Content

1 ### Library Import ### 1.1 ### Widgets 1.2 ### CSV/Data Ops 1.3 ### Drawing 2 ### Widgets ### 2.1 ### Tabs 2.2 ### FileUpload 2.3 ### Output 2.4 ### SelectMultiple 2.5 ### RadioButton 2.6 ### IntSlider 2.7 ### Accordion 2.8 ### Button 2.9 ### HBox/VBox 2.10 ### ToggleButtons 2.11 ### Dropdown 2.12 ### ColorPicker 3 ### Value Import ### 4 ### Button Action ### 4.1 ### Preview Button 4.2 ### Upload Button 4.3 ### Description ToggleButton 4.4 ### Plot Button

## Library Import

1.1 ### Widgets 1.2 ### CSV/Data Ops 1.3 ### Drawing

In [1]:

```
pip install ipywidgets
```

```
Requirement already satisfied: ipywidgets in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (8.1.2)
Requirement already satisfied: ipython>=6.1.0 in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from ipywidgets) (8.10.0)
Requirement already satisfied: widgetsnbextension~=4.0.10 in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from ipywidgets) (4.0.10)
Requirement already satisfied: jupyterlab-widgets~=3.0.10 in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from ipywidgets) (3.0.10)
Requirement already satisfied: comm>=0.1.3 in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from ipywidgets) (0.2.1)
Requirement already satisfied: traitlets>=4.3.1 in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from ipywidgets) (5.7.1)
Requirement already satisfied: pygments>=2.4.0 in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from ipython>=6.1.0->ipywidgets) (2.11.2)
Requirement already satisfied: backcall in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from ipython>=6.1.0->ipywidgets) (0.2.0)
Requirement already satisfied: decorator in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from ipython>=6.1.0->ipywidgets) (5.1.1)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.30 in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from ipython>=6.1.0->ipywidgets) (3.0.36)
Requirement already satisfied: pickleshare in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from ipython>=6.1.0->ipywidgets) (0.7.5)
Requirement already satisfied: appnope in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from ipython>=6.1.0->ipywidgets) (0.1.2)
Requirement already satisfied: jedi>=0.16 in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from ipython>=6.1.0->ipywidgets) (0.18.1)
Requirement already satisfied: stack-data in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from ipython>=6.1.0->ipywidgets) (0.2.0)
Requirement already satisfied: matplotlib-inline in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from ipython>=6.1.0->ipywidgets) (0.1.6)
Requirement already satisfied: pexpect>4.3 in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from ipython>=6.1.0->ipywidgets) (4.8.0)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from jedi>=0.16->ipython>=6.1.0->ipywidgets) (0.8.3)
Requirement already satisfied: ptyprocess>=0.5 in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from pexpect>4.3->ipython>=6.1.0->ipywidgets) (0.7.0)
Requirement already satisfied: wcwidth in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from prompt-toolkit<3.1.0,>=3.0.30->ipython>=6.1.0->ipywidgets)
Requirement already satisfied: pure-eval in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from stack-data->ipython>=6.1.0->ipywidgets) (0.2.2)
Requirement already satisfied: executing in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from stack-data->ipython>=6.1.0->ipywidgets) (0.8.3)
Requirement already satisfied: asttokens in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from stack-data->ipython>=6.1.0->ipywidgets) (2.0.5)
Requirement already satisfied: six in /Users/yavuzsebe/anaconda3/lib/python3.10/site-packages (from asttokens->stack-data->ipython>=6.1.0->ipywidgets) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:

```
# bash
# jupyter labextension install @jupyter-widgets/jupyterlab-manager
```

In [3]:

```
# Widgets:
import ipywidgets as widgets

# CSV/DataFrame:
import pandas as pd
import numpy as np
from io import StringIO

# Drawing
import matplotlib.pyplot as plt
%matplotlib inline

from IPython.display import display
```

## Widgets

2.1 ### Tabs 2.2 ### FileUpload 2.3 ### Output 2.4 ### SelectMultiple 2.5 ### RadioButton 2.6 ### IntSlider 2.7 ### Accordion 2.8 ### Button 2.9 ### HBox/VBox 2.10 ### ToggleButtons 2.11 ### Dropdown 2.12 ### ColorPicker

### Tabs

In [4]:

```
tab = widgets.Tab()
placeholder = widgets.Label() # başlığı yazmak için önemli sonrasındaki childrenlardan ayrı tutulmalı her türlü yazılmalı..
tab.children = [placeholder, placeholder, placeholder] # ... her türlü yazılmalı
tab.set_title(0, "Upload")
tab.set_title(1, "Describer")
tab.set_title(2, "Plotter")
tab
```

Out [4]:

```
Tab(children=(Label(value=''), Label(value=''), Label(value='')), selected_index=0, titles=('Upload', 'Describ…
```

FileUpload

In [5]:

```
up = widgets.FileUpload(accept="", multiple=False)
up
```

Out [5]:

```
FileUpload(value=(), description='Upload')
```

Output

In [6]:

```
out = widgets.Output(layout={"border": "1px solid black"})
tab.children = [up, out, out]
tab
```

Out [6]:

```
Tab(children=(FileUpload(value=(), description='Upload'), Output(layout=Layout(border_bottom='1px solid black'…
```

SelectMultiple

In [7]:

```
eraser = widgets.SelectMultiple(
    options = ["tab", '"'],
    value = ["tab"],
    #rows=10,
    description = "Eraser: ",
    disabled = False
)
eraser
```

Out [7]:

```
SelectMultiple(description='Eraser: ', index=(0,), options=('tab', '"'), value=('tab',))
```

RadioButtons

In [8]:

```
delim = widgets.RadioButtons(
    options = [";", ",", " "],
    description="Seperator: ",
    disabled = False
)
delim
```

Out [8]:

```
RadioButtons(description='Seperator: ', options=(';', ',', ' '), value=';')
```

IntSlider

In [9]:

```
rows = widgets.IntSlider(
    value = 0, #0la başlayan
    step = 1, #her basamakta 1 artan
    description = "# of lines:",
    disabled = False,
    continous_update = False,
    orientation = "horizontal",
    readout = True,
    readout_format= "d"
)
rows
```

Out [9]:

```
IntSlider(value=0, description='# of lines:')
```

Accordion

In [10]:

```
accordion = widgets.Accordion()
accordion.children = [
    up,
    delim,
    rows]
accordion.set_title(0, "File Selection")
accordion.set_title(1, "Delimiter")
accordion.set_title(2, "Skip Rows")
accordion
```

Out [10]:

```
Accordion(children=(FileUpload(value=(), description='Upload'), RadioButtons(description='Seperator: ', option…
```

Button

In [11]:

```
button_upload = widgets.Button(
    description = "Upload",
    disabled = False,
    button_style = "warning",
    tooltip = "Click to Upload",
    icon = "check"
)
button_upload
```

Out [11]:

```
Button(button_style='warning', description='Upload', icon='check', style=ButtonStyle(), tooltip='Click to Uplo…
```

In [12]:

```
button_preview = widgets.Button(
    description = "Preview",
    disabled = False,
    button_style = "info",
    tooltip = "Click to Preview",
    icon = "search"
)
button_preview
```

Out [12]:

```
Button(button_style='info', description='Preview', icon='search', style=ButtonStyle(), tooltip='Click to Previ…
```

In [13]:

```
button_plot = widgets.Button(
    description = "Plot",
    disabled = False,
    button_style = "danger",
    tooltip = "Click to Plot",
    icon = "pencil"
)
button_plot
```

```
Out [13]: Button(button_style='danger', description='Plot', icon='pencil', style=ButtonStyle(), tooltip='Click to Plot')
```

HBox/VBox - Groupping

In [14]:

```
vb = widgets.VBox([delim, eraser]) # bir accordion a gruplar eklenebilir
vb
```

```
Out [14]: VBox(children=(RadioButtons(description='Seperator: ', options=(';', ',', ' '), value=';'), SelectMultiple(des…
```

In [15]:

```
hb = widgets.HBox([delim, eraser])
hb
```

```
Out [15]: HBox(children=(RadioButtons(description='Seperator: ', options=(';', ',', ' '), value=';'), SelectMultiple(des…
```

In [16]:

```
accordion.children = [
    up,
    widgets.VBox([delim, eraser]), # bir accordion a gruplar eklenebilir
    rows]

accordion_box = widgets.VBox([
    accordion,
    widgets.HBox([button_preview, button_upload]), # bir accordion a gruplar eklenebilir
    out
])
accordion_box
```

```
Out [16]: VBox(children=(Accordion(children=(FileUpload(value=(), description='Upload'), VBox(children=(RadioButtons(des…
```

ToggleButtons

In [17]:

```
toggle = widgets.ToggleButtons(
    options = ["Preview  ", "Info  ", "Stats  "],
    #assignmentlar butona özgü değil hangi butonun seçili olduğuna göre
    #radiobutton gibi
    description = "Options",
    disabled = False,
    button_Style = "warning",
    icons = ["search", "info", "tachometer"]
)
#toggle.observe(desc_clicked, "value")
toggle
```

```
Out [17]: ToggleButtons(description='Options', icons=('search', 'info', 'tachometer'), options=('Preview  ', 'Info  ', '…
```

Dropdown

In [18]:

```
x_axis = widgets.Dropdown(
    options = [""],
    value = "",
    description = "X-Axis:",
    disabled = False)

y_axis = widgets.Dropdown(
    options = [""],
    value = "",
    description = "Y-Axis:",
    disabled = False)

graph_type = widgets.Dropdown(
    options = ["Bar Chart", "Line Chart"],
    value = "Bar Chart",
    description = "Chart Type:",
    disabled = False)
graph_type
```

```
Out [18]: Dropdown(description='Chart Type:', options=('Bar Chart', 'Line Chart'), value='Bar Chart')
```

ColorPicker

In [19]:

```
color_picker = widgets.ColorPicker(
    concise = False,
    description = "Color Picker: ",
    value = "lightblue",
    disabled = False)
color_picker
```

```
Out [19]: ColorPicker(value='lightblue', description='Color Picker: ')
```

Şu ana kadar yaptıklarımız hepsinin çıktısı:

In [20]:
```python
children = [
    accordion_box,
    widgets.VBox([toggle, out]),
    widgets.VBox([
        widgets.HBox([graph_type, color_picker]),
        widgets.HBox([x_axis, y_axis]),
        button_plot,
        out])]
tab.children = children
tab
```

Out [20]:
```
Tab(children=(VBox(children=(Accordion(children=(FileUpload(value=(), description='Upload'), VBox(children=(Ra…
```

Value Import

In [21]:
```python
def content_parser():
    if not up.value or "type" not in up.value[0]:
        with out:
            print("No CSV loaded.")
    else:
        typ, content = "", ""
        up_value = up.value[0]["type"]
        #up_value = up.value
        #for i in up_value.keys():
            #typ = up_value[i]["metadata"]["type"]

        if up_value == "text/csv":
            #content = up_value[i]["content"]
            content = up.value[0]["content"]
            content_str = str(content, "utf-8")

            #if eraser.value != {}:
            if eraser.value:
                for val in eraser.value:
                    if val == "tab":
                        content_str = content_str.replace("\t", "")
                    else:
                        content_str = content_str.replace(val, "")
            #if content_str != "":
            if content_str:
                str_io = StringIO(content_str)
                return str_io

def df_converter():
    content = content_parser()
    if content is not None:
        #df = pd.read_csv(content, sep = delim.value, index_Col = False, skiprows = rows.value)
        df = pd.read_csv(content, sep=delim.value)
        return df
    else:
        return None
```

In [22]:
```python
tab
```

Out [22]:
```
Tab(children=(VBox(children=(Accordion(children=(FileUpload(value=(), description='Upload'), VBox(children=(Ra…
```

In [36]:
```python
df_converter()
```

Out [36]:

	oscar_no	oscar_yr	award	name	movie	age	birth_pl	birth_date	birth_mo	birth_d	birth_y
0	1	1929	Best actress	Janet Gaynor	7th Heaven	22	Pennsylvania	1906-10-06	10	6	1906
1	2	1930	Best actress	Mary Pickford	Coquette	37	Canada	1892-04-08	4	8	1892
2	3	1931	Best actress	Norma Shearer	The Divorcee	28	Canada	1902-08-10	8	10	1902
3	4	1932	Best actress	Marie Dressler	Min and Bill	63	Canada	1868-11-09	11	9	1868
4	5	1933	Best actress	Helen Hayes	The Sin of Madelon Claudet	32	Washington DC	1900-10-10	10	10	1900
...	...	...	...	...	...	...	...	...	...	...	...
179	87	2015	Best actor	Eddie Redmayne	The Theory of Everything	32	England	1982-01-06	1	6	1982
180	88	2016	Best actor	Leonardo Di Caprio	The Revenant	41	California	1974-11-11	11	11	1974
181	89	2017	Best actor	Casey Affleck	Manchester by the Sa	41	Massachusetts	1975-08-12	8	12	1975
182	90	2018	Best actor	Gary Oldman	Darkest Hour	59	England	1958-03-21	3	21	1958
183	91	2019	Best actor	Rami Malek	Bohemian Rhapsody	37	California	1981-05-12	5	12	1981

184 rows × 11 columns

Button Action

4.1 ### Preview Button 4.2 ### Upload Button 4.3 ### Description ToggleButton 4.4 ### Plot Button

Preview Button

In [24]:
```
def preview():
    df = df_converter()
    with out:
        out.clear_output()
        print("\n -------------------- Now this is how your DF looks like: -------------------- \n")
        if df is not None:
            print(df.head(10))
        else:
            print("Configuration is wrong/missing...")
```

In [25]:
```
def preview_clicked(b): # butona fonksiyonu direkt atayamıyoruz, parantez içine kullanmayacağımızı düşündüğümüz bir paramet
    preview() # başka bir fonksiyon içerisinde çalıştırıp...
button_preview.on_click(preview_clicked) # fonksiyonu on_click komutu içinde butona atayabiliyoruz
```

Upload Button

In [26]:
```
def upload():
    df = df_converter()
    with out:
        out.clear_output()
        print("\n -------------------- Your uploaded DF looks like: -------------------- \n")
        if df is not None:
            print(df)
            x_axis.options = df.columns
            y_axis.options = df.columns
        else:
            print("Configuration is wrong/missing...")
```

In [27]:
```
def upload_clicked(b): #önceki atadığımız parametrenin aynısını koyabiliriz
    upload()
button_upload.on_click(upload_clicked)
```

In [28]:
```
tab
```

Out [28]:
```
Tab(children=(VBox(children=(Accordion(children=(FileUpload(value=(), description='Upload'), VBox(children=(Ra…
```

Description ToggleButton

In [29]:
```
def desc():
    info_level = toggle.value
    if info_level != {}:
        df = df_converter()
        with out:
            out.clear_output()
            print("\n -------------------- Your  {}looks like: -------------------- \n".format(info_level))
            if df is not None:
                if info_level == "Info  ":
                    print(df.info(verbose=True))
                elif info_level == "Stats  ":
                    print(df.describe())
                elif info_level == "Preview  ":
                    print(df.head(5))
            else:
                print("Configuration is wrong/missing...")
```

In [30]:
```
def desc_clicked(b):
    desc()
toggle.observe(desc_clicked, "value") #value değerince seçim yapmasını istiyoruz
```

In [31]:
```
tab
```

Out [31]:
```
Tab(children=(VBox(children=(Accordion(children=(FileUpload(value=(), description='Upload'), VBox(children=(Ra…
```

Plot Button

In [32]:
```
def plot():
    graph = graph_type.value
    if graph != {}:
        df = df_converter()
        with out:
            out.clear_output()
            print("\n -------------------- Your  {}looks like: -------------------- \n".format(graph))
            if (df is not None):
                #df = df.head(5)
                height = df[y_axis.value]
                bars = df[x_axis.value]
                y_pos = np.arange(len(height))
                plt.figure(figsize = (10,4))
                if graph == "Bar Chart":

                    # create bars
                    plt.bar(
                        y_pos,
                        height,
                        color = color_picker.value)
                    # create names on the x-axis
                    plt.xticks(y_pos, bars)

                elif graph == "Line Chart":
                    plt.plot(
                        bars,
                        height,
                        color = color_picker.value,
                        marker = "o",
                        linestyle = "solid")
                    plt.xticks(bars)

                plt.show()
```

In [33]:

```
def plotter_clicked(b):
    plot()
button_plot.on_click(plotter_clicked)
```

In [34]:

```
tab
```

```
Out [34]: Tab(children=(VBox(children=(Accordion(children=(FileUpload(value=(), description='Upload'), VBox(children=(Ra…
```

Sonuç

In [35]:

```
tab
```

```
Out [35]: Tab(children=(VBox(children=(Accordion(children=(FileUpload(value=(), description='Upload'), VBox(children=(Ra…
```

In [ ]: