INDR 450/550

Spring 2022

Lecture 5: Transformations and Auto-correlation

Feb. 28, 2022

Fikri Karaesmen

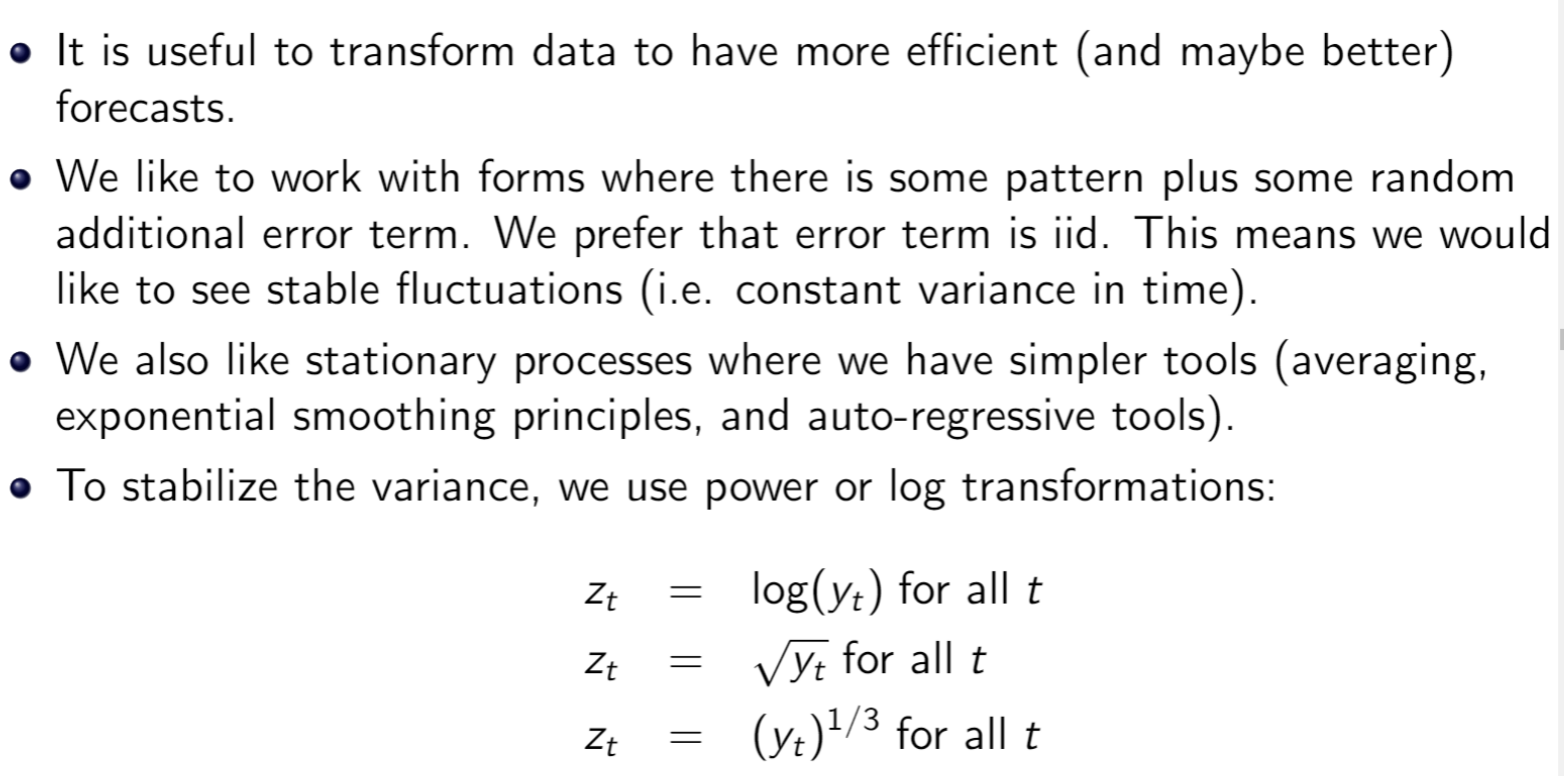
Announcements

- The first lab video was uploaded on Friday. The second lab will be uploaded on Friday this week. Please follow them.
- First HW will be assigned this week
  - You can work in groups of two for the homeworks
- I’ll also try to upload the python scripts that I use in the lectures.

Poll Exercise: INDR 450 Feb. 28 (1)

- Error Performance Measures etc

Data Transformations



- It is useful to transform data to have more efficient (and maybe better) forecasts.
- We like to work with forms where there is some pattern plus some random additional error term. We prefer that error term is iid. This means we would like to see stable fluctuations (i.e. constant variance in time).
- We also like stationary processes where we have simpler tools (averaging, exponential smoothing principles, and auto-regressive tools).
- To stabilize the variance, we use power or log transformations:

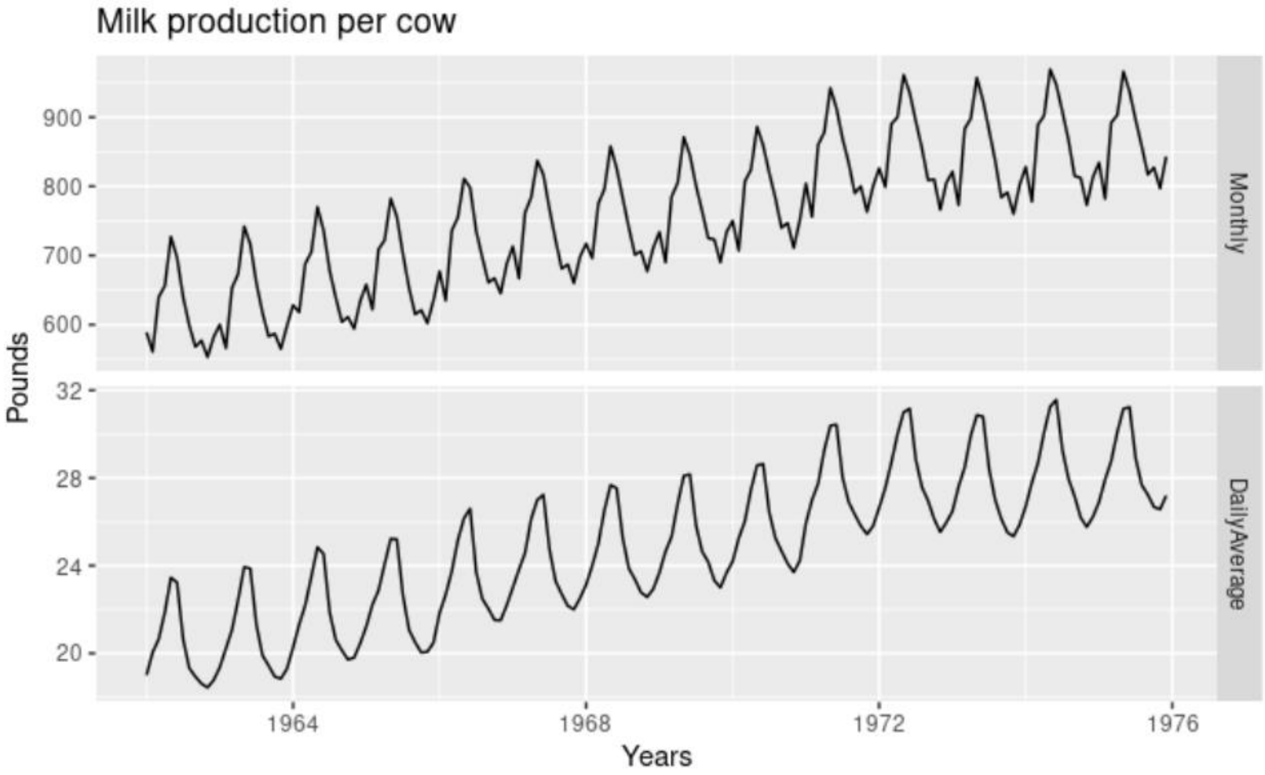
$$
\begin{aligned}
z_t &= \log(y_t) \text{ for all } t \\
z_t &= \sqrt{y_t} \text{ for all } t \\
z_t &= (y_t)^{1/3} \text{ for all } t
\end{aligned}
$$

Data Transformations

- We can the forecast for the transformed series and undo the transformation to have a prediction for the original model.
- Note that these transformations are non-linear and we need to be careful in computing prediction intervals (i.e. first compute them for the transformed model then convert them to the original scale).

Data Transformations: Example

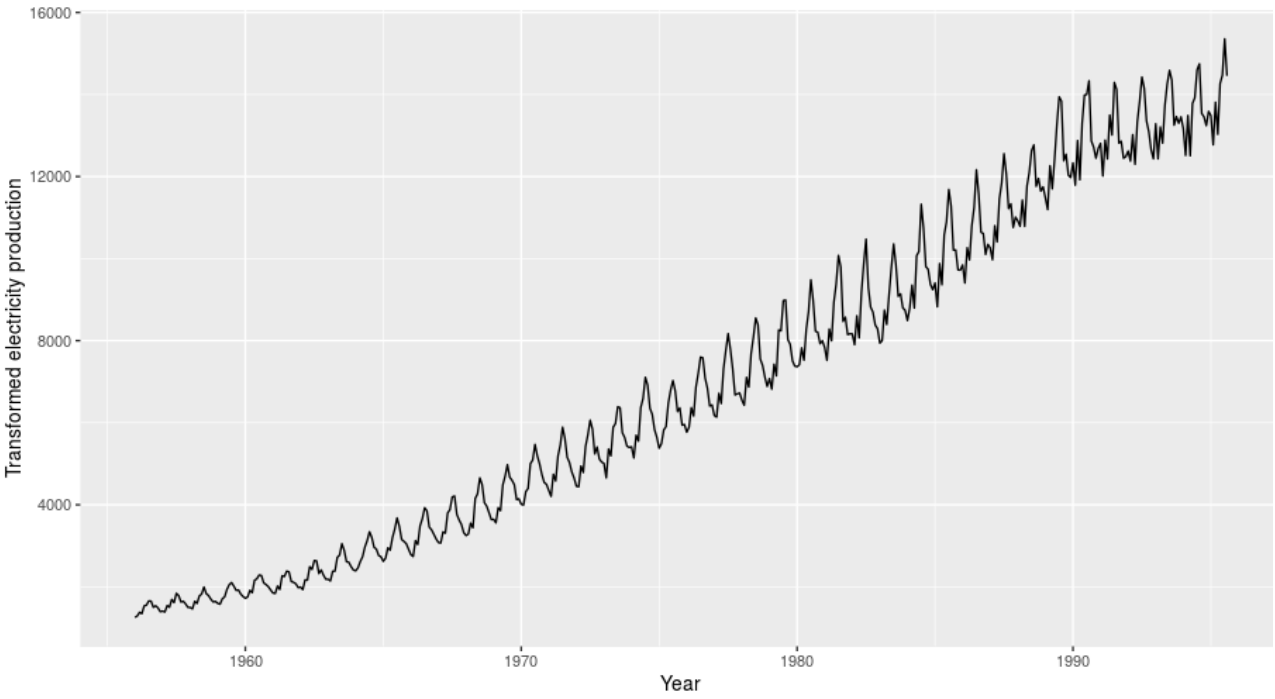
Calendar adjustments: monthly data is sometimes deceitful because each month does not have the same number of days. Convert months to a Daily average.



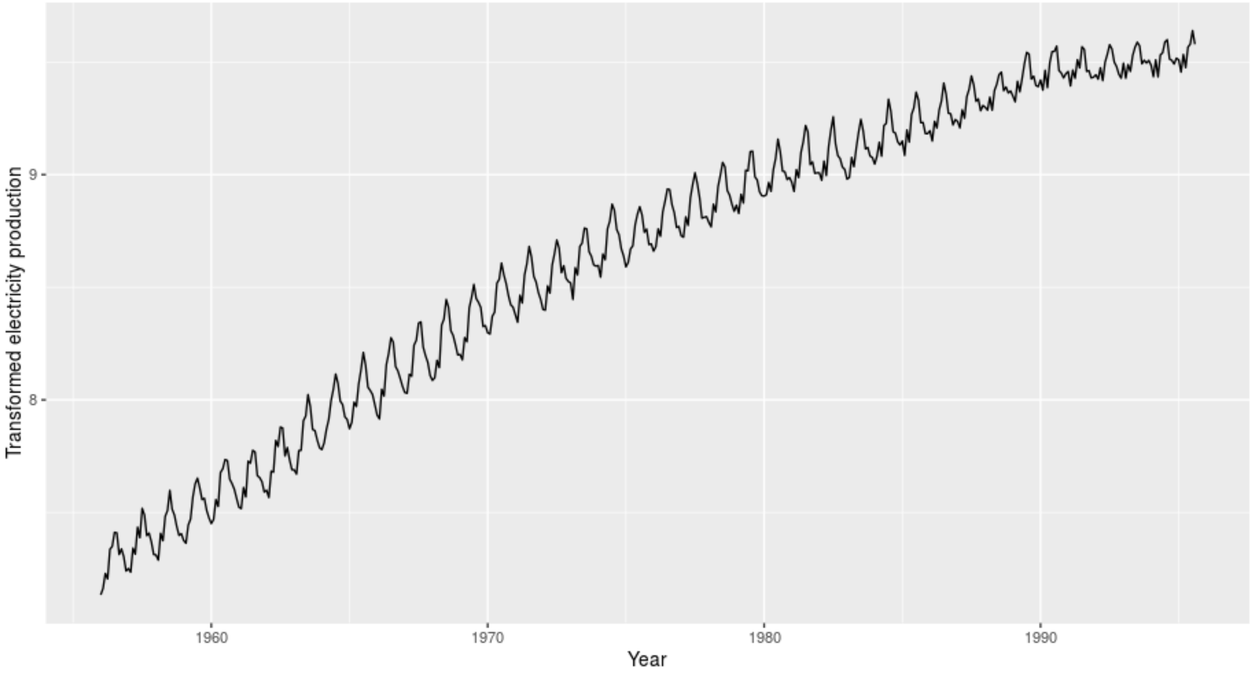
Source: Hydman and Athanasopoulos

Data Transformations: Example

Australian Monthly Electricity Production



Log Transformation



Source: Hydman and Athanasopoulos

Data Transformations: detrending (differencing)

- We can also transform the data to convert the original series to a stationary series.
- Take the example of:

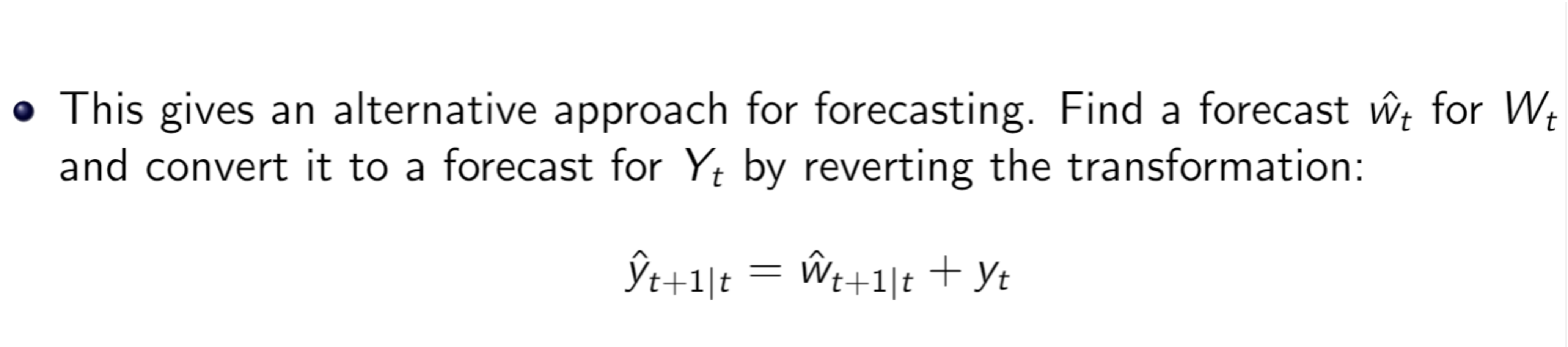
$$Y_t = c + bt + \epsilon_t$$

- Now consider the transformation:

$$W_t = Y_t - Y_{t-1} = b + \epsilon_t - \epsilon_{t-1} = b + \epsilon'_t$$

- This gives an alternative approach for forecasting. Find a forecast $\hat{w}_t$ for W_t and convert it to a forecast for Y_t by reverting the transformation.

Data Transformations



- This gives an alternative approach for forecasting. Find a forecast $\hat{w}_t$ for W_t and convert it to a forecast for Y_t by reverting the transformation:

$$\hat{y}_{t+1|t} = \hat{w}_{t+1|t} + y_t$$

Data Transformations: deseasonalizing

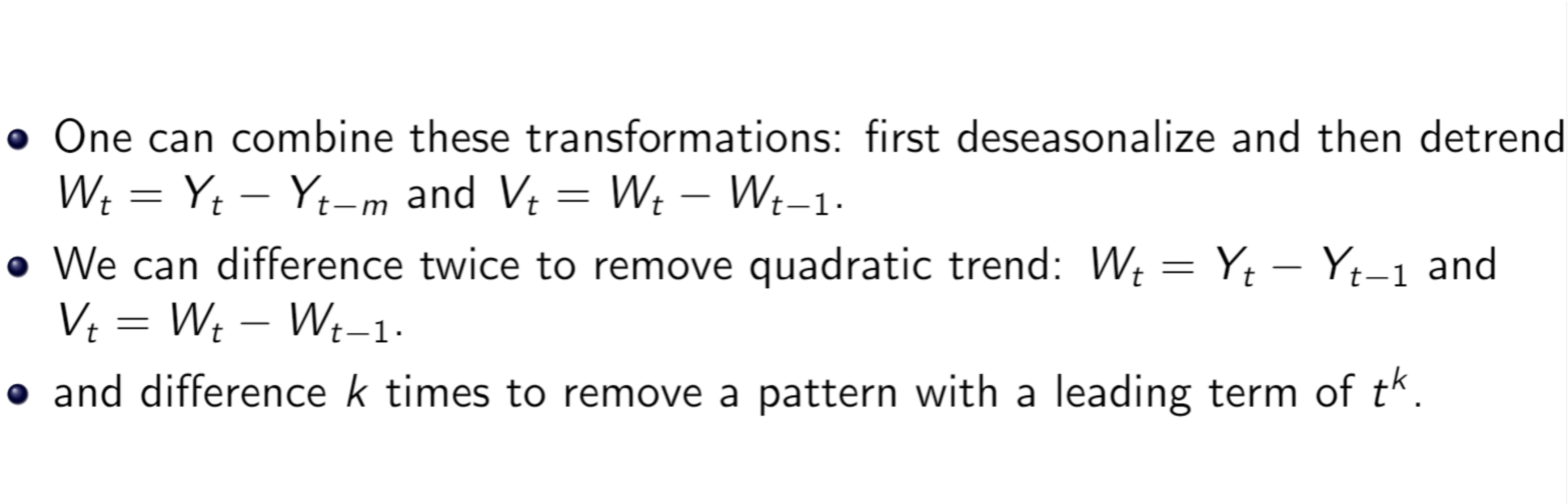
- The transformation $W_t = Y_t - Y_{t-1}$ is known as simple detrending, it removes a linear trend.
- A similar transformation that applies to (simple) seasonal data is called deseasonalizing:

$$W_t = Y_t - Y_{t-m}$$

for a season length of m. A reasonable forecast is then

$$\hat{y}_{t+1|t} = \hat{w}_{t+1|t} + y_{t+1-m}$$

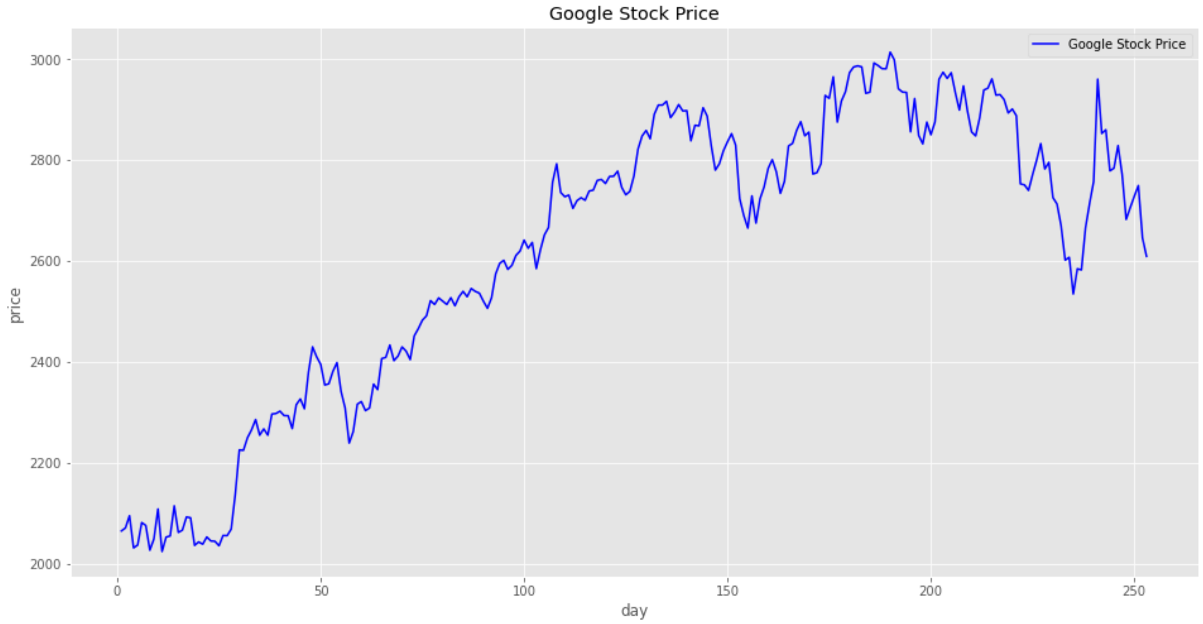
Data Transformations



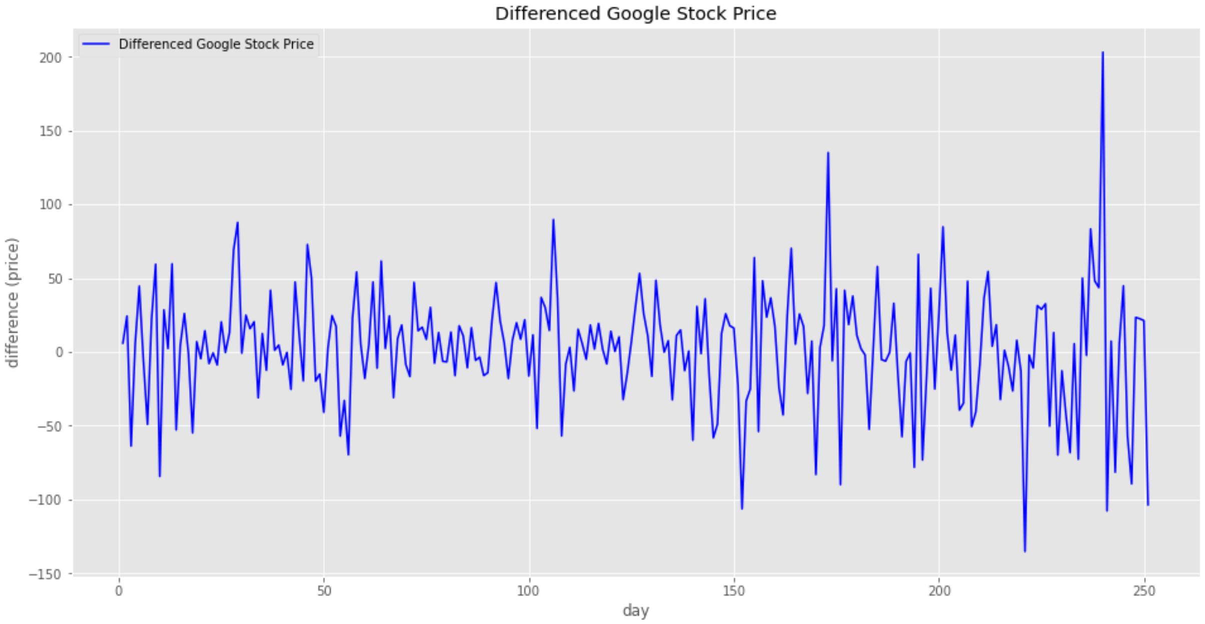
- One can combine these transformations: first deseasonalize and then detrend $W_t = Y_t - Y_{t-m}$ and $V_t = W_t - W_{t-1}$.
- We can difference twice to remove quadratic trend: $W_t = Y_t - Y_{t-1}$ and $V_t = W_t - W_{t-1}$.
- and difference k times to remove a pattern with a leading term of t^k.

Data Transformations: Google Stock Price Example

Google (Alphabet) Stock Price over the last year



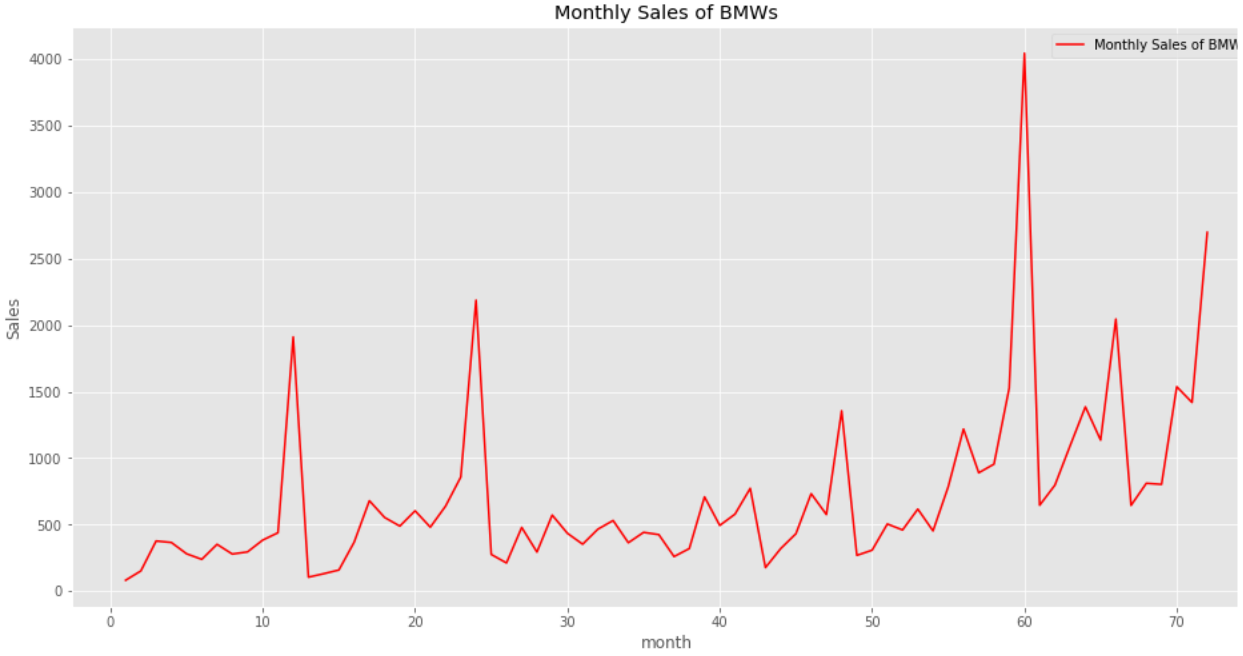
Differenced Google (Alphabet) Stock Price over the last year



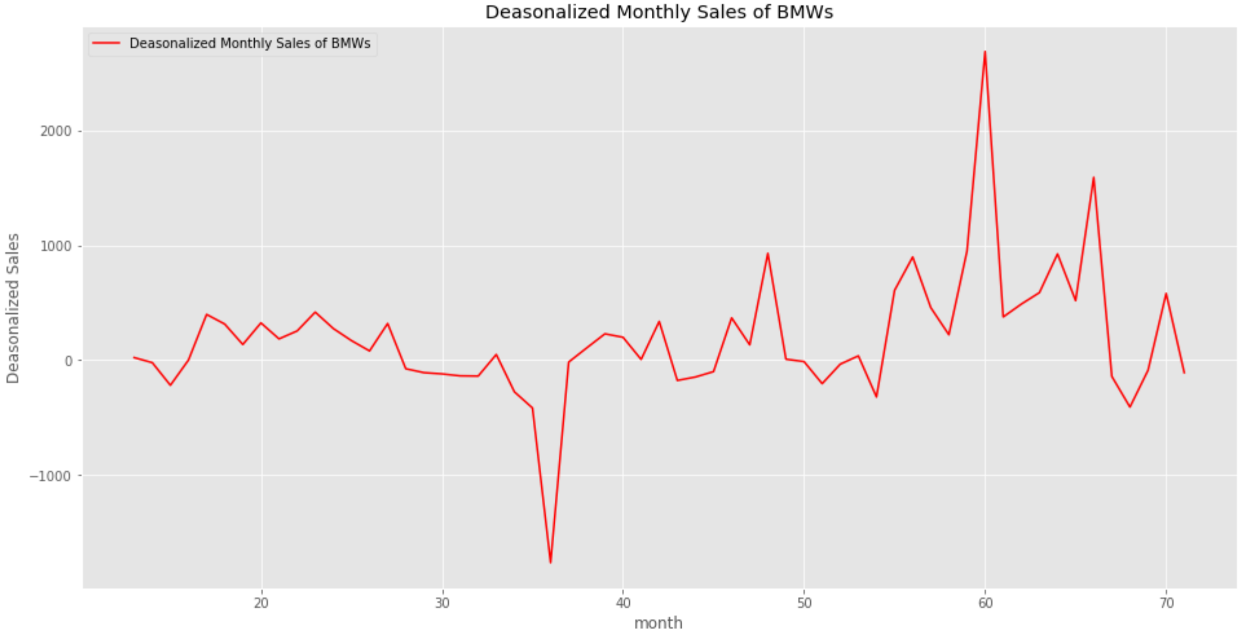
Data compiled on Feb. 22, 2022

Data Transformations: BMW Sales Example

Monthly BMW Sales since 2016



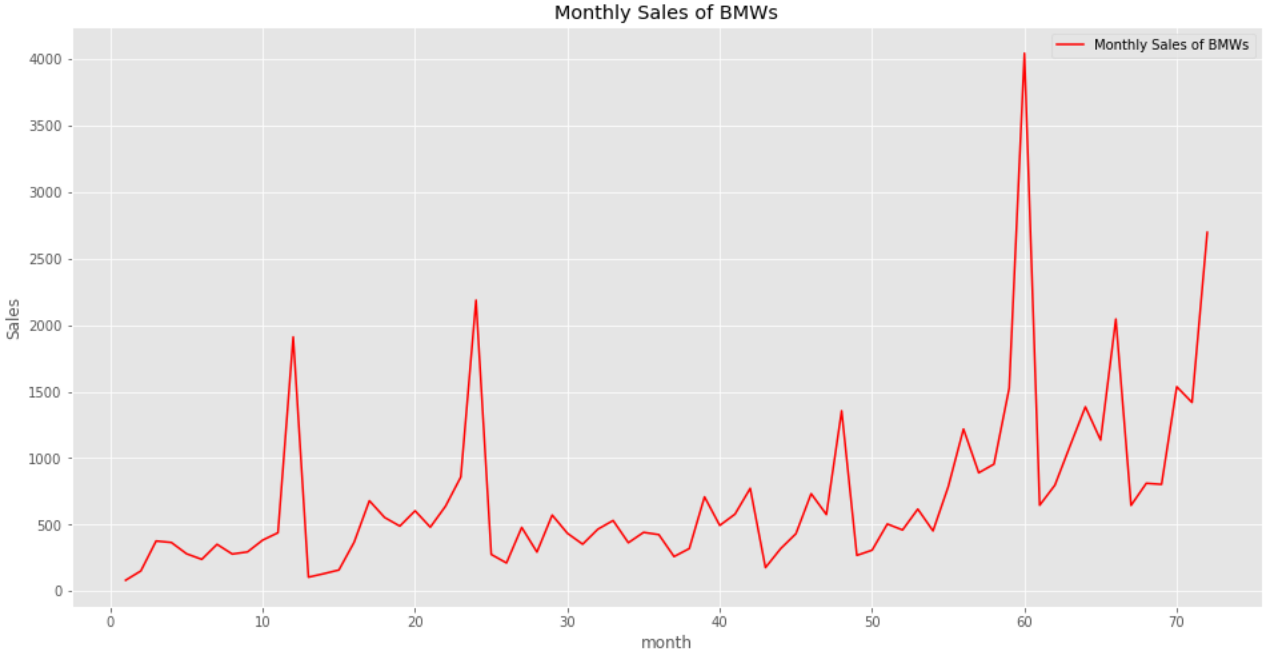
Deseasonalized Monthly BMW Sales



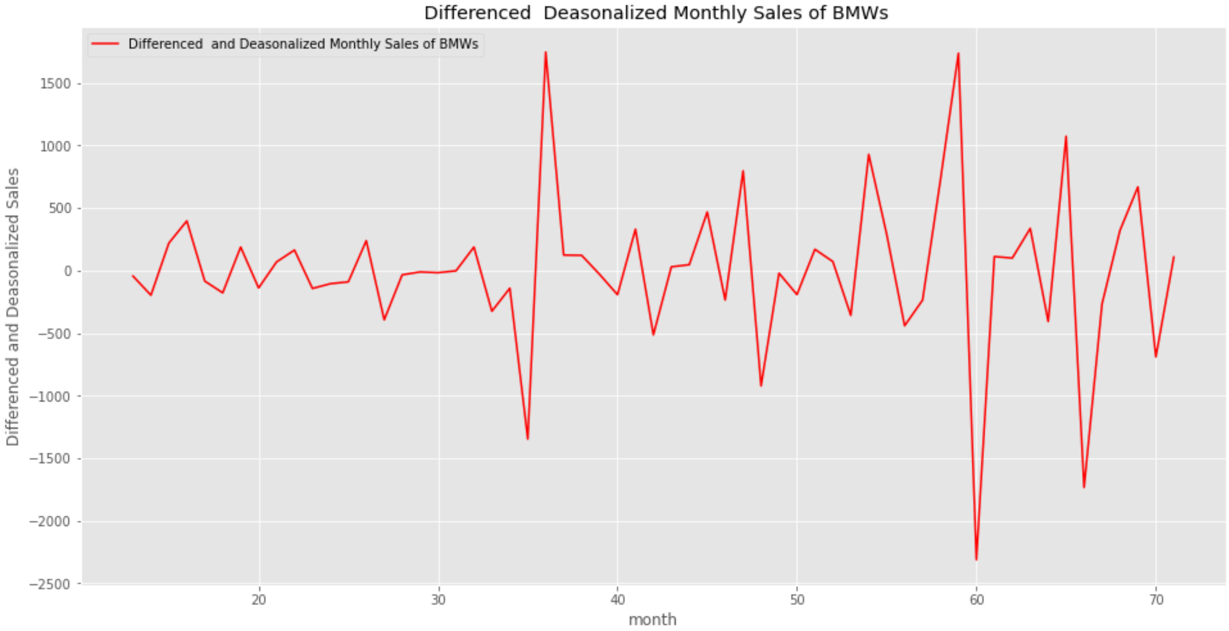
Data until Dec. 2021 (included)

Data Transformations: BMW Sales Example

Monthly BMW Sales since 2016

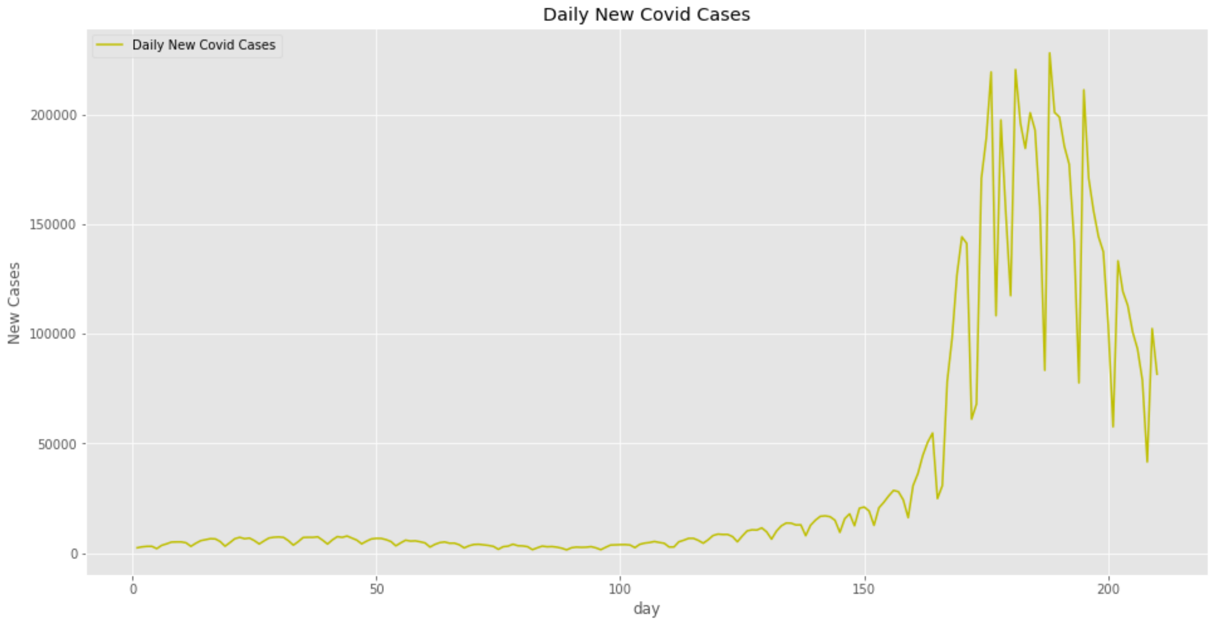


Deseasonalized and Differenced Monthly BMW Sales

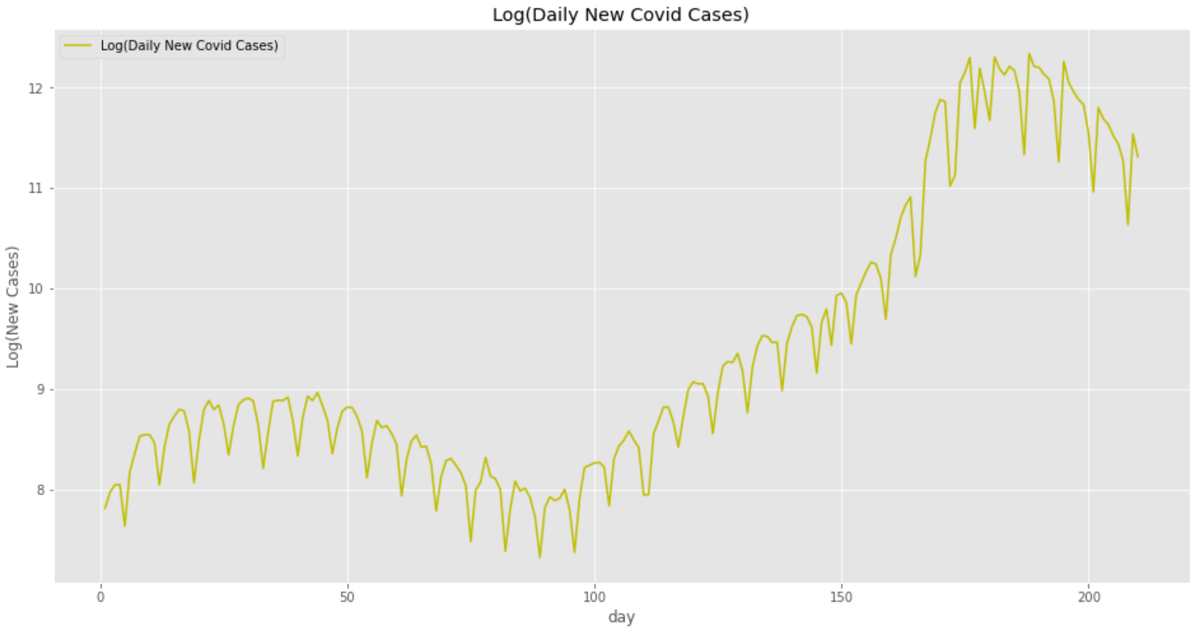


Data Transformations: Daily New Covid Cases in Italy (2021)

Daily New Covid Cases



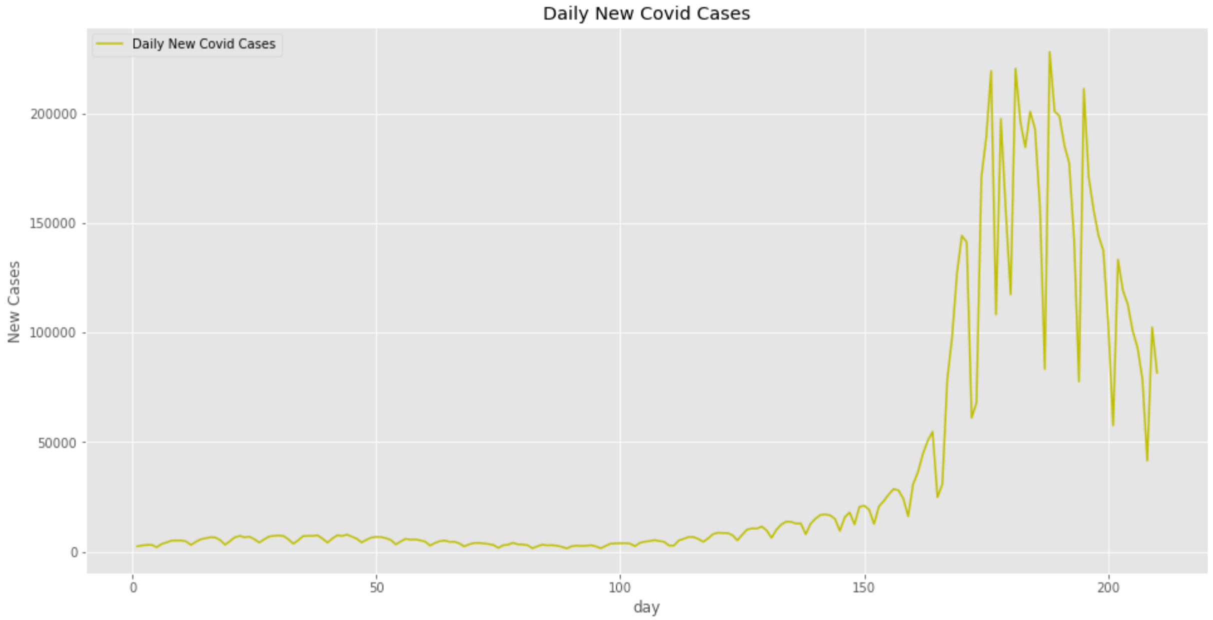
Log(Daily New Covid Cases)



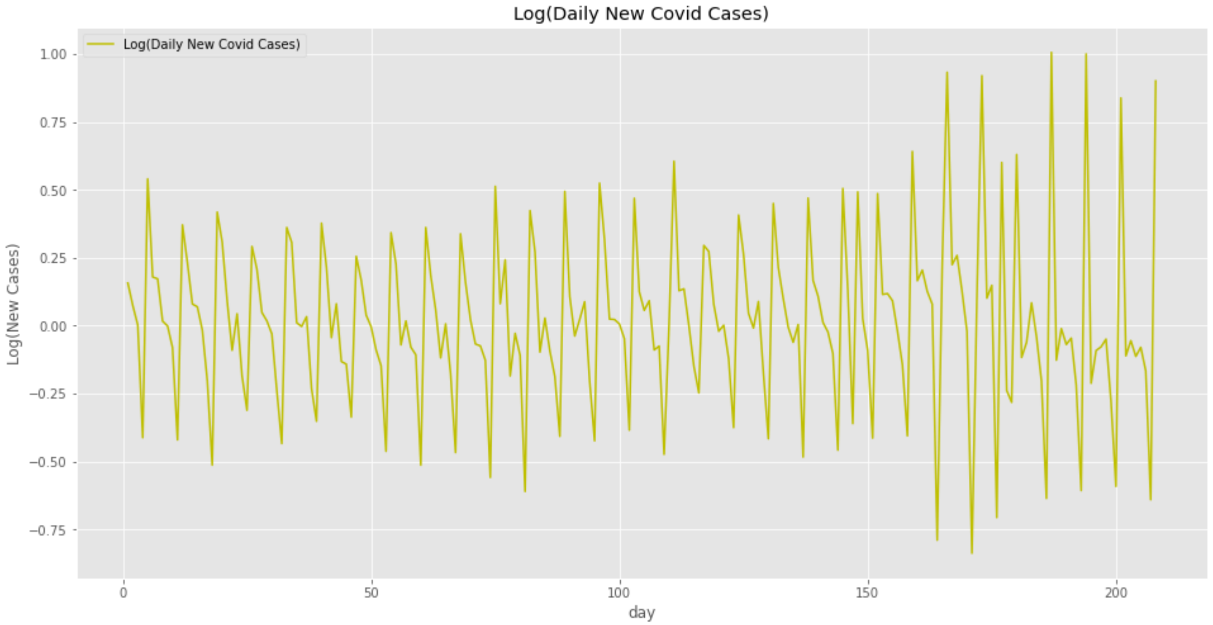
Data from mid 2021

Data Transformations: Daily New Covid Cases in Italy (2021)

Daily New Covid Cases



Diff(Log(Daily New Covid Cases))

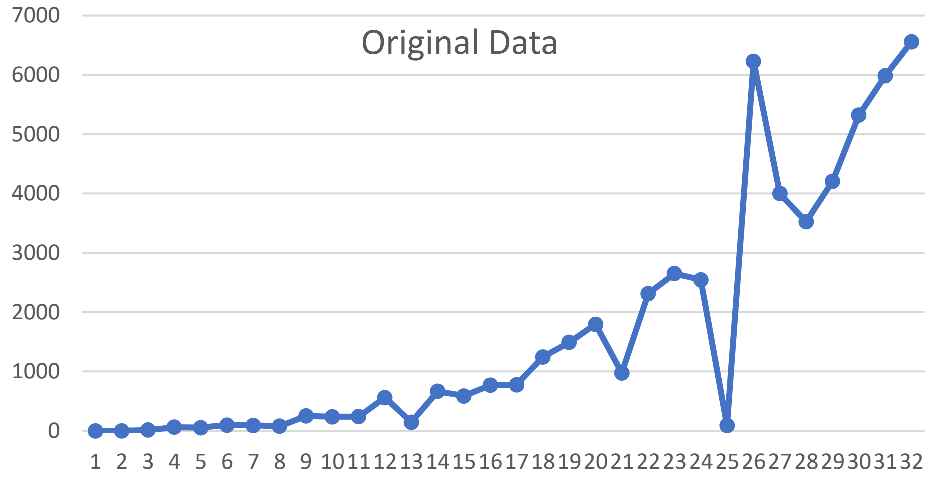


Short Case Study: Predicting daily covid cases

- The goal here is not to reach sophisticated predictions that can guide public policy but to perform a rapid analysis using actual data.
  - using what we learned in the course so far
- I'll look at data from Italy (appears reliable)
  - Regulary updated
- This analysis looks at the beginning of the pandemic. Official data until March 21, 2020 (included).
- I start at the time of first positive case in Italy.

Predicting daily covid cases: Italy

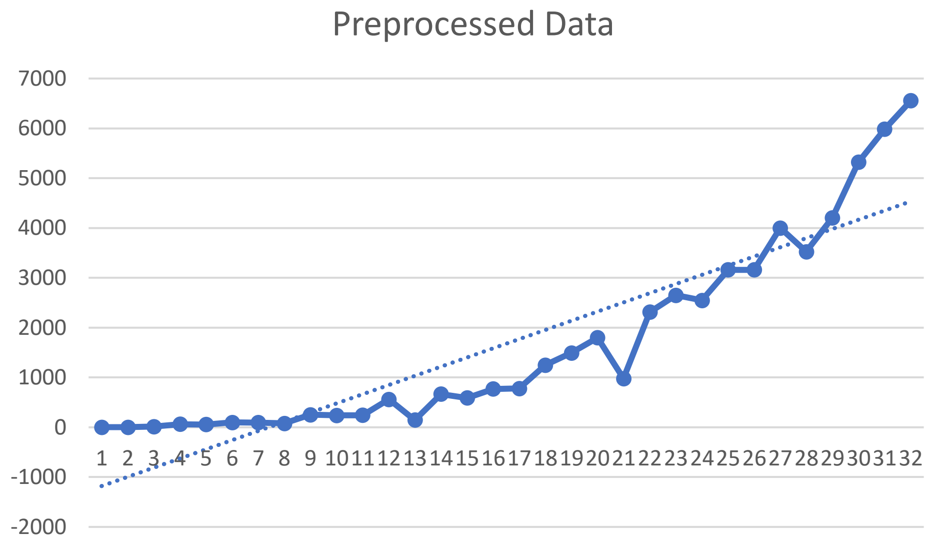
- 32 days of data.
- First some preprocessing:



- Apparently, some observation/reporting problem took place on day 25 where only 90 cases were reported. On day 24 there were 2547 cases and day 26 there were 6130 cases!
- I'll replace the data on day 25 by (2547+6130)/2, the average of the previous day and the next day.
  - There could be other ways of correcting. My choice of correction increases the total number of cases.

Predicting daily covid cases: Italy

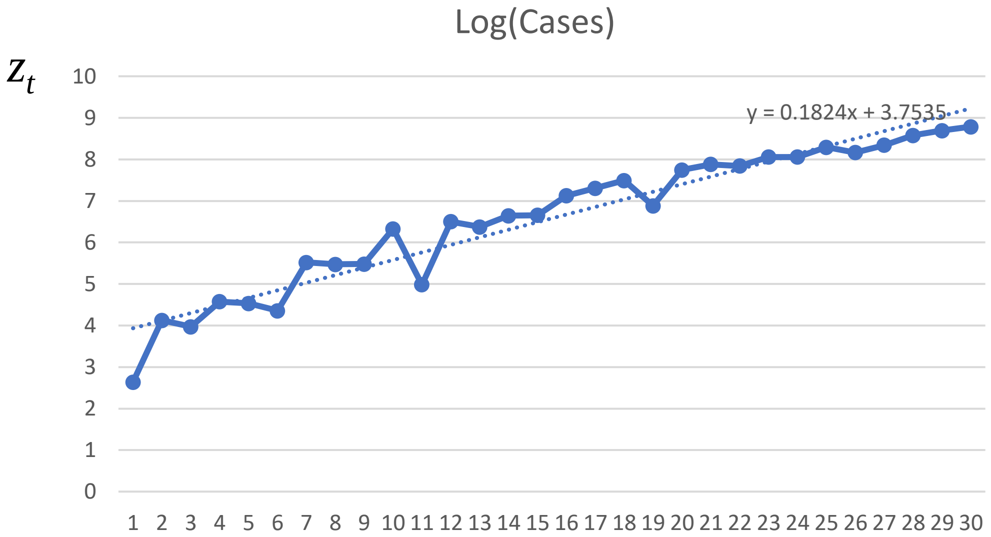
- This is the cleaned data.



- As we keep hearing, there is an exponential increase.
- Let’s do a log transformation.

Predicting daily covid cases: Italy

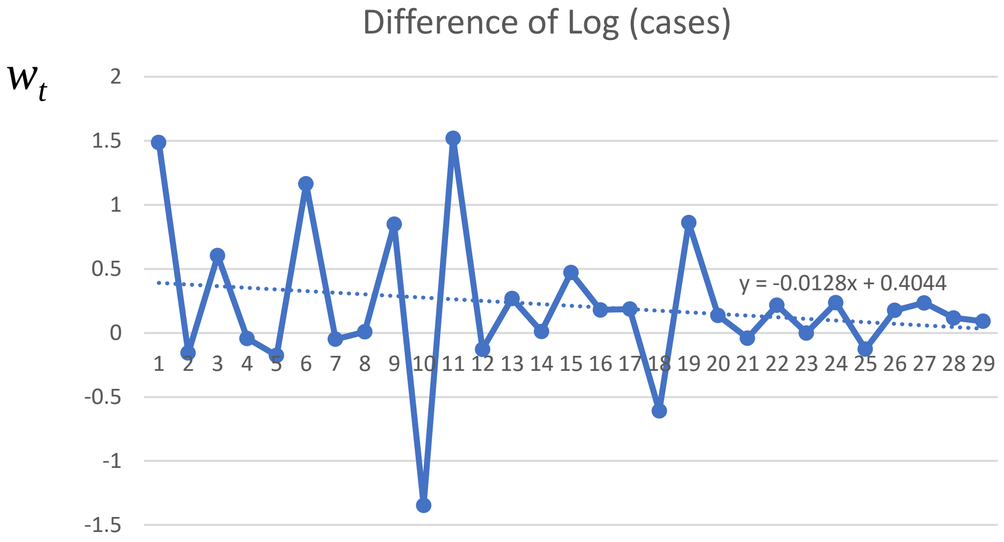
- $z_t = \ln(y_t)$



- There is a significant linear trend. This confirms the exponential growth of the original data. We'll take differences to remove it.

Predicting daily covid cases: Italy

- $w_t = z_t - z_{t-1}$



- There is a small decreasing linear trend after differencing. We can run a linear regression to decide whether that's statistically significant. If so, we might want to take a second difference.

Predicting daily covid cases: Italy

- We test the following linear model to see whether w_t has trend:

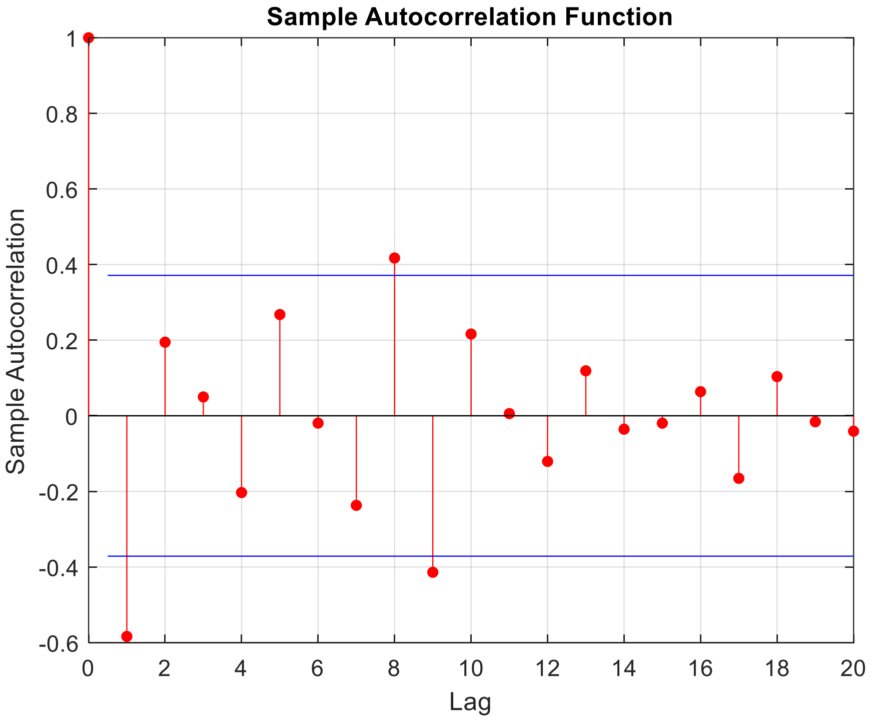
$$w_t=b_0+b_1t+\varepsilon_t$$

- The results are as follows:
- The slope is statistically insignificant (p-value->0.324).
- So first order differencing was sufficient to remove the trend.

```
Linear regression model:
   y ~ 1 + x1

Estimated Coefficients:
                   Estimate        SE          tStat        pValue
                 __________    _________    _________    _________

   (Intercept)       0.4044      0.21934       1.8437     0.076231
   x1             -0.012824     0.012771      -1.0042      0.32422


Number of observations: 29, Error degrees of freedom: 27
Root Mean Squared Error: 0.575
R-squared: 0.036,  Adjusted R-Squared 0.000298
F-statistic vs. constant model: 1.01, p-value = 0.324
```

Predicting daily covid cases: Italy

- Finally, we can check the autocorrelation plot of w_t.



- There is some AC at lags 1, 8 and 9. But our data set is too small to fit an AR model so we'll ignore that and fit a simple model.

Predicting daily covid cases: Italy

- Proposed model:

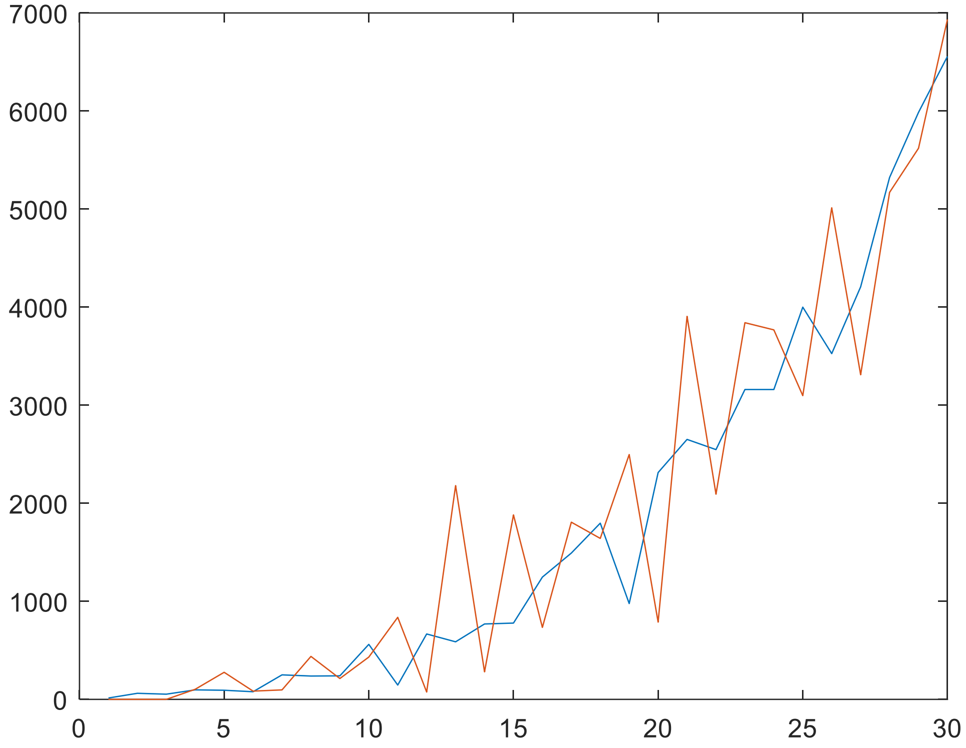
$$G_t = z_{t\text{-}1} + ((z_{t\text{-}1} - z_{t\text{-}2}) + (z_{t\text{-}3} - z_{t\text{-}4}))/2$$

$$F_t = e^{Gt}$$

- G_t estimates the current slope by averaging the last two independent differences and uses the last observation as an estimator of the current level.
- The final forecast F_t reverses the log transformation.
- Note that F_t is an unbiased estimator for the mean of: $y_t = e^{a+bt+\varepsilon t}$

Predicting daily covid cases: Italy

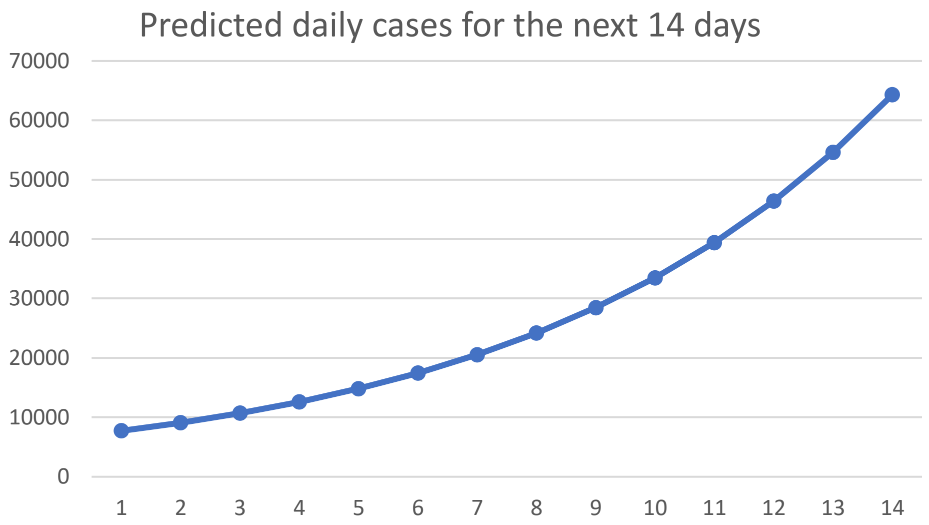
- Blue: observed new daily covid cases
- Orange: prediction from the model



Root Mean Squared Error = 627.983

Predicting daily covid cases: Italy

- We are hoping that the rate of infection growth slows soon. If it does not, our model predicts that in a week the new daily number of infections would be around 20000 and in two weeks they would be around 64000.

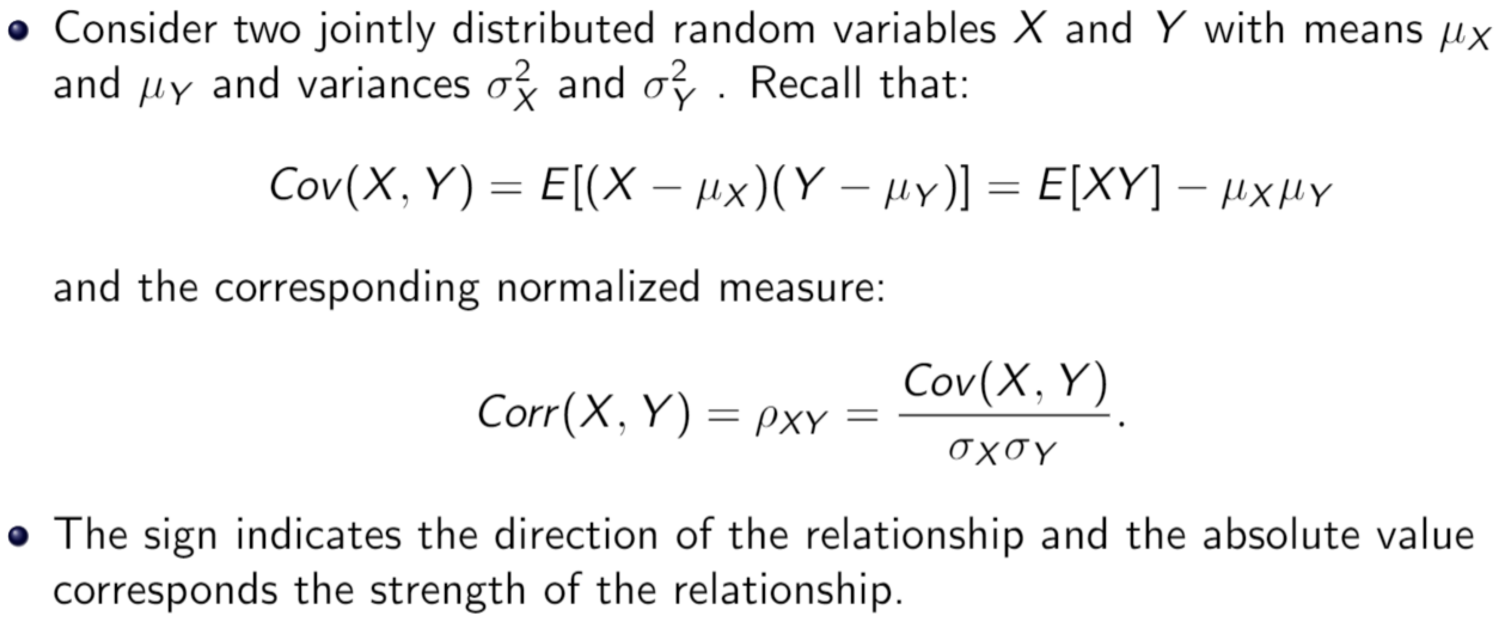


- “The lockdown should hopefully prove these predictions wrong.” was my comment from March 2020. (This turned out to be true for the first peak)

Poll Exercise

- Data transformations

Correlation Across Time: ARIMA forecasts



- Consider two jointly distributed random variables X and Y with means μ_X and μ_Y and variances σ_X^2 and σ_Y^2 . Recall that:

$$Cov(X, Y) = E[(X - \mu_X)(Y - \mu_Y)] = E[XY] - \mu_X \mu_Y$$

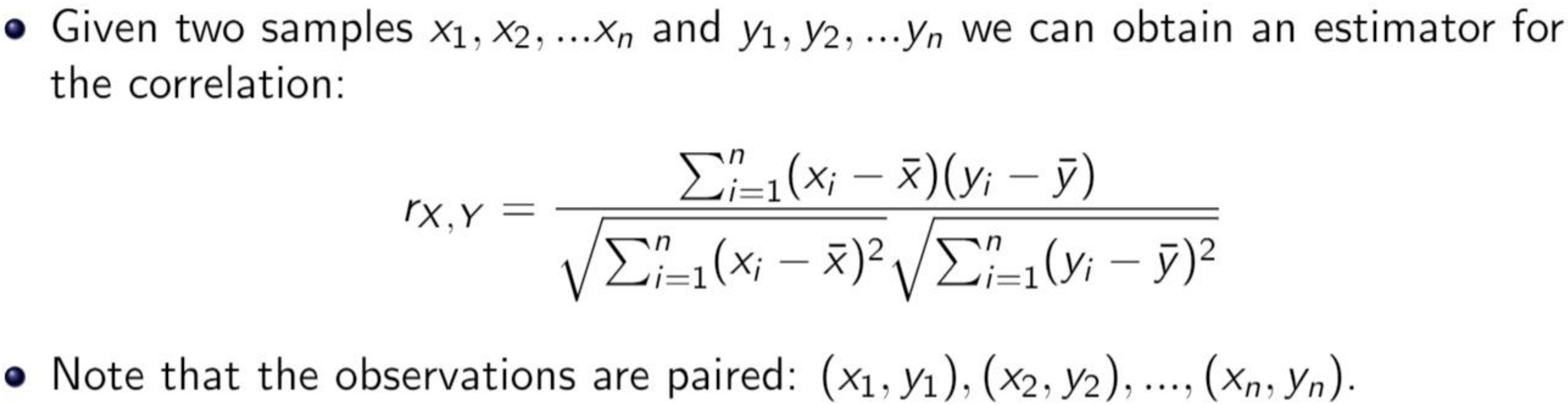
  and the corresponding normalized measure:

$$Corr(X, Y) = \rho_{XY} = \frac{Cov(X, Y)}{\sigma_X \sigma_Y}.$$

- The sign indicates the direction of the relationship and the absolute value corresponds the strength of the relationship.

Covariance and Correlation

- Sample estimators:



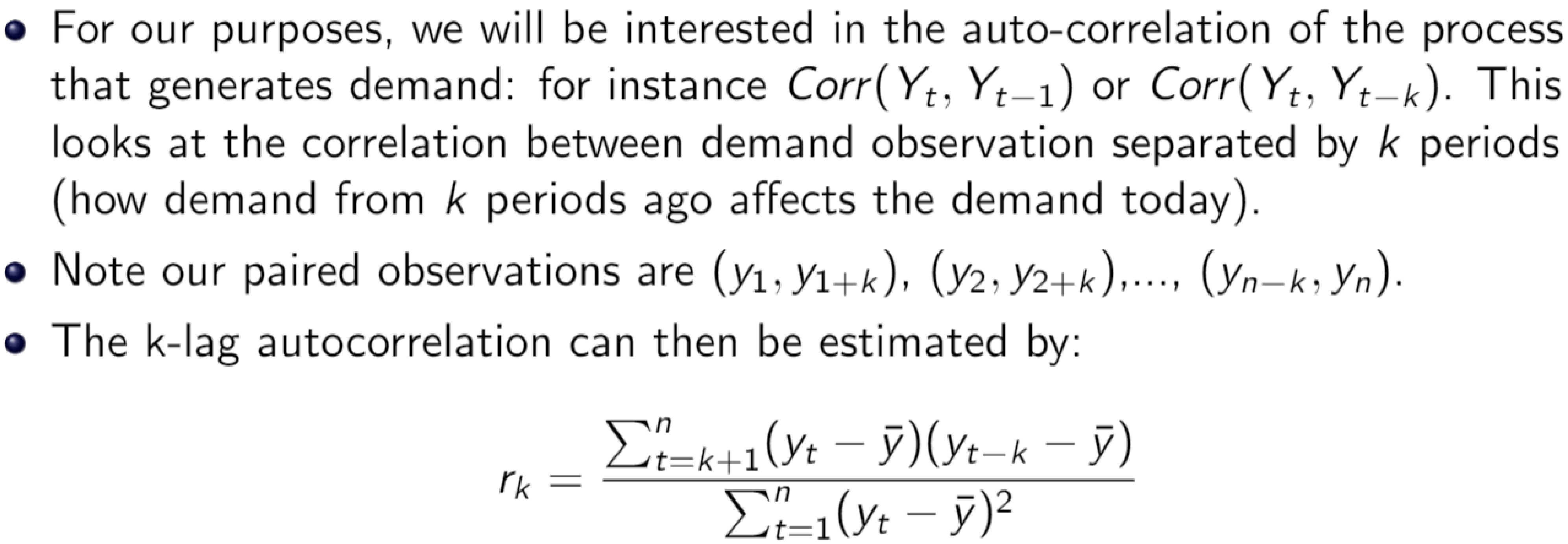
- Given two samples $x_1, x_2, ...x_n$ and $y_1, y_2, ...y_n$ we can obtain an estimator for the correlation:

$$r_{X,Y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- Note that the observations are paired: $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$.

Auto-Correlation

- Sample estimators:



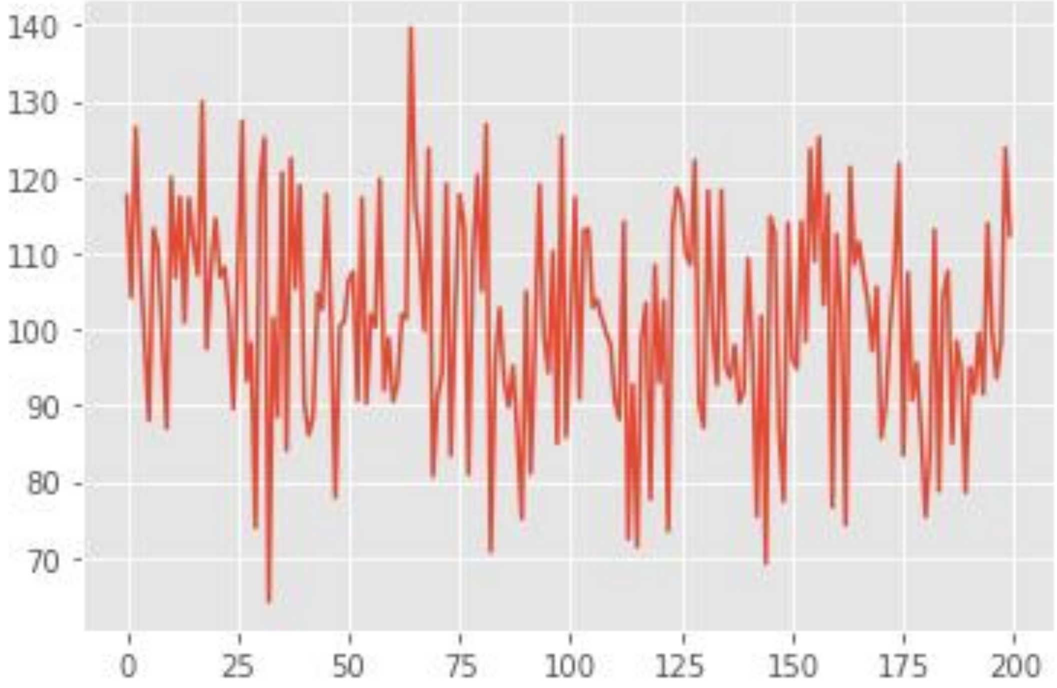
- For our purposes, we will be interested in the auto-correlation of the process that generates demand: for instance $Corr(Y_t, Y_{t-1})$ or $Corr(Y_t, Y_{t-k})$. This looks at the correlation between demand observation separated by k periods (how demand from k periods ago affects the demand today).
- Note our paired observations are (y_1, y_{1+k}), (y_2, y_{2+k}),..., (y_{n-k}, y_n).
- The k-lag autocorrelation can then be estimated by:

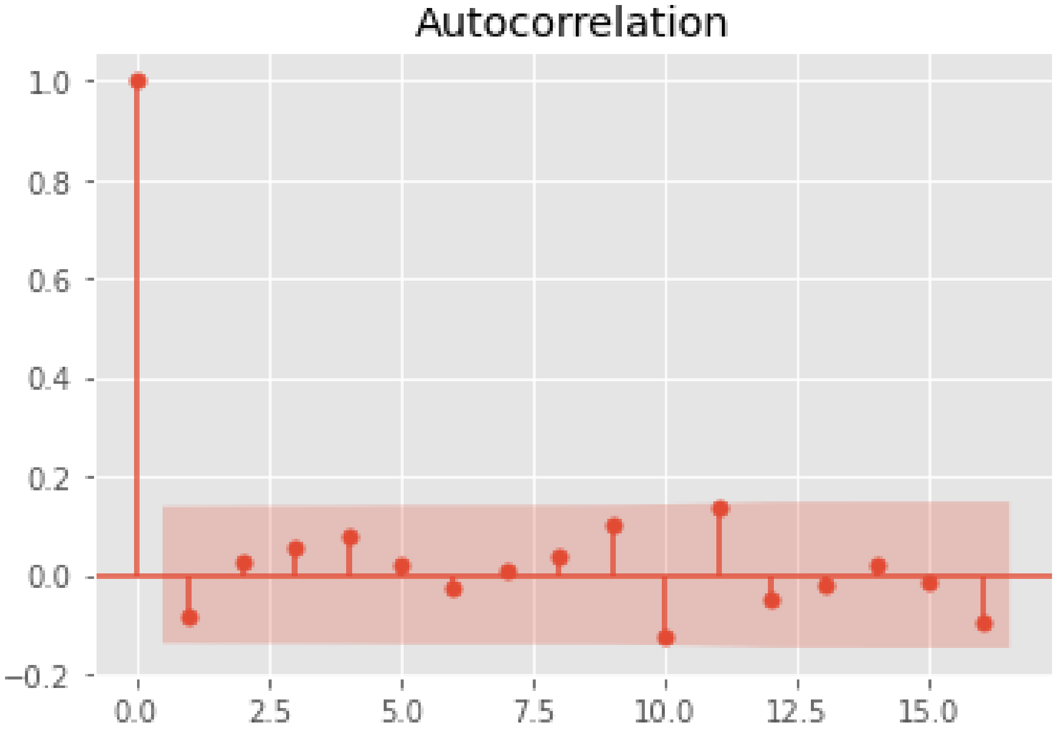
$$r_k = \frac{\sum_{t=k+1}^{n}(y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{n}(y_t - \bar{y})^2}$$

Auto-Correlation: stationary i.i.d demand

- Recall the simple model: $Y_t = c + \varepsilon_t$. Here's the autocorrelation structure:

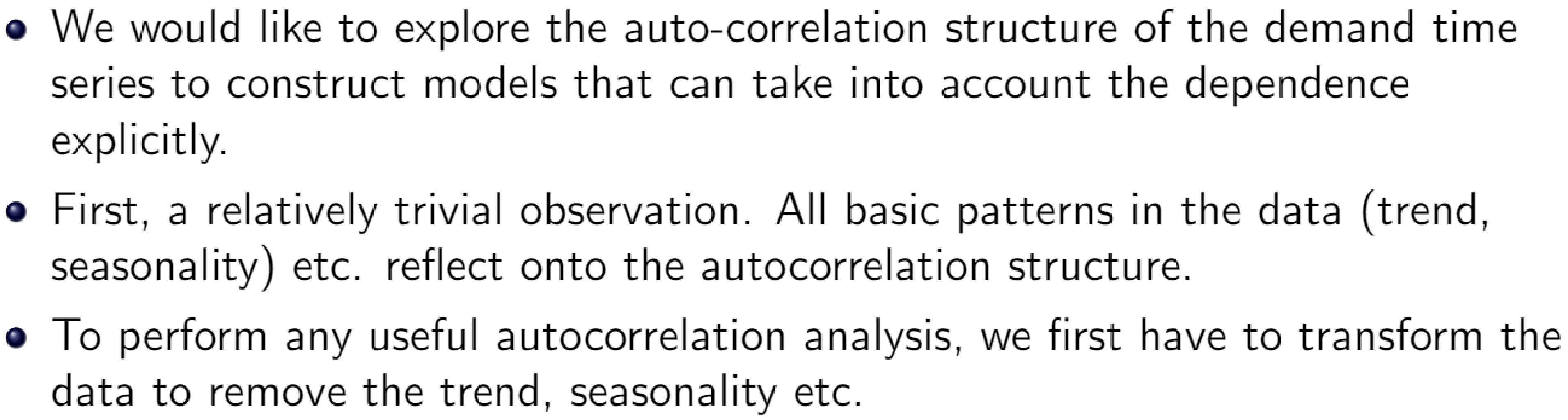
The data (generated randomly)





Auto-Correlation: The effects of patterns

- The effect of patterns:

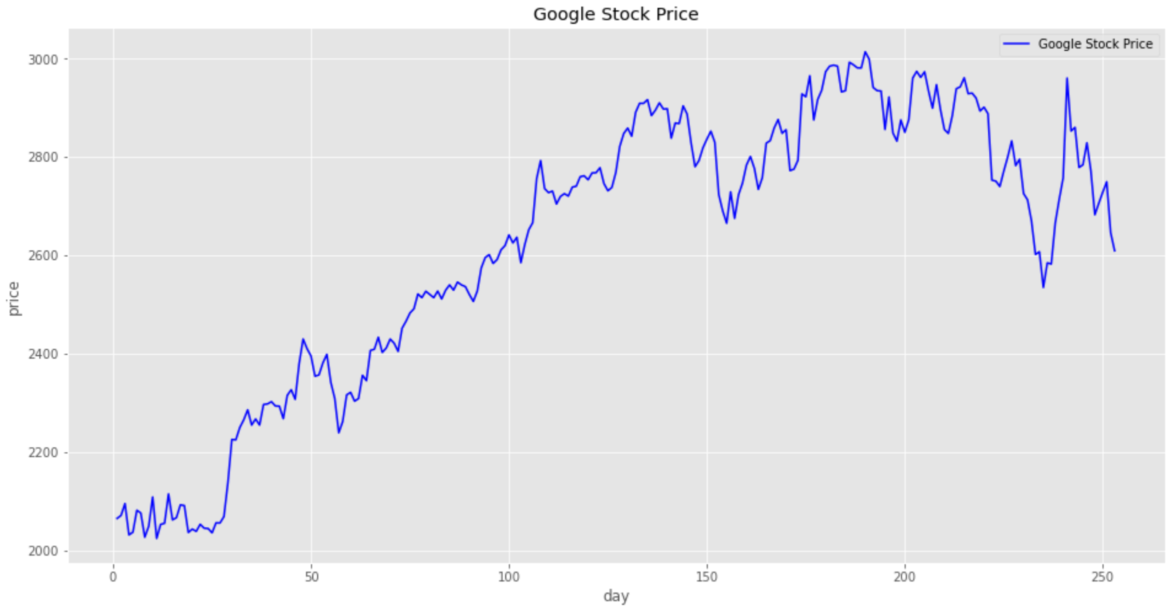


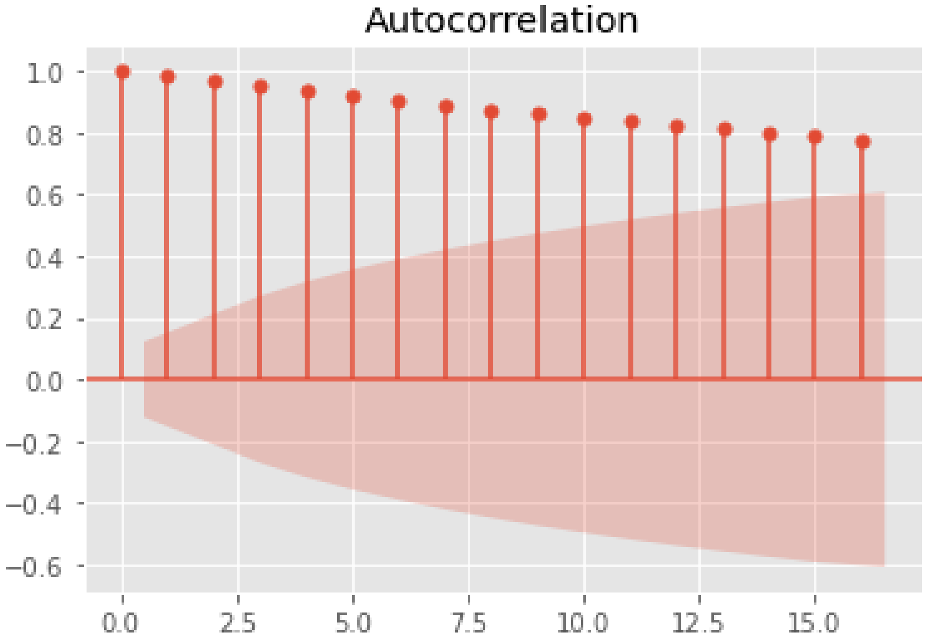
- We would like to explore the auto-correlation structure of the demand time series to construct models that can take into account the dependence explicitly.
- First, a relatively trivial observation. All basic patterns in the data (trend, seasonality) etc. reflect onto the autocorrelation structure.
- To perform any useful autocorrelation analysis, we first have to transform the data to remove the trend, seasonality etc.

Auto-Correlation: The effects of patterns

- The effect of patterns:

Daily Closing Price of Google – Alphabet Stock

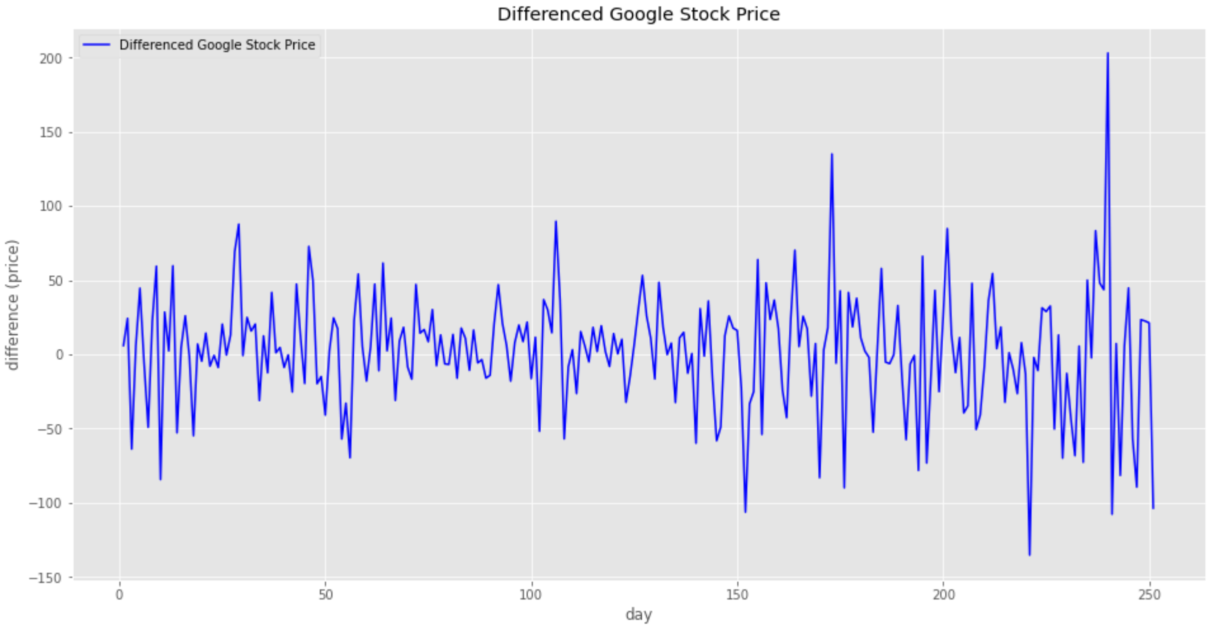


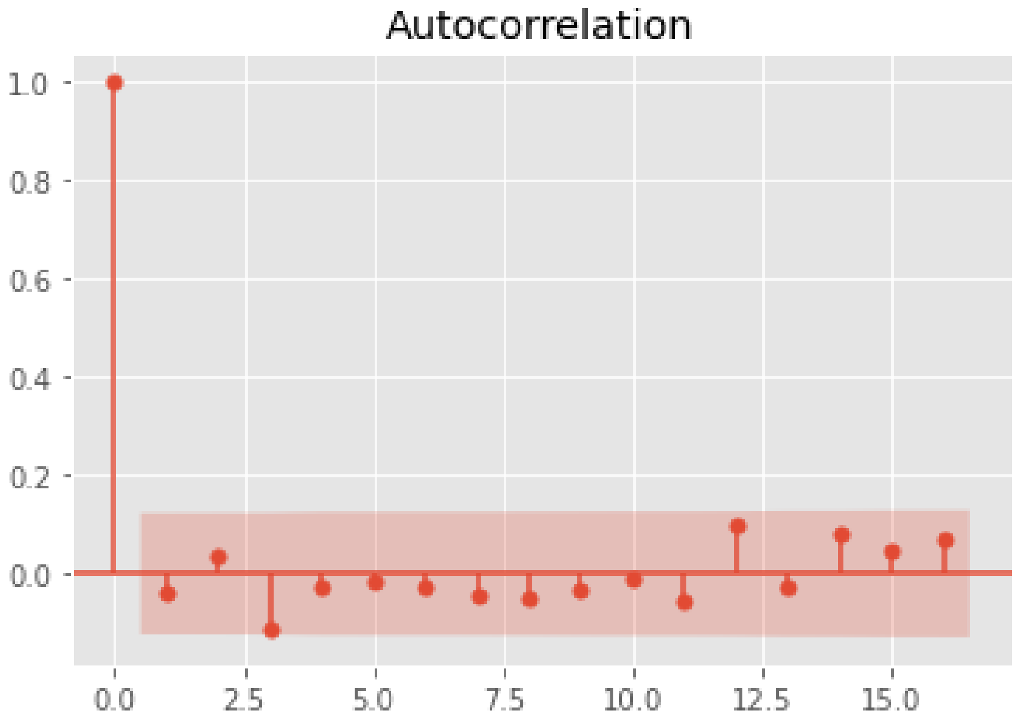


Auto-Correlation: The effects of patterns

- Let’s detrend by taking differences

Differenced Daily Closing Price of Google – Alphabet Stock



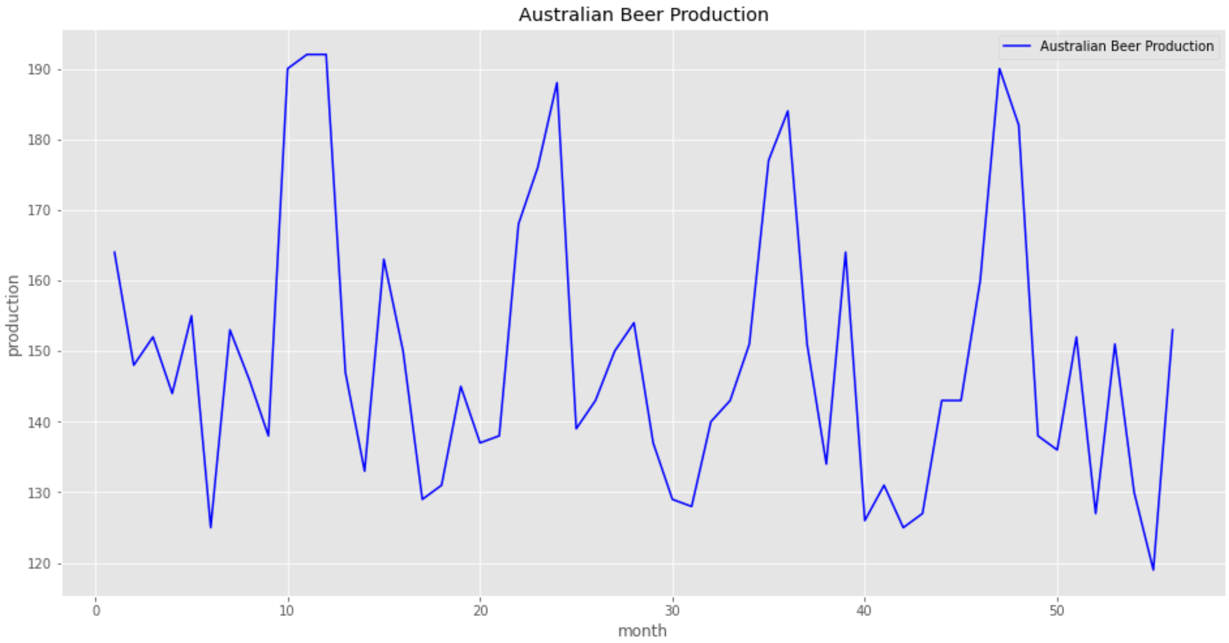


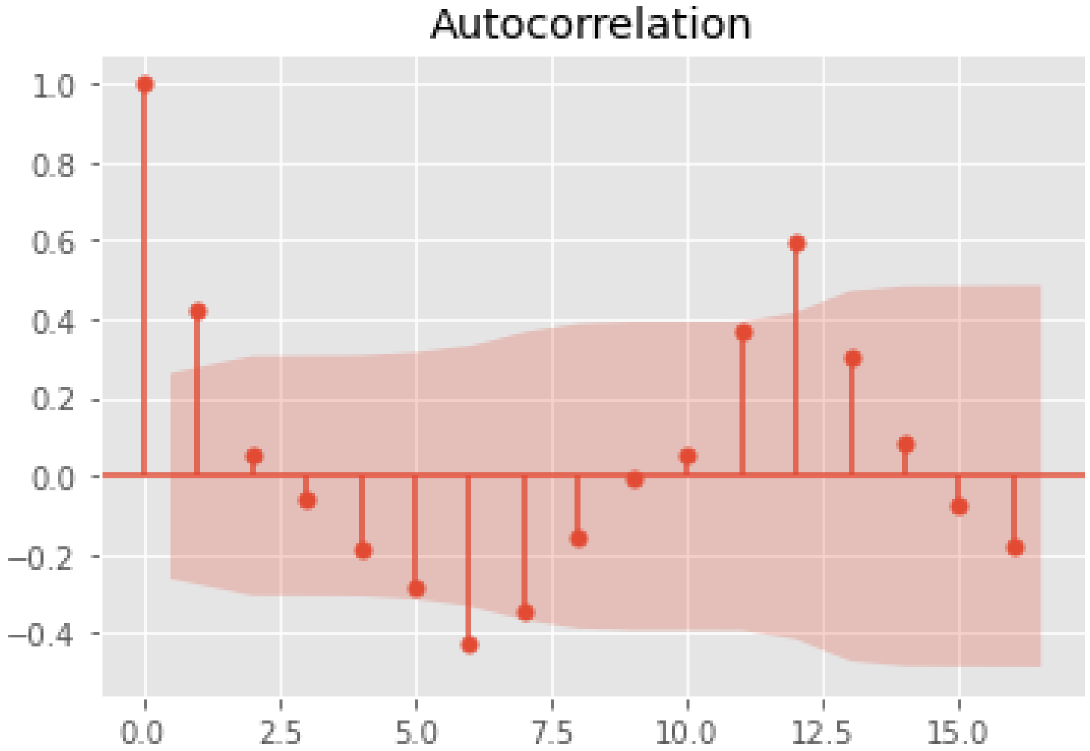
This is usually the case with time series for stock markets, after a transformation there is little AC left.

Auto-Correlation: The effects of patterns

- The effect of patterns:

Australian Monthly Beer Production

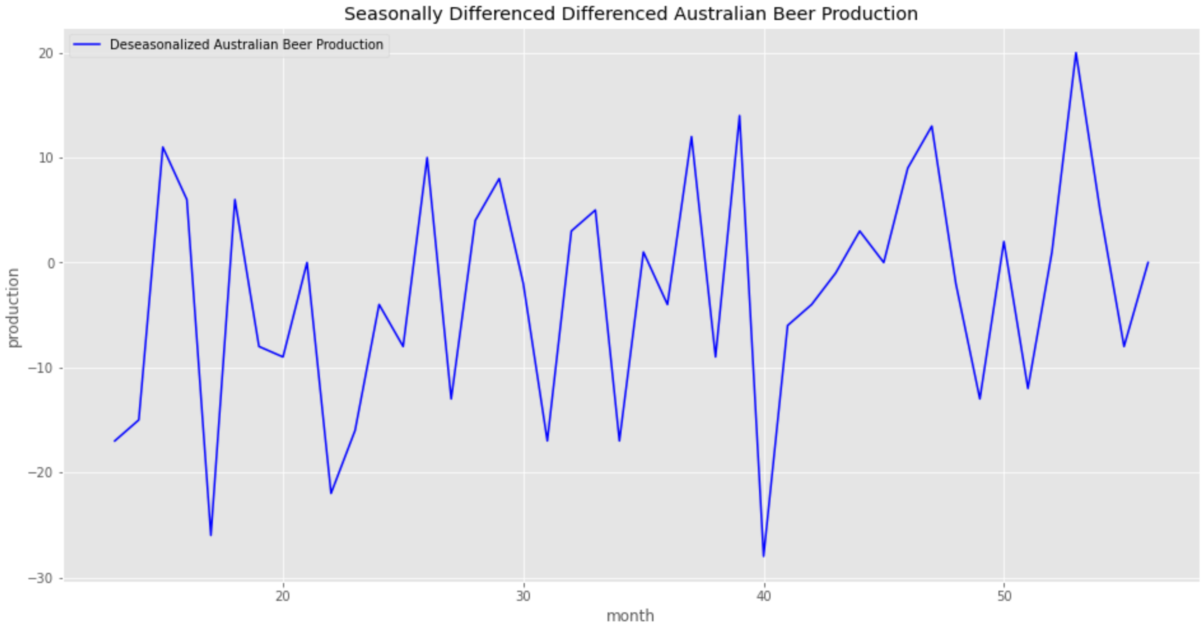


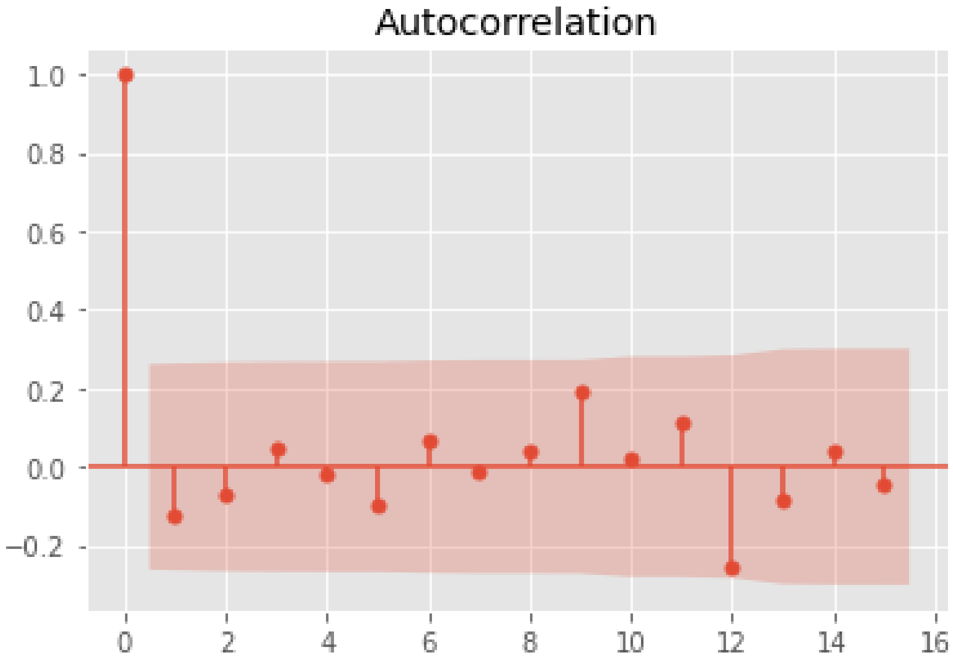


Auto-Correlation: The effects of patterns

- The effect of patterns: deseasonalized Australian Beer Production

Deseasonalized Australian Monthly Beer Production





Auto-Regressive (AR) models

- We started our modeling analysis with demand models that were in the form of $Y_t = f(t) + \epsilon_t$ (where ϵ_t are iid). Note that if we know $f(t)$ or once we figure out the functional form of its pattern from existing data, there is no remaining auto-correlation.
- We'll now consider models with a dependence structure. For instance, the AR model has the following structure:

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + \epsilon_t$$

  This is referred to as an AR-p model since it has p auto-regressive terms. Note that this is different than a typical regression because the right hand side involves terms from the same series (hence auto-regression).

Auto-Regressive models: AR(1)

- Let us consider the simplest model of this type, AR-1

$$Y_t = c + \phi_1 Y_{t-1} + \epsilon_t$$

- We can already figure out some of the basic properties. First, we have to have the AR coefficient: $-1 < \phi_1 < 1$, otherwise the series would diverge (in expectation). Note that for general AR-p processes the stability conditions for the parameters are more complicated (please see Hyndman and Athansapoulos, Chapter 8).
- If we take ϕ_1 to be positive and high (i.e. close to 1), it is clear that Y_{t-1} and Y_t are highly correlated. In fact, we can verify that $Corr(Y_{t-1}, Y_t) = \phi$.
- But due to the recursive structure, Y_{t-2} and Y_t are also correlated. In fact, we can verify that $Corr(Y_{t-2}, Y_t) = \phi_1^2$ and in general $Corr(Y_{t-k}, Y_t) = \phi_1^k$.