Monographic Lecture in Mathematics: Time Series, Lab № 1

Bartosz Banasik, 229760 09/03/2020

Task 1

Load `co2.csv` using Pandas library and set date as the data frame's index. File consists of four columns: date, mean global CO2 concentration, mean CO2 concentration in northern hemisphere, and mean CO2 concentration southern hemisphere.

Listing 1: "Solution"

```
import pandas as pd

df = pd.read_csv('co2.csv', delimiter=";", index_col=0)
df = df[df.index >= "1950"]
df.index = pd.to_datetime(df.index, format="%Y-%m-%d")
```

The Listing 1 shows how to read data from the `.csv` file (default with header) and filter only desired range. Then transform the index to `pandas.DatetimeIndex` type.

Task 2

Using line plot, plot all three mean annual CO2 concentrations in one figure as a function of time.

To plot annual CO2 concentrations we need to average the values over the year when they were recorded. This part of code will do it for us.

Listing 2: "Solution"

```
annual_mean = df.groupby(df.index.year).mean()
```

On the figure 1 there is presented a graph visualizing above data.

Task 3

Using `Seaborn` library, visualize the spread of monthly CO2 concentrations with boxplots.

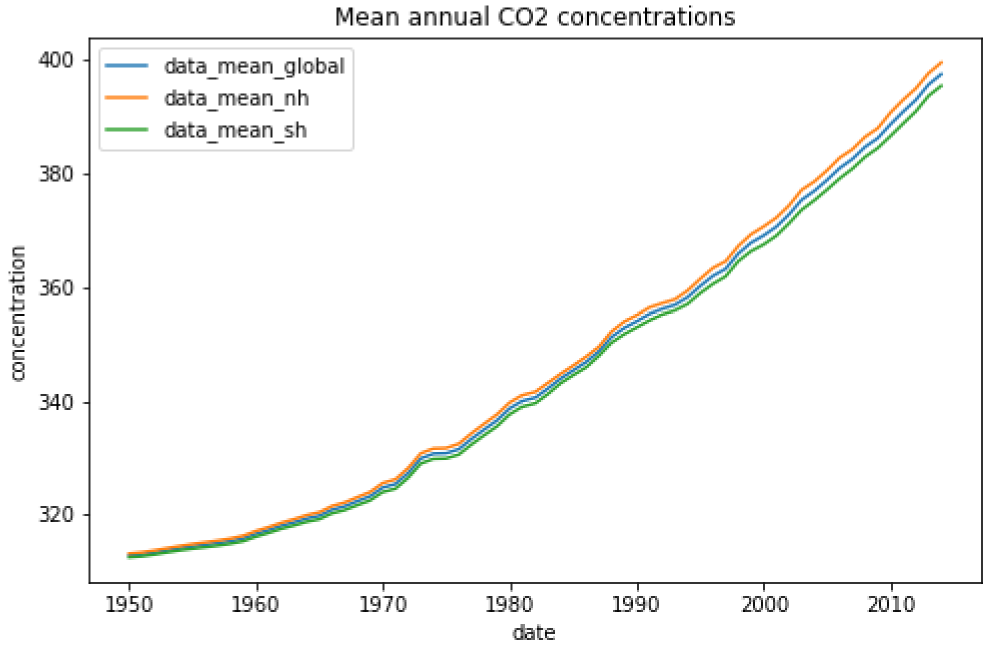

Figure 1:

To visualize this data we need to aggregate the data by month when they were recorded. There are many different ways to parse this data, here is my example:

Listing 3: "Parsing monthly data"

```
monthly_df = pd.DataFrame(df)
nh_df = pd.DataFrame(df['data_mean_nh'])
nh_df['geo'] = "data_mean_nh"
nh_df.columns = ['value', 'geo']
gl_df = pd.DataFrame(df['data_mean_global'])
gl_df['geo'] = "data_mean_global"
gl_df.columns = ['value', 'geo']
sh_df = pd.DataFrame(df['data_mean_sh'])
sh_df['geo'] = "data_mean_sh"
sh_df.columns = ['value', 'geo']
monthly_df = pd.concat([nh_df, gl_df, sh_df])
monthly_df['month'] = monthly_df.index.strftime('%b')
```

And plotting this data using `Seaborn` library and `boxplot()` function.

Listing 4: "Plotting monthly data"

```
plt.figure(figsize=(12, 6))
sns.set(font_scale=1.5)
```

```
ax = sns.boxplot(
    x='month',y='value',hue='geo', data=monthly_df,
    palette="bright")
handles, labels = ax.get_legend_handles_labels()
ax.set_title("Spread of monthly CO2 concentrations")
ax.legend(handles, labels, bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0.)
ax.set_ylabel('concentration')
```

On the figure 2 we can see the result of above's code execution.

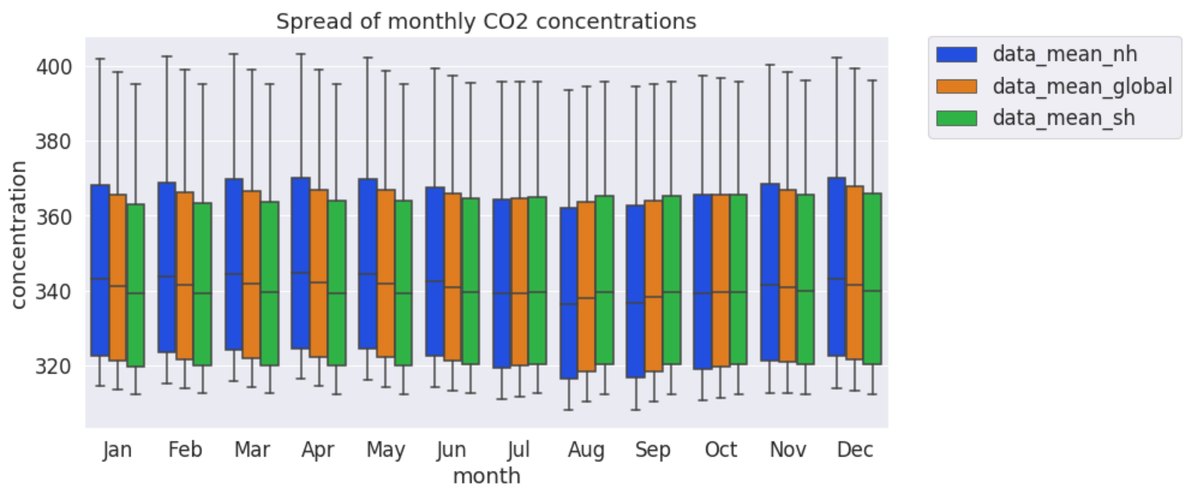

Figure 2:

Task 4

Choose one year and compare monthly CO2 concentrations in both hemispheres using bar plot.

To complete this task we can use function `barplot()` from the `Seaborn` package.

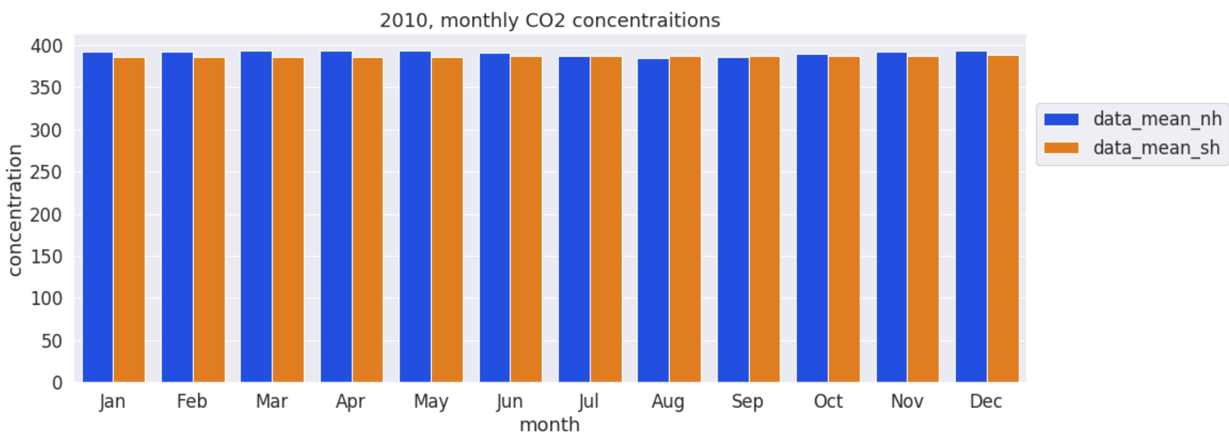

Figure 3:

Task 5

Plot monthly CO2 concentrations in both hemispheres. Examine seasonality using autocorrelation function (`acf` in `Statsmodels` library) with lags up to 15. Interpret the results.

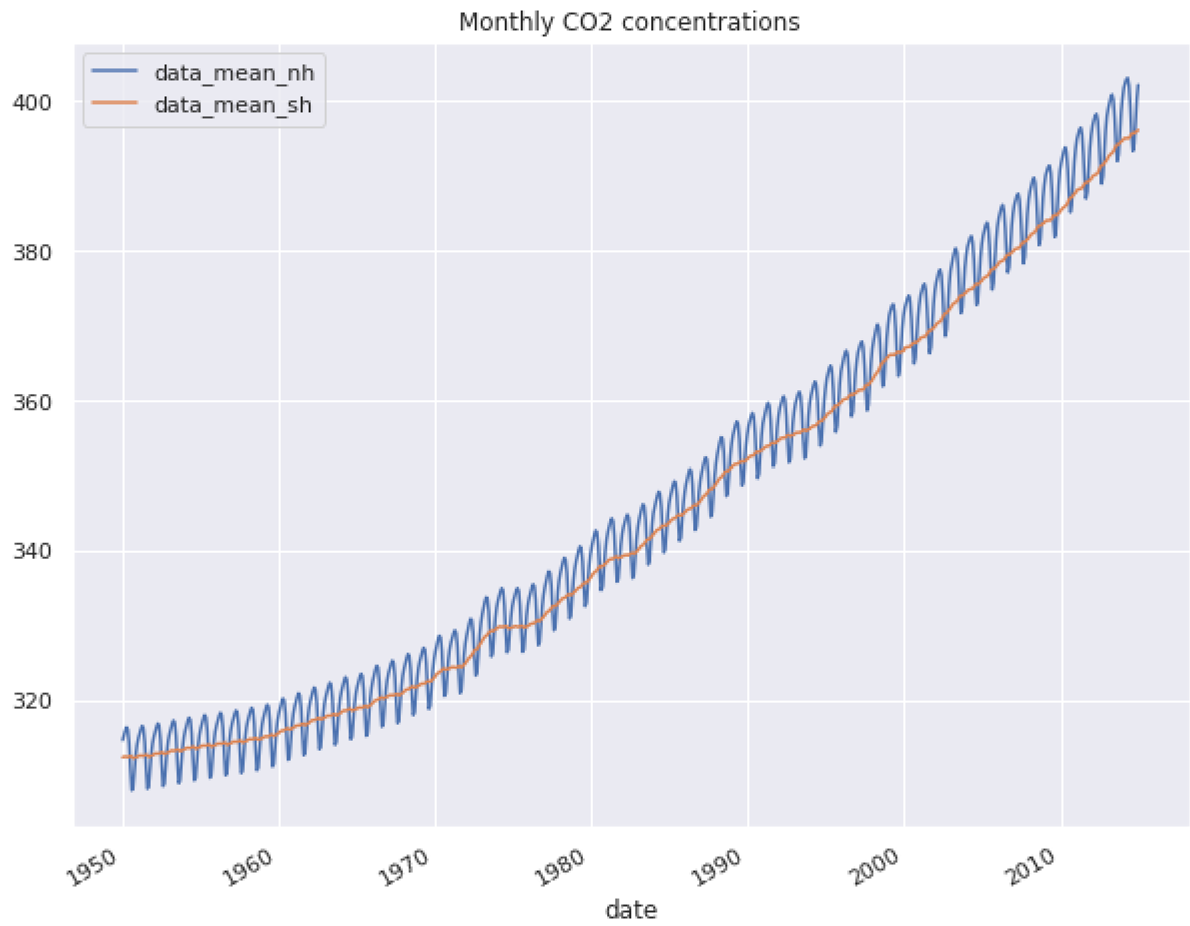

Figure 4:

From the figure 5 and 6 we can conclude that we expect the ACF for the time series to be strong to a lag of k and the inertia of that relationship would carry on to subsequent lag values.

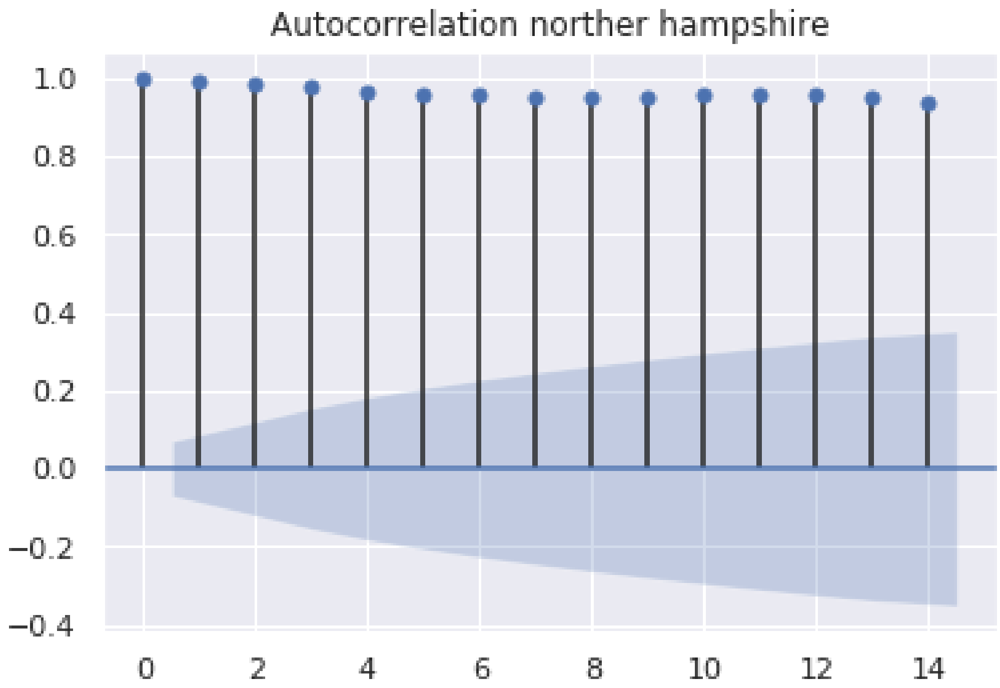

Figure 5:

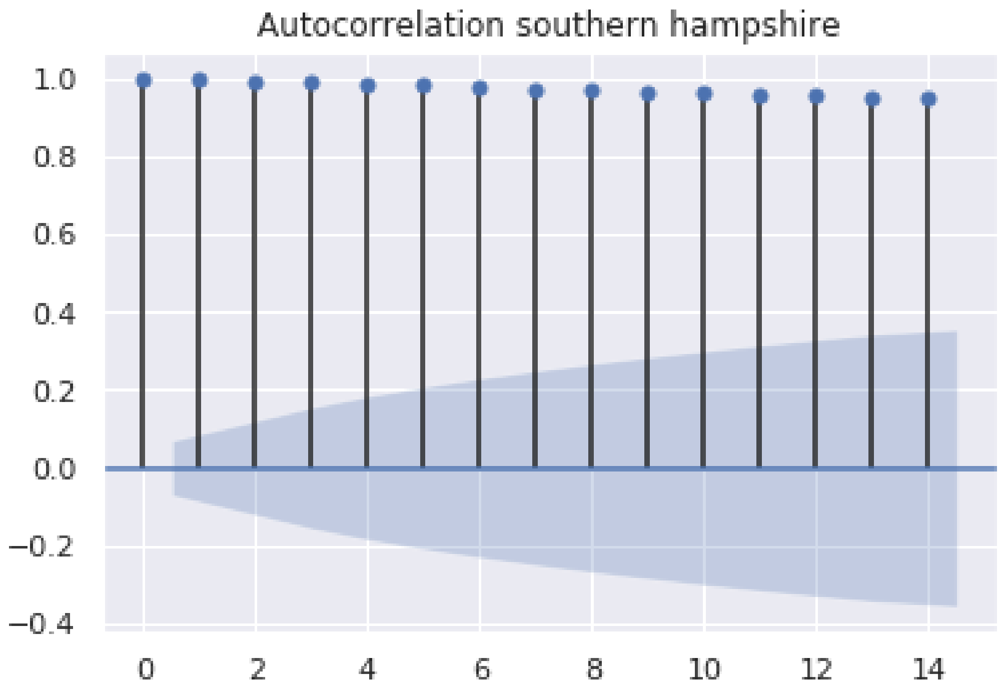

Figure 6:

Task 6

Decompose all series using additive and multiplicative models. Examine trends, seasonal, and residual components.

To decompose the series we will use `seasonal_decompose` function from `Statsmodels` package as follows:

Listing 5: "Plotting seasonality"

```
result = seasonal_decompose(sh_df, model='additive')
```

On the graphs 7 and 8 we clearly see that the trend is positive and the seasonality information extracted from the series does seem reasonable. Comparing northern and southern hemisphere seasonality is more clearly in northern hemisphere.

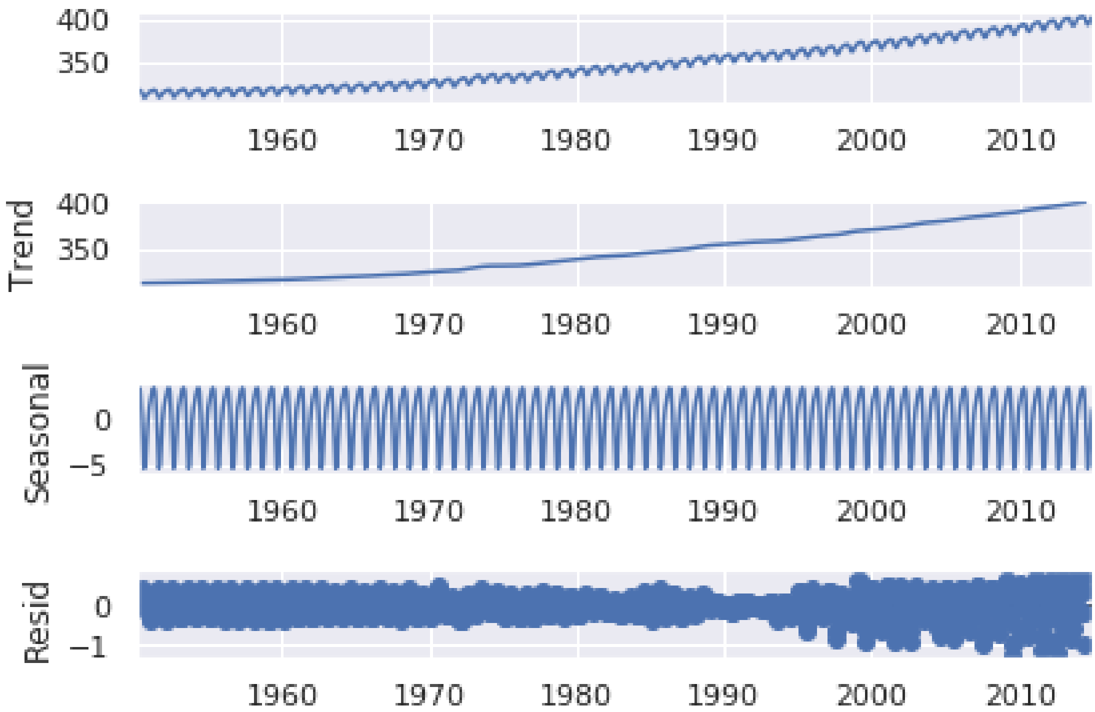

Figure 7: Seasonal decomposition for northern hemisphere (additive)

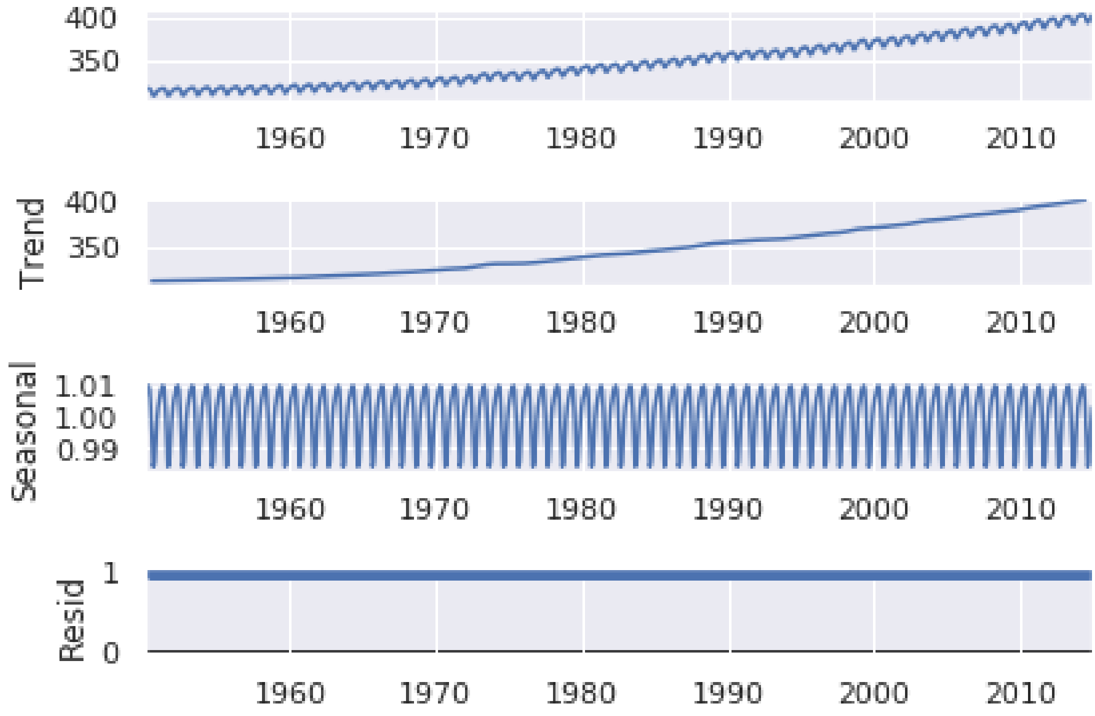

Figure 8: Seasonal decomposition for northern hemisphere (multiplicative)

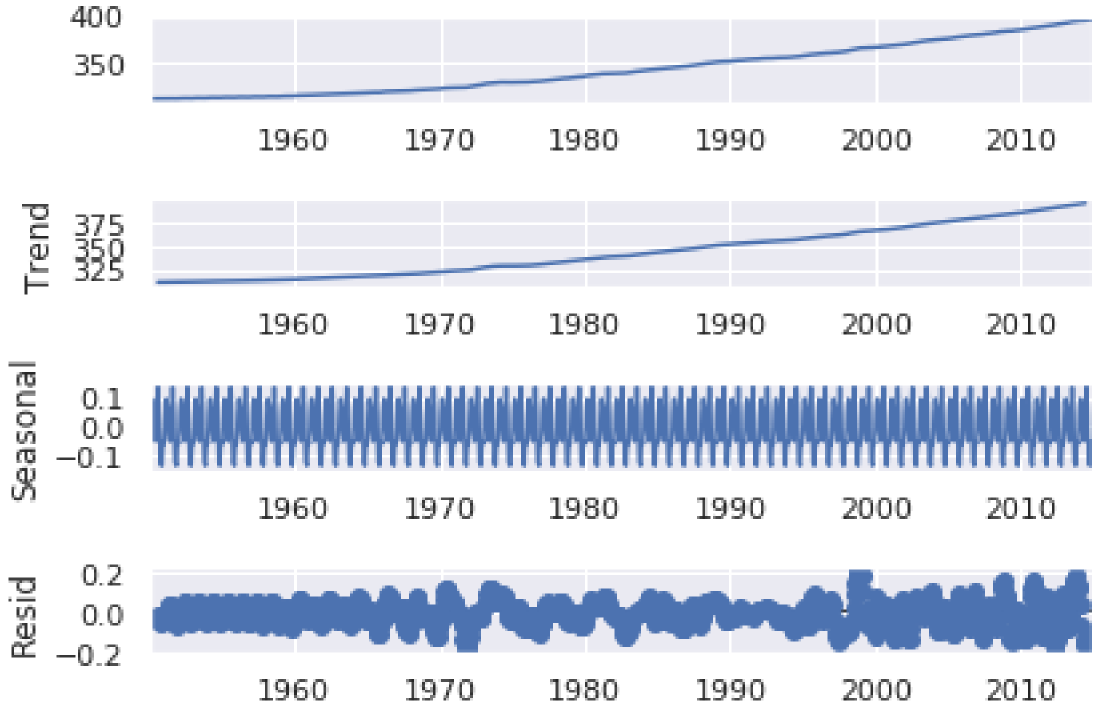

Figure 9: Seasonal decomposition for southern hemisphere (additive)

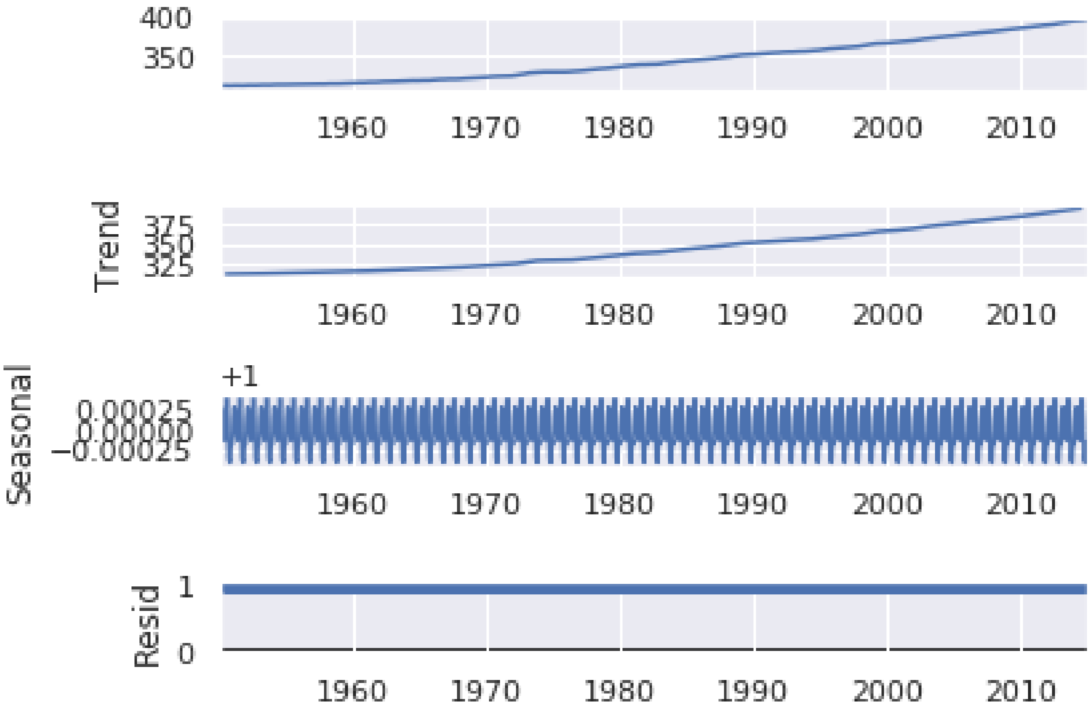

Figure 10: Seasonal decomposition for southern hemisphere (multiplicative)