



SASPY in a Flask

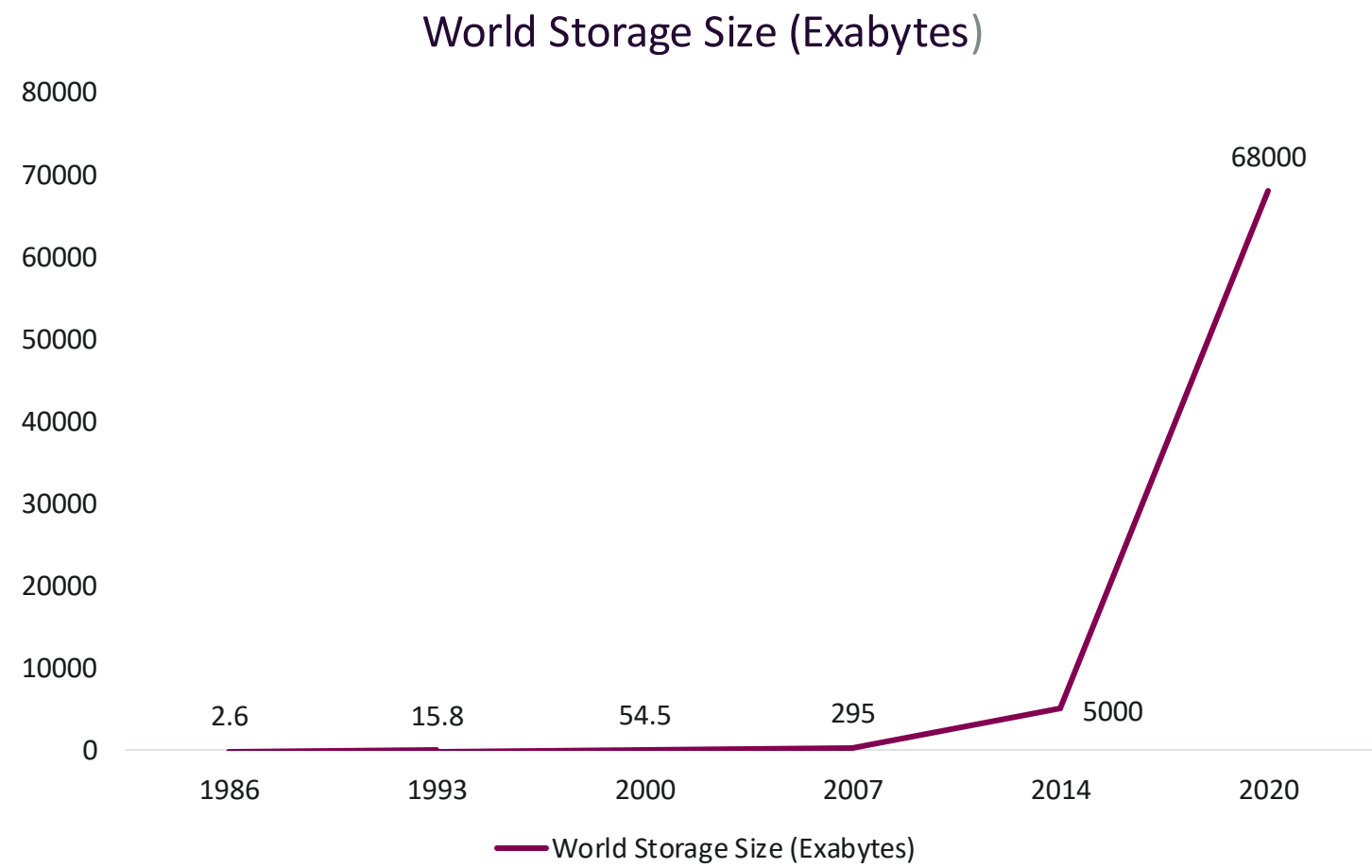
Jamie Pilkington

Statistical Programmer I

AstraZeneca

17/11/2025

This is not the data you are looking for



The Source Is Strong with This App



This is where the fun begins

Why use SAS as
a back-end?

- Broad codebase of historic programs
- The flexibility is what the industry expects

Key SAS
interactive
features

- Macros
- HTML



Perfectly balanced as all things should be

SAS Dataset creation	Dataframe and Array in Python																								
<pre>data data2_ex ; set data2 ; where a=2 or a=3 ; run ; proc print data=data2_ex ; run ;</pre>	<pre># Extract the records where a=2 or 3 data2_ex=data2[(data2.a==2) (data2.a==3)] print(data2_ex)</pre>																								
Output																									
<table><tr><th>Obs</th><th>a</th><th>b</th><th>c</th></tr><tr><td>1</td><td>2</td><td>4</td><td>6</td></tr><tr><td>2</td><td>3</td><td>6</td><td>9</td></tr></table>	Obs	a	b	c	1	2	4	6	2	3	6	9	<table><tr><th></th><th>a</th><th>b</th><th>c</th></tr><tr><td>1</td><td>2</td><td>4</td><td>6</td></tr><tr><td>2</td><td>3</td><td>6</td><td>9</td></tr></table>		a	b	c	1	2	4	6	2	3	6	9
Obs	a	b	c																						
1	2	4	6																						
2	3	6	9																						
	a	b	c																						
1	2	4	6																						
2	3	6	9																						

Choices	Description	Example code of data sorting in Jupyter notebook cell <>: SAS dataset, []: Pandas DataFrame
Pandas DataFrame	Pandas is a third-party package to handle one dimension data (Vector: Series) and 2 dimension data (Matrix: DataFrame) with Pandas analytic functions. Set "import Pandas" to use Pandas first.	[BBBB]=<AAAA>.sort_values(by=['USUBJID', 'AESEV'], ascending=[True, False])

SAS Operators	Python Operators
<pre>*--- DO and IF statement ; do i = 1 to 3 ; if i = 1 then y = 1 ; else if i = 2 then y = 2 ; else y = 3 ; end ;</pre>	<pre># for and if operators x = [1,2,3] for i in x: if i == 1: print('i=',1) elif i == 2: print('i=',2) else: print('i=',3)</pre>
<pre>*--- DO statement with decimal numbers ; do i = 10 to 11 by 0.1 ; output ; end ;</pre>	<pre># for operators with decimal numbers for x in range(10, 12, 1): for y in [0.1*z for z in range(10)]: x1 = round(x + y,1)</pre>
<pre>*--- Arithmetic operators ; data xxx ; x1 = 13 ; x2 = x1+3 ; x3 = x1-3 ; x4 = x1*3 ; x5 = round(x1/3, .001) ; x6 = int(x1/3) ; x7 = mod(x1,3) ; x8 = x1**3 ; run ;</pre>	<pre># Arithmetic operators x1 = 13 x2 = x1+3 x3 = x1-3 x4 = x1*3 x5 = round(x1/3, 3) x6 = x1//3 # divmod(x1,3) returns (4, 1) x7 = x1%3 x8 = x1**3</pre>
Results: 13 16 10 39 4.333 4 1 2197	

Table 7. Basic Logical and Arithmetic Operators in SAS and Python

Connecting between SAS and Python

- SASPY connection to ODA
- SAS submit function
- HTML Output

```
1 import saspy
2 sas = saspy.SASsession(
3     java='/usr/bin/java',
4     iomport=8591,
5     encoding='utf-8',
6     iomhost = ['odaws01-euw1.oda.sas.com', 'odaws02-euw1.oda.sas.com']
7 )
8 print(sas)
```

```
1 variable_change = "sex"
2 sas_code = f'''
3     proc sgplot data=sashelp.class;
4         vbar {variable_change};
5     run;
6 '''
7 sas_output = sas.submit(sas_code)
8 print(sas_output["LST"])
```

```
<!DOCTYPE html>
<html lang="en" xml:lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta charset="utf-8"/>
<meta content="SAS 9.4" name="generator"/>
<title>SAS Output</title>
<style>
```



Python Webapps

Dropdown Box

ADAM

WORK

TFLDATA

ADAM

SASDATA

STPSAMP

SASHELP

Buttons

Population Filter

X-axis Controls

Y-axis Controls

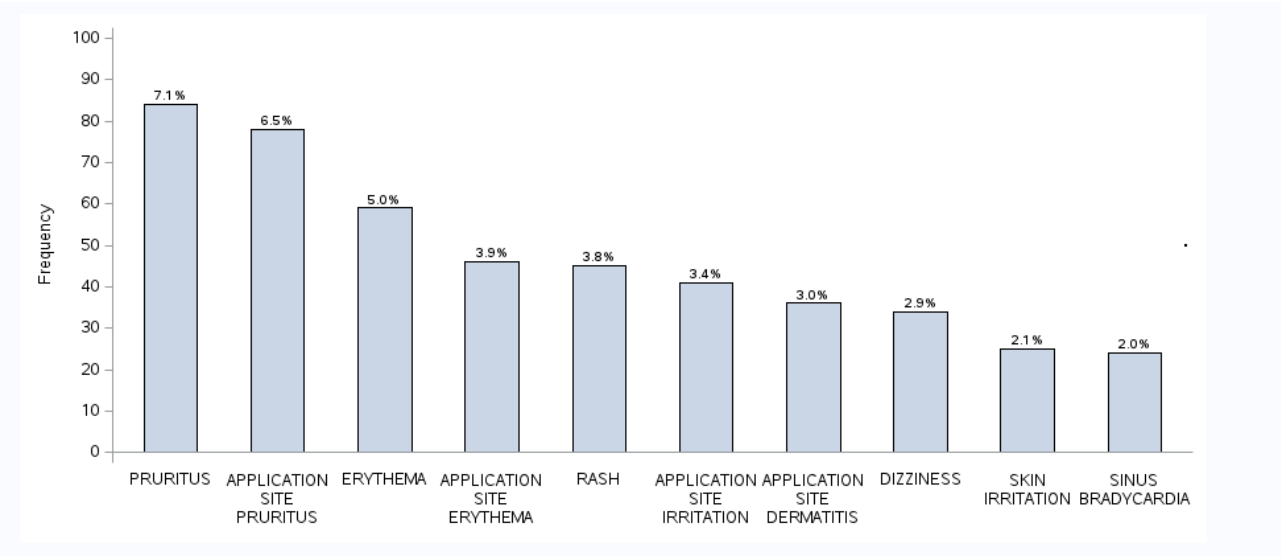
Update HTML Content

Iframes

Check Boxes

Population Flags:

- ☐SAFFL
- ☐TRTEMFL
- ☐AOCCFL
- ☐AOCCSFL
- ☐AOCCPFL
- ☐AOCC02FL
- ☐AOCC03FL
- ☐AOCC04FL
- ☐AOCC01FL



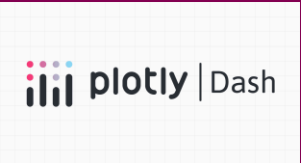
Flask vs Dash

Flask



- Creates a raw connection to a server
- General purpose for any web application
- Requires HTML, CSS and Javascript
- Has basic components built into it but lots of functionality must be code

Dash



- Built on flask and uses it to create the final website
- Specifically designed for dashboards
- All coding can be done in python
- Has more complicated components with built in styling but less flexibility



Flask Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Basic HTML Components Example</title>
</head>
<body>

  <!-- Checkbox -->
  <label>
    <input type="checkbox" id="myCheckbox">
    Check me
  </label>
  <br><br>

  <!-- Dropdown box -->
  <label for="myDropdown">Choose an option:</label>
  <select id="myDropdown">
    <option value="option1">Option 1</option>
    <option value="option2">Option 2</option>
    <option value="option3">Option 3</option>
  </select>
  <br><br>

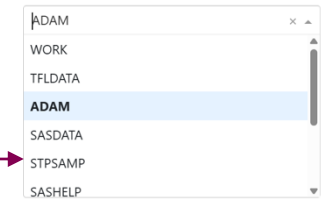
  <!-- Button -->
  <button type="button" id="myButton">Click Me</button>
  <br><br>

  <!-- Iframe with text using srcdoc -->
  <iframe srcdoc="<p>This is some text inside the iframe.</p>"
width="300" height="100" style="border:1px solid #ccc;"></iframe>

</body>
</html>
```

Population Flags:

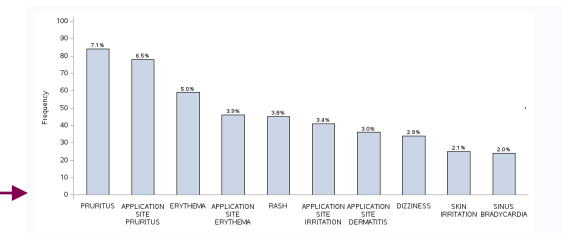
- ☐SAFFL
- ☐TRTEMFL
- ☐AOCCFL
- ☐AOCCSFL
- ☐AOCCPFL
- ☐AOCC02FL
- ☐AOCC03FL
- ☐AOCC04FL
- ☐AOCC01FL



Population Filter

X-axis Controls

Y-axis Controls



Dash Code

```
app.layout = html.Div([
    # Checkbox
    html.Label([
        dcc.Checklist(
            options=[{'label': 'Check me', 'value': 'checked'}],
            value=[],
            id='myCheckbox'))],
    html.Br(),

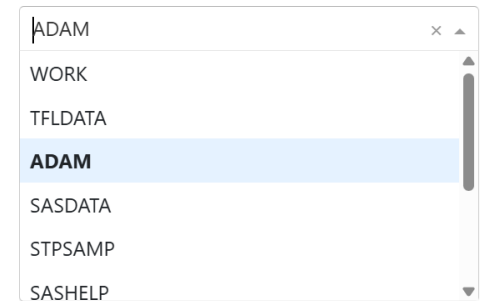
    # Dropdown box
    html.Label('Choose an option:'),
    dcc.Dropdown(
        options=[
            {'label': 'Option 1', 'value': 'option1'},
            {'label': 'Option 2', 'value': 'option2'},
            {'label': 'Option 3', 'value': 'option3'}],
        value='option1',
        id='myDropdown'),
    html.Br(),

    # Button
    html.Button('Click Me', id='myButton'),
    html.Br(),

    # Iframe-like component with text (using html.Iframe or a div)
    html.Label('Iframe Area:'),
    html.Iframe(
        srcDoc="<p>This is some text inside the iframe.</p>",
        style={'width': '300px', 'height': '100px', 'border': '1px solid #ccc'},
        id='myIframe'))]
```

Population Flags:

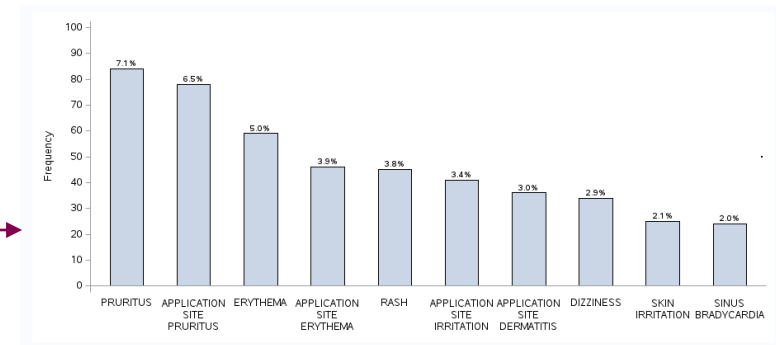
- ☐SAFFL
- ☐TRTEMFL
- ☐AOCCFL
- ☐AOCCSFL
- ☐AOCCPFL
- ☐AOCC02FL
- ☐AOCC03FL



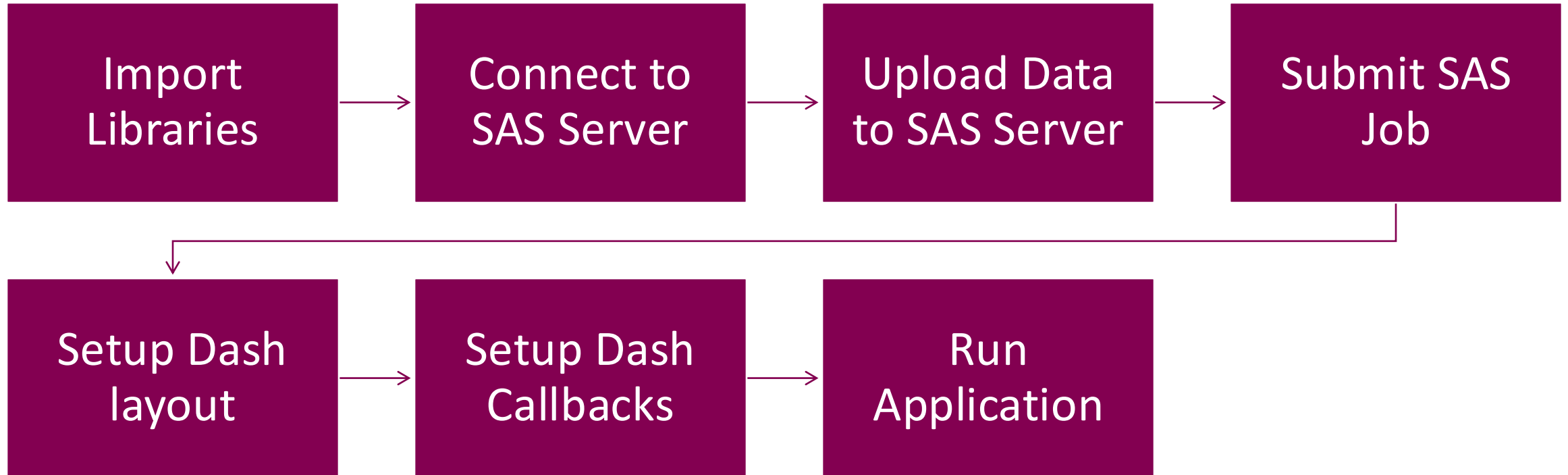
Population Filter

X-axis Controls

Y-axis Controls

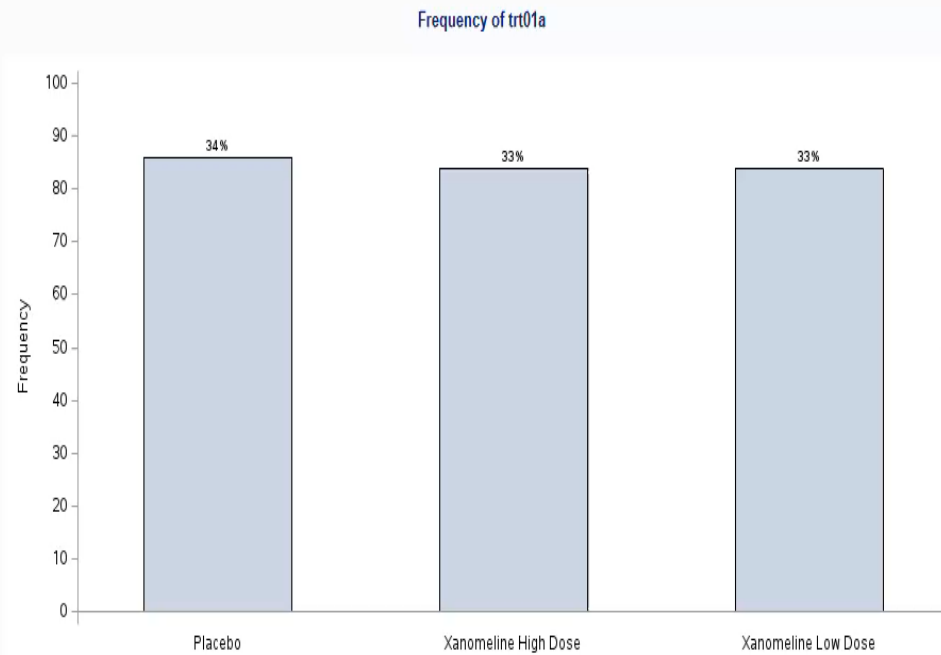


Dashboard Application Flowchart



Subject count by treatment group

SAS Based Data Dashboard



Population Filter

X-axis Controls

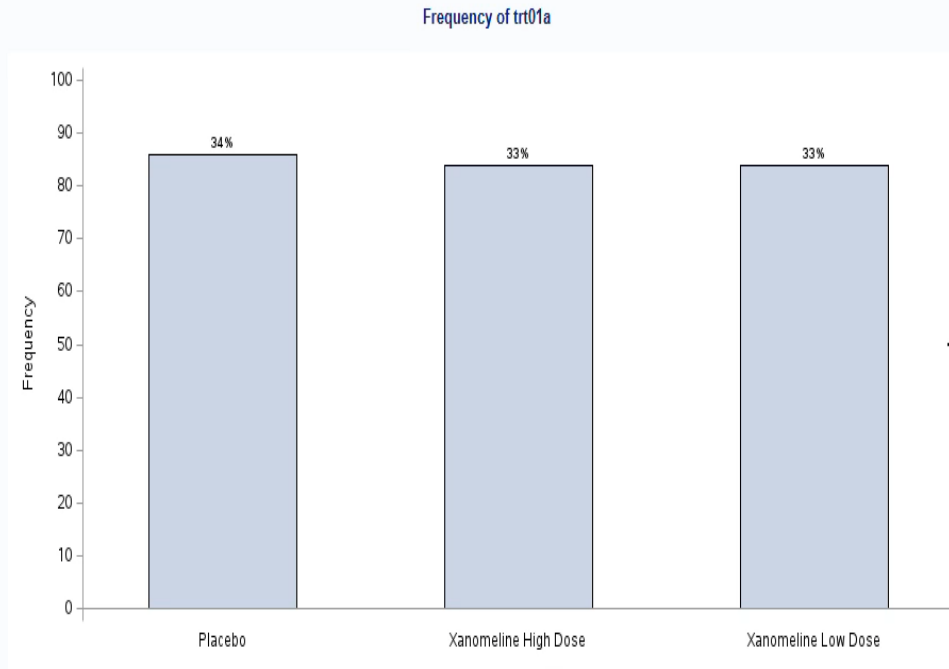
Y-axis Controls

Update HTML Content



Subject count by treatment group

SAS Based Data Dashboard



Population Filter

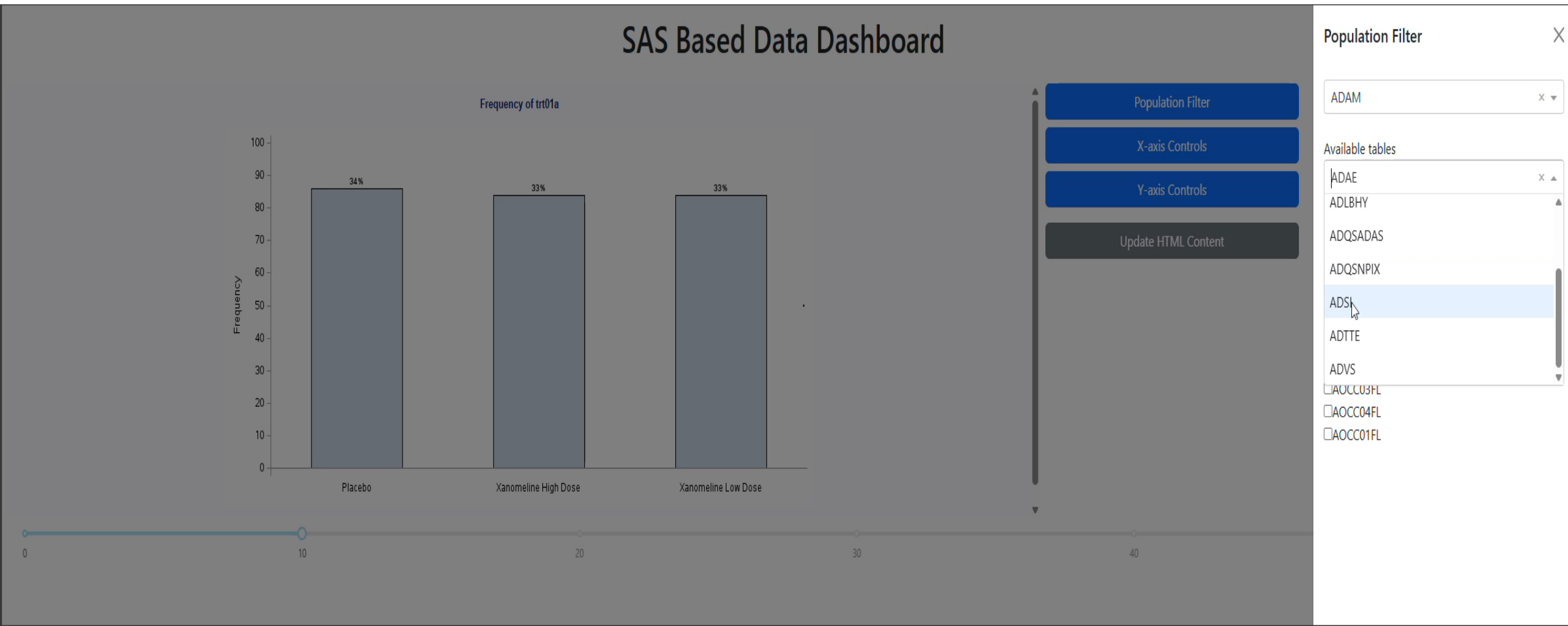
X-axis Controls

Y-axis Controls

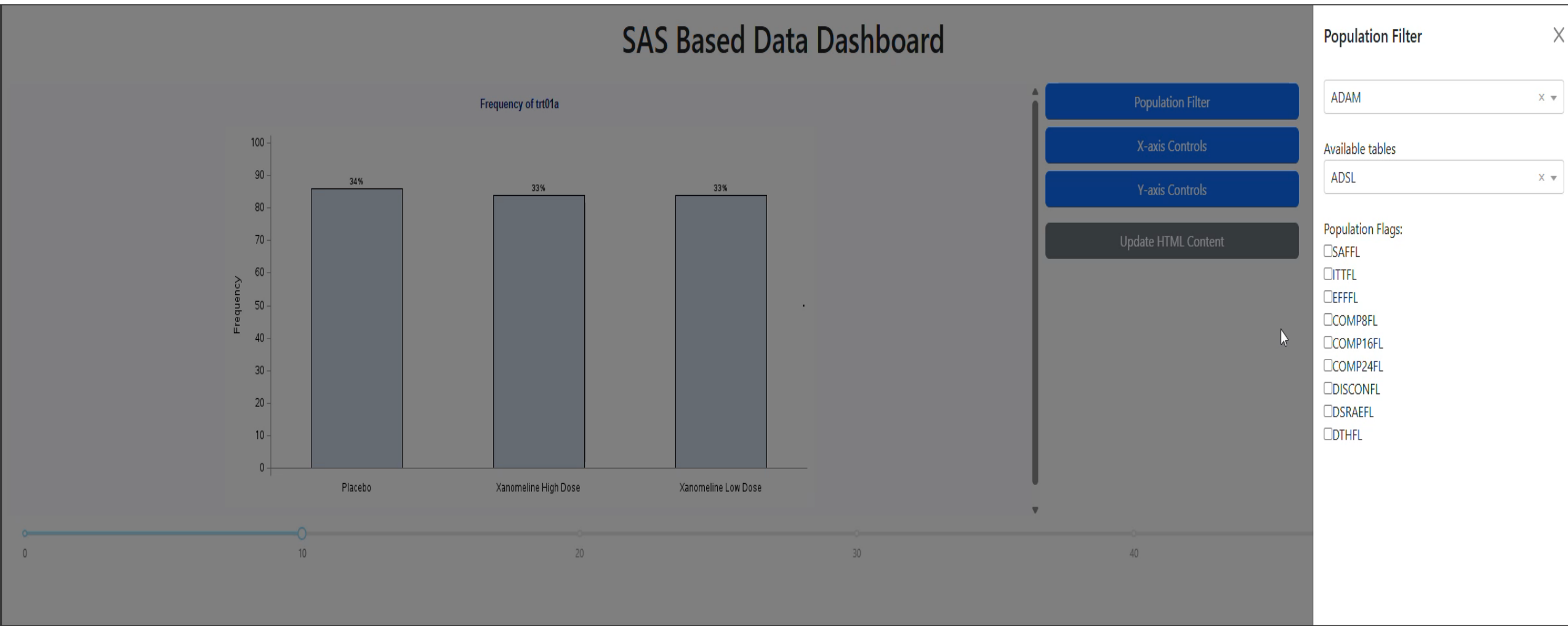
Update HTML Content



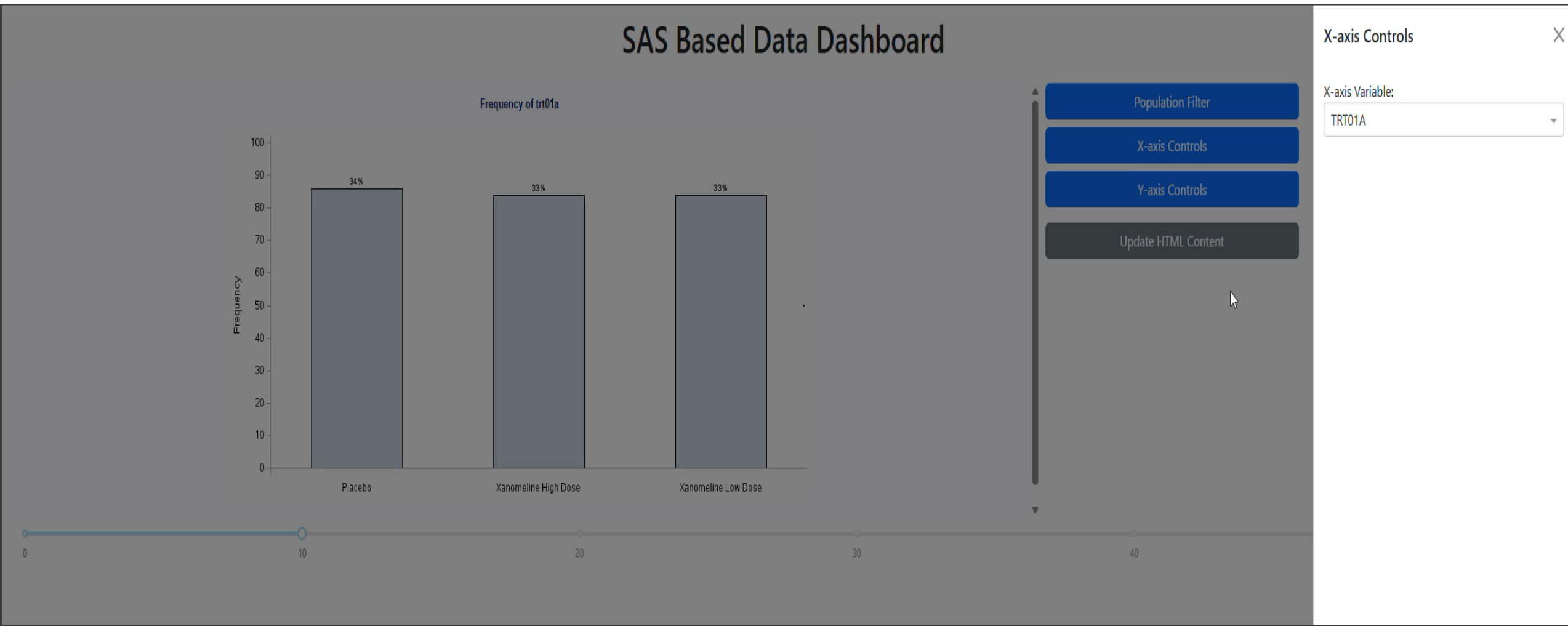
Subject count by treatment group



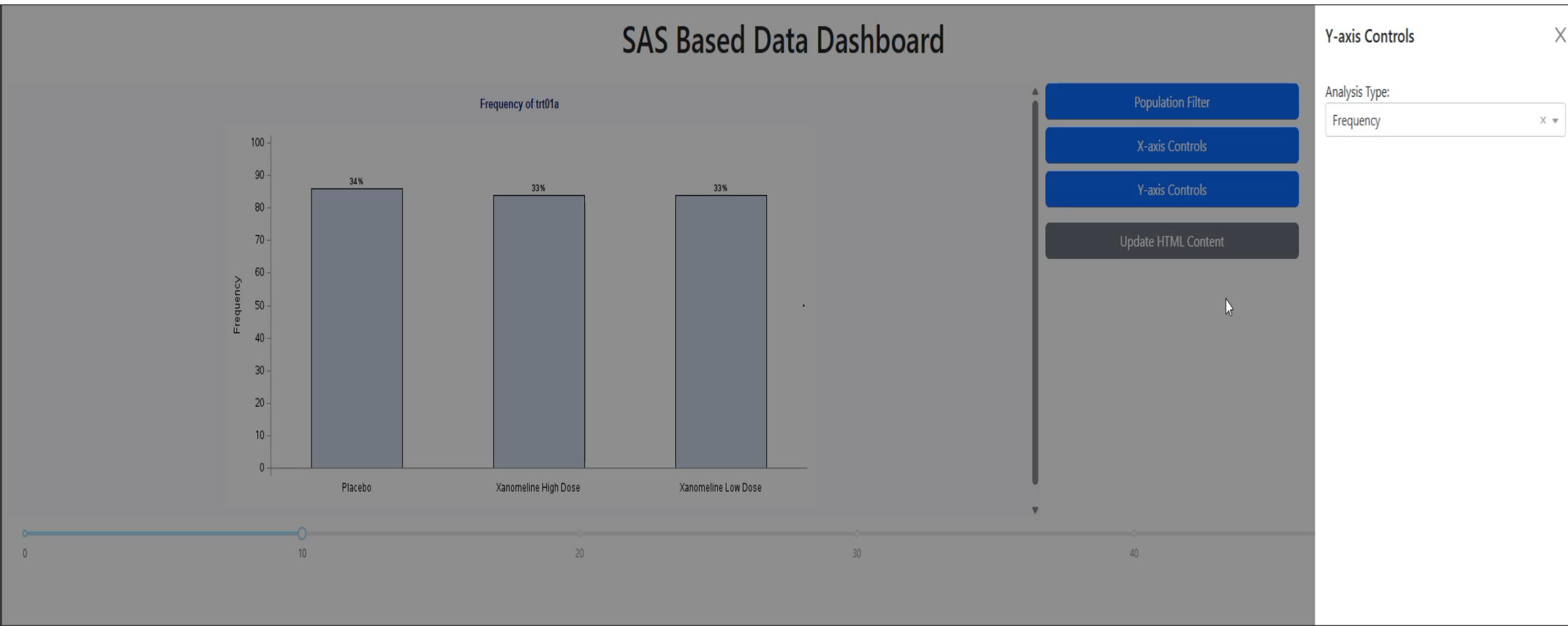
Subject count by treatment group



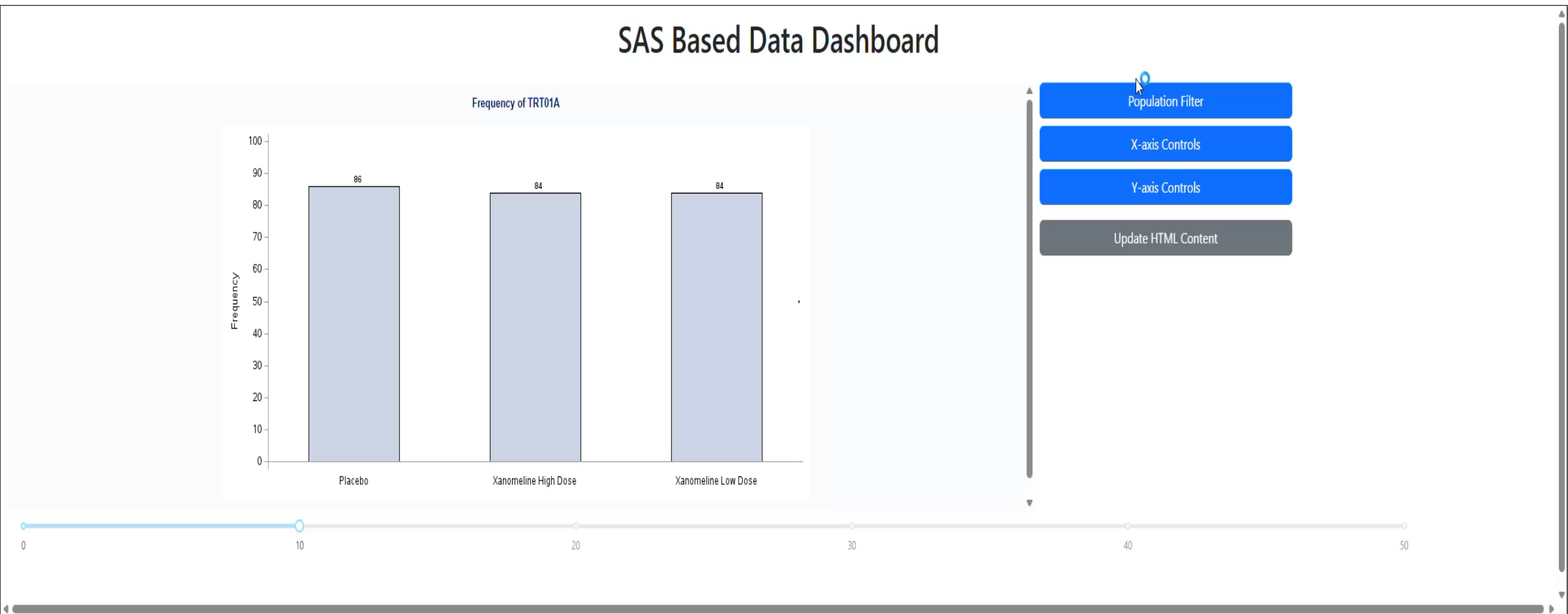
Subject count by treatment group



Subject count by treatment group

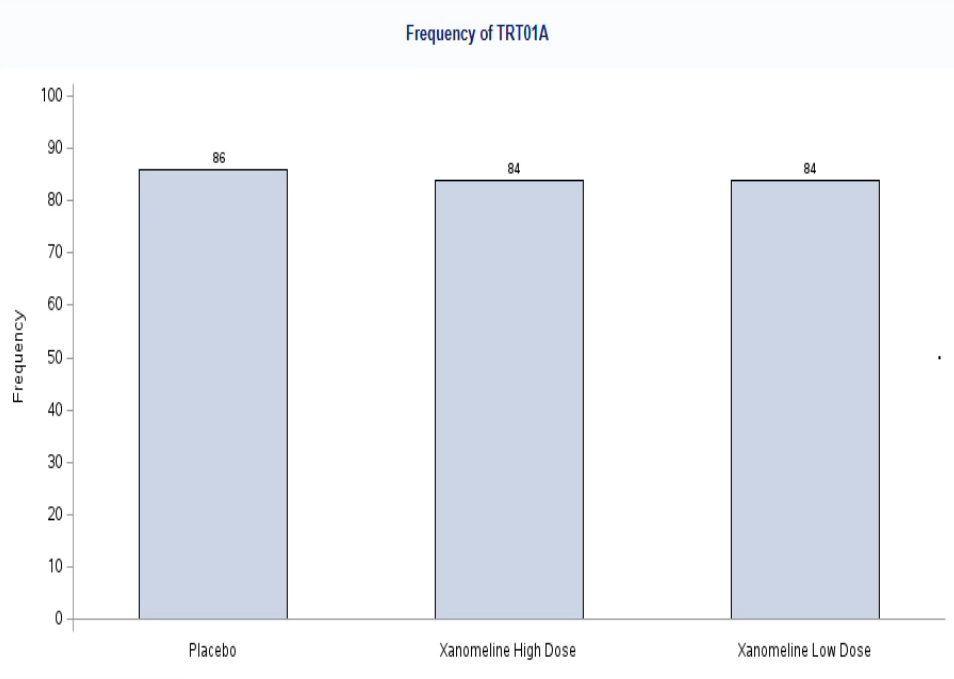


Frequency of Deaths



Frequency of Deaths

SAS Based Data Dashboard



Population Filter

X-axis Controls

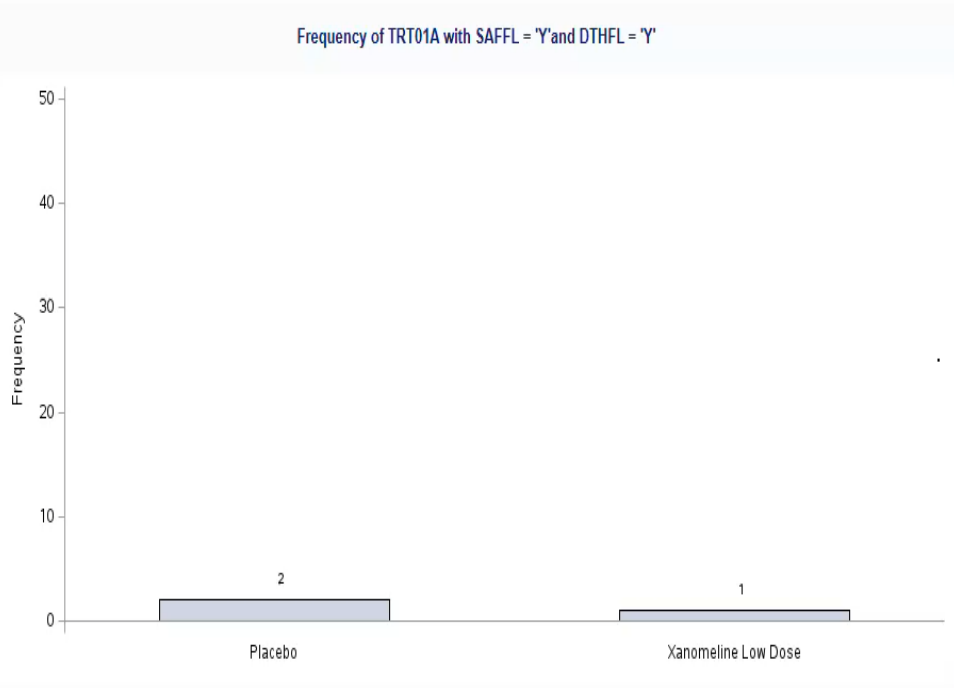
Y-axis Controls

Update HTML Content



Frequency of AEs by AETERM

SAS Based Data Dashboard



Population Filter

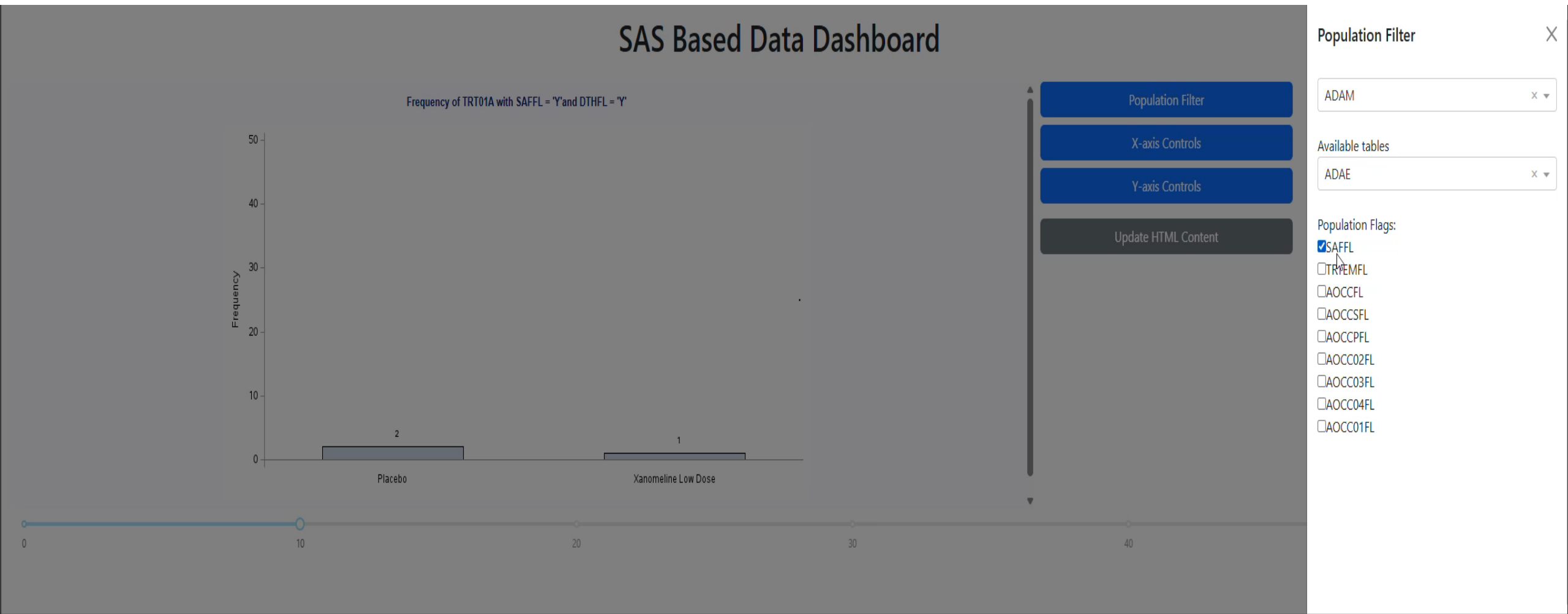
X-axis Controls

Y-axis Controls

Update HTML Content

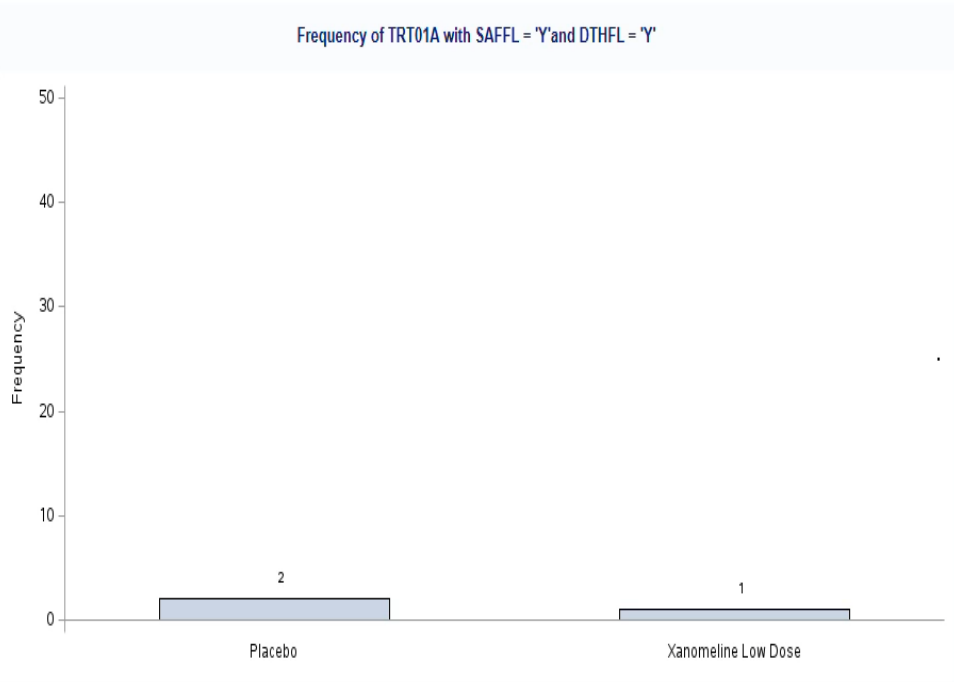


Frequency of Aes by AETERM



Frequency of AEs by AETERM

SAS Based Data Dashboard



Population Filter

X-axis Controls

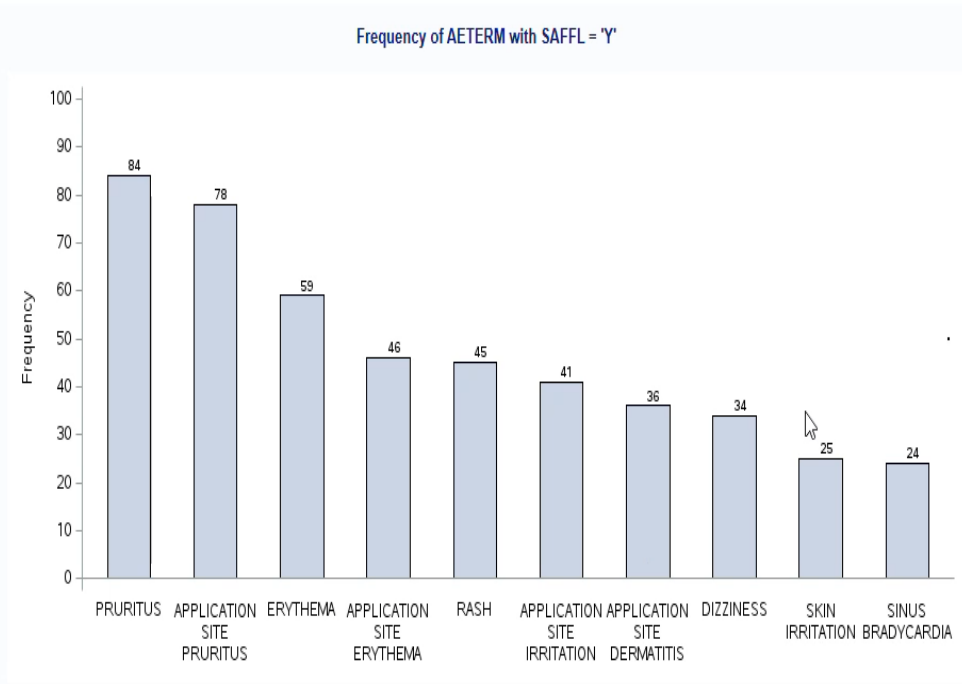
Y-axis Controls

Update HTML Content



Frequency of AEs by AETERM

SAS Based Data Dashboard



Population Filter

X-axis Controls

Y-axis Controls

Update HTML Content



How to get started

Google Account

- Go to [Colab](#)
- Load up example notebook

SAS ODA

- Sign up for SAS On demand for Academics
- Make a note of the filepath where you want to load the data and programs

Editing Colab

- Update the filepaths in the code to be the ones from your ODA account
- Hit run, wait for the data to upload then scroll down to see the demo



References

Bartley, K. (2020). *Big Data Statistics: How Much Data Is There in the World?* Rivery.
Available at: <https://rivery.io/blog/big-data-statistics-how-much-data-is-there-in-the-world/>.

Flask (2010). *Welcome to Flask — Flask Documentation (3.0.x)*. Palletsprojects.com.
Available at: <https://flask.palletsprojects.com/en/stable/>.

Gottula, J. (2020). *How SAS grew from its strong roots in agriculture*. SAS Voices.
Available at: <https://blogs.sas.com/content/sascom/2020/06/22/sas-born-on-a-farm-raised-on-drugs-and-grew-up-in-the-industry/>.

Lankham, I. and Slaughter, M. (2019). *Everything is better with friends: Executing SAS ® code in Python scripts with SASPy*.
Available at: <https://support.sas.com/resources/papers/proceedings19/3189-2019.pdf>.

Nakajima, Y. and Pharma, N. (n.d.). *Utilization of Python in clinical study by SASPy*.
Available at: <https://support.sas.com/resources/papers/proceedings19/3191-2019.pdf>.

Plotly (n.d.). *Dash Documentation & User Guide | Plotly*. dash.plotly.com.
Available at: <https://dash.plotly.com/>.

Takanami, Y. and Pharmaceuticals, T. (2019). *Interaction between SAS ® and Python for Data Handling and Visualization*.
Available at: <https://support.sas.com/resources/papers/proceedings19/3260-2019.pdf>.

Weber, T. (2021). *SASPy — saspy 5.103.2 documentation*. Github.io.
Available at: <https://sassoftware.github.io/saspy/>



Confidentiality Notice

This file is private and may contain confidential and proprietary information. If you have received this file in error, please notify us and remove it from your system and note that you must not copy, distribute or take any action in reliance on it. Any unauthorized use or disclosure of the contents of this file is not permitted and may be unlawful.

AstraZeneca PLC, 1 Francis Crick Avenue, Cambridge Biomedical Campus, Cambridge, CB2 0AA, UK
+44(0)203 749 5000
www.astrazeneca.com

