

t-Test

Data Analyst: Gyro A. Madrona
Department: Electrical Engineering

```
1 # imports and packages
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from scipy import stats
```

Python

```
1 # load dataset
2 df = pd.read_csv(r"raw\electricity-normal-sample.csv",
3                 delimiter=",")
4 df
```

Open 'df' in Data Wrangler

Python

```
1 # summary of dataframe
2 df.info()
```

Python

```
1 # summary of statistics
2 df.describe()
```

Python

Data Cleaning

```
1 # check for outliers
2 plt.figure(figsize=(10,5))
3 plt.boxplot(df[['Nuclear', 'Wind', 'Hydroelectric', 'Oil and Gas', 'Coal', 'Solar', 'Biomass']],
4             tick_labels=df.columns)
5 plt.title('Production Type')
6 plt.ylabel('MWh')
7 #plt.xticks(rotation=45)
8 plt.show()
```

Python

1-Sample t-Test

```
1 # left-tailed test
2 # Ho: u1 >= 1283.78
3 # Ha: u1 < 1283.78
4 pop_mean = 1283.78
5 t_test, p_value = stats.ttest_1samp(df['Nuclear'], pop_mean,
6                                     alternative='less')
7
```

Python

2-Sample t-Test

F-Test

```
1 # nuclear and coal
2 # Ho: var_nuclear = var_coal
3 # Ha: var_nuclear != var_coal
4 var_nuc = df['Nuclear'].var(ddof=1)
5 var_coal = df['Coal'].var(ddof=1)
6
7 # F-statistic
8 if var_nuc > var_coal:
9     f_stat = var_nuc/var_coal
10 else:
11     f_stat = var_coal/var_nuc
12
13 f_stat
```

Python

Nuclear and Coal

```
1 # nuclear and coal
2 # Ho: Nuclear <= Coal
3 # Ha: Nuclear > Coal
4 t_stat, p_value = stats.ttest_ind(df['Nuclear'],df['Coal'],
5                                   alternative='greater',
6                                   equal_var=True)
```

Python

Paired t-Test

```
1 # load dataset
2 df = pd.read_csv(r"raw\nuclear-maintenance.csv",
3                 delimiter=",")
4 df
```

Open 'df' in Data Wrangler

Python

```
1 # summary of dataframe
2 df.info()
```

Python

```
1 # summary of statistics
2 df.describe()
```

Python

Data Cleaning

```
1 # check for outliers
2 plt.boxplot(df[['Before','After']],tick_labels=df.columns)
3 plt.title("Nuclear Maintenance")
4 plt.show()
```

Python



Prepared by:

Gyro A. Madrona

Electronics Engineer

F-Test

```
1 # before and after
2 # Ho: var_before = var_after
3 # Ha: var_before != var_after
4 var_before = df['Before'].var(ddof=1)
5 var_after = df['After'].var(ddof=1)
6
7 # F-statistic
8 if var_before > var_after:
9     f_stat = var_after/var_before
10 else:
11     f_stat = var_before/var_after
12
13 f_stat
```

Python

Before and After Maintenance

```
1 # paired t-test
2 t_stat, p_value = stats.ttest_rel(df['Before'],df['After'])
```

Python