autoeda

January 2, 2024

[ ]:
```
# Load the Data
import pandas as pd

df = pd.read_csv(r"education.csv")
df
```

[ ]:
```
     datasrno  workex  gmat
0           1      21   720
1           2     107   640
2           3      57   740
3           4      99   690
4           5     208   710
..        …      …    …
768       769      88   620
769       770     132   670
770       771      28   610
771       772      10   610
772       773      52   620

[773 rows x 3 columns]
```

[ ]:
```
# Auto EDA
# ---------
# Sweetviz
# Autoviz
# Dtale
# Pandas Profiling
# Dataprep
```

[ ]:
```
!pip install sweetviz
```

```
Collecting sweetviz
  Downloading sweetviz-2.1.4-py3-none-any.whl (15.1 MB)
                           15.1/15.1 MB
29.3 MB/s eta 0:00:00
Requirement already satisfied: pandas!=1.0.0,!=1.0.1,!=1.0.2,>=0.25.3 in
/usr/local/lib/python3.10/dist-packages (from sweetviz) (1.5.3)
Requirement already satisfied: numpy>=1.16.0 in /usr/local/lib/python3.10/dist-
```

```
packages (from sweetviz) (1.23.5)
Requirement already satisfied: matplotlib>=3.1.3 in
/usr/local/lib/python3.10/dist-packages (from sweetviz) (3.7.1)
Requirement already satisfied: tqdm>=4.43.0 in /usr/local/lib/python3.10/dist-
packages (from sweetviz) (4.66.1)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-
packages (from sweetviz) (1.10.1)
Requirement already satisfied: jinja2>=2.11.1 in /usr/local/lib/python3.10/dist-
packages (from sweetviz) (3.1.2)
Requirement already satisfied: importlib-resources>=1.2.0 in
/usr/local/lib/python3.10/dist-packages (from sweetviz) (6.0.1)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2>=2.11.1->sweetviz) (2.1.3)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.3->sweetviz)
(1.1.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
packages (from matplotlib>=3.1.3->sweetviz) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.3->sweetviz)
(4.42.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.3->sweetviz)
(1.4.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.3->sweetviz)
(23.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-
packages (from matplotlib>=3.1.3->sweetviz) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.3->sweetviz)
(3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.3->sweetviz)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas!=1.0.0,!=1.0.1,!=1.0.2,>=0.25.3->sweetviz) (2023.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.7->matplotlib>=3.1.3->sweetviz) (1.16.0)
Installing collected packages: sweetviz
Successfully installed sweetviz-2.1.4
```

[ ]:
```python
# Sweetviz
###########
#pip install sweetviz
import sweetviz as sv
```

```
s = sv.analyze(df)
s.show_notebook()
```

```
/usr/local/lib/python3.10/dist-packages/sweetviz/dataframe_report.py:74:
FutureWarning: iteritems is deprecated and will be removed in a future version.
Use .items instead.
  all_source_names = [cur_name for cur_name, cur_series in
source_df.iteritems()]
/usr/local/lib/python3.10/dist-packages/sweetviz/dataframe_report.py:109:
FutureWarning: iteritems is deprecated and will be removed in a future version.
Use .items instead.
  filtered_series_names_in_source = [cur_name for cur_name, cur_series in
source_df.iteritems()

                                                |          | [  0%]   00:00 -> (?␣
 ↪left)

/usr/local/lib/python3.10/dist-packages/sweetviz/series_analyzer_numeric.py:25:
FutureWarning: The 'mad' method is deprecated and will be removed in a future
version. To compute the same result, you may do `(df - df.mean()).abs().mean()`.
  stats["mad"] = series.mad()
/usr/local/lib/python3.10/dist-packages/sweetviz/series_analyzer_numeric.py:25:
FutureWarning: The 'mad' method is deprecated and will be removed in a future
version. To compute the same result, you may do `(df - df.mean()).abs().mean()`.
  stats["mad"] = series.mad()
/usr/local/lib/python3.10/dist-packages/sweetviz/series_analyzer_numeric.py:25:
FutureWarning: The 'mad' method is deprecated and will be removed in a future
version. To compute the same result, you may do `(df - df.mean()).abs().mean()`.
  stats["mad"] = series.mad()

<IPython.core.display.HTML object>
```

[ ]:
```
pip install autoviz
```

```
Collecting autoviz
  Downloading autoviz-0.1.730-py3-none-any.whl (67 kB)
                           67.0/67.0 kB
1.6 MB/s eta 0:00:00
Collecting bokeh~=2.4.2 (from autoviz)
  Downloading bokeh-2.4.3-py3-none-any.whl (18.5 MB)
                           18.5/18.5 MB
32.9 MB/s eta 0:00:00
Collecting emoji (from autoviz)
  Downloading emoji-2.8.0-py2.py3-none-any.whl (358 kB)
                          358.9/358.9 kB
29.3 MB/s eta 0:00:00
Requirement already satisfied: fsspec>=0.8.3 in
/usr/local/lib/python3.10/dist-packages (from autoviz) (2023.6.0)
Collecting holoviews~=1.14.9 (from autoviz)
```

```
  Downloading holoviews-1.14.9-py2.py3-none-any.whl (4.3 MB)
                             4.3/4.3 MB
55.0 MB/s eta 0:00:00
Collecting hvplot~=0.7.3 (from autoviz)
  Downloading hvplot-0.7.3-py2.py3-none-any.whl (3.1 MB)
                             3.1/3.1 MB
26.4 MB/s eta 0:00:00
Requirement already satisfied: ipython in /usr/local/lib/python3.10/dist-
packages (from autoviz) (7.34.0)
Collecting jupyter (from autoviz)
  Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Requirement already satisfied: matplotlib>=3.3.3 in
/usr/local/lib/python3.10/dist-packages (from autoviz) (3.7.1)
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages
(from autoviz) (3.8.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(from autoviz) (1.23.5)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages
(from autoviz) (1.5.3)
Collecting pandas-dq==1.28 (from autoviz)
  Downloading pandas_dq-1.28-py3-none-any.whl (25 kB)
Requirement already satisfied: panel>=0.12.6 in /usr/local/lib/python3.10/dist-
packages (from autoviz) (1.2.1)
Collecting pyamg (from autoviz)
  Downloading
pyamg-5.0.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.9 MB)
                             1.9/1.9 MB
59.4 MB/s eta 0:00:00
Requirement already satisfied: scikit-learn in
/usr/local/lib/python3.10/dist-packages (from autoviz) (1.2.2)
Requirement already satisfied: seaborn>=0.11.1 in
/usr/local/lib/python3.10/dist-packages (from autoviz) (0.12.2)
Requirement already satisfied: statsmodels in /usr/local/lib/python3.10/dist-
packages (from autoviz) (0.14.0)
Requirement already satisfied: textblob in /usr/local/lib/python3.10/dist-
packages (from autoviz) (0.17.1)
Requirement already satisfied: typing-extensions>=4.1.1 in
/usr/local/lib/python3.10/dist-packages (from autoviz) (4.7.1)
Requirement already satisfied: wordcloud in /usr/local/lib/python3.10/dist-
packages (from autoviz) (1.9.2)
Requirement already satisfied: xgboost>=0.82 in /usr/local/lib/python3.10/dist-
packages (from autoviz) (1.7.6)
Requirement already satisfied: xlrd in /usr/local/lib/python3.10/dist-packages
(from autoviz) (2.0.1)
Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/python3.10/dist-
packages (from bokeh~=2.4.2->autoviz) (3.1.2)
Requirement already satisfied: packaging>=16.8 in
/usr/local/lib/python3.10/dist-packages (from bokeh~=2.4.2->autoviz) (23.1)
```

```
Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/python3.10/dist-
packages (from bokeh~=2.4.2->autoviz) (9.4.0)
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/python3.10/dist-
packages (from bokeh~=2.4.2->autoviz) (6.0.1)
Requirement already satisfied: tornado>=5.1 in /usr/local/lib/python3.10/dist-
packages (from bokeh~=2.4.2->autoviz) (6.3.2)
Requirement already satisfied: param<2.0,>=1.9.3 in
/usr/local/lib/python3.10/dist-packages (from holoviews~=1.14.9->autoviz)
(1.13.0)
Requirement already satisfied: pyviz-comms>=0.7.4 in
/usr/local/lib/python3.10/dist-packages (from holoviews~=1.14.9->autoviz)
(3.0.0)
Requirement already satisfied: colorcet in /usr/local/lib/python3.10/dist-
packages (from holoviews~=1.14.9->autoviz) (3.0.1)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.3->autoviz)
(1.1.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
packages (from matplotlib>=3.3.3->autoviz) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.3->autoviz)
(4.42.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.3->autoviz)
(1.4.4)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.3->autoviz)
(3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.3->autoviz)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas->autoviz) (2023.3)
INFO: pip is looking at multiple versions of panel to determine which version is
compatible with other requirements. This could take a while.
Collecting panel>=0.12.6 (from autoviz)
  Downloading panel-1.2.0-py2.py3-none-any.whl (20.0 MB)
                           20.0/20.0 MB
18.6 MB/s eta 0:00:00
  Downloading panel-1.1.1-py2.py3-none-any.whl (20.0 MB)
                           20.0/20.0 MB
29.5 MB/s eta 0:00:00
  Downloading panel-1.1.0-py2.py3-none-any.whl (20.0 MB)
                           20.0/20.0 MB
26.5 MB/s eta 0:00:00
  Downloading panel-1.0.4-py2.py3-none-any.whl (20.0 MB)
                           20.0/20.0 MB
63.3 MB/s eta 0:00:00
```

```
  Downloading panel-1.0.3-py2.py3-none-any.whl (19.9 MB)
                             19.9/19.9 MB
11.1 MB/s eta 0:00:00
  Downloading panel-1.0.2-py2.py3-none-any.whl (19.9 MB)
                             19.9/19.9 MB
58.6 MB/s eta 0:00:00
  Downloading panel-1.0.1-py2.py3-none-any.whl (19.9 MB)
                             19.9/19.9 MB
30.3 MB/s eta 0:00:00
INFO: pip is looking at multiple versions of panel to determine which
version is compatible with other requirements. This could take a while.
  Downloading panel-1.0.0-py2.py3-none-any.whl (19.9 MB)
                             19.9/19.9 MB
23.3 MB/s eta 0:00:00
  Downloading panel-0.14.4-py2.py3-none-any.whl (20.8 MB)
                             20.8/20.8 MB
13.3 MB/s eta 0:00:00
Requirement already satisfied: markdown in /usr/local/lib/python3.10/dist-
packages (from panel>=0.12.6->autoviz) (3.4.4)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from panel>=0.12.6->autoviz) (2.31.0)
Requirement already satisfied: tqdm>=4.48.0 in /usr/local/lib/python3.10/dist-
packages (from panel>=0.12.6->autoviz) (4.66.1)
Requirement already satisfied: pyct>=0.4.4 in /usr/local/lib/python3.10/dist-
packages (from panel>=0.12.6->autoviz) (0.5.0)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages
(from panel>=0.12.6->autoviz) (6.0.0)
Requirement already satisfied: setuptools>=42 in /usr/local/lib/python3.10/dist-
packages (from panel>=0.12.6->autoviz) (67.7.2)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn->autoviz) (1.10.1)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn->autoviz) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn->autoviz) (3.2.0)
Collecting jedi>=0.16 (from ipython->autoviz)
  Downloading jedi-0.19.0-py2.py3-none-any.whl (1.6 MB)
                             1.6/1.6 MB
51.8 MB/s eta 0:00:00
Requirement already satisfied: decorator in
/usr/local/lib/python3.10/dist-packages (from ipython->autoviz) (4.4.2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dist-
packages (from ipython->autoviz) (0.7.5)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.10/dist-
packages (from ipython->autoviz) (5.7.1)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from ipython->autoviz) (3.0.39)
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-
```

```
packages (from ipython->autoviz) (2.16.1)
Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-
packages (from ipython->autoviz) (0.2.0)
Requirement already satisfied: matplotlib-inline in
/usr/local/lib/python3.10/dist-packages (from ipython->autoviz) (0.1.6)
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dist-
packages (from ipython->autoviz) (4.8.0)
Requirement already satisfied: notebook in /usr/local/lib/python3.10/dist-
packages (from jupyter->autoviz) (6.5.5)
Collecting qtconsole (from jupyter->autoviz)
  Downloading qtconsole-5.4.3-py3-none-any.whl (121 kB)
                           121.9/121.9
kB 9.2 MB/s eta 0:00:00
Requirement already satisfied: jupyter-console in
/usr/local/lib/python3.10/dist-packages (from jupyter->autoviz) (6.1.0)
Requirement already satisfied: nbconvert in /usr/local/lib/python3.10/dist-
packages (from jupyter->autoviz) (6.5.4)
Requirement already satisfied: ipykernel in /usr/local/lib/python3.10/dist-
packages (from jupyter->autoviz) (5.5.6)
Requirement already satisfied: ipywidgets in /usr/local/lib/python3.10/dist-
packages (from jupyter->autoviz) (7.7.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages
(from nltk->autoviz) (8.1.7)
Requirement already satisfied: regex>=2021.8.3 in
/usr/local/lib/python3.10/dist-packages (from nltk->autoviz) (2023.6.3)
Requirement already satisfied: patsy>=0.5.2 in /usr/local/lib/python3.10/dist-
packages (from statsmodels->autoviz) (0.5.3)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in
/usr/local/lib/python3.10/dist-packages (from jedi>=0.16->ipython->autoviz)
(0.8.3)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from
Jinja2>=2.9->bokeh~=2.4.2->autoviz) (2.1.3)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages
(from patsy>=0.5.2->statsmodels->autoviz) (1.16.0)
Requirement already satisfied: ptyprocess>=0.5 in
/usr/local/lib/python3.10/dist-packages (from pexpect>4.3->ipython->autoviz)
(0.7.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-
packages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython->autoviz)
(0.2.6)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-
packages (from bleach->panel>=0.12.6->autoviz) (0.5.1)
Requirement already satisfied: ipython-genutils in
/usr/local/lib/python3.10/dist-packages (from ipykernel->jupyter->autoviz)
(0.2.0)
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.10/dist-
```

```
packages (from ipykernel->jupyter->autoviz) (6.1.12)
Requirement already satisfied: widgetsnbextension~=3.6.0 in
/usr/local/lib/python3.10/dist-packages (from ipywidgets->jupyter->autoviz)
(3.6.5)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in
/usr/local/lib/python3.10/dist-packages (from ipywidgets->jupyter->autoviz)
(3.0.8)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packages
(from nbconvert->jupyter->autoviz) (4.9.3)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-
packages (from nbconvert->jupyter->autoviz) (4.11.2)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.10/dist-
packages (from nbconvert->jupyter->autoviz) (0.7.1)
Requirement already satisfied: entrypoints>=0.2.2 in
/usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter->autoviz) (0.4)
Requirement already satisfied: jupyter-core>=4.7 in
/usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter->autoviz)
(5.3.1)
Requirement already satisfied: jupyterlab-pygments in
/usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter->autoviz)
(0.2.2)
Requirement already satisfied: mistune<2,>=0.8.1 in
/usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter->autoviz)
(0.8.4)
Requirement already satisfied: nbclient>=0.5.0 in
/usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter->autoviz)
(0.8.0)
Requirement already satisfied: nbformat>=5.1 in /usr/local/lib/python3.10/dist-
packages (from nbconvert->jupyter->autoviz) (5.9.2)
Requirement already satisfied: pandocfilters>=1.4.1 in
/usr/local/lib/python3.10/dist-packages (from nbconvert->jupyter->autoviz)
(1.5.0)
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.10/dist-
packages (from nbconvert->jupyter->autoviz) (1.2.1)
Requirement already satisfied: pyzmq<25,>=17 in /usr/local/lib/python3.10/dist-
packages (from notebook->jupyter->autoviz) (23.2.1)
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.10/dist-
packages (from notebook->jupyter->autoviz) (23.1.0)
Requirement already satisfied: nest-asyncio>=1.5 in
/usr/local/lib/python3.10/dist-packages (from notebook->jupyter->autoviz)
(1.5.7)
Requirement already satisfied: Send2Trash>=1.8.0 in
/usr/local/lib/python3.10/dist-packages (from notebook->jupyter->autoviz)
(1.8.2)
Requirement already satisfied: terminado>=0.8.3 in
/usr/local/lib/python3.10/dist-packages (from notebook->jupyter->autoviz)
(0.17.1)
Requirement already satisfied: prometheus-client in
```

```
/usr/local/lib/python3.10/dist-packages (from notebook->jupyter->autoviz)
(0.17.1)
Requirement already satisfied: nbclassic>=0.4.7 in
/usr/local/lib/python3.10/dist-packages (from notebook->jupyter->autoviz)
(1.0.0)
Collecting qtpy>=2.0.1 (from qtconsole->jupyter->autoviz)
  Downloading QtPy-2.3.1-py3-none-any.whl (84 kB)
                           84.9/84.9 kB
9.6 MB/s eta 0:00:00
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->panel>=0.12.6->autoviz)
(3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests->panel>=0.12.6->autoviz) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->panel>=0.12.6->autoviz)
(2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->panel>=0.12.6->autoviz)
(2023.7.22)
Requirement already satisfied: platformdirs>=2.5 in
/usr/local/lib/python3.10/dist-packages (from jupyter-
core>=4.7->nbconvert->jupyter->autoviz) (3.10.0)
Requirement already satisfied: jupyter-server>=1.8 in
/usr/local/lib/python3.10/dist-packages (from
nbclassic>=0.4.7->notebook->jupyter->autoviz) (1.24.0)
Requirement already satisfied: notebook-shim>=0.2.3 in
/usr/local/lib/python3.10/dist-packages (from
nbclassic>=0.4.7->notebook->jupyter->autoviz) (0.2.3)
Requirement already satisfied: fastjsonschema in /usr/local/lib/python3.10/dist-
packages (from nbformat>=5.1->nbconvert->jupyter->autoviz) (2.18.0)
Requirement already satisfied: jsonschema>=2.6 in
/usr/local/lib/python3.10/dist-packages (from
nbformat>=5.1->nbconvert->jupyter->autoviz) (4.19.0)
Requirement already satisfied: argon2-cffi-bindings in
/usr/local/lib/python3.10/dist-packages (from
argon2-cffi->notebook->jupyter->autoviz) (21.2.0)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-
packages (from beautifulsoup4->nbconvert->jupyter->autoviz) (2.4.1)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/dist-
packages (from jsonschema>=2.6->nbformat>=5.1->nbconvert->jupyter->autoviz)
(23.1.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
/usr/local/lib/python3.10/dist-packages (from
jsonschema>=2.6->nbformat>=5.1->nbconvert->jupyter->autoviz) (2023.7.1)
Requirement already satisfied: referencing>=0.28.4 in
/usr/local/lib/python3.10/dist-packages (from
jsonschema>=2.6->nbformat>=5.1->nbconvert->jupyter->autoviz) (0.30.2)
```

```
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-
packages (from jsonschema>=2.6->nbformat>=5.1->nbconvert->jupyter->autoviz)
(0.9.2)
Requirement already satisfied: anyio<4,>=3.1.0 in
/usr/local/lib/python3.10/dist-packages (from jupyter-
server>=1.8->nbclassic>=0.4.7->notebook->jupyter->autoviz) (3.7.1)
Requirement already satisfied: websocket-client in
/usr/local/lib/python3.10/dist-packages (from jupyter-
server>=1.8->nbclassic>=0.4.7->notebook->jupyter->autoviz) (1.6.1)
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.10/dist-
packages (from argon2-cffi-bindings->argon2-cffi->notebook->jupyter->autoviz)
(1.15.1)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-
packages (from anyio<4,>=3.1.0->jupyter-
server>=1.8->nbclassic>=0.4.7->notebook->jupyter->autoviz) (1.3.0)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-
packages (from anyio<4,>=3.1.0->jupyter-
server>=1.8->nbclassic>=0.4.7->notebook->jupyter->autoviz) (1.1.3)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-
packages (from cffi>=1.0.1->argon2-cffi-
bindings->argon2-cffi->notebook->jupyter->autoviz) (2.21)
Installing collected packages: qtpy, jedi, emoji, pyamg, bokeh, panel, pandas-
dq, qtconsole, holoviews, hvplot, jupyter, autoviz
  Attempting uninstall: bokeh
    Found existing installation: bokeh 3.2.2
    Uninstalling bokeh-3.2.2:
      Successfully uninstalled bokeh-3.2.2
  Attempting uninstall: panel
    Found existing installation: panel 1.2.1
    Uninstalling panel-1.2.1:
      Successfully uninstalled panel-1.2.1
  Attempting uninstall: holoviews
    Found existing installation: holoviews 1.17.1
    Uninstalling holoviews-1.17.1:
      Successfully uninstalled holoviews-1.17.1
Successfully installed autoviz-0.1.730 bokeh-2.4.3 emoji-2.8.0 holoviews-1.14.9
hvplot-0.7.3 jedi-0.19.0 jupyter-1.0.0 pandas-dq-1.28 panel-0.14.4 pyamg-5.0.1
qtconsole-5.4.3 qtpy-2.3.1
```

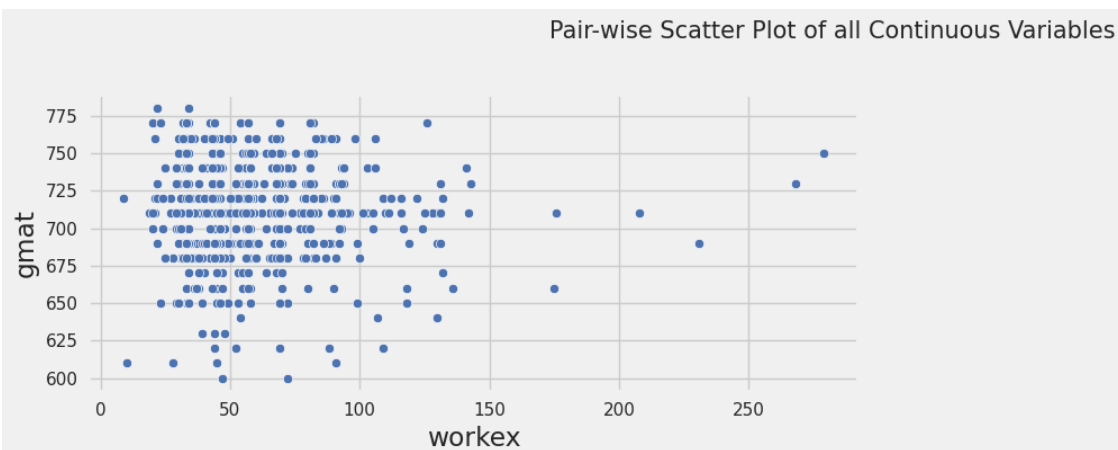
[ ]:
```python
# Autoviz
###########
# pip install autoviz
from autoviz.AutoViz_Class import AutoViz_Class

av = AutoViz_Class()
%matplotlib inline
# a = av.AutoViz(r"education.csv", chart_format = 'html')
```

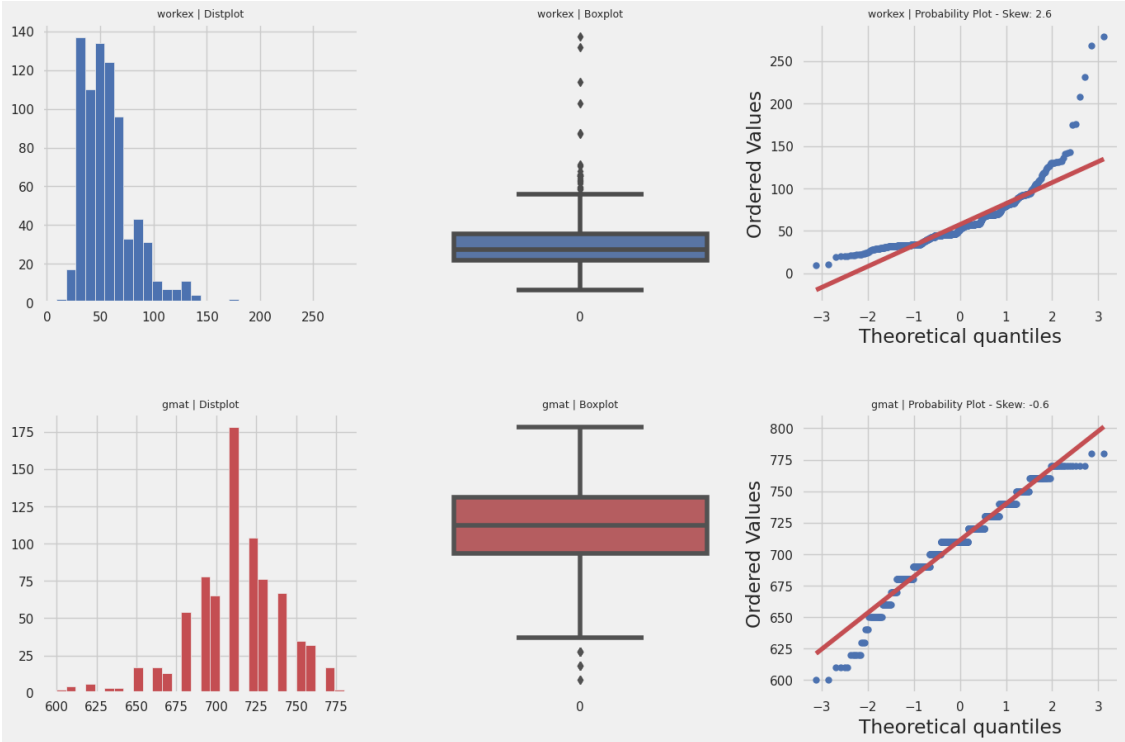
```
a = av.AutoViz(r"education.csv")
```

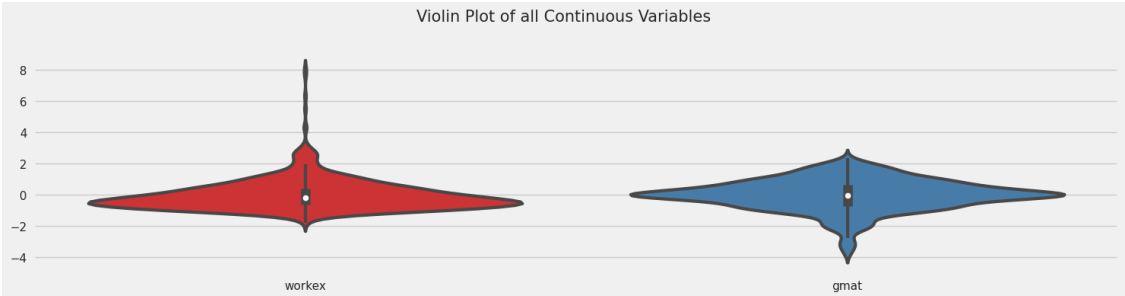
```
Imported v0.1.730. After importing autoviz, execute '%matplotlib inline' to
display charts inline.
    AV = AutoViz_Class()
    dfte = AV.AutoViz(filename, sep=',', depVar='', dfte=None, header=0,
verbose=1, lowess=False,
                chart_format='svg',max_rows_analyzed=150000,max_cols_analyzed=30,
save_plot_dir=None)
Shape of your Data Set loaded: (773, 3)
##############################################################################
#######
######################### C L A S S I F Y I N G  V A R I A B L E S
####################
##############################################################################
#######
Classifying variables in data set…
    3 Predictors classified…
        1 variable(s) removed since they were ID or low-information variables
        List of variables removed: ['datasrno']
To fix data quality issues automatically, import FixDQ from autoviz…
```

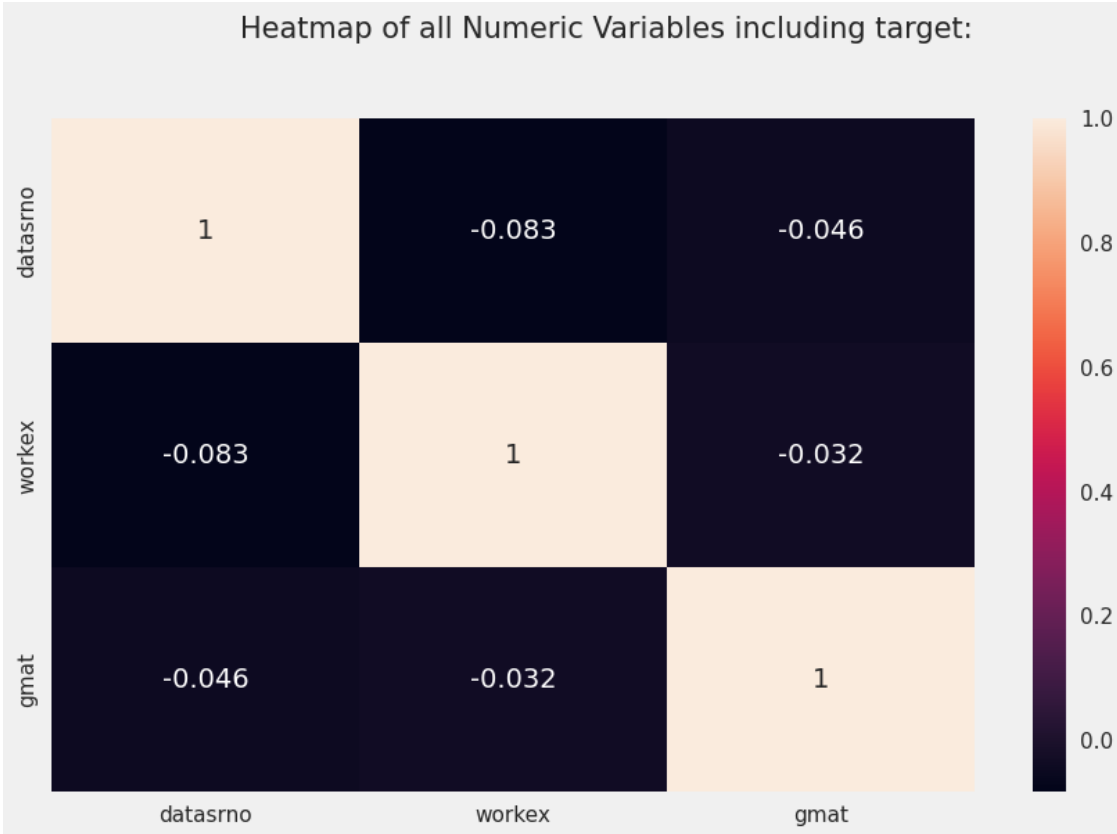
```
<pandas.io.formats.style.Styler at 0x7a341de9be50>
```

```
Number of All Scatter Plots = 3
```




workex | Distplot
workex | Boxplot
workex | Probability Plot - Skew: 2.6
Ordered Values
Theoretical quantiles
gmat | Distplot
gmat | Boxplot
gmat | Probability Plot - Skew: -0.6
Ordered Values
Theoretical quantiles


Violin Plot of all Continuous Variables
workex
gmat



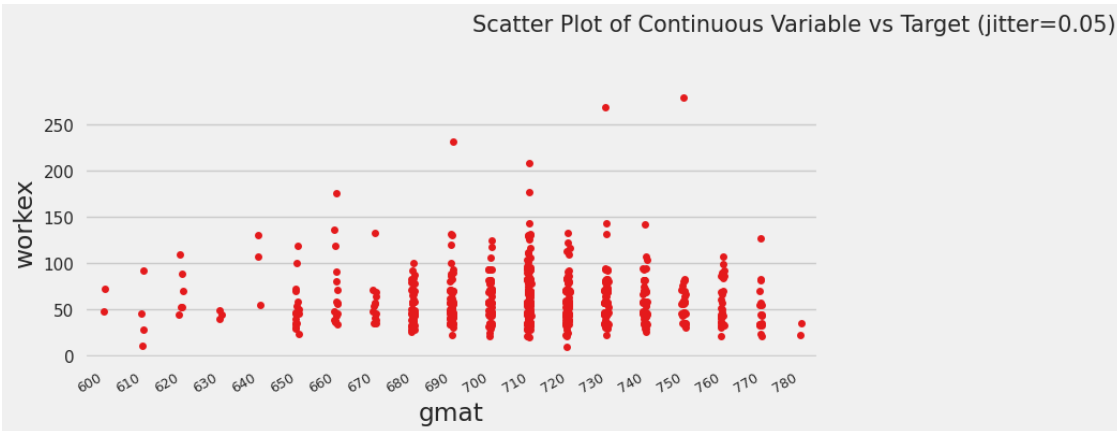

```
All Plots done
Time to run AutoViz = 9 seconds

 ###################### AUTO VISUALIZATION Completed ######################
```

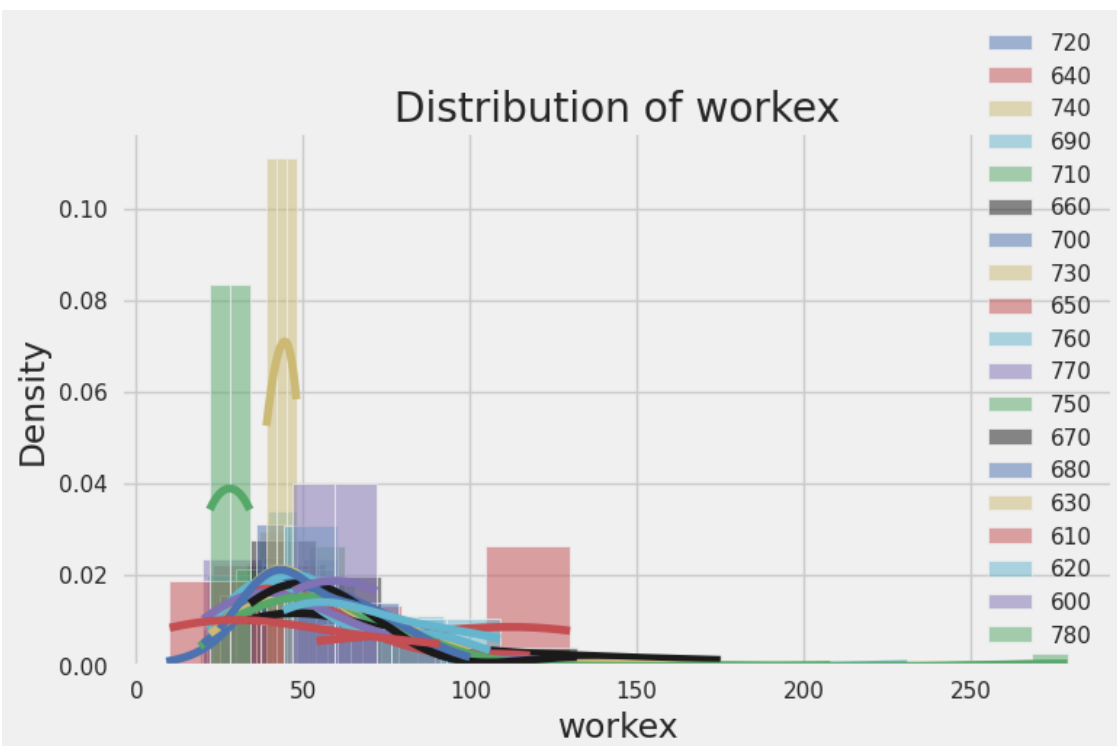
[ ]:
```
import os
os.getcwd()

# If the dependent variable is known:
a = av.AutoViz(r"education.csv", depVar = 'gmat') # depVar - target variable in␣
 ↪your dataset
%matplotlib inline
```

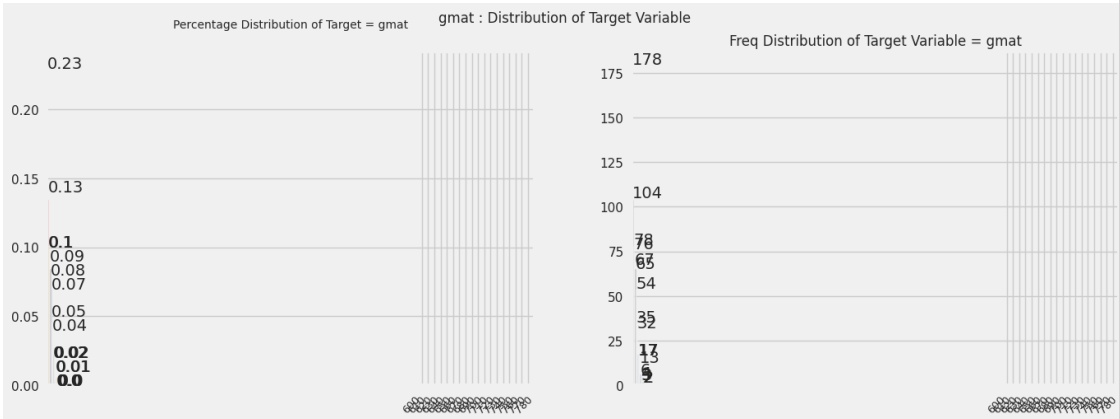
```
Shape of your Data Set loaded: (773, 3)
################################################################################
#######
######################## C L A S S I F Y I N G  V A R I A B L E S
####################
################################################################################
#######
Classifying variables in data set…
```

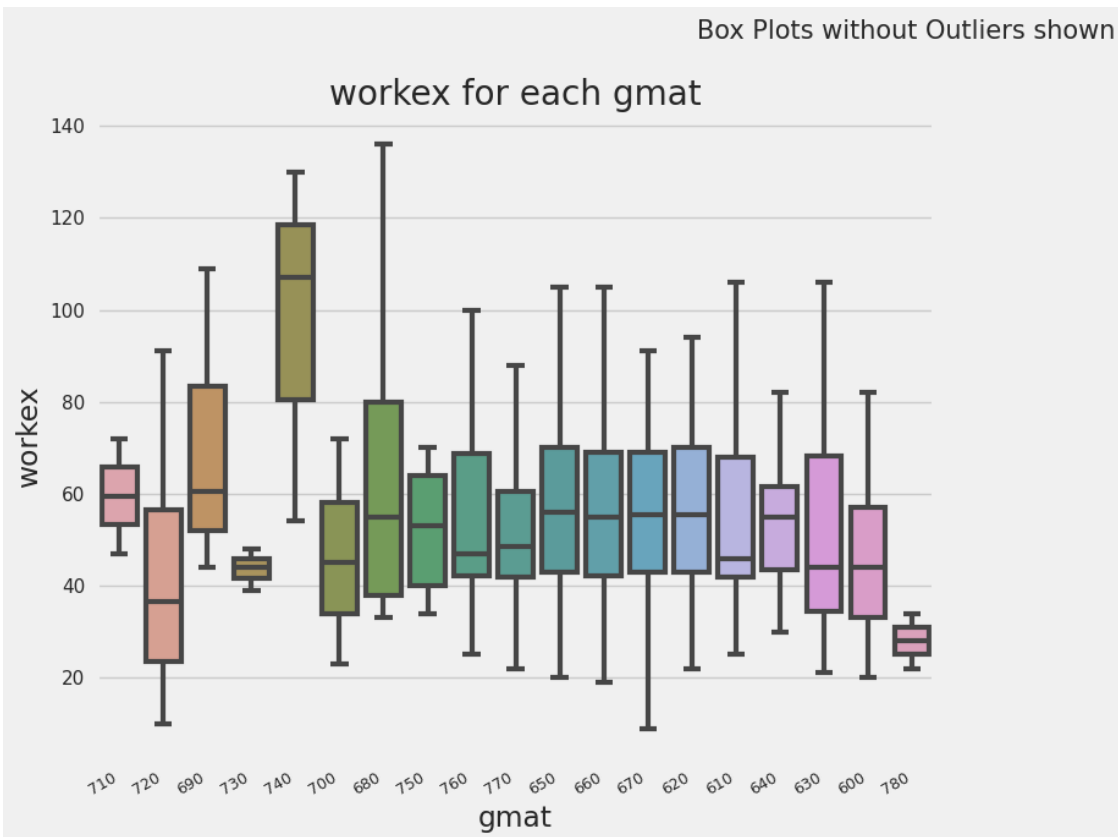
```
    2 Predictors classified…
        1 variable(s) removed since they were ID or low-information variables
        List of variables removed: ['datasrno']

################ Multi_Classification problem #####################
To fix data quality issues automatically, import FixDQ from autoviz…
    Alert: Dropping 369 duplicate rows can sometimes cause column data types to
change to object. Double-check!
```

```
<pandas.io.formats.style.Styler at 0x7a3439d61960>
```










```
All Plots done
Time to run AutoViz = 4 seconds

 ###################### AUTO VISUALIZATION Completed ######################
```

[ ]:
```
# D-Tale
########

!pip install dtale   # In case of any error then please install werkzeug␣
↪appropriate version (pip install werkzeug==2.0.3)
```

```
Collecting dtale
  Downloading dtale-3.3.0-py2.py3-none-any.whl (14.2 MB)
                            14.2/14.2 MB
27.6 MB/s eta 0:00:00
Collecting dash-colorscales (from dtale)
  Downloading dash_colorscales-0.0.4.tar.gz (62 kB)
                            62.3/62.3 kB
7.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting dash-daq (from dtale)
  Downloading dash_daq-0.5.0.tar.gz (642 kB)
                           642.7/642.7 kB
36.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting Flask-Compress (from dtale)
  Downloading Flask_Compress-1.13-py3-none-any.whl (7.9 kB)
Requirement already satisfied: future>=0.14.0 in /usr/local/lib/python3.10/dist-
packages (from dtale) (0.18.3)
Collecting kaleido (from dtale)
  Downloading kaleido-0.2.1-py2.py3-none-manylinux1_x86_64.whl (79.9 MB)
                            79.9/79.9 MB
9.1 MB/s eta 0:00:00
Requirement already satisfied: missingno in
/usr/local/lib/python3.10/dist-packages (from dtale) (0.5.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages
(from dtale) (1.5.3)
Collecting squarify (from dtale)
  Downloading squarify-0.4.3-py3-none-any.whl (4.3 kB)
Collecting strsimpy (from dtale)
  Downloading strsimpy-0.2.1-py3-none-any.whl (45 kB)
                            45.9/45.9 kB
5.9 MB/s eta 0:00:00
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-
packages (from dtale) (1.16.0)
Requirement already satisfied: xlrd in /usr/local/lib/python3.10/dist-packages
(from dtale) (2.0.1)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-
packages (from dtale) (4.11.2)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-
packages (from dtale) (2023.7.22)
Requirement already satisfied: cycler in /usr/local/lib/python3.10/dist-packages
```

```
(from dtale) (0.11.0)
Collecting flask-ngrok (from dtale)
  Downloading flask_ngrok-0.0.25-py3-none-any.whl (3.1 kB)
Collecting lz4 (from dtale)
  Downloading
lz4-4.3.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.3 MB)
                           1.3/1.3 MB
71.6 MB/s eta 0:00:00
Collecting dash-bootstrap-components<=1.3.1 (from dtale)
  Downloading dash_bootstrap_components-1.3.1-py3-none-any.whl (219 kB)
                          219.7/219.7 kB
24.1 MB/s eta 0:00:00
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-
packages (from dtale) (0.12.2)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-
packages (from dtale) (3.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-
packages (from dtale) (1.2.2)
Requirement already satisfied: statsmodels in /usr/local/lib/python3.10/dist-
packages (from dtale) (0.14.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(from dtale) (1.23.5)
Requirement already satisfied: openpyxl!=3.2.0b1 in
/usr/local/lib/python3.10/dist-packages (from dtale) (3.1.2)
Requirement already satisfied: xarray in /usr/local/lib/python3.10/dist-packages
(from dtale) (2023.7.0)
Collecting dash (from dtale)
  Downloading dash-2.12.1-py3-none-any.whl (10.4 MB)
                           10.4/10.4 MB
87.7 MB/s eta 0:00:00
Requirement already satisfied: et-xmlfile in
/usr/local/lib/python3.10/dist-packages (from dtale) (1.1.0)
Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packages
(from dtale) (5.15.0)
Requirement already satisfied: Flask<2.3 in /usr/local/lib/python3.10/dist-
packages (from dtale) (2.2.5)
Requirement already satisfied: itsdangerous in /usr/local/lib/python3.10/dist-
packages (from dtale) (2.1.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from dtale) (2.31.0)
Collecting werkzeug<2.3 (from dtale)
  Downloading Werkzeug-2.2.3-py3-none-any.whl (233 kB)
                          233.6/233.6 kB
25.5 MB/s eta 0:00:00
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.10/dist-packages (from dtale) (3.7.1)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages
(from dtale) (1.10.1)
```

```
Collecting dash-html-components==2.0.0 (from dash->dtale)
  Downloading dash_html_components-2.0.0-py3-none-any.whl (4.1 kB)
Collecting dash-core-components==2.0.0 (from dash->dtale)
  Downloading dash_core_components-2.0.0-py3-none-any.whl (3.8 kB)
Collecting dash-table==5.0.0 (from dash->dtale)
  Downloading dash_table-5.0.0-py3-none-any.whl (3.9 kB)
Requirement already satisfied: typing-extensions>=4.1.1 in
/usr/local/lib/python3.10/dist-packages (from dash->dtale) (4.7.1)
Collecting retrying (from dash->dtale)
  Downloading retrying-1.3.4-py3-none-any.whl (11 kB)
Collecting ansi2html (from dash->dtale)
  Downloading ansi2html-1.8.0-py3-none-any.whl (16 kB)
Requirement already satisfied: nest-asyncio in /usr/local/lib/python3.10/dist-
packages (from dash->dtale) (1.5.7)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-
packages (from dash->dtale) (67.7.2)
Requirement already satisfied: Jinja2>=3.0 in /usr/local/lib/python3.10/dist-
packages (from Flask<2.3->dtale) (3.1.2)
Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.10/dist-
packages (from Flask<2.3->dtale) (8.1.7)
Requirement already satisfied: tenacity>=6.2.0 in
/usr/local/lib/python3.10/dist-packages (from plotly->dtale) (8.2.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from plotly->dtale) (23.1)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.10/dist-packages (from werkzeug<2.3->dtale) (2.1.3)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-
packages (from beautifulsoup4->dtale) (2.4.1)
Collecting brotli (from Flask-Compress->dtale)
  Downloading Brotli-1.0.9-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.ma
nylinux_2_12_x86_64.manylinux2010_x86_64.whl (2.7 MB)
                           2.7/2.7 MB
88.1 MB/s eta 0:00:00
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->dtale) (1.1.0)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->dtale) (4.42.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->dtale) (1.4.4)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-
packages (from matplotlib->dtale) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->dtale) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->dtale) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas->dtale) (2023.3)
Requirement already satisfied: charset-normalizer<4,>=2 in
```

```
/usr/local/lib/python3.10/dist-packages (from requests->dtale) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests->dtale) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->dtale) (2.0.4)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn->dtale) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn->dtale) (3.2.0)
Requirement already satisfied: patsy>=0.5.2 in /usr/local/lib/python3.10/dist-
packages (from statsmodels->dtale) (0.5.3)
Building wheels for collected packages: dash-colorscales, dash-daq
  Building wheel for dash-colorscales (setup.py) … done
  Created wheel for dash-colorscales: filename=dash_colorscales-0.0.4-py3-none-
any.whl size=62566
sha256=cddc9bdbe8808293ad49036882e4b6a0dd4b0682581e2f7dc2b9b5e78d53b6d1
  Stored in directory: /root/.cache/pip/wheels/70/6a/1f/95b2135cd2c895f0cd8b5d6d
6ae7d5ed0a883580b34a31a14d
  Building wheel for dash-daq (setup.py) … done
  Created wheel for dash-daq: filename=dash_daq-0.5.0-py3-none-any.whl
size=669691
sha256=c56685c71eb09d69d2e722f664aca82a9085520d5fa796e408187dfea15aa183
  Stored in directory: /root/.cache/pip/wheels/75/14/1b/208d09d5e239391048bdc167
759977b41ba65a3d4063aebf6b
Successfully built dash-colorscales dash-daq
Installing collected packages: strsimpy, squarify, kaleido, dash-table, dash-
html-components, dash-core-components, dash-colorscales, brotli, werkzeug,
retrying, lz4, ansi2html, flask-ngrok, Flask-Compress, dash, dash-daq, dash-
bootstrap-components, dtale
  Attempting uninstall: werkzeug
    Found existing installation: Werkzeug 2.3.7
    Uninstalling Werkzeug-2.3.7:
      Successfully uninstalled Werkzeug-2.3.7
Successfully installed Flask-Compress-1.13 ansi2html-1.8.0 brotli-1.0.9
dash-2.12.1 dash-bootstrap-components-1.3.1 dash-colorscales-0.0.4 dash-core-
components-2.0.0 dash-daq-0.5.0 dash-html-components-2.0.0 dash-table-5.0.0
dtale-3.3.0 flask-ngrok-0.0.25 kaleido-0.2.1 lz4-4.3.2 retrying-1.3.4
squarify-0.4.3 strsimpy-0.2.1 werkzeug-2.2.3
```

[ ]:
```python
import dtale
import dtale.app as dtale_app
df = pd.read_csv(r"education.csv")
dtale_app.USE_COLAB = True
d = dtale.show(df,notebook=True)
# d.open_browser()
```

```
<IPython.lib.display.IFrame at 0x7a34139a7a60>
```

[ ]:
```
# Pandas Profiling
###################

!pip install pandas_profiling
```

```
Collecting pandas_profiling
  Downloading pandas_profiling-3.6.6-py2.py3-none-any.whl (324 kB)
                            324.4/324.4

kB 4.3 MB/s eta 0:00:00
Collecting ydata-profiling (from pandas_profiling)
  Downloading ydata_profiling-4.5.1-py2.py3-none-any.whl (357 kB)
                           357.3/357.3 kB
10.5 MB/s eta 0:00:00
Requirement already satisfied: scipy<1.12,>=1.4.1 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling)
(1.10.1)
Requirement already satisfied: pandas!=1.4.0,<2.1,>1.1 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling)
(1.5.3)
Requirement already satisfied: matplotlib<4,>=3.2 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling)
(3.7.1)
Collecting pydantic<2,>=1.8.1 (from ydata-profiling->pandas_profiling)
  Downloading
pydantic-1.10.12-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.1
MB)
                             3.1/3.1 MB
18.3 MB/s eta 0:00:00
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling)
(6.0.1)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling)
(3.1.2)
Collecting visions[type_image_path]==0.7.5 (from ydata-
profiling->pandas_profiling)
  Downloading visions-0.7.5-py3-none-any.whl (102 kB)
                           102.7/102.7 kB
11.6 MB/s eta 0:00:00
Requirement already satisfied: numpy<1.24,>=1.16.0 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling)
(1.23.5)
Collecting htmlmin==0.1.12 (from ydata-profiling->pandas_profiling)
  Downloading htmlmin-0.1.12.tar.gz (19 kB)
  Preparing metadata (setup.py) … done
Collecting phik<0.13,>=0.11.1 (from ydata-profiling->pandas_profiling)
  Downloading
```

```
phik-0.12.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (679 kB)
                           679.5/679.5 kB
22.7 MB/s eta 0:00:00
Requirement already satisfied: requests<3,>=2.24.0 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling)
(2.31.0)
Requirement already satisfied: tqdm<5,>=4.48.2 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling)
(4.66.1)
Requirement already satisfied: seaborn<0.13,>=0.10.1 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling)
(0.12.2)
Collecting multimethod<2,>=1.4 (from ydata-profiling->pandas_profiling)
  Downloading multimethod-1.9.1-py3-none-any.whl (10 kB)
Requirement already satisfied: statsmodels<1,>=0.13.2 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling)
(0.14.0)
Collecting typeguard<3,>=2.13.2 (from ydata-profiling->pandas_profiling)
  Downloading typeguard-2.13.3-py3-none-any.whl (17 kB)
Collecting imagehash==4.3.1 (from ydata-profiling->pandas_profiling)
  Downloading ImageHash-4.3.1-py2.py3-none-any.whl (296 kB)
                           296.5/296.5 kB
17.6 MB/s eta 0:00:00
Requirement already satisfied: wordcloud>=1.9.1 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling)
(1.9.2)
Collecting dacite>=1.8 (from ydata-profiling->pandas_profiling)
  Downloading dacite-1.8.1-py3-none-any.whl (14 kB)
Requirement already satisfied: PyWavelets in /usr/local/lib/python3.10/dist-
packages (from imagehash==4.3.1->ydata-profiling->pandas_profiling) (1.4.1)
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages
(from imagehash==4.3.1->ydata-profiling->pandas_profiling) (9.4.0)
Requirement already satisfied: attrs>=19.3.0 in /usr/local/lib/python3.10/dist-
packages (from visions[type_image_path]==0.7.5->ydata-
profiling->pandas_profiling) (23.1.0)
Requirement already satisfied: networkx>=2.4 in /usr/local/lib/python3.10/dist-
packages (from visions[type_image_path]==0.7.5->ydata-
profiling->pandas_profiling) (3.1)
Collecting tangled-up-in-unicode>=0.0.4 (from
visions[type_image_path]==0.7.5->ydata-profiling->pandas_profiling)
  Downloading tangled_up_in_unicode-0.2.0-py3-none-any.whl (4.7 MB)
                            4.7/4.7 MB
38.9 MB/s eta 0:00:00
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2<3.2,>=2.11.1->ydata-
profiling->pandas_profiling) (2.1.3)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib<4,>=3.2->ydata-
```

```
profiling->pandas_profiling) (1.1.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
packages (from matplotlib<4,>=3.2->ydata-profiling->pandas_profiling) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib<4,>=3.2->ydata-
profiling->pandas_profiling) (4.42.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib<4,>=3.2->ydata-
profiling->pandas_profiling) (1.4.4)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib<4,>=3.2->ydata-
profiling->pandas_profiling) (23.1)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib<4,>=3.2->ydata-
profiling->pandas_profiling) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib<4,>=3.2->ydata-
profiling->pandas_profiling) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas!=1.4.0,<2.1,>1.1->ydata-profiling->pandas_profiling)
(2023.3)
Requirement already satisfied: joblib>=0.14.1 in /usr/local/lib/python3.10/dist-
packages (from phik<0.13,>=0.11.1->ydata-profiling->pandas_profiling) (1.3.2)
Requirement already satisfied: typing-extensions>=4.2.0 in
/usr/local/lib/python3.10/dist-packages (from pydantic<2,>=1.8.1->ydata-
profiling->pandas_profiling) (4.7.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.24.0->ydata-
profiling->pandas_profiling) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests<3,>=2.24.0->ydata-profiling->pandas_profiling) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.24.0->ydata-
profiling->pandas_profiling) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.24.0->ydata-
profiling->pandas_profiling) (2023.7.22)
Requirement already satisfied: patsy>=0.5.2 in /usr/local/lib/python3.10/dist-
packages (from statsmodels<1,>=0.13.2->ydata-profiling->pandas_profiling)
(0.5.3)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages
(from patsy>=0.5.2->statsmodels<1,>=0.13.2->ydata-profiling->pandas_profiling)
(1.16.0)
Building wheels for collected packages: htmlmin
  Building wheel for htmlmin (setup.py) … done
  Created wheel for htmlmin: filename=htmlmin-0.1.12-py3-none-any.whl size=27082
sha256=149c8928847f1a145e09d9e14166e516bfefc3dbfbbe6292dbebfa532ed9e845
  Stored in directory: /root/.cache/pip/wheels/dd/91/29/a79cecb328d01739e64017b6
```

```
fb9a1ab9d8cb1853098ec5966d
Successfully built htmlmin
Installing collected packages: htmlmin, typeguard, tangled-up-in-unicode,
pydantic, multimethod, dacite, imagehash, visions, phik, ydata-profiling,
pandas_profiling
  Attempting uninstall: pydantic
    Found existing installation: pydantic 2.2.0
    Uninstalling pydantic-2.2.0:
      Successfully uninstalled pydantic-2.2.0
Successfully installed dacite-1.8.1 htmlmin-0.1.12 imagehash-4.3.1
multimethod-1.9.1 pandas_profiling-3.6.6 phik-0.12.3 pydantic-1.10.12 tangled-
up-in-unicode-0.2.0 typeguard-2.13.3 visions-0.7.5 ydata-profiling-4.5.1
```

[ ]:
```python
from pandas_profiling import ProfileReport

p = ProfileReport(df)
p
```

```
2023-08-25 07:14:58,130 - INFO     - Pandas backend loaded 1.5.3
2023-08-25 07:14:58,143 - INFO     - Numpy backend loaded 1.23.5
2023-08-25 07:14:58,146 - INFO     - Pyspark backend NOT loaded
2023-08-25 07:14:58,149 - INFO     - Python backend loaded

Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]

<IPython.core.display.HTML object>
```

[ ]:

[ ]:
```python
# Dataprep
##########

!pip install dataprep
```

```
Collecting dataprep
  Downloading dataprep-0.4.5-py3-none-any.whl (9.9 MB)
                           9.9/9.9 MB
19.0 MB/s eta 0:00:00
Requirement already satisfied: aiohttp<4.0,>=3.6 in
/usr/local/lib/python3.10/dist-packages (from dataprep) (3.8.5)
Requirement already satisfied: bokeh<3,>=2 in /usr/local/lib/python3.10/dist-
packages (from dataprep) (2.4.3)
Requirement already satisfied: dask[array,dataframe,delayed]>=2022.3.0 in
/usr/local/lib/python3.10/dist-packages (from dataprep) (2023.8.0)
Requirement already satisfied: flask<3,>=2 in /usr/local/lib/python3.10/dist-
packages (from dataprep) (2.2.5)
```

```
Collecting flask_cors<4.0.0,>=3.0.10 (from dataprep)
  Downloading Flask_Cors-3.0.10-py2.py3-none-any.whl (14 kB)
Requirement already satisfied: ipywidgets<8.0,>=7.5 in
/usr/local/lib/python3.10/dist-packages (from dataprep) (7.7.1)
Collecting jinja2<3.1,>=3.0 (from dataprep)
  Downloading Jinja2-3.0.3-py3-none-any.whl (133 kB)
                           133.6/133.6 kB
11.6 MB/s eta 0:00:00
Collecting jsonpath-ng<2.0,>=1.5 (from dataprep)
  Downloading jsonpath_ng-1.5.3-py3-none-any.whl (29 kB)
Collecting metaphone<0.7,>=0.6 (from dataprep)
  Downloading Metaphone-0.6.tar.gz (14 kB)
  Preparing metadata (setup.py) … done
Requirement already satisfied: nltk<4.0.0,>=3.6.7 in
/usr/local/lib/python3.10/dist-packages (from dataprep) (3.8.1)
Requirement already satisfied: numpy<2.0,>=1.21 in
/usr/local/lib/python3.10/dist-packages (from dataprep) (1.23.5)
Requirement already satisfied: pandas<2.0,>=1.1 in
/usr/local/lib/python3.10/dist-packages (from dataprep) (1.5.3)
Requirement already satisfied: pydantic<2.0,>=1.6 in
/usr/local/lib/python3.10/dist-packages (from dataprep) (1.10.12)
Requirement already satisfied: pydot<2.0.0,>=1.4.2 in
/usr/local/lib/python3.10/dist-packages (from dataprep) (1.4.2)
Collecting python-crfsuite==0.9.8 (from dataprep)
  Downloading
python_crfsuite-0.9.8-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(1.0 MB)
                            1.0/1.0 MB
27.5 MB/s eta 0:00:00
Collecting python-stdnum<2.0,>=1.16 (from dataprep)
  Downloading python_stdnum-1.19-py2.py3-none-any.whl (1.0 MB)
                            1.0/1.0 MB
40.7 MB/s eta 0:00:00
Collecting rapidfuzz<3.0.0,>=2.1.2 (from dataprep)
  Downloading
rapidfuzz-2.15.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.0
MB)
                            3.0/3.0 MB
30.4 MB/s eta 0:00:00
Collecting regex<2022.0.0,>=2021.8.3 (from dataprep)
  Downloading
regex-2021.11.10-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (764
kB)
                           764.0/764.0 kB
31.0 MB/s eta 0:00:00
Requirement already satisfied: scipy<2.0,>=1.8 in
/usr/local/lib/python3.10/dist-packages (from dataprep) (1.10.1)
Collecting sqlalchemy==1.3.24 (from dataprep)
```

```
  Downloading SQLAlchemy-1.3.24.tar.gz (6.4 MB)
                             6.4/6.4 MB
55.4 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Requirement already satisfied: tqdm<5.0,>=4.48 in
/usr/local/lib/python3.10/dist-packages (from dataprep) (4.66.1)
Collecting varname<0.9.0,>=0.8.1 (from dataprep)
  Downloading varname-0.8.3-py3-none-any.whl (21 kB)
Requirement already satisfied: wordcloud<2.0,>=1.8 in
/usr/local/lib/python3.10/dist-packages (from dataprep) (1.9.2)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-
packages (from aiohttp<4.0,>=3.6->dataprep) (23.1.0)
Requirement already satisfied: charset-normalizer<4.0,>=2.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp<4.0,>=3.6->dataprep)
(3.2.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.10/dist-packages (from aiohttp<4.0,>=3.6->dataprep)
(6.0.4)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in
/usr/local/lib/python3.10/dist-packages (from aiohttp<4.0,>=3.6->dataprep)
(4.0.3)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-
packages (from aiohttp<4.0,>=3.6->dataprep) (1.9.2)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from aiohttp<4.0,>=3.6->dataprep)
(1.4.0)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.10/dist-packages (from aiohttp<4.0,>=3.6->dataprep)
(1.3.1)
Requirement already satisfied: packaging>=16.8 in
/usr/local/lib/python3.10/dist-packages (from bokeh<3,>=2->dataprep) (23.1)
Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/python3.10/dist-
packages (from bokeh<3,>=2->dataprep) (9.4.0)
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/python3.10/dist-
packages (from bokeh<3,>=2->dataprep) (6.0.1)
Requirement already satisfied: tornado>=5.1 in /usr/local/lib/python3.10/dist-
packages (from bokeh<3,>=2->dataprep) (6.3.2)
Requirement already satisfied: typing-extensions>=3.10.0 in
/usr/local/lib/python3.10/dist-packages (from bokeh<3,>=2->dataprep) (4.7.1)
Requirement already satisfied: click>=8.0 in /usr/local/lib/python3.10/dist-
packages (from dask[array,dataframe,delayed]>=2022.3.0->dataprep) (8.1.7)
Requirement already satisfied: cloudpickle>=1.5.0 in
/usr/local/lib/python3.10/dist-packages (from
dask[array,dataframe,delayed]>=2022.3.0->dataprep) (2.2.1)
Requirement already satisfied: fsspec>=2021.09.0 in
/usr/local/lib/python3.10/dist-packages (from
dask[array,dataframe,delayed]>=2022.3.0->dataprep) (2023.6.0)
Requirement already satisfied: partd>=1.2.0 in /usr/local/lib/python3.10/dist-
```

```
packages (from dask[array,dataframe,delayed]>=2022.3.0->dataprep) (1.4.0)
Requirement already satisfied: toolz>=0.10.0 in /usr/local/lib/python3.10/dist-
packages (from dask[array,dataframe,delayed]>=2022.3.0->dataprep) (0.12.0)
Requirement already satisfied: importlib-metadata>=4.13.0 in
/usr/local/lib/python3.10/dist-packages (from
dask[array,dataframe,delayed]>=2022.3.0->dataprep) (6.8.0)
Requirement already satisfied: Werkzeug>=2.2.2 in
/usr/local/lib/python3.10/dist-packages (from flask<3,>=2->dataprep) (2.2.3)
Requirement already satisfied: itsdangerous>=2.0 in
/usr/local/lib/python3.10/dist-packages (from flask<3,>=2->dataprep) (2.1.2)
Requirement already satisfied: Six in /usr/local/lib/python3.10/dist-packages
(from flask_cors<4.0.0,>=3.0.10->dataprep) (1.16.0)
Requirement already satisfied: ipykernel>=4.5.1 in
/usr/local/lib/python3.10/dist-packages (from ipywidgets<8.0,>=7.5->dataprep)
(5.5.6)
Requirement already satisfied: ipython-genutils~=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from ipywidgets<8.0,>=7.5->dataprep)
(0.2.0)
Requirement already satisfied: traitlets>=4.3.1 in
/usr/local/lib/python3.10/dist-packages (from ipywidgets<8.0,>=7.5->dataprep)
(5.7.1)
Requirement already satisfied: widgetsnbextension~=3.6.0 in
/usr/local/lib/python3.10/dist-packages (from ipywidgets<8.0,>=7.5->dataprep)
(3.6.5)
Requirement already satisfied: ipython>=4.0.0 in /usr/local/lib/python3.10/dist-
packages (from ipywidgets<8.0,>=7.5->dataprep) (7.34.0)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in
/usr/local/lib/python3.10/dist-packages (from ipywidgets<8.0,>=7.5->dataprep)
(3.0.8)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2<3.1,>=3.0->dataprep)
(2.1.3)
Collecting ply (from jsonpath-ng<2.0,>=1.5->dataprep)
  Downloading ply-3.11-py2.py3-none-any.whl (49 kB)
                           49.6/49.6 kB
5.4 MB/s eta 0:00:00
Requirement already satisfied: decorator in
/usr/local/lib/python3.10/dist-packages (from jsonpath-ng<2.0,>=1.5->dataprep)
(4.4.2)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages
(from nltk<4.0.0,>=3.6.7->dataprep) (1.3.2)
Requirement already satisfied: python-dateutil>=2.8.1 in
/usr/local/lib/python3.10/dist-packages (from pandas<2.0,>=1.1->dataprep)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas<2.0,>=1.1->dataprep) (2023.3)
Requirement already satisfied: pyparsing>=2.1.4 in
/usr/local/lib/python3.10/dist-packages (from pydot<2.0.0,>=1.4.2->dataprep)
```

```
(3.1.1)
Collecting asttokens<3.0.0,>=2.0.0 (from varname<0.9.0,>=0.8.1->dataprep)
  Downloading asttokens-2.2.1-py2.py3-none-any.whl (26 kB)
Collecting executing<0.9.0,>=0.8.3 (from varname<0.9.0,>=0.8.1->dataprep)
  Downloading executing-0.8.3-py2.py3-none-any.whl (16 kB)
Collecting pure_eval<1.0.0 (from varname<0.9.0,>=0.8.1->dataprep)
  Downloading pure_eval-0.2.2-py3-none-any.whl (11 kB)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-
packages (from wordcloud<2.0,>=1.8->dataprep) (3.7.1)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.10/dist-
packages (from importlib-
metadata>=4.13.0->dask[array,dataframe,delayed]>=2022.3.0->dataprep) (3.16.2)
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.10/dist-
packages (from ipykernel>=4.5.1->ipywidgets<8.0,>=7.5->dataprep) (6.1.12)
Requirement already satisfied: setuptools>=18.5 in
/usr/local/lib/python3.10/dist-packages (from
ipython>=4.0.0->ipywidgets<8.0,>=7.5->dataprep) (67.7.2)
Requirement already satisfied: jedi>=0.16 in /usr/local/lib/python3.10/dist-
packages (from ipython>=4.0.0->ipywidgets<8.0,>=7.5->dataprep) (0.19.0)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.10/dist-
packages (from ipython>=4.0.0->ipywidgets<8.0,>=7.5->dataprep) (0.7.5)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from
ipython>=4.0.0->ipywidgets<8.0,>=7.5->dataprep) (3.0.39)
Requirement already satisfied: pygments in /usr/local/lib/python3.10/dist-
packages (from ipython>=4.0.0->ipywidgets<8.0,>=7.5->dataprep) (2.16.1)
Requirement already satisfied: backcall in /usr/local/lib/python3.10/dist-
packages (from ipython>=4.0.0->ipywidgets<8.0,>=7.5->dataprep) (0.2.0)
Requirement already satisfied: matplotlib-inline in
/usr/local/lib/python3.10/dist-packages (from
ipython>=4.0.0->ipywidgets<8.0,>=7.5->dataprep) (0.1.6)
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.10/dist-
packages (from ipython>=4.0.0->ipywidgets<8.0,>=7.5->dataprep) (4.8.0)
Requirement already satisfied: locket in /usr/local/lib/python3.10/dist-packages
(from partd>=1.2.0->dask[array,dataframe,delayed]>=2022.3.0->dataprep) (1.0.0)
Requirement already satisfied: notebook>=4.4.1 in
/usr/local/lib/python3.10/dist-packages (from
widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (6.5.5)
Requirement already satisfied: idna>=2.0 in /usr/local/lib/python3.10/dist-
packages (from yarl<2.0,>=1.0->aiohttp<4.0,>=3.6->dataprep) (3.4)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from
matplotlib->wordcloud<2.0,>=1.8->dataprep) (1.1.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
packages (from matplotlib->wordcloud<2.0,>=1.8->dataprep) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from
matplotlib->wordcloud<2.0,>=1.8->dataprep) (4.42.0)
```

```
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from
matplotlib->wordcloud<2.0,>=1.8->dataprep) (1.4.4)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in
/usr/local/lib/python3.10/dist-packages (from
jedi>=0.16->ipython>=4.0.0->ipywidgets<8.0,>=7.5->dataprep) (0.8.3)
Requirement already satisfied: pyzmq<25,>=17 in /usr/local/lib/python3.10/dist-
packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep)
(23.2.1)
Requirement already satisfied: argon2-cffi in /usr/local/lib/python3.10/dist-
packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep)
(23.1.0)
Requirement already satisfied: jupyter-core>=4.6.1 in
/usr/local/lib/python3.10/dist-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep)
(5.3.1)
Requirement already satisfied: nbformat in /usr/local/lib/python3.10/dist-
packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep)
(5.9.2)
Requirement already satisfied: nbconvert>=5 in /usr/local/lib/python3.10/dist-
packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep)
(6.5.4)
Requirement already satisfied: nest-asyncio>=1.5 in
/usr/local/lib/python3.10/dist-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep)
(1.5.7)
Requirement already satisfied: Send2Trash>=1.8.0 in
/usr/local/lib/python3.10/dist-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep)
(1.8.2)
Requirement already satisfied: terminado>=0.8.3 in
/usr/local/lib/python3.10/dist-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep)
(0.17.1)
Requirement already satisfied: prometheus-client in
/usr/local/lib/python3.10/dist-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep)
(0.17.1)
Requirement already satisfied: nbclassic>=0.4.7 in
/usr/local/lib/python3.10/dist-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep)
(1.0.0)
Requirement already satisfied: ptyprocess>=0.5 in
/usr/local/lib/python3.10/dist-packages (from
```

```
pexpect>4.3->ipython>=4.0.0->ipywidgets<8.0,>=7.5->dataprep) (0.7.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-
packages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython>=4.0.0->ipy
widgets<8.0,>=7.5->dataprep) (0.2.6)
Requirement already satisfied: platformdirs>=2.5 in
/usr/local/lib/python3.10/dist-packages (from jupyter-core>=4.6.1->notebook>=4.4
.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (3.10.0)
Requirement already satisfied: jupyter-server>=1.8 in
/usr/local/lib/python3.10/dist-packages (from nbclassic>=0.4.7->notebook>=4.4.1-
>widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (1.24.0)
Requirement already satisfied: notebook-shim>=0.2.3 in
/usr/local/lib/python3.10/dist-packages (from nbclassic>=0.4.7->notebook>=4.4.1-
>widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (0.2.3)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packages
(from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>
=7.5->dataprep) (4.9.3)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-
packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidg
ets<8.0,>=7.5->dataprep) (4.11.2)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages
(from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>
=7.5->dataprep) (6.0.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.10/dist-
packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidg
ets<8.0,>=7.5->dataprep) (0.7.1)
Requirement already satisfied: entrypoints>=0.2.2 in
/usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->wid
getsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (0.4)
Requirement already satisfied: jupyterlab-pygments in
/usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->wid
getsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (0.2.2)
Requirement already satisfied: mistune<2,>=0.8.1 in
/usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->wid
getsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (0.8.4)
Requirement already satisfied: nbclient>=0.5.0 in
/usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->wid
getsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (0.8.0)
Requirement already satisfied: pandocfilters>=1.4.1 in
/usr/local/lib/python3.10/dist-packages (from nbconvert>=5->notebook>=4.4.1->wid
getsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (1.5.0)
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.10/dist-
packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidg
ets<8.0,>=7.5->dataprep) (1.2.1)
Requirement already satisfied: fastjsonschema in /usr/local/lib/python3.10/dist-
packages (from nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<
8.0,>=7.5->dataprep) (2.18.0)
Requirement already satisfied: jsonschema>=2.6 in
/usr/local/lib/python3.10/dist-packages (from nbformat->notebook>=4.4.1->widgets
```

```
nbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (4.19.0)
Requirement already satisfied: argon2-cffi-bindings in
/usr/local/lib/python3.10/dist-packages (from argon2-cffi->notebook>=4.4.1->widg
etsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (21.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
/usr/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nbformat->noteboo
k>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (2023.7.1)
Requirement already satisfied: referencing>=0.28.4 in
/usr/local/lib/python3.10/dist-packages (from jsonschema>=2.6->nbformat->noteboo
k>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (0.30.2)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-
packages (from jsonschema>=2.6->nbformat->notebook>=4.4.1->widgetsnbextension~=3
.6.0->ipywidgets<8.0,>=7.5->dataprep) (0.9.2)
Requirement already satisfied: anyio<4,>=3.1.0 in
/usr/local/lib/python3.10/dist-packages (from jupyter-server>=1.8->nbclassic>=0.
4.7->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep)
(3.7.1)
Requirement already satisfied: websocket-client in
/usr/local/lib/python3.10/dist-packages (from jupyter-server>=1.8->nbclassic>=0.
4.7->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep)
(1.6.1)
Requirement already satisfied: cffi>=1.0.1 in /usr/local/lib/python3.10/dist-
packages (from argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1->widgetsnbexte
nsion~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (1.15.1)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-
packages (from beautifulsoup4->nbconvert>=5->notebook>=4.4.1->widgetsnbextension
~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (2.4.1)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-
packages (from bleach->nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0-
>ipywidgets<8.0,>=7.5->dataprep) (0.5.1)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-
packages (from anyio<4,>=3.1.0->jupyter-server>=1.8->nbclassic>=0.4.7->notebook>
=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (1.3.0)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-
packages (from anyio<4,>=3.1.0->jupyter-server>=1.8->nbclassic>=0.4.7->notebook>
=4.4.1->widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (1.1.3)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-
packages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1->
widgetsnbextension~=3.6.0->ipywidgets<8.0,>=7.5->dataprep) (2.21)
Building wheels for collected packages: sqlalchemy, metaphone
  Building wheel for sqlalchemy (setup.py) … done
  Created wheel for sqlalchemy:
filename=SQLAlchemy-1.3.24-cp310-cp310-linux_x86_64.whl size=1252698
sha256=b2efe44418dca6adc05a035dd5f1e8f1acb1eda2435a6efc912f7b64a3a7aca0
  Stored in directory: /root/.cache/pip/wheels/27/51/b3/3481e88d5a5ba95dd4aafedc
9316774d941c4ba61cfb93add8
  Building wheel for metaphone (setup.py) … done
  Created wheel for metaphone: filename=Metaphone-0.6-py3-none-any.whl
```

```
size=13901
sha256=210167338c6671eaf5f9f9aa1c1b08204c75191cbbf5e856bb9f7b794c046f11
  Stored in directory: /root/.cache/pip/wheels/23/dd/1d/6cdd346605db62bde1f60954
155e9ce48f4681c243f265b704
Successfully built sqlalchemy metaphone
Installing collected packages: regex, python-stdnum, python-crfsuite, pure_eval,
ply, metaphone, executing, sqlalchemy, rapidfuzz, jsonpath-ng, jinja2,
asttokens, varname, flask_cors, dataprep
  Attempting uninstall: regex
    Found existing installation: regex 2023.6.3
    Uninstalling regex-2023.6.3:
      Successfully uninstalled regex-2023.6.3
  Attempting uninstall: sqlalchemy
    Found existing installation: SQLAlchemy 2.0.20
    Uninstalling SQLAlchemy-2.0.20:
      Successfully uninstalled SQLAlchemy-2.0.20
  Attempting uninstall: jinja2
    Found existing installation: Jinja2 3.1.2
    Uninstalling Jinja2-3.1.2:
      Successfully uninstalled Jinja2-3.1.2
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
ipython-sql 0.5.0 requires sqlalchemy>=2.0, but you have sqlalchemy 1.3.24 which
is incompatible.
Successfully installed asttokens-2.2.1 dataprep-0.4.5 executing-0.8.3
flask_cors-3.0.10 jinja2-3.0.3 jsonpath-ng-1.5.3 metaphone-0.6 ply-3.11
pure_eval-0.2.2 python-crfsuite-0.9.8 python-stdnum-1.19 rapidfuzz-2.15.1
regex-2021.11.10 sqlalchemy-1.3.24 varname-0.8.3
```

[ ]:
```python
from dataprep.eda import create_report

report = create_report(df, title = 'My Report')

report
```

[ ]:

[ ]:
```python

```

data-processing-ds-project1

January 2, 2024

Data Pre-processing

Type casting

[117]:
```
import pandas as pd
```

[118]:
```
project = pd.read_csv(r"/content/Datasets.csv")
```

[119]:
```
project.dtypes
```

[119]:
```
Year                      int64
Month                     int64
DayofMonth                int64
DayOfWeek                 int64
Actual_Shipment_Time    float64
Planned_Shipment_Time     int64
Planned_Delivery_Time     int64
Carrier_Name             object
Carrier_Num               int64
Planned_TimeofTravel      int64
Shipment_Delay          float64
Source                   object
Destination              object
Distance                  int64
Delivery_Status         float64
dtype: object
```

[ ]:
```
help(project.astype)
```

```
Help on method astype in module pandas.core.generic:

astype(dtype, copy: 'bool_t' = True, errors: 'IgnoreRaise' = 'raise') ->
'NDFrameT' method of pandas.core.frame.DataFrame instance
    Cast a pandas object to a specified dtype ``dtype``.

    Parameters
    ----------
    dtype : data type, or dict of column name -> data type
        Use a numpy.dtype or Python type to cast entire pandas object to
```

```
    the same type. Alternatively, use {col: dtype, …}, where col is a
    column label and dtype is a numpy.dtype or Python type to cast one
    or more of the DataFrame's columns to column-specific types.
copy : bool, default True
    Return a copy when ``copy=True`` (be very careful setting
    ``copy=False`` as changes to values then may propagate to other
    pandas objects).
errors : {'raise', 'ignore'}, default 'raise'
    Control raising of exceptions on invalid data for provided dtype.

    - ``raise`` : allow exceptions to be raised
    - ``ignore`` : suppress exceptions. On error return original object.

Returns
-------
casted : same type as caller

See Also
--------
to_datetime : Convert argument to datetime.
to_timedelta : Convert argument to timedelta.
to_numeric : Convert argument to a numeric type.
numpy.ndarray.astype : Cast a numpy array to a specified type.

Notes
-----
.. deprecated:: 1.3.0

    Using ``astype`` to convert from timezone-naive dtype to
    timezone-aware dtype is deprecated and will raise in a
    future version.  Use :meth:`Series.dt.tz_localize` instead.

Examples
--------
Create a DataFrame:

>>> d = {'col1': [1, 2], 'col2': [3, 4]}
>>> df = pd.DataFrame(data=d)
>>> df.dtypes
col1    int64
col2    int64
dtype: object

Cast all columns to int32:

>>> df.astype('int32').dtypes
col1    int32
col2    int32
```

```
dtype: object

Cast col1 to int32 using a dictionary:

>>> df.astype({'col1': 'int32'}).dtypes
col1    int32
col2    int64
dtype: object

Create a series:

>>> ser = pd.Series([1, 2], dtype='int32')
>>> ser
0    1
1    2
dtype: int32
>>> ser.astype('int64')
0    1
1    2
dtype: int64

Convert to categorical type:

>>> ser.astype('category')
0    1
1    2
dtype: category
Categories (2, int64): [1, 2]

Convert to ordered categorical type with custom ordering:

>>> from pandas.api.types import CategoricalDtype
>>> cat_dtype = CategoricalDtype(
…     categories=[2, 1], ordered=True)
>>> ser.astype(cat_dtype)
0    1
1    2
dtype: category
Categories (2, int64): [2 < 1]

Note that using ``copy=False`` and changing data on a new
pandas object may propagate changes:

>>> s1 = pd.Series([1, 2])
>>> s2 = s1.astype('int64', copy=False)
>>> s2[0] = 10
>>> s1  # note that s1[0] has changed too
0    10
```

```
1     2
dtype: int64

Create a series of dates:

>>> ser_date = pd.Series(pd.date_range('20200101', periods=3))
>>> ser_date
0   2020-01-01
1   2020-01-02
2   2020-01-03
dtype: datetime64[ns]
```

1 Convert 'int64' to 'str' (string) type.

[ ]:
```
project.Year = project.Year.astype('str')
```

[ ]:
```
project.dtypes
```

[ ]:
```
Year                      object
Month                      int64
DayofMonth                 int64
DayOfWeek                  int64
Actual_Shipment_Time     float64
Planned_Shipment_Time      int64
Planned_Delivery_Time      int64
Carrier_Name              object
Carrier_Num                int64
Planned_TimeofTravel       int64
Shipment_Delay           float64
Source                    object
Destination               object
Distance                   int64
Delivery_Status          float64
dtype: object
```

[ ]:
```
project.Month = project.Month.astype('str')
```

[ ]:
```
project.dtypes
```

[ ]:
```
Year                      object
Month                     object
DayofMonth                 int64
DayOfWeek                  int64
Actual_Shipment_Time     float64
Planned_Shipment_Time      int64
```

```
Planned_Delivery_Time      int64
Carrier_Name              object
Carrier_Num                int64
Planned_TimeofTravel       int64
Shipment_Delay           float64
Source                    object
Destination               object
Distance                   int64
Delivery_Status          float64
dtype: object
```

[ ]:
```
project.Planned_Shipment_Time = project.Planned_Shipment_Time.astype('str')
```

[ ]:
```
project.dtypes
```

[ ]:
```
Year                      object
Month                     object
DayofMonth                 int64
DayOfWeek                  int64
Actual_Shipment_Time     float64
Planned_Shipment_Time     object
Planned_Delivery_Time      int64
Carrier_Name              object
Carrier_Num                int64
Planned_TimeofTravel       int64
Shipment_Delay           float64
Source                    object
Destination               object
Distance                   int64
Delivery_Status          float64
dtype: object
```

2 convert 'str' to 'int64' type.

[ ]:
```
project.Year = project.Year.astype('int64')
```

[ ]:
```
project.dtypes
```

[ ]:
```
Year                       int64
Month                     object
DayofMonth                 int64
DayOfWeek                  int64
Actual_Shipment_Time     float64
Planned_Shipment_Time     object
Planned_Delivery_Time      int64
Carrier_Name              object
```

```
Carrier_Num                int64
Planned_TimeofTravel       int64
Shipment_Delay           float64
Source                    object
Destination               object
Distance                   int64
Delivery_Status          float64
dtype: object
```

[ ]:
```
project.Month = project.Month.astype('int64')
```

[ ]:
```
project.dtypes
```

[ ]:
```
Year                       int64
Month                      int64
DayofMonth                 int64
DayOfWeek                  int64
Actual_Shipment_Time     float64
Planned_Shipment_Time     object
Planned_Delivery_Time      int64
Carrier_Name              object
Carrier_Num                int64
Planned_TimeofTravel       int64
Shipment_Delay           float64
Source                    object
Destination               object
Distance                   int64
Delivery_Status          float64
dtype: object
```

[ ]:
```
project.Planned_Shipment_Time = project.Planned_Shipment_Time.astype('int64')
```

[ ]:
```
project.dtypes
```

[ ]:
```
Year                       int64
Month                      int64
DayofMonth                 int64
DayOfWeek                  int64
Actual_Shipment_Time     float64
Planned_Shipment_Time      int64
Planned_Delivery_Time      int64
Carrier_Name              object
Carrier_Num                int64
Planned_TimeofTravel       int64
Shipment_Delay           float64
Source                    object
Destination               object
```

```
Distance                    int64
Delivery_Status           float64
dtype: object
```

3 'float64' into 'int64' type

[ ]:
```
project.Actual_Shipment_Time = project.Actual_Shipment_Time.astype('int64')
```

```
---------------------------------------------------------------------------
IntCastingNaNError                        Traceback (most recent call last)
<ipython-input-17-a261dbd43cc9> in <cell line: 1>()
----> 1 project.Actual_Shipment_Time = project.Actual_Shipment_Time.
 ↪astype('int64')

/usr/local/lib/python3.10/dist-packages/pandas/core/generic.py in astype(self,␣
 ↪dtype, copy, errors)
   6238         else:
   6239             # else, only a single dtype is given
-> 6240             new_data = self._mgr.astype(dtype=dtype, copy=copy,␣
 ↪errors=errors)
   6241             return self._constructor(new_data).__finalize__(self,␣
 ↪method="astype")
   6242

/usr/local/lib/python3.10/dist-packages/pandas/core/internals/managers.py in␣
 ↪astype(self, dtype, copy, errors)
    446
    447     def astype(self: T, dtype, copy: bool = False, errors: str =␣
 ↪"raise") -> T:
--> 448         return self.apply("astype", dtype=dtype, copy=copy,␣
 ↪errors=errors)
    449
    450     def convert(

/usr/local/lib/python3.10/dist-packages/pandas/core/internals/managers.py in␣
 ↪apply(self, f, align_keys, ignore_failures, **kwargs)
    350                     applied = b.apply(f, **kwargs)
    351                 else:
--> 352                     applied = getattr(b, f)(**kwargs)
    353             except (TypeError, NotImplementedError):
    354                 if not ignore_failures:

/usr/local/lib/python3.10/dist-packages/pandas/core/internals/blocks.py in␣
 ↪astype(self, dtype, copy, errors)
    524         values = self.values
    525
```

```
--> 526         new_values = astype_array_safe(values, dtype, copy=copy,␣
 ↪errors=errors)
    527
    528         new_values = maybe_coerce_values(new_values)

/usr/local/lib/python3.10/dist-packages/pandas/core/dtypes/astype.py in␣
 ↪astype_array_safe(values, dtype, copy, errors)
    297
    298     try:
--> 299         new_values = astype_array(values, dtype, copy=copy)
    300     except (ValueError, TypeError):
    301         # e.g. astype_nansafe can fail on object-dtype of strings

/usr/local/lib/python3.10/dist-packages/pandas/core/dtypes/astype.py in␣
 ↪astype_array(values, dtype, copy)
    228
    229     else:
--> 230         values = astype_nansafe(values, dtype, copy=copy)
    231
    232     # in pandas we don't store numpy str dtypes, so convert to object

/usr/local/lib/python3.10/dist-packages/pandas/core/dtypes/astype.py in␣
 ↪astype_nansafe(arr, dtype, copy, skipna)
    138
    139     elif np.issubdtype(arr.dtype, np.floating) and␣
 ↪is_integer_dtype(dtype):
--> 140         return _astype_float_to_int_nansafe(arr, dtype, copy)
    141
    142     elif is_object_dtype(arr.dtype):

/usr/local/lib/python3.10/dist-packages/pandas/core/dtypes/astype.py in␣
 ↪_astype_float_to_int_nansafe(values, dtype, copy)
    180     """
    181     if not np.isfinite(values).all():
--> 182         raise IntCastingNaNError(
    183             "Cannot convert non-finite values (NA or inf) to integer"
    184         )

IntCastingNaNError: Cannot convert non-finite values (NA or inf) to integer
```

[ ]:
```
project.dtypes
```

[ ]:
```
Year                        int64
Month                       int64
DayofMonth                  int64
DayOfWeek                   int64
```

```
Actual_Shipment_Time     float64
Planned_Shipment_Time      int64
Planned_Delivery_Time      int64
Carrier_Name              object
Carrier_Num                int64
Planned_TimeofTravel       int64
Shipment_Delay           float64
Source                    object
Destination               object
Distance                   int64
Delivery_Status          float64
dtype: object
```

[ ]:
```
project.Shipment_Delay = project.Shipment_Delay.astype('int64')
```

```
---------------------------------------------------------------------------
IntCastingNaNError                        Traceback (most recent call last)
<ipython-input-19-ab5ba5d19076> in <cell line: 1>()
----> 1 project.Shipment_Delay = project.Shipment_Delay.astype('int64')

/usr/local/lib/python3.10/dist-packages/pandas/core/generic.py in astype(self,␣
↪dtype, copy, errors)
   6238         else:
   6239             # else, only a single dtype is given
-> 6240             new_data = self._mgr.astype(dtype=dtype, copy=copy,␣
↪errors=errors)
   6241             return self._constructor(new_data).__finalize__(self,␣
↪method="astype")
   6242

/usr/local/lib/python3.10/dist-packages/pandas/core/internals/managers.py in␣
↪astype(self, dtype, copy, errors)
    446
    447     def astype(self: T, dtype, copy: bool = False, errors: str =␣
↪"raise") -> T:
--> 448         return self.apply("astype", dtype=dtype, copy=copy,␣
↪errors=errors)
    449
    450     def convert(

/usr/local/lib/python3.10/dist-packages/pandas/core/internals/managers.py in␣
↪apply(self, f, align_keys, ignore_failures, **kwargs)
    350                     applied = b.apply(f, **kwargs)
    351                 else:
--> 352                     applied = getattr(b, f)(**kwargs)
    353             except (TypeError, NotImplementedError):
    354                 if not ignore_failures:
```

```
/usr/local/lib/python3.10/dist-packages/pandas/core/internals/blocks.py in␣
↪astype(self, dtype, copy, errors)
    524         values = self.values
    525
--> 526         new_values = astype_array_safe(values, dtype, copy=copy,␣
↪errors=errors)
    527
    528         new_values = maybe_coerce_values(new_values)

/usr/local/lib/python3.10/dist-packages/pandas/core/dtypes/astype.py in␣
↪astype_array_safe(values, dtype, copy, errors)
    297
    298     try:
--> 299         new_values = astype_array(values, dtype, copy=copy)
    300     except (ValueError, TypeError):
    301         # e.g. astype_nansafe can fail on object-dtype of strings

/usr/local/lib/python3.10/dist-packages/pandas/core/dtypes/astype.py in␣
↪astype_array(values, dtype, copy)
    228
    229     else:
--> 230         values = astype_nansafe(values, dtype, copy=copy)
    231
    232     # in pandas we don't store numpy str dtypes, so convert to object

/usr/local/lib/python3.10/dist-packages/pandas/core/dtypes/astype.py in␣
↪astype_nansafe(arr, dtype, copy, skipna)
    138
    139     elif np.issubdtype(arr.dtype, np.floating) and␣
↪is_integer_dtype(dtype):
--> 140         return _astype_float_to_int_nansafe(arr, dtype, copy)
    141
    142     elif is_object_dtype(arr.dtype):

/usr/local/lib/python3.10/dist-packages/pandas/core/dtypes/astype.py in␣
↪_astype_float_to_int_nansafe(values, dtype, copy)
    180     """
    181     if not np.isfinite(values).all():
--> 182         raise IntCastingNaNError(
    183             "Cannot convert non-finite values (NA or inf) to integer"
    184         )

IntCastingNaNError: Cannot convert non-finite values (NA or inf) to integer
```

[ ]:
```
project.dtypes
```

[ ]:
```
Year                        int64
Month                       int64
DayofMonth                  int64
DayOfWeek                   int64
Actual_Shipment_Time      float64
Planned_Shipment_Time       int64
Planned_Delivery_Time       int64
Carrier_Name               object
Carrier_Num                 int64
Planned_TimeofTravel        int64
Shipment_Delay            float64
Source                     object
Destination                object
Distance                    int64
Delivery_Status           float64
dtype: object
```

3.0.1 Identify duplicate records in the data

[ ]:
```
import pandas as pd
```

[ ]:
```
project = pd.read_csv(r"/content/Datasets.csv")
```

4 Duplicates in rows

[ ]:
```
help(project.duplicated)
```

```
Help on method duplicated in module pandas.core.frame:

duplicated(subset: 'Hashable | Sequence[Hashable] | None' = None, keep:
"Literal['first', 'last', False]" = 'first') -> 'Series' method of
pandas.core.frame.DataFrame instance
    Return boolean Series denoting duplicate rows.

    Considering certain columns is optional.

    Parameters
    ----------
    subset : column label or sequence of labels, optional
        Only consider certain columns for identifying duplicates, by
        default use all of the columns.
    keep : {'first', 'last', False}, default 'first'
        Determines which duplicates (if any) to mark.

        - ``first`` : Mark duplicates as ``True`` except for the first
occurrence.
```

```
    - ``last`` : Mark duplicates as ``True`` except for the last occurrence.
    - False : Mark all duplicates as ``True``.

Returns
-------
Series
    Boolean series for each duplicated rows.

See Also
--------
Index.duplicated : Equivalent method on index.
Series.duplicated : Equivalent method on Series.
Series.drop_duplicates : Remove duplicate values from Series.
DataFrame.drop_duplicates : Remove duplicate values from DataFrame.

Examples
--------
Consider dataset containing ramen rating.

>>> df = pd.DataFrame({
…     'brand': ['Yum Yum', 'Yum Yum', 'Indomie', 'Indomie', 'Indomie'],
…     'style': ['cup', 'cup', 'cup', 'pack', 'pack'],
…     'rating': [4, 4, 3.5, 15, 5]
… })
>>> df
    brand style  rating
0  Yum Yum   cup     4.0
1  Yum Yum   cup     4.0
2  Indomie   cup     3.5
3  Indomie  pack    15.0
4  Indomie  pack     5.0

By default, for each set of duplicated values, the first occurrence
is set on False and all others on True.

>>> df.duplicated()
0    False
1     True
2    False
3    False
4    False
dtype: bool

By using 'last', the last occurrence of each set of duplicated values
is set on False and all others on True.

>>> df.duplicated(keep='last')
0     True
```

```
1    False
2    False
3    False
4    False
dtype: bool

By setting ``keep`` on False, all duplicates are True.

>>> df.duplicated(keep=False)
0     True
1     True
2    False
3    False
4    False
dtype: bool

To find duplicates on specific column(s), use ``subset``.

>>> df.duplicated(subset=['brand'])
0    False
1     True
2    False
3     True
4     True
dtype: bool
```

[ ]:
```
duplicate = project.duplicated()
```

[ ]:
```
duplicate
```

[ ]:
```
0       False
1       False
2       False
3       False
4       False
        …
7994    False
7995    False
7996    False
7997    False
7998    False
Length: 7999, dtype: bool
```

[ ]:
```
sum(duplicate)
```

[ ]:
```
0
```

5 Duplicates in rows

[ ]:
```
help(project.duplicated)
```

```
Help on method duplicated in module pandas.core.frame:

duplicated(subset: 'Hashable | Sequence[Hashable] | None' = None, keep:
"Literal['first', 'last', False]" = 'first') -> 'Series' method of
pandas.core.frame.DataFrame instance
    Return boolean Series denoting duplicate rows.

    Considering certain columns is optional.

    Parameters
    ----------
    subset : column label or sequence of labels, optional
        Only consider certain columns for identifying duplicates, by
        default use all of the columns.
    keep : {'first', 'last', False}, default 'first'
        Determines which duplicates (if any) to mark.

        - ``first`` : Mark duplicates as ``True`` except for the first
occurrence.
        - ``last`` : Mark duplicates as ``True`` except for the last occurrence.
        - False : Mark all duplicates as ``True``.

    Returns
    -------
    Series
        Boolean series for each duplicated rows.

    See Also
    --------
    Index.duplicated : Equivalent method on index.
    Series.duplicated : Equivalent method on Series.
    Series.drop_duplicates : Remove duplicate values from Series.
    DataFrame.drop_duplicates : Remove duplicate values from DataFrame.

    Examples
    --------
    Consider dataset containing ramen rating.

    >>> df = pd.DataFrame({
    …     'brand': ['Yum Yum', 'Yum Yum', 'Indomie', 'Indomie', 'Indomie'],
    …     'style': ['cup', 'cup', 'cup', 'pack', 'pack'],
    …     'rating': [4, 4, 3.5, 15, 5]
    … })
```

```
>>> df
    brand style  rating
0  Yum Yum   cup     4.0
1  Yum Yum   cup     4.0
2  Indomie   cup     3.5
3  Indomie  pack    15.0
4  Indomie  pack     5.0

By default, for each set of duplicated values, the first occurrence
is set on False and all others on True.

>>> df.duplicated()
0    False
1     True
2    False
3    False
4    False
dtype: bool

By using 'last', the last occurrence of each set of duplicated values
is set on False and all others on True.

>>> df.duplicated(keep='last')
0     True
1    False
2    False
3    False
4    False
dtype: bool

By setting ``keep`` on False, all duplicates are True.

>>> df.duplicated(keep=False)
0     True
1     True
2    False
3    False
4    False
dtype: bool

To find duplicates on specific column(s), use ``subset``.

>>> df.duplicated(subset=['brand'])
0    False
1     True
2    False
3     True
4     True
```

```
dtype: bool
```

[ ]:
```
duplicate = project.duplicated()  # Returns Boolean Series denoting duplicate␣
↪rows.
```

[ ]:
```
duplicate
```

[ ]:
```
0       False
1       False
2       False
3       False
4       False
        …
7994    False
7995    False
7996    False
7997    False
7998    False
Length: 7999, dtype: bool
```

[ ]:
```
sum(duplicate)
```

[ ]: 0

6 Parameters

[ ]:
```
duplicate = project.duplicated(keep = 'last')
```

[ ]:
```
duplicate
```

[ ]:
```
0       False
1       False
2       False
3       False
4       False
        …
7994    False
7995    False
7996    False
7997    False
7998    False
Length: 7999, dtype: bool
```

[ ]:
```
duplicate = project.duplicated(keep = False)
```

[ ]:
```
duplicate
```

[ ]:
```
0       False
1       False
2       False
3       False
4       False
        …
7994    False
7995    False
7996    False
7997    False
7998    False
Length: 7999, dtype: bool
```

7 Removing Duplicates

[ ]:
```
project1 = project.drop_duplicates() # Returns DataFrame with duplicate rows␣
↪removed.
```

8 Parameters

[ ]:
```
project1 = project.drop_duplicates(keep = 'last')
```

[ ]:
```
project1 = project.drop_duplicates(keep = False)
```

9 Duplicates in Columns

10 We can use correlation coefficient values to identify columns which have duplicate information

[ ]:
```
import pandas as pd
```

[ ]:
```
project = pd.read_csv(r"/content/Datasets.csv")
```

11 Correlation coefficient

[ ]:
```
project.corr()
```

```
<ipython-input-40-f63a631e511a>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
```

```
to silence this warning.
  project.corr()
```

[ ]:

	Year	Month	DayofMonth	DayOfWeek	\
Year	NaN	NaN	NaN	NaN	
Month	NaN	NaN	NaN	NaN	
DayofMonth	NaN	NaN	1.000000	1.000000	
DayOfWeek	NaN	NaN	1.000000	1.000000	
Actual_Shipment_Time	NaN	NaN	-0.014877	-0.014877	
Planned_Shipment_Time	NaN	NaN	-0.004550	-0.004550	
Planned_Delivery_Time	NaN	NaN	-0.000644	-0.000644	
Carrier_Num	NaN	NaN	-0.053688	-0.053688	
Planned_TimeofTravel	NaN	NaN	0.022032	0.022032	
Shipment_Delay	NaN	NaN	-0.077483	-0.077483	
Distance	NaN	NaN	0.016220	0.016220	
Delivery_Status	NaN	NaN	-0.121121	-0.121121	

	Actual_Shipment_Time	Planned_Shipment_Time	\
Year	NaN	NaN	
Month	NaN	NaN	
DayofMonth	-0.014877	-0.004550	
DayOfWeek	-0.014877	-0.004550	
Actual_Shipment_Time	1.000000	0.992386	
Planned_Shipment_Time	0.992386	1.000000	
Planned_Delivery_Time	0.847986	0.858210	
Carrier_Num	0.005744	0.005147	
Planned_TimeofTravel	-0.063763	-0.070638	
Shipment_Delay	0.434833	0.338752	
Distance	-0.053634	-0.062261	
Delivery_Status	0.459595	0.397657	

	Planned_Delivery_Time	Carrier_Num	\
Year	NaN	NaN	
Month	NaN	NaN	
DayofMonth	-0.000644	-0.053688	
DayOfWeek	-0.000644	-0.053688	
Actual_Shipment_Time	0.847986	0.005744	
Planned_Shipment_Time	0.858210	0.005147	
Planned_Delivery_Time	1.000000	-0.004370	
Carrier_Num	-0.004370	1.000000	
Planned_TimeofTravel	0.030032	0.045030	
Shipment_Delay	0.270309	0.004711	
Distance	0.038032	0.035700	
Delivery_Status	0.341430	0.005415	

	Planned_TimeofTravel	Shipment_Delay	Distance	\
Year	NaN	NaN	NaN	

```
Month                                 NaN             NaN       NaN
DayofMonth                       0.022032       -0.077483  0.016220
DayOfWeek                        0.022032       -0.077483  0.016220
Actual_Shipment_Time            -0.063763        0.434833 -0.053634
Planned_Shipment_Time           -0.070638        0.338752 -0.062261
Planned_Delivery_Time            0.030032        0.270309  0.038032
Carrier_Num                      0.045030        0.004711  0.035700
Planned_TimeofTravel             1.000000        0.032342  0.980355
Shipment_Delay                   0.032342        1.000000  0.050998
Distance                         0.980355        0.050998  1.000000
Delivery_Status                  0.025275        0.692433  0.044404

                       Delivery_Status
Year                               NaN
Month                              NaN
DayofMonth                   -0.121121
DayOfWeek                    -0.121121
Actual_Shipment_Time          0.459595
Planned_Shipment_Time         0.397657
Planned_Delivery_Time         0.341430
Carrier_Num                   0.005415
Planned_TimeofTravel          0.025275
Shipment_Delay                0.692433
Distance                      0.044404
Delivery_Status               1.000000
```

Missing Values - Imputation

[ ]:
```python
import numpy as np
```

[ ]:
```python
import pandas as pd
```

[ ]:
```python
project = pd.read_csv(r"/content/Datasets.csv")
```

[ ]:
```python
project.isna().sum()
```

[ ]:
```
Year                      0
Month                     0
DayofMonth                0
DayOfWeek                 0
Actual_Shipment_Time    139
Planned_Shipment_Time     0
Planned_Delivery_Time     0
Carrier_Name              0
Carrier_Num               0
Planned_TimeofTravel      0
Shipment_Delay          139
```

```
Source                     0
Destination                0
Distance                   0
Delivery_Status          139
dtype: int64
```

12 For Mean, Median, Mode imputation we can use Simple Imputer or df.fillna()

[ ]:
```
from sklearn.impute import SimpleImputer
```

13 Mean Imputer

[ ]:
```
mean_imputer1 = SimpleImputer(missing_values = np.nan, strategy = 'mean')
```

[ ]:
```
project["Actual_Shipment_Time"] = pd.DataFrame(mean_imputer1.
 ↪fit_transform(project[["Actual_Shipment_Time"]]))
```

[ ]:
```
project["Actual_Shipment_Time"].isna().sum()
```

[ ]: 0

14 Median Imputer

[ ]:
```
median_imputer1 = SimpleImputer(missing_values = np.nan, strategy = 'median')
```

[ ]:
```
project["Planned_Shipment_Time"] = pd.DataFrame(median_imputer1.
 ↪fit_transform(project[["Planned_Shipment_Time"]]))
```

[ ]:
```
project["Planned_Shipment_Time"].isna().sum()
```

[ ]: 0

[ ]:
```
project.isna().sum()
```

[ ]:
```
Year                       0
Month                      0
DayofMonth                 0
DayOfWeek                  0
Actual_Shipment_Time       0
Planned_Shipment_Time      0
Planned_Delivery_Time      0
Carrier_Name               0
```

```
Carrier_Num                0
Planned_TimeofTravel       0
Shipment_Delay           139
Source                     0
Destination                0
Distance                   0
Delivery_Status          139
dtype: int64
```

15 Mode Imputer

[ ]:
```
mode_imputer1 = SimpleImputer(missing_values = np.nan, strategy =␣
↪'most_frequent')
```

[ ]:
```
project["Planned_Delivery_Time"] = pd.DataFrame(mode_imputer1.
↪fit_transform(project[["Planned_Delivery_Time"]]))
```

[ ]:
```
project["Planned_Delivery_Time"] = pd.DataFrame(mode_imputer1.
↪fit_transform(project[["Planned_Delivery_Time"]]))
```

[ ]:
```
project.isnull().sum()
```

[ ]:
```
Year                       0
Month                      0
DayofMonth                 0
DayOfWeek                  0
Actual_Shipment_Time       0
Planned_Shipment_Time      0
Planned_Delivery_Time      0
Carrier_Name               0
Carrier_Num                0
Planned_TimeofTravel       0
Shipment_Delay           139
Source                     0
Destination                0
Distance                   0
Delivery_Status          139
dtype: int64
```

Outlier Treatment

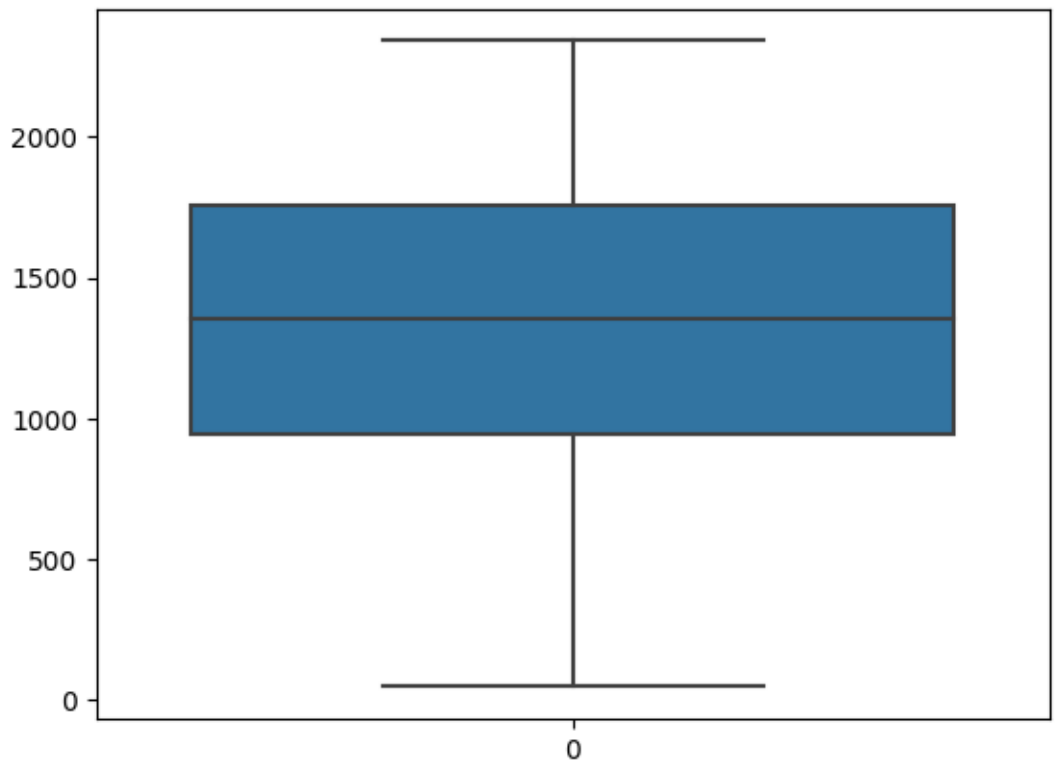
[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
```

[2]:
```
project = pd.read_csv(r"/content/Datasets.csv")
```

16 Let's find outliers in Actual_Shipment_Time

[3]:
```
sns.boxplot(project.Actual_Shipment_Time)
```

[3]: <Axes: >



17 No outliers in Actual_Shipment_Time column

18 Detection of outliers (find limits for Actual_Shipment_Time based on IQR)

[4]:
```
IQR = project['Actual_Shipment_Time'].quantile(0.75) -␣
↪project['Actual_Shipment_Time'].quantile(0.25)
```

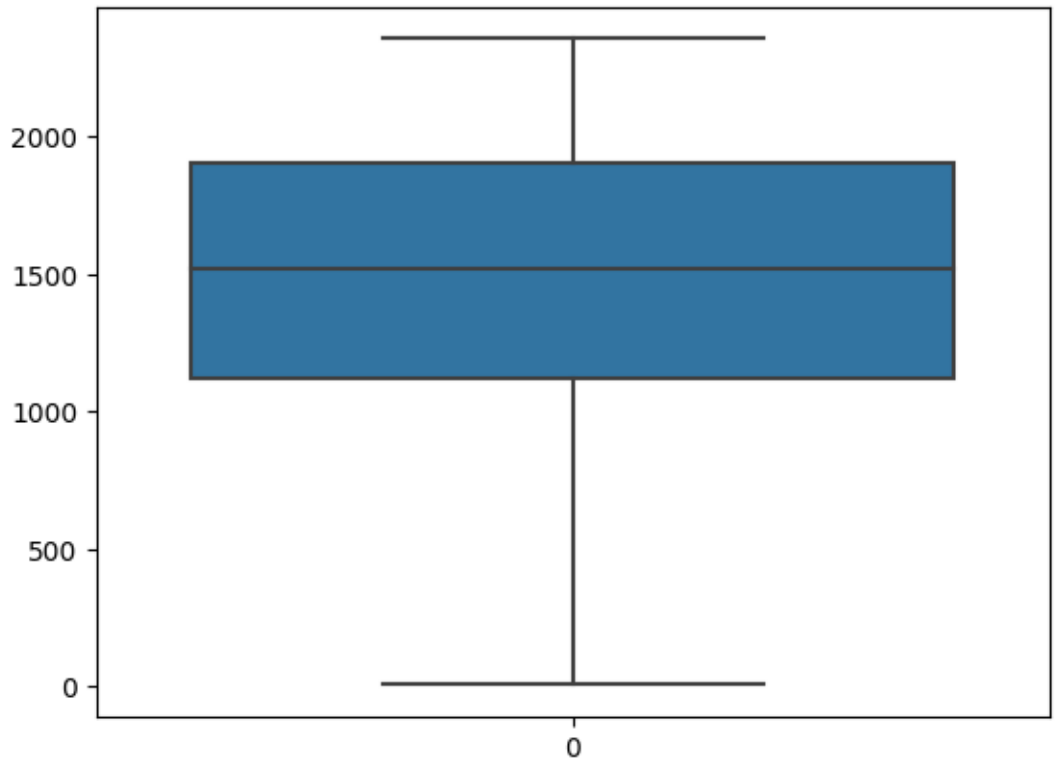
[5]:
```
lower_limit1 = project['Actual_Shipment_Time'].quantile(0.25) - (IQR * 1.5)
```

[6]:
```
upper_limit1 = project['Actual_Shipment_Time'].quantile(0.75) + (IQR * 1.5)
```

19 Let's find outliers in Planned_Delivered_Time

[7]:
```
sns.boxplot(project.Planned_Delivery_Time)
```

[7]: <Axes: >



20 No outliers in Planned_Delivered_Time column

21 Detection of outliers (find limits for Planned_Delivery_Time based on IQR)

[8]:
```
IQR = project['Planned_Delivery_Time'].quantile(0.75) -␣
↪project['Planned_Delivery_Time'].quantile(0.25)
```

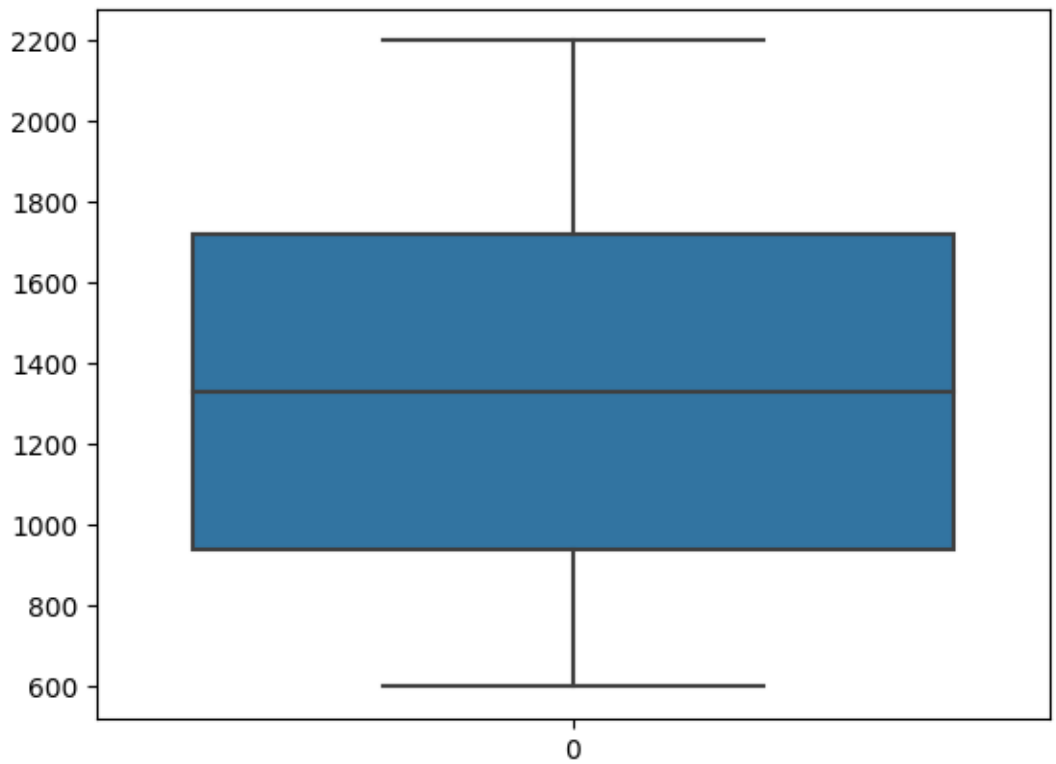
[9]:
```
lower_limit3 = project['Planned_Delivery_Time'].quantile(0.25) - (IQR * 1.5)
```

[10]:
```
upper_limit3 = project['Planned_Delivery_Time'].quantile(0.75) + (IQR * 1.5)
```

22 Let's find outliers in Planned_Shipment_Time

[11]:
```
sns.boxplot(project.Planned_Shipment_Time)
```

[11]: <Axes: >



23 No outliers in Planned_Shipment_Time column

24 Detection of outliers (find limits for Planned_Shipment_Time based on IQR)

[12]:
```
IQR = project['Planned_Shipment_Time'].quantile(0.75) -␣
↪project['Planned_Shipment_Time'].quantile(0.25)
```

[13]:
```
lower_limit2 = project['Planned_Shipment_Time'].quantile(0.25) - (IQR * 1.5)
```

[14]:
```
upper_limit2 = project['Planned_Shipment_Time'].quantile(0.75) + (IQR * 1.5)
```

1. Remove (let's trim the dataset)

25 Trimming Technique

26 Let's flag the outliers in the dataset

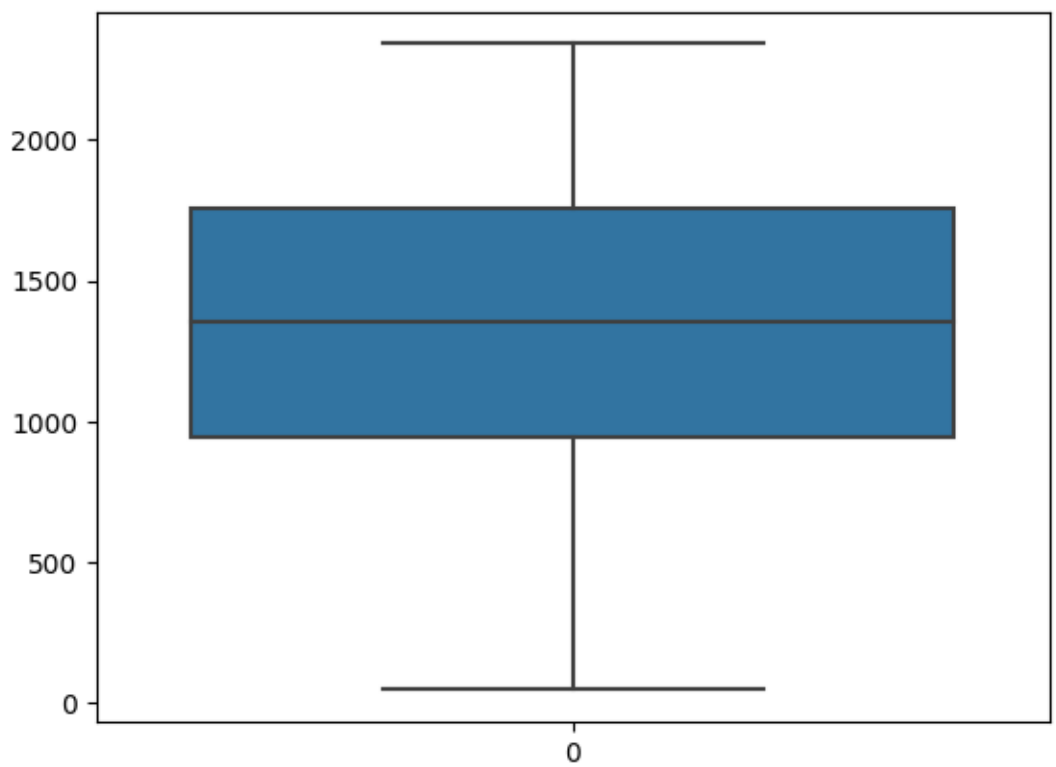
[15]:
```
outliers_project1 = np.where(project.Actual_Shipment_Time > upper_limit1, True,␣
↪np.where(project.Actual_Shipment_Time < lower_limit1, True, False))
```

[16]:
```
# outliers data
project_out1 = project.loc[outliers_project1, ]
project_trimmed1 = project.loc[~(outliers_project1), ]
project.shape, project_trimmed1.shape
```

[16]: ((7999, 15), (7999, 15))

[17]:
```
# Let's explore outliers in the trimmed dataset
sns.boxplot(project_trimmed1.Actual_Shipment_Time)
```

[17]: <Axes: >



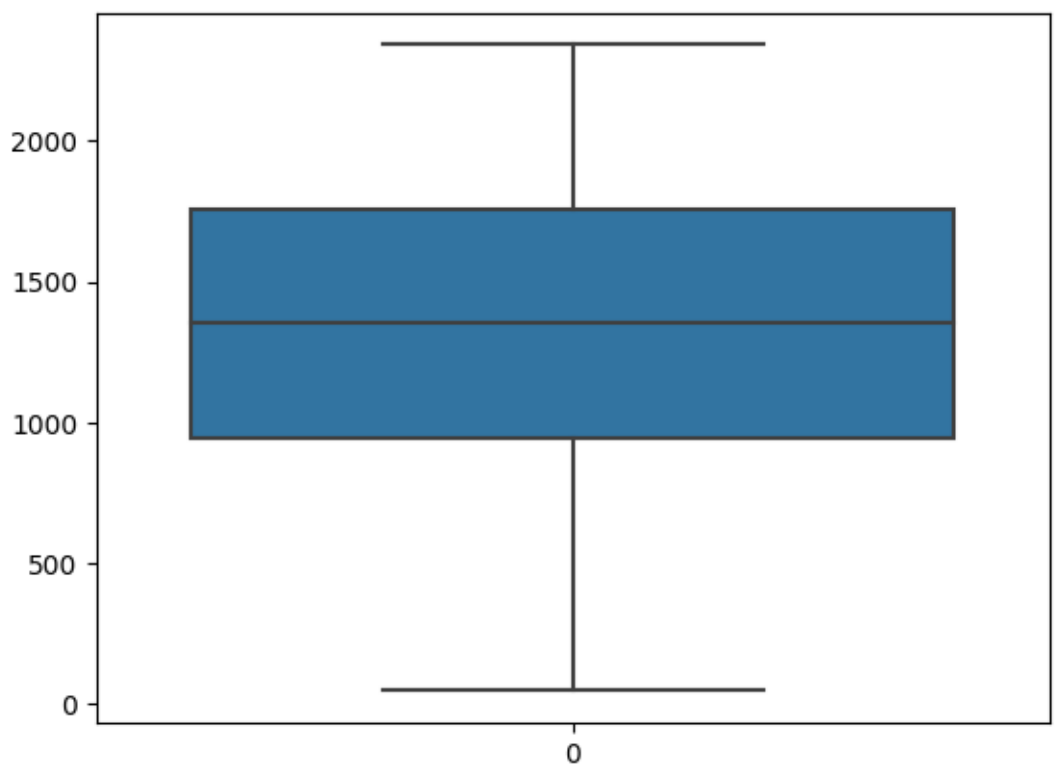
[18]:
```
outliers_project2 = np.where(project.Planned_Shipment_Time > upper_limit1,␣
↪True, np.where(project.Planned_Shipment_Time < lower_limit1, True, False))
```

```
# outliers data
project_out2 = project.loc[outliers_project2, ]
project_trimmed2 = project.loc[~(outliers_project2), ]
project.shape, project_trimmed2.shape
```

[18]: ((7999, 15), (7999, 15))

[19]:
```
# Let's explore outliers in the trimmed dataset
sns.boxplot(project_trimmed2.Actual_Shipment_Time)
```

[19]: <Axes: >

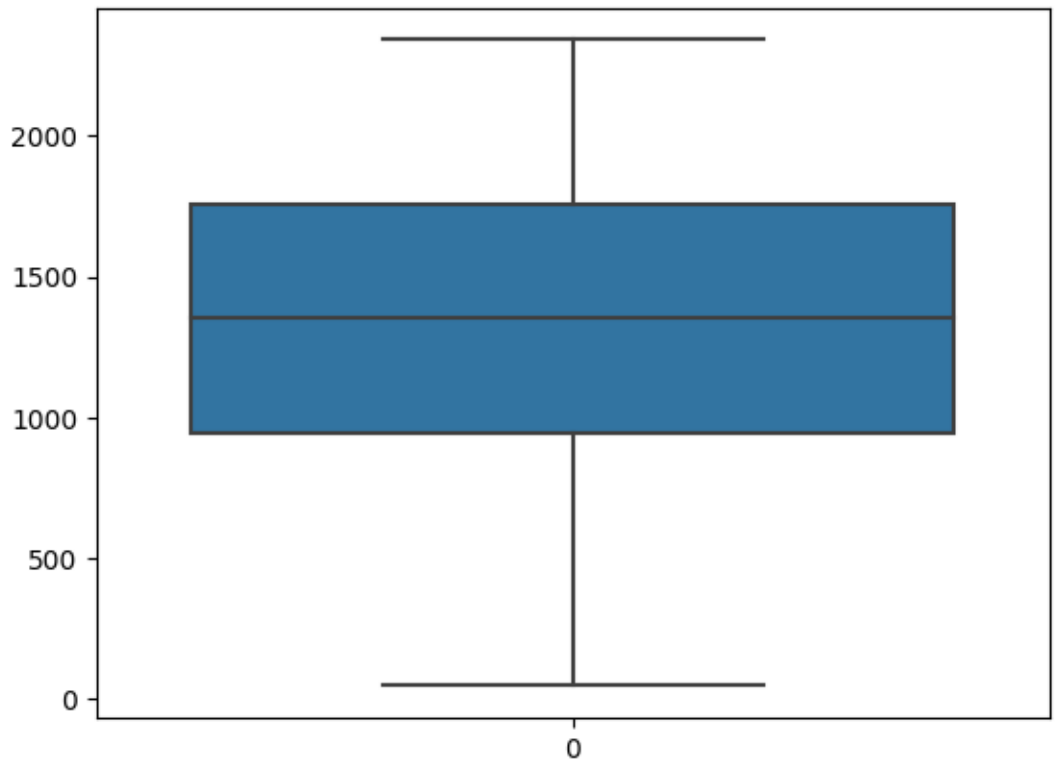


[20]:
```
outliers_project3 = np.where(project.Planned_Delivery_Time > upper_limit1,␣
↪True, np.where(project.Planned_Delivery_Time < lower_limit1, True, False))

# outliers data
project_out3 = project.loc[outliers_project3, ]
project_trimmed3 = project.loc[~(outliers_project3), ]
project.shape, project_trimmed3.shape
```

[20]: ((7999, 15), (7999, 15))

[21]:
```
# Let's explore outliers in the trimmed dataset
sns.boxplot(project_trimmed3.Actual_Shipment_Time)
```

[21]: <Axes: >

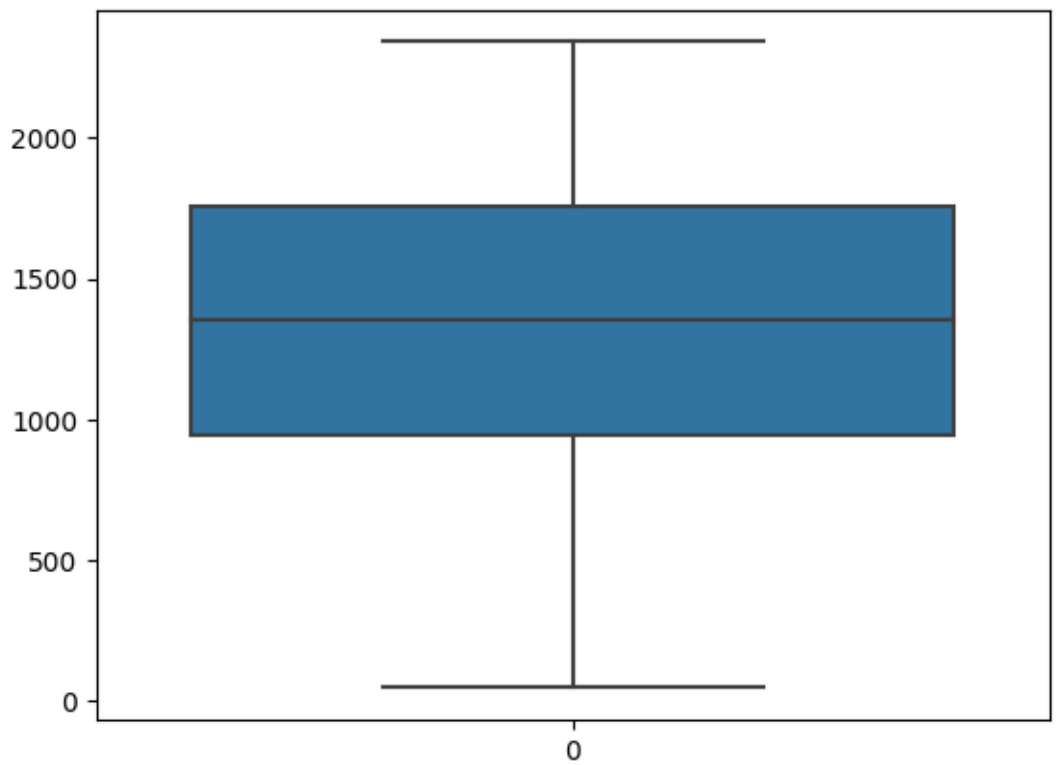


2. Replace

27 Replace the outliers by the maximum and minimum limit

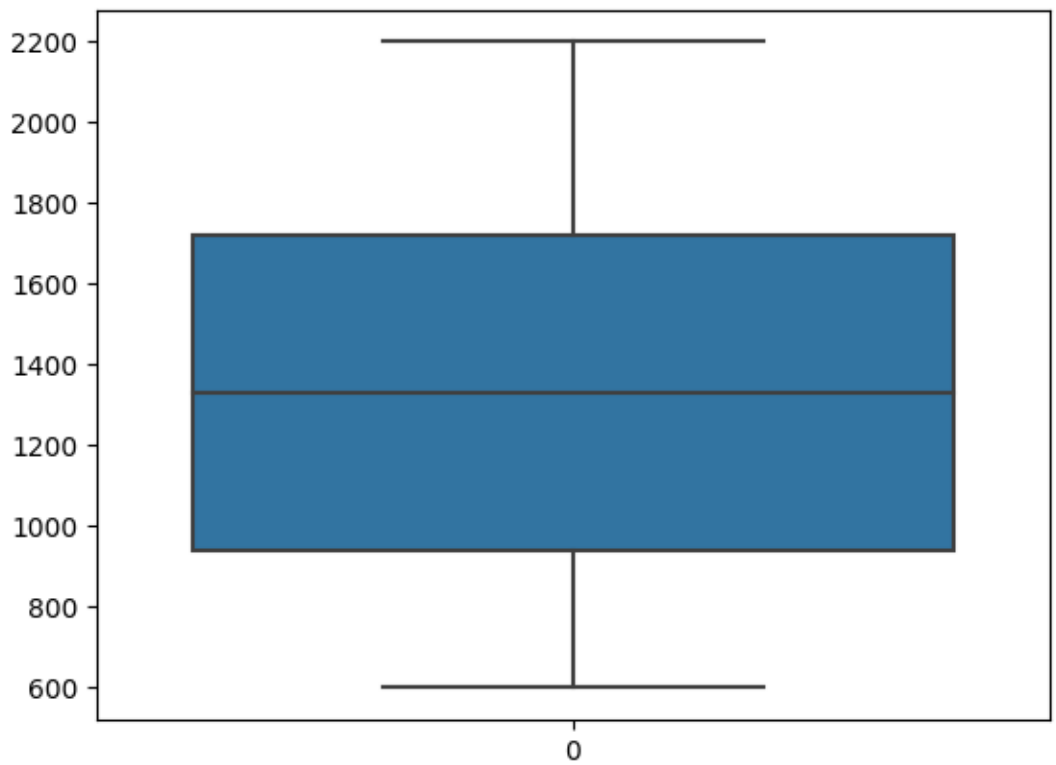
[22]:
```
project['project_replaced1'] = pd.DataFrame(np.
↪where(project['Actual_Shipment_Time'] > upper_limit1, upper_limit1, np.
↪where(project['Actual_Shipment_Time'] < lower_limit1, lower_limit1,␣
↪project['Actual_Shipment_Time'])))
sns.boxplot(project.project_replaced1)
```

[22]: <Axes: >



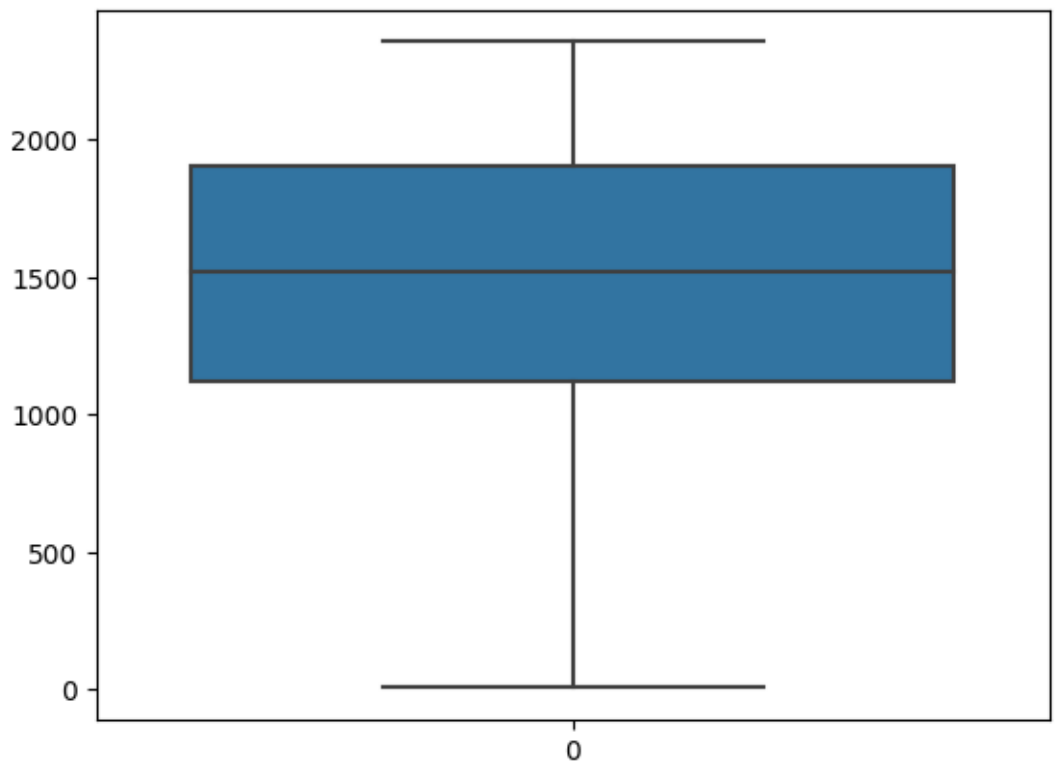
[23]:
```
project['project_replaced2'] = pd.DataFrame(np.
↪where(project['Planned_Shipment_Time'] > upper_limit2, upper_limit2, np.
↪where(project['Planned_Shipment_Time'] < lower_limit2, lower_limit2,␣
↪project['Planned_Shipment_Time'])))
sns.boxplot(project.project_replaced2)
```

[23]: <Axes: >



[24]:
```
project['project_replaced3'] = pd.DataFrame(np.
↪where(project['Planned_Delivery_Time'] > upper_limit1, upper_limit1, np.
↪where(project['Planned_Delivery_Time'] < lower_limit1, lower_limit1,␣
↪project['Planned_Delivery_Time'])))
sns.boxplot(project.project_replaced3)
```

[24]: <Axes: >



Dummy Variables

[25]:
```
import pandas as pd
import numpy as np
```

[26]:
```
project = pd.read_csv(r"/content/Datasets.csv")
```

[27]:
```
project.columns
```

[27]:
```
Index(['Year', 'Month', 'DayofMonth', 'DayOfWeek', 'Actual_Shipment_Time',
       'Planned_Shipment_Time', 'Planned_Delivery_Time', 'Carrier_Name',
       'Carrier_Num', 'Planned_TimeofTravel', 'Shipment_Delay', 'Source',
       'Destination', 'Distance', 'Delivery_Status'],
      dtype='object')
```

[28]:
```
project.shape
```

[28]:
```
(7999, 15)
```

[29]:
```
project.dtypes
```

[29]:
```
Year                       int64
Month                      int64
DayofMonth                 int64
DayOfWeek                  int64
Actual_Shipment_Time     float64
Planned_Shipment_Time      int64
Planned_Delivery_Time      int64
Carrier_Name              object
Carrier_Num                int64
Planned_TimeofTravel       int64
Shipment_Delay           float64
Source                    object
Destination               object
Distance                   int64
Delivery_Status          float64
dtype: object
```

[30]:
```python
project.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7999 entries, 0 to 7998
Data columns (total 15 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Year                   7999 non-null   int64
 1   Month                  7999 non-null   int64
 2   DayofMonth             7999 non-null   int64
 3   DayOfWeek              7999 non-null   int64
 4   Actual_Shipment_Time   7860 non-null   float64
 5   Planned_Shipment_Time  7999 non-null   int64
 6   Planned_Delivery_Time  7999 non-null   int64
 7   Carrier_Name           7999 non-null   object
 8   Carrier_Num            7999 non-null   int64
 9   Planned_TimeofTravel   7999 non-null   int64
 10  Shipment_Delay         7860 non-null   float64
 11  Source                 7999 non-null   object
 12  Destination            7999 non-null   object
 13  Distance               7999 non-null   int64
 14  Delivery_Status        7860 non-null   float64
dtypes: float64(3), int64(9), object(3)
memory usage: 937.5+ KB
```

[31]:
```python
# Drop Actual_Shipment_Time column

project1 = project.drop(['Carrier_Name' , 'Planned_Shipment_Time' ,␣
 ↪'Planned_Delivery_Time'], axis = 1)
```

```
project.drop(['Carrier_Name' , 'Planned_Shipment_Time' ,␣
↪'Planned_Delivery_Time'], axis = 1, inplace = True)
```

[32]:
```
# Create dummy variables
project_new = pd.get_dummies(project)
project_new_1 = pd.get_dummies(project, drop_first = True)
```

[34]:
```
# Created dummies for all categorical columns
##### One Hot Encoding works
project.columns
project = project[['Year', 'Month', 'DayofMonth', 'DayOfWeek',␣
↪'Actual_Shipment_Time',
          'Carrier_Num', 'Planned_TimeofTravel', 'Shipment_Delay', 'Source',
       'Destination', 'Distance', 'Delivery_Status']]
```

[35]:
```
a = project['DayofMonth']
b = project[['DayofMonth']]
```

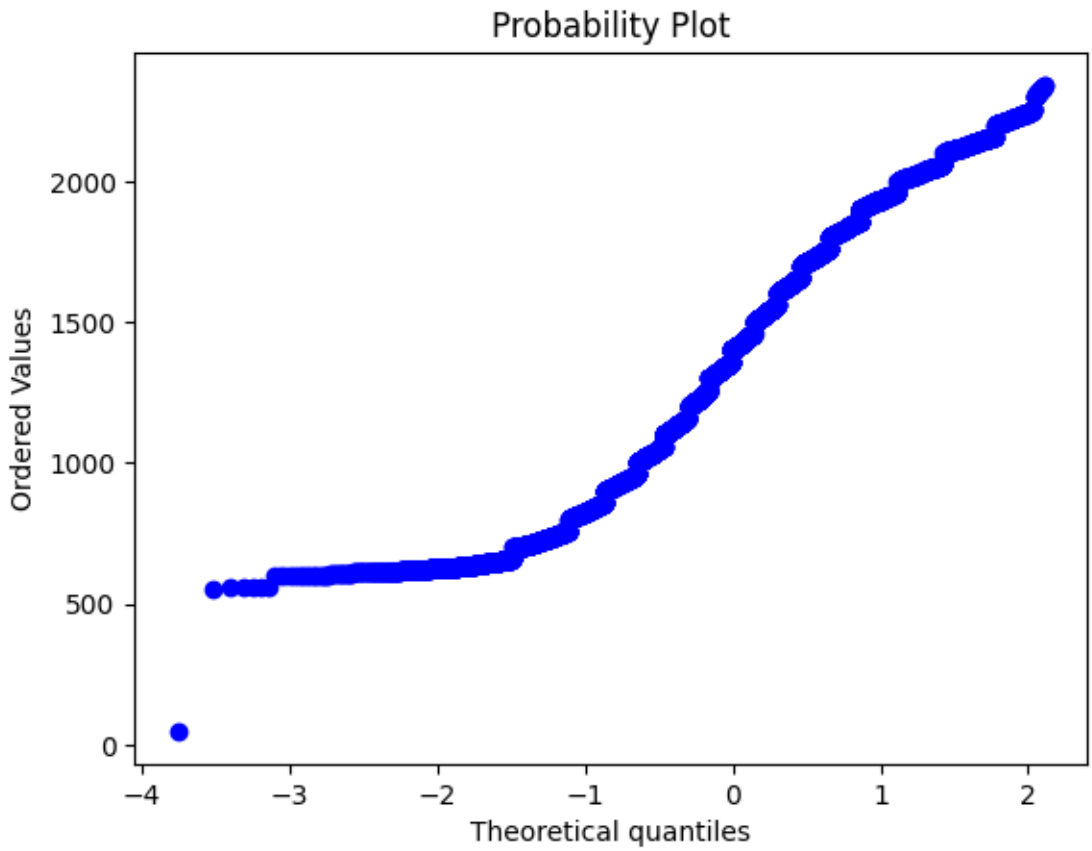
Transformation

[36]:
```
import pandas as pd
import numpy as np
import scipy.stats as stats
import pylab
```

[37]:
```
project = pd.read_csv(r"/content/Datasets.csv")
```

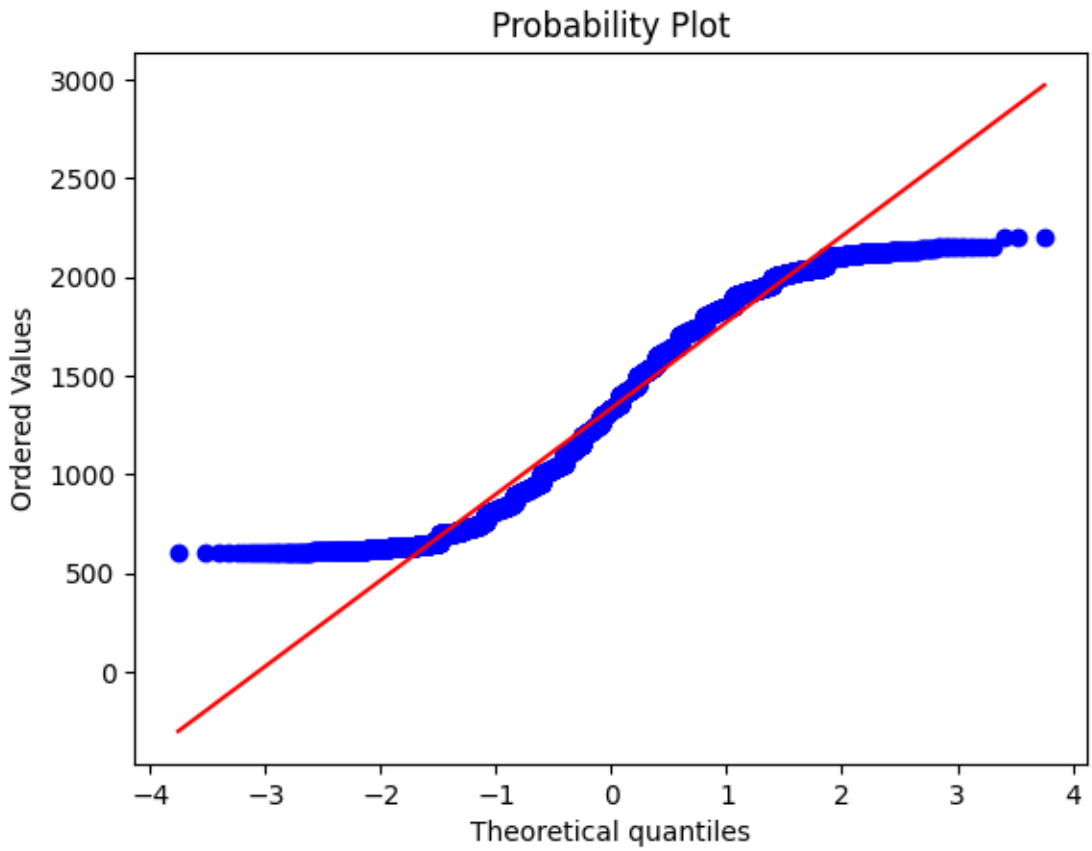
[38]:
```
#  normally distributed
stats.probplot(project.Actual_Shipment_Time, dist = "norm", plot = pylab)
```

[38]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
  array([ 47., 555., 558., …,  nan,  nan,  nan])),
 (nan, nan, nan))
```



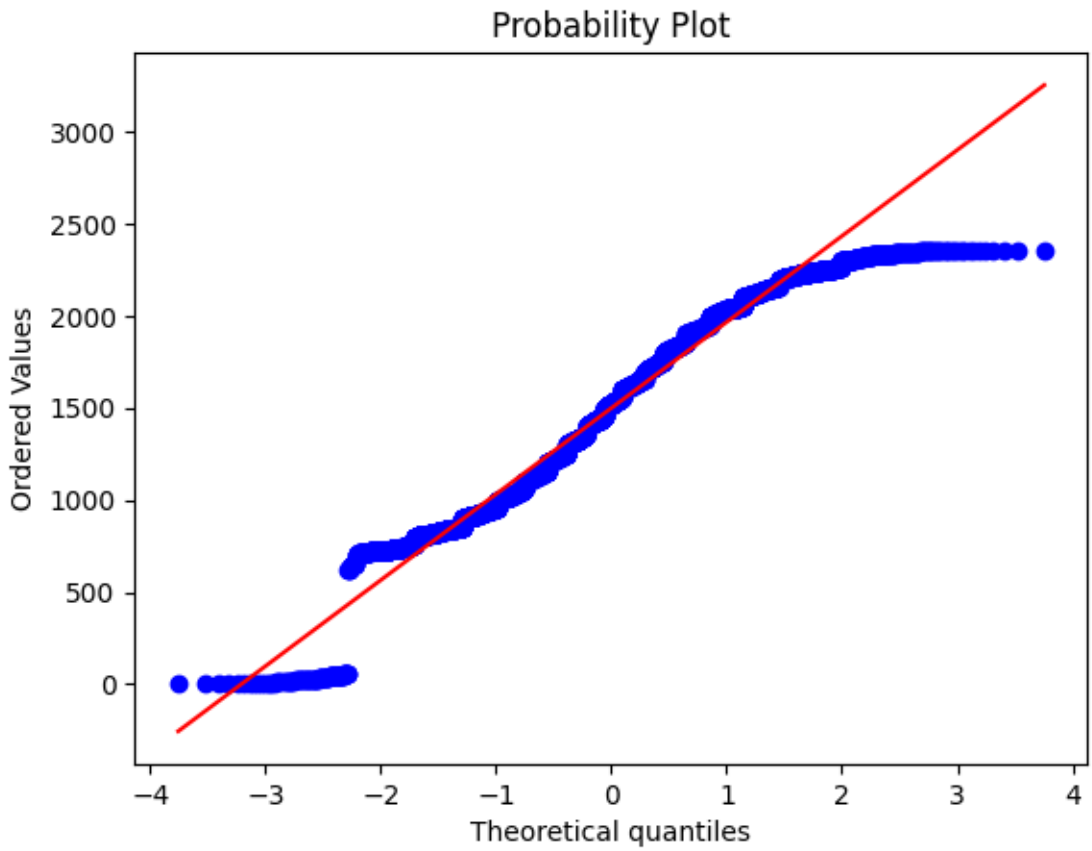
[39]:
```
stats.probplot(project.Planned_Shipment_Time, dist = "norm", plot = pylab)
```

[39]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
  array([ 600,  600,  600, …, 2200, 2200, 2200])),
 (435.96579572678934, 1335.3175396924617, 0.9768036704470942))
```



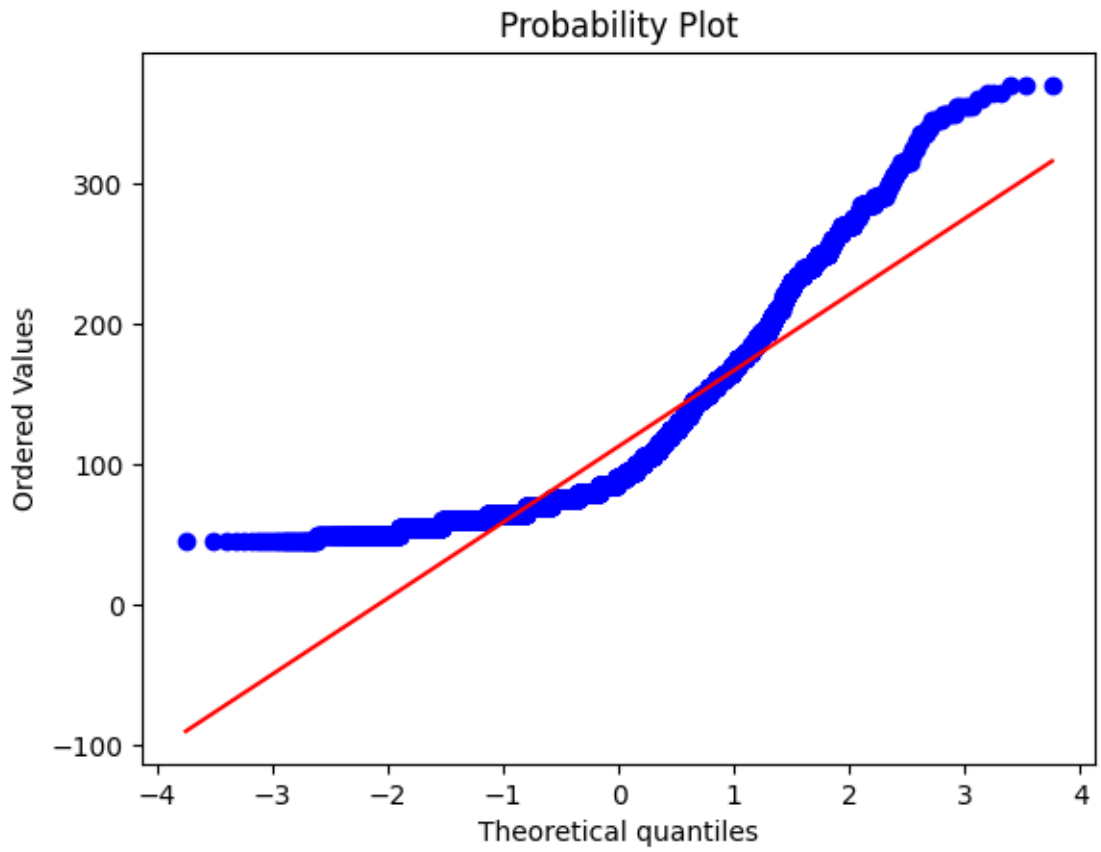
[42]:
```
stats.probplot(project.Planned_Delivery_Time, dist = "norm", plot = pylab)
```

[42]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
  array([   5,    5,    5, …, 2355, 2355, 2355])),
 (466.9271375575307, 1498.2554069258658, 0.9851475869914547))
```



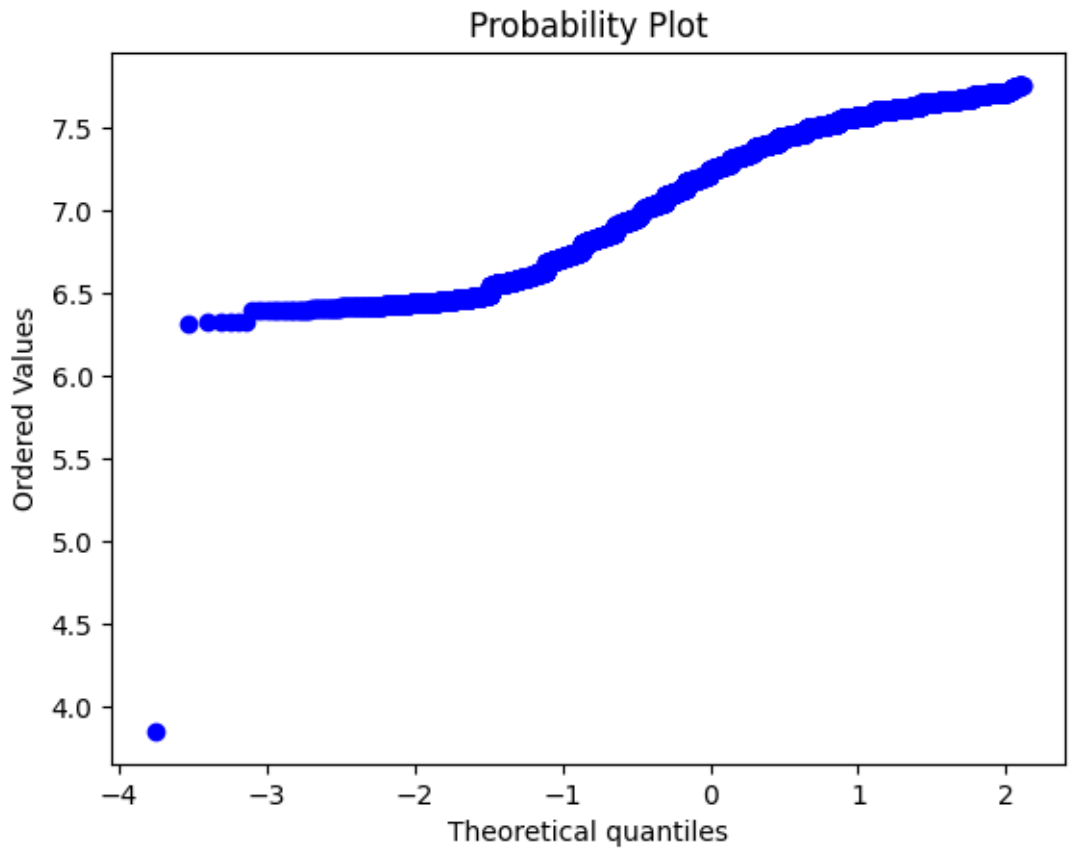
[41]:
```
stats.probplot(project.Planned_TimeofTravel, dist = "norm", plot = pylab)
```

[41]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
  array([ 45,  45,  45, …, 370, 370, 370])),
 (54.05385289105414, 112.89911238904863, 0.9194687259293179))
```



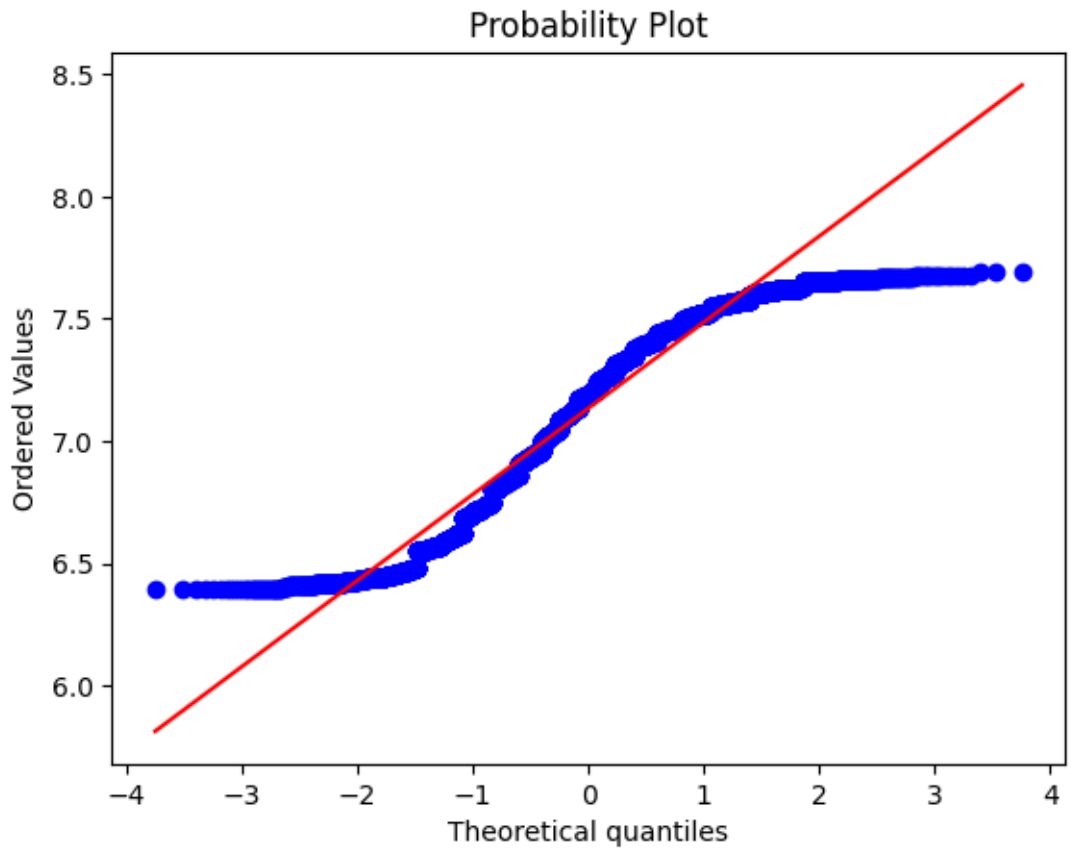
[46]:
```
# log Transformation

stats.probplot (np.log(project.Actual_Shipment_Time), dist = "norm", plot =␣
↪pylab)
```

[46]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
         3.52677228,  3.75505857]),
  array([3.8501476 , 6.31896811, 6.32435896, …,        nan,        nan,
                nan])),
 (nan, nan, nan))
```



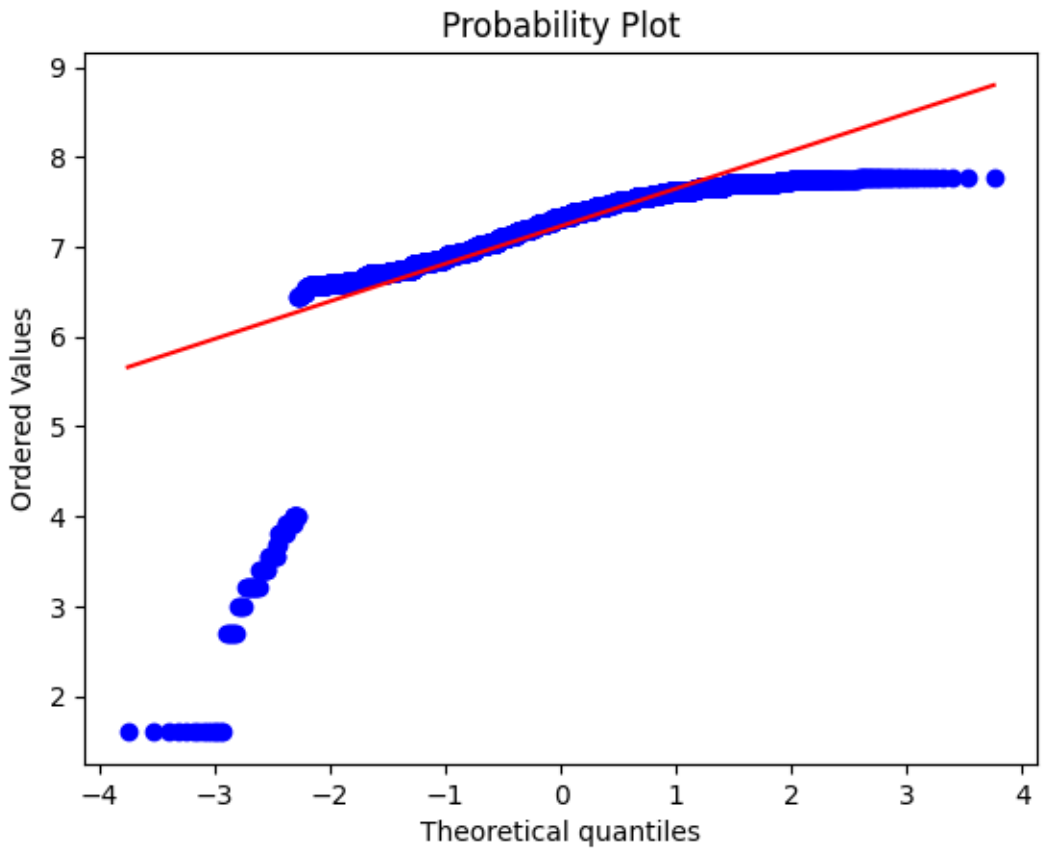
[45]:
```
stats.probplot (np.log(project.Planned_Shipment_Time), dist = "norm", plot =␣
↪pylab)
```

[45]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
  array([6.39692966, 6.39692966, 6.39692966, …, 7.69621264, 7.69621264,
         7.69621264])),
 (0.3516247486607927, 7.1351139383757465, 0.9706500937589305))
```



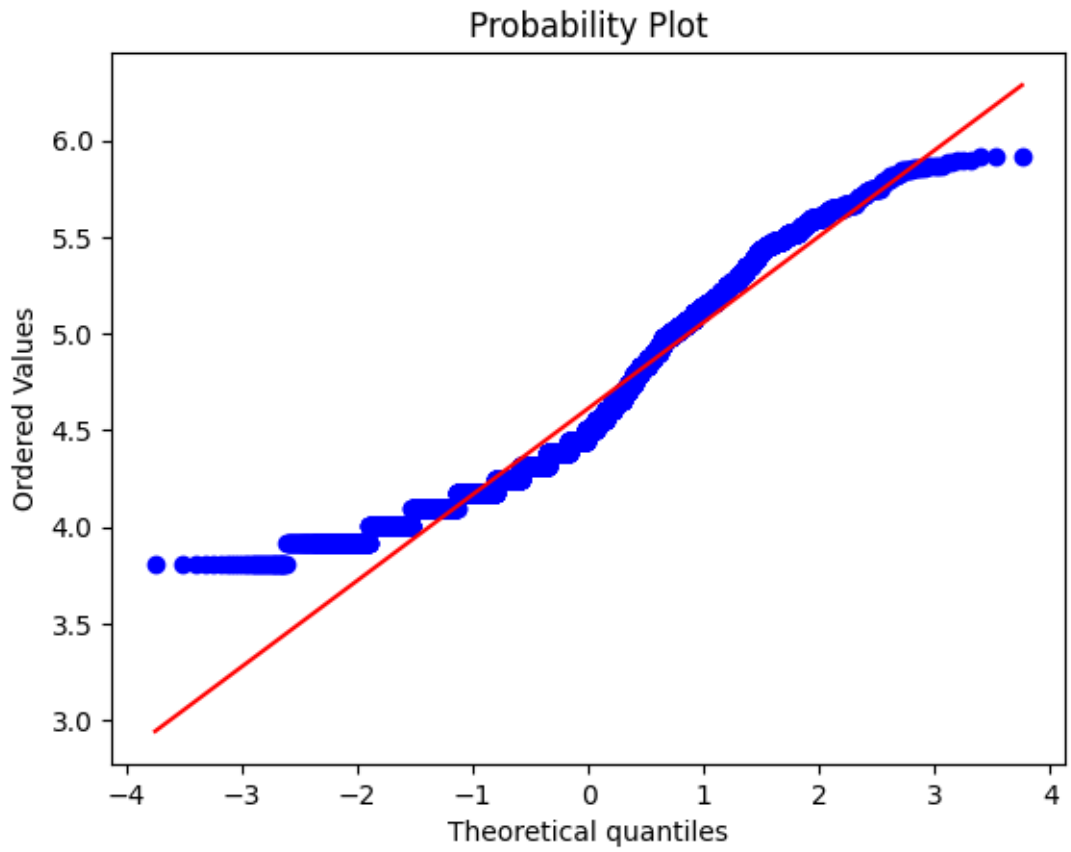
[47]:
```
stats.probplot(np.log(project.Planned_Delivery_Time), dist = "norm", plot =␣
↪pylab)
```

[47]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
 array([1.60943791, 1.60943791, 1.60943791, …, 7.76429601, 7.76429601,
        7.76429601])),
 (0.41791891269380516, 7.2296711133439935, 0.7793248784604291))
```



[48]:
```
stats.probplot(np.log(project.Planned_TimeofTravel), dist = "norm", plot =␣
↪pylab)
```

[48]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
 array([3.80666249, 3.80666249, 3.80666249, …, 5.91350301, 5.91350301,
        5.91350301])),
 (0.4452819531294663, 4.613987512354753, 0.973204139040742))
```



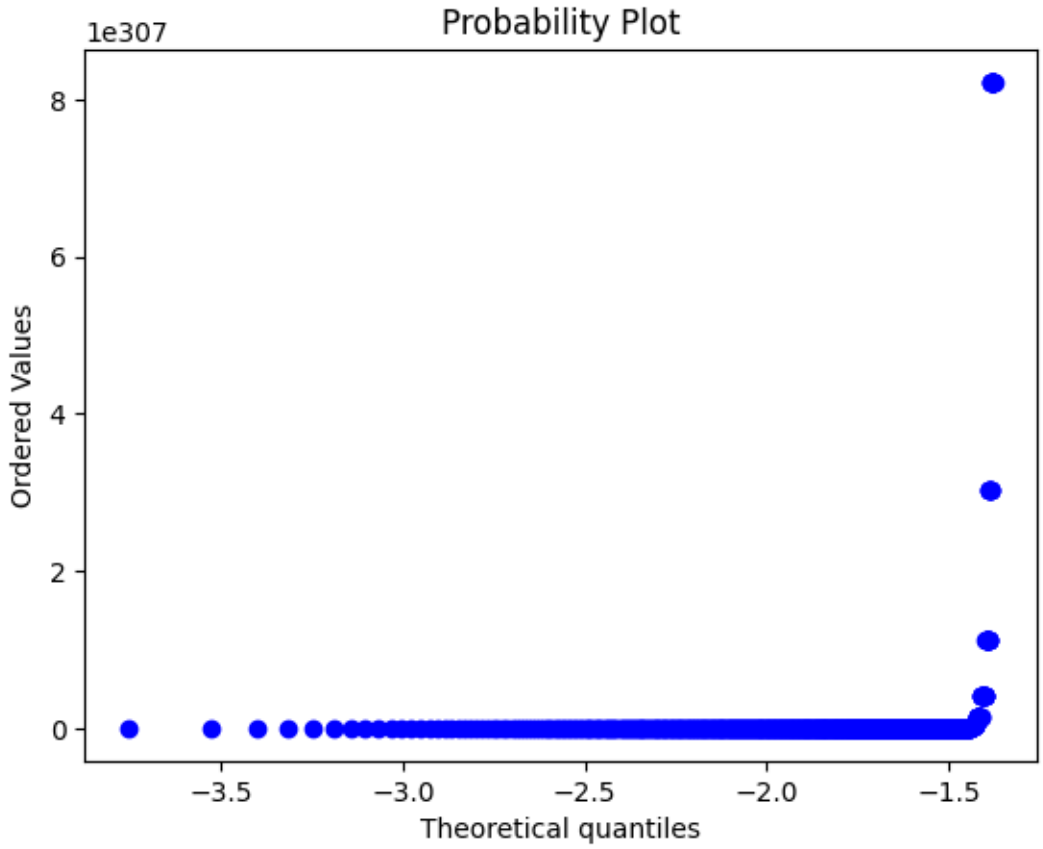
[49]:
```
# exp Transformation

stats.probplot (np.exp(project.Actual_Shipment_Time), dist = "norm", plot =␣
↪pylab)
```

```
/usr/local/lib/python3.10/dist-packages/pandas/core/arraylike.py:402:
RuntimeWarning: overflow encountered in exp
  result = getattr(ufunc, method)(*inputs, **kwargs)
/usr/local/lib/python3.10/dist-packages/numpy/core/_methods.py:180:
RuntimeWarning: overflow encountered in reduce
  ret = umr_sum(arr, axis, dtype, out, keepdims, where=where)
```

[49]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
         3.52677228,  3.75505857]),
 array([2.58131289e+020, 1.08003407e+241, 2.16930642e+242, …,
                    nan,             nan,             nan])),
 (nan, nan, nan))
```

```
/usr/local/lib/python3.10/dist-packages/matplotlib/ticker.py:2094:
RuntimeWarning: overflow encountered in multiply
```

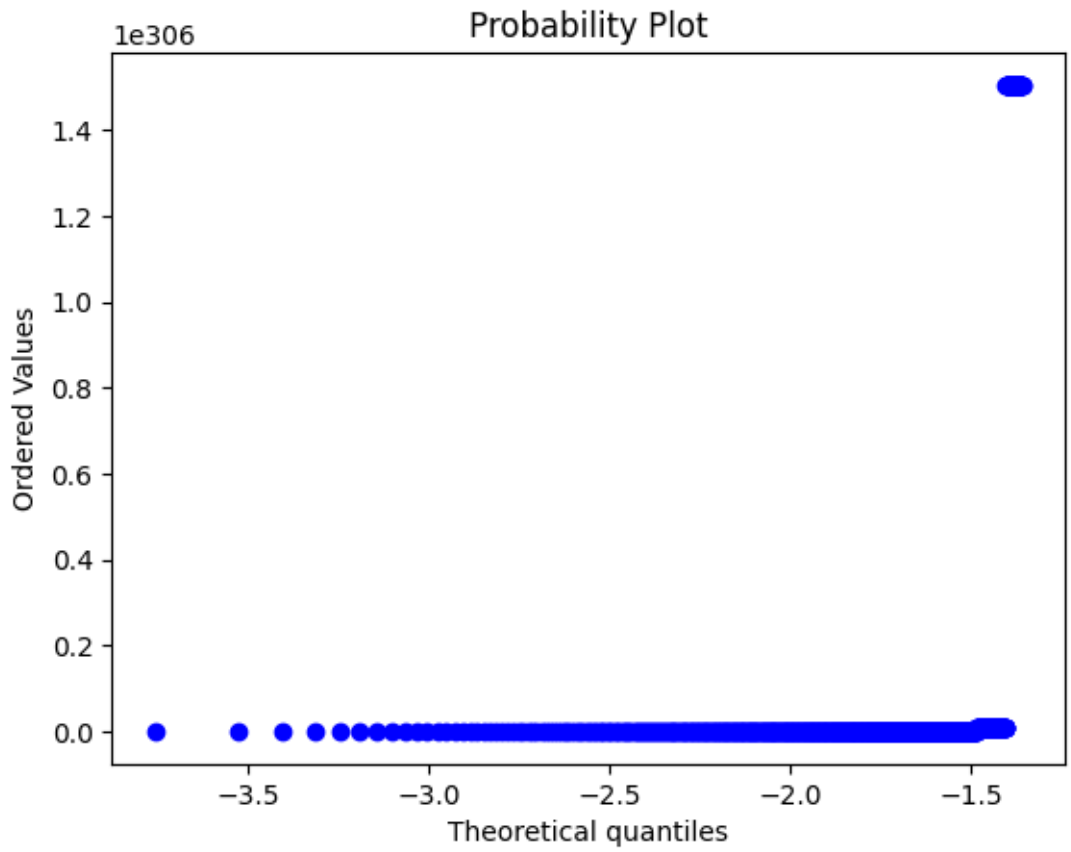
```
steps = self._extended_steps * scale
```



[50]:
```
stats.probplot (np.exp(project.Planned_Shipment_Time), dist = "norm", plot =␣
↪pylab)
```

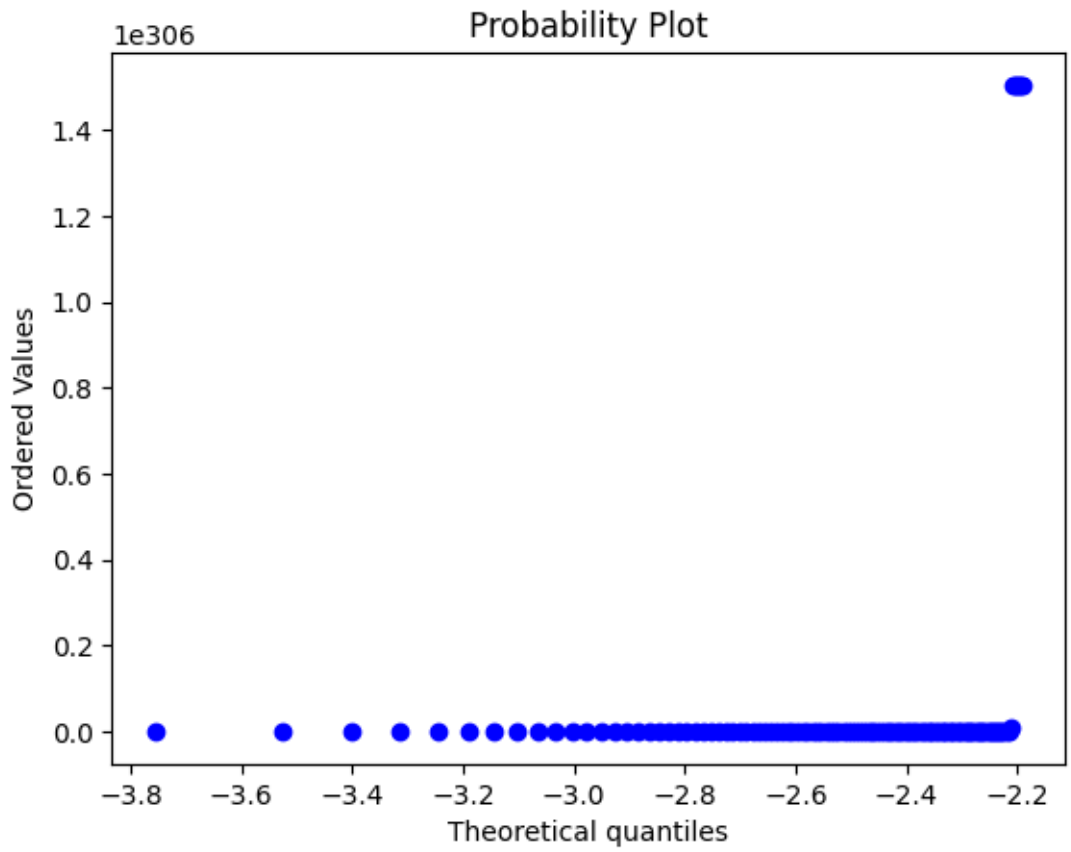
```
/usr/local/lib/python3.10/dist-packages/numpy/lib/function_base.py:2698:
RuntimeWarning: invalid value encountered in subtract
  X -= avg[:, None]
```

[50]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
 array([3.7730203e+260, 3.7730203e+260, 3.7730203e+260, …,
                   inf,            inf,            inf])),
 (nan, nan, nan))
```



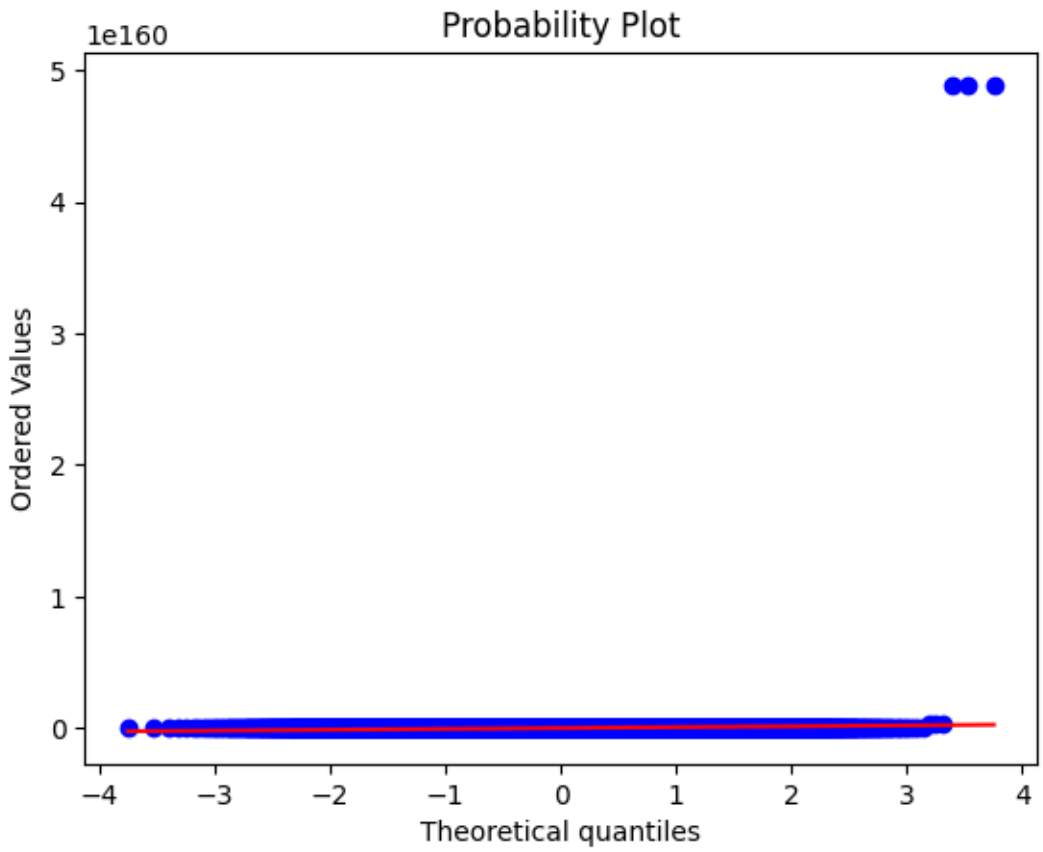
[51]:
```
stats.probplot(np.exp(project.Planned_Delivery_Time), dist = "norm", plot =␣
↪pylab)
```

[51]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
 array([148.4131591, 148.4131591, 148.4131591, …,         inf,
                inf,         inf])),
 (nan, nan, nan))
```



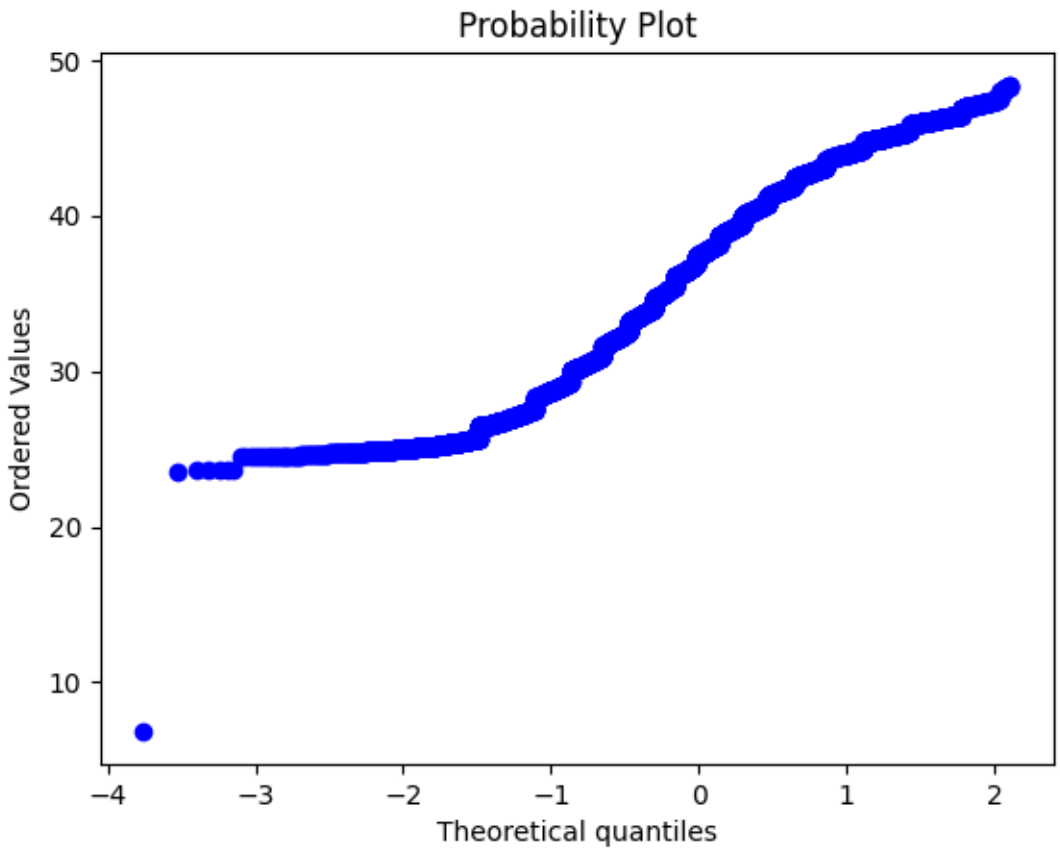
[52]:
```
stats.probplot(np.exp(project.Planned_TimeofTravel), dist = "norm", plot =␣
↪pylab)
```

[52]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
  array([3.49342711e+019, 3.49342711e+019, 3.49342711e+019, …,
         4.88605447e+160, 4.88605447e+160, 4.88605447e+160])),
 (6.571660879246802e+157, 1.844903365428962e+157, 0.0))
```



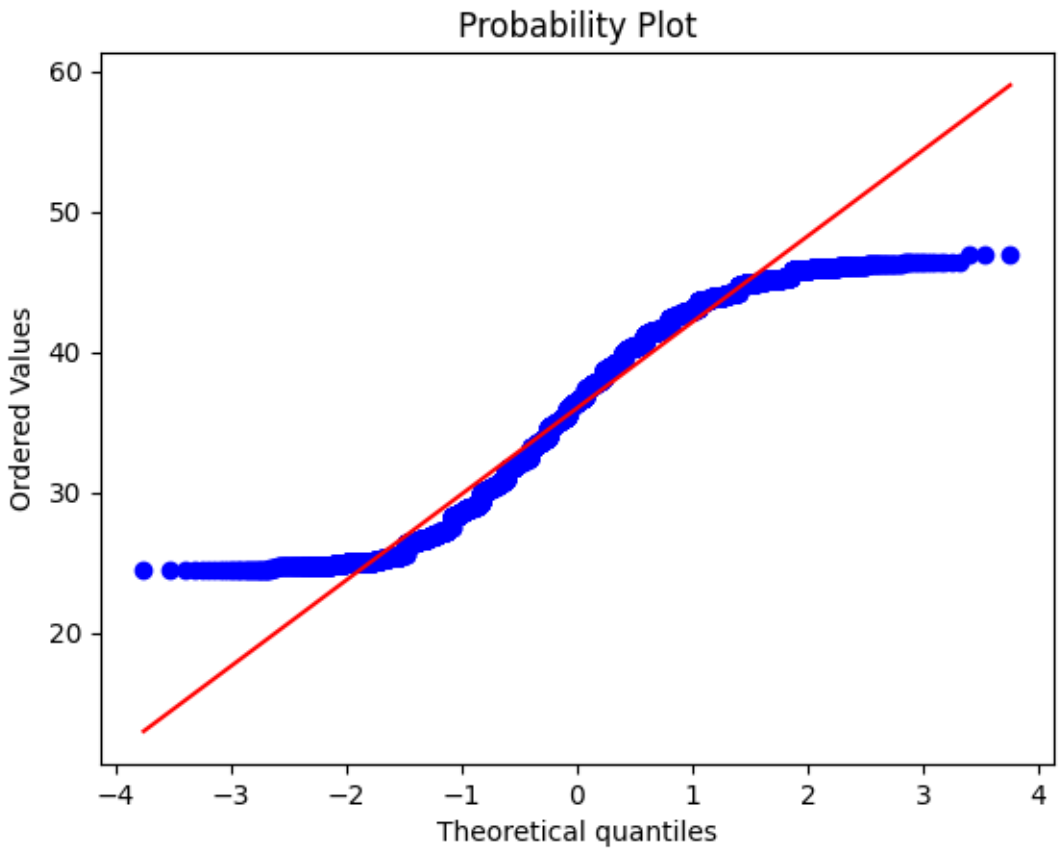
[53]:
```
# sqrt Transformation

stats.probplot (np.sqrt(project.Actual_Shipment_Time), dist = "norm", plot =␣
↪pylab)
```

[53]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
         3.52677228,  3.75505857]),
  array([ 6.8556546 , 23.55843798, 23.62202362, …,         nan,
                nan,         nan])),
 (nan, nan, nan))
```



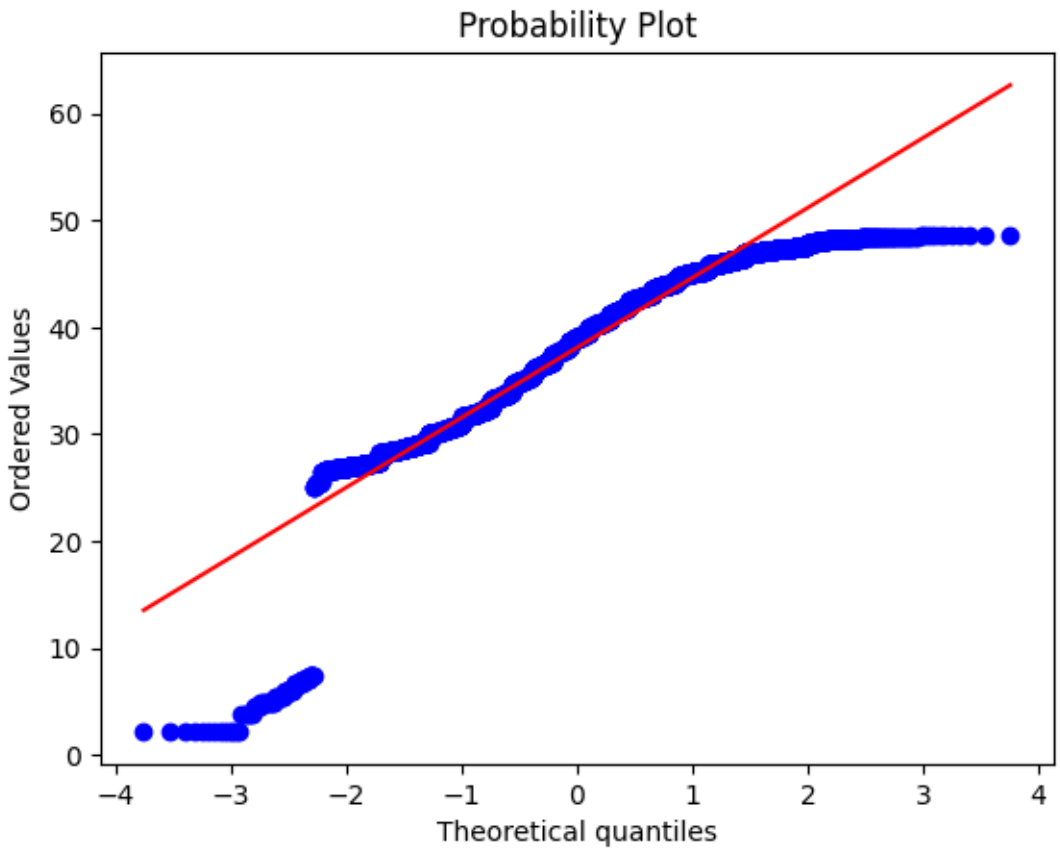
[54]:
```
stats.probplot (np.sqrt(project.Planned_Shipment_Time), dist = "norm", plot =␣
↪pylab)
```

[54]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
  array([24.49489743, 24.49489743, 24.49489743, …, 46.9041576 ,
         46.9041576 , 46.9041576 ])),
 (6.1251327297262055, 35.999726417463044, 0.9761656641251333))
```



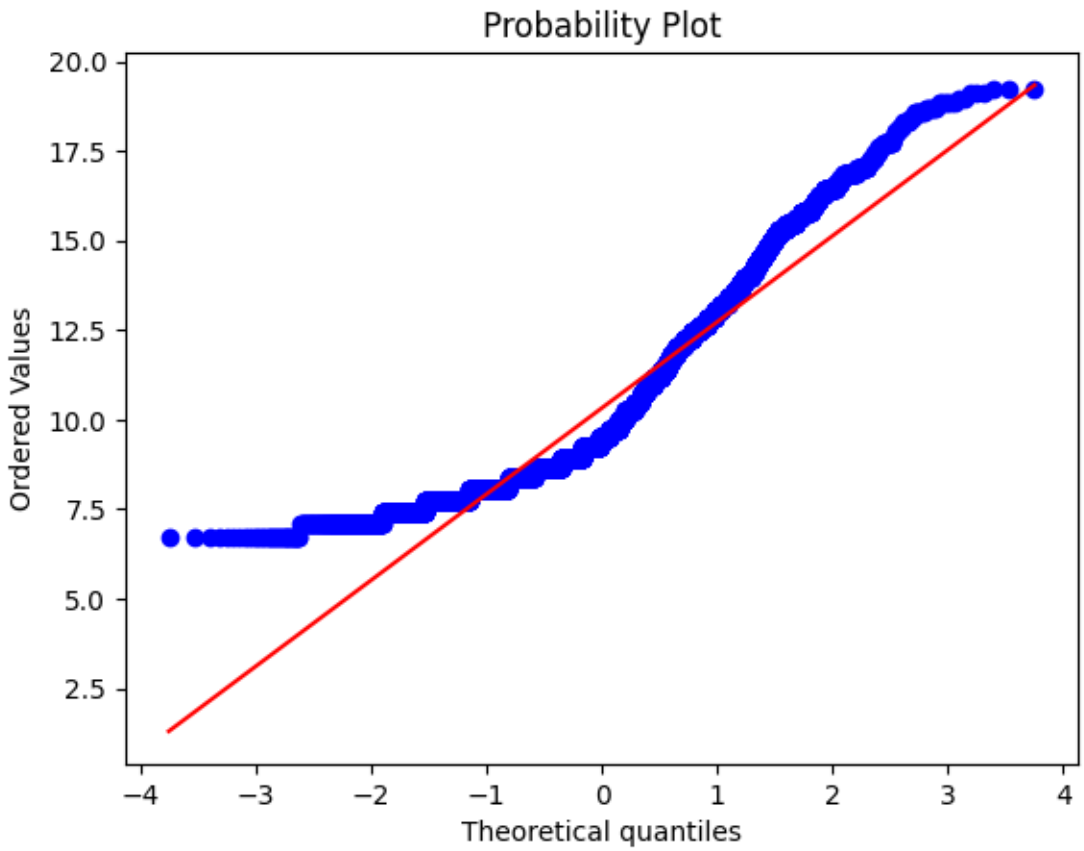
[55]:
```
stats.probplot(np.sqrt(project.Planned_Delivery_Time), dist = "norm", plot =␣
↪pylab)
```

[55]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
  array([ 2.23606798,  2.23606798,  2.23606798, …, 48.52834223,
         48.52834223, 48.52834223])),
 (6.535515605318326, 38.097194290880964, 0.954316430509373))
```



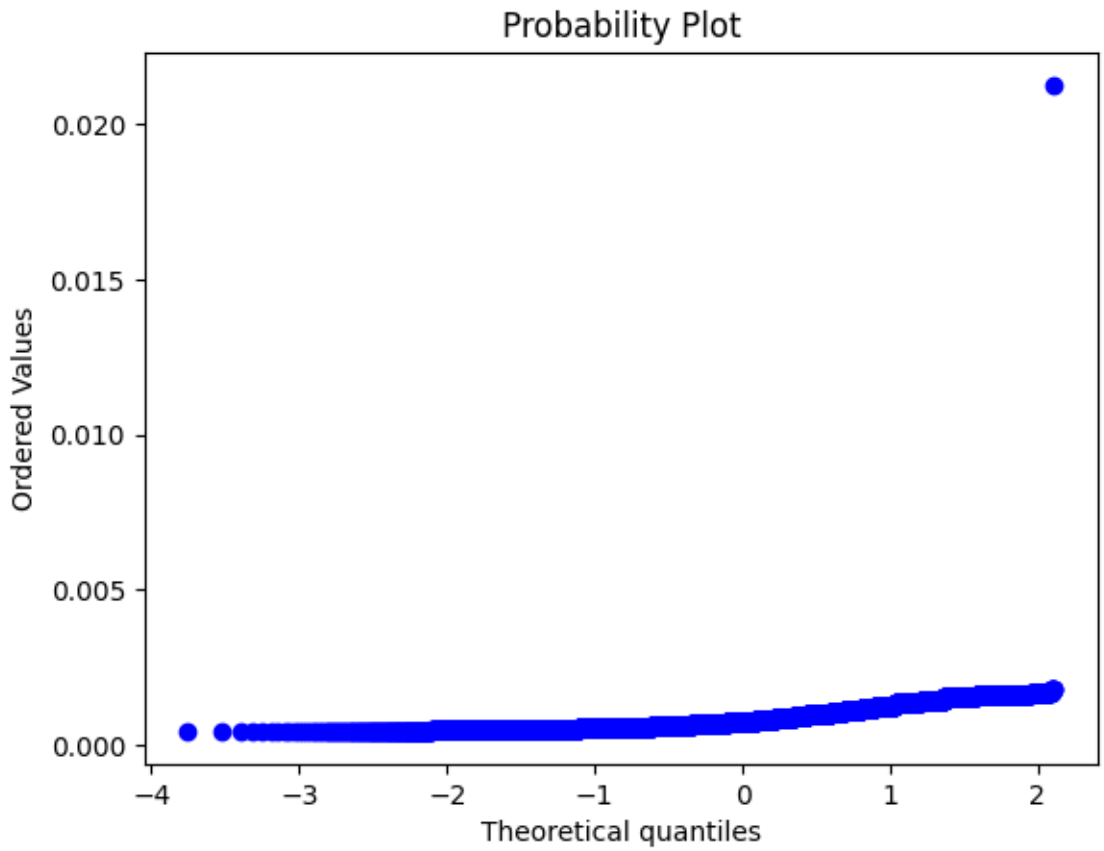
[56]:
```
stats.probplot(np.sqrt(project.Planned_TimeofTravel), dist = "norm", plot =␣
↪pylab)
```

[56]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
  array([ 6.70820393,  6.70820393,  6.70820393, …, 19.23538406,
         19.23538406, 19.23538406])),
 (2.4016185165026167, 10.321561127388817, 0.9515513514518762))
```



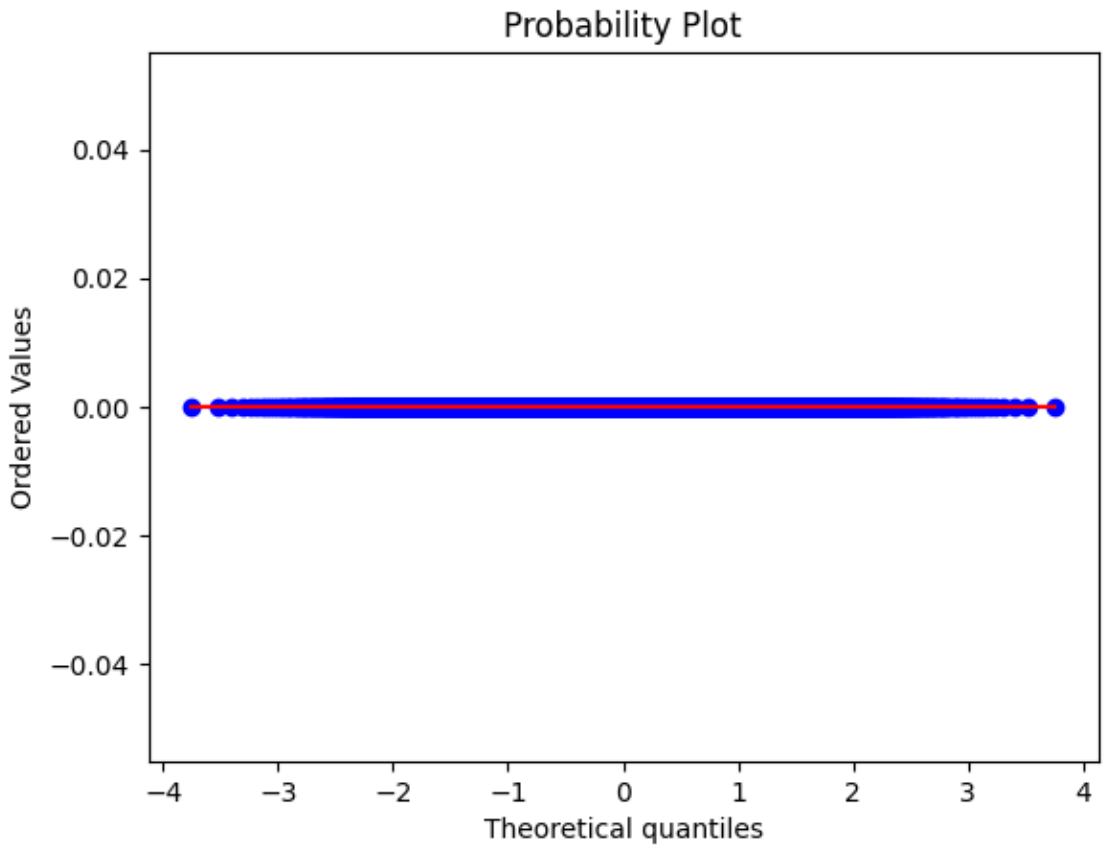
[57]:
```
# reciprocal Transformation

stats.probplot (np.reciprocal(project.Actual_Shipment_Time), dist = "norm",␣
↪plot = pylab)
```

[57]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
  array([0.00042717, 0.00042717, 0.00042845, …,        nan,        nan,
               nan])),
 (nan, nan, nan))
```



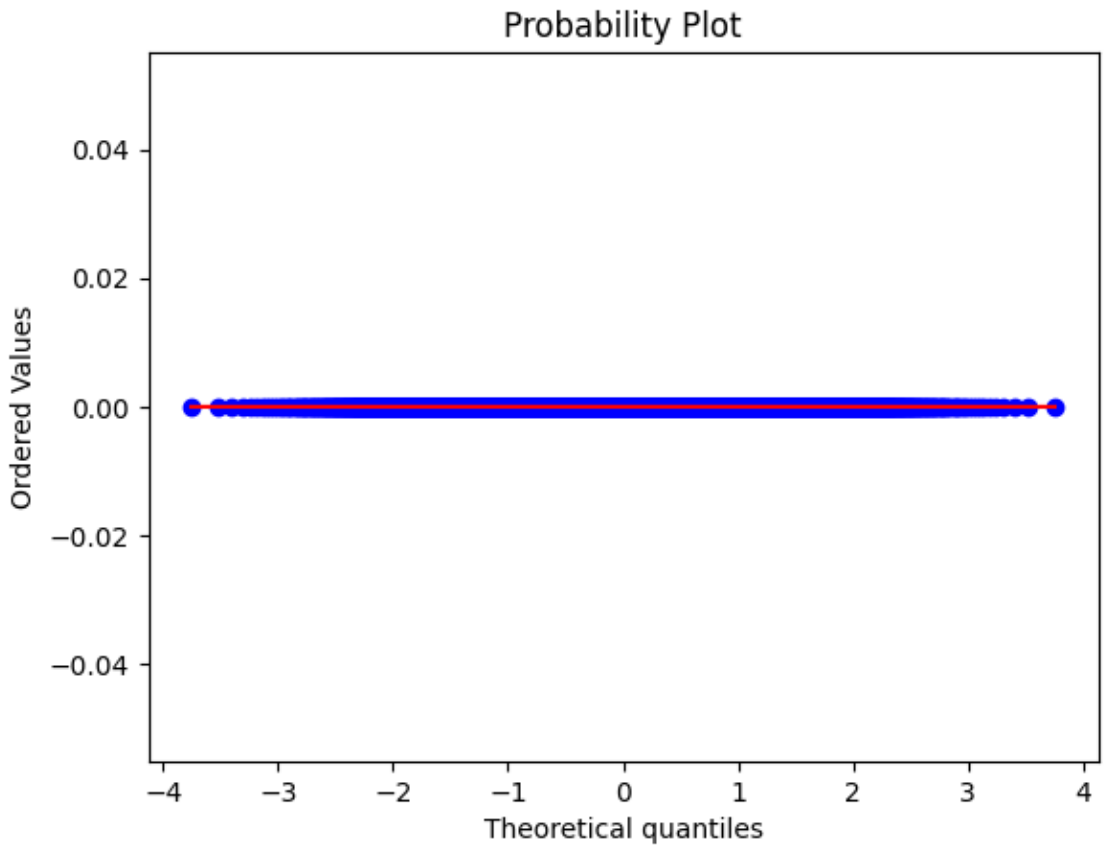
[58]:
```
stats.probplot (np.reciprocal(project.Planned_Shipment_Time), dist = "norm",␣
↪plot = pylab)
```

[58]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
 array([0, 0, 0, …, 0, 0, 0])),
(0.0, 0.0, 0.0))
```



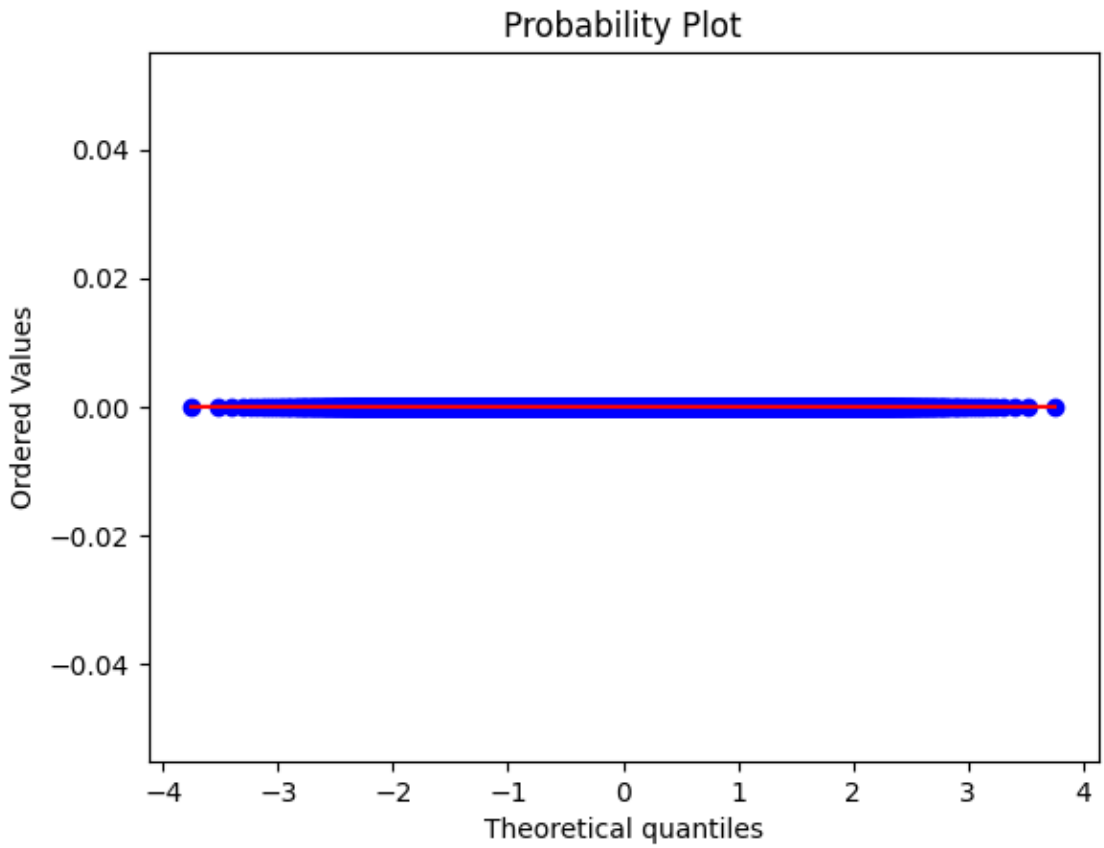
[59]:
```
stats.probplot(np.reciprocal(project.Planned_Delivery_Time), dist = "norm",␣
↪plot = pylab)
```

[59]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
  array([0, 0, 0, …, 0, 0, 0])),
 (0.0, 0.0, 0.0))
```



[60]:
```
stats.probplot(np.reciprocal(project.Planned_TimeofTravel), dist = "norm", plot␣
↪= pylab)
```

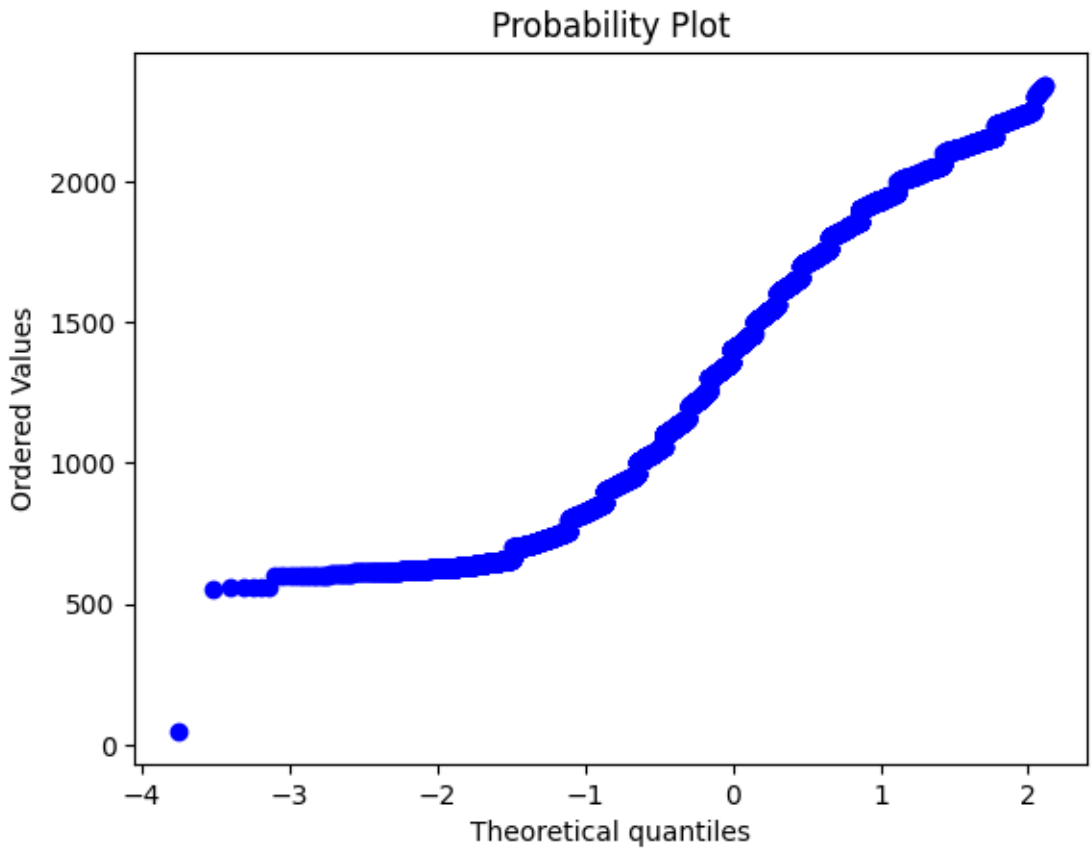
[60]:
```
((array([-3.75505857, -3.52677228, -3.40129331, …,  3.40129331,
          3.52677228,  3.75505857]),
  array([0, 0, 0, …, 0, 0, 0])),
 (0.0, 0.0, 0.0))
```



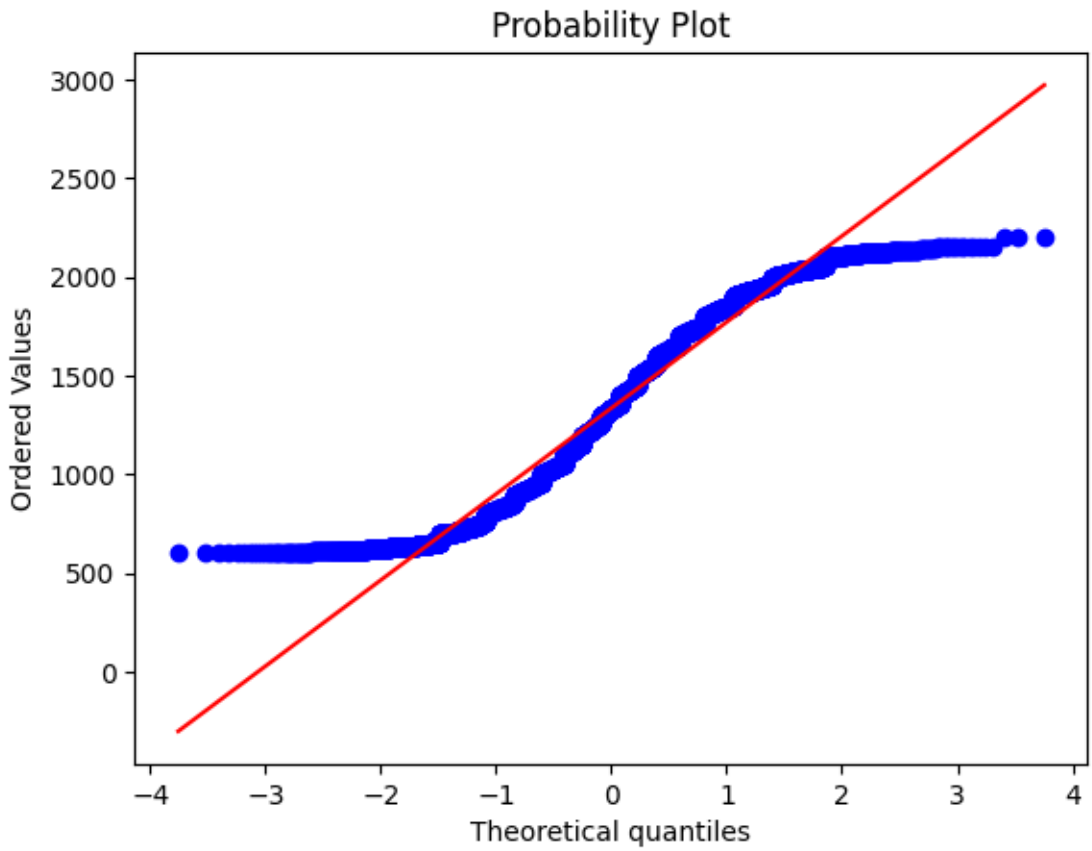
[61]:
```
# Box cox Transformation

import pandas as pd
from scipy import stats

# Plotting related modules
import seaborn as sns
import matplotlib.pyplot as plt
import pylab
```

[62]:
```
project = pd.read_csv(r"/content/Datasets.csv")
```

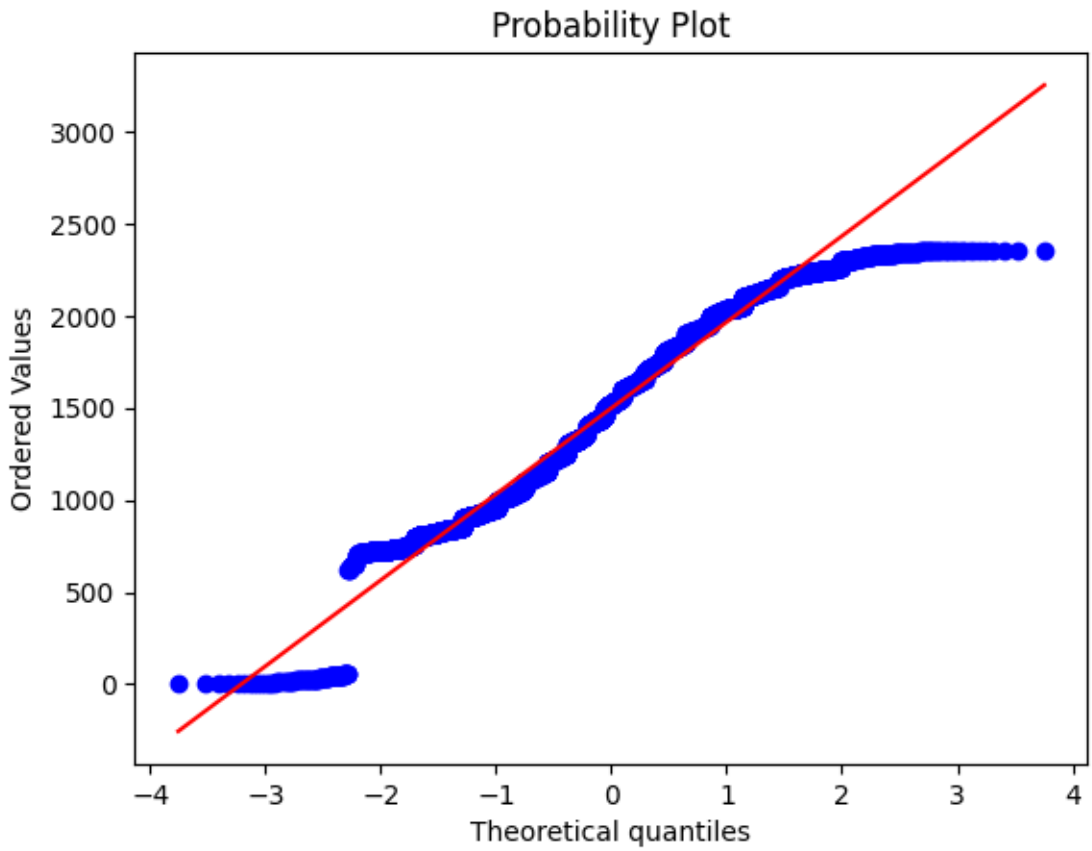
[63]:
```
# Original data
prob1 = stats.probplot(project.Actual_Shipment_Time, dist = stats.norm, plot =␣
↪pylab)
```



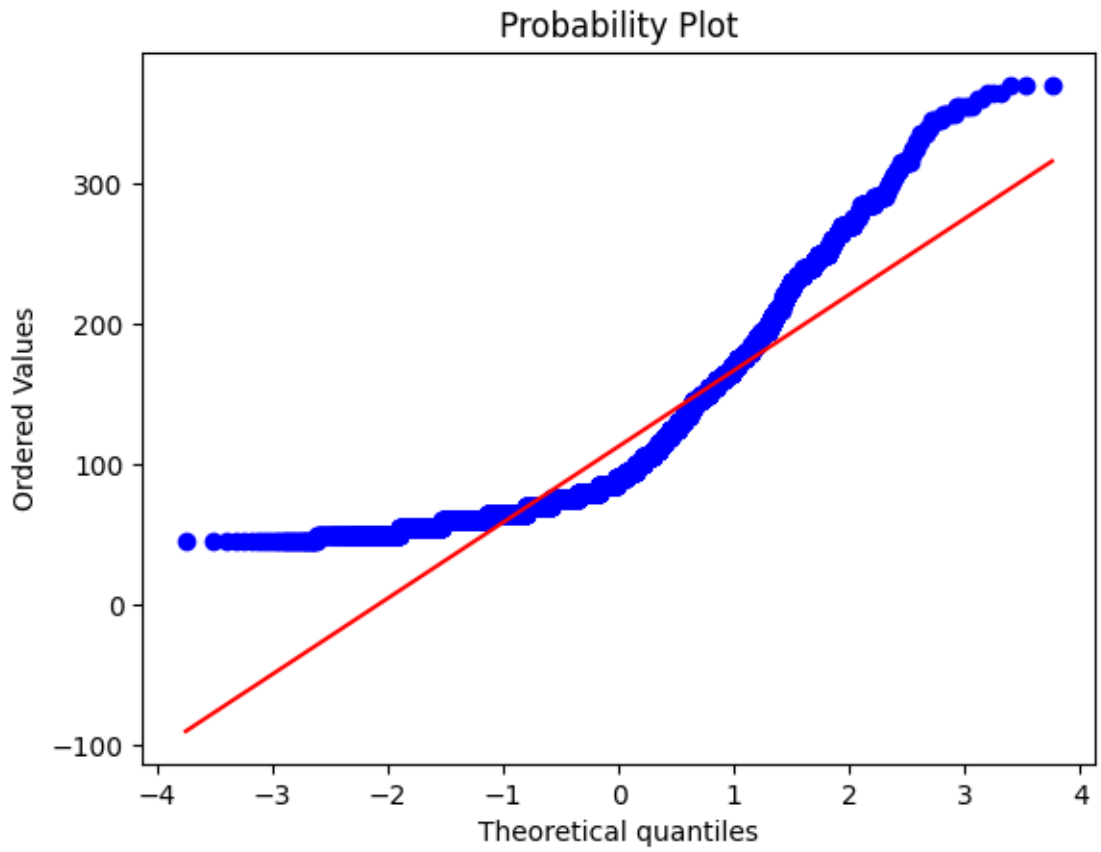
[64]:
```
prob2 = stats.probplot(project.Planned_Shipment_Time, dist = stats.norm, plot =␣
↪pylab)
```



[65]:
```
prob3 = stats.probplot(project.Planned_Delivery_Time, dist = stats.norm, plot =␣
↪pylab)
```



[66]:
```
prob4= stats.probplot(project.Planned_TimeofTravel, dist = stats.norm, plot =
 ↪pylab)
```



28 Transform training data & save lambda value

[78]:
```
fitted_data1, fitted_lambda1 = stats.boxcox(project.Actual_Shipment_Time)
```

```
---------------------------------------------------------------------------
BracketError                              Traceback (most recent call last)
<ipython-input-78-b0e564be35d7> in <cell line: 1>()
----> 1 fitted_data1, fitted_lambda1 = stats.boxcox(project.Actual_Shipment_Time)

/usr/local/lib/python3.10/dist-packages/scipy/stats/_morestats.py in boxcox(x,␣
↪lmbda, alpha, optimizer)
   1103
   1104     # If lmbda=None, find the lmbda that maximizes the log-likelihood␣
↪function.
-> 1105     lmax = boxcox_normmax(x, method='mle', optimizer=optimizer)
   1106     y = boxcox(x, lmax)
   1107
```

```
/usr/local/lib/python3.10/dist-packages/scipy/stats/_morestats.py in␣
↪boxcox_normmax(x, brack, method, optimizer)
   1274
   1275     optimfunc = methods[method]
-> 1276     res = optimfunc(x)
   1277     if res is None:
   1278         message = ("`optimizer` must return an object containing the␣
↪optimal "

/usr/local/lib/python3.10/dist-packages/scipy/stats/_morestats.py in _mle(x)
   1259             return -boxcox_llf(lmb, data)
   1260
-> 1261         return _optimizer(_eval_mle, args=(x,))
   1262
   1263     def _all(x):

/usr/local/lib/python3.10/dist-packages/scipy/stats/_morestats.py in␣
↪_optimizer(func, args)
   1221
   1222         def _optimizer(func, args):
-> 1223             return optimize.brent(func, args=args, brack=brack)
   1224
   1225     # Otherwise check optimizer.

/usr/local/lib/python3.10/dist-packages/scipy/optimize/_optimize.py in␣
↪brent(func, args, brack, tol, full_output, maxiter)
   2640     options = {'xtol': tol,
   2641                'maxiter': maxiter}
-> 2642     res = _minimize_scalar_brent(func, brack, args, **options)
   2643     if full_output:
   2644         return res['x'], res['fun'], res['nit'], res['nfev']

/usr/local/lib/python3.10/dist-packages/scipy/optimize/_optimize.py in␣
↪_minimize_scalar_brent(func, brack, args, xtol, maxiter, disp,␣
↪**unknown_options)
   2677                    full_output=True, maxiter=maxiter, disp=disp)
   2678     brent.set_bracket(brack)
-> 2679     brent.optimize()
   2680     x, fval, nit, nfev = brent.get_result(full_output=True)
   2681

/usr/local/lib/python3.10/dist-packages/scipy/optimize/_optimize.py in␣
↪optimize(self)
   2447         # set up for optimization
   2448         func = self.func
-> 2449         xa, xb, xc, fa, fb, fc, funcalls = self.get_bracket_info()
   2450         _mintol = self._mintol
   2451         _cg = self._cg
```

```
/usr/local/lib/python3.10/dist-packages/scipy/optimize/_optimize.py in␣
↪get_bracket_info(self)
   2416             xa, xb, xc, fa, fb, fc, funcalls = bracket(func, args=args)
   2417         elif len(brack) == 2:
-> 2418             xa, xb, xc, fa, fb, fc, funcalls = bracket(func, xa=brack[0],
   2419                                                        xb=brack[1],␣
↪args=args)
   2420         elif len(brack) == 3:

/usr/local/lib/python3.10/dist-packages/scipy/optimize/_optimize.py in␣
↪bracket(func, xa, xb, args, grow_limit, maxiter)
   3046         e = BracketError(msg)
   3047         e.data = (xa, xb, xc, fa, fb, fc, funcalls)
-> 3048         raise e
   3049
   3050     return xa, xb, xc, fa, fb, fc, funcalls

BracketError: The algorithm terminated without finding a valid bracket. Consider␣
↪trying different initial points.
```

[68]:
```
fitted_data2, fitted_lambda2 = stats.boxcox(project.Planned_Shipment_Time)
```

[69]:
```
fitted_data3, fitted_lambda3 = stats.boxcox(project.Planned_Delivery_Time)
```

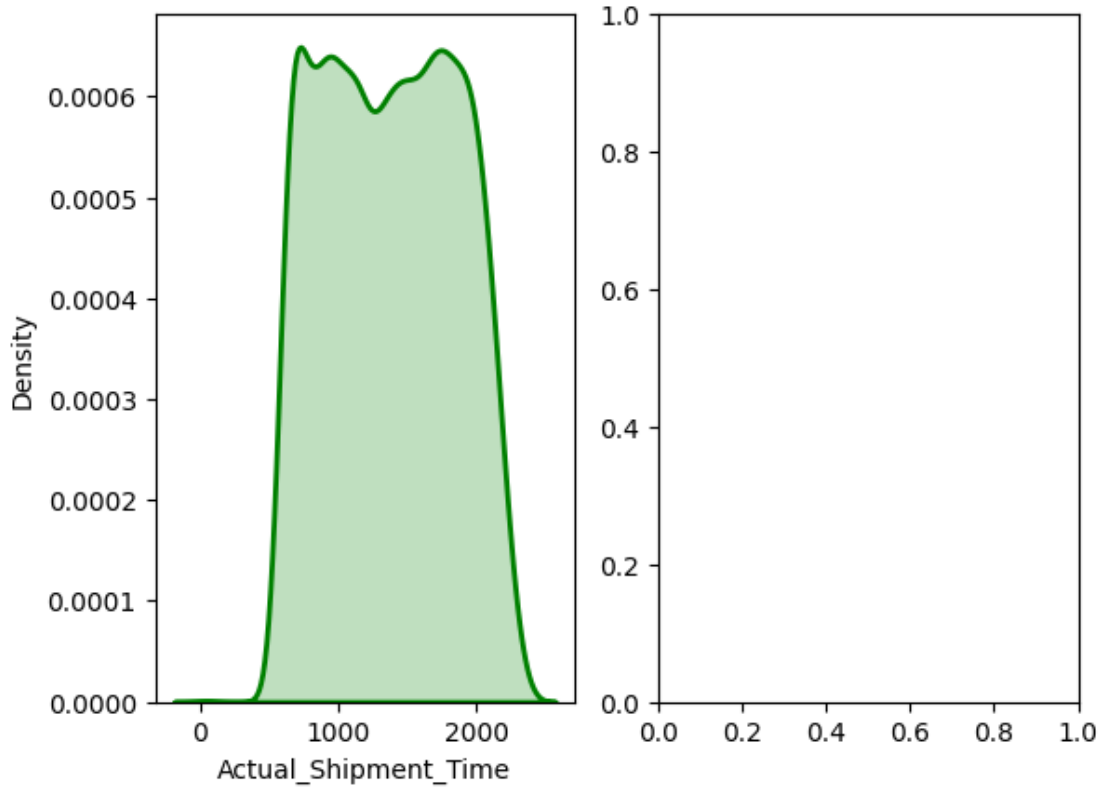
[70]:
```
fitted_data4, fitted_lambda4 = stats.boxcox(project.Planned_TimeofTravel)
```

[73]:
```
# creating axes to draw plots
fig, ax = plt.subplots(1 ,2)

# Plotting the original data (non-normal) and fitted data (normal)
sns.distplot(project.Actual_Shipment_Time, hist = False, kde = True,
             kde_kws = {'shade': True, 'linewidth': 2},
             label = "Non-Normal", color = "green", ax = ax[0])

sns.distplot(fitted_data1, hist = False, kde = True,
             kde_kws = {'shade': True, 'linewidth': 2},
             label = "Normal", color = "green", ax = ax[1])
# adding legends to the subplots
plt.legend(loc = "upper right")

# rescaling the subplots
fig.set_figheight(5)
fig.set_figwidth(10)
```

```
<ipython-input-73-4db82e9e50f1>:5: UserWarning:
```

```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density
plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(project.Actual_Shipment_Time, hist = False, kde = True,
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2511:
FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
```

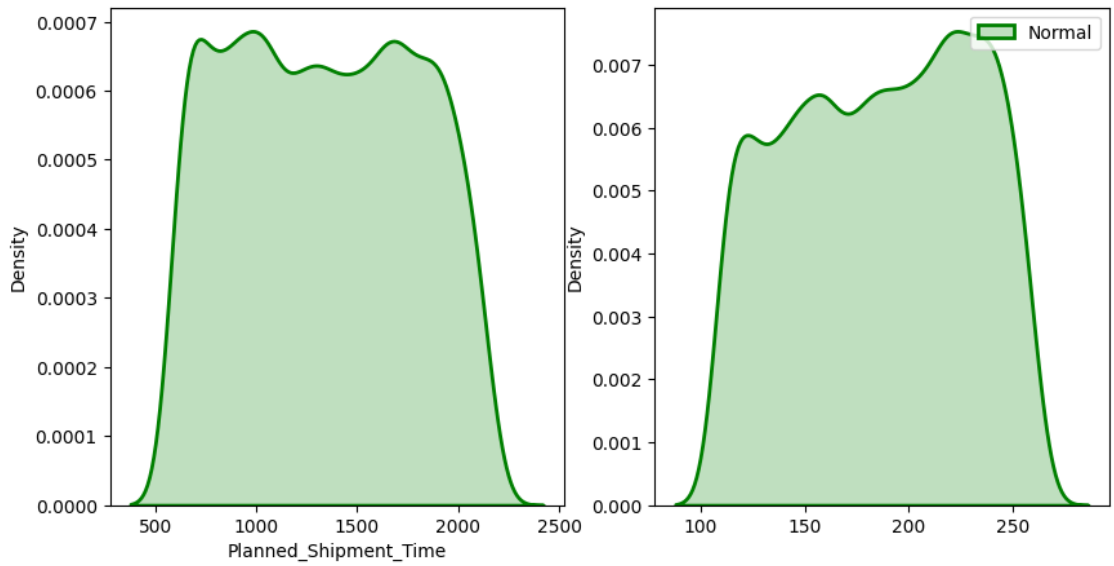
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-73-4db82e9e50f1> in <cell line: 9>()
      7                 label = "Non-Normal", color = "green", ax = ax[0])
      8
----> 9 sns.distplot(fitted_data1, hist = False, kde = True,
     10                 kde_kws = {'shade': True, 'linewidth': 2},
     11                 label = "Normal", color = "green", ax = ax[1])

NameError: name 'fitted_data1' is not defined
```



[75]:
```
# creating axes to draw plots
fig, ax = plt.subplots(1 ,2)
# Plotting the original data (non-normal) and fitted data (normal)
sns.distplot(project.Planned_Shipment_Time, hist = False, kde = True,
             kde_kws = {'shade': True, 'linewidth': 2},
             label = "Non-Normal", color = "green", ax = ax[0])

sns.distplot(fitted_data2, hist = False, kde = True,
             kde_kws = {'shade': True, 'linewidth': 2},
             label = "Normal", color = "green", ax = ax[1])
# adding legends to the subplots
plt.legend(loc = "upper right")

# rescaling the subplots
fig.set_figheight(5)
fig.set_figwidth(10)
```

```
<ipython-input-75-21443b738d73>:4: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
```

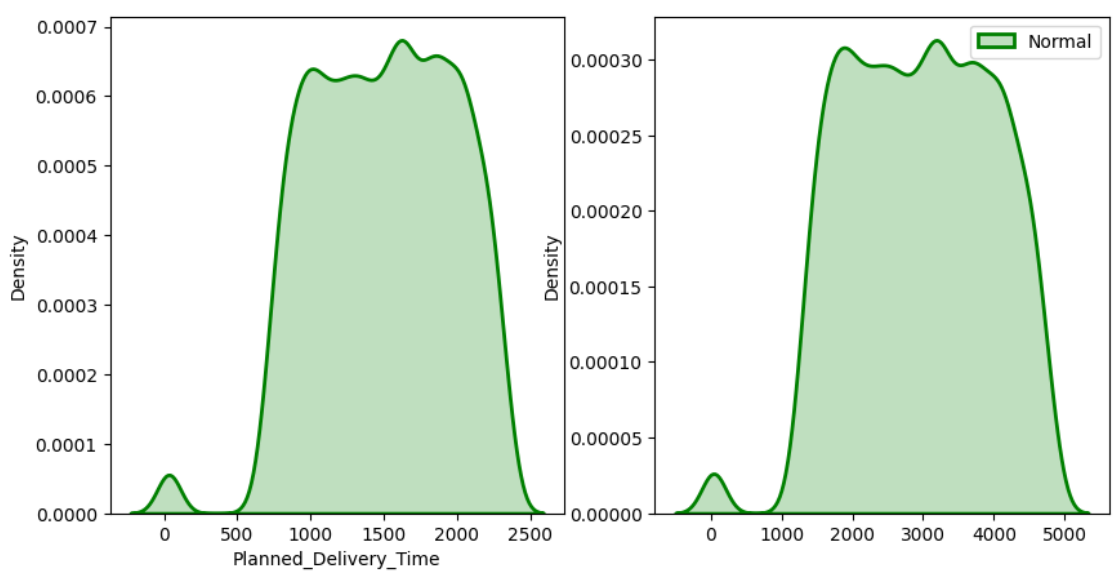
```
similar flexibility) or `kdeplot` (an axes-level function for kernel density
plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(project.Planned_Shipment_Time, hist = False, kde = True,
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2511:
FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-75-21443b738d73>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density
plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(fitted_data2, hist = False, kde = True,
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2511:
FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
```



[76]:
```
# creating axes to draw plots
fig, ax = plt.subplots(1 ,2)

# Plotting the original data (non-normal) and fitted data (normal)
sns.distplot(project.Planned_Delivery_Time, hist = False, kde = True,
             kde_kws = {'shade': True, 'linewidth': 2},
             label = "Non-Normal", color = "green", ax = ax[0])

sns.distplot(fitted_data3, hist = False, kde = True,
             kde_kws = {'shade': True, 'linewidth': 2},
             label = "Normal", color = "green", ax = ax[1])

# adding legends to the subplots
plt.legend(loc = "upper right")

# rescaling the subplots
fig.set_figheight(5)
fig.set_figwidth(10)
```

```
<ipython-input-76-d623d876a3c6>:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density
plots).

For a guide to updating your code to use the new functions, please see
```

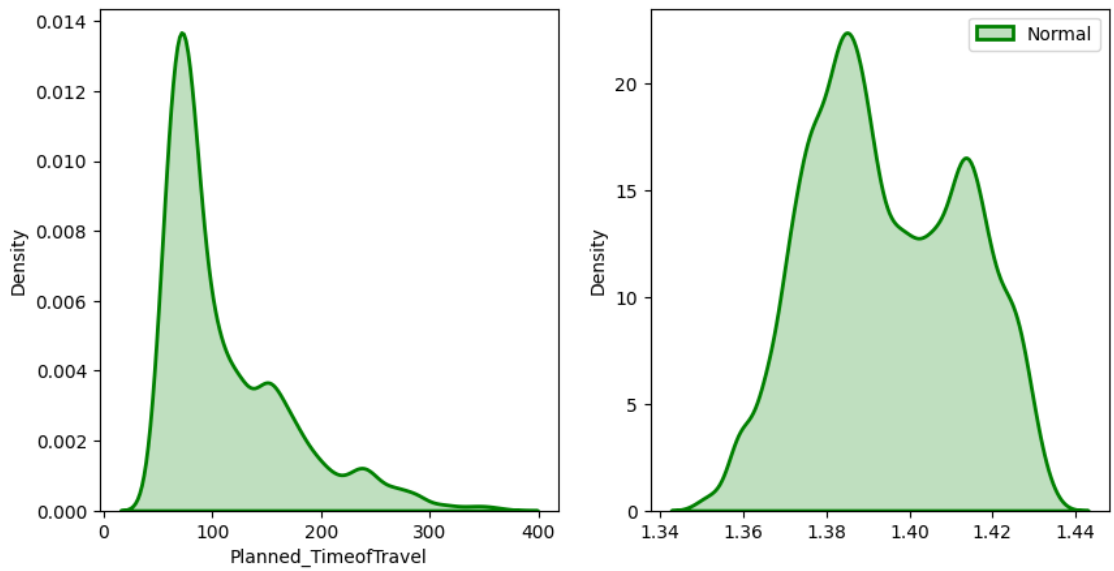
```
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(project.Planned_Delivery_Time, hist = False, kde = True,
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2511:
FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-76-d623d876a3c6>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density
plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(fitted_data3, hist = False, kde = True,
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2511:
FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
```



[77]:
```
# creating axes to draw plots
fig, ax = plt.subplots(1 ,2)

# Plotting the original data (non-normal) and fitted data (normal)
sns.distplot(project.Planned_TimeofTravel, hist = False, kde = True,
             kde_kws = {'shade': True, 'linewidth': 2},
             label = "Non-Normal", color = "green", ax = ax[0])

sns.distplot(fitted_data4, hist = False, kde = True,
             kde_kws = {'shade': True, 'linewidth': 2},
             label = "Normal", color = "green", ax = ax[1])

# adding legends to the subplots
plt.legend(loc = "upper right")

# rescaling the subplots
fig.set_figheight(5)
fig.set_figwidth(10)
```

```
<ipython-input-77-12dfa2effae1>:5: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density
plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(project.Planned_TimeofTravel, hist = False, kde = True,
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2511:
FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
<ipython-input-77-12dfa2effae1>:9: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density
plots).
```

```
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(fitted_data4, hist = False, kde = True,
/usr/local/lib/python3.10/dist-packages/seaborn/distributions.py:2511:
FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  kdeplot(**{axis: a}, ax=ax, color=kde_color, **kde_kws)
```



29 Transformed data

[79]:
```
print(f"Lambda value used for Transformation: {fitted_lambda1}")
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-79-e91fe5b15a3f> in <cell line: 1>()
----> 1 print(f"Lambda value used for Transformation: {fitted_lambda1}")

NameError: name 'fitted_lambda1' is not defined
```

[80]:
```
print(f"Lambda value used for Transformation: {fitted_lambda2}")
```

```
Lambda value used for Transformation: 0.6744165965976693
```

[81]:
```
print(f"Lambda value used for Transformation: {fitted_lambda3}")
```

```
Lambda value used for Transformation: 1.1053861600158499
```

[82]:
```
print(f"Lambda value used for Transformation: {fitted_lambda4}")
```

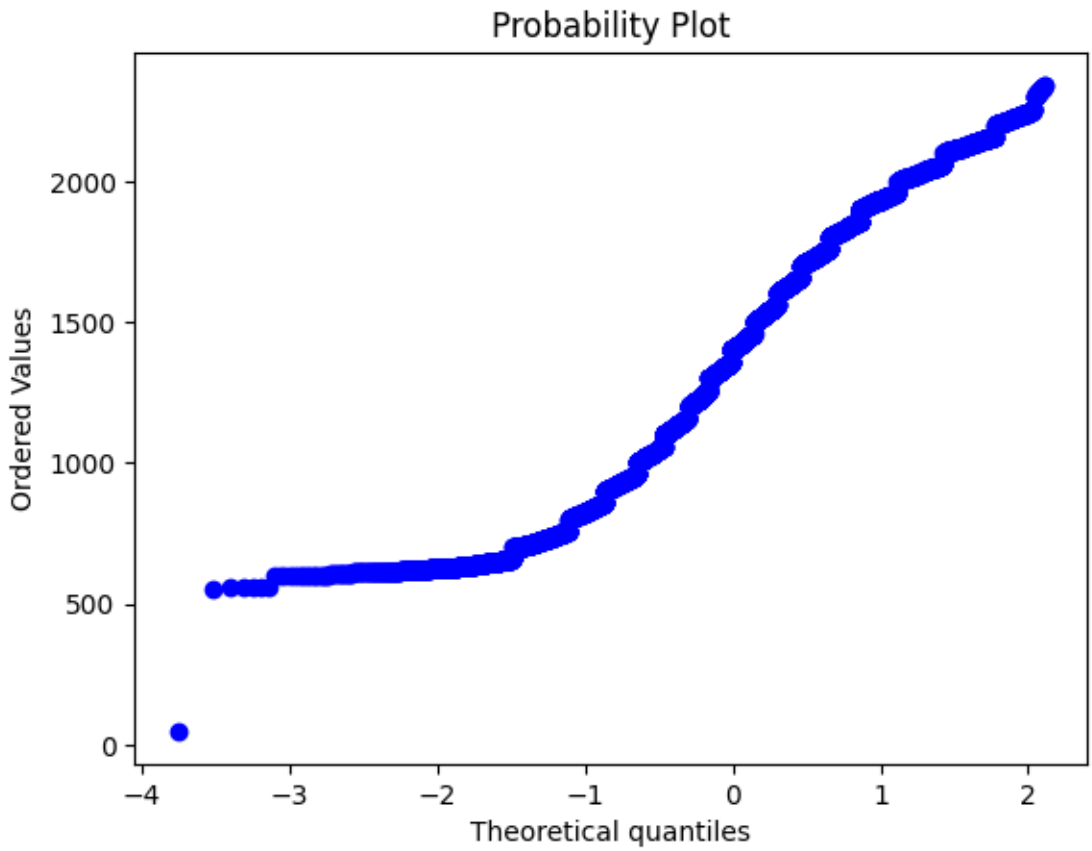
```
Lambda value used for Transformation: -0.685297253290949
```

30 Yeo-Johnson Transform

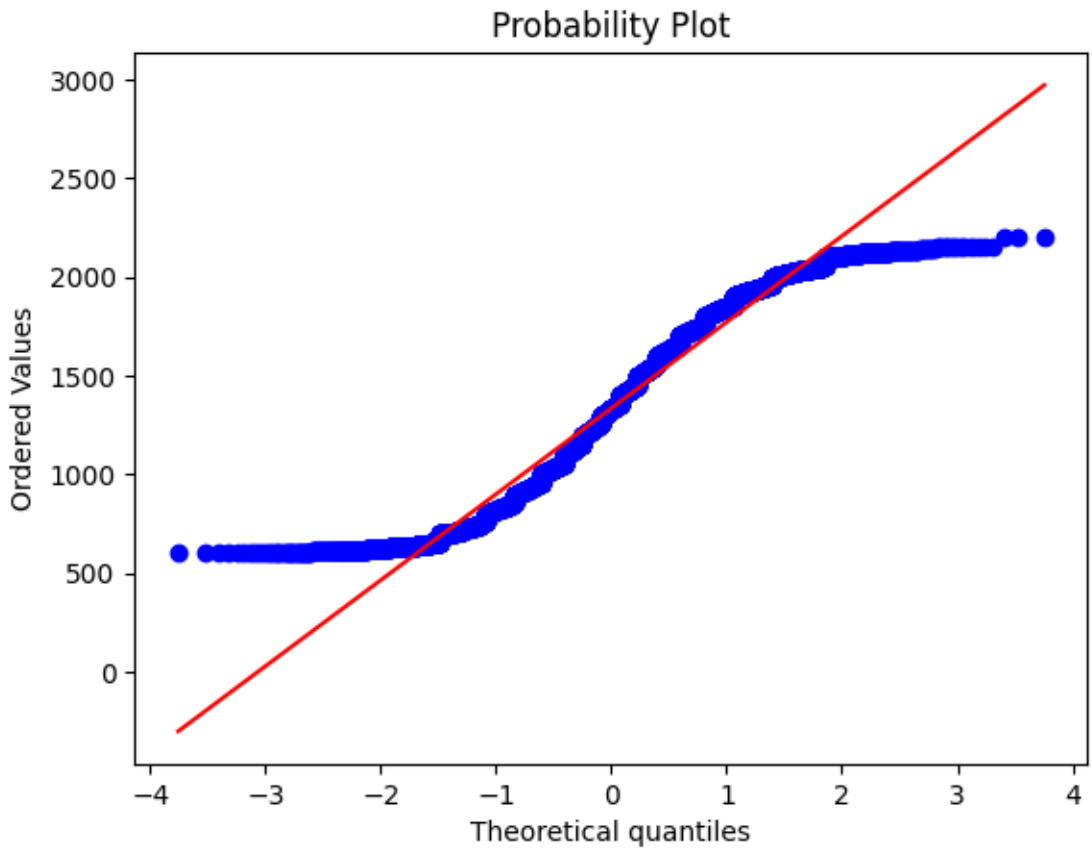
[83]:
```
# import modules
import pandas as pd
from scipy import stats

# Plotting modules
import seaborn as sns
import matplotlib.pyplot as plt
import pylab
```

[84]:
```
# Read data into Python
project = pd.read_csv(r"/content/Datasets.csv")
```

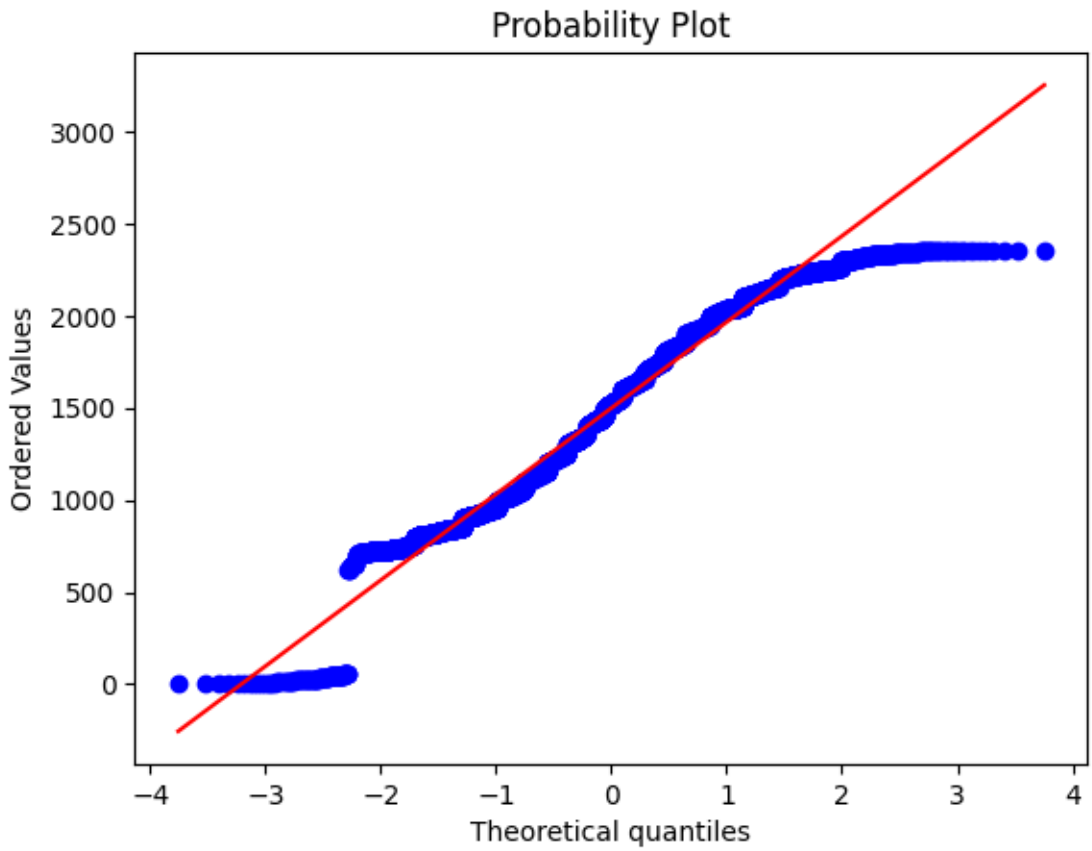
[85]:
```
# Original data

prob1 = stats.probplot(project.Actual_Shipment_Time, dist = stats.norm, plot =␣
↪pylab)
```



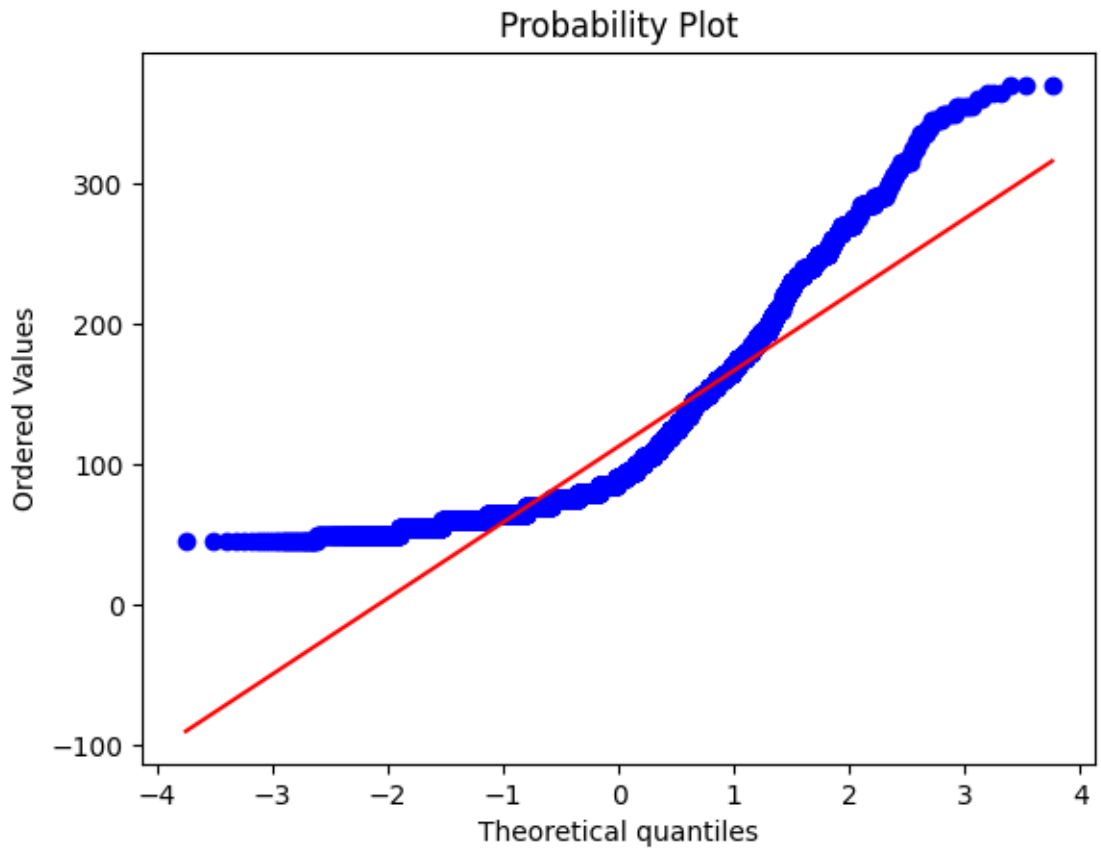
[86]:
```
prob2 = stats.probplot(project.Planned_Shipment_Time, dist = stats.norm, plot =␣
↪pylab)
```



[87]:
```
prob3 = stats.probplot(project.Planned_Delivery_Time, dist = stats.norm, plot =␣
↪pylab)
```



[88]:
```
prob4= stats.probplot(project.Planned_TimeofTravel, dist = stats.norm, plot =␣
↪pylab)
```



[89]:
```
pip install feature_engine
```

```
Collecting feature_engine
  Downloading feature_engine-1.6.2-py2.py3-none-any.whl (328 kB)
                              328.9/328.9

kB 6.9 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.18.2 in
/usr/local/lib/python3.10/dist-packages (from feature_engine) (1.23.5)
Requirement already satisfied: pandas>=1.0.3 in /usr/local/lib/python3.10/dist-
packages (from feature_engine) (1.5.3)
Requirement already satisfied: scikit-learn>=1.0.0 in
/usr/local/lib/python3.10/dist-packages (from feature_engine) (1.2.2)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-
packages (from feature_engine) (1.11.3)
Requirement already satisfied: statsmodels>=0.11.1 in
/usr/local/lib/python3.10/dist-packages (from feature_engine) (0.14.0)
Requirement already satisfied: python-dateutil>=2.8.1 in
/usr/local/lib/python3.10/dist-packages (from pandas>=1.0.3->feature_engine)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
```

```
packages (from pandas>=1.0.3->feature_engine) (2023.3.post1)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn>=1.0.0->feature_engine) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-
learn>=1.0.0->feature_engine) (3.2.0)
Requirement already satisfied: patsy>=0.5.2 in /usr/local/lib/python3.10/dist-
packages (from statsmodels>=0.11.1->feature_engine) (0.5.3)
Requirement already satisfied: packaging>=21.3 in
/usr/local/lib/python3.10/dist-packages (from
statsmodels>=0.11.1->feature_engine) (23.2)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages
(from patsy>=0.5.2->statsmodels>=0.11.1->feature_engine) (1.16.0)
Installing collected packages: feature_engine
Successfully installed feature_engine-1.6.2
```

[90]:
```python
from feature_engine import transformation
```

[91]:
```python
# Set up the variable transformer

ts1 = transformation.YeoJohnsonTransformer(variables = 'Actual_Shipment_Time')

ts2 = transformation.YeoJohnsonTransformer(variables = 'Planned_Shipment_Time')

ts3 = transformation.YeoJohnsonTransformer(variables = 'Planned_Delivery_Time')

ts4 = transformation.YeoJohnsonTransformer(variables = 'Planned_TimeofTravel')
```

[94]:
```python
rx1 = ts1.fit_transform(project)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-94-158263fb888e> in <cell line: 1>()
----> 1 rx1 = ts1.fit_transform(project)

/usr/local/lib/python3.10/dist-packages/sklearn/utils/_set_output.py in␣
↪wrapped(self, X, *args, **kwargs)
    138     @wraps(f)
    139     def wrapped(self, X, *args, **kwargs):
--> 140         data_to_wrap = f(self, X, *args, **kwargs)
    141         if isinstance(data_to_wrap, tuple):
    142             # only wrap the first output for cross decomposition

/usr/local/lib/python3.10/dist-packages/sklearn/base.py in fit_transform(self,␣
↪X, y, **fit_params)
    876         if y is None:
    877             # fit method of arity 1 (unsupervised transformation)
```

```
--> 878             return self.fit(X, **fit_params).transform(X)
    879         else:
    880             # fit method of arity 2 (supervised transformation)

/usr/local/lib/python3.10/dist-packages/feature_engine/transformation/yeojohnso
 ↪py in fit(self, X, y)
    129
    130         # check input dataframe
--> 131         X = super().fit(X)
    132
    133         self.lambda_dict_ = {}

/usr/local/lib/python3.10/dist-packages/feature_engine/_base_transformers/
 ↪base_numerical.py in fit(self, X)
     62
     63         # check if dataset contains na or inf
---> 64         _check_contains_na(X, self.variables_)
     65         _check_contains_inf(X, self.variables_)
     66

/usr/local/lib/python3.10/dist-packages/feature_engine/dataframe_checks.py in␣
 ↪_check_contains_na(X, variables)
    266
    267     if X[variables].isnull().any().any():
--> 268         raise ValueError(
    269             "Some of the variables in the dataset contain NaN. Check an␣
 ↪"
    270             "remove those before using this transformer."

ValueError: Some of the variables in the dataset contain NaN. Check and remove␣
 ↪those before using this transformer.
```

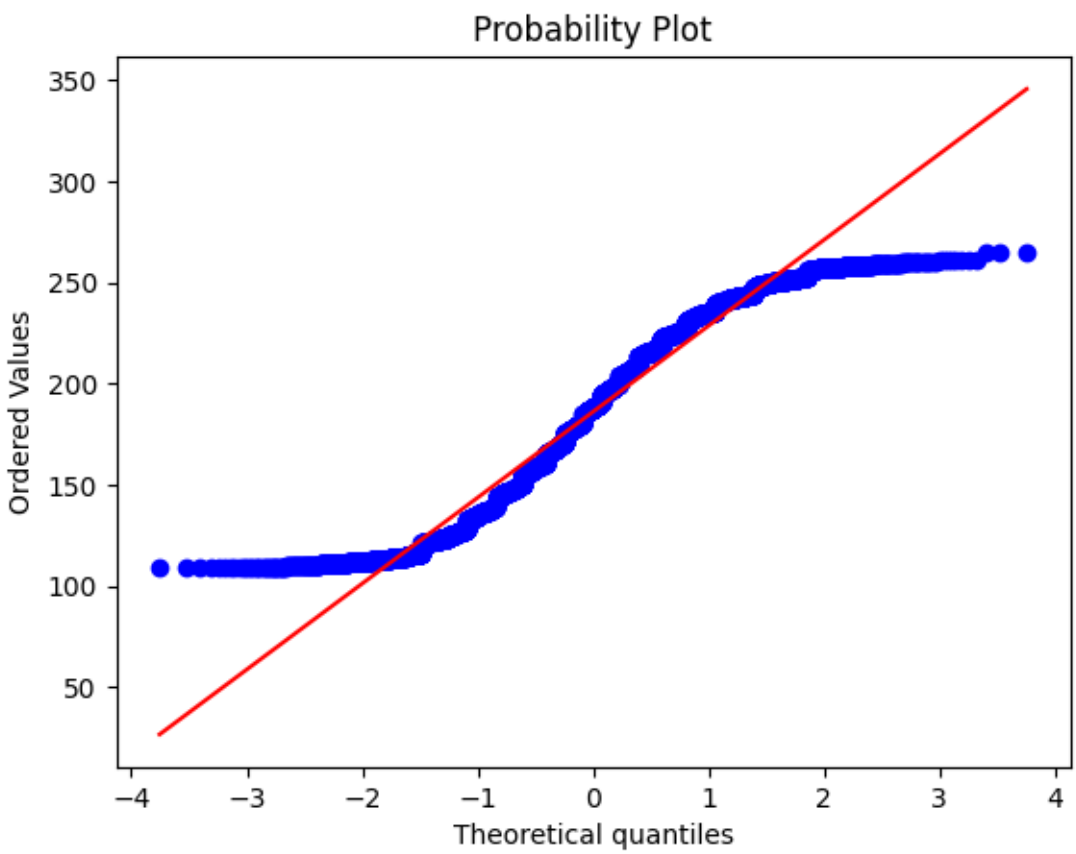
[93]:
```
rx2 = ts2.fit_transform(project)

rx3 = ts3.fit_transform(project)

rx4 = ts4.fit_transform(project)
```

[95]:
```
# Transformed data

prob1 = stats.probplot(rx1.Actual_Shipment_Time, dist = stats.norm, plot =␣
 ↪pylab)

prob2 = stats.probplot(rx2.Planned_Shipment_Time, dist = stats.norm, plot =␣
 ↪pylab)
```

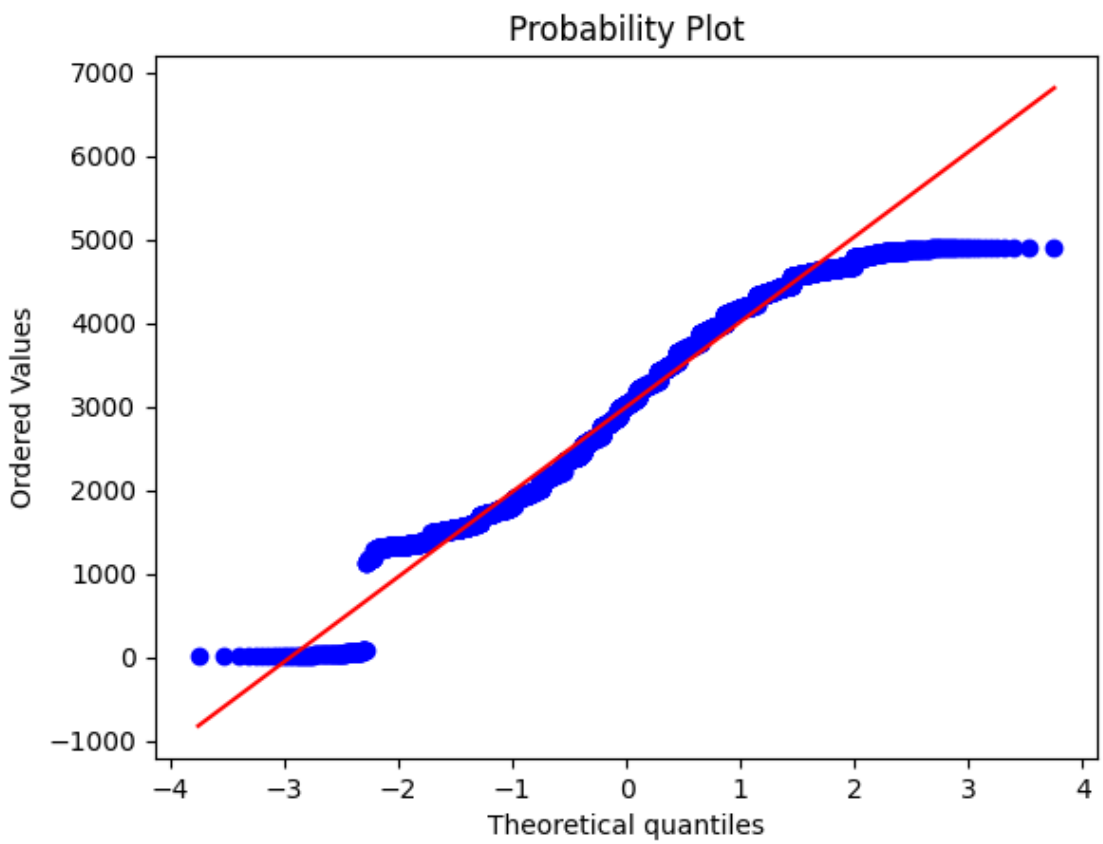
```
prob3 = stats.probplot(rx3.Planned_Delivery_Time, dist = stats.norm, plot =␣
↪pylab)

prob4 = stats.probplot(rx4.Planned_TimeofTravel, dist = stats.norm, plot =␣
↪pylab)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-95-9cd511449d88> in <cell line: 3>()
      1 # Transformed data
      2
----> 3 prob1 = stats.probplot(rx1.Actual_Shipment_Time, dist = stats.norm, plot␣
↪= pylab)
      4
      5 prob2 = stats.probplot(rx2.Planned_Shipment_Time, dist = stats.norm,␣
↪plot = pylab)

NameError: name 'rx1' is not defined
```

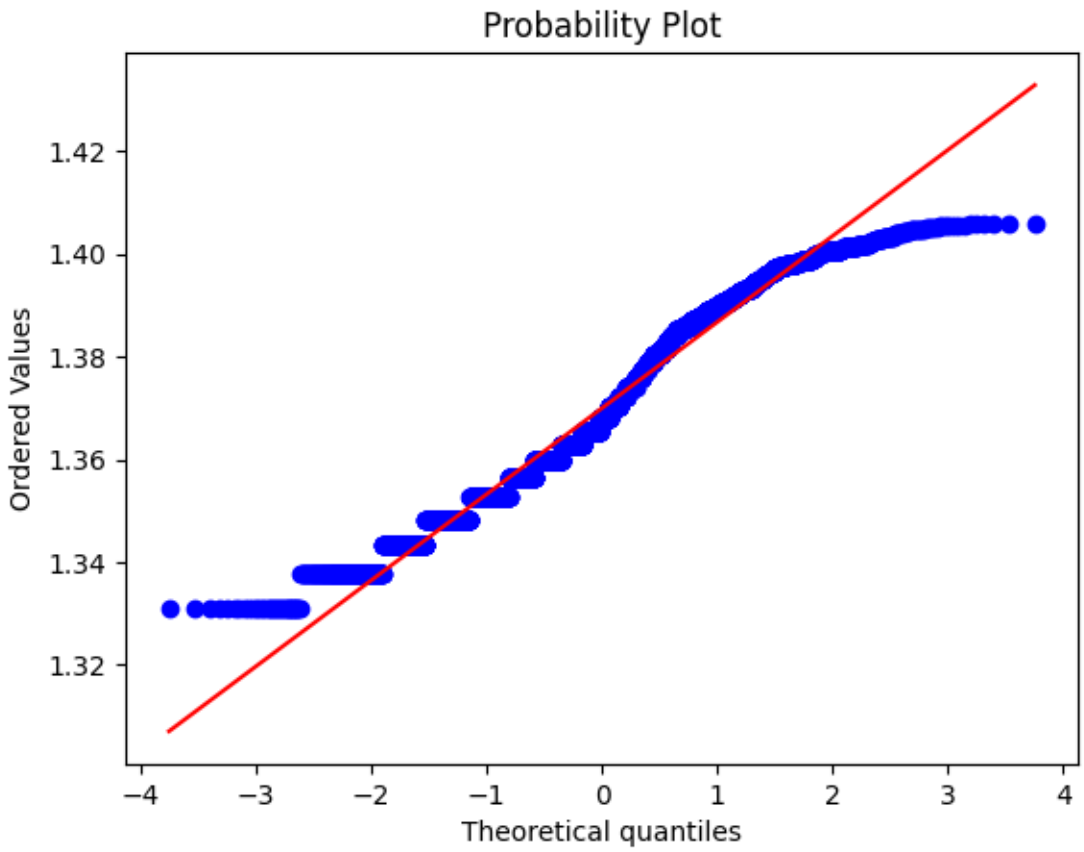
[96]:
```
prob2 = stats.probplot(rx2.Planned_Shipment_Time, dist = stats.norm, plot =␣
↪pylab)
```



[97]:
```
prob3 = stats.probplot(rx3.Planned_Delivery_Time, dist = stats.norm, plot =␣
↪pylab)
```



[98]:
```
prob4 = stats.probplot(rx4.Planned_TimeofTravel, dist = stats.norm, plot =␣
↪pylab)
```



Standardization and Normalization

[99]:
```
import pandas as pd
import numpy as np
```

[100]:
```
project = pd.read_csv(r"/content/Datasets.csv")
```

[101]:
```
ps = project.describe()
```

[102]:
```
### Standardization
from sklearn.preprocessing import StandardScaler
```

[104]:
```
# Initialise the Scaler
scaler = StandardScaler()
```

[105]:
```
# To scale data
dn = scaler.fit_transform(project)
# Convert the array back to a dataframe
dataset = pd.DataFrame(dn)
res = dataset.describe()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-105-d7d61a45e15e> in <cell line: 2>()
      1 # To scale data
----> 2 dn = scaler.fit_transform(project)
      3 # Convert the array back to a dataframe
      4 dataset = pd.DataFrame(dn)
      5 res = dataset.describe()

/usr/local/lib/python3.10/dist-packages/sklearn/utils/_set_output.py in␣
↪wrapped(self, X, *args, **kwargs)
    138     @wraps(f)
    139     def wrapped(self, X, *args, **kwargs):
--> 140         data_to_wrap = f(self, X, *args, **kwargs)
    141         if isinstance(data_to_wrap, tuple):
    142             # only wrap the first output for cross decomposition

/usr/local/lib/python3.10/dist-packages/sklearn/base.py in fit_transform(self,␣
↪X, y, **fit_params)
    876         if y is None:
    877             # fit method of arity 1 (unsupervised transformation)
--> 878             return self.fit(X, **fit_params).transform(X)
    879         else:
    880             # fit method of arity 2 (supervised transformation)

/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_data.py in␣
↪fit(self, X, y, sample_weight)
    822         # Reset internal state before fitting
    823         self._reset()
--> 824         return self.partial_fit(X, y, sample_weight)
    825
    826     def partial_fit(self, X, y=None, sample_weight=None):

/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_data.py in␣
↪partial_fit(self, X, y, sample_weight)
    859
    860         first_call = not hasattr(self, "n_samples_seen_")
--> 861         X = self._validate_data(
    862             X,
    863             accept_sparse=("csr", "csc"),

/usr/local/lib/python3.10/dist-packages/sklearn/base.py in _validate_data(self,␣
↪X, y, reset, validate_separately, **check_params)
    563             raise ValueError("Validation should be done on X, y or both
↪")
    564         elif not no_val_X and no_val_y:
```

```
--> 565             X = check_array(X, input_name="X", **check_params)
    566             out = X
    567         elif no_val_X and not no_val_y:

/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py in␣
↪check_array(array, accept_sparse, accept_large_sparse, dtype, order, copy,␣
↪force_all_finite, ensure_2d, allow_nd, ensure_min_samples,␣
↪ensure_min_features, estimator, input_name)
    877                     array = xp.astype(array, dtype, copy=False)
    878                 else:
--> 879                     array = _asarray_with_order(array, order=order,␣
↪dtype=dtype, xp=xp)
    880             except ComplexWarning as complex_warning:
    881                 raise ValueError(

/usr/local/lib/python3.10/dist-packages/sklearn/utils/_array_api.py in␣
↪_asarray_with_order(array, dtype, order, copy, xp)
    183     if xp.__name__ in {"numpy", "numpy.array_api"}:
    184         # Use NumPy API to support order
--> 185         array = numpy.asarray(array, order=order, dtype=dtype)
    186         return xp.asarray(array, copy=copy)
    187     else:

/usr/local/lib/python3.10/dist-packages/pandas/core/generic.py in␣
↪__array__(self, dtype)
   2068
   2069     def __array__(self, dtype: npt.DTypeLike | None = None) -> np.
↪ndarray:
-> 2070         return np.asarray(self._values, dtype=dtype)
   2071
   2072     def __array_wrap__(

ValueError: could not convert string to float: 'WN'
```

[106]:
```
# Normalization
''' Alternatively we can use the below function'''
from sklearn.preprocessing import MinMaxScaler
minmaxscale = MinMaxScaler()

dn_n = minmaxscale.fit_transform(dn)
dataset1 = pd.DataFrame(dn_n)

res1 = dataset1.describe()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-106-df2a998530e8> in <cell line: 6>()
```

```
      4 minmaxscale = MinMaxScaler()
      5
----> 6 dn_n = minmaxscale.fit_transform(dn)
      7 dataset1 = pd.DataFrame(dn_n)
      8

NameError: name 'dn' is not defined
```

[108]:
```
### Normalization
## load dataset
project = pd.read_csv(r"/content/Datasets.csv")

project.columns
project.drop([ 'Actual_Shipment_Time','Planned_Shipment_Time',␣
 ↪'Planned_Delivery_Time', 'Carrier_Name',
       'Carrier_Num', 'Planned_TimeofTravel', ], axis = 1, inplace = True)
```

[109]:
```
a2 = project.describe()
```

[110]:
```
# Get dummies
ethnic1 = pd.get_dummies(project, drop_first = True)

a3 = ethnic1.describe()
```

[111]:
```
### Normalization function - Custom Function
# Range converts to: 0 to 1
def norm_func(i):
    x = (i-i.min())/(i.max()-i.min())
    return(x)

df_norm = norm_func(ethnic1)
b = df_norm.describe()
```

[112]:
```
''' Alternatively we can use the below function'''
from sklearn.preprocessing import MinMaxScaler
minmaxscale = MinMaxScaler()

ethnic1_minmax = minmaxscale.fit_transform(ethnic1)
df_ethnic1 = pd.DataFrame(ethnic1_minmax)
minmax_res = df_ethnic1.describe()
```

[113]:
```
'''Robust Scaling
Scale features using statistics that are robust to outliers'''

from sklearn.preprocessing import RobustScaler
```

```
robust_model = RobustScaler()

df_robust = robust_model.fit_transform(ethnic1)

dataset_robust = pd.DataFrame(df_robust)
res_robust = dataset_robust.describe()
```

[114]:
```
import pandas as pd
```

31 clean data

[115]:
```
project = pd.read_csv(r"/content/Datasets.csv")
```

[116]:
```
print(f"Cleaned data saved to: {project}")
```

```
Cleaned data saved to:       Year  Month  DayofMonth  DayOfWeek
Actual_Shipment_Time  \
0     2008      1           3          4                2003.0
1     2008      1           3          4                 754.0
2     2008      1           3          4                 628.0
3     2008      1           3          4                 926.0
4     2008      1           3          4                1829.0
…     …     …           …          …                     …
7994  2008      1           5          6                1534.0
7995  2008      1           5          6                1200.0
7996  2008      1           5          6                 902.0
7997  2008      1           5          6                1722.0
7998  2008      1           5          6                 721.0

      Planned_Shipment_Time  Planned_Delivery_Time Carrier_Name  Carrier_Num  \
0                      1955                   2225           WN          335
1                       735                   1000           WN         3231
2                       620                    750           WN          448
3                       930                   1100           WN         1746
4                      1755                   1925           WN         3920
…                       …                      …           …           …
7994                   1520                   1620           WN         1516
7995                   1200                   1255           WN         2621
7996                    900                   1000           WN         3569
7997                   1715                   1930           WN          383
7998                    715                    930           WN         1945

      Planned_TimeofTravel  Shipment_Delay Source Destination  Distance  \
0                      150             8.0    IAD         TPA       810
1                      145            19.0    IAD         TPA       810
2                       90             8.0    IND         BWI       515
```

```
3                       90            -4.0    IND         BWI       515
4                       90            34.0    IND         BWI       515
…                     …             …    …         …       …
7994                    60            14.0    RDU         BWI       255
7995                    55             0.0    RDU         BWI       255
7996                    60             2.0    RDU         BWI       255
7997                   315             7.0    RDU         LAS      2027
7998                   315             6.0    RDU         LAS      2027

      Delivery_Status
0                 0.0
1                 1.0
2                 0.0
3                 0.0
4                 1.0
…               …
7994              0.0
7995              0.0
7996              0.0
7997              0.0
7998              0.0

[7999 rows x 15 columns]
```

[ ]:

```
```

eda-ds-project-1

January 2, 2024

1 **Explanation of the dataset:**

2 **No. of Columns: 15**

3 **Dependent variable: Delivery_Status**

4 **Task: Classification**

5 **NA's: Yes**

6 **Explanation of the Columns:**

7 **Year: 2008**

8 **Month:1 month time is needed**

9 **DayofMonth: 3rd or 4th day of month**

10 **DayofWeek: 4th or 5th day of Week**

11 **Actual_Shipment_Time: The Actual time when the package was sent for shipment. (ex: 1955 means 19 hours and 55 minutes i.e 7:55 PM)**

12 **Planned_Shipment_Time: The time when the package should have been sent for shipment. (ex: 1955 means 19 hours and 55 minutes i.e 7:55 PM)**

13 **Planned_Delivery_Time: The time when the package should be delivered. (ex: 1955 means 19 hours and 55 minutes i.e 7:55 PM)**

14 **Carrier_Name: The name of the Carrier which carried the package.**

15 **Carrier_Num: The number of the Carrier which carried the package.**

16 **Planned_TimeofTravel: The estimated time to reach from Source to Destination. (in minutes)**

17 **Shipment_Delay: The time by which the package was shipped late. (in minutes. Negative value indicates that the package was shipped early. Ex: 4 indicates that the package was shipped 4 minutes late, whereas, -4 indicates that the package was shipped 4 minutes early)**

[2]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages
(1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas) (2023.3.post1)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-
packages (from pandas) (1.23.5)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
```

[4]:
```
import pandas as pd

dir(pd)

project = pd.read_csv(r"/content/Datasets.csv")
```

23 Measures of Central Tendency / First moment business decision

MEAN

[7]:
```
project.Actual_Shipment_Time.mean()
```

[7]: 1370.203435114504

[8]:
```
project.Planned_Shipment_Time.mean()
```

[8]: 1335.3175396924617

[9]:
```
project.Carrier_Num.mean()
```

[9]: 1422.2832854106764

[10]:
```
project.Planned_TimeofTravel.mean()
```

[10]: 112.89911238904863

[11]:
```
project.Distance.mean()
```

[11]: 637.847230903863

[12]:
```
project.Shipment_Delay.mean()
```

[12]: 21.389185750636134

MEDIAN

[13]:
```
project.Actual_Shipment_Time.median()
```

[13]:
```
1356.0
```

[14]:
```
project.Planned_Shipment_Time.median()
```

[14]:
```
1330.0
```

[15]:
```
project.Carrier_Num.median()
```

[15]:
```
1023.0
```

[16]:
```
project.Planned_TimeofTravel.median()
```

[16]:
```
90.0
```

[17]:
```
project.Distance.median()
```

[17]:
```
447.0
```

[18]:
```
project.Shipment_Delay.median()
```

[18]:
```
9.0
```

MODE

[19]:
```
project.Actual_Shipment_Time.mode()
```

[19]:
```
0    700.0
Name: Actual_Shipment_Time, dtype: float64
```

[20]:
```
project.Planned_Shipment_Time.mode()
```

[20]:
```
0    630
Name: Planned_Shipment_Time, dtype: int64
```

[61]:
```
project.Carrier_Num.mode()
```

[61]:
```
0     102
1    1414
2    2361
Name: Carrier_Num, dtype: int64
```

[22]:
```
project.Planned_TimeofTravel.mode()
```

[22]: 0 75
Name: Planned_TimeofTravel, dtype: int64

[23]:
```
project.Distance.mode()
```

[23]: 0 337
Name: Distance, dtype: int64

[24]:
```
project.Shipment_Delay.mode()
```

[24]: 0 0.0
Name: Shipment_Delay, dtype: float64

24 Measures of Dispersion / Second moment business decision

[26]:
```
project.Actual_Shipment_Time.var()
```

[26]: 219064.81202535334

[27]:
```
project.Actual_Shipment_Time.std()
```

[27]: 468.0436005601971

[28]:
```
range = max(project.Actual_Shipment_Time) - min(project.Actual_Shipment_Time)
range
```

[28]: 2294.0

[29]:
```
project.Planned_Shipment_Time.var()
```

[29]: 199051.0494435085

[30]:
```
project.Planned_Shipment_Time.std()
```

[30]: 446.1513750326323

[31]:
```
range = max(project.Planned_Shipment_Time) - min(project.Planned_Shipment_Time)
range
```

[31]: 1600

[33]:
```
project.Carrier_Num.var()
```

[33]: 1334677.2670761766

[34]:
```
project.Carrier_Num.std()
```

[34]: 1155.282332192515

[35]:
```
range = max(project.Carrier_Num) - min(project.Carrier_Num)
range
```

[35]: 3948

[37]:
```
project.Planned_TimeofTravel.var()
```

[37]: 3453.4533112900676

[36]:
```
project.Planned_TimeofTravel.std()
```

[36]: 58.766089807728974

[38]:
```
range = max(project.Planned_TimeofTravel) - min(project.Planned_TimeofTravel)
range
```

[38]: 325

[42]:
```
project.Distance.var()
```

[42]: 204261.43802402657

[41]:
```
project.Distance.std()
```

[41]: 451.952915715815

[40]:
```
range = max(project.Distance) - min(project.Distance)
range
```

[40]: 2230

[44]:
```
project.Shipment_Delay.var()
```

[44]: 1060.3784808231076

[45]:
```
project.Shipment_Delay.std()
```

[45]: 32.56345314648168

[46]:
```
range = max(project.Shipment_Delay) - min(project.Shipment_Delay)
range
```

[46]: 325.0

25 Third moment business decision

[47]:
```
project.Actual_Shipment_Time.skew()
```

[47]: 0.03738851063385681

[48]:
```
project.Planned_Shipment_Time.skew()
```

[48]: 0.038648946989911614

[49]:
```
project.Carrier_Num.skew()
```

[49]: 0.6508091440601923

[50]:
```
project.Planned_TimeofTravel.skew()
```

[50]: 1.423396585192246

[51]:
```
project.Distance.skew()
```

[51]: 1.4608790007256947

[53]:
```
project.Shipment_Delay.skew()
```

[53]: 2.740589193560789

26 Fourth moment business decision

[54]:
```
project.Actual_Shipment_Time.kurt()
```

[54]: -1.1777053461904525

[55]:
```
project.Planned_Shipment_Time.kurt()
```

[55]: -1.2031904488780858

[56]:
```
project.Carrier_Num.kurt()
```

[56]: -0.847418945861568

[57]:
```
project.Planned_TimeofTravel.kurt()
```

[57]: 1.7054182837994971

[58]:
```
project.Distance.kurt()
```

[58]: 1.6504619593485272

[60]:
```
project.Shipment_Delay.kurt()
```

[60]: 10.944013976268785

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

phical-representation-ds-project-1

January 2, 2024

1 Data Visualization or Graphical Representation

[1]:
```
import matplotlib.pyplot as plt
```

[2]:
```
import numpy as np
import pandas as pd
```

2 Read data into Python

[5]:
```
project = pd.read_csv(r"/content/Datasets.csv")
```

3 Read data into Python

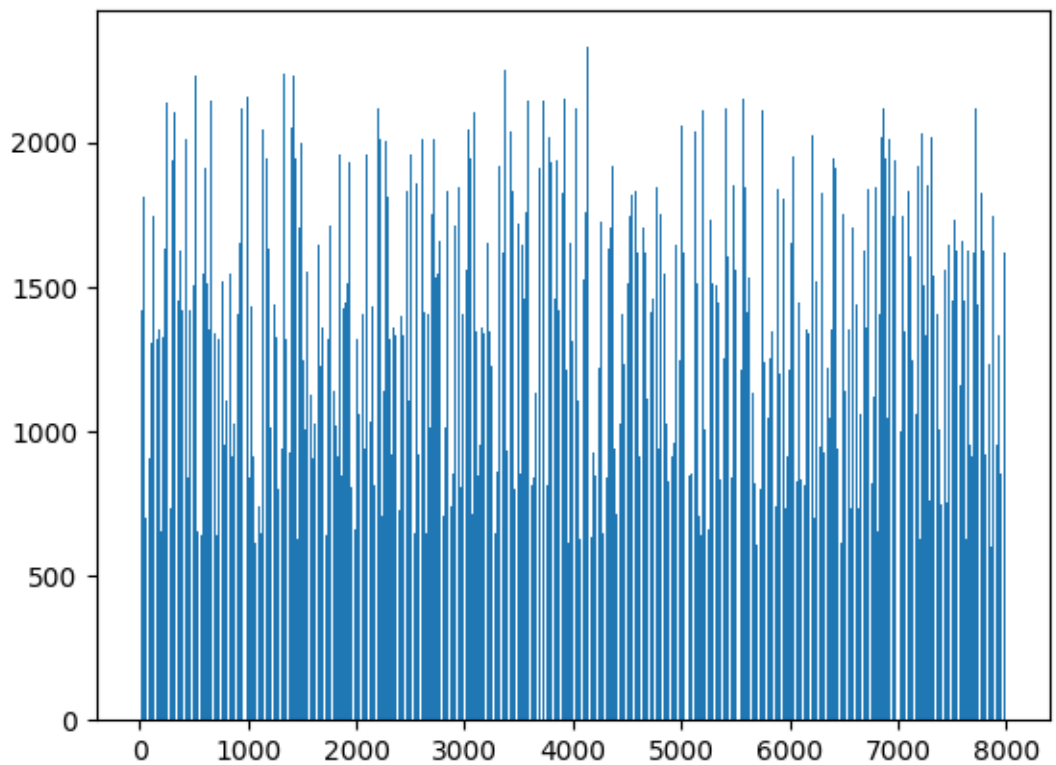
[6]:
```
project.shape
```

[6]:
```
(7999, 15)
```

4 barplot

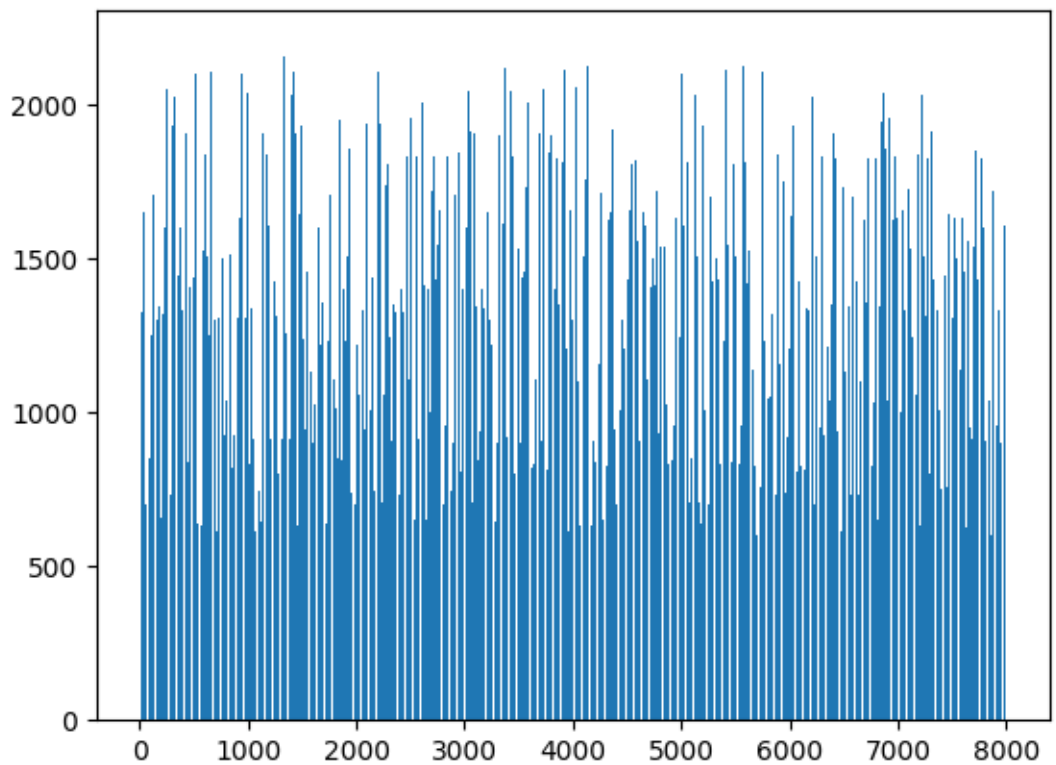
[7]:
```
plt.bar(height = project.Actual_Shipment_Time, x = np.arange(1, 8000, 1))
```

[7]:
```
<BarContainer object of 7999 artists>
```



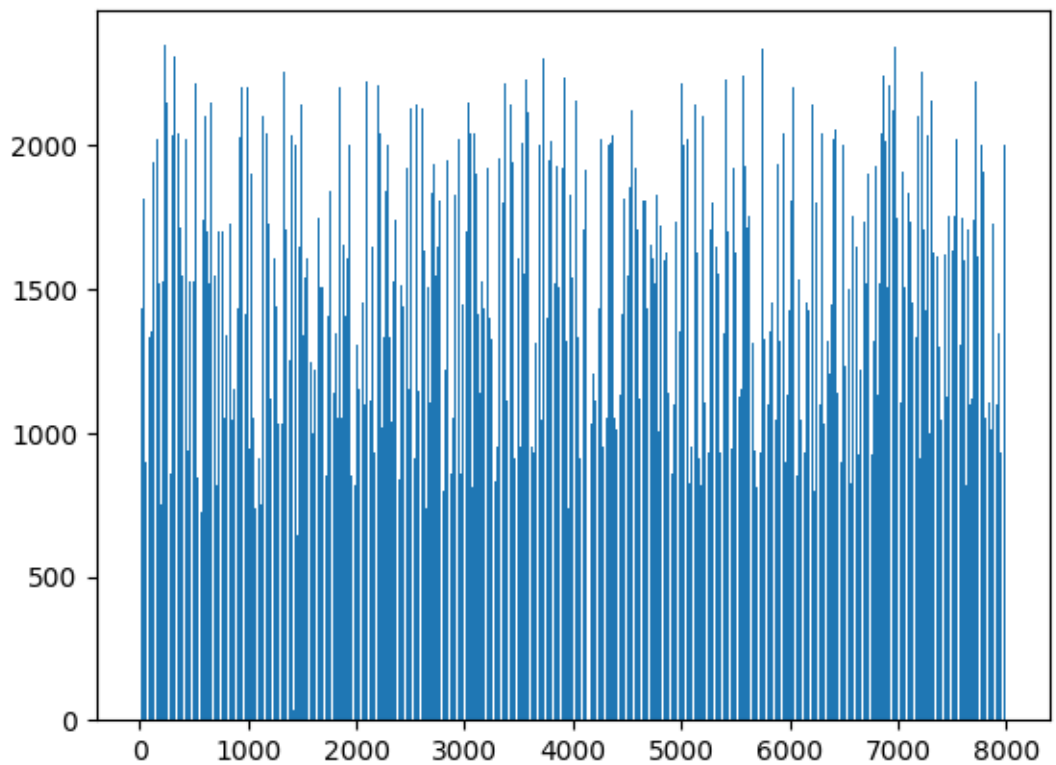
[8]:
```
plt.bar(height = project.Planned_Shipment_Time, x = np.arange(1, 8000, 1))
```

[8]: <BarContainer object of 7999 artists>



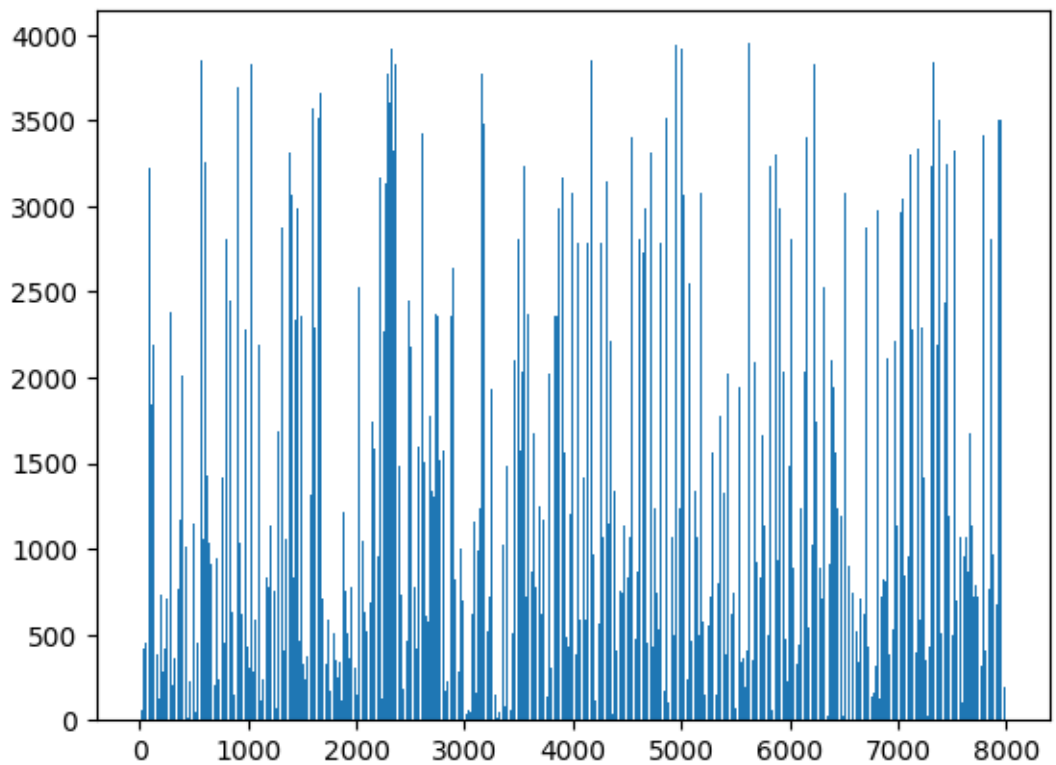
[9]:
```
plt.bar(height = project.Planned_Delivery_Time, x = np.arange(1, 8000, 1))
```

[9]: <BarContainer object of 7999 artists>



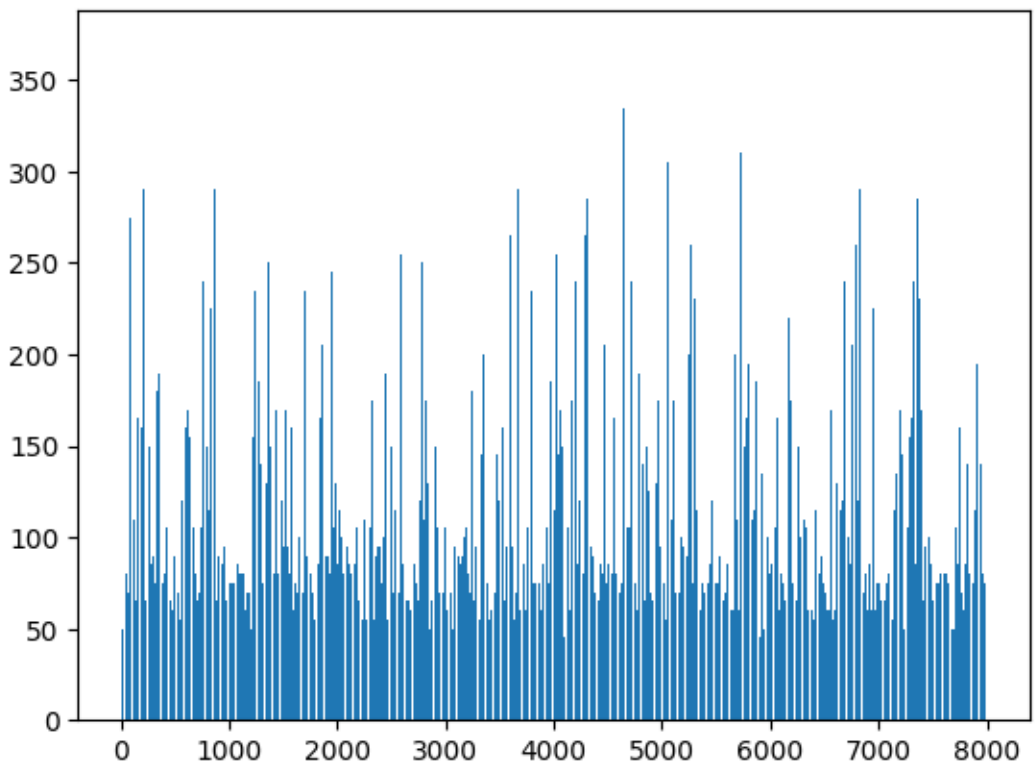
[10]:
```
plt.bar(height = project.Carrier_Num, x = np.arange(1, 8000, 1))
```

[10]: <BarContainer object of 7999 artists>



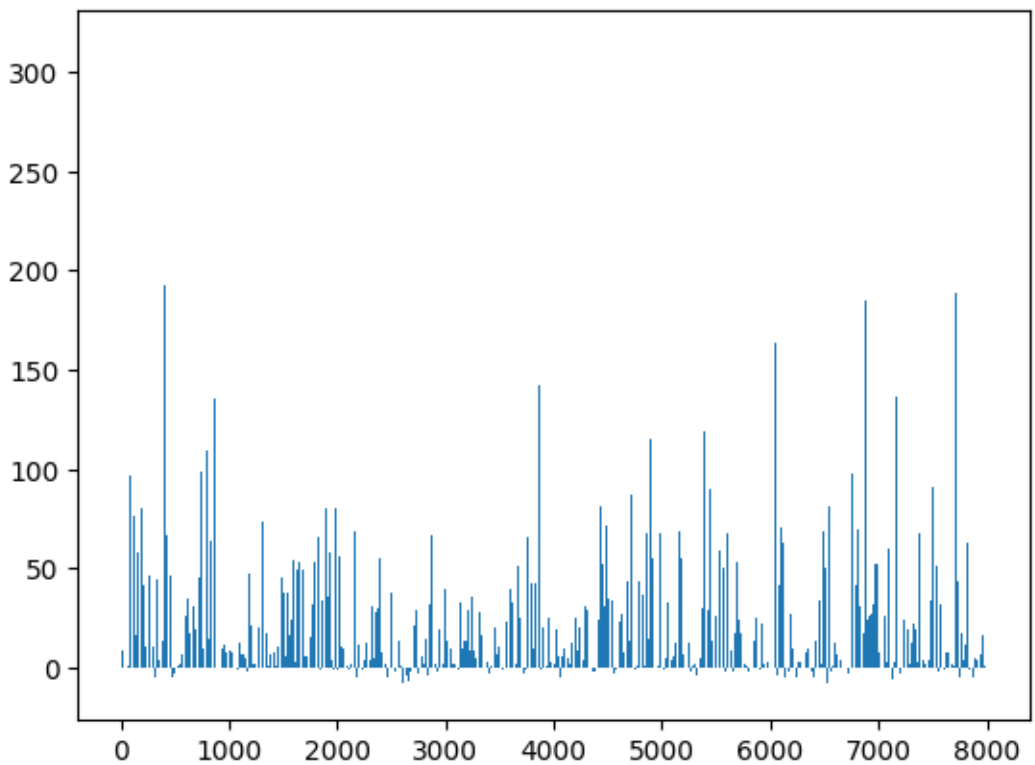
[11]:
```
plt.bar(height = project.Planned_TimeofTravel, x = np.arange(1, 8000, 1))
```

[11]: <BarContainer object of 7999 artists>



[12]:
```
plt.bar(height = project.Shipment_Delay, x = np.arange(1, 8000, 1))
```

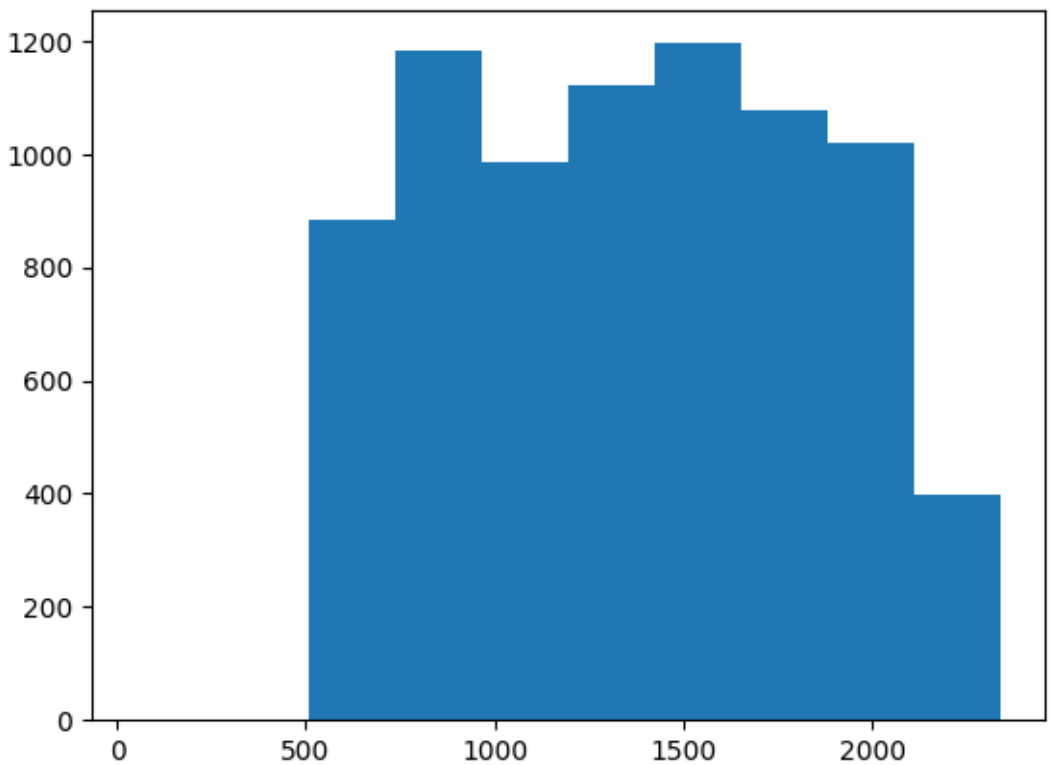
[12]: <BarContainer object of 7999 artists>



5 Histogram

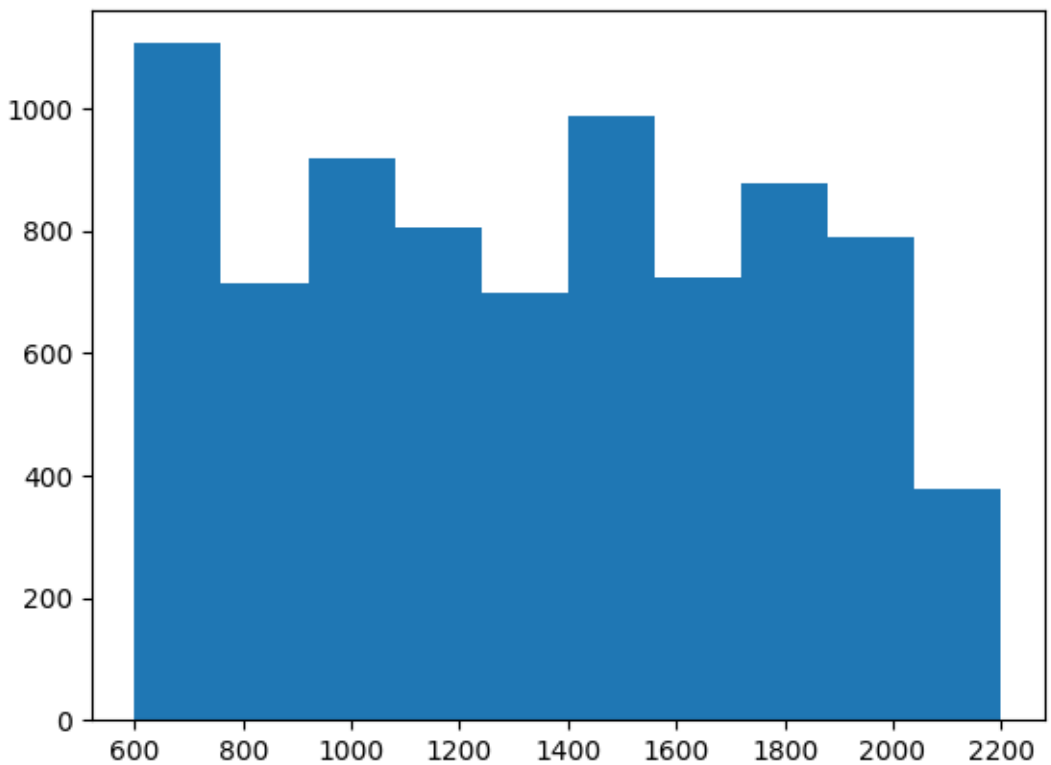
[14]:
```
plt.hist(project.Actual_Shipment_Time)
```

[14]:
```
(array([1.000e+00, 0.000e+00, 8.830e+02, 1.184e+03, 9.840e+02, 1.120e+03,
        1.195e+03, 1.076e+03, 1.020e+03, 3.970e+02]),
 array([  47. ,  276.4,  505.8,  735.2,  964.6, 1194. , 1423.4, 1652.8,
        1882.2, 2111.6, 2341. ]),
 <BarContainer object of 10 artists>)
```



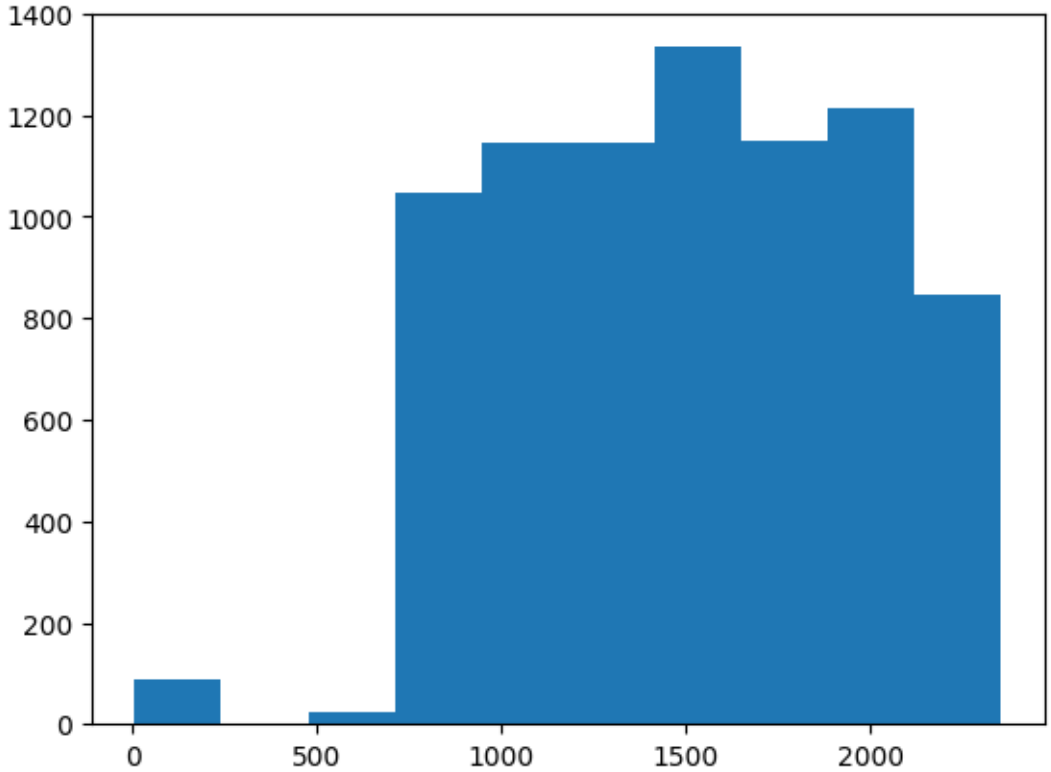
[15]:
```
plt.hist(project.Planned_Shipment_Time)
```

[15]:
```
(array([1106.,  714.,  917.,  805.,  699.,  988.,  725.,  878.,  788.,
         379.]),
 array([ 600.,  760.,  920., 1080., 1240., 1400., 1560., 1720., 1880.,
        2040., 2200.]),
 <BarContainer object of 10 artists>)
```



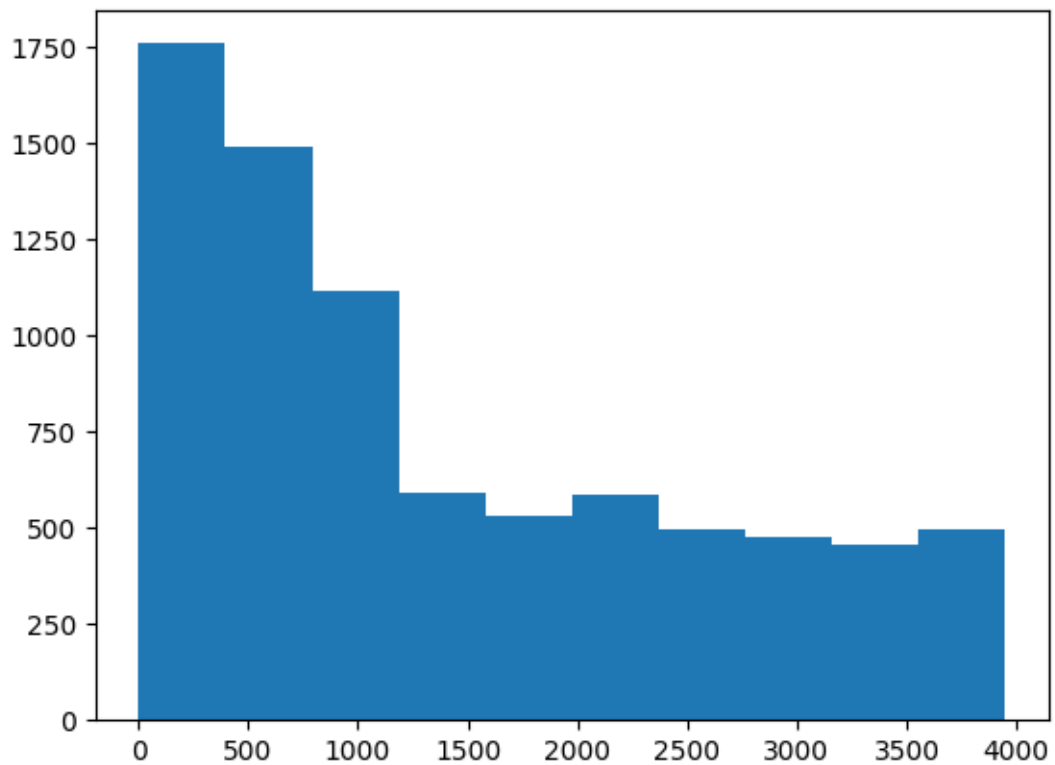
[16]:
```
plt.hist(project.Planned_Delivery_Time)
```

[16]:
```
(array([  89.,    0.,   25., 1047., 1147., 1146., 1334., 1149., 1215.,
         847.]),
 array([   5.,  240.,  475.,  710.,  945., 1180., 1415., 1650., 1885.,
        2120., 2355.]),
 <BarContainer object of 10 artists>)
```



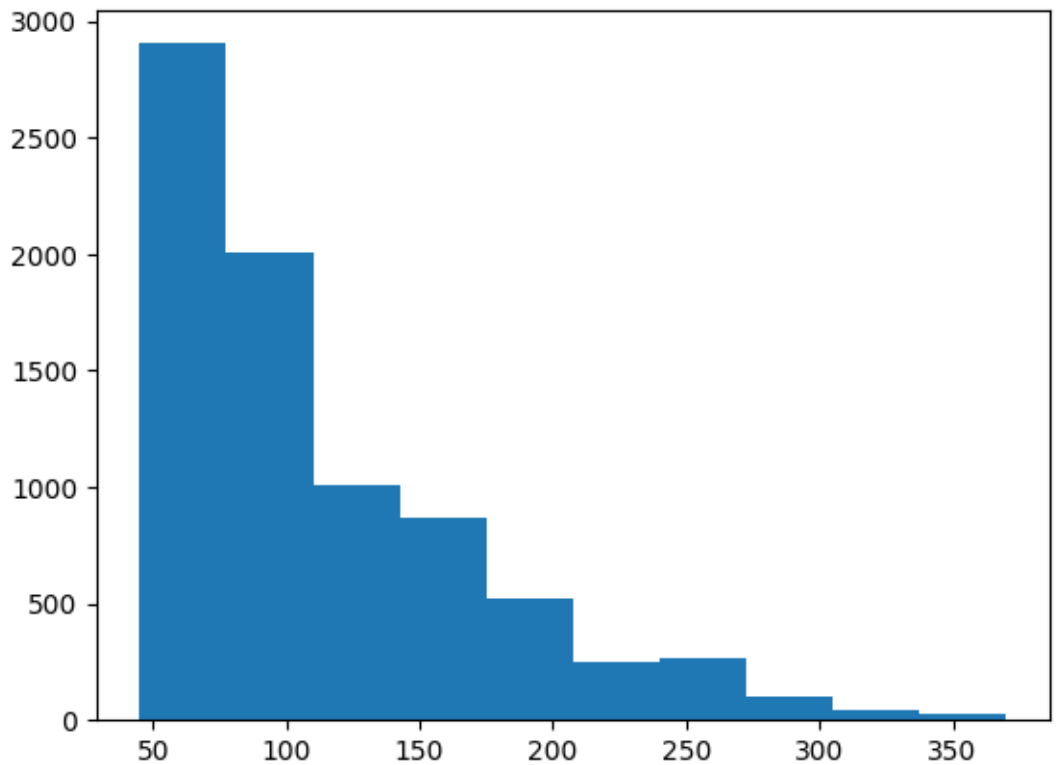
[17]:
```
plt.hist(project.Carrier_Num)
```

[17]:
```
(array([1758., 1490., 1115.,  593.,  531.,  588.,  497.,  476.,  455.,
         496.]),
 array([1.0000e+00, 3.9580e+02, 7.9060e+02, 1.1854e+03, 1.5802e+03,
        1.9750e+03, 2.3698e+03, 2.7646e+03, 3.1594e+03, 3.5542e+03,
        3.9490e+03]),
 <BarContainer object of 10 artists>)
```



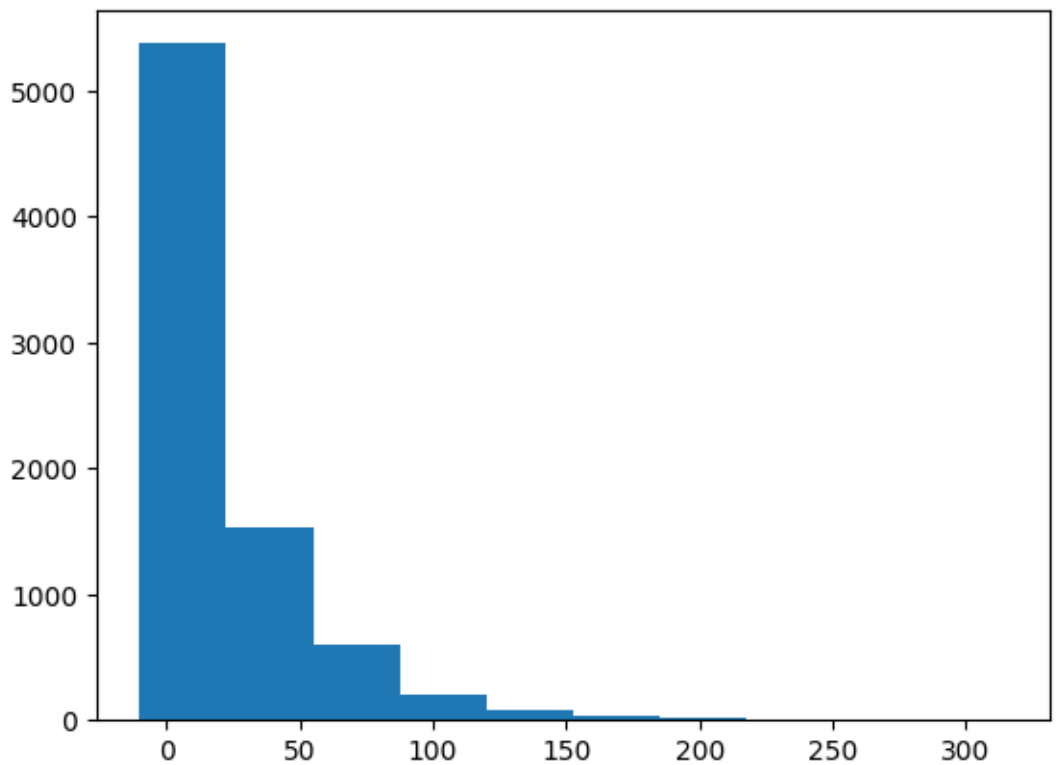
[18]:
```
plt.hist(project.Planned_TimeofTravel)
```

[18]:
```
(array([2904., 2007., 1009.,  869.,  520.,  252.,  269.,   98.,   41.,
          30.]),
 array([ 45. ,  77.5, 110. , 142.5, 175. , 207.5, 240. , 272.5, 305. ,
        337.5, 370. ]),
 <BarContainer object of 10 artists>)
```



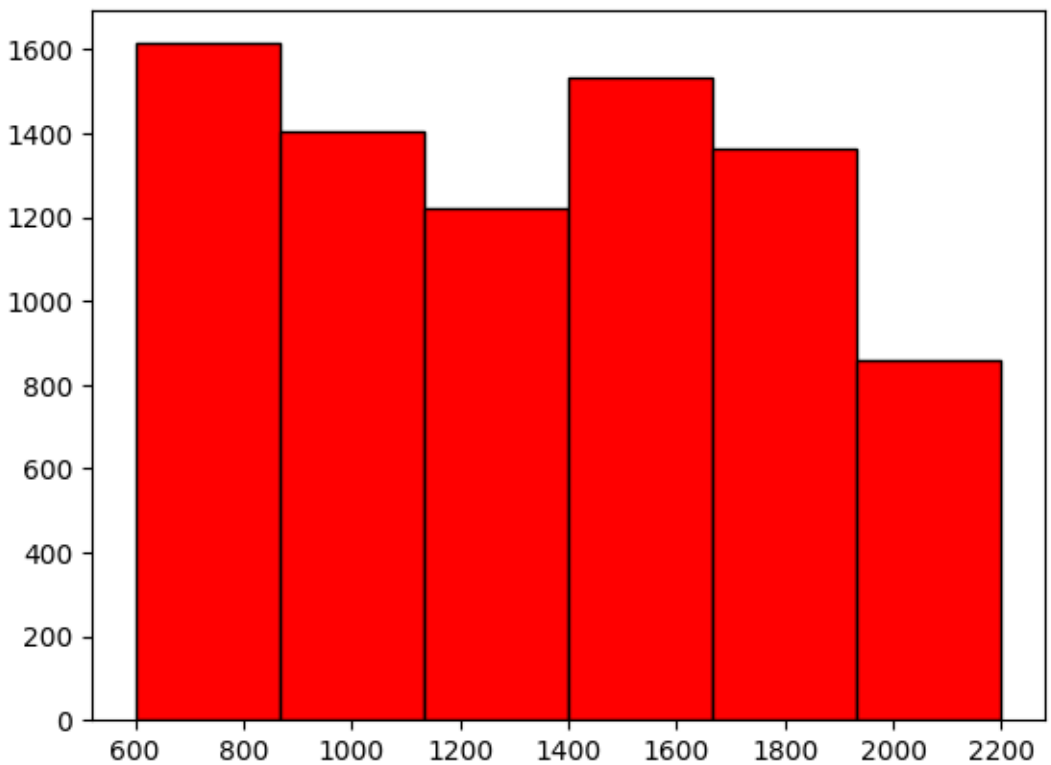
[19]:
```
plt.hist(project.Shipment_Delay)
```

[19]:
```
(array([5.373e+03, 1.532e+03, 5.950e+02, 2.020e+02, 7.900e+01, 4.000e+01,
        2.000e+01, 1.000e+01, 7.000e+00, 2.000e+00]),
 array([-10. ,  22.5,  55. ,  87.5, 120. , 152.5, 185. , 217.5, 250. ,
        282.5, 315. ]),
 <BarContainer object of 10 artists>)
```



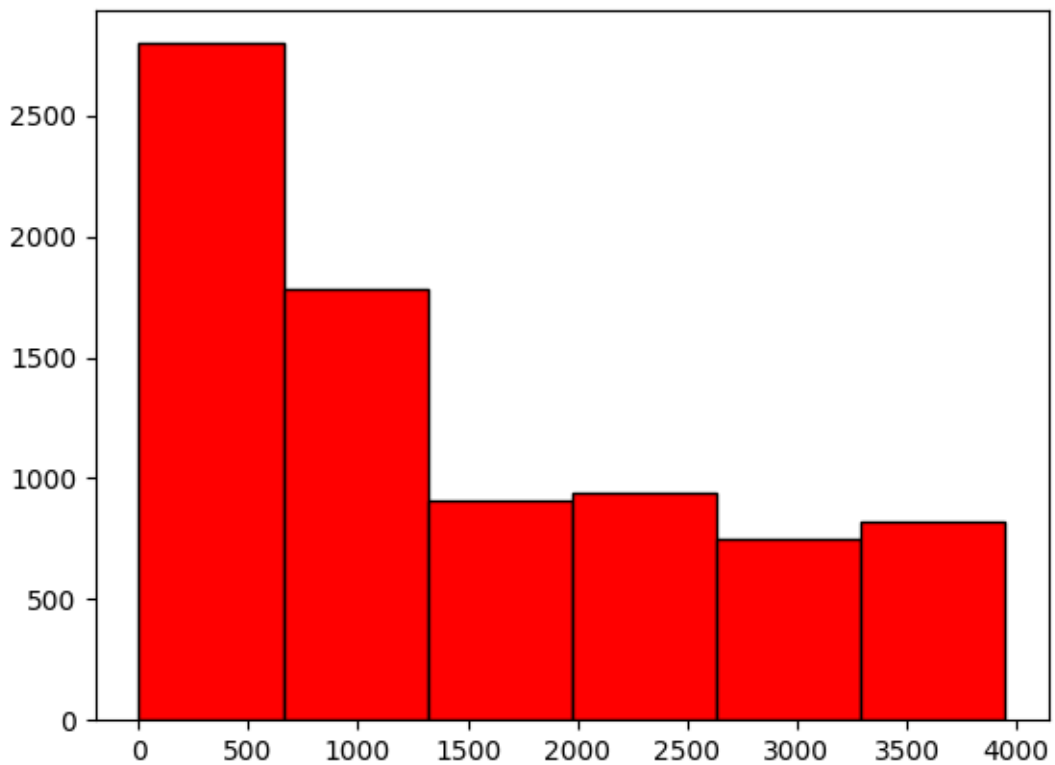
[20]:
```
plt.hist(project.Planned_Shipment_Time, color='red', edgecolor = "black", bins␣
↪= 6)
```

[20]:
```
(array([1614., 1406., 1221., 1533., 1364.,  861.]),
 array([ 600.        ,  866.66666667, 1133.33333333, 1400.        ,
        1666.66666667, 1933.33333333, 2200.        ]),
 <BarContainer object of 6 artists>)
```



[21]:
```
plt.hist(project.Carrier_Num, color='red', edgecolor = "black", bins = 6)
```

[21]:
```
(array([2797., 1783.,  907.,  940.,  748.,  824.]),
 array([1.000e+00, 6.590e+02, 1.317e+03, 1.975e+03, 2.633e+03, 3.291e+03,
        3.949e+03]),
 <BarContainer object of 6 artists>)
```



[22]:
```
help(plt.hist)
```

```
Help on function hist in module matplotlib.pyplot:

hist(x, bins=None, range=None, density=False, weights=None, cumulative=False,
bottom=None, histtype='bar', align='mid', orientation='vertical', rwidth=None,
log=False, color=None, label=None, stacked=False, *, data=None, **kwargs)
    Compute and plot a histogram.

    This method uses `numpy.histogram` to bin the data in *x* and count the
    number of values in each bin, then draws the distribution either as a
    `.BarContainer` or `.Polygon`. The *bins*, *range*, *density*, and
    *weights* parameters are forwarded to `numpy.histogram`.

    If the data has already been binned and counted, use `~.bar` or
    `~.stairs` to plot the distribution::

        counts, bins = np.histogram(x)
        plt.stairs(counts, bins)

    Alternatively, plot pre-computed bins and counts using ``hist()`` by
    treating each bin as a single point with a weight equal to its count::
```

```
    plt.hist(bins[:-1], bins, weights=counts)

The data input *x* can be a singular array, a list of datasets of
potentially different lengths ([*x0*, *x1*, …]), or a 2D ndarray in
which each column is a dataset. Note that the ndarray form is
transposed relative to the list form. If the input is an array, then
the return value is a tuple (*n*, *bins*, *patches*); if the input is a
sequence of arrays, then the return value is a tuple
([*n0*, *n1*, …], *bins*, [*patches0*, *patches1*, …]).

Masked arrays are not supported.

Parameters
----------
x : (n,) array or sequence of (n,) arrays
    Input values, this takes either a single array or a sequence of
    arrays which are not required to be of the same length.

bins : int or sequence or str, default: :rc:`hist.bins`
    If *bins* is an integer, it defines the number of equal-width bins
    in the range.

    If *bins* is a sequence, it defines the bin edges, including the
    left edge of the first bin and the right edge of the last bin;
    in this case, bins may be unequally spaced.  All but the last
    (righthand-most) bin is half-open.  In other words, if *bins* is::

        [1, 2, 3, 4]

    then the first bin is ``[1, 2)`` (including 1, but excluding 2) and
    the second ``[2, 3)``.  The last bin, however, is ``[3, 4]``, which
    *includes* 4.

    If *bins* is a string, it is one of the binning strategies
    supported by `numpy.histogram_bin_edges`: 'auto', 'fd', 'doane',
    'scott', 'stone', 'rice', 'sturges', or 'sqrt'.

range : tuple or None, default: None
    The lower and upper range of the bins. Lower and upper outliers
    are ignored. If not provided, *range* is ``(x.min(), x.max())``.
    Range has no effect if *bins* is a sequence.

    If *bins* is a sequence or *range* is specified, autoscaling
    is based on the specified bin range instead of the
    range of x.

density : bool, default: False
```

```
    If ``True``, draw and return a probability density: each bin
    will display the bin's raw count divided by the total number of
    counts *and the bin width*
    (``density = counts / (sum(counts) * np.diff(bins))``),
    so that the area under the histogram integrates to 1
    (``np.sum(density * np.diff(bins)) == 1``).

    If *stacked* is also ``True``, the sum of the histograms is
    normalized to 1.

weights : (n,) array-like or None, default: None
    An array of weights, of the same shape as *x*.  Each value in
    *x* only contributes its associated weight towards the bin count
    (instead of 1).  If *density* is ``True``, the weights are
    normalized, so that the integral of the density over the range
    remains 1.

cumulative : bool or -1, default: False
    If ``True``, then a histogram is computed where each bin gives the
    counts in that bin plus all bins for smaller values. The last bin
    gives the total number of datapoints.

    If *density* is also ``True`` then the histogram is normalized such
    that the last bin equals 1.

    If *cumulative* is a number less than 0 (e.g., -1), the direction
    of accumulation is reversed.  In this case, if *density* is also
    ``True``, then the histogram is normalized such that the first bin
    equals 1.

bottom : array-like, scalar, or None, default: None
    Location of the bottom of each bin, i.e. bins are drawn from
    ``bottom`` to ``bottom + hist(x, bins)`` If a scalar, the bottom
    of each bin is shifted by the same amount. If an array, each bin
    is shifted independently and the length of bottom must match the
    number of bins. If None, defaults to 0.

histtype : {'bar', 'barstacked', 'step', 'stepfilled'}, default: 'bar'
    The type of histogram to draw.

    - 'bar' is a traditional bar-type histogram.  If multiple data
      are given the bars are arranged side by side.
    - 'barstacked' is a bar-type histogram where multiple
      data are stacked on top of each other.
    - 'step' generates a lineplot that is by default unfilled.
    - 'stepfilled' generates a lineplot that is by default filled.

align : {'left', 'mid', 'right'}, default: 'mid'
```

```
    The horizontal alignment of the histogram bars.

    - 'left': bars are centered on the left bin edges.
    - 'mid': bars are centered between the bin edges.
    - 'right': bars are centered on the right bin edges.

orientation : {'vertical', 'horizontal'}, default: 'vertical'
    If 'horizontal', `~.Axes.barh` will be used for bar-type histograms
    and the *bottom* kwarg will be the left edges.

rwidth : float or None, default: None
    The relative width of the bars as a fraction of the bin width.  If
    ``None``, automatically compute the width.

    Ignored if *histtype* is 'step' or 'stepfilled'.

log : bool, default: False
    If ``True``, the histogram axis will be set to a log scale.

color : color or array-like of colors or None, default: None
    Color or sequence of colors, one per dataset.  Default (``None``)
    uses the standard line color sequence.

label : str or None, default: None
    String, or sequence of strings to match multiple datasets.  Bar
    charts yield multiple patches per dataset, but only the first gets
    the label, so that `~.Axes.legend` will work as expected.

stacked : bool, default: False
    If ``True``, multiple data are stacked on top of each other If
    ``False`` multiple data are arranged side by side if histtype is
    'bar' or on top of each other if histtype is 'step'

Returns
-------
n : array or list of arrays
    The values of the histogram bins. See *density* and *weights* for a
    description of the possible semantics.  If input *x* is an array,
    then this is an array of length *nbins*. If input is a sequence of
    arrays ``[data1, data2, …]``, then this is a list of arrays with
    the values of the histograms for each of the arrays in the same
    order.  The dtype of the array *n* (or of its element arrays) will
    always be float even if no weighting or normalization is used.

bins : array
    The edges of the bins. Length nbins + 1 (nbins left edges and right
    edge of last bin).  Always a single array even when multiple data
    sets are passed in.
```

```
    patches : `.BarContainer` or list of a single `.Polygon` or list of such
objects
        Container of individual artists used to create the histogram
        or list of such containers if there are multiple input datasets.

    Other Parameters
    ----------------
    data : indexable object, optional
        If given, the following parameters also accept a string ``s``, which is
        interpreted as ``data[s]`` (unless this raises an exception):

        *x*, *weights*

    **kwargs
        `~matplotlib.patches.Patch` properties

    See Also
    --------
    hist2d : 2D histogram with rectangular bins
    hexbin : 2D histogram with hexagonal bins

    Notes
    -----
    For large numbers of bins (>1000), plotting can be significantly faster
    if *histtype* is set to 'step' or 'stepfilled' rather than 'bar' or
    'barstacked'.
```

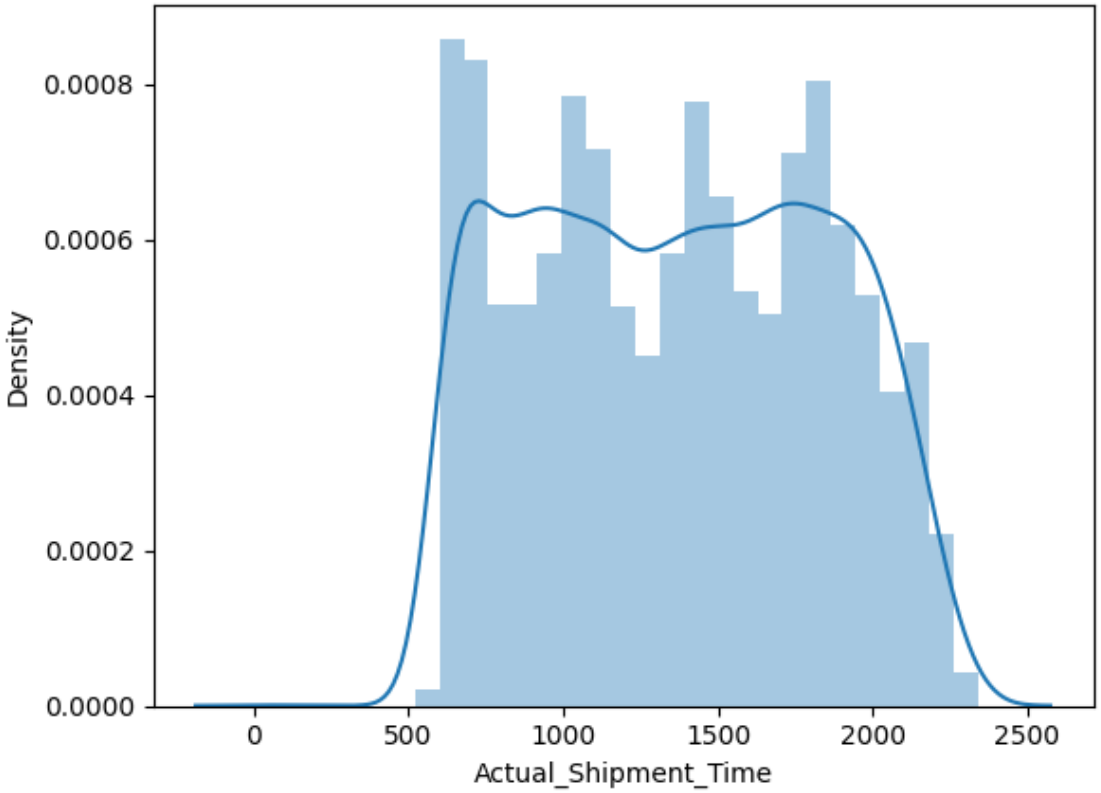
6 Histogram using Seaborn

[23]:
```
import seaborn as sns
```

[24]:
```
sns.distplot(project.Actual_Shipment_Time)
```

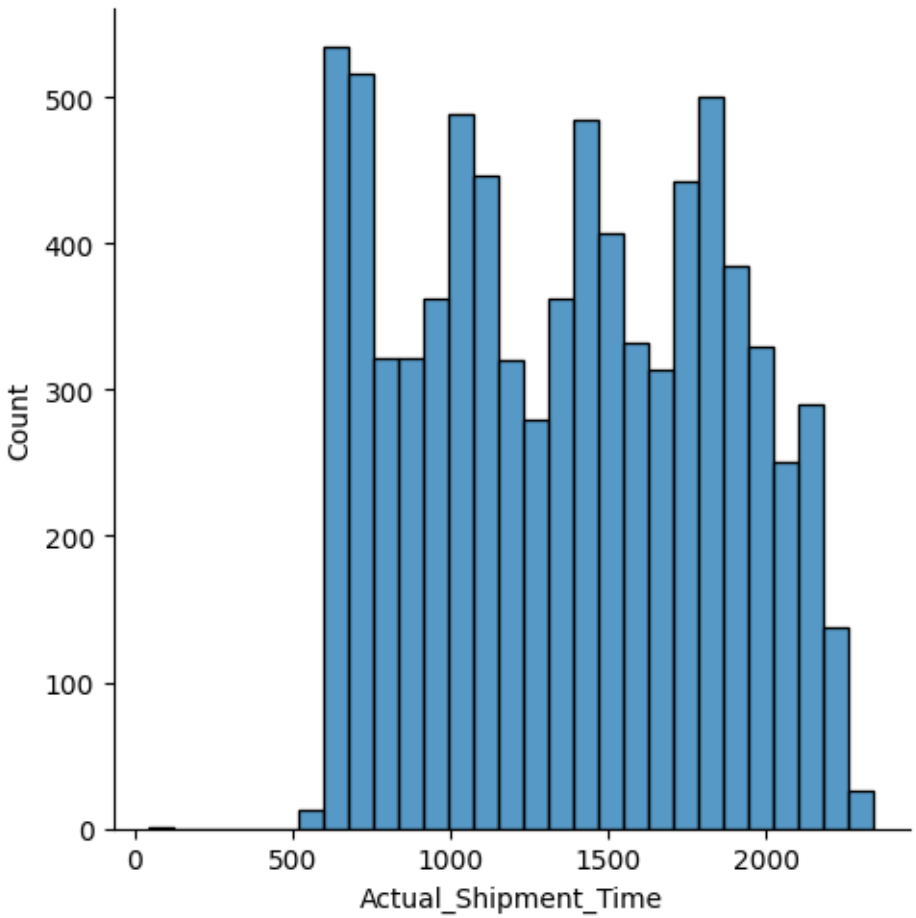
```
<ipython-input-24-67120540fb27>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(project.Actual_Shipment_Time)
```

[24]: <Axes: xlabel='Actual_Shipment_Time', ylabel='Density'>



[25]:
```
sns.displot(project.Actual_Shipment_Time)
```

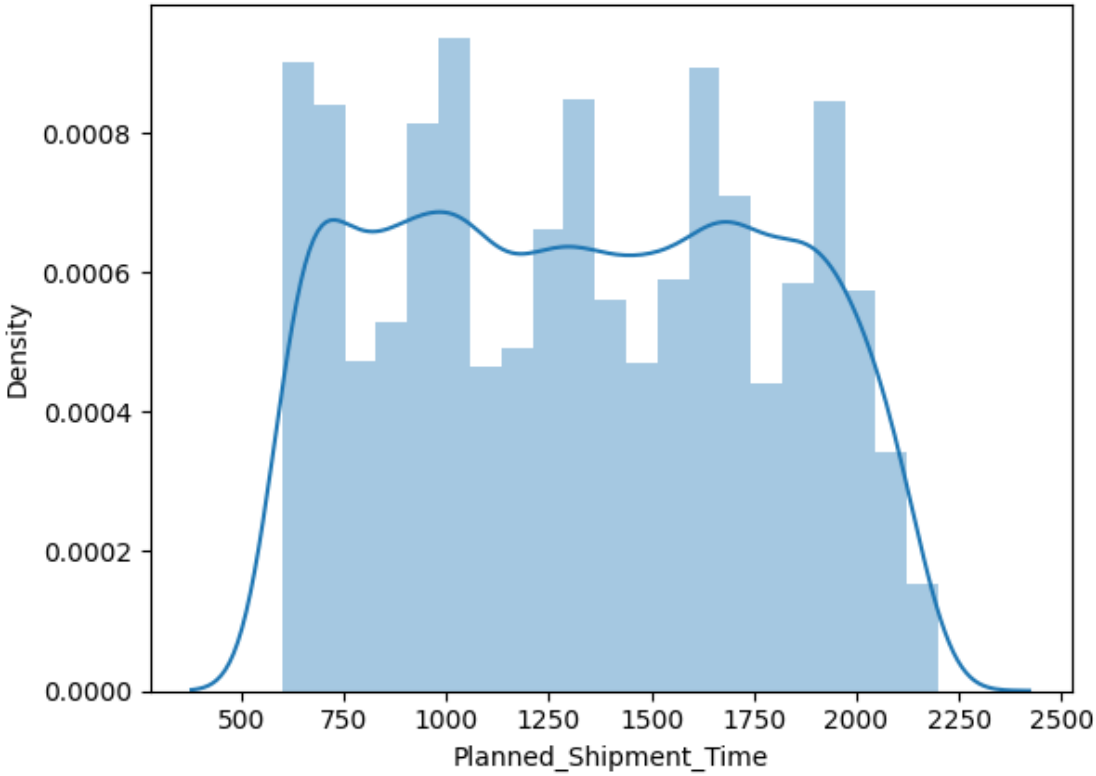
[25]: <seaborn.axisgrid.FacetGrid at 0x7fe0564effd0>



[26]:
```
sns.distplot(project.Planned_Shipment_Time)
```

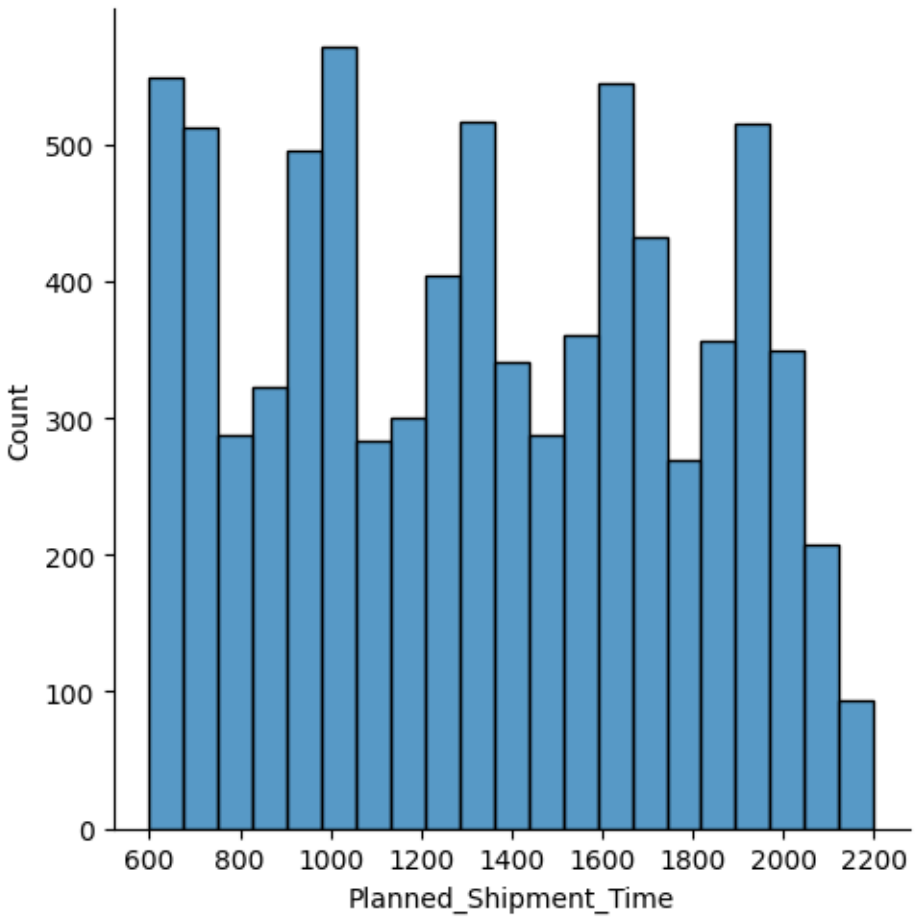
```
<ipython-input-26-7a7907fb2066>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(project.Planned_Shipment_Time)
```

[26]:
```
<Axes: xlabel='Planned_Shipment_Time', ylabel='Density'>
```



[27]:
```
sns.displot(project.Planned_Shipment_Time)
```

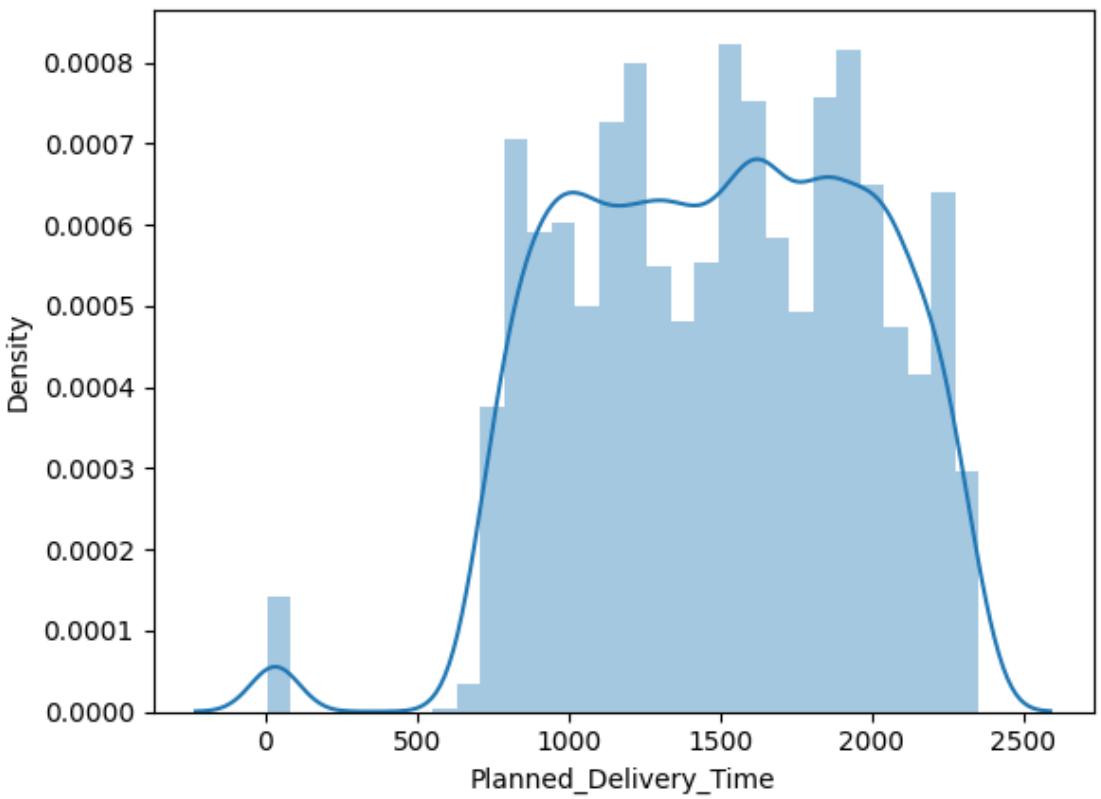
[27]: <seaborn.axisgrid.FacetGrid at 0x7fe0543efc40>



[28]:
```
sns.distplot(project.Planned_Delivery_Time)
```

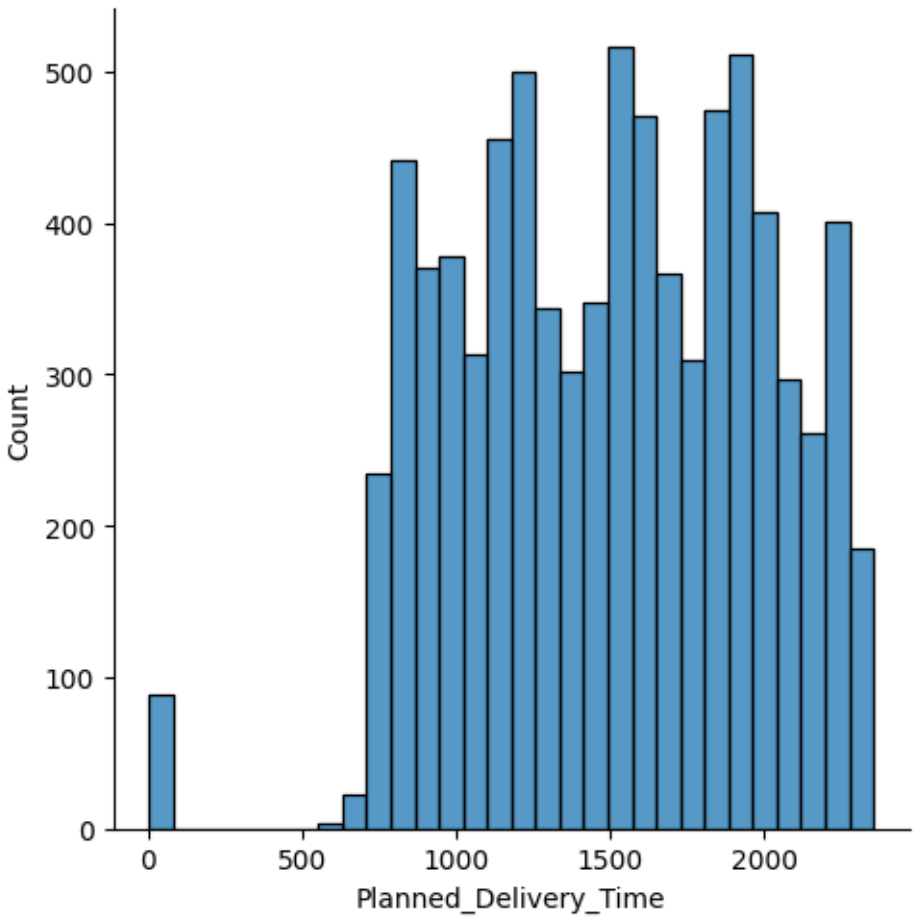
```
<ipython-input-28-4422adac3f3c>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(project.Planned_Delivery_Time)
```

[28]:
```
<Axes: xlabel='Planned_Delivery_Time', ylabel='Density'>
```



[29]:
```
sns.displot(project.Planned_Delivery_Time)
```

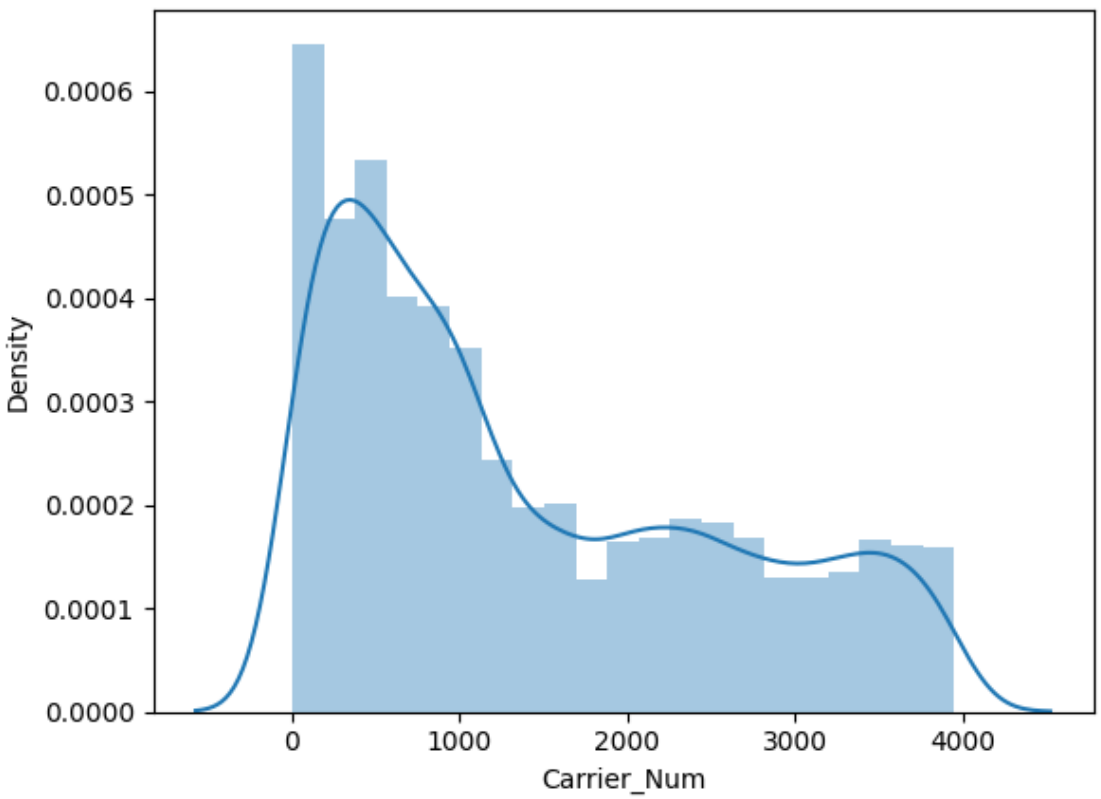
[29]: <seaborn.axisgrid.FacetGrid at 0x7fe0541f1c60>



[30]:
```
sns.distplot(project.Carrier_Num)
```

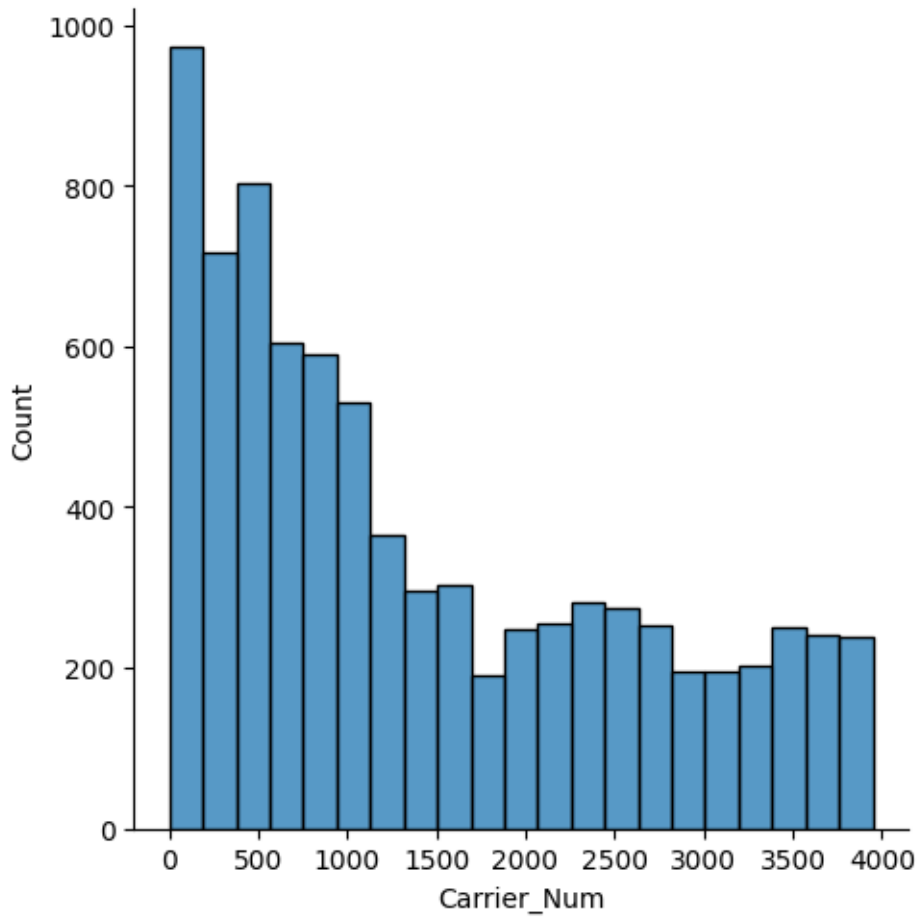
```
<ipython-input-30-1ac42355d1d7>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(project.Carrier_Num)
```

[30]:
```
<Axes: xlabel='Carrier_Num', ylabel='Density'>
```



[31]:
```
sns.displot(project.Carrier_Num)
```

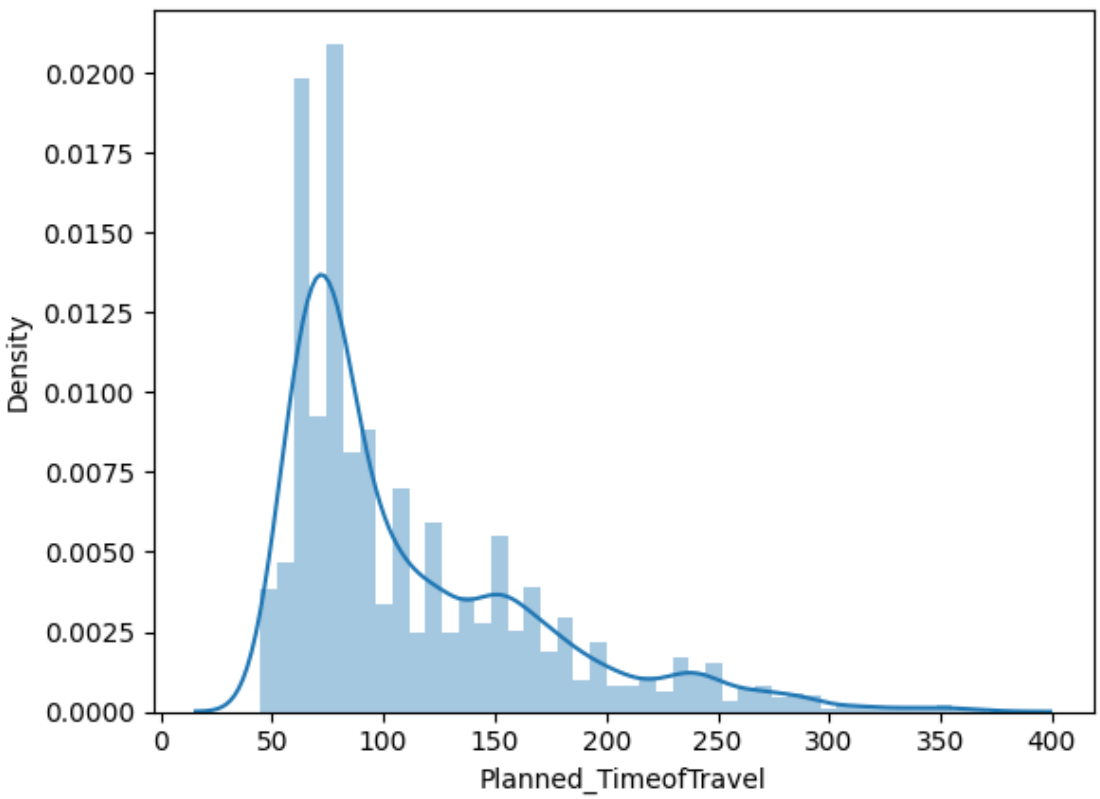
[31]: <seaborn.axisgrid.FacetGrid at 0x7fe0541b7760>



[32]:
```
sns.distplot(project.Planned_TimeofTravel)
```

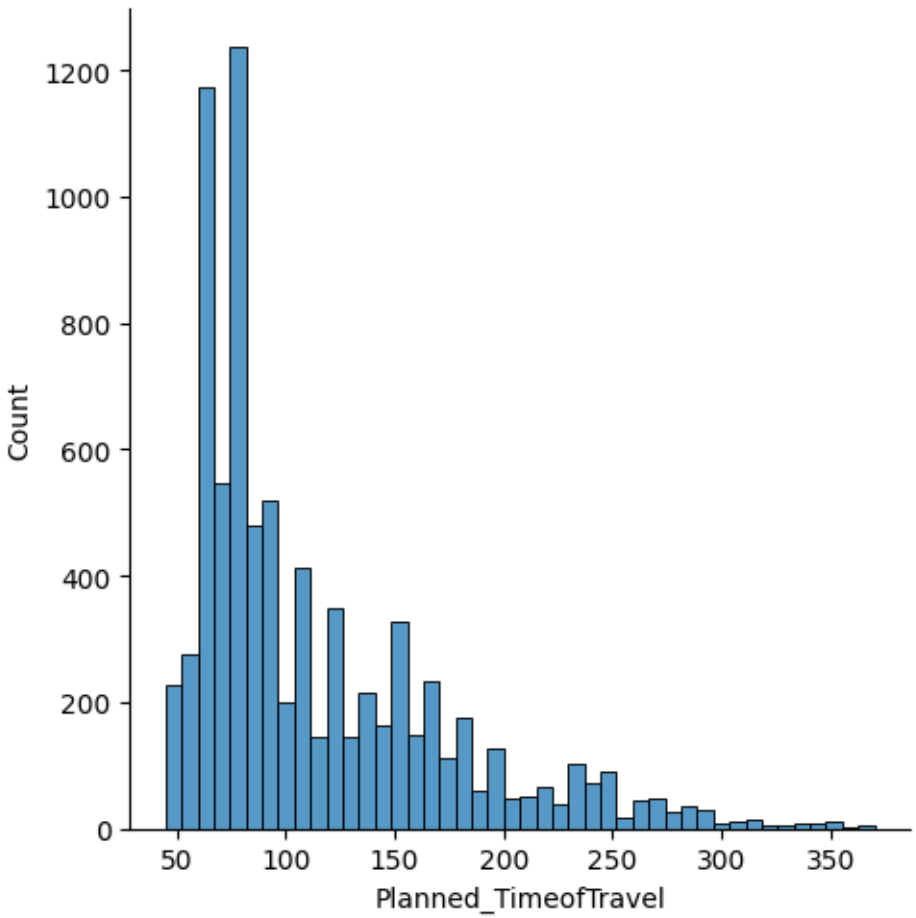
```
<ipython-input-32-d8bb81c4b701>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(project.Planned_TimeofTravel)
```

[32]:
```
<Axes: xlabel='Planned_TimeofTravel', ylabel='Density'>
```



[33]:
```
sns.displot(project.Planned_TimeofTravel)
```

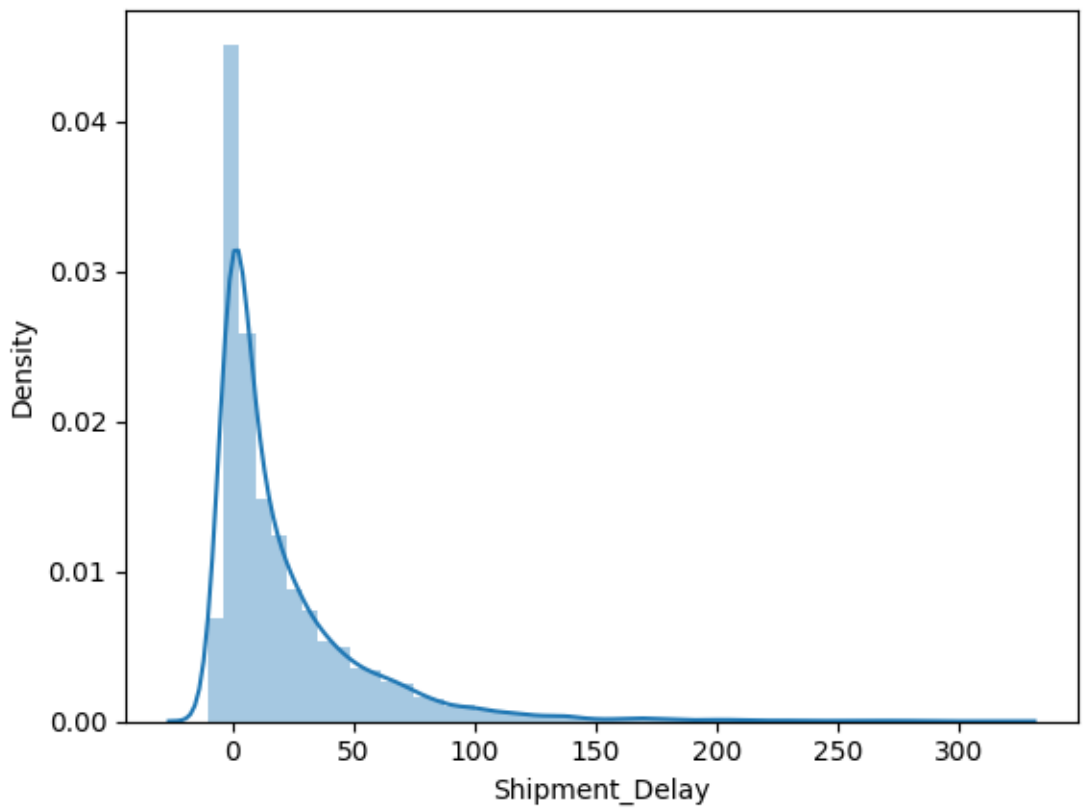
[33]: <seaborn.axisgrid.FacetGrid at 0x7fe0567cff10>



[34]:
```
sns.distplot(project.Shipment_Delay)
```

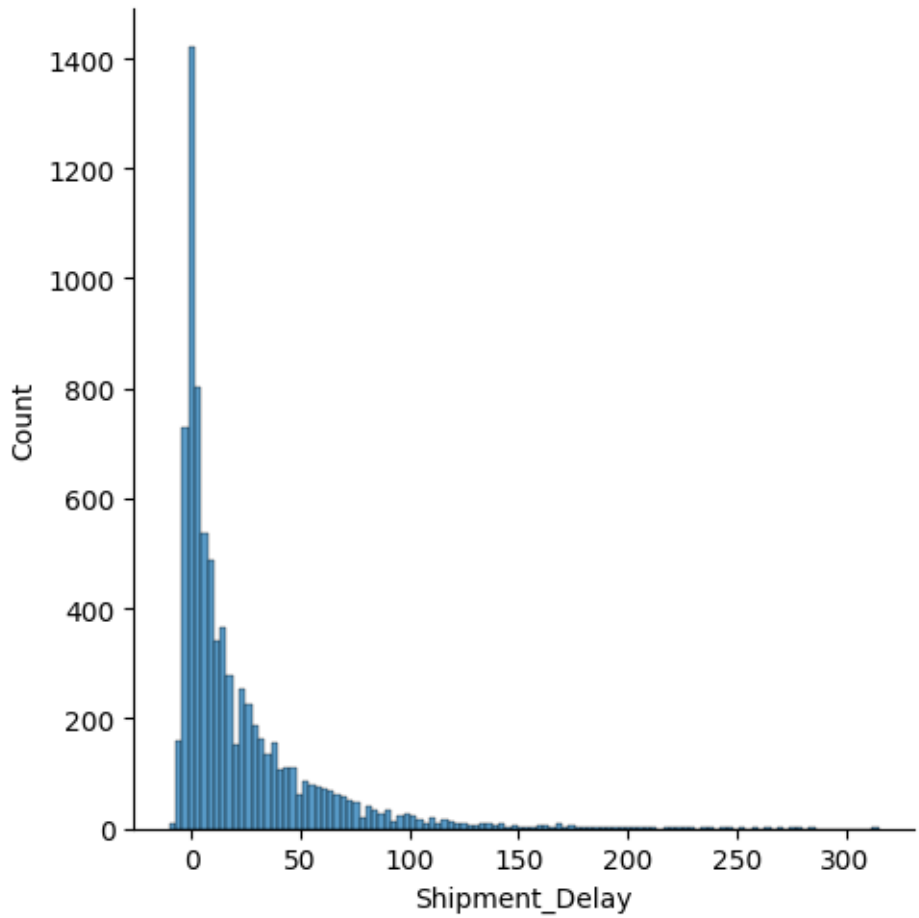
```
<ipython-input-34-144c4f121fc1>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(project.Shipment_Delay)
```

[34]:
```
<Axes: xlabel='Shipment_Delay', ylabel='Density'>
```



[35]:
```
sns.displot(project.Shipment_Delay)
```

[35]: <seaborn.axisgrid.FacetGrid at 0x7fe053ce6860>



7 Boxplot

[36]:
```
plt.figure()
```

[36]:
```
<Figure size 640x480 with 0 Axes>
```

```
<Figure size 640x480 with 0 Axes>
```

[37]:
```
plt.boxplot(project.Actual_Shipment_Time)
```

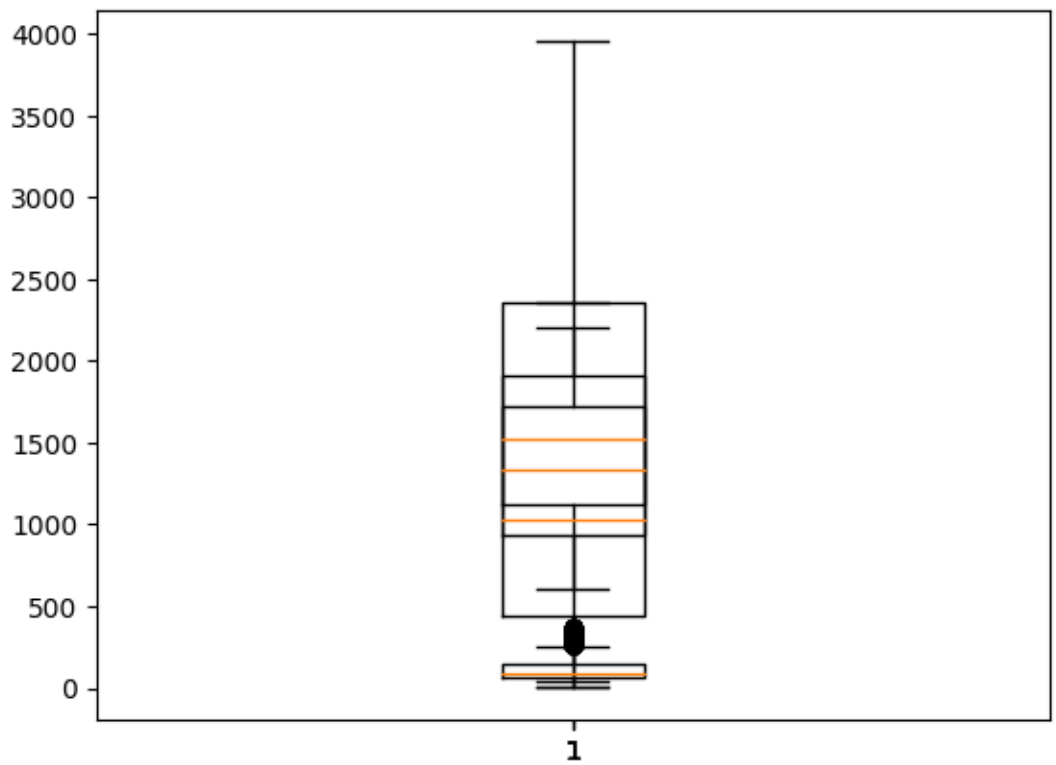
```
Help on function boxplot in module matplotlib.pyplot:

boxplot(x, notch=None, sym=None, vert=None, whis=None, positions=None,
widths=None, patch_artist=None, bootstrap=None, usermedians=None,
conf_intervals=None, meanline=None, showmeans=None, showcaps=None, showbox=None,
showfliers=None, boxprops=None, labels=None, flierprops=None, medianprops=None,
meanprops=None, capprops=None, whiskerprops=None, manage_ticks=True,
```

```
autorange=False, zorder=None, capwidths=None, *, data=None)
    Draw a box and whisker plot.

    The box extends from the first quartile (Q1) to the third
    quartile (Q3) of the data, with a line at the median.  The
    whiskers extend from the box by 1.5x the inter-quartile range
    (IQR).  Flier points are those past the end of the whiskers.
    See https://en.wikipedia.org/wiki/Box_plot for reference.

    .. code-block:: none

              Q1-1.5IQR   Q1   median  Q3   Q3+1.5IQR
                           |-----:-----|
            o      |--------|     :     |--------|    o  o
                           |-----:-----|
         flier             <----------->            fliers
                                IQR


    Parameters
    ----------
    x : Array or a sequence of vectors.
        The input data.  If a 2D array, a boxplot is drawn for each column
        in *x*.  If a sequence of 1D arrays, a boxplot is drawn for each
        array in *x*.

    notch : bool, default: False
        Whether to draw a notched boxplot (`True`), or a rectangular
        boxplot (`False`).  The notches represent the confidence interval
        (CI) around the median.  The documentation for *bootstrap*
        describes how the locations of the notches are computed by
        default, but their locations may also be overridden by setting the
        *conf_intervals* parameter.

        .. note::

            In cases where the values of the CI are less than the
            lower quartile or greater than the upper quartile, the
            notches will extend beyond the box, giving it a
            distinctive "flipped" appearance. This is expected
            behavior and consistent with other statistical
            visualization packages.

    sym : str, optional
        The default symbol for flier points.  An empty string ('') hides
        the fliers.  If `None`, then the fliers default to 'b+'.  More
        control is provided by the *flierprops* parameter.
```

```
vert : bool, default: True
    If `True`, draws vertical boxes.
    If `False`, draw horizontal boxes.

whis : float or (float, float), default: 1.5
    The position of the whiskers.

    If a float, the lower whisker is at the lowest datum above
    ``Q1 - whis*(Q3-Q1)``, and the upper whisker at the highest datum
    below ``Q3 + whis*(Q3-Q1)``, where Q1 and Q3 are the first and
    third quartiles.  The default value of ``whis = 1.5`` corresponds
    to Tukey's original definition of boxplots.

    If a pair of floats, they indicate the percentiles at which to
    draw the whiskers (e.g., (5, 95)).  In particular, setting this to
    (0, 100) results in whiskers covering the whole range of the data.

    In the edge case where ``Q1 == Q3``, *whis* is automatically set
    to (0, 100) (cover the whole range of the data) if *autorange* is
    True.

    Beyond the whiskers, data are considered outliers and are plotted
    as individual points.

bootstrap : int, optional
    Specifies whether to bootstrap the confidence intervals
    around the median for notched boxplots. If *bootstrap* is
    None, no bootstrapping is performed, and notches are
    calculated using a Gaussian-based asymptotic approximation
    (see McGill, R., Tukey, J.W., and Larsen, W.A., 1978, and
    Kendall and Stuart, 1967). Otherwise, bootstrap specifies
    the number of times to bootstrap the median to determine its
    95% confidence intervals. Values between 1000 and 10000 are
    recommended.

usermedians : 1D array-like, optional
    A 1D array-like of length ``len(x)``.  Each entry that is not
    `None` forces the value of the median for the corresponding
    dataset.  For entries that are `None`, the medians are computed
    by Matplotlib as normal.

conf_intervals : array-like, optional
    A 2D array-like of shape ``(len(x), 2)``.  Each entry that is not
    None forces the location of the corresponding notch (which is
    only drawn if *notch* is `True`).  For entries that are `None`,
    the notches are computed by the method specified by the other
    parameters (e.g., *bootstrap*).
```

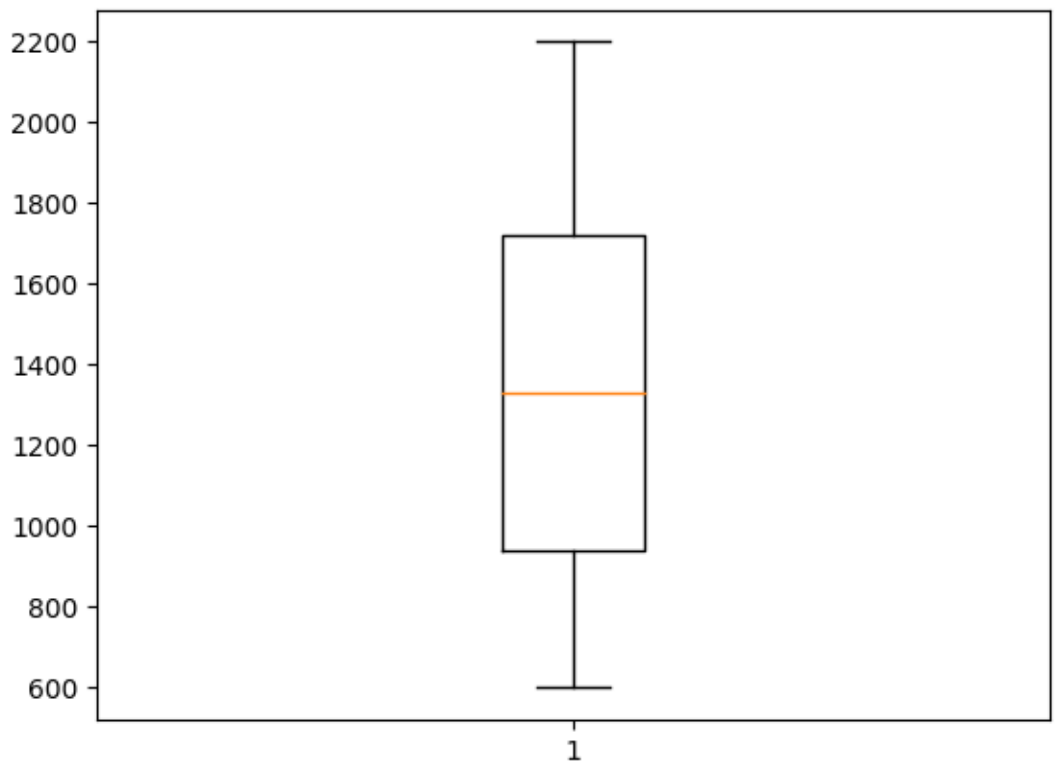
```
positions : array-like, optional
    The positions of the boxes. The ticks and limits are
    automatically set to match the positions. Defaults to
    ``range(1, N+1)`` where N is the number of boxes to be drawn.

widths : float or array-like
    The widths of the boxes.  The default is 0.5, or ``0.15*(distance
    between extreme positions)``, if that is smaller.

patch_artist : bool, default: False
    If `False` produces boxes with the Line2D artist. Otherwise,
    boxes are drawn with Patch artists.

labels : sequence, optional
    Labels for each dataset (one per dataset).

manage_ticks : bool, default: True
    If True, the tick locations and labels will be adjusted to match
    the boxplot positions.

autorange : bool, default: False
    When `True` and the data are distributed such that the 25th and
    75th percentiles are equal, *whis* is set to (0, 100) such
    that the whisker ends are at the minimum and maximum of the data.

meanline : bool, default: False
    If `True` (and *showmeans* is `True`), will try to render the
    mean as a line spanning the full width of the box according to
    *meanprops* (see below).  Not recommended if *shownotches* is also
    True.  Otherwise, means will be shown as points.

zorder : float, default: ``Line2D.zorder = 2``
    The zorder of the boxplot.

Returns
-------
dict
  A dictionary mapping each component of the boxplot to a list
  of the `.Line2D` instances created. That dictionary has the
  following keys (assuming vertical boxplots):

  - ``boxes``: the main body of the boxplot showing the
    quartiles and the median's confidence intervals if
    enabled.

  - ``medians``: horizontal lines at the median of each box.

  - ``whiskers``: the vertical lines extending to the most
```

```
    extreme, non-outlier data points.

  - ``caps``: the horizontal lines at the ends of the
    whiskers.

  - ``fliers``: points representing data that extend beyond
    the whiskers (fliers).

  - ``means``: points or lines representing the means.

Other Parameters
----------------
showcaps : bool, default: True
    Show the caps on the ends of whiskers.
showbox : bool, default: True
    Show the central box.
showfliers : bool, default: True
    Show the outliers beyond the caps.
showmeans : bool, default: False
    Show the arithmetic means.
capprops : dict, default: None
    The style of the caps.
capwidths : float or array, default: None
    The widths of the caps.
boxprops : dict, default: None
    The style of the box.
whiskerprops : dict, default: None
    The style of the whiskers.
flierprops : dict, default: None
    The style of the fliers.
medianprops : dict, default: None
    The style of the median.
meanprops : dict, default: None
    The style of the mean.
data : indexable object, optional
    If given, all parameters also accept a string ``s``, which is
    interpreted as ``data[s]`` (unless this raises an exception).

See Also
--------
violinplot : Draw an estimate of the probability density function.
```



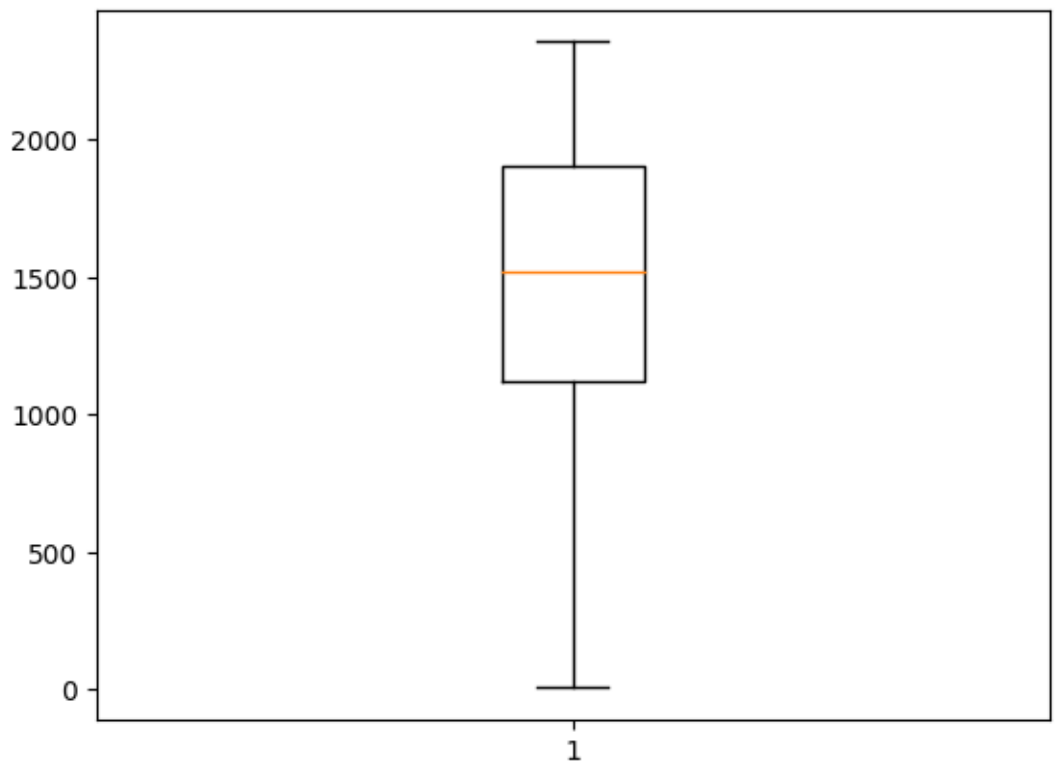
[38]:
```
plt.boxplot(project.Planned_Shipment_Time)
```

[38]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fe053abfd00>,
  <matplotlib.lines.Line2D at 0x7fe053abffa0>],
 'caps': [<matplotlib.lines.Line2D at 0x7fe0538ec280>,
  <matplotlib.lines.Line2D at 0x7fe0538ec520>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fe053abfa60>],
 'medians': [<matplotlib.lines.Line2D at 0x7fe0538ec7c0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fe0538eca60>],
 'means': []}
```



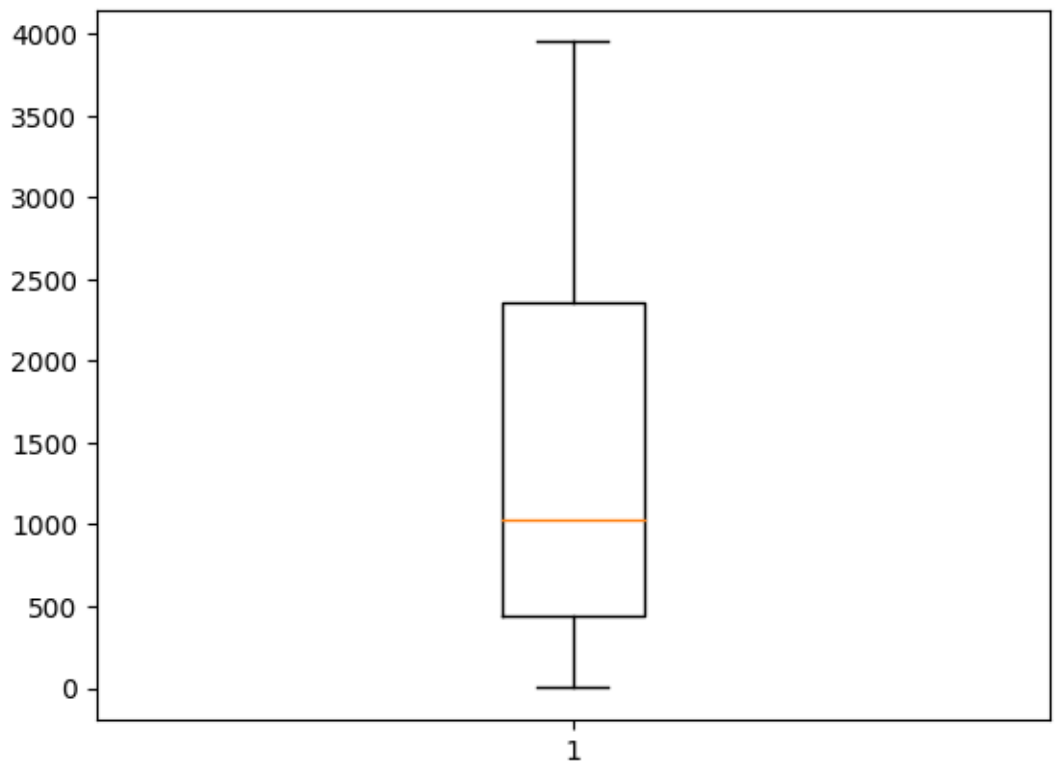
[39]:
```
plt.boxplot(project.Planned_Delivery_Time)
```

[39]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fe053946c50>,
  <matplotlib.lines.Line2D at 0x7fe053946ef0>],
 'caps': [<matplotlib.lines.Line2D at 0x7fe053947190>,
  <matplotlib.lines.Line2D at 0x7fe053947430>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fe0539469b0>],
 'medians': [<matplotlib.lines.Line2D at 0x7fe0539476d0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fe053947970>],
 'means': []}
```



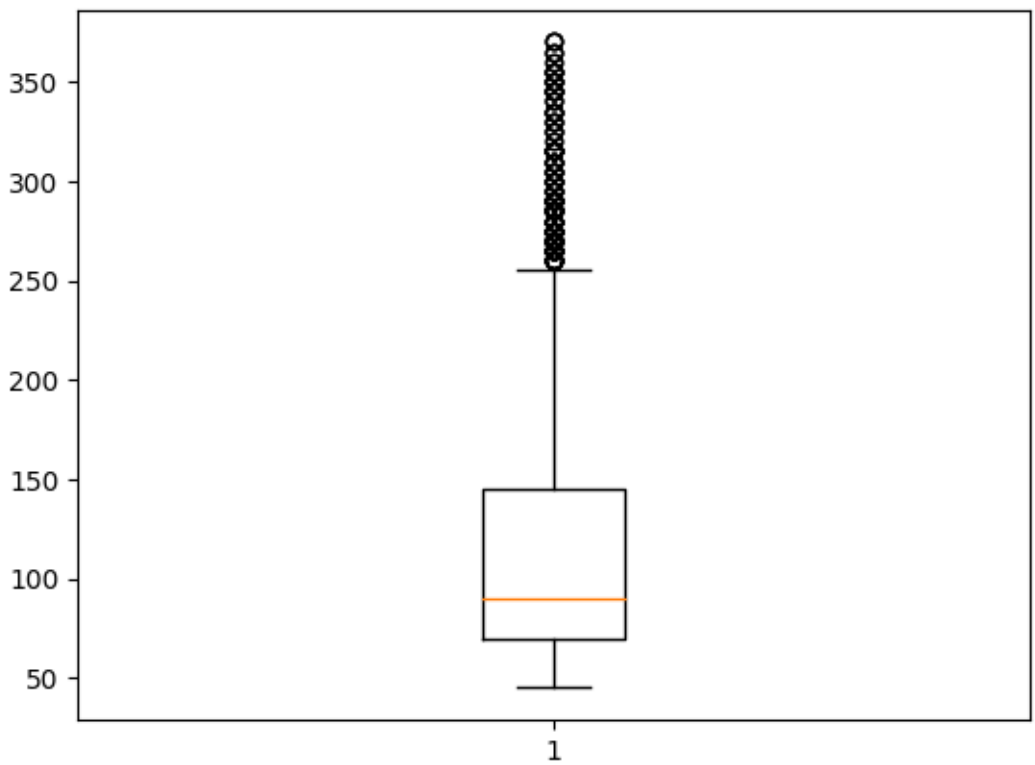
[40]:
```
plt.boxplot(project.Carrier_Num)
```

[40]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fe0539a2d10>,
  <matplotlib.lines.Line2D at 0x7fe0539d4160>],
 'caps': [<matplotlib.lines.Line2D at 0x7fe0539d4340>,
  <matplotlib.lines.Line2D at 0x7fe0539d45e0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fe0539a3f10>],
 'medians': [<matplotlib.lines.Line2D at 0x7fe0539d4880>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fe0539d4b20>],
 'means': []}
```



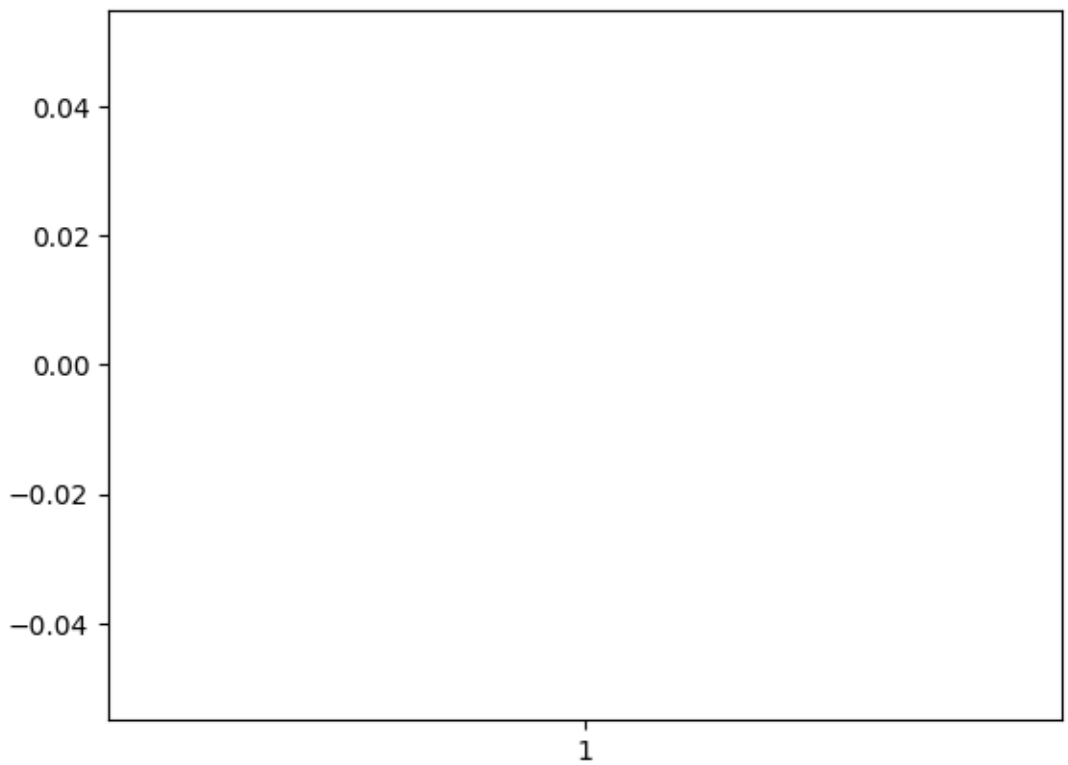
[41]:
```
plt.boxplot(project.Planned_TimeofTravel)
```

[41]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fe053823130>,
  <matplotlib.lines.Line2D at 0x7fe0538233d0>],
 'caps': [<matplotlib.lines.Line2D at 0x7fe053823670>,
  <matplotlib.lines.Line2D at 0x7fe053823910>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fe053822e90>],
 'medians': [<matplotlib.lines.Line2D at 0x7fe053823bb0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fe053823e50>],
 'means': []}
```



[42]:
```
plt.boxplot(project.Shipment_Delay)
```

[42]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fe0538a8be0>,
  <matplotlib.lines.Line2D at 0x7fe0538a8e80>],
 'caps': [<matplotlib.lines.Line2D at 0x7fe0538a9120>,
  <matplotlib.lines.Line2D at 0x7fe0538a93c0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fe0539d59f0>],
 'medians': [<matplotlib.lines.Line2D at 0x7fe0538a9660>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fe0538a9900>],
 'means': []}
```



[43]:
```
help(plt.boxplot)
```

```
Help on function boxplot in module matplotlib.pyplot:

boxplot(x, notch=None, sym=None, vert=None, whis=None, positions=None,
widths=None, patch_artist=None, bootstrap=None, usermedians=None,
conf_intervals=None, meanline=None, showmeans=None, showcaps=None, showbox=None,
showfliers=None, boxprops=None, labels=None, flierprops=None, medianprops=None,
meanprops=None, capprops=None, whiskerprops=None, manage_ticks=True,
autorange=False, zorder=None, capwidths=None, *, data=None)
    Draw a box and whisker plot.

    The box extends from the first quartile (Q1) to the third
    quartile (Q3) of the data, with a line at the median.  The
    whiskers extend from the box by 1.5x the inter-quartile range
    (IQR).  Flier points are those past the end of the whiskers.
    See https://en.wikipedia.org/wiki/Box_plot for reference.

    .. code-block:: none

              Q1-1.5IQR   Q1   median  Q3   Q3+1.5IQR
                           |-----:-----|
```

```
       o      |--------|     :     |--------|    o  o
                         |-----:-----|
     flier               <----------->            fliers
                              IQR


Parameters
----------
x : Array or a sequence of vectors.
    The input data.  If a 2D array, a boxplot is drawn for each column
    in *x*.  If a sequence of 1D arrays, a boxplot is drawn for each
    array in *x*.

notch : bool, default: False
    Whether to draw a notched boxplot (`True`), or a rectangular
    boxplot (`False`).  The notches represent the confidence interval
    (CI) around the median.  The documentation for *bootstrap*
    describes how the locations of the notches are computed by
    default, but their locations may also be overridden by setting the
    *conf_intervals* parameter.

    .. note::

        In cases where the values of the CI are less than the
        lower quartile or greater than the upper quartile, the
        notches will extend beyond the box, giving it a
        distinctive "flipped" appearance. This is expected
        behavior and consistent with other statistical
        visualization packages.

sym : str, optional
    The default symbol for flier points.  An empty string ('') hides
    the fliers.  If `None`, then the fliers default to 'b+'.  More
    control is provided by the *flierprops* parameter.

vert : bool, default: True
    If `True`, draws vertical boxes.
    If `False`, draw horizontal boxes.

whis : float or (float, float), default: 1.5
    The position of the whiskers.

    If a float, the lower whisker is at the lowest datum above
    ``Q1 - whis*(Q3-Q1)``, and the upper whisker at the highest datum
    below ``Q3 + whis*(Q3-Q1)``, where Q1 and Q3 are the first and
    third quartiles.  The default value of ``whis = 1.5`` corresponds
    to Tukey's original definition of boxplots.
```

```
    If a pair of floats, they indicate the percentiles at which to
    draw the whiskers (e.g., (5, 95)).  In particular, setting this to
    (0, 100) results in whiskers covering the whole range of the data.

    In the edge case where ``Q1 == Q3``, *whis* is automatically set
    to (0, 100) (cover the whole range of the data) if *autorange* is
    True.

    Beyond the whiskers, data are considered outliers and are plotted
    as individual points.

bootstrap : int, optional
    Specifies whether to bootstrap the confidence intervals
    around the median for notched boxplots. If *bootstrap* is
    None, no bootstrapping is performed, and notches are
    calculated using a Gaussian-based asymptotic approximation
    (see McGill, R., Tukey, J.W., and Larsen, W.A., 1978, and
    Kendall and Stuart, 1967). Otherwise, bootstrap specifies
    the number of times to bootstrap the median to determine its
    95% confidence intervals. Values between 1000 and 10000 are
    recommended.

usermedians : 1D array-like, optional
    A 1D array-like of length ``len(x)``.  Each entry that is not
    `None` forces the value of the median for the corresponding
    dataset.  For entries that are `None`, the medians are computed
    by Matplotlib as normal.

conf_intervals : array-like, optional
    A 2D array-like of shape ``(len(x), 2)``.  Each entry that is not
    None forces the location of the corresponding notch (which is
    only drawn if *notch* is `True`).  For entries that are `None`,
    the notches are computed by the method specified by the other
    parameters (e.g., *bootstrap*).

positions : array-like, optional
    The positions of the boxes. The ticks and limits are
    automatically set to match the positions. Defaults to
    ``range(1, N+1)`` where N is the number of boxes to be drawn.

widths : float or array-like
    The widths of the boxes.  The default is 0.5, or ``0.15*(distance
    between extreme positions)``, if that is smaller.

patch_artist : bool, default: False
    If `False` produces boxes with the Line2D artist. Otherwise,
    boxes are drawn with Patch artists.
```

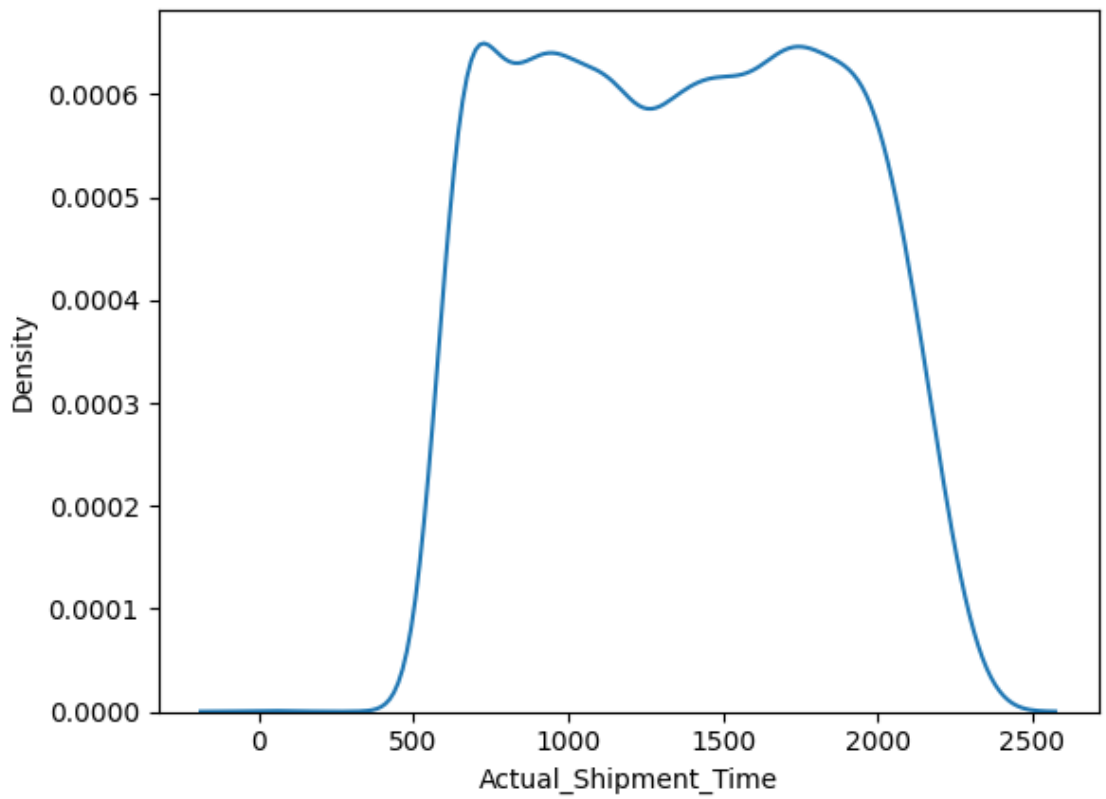
```
labels : sequence, optional
    Labels for each dataset (one per dataset).

manage_ticks : bool, default: True
    If True, the tick locations and labels will be adjusted to match
    the boxplot positions.

autorange : bool, default: False
    When `True` and the data are distributed such that the 25th and
    75th percentiles are equal, *whis* is set to (0, 100) such
    that the whisker ends are at the minimum and maximum of the data.

meanline : bool, default: False
    If `True` (and *showmeans* is `True`), will try to render the
    mean as a line spanning the full width of the box according to
    *meanprops* (see below).  Not recommended if *shownotches* is also
    True.  Otherwise, means will be shown as points.

zorder : float, default: ``Line2D.zorder = 2``
    The zorder of the boxplot.

Returns
-------
dict
  A dictionary mapping each component of the boxplot to a list
  of the `.Line2D` instances created. That dictionary has the
  following keys (assuming vertical boxplots):

  - ``boxes``: the main body of the boxplot showing the
    quartiles and the median's confidence intervals if
    enabled.

  - ``medians``: horizontal lines at the median of each box.

  - ``whiskers``: the vertical lines extending to the most
    extreme, non-outlier data points.

  - ``caps``: the horizontal lines at the ends of the
    whiskers.

  - ``fliers``: points representing data that extend beyond
    the whiskers (fliers).

  - ``means``: points or lines representing the means.

Other Parameters
----------------
showcaps : bool, default: True
```

```
    Show the caps on the ends of whiskers.
showbox : bool, default: True
    Show the central box.
showfliers : bool, default: True
    Show the outliers beyond the caps.
showmeans : bool, default: False
    Show the arithmetic means.
capprops : dict, default: None
    The style of the caps.
capwidths : float or array, default: None
    The widths of the caps.
boxprops : dict, default: None
    The style of the box.
whiskerprops : dict, default: None
    The style of the whiskers.
flierprops : dict, default: None
    The style of the fliers.
medianprops : dict, default: None
    The style of the median.
meanprops : dict, default: None
    The style of the mean.
data : indexable object, optional
    If given, all parameters also accept a string ``s``, which is
    interpreted as ``data[s]`` (unless this raises an exception).

See Also
--------
violinplot : Draw an estimate of the probability density function.
```

8 Density Plot

[44]:
```
sns.kdeplot(project.Actual_Shipment_Time)
```

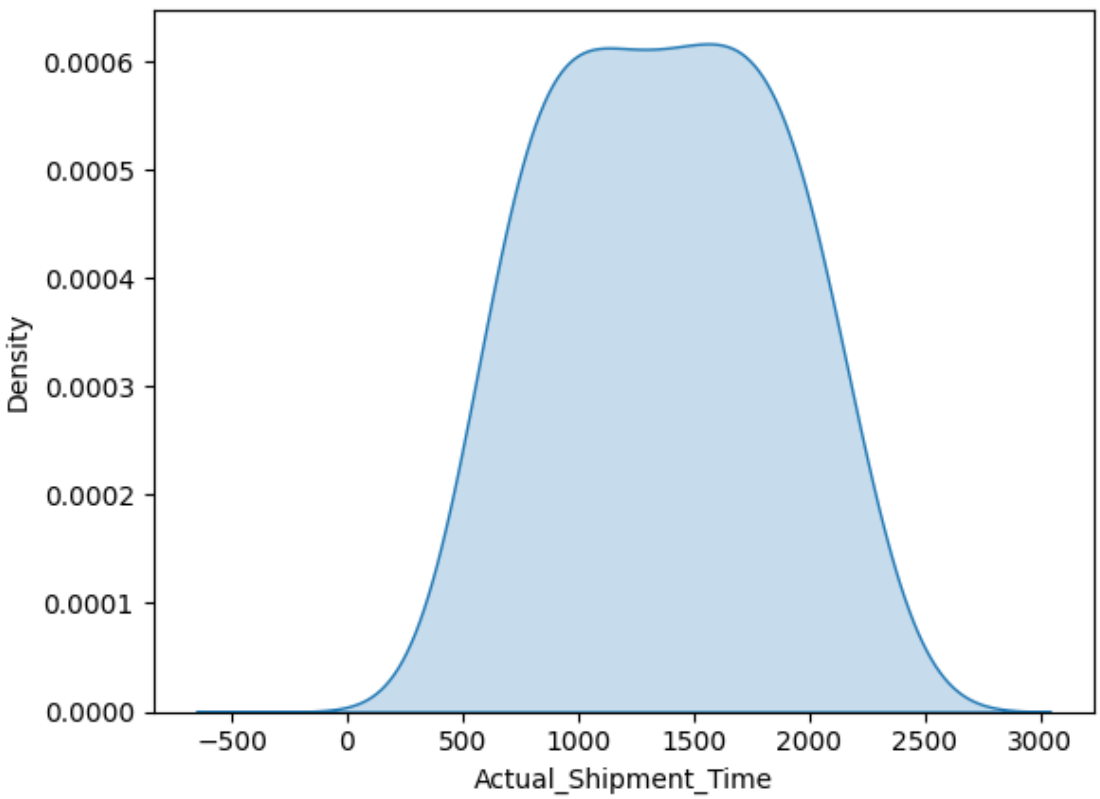
[44]:
```
<Axes: xlabel='Actual_Shipment_Time', ylabel='Density'>
```



[45]:
```
sns.kdeplot(project.Actual_Shipment_Time, bw = 0.5 , fill = True)
```

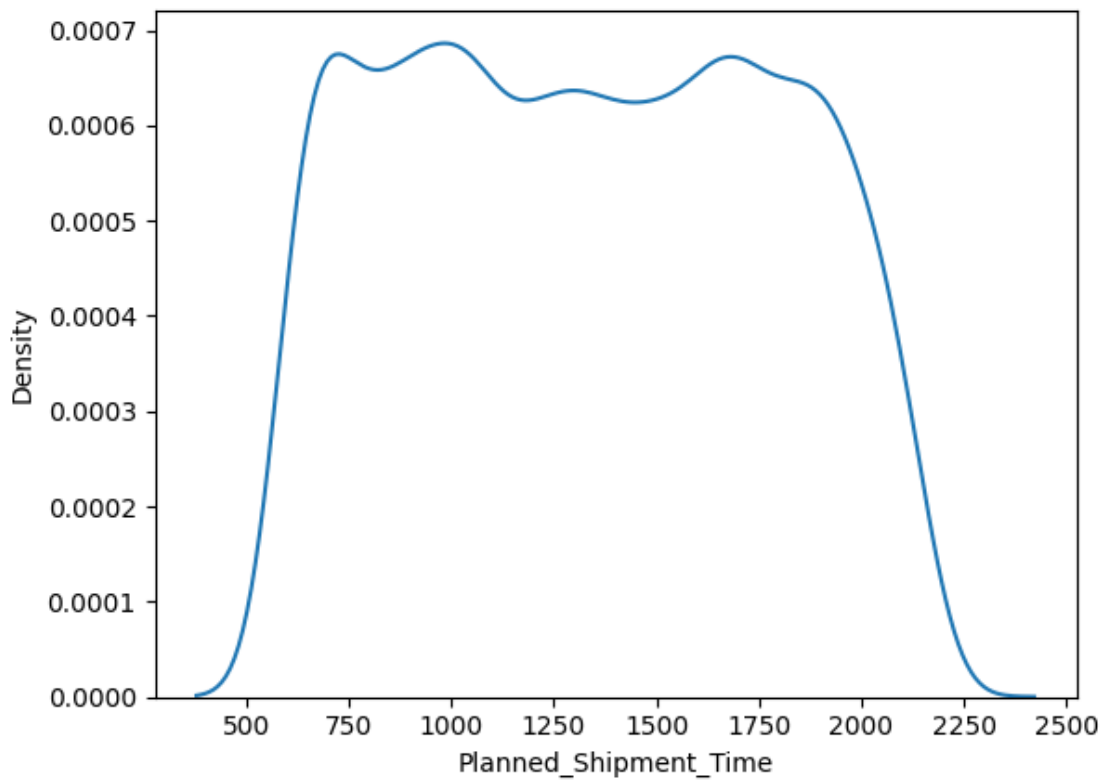
```
<ipython-input-45-5a6e0bd28785>:1: UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and `bw_adjust`.
Setting `bw_method=0.5`, but please see the docs for the new parameters
and update your code. This will become an error in seaborn v0.13.0.

  sns.kdeplot(project.Actual_Shipment_Time, bw = 0.5 , fill = True)
```

[45]:
```
<Axes: xlabel='Actual_Shipment_Time', ylabel='Density'>
```



[46]:
```
sns.kdeplot(project.Planned_Shipment_Time)
```

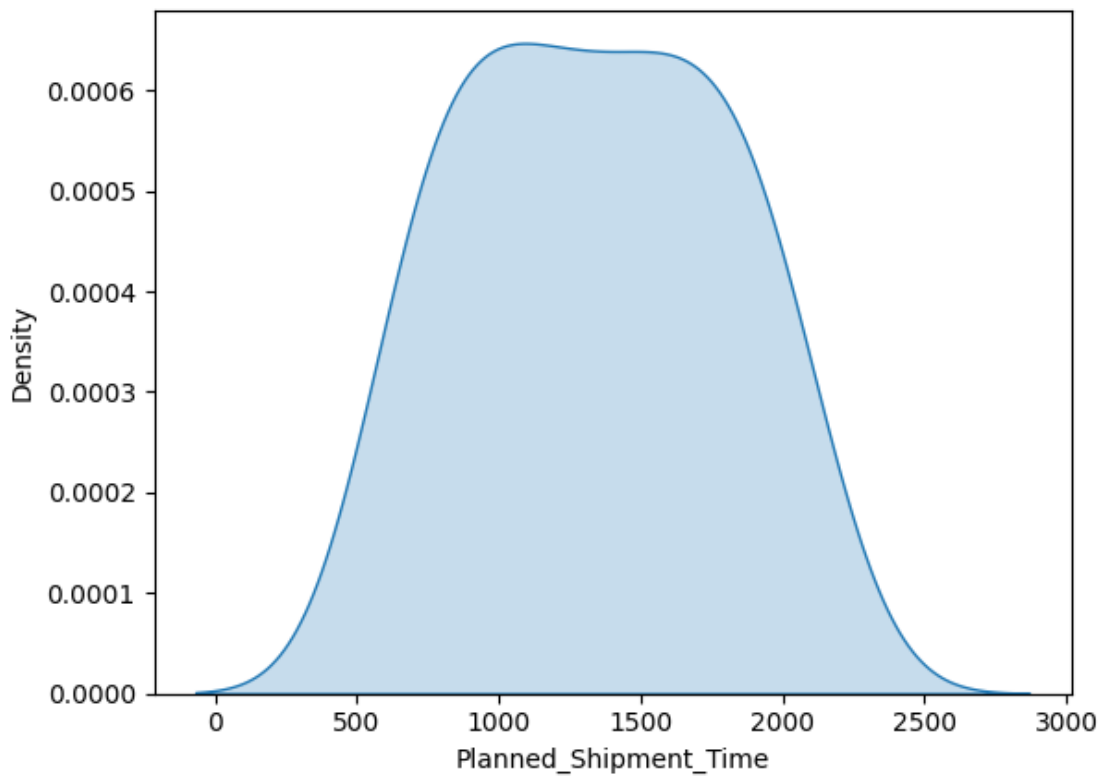
[46]: <Axes: xlabel='Planned_Shipment_Time', ylabel='Density'>



[47]:
```
sns.kdeplot(project.Planned_Shipment_Time, bw = 0.5 , fill = True)
```

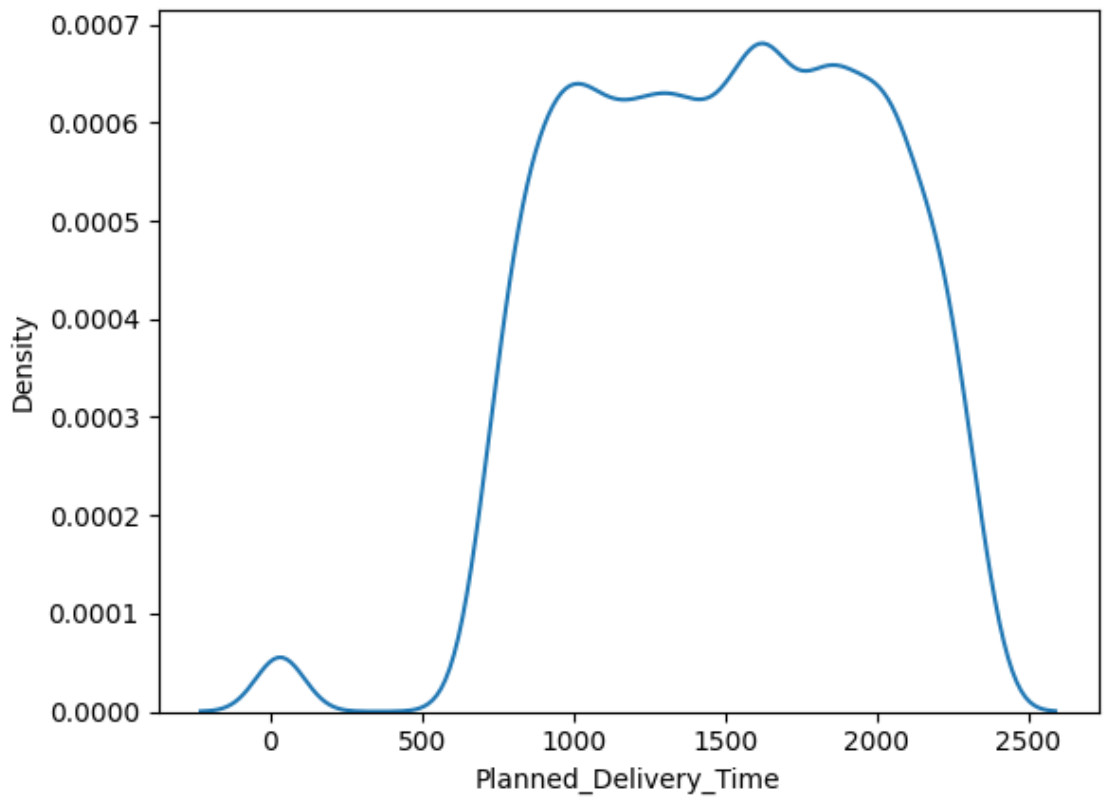
```
<ipython-input-47-14589e1aab3d>:1: UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and `bw_adjust`.
Setting `bw_method=0.5`, but please see the docs for the new parameters
and update your code. This will become an error in seaborn v0.13.0.

  sns.kdeplot(project.Planned_Shipment_Time, bw = 0.5 , fill = True)
```

[47]:
```
<Axes: xlabel='Planned_Shipment_Time', ylabel='Density'>
```



[48]:
```
sns.kdeplot(project.Planned_Delivery_Time)
```

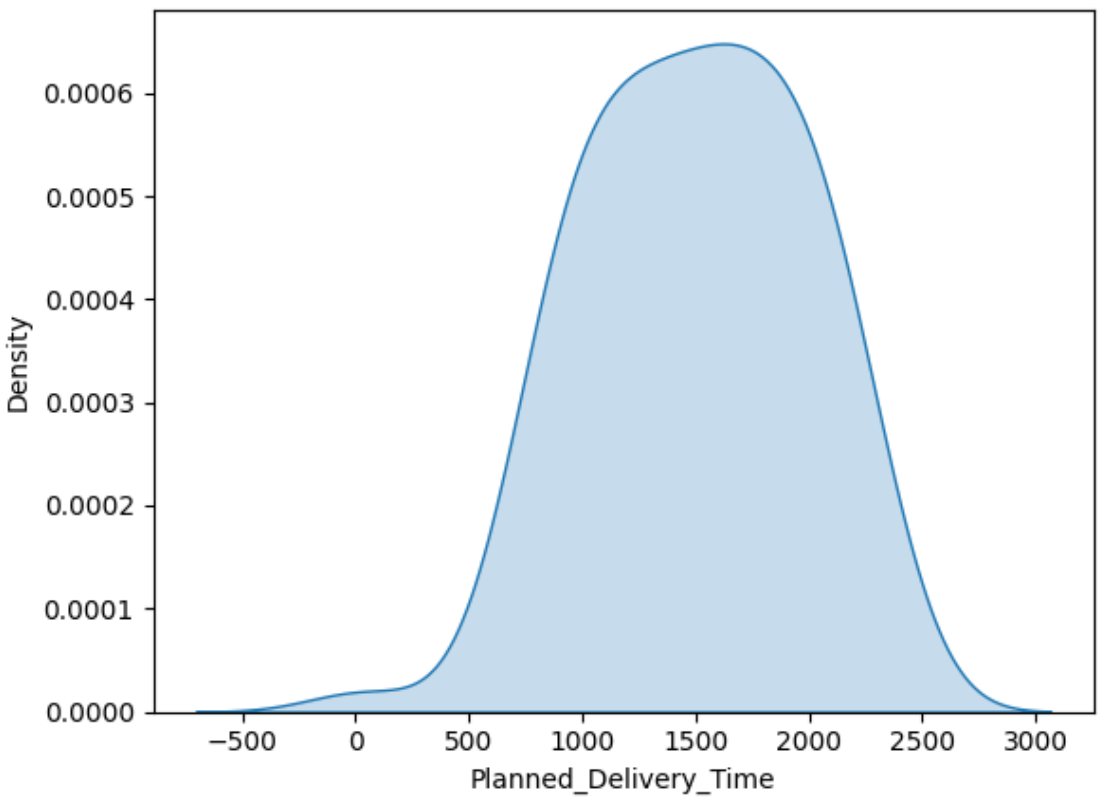
[48]: <Axes: xlabel='Planned_Delivery_Time', ylabel='Density'>



[49]:
```
sns.kdeplot(project.Planned_Delivery_Time, bw = 0.5 , fill = True)
```

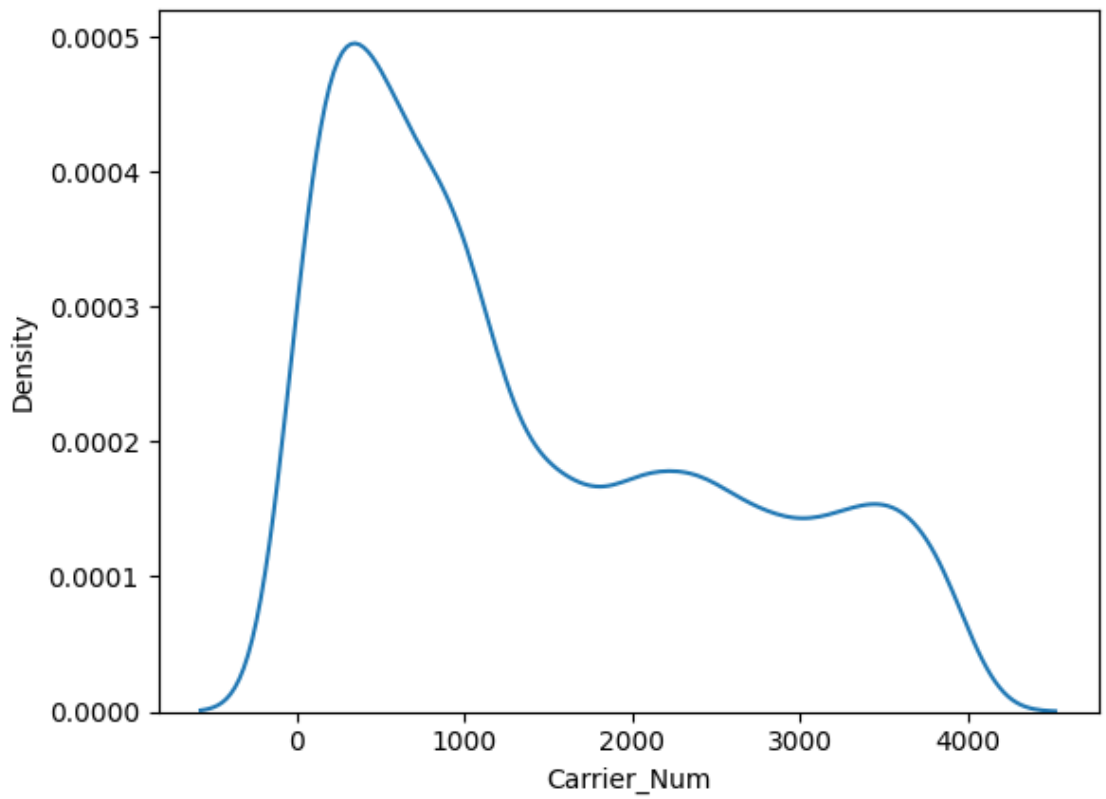
```
<ipython-input-49-87f25f9ee559>:1: UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and `bw_adjust`.
Setting `bw_method=0.5`, but please see the docs for the new parameters
and update your code. This will become an error in seaborn v0.13.0.

  sns.kdeplot(project.Planned_Delivery_Time, bw = 0.5 , fill = True)
```

[49]:
```
<Axes: xlabel='Planned_Delivery_Time', ylabel='Density'>
```



[50]:
```
sns.kdeplot(project.Carrier_Num)
```

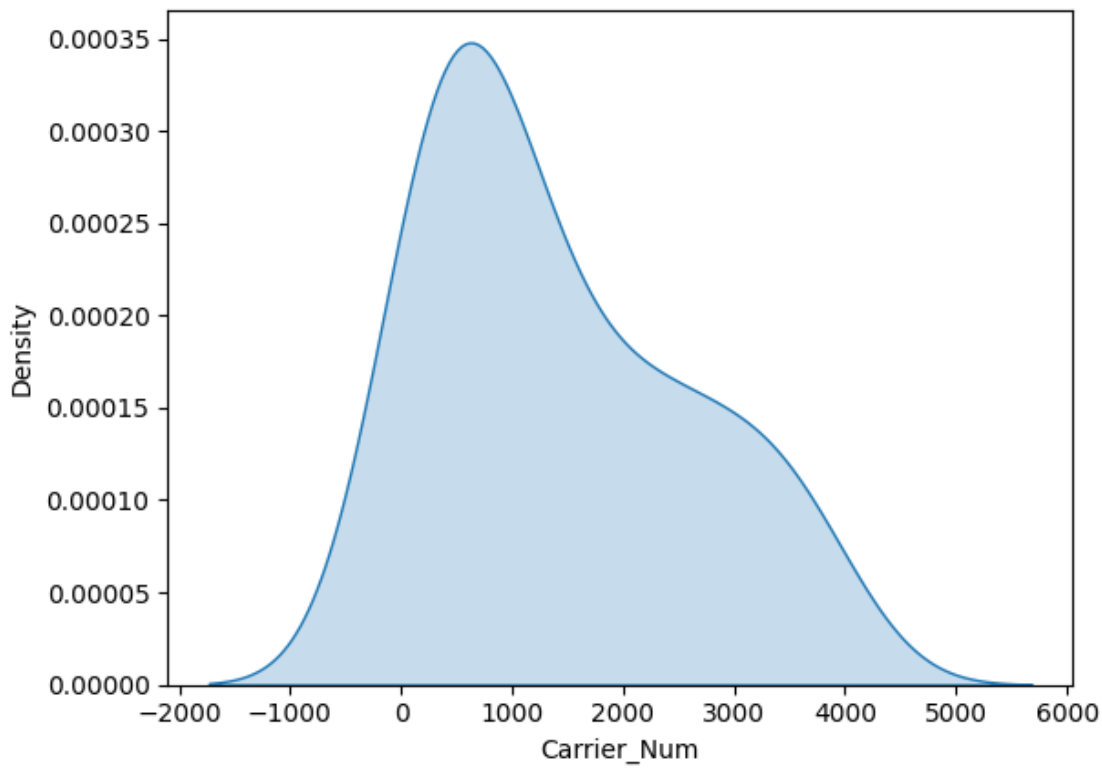
[50]: <Axes: xlabel='Carrier_Num', ylabel='Density'>



[51]:
```
sns.kdeplot(project.Carrier_Num, bw = 0.5 , fill = True)
```

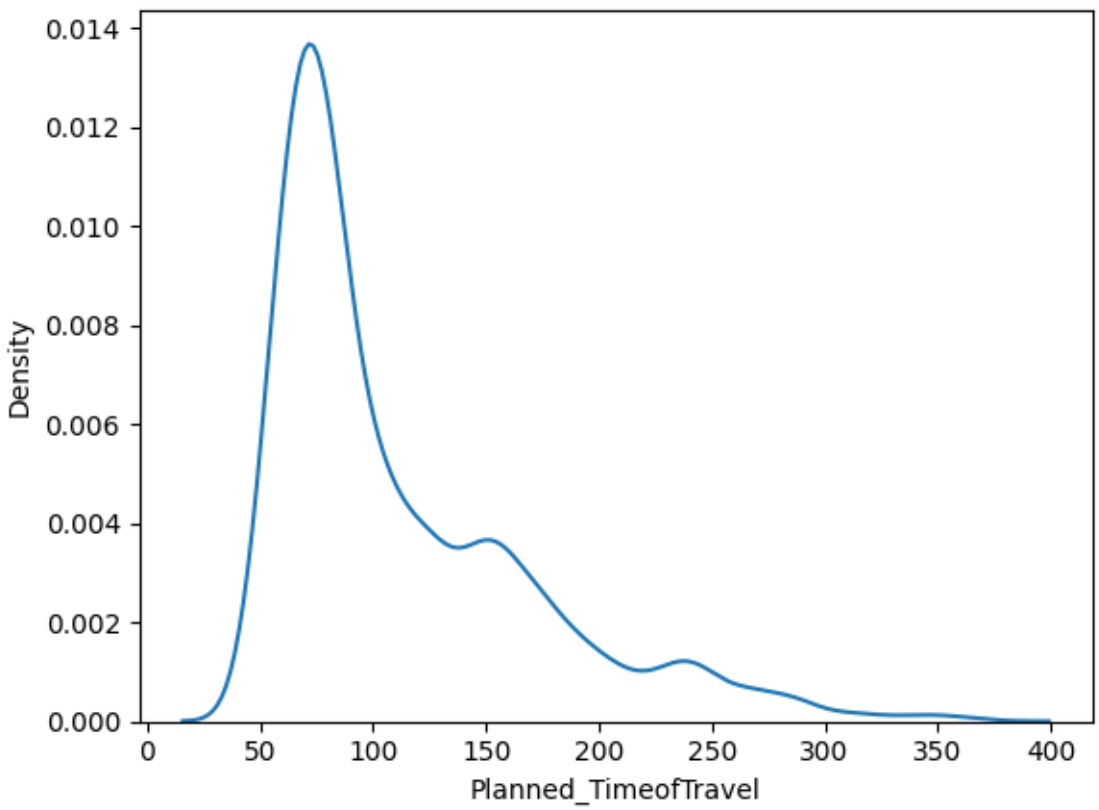
```
<ipython-input-51-e98e31b74cd8>:1: UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and `bw_adjust`.
Setting `bw_method=0.5`, but please see the docs for the new parameters
and update your code. This will become an error in seaborn v0.13.0.

  sns.kdeplot(project.Carrier_Num, bw = 0.5 , fill = True)
```

[51]:
```
<Axes: xlabel='Carrier_Num', ylabel='Density'>
```



[52]:
```
sns.kdeplot(project.Planned_TimeofTravel)
```

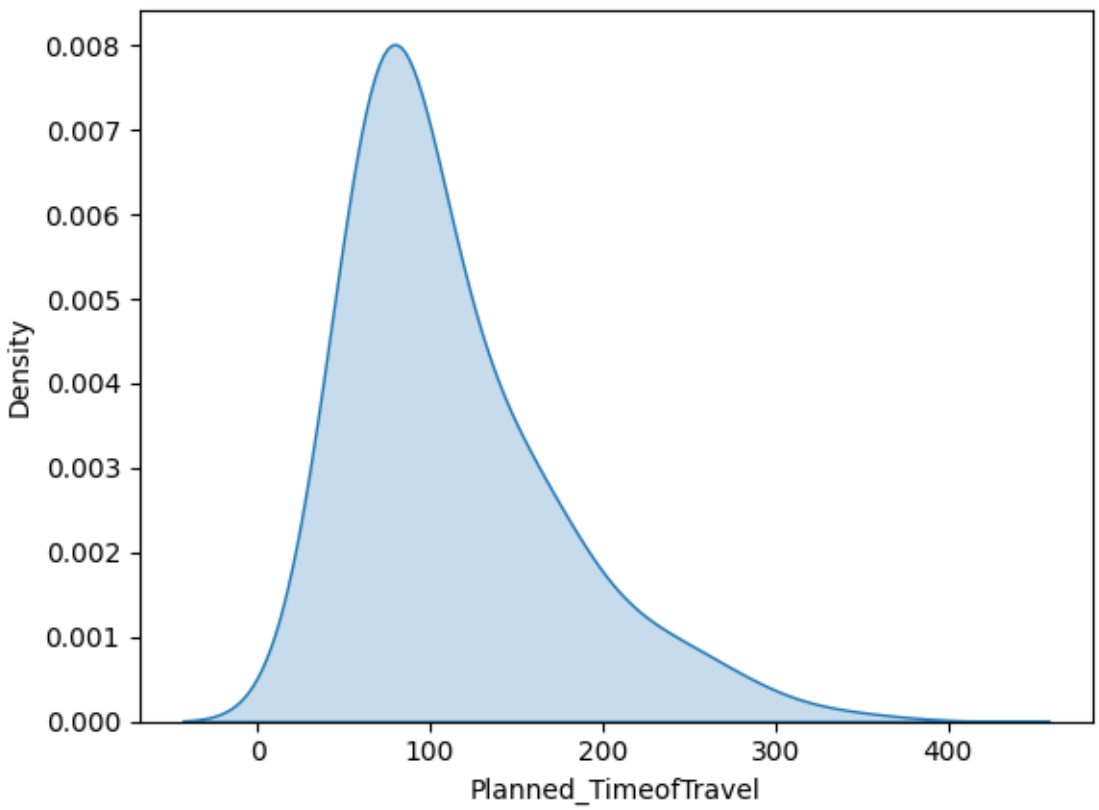
[52]: <Axes: xlabel='Planned_TimeofTravel', ylabel='Density'>



[53]:
```
sns.kdeplot(project.Planned_TimeofTravel, bw = 0.5 , fill = True)
```

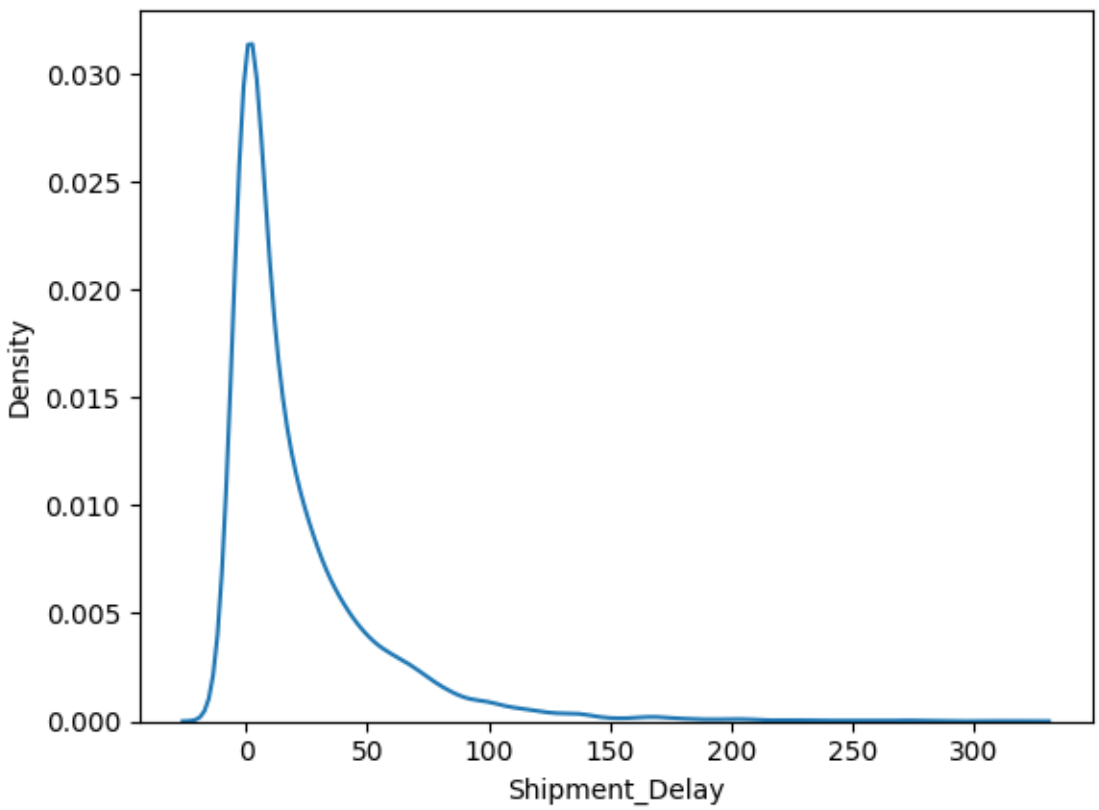
```
<ipython-input-53-80c2458b5a2e>:1: UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and `bw_adjust`.
Setting `bw_method=0.5`, but please see the docs for the new parameters
and update your code. This will become an error in seaborn v0.13.0.

  sns.kdeplot(project.Planned_TimeofTravel, bw = 0.5 , fill = True)
```

[53]:
```
<Axes: xlabel='Planned_TimeofTravel', ylabel='Density'>
```



[54]:
```
sns.kdeplot(project.Shipment_Delay)
```

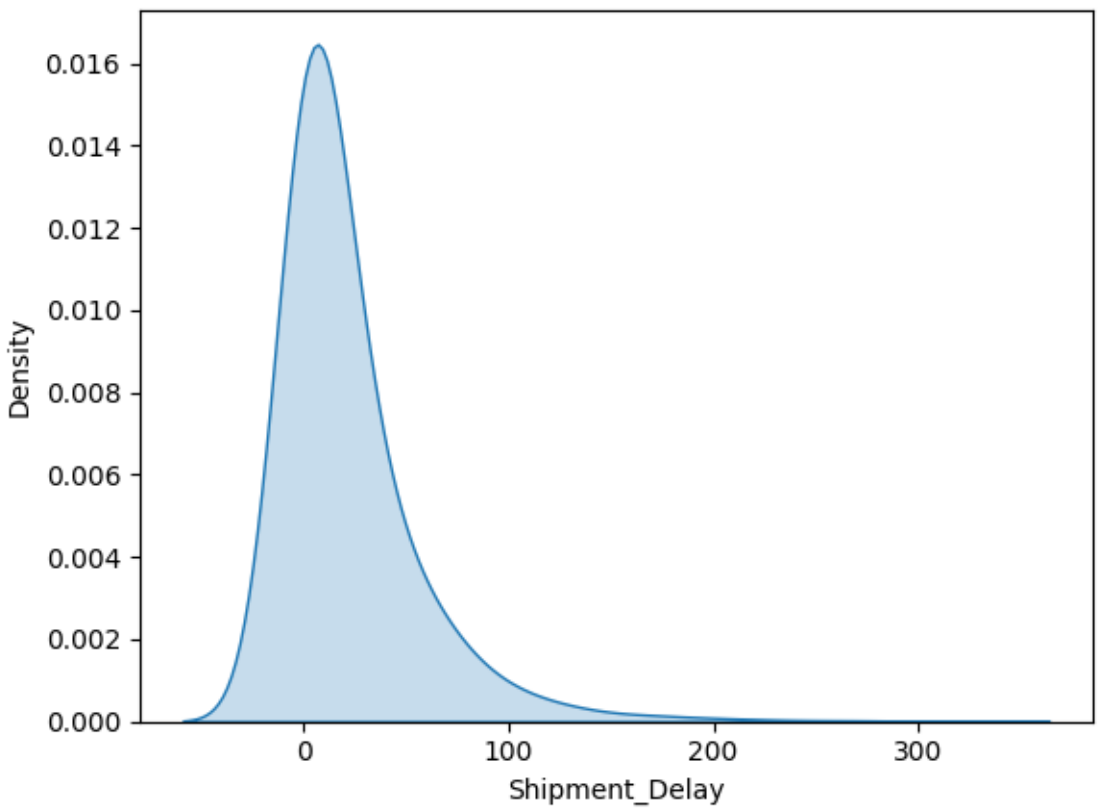
[54]: <Axes: xlabel='Shipment_Delay', ylabel='Density'>



[55]:
```
sns.kdeplot(project.Shipment_Delay, bw = 0.5 , fill = True)
```

```
<ipython-input-55-e8dac5fe4c2c>:1: UserWarning:

The `bw` parameter is deprecated in favor of `bw_method` and `bw_adjust`.
Setting `bw_method=0.5`, but please see the docs for the new parameters
and update your code. This will become an error in seaborn v0.13.0.

  sns.kdeplot(project.Shipment_Delay, bw = 0.5 , fill = True)
```

[55]:
```
<Axes: xlabel='Shipment_Delay', ylabel='Density'>
```



9 Descriptive Statistics

10 describe function will return descriptive statistics including the

11 central tendency, dispersion and shape of a dataset's distribution.

[56]:
```
project.describe()
```

[56]:
```
          Year   Month   DayofMonth    DayOfWeek  Actual_Shipment_Time  \
count  7999.0  7999.0  7999.000000  7999.000000           7860.000000
mean   2008.0     1.0     3.978372     4.978372           1370.203435
std       0.0     0.0     0.754851     0.754851            468.043601
min    2008.0     1.0     3.000000     4.000000             47.000000
25%    2008.0     1.0     3.000000     4.000000            947.000000
50%    2008.0     1.0     4.000000     5.000000           1356.000000
75%    2008.0     1.0     5.000000     6.000000           1754.000000
max    2008.0     1.0     5.000000     6.000000           2341.000000
```

```
       Planned_Shipment_Time  Planned_Delivery_Time  Carrier_Num  \
count            7999.000000            7999.000000  7999.000000
mean             1335.317540            1498.255407  1422.283285
std               446.151375             473.788941  1155.282332
min               600.000000               5.000000     1.000000
25%               940.000000            1120.000000   445.500000
50%              1330.000000            1520.000000  1023.000000
75%              1720.000000            1905.000000  2358.500000
max              2200.000000            2355.000000  3949.000000

       Planned_TimeofTravel  Shipment_Delay     Distance  Delivery_Status
count           7999.000000     7860.000000  7999.000000      7860.000000
mean             112.899112       21.389186   637.847231         0.397074
std               58.766090       32.563453   451.952916         0.489323
min               45.000000      -10.000000   133.000000         0.000000
25%               70.000000        1.000000   325.000000         0.000000
50%               90.000000        9.000000   447.000000         0.000000
75%              145.000000       30.000000   861.000000         1.000000
max              370.000000      315.000000  2363.000000         1.000000
```

12 Bivariate visualization

13 Scatter plot

[58]:
```
import pandas as pd
```

[57]:
```
import matplotlib.pyplot as plt
```

[59]:
```
project = pd.read_csv(r"/content/Datasets.csv")
```

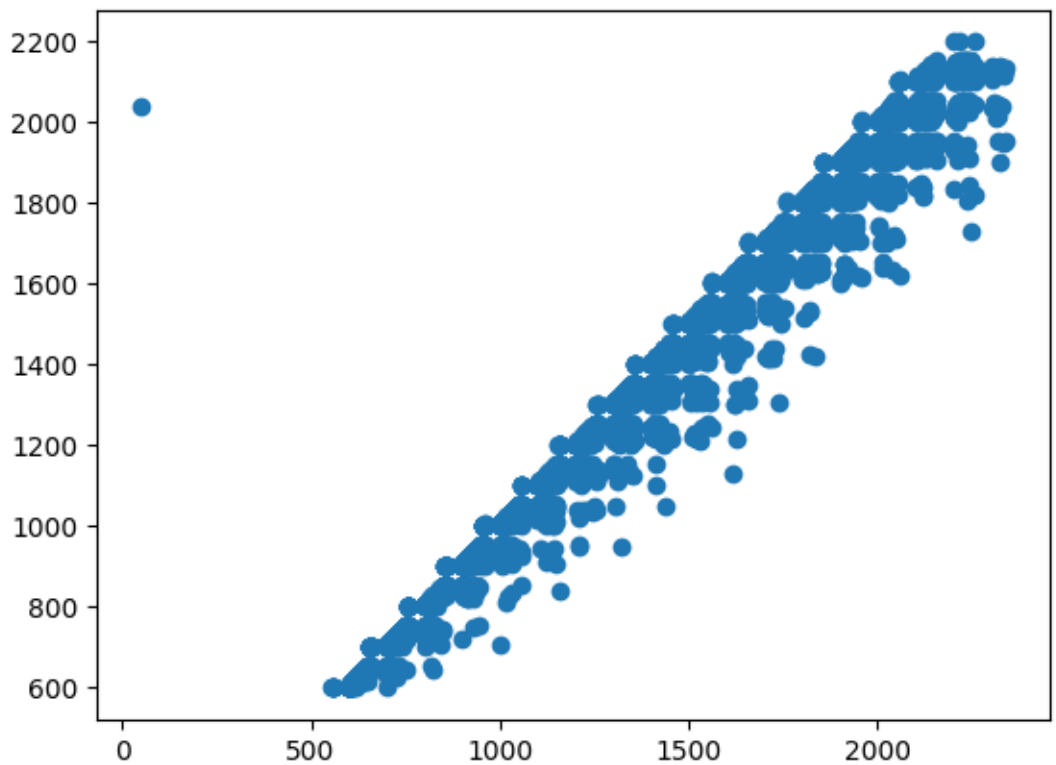
[60]:
```
project.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7999 entries, 0 to 7998
Data columns (total 15 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Year                   7999 non-null   int64
 1   Month                  7999 non-null   int64
 2   DayofMonth             7999 non-null   int64
 3   DayOfWeek              7999 non-null   int64
 4   Actual_Shipment_Time   7860 non-null   float64
 5   Planned_Shipment_Time  7999 non-null   int64
 6   Planned_Delivery_Time  7999 non-null   int64
 7   Carrier_Name           7999 non-null   object
 8   Carrier_Num            7999 non-null   int64
```

```
 9   Planned_TimeofTravel   7999 non-null   int64
 10  Shipment_Delay         7860 non-null   float64
 11  Source                 7999 non-null   object
 12  Destination            7999 non-null   object
 13  Distance               7999 non-null   int64
 14  Delivery_Status        7860 non-null   float64
dtypes: float64(3), int64(9), object(3)
memory usage: 937.5+ KB
```

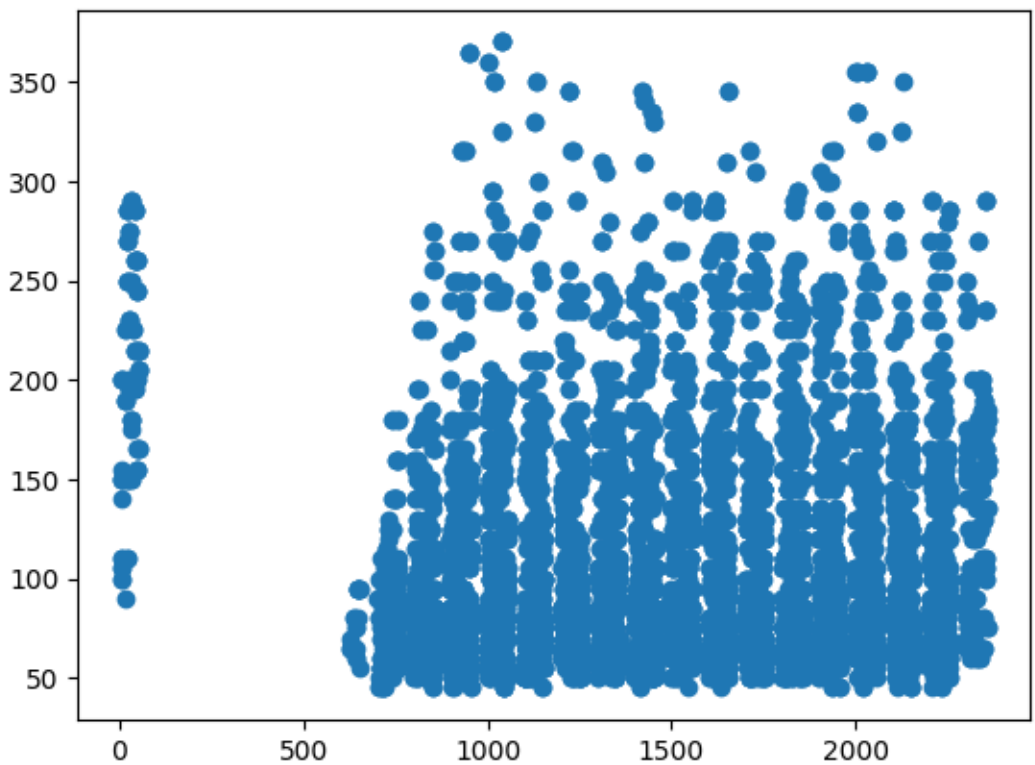
[61]:
```
plt.scatter(x = project['Actual_Shipment_Time'], y =␣
↪project['Planned_Shipment_Time'])
```

[61]: <matplotlib.collections.PathCollection at 0x7fe0530b2710>



[62]:
```
plt.scatter(x = project['Planned_Delivery_Time'], y =␣
↪project['Planned_TimeofTravel'])
```

[62]: <matplotlib.collections.PathCollection at 0x7fe052f33f70>



[63]:
```
plt.scatter(x = project['Source'], y = project['Destination'], color = 'green')
```

[63]: <matplotlib.collections.PathCollection at 0x7fe052faefe0>