



# Template based graph generation in Python and its maintenance in Git



Pournima Prabhakaran



EU Connect 2020, [pournima.prabhakaran@ocs-consulting.com](mailto:pournima.prabhakaran@ocs-consulting.com)

# Visualization in the Clinical World

---



## Data Quality

- Trends of missing records
- Duplicate Records
- Unusual trends in findings measurements

## Clinical Safety Review

- Risk Evaluation

## Clinical Outcome

- Endpoint Analysis

## Real World Data

# Data visualization tools



TIBCO<sup>™</sup>  
Spotfire<sup>®</sup>



sas



python<sup>™</sup>



+tableau

sas VISUAL  
ANALYTICS



# Data visualization in Python

---

- Open source for experts and beginners
- Do more with less code
- Better code readability
- Fast debugging
- Great packages for graphs : matplotlib, seaborn, ggplot, plotly
- Powerful data processing
- Big data analysis

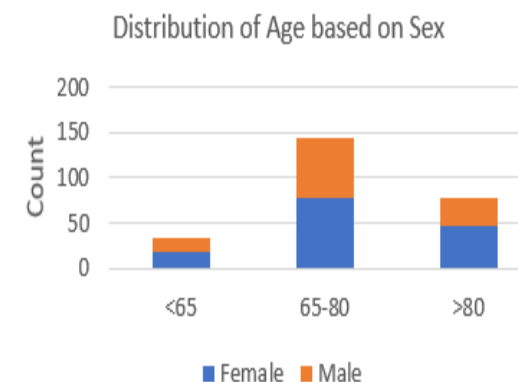


# Current Plotting Routine



|    | USUBJID     | TRTP                 | AGEGRP | SEX |
|----|-------------|----------------------|--------|-----|
| 1  | 01-701-1015 | Placebo              | <65    | F   |
| 2  | 01-701-1023 | Placebo              | <65    | M   |
| 3  | 01-701-1028 | Xanomeline High Dose | 65-80  | M   |
| 4  | 01-701-1033 | Xanomeline Low Dose  | 65-80  | M   |
| 5  | 01-701-1034 | Xanomeline High Dose | 65-80  | F   |
| 6  | 01-701-1047 | Placebo              | >80    | F   |
| 7  | 01-701-1097 | Xanomeline Low Dose  | 65-80  | M   |
| 8  | 01-701-1111 | Xanomeline Low Dose  | >80    | F   |
| 9  | 01-701-1115 | Xanomeline Low Dose  | >80    | M   |
| 10 | 01-701-1118 | Placebo              | <65    | M   |
| 11 | 01-701-1130 | Placebo              | >80    | M   |
| 12 | 01-701-1133 | Xanomeline High Dose | >80    | F   |
| 13 | 01-701-1146 | Xanomeline High Dose | 65-80  | F   |
| 14 | 01-701-1148 | Xanomeline High Dose | <65    | M   |
| 15 | 01-701-1153 | Placebo              | 65-80  | F   |
| 16 | 01-701-1180 | Xanomeline High Dose | <65    | M   |
| 17 | 01-701-1181 | Xanomeline High Dose | 65-80  | F   |
| 18 | 01-701-1188 | Xanomeline Low Dose  | 65-80  | M   |
| 19 | 01-701-1192 | Xanomeline Low Dose  | 65-80  | F   |
| 20 | 01-701-1203 | Placebo              | >80    | F   |
| 21 | 01-701-1211 | Xanomeline Low Dose  | 65-80  | F   |

| AGEGRP | F  | M  |
|--------|----|----|
| <65    | 19 | 14 |
| 65-80  | 78 | 66 |
| >80    | 46 | 31 |

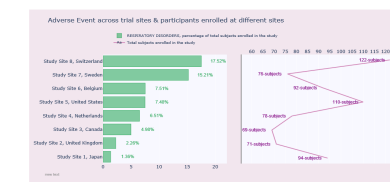
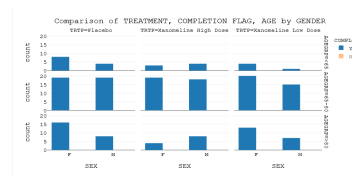
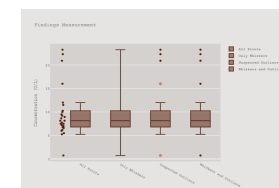
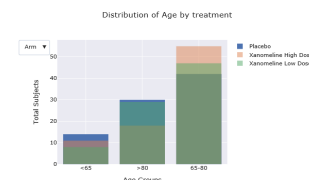


# Plotting using template graphs

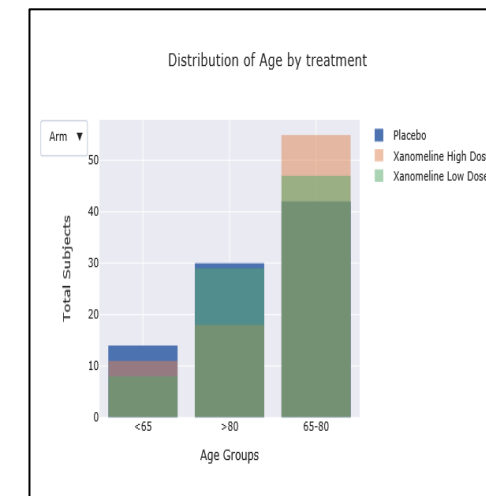


|    | USUBJID     | TRTP                 | AGEGRP | SEX |
|----|-------------|----------------------|--------|-----|
| 1  | 01-701-1015 | Placebo              | <65    | F   |
| 2  | 01-701-1023 | Placebo              | <65    | M   |
| 3  | 01-701-1028 | Xanomeline High Dose | 65-80  | M   |
| 4  | 01-701-1033 | Xanomeline Low Dose  | 65-80  | M   |
| 5  | 01-701-1034 | Xanomeline High Dose | 65-80  | F   |
| 6  | 01-701-1047 | Placebo              | >80    | F   |
| 7  | 01-701-1097 | Xanomeline Low Dose  | 65-80  | M   |
| 8  | 01-701-1111 | Xanomeline Low Dose  | >80    | F   |
| 9  | 01-701-1115 | Xanomeline Low Dose  | >80    | M   |
| 10 | 01-701-1118 | Placebo              | <65    | M   |
| 11 | 01-701-1130 | Placebo              | >80    | M   |
| 12 | 01-701-1133 | Xanomeline High Dose | >80    | F   |
| 13 | 01-701-1146 | Xanomeline High Dose | 65-80  | F   |
| 14 | 01-701-1148 | Xanomeline High Dose | <65    | M   |
| 15 | 01-701-1153 | Placebo              | 65-80  | F   |
| 16 | 01-701-1180 | Xanomeline High Dose | <65    | M   |
| 17 | 01-701-1181 | Xanomeline High Dose | 65-80  | F   |
| 18 | 01-701-1188 | Xanomeline Low Dose  | 65-80  | M   |
| 19 | 01-701-1192 | Xanomeline Low Dose  | 65-80  | F   |
| 20 | 01-701-1203 | Placebo              | >80    | F   |
| 21 | 01-701-1211 | Xanomeline Low Dose  | 65-80  | F   |

Data processing



Plotting

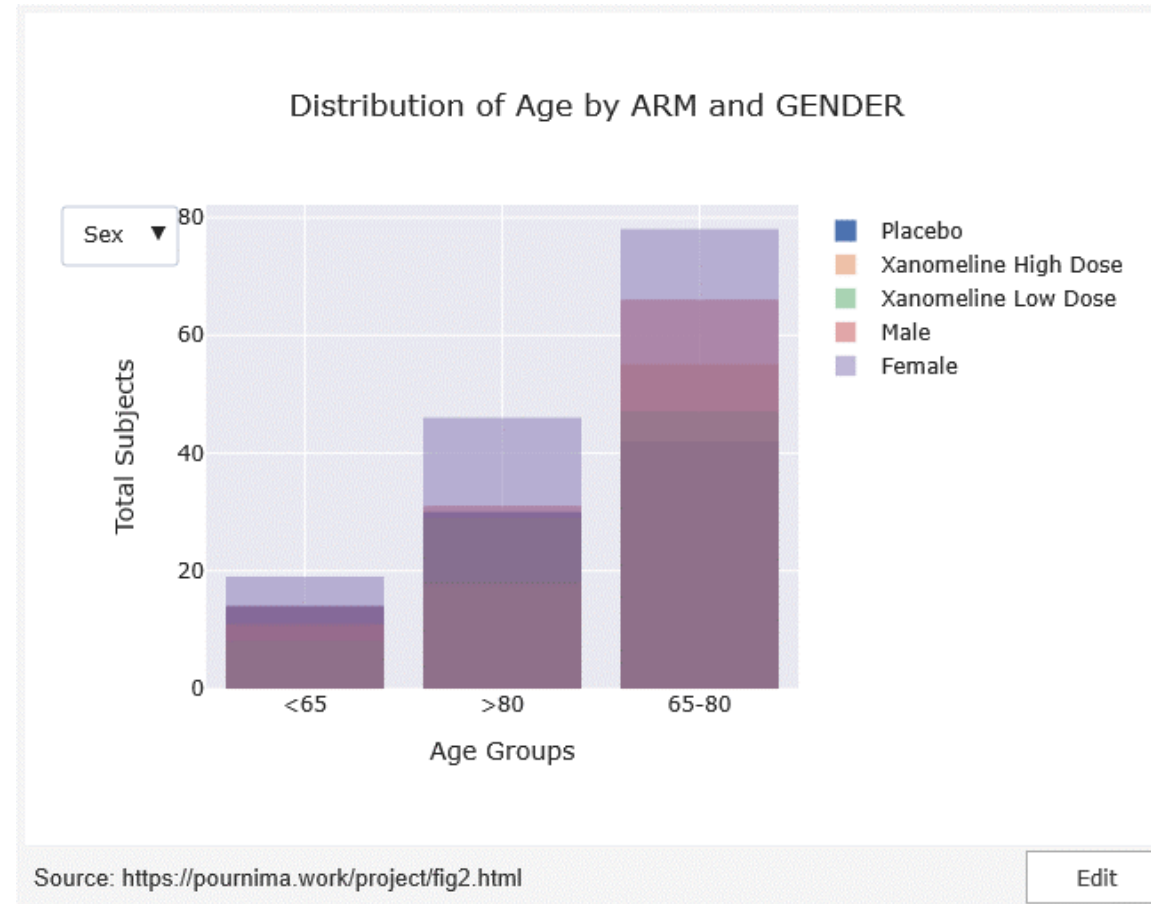


TEMPLATE GRAPH LIBRARY



# Example 1: Interactive vertical bar

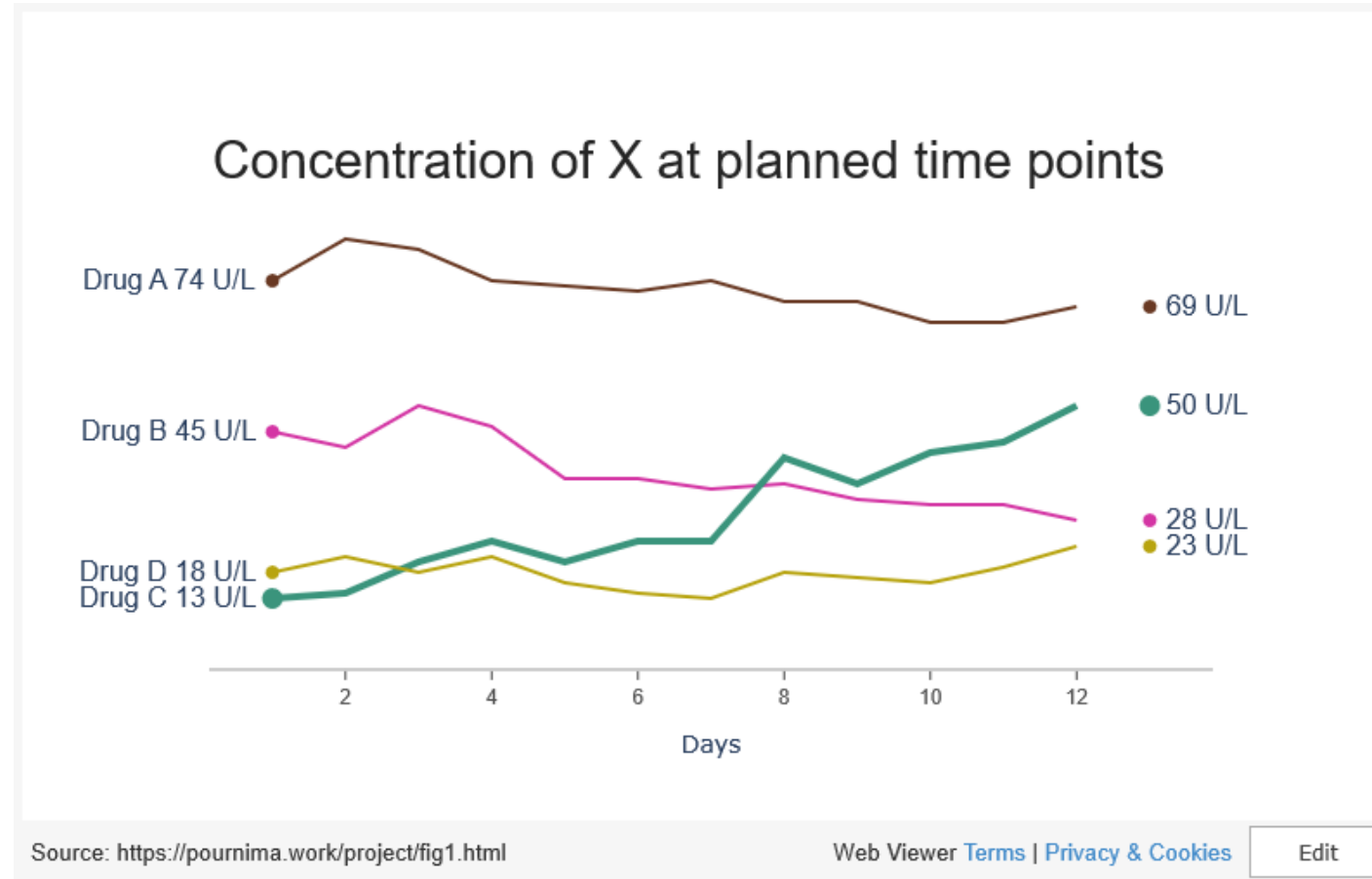
- ✓ Distance between bars
- ✓ Size of titles
- ✓ Size of legends
- ✓ Colour schemes
- ✓ Hover data of your choice
- ✓ Dropdown variables
- ✓ Themes





# Example 2: Interactive line plot

- ✓ Reference lines
- ✓ Marking values
- ✓ Flexible axis



# Template codes in Python



```
import plotly.graph_objects as go
```

```
import pandas as pd
```

```
def intscatter(source, destination, xlabel, ylabel)
```

```
    df = pd.read_csv(source)
```

```
    fig = go.Figure()
```

```
    fig.add_trace(go.Scatter(x = age ,y = weight ))
```

```
    fig.update_layout(xaxis_title = xlabel, yaxis_title = ylabel)
```

```
    fig.write_html(destination)
```

#1 Create data frame

#2 Instantiate a Figure object

#3 Add Traces

#4 Update Layout

#5 Create Output

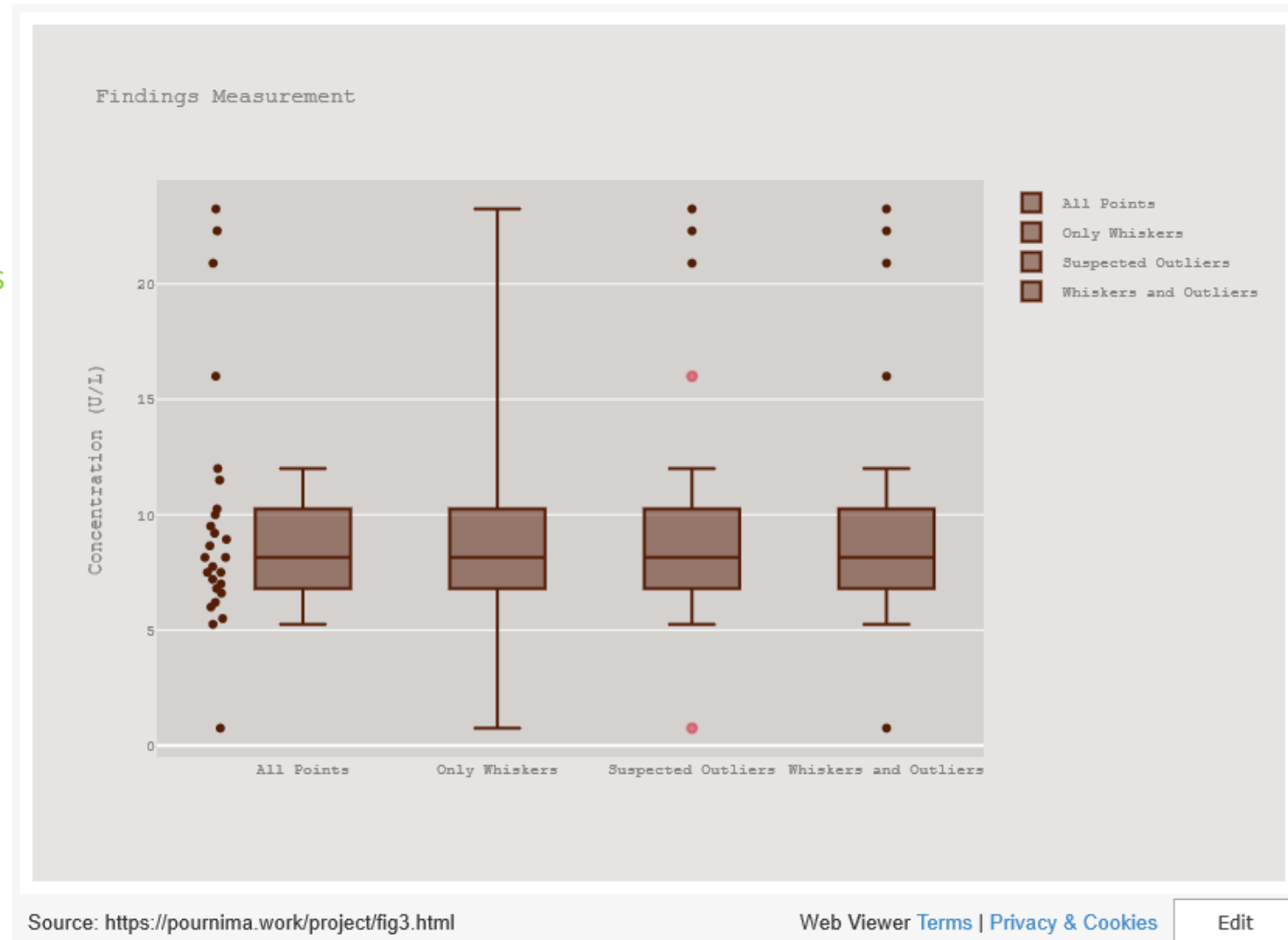


# Example 3: Interactive boxplot

## ✓ Statistical tasks

```
fig.add_trace(go.Box(  
    y = ['Concentration'],  
    name = "Whiskers and Outliers",  
    boxpoints = 'outliers', # only outliers  
    marker_color = 'rgb(84, 29, 7)',  
    line_color = 'rgb(84, 29, 7)'  
))
```

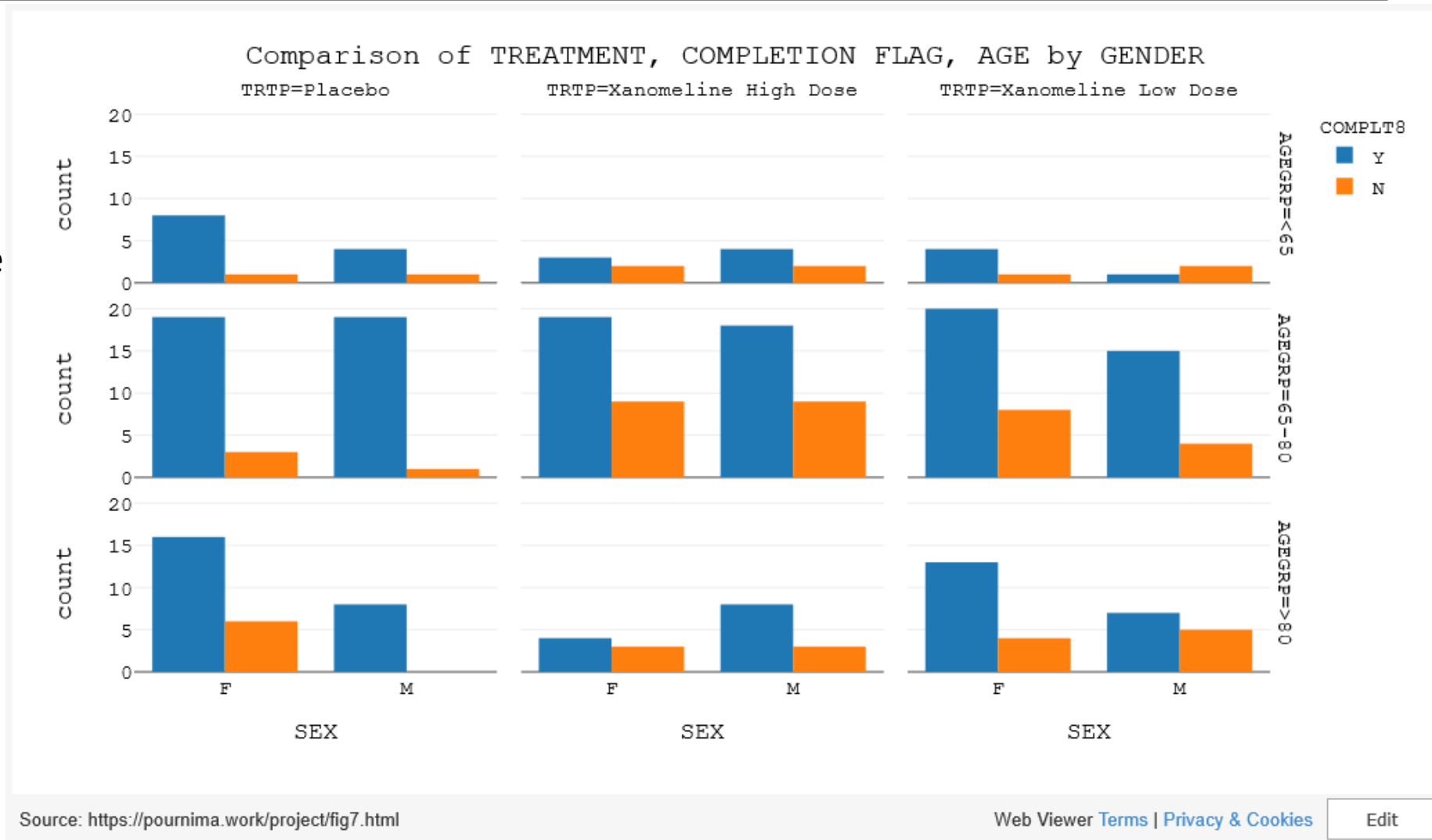
```
fig.add_trace(go.Box(  
    y = ['Concentration'],  
    name = "Only Whiskers",  
    boxpoints = False, # no data points  
    marker_color = 'rgb(84, 29, 7)',  
    line_color = 'rgb(84, 29, 7)'  
)
```





# Example 4: Interactive subplot

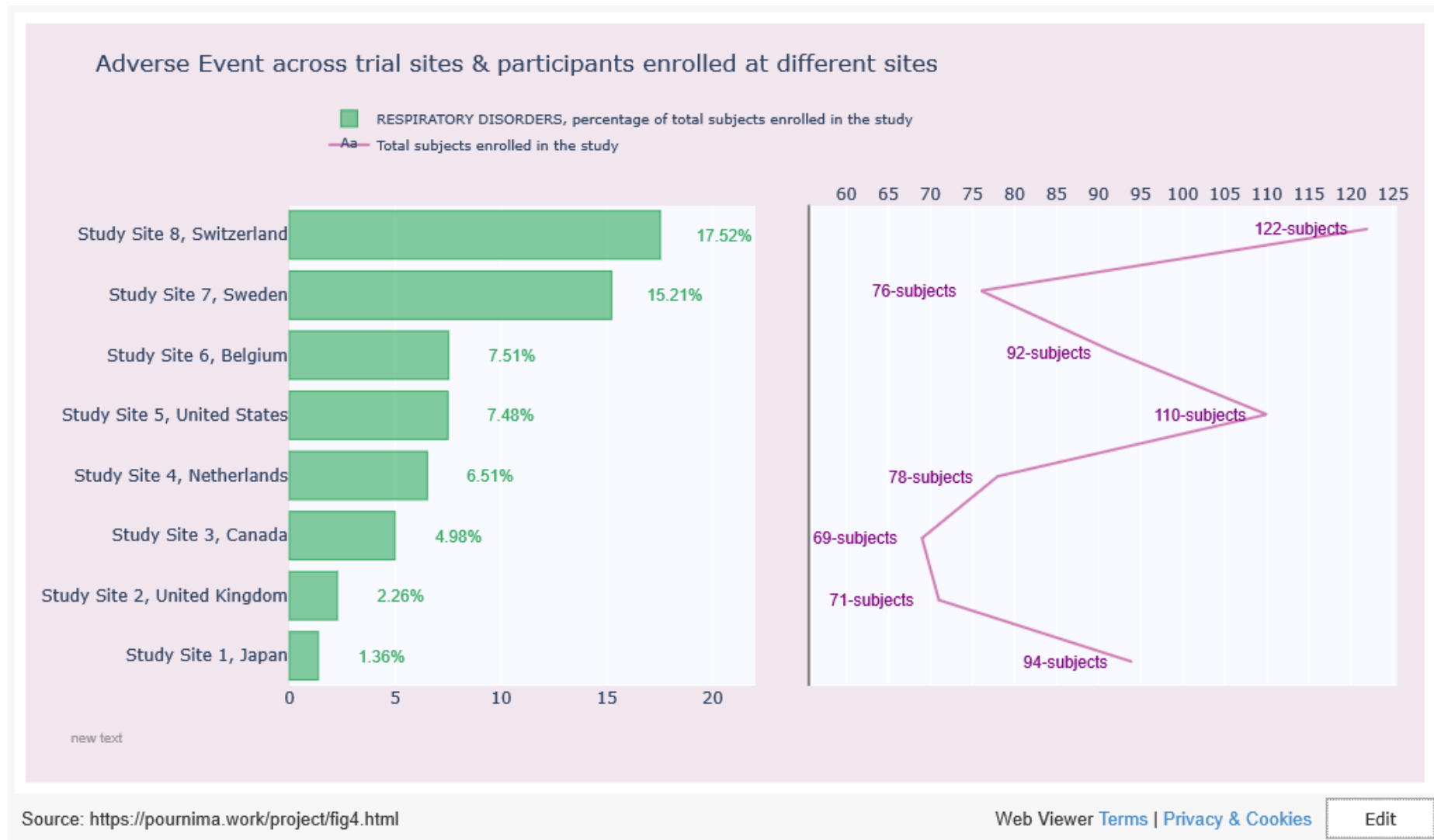
- ✓ Multiple axis
- ✓ Interactive column variable





# Example 5: Interactive bar-line chart

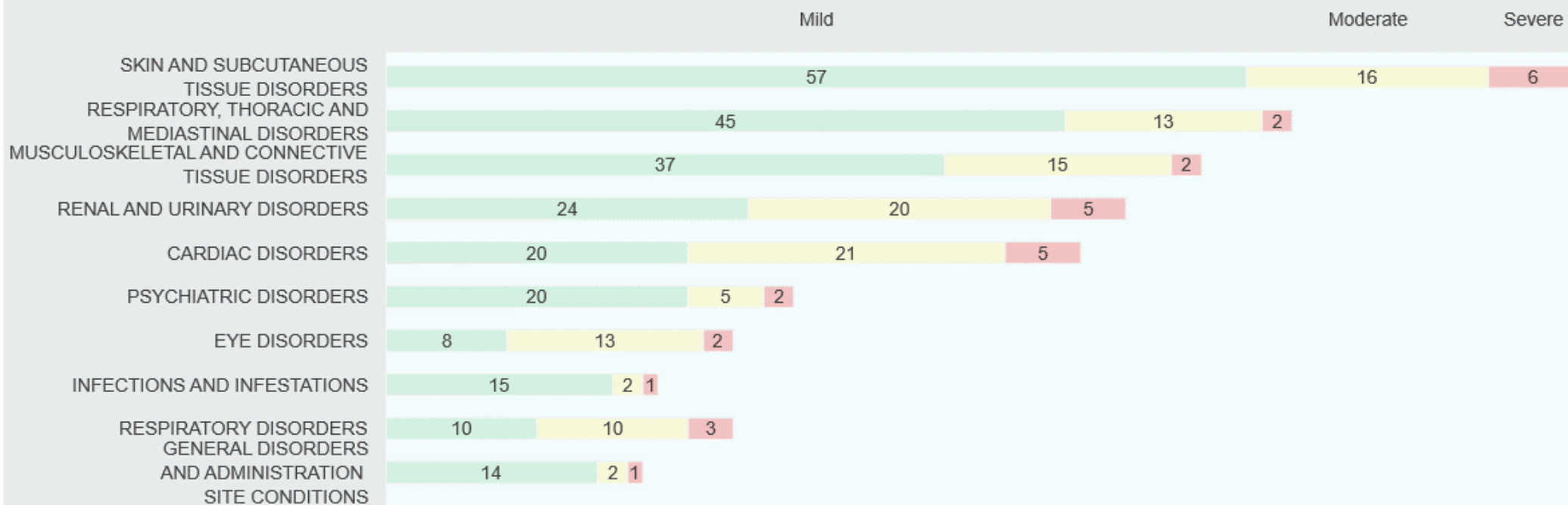
- ✓ Different chart types



# Example 6: Interactive horizontal bar



## Adverse Events and Severity



Source: <https://pournima.work/project/fig5.html>

Web Viewer [Terms](#) | [Privacy & Cookies](#)

Edit

# Template graphs in Python

---



➤ One time use graphs



✓ Reusable graphs

➤ Limited features



✓ Flexible features

➤ General



✓ Customized

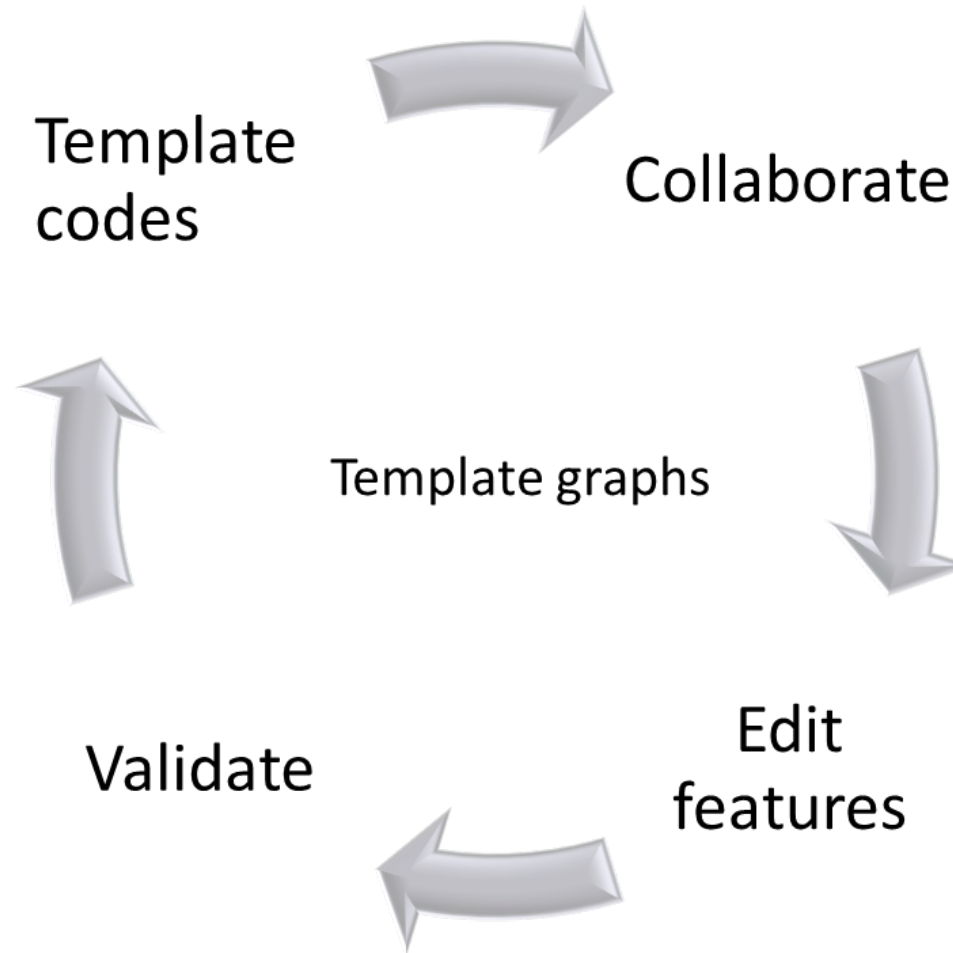
➤ No Version Control



✓ Version controlled

# Workflow

---



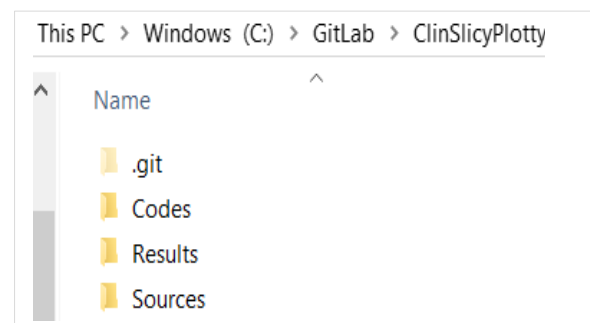
# Workflow

- ✓ `Open source
- ✓ Version control



## Git repository

|         |   |                                 |
|---------|---|---------------------------------|
| Codes   |   | Project name modified           |
| Results | ✕ | Sample codes and graphs updated |
| Sources | ✕ | Code_init_constructor created   |



## System architecture

```
def simplestackedbar(source, ylabel, xlabel, title, separator, destination, meantrend=False):  
    # -----  
    # Zip and arrange the plotter and the plotted  
    # -----  
    dcraw = pd.read_csv(source, sep=separator, error_bad_lines=False)  
  
    numerics = ['int16', 'int32', 'int64', 'float16', 'float32', 'float64']  
  
    newdf = dcraw.select_dtypes(include=numerics)  
  
    data = np.array(newdf)  
    data = np.transpose(data)  
    attributes = dcraw.columns[1:len(dcraw.columns)]  
    columns=dcraw.columns  
    N = np.shape(data)  
    x = dcraw[columns[0]]  
    ind = np.arange(len(x))  
  
    totalindex = N[1]  
    barsperindex = N[1] - 1  
  
    # -----  
    # Parameters  
    # -----
```

## Python template code





# Visualization of Git process

- Keep track of code activity
- Non-linear development
- Branch history



# Integration with SAS Viya

- Collaborate with non-Python users
- Faster processing



#CAS Connection

```
import swat
```

```
conn = swat.CAS("ocsnl-viya1.internal.ocs-consulting.com",  
                5570,"userid","password")
```



# Success factors

---



- Choosing the right tool and not replacing
- Experience in correlating data
- Collaboration
- Validation
- Open Mind



# Key Takeaways

---

- ✓ Data **visualization** becoming popular in the clinical world
- ✓ Customized library of **template graphs** for plotting
- ✓ Interactive plotting in **Python**
- ✓ Framework for collaboration using **Git**
- ✓ Integration with **SAS Viya** Platform



# Questions?

Email: [pournima.prabhakaran@ocs-consulting.com](mailto:pournima.prabhakaran@ocs-consulting.com)

LinkedIn: <https://www.linkedin.com/in/pournimaprabha/>