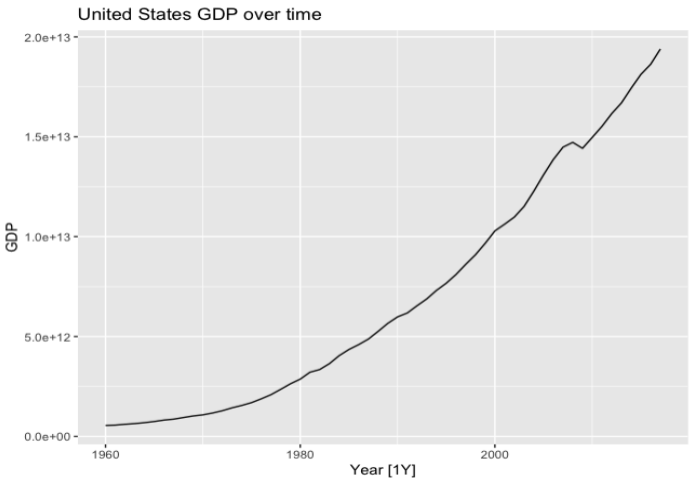
Assignment 1

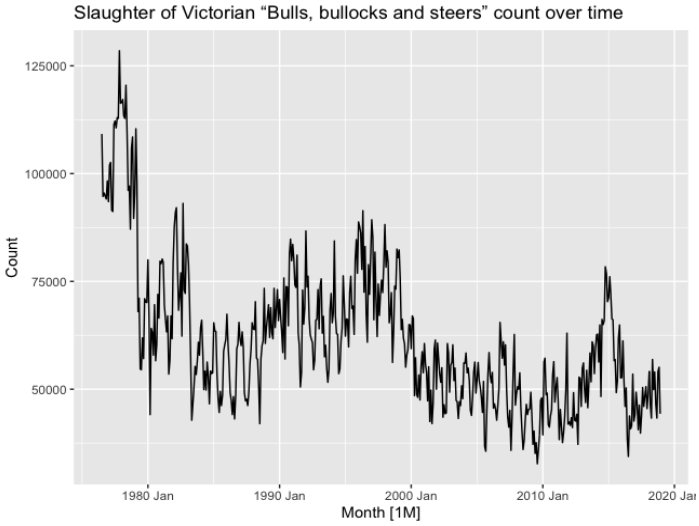
Dragos Pop

Exercise 1

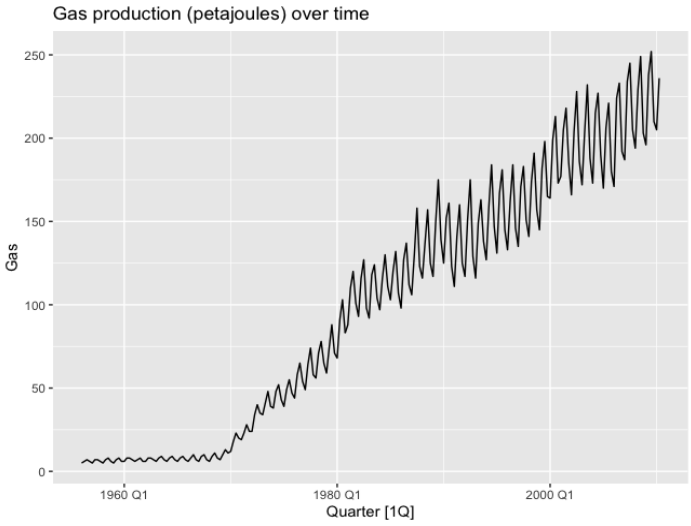
1.1) After filtering the country from the *global_economy* tsibble, the United States GDP over time was plotted using the function *autoplot()* with *GDP* as argument. As one can see from the image, the GDP follows an upward trend from the ’60 until 2017, with a small dip around 2008 which is most likely due to the financial crisis. Accordingly, the GDP evolved from 0,5 quadrillions (10^{15}) to 19,4 quadrillions. As the graph shows a clear trend with no variation, no transformation is needed.



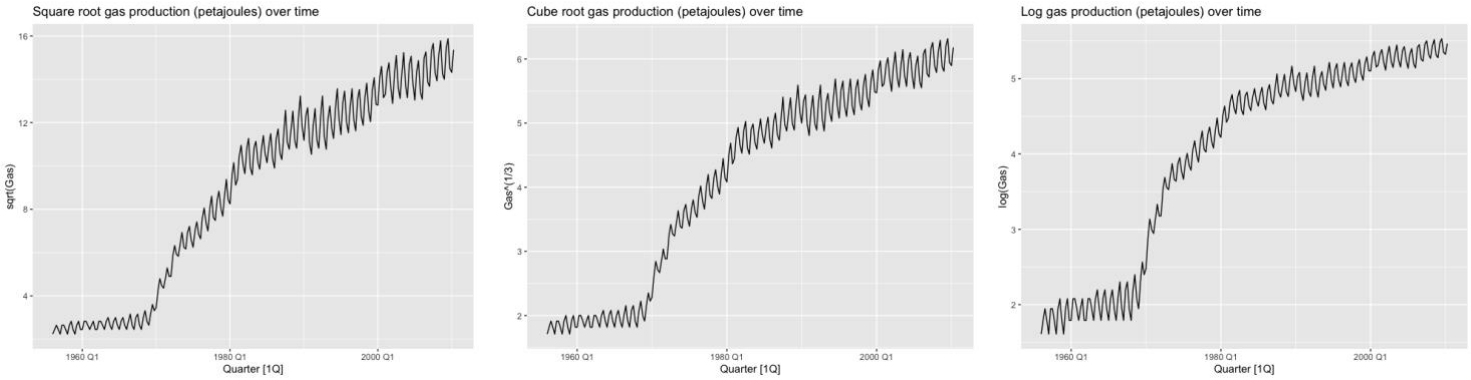
1.2) Similarly, the count of Victorian bulls, bullocks, and steers slaughtered was selected filtering the Victoria state and the respective animals from *aus_livestock* tsibble and then plotted with the help of the function *autoplot()* with parameter *Count*. Since the data does not appear to vary proportionally to the level of the series, no transformation was applied. Hence, the plot shows a drastic decrease of animals slaughtered after 1978, from a monthly average of one million to about 0,7 millions. Afterwards, the trend looks slightly downwards with two periods of growth in the 90’ and around 2005.



1.3) The gas production was simply represented using *autoplot()* on the *Gas* variable of *aus_production* dataset. As it is visible from the image on the right, the line plot shows the variation increasing with the petajoules of Gas over time, thus, a transformation is useful in this case to make the variation consistent across the data. After trying the main mathematical transformations, namely



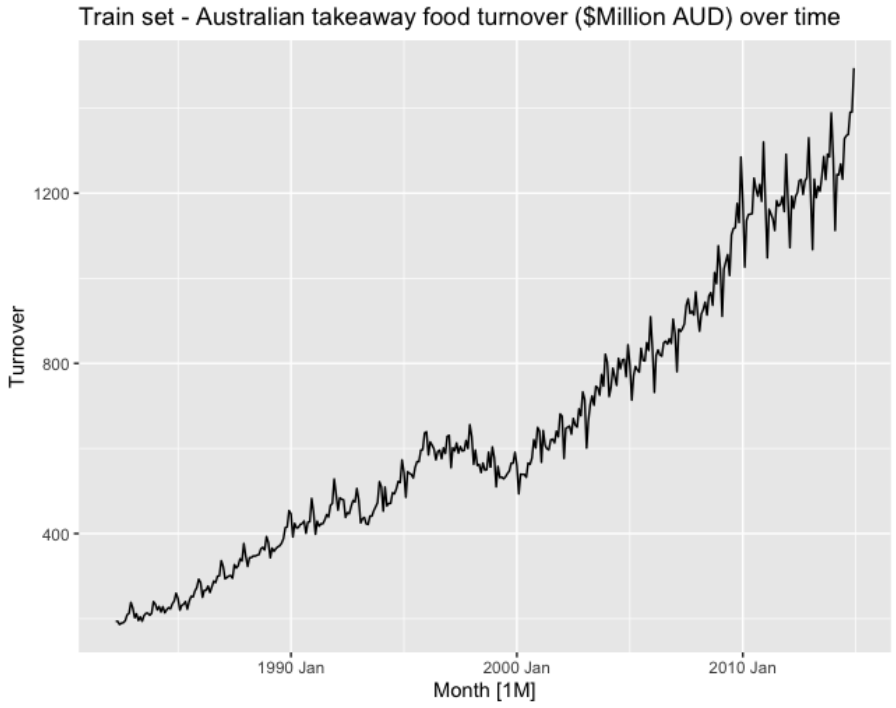
square root, cube root and natural logarithm, the last one was picked as it led to the most stabilized variation, as it can be seen below. Another advantage of the log transformation is that it is interpretable, meaning that the changes in the log value are proportional to changes in the original scale [1], making the forecasting model simpler.



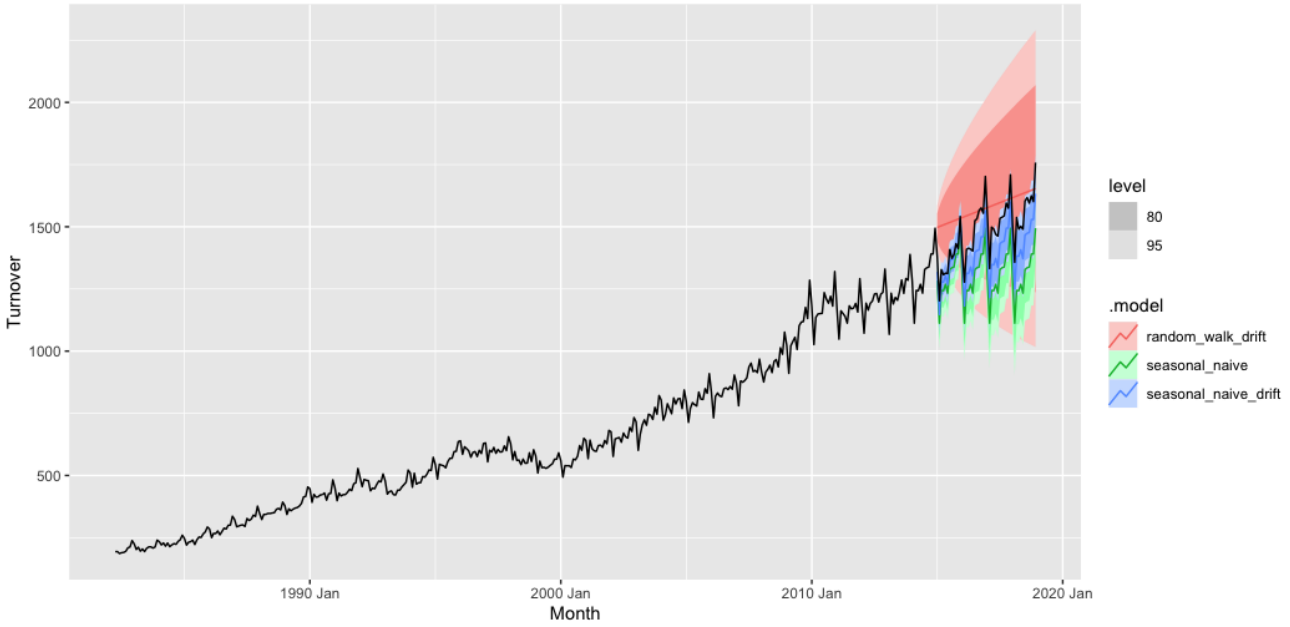
Accordingly, the gas production follows a clear upwards trend from 1950 to 2010, with a drastic increase around 1970. Besides, the series is seasonal, with Q2 and Q3 peaking over Q1 and Q4.

Exercise 2

2.1) In order to select the appropriate data, the *aus_retail* tsibble was filtered for the takeaway food services industry and the turnover was summed per month. Then, the training set was selected using *slice(1:(n() - 4*12))* to slice out the last 48 months (4 years) from the takeaway series, resulting in turnovers up to December 2014. Their trend can be observed in the image below, following an upward trend with a small dip before 2000, and a seasonal pattern, peaking at the end of the year and falling right at the bottom at the beginning of the year.



2.2) The appropriate benchmark methods in this case are seasonal naïve because the data has a strong seasonal pattern, random walk with drift because the series has a clear upwards trend, and seasonal naïve with drift because of the strong seasonality and trend. Accordingly, the three methods were fitted to the train set using the *model()* function and a forecast for the next 4 years was generated by each using the function *forecast(h = "4 years")*. Lastly, the forecasts were attached to the time series as it can be seen below.



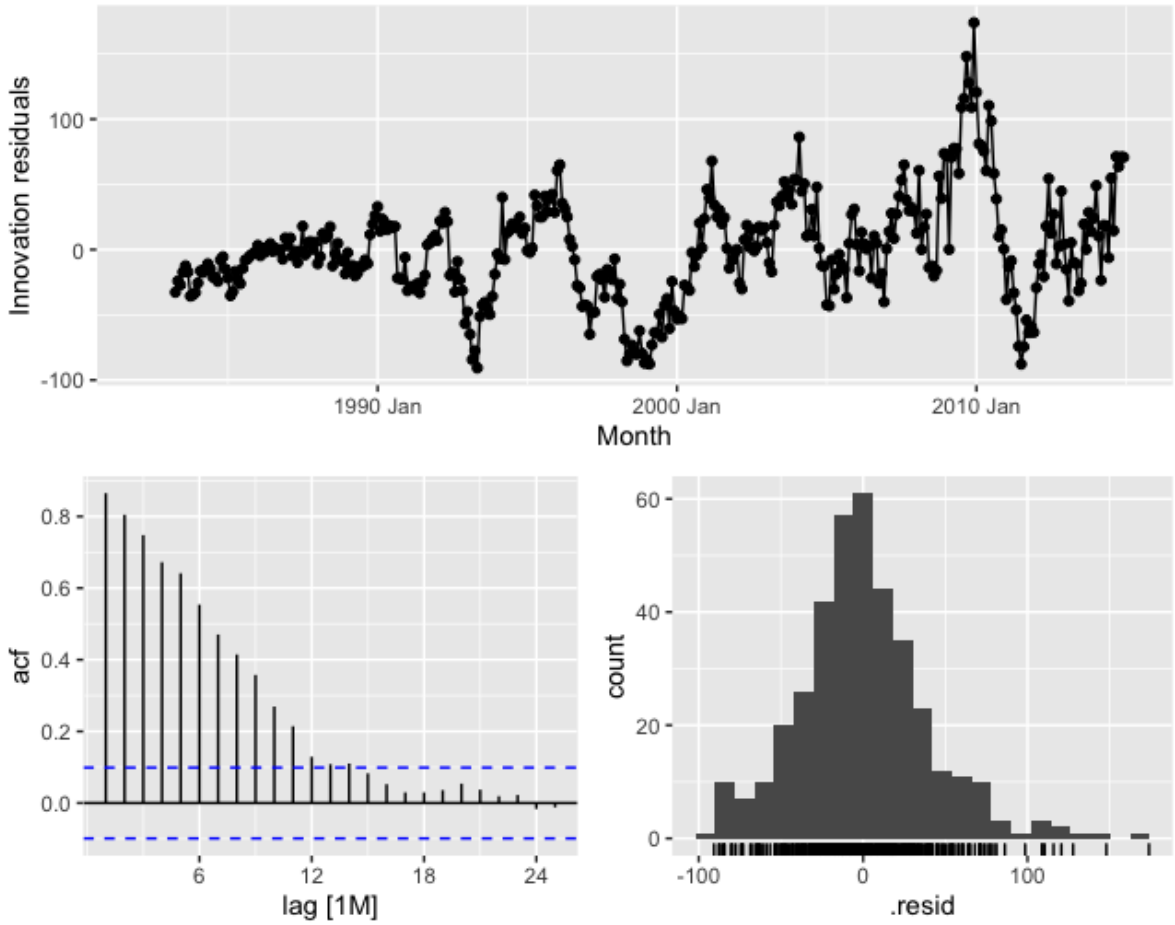
By looking at the graph, one can quickly observe the inaccuracy of the random walk with drift forecast that ranges in the right interval but does not follow the seasonality of the data. Additionally, it is visible that the blue line, namely the predictions of the seasonal naïve with drift are closer to the real values depicted with black in comparison with the predictions of the seasonal naïve model. Consequently, one would pick the seasonal naïve with drift as the best model.

2.3) The accuracy of the three models is simply done calling the function *accuracy()* on the forecasts and test set, whose results are in the table below.

	.model	.type	ME	RMSE	MAE	MPE	MAPE	MASE	RMSSE	ACF1
	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	random_walk_drift	Test	-93.7	130.	108.	-6.82	7.67	NaN	NaN	0.403
2	seasonal_naive	Test	177.	192.	177.	11.7	11.7	NaN	NaN	0.902
3	seasonal_naive_drift	Test	90.0	100.	90.5	5.98	6.02	NaN	NaN	0.799

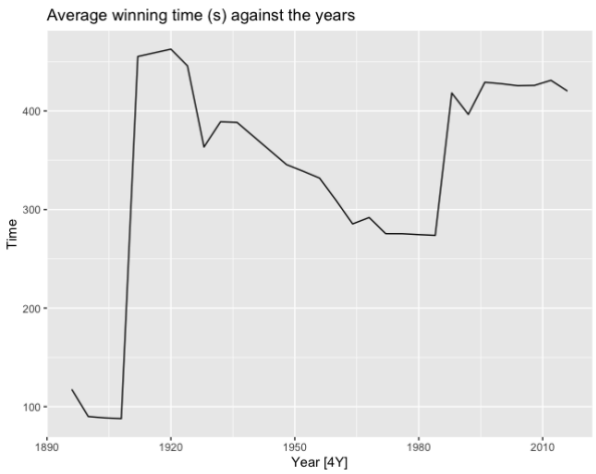
From the table, same ranking of the three models is the following: seasonal naïve with drift the best model, followed by the random walk with drift and then by the seasonal naïve. The reason for reaching this conclusion is that the (absolute) errors are the smallest in case of the seasonal naïve with drift and the largest of the seasonal naïve. Hence, the seasonal naïve with drift does the best forecasts.

Next, the residuals of the seasonal_naive_drift fit were represented with *gg_tsresiduals()*. Looking at the ACF, it shows the data contains indeed a trend since it has many significant lags. Consequently, even if the distributions of the residuals looks standard normal in the histogram, the residuals do not resemble white noise due to the many significant lags in the ACF. This means the residuals are correlated and there is information left in the residuals that could be used in the forecast.

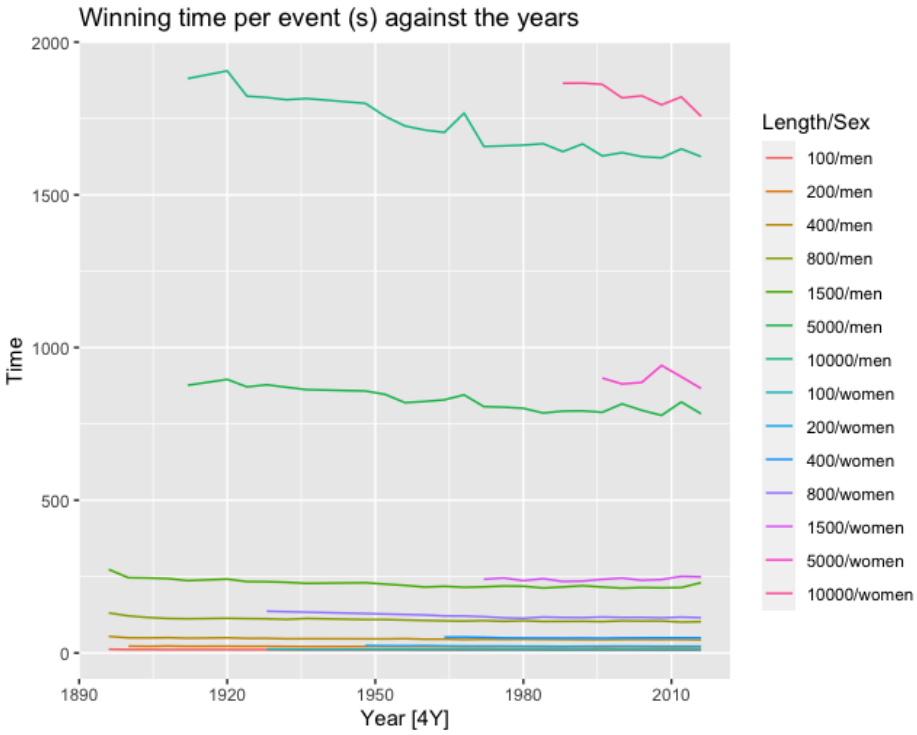


Exercise 3

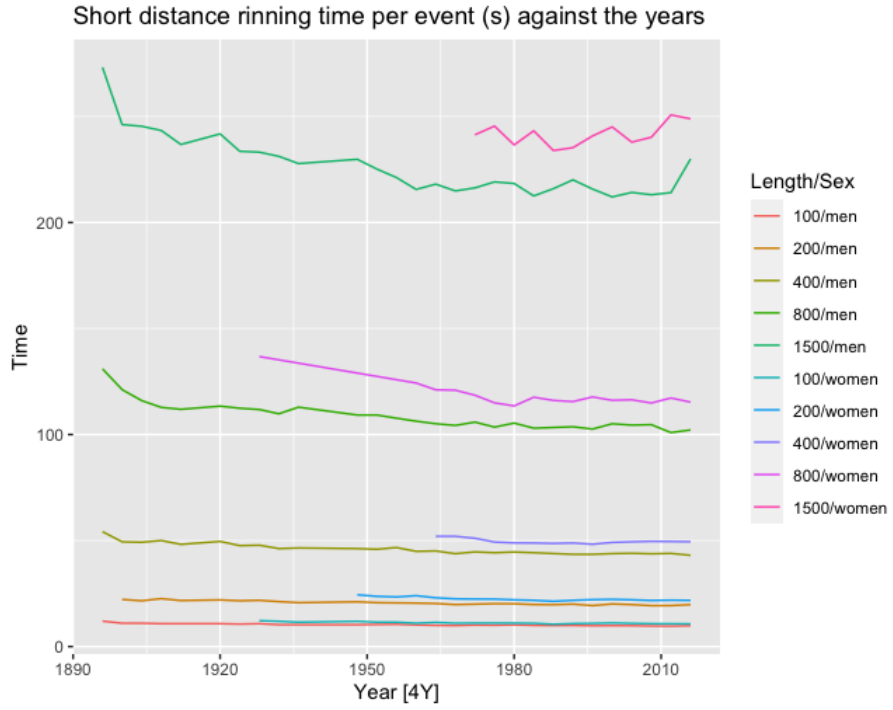
3.1) In the image on the right, one can see the average winning time in seconds over the years. The reason why this is looking quite strange is because the graph was generated averaging over all the events in the respective years, both men and women, and some events did not take place in certain years. For this reason, the winning time per event was generated. From this, one can distinguish the three types of events, long, medium, and short distance running. From the first two, represented by the upper four line plots, it is visible that men events exist from a longer time, men are generally significantly faster, and the



winning time is on a downward trend, reflecting the advance in technology (e.g. medicine, training, shoes, nutrition, etc).



Zooming in for the short-distance running events, the same conclusion can be drawn as above, excepting the women competing in the 1500m race, which are not getting faster over the year. Moreover, it is visible that as the distance decreases, the line gets flatter, suggesting smaller variance in the winning times.



3.2) First, the three types of events were created, sprint events (finished under 500s), middle (finished between 500s and 1000s) and long (over 1000s). Then the regression lines were fitted using a TSLM model.

3.3)

3.4)

References

[1] https://otexts.com/fpp3/transformations.html

Appendix

```
## Assignment 1
library(fpp3)

# 1.1 United States GDP
global_economy %>%
  filter(Country == "United States") %>%
  autoplot(GDP)+
  labs(title = "United States GDP over time")

# 1.2 Slaughter of Victorian “Bulls, bullocks and steers”
aus_livestock %>%
  filter(Animal == "Bulls, bullocks and steers") %>%
  filter(State == "Victoria") %>%
  autoplot(Count)+
  labs(title = "Slaughter of Victorian “Bulls, bullocks and steers” count over time")

# 1.3 Gas production
aus_production %>%
  autoplot((Gas))+
  labs(title = "Log gas production (petajoules) over time")

# 2.1
takeaway <- aus_retail %>%
  filter(Industry == "Takeaway food services") %>%
  summarise(Turnover = sum(Turnover))
takeaway

train_set <- takeaway %>%
  slice(1:(n() - 4*12))
train_set

test_set <- takeaway %>%
```

```
  slice((n() - 4*12) : n())
test_set

train_set %>%
  autoplot(Turnover)+labs(title = "Train set - Australian takeaway food turnover ($Million AUD) over
time")

# 2.2
fc <- bind_cols(
  train_set %>% model(seasonal_naive = SNAIVE(Turnover)),
  train_set %>% model(random_walk_drift = RW(Turnover ~ drift())),
  train_set %>% model(seasonal_naive_drift = SNAIVE(Turnover ~ drift()))) %>%
  forecast(h = "4 years")

fc %>%
  autoplot(takeaway)

# 2.3
fc %>% accuracy(test_set)

fit <- train_set %>%
  model(seasonal_naive_drift = SNAIVE(Turnover ~ drift()))

fit %>% gg_tsresiduals()

# 3.1
olympic_running %>%
  filter(!is.na(Time)) %>%
  summarise(Time = mean(Time)) %>%
  autoplot(Time) +labs(title = "Average winning time (s) against the years")

olympic_running %>%
  filter(!is.na(Time)) %>%
  autoplot(Time) +labs(title = "Winning time per event (s) against the years")

olympic_running %>%
  filter(!is.na(Time)) %>%
  filter(Time < 410) %>%
  autoplot() +labs(title = "Short distance rinning time per event (s) against the years")

# 3.2

sprint <- olympic_running %>%
  filter(!is.na(Time)) %>%
  filter(Time < 500)

middle <- olympic_running %>%
```

```
  filter(!is.na(Time)) %>%
  filter(Time < 1000) %>%
  filter(Time > 500)

long <- olympic_running %>%
  filter(!is.na(Time)) %>%
  filter(Time > 1000)

fit <- sprint %>%
  model(TSLM(Time ~ Length() + Sex() + Year()))
fit %>% report()
```