**D213 TIP SHEET AND COURSE NOTES**

**TASK ONE:**

**Documents on ARIMA**

<https://towardsdatascience.com/arima-simplified-b63315f27cbc>

ARIMA model is based on a number of assumptions including:

* Data does not contain anomalies
* Model parameters and error term is constant
* Historic timepoints dictate behavior of present timepoints which might not hold in stressed market data conditions
* Time series is stationary

**Duke University (All about)**

<https://people.duke.edu/~rnau/411arim.htm>

**Simple ARIMA example (how-to in R)**

<http://ucanalytics.com/blogs/step-by-step-graphic-guide-to-forecasting-through-arima-modeling-in-r-manufacturing-case-study-example/>

**Videos:**

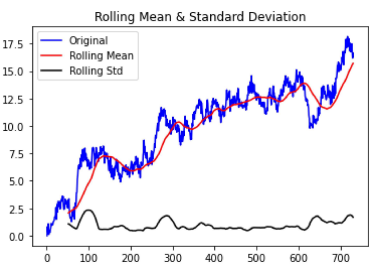
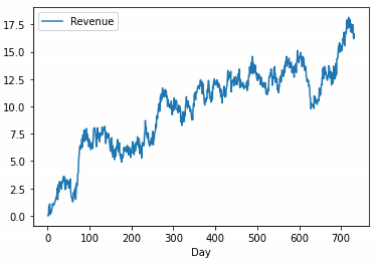
**How to use ARIMA in Python:**

<https://www.youtube.com/watch?v=o7Ux5jKEbcw>   <- 7 minute video

**Nice ARIMA in R video:**

<https://www.youtube.com/watch?v=iwRtpJDDw5M>

What you are shooting for is the revenue v. ARIMA Predictions



**TASK TWO:**

TensorFlow in 10 minutes (YouTube):

[https://youtu.be/tXVNS-V39A0](https://protect-us.mimecast.com/s/HhkECzp4LKCRnnkNPsXwhKe?domain=youtu.be)

TensorFlow Crash course (YouTube):

[https://youtu.be/6g4O5UOH304](https://protect-us.mimecast.com/s/5ZVsCjRgoqCjAABP6CRbRFF?domain=youtu.be)

**TensorFlow for Beginners (YouTube):**

[https://youtu.be/QPDsEtUK\_D4](https://protect-us.mimecast.com/s/3PvlCn5mvruGxxPy5fJyj1X?domain=youtu.be)

TensorFlow (Specifically NeuralNet YouTube):

[https://youtu.be/tPYj3fFJGjk](https://protect-us.mimecast.com/s/H8_YCjRgoqCjAABgJtRR7Jp?domain=youtu.be)

Good step-by-step reading series for TensorFlow (Includes Neural Net):

[https://www.guru99.com/tensorflow-tutorial.html](https://protect-us.mimecast.com/s/5Sm1CrkqzRtA66kRZI7S_8A?domain=guru99.com)

I recommend these steps for Task Two:

1. Analyzing the data and review length.
2. Creating a dictionary and applying it to remove extraneous characters.
3. Deciding on a typical review length.
4. Tokenization.
5. Applying Tensorflow’s keras and layers methods
6. Splitting the dataset into train and test
7. Fitting the model
8. Model evaluation and reporting.

Yes, scipy has the ability to create a nice periodogram: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.periodogram.html>

And so does matploblib: <https://scipy-lectures.org/intro/scipy/auto_examples/plot_spectrogram.html>

A math example of calculating spectral density: <https://www.youtube.com/watch?v=Wio7OKc4-uE>

A theoretical video of psd is: <https://www.youtube.com/watch?v=-Nt0FaofWL4>

Remember, psd is related to Fourier Transform of autocorrelation, so this video should help: <https://www.youtube.com/watch?v=s2K1JfNR7Sc>

Vaishali explains a little bit more but is somewhat incomplete on the theory as the video comments state:  <https://www.youtube.com/watch?v=Cg0_k1Dy0b4>

However, if you do a stationary test (Dickey-Fuller) with rolling mean and std. dev.  You should be pretty good-to-go.  You can run your autocorrelation and set your lags using statsmodel and importing pacf.  Then create your periodogram using scipy on your training set.  Using ARIMA from statsmodels, you can do your model fit and it should yield your regression table with p-values and coefficients.  Run your plot with ARIMA v. Revenue predictions and you are done!

In the LSTM model from Keras library, you set the max\_features, embedding\_dim, sequence\_Length, and the number of nodes = to max\_features + 1.

The embedding\_dim if set to 16 will convert each word to a 16 bit vector.  If 32, each word will be converted to a 32 bit vector.

Example:

max\_features = 50000

embedding\_dim =16

sequence\_length = 70

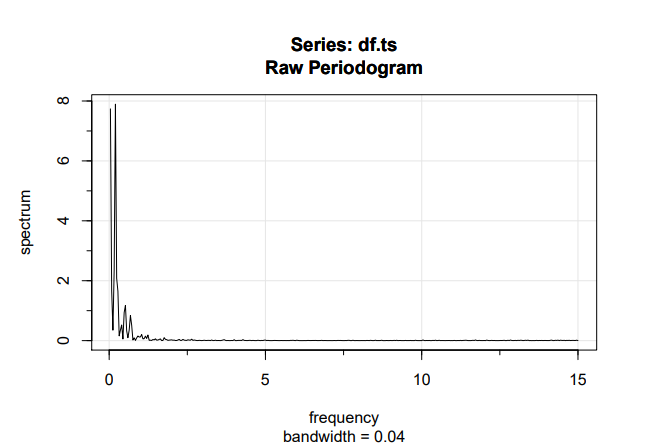
model = tf.keras.Sequential()

model.add(tf.keras.layers.Embedding(max\_features +1, embedding\_dim,

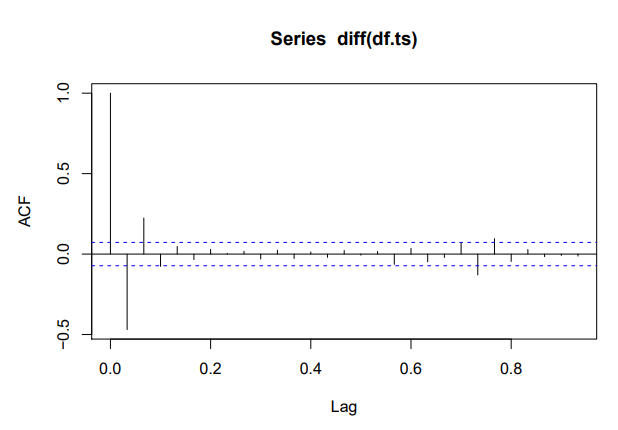
input\_length=sequence\_length))

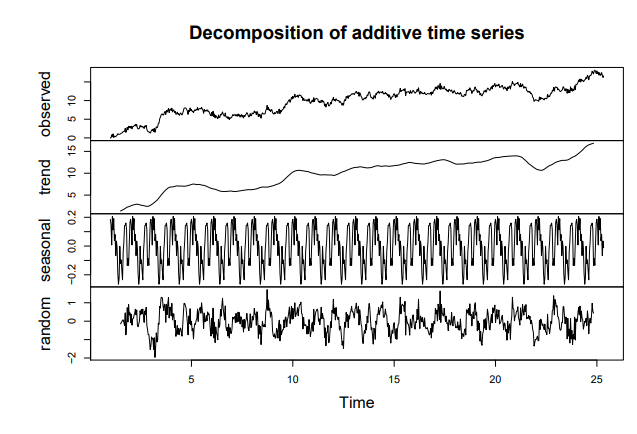
model.add(tf.keras.layers.Dropout(0.5))

Usually, it is done by max\_df\_sentence\_length = df['Num\_words\_text'].max()



Acf(diff(df.ts))





For Task Two: Actually, it is pretty simple.  There are three text files, Amazon, IMDB, and Yelp.  You want to do a neural net to find the Sentiment index among the reviews.  These are the high-level ten basic steps:

1. If you run a couple of FOR loops to translate characters (essentially “cleaning up”).   There will be 52 unique characters remaining.
2. Plot a histogram to find the length of the reviews. Most of the 500 reviews will be under 150 characters in length.
3. Create a dictionary for characters and tokens.
4. Use arrays to hold each review and assign to it a list of one-hot-encoded characters.
5. Code the output array to represent the review as positive (1) or negative (0).  The target value is binary.
6. Add terminators to the end of the data to fill the fixed length strings.
7. Using sklearn, numpy, and keras train a neural net.  Split data prior to training into test and validation.
8. By reading through the data in the network, weights are adjusted until errors are minimized.  Each period or cycle is referred to as an epoch.
9. Then, you run a sequential neural net using keras.Sequential()
10. After using the training process iteratively, the model.evaluate(x, y) will confirm the reliability.

You may have to downgrade Python environment to python 3.6. There is an issue with python 3.8 and TensorFlow.

An option is to install it for Python 3.7 which is supported for Tensorflow 2.0. If you have anaconda navigator, you can create an environment with Python 3.7 specifically for installing tensorflow. Open your terminal in this environment and then type:

conda install tensorflow

[How to install TensorFlow with Python 3.8 - Stack Overflow](https://stackoverflow.com/questions/59809495/how-to-install-tensorflow-with-python-3-8)

**System requirements**

* Python 3.6–3.9
  + Python 3.9 support requires TensorFlow 2.5 or later.
  + Python 3.8 support requires TensorFlow 2.2 or later.
* pip 19.0 or later

[Install TensorFlow with pip](https://www.tensorflow.org/install/pip)

Here is how to check your pip version:

## 1. Install the Python development environment on your system

Check if your Python environment is already configured:

Requires Python 3.6–3.9, and pip >= 19.0

python3 --version  
pip3 --version