Lab_3

September 24, 2024

1 Lab 3 - Assignment 2 and related concepts

1.1 1. Import libraries

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
```

1.2 2. Read data

1.2.1 2.1 Read from GitHub link

[2]:
```
url_futures = 'https://raw.githubusercontent.com/vercammen/MFRE/master/
 ↪fre501Assign2/canola_futures.csv'
canola = pd.read_csv(url_futures)
canola.head()
```

[2]:
```
         date contract  front_old  front_new  RSF16  RSH16  RSK16  RSN16  \
0  10/16/2015    RSF16        NaN      477.1  477.1  479.8  480.5  480.0
1  10/19/2015    RSF16        NaN      476.2  476.2  478.9  479.6  479.2
2  10/20/2015    RSF16        NaN      478.3  478.3  481.6  482.0  480.9
3  10/21/2015    RSF16        NaN      482.4  482.4  486.8  488.3  487.3
4  10/22/2015    RSF16        NaN      481.2  481.2  485.2  485.8  484.0

   RSX16  RSF17  …  RSF24  RSH24  RSK24  RSN24  RSX24  RSF25  RSH25  RSK25  \
0  469.7  467.9  …    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN
1  468.6  466.8  …    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN
2  471.1  469.3  …    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN
3  477.2  475.4  …    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN
4  473.4  471.6  …    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN

   RSN25  RSX25
0    NaN    NaN
1    NaN    NaN
2    NaN    NaN
3    NaN    NaN
4    NaN    NaN
```

[5 rows x 54 columns]

1.2.2 2.2 Read from Colab

[3]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[4]:
```
pd.read_csv("/content/drive/MyDrive/data/wheat_historical.csv").head()
```

[4]:

	Time	Open	High	Low	Last	Change	%Chg	Volume	Open Int
0	08/20/2024	551.0	559.25	549.5	556.50	4.25	0.77%	112067	415877
1	08/19/2024	554.5	554.50	545.0	552.25	-0.25	-0.05%	125305	416853
2	08/16/2024	549.0	555.50	545.5	552.50	2.25	0.41%	123773	416551
3	08/15/2024	557.5	568.00	548.5	550.25	-6.00	-1.08%	161435	412639
4	08/14/2024	553.0	559.25	547.5	556.25	4.50	0.82%	201597	410420

1.3 3. Convert “date” column to “date” datatype

[5]:
```
canola['date'] = pd.to_datetime(canola['date'])
```

[6]:
```
canola.head()
```

[6]:

	date	contract	front_old	front_new	RSF16	RSH16	RSK16	RSN16	\
0	2015-10-16	RSF16	NaN	477.1	477.1	479.8	480.5	480.0	
1	2015-10-19	RSF16	NaN	476.2	476.2	478.9	479.6	479.2	
2	2015-10-20	RSF16	NaN	478.3	478.3	481.6	482.0	480.9	
3	2015-10-21	RSF16	NaN	482.4	482.4	486.8	488.3	487.3	
4	2015-10-22	RSF16	NaN	481.2	481.2	485.2	485.8	484.0	

	RSX16	RSF17	…	RSF24	RSH24	RSK24	RSN24	RSX24	RSF25	RSH25	RSK25	\
0	469.7	467.9	…	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	468.6	466.8	…	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2	471.1	469.3	…	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3	477.2	475.4	…	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
4	473.4	471.6	…	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

	RSN25	RSX25
0	NaN	NaN
1	NaN	NaN
2	NaN	NaN
3	NaN	NaN
4	NaN	NaN

```
[5 rows x 54 columns]
```

1.4 4. Data Processing

Task: Find rows where the contract changes

[7]:
```
# Identify if the contract column changes (True) or not (False)
changes = canola['contract'] != canola['contract'].shift()

# Select rows where the contract changes and assign to 'fwrd' df
fwrd = canola[changes].copy()
fwrd.head()
```

[7]:

	date	contract	front_old	front_new	RSF16	RSH16	RSK16	RSN16	\
0	2015-10-16	RSF16	NaN	477.1	477.1	479.8	480.5	480.0	
32	2015-12-02	RSH16	471.4	479.5	471.4	479.5	486.8	491.7	
79	2016-02-10	RSK16	462.8	471.6	NaN	462.8	471.6	477.2	
121	2016-04-12	RSN16	473.9	479.5	NaN	NaN	473.9	479.5	
157	2016-06-02	RSX16	520.6	526.3	NaN	NaN	NaN	520.6	

	RSX16	RSF17	…	RSF24	RSH24	RSK24	RSN24	RSX24	RSF25	RSH25	\
0	469.7	467.9	…	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
32	476.6	480.3	…	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
79	481.1	483.8	…	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
121	478.4	483.2	…	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
157	526.3	529.3	…	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

	RSK25	RSN25	RSX25
0	NaN	NaN	NaN
32	NaN	NaN	NaN
79	NaN	NaN	NaN
121	NaN	NaN	NaN
157	NaN	NaN	NaN

```
[5 rows x 54 columns]
```

1.4.1 4.1 Shift() funtion

Q. What does shift() function do? A. Shift() moves the data along the specifies axis. The default is column-wise.

https://www.geeksforgeeks.org/python-pandas-dataframe-shift/

[8]:
```
df = pd.DataFrame({
    "employee_ID": [101, 102, 103, 104],
    "age": [23, 45, 20, 34],
    "name": ["Amy", "Bob", "Catherine", "Jack"]
})
```

```
# df.shift(axis=0)
df.shift(axis = 1)
```

[8]:
```
   employee_ID  age  name
0          NaN  101    23
1          NaN  102    45
2          NaN  103    20
3          NaN  104    34
```

[9]:
```
# Remove the "front_old" and "front_new" columns and the RSF16 column (since␣
 ↪March contract is featured)
fwrd = fwrd.drop(['front_old','front_new','RSF16'], axis=1)

# Reset the index to "contract"
fwrd.set_index('contract', inplace=True)
fwrd.head()
```

[9]:
```
                date  RSH16  RSK16  RSN16  RSX16  RSF17  RSH17  RSK17  RSN17  \
contract
RSF16     2015-10-16  479.8  480.5  480.0  469.7  467.9  469.6  469.6  469.6
RSH16     2015-12-02  479.5  486.8  491.7  476.6  480.3  480.3  480.3  480.3
RSK16     2016-02-10  462.8  471.6  477.2  481.1  483.8  484.0  482.7  482.7
RSN16     2016-04-12    NaN  473.9  479.5  478.4  483.2  485.0  485.3  485.5
RSX16     2016-06-02    NaN    NaN  520.6  526.3  529.3  530.6  531.8  533.4

           RSX17  …  RSF24  RSH24  RSK24  RSN24  RSX24  RSF25  RSH25  RSK25  \
contract         …
RSF16      469.6  …    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN
RSH16      480.3  …    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN
RSK16      482.7  …    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN
RSN16      477.9  …    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN
RSX16      516.5  …    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN

           RSN25  RSX25
contract
RSF16        NaN    NaN
RSH16        NaN    NaN
RSK16        NaN    NaN
RSN16        NaN    NaN
RSX16        NaN    NaN

[5 rows x 50 columns]
```

1.5 5. Convert a dataframe into a CSV file

Syntax: df.to_csv(‘out.csv’)

[10]:
```
fwrd.to_csv('fwrd.csv', index=True)
```

1.6 6. Select columns and rows using loc() and iloc()

[11]:
```
# Select a column
fwrd["RSH16"]
```

[11]:
```
contract
RSF16    479.8
RSH16    479.5
RSK16    462.8
RSN16      NaN
RSX16      NaN
RSF17      NaN
RSH17      NaN
RSK17      NaN
RSN17      NaN
RSX17      NaN
RSF18      NaN
RSH18      NaN
RSK18      NaN
RSN18      NaN
RSX18      NaN
RSF19      NaN
RSH19      NaN
RSK19      NaN
RSN19      NaN
RSX19      NaN
RSF20      NaN
RSH20      NaN
RSK20      NaN
RSN20      NaN
RSX20      NaN
RSF21      NaN
RSH21      NaN
RSN21      NaN
RSX21      NaN
RSF22      NaN
RSH22      NaN
RSK22      NaN
RSN22      NaN
RSX22      NaN
RSF23      NaN
RSH23      NaN
RSK23      NaN
RSN23      NaN
RSX23      NaN
```

```
RSF24      NaN
RSH24      NaN
RSK24      NaN
RSN24      NaN
RSX24      NaN
Name: RSH16, dtype: float64
```

[12]:
```
# Select a row using index (contract)
fwrd.loc["RSH16"]
```

[12]:
```
date     2015-12-02 00:00:00
RSH16                  479.5
RSK16                  486.8
RSN16                  491.7
RSX16                  476.6
RSF17                  480.3
RSH17                  480.3
RSK17                  480.3
RSN17                  480.3
RSX17                  480.3
RSF18                  480.3
RSH18                    NaN
RSK18                    NaN
RSN18                    NaN
RSX18                    NaN
RSF19                    NaN
RSH19                    NaN
RSK19                    NaN
RSN19                    NaN
RSX19                    NaN
RSF20                    NaN
RSH20                    NaN
RSK20                    NaN
RSN20                    NaN
RSX20                    NaN
RSF21                    NaN
RSH21                    NaN
RSK21                    NaN
RSN21                    NaN
RSX21                    NaN
RSF22                    NaN
RSH22                    NaN
RSK22                    NaN
RSN22                    NaN
RSX22                    NaN
RSF23                    NaN
RSH23                    NaN
```

```
RSK23                    NaN
RSN23                    NaN
RSX23                    NaN
RSF24                    NaN
RSH24                    NaN
RSK24                    NaN
RSN24                    NaN
RSX24                    NaN
RSF25                    NaN
RSH25                    NaN
RSK25                    NaN
RSN25                    NaN
RSX25                    NaN
Name: RSH16, dtype: object
```

[13]:
```
# Select a row and column
fwrd.loc["RSH16", "date"]
```

[13]:
```
Timestamp('2015-12-02 00:00:00')
```

[14]:
```
# Select a range of rows and columns
fwrd.loc["RSH16":"RSX19", "date":"RSH17"]
```

[14]:

contract	date	RSH16	RSK16	RSN16	RSX16	RSF17	RSH17
RSH16	2015-12-02	479.5	486.8	491.7	476.6	480.3	480.3
RSK16	2016-02-10	462.8	471.6	477.2	481.1	483.8	484.0
RSN16	2016-04-12	NaN	473.9	479.5	478.4	483.2	485.0
RSX16	2016-06-02	NaN	NaN	520.6	526.3	529.3	530.6
RSF17	2016-10-19	NaN	NaN	NaN	504.4	511.4	515.5
RSH17	2016-12-15	NaN	NaN	NaN	NaN	518.6	525.6
RSK17	2017-02-08	NaN	NaN	NaN	NaN	NaN	524.0
RSN17	2017-04-10	NaN	NaN	NaN	NaN	NaN	NaN
RSX17	2017-06-05	NaN	NaN	NaN	NaN	NaN	NaN
RSF18	2017-10-16	NaN	NaN	NaN	NaN	NaN	NaN
RSH18	2017-12-13	NaN	NaN	NaN	NaN	NaN	NaN
RSK18	2018-02-08	NaN	NaN	NaN	NaN	NaN	NaN
RSN18	2018-04-05	NaN	NaN	NaN	NaN	NaN	NaN
RSX18	2018-06-01	NaN	NaN	NaN	NaN	NaN	NaN
RSF19	2018-10-12	NaN	NaN	NaN	NaN	NaN	NaN
RSH19	2018-12-11	NaN	NaN	NaN	NaN	NaN	NaN
RSK19	2019-02-12	NaN	NaN	NaN	NaN	NaN	NaN
RSN19	2019-04-11	NaN	NaN	NaN	NaN	NaN	NaN
RSX19	2019-06-07	NaN	NaN	NaN	NaN	NaN	NaN

[15]:
```
# To do the same using indexes
fwrd.iloc[1:5, 2:3]
```

[15]:
```
          RSK16
contract
RSH16     486.8
RSK16     471.6
RSN16     473.9
RSX16       NaN
```

[16]:
```
fwrd.iloc[ :3, 2:]
```

[16]:
```
          RSK16  RSN16  RSX16  RSF17  RSH17  RSK17  RSN17  RSX17  RSF18  \
contract
RSF16     480.5  480.0  469.7  467.9  469.6  469.6  469.6  469.6    NaN
RSH16     486.8  491.7  476.6  480.3  480.3  480.3  480.3  480.3  480.3
RSK16     471.6  477.2  481.1  483.8  484.0  482.7  482.7  482.7  482.7

          RSH18  …  RSF24  RSH24  RSK24  RSN24  RSX24  RSF25  RSH25  RSK25  \
contract        …
RSF16       NaN  …    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN
RSH16       NaN  …    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN
RSK16     482.7  …    NaN    NaN    NaN    NaN    NaN    NaN    NaN    NaN

          RSN25  RSX25
contract
RSF16       NaN    NaN
RSH16       NaN    NaN
RSK16       NaN    NaN

[3 rows x 48 columns]
```

1.7 7. Plot in python

1.7.1 7.1 Using Pandas and matplotlib

[17]:
```
fwrd16 = fwrd.loc['RSH16'].iloc[1:11]
```

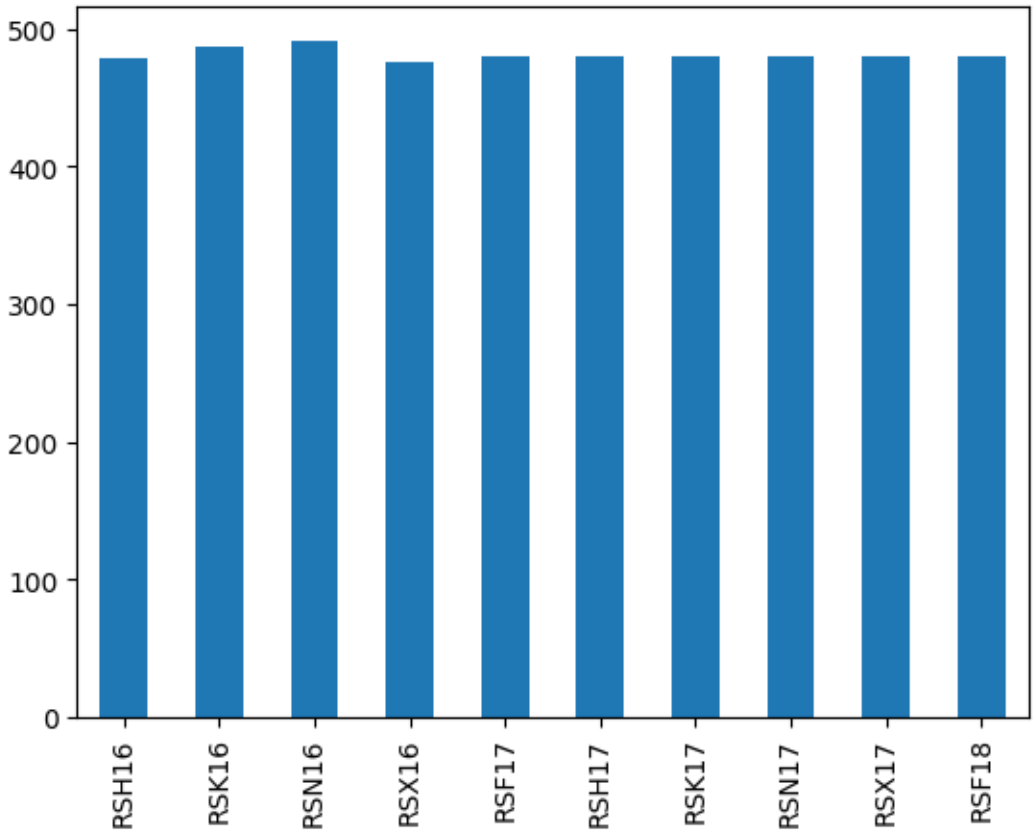
[18]:
```
fwrd16
```

[18]:
```
RSH16    479.5
RSK16    486.8
RSN16    491.7
RSX16    476.6
RSF17    480.3
RSH17    480.3
RSK17    480.3
RSN17    480.3
RSX17    480.3
RSF18    480.3
```

```
Name: RSH16, dtype: object
```

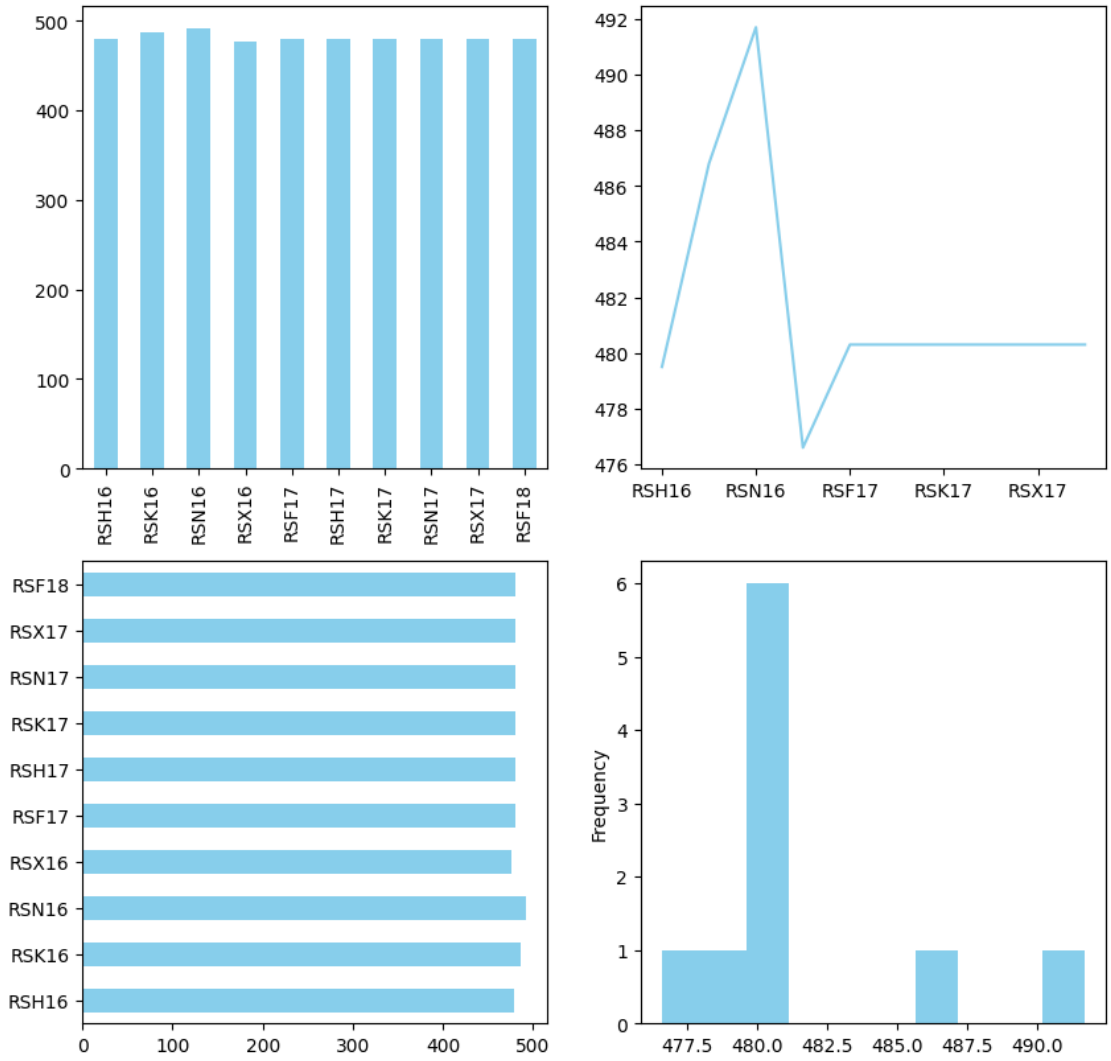
[19]:
```
fwrd16.plot(kind = 'bar')
```

[19]:
```
<Axes: >
```



[20]:
```
fig, axs = plt.subplots(2, 2, figsize=(10, 10))
fwrd16.plot(kind='bar', ax=axs[0, 0], color='skyblue')
fwrd16.plot(kind='line', ax=axs[0, 1], color='skyblue')
fwrd16.plot(kind='barh', ax=axs[1, 0], color='skyblue')
fwrd16.plot(kind='hist', ax=axs[1, 1], color='skyblue')
```

[20]:
```
<Axes: ylabel='Frequency'>
```



1.7.2 7.2 Plot with Altair

[21]:
```
# Install altair
# pip install altair vega_datasets
```

[22]:
```
import altair as alt
```

[23]:
```
data = pd.DataFrame({
    'Contract': ['RSH16', 'RSK16', 'RSN16', 'RSX16', 'RSF17', 'RSH17', 'RSK17',␣
 ↪'RSN17', 'RSX17', 'RSF18'],
    'Price': [479.5, 486.8, 491.7, 476.6, 480.3, 480.3, 480.3, 480.3, 480.3,␣
 ↪480.3]
})
```

```
data
```

[23]:

	Contract	Price
0	RSH16	479.5
1	RSK16	486.8
2	RSN16	491.7
3	RSX16	476.6
4	RSF17	480.3
5	RSH17	480.3
6	RSK17	480.3
7	RSN17	480.3
8	RSX17	480.3
9	RSF18	480.3

[24]:
```
chart1 = alt.Chart(data).mark_bar().encode(
    x='Contract',
    y='Price',
    color=alt.value('skyblue')
).properties(
    title='Price per Contract'
)

chart1
```

```
/usr/local/lib/python3.10/dist-packages/altair/utils/core.py:384: FutureWarning:
the convert_dtype parameter is deprecated and will be removed in a future
version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[24]:
```
alt.Chart(…)
```

[25]:
```
# To make 2x2 grids

top_row = alt.hconcat(chart1, chart1)
bottom_row = alt.hconcat(chart1, chart1)
final_chart = alt.vconcat(top_row, bottom_row)

final_chart
```

```
/usr/local/lib/python3.10/dist-packages/altair/utils/core.py:384: FutureWarning:
the convert_dtype parameter is deprecated and will be removed in a future
version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[25]:
```
alt.VConcatChart(…)
```

[26]:
```
!pip install nbconvert
```

```
Requirement already satisfied: nbconvert in /usr/local/lib/python3.10/dist-
packages (6.5.4)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packages
(from nbconvert) (4.9.4)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-
packages (from nbconvert) (4.12.3)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages
(from nbconvert) (6.1.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.10/dist-
packages (from nbconvert) (0.7.1)
Requirement already satisfied: entrypoints>=0.2.2 in
/usr/local/lib/python3.10/dist-packages (from nbconvert) (0.4)
Requirement already satisfied: jinja2>=3.0 in /usr/local/lib/python3.10/dist-
packages (from nbconvert) (3.1.4)
Requirement already satisfied: jupyter-core>=4.7 in
/usr/local/lib/python3.10/dist-packages (from nbconvert) (5.7.2)
Requirement already satisfied: jupyterlab-pygments in
/usr/local/lib/python3.10/dist-packages (from nbconvert) (0.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from nbconvert) (2.1.5)
Requirement already satisfied: mistune<2,>=0.8.1 in
/usr/local/lib/python3.10/dist-packages (from nbconvert) (0.8.4)
Requirement already satisfied: nbclient>=0.5.0 in
/usr/local/lib/python3.10/dist-packages (from nbconvert) (0.10.0)
Requirement already satisfied: nbformat>=5.1 in /usr/local/lib/python3.10/dist-
packages (from nbconvert) (5.10.4)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from nbconvert) (24.1)
Requirement already satisfied: pandocfilters>=1.4.1 in
/usr/local/lib/python3.10/dist-packages (from nbconvert) (1.5.1)
Requirement already satisfied: pygments>=2.4.1 in
/usr/local/lib/python3.10/dist-packages (from nbconvert) (2.18.0)
Requirement already satisfied: tinycss2 in /usr/local/lib/python3.10/dist-
packages (from nbconvert) (1.3.0)
Requirement already satisfied: traitlets>=5.0 in /usr/local/lib/python3.10/dist-
packages (from nbconvert) (5.7.1)
Requirement already satisfied: platformdirs>=2.5 in
/usr/local/lib/python3.10/dist-packages (from jupyter-core>=4.7->nbconvert)
(4.3.6)
Requirement already satisfied: jupyter-client>=6.1.12 in
/usr/local/lib/python3.10/dist-packages (from nbclient>=0.5.0->nbconvert)
(6.1.12)
Requirement already satisfied: fastjsonschema>=2.15 in
/usr/local/lib/python3.10/dist-packages (from nbformat>=5.1->nbconvert) (2.20.0)
Requirement already satisfied: jsonschema>=2.6 in
```

```
/usr/local/lib/python3.10/dist-packages (from nbformat>=5.1->nbconvert) (4.23.0)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-
packages (from beautifulsoup4->nbconvert) (2.6)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.10/dist-
packages (from bleach->nbconvert) (1.16.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-
packages (from bleach->nbconvert) (0.5.1)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/dist-
packages (from jsonschema>=2.6->nbformat>=5.1->nbconvert) (24.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
/usr/local/lib/python3.10/dist-packages (from
jsonschema>=2.6->nbformat>=5.1->nbconvert) (2023.12.1)
Requirement already satisfied: referencing>=0.28.4 in
/usr/local/lib/python3.10/dist-packages (from
jsonschema>=2.6->nbformat>=5.1->nbconvert) (0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-
packages (from jsonschema>=2.6->nbformat>=5.1->nbconvert) (0.20.0)
Requirement already satisfied: pyzmq>=13 in /usr/local/lib/python3.10/dist-
packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert) (24.0.1)
Requirement already satisfied: python-dateutil>=2.1 in
/usr/local/lib/python3.10/dist-packages (from jupyter-
client>=6.1.12->nbclient>=0.5.0->nbconvert) (2.8.2)
Requirement already satisfied: tornado>=4.1 in /usr/local/lib/python3.10/dist-
packages (from jupyter-client>=6.1.12->nbclient>=0.5.0->nbconvert) (6.3.3)
```

[27]:
```
!apt-get install texlive texlive-xetex texlive-latex-extra pandoc
```

```
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
pandoc is already the newest version (2.9.2.1-3ubuntu2).
texlive is already the newest version (2021.20220204-1).
texlive-latex-extra is already the newest version (2021.20220204-1).
texlive-xetex is already the newest version (2021.20220204-1).
0 upgraded, 0 newly installed, 0 to remove and 49 not upgraded.
```

[29]:
```
!jupyter nbconvert --to pdf /content/drive/MyDrive/Colab Notebooks/Lab_3.ipynb
```

```
[NbConvertApp] WARNING | pattern '/content/drive/MyDrive/Colab' matched no files
[NbConvertApp] WARNING | pattern 'Notebooks/Lab_3.ipynb' matched no files
This application is used to convert notebook files (*.ipynb)
        to various other formats.

        WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.

Options
=======
The options below are convenience aliases to configurable class-options,
```

```
as listed in the "Equivalent to" description-line of the aliases.
To see all configurable class-options for some <cmd>, use:
    <cmd> --help-all

--debug
    set log level to logging.DEBUG (maximize logging output)
    Equivalent to: [--Application.log_level=10]
--show-config
    Show the application's configuration (human-readable format)
    Equivalent to: [--Application.show_config=True]
--show-config-json
    Show the application's configuration (json format)
    Equivalent to: [--Application.show_config_json=True]
--generate-config
    generate default config file
    Equivalent to: [--JupyterApp.generate_config=True]
-y
    Answer yes to any questions instead of prompting.
    Equivalent to: [--JupyterApp.answer_yes=True]
--execute
    Execute the notebook prior to export.
    Equivalent to: [--ExecutePreprocessor.enabled=True]
--allow-errors
    Continue notebook execution even if one of the cells throws an error and
include the error message in the cell output (the default behaviour is to abort
conversion). This flag is only relevant if '--execute' was specified, too.
    Equivalent to: [--ExecutePreprocessor.allow_errors=True]
--stdin
    read a single notebook file from stdin. Write the resulting notebook with
default basename 'notebook.*'
    Equivalent to: [--NbConvertApp.from_stdin=True]
--stdout
    Write notebook output to stdout instead of files.
    Equivalent to: [--NbConvertApp.writer_class=StdoutWriter]
--inplace
    Run nbconvert in place, overwriting the existing notebook (only
            relevant when converting to notebook format)
    Equivalent to: [--NbConvertApp.use_output_suffix=False
--NbConvertApp.export_format=notebook --FilesWriter.build_directory=]
--clear-output
    Clear output of current file and save in place,
            overwriting the existing notebook.
    Equivalent to: [--NbConvertApp.use_output_suffix=False
--NbConvertApp.export_format=notebook --FilesWriter.build_directory=
--ClearOutputPreprocessor.enabled=True]
--no-prompt
    Exclude input and output prompts from converted document.
    Equivalent to: [--TemplateExporter.exclude_input_prompt=True
```

```
--TemplateExporter.exclude_output_prompt=True]
--no-input
    Exclude input cells and output prompts from converted document.
            This mode is ideal for generating code-free reports.
    Equivalent to: [--TemplateExporter.exclude_output_prompt=True
--TemplateExporter.exclude_input=True
--TemplateExporter.exclude_input_prompt=True]
--allow-chromium-download
    Whether to allow downloading chromium if no suitable version is found on the
system.
    Equivalent to: [--WebPDFExporter.allow_chromium_download=True]
--disable-chromium-sandbox
    Disable chromium security sandbox when converting to PDF..
    Equivalent to: [--WebPDFExporter.disable_sandbox=True]
--show-input
    Shows code input. This flag is only useful for dejavu users.
    Equivalent to: [--TemplateExporter.exclude_input=False]
--embed-images
    Embed the images as base64 dataurls in the output. This flag is only useful
for the HTML/WebPDF/Slides exports.
    Equivalent to: [--HTMLExporter.embed_images=True]
--sanitize-html
    Whether the HTML in Markdown cells and cell outputs should be sanitized..
    Equivalent to: [--HTMLExporter.sanitize_html=True]
--log-level=<Enum>
    Set the log level by value or name.
    Choices: any of [0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR',
'CRITICAL']
    Default: 30
    Equivalent to: [--Application.log_level]
--config=<Unicode>
    Full path of a config file.
    Default: ''
    Equivalent to: [--JupyterApp.config_file]
--to=<Unicode>
    The export format to be used, either one of the built-in formats
            ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook',
'pdf', 'python', 'rst', 'script', 'slides', 'webpdf']
            or a dotted object name that represents the import path for an
            ``Exporter`` class
    Default: ''
    Equivalent to: [--NbConvertApp.export_format]
--template=<Unicode>
    Name of the template to use
    Default: ''
    Equivalent to: [--TemplateExporter.template_name]
--template-file=<Unicode>
    Name of the template file to use
```

```
    Default: None
    Equivalent to: [--TemplateExporter.template_file]
--theme=<Unicode>
    Template specific theme(e.g. the name of a JupyterLab CSS theme distributed
    as prebuilt extension for the lab template)
    Default: 'light'
    Equivalent to: [--HTMLExporter.theme]
--sanitize_html=<Bool>
    Whether the HTML in Markdown cells and cell outputs should be sanitized.This
    should be set to True by nbviewer or similar tools.
    Default: False
    Equivalent to: [--HTMLExporter.sanitize_html]
--writer=<DottedObjectName>
    Writer class used to write the
                                        results of the conversion
    Default: 'FilesWriter'
    Equivalent to: [--NbConvertApp.writer_class]
--post=<DottedOrNone>
    PostProcessor class used to write the
                                        results of the conversion
    Default: ''
    Equivalent to: [--NbConvertApp.postprocessor_class]
--output=<Unicode>
    overwrite base name use for output files.
                can only be used when converting one notebook at a time.
    Default: ''
    Equivalent to: [--NbConvertApp.output_base]
--output-dir=<Unicode>
    Directory to write output(s) to. Defaults
                                    to output to the directory of each notebook.
To recover
                                    previous default behaviour (outputting to the
current
                                    working directory) use . as the flag value.
    Default: ''
    Equivalent to: [--FilesWriter.build_directory]
--reveal-prefix=<Unicode>
    The URL prefix for reveal.js (version 3.x).
            This defaults to the reveal CDN, but can be any url pointing to a
copy
            of reveal.js.
            For speaker notes to work, this must be a relative path to a local
            copy of reveal.js: e.g., "reveal.js".
            If a relative path is given, it must be a subdirectory of the
            current directory (from which the server is run).
            See the usage documentation
            (https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-
html-slideshow)
```

```
            for more details.
    Default: ''
    Equivalent to: [--SlidesExporter.reveal_url_prefix]
--nbformat=<Enum>
    The nbformat version to write.
            Use this to downgrade notebooks.
    Choices: any of [1, 2, 3, 4]
    Default: 4
    Equivalent to: [--NotebookExporter.nbformat_version]

Examples
--------

    The simplest way to use nbconvert is

            > jupyter nbconvert mynotebook.ipynb --to html

            Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown',
'notebook', 'pdf', 'python', 'rst', 'script', 'slides', 'webpdf'].

            > jupyter nbconvert --to latex mynotebook.ipynb

            Both HTML and LaTeX support multiple output templates. LaTeX
includes
            'base', 'article' and 'report'.  HTML includes 'basic', 'lab' and
            'classic'. You can specify the flavor of the format used.

            > jupyter nbconvert --to html --template lab mynotebook.ipynb

            You can also pipe the output to stdout, rather than a file

            > jupyter nbconvert mynotebook.ipynb --stdout

            PDF is generated via latex

            > jupyter nbconvert mynotebook.ipynb --to pdf

            You can get (and serve) a Reveal.js-powered slideshow

            > jupyter nbconvert myslides.ipynb --to slides --post serve

            Multiple notebooks can be given at the command line in a couple of
            different ways:

            > jupyter nbconvert notebook*.ipynb
            > jupyter nbconvert notebook1.ipynb notebook2.ipynb

            or you can specify the notebooks list in a config file, containing::
```

```
                c.NbConvertApp.notebooks = ["my_notebook.ipynb"]

            > jupyter nbconvert --config mycfg.py

To see all available configurables, use `--help-all`.
```