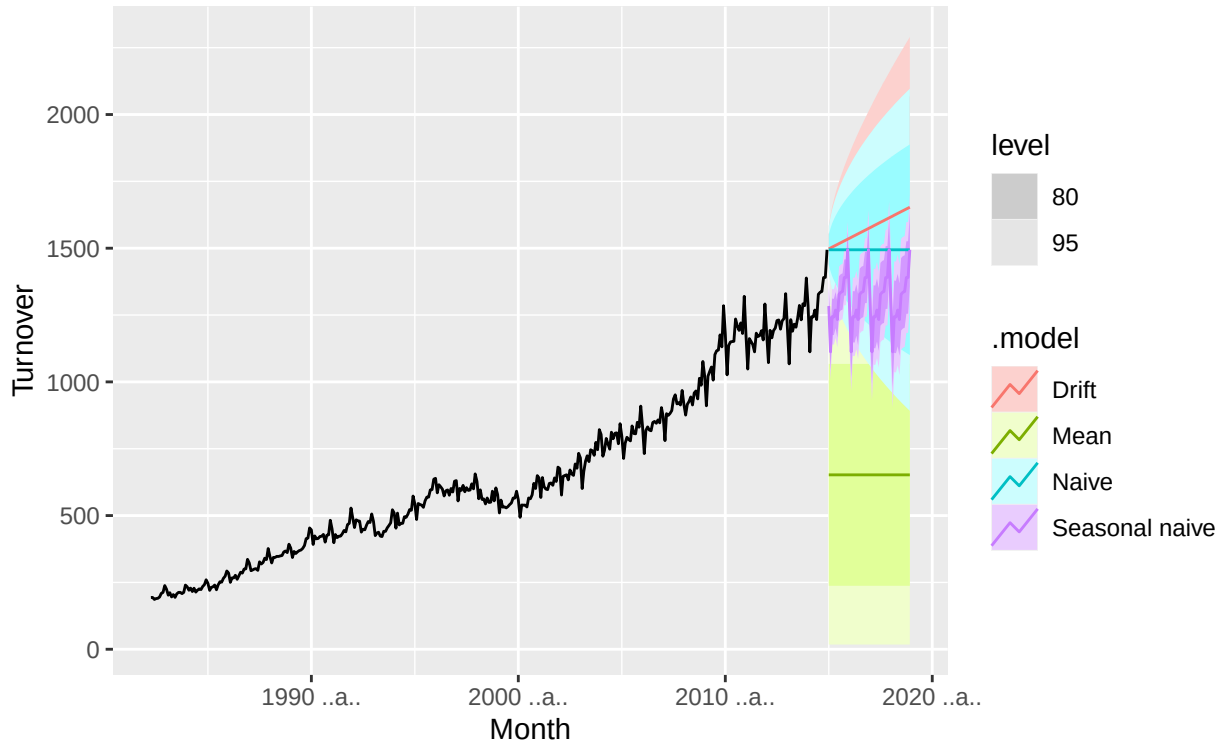
Exercise 2

We summarised the Turnover for all of the states and filtered for Takeaway food services

```
filteredAusRetail =  aus_retail %>% filter(
  Industry == "Takeaway food services")  %>%
  summarise(Turnover = sum(Turnover))



train = filteredAusRetail %>% slice(0:47- n())
test = filteredAusRetail %>% slice(n()-47 : 0 )
modelFit = train %>%  model(
    Mean = MEAN(Turnover),
    `Naïve` = NAIVE(Turnover),
    `Seasonal naïve` = SNAIVE(Turnover),
    Drift = RW(Turnover ~ drift())
  )
turnoverForecast = modelFit %>% forecast(h=48)
turnoverForecast %>% autoplot(train)
```



Our dataset uses as time window months, so in order to get the last 4 years, we have sliced the last 48 entries, which correspond to 4 years, we opted not to transform our dataset here because the variance does not appear to increase or decrease with time. We then fitted our model using training data and applied 4 benchmark methods, namely Naive, Seasonal Naive, Mean and Drift to create some basic and simple forecasts, which we can see plotted in the graph above, along with their prediction intervals(80%,95%).

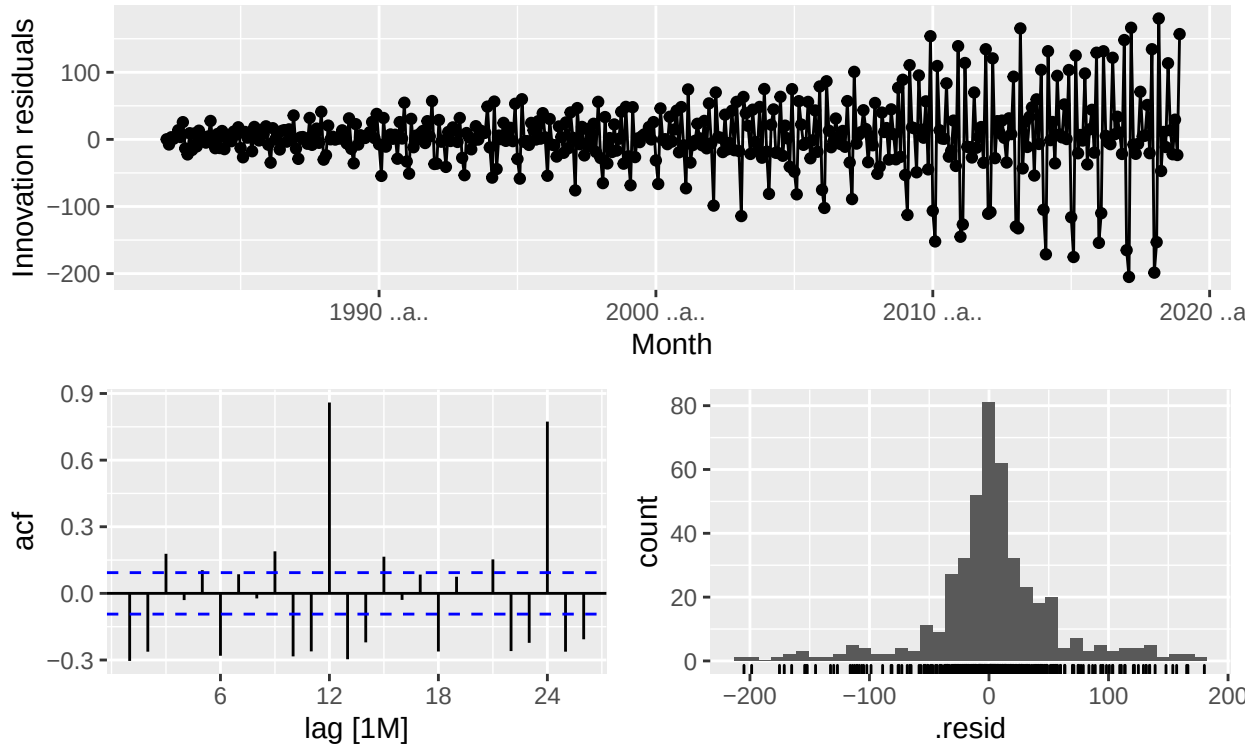
```
accuracy(turnoverForecast,filteredAusRetail)
```

```
## # A tibble: 4 x 10
##   .model         .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>          <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Drift          Test  -93.7  130. 108.  -6.82  7.67  2.58  2.46 0.403
## 2 Mean           Test  829.   838. 829.  55.7  55.7  19.8  15.8  0.613
## 3 Naive          Test  -12.4  119.  96.4 -1.49  6.66  2.30  2.25 0.613
## 4 Seasonal naive Test  177.   192. 177.  11.7  11.7   4.22  3.64 0.902
```

First we will take a look out our scale dependant errors like RMSE and MAE, for these errors Naive perform the best since it produces the smallest error in our scale (Turnover scale in this scenario). For percentage errors (MAPE) and scaled errors (MASE, RMSSE) the same conclusion is to be made, since again Drift produces the smallest differences. Hence we can conclude that Drift is the benchmark methods that forecast the best for our model. We will perform some residual diagnostics to assess how well our forecasts have captured the data they are supposed to forecast. The innovation residuals do not have homoscedasticity since towards the tail, the part we forecasted in fact, they appear to express increasing variance. Their autocorrelation apart from seemingly showing some small correlation, also have values that are well above our confidence

intervals. Their distribution also could not be characterized as normal since the left and the right tail are way too long. From the above we can reject that our residuals are a realization of a white noise process, which means that our forecasting method still has more improvements.

```
filteredAusRetail %>%
  model(`Naïve` = NAIVE(Turnover)) %>%
  gg_tsresiduals()
```

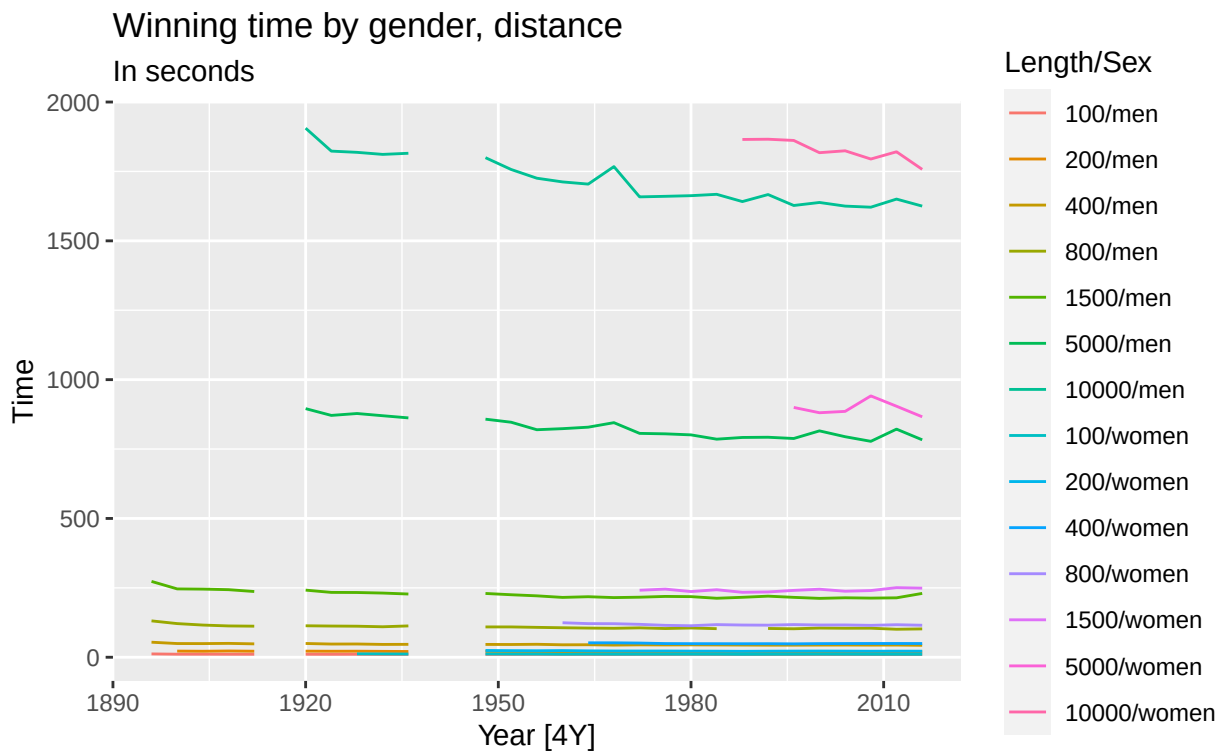


Exercise 3

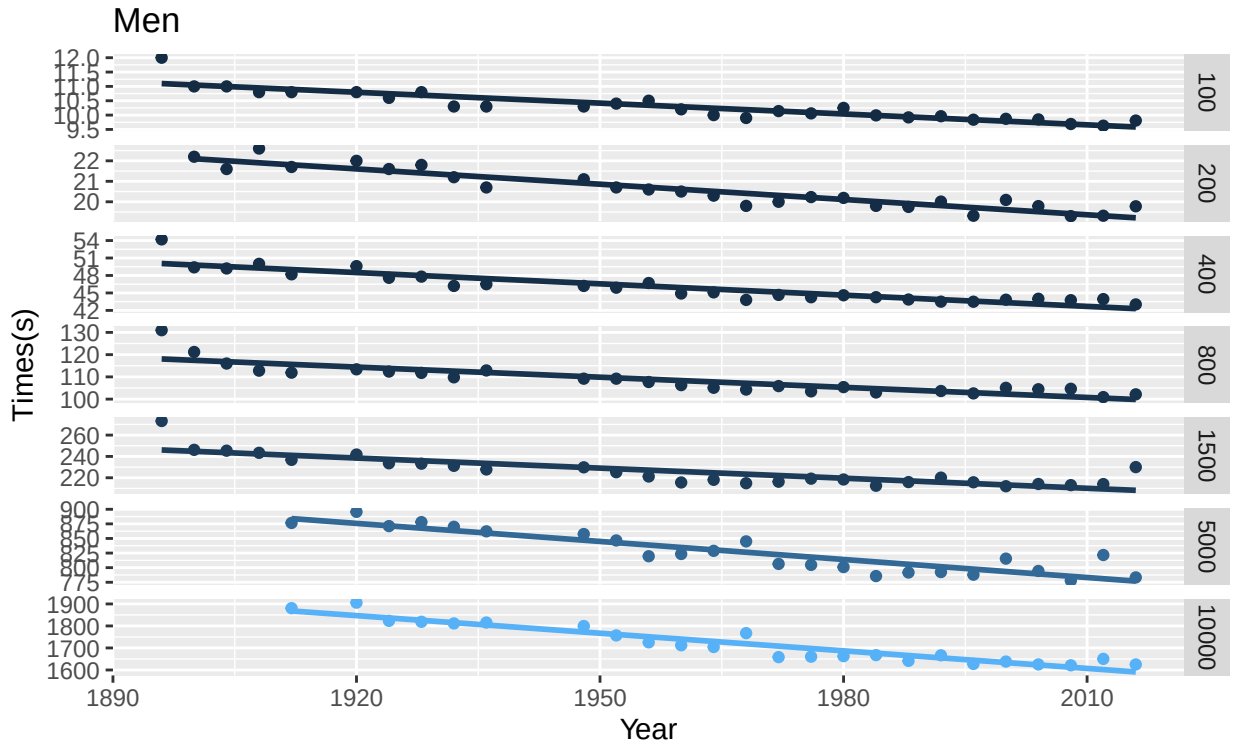
Below we can see the winning times plotted against each year measured in seconds for each course, categorized by gender and measured in seconds. As we can see, the winning time has a slight declining trend although there appears to be some gaps in our datasets, possibly due to the world wars in these eras.

```
autoplot(olympic_running)+ labs(title = "Winning time by gender, distance",
                                subtitle = "In seconds")
```

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

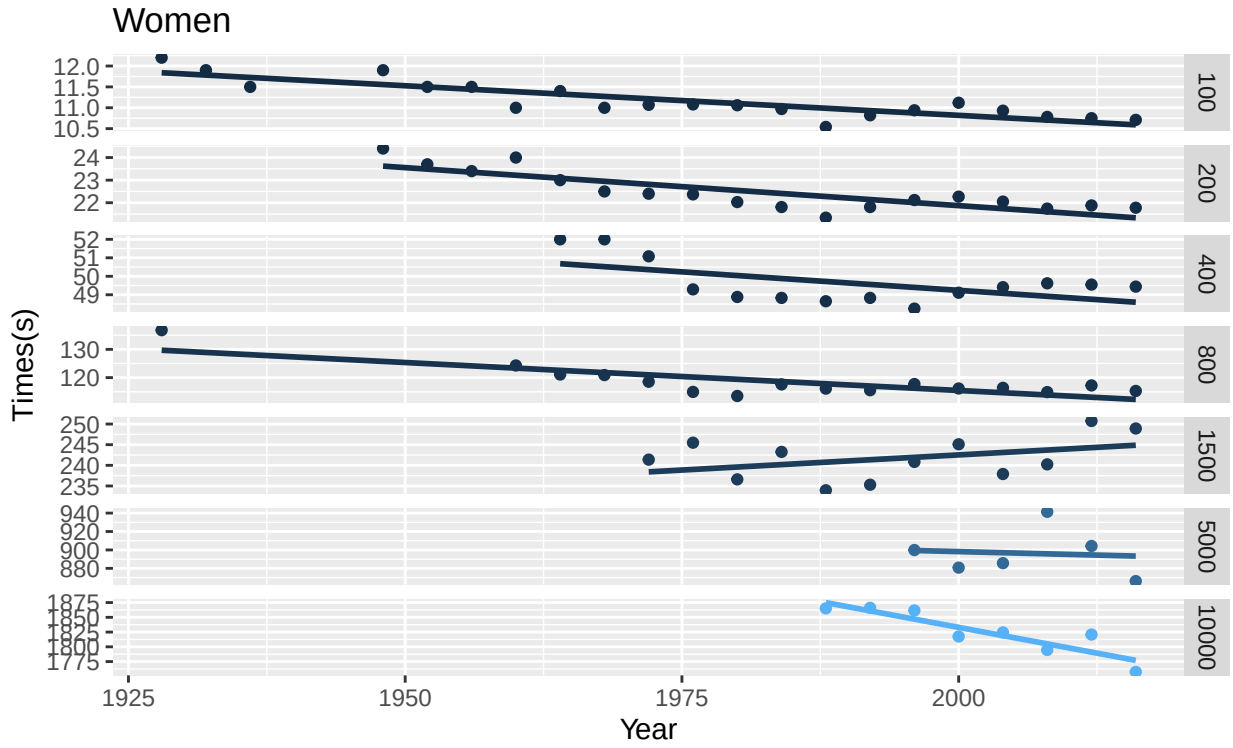


We will try and fit a regression line to each event, below we can see the fitted lines for men with regards to each distance



While here we can see the fitted lines for women for each distance. In the 1500m distance the trend seems to be increasing, but that could be due to the fact that we do not have enough points

for distances 1500, 5000, 10000.



In order to get the rate of change for each year, all we have to do is find the slope of our fitted lines. An indication as to how we can find this , is shown below, due to repetitiveness the rest of the code can be found in the appendix.

```
olympic_running %>% filter(Sex == "men", Length == 100) %>%
  model(TSLM(Time ~ Year)) %>%
  coefficients() %>% filter(term == "Year") %>% select(estimate)
```

For 100m : Times for men are decreasing at a rate of 0.012 and 0.014 for women each year

For 200m : Times for men are decreasing at a rate of 0.024 and 0.033 for women each year

For 400m : Times for men are decreasing at a rate of 0.064 and 0.04 for women each year

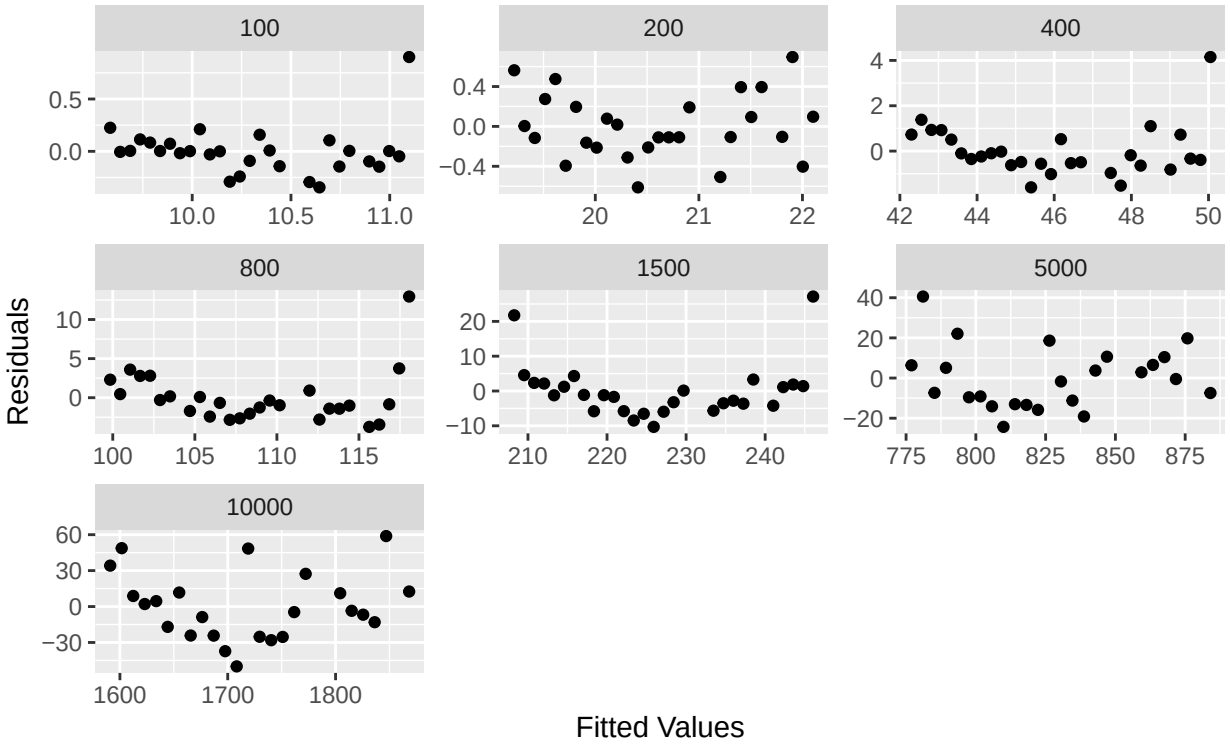
For 800m : Times for men are decreasing at a rate of 0.151 and 0.197 for women each year

For 1500m : Times for men are decreasing at a rate of 0.315 and increasing at a rate of 0.146 for women each year

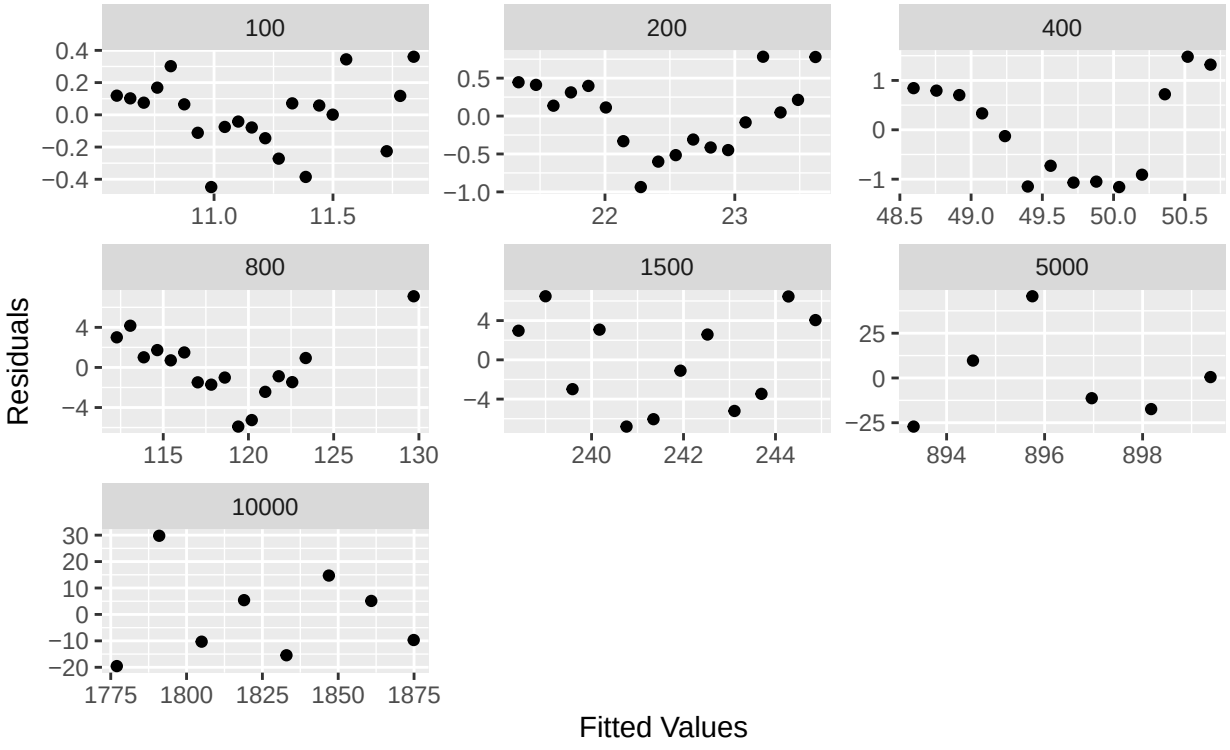
For 5000m : Times for men are decreasing at a rate of 1.029 and 0.303 for women each year

For 10000m : Times for men are decreasing at a rate of 2.665 and 3.49 for women each year

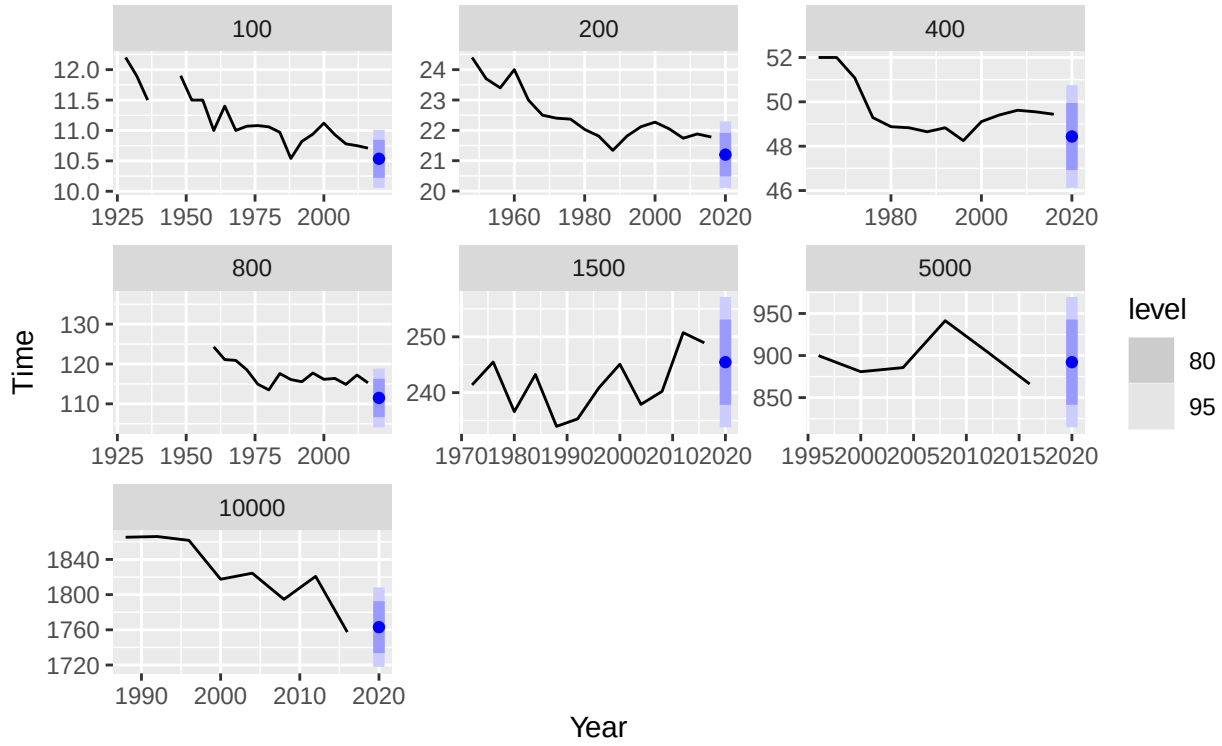
Below we can see that the residuals for men appear random and do not have big values, except for the distances 1500,5000 and 10000 where the residuals have big values. This could be due to the fact that we do not have enough data points for a more accurate fit.



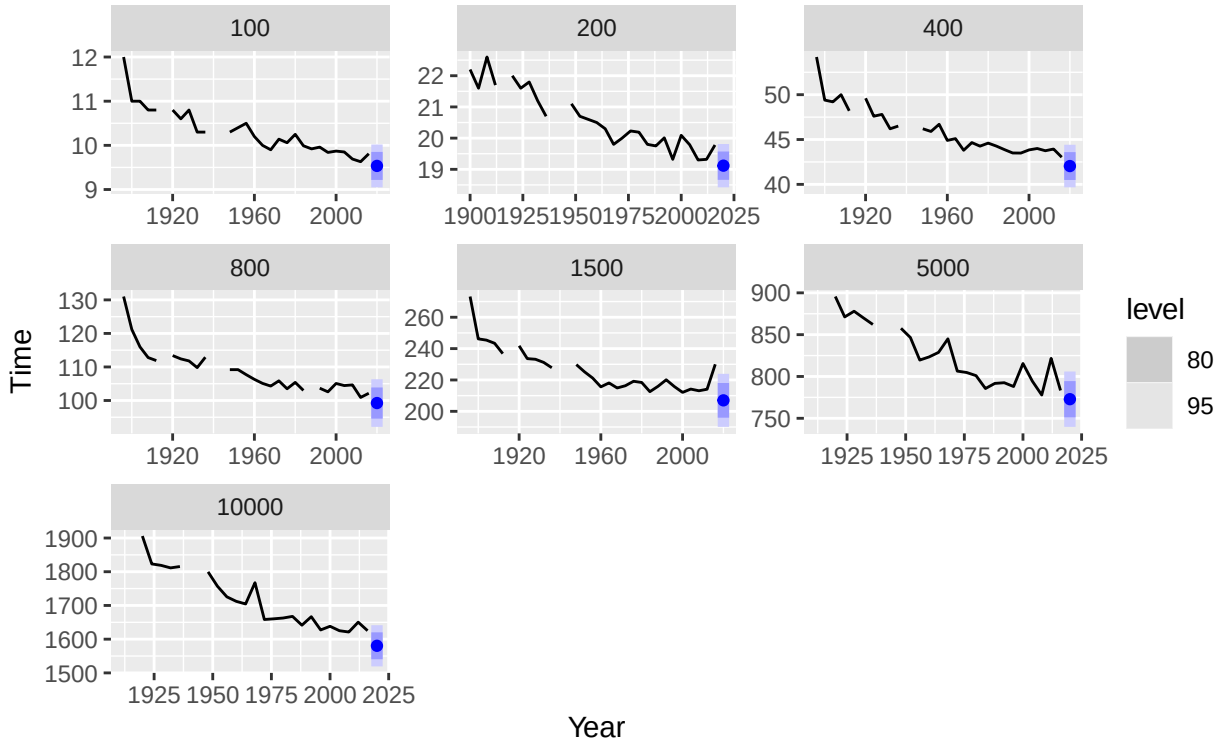
While for women our residuals express the same behaviour as above, where 1500, 5000 and 10000 meters have residuals with very big values. There are some residuals that we could deem problematic in the 800m distance, but since most of them are around the [-1,1] range we could say that this is an okay model, but not perfect.



Below we can see the prediction intervals for 95% confidence interval for the next olympic games in 2020. Which means that we are 95% confident that the next best time is going to be between [10.1,11] for 100m, [20.1,22.4] for 200m, [46.1,50.9] for 400m, [105,118] for 800m, [234,257] for 1500m, [824,974] for 5km and [1720,1810] for 10km. These interval bounds are given as approximation due to the fact that we can not discern with great detail the intervals, if we plotted each one on its own we could have a better idea.



As for men the next best time is going to be between [9.1,10] for 100m, [18.5,19.8] for 200m, [40,44] for 400m, [93,106] for 800m, [200,225] for 1500m, [745,809] for 5km and [1525,1650] for 10km. These interval bounds are given as approximation due to the fact that we can not discern with great detail the intervals, if we plotted each one on its own we could have a better idea.



We have assumed that the relationship between Time and Year to be linear, observations are independent of each other. We have also assumed, even though it is incorrect, that the homoscedasticity of the residuals is the same for every value of X and Normality. If we were given the option as to which models to forecast, we would not have selected the models whose residuals expressed heteroscedasticity

Appendix

The model reports for Exercise 3.2

```
olympic_running %>% filter(Sex == "women", Length == 100) %>%
  model(TSLM(Time ~ Year)) %>%
  coefficients() %>% filter(term == "Year") %>% select(estimate)

olympic_running %>% filter(Sex == "women", Length == 200) %>%
  model(TSLM(Time ~ Year)) %>%
  coefficients() %>% filter(term == "Year") %>% select(estimate)

olympic_running %>% filter(Sex == "women", Length == 400) %>%
  model(TSLM(Time ~ Year)) %>%
  coefficients() %>% filter(term == "Year") %>% select(estimate)

olympic_running %>% filter(Sex == "women", Length == 800) %>%
  model(TSLM(Time ~ Year)) %>%
  coefficients() %>% filter(term == "Year") %>% select(estimate)
```

```
olympic_running %>% filter(Sex == "women", Length == 1500) %>%
  model(TSLM(Time ~ Year)) %>%
  coefficients() %>% filter(term == "Year") %>% select(estimate)

olympic_running %>% filter(Sex == "women", Length == 5000) %>%
  model(TSLM(Time ~ Year)) %>%
  coefficients() %>% filter(term == "Year") %>% select(estimate)

olympic_running %>% filter(Sex == "women", Length == 10000) %>%
  model(TSLM(Time ~ Year)) %>%
  coefficients() %>% filter(term == "Year") %>% select(estimate)

olympic_running %>% filter(Sex == "men", Length == 200) %>%
  model(TSLM(Time ~ Year)) %>%
  coefficients() %>% filter(term == "Year") %>% select(estimate)

olympic_running %>% filter(Sex == "men", Length == 400) %>%
  model(TSLM(Time ~ Year)) %>%
  coefficients() %>% filter(term == "Year") %>% select(estimate)

olympic_running %>% filter(Sex == "men", Length == 800) %>%
  model(TSLM(Time ~ Year)) %>%
  coefficients() %>% filter(term == "Year") %>% select(estimate)

olympic_running %>% filter(Sex == "men", Length == 1500) %>%
  model(TSLM(Time ~ Year)) %>%
  coefficients() %>% filter(term == "Year") %>% select(estimate)

olympic_running %>% filter(Sex == "men", Length == 5000) %>%
  model(TSLM(Time ~ Year)) %>%
  coefficients() %>% filter(term == "Year") %>% select(estimate)

olympic_running %>% filter(Sex == "men", Length == 10000) %>%
  model(TSLM(Time ~ Year)) %>%
  coefficients() %>% filter(term == "Year") %>% select(estimate)
```