Module 2 – Part I

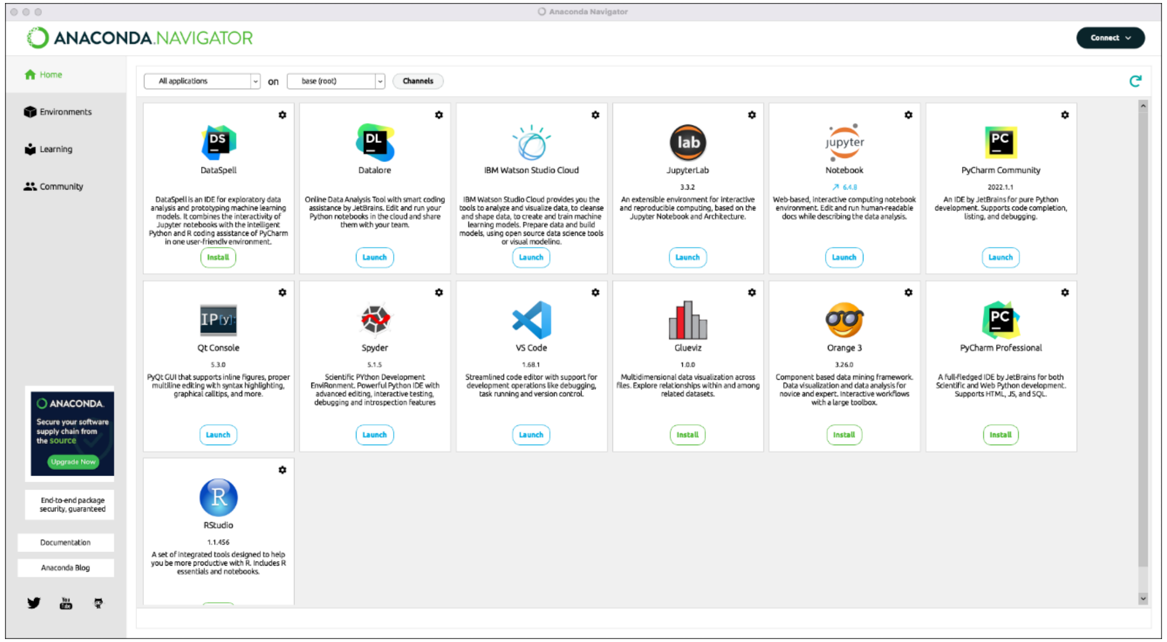
Setting up Deep Forecasting Environment (Python)

Road map!

- Module 1- Introduction to Deep Forecasting
- **Module 2- Setting up Deep Forecasting Environment (Python)**
- Module 3- Exponential Smoothing
- Module 4- ARIMA models
- Module 5- Machine Learning for Time series Forecasting
- Module 6- Deep Neural Networks
- Module 7- Deep Sequence Modeling (RNN, LSTM)
- Module 8- Transformers (Attention is all you need!)
- Module 9- Prophet and Neural Prophet

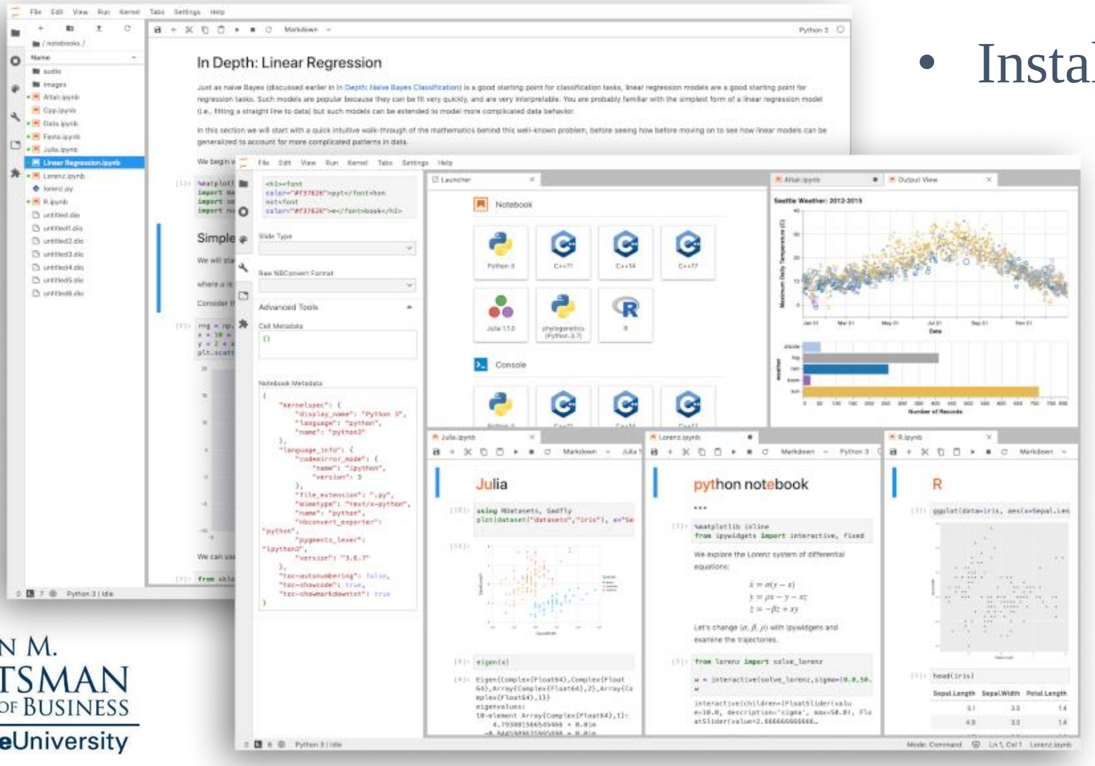
Install through ANACONDA

- Anaconda is a distribution of the Python and R programming languages for scientific computing, that aims to simplify package management with conda environments.
- Anaconda offers the easiest way to perform data science and machine learning on a single machine.
- Install Anaconda @ https://www.anaconda.com/



JupyterLab

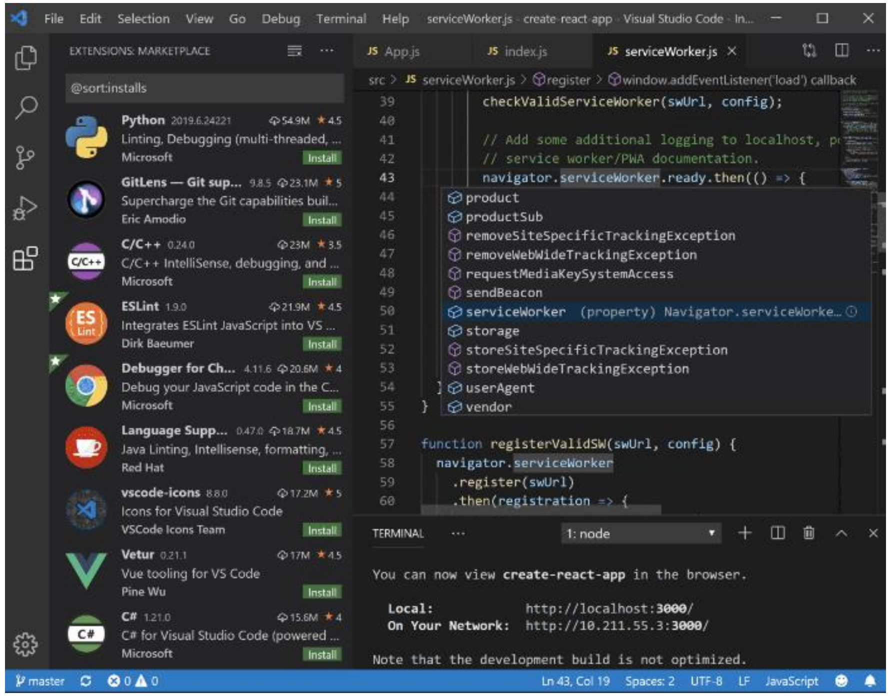
- <u>JupyterLab</u> is the latest web-based interactive development environment for notebooks, code, and data
- Jupyter's name is a reference to the three core programming languages supported by Jupyter, which are Julia, Python and R



- Install JupyterLab @ http://jupyter.org/install

VS Code

- VS Code is one of the most popular source code editors
- Features include support for debugging, syntax highlighting, intelligent code completion, code refactoring, and embedded Git.
- Install VS code @ https://code.visualstudio.com/

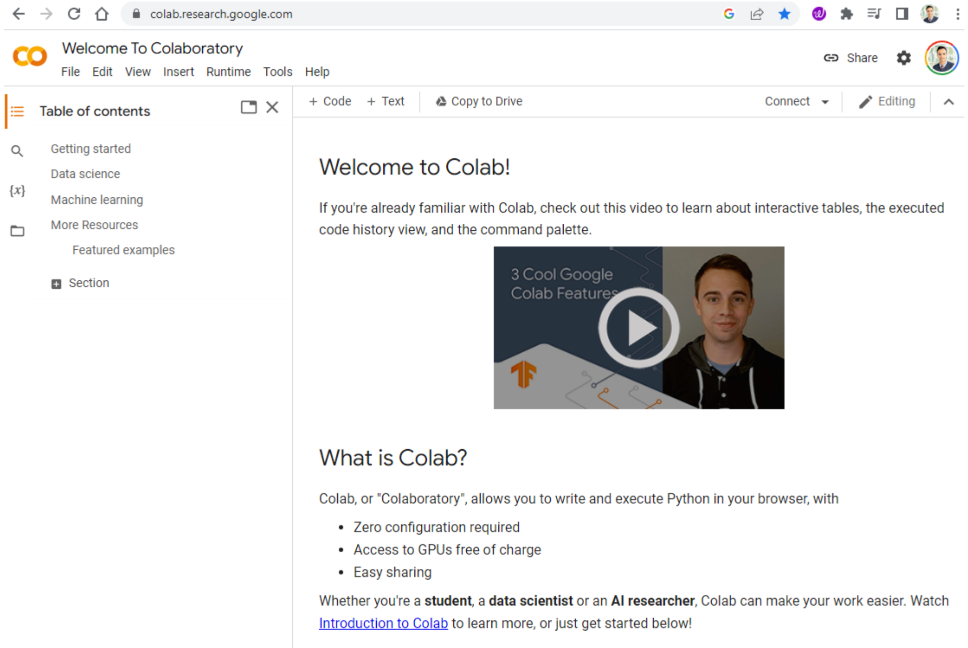


JON M. HUNTSMAN SCHOOL OF BUSINESS
UtahStateUniversity

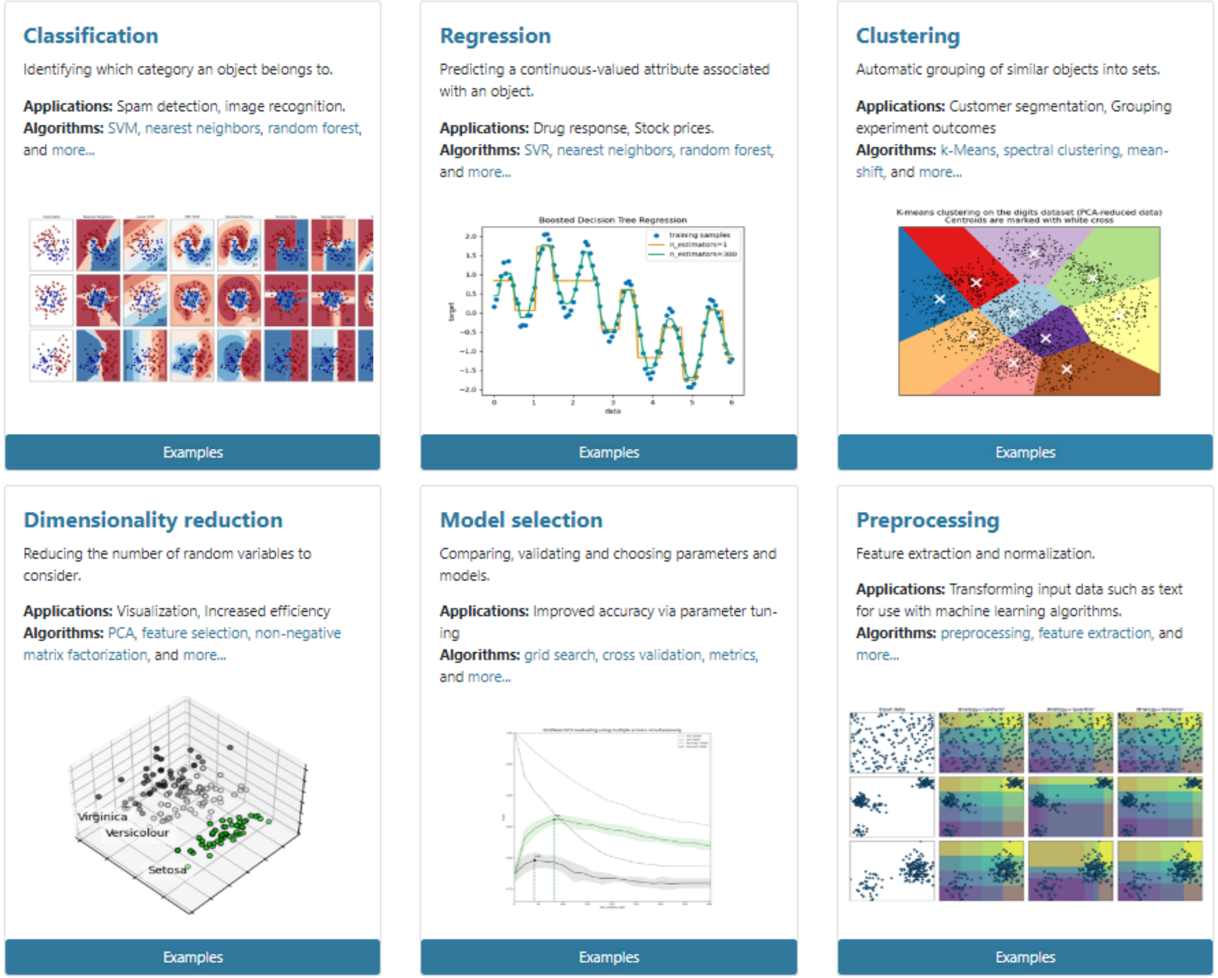
Pedram Jahangiry

Google colab

- Colab is a free hosted Jupyter notebook-style environment that runs entirely in the cloud and requires no setup to use. It also provides access to machine learning libraries and computing resources including GPU.
- Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. https://colab.research.google.com/



scikit learn



- Scikit-learn is an open-sourced Python library and includes a variety of unsupervised and supervised learning techniques.
- It is based on technologies and libraries like Matplotlib, Pandas and NumPy and helps simplify the coding task.
- Install Scikit-learn @ https://scikit-learn.org/stable/install.html

PYCARET

- PyCaret is an open-source, low-code machine learning library in Python that automates machine learning workflows.
- PyCaret is essentially a Python wrapper around several machine learning libraries and frameworks
- Install PyCaret @ https://pycaret.gitbook.io/docs/get-started/installation

```python
# load dataset
import pandas as pd
train = pd.read_csv('train.csv')
test  = pd.read_csv('test.csv')

# init setup
from pycaret.classification import *
s = setup(train, target= 'target')

# model training and selection
best = compare_models()

# analyze best model
evaluate_model(best)


# predict on new data
predictions = predict_model(best, data =test )

# save best pipeline
save_model(best, 'my_best_pipeline')
```

ID	Name
'lr'	Logistic Regression
'knn'	K Nearest Neighbour
'nb'	Naives Bayes
'dt'	Decision Tree Classifier
'svm'	SVM – Linear Kernel
'rbfsvm'	SVM – Radial Kernel
'gpc'	Gaussian Process Classifier
'mlp'	Multi Level Perceptron
'ridge'	Ridge Classifier
'rf'	Random Forest Classifier
'qda'	Quadratic Discriminant Analysis
'ada'	Ada Boost Classifier
'gbc'	Gradient Boosting Classifier
'lda'	Linear Discriminant Analysis
'et'	Extra Trees Classifier
'xgboost'	Extreme Gradient Boosting
'lightgbm'	Light Gradient Boosting
'catboost'	CatBoost Classifier

Keras

- Keras is a high-level, open-source neural network library written in Python. It was developed to make it easier for researchers and developers to build and experiment with deep learning models.
- The Keras API became the official high-level API for TensorFlow 2.0 in 2019. https://keras.io/

```
model = keras.Sequential(
    [
        keras.Input(shape=input_shape),
        layers.Conv2D(32, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Flatten(),
        layers.Dropout(0.5),
        layers.Dense(num_classes, activation="softmax"),
    ]
)

model.summary()
```

Jon M. Huntsman School of Business
Utah State University

Pedram Jahangiry

TensorFlow

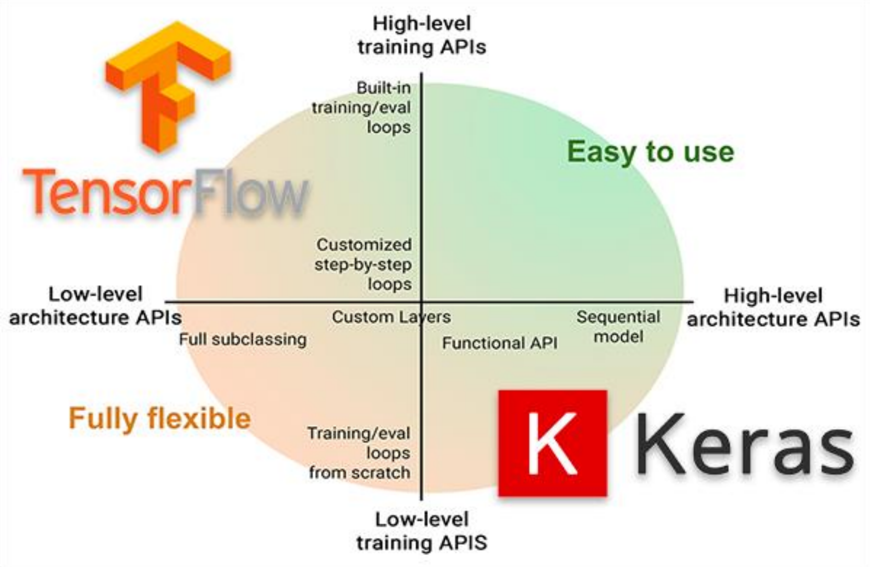
- TensorFlow is a Google-maintained open-source end-to-end platform for prototyping and assessing machine learning models, primarily neural networks.
- TensorFlow also offers TensorBoard, a visualization tool for comparing and tracking our learned models.
- It can scale from a single CPU, to a GPU or cluster of GPUs all the way up to a multi-node TPU infrastructure.
- Build-in Google Colab. For local installation visit https://www.tensorflow.org/install

Jon M. Huntsman School of Business
Utah State University

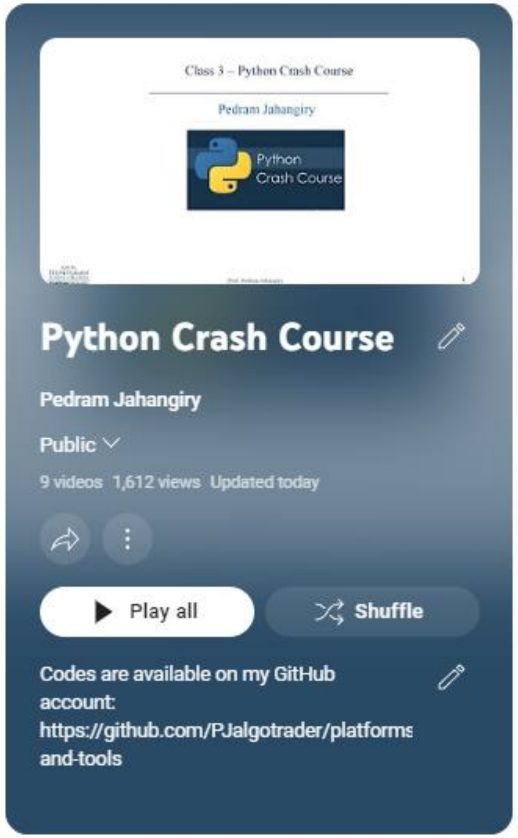
Pedram Jahangiry

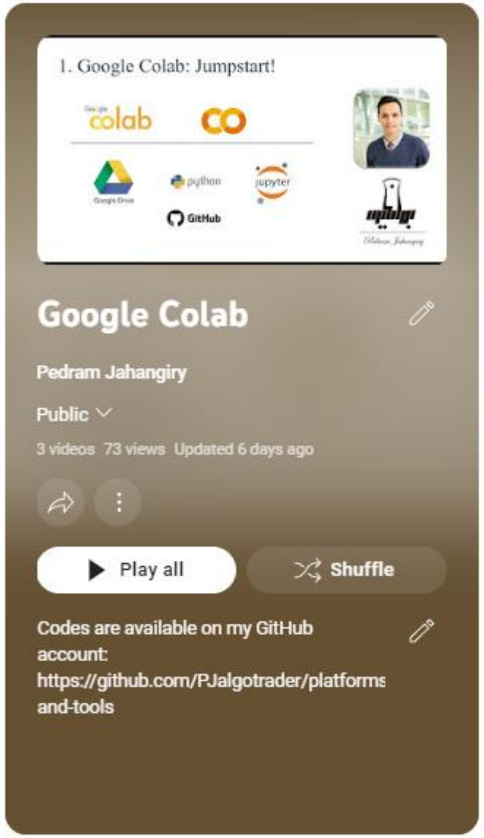
Keras Vs TensorFlow

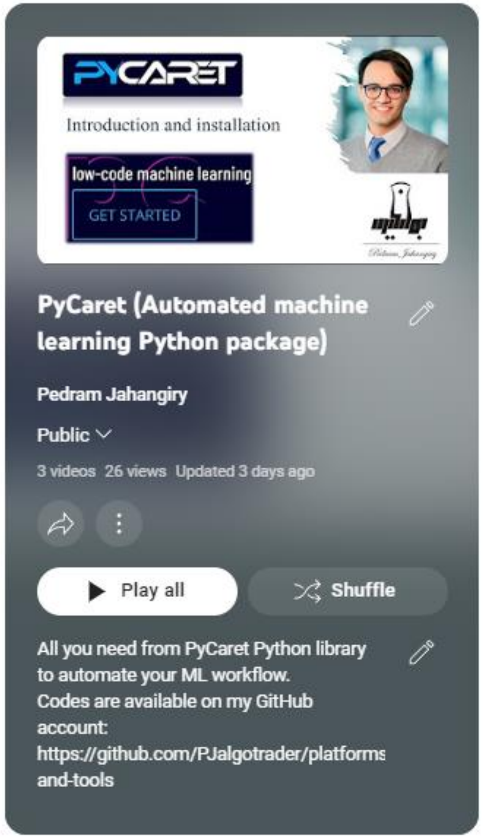
- Level of abstraction: Keras is a higher-level library that provides a more intuitive interface for building and training models, while TensorFlow is a lower-level library that provides more flexibility but requires the user to specify more details of the model.
- Keras is a standalone library, while TensorFlow includes both a low-level library for numerical computations and a high-level library for building and training machine learning models.
- Keras is a user-friendly interface to TensorFlow



Available YouTube playlists







Jon M. Huntsman School of Business
Utah State University

Pedram Jahangiry

Platforms and Packages

Listed below are some Python packages and platforms that will be used in the deep learning and deep forecasting courses. Google colab

General Python libraries

- NumPy
- pandas
- matplotlib
- seaborn

Machine Learning libraries

- scikit learn
- dmlc XGBoost
- LightGBM
- PYCARET
- SKTIME

Deep Learning libraries

- Keras
- TensorFlow
- HUGGING FACE

Setting up Deep Learning Environment

	Personal Workstation	Cloud Platforms	Google Colaboratory
Pros	• Full control over hardware and software • Work offline • Fixed cost	• Powerful computing resources • Scalability • Ease of use • Cost-effective: Pay-as-you-go • Collaboration	• Powerful computing resources (GPU, TPU) • Ease of use • Collaboration • No need to set up a local environment
Cons	• Scalability • Maintenance (both hardware and software)	• Expensive for large-scale experiments • Dependency on the provider • Limited control • Internet connection • Security	• Time limit • Hardware limitation • Data storage • Limited control • Internet connection • Security

Jon M. Huntsman School of Business
Utah State University

Pedram Jahangiry

The modern machine learning landscape

- From 2016 to 2020, the entire machine learning and data science industry has been dominated by these two approaches:
  1. Deep learning
  2. Gradient boosted trees

- Most practitioners of deep learning use Keras, often in combination with its parent framework TensorFlow.
- This means you'll need to be familiar with Scikit-learn, XGBoost, and Keras

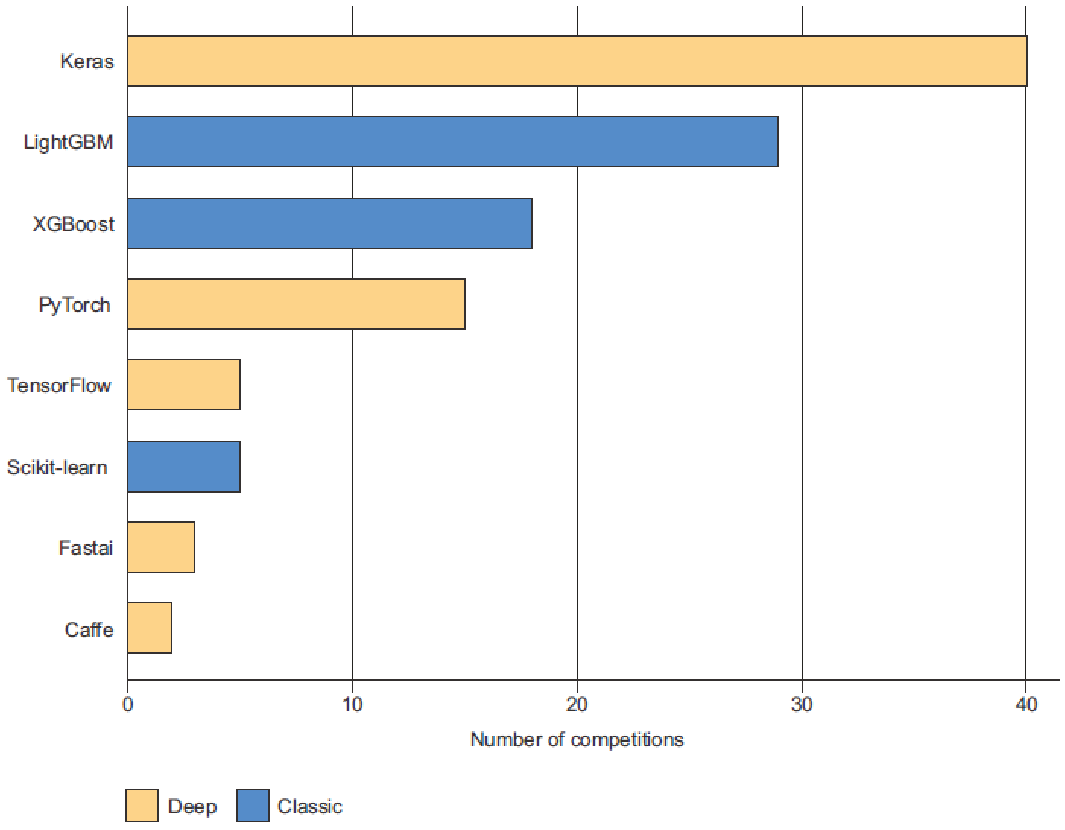


Figure 1.12 Machine learning tools used by top teams on Kaggle

The modern machine learning landscape

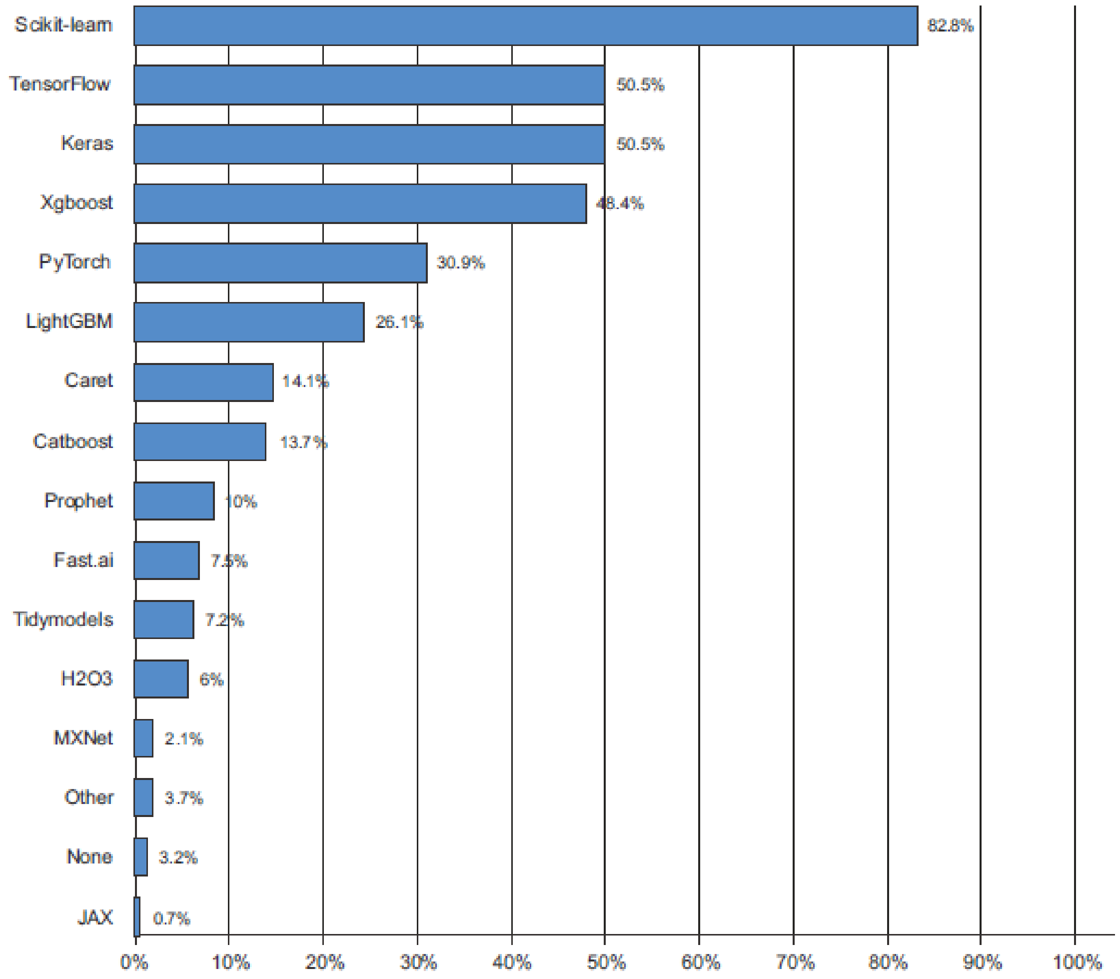


Figure 1.13 Tool usage across the machine learning and data science industry (Source: www.kaggle.com/kaggle-survey-2020)

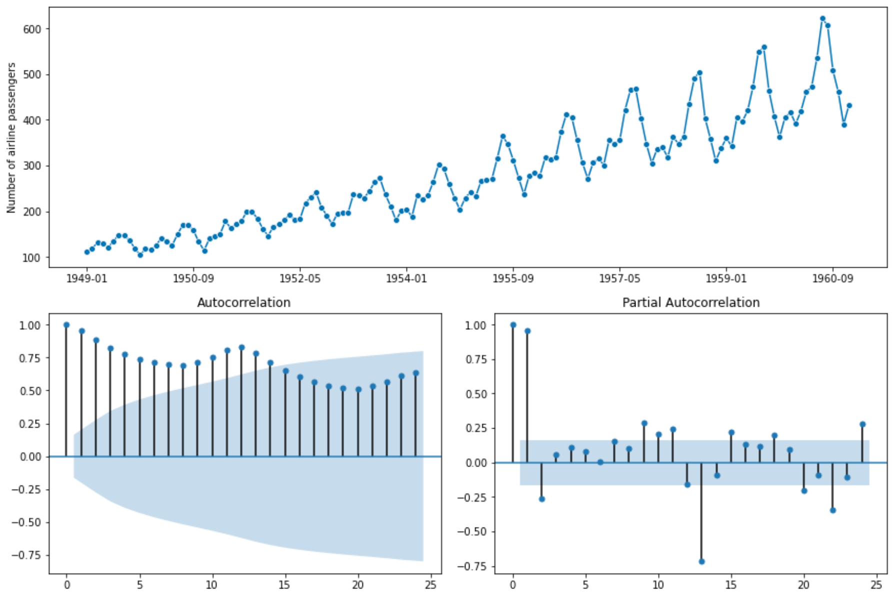
- Kaggle also runs a yearly survey among machine learning and data science professionals worldwide.
- This survey is one of our most reliable sources about the state of the industry!!!
- This figure shows the percentage of usage of different machine learning software frameworks.

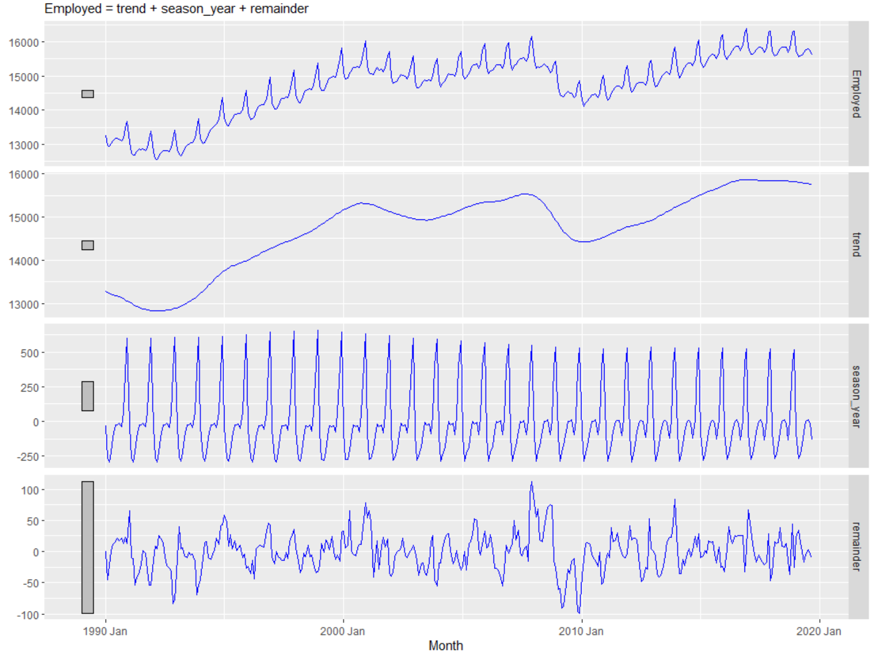
Pedram Jahangiry

Module 2 – Part II

Setting up Deep Forecasting Environment

(Time Series Basics)





JON M. HUNTSMAN SCHOOL OF BUSINESS
UtahStateUniversity

Pedram Jahangiry

Road map!

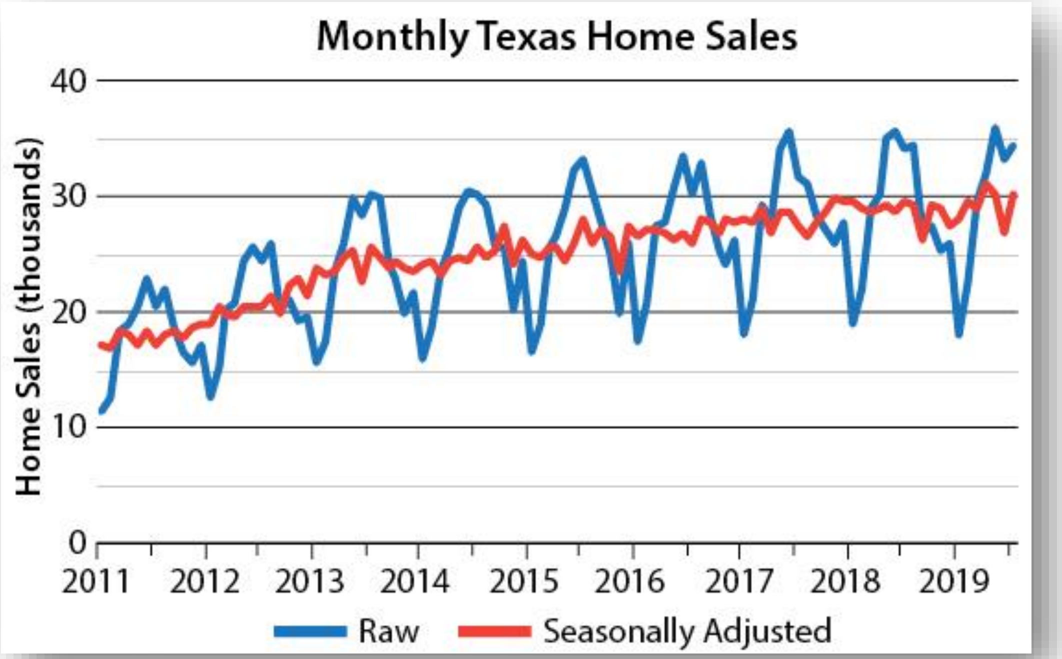
- Module 1- Introduction to Deep Forecasting
- Module 2- Setting up Deep Forecasting Environment (Time Series Basics)
- Module 3- Exponential Smoothing
- Module 4- ARIMA models
- Module 5- Machine Learning for Time series Forecasting
- Module 6- Deep Neural Networks
- Module 7- Deep Sequence Modeling (RNN, LSTM)
- Module 8- Transformers (Attention is all you need!)
- Module 9- Prophet and Neural Prophet

Time Series Patterns

- Trend: A trend is a long-term movement in the data, either upwards or downwards. A trend can be linear, meaning that it follows a straight line, or it can be nonlinear, meaning that it follows a more complex curve.

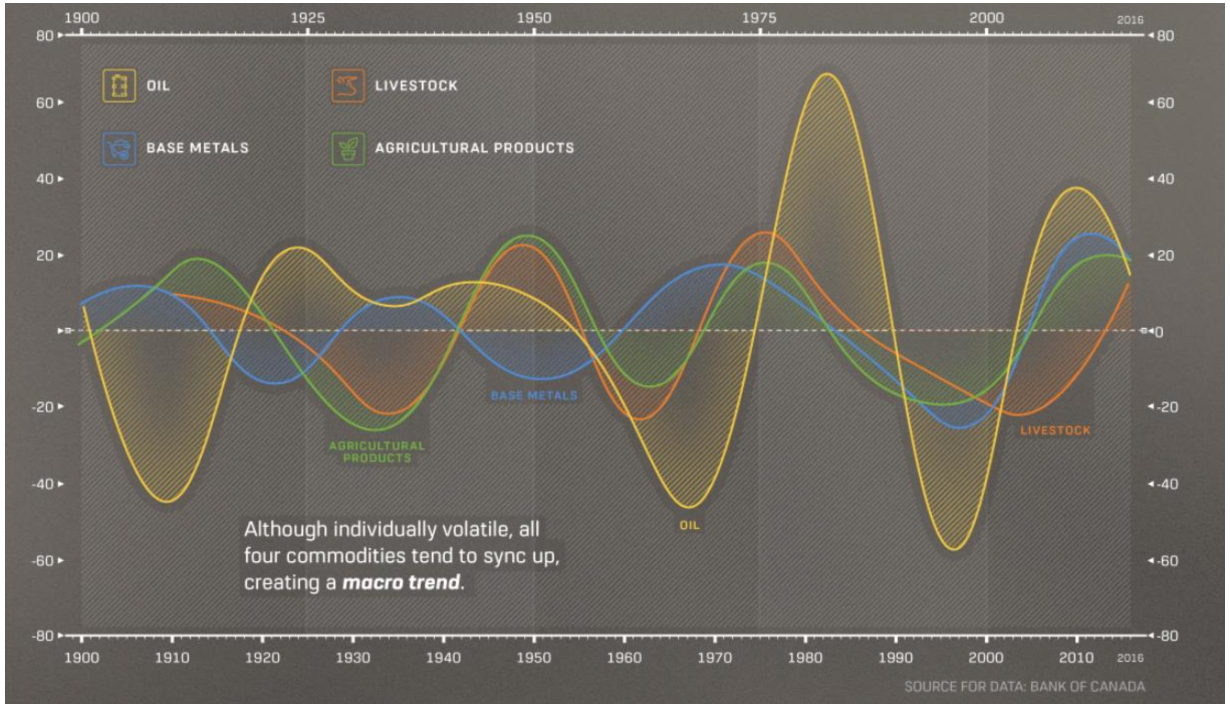
Time Series Patterns

- Seasonal: A seasonal pattern is a regular fluctuation in the data that occurs at a specific time of the year, month or week. Seasonality is always of a fixed and known period.



Time Series Patterns

- Cyclic: There is a cycle when the data exhibits peaks and valleys that do not occur at a fixed frequency. Economic conditions and the "business cycle" are typically responsible for these fluctuations.
- In general, the average length of cycles is longer than the length of a seasonal pattern, and the magnitudes of cycles tend to be more variable than the magnitudes of seasonal patterns.



Lag variables

- Lag variables are variables that are measured at a previous point in time relative to the current time period being studied.
- Lag variables can be useful for identifying trends, patterns, and relationships in time series data and how they change over time.

Date	Value	$Value_{t-1}$	$Value_{t-2}$
1/1/2017	200	NA	NA
1/2/2017	220	200	NA
1/3/2017	215	220	200
1/4/2017	230	215	220
1/5/2017	235	230	215
1/6/2017	225	235	230
1/7/2017	220	225	235
1/8/2017	225	220	225
1/9/2017	240	225	220
1/10/2017	245	240	225

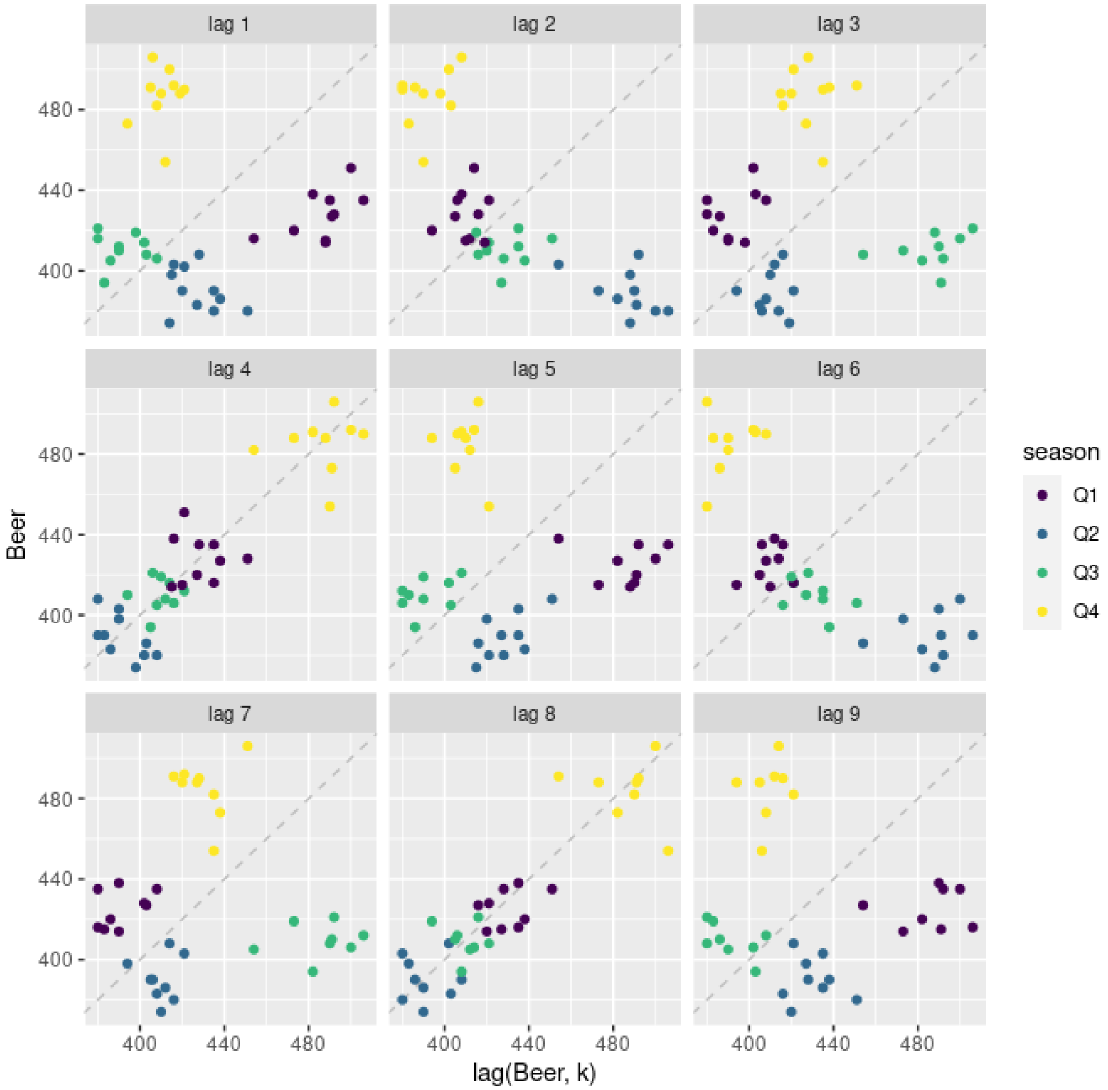
Jon M. Huntsman School of Business
UtahStateUniversity

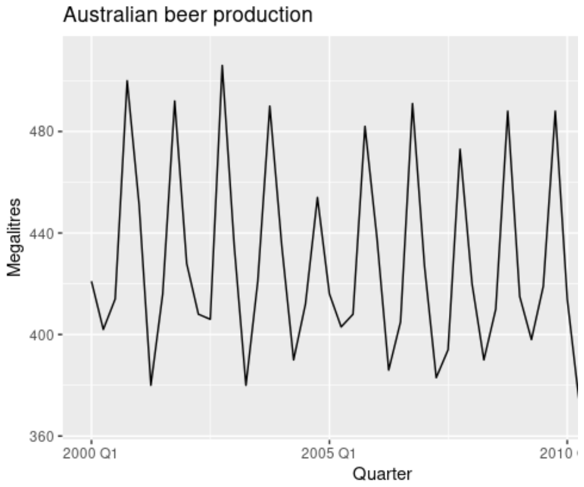
Pedram Jahangiry

Lag plots

- y_t plotted against $y_{(t-k)}$
- The colors indicate the quarter of the beer production
- The relationship is strongly positive at lags 4 and 8, reflecting the strong seasonality in the data.

Lagged scatterplots for quarterly beer production





JON M. HUNTSMAN SCHOOL OF BUSINESS
UtahStateUniversity

Pedram Jahangiry

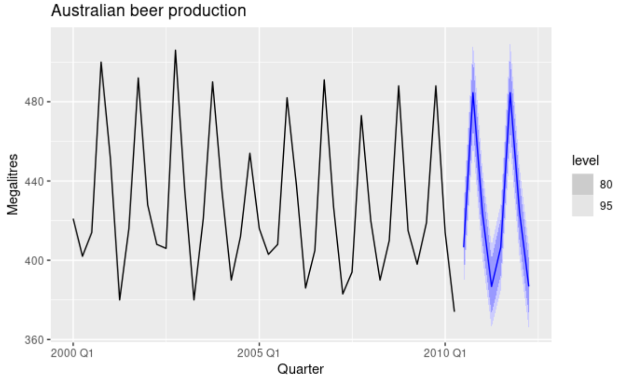
Autocorrelation

- Autocorrelation, also known as serial correlation, is a measure of the correlation between a time series and a lagged version of itself.
- It is used to assess the degree to which the past values of a time series are predictive of its future values.

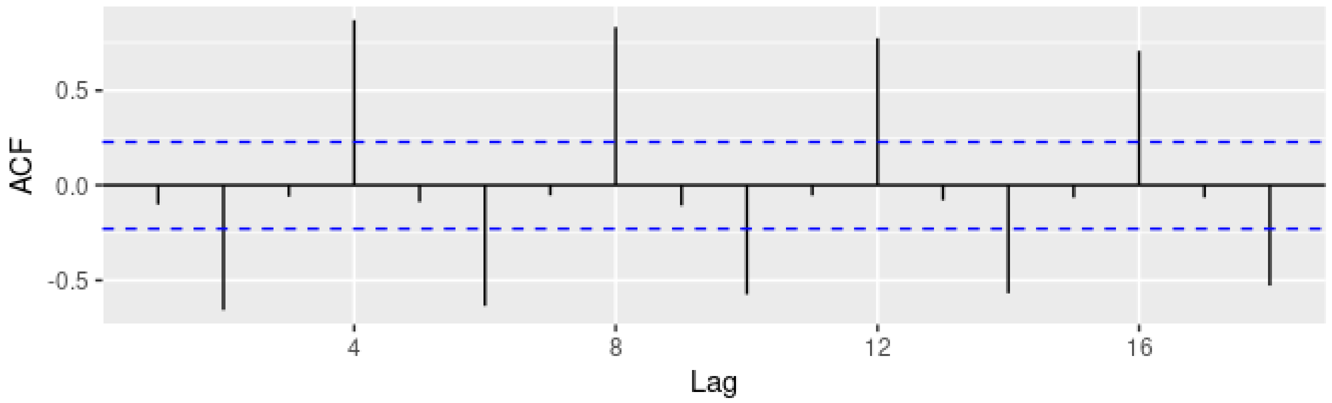
$$r_k = \frac{\sum_{t=k+1}^{T} (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{T} (y_t - \bar{y})^2}$$

ACF: Autocorrelation Function

- The autocorrelation function (ACF) is a statistical tool that can be used to measure the autocorrelation of a time series.
- It calculates the correlation between the time series and lagged versions of itself at different lag periods.



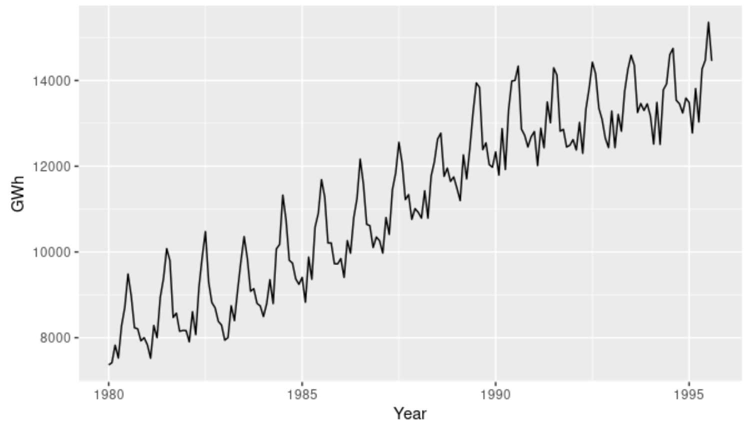
r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9
-0.102	-0.657	-0.060	0.869	-0.089	-0.635	-0.054	0.832	-0.108

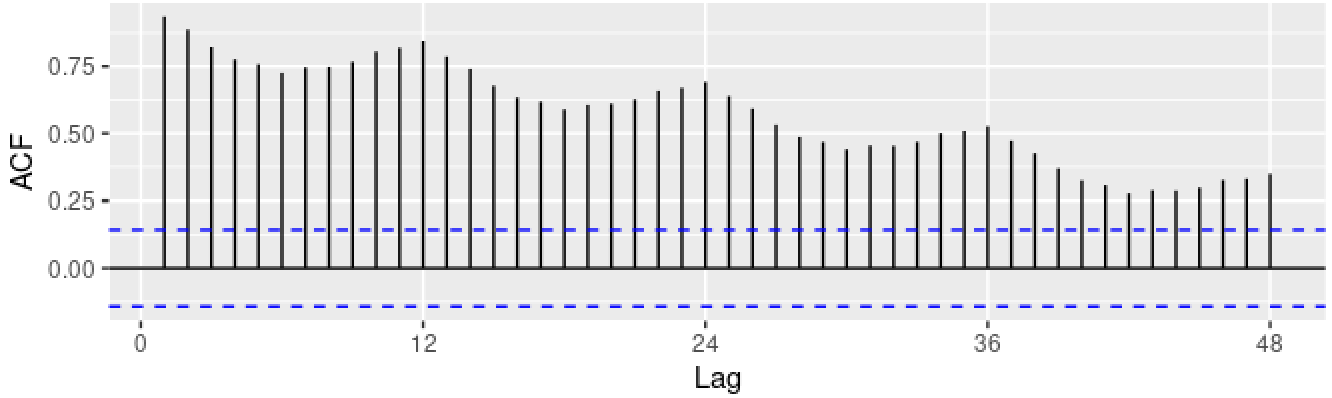


Trend and seasonality in ACF plots

- Autocorrelation can be useful for identifying patterns and trends in time series data.
- The ACF of trended time series tend to have positive values that slowly decrease as the lags increase.
- Fore seasonal data, the autocorrelations are larger for the seasonal lags than for other lags.

Monthly Electricity Demand

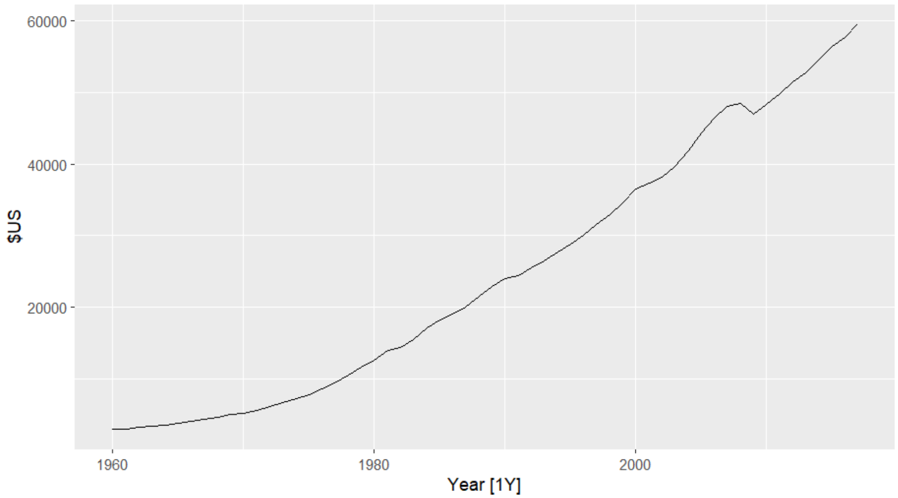




Partial Autocorrelation

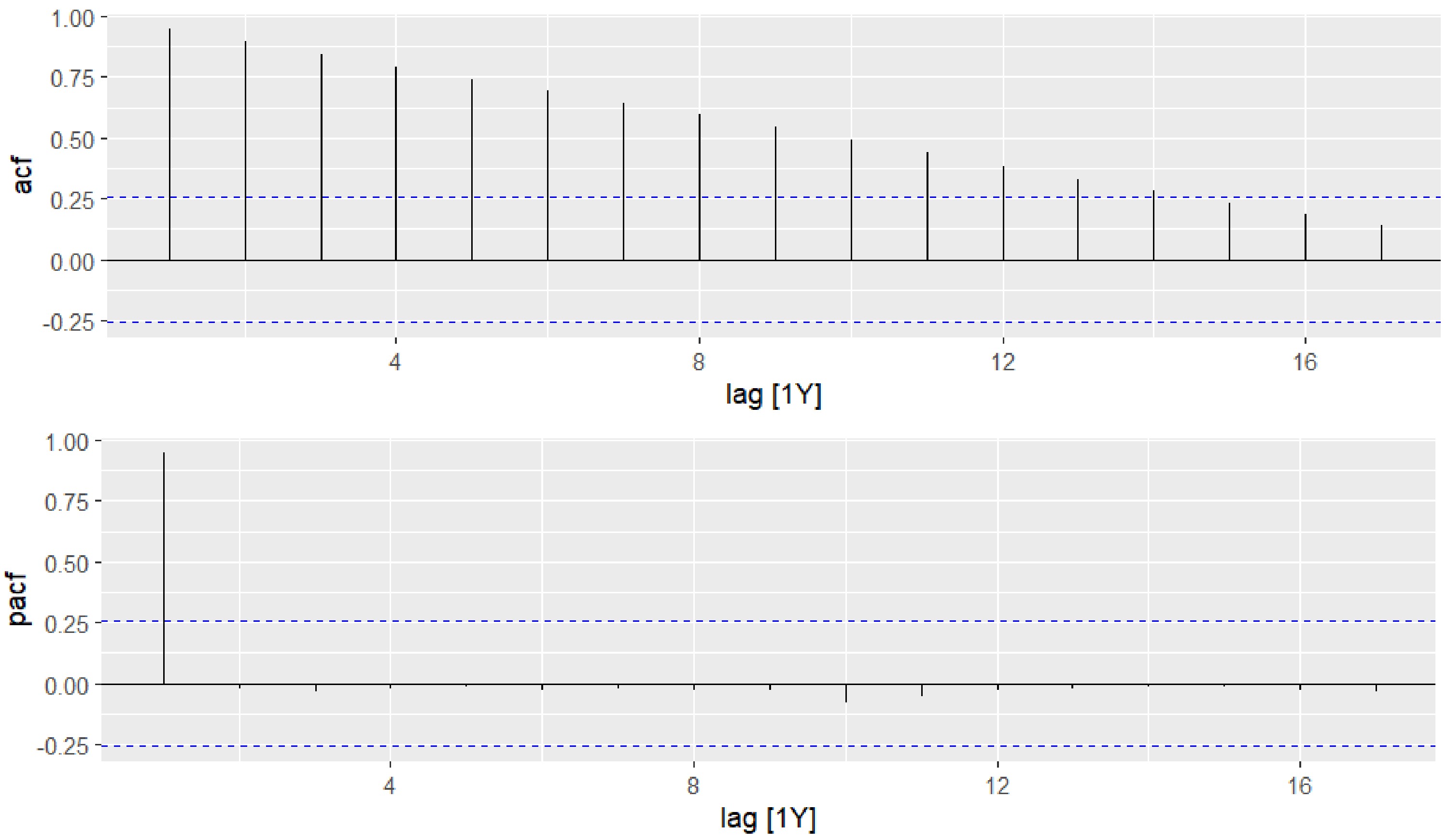
- Partial autocorrelation, also known as partial serial correlation, is a measure of the correlation between a time series and a lagged version of itself, controlling for the effects of intermediate lag periods.
- y_t and y_{t-2} might be correlated, simply because they are both connected to y_{t-1}, rather than because of any new information contained in y_{t-2}. Partial autocorrelation overcomes this problem.

US Annual GDP per capita (1960-2017)



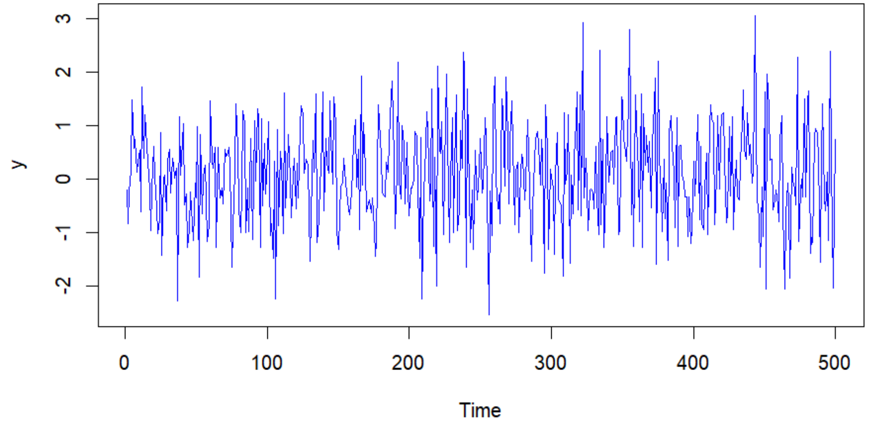
PACF: Partial Autocorrelation Function

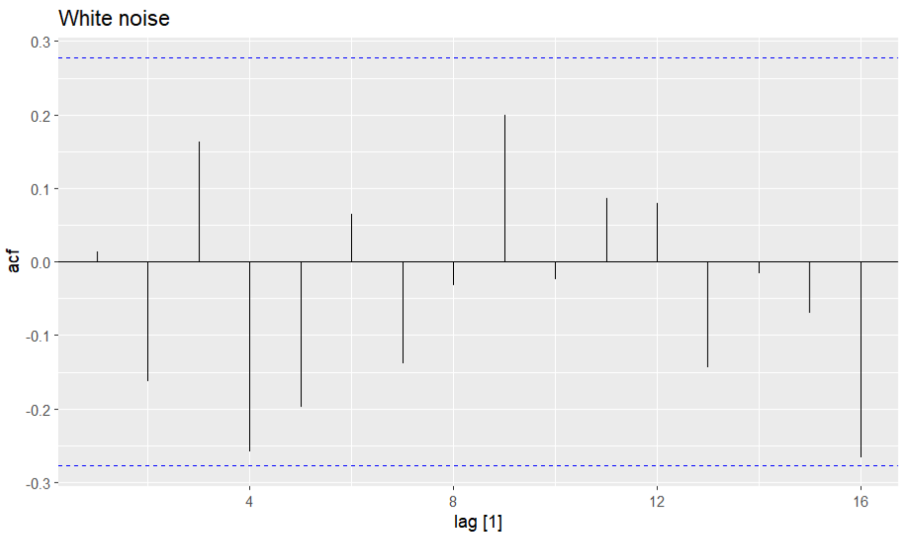
- PACF is a statistical tool that can be used to measure the partial autocorrelation of a time series.



White Noise

- White noise can be thought of as a random sequence of iid values (independent and identically distributed) characterized by a distribution.
- By comparing the characteristics of the time series data to the characteristics of white noise, we can determine whether there are any significant patterns or trends present in the data.
- White noise data show no autocorrelation.





Transformations

- The purpose of transformations is to simplify the patterns in the historical data by making the pattern more consistent across the whole data set.
- Mathematical Transformations:
  - Logarithmic
  - Power
  - Box-Cox

- Note that these transformations should be used with caution, as they can significantly alter the interpretation of the data.
- It's generally a good idea to try multiple transformations and compare the results to determine which is most appropriate for your data.

Transformations: Log vs Power vs Box-Cox

Method	Description	Formula
Logarithmic	• Transforms data by taking the logarithm of the values. • Log transformation is very interpretable!	$w_t = \log(y_t)$
Power	• Transforms data by raising it to a power. (square, cube, …) • Not so interpretable.	$w_t = y_t^p$
Box-Cox	• Includes both log and power transformations. • Depends on parameter lambda. • The log is always natural log.	$w_t = \begin{cases} \log(y_t) & \text{if } \lambda = 0; \\ (\text{sign}(y_t)\lvert y_t\rvert^\lambda - 1)/\lambda & \text{otherwise.} \end{cases}$

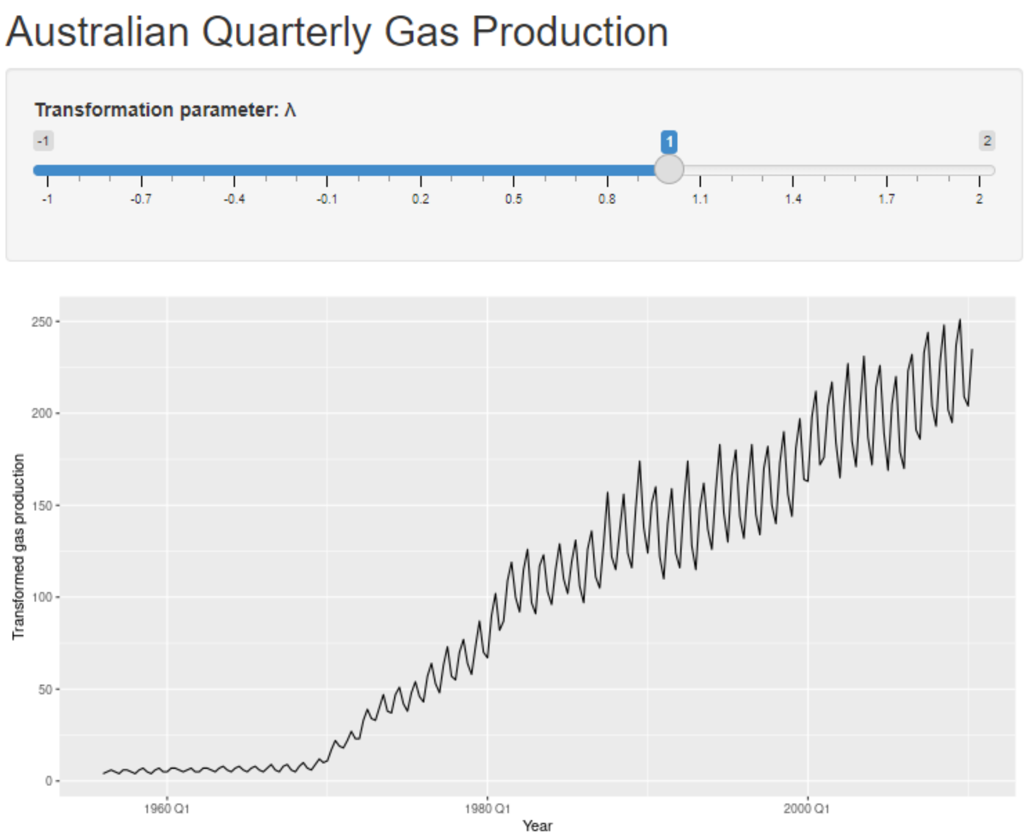
- Common use case: Reducing the impact of outliers and making patterns in the data more visible.

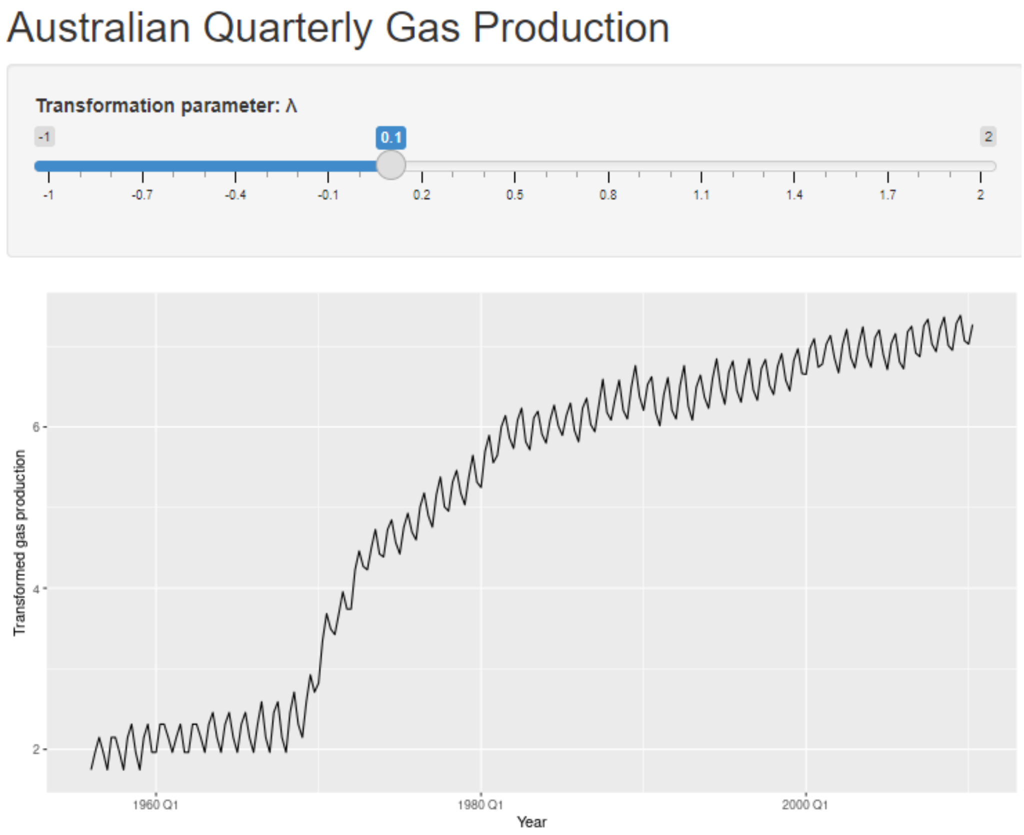
Jon M. Huntsman School of Business Utah State University

Pedram Jahangiry

Box-Cox transformation and lambda

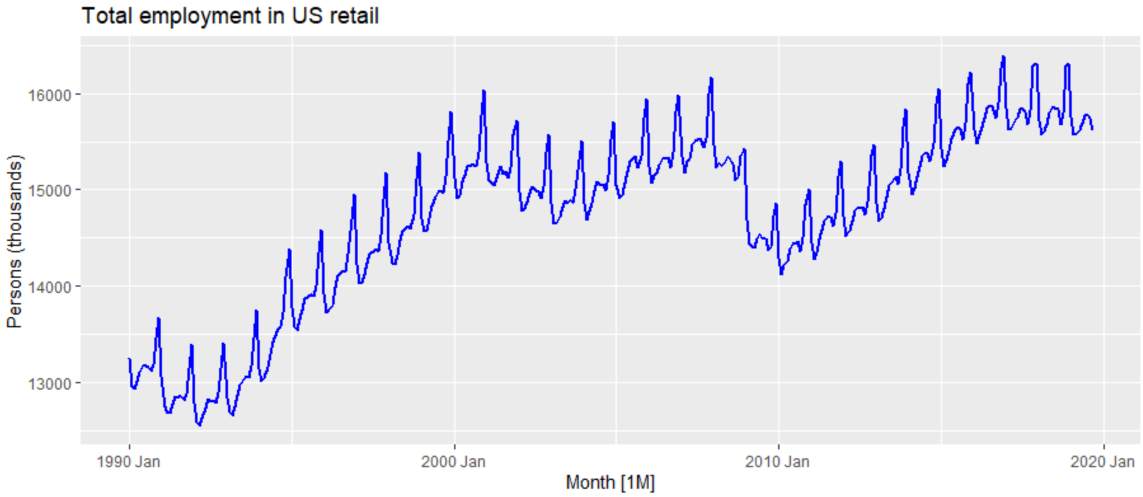
- A good value of λ is one which makes the size of the seasonal variation about the same across the whole series





Time Series Components

- Time series patterns: Trend, Seasonality, Cycle
- Decomposing Time Series:
  - Trend-Cycle component
  - Seasonal component
  - Remainder (error) component



- The decomposition of a time series is done to improve understanding, but it can also be used to improve forecasting accuracy.
- For simplified decomposition (and later analysis), it is sometimes helpful to transform or adjust the series first.

Jon M. Huntsman School of Business UtahStateUniversity

Pedram Jahangiry

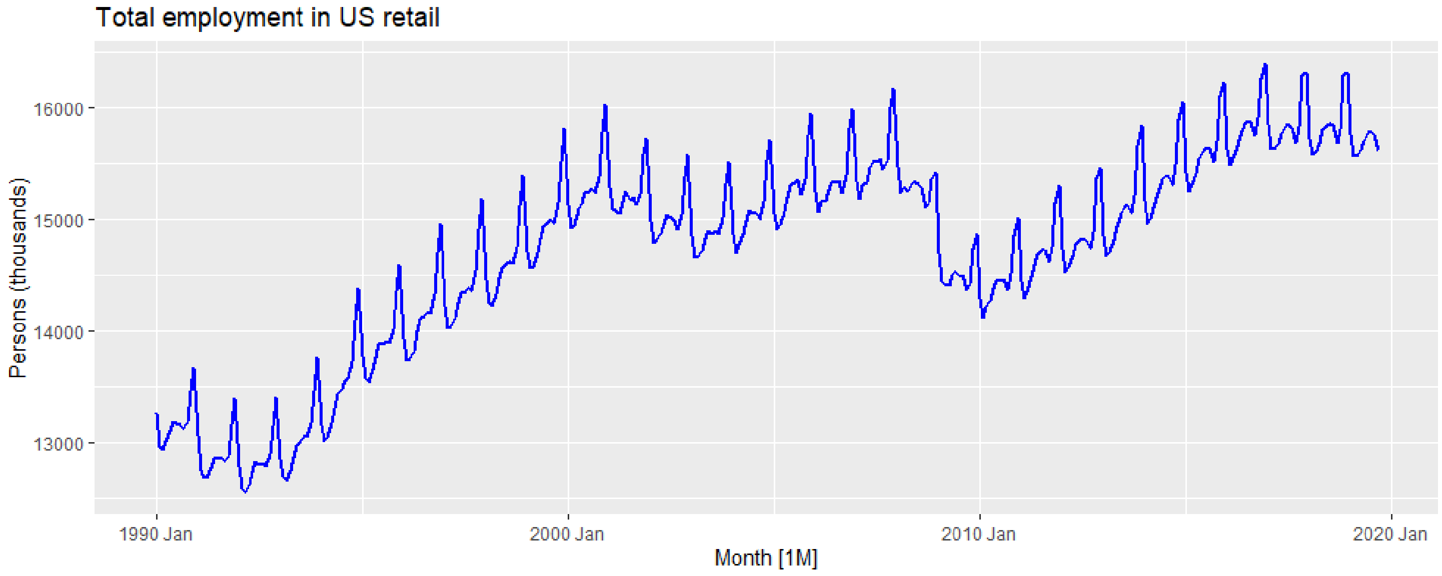
Time Series Components Decomposition

TS Decomposition

- Additive: $y_t = S_t + T_t + R_t$
- Multiplicative: $y_t = S_t * T_t * R_t$

$$\log y_t = \log S_t + \log T_t + \log R_t$$

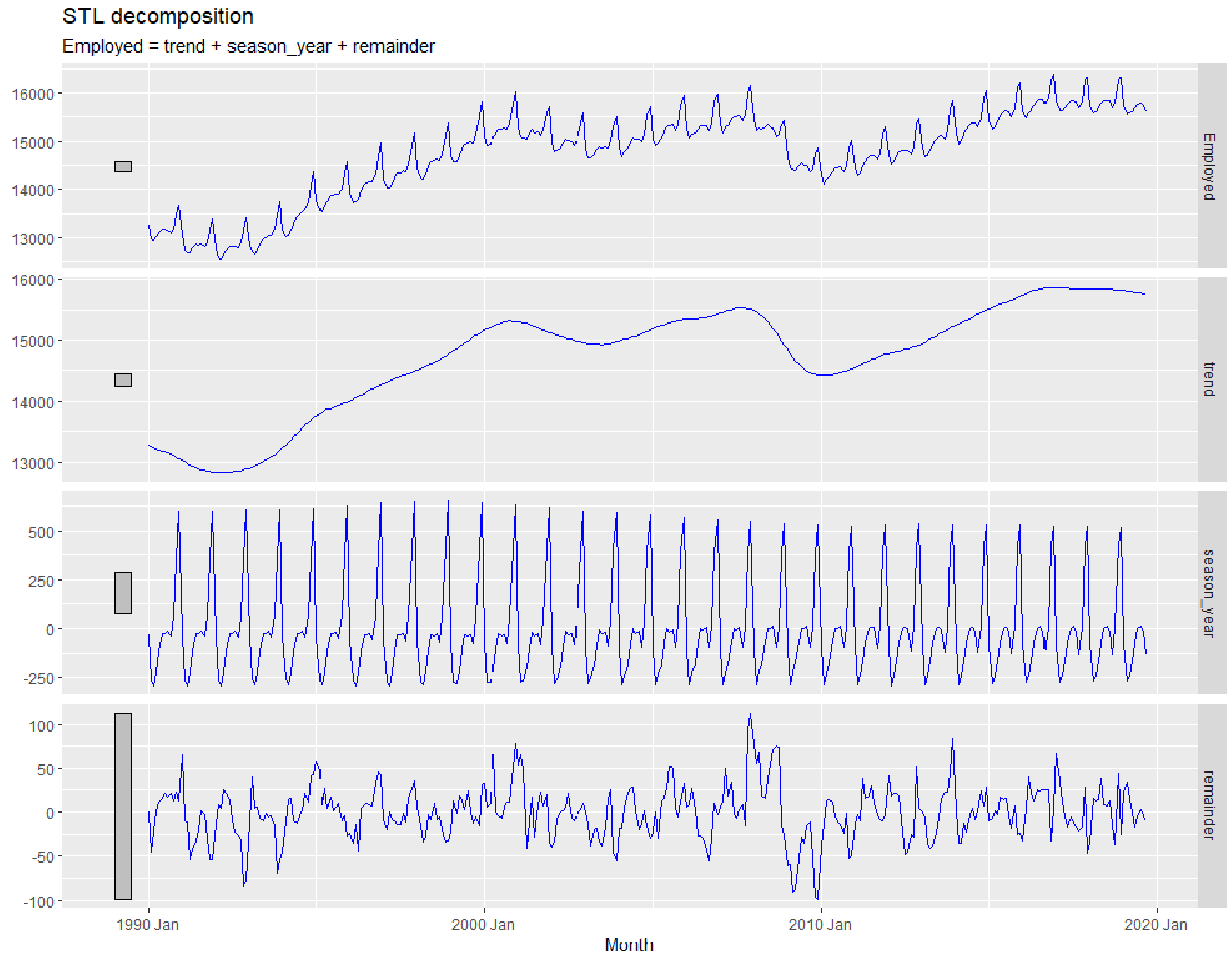
- The additive decomposition is appropriate if the magnitude of seasonal fluctuations or variation around trend-cycles do not change with time series level.



Jon M. Huntsman School of Business
Utah State University

Pedram Jahangiry

Time Series Components Decomposition Example

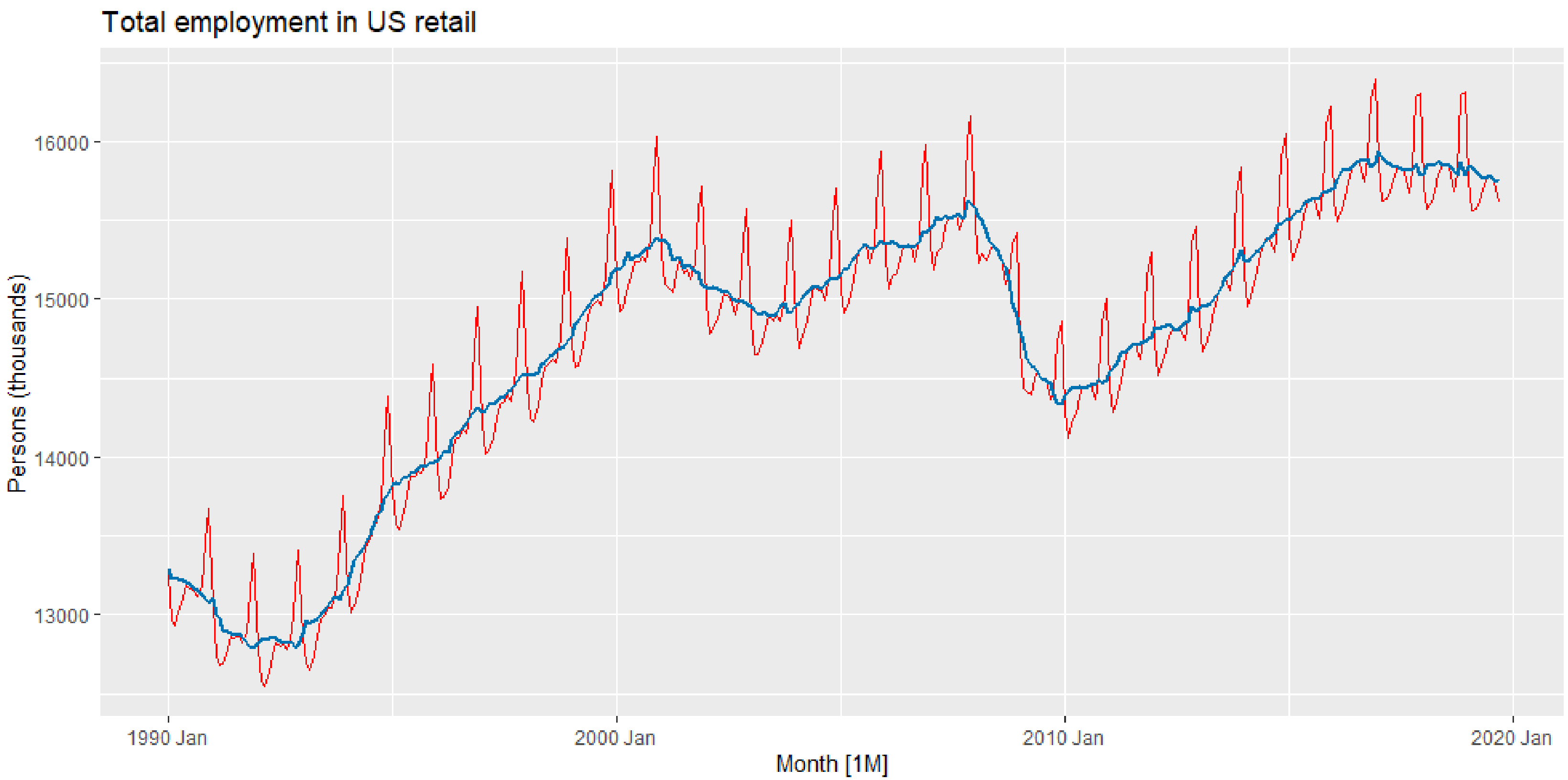


Jon M. Huntsman School of Business
Utah State University

Pedram Jahangiry

Seasonally Adjusted Data

- If the seasonal component is removed from the original data, the resulting values are the "seasonally adjusted" data.



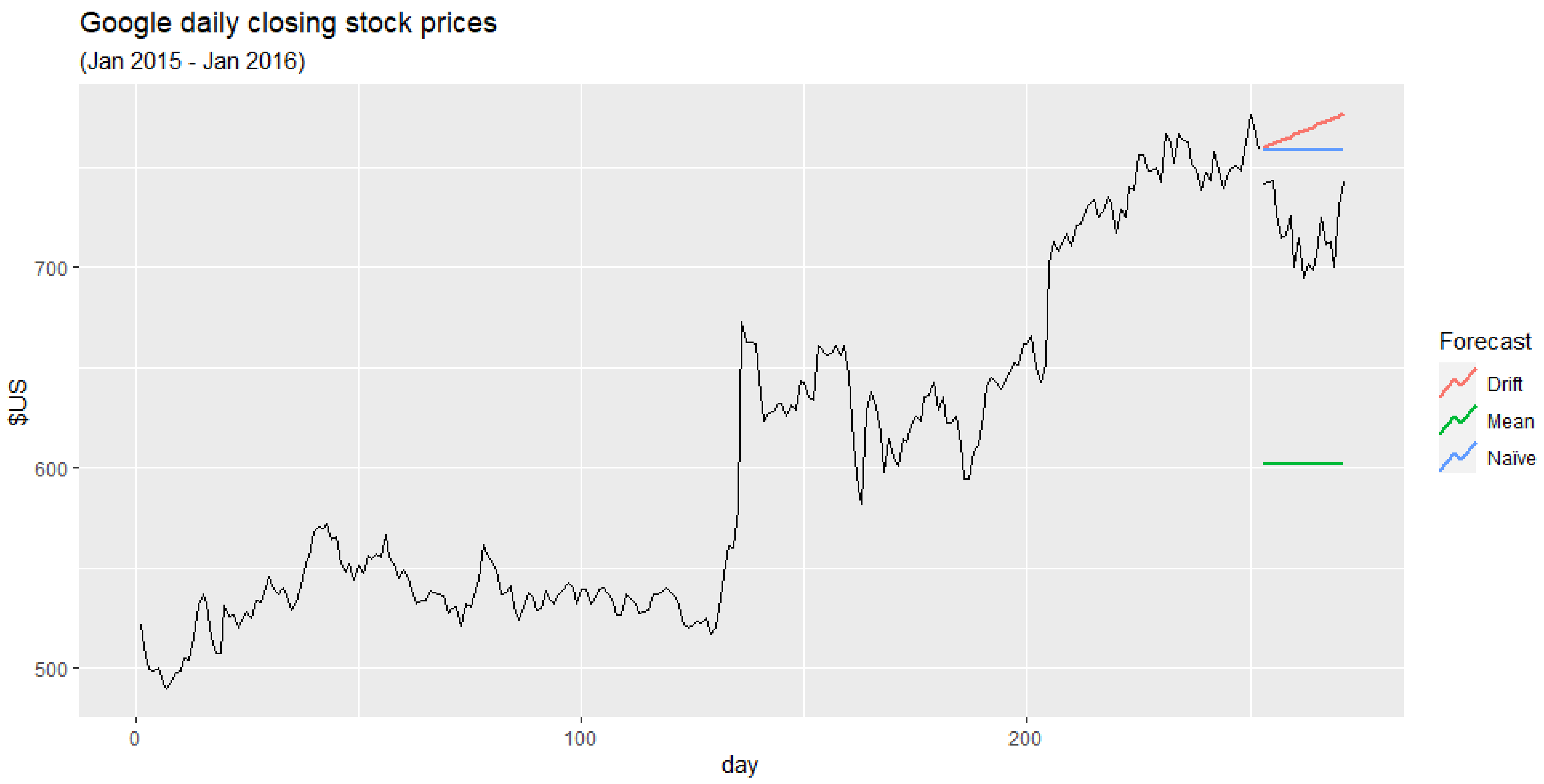
Some Forecasting Benchmarks

- Some forecasting methods are extremely simple and surprisingly effective.

1. Mean method: the forecasts of all future values are equal to the average of the historical data.
2. Naïve method: all forecasts are set to be the value of the last observation
3. Seasonal Naïve: each forecast set to be equal to the last observed value from the same season
4. Drift method: allows forecasts to increase or decrease over time, where the drift is set to be the average change in historical data.

- These methods will serve as benchmarks rather than the method of choice.
- If your model cannot beat the benchmark, it is not worth considering!!!

Some Forecasting Benchmarks (Google Stock price forecasting)



Road map!

- ✓ Module 1- Introduction to Deep Forecasting
- ✓ Module 2- Setting up Deep Forecasting Environment
- Module 3- ETS and Exponential Smoothing
- Module 4- ARIMA models
- Module 5- Machine Learning for Time series Forecasting
- Module 6- Deep Neural Networks
- Module 7- Deep Sequence Modeling (RNN, LSTM)
- Module 8- Transformers (Attention is all you need!)
- Module 9- Prophet and Neural Prophet

Jon M. Huntsman School of Business
UtahStateUniversity

Pedram Jahangiry