# **Beer Production in Australia**

**Seasonal Time Series Analysis**

**Dataset** – ‘monthly-beer-production-in-australia.csv’

**Dataset Dates** – From 1956-01 to 1995-08, in monthly intervals.

**Source of Data –** Kaggle

**Code –** Completed in RStudio

**Reading the Data**

First thing that I needed to do was to convert the dataset to a time series and plot the initial data.

Text

Description automatically generated

Graphical user interface

Description automatically generated with low confidence

I then had a look at the column names and values in the dataset.



Text

Description automatically generated

This informed me that there were two column names that I would be working with, ‘Month’ and ‘Monthly Beer Production’. The summary showed that there were a total number of 476 rows of data in the dataset, with the min amount of beer produced at 64.8 per month and the max at 217.8.

To view the first values in the dataset:





I then plotted a linear mean line on the current plot, and this was the output.



Graphical user interface

Description automatically generated with low confidence

As you can see it doesn’t represent the overall mean for each of the values.

I then plotted a graph which showed the mean aggregate for each year which gave a more representable series.



Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

Above you see the mean line on the aggregate mean chart, not giving the same data.

**Decomposition**

To decompose data, you must split it into different components. This allows you do extract seasonality from it, making it easier to forecast. It is split up into four components: Level, trend, seasonality and noise/random.



Chart

Description automatically generated

To get a better look at each of the time series, I plot them individually.

Trend:

Chart, line chart

Description automatically generated

Seasonal:

A picture containing application

Description automatically generated

Random:

Graphical user interface

Description automatically generated with medium confidence

**Seasonality**

From the original time series, you can see that it is seasonal with an increase in production over time. To make this clearer, I plot the time series using the first 5 years of values.



Chart, line chart

Description automatically generated

As you can clearly see from the time series plotted, there is a dip in production levels every 12 months, starting from the sixth month of each year. You can also tell that there is a high at every twelfth month of each year.

I demonstrated this seasonality of the data by plotting a boxplot.



Chart, box and whisker chart

Description automatically generated

From the chart you can clearly see that in the 6th month of June, the production of beer in Australia is significantly lower than in the 12th month of December. This is probably because the demand for beer in the colder months of June or July is a lot less than the months with warmer weather.

**Stationarity**

To test for stationarity, I will use the autocorrelation function. From the plot below you can tell that the data isn’t stationary. To further test for stationarity, we can conduct the ADF test.



Chart

Description automatically generated

The Augmented Dickey-Fuller Test provided the following information.

Text

Description automatically generated

With a p-value of 0.01, we can reject the null hypothesis of non-stationarity.

To use the arima model the data must be stationary. I will do this using the log and diff functions to plot the data with equal variances and an equal trend.



A picture containing graphical user interface

Description automatically generated



A picture containing bar chart

Description automatically generated

Now we can get the arima model. To get the ar value, we use an acf graph. To get the ma value we use a pacf graph.

Timeline

Description automatically generated

From the acf model, we can determine that the ma value is 0.

Chart

Description automatically generated with low confidence

The pacf model gives a ar value of 1. As we have differentiated the terms once, the d value will also be one. This gives us an arima of (1,1,0).

**Prediction**

To predict future values, using the arima model.



A picture containing graphical user interface

Description automatically generated

To then test these results, I used the predict function, to predict the result of the year 1994/95, which had already happened. This showed me that the prediction was actually very accurate, as the figures for these years were similar.

Text

Description automatically generated

Text

Description automatically generated