numerical_data_processing

May 6, 2024

1 0- Installation

Install all the required packages first by running the following line:

[3]:

```
!pip3 install -r requirements.txt
```

```
Requirement already satisfied: anyio==4.3.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 1)) (4.3.0)
Requirement already satisfied: appnope==0.1.4 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 2)) (0.1.4)
Requirement already satisfied: argon2-cffi==23.1.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 3)) (23.1.0)
Requirement already satisfied: argon2-cffi-bindings==21.2.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 4)) (21.2.0)
Requirement already satisfied: arrow==1.3.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 5)) (1.3.0)
Requirement already satisfied: asttokens==2.4.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 6)) (2.4.1)
Requirement already satisfied: async-lru==2.0.4 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 7)) (2.0.4)
Requirement already satisfied: attrs==23.2.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 8)) (23.2.0)
Requirement already satisfied: Babel==2.14.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 9)) (2.14.0)
Requirement already satisfied: beautifulsoup4==4.12.3 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 10)) (4.12.3)
Requirement already satisfied: bleach==6.1.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
```

```
(from -r requirements.txt (line 11)) (6.1.0)
Requirement already satisfied: certifi==2024.2.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 12)) (2024.2.2)
Requirement already satisfied: cffi==1.16.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 13)) (1.16.0)
Requirement already satisfied: charset-normalizer==3.3.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 14)) (3.3.2)
Requirement already satisfied: comm==0.2.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 15)) (0.2.2)
Requirement already satisfied: contourpy==1.2.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 16)) (1.2.1)
Requirement already satisfied: cycler==0.12.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 17)) (0.12.1)
Requirement already satisfied: debugpy==1.8.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 18)) (1.8.1)
Requirement already satisfied: decorator==5.1.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 19)) (5.1.1)
Requirement already satisfied: defusedxml==0.7.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 20)) (0.7.1)
Requirement already satisfied: exceptiongroup==1.2.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 21)) (1.2.0)
Requirement already satisfied: executing==2.0.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 22)) (2.0.1)
Requirement already satisfied: fastjsonschema==2.19.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 23)) (2.19.1)
Requirement already satisfied: filelock==3.13.4 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 24)) (3.13.4)
Requirement already satisfied: fonttools==4.51.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 25)) (4.51.0)
Requirement already satisfied: fqdn==1.5.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 26)) (1.5.1)
Requirement already satisfied: fsspec==2024.3.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
```

```
(from -r requirements.txt (line 27)) (2024.3.1)
Requirement already satisfied: h11==0.14.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 28)) (0.14.0)
Requirement already satisfied: httpcore==1.0.5 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 29)) (1.0.5)
Requirement already satisfied: httpx==0.27.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 30)) (0.27.0)
Requirement already satisfied: huggingface-hub==0.22.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 31)) (0.22.2)
Requirement already satisfied: idna==3.7 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 32)) (3.7)
Requirement already satisfied: importlib_metadata==7.1.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 33)) (7.1.0)
Requirement already satisfied: importlib_resources==6.4.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 34)) (6.4.0)
Requirement already satisfied: ipykernel==6.29.4 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 35)) (6.29.4)
Requirement already satisfied: ipython==8.18.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 36)) (8.18.1)
Requirement already satisfied: ipywidgets==8.1.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 37)) (8.1.2)
Requirement already satisfied: isoduration==20.11.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 38)) (20.11.0)
Requirement already satisfied: jedi==0.19.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 39)) (0.19.1)
Requirement already satisfied: Jinja2==3.1.3 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 40)) (3.1.3)
Requirement already satisfied: json5==0.9.25 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 41)) (0.9.25)
Requirement already satisfied: jsonpointer==2.4 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 42)) (2.4)
Requirement already satisfied: jsonschema==4.21.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
```

```
(from -r requirements.txt (line 43)) (4.21.1)
Requirement already satisfied: jsonschema-specifications==2023.12.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 44)) (2023.12.1)
Requirement already satisfied: jupyter==1.0.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 45)) (1.0.0)
Requirement already satisfied: jupyter_client==8.6.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 46)) (8.6.1)
Requirement already satisfied: jupyter-console==6.6.3 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 47)) (6.6.3)
Requirement already satisfied: jupyter_core==5.7.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 48)) (5.7.2)
Requirement already satisfied: jupyter-events==0.10.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 49)) (0.10.0)
Requirement already satisfied: jupyter-lsp==2.2.5 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 50)) (2.2.5)
Requirement already satisfied: jupyter_server==2.14.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 51)) (2.14.0)
Requirement already satisfied: jupyter_server_terminals==0.5.3 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 52)) (0.5.3)
Requirement already satisfied: jupyterlab==4.1.6 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 53)) (4.1.6)
Requirement already satisfied: jupyterlab_pygments==0.3.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 54)) (0.3.0)
Requirement already satisfied: jupyterlab_server==2.26.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 55)) (2.26.0)
Requirement already satisfied: jupyterlab_widgets==3.0.10 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 56)) (3.0.10)
Requirement already satisfied: kiwisolver==1.4.5 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 57)) (1.4.5)
Requirement already satisfied: MarkupSafe==2.1.5 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 58)) (2.1.5)
Requirement already satisfied: matplotlib==3.8.4 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
```

```
(from -r requirements.txt (line 59)) (3.8.4)
Requirement already satisfied: matplotlib-inline==0.1.7 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 60)) (0.1.7)
Requirement already satisfied: mistune==3.0.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 61)) (3.0.2)
Requirement already satisfied: mpmath==1.3.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 62)) (1.3.0)
Requirement already satisfied: nbclient==0.10.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 63)) (0.10.0)
Requirement already satisfied: nbconvert==7.16.3 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 64)) (7.16.3)
Requirement already satisfied: nbformat==5.10.4 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 65)) (5.10.4)
Requirement already satisfied: nest_asyncio==1.6.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 66)) (1.6.0)
Requirement already satisfied: networkx==3.3 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 67)) (3.3)
Requirement already satisfied: notebook==7.1.3 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 68)) (7.1.3)
Requirement already satisfied: notebook_shim==0.2.4 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 69)) (0.2.4)
Requirement already satisfied: numpy==1.26.4 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 70)) (1.26.4)
Requirement already satisfied: overrides==7.7.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 71)) (7.7.0)
Requirement already satisfied: packaging==24.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 72)) (24.0)
Requirement already satisfied: pandas==2.2.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 73)) (2.2.2)
Requirement already satisfied: pandocfilters==1.5.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 74)) (1.5.1)
Requirement already satisfied: parso==0.8.4 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
```

```
(from -r requirements.txt (line 75)) (0.8.4)
Requirement already satisfied: pexpect==4.9.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 76)) (4.9.0)
Requirement already satisfied: pickleshare==0.7.5 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 77)) (0.7.5)
Requirement already satisfied: pillow==10.3.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 78)) (10.3.0)
Requirement already satisfied: pip==23.3.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 79)) (23.3.1)
Requirement already satisfied: platformdirs==4.2.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 80)) (4.2.0)
Requirement already satisfied: prometheus_client==0.20.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 81)) (0.20.0)
Requirement already satisfied: prompt-toolkit==3.0.43 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 82)) (3.0.43)
Requirement already satisfied: psutil==5.9.8 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 83)) (5.9.8)
Requirement already satisfied: ptyprocess==0.7.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 84)) (0.7.0)
Requirement already satisfied: pure-eval==0.2.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 85)) (0.2.2)
Requirement already satisfied: pycparser==2.22 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 86)) (2.22)
Requirement already satisfied: Pygments==2.17.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 87)) (2.17.2)
Requirement already satisfied: pyparsing==3.1.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 88)) (3.1.2)
Requirement already satisfied: python-dateutil==2.9.0.post0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 89)) (2.9.0.post0)
Requirement already satisfied: python-json-logger==2.0.7 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 90)) (2.0.7)
Requirement already satisfied: pytz==2024.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
```

```
(from -r requirements.txt (line 91)) (2024.1)
Requirement already satisfied: PyYAML==6.0.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 92)) (6.0.1)
Requirement already satisfied: pyzmq==26.0.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 93)) (26.0.0)
Requirement already satisfied: qtconsole==5.5.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 94)) (5.5.1)
Requirement already satisfied: QtPy==2.4.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 95)) (2.4.1)
Requirement already satisfied: referencing==0.34.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 96)) (0.34.0)
Requirement already satisfied: regex==2024.4.16 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 97)) (2024.4.16)
Requirement already satisfied: requests==2.31.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 98)) (2.31.0)
Requirement already satisfied: rfc3339-validator==0.1.4 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 99)) (0.1.4)
Requirement already satisfied: rfc3986-validator==0.1.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 100)) (0.1.1)
Requirement already satisfied: rpds-py==0.18.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 101)) (0.18.0)
Requirement already satisfied: safetensors==0.4.3 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 102)) (0.4.3)
Requirement already satisfied: seaborn==0.13.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 103)) (0.13.2)
Requirement already satisfied: Send2Trash==1.8.3 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 104)) (1.8.3)
Requirement already satisfied: setuptools==68.2.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 105)) (68.2.2)
Requirement already satisfied: six==1.16.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 106)) (1.16.0)
Requirement already satisfied: sniffio==1.3.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
```

```
(from -r requirements.txt (line 107)) (1.3.1)
Requirement already satisfied: soupsieve==2.5 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 108)) (2.5)
Requirement already satisfied: stack-data==0.6.3 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 109)) (0.6.3)
Requirement already satisfied: sympy==1.12 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 110)) (1.12)
Requirement already satisfied: terminado==0.18.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 111)) (0.18.1)
Requirement already satisfied: tinycss2==1.2.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 112)) (1.2.1)
Requirement already satisfied: tokenizers==0.19.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 113)) (0.19.1)
Requirement already satisfied: tomli==2.0.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 114)) (2.0.1)
Requirement already satisfied: torch==2.2.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 115)) (2.2.2)
Requirement already satisfied: tornado==6.4 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 116)) (6.4)
Requirement already satisfied: tqdm==4.66.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 117)) (4.66.2)
Requirement already satisfied: traitlets==5.14.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 118)) (5.14.2)
Requirement already satisfied: transformers==4.40.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 119)) (4.40.0)
Requirement already satisfied: types-python-dateutil==2.9.0.20240316 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 120)) (2.9.0.20240316)
Requirement already satisfied: typing_extensions==4.11.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 121)) (4.11.0)
Requirement already satisfied: tzdata==2024.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 122)) (2024.1)
Requirement already satisfied: uri-template==1.3.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
```

```
(from -r requirements.txt (line 123)) (1.3.0)
Requirement already satisfied: urllib3==2.2.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 124)) (2.2.1)
Requirement already satisfied: wcwidth==0.2.13 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 125)) (0.2.13)
Requirement already satisfied: webcolors==1.13 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 126)) (1.13)
Requirement already satisfied: webencodings==0.5.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 127)) (0.5.1)
Requirement already satisfied: websocket-client==1.7.0 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 128)) (1.7.0)
Requirement already satisfied: wheel==0.41.2 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 129)) (0.41.2)
Requirement already satisfied: widgetsnbextension==4.0.10 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 130)) (4.0.10)
Requirement already satisfied: zipp==3.18.1 in
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages
(from -r requirements.txt (line 131)) (3.18.1)

[notice] A new release of pip is
available: 23.3.1 -> 24.0
[notice] To update, run:
pip install --upgrade pip
```

2 1- Introduction to Numpy

[4]:

```python
import numpy as np
```

[5]:

```python
np.random.seed(0)  # the seed ensures reproducibility: same 'randomness' every␣
 ↪time you run this notebook
```

[6]:

```python
one_dim = np.random.randint(10, size=2)  # One-dimensional array
two_dim = np.random.randint(10, size=(3, 5))  # Two-dimensional array
three_dim = np.random.randint(10, size=(3, 4, 5))  # Three-dimensional array
```

Exercise Analyze the three created arrays using the `ndim`, `shape` and `size` attributes.

[7]:

```python
def analyze_array(array: list) -> tuple:
    """
    Returns the statistics of an array
    Args:
        array (list): numpy array

    Returns:
        tuple: array, size and byte-size
    """
    return array.ndim, array.shape, array.size
# ndim == dimensions
# shape == columns and rows
# size == byte-size

print(analyze_array(one_dim))
print(analyze_array(two_dim))
print(analyze_array(three_dim))
```

```
(1, (2,), 2)
(2, (3, 5), 15)
(3, (3, 4, 5), 60)
```

Exercise Create an array filled with ascending integer values from 0 to 14. Then change its shape to (3,5).

[8]:

```
ascending_array = np.array(list(range(15))[::-1])
print(np.reshape(ascending_array, (3, 5)))
```

```
[[14 13 12 11 10]
 [ 9  8  7  6  5]
 [ 4  3  2  1  0]]
```

2.0.1 Indexing and Slicing

[9]:

```
A = np.array([[1,2,3,4], [5,6,7,8], [9,10,11,12]])
A
```

[9]:

```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
```

[10]:

```
A[1]  # Indexes second row
```

[10]:

```
array([5, 6, 7, 8])
```

[11]:

```
A[2, 1] # Index element at third row, second column
```

[11]:

```
10
```

Exercise *get the indexes from the third column*

```
[12]: A[:, 2:]
```

```
[12]: array([[ 3,  4],
             [ 7,  8],
             [11, 12]])
```

Exercise *get subset of elements: first two rows and three columns*

```
[13]: A[:2, :3]
```

```
[13]: array([[1, 2, 3],
             [5, 6, 7]])
```

Exercise *get subset of elements: last two rows and three columns*

```
[14]: A[-2:, -3:]
```

```
[14]: array([[ 6,  7,  8],
             [10, 11, 12]])
```

Exercise reverse all elements, get only every other column (hint: `::2`)

```
[15]: A[::-1, ::-2]
```

```
[15]: array([[12, 10],
             [ 8,  6],
             [ 4,  2]])
```

2.0.2 Numpy basic operations

```
[16]: a = np.array([20,30,40,50])
      a
```

```
[16]: array([20, 30, 40, 50])
```

```
[17]: b = np.arange(4)
      b
```

```
[17]: array([0, 1, 2, 3])
```

```
[18]: subtraction = a - b  # subtraction
      print(subtraction)
      print(a + b)  # addition
```

```
[20 29 38 47]
[20 31 42 53]
```

```
[19]: a == 20  # conditional
```

```
[19]: array([ True, False, False, False])
```

```
[20]: a[a == 20]  # apply condition to get elements
```

```
[20]: array([20])
```

Exercise *get only elements from* ***`subtraction`*** *that are divisible by 2* (hint: modulo (%) of elements divisible by two is 0)

```
[21]: subtraction[subtraction % 2 == 0]
```

```
[21]: array([20, 38])
```

Exercise *get the cosine of each element in* ***`a`*** *(check the numpy documentation)*

```
[22]: np.cos(a)
```

```
[22]: array([ 0.40808206,  0.15425145, -0.66693806,  0.96496603])
```

Exercise *practise with aggregates:* ***`np.min`****,* ***`np.max`*** *and* ***`np.sum`***

```
[23]: np.min(a), np.max(a), np.sum(a)
```

```
[23]: (20, 50, 140)
```

3 2 - Pandas

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

```
[25]: df = pd.read_csv("top50spotify.csv", encoding = "latin", header=0, index_col=0)␣
       ↪# load csv file
```

```
[26]: df.head(3)
```

```
[26]:                        Track.Name    Artist.Name           Genre  \
      1                          Señorita   Shawn Mendes    canadian pop
      2                             China       Anuel AA  reggaeton flow
      3  boyfriend (with Social House)  Ariana Grande       dance pop

         Beats.Per.Minute  Energy  Danceability  Loudness..dB..  Liveness  Valence.  \
      1               117      55            76              -6         8        75
      2               105      81            79              -4         8        61
      3               190      80            40              -4        16        70

         Length.  Acousticness..  Speechiness.  Popularity
      1      191               4             3          79
      2      302               8             9          92
      3      186              12            46          85
```

```
[27]: df.shape
```

[27]:
```
(50, 13)
```

[28]:
```
df.columns
```

[28]:
```
Index(['Track.Name', 'Artist.Name', 'Genre', 'Beats.Per.Minute', 'Energy',
       'Danceability', 'Loudness..dB..', 'Liveness', 'Valence.', 'Length.',
       'Acousticness..', 'Speechiness.', 'Popularity'],
      dtype='object')
```

[29]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 50 entries, 1 to 50
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Track.Name        50 non-null     object
 1   Artist.Name       50 non-null     object
 2   Genre             50 non-null     object
 3   Beats.Per.Minute  50 non-null     int64
 4   Energy            50 non-null     int64
 5   Danceability      50 non-null     int64
 6   Loudness..dB..    50 non-null     int64
 7   Liveness          50 non-null     int64
 8   Valence.          50 non-null     int64
 9   Length.           50 non-null     int64
 10  Acousticness..    50 non-null     int64
 11  Speechiness.      50 non-null     int64
 12  Popularity        50 non-null     int64
dtypes: int64(10), object(3)
memory usage: 5.5+ KB
```

[30]:
```
df.describe()
```

[30]:
```
       Beats.Per.Minute     Energy  Danceability  Loudness..dB..   Liveness  \
count         50.000000  50.000000      50.00000       50.000000  50.000000
mean         120.060000  64.060000      71.38000       -5.660000  14.660000
std           30.898392  14.231913      11.92988        2.056448  11.118306
min           85.000000  32.000000      29.00000      -11.000000   5.000000
25%           96.000000  55.250000      67.00000       -6.750000   8.000000
50%          104.500000  66.500000      73.50000       -6.000000  11.000000
75%          137.500000  74.750000      79.75000       -4.000000  15.750000
max          190.000000  88.000000      90.00000       -2.000000  58.000000

        Valence.     Length.  Acousticness..  Speechiness.  Popularity
count  50.000000   50.000000       50.000000     50.000000   50.000000
mean   54.600000  200.960000       22.160000     12.480000   87.500000
std    22.336024   39.143879       18.995553     11.161596    4.491489
```

```
min    10.000000  115.000000        1.000000       3.000000   70.000000
25%    38.250000  176.750000        8.250000       5.000000   86.000000
50%    55.500000  198.000000       15.000000       7.000000   88.000000
75%    69.500000  217.500000       33.750000      15.000000   90.750000
max    95.000000  309.000000       75.000000      46.000000   95.000000
```

[31]:

```
df[0:3] # select first three rows of the dataframe
```

[31]:

```
                      Track.Name    Artist.Name           Genre  \
1                       Señorita   Shawn Mendes    canadian pop
2                          China       Anuel AA  reggaeton flow
3  boyfriend (with Social House)  Ariana Grande       dance pop

   Beats.Per.Minute  Energy  Danceability  Loudness..dB..  Liveness  Valence.  \
1               117      55            76              -6         8        75
2               105      81            79              -4         8        61
3               190      80            40              -4        16        70

   Length.  Acousticness..  Speechiness.  Popularity
1      191               4             3          79
2      302               8             9          92
3      186              12            46          85
```

[32]:

```
df.loc[0:3] # select first three rows of the dataframe
```

[32]:

```
                      Track.Name    Artist.Name           Genre  \
1                       Señorita   Shawn Mendes    canadian pop
2                          China       Anuel AA  reggaeton flow
3  boyfriend (with Social House)  Ariana Grande       dance pop

   Beats.Per.Minute  Energy  Danceability  Loudness..dB..  Liveness  Valence.  \
1               117      55            76              -6         8        75
2               105      81            79              -4         8        61
3               190      80            40              -4        16        70

   Length.  Acousticness..  Speechiness.  Popularity
1      191               4             3          79
2      302               8             9          92
3      186              12            46          85
```

Exercise: *select by position, last three rows, cols* `Track.Name` *and* `Artist.Name`

[33]:

```
df.iloc[:-3, [0, 1]]
```

[33]:

```
                                           Track.Name       Artist.Name
1                                            Señorita      Shawn Mendes
2                                               China          Anuel AA
3                       boyfriend (with Social House)     Ariana Grande
```

```
4                      Beautiful People (feat. Khalid)        Ed Sheeran
5                          Goodbyes (Feat. Young Thug)       Post Malone
6                    I Don't Care (with Justin Bieber)        Ed Sheeran
7                                               Ransom         Lil Tecca
8                                    How Do You Sleep?         Sam Smith
9                                Old Town Road - Remix         Lil Nas X
10                                             bad guy     Billie Eilish
11                                            Callaita         Bad Bunny
12               Loco Contigo (feat. J. Balvin & Tyga)          DJ Snake
13                                   Someone You Loved     Lewis Capaldi
14                                  Otro Trago - Remix              Sech
15            Money In The Grave (Drake ft. Rick Ross)             Drake
16                           No Guidance (feat. Drake)       Chris Brown
17                                          LA CANCIÓN          J Balvin
18       Sunflower - Spider-Man: Into the Spider-Verse       Post Malone
19                                              Lalala               Y2K
20                                         Truth Hurts             Lizzo
21                                 Piece Of Your Heart            MEDUZA
22                                              Panini         Lil Nas X
23                                No Me Conoce - Remix       Jhay Cortez
24                                     Soltera - Remix             Lunay
25                        bad guy (with Justin Bieber)     Billie Eilish
26                                 If I Can't Have You      Shawn Mendes
27                                        Dance Monkey       Tones and I
28                                            It's You         Ali Gatie
29                                           Con Calma      Daddy Yankee
30                                       QUE PRETENDES          J Balvin
31                                            Takeaway  The Chainsmokers
32                                             7 rings     Ariana Grande
33                                   0.958333333333333            Maluma
34           The London (feat. J. Cole & Travis Scott)        Young Thug
35                                   Never Really Over        Katy Perry
36  Summer Days (feat. Macklemore & Patrick Stump …     Martin Garrix
37                                          Otro Trago              Sech
38                      Antisocial (with Travis Scott)        Ed Sheeran
39                                              Sucker    Jonas Brothers
40  fuck, i'm lonely (with Anne-Marie) - from  13 …             Lauv
41                                         Higher Love              Kygo
42                               You Need To Calm Down      Taylor Swift
43                                             Shallow         Lady Gaga
44                                                Talk            Khalid
45                                          Con Altura           ROSALÍA
46                                     One Thing Right        Marshmello
47                                           Te Robaré         Nicky Jam
```

Exercise: *find out how many songs there are per* `Genre`

```
[34]: df["Genre"].value_counts()
```

```
[34]: Genre
      dance pop           8
      pop                 7
      latin               5
      canadian hip hop    3
      edm                 3
      reggaeton           2
      reggaeton flow      2
      panamanian pop      2
      canadian pop        2
      electropop          2
      country rap         2
      dfw rap             2
      brostep             2
      trap music          1
      escape room         1
      pop house           1
      australian pop      1
      atl hip hop         1
      big room            1
      boy band            1
      r&b en espanol      1
      Name: count, dtype: int64
```

Exercise: *get all entries with* ***`Popularity`*** *higher than 90*

```
[35]: df[df["Popularity"] > 90]
```

```
[35]:                                        Track.Name    Artist.Name  \
      2                                           China       Anuel AA
      5                     Goodbyes (Feat. Young Thug)    Post Malone
      7                                          Ransom      Lil Tecca
      10                                        bad guy  Billie Eilish
      11                                       Callaita      Bad Bunny
      15       Money In The Grave (Drake ft. Rick Ross)          Drake
      18  Sunflower - Spider-Man: Into the Spider-Verse    Post Malone
      20                                    Truth Hurts          Lizzo
      21                            Piece Of Your Heart         MEDUZA
      22                                         Panini      Lil Nas X
      24                                Soltera - Remix          Lunay
      29                                      Con Calma   Daddy Yankee
      37                                     Otro Trago           Sech

                     Genre  Beats.Per.Minute  Energy  Danceability  Loudness..dB..  \
      2     reggaeton flow               105      81            79              -4
      5            dfw rap               150      65            58              -4
```

```
7          trap music               180      64            75              -6
10        electropop               135      43            70             -11
11         reggaeton               176      62            61              -5
15  canadian hip hop               101      50            83              -4
18           dfw rap                90      48            76              -6
20       escape room               158      62            72              -3
21         pop house               124      74            68              -7
22       country rap               154      59            70              -6
24             latin                92      78            80              -4
29             latin                94      86            74              -3
37    panamanian pop               176      70            75              -5

    Liveness  Valence.  Length.  Acousticness..  Speechiness.  Popularity
2          8        61      302               8             9          92
5         11        18      175              45             7          94
7          7        23      131               2            29          92
10        10        56      194              33            38          95
11        24        24      251              60            31          93
15        12        10      205              10             5          92
18         7        91      158              56             5          91
20        12        41      173              11            11          91
21         7        63      153               4             3          91
22        12        48      115              34             8          91
24        44        80      266              36             4          91
29         6        66      193              11             6          91
37        11        62      226              14            34          91
```

Exercise: *group by genre and get the mean of the* ***`Beats.Per.Minute`***

[36]:
```
df.groupby("Genre").mean("Beats.Per.Minute")
```

[36]:
```
                  Beats.Per.Minute     Energy  Danceability  Loudness..dB..  \
Genre
atl hip hop              98.000000  59.000000     80.000000       -7.000000
australian pop           98.000000  59.000000     82.000000       -6.000000
big room                114.000000  72.000000     66.000000       -7.000000
boy band                138.000000  73.000000     84.000000       -5.000000
brostep                  94.000000  70.500000     67.500000       -2.500000
canadian hip hop        109.000000  45.000000     80.000000       -6.333333
canadian pop            120.500000  68.500000     72.500000       -5.000000
country rap             145.000000  60.500000     79.000000       -6.000000
dance pop               111.875000  59.875000     70.250000       -6.125000
dfw rap                 120.000000  56.500000     67.000000       -5.000000
edm                      97.666667  63.000000     52.333333       -7.000000
electropop              135.000000  44.000000     68.500000      -11.000000
escape room             158.000000  62.000000     72.000000       -3.000000
latin                   126.200000  76.600000     72.000000       -4.200000
panamanian pop          176.000000  74.500000     74.000000       -3.500000
```

```
pop                     114.142857  63.285714     68.428571       -6.285714
pop house               124.000000  74.000000     68.000000       -7.000000
r&b en espanol           98.000000  69.000000     88.000000       -4.000000
reggaeton               136.000000  66.500000     69.500000       -5.000000
reggaeton flow           98.500000  80.000000     80.000000       -4.000000
trap music              180.000000  64.000000     75.000000       -6.000000

                   Liveness   Valence.     Length.  Acousticness..  \
Genre
atl hip hop       13.000000  18.000000  200.000000        2.000000
australian pop    18.000000  54.000000  210.000000       69.000000
big room          14.000000  32.000000  164.000000       18.000000
boy band          11.000000  95.000000  181.000000        4.000000
brostep           37.500000  55.500000  198.000000       13.000000
canadian hip hop  15.000000  33.333333  193.000000       21.666667
canadian pop      10.500000  81.000000  191.000000       26.500000
country rap       11.500000  56.000000  136.000000       19.500000
dance pop         15.500000  45.875000  202.625000       27.000000
dfw rap            9.000000  54.500000  166.500000       50.500000
edm               20.333333  42.000000  218.666667       12.333333
electropop        11.000000  62.000000  194.500000       29.000000
escape room       12.000000  41.000000  173.000000       11.000000
latin             21.000000  72.600000  225.200000       17.800000
panamanian pop     8.500000  69.000000  257.000000       10.500000
pop               12.142857  58.000000  195.428571       21.428571
pop house          7.000000  63.000000  153.000000        4.000000
r&b en espanol     5.000000  75.000000  162.000000       39.000000
reggaeton         16.500000  46.000000  213.500000       41.000000
reggaeton flow     8.500000  59.500000  305.500000       11.000000
trap music         7.000000  23.000000  131.000000        2.000000

                  Speechiness.  Popularity
Genre
atl hip hop          15.000000   89.000000
australian pop       10.000000   83.000000
big room              6.000000   89.000000
boy band              6.000000   80.000000
brostep               5.000000   88.000000
canadian hip hop      5.333333   89.666667
canadian pop          4.500000   74.500000
country rap           9.000000   89.000000
dance pop            15.250000   85.750000
dfw rap               6.000000   92.500000
edm                   3.333333   86.666667
electropop           34.000000   92.000000
escape room          11.000000   91.000000
latin                14.600000   89.800000
```

```
panamanian pop       27.000000   89.000000
pop                   9.285714   85.857143
pop house             3.000000   91.000000
r&b en espanol       12.000000   88.000000
reggaeton            29.500000   91.000000
reggaeton flow        8.000000   87.500000
trap music           29.000000   92.000000
```

4 3- Seaborn

Use seaborn to visualize the "top50spotify.csv" dataset.

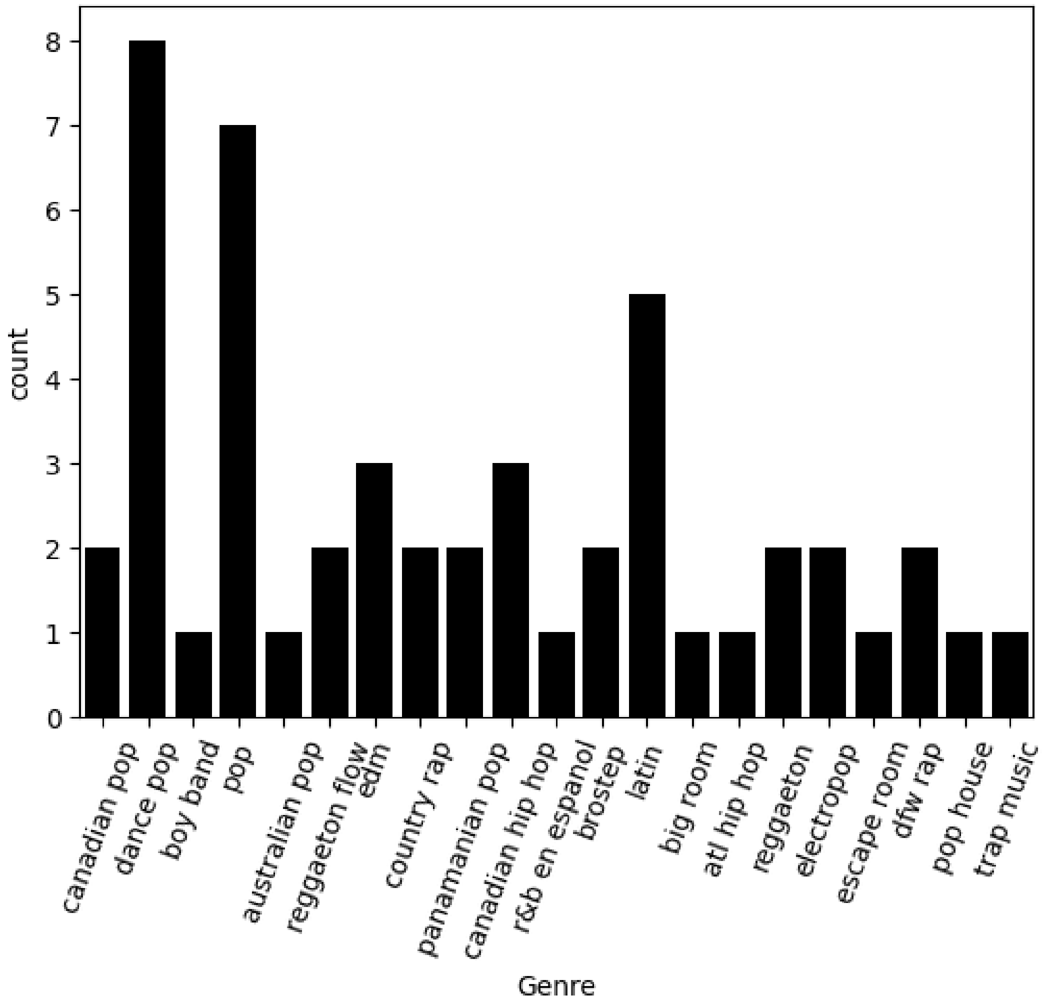
[37]:

```python
import matplotlib.pyplot as plt
import seaborn as sns
```

Exercise *How are the top 50 songs distributed across genres? (Hint: use a* `countplot`*)*

[38]:

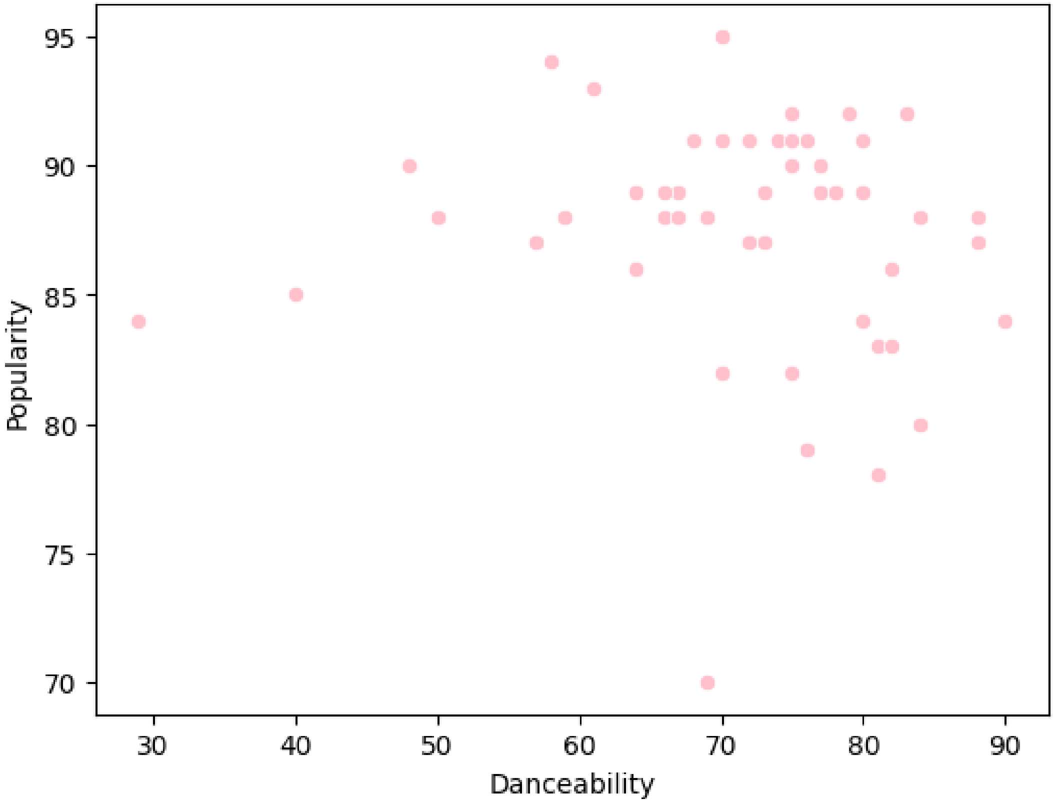
```python
# Top 50 songs
top_50 = df.sort_values(by="Popularity").head(50)
sns.countplot(x="Genre", data= top_50, color="black")

# There was some overlapping of the x-value labels, thus i Googled how to prevent that and one can rotate the labels
plt.xticks(rotation=70)

plt.show()
```



Exercise *How are **Danceability** and **Popularity** related?*

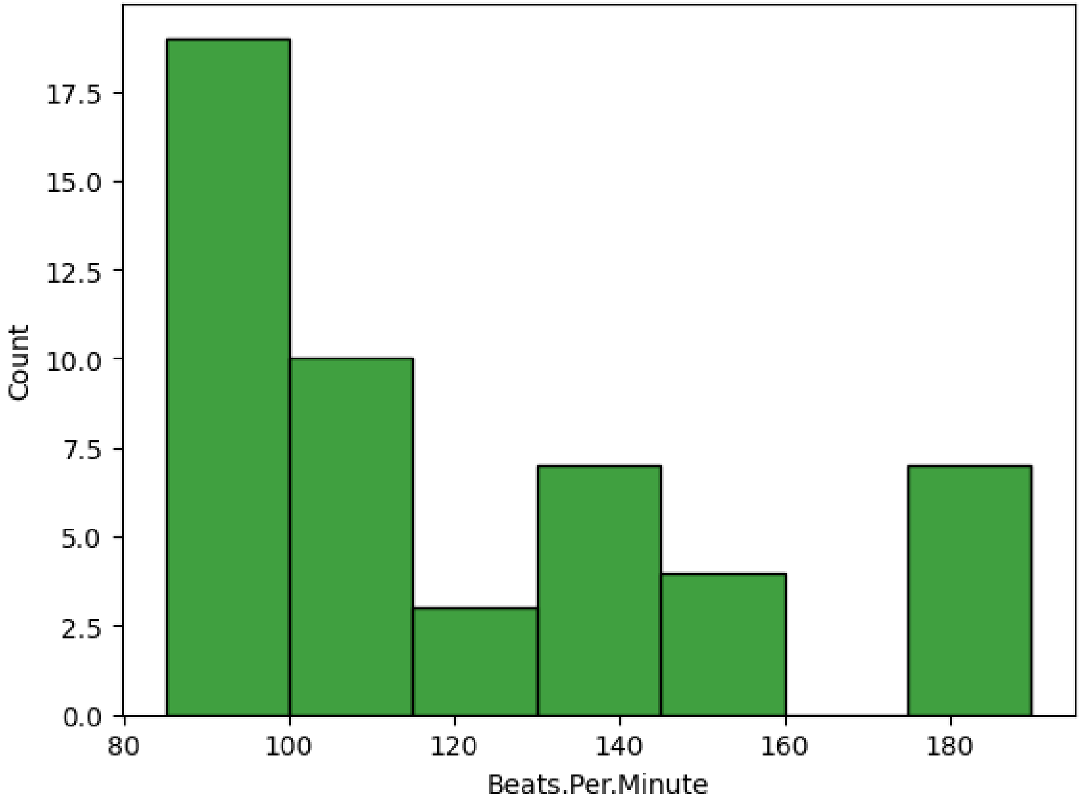
[39]:

```
sns.scatterplot(x="Danceability", y="Popularity", data=df, color="pink")
plt.show()
```



Exercise *How is* `Beats.Per.Minute` *distributed across songs?*

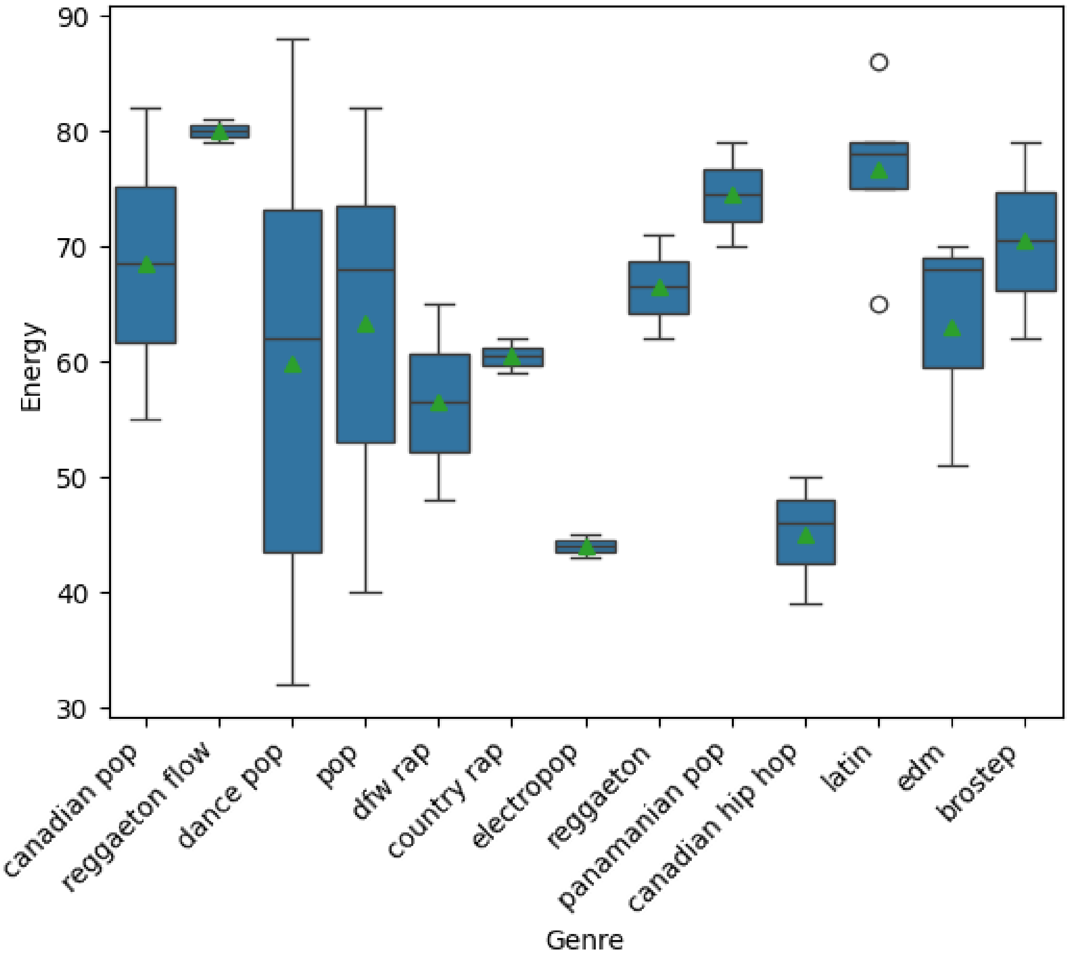
[40]:
```
sns.histplot(df["Beats.Per.Minute"], color="green")
plt.show()
```



Exercise *How do **Energy** levels vary across genres? Use a boxplot to visualize the dataset.*

[41]:
```
# filter out genres with only a single song in the dataset
multiple_songs_per_genre = df.groupby('Genre').filter(lambda x: len(x) > 1)


plot = sns.boxplot(x = "Genre", y= "Energy", showmeans=True,␣
 ↪data=multiple_songs_per_genre)
plot.set_xticklabels(plot.get_xticklabels(), rotation=45,␣
 ↪horizontalalignment='right')
plt.show()
```

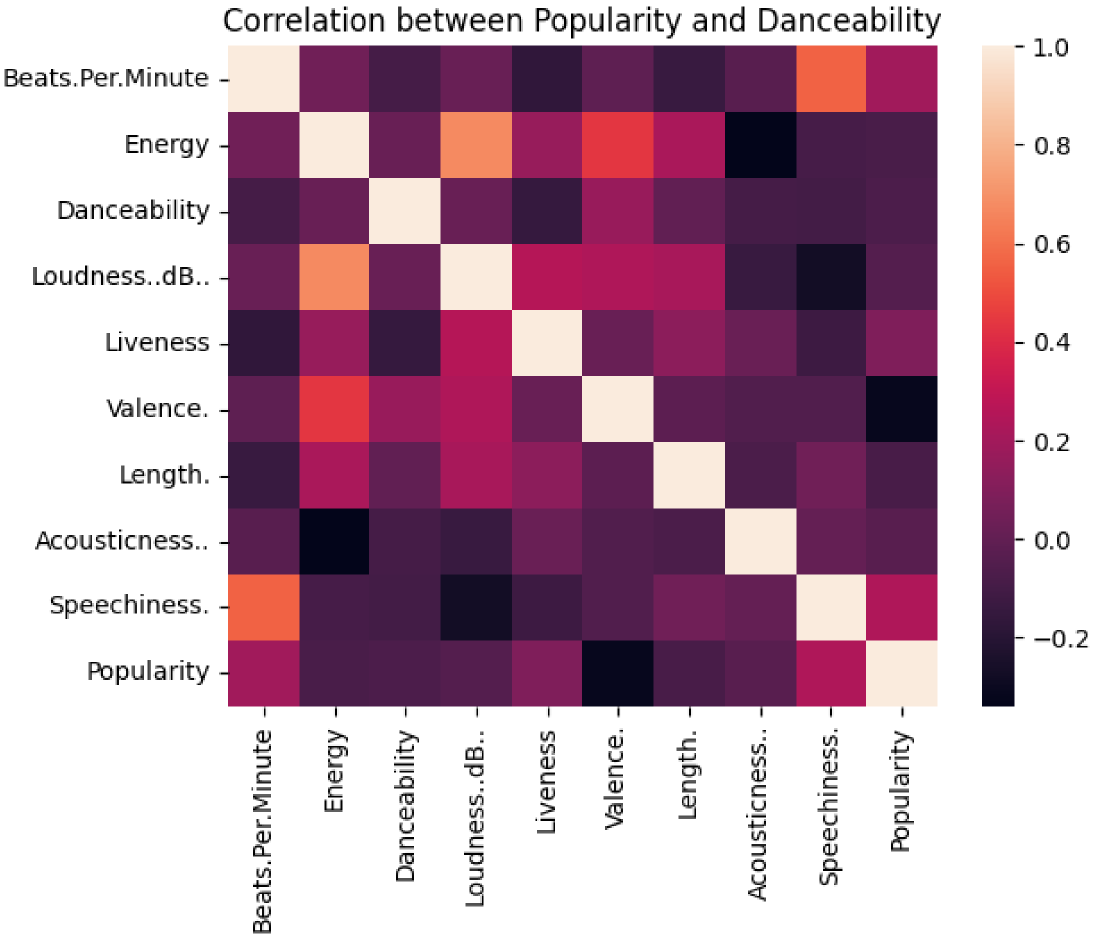
```
/var/folders/wn/61694zz176n_dm0b5c1stkv40000gn/T/ipykernel_83775/91947770.py:6:
UserWarning: set_ticklabels() should only be used with a fixed number of ticks,
i.e. after set_ticks() or using a FixedLocator.
  plot.set_xticklabels(plot.get_xticklabels(), rotation=45,
horizontalalignment='right')
```



5 4- Matplotlib

Exercise *Is there a correlation between* ***Popularity*** *and* ***Danceability****? Use a heatmap to visualize the dataset. Begin by excluding non-numerical data. Hint: use* ***df.select_dtypes()*** *and* ***.corr()***

[42]:
```
# np.number represents numerical data in the array
numerical_data = df.select_dtypes(include= np.number)

# Next calculate the correlation matrix, excluding non-numerical data
correlation = numerical_data.corr()

# Create a heatmap with sns
sns.heatmap(correlation)

plt.title("Correlation between Popularity and Danceability")
plt.show()
```



Exercise (bonus): Experiment data visualisation other datasets! We have provided some additional ones, but feel free to find your own (e.g. on Kaggle.com)

6 5- Exploration of Penguins

6.0.1 Penguins Dataset

In this exercise we are using the Penguins dataset from the `seaborn` package.

Feel free to explore other datasets, too.

[43]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# list all the available datasets via the seaborn package
sns.get_dataset_names()
```

```
[43]: ['anagrams',
       'anscombe',
       'attention',
       'brain_networks',
       'car_crashes',
       'diamonds',
       'dots',
       'dowjones',
       'exercise',
       'flights',
       'fmri',
       'geyser',
       'glue',
       'healthexp',
       'iris',
       'mpg',
       'penguins',
       'planets',
       'seaice',
       'taxis',
       'tips',
       'titanic']
```

```
[44]: penguins = sns.load_dataset('penguins')
      penguins.head(10)
```

```
[44]:   species     island  bill_length_mm  bill_depth_mm  flipper_length_mm  \
      0  Adelie  Torgersen            39.1           18.7              181.0
      1  Adelie  Torgersen            39.5           17.4              186.0
      2  Adelie  Torgersen            40.3           18.0              195.0
      3  Adelie  Torgersen             NaN            NaN                NaN
      4  Adelie  Torgersen            36.7           19.3              193.0
      5  Adelie  Torgersen            39.3           20.6              190.0
      6  Adelie  Torgersen            38.9           17.8              181.0
      7  Adelie  Torgersen            39.2           19.6              195.0
      8  Adelie  Torgersen            34.1           18.1              193.0
      9  Adelie  Torgersen            42.0           20.2              190.0

         body_mass_g     sex
      0       3750.0    Male
      1       3800.0  Female
      2       3250.0  Female
      3          NaN     NaN
      4       3450.0  Female
      5       3650.0    Male
      6       3625.0  Female
      7       4675.0    Male
```

```
8       3475.0     NaN
9       4250.0     NaN
```

[45]:
```
penguins.shape
```

[45]:
```
(344, 7)
```

[46]:
```
penguins.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   species            344 non-null    object
 1   island             344 non-null    object
 2   bill_length_mm     342 non-null    float64
 3   bill_depth_mm      342 non-null    float64
 4   flipper_length_mm  342 non-null    float64
 5   body_mass_g        342 non-null    float64
 6   sex                333 non-null    object
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

As you can see in the previous cell, there are some `NaN` values in the dataset (i.e. the total of non-null counts is smaller than the total of entries)

> ***Exercise:*** *remove the null values* (hint: use `pandas.DataFrame.dropna()`)

[47]:
```
# penguins output without NaN values
penguins.dropna()
```

[47]:
```
    species     island  bill_length_mm  bill_depth_mm  flipper_length_mm  \
0    Adelie  Torgersen            39.1           18.7              181.0
1    Adelie  Torgersen            39.5           17.4              186.0
2    Adelie  Torgersen            40.3           18.0              195.0
4    Adelie  Torgersen            36.7           19.3              193.0
5    Adelie  Torgersen            39.3           20.6              190.0
..      …        …              …             …                 …
338  Gentoo     Biscoe            47.2           13.7              214.0
340  Gentoo     Biscoe            46.8           14.3              215.0
341  Gentoo     Biscoe            50.4           15.7              222.0
342  Gentoo     Biscoe            45.2           14.8              212.0
343  Gentoo     Biscoe            49.9           16.1              213.0

     body_mass_g     sex
0         3750.0    Male
1         3800.0  Female
2         3250.0  Female
```

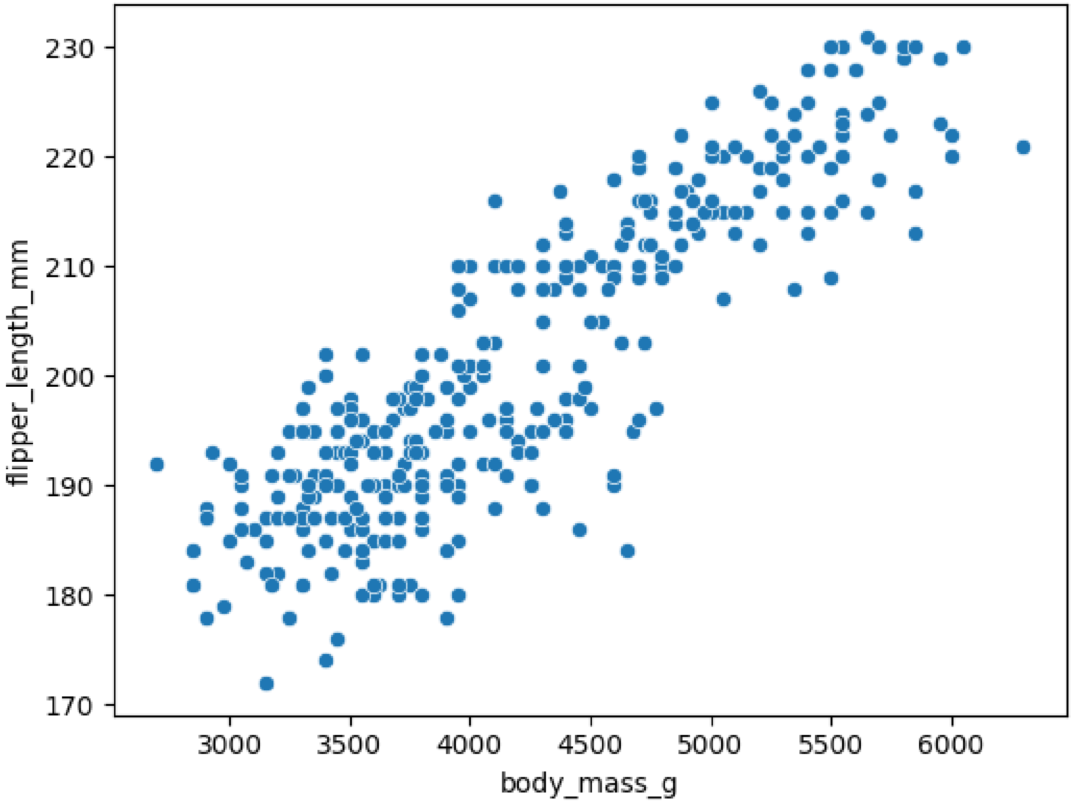
```
4          3450.0  Female
5          3650.0    Male
..            ...     ...
338        4925.0  Female
340        4850.0  Female
341        5750.0    Male
342        5200.0  Female
343        5400.0    Male

[333 rows x 7 columns]
```

6.0.2 Visualizations

Exercise: *How are `body_mass_g` and `flipper_length_mm` related?*

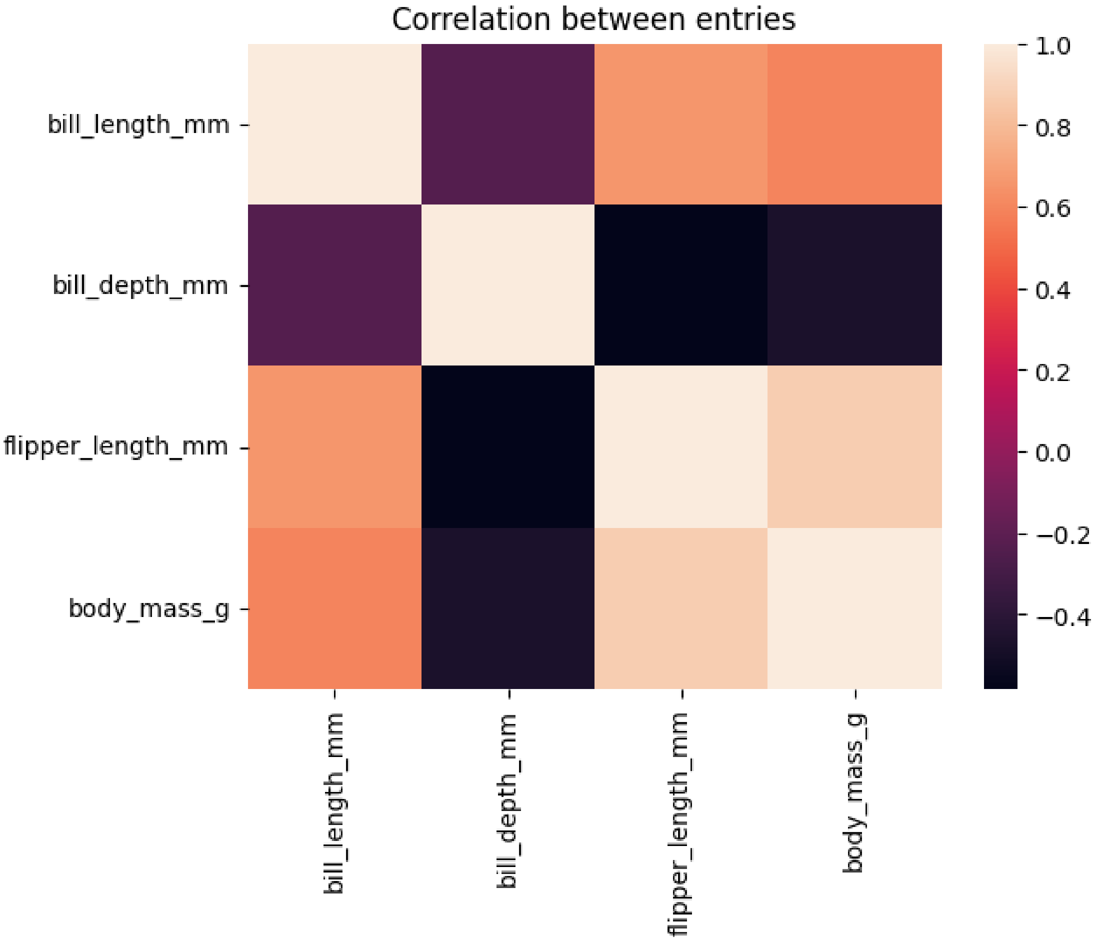
[48]:

```python
# I tried using hue, yet it raised an Error and I don't understand why
sns.scatterplot(x="body_mass_g", y="flipper_length_mm", data=penguins)
plt.show()
```



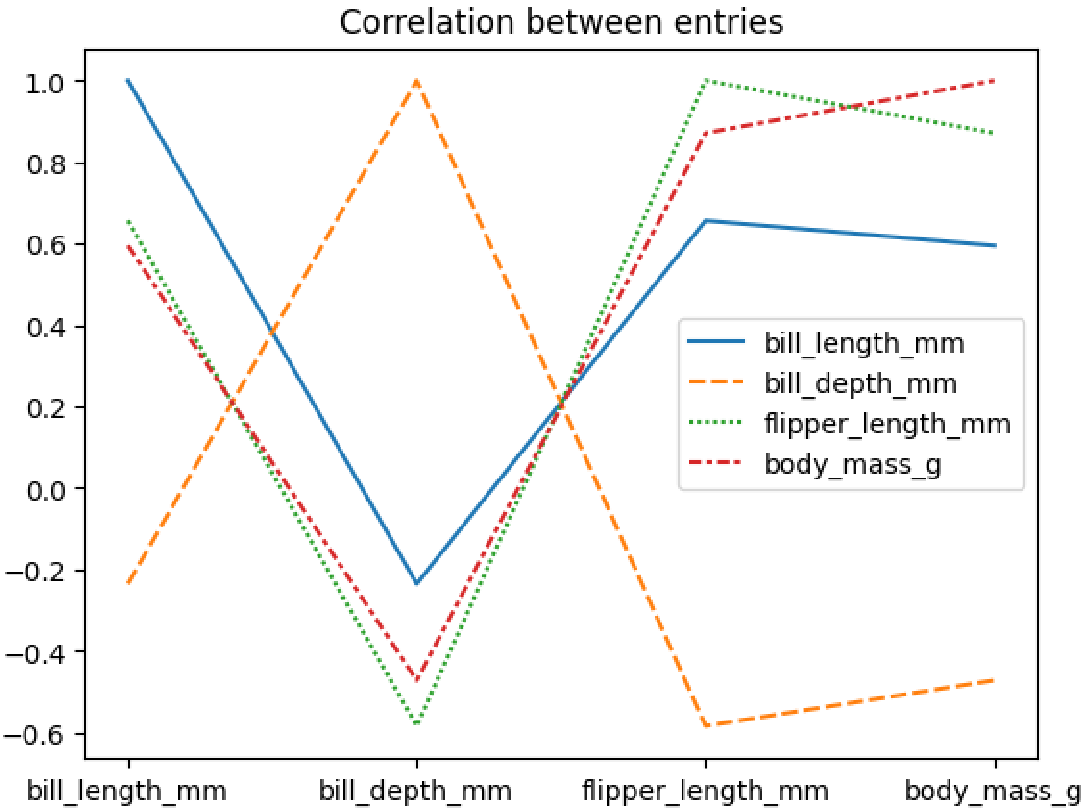
Exercise: *Experiment with other plots to reveal something intersting about the dataset!*

[49]:
```
# np.number represents numerical data in the array
numerical_data = penguins.select_dtypes(include= np.number)

# Next calculate the correlation matrix, excluding non-numerical data
correlation = numerical_data.corr()

# Create a heatmap with sns
sns.heatmap(correlation)

plt.title("Correlation between entries")
plt.show()
```

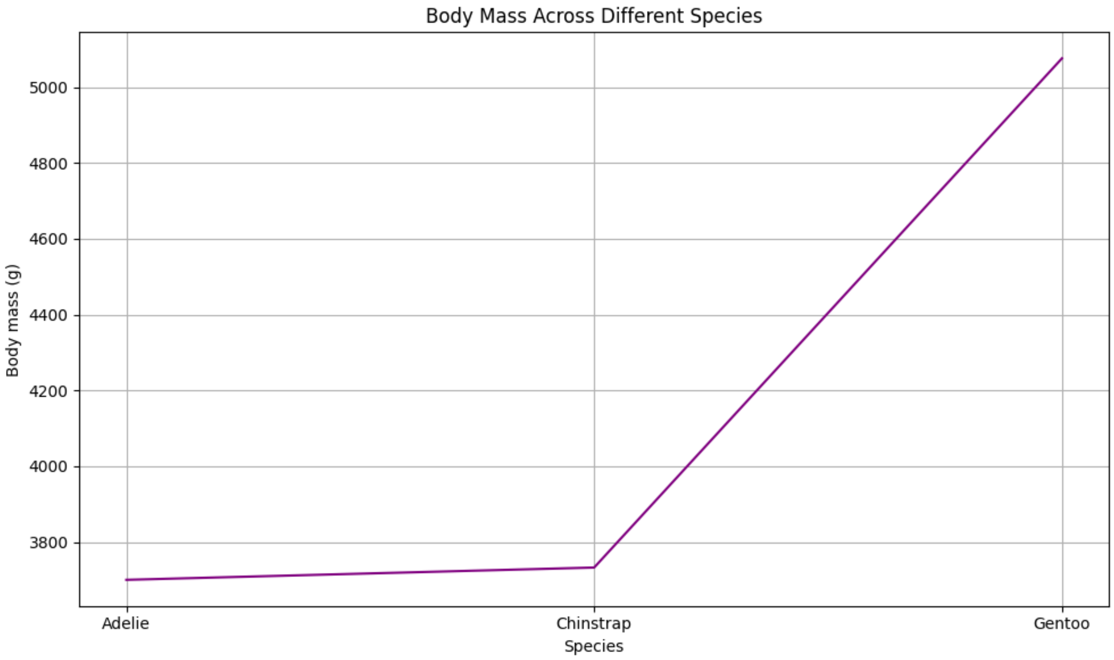


[50]:
```
# np.number represents numerical data in the array
numerical_data = penguins.select_dtypes(include= np.number)

# Next calculate the correlation matrix, excluding non-numerical data
correlation = numerical_data.corr()
```

```
# Create a heatmap with sns
sns.lineplot(correlation)

plt.title("Correlation between entries")
plt.show()
```



[51]:
```
plt.figure(figsize=(10, 6))
sns.lineplot(data=penguins, x='species', y='body_mass_g', errorbar=None,␣
 ↪color="purple")  # ci=None removes confidence intervals
plt.title('Body Mass Across Different Species')
plt.xlabel('Species')
plt.ylabel('Body mass (g)')
plt.grid(True)
plt.tight_layout()
plt.show()
```



7 6- Playing around with BERT

Import: *import the necessary libraries*

[52]:
```
from transformers import pipeline
```

Instantiate the Pipeline: *define your classifier and task*

[53]:
```
classifier = pipeline("sentiment-analysis", model="roberta-base")
```

```
Some weights of RobertaForSequenceClassification were not initialized from the
model checkpoint at roberta-base and are newly initialized:
['classifier.dense.bias', 'classifier.dense.weight', 'classifier.out_proj.bias',
'classifier.out_proj.weight']
You should probably TRAIN this model on a down-stream task to be able to use it
for predictions and inference.
```

Perform Sentiment Analysis: *Enter a piece of text that you want to analyze for sentiment*

[54]:
```
result = classifier("Alice was beginning to get very tired of sitting by her␣
  ↪sister on the bank, and of having nothing to do.")
result
```

[54]:
```
[{'label': 'LABEL_1', 'score': 0.511226236820221}]
```

7.0.1 Ryanair Reviews Analysis

Introduction This Jupyter Notebook is designed to guide you through analyzing customer reviews of Ryanair flights. We will perform various NLP tasks to extract insights from textual data and explore the relationships between different numeric variables.

Data Loading Let's start by loading the necessary libraries and the dataset.

```
[55]: ryanair_reviews_df = pd.read_csv("ryanair_reviews.csv")

      # Display the first few rows of the dataset to understand its structure and␣
       ↪contents
      # TODO
      ryanair_reviews_df.head(10)
```

```
[55]:    Unnamed: 0 Date Published  Overall Rating Passenger Country  Trip_verified  \
      0           0     2024-02-03            10.0    United Kingdom   Not Verified
      1           1     2024-01-26            10.0    United Kingdom  Trip Verified
      2           2     2024-01-20            10.0    United Kingdom  Trip Verified
      3           3     2024-01-07             6.0    United Kingdom  Trip Verified
      4           4     2024-01-06            10.0            Israel  Trip Verified
      5           5     2024-01-06             1.0           Denmark   Not Verified
      6           6     2024-01-03             5.0    United Kingdom   Not Verified
      7           7     2024-01-03             1.0         Australia  Trip Verified
      8           8     2023-12-25             1.0    United Kingdom  Trip Verified
      9           9     2023-12-08             1.0           Germany   Not Verified

                                     Comment title  \
      0         "bang on time and smooth flights"
      1          "Another good affordable flight"
      2                       "Really impressed!"
      3          "a decent offering from Ryanair"
      4   "cabin crew were welcoming and friendly"
      5      "close online checkin 3 hours before"
      6         "they are really not better value"
      7         "asked me to pay for the backpack"
      8       "ground service staff is really bad"
      9          "they made us pay a No show fee"

                                                   Comment         Aircraft  \
      0  Flew back from Faro to London Luton Friday 2nd…   Boeing 737 900
      1  Another good affordable flight with Ryanair. O…              NaN
      2  Really impressed! You get what you pay for, th…   Boeing 737-800
      3  I should like to review my flight from Faro to…       Boeing 737
      4  Flight left the gate ahead of schedule, fare w…   Boeing 737-800
      5  Booked a fight from Copenhagen to Poland thoug…              NaN
      6  The flight itself is operated by Malta air and…       Boeing 737
      7  Staff is rude and has no manners, let alone be…              NaN
```

```
8  Ryanair ground service staff is really bad. If…            NaN
9  I wanted to check in online a night before our…            NaN

  Type Of Traveller      Seat Type  …        Destination     Date Flown  \
0    Family Leisure  Economy Class  …              Luton  February 2024
1    Couple Leisure  Economy Class  …           Alicante   January 2024
2    Couple Leisure  Economy Class  …     Paris Beauvais   October 2023
3      Solo Leisure  Economy Class  …          Liverpool   January 2024
4      Solo Leisure  Economy Class  …         Manchester   January 2024
5      Solo Leisure  Economy Class  …             Gdansk   January 2024
6          Business  Economy Class  …               Pisa  December 2023
7      Solo Leisure  Economy Class  …          Barcelona   January 2024
8    Family Leisure  Economy Class  …             Tirana  December 2023
9    Couple Leisure  Economy Class  …  Palma de Mallorca  November 2023

  Seat Comfort  Cabin Staff Service  Food & Beverages  Ground Service  \
0          4.0                  5.0               3.0             4.0
1          3.0                  5.0               3.0             5.0
2          5.0                  5.0               4.0             5.0
3          3.0                  2.0               1.0             3.0
4          4.0                  5.0               NaN             4.0
5          2.0                  2.0               2.0             1.0
6          2.0                  5.0               2.0             1.0
7          NaN                  NaN               NaN             1.0
8          1.0                  NaN               NaN             1.0
9          1.0                  1.0               NaN             1.0

   Value For Money  Recommended Inflight Entertainment  Wifi & Connectivity
0              4.0          yes                    NaN                  NaN
1              5.0          yes                    NaN                  NaN
2              5.0          yes                    NaN                  NaN
3              3.0          yes                    NaN                  NaN
4              5.0          yes                    NaN                  NaN
5              1.0           no                    2.0                  2.0
6              1.0          yes                    NaN                  NaN
7              1.0           no                    NaN                  NaN
8              1.0           no                    NaN                  NaN
9              1.0           no                    NaN                  NaN

[10 rows x 21 columns]
```

7.0.2 Data Cleaning and Preprocessing

In this section, we will prepare the data for analysis by cleaning and preprocessing it. We will perform the following tasks:

1. Convert data types if necessary to ensure correct data formats for analysis, e.g convert dates (using `.to_datetime`)

2. We need to filter out rows where the columns might be too long to process. BERT can process a max_len of 512 tokens…

   Convert Data Types: *to ensure correct data types for analysis* (Here just an example of what you might need in real world applications)

[56]:

```
# Convert data types
ryanair_reviews_df['Date Published'] = pd.to_datetime(ryanair_reviews_df['Date␣
 ↪Published'])
ryanair_reviews_df['Date Flown'] = pd.to_datetime(ryanair_reviews_df['Date␣
 ↪Flown'], format='%B %Y', errors='coerce')
```

7.0.3 Preprocessing Text Length: why do we need to pay attention to this?

Language models like BERT, RoBERTa, and GPT-2 have limitations on the maximum sequence length they can process due to their tokenization methods. Effective preprocessing of text length is crucial to ensure the data is compatible with these limits, which improves computational efficiency and model performance.

7.0.4 Practical Strategy for Our Case: Dropping Long Texts

For simplicity, we will drop rows where texts exceed a certain length. This approach ensures all input data fits the model's constraints without the need for complex preprocessing steps like truncation or segmentation. However, depending on the context, other strategies like segmenting long texts into smaller parts or dynamically batching texts of varying lengths could also be considered to preserve information and enhance processing.

7.0.5 Understanding Tokenization with BERT, RoBERTa, and GPT-2

To ensure clarity in our examples and practical application of tokenization methods, let's consider how text length changes when tokenized using different models such as BERT, RoBERTa, and GPT-2.

BERT (Bidirectional Encoder Representations from Transformers) and **RoBERTa (Robustly Optimized BERT Approach)** use a WordPiece tokenization mechanism. In contrast, **GPT-2 (Generative Pre-trained Transformer 2)** employs a byte pair encoding (BPE) tokenization. Let's see how these tokenization methods affect text length.

Example Sentence: "Quick brown foxes leap over lazy dogs multiple times."

BERT Tokenization:

- Pre-tokenization: "Quick brown foxes leap over lazy dogs multiple times."
- Post-tokenization: `[CLS] Quick brown fox ##es leap over lazy dogs multiple times [SEP]`
- Token count: 12 tokens

RoBERTa Tokenization:

- Pre-tokenization: "Quick brown foxes leap over lazy dogs multiple times."

- Post-tokenization: `<s> Quick brown foxes leap over lazy dogs multiple times </s>`
- Token count: 11 tokens

GPT-2 Tokenization:

- Pre-tokenization: “Quick brown foxes leap over lazy dogs multiple times.”
- Post-tokenization: `Quick Ġbrown Ġfoxes Ġleap Ġover Ġlazy Ġdogs Ġmultiple Ġtimes`
- Token count: 9 tokens

7.0.6 Notes:

1. [**CLS**] and [**SEP**] are special tokens used by BERT to mark the beginning and end of a sentence. RoBERTa uses and as its special boundary tokens.
2. The difference in token count is due to the various subword divisions by each model’s tokenization algorithm.
3. Subword tokenization helps in handling unknown words more effectively by breaking them down into meaningful sub-units.

To ensure no issues in our examples, let’s filter out the longer comments. BERT can only process sequences with a max length of 512. As we’ve just discussed, we need to account for extra length after tokenization. Feel free to experiment with different lengths to find what works best without having to sacrifice too much data. We suggest starting with a length of 200. You can check the number of rows in the original dataframe vs the filtered dataframe to see how many texts you’ve lost:

here’s how you might do that:

```
# Check the number of rows before and after filtering
original_count = len(df)
filtered_count = len(filtered_df)

print(f'Original DataFrame size: {original_count}')
print(f'Filtered DataFrame size: {filtered_count}')
print(f'Number of texts lost: {original_count - filtered_count}')
```

Filter Out Longer Texts from the Dataframe*

[57]:
```
# function to count the number of words in a string
def word_count(string):
    return len(string.split())

# apply the function to the 'Comment' column of the dataframe, to apply a␣
 ↪function to each row of the column we use the apply method
ryanair_reviews_df['word_count'] = ryanair_reviews_df['Comment'].
 ↪apply(word_count)

# now filter out the rows where the word count is greater than 200, hint: you␣
 ↪might want to create a new dataframe to store the filtered rows
# TODO
filtered_df = ryanair_reviews_df[ryanair_reviews_df['word_count'] < 200]
```

```
# Check the number of rows before and after filtering
original_count = len(ryanair_reviews_df)
filtered_count = len(filtered_df)

print(f'Original DataFrame size: {original_count}')
print(f'Filtered DataFrame size: {filtered_count}')
print(f'Number of texts lost: {original_count - filtered_count}')
```

```
Original DataFrame size: 2249
Filtered DataFrame size: 1885
Number of texts lost: 364
```

7.0.7 Exploratory Data Analysis (EDA)

Now that our data is clean, let's explore it to uncover some initial insights:

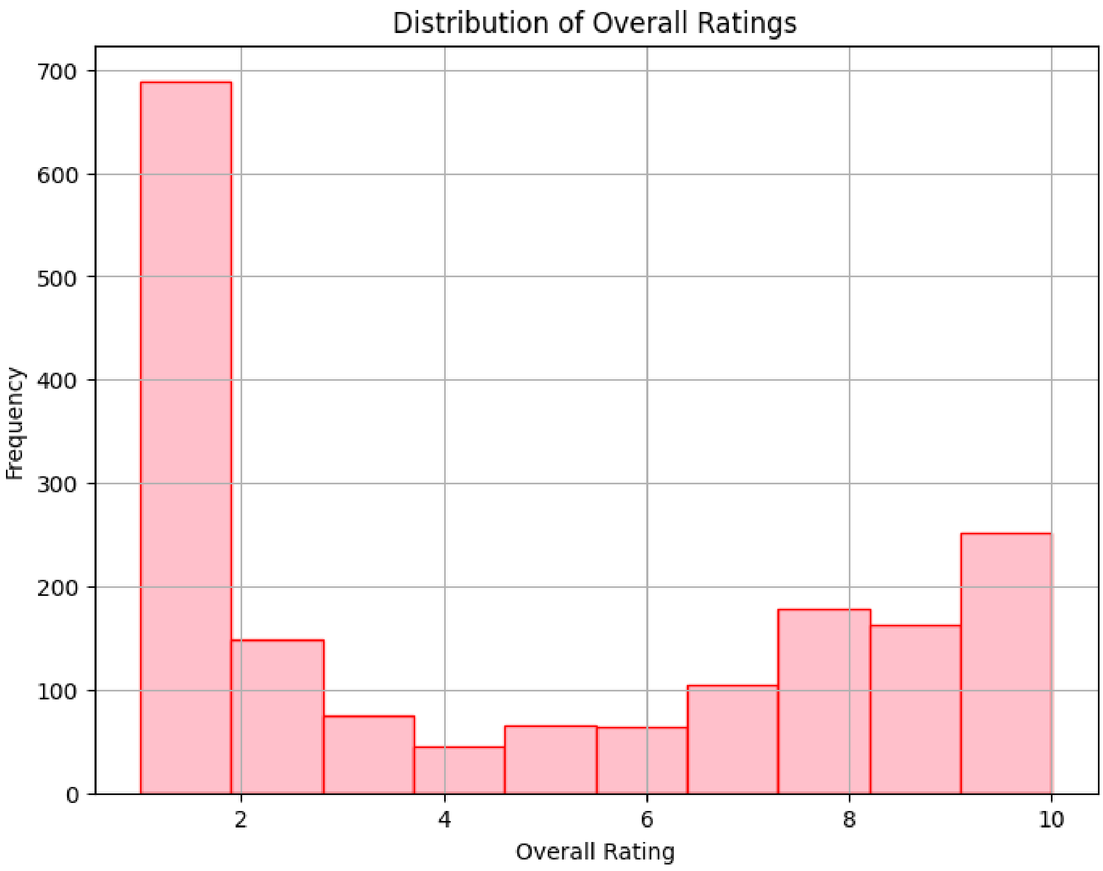
1. Analyze the distribution of overall ratings to see how passengers generally feel about Ryanair.
2. Investigate the frequency of different types of travellers and their experiences.

[58]:

```
#import matplotlib.pyplot as plt       uncomment if for some reason you didnt␣
 ↪import before
#import seaborn as sns
```

[59]:

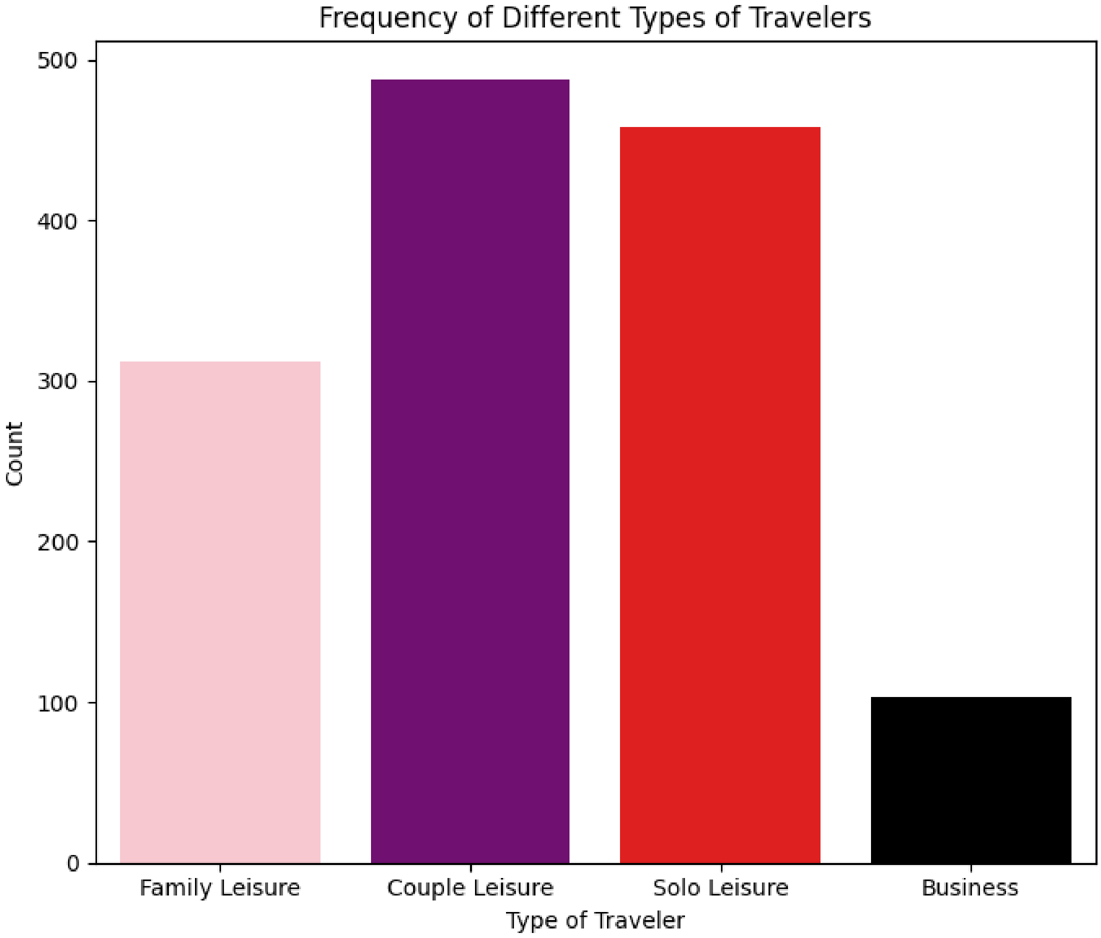
```
# Analyze the distribution of overall ratings in the dataset with a histogram
# TODO
plt.figure(figsize=(8, 6))
plt.hist(filtered_df['Overall Rating'], bins=10, color='pink', edgecolor='red')
plt.title('Distribution of Overall Ratings')
plt.xlabel('Overall Rating')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



[60]:
```
# Investigate the frequency of different types of travellers using a countplot␣
↪(countplots are used to show the counts of observations in each categorical␣
↪bin using bars)
# TODO
plt.figure(figsize=(7, 6))
sns.countplot(data=filtered_df, x='Type Of Traveller', palette=['pink',␣
↪'purple', 'red', 'black'])
plt.title('Frequency of Different Types of Travelers')
plt.xlabel('Type of Traveler')
plt.ylabel('Count')
plt.tight_layout()
plt.show()
```

```
/var/folders/wn/6l694zz176n_dm0b5c1stkv40000gn/T/ipykernel_83775/4080108712.py:4
: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.
```

```
  sns.countplot(data=filtered_df, x='Type Of Traveller', palette=['pink',
'purple', 'red', 'black'])
```



7.0.8 Text-based NLP Tasks

We will now apply various NLP techniques to analyze the text data from the comments:

1. **Sentiment Analysis**: Determine the sentiment expressed in the comments.
2. **Named Entity Recognition (NER)**: Extract names, places, and other entities from the comments.
3. **Text Summarization**: Summarize longer comments to get quick insights.
4. **Text Generation**: Generate follow-up comments based on the originals.

We're using models from the Huggingface website. Feel free to explore and try out different ones. https://huggingface.co/

Disclaimer: The larger (and usually better) the model, the longer it will take to load. Some will most likely not run on your computers. If it's taking too long (more than a few minutes), try a different model.

Some other pipelines to try out: `ner = pipeline("ner", model="")`,

`summarizer = pipeline("summarization", model="")`,

`text_generator = pipeline("text-generation", model="")`

[61]:
```
#from transformers import pipeline

# Initialize NLP pipelines with specified models (we already ran this earlier,␣
↪but just to show you how to do it)
classifier = pipeline("sentiment-analysis", model="roberta-base")
```

```
Some weights of RobertaForSequenceClassification were not initialized from the
model checkpoint at roberta-base and are newly initialized:
['classifier.dense.bias', 'classifier.dense.weight', 'classifier.out_proj.bias',
'classifier.out_proj.weight']
You should probably TRAIN this model on a down-stream task to be able to use it
for predictions and inference.
```

[62]:
```
# Apply the sentiment analysis model to the 'Comment' column of the dataframe␣
↪to get the sentiment of each review, hint you can use the apply method and a␣
↪lambda function
# TODO
filtered_df["Sentiment"] = filtered_df['Comment'].apply(lambda x: classifier(x))
```

```
/var/folders/wn/61694zz176n_dm0b5c1stkv40000gn/T/ipykernel_83775/2732592695.py:3
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_df["Sentiment"] = filtered_df['Comment'].apply(lambda x:
classifier(x))
```

[63]:
```
ner = pipeline("ner", model="mdarhri00/named-entity-recognition")
```

[64]:
```
# Named Entity Recognition
filtered_df['entities'] = filtered_df['Comment'].apply(lambda x:␣
↪[entity['word'] for entity in ner(x)])
print(filtered_df['entities'])
```

```
0        [Faro, London, Lu, ##ton, Friday, 2nd, Februar…
1                                   [Ryan, ##air, Ryan, ##air]
2                                                           []
3        [Faro, Liverpool, Ryan, ##air, Ryan, ##air, mo…
4                         [A, ##er, Ling, ##us, Ryan, ##air]
                                     …
2243                                              [Manchester]
2245                                  [P, ##ula, Ryan, ##air]
```

```
2246                               [check, in, lady, Malta]
2247    [Budapest, -, Manchester, and, back, 5, month,…
2248                               [Barcelona, Ryan, ##air]
Name: entities, Length: 1885, dtype: object
```

```
/var/folders/wn/6l694zz176n_dm0b5c1stkv40000gn/T/ipykernel_83775/601807711.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_df['entities'] = filtered_df['Comment'].apply(lambda x:
[entity['word'] for entity in ner(x)])
```

7.0.9 How to apply the other pipelines

Named Entity Recognition (NER) `filtered_df['entities'] = filtered_df['Comment'].apply(lambda x: [entity['word'] for entity in ner(x)])`

Text Summarization `filtered_df['summary'] = filtered_df['Comment'].apply(lambda x: summarizer(x, max_length=50, min_length=10, do_sample=False)[0]['summary_text'] if len(x.split()) > 30 else x)`

Text Generation `filtered_df['generated_text'] = filtered_df['Comment'].apply(lambda x: text_generator(x, max_length=50, do_sample=False)[0]['generated_text'])`

Exercise: *print the output of the Sentiment column*

[65]:
```
#TODO
print(filtered_df['Sentiment'])
```

```
0       [{'label': 'LABEL_0', 'score': 0.5557076930999…
1       [{'label': 'LABEL_0', 'score': 0.5575989484786…
2       [{'label': 'LABEL_0', 'score': 0.5576185584068…
3       [{'label': 'LABEL_0', 'score': 0.5564654469490…
4       [{'label': 'LABEL_0', 'score': 0.5586391687393…
                                  …
2243    [{'label': 'LABEL_0', 'score': 0.5535437464714…
2245    [{'label': 'LABEL_0', 'score': 0.5557156801223…
2246    [{'label': 'LABEL_0', 'score': 0.5546550154685…
2247    [{'label': 'LABEL_0', 'score': 0.5556036829948…
```

```
2248    [{'label': 'LABEL_0', 'score': 0.5545305013656…
Name: Sentiment, Length: 1885, dtype: object
```

> ***Exercise:*** *sample random rows from the dataframe and compare the* `Comment` *and* `sentiment` *column by selecting only them*

```
[66]: #TODO
      random_sample = filtered_df.sample(n=5)
      print(random_sample[['Comment', 'Sentiment']])
```

```
                                                Comment  \
1581  Flying with Ryanair, you get exactly what you …
951   Palma to Dublin. Ryanair does a great job maki…
1259  Stansted to Poznań with Ryanair. I usually pic…
736   Malta to Stansted. Never again Ryanair. I real…
301   I was not allowed on the Frankfurt - Stansted …

                                              Sentiment
1581  [{'label': 'LABEL_0', 'score': 0.5582830905914…
951   [{'label': 'LABEL_0', 'score': 0.5541141033172…
1259  [{'label': 'LABEL_0', 'score': 0.5583271384239…
736   [{'label': 'LABEL_0', 'score': 0.5585727691650…
301   [{'label': 'LABEL_0', 'score': 0.5575473904609…
```

7.0.10 Visualization

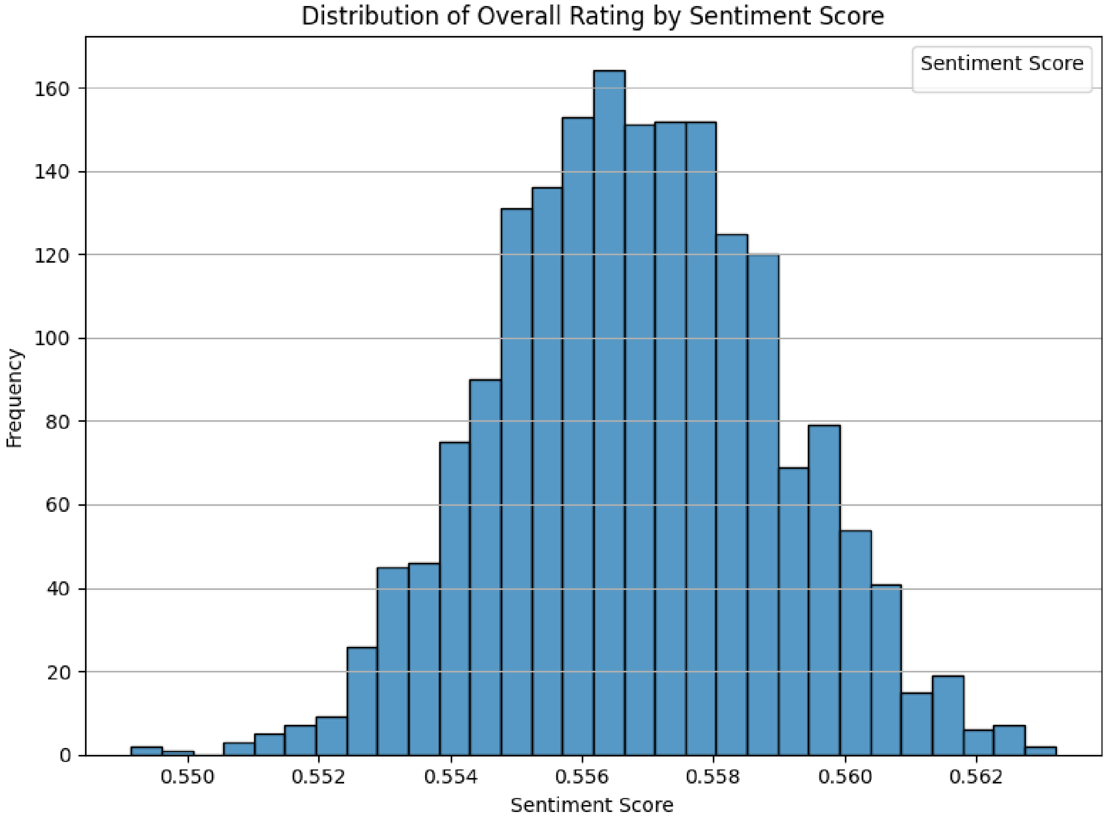
Let’s visually represent some of our findings from the NLP tasks:

1. Display sentiment analysis results.
2. Does the sentiment correlate with the Overall Rating?
3. If you run more pipelines, think of other plots.

```
[70]: #TODO
      # Extract sentiment scores from the 'Sentiment' column
      filtered_df['Sentiment Score'] = filtered_df['Sentiment'].apply(lambda x:␣
        ↪x[0]['score'])

      # Create the histogram with sentiment scores compared to Overall Rating
      plt.figure(figsize=(8, 6))
      sns.histplot(data=filtered_df, x='Sentiment Score')
      plt.title('Distribution of Overall Rating by Sentiment Score')
      plt.xlabel('Sentiment Score')
      plt.ylabel('Frequency')
      plt.legend(title='Sentiment Score')
      plt.grid(True, axis='y')  # Add gridlines for y-axis only
      plt.tight_layout()
      plt.show()
```

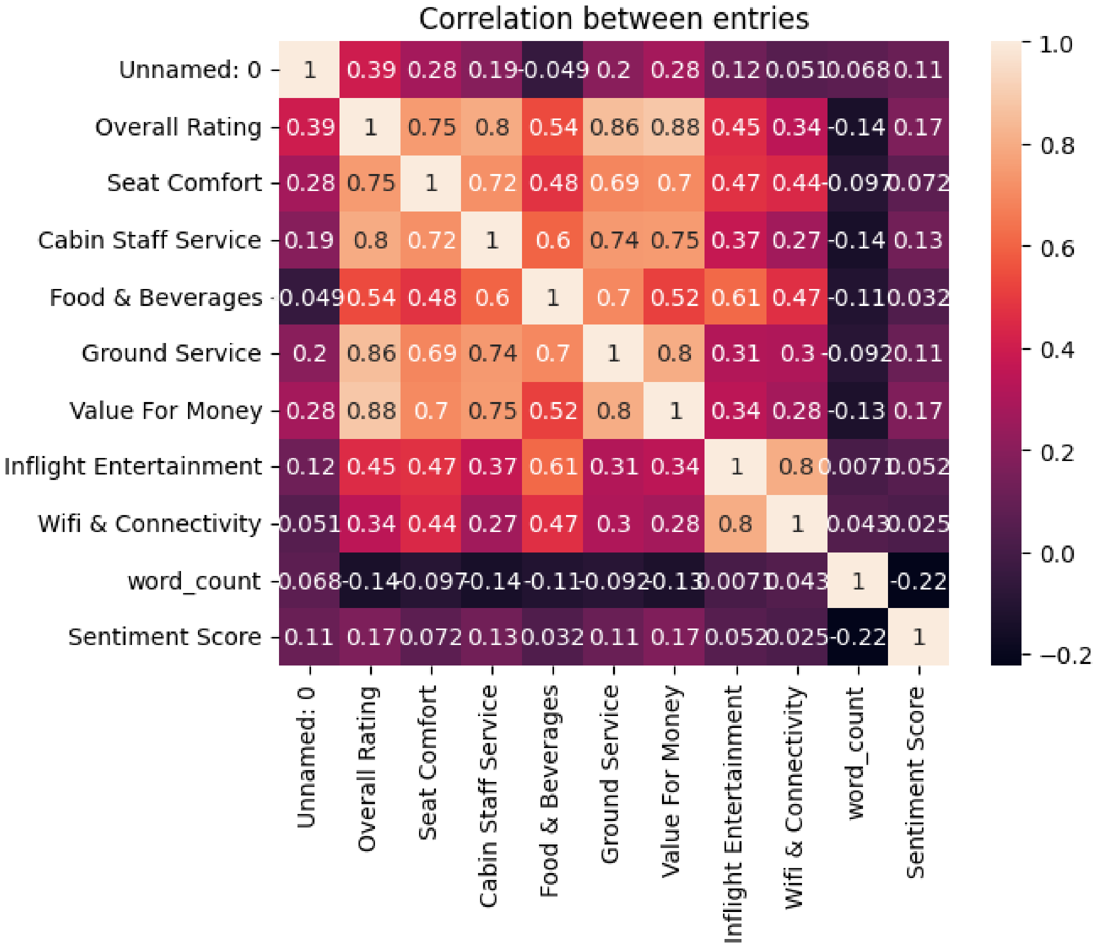
```
/var/folders/wn/6l694zz176n_dm0b5c1stkv40000gn/T/ipykernel_83775/2750175269.py:3
: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_df['Sentiment Score'] = filtered_df['Sentiment'].apply(lambda x:
x[0]['score'])
No artists with labels found to put in legend.  Note that artists whose label
start with an underscore are ignored when legend() is called with no argument.
```



[73]:

```python
# np.number represents numerical data in the array
numerical_data = filtered_df.select_dtypes(include= np.number)

# Next calculate the correlation matrix, excluding non-numerical data
correlation = numerical_data.corr()

# Create a heatmap with sns
sns.heatmap(correlation, annot=True)

plt.title("Correlation between entries")
plt.show()
```



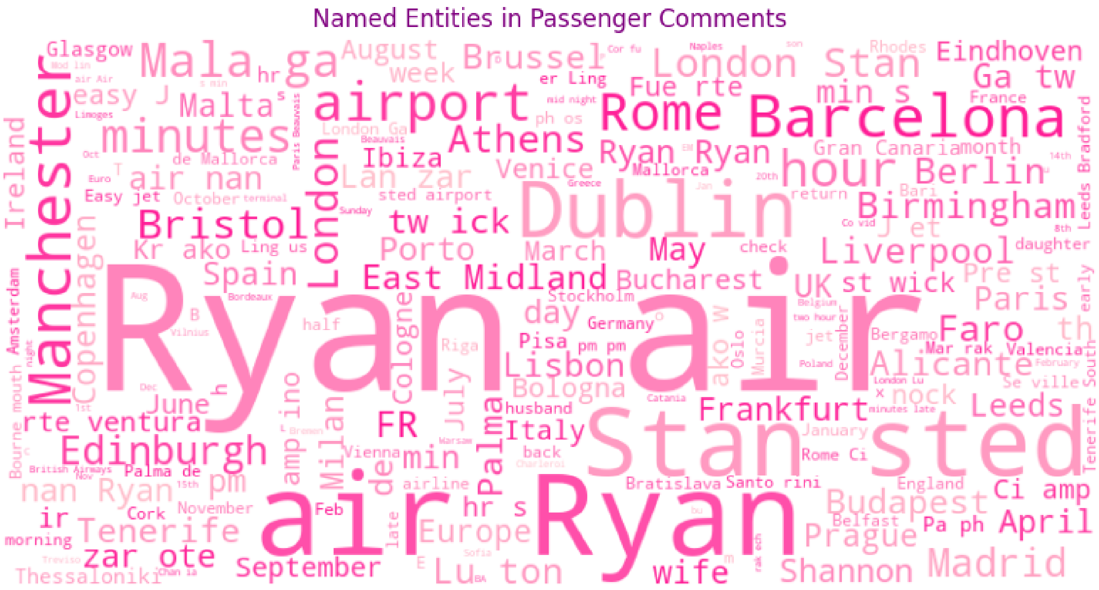
[68]:

```
from wordcloud import WordCloud
from matplotlib.colors import LinearSegmentedColormap # This module helps us change the colors of graphs
```

[69]:

```
# Let's create a word cloud of the different Named Entities in the comments

#Concatenate all named entities into a single string. This I had to google how to do it
all_entities = ' '.join(filtered_df['entities'].explode().astype(str))
# explode(): Transform each element of a list-like to a row, replicating index values.
# Then turned into a string and all are joined with a white space

# We wanted to make it a bit prettier and thus tried making it use pink shades
pink_shades = LinearSegmentedColormap.from_list('pink_cmap', ['#FFC0CB', '#FF69B4', '#FF1493']) # Here we also had to Google how to use the Module and came across a GitHub website that explained how to use it
```

```
word_cloud = WordCloud(width=800, height=400, background_color='white',␣
 ↪colormap= pink_shades).generate(all_entities)

plt.figure(figsize=(10, 5))
plt.imshow(word_cloud, interpolation='bilinear')
plt.axis('off')
plt.title('Named Entities in Passenger Comments', color='purple')
plt.show()
```



8 Have fun!

Explore the other datasets and practise what you learned in this session!