Chapter 2 - Statistical Learning

September 14, 2023

[97]:
```
!pip install ISLP
```

```
Requirement already satisfied: ISLP in
/Users/barnana/anaconda3/lib/python3.10/site-packages (0.3.16)
Requirement already satisfied: pandas>=0.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from ISLP) (1.5.3)
Requirement already satisfied: numpy>=0.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from ISLP) (1.25.1)
Requirement already satisfied: pygam>=0.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from ISLP) (0.9.0)
Requirement already satisfied: scipy>=0.9 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from ISLP) (1.11.1)
Requirement already satisfied: jupyter>=0.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from ISLP) (1.0.0)
Requirement already satisfied: statsmodels>=0.13 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from ISLP) (0.13.5)
Requirement already satisfied: lifelines>=0.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from ISLP) (0.27.7)
Requirement already satisfied: scikit-learn>=1.2 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from ISLP) (1.3.0)
Requirement already satisfied: joblib>=0.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from ISLP) (1.1.1)
Requirement already satisfied: matplotlib>=3.3.3 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from ISLP) (3.7.0)
Requirement already satisfied: lxml>=0.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from ISLP) (4.9.1)
Requirement already satisfied: ipykernel in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from jupyter>=0.0->ISLP)
(6.19.2)
Requirement already satisfied: nbconvert in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from jupyter>=0.0->ISLP)
(6.5.4)
Requirement already satisfied: qtconsole in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from jupyter>=0.0->ISLP)
(5.4.0)
Requirement already satisfied: jupyter-console in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from jupyter>=0.0->ISLP)
(6.6.3)
```

```
Requirement already satisfied: ipywidgets in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from jupyter>=0.0->ISLP)
(8.0.7)
Requirement already satisfied: notebook in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from jupyter>=0.0->ISLP)
(6.5.2)
Requirement already satisfied: autograd>=1.5 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
lifelines>=0.0->ISLP) (1.6.2)
Requirement already satisfied: formulaic>=0.2.2 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
lifelines>=0.0->ISLP) (0.6.4)
Requirement already satisfied: autograd-gamma>=0.3 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
lifelines>=0.0->ISLP) (0.5.0)
Requirement already satisfied: python-dateutil>=2.7 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
matplotlib>=3.3.3->ISLP) (2.8.2)
Requirement already satisfied: cycler>=0.10 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
matplotlib>=3.3.3->ISLP) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
matplotlib>=3.3.3->ISLP) (9.4.0)
Requirement already satisfied: packaging>=20.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
matplotlib>=3.3.3->ISLP) (22.0)
Requirement already satisfied: contourpy>=1.0.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
matplotlib>=3.3.3->ISLP) (1.0.5)
Requirement already satisfied: kiwisolver>=1.0.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
matplotlib>=3.3.3->ISLP) (1.4.4)
Requirement already satisfied: pyparsing>=2.3.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
matplotlib>=3.3.3->ISLP) (3.0.9)
Requirement already satisfied: fonttools>=4.22.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
matplotlib>=3.3.3->ISLP) (4.25.0)
Requirement already satisfied: pytz>=2020.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from pandas>=0.0->ISLP)
(2022.7)
Requirement already satisfied: progressbar2<5.0.0,>=4.2.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from pygam>=0.0->ISLP)
(4.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from scikit-
learn>=1.2->ISLP) (3.2.0)
```

```
Requirement already satisfied: patsy>=0.5.2 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
statsmodels>=0.13->ISLP) (0.5.3)
Requirement already satisfied: future>=0.15.2 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
autograd>=1.5->lifelines>=0.0->ISLP) (0.18.3)
Requirement already satisfied: wrapt>=1.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
formulaic>=0.2.2->lifelines>=0.0->ISLP) (1.14.1)
Requirement already satisfied: typing-extensions>=4.2.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
formulaic>=0.2.2->lifelines>=0.0->ISLP) (4.4.0)
Requirement already satisfied: astor>=0.8 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
formulaic>=0.2.2->lifelines>=0.0->ISLP) (0.8.1)
Requirement already satisfied: interface-meta>=1.2.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
formulaic>=0.2.2->lifelines>=0.0->ISLP) (1.3.0)
Requirement already satisfied: six in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
patsy>=0.5.2->statsmodels>=0.13->ISLP) (1.16.0)
Requirement already satisfied: python-utils>=3.0.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
progressbar2<5.0.0,>=4.2.0->pygam>=0.0->ISLP) (3.7.0)
Requirement already satisfied: comm>=0.1.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipykernel->jupyter>=0.0->ISLP) (0.1.2)
Requirement already satisfied: ipython>=7.23.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipykernel->jupyter>=0.0->ISLP) (8.10.0)
Requirement already satisfied: pyzmq>=17 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipykernel->jupyter>=0.0->ISLP) (23.2.0)
Requirement already satisfied: psutil in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipykernel->jupyter>=0.0->ISLP) (5.9.0)
Requirement already satisfied: traitlets>=5.4.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipykernel->jupyter>=0.0->ISLP) (5.7.1)
Requirement already satisfied: tornado>=6.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipykernel->jupyter>=0.0->ISLP) (6.1)
Requirement already satisfied: matplotlib-inline>=0.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipykernel->jupyter>=0.0->ISLP) (0.1.6)
Requirement already satisfied: nest-asyncio in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipykernel->jupyter>=0.0->ISLP) (1.5.6)
```

```
Requirement already satisfied: jupyter-client>=6.1.12 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipykernel->jupyter>=0.0->ISLP) (7.3.4)
Requirement already satisfied: debugpy>=1.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipykernel->jupyter>=0.0->ISLP) (1.5.1)
Requirement already satisfied: appnope in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipykernel->jupyter>=0.0->ISLP) (0.1.2)
Requirement already satisfied: widgetsnbextension~=4.0.7 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipywidgets->jupyter>=0.0->ISLP) (4.0.8)
Requirement already satisfied: jupyterlab-widgets~=3.0.7 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipywidgets->jupyter>=0.0->ISLP) (3.0.8)
Requirement already satisfied: prompt-toolkit>=3.0.30 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from jupyter-
console->jupyter>=0.0->ISLP) (3.0.36)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from jupyter-
console->jupyter>=0.0->ISLP) (5.2.0)
Requirement already satisfied: pygments in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from jupyter-
console->jupyter>=0.0->ISLP) (2.11.2)
Requirement already satisfied: pandocfilters>=1.4.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbconvert->jupyter>=0.0->ISLP) (1.5.0)
Requirement already satisfied: tinycss2 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbconvert->jupyter>=0.0->ISLP) (1.2.1)
Requirement already satisfied: entrypoints>=0.2.2 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbconvert->jupyter>=0.0->ISLP) (0.4)
Requirement already satisfied: MarkupSafe>=2.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbconvert->jupyter>=0.0->ISLP) (2.1.1)
Requirement already satisfied: nbformat>=5.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbconvert->jupyter>=0.0->ISLP) (5.7.0)
Requirement already satisfied: bleach in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbconvert->jupyter>=0.0->ISLP) (4.1.0)
Requirement already satisfied: jinja2>=3.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbconvert->jupyter>=0.0->ISLP) (3.1.2)
Requirement already satisfied: beautifulsoup4 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbconvert->jupyter>=0.0->ISLP) (4.11.1)
```

```
Requirement already satisfied: mistune<2,>=0.8.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbconvert->jupyter>=0.0->ISLP) (0.8.4)
Requirement already satisfied: nbclient>=0.5.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbconvert->jupyter>=0.0->ISLP) (0.5.13)
Requirement already satisfied: jupyterlab-pygments in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbconvert->jupyter>=0.0->ISLP) (0.1.2)
Requirement already satisfied: defusedxml in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbconvert->jupyter>=0.0->ISLP) (0.7.1)
Requirement already satisfied: ipython-genutils in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
notebook->jupyter>=0.0->ISLP) (0.2.0)
Requirement already satisfied: prometheus-client in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
notebook->jupyter>=0.0->ISLP) (0.14.1)
Requirement already satisfied: Send2Trash>=1.8.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
notebook->jupyter>=0.0->ISLP) (1.8.0)
Requirement already satisfied: terminado>=0.8.3 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
notebook->jupyter>=0.0->ISLP) (0.17.1)
Requirement already satisfied: argon2-cffi in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
notebook->jupyter>=0.0->ISLP) (21.3.0)
Requirement already satisfied: nbclassic>=0.4.7 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
notebook->jupyter>=0.0->ISLP) (0.5.2)
Requirement already satisfied: qtpy>=2.0.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
qtconsole->jupyter>=0.0->ISLP) (2.2.0)
Requirement already satisfied: decorator in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipython>=7.23.1->ipykernel->jupyter>=0.0->ISLP) (5.1.1)
Requirement already satisfied: jedi>=0.16 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipython>=7.23.1->ipykernel->jupyter>=0.0->ISLP) (0.18.1)
Requirement already satisfied: stack-data in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipython>=7.23.1->ipykernel->jupyter>=0.0->ISLP) (0.2.0)
Requirement already satisfied: pexpect>4.3 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipython>=7.23.1->ipykernel->jupyter>=0.0->ISLP) (4.8.0)
Requirement already satisfied: pickleshare in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipython>=7.23.1->ipykernel->jupyter>=0.0->ISLP) (0.7.5)
```

```
Requirement already satisfied: backcall in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
ipython>=7.23.1->ipykernel->jupyter>=0.0->ISLP) (0.2.0)
Requirement already satisfied: platformdirs>=2.5 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from jupyter-
core!=5.0.*,>=4.12->jupyter-console->jupyter>=0.0->ISLP) (2.5.2)
Requirement already satisfied: notebook-shim>=0.1.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbclassic>=0.4.7->notebook->jupyter>=0.0->ISLP) (0.2.2)
Requirement already satisfied: jupyter-server>=1.8 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbclassic>=0.4.7->notebook->jupyter>=0.0->ISLP) (1.23.4)
Requirement already satisfied: fastjsonschema in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbformat>=5.1->nbconvert->jupyter>=0.0->ISLP) (2.16.2)
Requirement already satisfied: jsonschema>=2.6 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
nbformat>=5.1->nbconvert->jupyter>=0.0->ISLP) (4.17.3)
Requirement already satisfied: wcwidth in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from prompt-
toolkit>=3.0.30->jupyter-console->jupyter>=0.0->ISLP) (0.2.5)
Requirement already satisfied: ptyprocess in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
terminado>=0.8.3->notebook->jupyter>=0.0->ISLP) (0.7.0)
Requirement already satisfied: argon2-cffi-bindings in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
argon2-cffi->notebook->jupyter>=0.0->ISLP) (21.2.0)
Requirement already satisfied: soupsieve>1.2 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
beautifulsoup4->nbconvert->jupyter>=0.0->ISLP) (2.3.2.post1)
Requirement already satisfied: webencodings in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
bleach->nbconvert->jupyter>=0.0->ISLP) (0.5.1)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
jedi>=0.16->ipython>=7.23.1->ipykernel->jupyter>=0.0->ISLP) (0.8.3)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
jsonschema>=2.6->nbformat>=5.1->nbconvert->jupyter>=0.0->ISLP) (0.18.0)
Requirement already satisfied: attrs>=17.4.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
jsonschema>=2.6->nbformat>=5.1->nbconvert->jupyter>=0.0->ISLP) (22.1.0)
Requirement already satisfied: websocket-client in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from jupyter-
server>=1.8->nbclassic>=0.4.7->notebook->jupyter>=0.0->ISLP) (0.58.0)
Requirement already satisfied: anyio<4,>=3.1.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from jupyter-
server>=1.8->nbclassic>=0.4.7->notebook->jupyter>=0.0->ISLP) (3.5.0)
```

```
Requirement already satisfied: cffi>=1.0.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from argon2-cffi-
bindings->argon2-cffi->notebook->jupyter>=0.0->ISLP) (1.15.1)
Requirement already satisfied: executing in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from stack-
data->ipython>=7.23.1->ipykernel->jupyter>=0.0->ISLP) (0.8.3)
Requirement already satisfied: pure-eval in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from stack-
data->ipython>=7.23.1->ipykernel->jupyter>=0.0->ISLP) (0.2.2)
Requirement already satisfied: asttokens in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from stack-
data->ipython>=7.23.1->ipykernel->jupyter>=0.0->ISLP) (2.0.5)
Requirement already satisfied: sniffio>=1.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
anyio<4,>=3.1.0->jupyter-
server>=1.8->nbclassic>=0.4.7->notebook->jupyter>=0.0->ISLP) (1.2.0)
Requirement already satisfied: idna>=2.8 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
anyio<4,>=3.1.0->jupyter-
server>=1.8->nbclassic>=0.4.7->notebook->jupyter>=0.0->ISLP) (3.4)
Requirement already satisfied: pycparser in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi->notebook->jupyter>=0.0->ISLP)
(2.21)
```

[98]:
```
!pip install pytorch-lightning
```

```
Requirement already satisfied: pytorch-lightning in
/Users/barnana/anaconda3/lib/python3.10/site-packages (2.0.6)
Requirement already satisfied: typing-extensions>=4.0.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from pytorch-lightning)
(4.4.0)
Requirement already satisfied: fsspec[http]>2021.06.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from pytorch-lightning)
(2022.11.0)
Requirement already satisfied: tqdm>=4.57.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from pytorch-lightning)
(4.64.1)
Requirement already satisfied: lightning-utilities>=0.7.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from pytorch-lightning)
(0.9.0)
Requirement already satisfied: PyYAML>=5.4 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from pytorch-lightning)
(6.0)
Requirement already satisfied: torch>=1.11.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from pytorch-lightning)
(1.12.1)
Requirement already satisfied: torchmetrics>=0.7.0 in
```

```
/Users/barnana/anaconda3/lib/python3.10/site-packages (from pytorch-lightning)
(1.0.1)
Requirement already satisfied: numpy>=1.17.2 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from pytorch-lightning)
(1.25.1)
Requirement already satisfied: packaging>=17.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from pytorch-lightning)
(22.0)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
fsspec[http]>2021.06.0->pytorch-lightning) (3.8.5)
Requirement already satisfied: requests in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
fsspec[http]>2021.06.0->pytorch-lightning) (2.28.1)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-lightning) (4.0.2)
Requirement already satisfied: multidict<7.0,>=4.5 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-lightning) (6.0.4)
Requirement already satisfied: aiosignal>=1.1.2 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-lightning) (1.3.1)
Requirement already satisfied: charset-normalizer<4.0,>=2.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-lightning) (2.0.4)
Requirement already satisfied: attrs>=17.3.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-lightning) (22.1.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-lightning) (1.4.0)
Requirement already satisfied: yarl<2.0,>=1.0 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>2021.06.0->pytorch-lightning) (1.9.2)
Requirement already satisfied: certifi>=2017.4.17 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
requests->fsspec[http]>2021.06.0->pytorch-lightning) (2023.5.7)
Requirement already satisfied: idna<4,>=2.5 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
requests->fsspec[http]>2021.06.0->pytorch-lightning) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/Users/barnana/anaconda3/lib/python3.10/site-packages (from
requests->fsspec[http]>2021.06.0->pytorch-lightning) (1.26.14)
```

[99]:
```python
from ISLP import load_data
import pandas as pd
```

```
import numpy as np
import seaborn as sns
import math
import matplotlib.pyplot as plt
import statsmodels.api as sm
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.model_selection import train_test_split, GridSearchCV, KFold,␣
↪cross_val_score
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import RidgeCV
from sklearn.linear_model import LassoCV
from sklearn.decomposition import PCA
from sklearn.pipeline import Pipeline
from sklearn.cross_decomposition import PLSRegression
import itertools
from sklearn.tree import plot_tree
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import BaggingRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
warnings.filterwarnings("ignore", category=DeprecationWarning)
import torch
import torch.nn as nn
import torch.nn.functional as F
from sklearn.metrics import classification_report
```

[100]:

```
# Creating a function to print output in green and bold
# ANSI escape code for green color and bold font
GREEN_BOLD = '\033[1;32m'

# ANSI escape code to reset colors and font style
RESET = '\033[0m'

def print_green_bold(*args):
    text = ' '.join(str(arg) for arg in args)
    print(GREEN_BOLD + text + RESET)
```

1 Chapter 2

1.1 Question 10

This exercise involves the Boston housing data set. (a) To begin, load in the Boston data set, which is part of the ISLP library.

[101]:
```python
boston_c2q10 = load_data('Boston')
boston_c2q10.head(20)
```

[101]:
```
       crim    zn  indus  chas    nox     rm    age     dis  rad  tax  \
0   0.00632  18.0   2.31     0  0.538  6.575   65.2  4.0900    1  296
1   0.02731   0.0   7.07     0  0.469  6.421   78.9  4.9671    2  242
2   0.02729   0.0   7.07     0  0.469  7.185   61.1  4.9671    2  242
3   0.03237   0.0   2.18     0  0.458  6.998   45.8  6.0622    3  222
4   0.06905   0.0   2.18     0  0.458  7.147   54.2  6.0622    3  222
5   0.02985   0.0   2.18     0  0.458  6.430   58.7  6.0622    3  222
6   0.08829  12.5   7.87     0  0.524  6.012   66.6  5.5605    5  311
7   0.14455  12.5   7.87     0  0.524  6.172   96.1  5.9505    5  311
8   0.21124  12.5   7.87     0  0.524  5.631  100.0  6.0821    5  311
9   0.17004  12.5   7.87     0  0.524  6.004   85.9  6.5921    5  311
10  0.22489  12.5   7.87     0  0.524  6.377   94.3  6.3467    5  311
11  0.11747  12.5   7.87     0  0.524  6.009   82.9  6.2267    5  311
12  0.09378  12.5   7.87     0  0.524  5.889   39.0  5.4509    5  311
13  0.62976   0.0   8.14     0  0.538  5.949   61.8  4.7075    4  307
14  0.63796   0.0   8.14     0  0.538  6.096   84.5  4.4619    4  307
15  0.62739   0.0   8.14     0  0.538  5.834   56.5  4.4986    4  307
16  1.05393   0.0   8.14     0  0.538  5.935   29.3  4.4986    4  307
17  0.78420   0.0   8.14     0  0.538  5.990   81.7  4.2579    4  307
18  0.80271   0.0   8.14     0  0.538  5.456   36.6  3.7965    4  307
19  0.72580   0.0   8.14     0  0.538  5.727   69.5  3.7965    4  307

    ptratio  lstat  medv
0      15.3   4.98  24.0
1      17.8   9.14  21.6
2      17.8   4.03  34.7
3      18.7   2.94  33.4
4      18.7   5.33  36.2
5      18.7   5.21  28.7
6      15.2  12.43  22.9
7      15.2  19.15  27.1
8      15.2  29.93  16.5
9      15.2  17.10  18.9
10     15.2  20.45  15.0
11     15.2  13.27  18.9
12     15.2  15.71  21.7
13     21.0   8.26  20.4
14     21.0  10.26  18.2
```

15	21.0	8.47	19.9
16	21.0	6.58	23.1
17	21.0	14.67	17.5
18	21.0	11.69	20.2
19	21.0	11.28	18.2

(b) How many rows are in this data set? How many columns? What do the rows and columns represent?

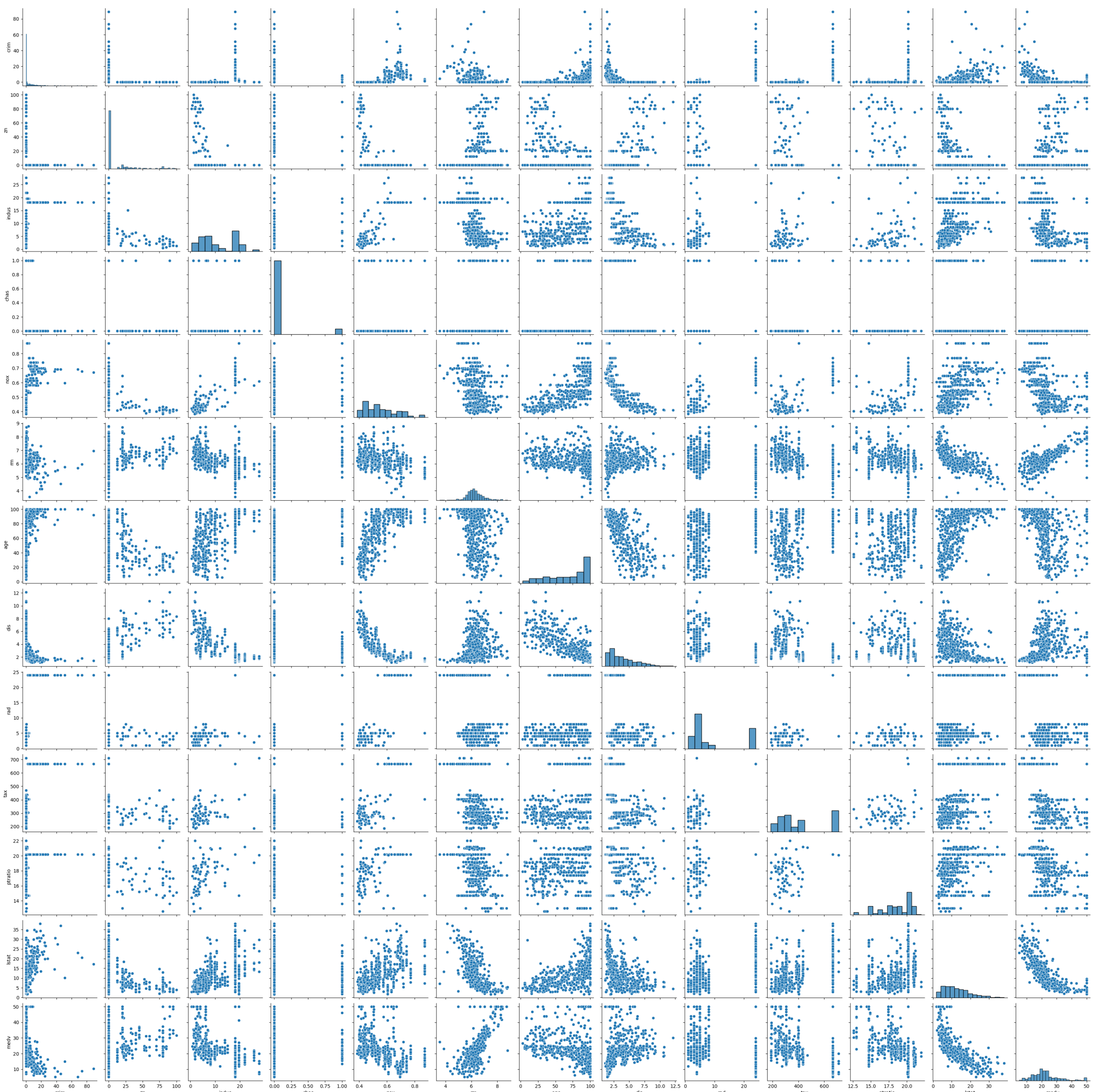
[102]:
```
##Number of rows and columns
r,c=boston_c2q10.shape
print_green_bold("The number of rows :",r)
print_green_bold("The number of columns :",c)
```

The number of rows : 506
The number of columns : 13

For a more detailed description of each variable, we can use DESCR but this can only be used on dataset objects from scikitlearn. The Boston dataset has been removed from scikitlearn so we can no longer use DESCR to get variable descriptions for it. For the sake of clarity, noting down the variable descriptions : crim: Represents the per capita crime rate by town. zn: Represents the proportion of residential land zoned for lots over 25,000 sq.ft. indus: Represents the proportion of non-retail business acres per town. chas: Represents whether the property is located along the Charles River (1 if it does, 0 if it doesn't). nox: Represents the nitric oxides concentration (parts per 10 million). rm: Represents the average number of rooms per dwelling. age: Represents the proportion of owner-occupied units built prior to 1940. dis: Represents the weighted distances to five Boston employment centers. rad: Represents the index of accessibility to radial highways. tax: Represents the full-value property tax rate per $10,000. ptratio: Represents the pupil-teacher ratio by town. lstat: Represents the percentage of lower status of the population. medv: Represents the median value of owner-occupied homes in $1000s.

(c) Make some pairwise scatterplots of the predictors (columns) in this data set. Describe your findings.

[103]:
```
sns.pairplot(boston_c2q10)
plt.show()
```



It appears that certain pairs of variables such as indus and nox, indus and tax, age and nox, etc. seem to be highly correlated. In order to confirm this, we look at the correlation matrix and filter for variable pairs where the correlation coefficient is high i.e. greater than 0.5. We've taken a threshold of 0.5 here but this number can be higher or lower depending on the context of the problem.

[104]:
```
# The correlation matrix
corr_matrix_c2q10=boston_c2q10.corr()
# Creating a dictionary with key as variable pairs and value as their␣
↪correlation
corr_dict_c2q10 = {(col1, col2): round(corr_matrix_c2q10.loc[col1, col2],1)
                    for col1 in corr_matrix_c2q10.columns
                    for col2 in corr_matrix_c2q10.columns if col1 < col2}
```

```python
# Filtering only pairs with correlation greater than 0.5
high_corr_dict_c2q10  = {key: value for key, value in corr_dict_c2q10.items()
 ↪if value > 0.5}

# Print the filtered dictionary
print_green_bold("Highly correlated variable pairs in the Boston dataset along
 ↪with their correlation coefficients :\n")
for key, value in high_corr_dict_c2q10.items():
    print_green_bold(f"{key}: {value}\n")
```

Highly correlated variable pairs in the Boston dataset along with their correlation coefficients :

('crim', 'rad'): 0.6

('crim', 'tax'): 0.6

('indus', 'nox'): 0.8

('indus', 'rad'): 0.6

('indus', 'tax'): 0.7

('indus', 'lstat'): 0.6

('nox', 'rad'): 0.6

('nox', 'tax'): 0.7

('age', 'indus'): 0.6

('age', 'nox'): 0.7

('age', 'lstat'): 0.6

('dis', 'zn'): 0.7

```
('rad', 'tax'): 0.9

('lstat', 'nox'): 0.6

('medv', 'rm'): 0.7
```

(d) Are any of the predictors associated with per capita crime rate? If so, explain the relationship.

[105]:

```python
# Calculate the correlation matrix between crim and all predictors
# Note that we can pull all the elements with crim from the overall correlation␣
↪matrix, but we're creating a new one
corr_mat_crim_c2q10 = boston_c2q10.corr()['crim']
corr_mat_crim_c2q10 = corr_mat_crim_c2q10.drop('crim')
# Sorting the correlations by value
corr_mat_crim_c2q10=corr_mat_crim_c2q10.sort_values(ascending=False)
print_green_bold(corr_mat_crim_c2q10)
```

```
rad        0.625505
tax        0.582764
lstat      0.455621
nox        0.420972
indus      0.406583
age        0.352734
ptratio    0.289946
chas      -0.055892
zn        -0.200469
rm        -0.219247
dis       -0.379670
medv      -0.388305
Name: crim, dtype: float64
```

The variables rad and tax are highly positively correlated (>0.5) to crim. medv and dis show the highest negative correlation (>0.35 in absolute value) with crim.

(e) Do any of the suburbs of Boston appear to have particularly high crime rates? Tax rates? Pupil-teacher ratios? Comment on the range of each predictor.

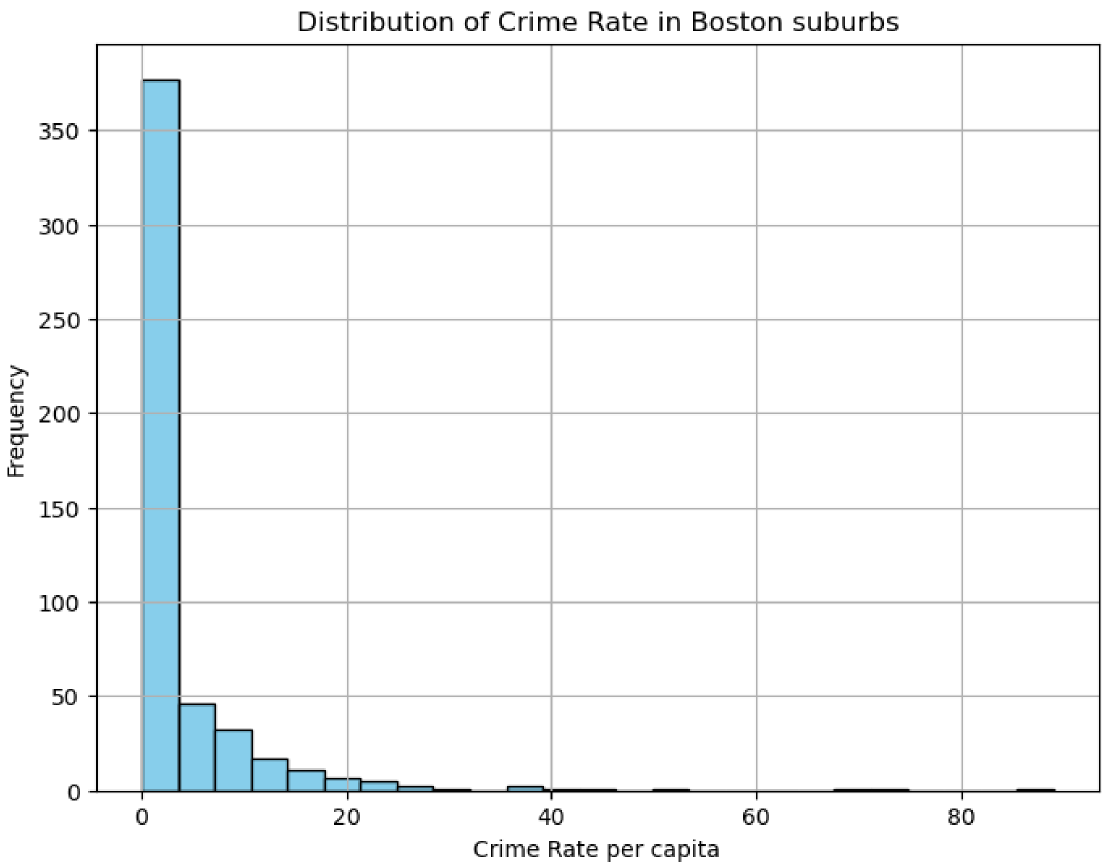
Crime Rates : Let us first look at the summary statistics for crime rate to understand the distribution of values.

[106]:
```
boston_c2q10['crim'].describe()
```

[106]:
```
count    506.000000
mean       3.613524
std        8.601545
min        0.006320
25%        0.082045
50%        0.256510
75%        3.677083
max       88.976200
Name: crim, dtype: float64
```

The median crime rate is 0.26 while that maximum crime rate is 89. This means that there are some suburbs with an exceptionally high level of crime. This is corroborated by the fact that the standard deviation (8.60) is relatively high compared to the mean crime rate of 3.61. Let us plot a histogram to look at the distribution of crime rates across suburbs.

[107]:
```
plt.figure(figsize=(8, 6))
plt.hist(boston_c2q10['crim'], bins=25, edgecolor='black', color='skyblue')
plt.xlabel('Crime Rate per capita')
plt.ylabel('Frequency')
plt.title('Distribution of Crime Rate in Boston suburbs')
plt.grid(True)
plt.show()
```



[108]:
```
# Split crim deciles with count
boston_c2q10['crim_decile'] = pd.qcut(boston_c2q10['crim'], 10)
crim_decile_count_c2q10 = boston_c2q10['crim_decile'].value_counts().
 ↪sort_index()
print_green_bold("Count of occurrences in each decile:")
print_green_bold(crim_decile_count_c2q10)
```

Count of occurrences in each decile:

```
(0.00532, 0.0382]    51
(0.0382, 0.0642]     51
(0.0642, 0.0992]     50
(0.0992, 0.15]       51
(0.15, 0.257]        50
(0.257, 0.55]        51
(0.55, 1.728]        50
(1.728, 5.581]       51
(5.581, 10.753]      50
(10.753, 88.976]     51
Name: crim_decile, dtype: int64
```

By examining the count of occurrences in each decile, we observe a clear trend: the majority of suburbs have lower crime rates, as evidenced by higher frequency in the lower deciles. However, there is a notable drop in the number of occurrences in the last decile (10.753 to 88.976), indicating that only a few suburbs have extremely high crime rates. This validates our initial point.

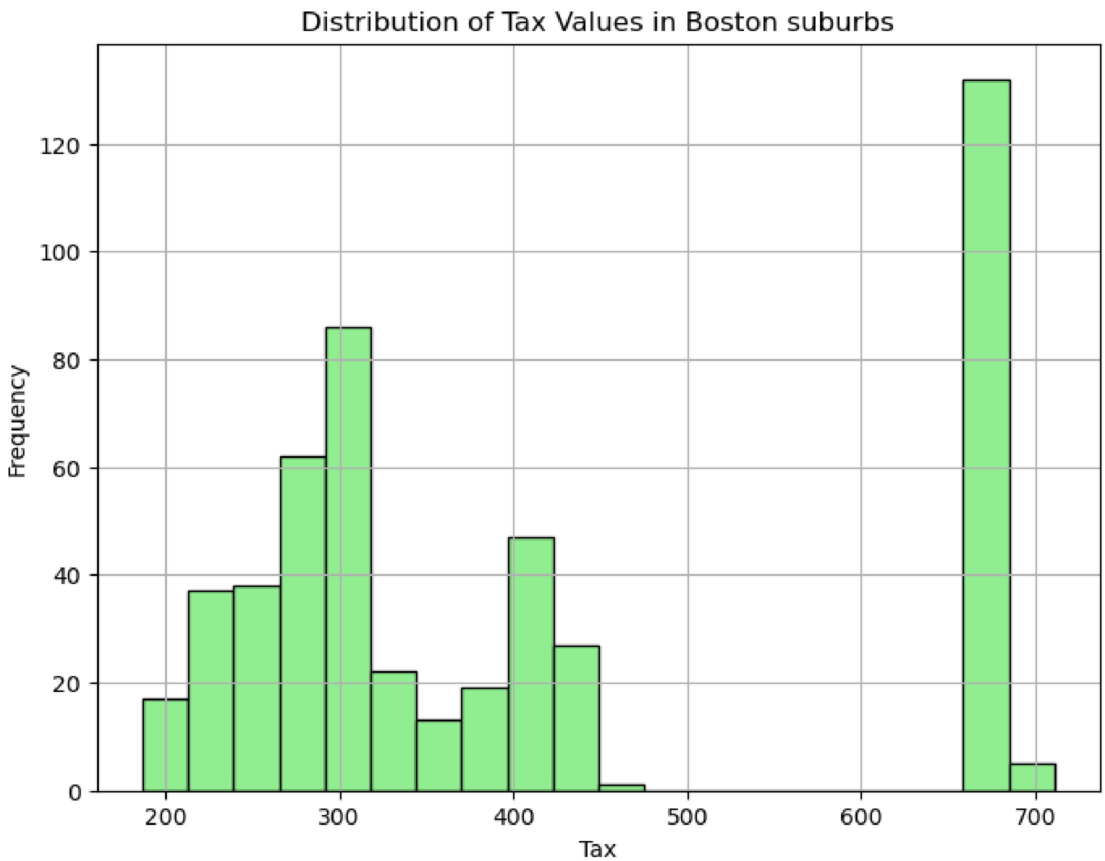
Tax : Let us first look at the summary statistics for tax to understand the distribution of values.

[109]:
```
boston_c2q10['tax'].describe()
```

[109]:
```
count    506.000000
mean     408.237154
std      168.537116
min      187.000000
25%      279.000000
50%      330.000000
75%      666.000000
max      711.000000
Name: tax, dtype: float64
```

The maximum tax in the dataset is 711 and is considerably higher than the mean tax of 408.24. This suggests that there are suburbs with exceptionally high taxes. Additionally, the 75th percentile (Q3) is closer to the maximum value, reinforcing the presence of suburbs with relatively higher taxes. Let's look at the histogram to understand the distribution better.

[110]:
```
# the distribution of tax values across suburbs using a histogram
plt.figure(figsize=(8, 6))
plt.hist(boston_c2q10['tax'], bins=20, edgecolor='black', color='lightgreen')
plt.xlabel('Tax')
plt.ylabel('Frequency')
plt.title('Distribution of Tax Values in Boston suburbs')
plt.grid(True)
plt.show()
```



It appears that while most suburbs have tax lower than 500, quite a few have values higher than 650. If we use 650 as the threshold, let us look at what percentage of suburbs have values above and below it.

[111]:
```
less_than_650_c2q10=len(boston_c2q10[boston_c2q10['tax']<650])
less_than_650_percentage_c2q10=round((less_than_650_c2q10/
 ↪len(boston_c2q10))*100,2)
greater_than_650_c2q10=len(boston_c2q10[boston_c2q10['tax']>=650])
greater_than_650_percentage_c2q10=round((greater_than_650_c2q10/
 ↪len(boston_c2q10))*100,2)
print_green_bold(f'The distribution of tax values in the Boston dataset aligns␣
 ↪with our earlier inference. Around {less_than_650_percentage_c2q10}% of␣
 ↪suburbs have lower tax rates (<650), while␣
 ↪{greater_than_650_percentage_c2q10}% have higher tax rates (>=650). This␣
 ↪confirms the presence of suburbs with exceptionally high tax rates,␣
 ↪supporting our previous observation.')
```

The distribution of tax values in the Boston dataset aligns with our earlier inference. Around 72.92% of suburbs have lower tax rates (<650), while 27.08% have higher tax rates (>=650). This confirms the presence of suburbs with exceptionally high tax rates, supporting our previous observation.

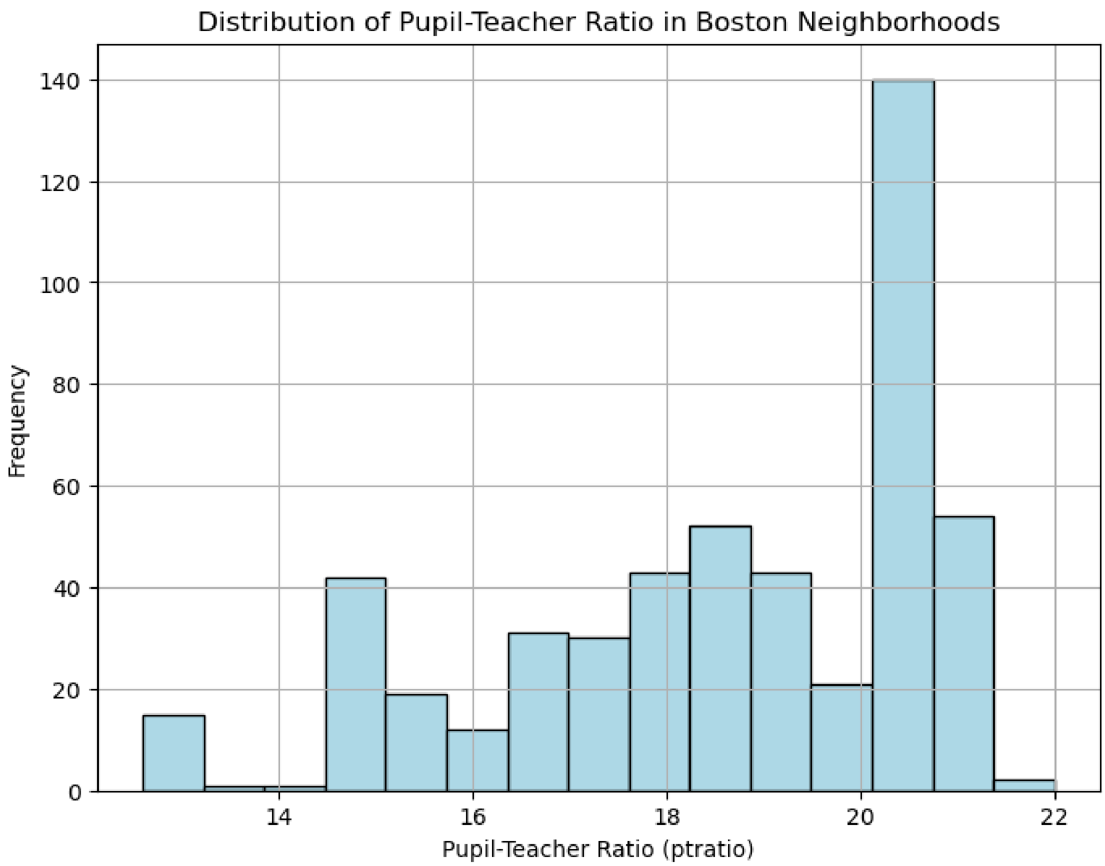
Pupil-Teacher Ratios : Let us first look at the summary statistics for ptratio to understand the distribution of values.

[112]:
```
boston_c2q10['ptratio'].describe()
```

[112]:
```
count    506.000000
mean      18.455534
std        2.164946
min       12.600000
25%       17.400000
50%       19.050000
75%       20.200000
max       22.000000
Name: ptratio, dtype: float64
```

The spread of the pupil-teacher ratio data is moderate, with a standard deviation of approximately 2.16. This indicates that while there is some variation in the number of students per teacher across suburbs, it is not as substantial as seen in other variables like crim or tax rates. In addition, the median is 19 and the max value is 22, which means that there are unlikely to be suburbs with exceptionally high pt ratios. Let us look at the histogram to confirm this.

[113]:
```
# The distribution of pupil-teacher ratio (ptratio) across neighborhoods using␣
↪a histogram
plt.figure(figsize=(8, 6))
plt.hist(boston_c2q10['ptratio'], bins=15, edgecolor='black', color='lightblue')
plt.xlabel('Pupil-Teacher Ratio (ptratio)')
plt.ylabel('Frequency')
plt.title('Distribution of Pupil-Teacher Ratio in Boston Neighborhoods')
plt.grid(True)
plt.show()
```



As expected there does not appear to be a great deal of spread in the data. Let's analyze the deciles to further corroborate this.

[114]:
```
# Split 'ptratio' into deciles and count occurrences in each decile
boston_c2q10['ptratio_decile'] = pd.qcut(boston_c2q10['ptratio'], 10,␣
↪duplicates='drop')
ptratio_decile_count_c2q10 = boston_c2q10['ptratio_decile'].value_counts().
↪sort_index()

# Print count of occurrences in each decile
print_green_bold("Count of occurrences in each decile:")
print_green_bold(ptratio_decile_count_c2q10)
```

```
Count of occurrences in each decile:
```

```
(12.599, 14.75]     51
(14.75, 16.6]       61
(16.6, 17.8]        62
(17.8, 18.4]        40
(18.4, 19.05]       39
(19.05, 19.7]       52
(19.7, 20.2]       145
(20.2, 20.9]        11
(20.9, 22.0]        45
Name: ptratio_decile, dtype: int64
```

The count of occurrences in each decile for ptratio confirms our previous point about pupil-teacher ratios not having a lot of spread. The counts remain relatively consistent in the middle deciles and show only small variations between adjacent deciles, indicating a low spread across neighborhoods.

(f) How many of the suburbs in this data set bound the Charles river?

[115]:
```
bound_by_criver_c2q10=len(boston_c2q10[boston_c2q10['chas']==1])
print_green_bold(f'The number of suburbs bound by the Charles River is␣
↪{bound_by_criver_c2q10}')
```

```
The number of suburbs bound by the Charles River is 35
```

(g) What is the median pupil-teacher ratio among the towns in this data set?

[116]:
```
median_ptratio_c2q10 = boston_c2q10['ptratio'].median()
print_green_bold(f'The median pupil-teacher ratio is {median_ptratio_c2q10}')
```

```
The median pupil-teacher ratio is 19.05
```

(h) Which suburb of Boston has lowest median value of owner- occupied homes? What are the values of the other predictors for that suburb, and how do those values compare to the overall ranges for those predictors? Comment on your findings.

[117]:
```
min_medv_c2q10=boston_c2q10['medv'].min()
min_medv_all_var_c2q10=boston_c2q10[boston_c2q10['medv']==min_medv_c2q10]
min_medv_all_var_c2q10
```

[117]:
```
        crim   zn  indus  chas    nox     rm    age     dis  rad  tax  \
398  38.3518  0.0   18.1     0  0.693  5.453  100.0  1.4896   24  666
405  67.9208  0.0   18.1     0  0.693  5.683  100.0  1.4254   24  666

     ptratio  lstat  medv        crim_decile ptratio_decile
398     20.2  30.59   5.0  (10.753, 88.976]   (19.7, 20.2]
405     20.2  22.98   5.0  (10.753, 88.976]   (19.7, 20.2]
```

[118]:
```
boston_c2q10.describe()
```

[118]:

	crim	zn	indus	chas	nox	rm
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000

	age	dis	rad	tax	ptratio	lstat
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	68.574901	3.795043	9.549407	408.237154	18.455534	12.653063
std	28.148861	2.105710	8.707259	168.537116	2.164946	7.141062
min	2.900000	1.129600	1.000000	187.000000	12.600000	1.730000
25%	45.025000	2.100175	4.000000	279.000000	17.400000	6.950000
50%	77.500000	3.207450	5.000000	330.000000	19.050000	11.360000
75%	94.075000	5.188425	24.000000	666.000000	20.200000	16.955000
max	100.000000	12.126500	24.000000	711.000000	22.000000	37.970000

	medv
count	506.000000
mean	22.532806
std	9.197104
min	5.000000
25%	17.025000
50%	21.200000
75%	25.000000
max	50.000000

Based on the comparison, we can see that the two suburbs with the lowest medv values have higher crime rates (crim) and a high percentage of lower-status population (lstat) compared to the overall range. Additionally, they both have no residential land zoned for large lots (zn = 0.0) and are situated near non-retail business areas (indus = 18.1). The pupil-teacher ratio (ptratio) and nitric oxides concentration (nox) for these suburbs are within the overall range.

Overall, these findings suggest that the suburbs with the lowest medv values have higher crime rates, higher percentage of lower-status population, and less residential land zoned for large lots compared to the rest of the suburbs in the dataset. The data also indicates that they are located near non-retail business areas. These factors may contribute to the lower median values of owner-occupied homes in these specific suburbs.

(i) In this data set, how many of the suburbs average more than seven rooms per dwelling? More than eight rooms per dwelling? Comment on the suburbs that average more than eight rooms per dwelling.

[119]:
```
rooms_7_c2q10=len(boston_c2q10[boston_c2q10['rm']>7])
rooms_8_c2q10=len(boston_c2q10[boston_c2q10['rm']>8])
print_green_bold(f'In the Boston dataset,the number of suburbs which average␣
↪more than 7 rooms per dwelling is {rooms_7_c2q10}. Around {rooms_8_c2q10}␣
↪suburbs average more than 8 rooms per dwelling.')
```

In the Boston dataset,the number of suburbs which average more than 7 rooms per dwelling is 64. Around 13 suburbs average more than 8 rooms per dwelling.

[120]:
```
rooms_8_df_c2q10=boston_c2q10[boston_c2q10['rm']>8]
rooms_8_df_c2q10.describe()
```

[120]:

	crim	zn	indus	chas	nox	rm
count	13.000000	13.000000	13.000000	13.000000	13.000000	13.000000
mean	0.718795	13.615385	7.078462	0.153846	0.539238	8.348538
std	0.901640	26.298094	5.392767	0.375534	0.092352	0.251261
min	0.020090	0.000000	2.680000	0.000000	0.416100	8.034000
25%	0.331470	0.000000	3.970000	0.000000	0.504000	8.247000
50%	0.520140	0.000000	6.200000	0.000000	0.507000	8.297000
75%	0.578340	20.000000	6.200000	0.000000	0.605000	8.398000
max	3.474280	95.000000	19.580000	1.000000	0.718000	8.780000

	age	dis	rad	tax	ptratio	lstat
count	13.000000	13.000000	13.000000	13.000000	13.000000	13.000000
mean	71.538462	3.430192	7.461538	325.076923	16.361538	4.310000
std	24.608723	1.883955	5.332532	110.971063	2.410580	1.373566
min	8.400000	1.801000	2.000000	224.000000	13.000000	2.470000
25%	70.400000	2.288500	5.000000	264.000000	14.700000	3.320000
50%	78.300000	2.894400	7.000000	307.000000	17.400000	4.140000
75%	86.500000	3.651900	8.000000	307.000000	17.400000	5.120000
max	93.900000	8.906700	24.000000	666.000000	20.200000	7.440000

	medv
count	13.000000
mean	44.200000
std	8.092383
min	21.900000
25%	41.700000
50%	48.300000
75%	50.000000
max	50.000000

If we compare with the summary stats of the overall Boston dataset, we notice the following : Crime rates (crim) are much lower compared to the overall dataset. By comparing the 75th percentile values we find that around 75% of these suburbs have crime rates lower than 0.58. The corresponding figure for the entire dataset is 3.68. lstat (% lower status of the population) values are much lower for these suburbs.The maximum lstat value here is 7.4 while it is 38 for the overall dataset. This makes sense as suburbs where the average house has more than 8 rooms are likely to be populated with wealthy people. As expected medv or median value of owner occupied homes is also much higher in these suburbs. The 25th percentile medv for these suburbs corresponds to 41.7 as compared to 17 for the overall data containing all suburbs. The zn values are also higher for these suburbs as compared to the overall data as demonstrated by a higher value of 20 vs 12.5 for the 75th percentile. This means that these suburbs have larger residential lots. These suburbs also have lower industrial development. The 75th percentile of indus is 6.2 as compared to 18.1 in the overall data. The nox values i.e. nitric oxide concentrations also appear to be lower for these suburbs (75th percentile is 0.605 vs 0.624 and the maximum value is 0.718 vs 0.871).

To summarise, the suburbs with more than an average of 8 rooms per dwelling have the following characteristics : - Lower crime rates - Lower percentage of lower status population - High median value of homes - Larger residential lots - Lower industrial development - Lower concentrations of nitric oxide