```
[1]: # Core libraries
import numpy as np, pandas as pd
import matplotlib as mpl, matplotlib.pyplot as plt, seaborn as sns

[2]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet

[3]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow

[4]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy

[5]: # Ignore warnings
warnings.filterwarnings('ignore')

[6]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__

[7]: # Core libraries
import numpy as np, pandas as pd
import matplotlib as mpl, matplotlib.pyplot as plt, seaborn as sns

[8]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet

[9]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow

[10]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy

[11]: # Ignore warnings
warnings.filterwarnings('ignore')

[12]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__

[13]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

TimeSeries 📈 ARIMA, Prophet, ADF, PACF.... 🎯 Beginner to Pro

In this project I want to keep alive the Traditional analysis so since now to review the data. Now it provides it and how to engineering.

I also want to explore some of the typical Timeframe topics such as:

- Stationarity
- ARIMA
- Prophet
- Exponential Smoothing Family (ADF)

Table of Content

- 1. Data Visualization 📊
  - 1.1 Chronological Order
  - 1.2 Calendar Heatmap
  - 1.3 Seasonal Decomposition
  - 1.4 Distributions
    - 1.4.1 Kernel Density Estimate (KDE)
    - 1.4.2 Boxplots
    - 1.4.3 Violin Plots
  - 1.5 Autocorrelations 🔁
    - 1.5.1 Autocorrelation Function
    - 1.5.2 Partial Autocorrelation
    - 1.5.3 Lag Plots
  - 1.6 Interactive Data Analysis 📊
    - 1.6.1 Interactive Boxplots
  - 1.7 Resampling
    - 1.7.1 Resampling Time Series Data Frames
    - 1.7.2 Downsampling Example
    - 1.7.3 Upsampling
    - 1.7.4 Upsampling Example
    - 1.7.5 Resampling Multivariate Time Series
    - 1.7.6 Rolling Aggregations Example
  - 1.8 Stationarity
  - 1.9 Differences


```
[14]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[15]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[16]: # Machine Learning
sklearn.__version__
```

```
[17]: # Time Series
statsmodels.__version__
```

```
[18]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[19]: # Other
prophet.__version__
pmdarima.__version__
```

```
[20]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[21]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[22]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[23]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[24]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[25]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[26]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[27]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[28]: # Machine Learning
sklearn.__version__
```

```
[29]: # Time Series
statsmodels.__version__
```

```
[30]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[31]: # Other
prophet.__version__
pmdarima.__version__
```

```
[32]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[33]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[34]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[35]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[36]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[37]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[38]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[39]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[40]: # Machine Learning
sklearn.__version__
```

```
[41]: # Time Series
statsmodels.__version__
```

```
[42]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[43]: # Other
prophet.__version__
pmdarima.__version__
```

```
[44]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[45]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[46]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[47]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[48]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[49]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[50]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[51]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[52]: # Machine Learning
sklearn.__version__
```

```
[53]: # Time Series
statsmodels.__version__
```

```
[54]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[55]: # Other
prophet.__version__
pmdarima.__version__
```

```
[56]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[57]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[58]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[59]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[60]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[61]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[62]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[63]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[64]: # Machine Learning
sklearn.__version__
```

```
[65]: # Time Series
statsmodels.__version__
```

```
[66]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[67]: # Other
prophet.__version__
pmdarima.__version__
```

```
[68]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[69]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[70]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[71]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[72]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[73]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[74]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[75]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[76]: # Machine Learning
sklearn.__version__
```

```
[77]: # Time Series
statsmodels.__version__
```

```
[78]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[79]: # Other
prophet.__version__
pmdarima.__version__
```

```
[80]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[81]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[82]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[83]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[84]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[85]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[86]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[87]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[88]: # Machine Learning
sklearn.__version__
```

```
[89]: # Time Series
statsmodels.__version__
```

```
[90]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[91]: # Other
prophet.__version__
pmdarima.__version__
```

```
[92]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[93]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[94]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[95]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[96]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[97]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[98]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[99]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[100]: # Machine Learning
sklearn.__version__
```

```
[101]: # Time Series
statsmodels.__version__
```

```
[102]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[103]: # Other
prophet.__version__
pmdarima.__version__
```

```
[104]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[105]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[106]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[107]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[108]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[109]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[110]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[111]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[112]: # Machine Learning
sklearn.__version__
```

```
[113]: # Time Series
statsmodels.__version__
```

```
[114]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[115]: # Other
prophet.__version__
pmdarima.__version__
```

```
[116]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[117]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[118]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[119]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[120]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[121]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[122]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[123]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[124]: # Machine Learning
sklearn.__version__
```

```
[125]: # Time Series
statsmodels.__version__
```

```
[126]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[127]: # Other
prophet.__version__
pmdarima.__version__
```

```
[128]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[129]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[130]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[131]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[132]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[133]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[134]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[135]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[136]: # Machine Learning
sklearn.__version__
```

```
[137]: # Time Series
statsmodels.__version__
```

```
[138]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[139]: # Other
prophet.__version__
pmdarima.__version__
```

```
[140]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[141]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[142]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[143]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[144]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[145]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[146]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[147]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[148]: # Machine Learning
sklearn.__version__
```

```
[149]: # Time Series
statsmodels.__version__
```

```
[150]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[151]: # Other
prophet.__version__
pmdarima.__version__
```

```
[152]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[153]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[154]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[155]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[156]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[157]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[158]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[159]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[160]: # Machine Learning
sklearn.__version__
```

```
[161]: # Time Series
statsmodels.__version__
```

```
[162]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[163]: # Other
prophet.__version__
pmdarima.__version__
```

```
[164]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[165]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[166]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[167]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[168]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[169]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[170]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[171]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[172]: # Machine Learning
sklearn.__version__
```

```
[173]: # Time Series
statsmodels.__version__
```

```
[174]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[175]: # Other
prophet.__version__
pmdarima.__version__
```

```
[176]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[177]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[178]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[179]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[180]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[181]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[182]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[183]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[184]: # Machine Learning
sklearn.__version__
```

```
[185]: # Time Series
statsmodels.__version__
```

```
[186]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[187]: # Other
prophet.__version__
pmdarima.__version__
```

```
[188]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[189]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[190]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[191]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[192]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[193]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[194]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[195]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[196]: # Machine Learning
sklearn.__version__
```

```
[197]: # Time Series
statsmodels.__version__
```

```
[198]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[199]: # Other
prophet.__version__
pmdarima.__version__
```

```
[200]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[201]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[202]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[203]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[204]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[205]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[206]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[207]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[208]: # Machine Learning
sklearn.__version__
```

```
[209]: # Time Series
statsmodels.__version__
```

```
[210]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[211]: # Other
prophet.__version__
pmdarima.__version__
```

```
[212]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[213]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[214]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[215]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[216]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[217]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[218]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[219]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[220]: # Machine Learning
sklearn.__version__
```

```
[221]: # Time Series
statsmodels.__version__
```

```
[222]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[223]: # Other
prophet.__version__
pmdarima.__version__
```

```
[224]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima, prophet
```

```
[225]: # Machine Learning Libraries
import sklearn, sklearn.preprocessing, sklearn.model_selection, sklearn.metrics, sklearn.ensemble, xgboost, keras, tensorflow
```

```
[226]: # Misc Libraries
import os, sys, gc, warnings, pickle, zipfile, datetime, requests, tqdm, PIL, IPython, joblib, psutil, numba, yaml, dill, jinja2, boto3, sqlalchemy
```

```
[227]: # Ignore warnings
warnings.filterwarnings('ignore')
```

```
[228]: # Check versions
!python -V
numpy.__version__, pandas.__version__
matplotlib.__version__, seaborn.__version__
sklearn.__version__
statsmodels.__version__
xgboost.__version__, keras.__version__, tensorflow.__version__
prophet.__version__
pmdarima.__version__
```

```
[229]: # Ignore specific warnings
import warnings, sklearn.exceptions, statsmodels.tsa.base.tsa_model, statsmodels.base.wrapper
warnings.filterwarnings('ignore', category=sklearn.exceptions.DataConversionWarning)
warnings.filterwarnings('ignore', category=statsmodels.tsa.base.tsa_model.ValueWarning)
warnings.filterwarnings('ignore', category=statsmodels.base.wrapper.FutureWarning)
```

```
[230]: # Data Manipulation
numpy.__version__, pandas.__version__
```

```
[231]: # Data Visualization
matplotlib.__version__, seaborn.__version__
```

```
[232]: # Machine Learning
sklearn.__version__
```

```
[233]: # Time Series
statsmodels.__version__
```

```
[234]: # Deep Learning
xgboost.__version__, keras.__version__, tensorflow.__version__
```

```
[235]: # Other
prophet.__version__
pmdarima.__version__
```

```
[236]: # Time Series Specific Libraries
import statsmodels, statsmodels.tsa.stattools, statsmodels.tsa.seasonal, statsmodels.tsa.arima.model, pmdarima,
```

```
In [24]: # Calculate Correlation Matrix for Time Series Variables
         corr_matrix = df[['Temperature', 'Humidity', 'Wind_Speed', 'Precipitation']].corr()

         # Plot the correlation matrix using a heatmap
         plt.figure(figure=(8, 6))
         sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', vmin=-1, vmin=1, linewidths=0.5)
         plt.title('Correlation Matrix for Time Series Variables')
         plt.show()
```

```
In [25]: # Scatter plots for pairwise relationships
         sns.pairplot(df[['Temperature', 'Humidity', 'Wind_Speed', 'Precipitation']])
         plt.suptitle('Pairwise Relationships', y=1.02)
         plt.show()
```

```
In [26]: # Time series plot for Temperature
         plt.figure(figure=(12, 6))
         plt.plot(df['Temperature'], label='Temperature')
         plt.title('Temperature Over Time')
         plt.xlabel('Date')
         plt.ylabel('Temperature')
         plt.legend()
         plt.show()
```

```
In [27]: # Time series plot for Humidity
         plt.figure(figure=(12, 6))
         plt.plot(df['Humidity'], label='Humidity')
         plt.title('Humidity Over Time')
         plt.xlabel('Date')
         plt.ylabel('Humidity')
         plt.legend()
         plt.show()
```

```
In [28]: # Time series plot for Wind_Speed
         plt.figure(figure=(12, 6))
         plt.plot(df['Wind_Speed'], label='Wind_Speed')
         plt.title('Wind_Speed Over Time')
         plt.xlabel('Date')
         plt.ylabel('Wind_Speed')
         plt.legend()
         plt.show()
```

```
In [29]: # Time series plot for Precipitation
         plt.figure(figure=(12, 6))
         plt.plot(df['Precipitation'], label='Precipitation')
         plt.title('Precipitation Over Time')
         plt.xlabel('Date')
         plt.ylabel('Precipitation')
         plt.legend()
         plt.show()
```

```
In [30]: # Boxplots for each variable to check for outliers
         plt.figure(figure=(12, 6))
         sns.boxplot(data=df[['Temperature', 'Humidity', 'Wind_Speed', 'Precipitation']])
         plt.title('Boxplots for Time Series Variables')
         plt.show()
```

3.2.2 Transforming

```
In [31]: # Log transform of precipitation
         df['log_precipitation'] = np.log(df['Precipitation'])

         # Square root transform of wind speed
         df['sqrt_wind_speed'] = np.sqrt(df['Wind_Speed'])

         # Plot the transformed precipitation
         plt.figure(figure=(12, 6))
         plt.plot(df['log_precipitation'], label='Log Transformed Precipitation')
         plt.title('Log Transformed Precipitation')
         plt.xlabel('Date')
         plt.ylabel('Log Precipitation')
         plt.legend()
         plt.show()

         # Plot the transformed wind speed
         plt.figure(figure=(12, 6))
         plt.plot(df['sqrt_wind_speed'], label='Square Root Transformed Wind Speed')
         plt.title('Square Root Transformed Wind Speed')
         plt.xlabel('Date')
         plt.ylabel('Square Root Wind Speed')
         plt.legend()
         plt.show()
```

Differencing can be done in different orders:

- First order differencing: lag-1 time shift ($y_t - y_{t-1}$)
- Second order differencing: applied twice with $y_t - 2y_{t-1} + y_{t-2}$
- and so on...

3.2.3 Differencing

```
In [32]: # First order differencing
         df['first_order_diff'] = df['Temperature'].diff()

         # Second order differencing
         df['second_order_diff'] = df['first_order_diff'].diff()

         # Plot the original, first order, and second order differences
         plt.figure(figure=(12, 6))
         plt.plot(df['Temperature'], label='Original')
         plt.plot(df['first_order_diff'], label='First Order Diff')
         plt.plot(df['second_order_diff'], label='Second Order Diff')
         plt.title('Original, First Order, and Second Order Differences')
         plt.xlabel('Date')
         plt.ylabel('Value')
         plt.legend()
         plt.show()
```

Differencing can be done in different orders:

- First order differencing: lag-1 time shift ($y_t - y_{t-1}$)
- Second order differencing: applied twice with $y_t - 2y_{t-1} + y_{t-2}$
- and so on...

3. Feature engineering 🔧

```
In [33]: df['Year'] = df.index.year
         df['Month'] = df.index.month
         df['Day'] = df.index.day
         df['Day_of_Week'] = df.index.dayofweek
         df['Is_Weekend'] = df['Day_of_Week'] >= 5

         # Display the first few rows with the new features
         print(df[['Year', 'Month', 'Day', 'Day_of_Week', 'Is_Weekend']].head())
```

```
            Year  Month  Day  Day_of_Week  Is_Weekend
Date
2010-01-01  2010      1    1            4       False
2010-01-02  2010      1    2            5        True
2010-01-03  2010      1    3            6        True
2010-01-04  2010      1    4            0       False
2010-01-05  2010      1    5            1       False
```

3.1 Encoding Cyclical Features

The new time features are cyclical. For example the feature month cycles between 1 and 12 for every year. While the difference between consecutive months is usually 1 (e.g., January and February), the difference between December (12) and January (1) is also 1. Neural networks treat 12 December as 12 potatoes. This results in a discontinuity that confuses our model.

```
In [34]: df['Month_sin'] = np.sin(2 * np.pi * df['Month'] / 12)
         df['Month_cos'] = np.cos(2 * np.pi * df['Month'] / 12)
```

Let's plot this cyclical feature:

```
In [35]: # Plot the cyclical encoding for month
         plt.figure(figure=(8, 8))
         plt.scatter(df['Month_sin'], df['Month_cos'], c=df['Month'], cmap='hsv')
         plt.title('Cyclical Encoding of Month')
         plt.xlabel('Month_sin')
         plt.ylabel('Month_cos')
         plt.colorbar(label='Month')
         plt.show()
```

3.2 Timeries Decomposition

Time series decomposition involves thinking of a series as a combination of level, trend, seasonality, and noise components.

These components can defined as follows:

- **Level:** The average value in the series.
- **Trend:** The rate at which value increases or decreases over time.
- **Seasonality:** The repeating short-term cycle in the series.
- **Noise:** Random variation in the series.

Decomposition provides a useful abstract model for thinking about time series generally and for better understanding problems during time series analysis and forecasting.

All series have a level and noise. The trend and seasonality components are optional.

It is helpful to think of the components as combining either additively or multiplicatively:

- **Additive:** series = Level + Trend + Seasonality + Noise
- **Multiplicative:** series = Level * Trend * Seasonality * Noise

A time series can be decomposed manually, demonstrated from this machinelearningmastery library.

```
In [36]: from statsmodels.tsa.seasonal import seasonal_decompose

         # Decompose the Temperature time series
         decomposition = seasonal_decompose(df['Temperature'], model='additive', period=365)

         # Plot the components
         plt.figure(figure=(12, 8))
         decomposition.plot()
         plt.suptitle('Decomposition of Temperature', y=1.02)
         plt.show()
```

```
In [37]: # Decompose the Humidity time series
         decomposition = seasonal_decompose(df['Humidity'], model='additive', period=365)

         # Plot the components
         plt.figure(figure=(12, 8))
         decomposition.plot()
         plt.suptitle('Decomposition of Humidity', y=1.02)
         plt.show()
```

3.3 Lag

We want to calculate each variable with an ACF(1). This ACF compares the correlation with the other variables.

```
In [38]: pd.plotting.lag_plot(df['Temperature'])
         plt.title('Lag Plot for Temperature')
         plt.show()
```

4. Exploratory Data Analysis 🔍

Now, we are going to pick the data sets by certain some knowledge.

```
In [39]: # Plot seasonal decomposition for Temperature
         decomposition = seasonal_decompose(df['Temperature'], model='additive', period=365)
         plt.figure(figure=(12, 8))
         decomposition.plot()
         plt.suptitle('Seasonal Decomposition of Temperature', y=1.02)
         plt.show()
```

We can see:

- **depth to groundwater:** depends the maximum amount they could vary by minimum around November
- **Temperature:** depends the maximum amount they could vary by minimum around September
- **Average wind speed:** depends the maximum amount they could vary by minimum around September
- **Precipitation:** depends the maximum amount they could vary by minimum around September

```
In [40]: # Calculate monthly average temperature
         monthly_avg = df['Temperature'].resample('M').mean()

         # Plot the monthly average temperature
         plt.figure(figure=(12, 6))
         monthly_avg.plot(kind='bar')
         plt.title('Monthly Average Temperature')
         plt.xlabel('Month')
         plt.ylabel('Average Temperature')
         plt.xticks(rotation=45)
         plt.show()
```

```
/var/folder/_/_0_0_0
```

```
In [18]: # ARIMA(1,1,1)
# ARIMA(2,1,2)
# ARIMA(3,1,3)

model_fit_1_1_1 = model_1_1_1.fit()
model_fit_2_1_2 = model_2_1_2.fit()
model_fit_3_1_3 = model_3_1_3.fit()

ax1.plot(df_test['Close'], label='Test', lw=2)
ax1.plot(model_fit_1_1_1.forecast(len(df_test)), label='ARIMA(1,1,1)', lw=2)
ax1.plot(model_fit_2_1_2.forecast(len(df_test)), label='ARIMA(2,1,2)', lw=2)
ax1.plot(model_fit_3_1_3.forecast(len(df_test)), label='ARIMA(3,1,3)', lw=2)
ax1.set_title('Forecast', fontsize=20)
ax1.set_ylabel('Close', fontsize=18)
ax1.set_xlabel('Time', fontsize=18)
ax1.legend(loc='upper left')
```

```
In [19]: # ARIMA(1,1,1)
# ARIMA(2,1,2)
# ARIMA(3,1,3)

pred_1_1_1 = model_fit_1_1_1.forecast(len(df_test))
pred_2_1_2 = model_fit_2_1_2.forecast(len(df_test))
pred_3_1_3 = model_fit_3_1_3.forecast(len(df_test))

print('ARIMA(1,1,1) MSE:', mean_squared_error(df_test['Close'], pred_1_1_1))
print('ARIMA(2,1,2) MSE:', mean_squared_error(df_test['Close'], pred_2_1_2))
print('ARIMA(3,1,3) MSE:', mean_squared_error(df_test['Close'], pred_3_1_3))
```

```
ARIMA(1,1,1) MSE: 1.7174.....7.....7.....7
```