```
1 import pandas as pd
```

```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
1 vr = pd.read_csv('/content/drive/MyDrive/VRImmersiondataset.csv')
```

```
1 vr.head()
```

| | headphones | width | height | fov | fps | stereopsis | antialiasing | textures |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1641.3370 | 911.8539 | 37 | 60 | 1 | 1 | 1 |
| **1** | 1 | 1282.1180 | 712.2877 | 39 | 55 | 1 | 1 | 1 |
| **2** | 1 | 1795.5310 | 997.5171 | 40 | 47 | 1 | 1 | 0 |
| **3** | 2 | 838.4885 | 465.8270 | 44 | 48 | 0 | 0 | 1 |
| **4** | 2 | 1924.7020 | 1069.2790 | 85 | 35 | 1 | 1 | 1 |

5 rows × 24 columns

```
# five number summary
vr.describe()
```

| | headphones | width | height | fov | fps | stereopsis | anti |
|---|---|---|---|---|---|---|---|
| **count** | 401.000000 | 401.000000 | 401.000000 | 401.000000 | 401.000000 | 401.000000 | 4 |
| **mean** | 1.149626 | 1204.965531 | 669.425287 | 64.628429 | 37.264339 | 0.541147 | |
| **std** | 0.701823 | 574.094938 | 318.941619 | 20.599492 | 13.359077 | 0.498927 | |
| **min** | 0.000000 | 216.410400 | 120.228000 | 30.000000 | 15.000000 | 0.000000 | |
| **25%** | 1.000000 | 719.343100 | 399.635000 | 46.000000 | 25.000000 | 0.000000 | |
| **50%** | 1.000000 | 1198.628000 | 665.904200 | 64.000000 | 38.000000 | 1.000000 | |
| **75%** | 2.000000 | 1714.976000 | 952.764500 | 84.000000 | 49.000000 | 1.000000 | |
| **max** | 2.000000 | 2155.729000 | 1197.627000 | 99.000000 | 60.000000 | 1.000000 | |

8 rows × 24 columns

```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

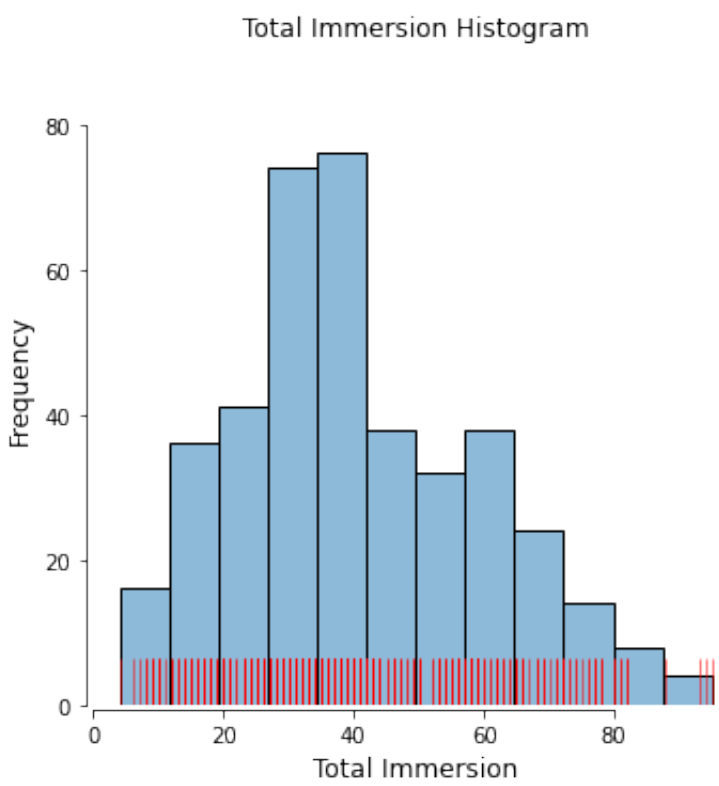
```
import cufflinks as cf
cf.go_offline()
cf.set_config_file(offline=False, world_readable=True)
```

```
def enable_plotly_in_cell():
  import IPython
  from plotly.offline import init_notebook_mode
  display(IPython.core.display.HTML('''<script
  src="/static/components/requirejs/require.js"></script>'''))
  init_notebook_mode(connected=False)
```

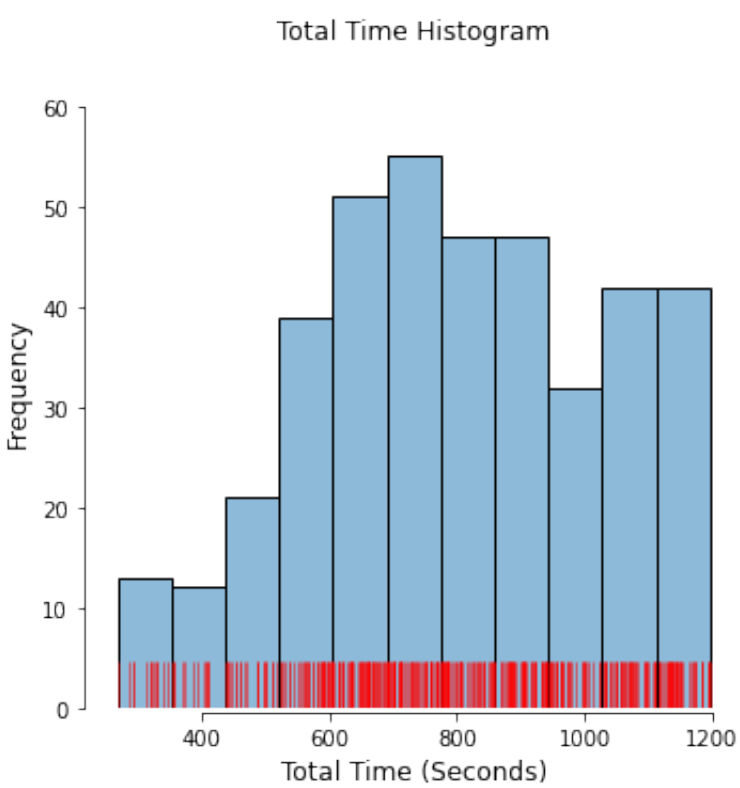
```
vr.columns
```

```
Index(['headphones', 'width', 'height', 'fov', 'fps', 'stereopsis',
       'antialiasing', 'textures', 'lightMode', 'saturation', 'brightness',
       'contrast', 'sharpness', 'shadowStrength', 'reflections',
       'modelsDetail', 'dof', 'particles', 'locomotion', 'ambientSound',
       'reverbZone', 'spatialSound', 'time', 'totalImmersion'],
      dtype='object')
```

```
sns.displot(vr['totalImmersion'],
            rug = True,
            rug_kws={'color':'r','alpha':0.5,'height':0.07},
            alpha = 0.5
            )
plt.title('Total Immersion Histogram')
plt.xlabel('Total Immersion', fontsize = 12)
plt.ylabel('Frequency', fontsize=12)

# clean up final results
sns.despine(trim = True, offset = 2);
```

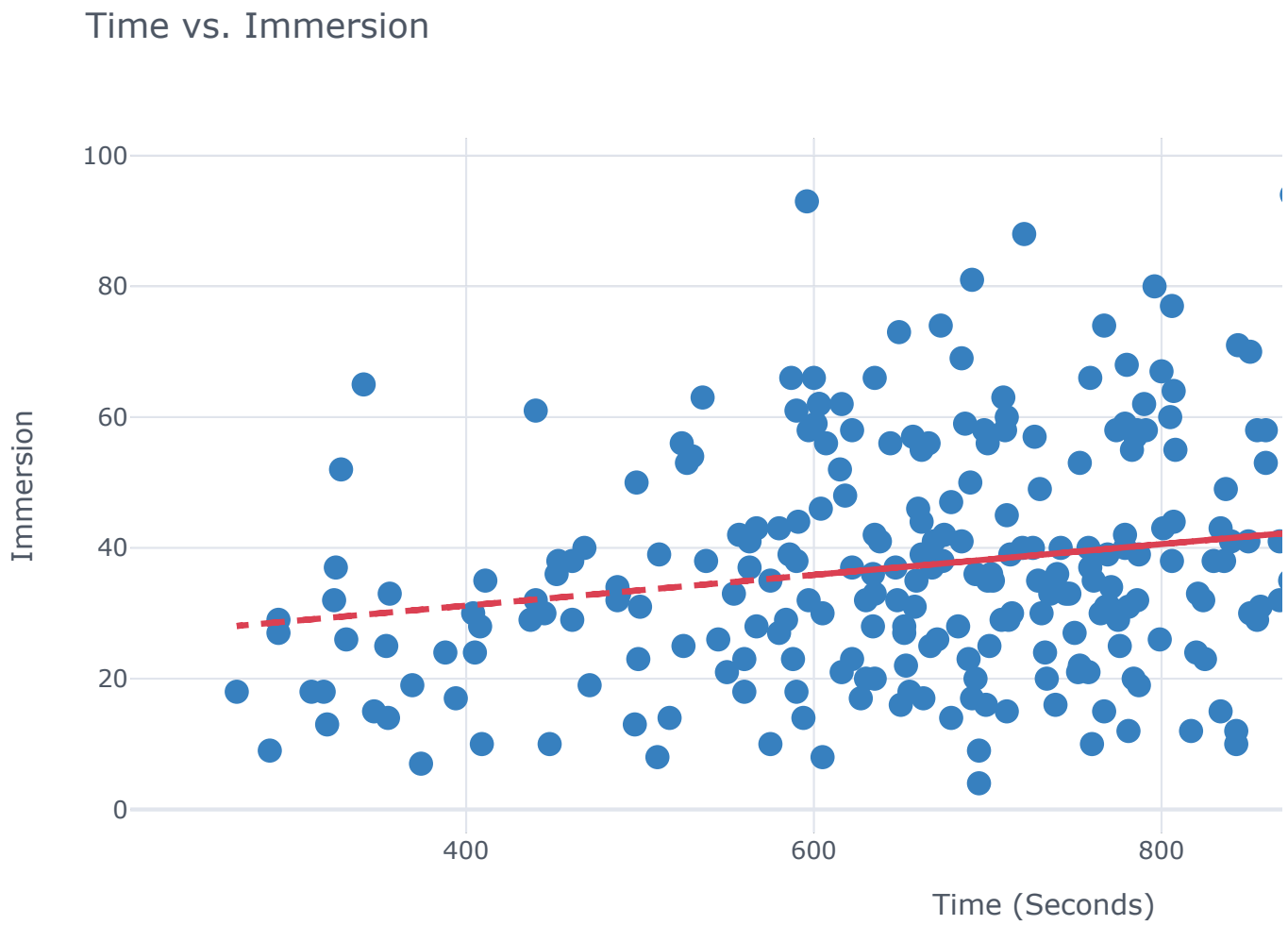


```
sns.displot(vr['time'],
            rug = True,
            rug_kws={'color':'r','alpha':0.5,'height':0.07},
            alpha = 0.5
            )
plt.title('Total Time Histogram')
plt.xlabel('Total Time (Seconds)', fontsize = 12)
plt.ylabel('Frequency', fontsize=12)

# clean up final results
sns.despine(trim = True, offset = 2);
```

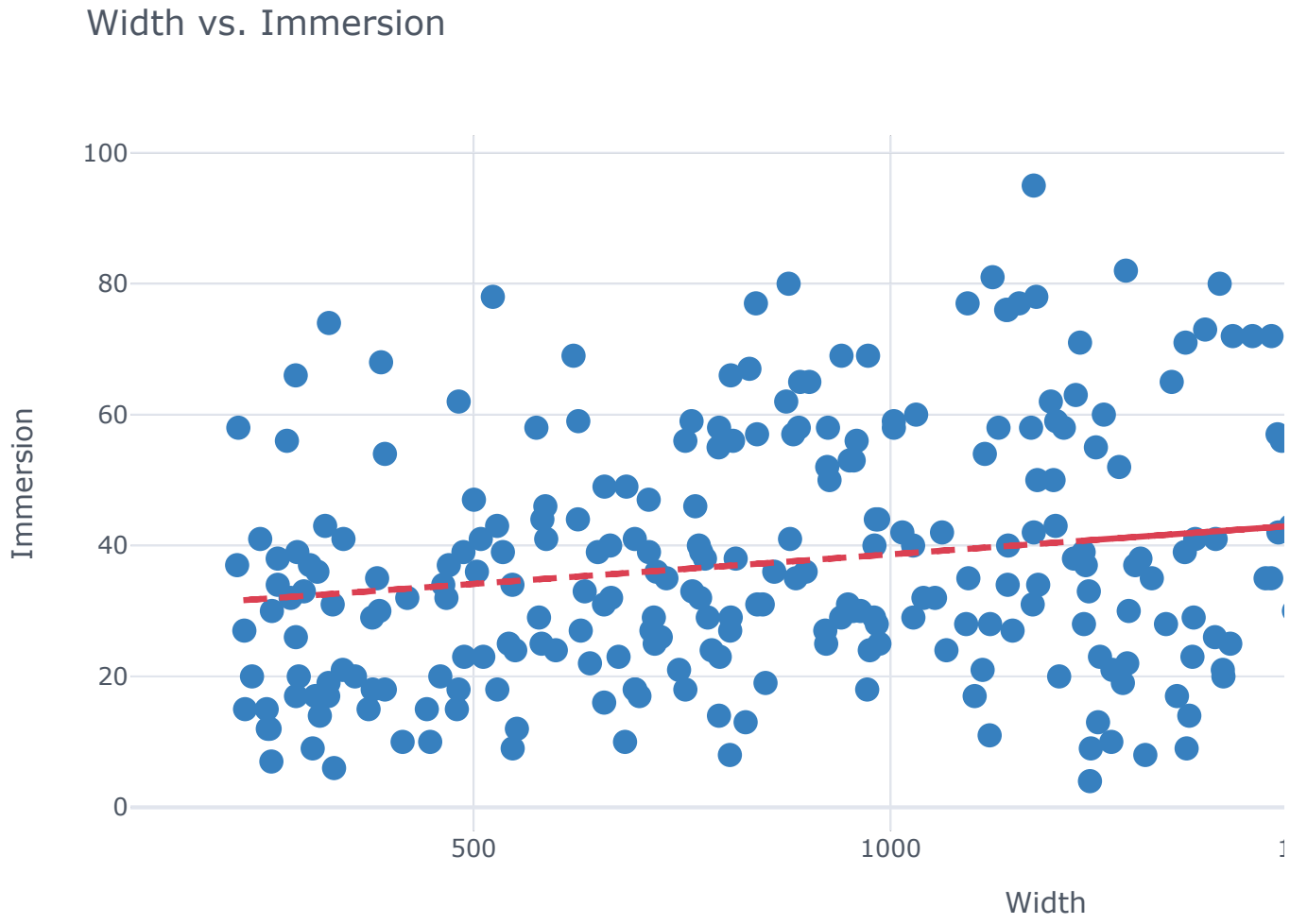


```
enable_plotly_in_cell()
vr.iplot(kind='scatter',
          x='time',
          y='totalImmersion',
          bestfit=True,
          bestfit_colors=['red'],
          mode='markers',
          color=['blue'],
          title='Time vs. Immersion',
          xTitle='Time (Seconds)',
          yTitle='Immersion')
```

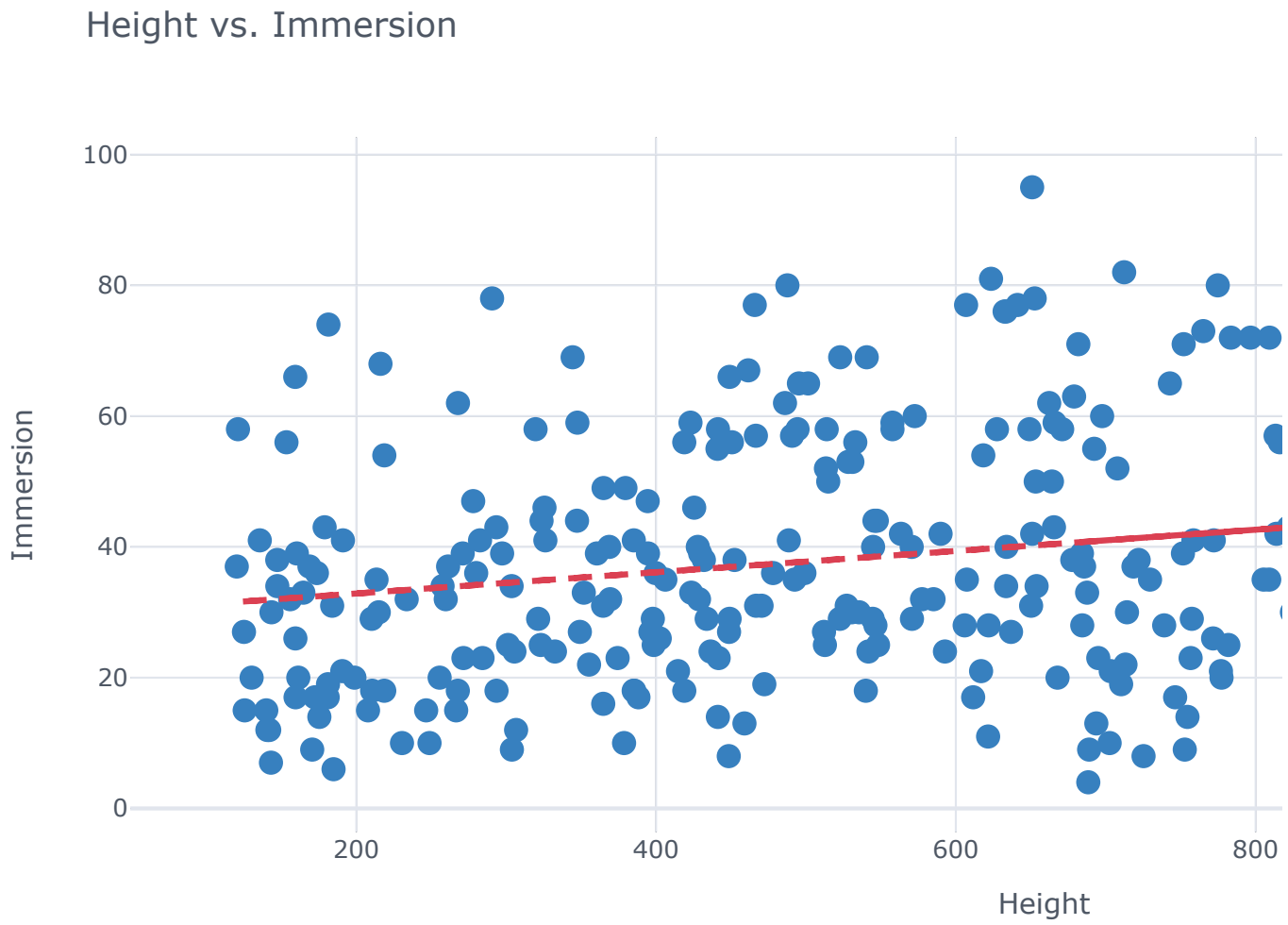
```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: Futur
  pandas.util.testing is deprecated. Use the functions in the public API at panc
```



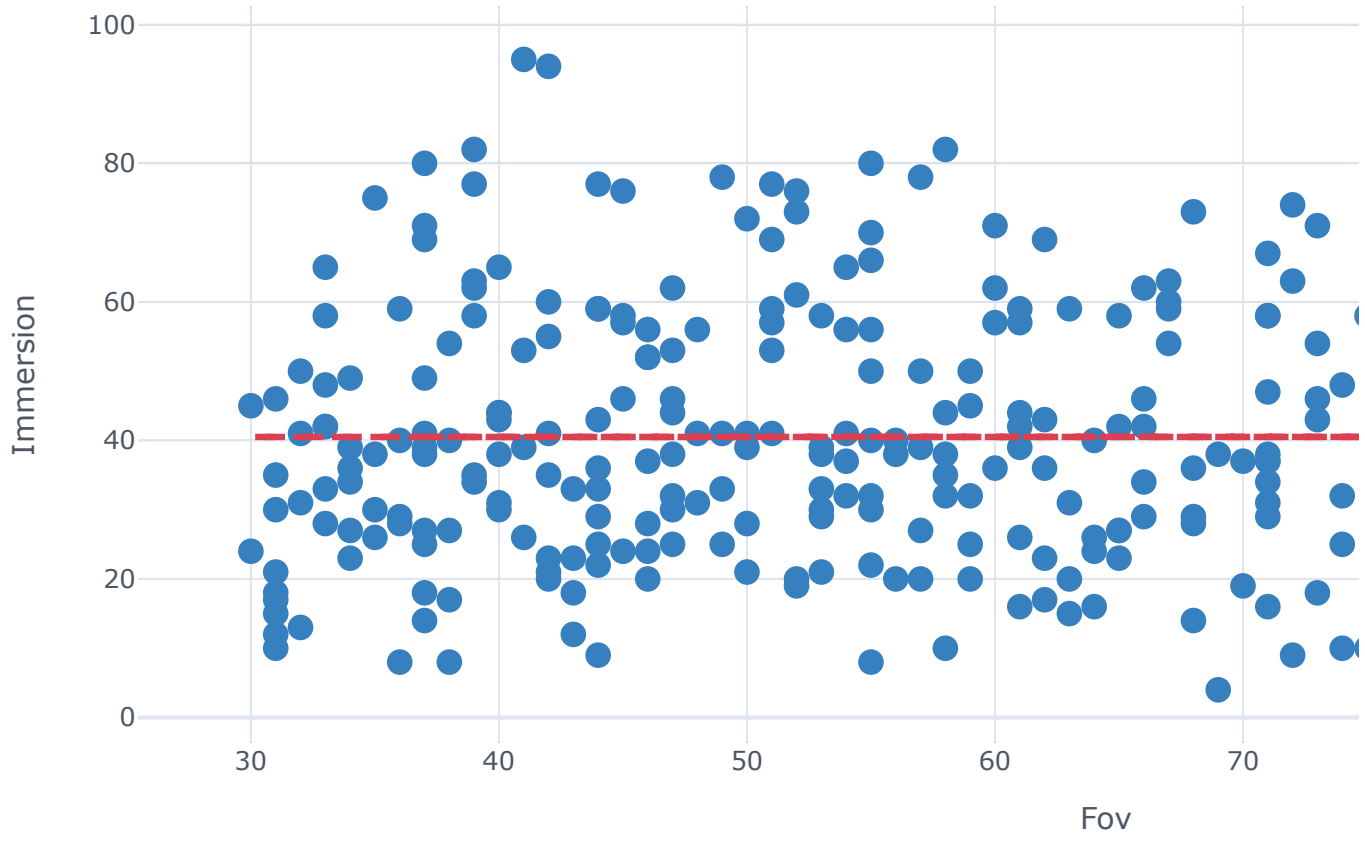
```
enable_plotly_in_cell()
vr.iplot(kind='scatter',
           x='width',
           y='totalImmersion',
           bestfit=True,
           bestfit_colors=['red'],
           mode='markers',
           color=['blue'],
           title='Width vs. Immersion',
           xTitle='Width',
           yTitle='Immersion')
```



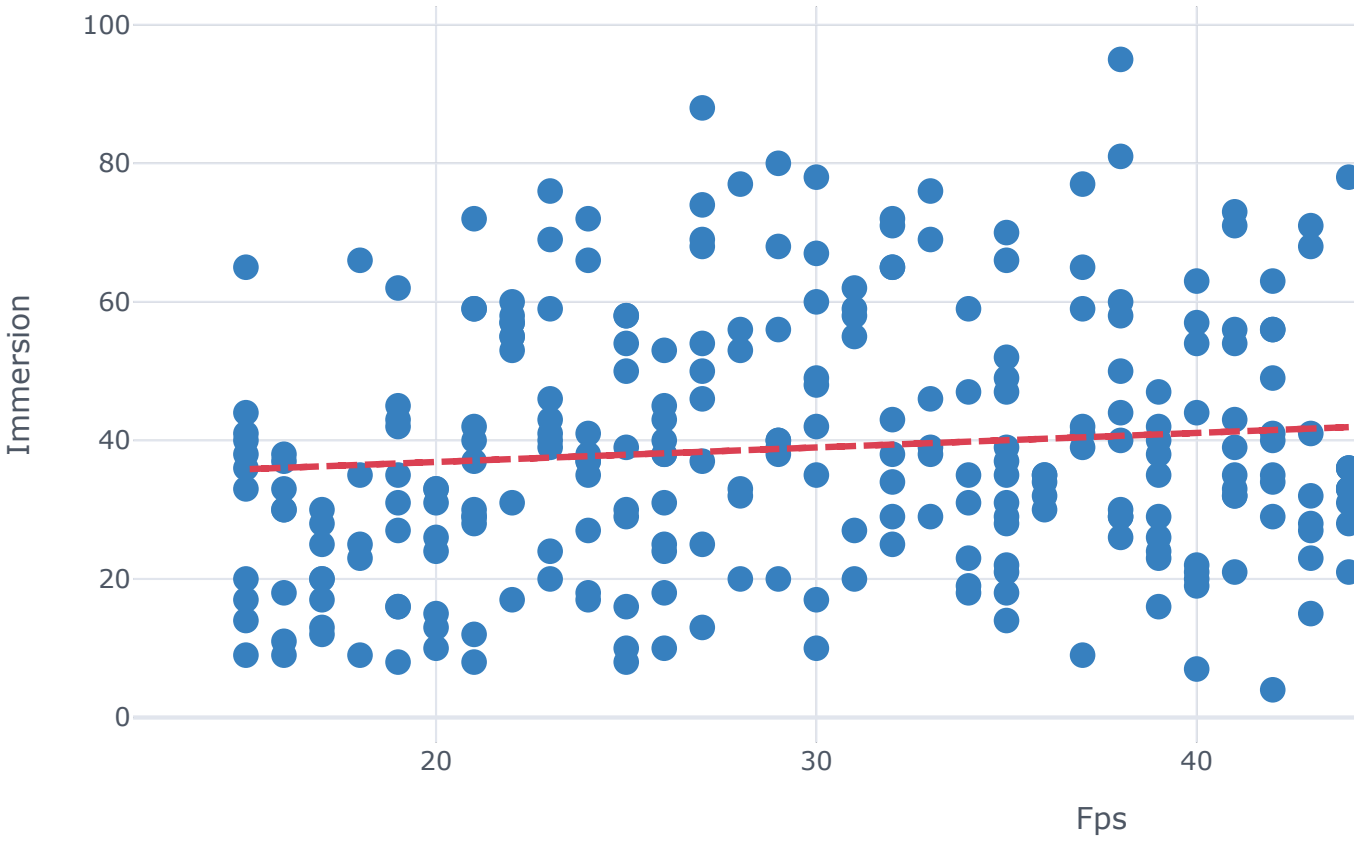
```
enable_plotly_in_cell()
vr.iplot(kind='scatter',
           x='height',
           y='totalImmersion',
           bestfit=True,
           bestfit_colors=['red'],
           mode='markers',
           color=['blue'],
           title='Height vs. Immersion',
           xTitle='Height',
           yTitle='Immersion')
```



```
enable_plotly_in_cell()
vr.iplot(kind='scatter',
           x='fov',
           y='totalImmersion',
           bestfit=True,
           bestfit_colors=['red'],
           mode='markers',
           color=['blue'],
           title='Fov vs. Immersion',
           xTitle='Fov',
           yTitle='Immersion')
```

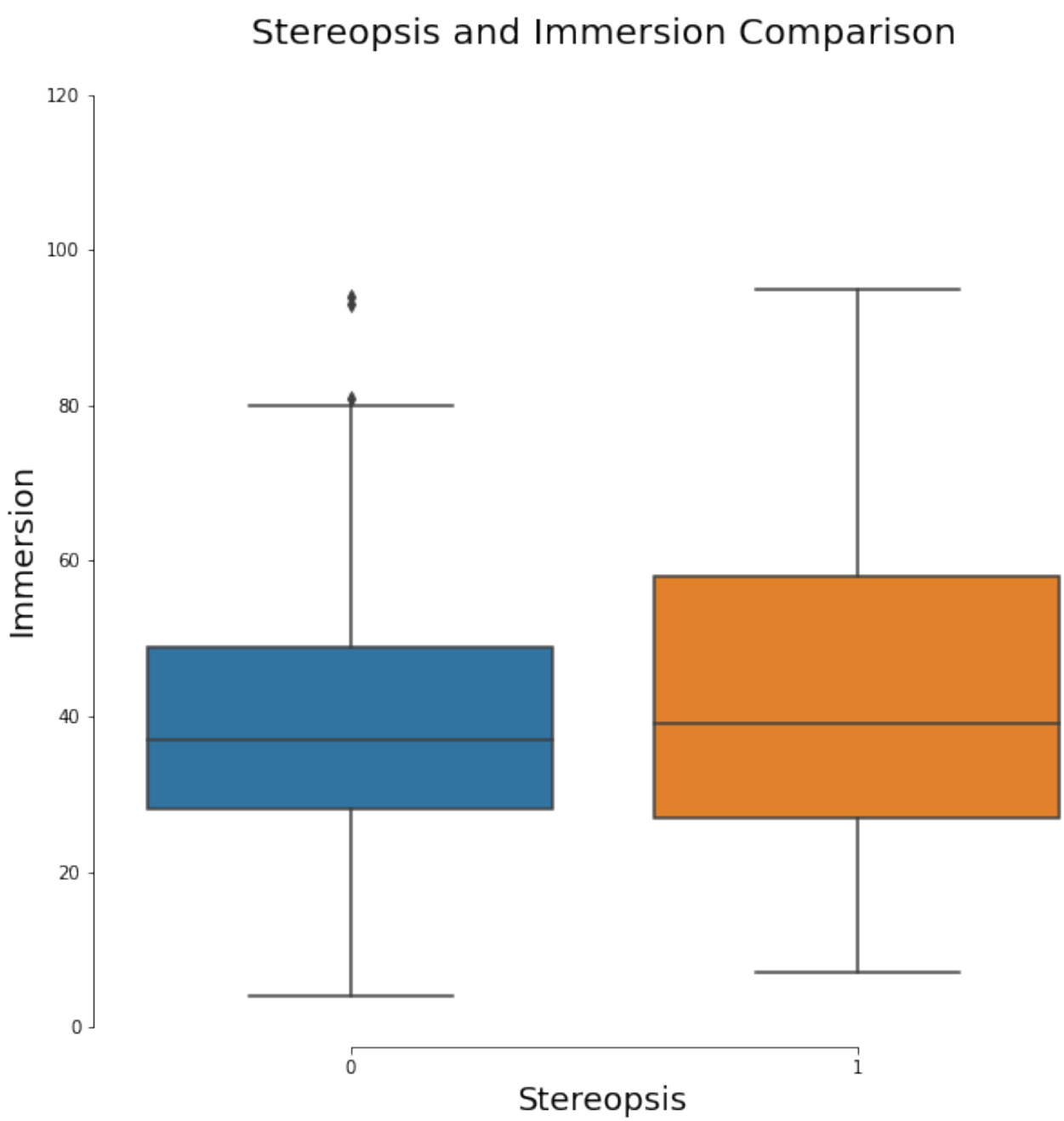


```
enable_plotly_in_cell()
vr.iplot(kind='scatter',
           x='fps',
           y='totalImmersion',
           bestfit=True,
           bestfit_colors=['red'],
           mode='markers',
           color=['blue'],
           title='Fps vs. Immersion',
           xTitle='Fps',
           yTitle='Immersion')
```



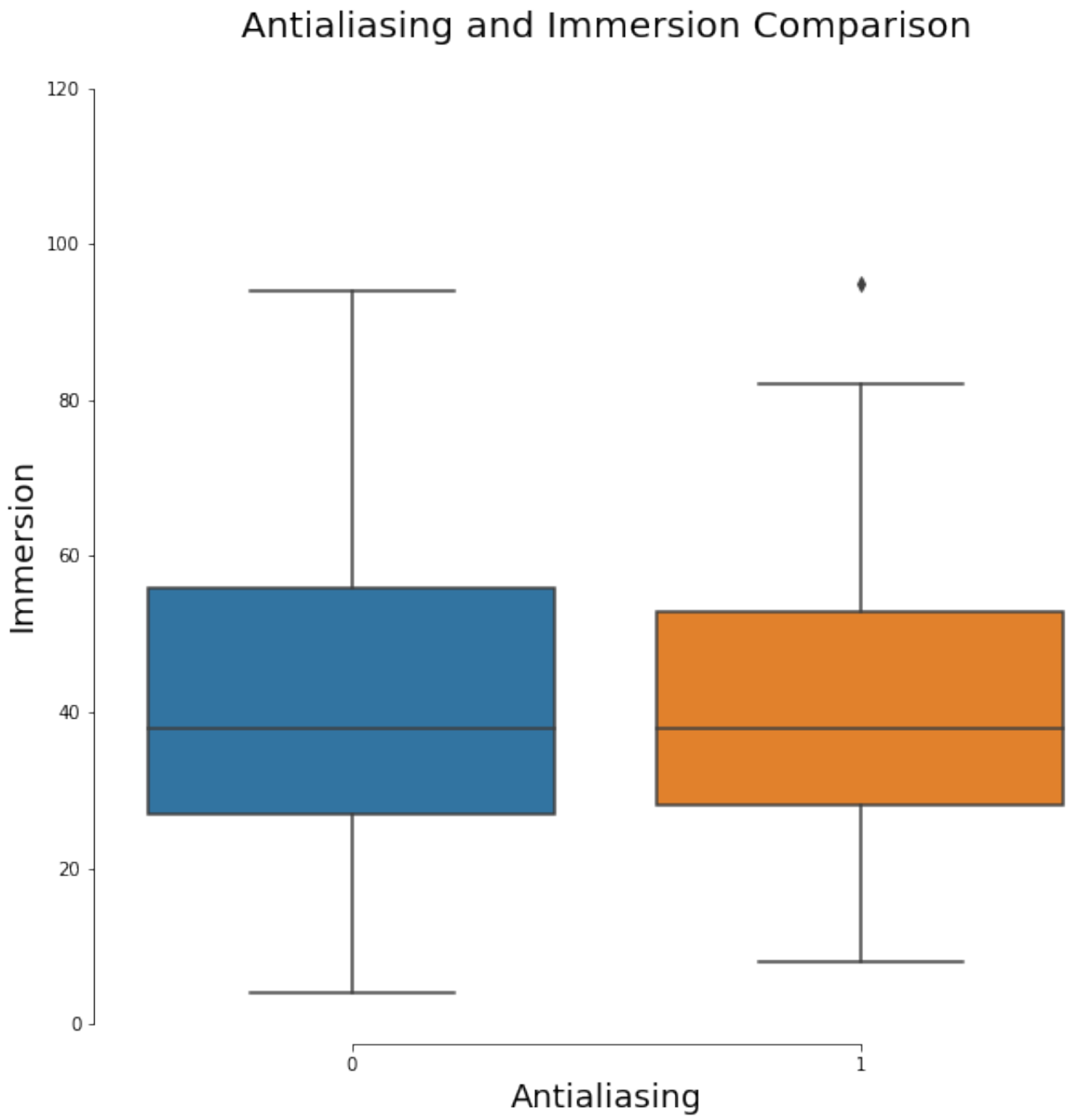

```
# Compare several distributions by using boxplots

fig, axs = plt.subplots(figsize=(10, 10))
```

```
4
5 # Boxplot
6 sns.boxplot(ax = axs, x='stereopsis', y='totalImmersion', data=vr)
7
8 # Define x-axis limits and label
9 axs.set_ylim(-2, 125)
10 axs.set_xlabel('Stereopsis', fontsize=18)
11 axs.set_ylabel('Immersion', fontsize=18)
12
13 # Title each plot
14 axs.set_title('Stereopsis and Immersion Comparison', ha='center', fontsize=20)
15
16 # Clean up each plot
17 sns.despine(ax=axs, offset=2, trim=True)
```



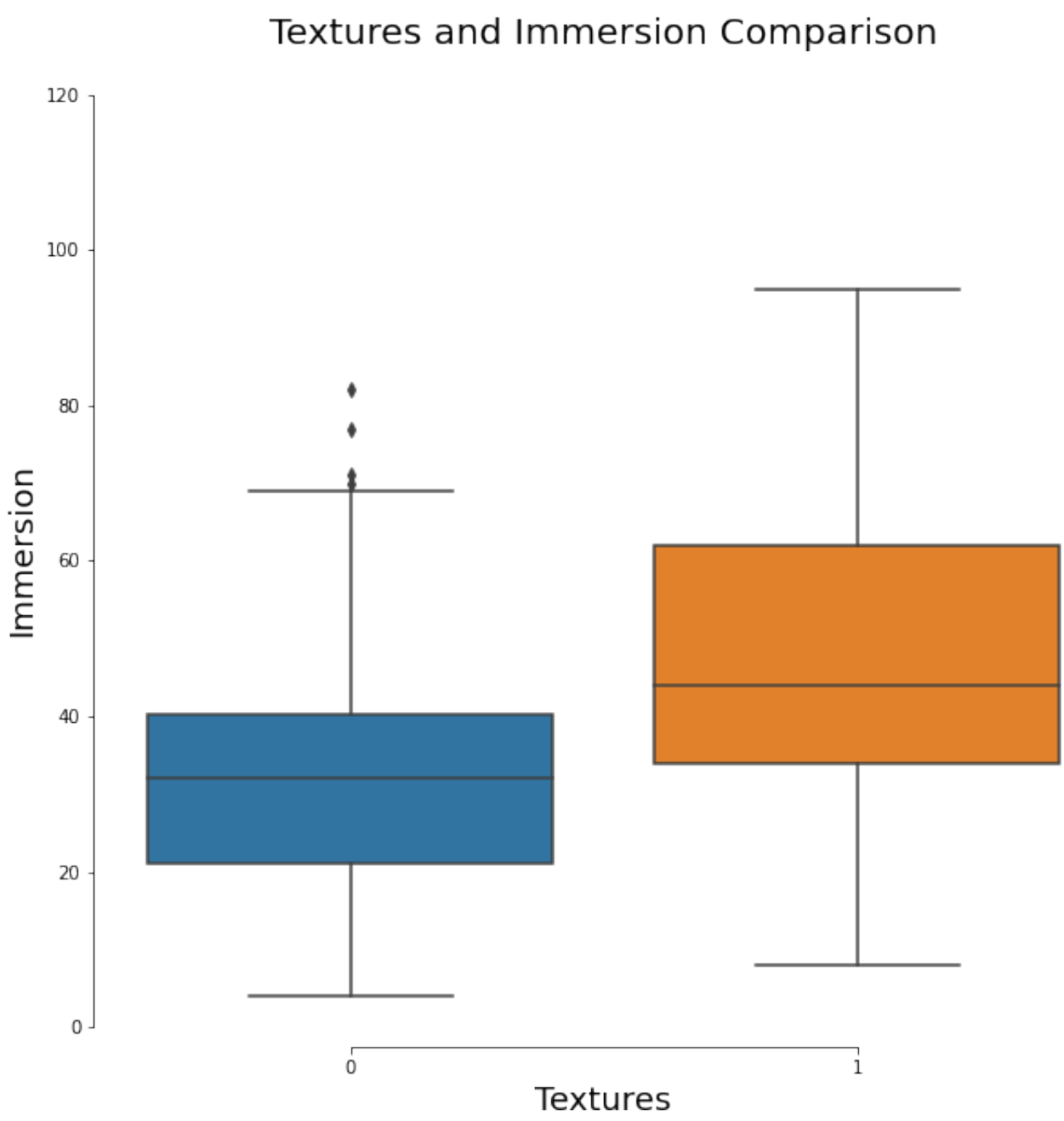

```
# Compare several distributions by using boxplots

fig, axs = plt.subplots(figsize=(10, 10))

# Boxplot
sns.boxplot(ax = axs, x='antialiasing', y='totalImmersion', data=vr)

# Define x-axis limits and label
axs.set_ylim(-2, 125)
axs.set_xlabel('Antialiasing', fontsize=18)
```

```
axs.set_ylabel('Immersion', fontsize=18)

# Title each plot
axs.set_title('Antialiasing and Immersion Comparison', ha='center', fontsize=2

# Clean up each plot
sns.despine(ax=axs, offset=2, trim=True)
```



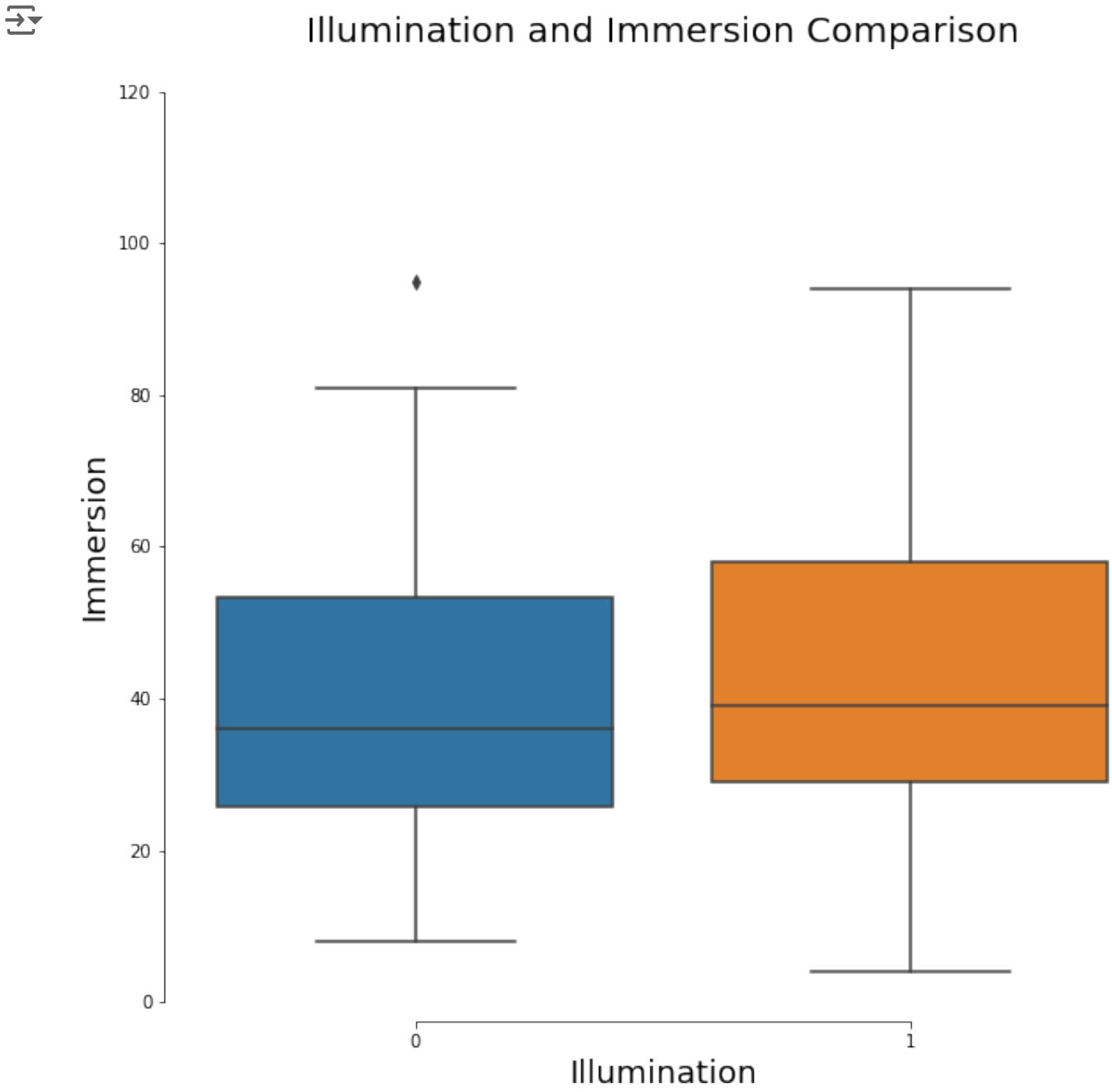

```
# Compare several distributions by using boxplots

```

```
fig, axs = plt.subplots(figsize=(10, 10))

# Boxplot
sns.boxplot(ax = axs, x='textures', y='totalImmersion', data=vr)

# Define x-axis limits and label
axs.set_ylim(-2, 125)
axs.set_xlabel('Textures', fontsize=18)
axs.set_ylabel('Immersion', fontsize=18)

# Title each plot
axs.set_title('Textures and Immersion Comparison', ha='center', fontsize=20)

# Clean up each plot
sns.despine(ax=axs, offset=2, trim=True)
```

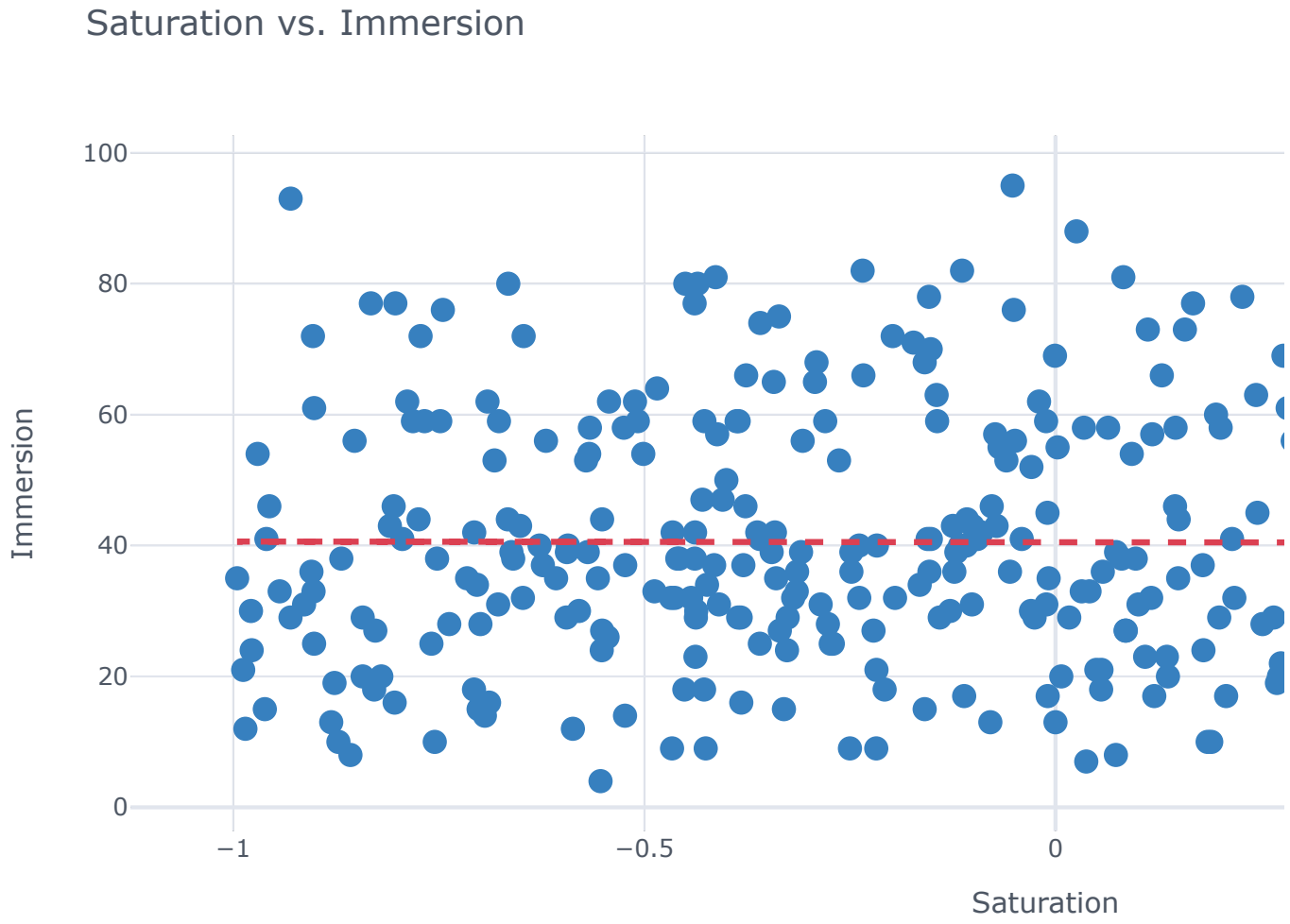



```
# Compare several distributions by using boxplots

fig, axs = plt.subplots(figsize=(10, 10))

# Boxplot
sns.boxplot(ax = axs, x='lightMode', y='totalImmersion', data=vr)

# Define x-axis limits and label
axs.set_ylim(-2, 125)
axs.set_xlabel('Illumination', fontsize=18)
```

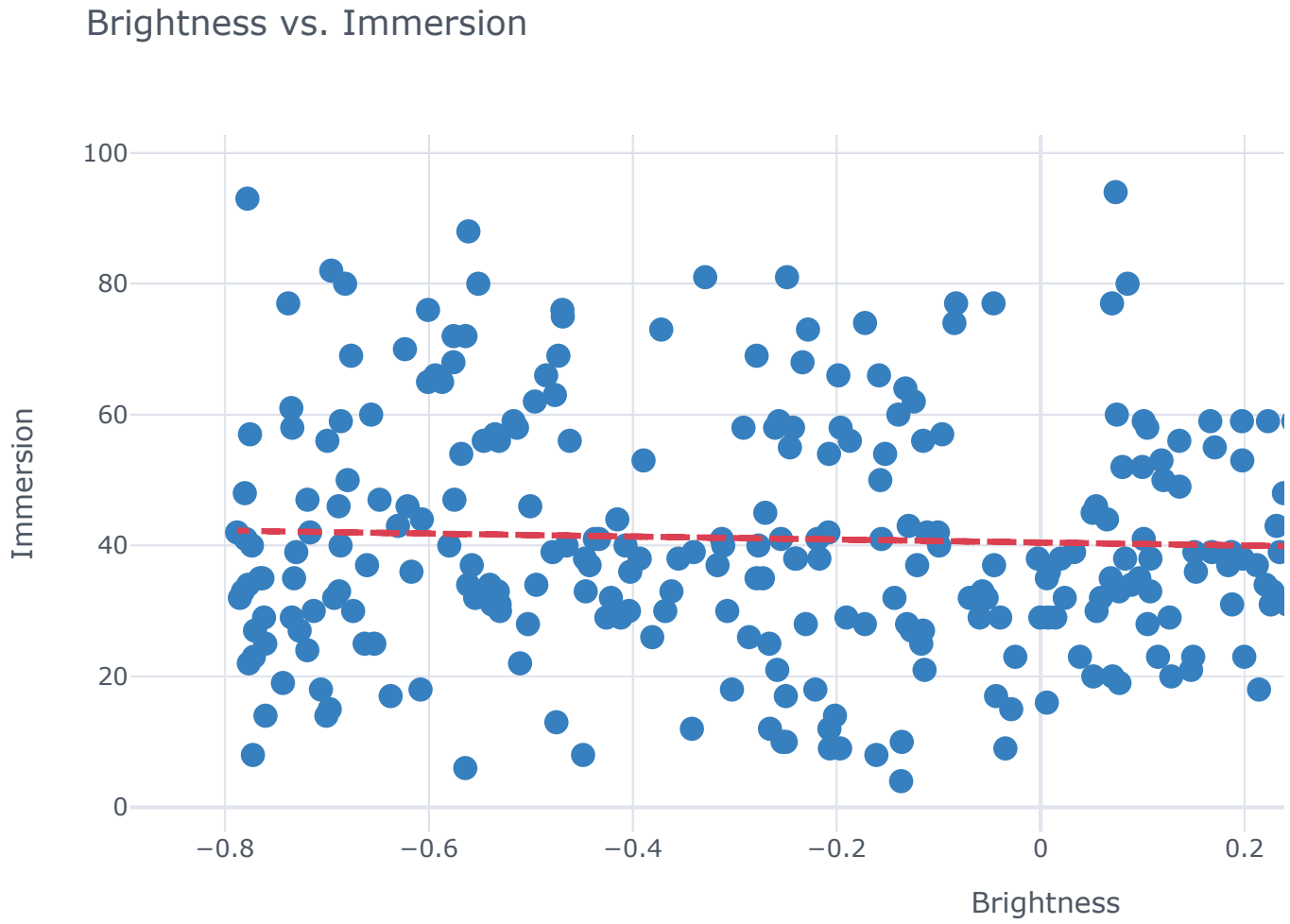
```
axs.set_ylabel('Immersion', fontsize=18)

# Title each plot
axs.set_title('Illumination and Immersion Comparison', ha='center', fontsize=2

# Clean up each plot
sns.despine(ax=axs, offset=2, trim=True)
```



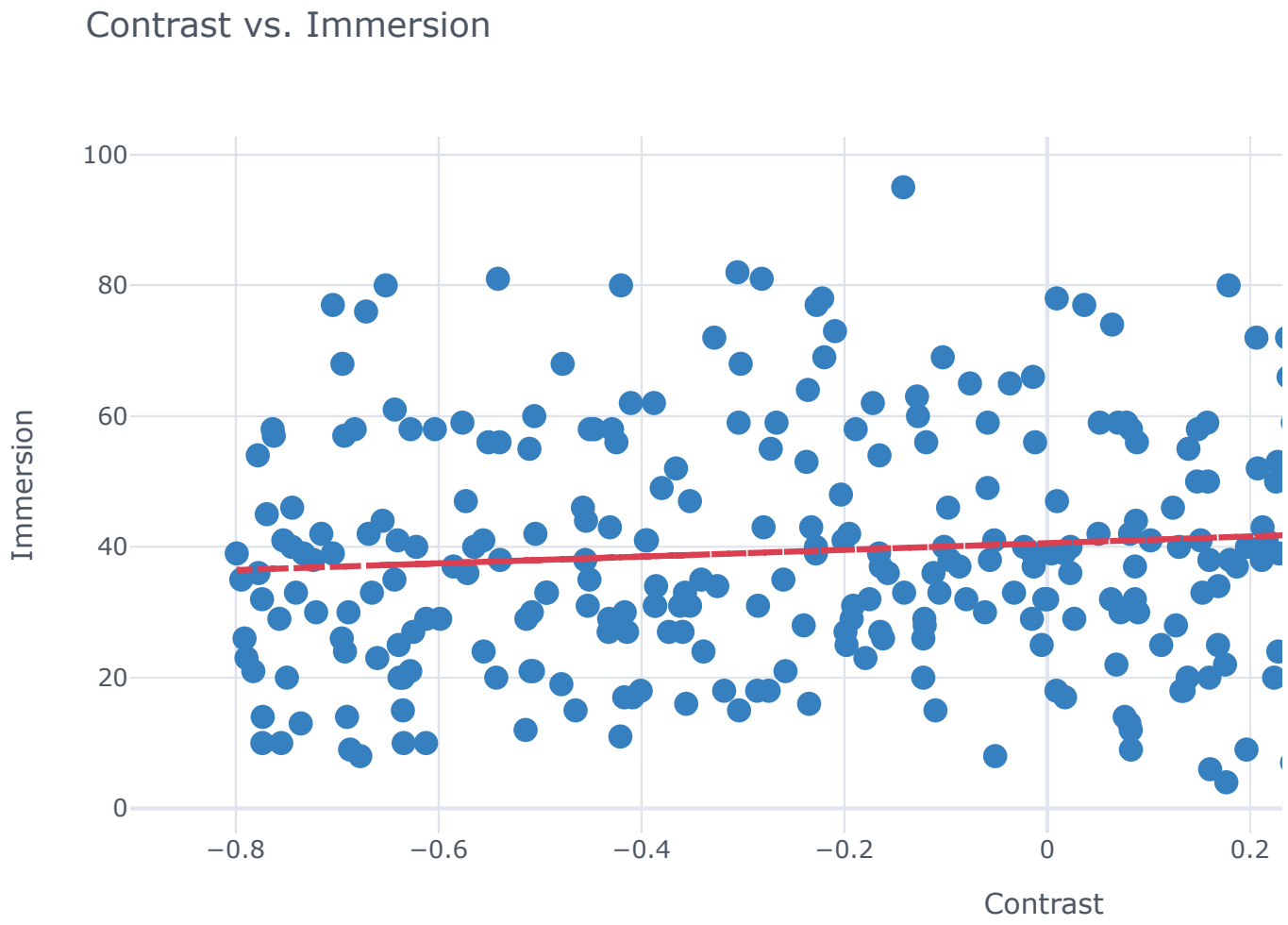
```
enable_plotly_in_cell()
vr.iplot(kind='scatter',
           x='saturation',
           y='totalImmersion',
           bestfit=True,
           bestfit_colors=['red'],
           mode='markers',
           color=['blue'],
           title='Saturation vs. Immersion',
           xTitle='Saturation',
           yTitle='Immersion')
```



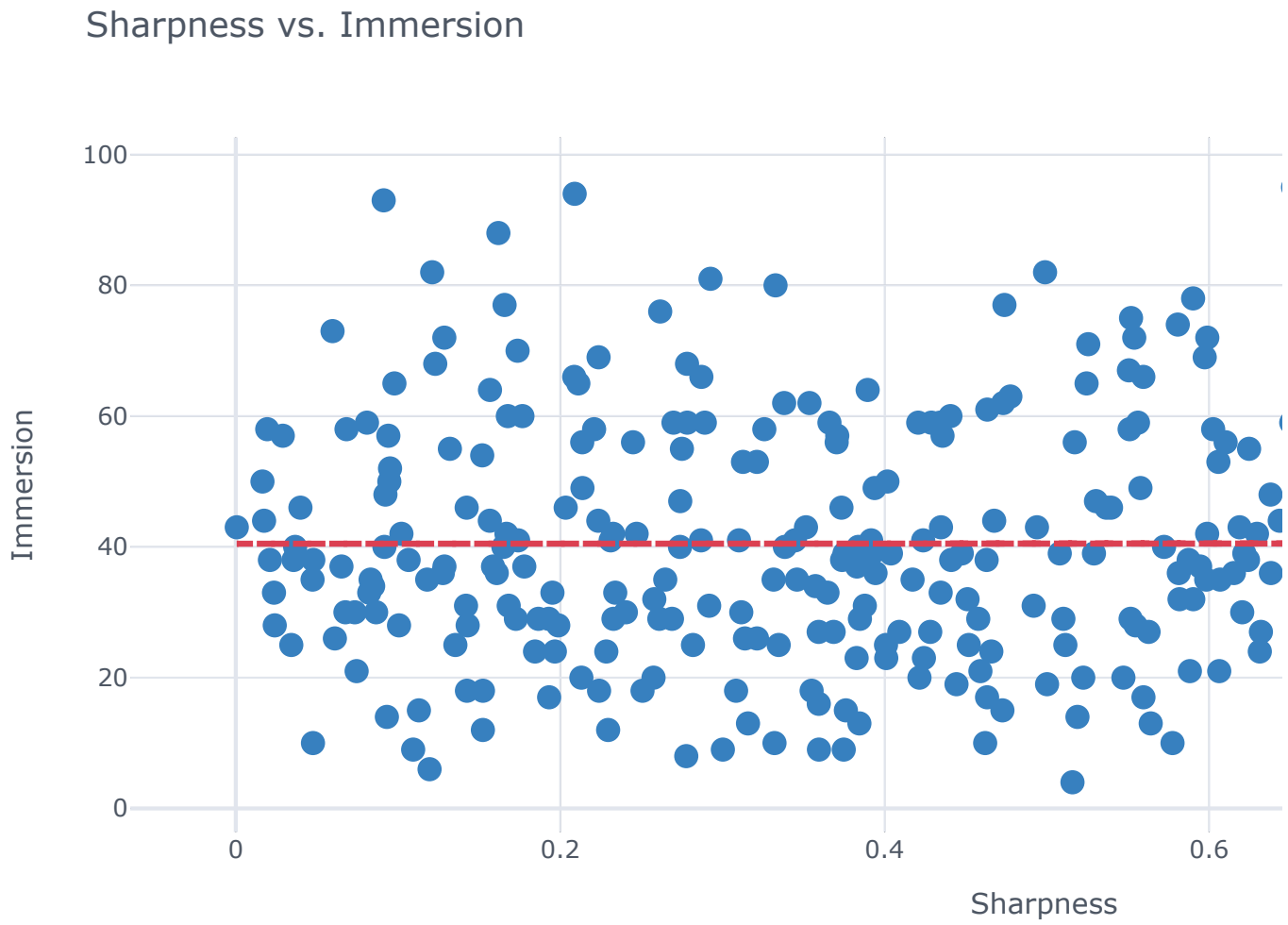
```
enable_plotly_in_cell()
vr.iplot(kind='scatter',
           x='brightness',
           y='totalImmersion',
           bestfit=True,
           bestfit_colors=['red'],
           mode='markers',
           color=['blue'],
           title='Brightness vs. Immersion',
           xTitle='Brightness',
           yTitle='Immersion')
```



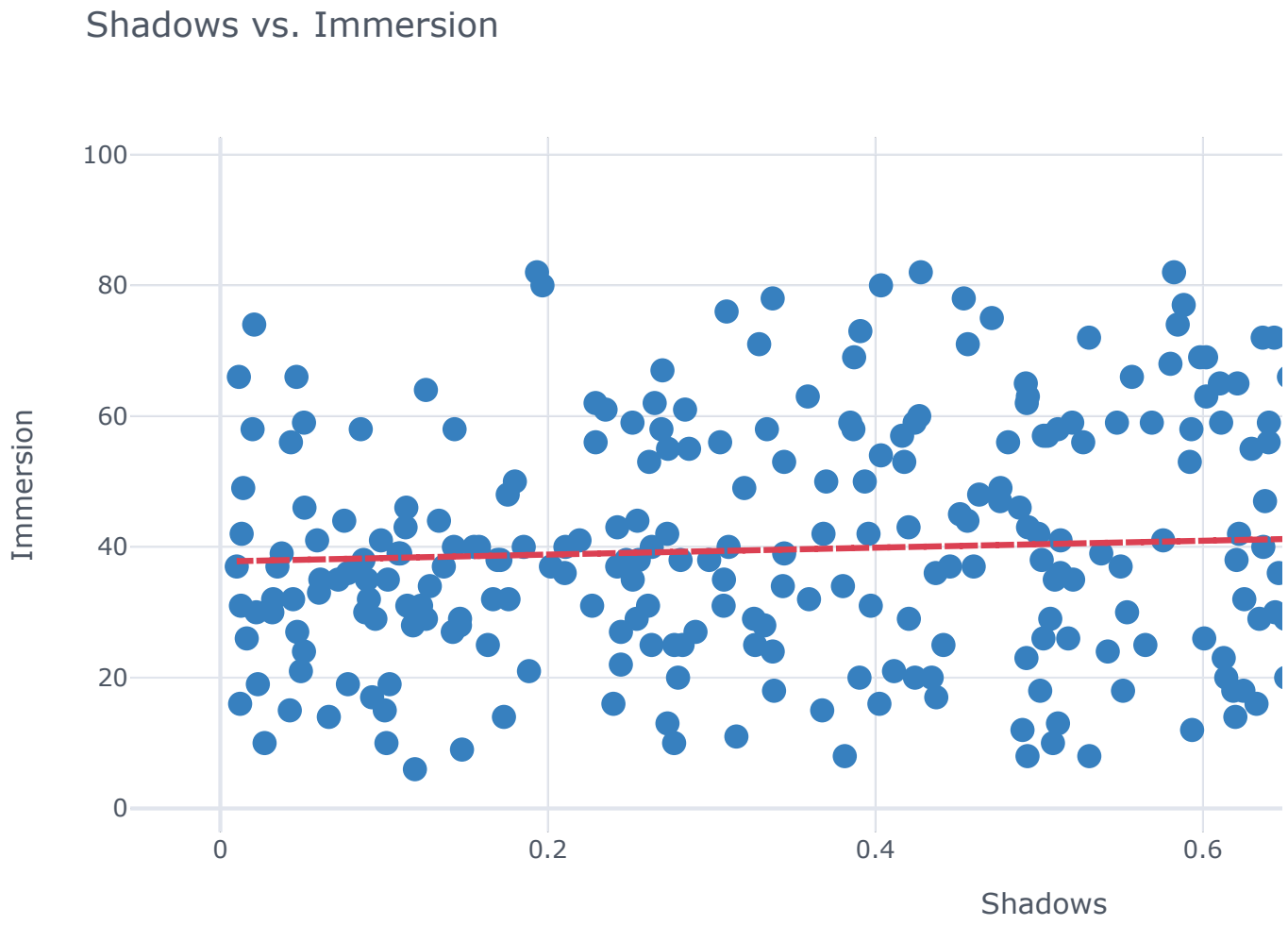
```
enable_plotly_in_cell()
vr.iplot(kind='scatter',
           x='contrast',
           y='totalImmersion',
           bestfit=True,
           bestfit_colors=['red'],
           mode='markers',
           color=['blue'],
           title='Contrast vs. Immersion',
           xTitle='Contrast',
           yTitle='Immersion')
```



```
enable_plotly_in_cell()
vr.iplot(kind='scatter',
           x='sharpness',
           y='totalImmersion',
           bestfit=True,
           bestfit_colors=['red'],
           mode='markers',
           color=['blue'],
           title='Sharpness vs. Immersion',
           xTitle='Sharpness',
           yTitle='Immersion')
```

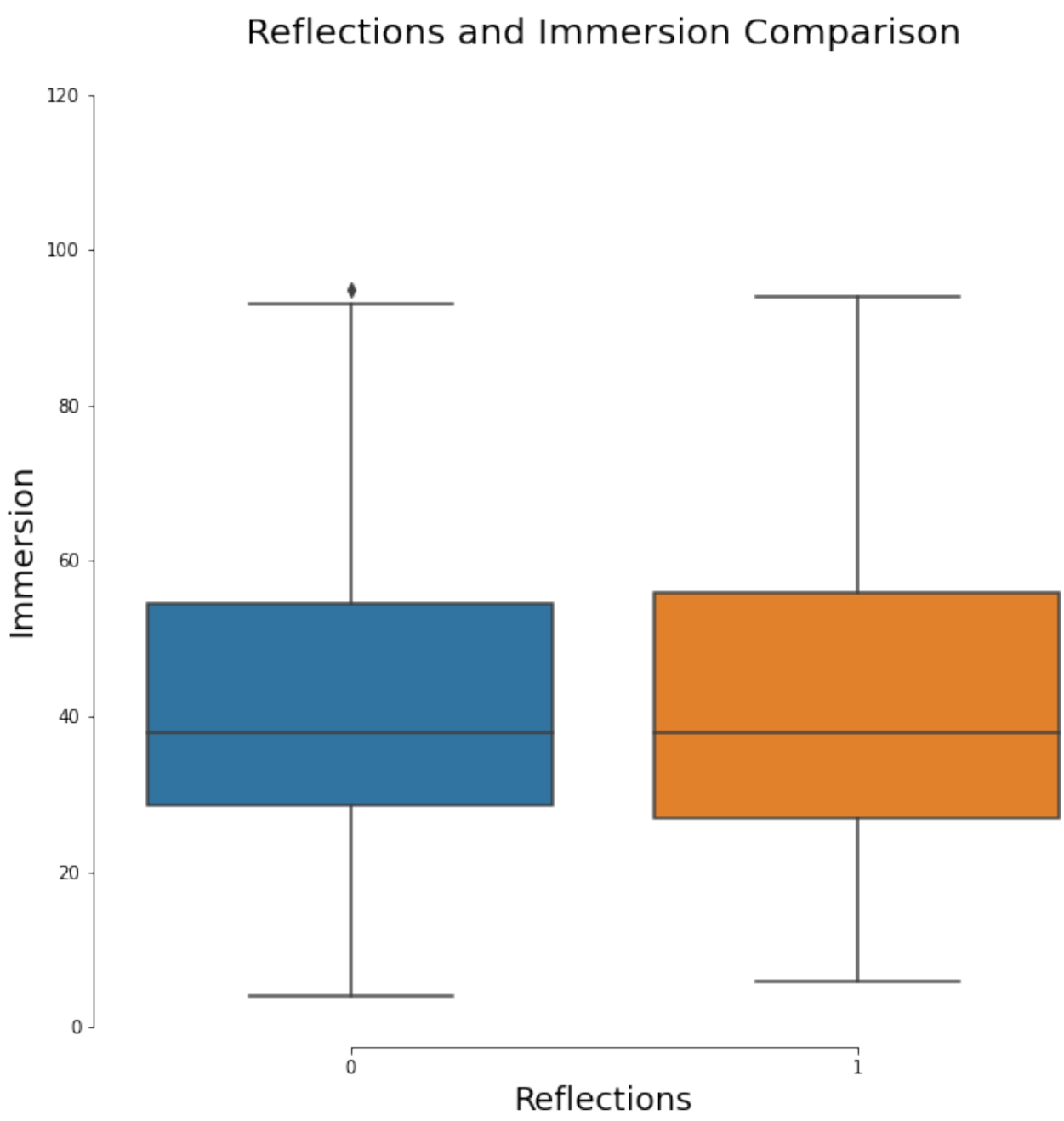


```
enable_plotly_in_cell()
vr.iplot(kind='scatter',
           x='shadowStrength',
           y='totalImmersion',
           bestfit=True,
           bestfit_colors=['red'],
           mode='markers',
           color=['blue'],
           title='Shadows vs. Immersion',
           xTitle='Shadows',
           yTitle='Immersion')
```



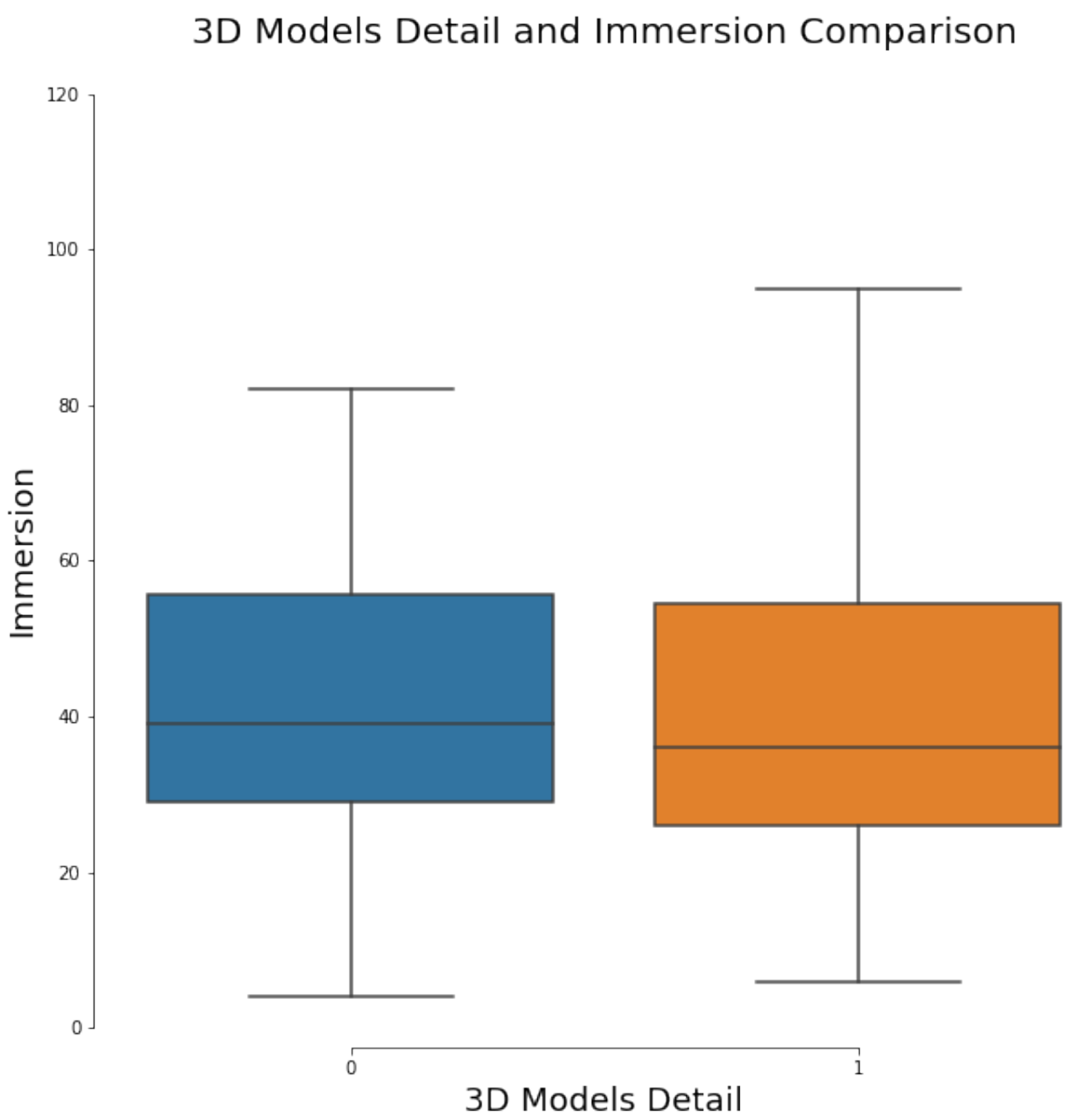

```
# Compare several distributions by using boxplots

fig, axs = plt.subplots(figsize=(10, 10))
```

```
4
5 # Boxplot
6 sns.boxplot(ax = axs, x='reflections', y='totalImmersion', data=vr)
7
8 # Define x-axis limits and label
9 axs.set_ylim(-2, 125)
10 axs.set_xlabel('Reflections', fontsize=18)
11 axs.set_ylabel('Immersion', fontsize=18)
12
13 # Title each plot
14 axs.set_title('Reflections and Immersion Comparison', ha='center', fontsize=20
15
16 # Clean up each plot
17 sns.despine(ax=axs, offset=2, trim=True)
```



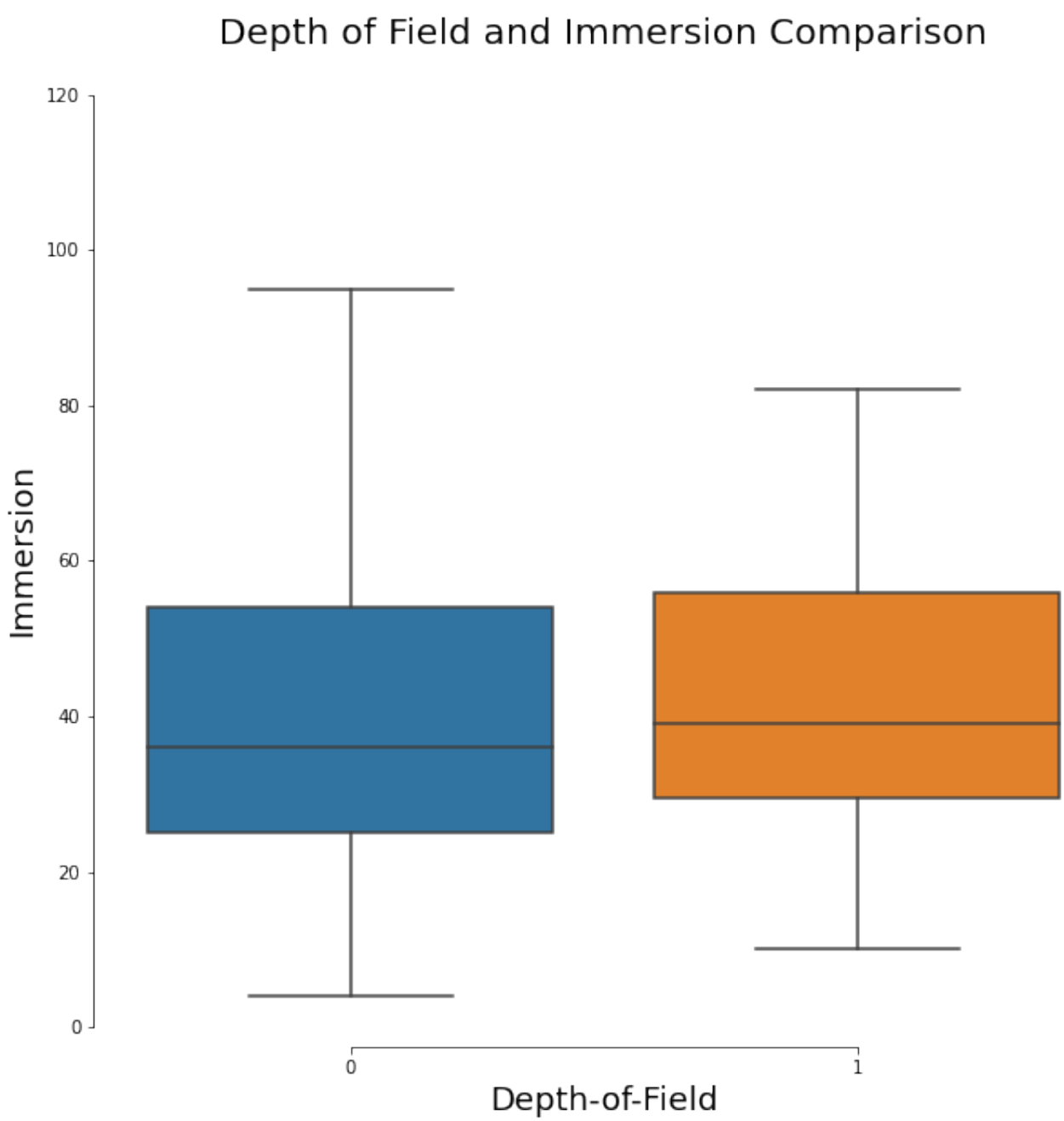

```
# Compare several distributions by using boxplots

fig, axs = plt.subplots(figsize=(10, 10))

# Boxplot
sns.boxplot(ax = axs, x='modelsDetail', y='totalImmersion', data=vr)

# Define x-axis limits and label
axs.set_ylim(-2, 125)
axs.set_xlabel('3D Models Detail', fontsize=18)
```

```
axs.set_ylabel('Immersion', fontsize=18)

# Title each plot
axs.set_title('3D Models Detail and Immersion Comparison', ha='center', fontsi

# Clean up each plot
sns.despine(ax=axs, offset=2, trim=True)
```



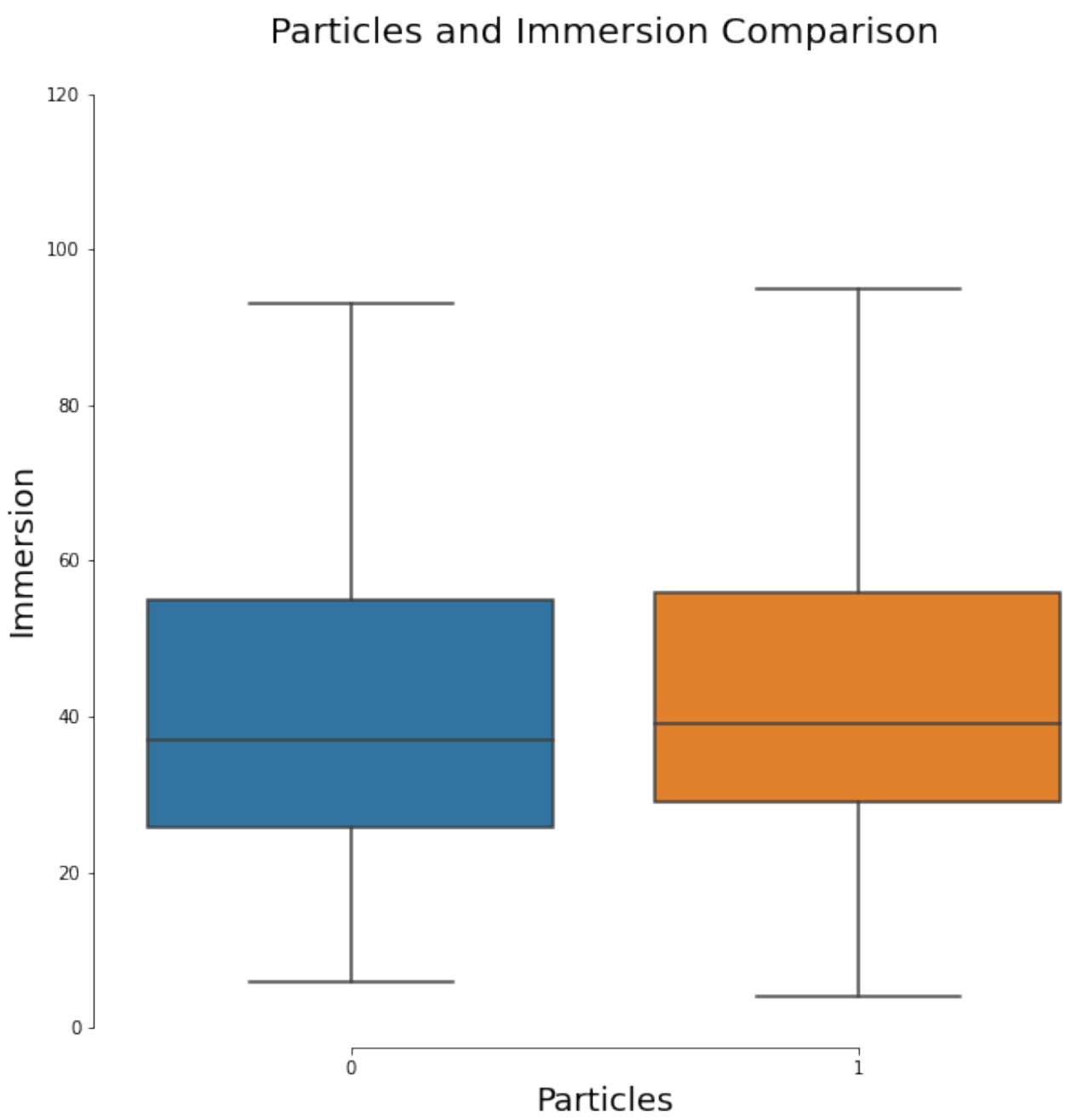

```
# Compare several distributions by using boxplots

```

```
fig, axs = plt.subplots(figsize=(10, 10))

# Boxplot
sns.boxplot(ax = axs, x='dof', y='totalImmersion', data=vr)

# Define x-axis limits and label
axs.set_ylim(-2, 125)
axs.set_xlabel('Depth-of-Field', fontsize=18)
axs.set_ylabel('Immersion', fontsize=18)

# Title each plot
axs.set_title('Depth of Field and Immersion Comparison', ha='center', fontsize
# Clean up each plot
sns.despine(ax=axs, offset=2, trim=True)
```



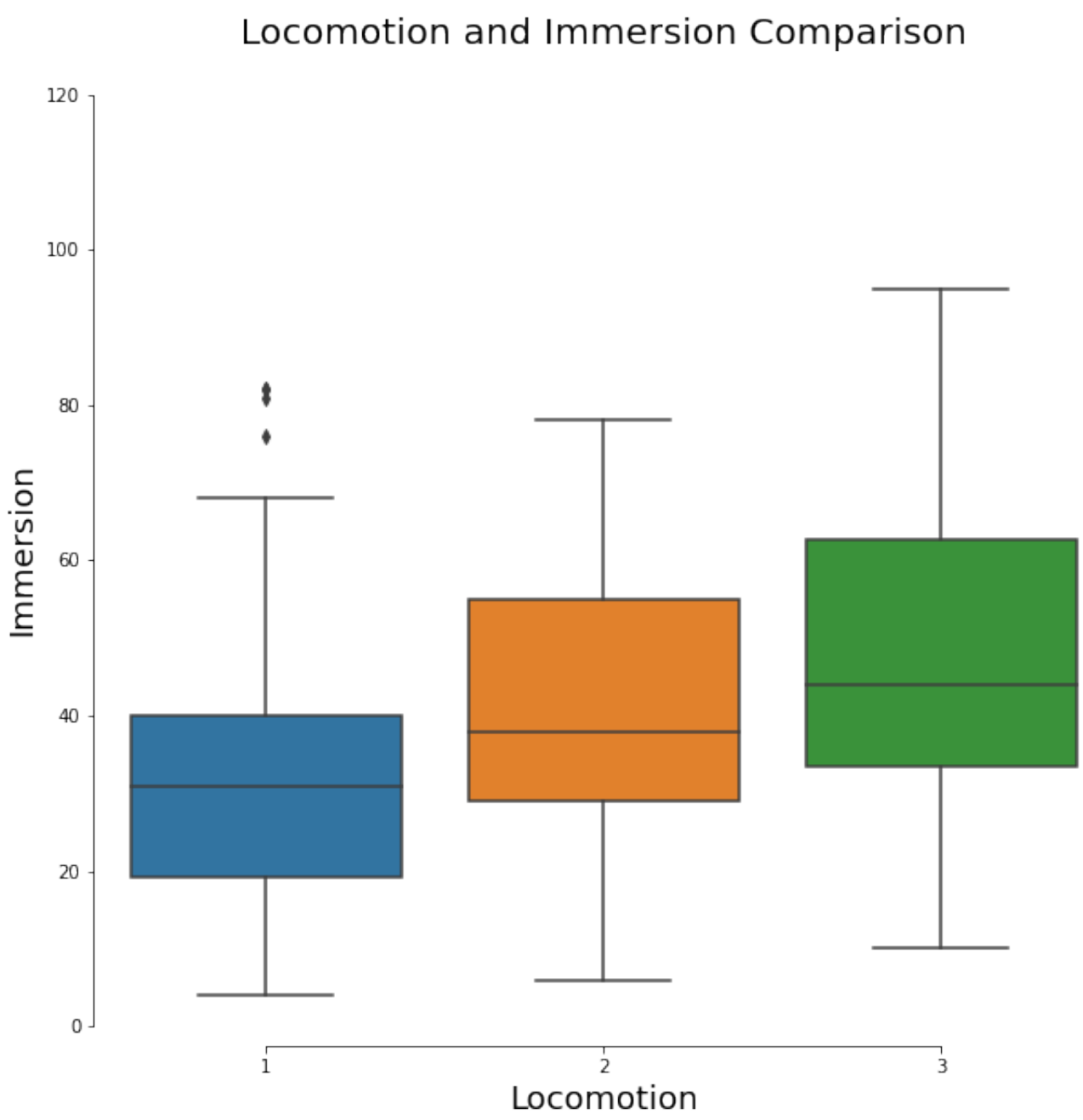

```
# Compare several distributions by using boxplots

fig, axs = plt.subplots(figsize=(10, 10))

# Boxplot
sns.boxplot(ax = axs, x='particles', y='totalImmersion', data=vr)

# Define x-axis limits and label
axs.set_ylim(-2, 125)
axs.set_xlabel('Particles', fontsize=18)
```

```
axs.set_ylabel('Immersion', fontsize=18)

# Title each plot
axs.set_title('Particles and Immersion Comparison', ha='center', fontsize=20)

# Clean up each plot
sns.despine(ax=axs, offset=2, trim=True)
```



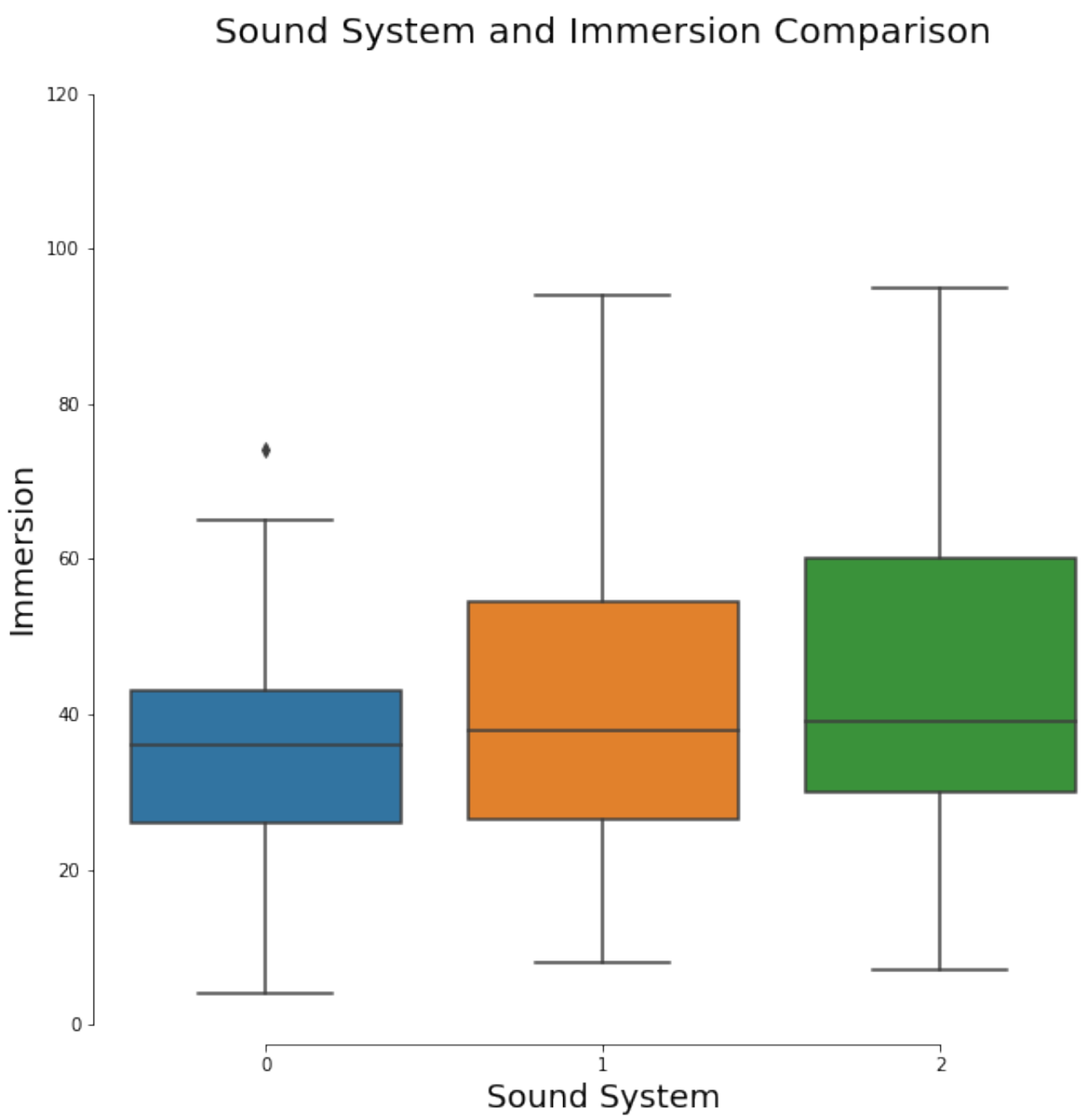

```
# Compare several distributions by using boxplots

```

```
fig, axs = plt.subplots(figsize=(10, 10))

# Boxplot
sns.boxplot(ax = axs, x='locomotion', y='totalImmersion', data=vr)

# Define x-axis limits and label
axs.set_ylim(-2, 125)
axs.set_xlabel('Locomotion', fontsize=18)
axs.set_ylabel('Immersion', fontsize=18)

# Title each plot
axs.set_title('Locomotion and Immersion Comparison', ha='center', fontsize=20)

# Clean up each plot
sns.despine(ax=axs, offset=2, trim=True)
```



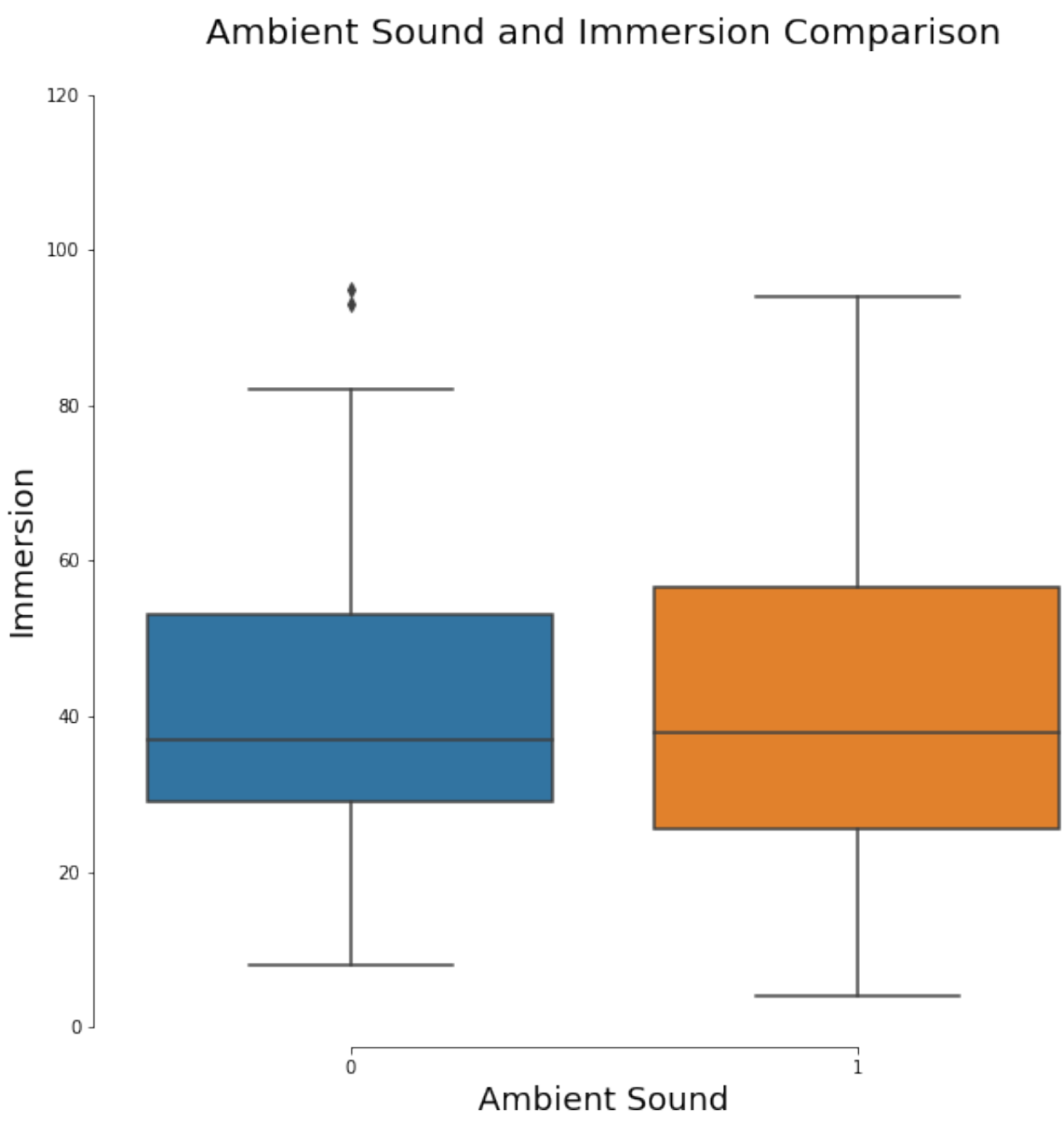

```
# Compare several distributions by using boxplots

fig, axs = plt.subplots(figsize=(10, 10))

# Boxplot
sns.boxplot(ax = axs, x='headphones', y='totalImmersion', data=vr)

# Define x-axis limits and label
axs.set_ylim(-2, 125)
axs.set_xlabel('Sound System', fontsize=18)
```

```
axs.set_ylabel('Immersion', fontsize=18)

# Title each plot
axs.set_title('Sound System and Immersion Comparison', ha='center', fontsize=2

# Clean up each plot
sns.despine(ax=axs, offset=2, trim=True)
```



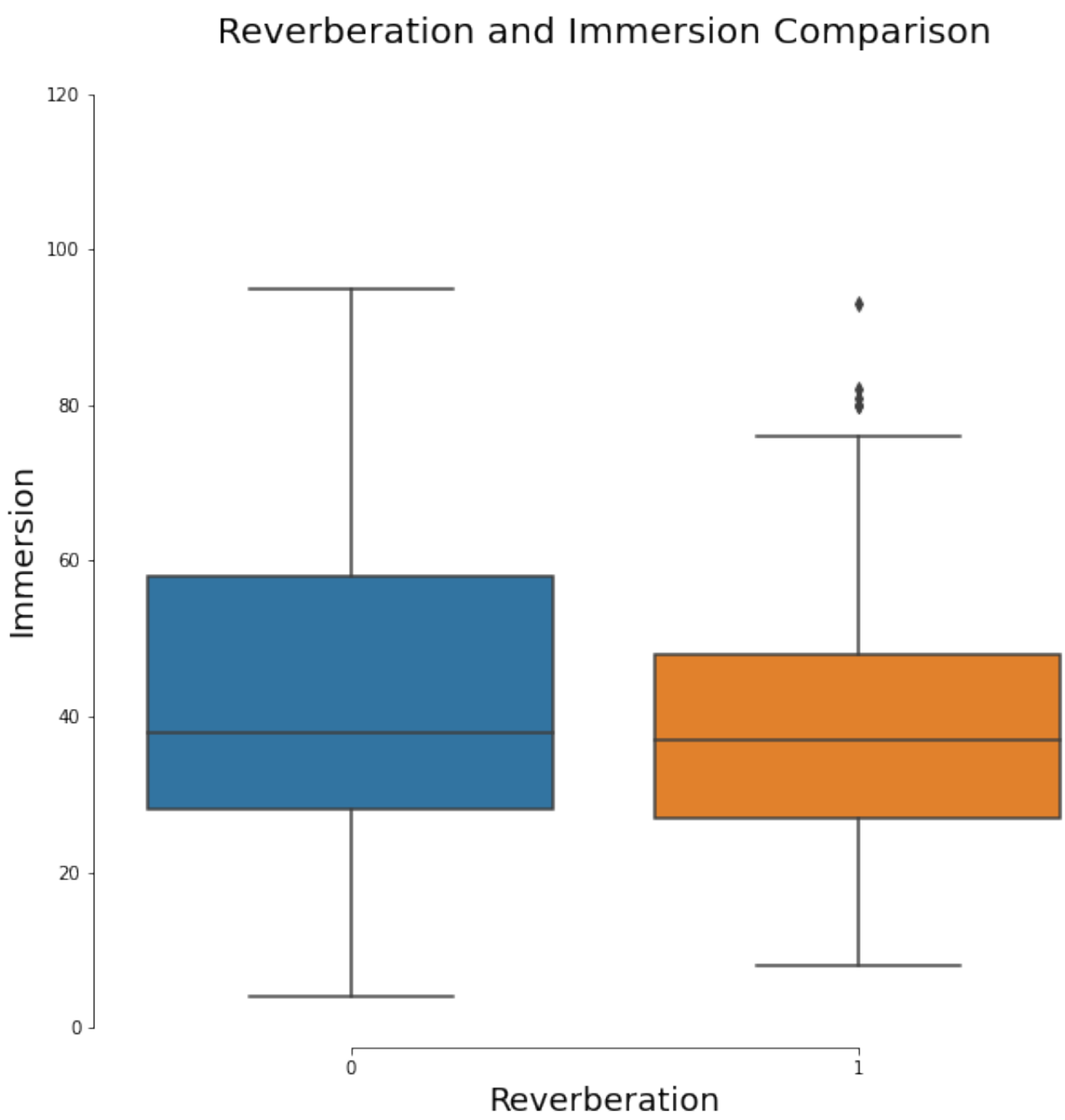

```
# Compare several distributions by using boxplots

```

```
fig, axs = plt.subplots(figsize=(10, 10))

# Boxplot
sns.boxplot(ax = axs, x='ambientSound', y='totalImmersion', data=vr)

# Define x-axis limits and label
axs.set_ylim(-2, 125)
axs.set_xlabel('Ambient Sound', fontsize=18)
axs.set_ylabel('Immersion', fontsize=18)

# Title each plot
axs.set_title('Ambient Sound and Immersion Comparison', ha='center', fontsize=

# Clean up each plot
sns.despine(ax=axs, offset=2, trim=True)
```



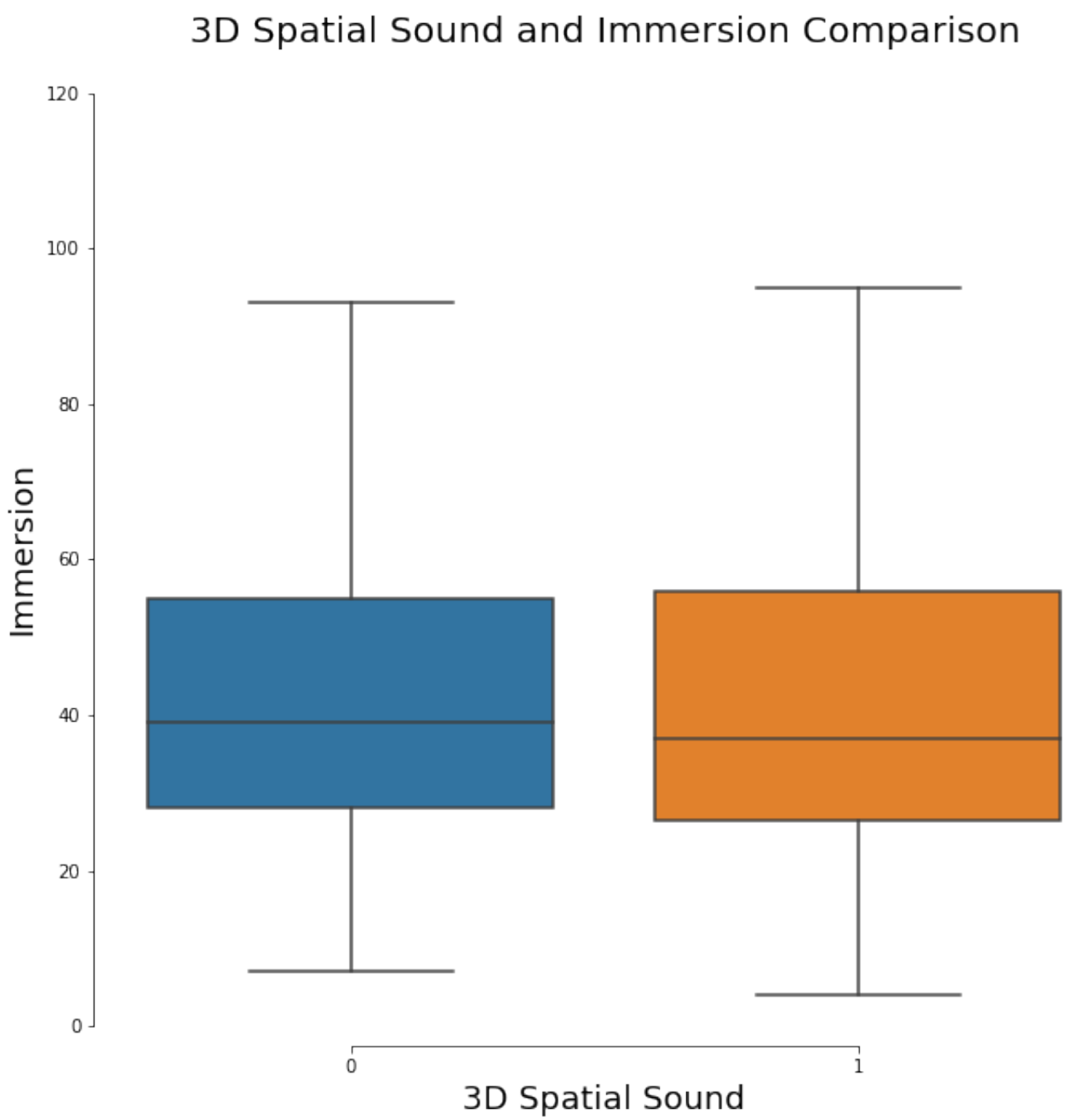

```
# Compare several distributions by using boxplots

fig, axs = plt.subplots(figsize=(10, 10))

# Boxplot
sns.boxplot(ax = axs, x='reverbZone', y='totalImmersion', data=vr)

# Define x-axis limits and label
axs.set_ylim(-2, 125)
axs.set_xlabel('Reverberation', fontsize=18)
```

```
axs.set_ylabel('Immersion', fontsize=18)

# Title each plot
axs.set_title('Reverberation and Immersion Comparison', ha='center', fontsize=

# Clean up each plot
sns.despine(ax=axs, offset=2, trim=True)
```




```
# Compare several distributions by using boxplots

```

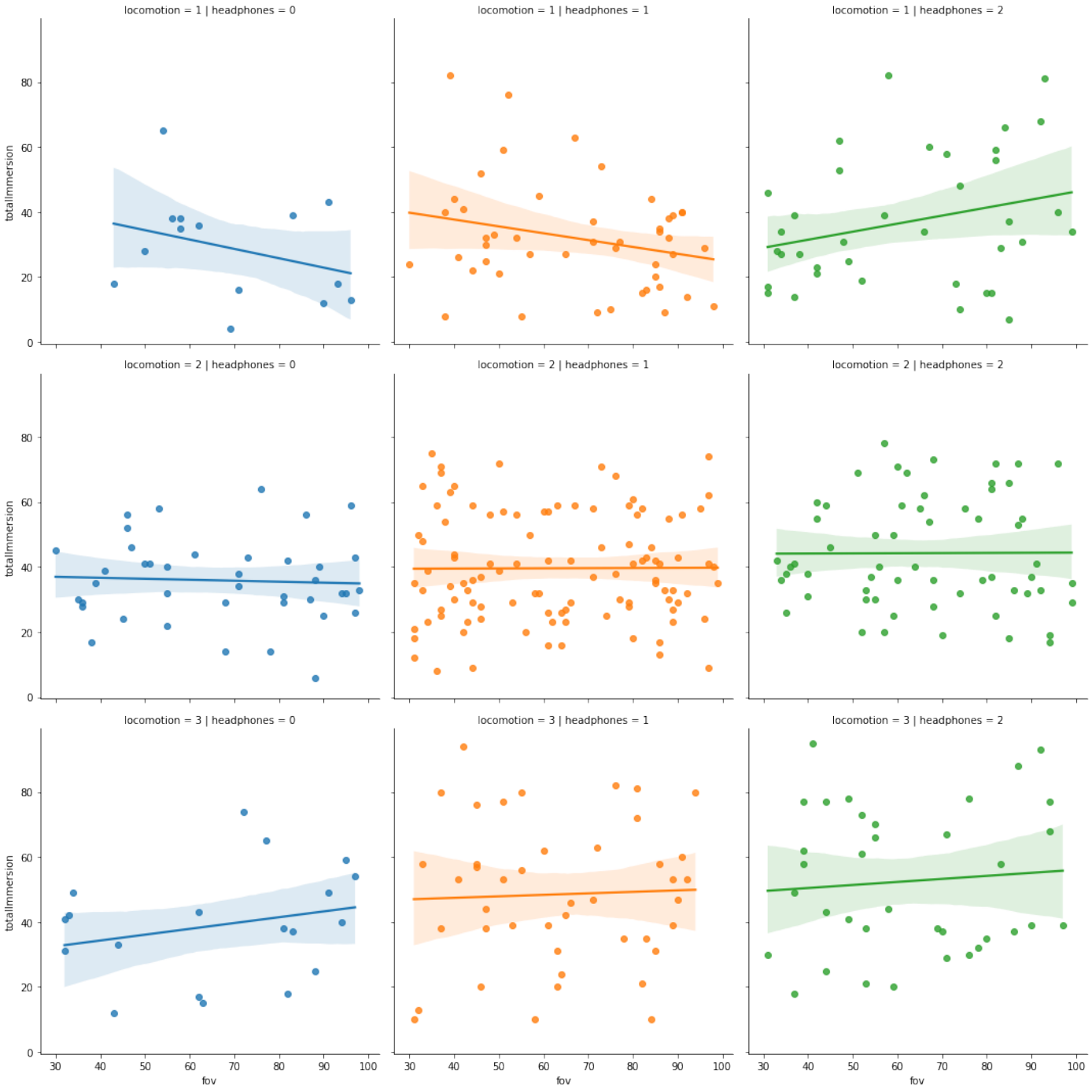
```
fig, axs = plt.subplots(figsize=(10, 10))

# Boxplot
sns.boxplot(ax = axs, x='spatialSound', y='totalImmersion', data=vr)

# Define x-axis limits and label
axs.set_ylim(-2, 125)
axs.set_xlabel('3D Spatial Sound', fontsize=18)
axs.set_ylabel('Immersion', fontsize=18)

# Title each plot
axs.set_title('3D Spatial Sound and Immersion Comparison', ha='center', fontsi

# Clean up each plot
sns.despine(ax=axs, offset=2, trim=True)
```




## Locomotion and Sound System

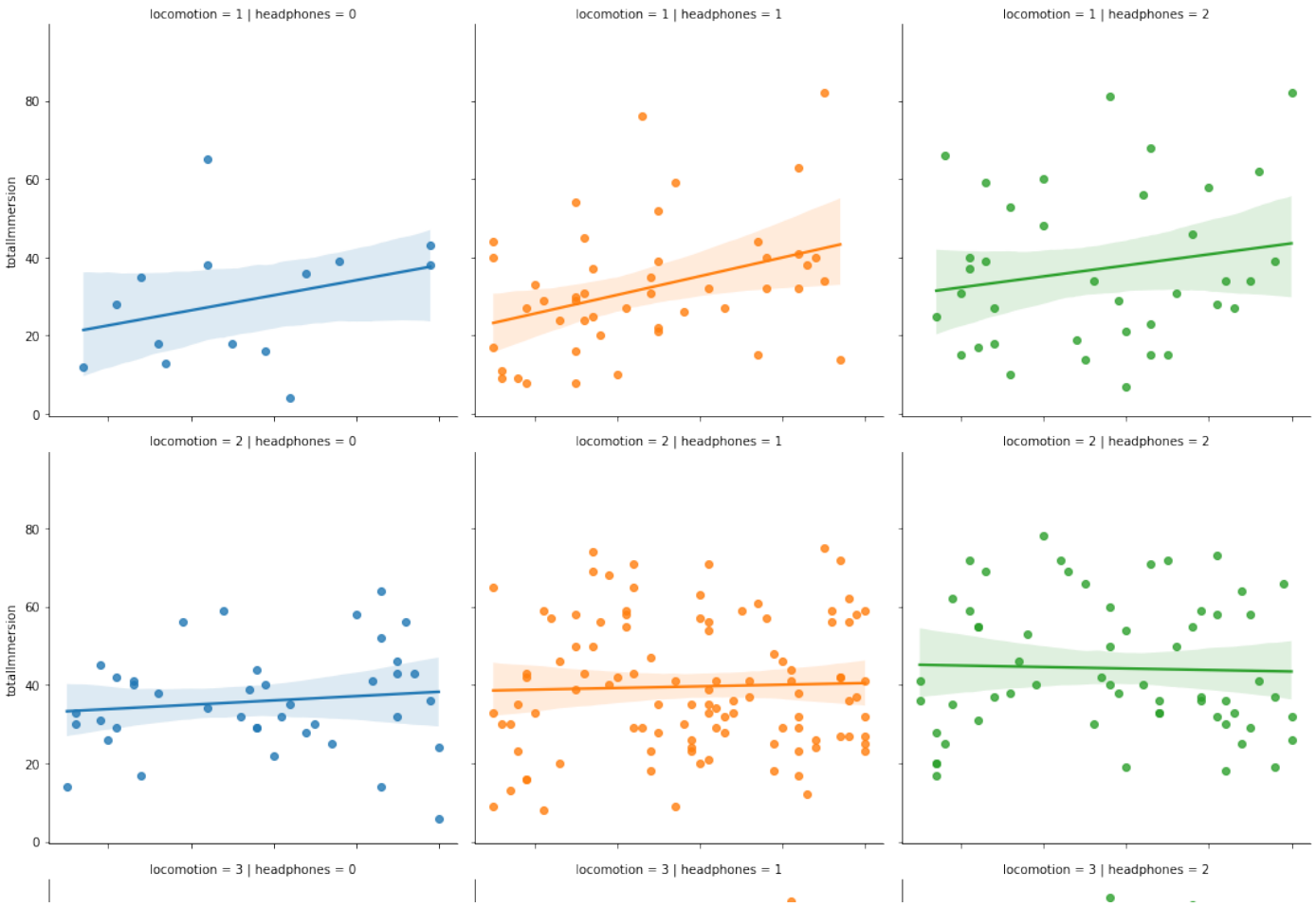
```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='headphones', row = 'locomotion', size = 5, hue=
facet.map(sns.regplot, 'fov', 'totalImmersion');
```

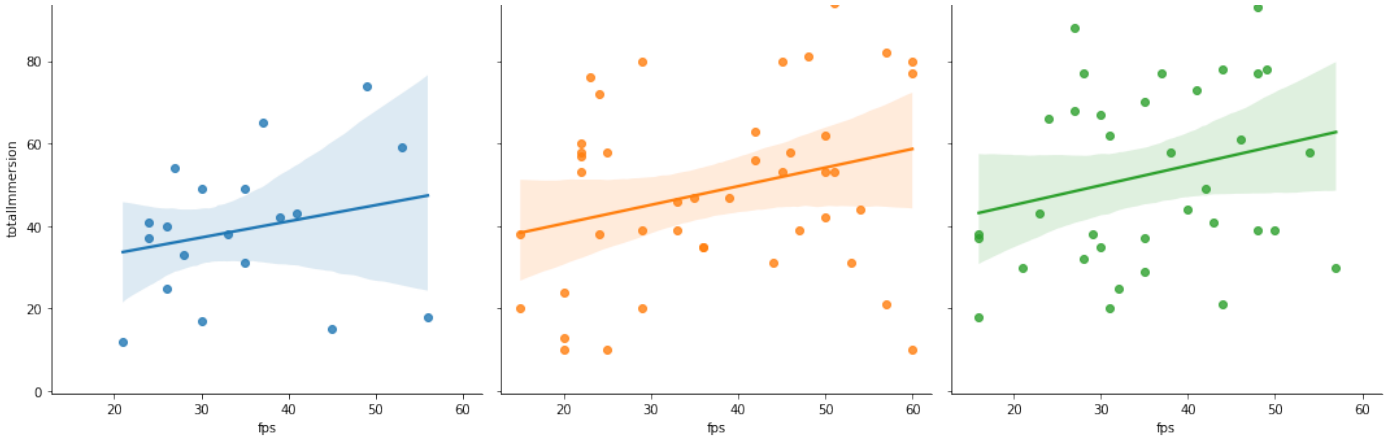
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:

The `size` parameter has been renamed to `height`; please update your code.
```


locomotion = 1 | headphones = 0
locomotion = 1 | headphones = 1
locomotion = 1 | headphones = 2
locomotion = 2 | headphones = 0
locomotion = 2 | headphones = 1
locomotion = 2 | headphones = 2
locomotion = 3 | headphones = 0
locomotion = 3 | headphones = 1
locomotion = 3 | headphones = 2
totalImmersion
fov
0
20
40
60
80
30
40
50
60
70
80
90
100

```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='headphones', row = 'locomotion', size = 5, hue=
facet.map(sns.regplot, 'fps', 'totalImmersion');
```

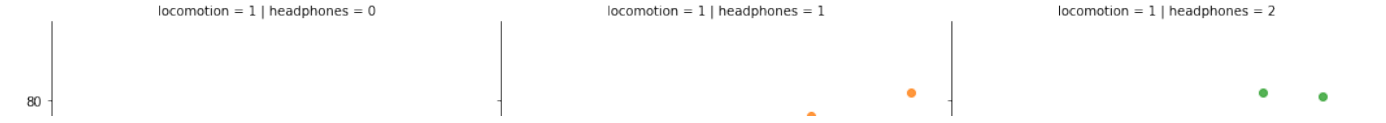
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:
The `size` parameter has been renamed to `height`; please update your code.
```

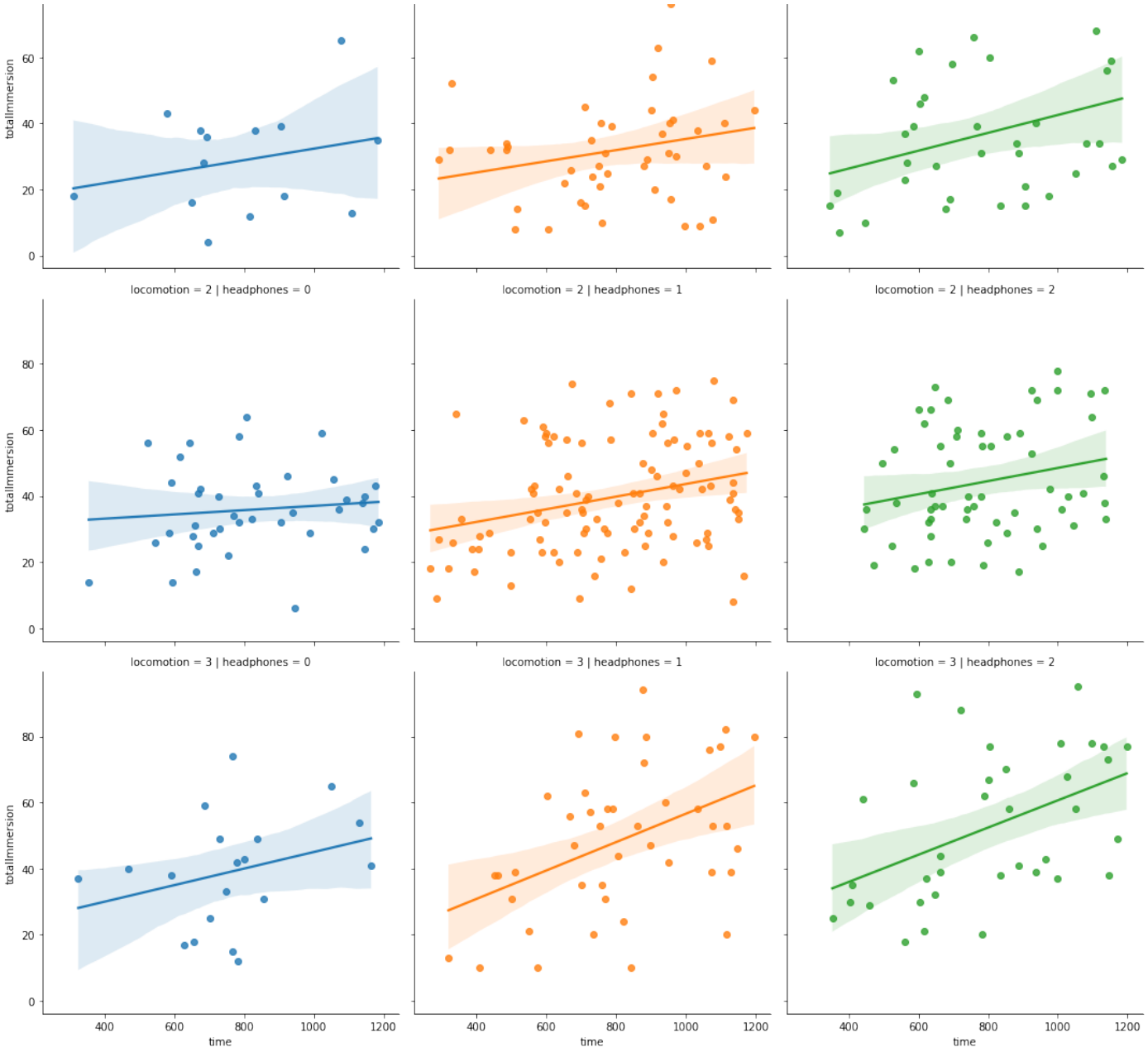





```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='headphones', row = 'locomotion', size = 5, hue=
facet.map(sns.regplot, 'time', 'totalImmersion');
```

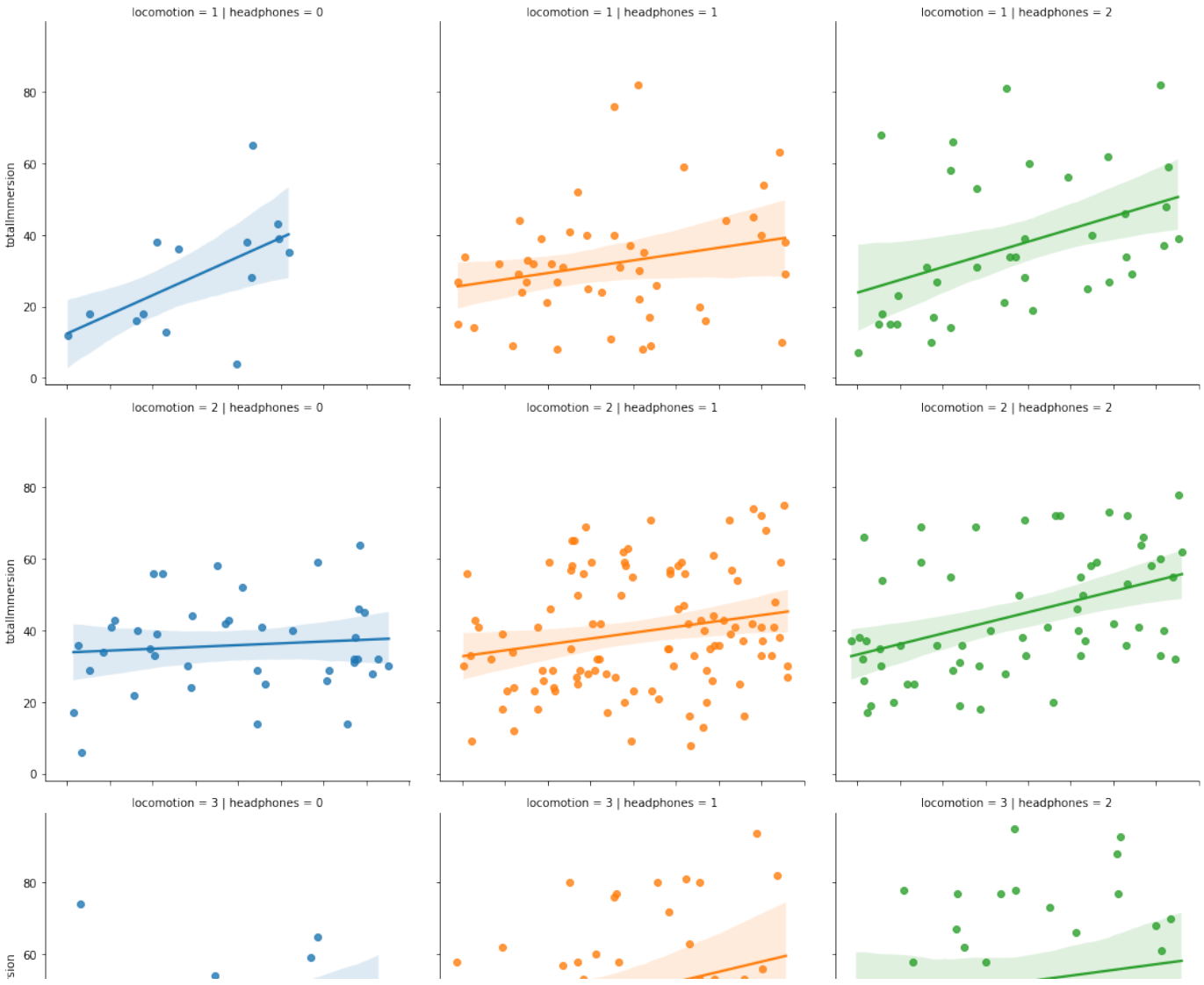
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:
The `size` parameter has been renamed to `height`; please update your code.
```

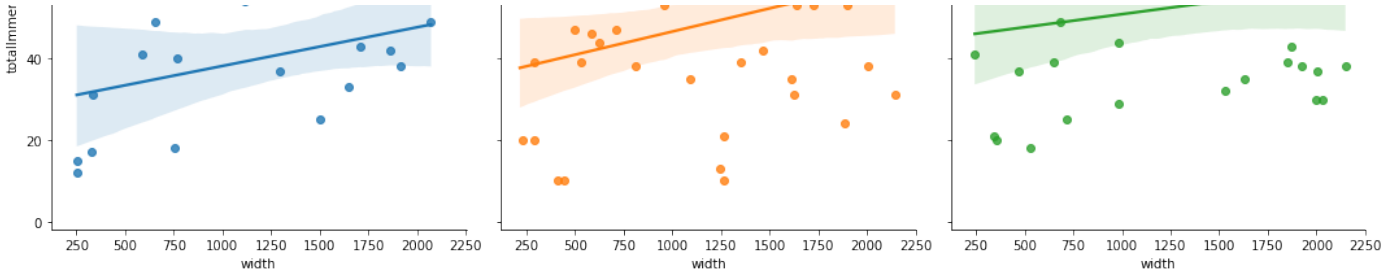



totalImmersion
locomotion = 2 | headphones = 0
locomotion = 2 | headphones = 1
locomotion = 2 | headphones = 2
locomotion = 3 | headphones = 0
locomotion = 3 | headphones = 1
locomotion = 3 | headphones = 2
time

```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='headphones', row = 'locomotion', size = 5, hue=
facet.map(sns.regplot, 'width', 'totalImmersion');
```

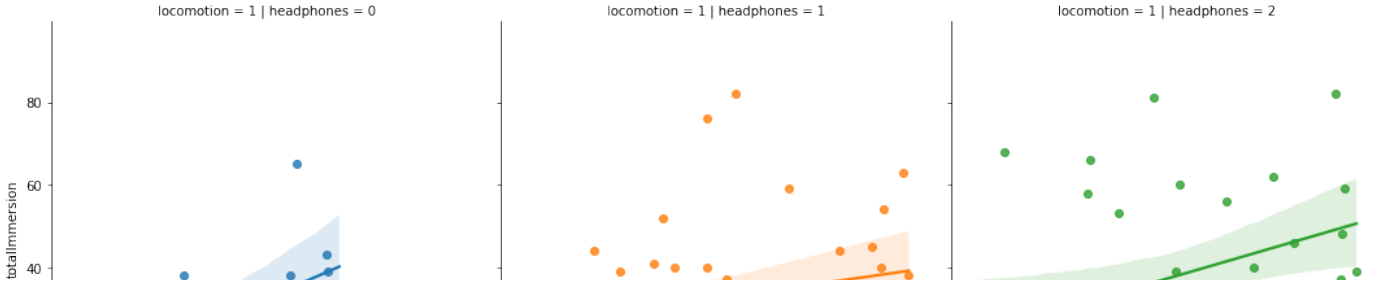
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:

The `size` parameter has been renamed to `height`; please update your code.
```






```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='headphones', row = 'locomotion', size = 5, hue=
facet.map(sns.regplot, 'height', 'totalImmersion');
```

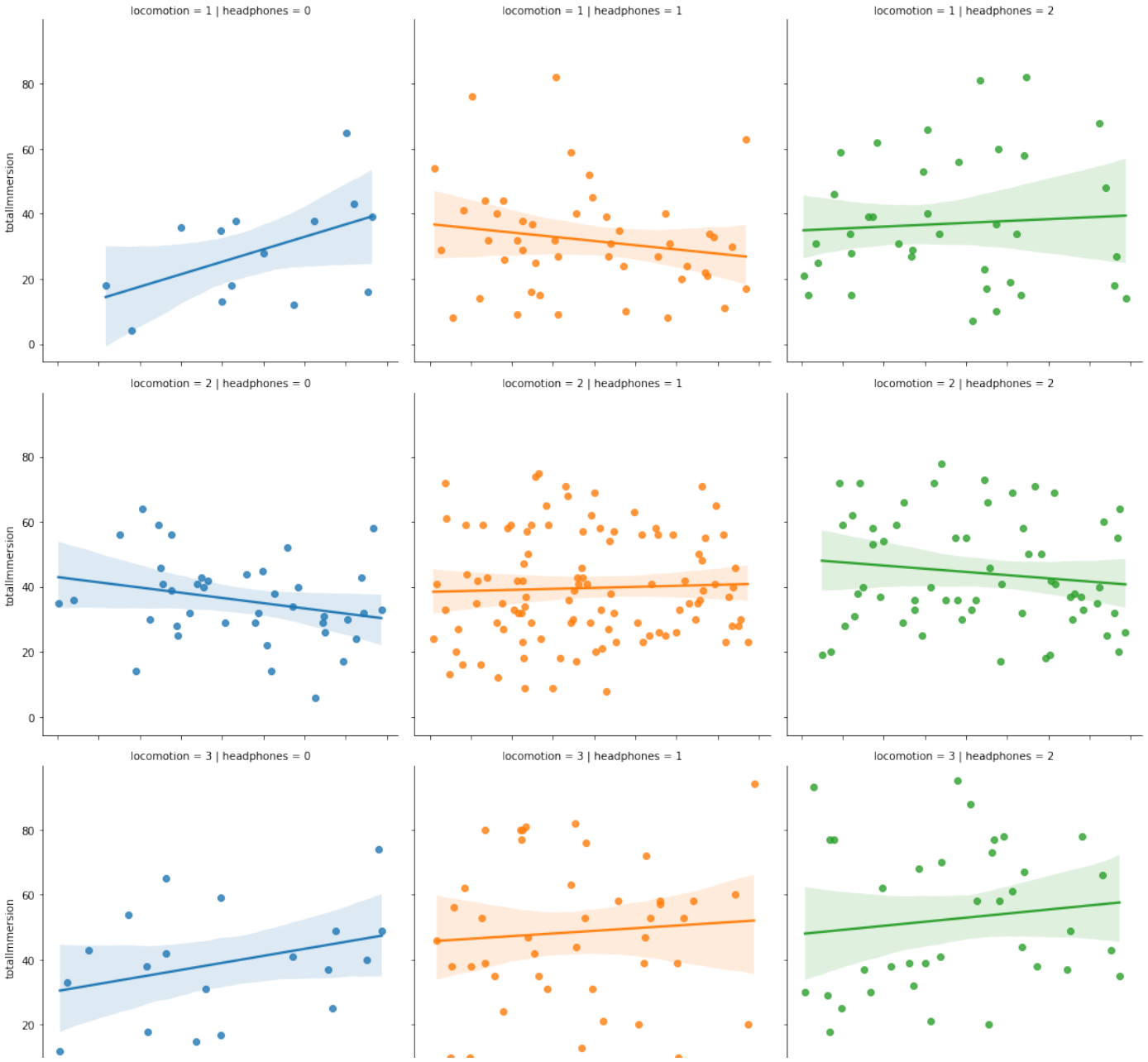
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:
The `size` parameter has been renamed to `height`; please update your code.
```

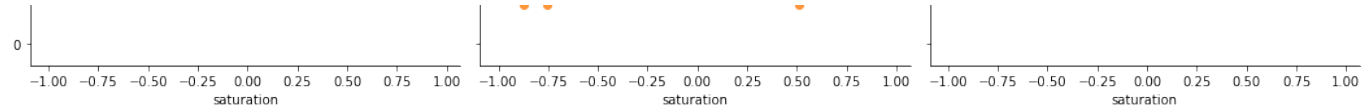



locomotion = 2 | headphones = 0
locomotion = 2 | headphones = 1
locomotion = 2 | headphones = 2
locomotion = 3 | headphones = 0
locomotion = 3 | headphones = 1
locomotion = 3 | headphones = 2
totalImmersion
height
0
20
40
60
80
200
400
600
800
1000
1200

```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='headphones', row = 'locomotion', size = 5, hue=
facet.map(sns.regplot, 'saturation', 'totalImmersion');
```

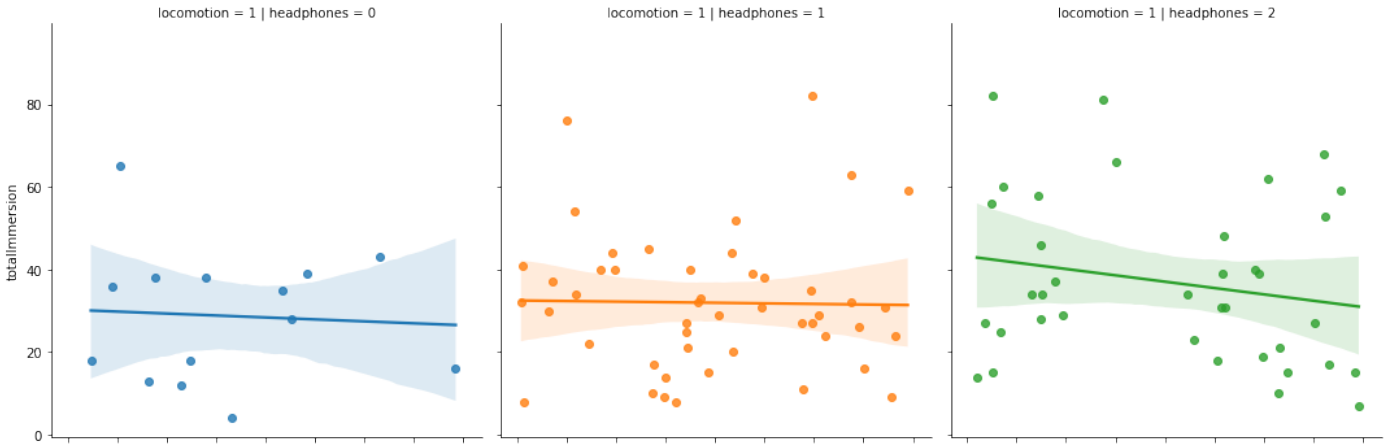
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:
The `size` parameter has been renamed to `height`; please update your code.
```

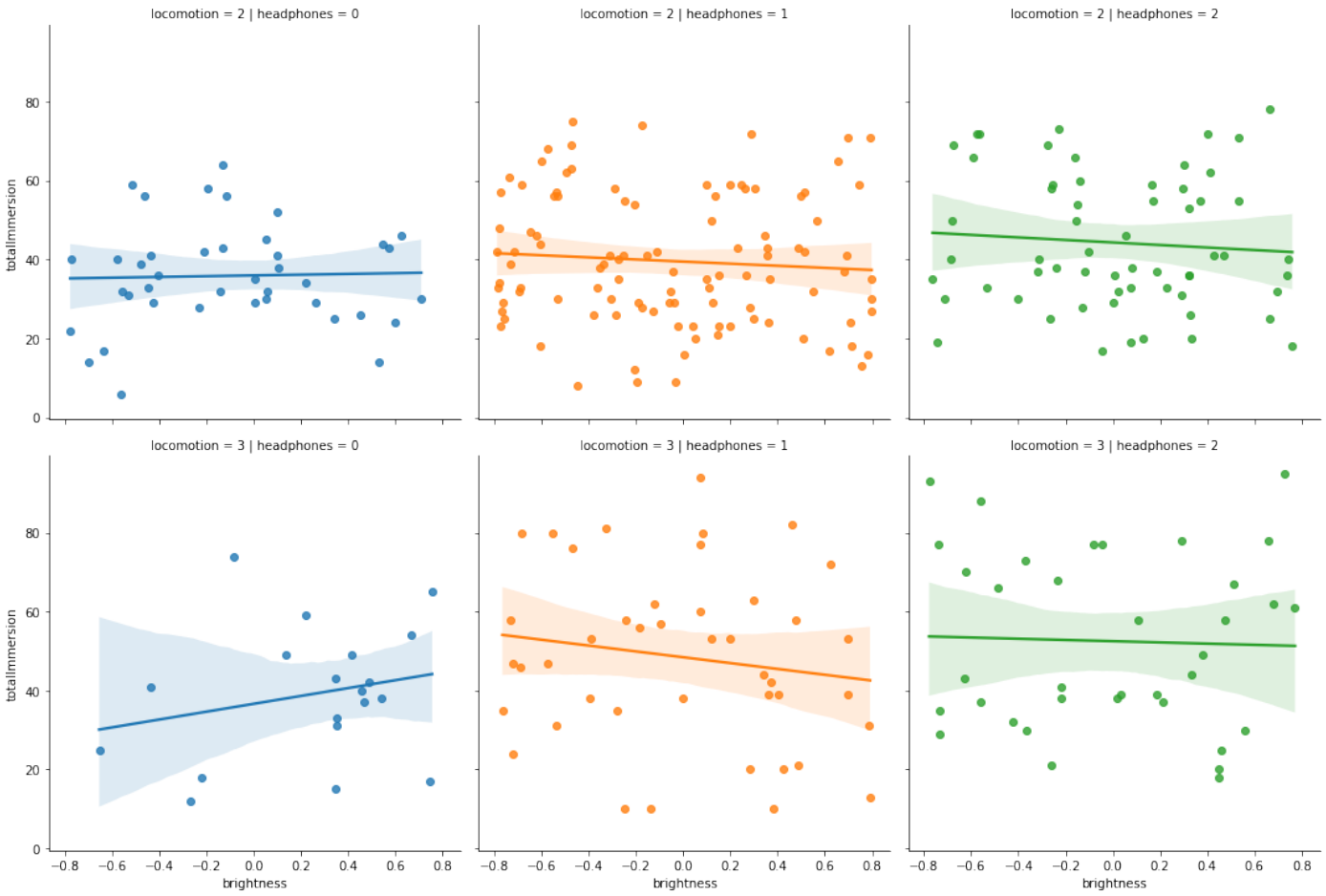





```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='headphones', row = 'locomotion', size = 5, hue=
facet.map(sns.regplot, 'brightness', 'totalImmersion');
```

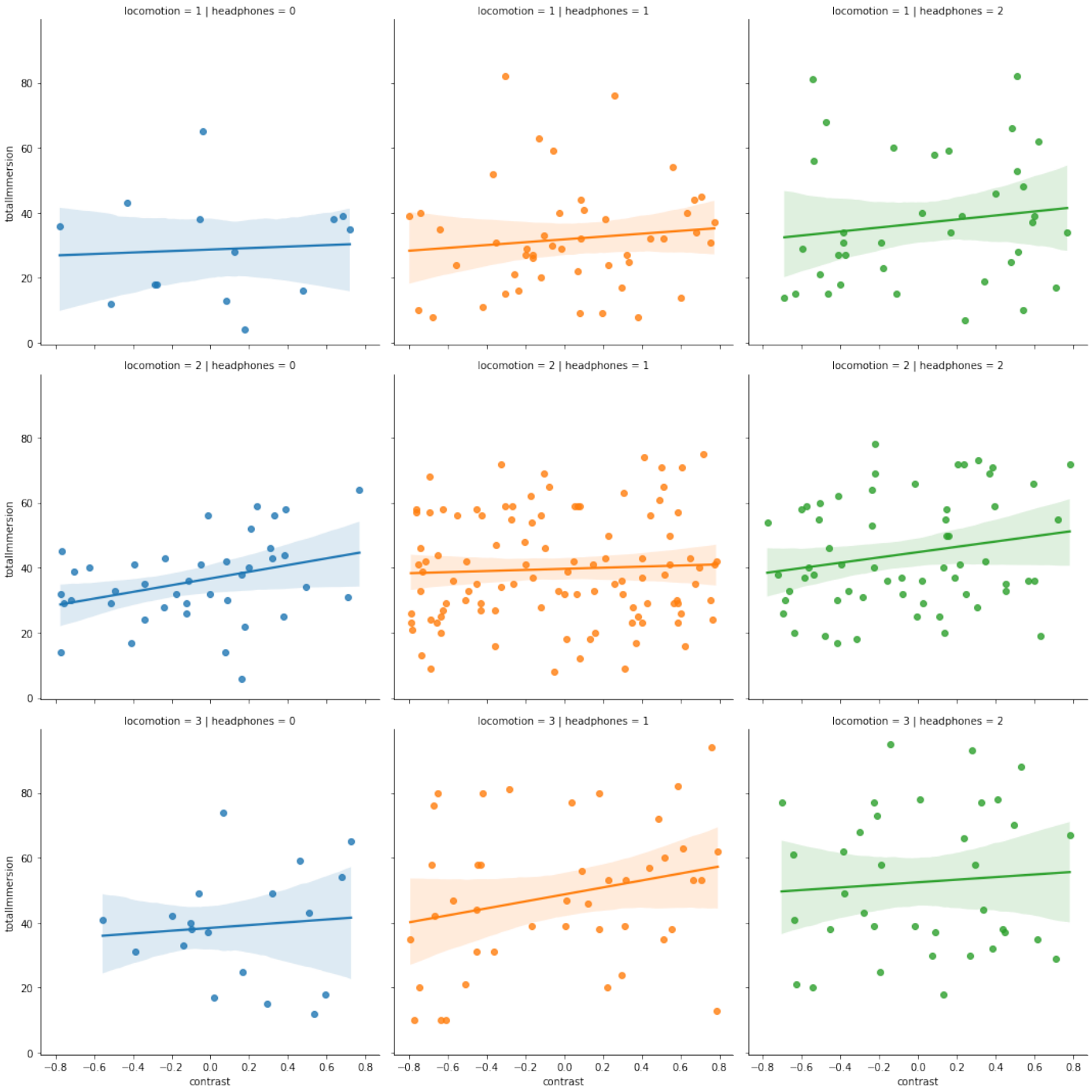
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:

The `size` parameter has been renamed to `height`; please update your code.
```




locomotion = 2 | headphones = 0
locomotion = 2 | headphones = 1
locomotion = 2 | headphones = 2
locomotion = 3 | headphones = 0
locomotion = 3 | headphones = 1
locomotion = 3 | headphones = 2
totalImmersion
brightness
0
20
40
60
80
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8

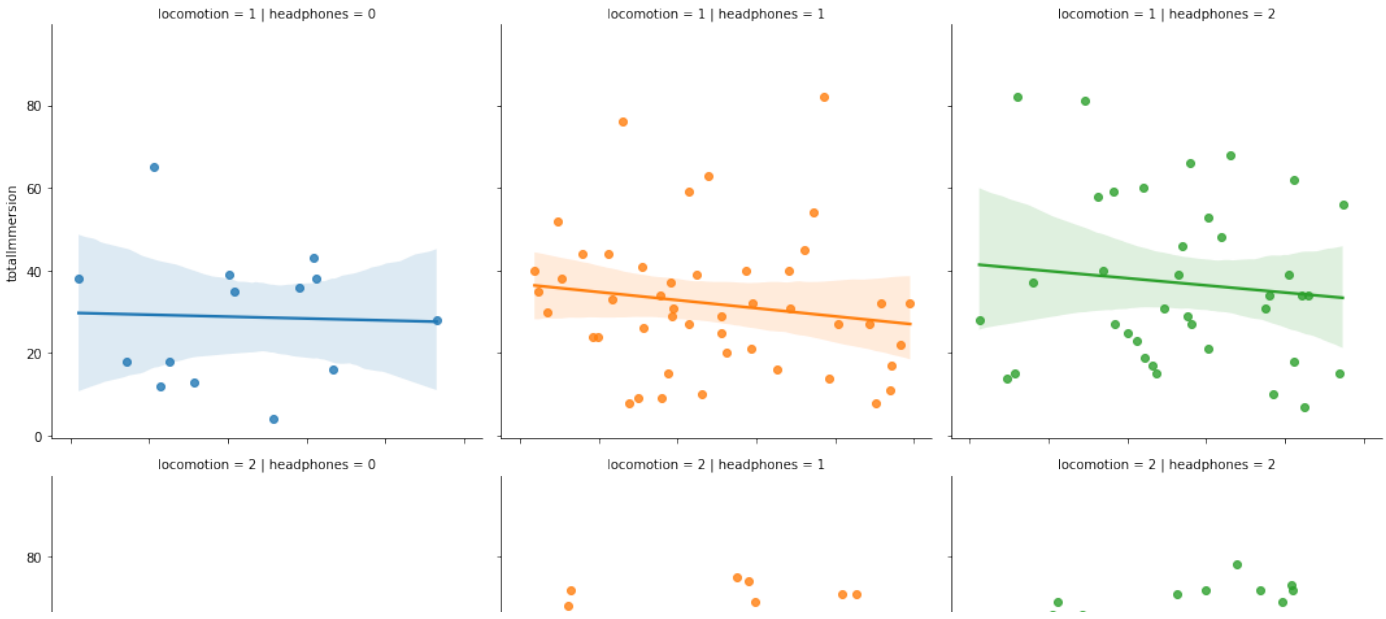
```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='headphones', row = 'locomotion', size = 5, hue=
facet.map(sns.regplot, 'contrast', 'totalImmersion');
```

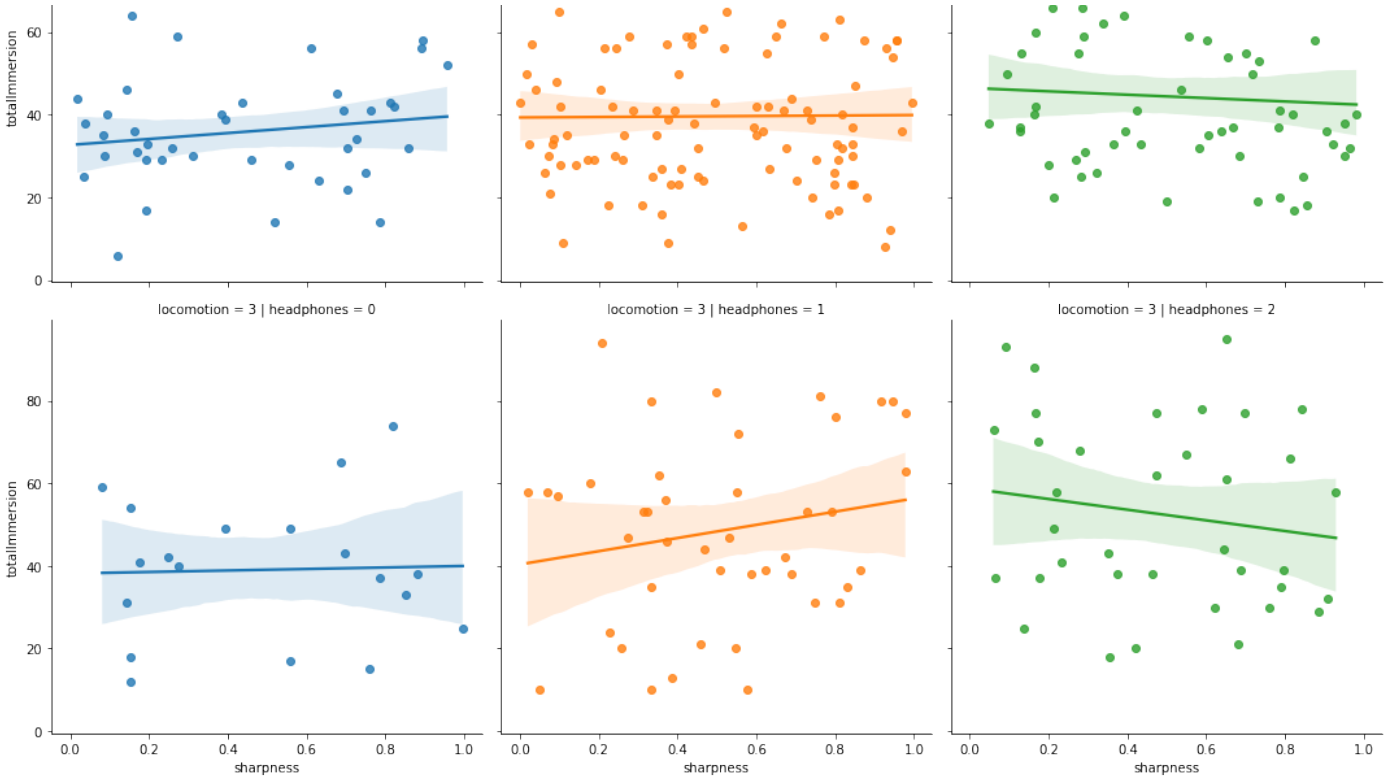
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:

The `size` parameter has been renamed to `height`; please update your code.
```



```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='headphones', row = 'locomotion', size = 5, hue=
facet.map(sns.regplot, 'sharpness', 'totalImmersion');
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:
The `size` parameter has been renamed to `height`; please update your code.
```

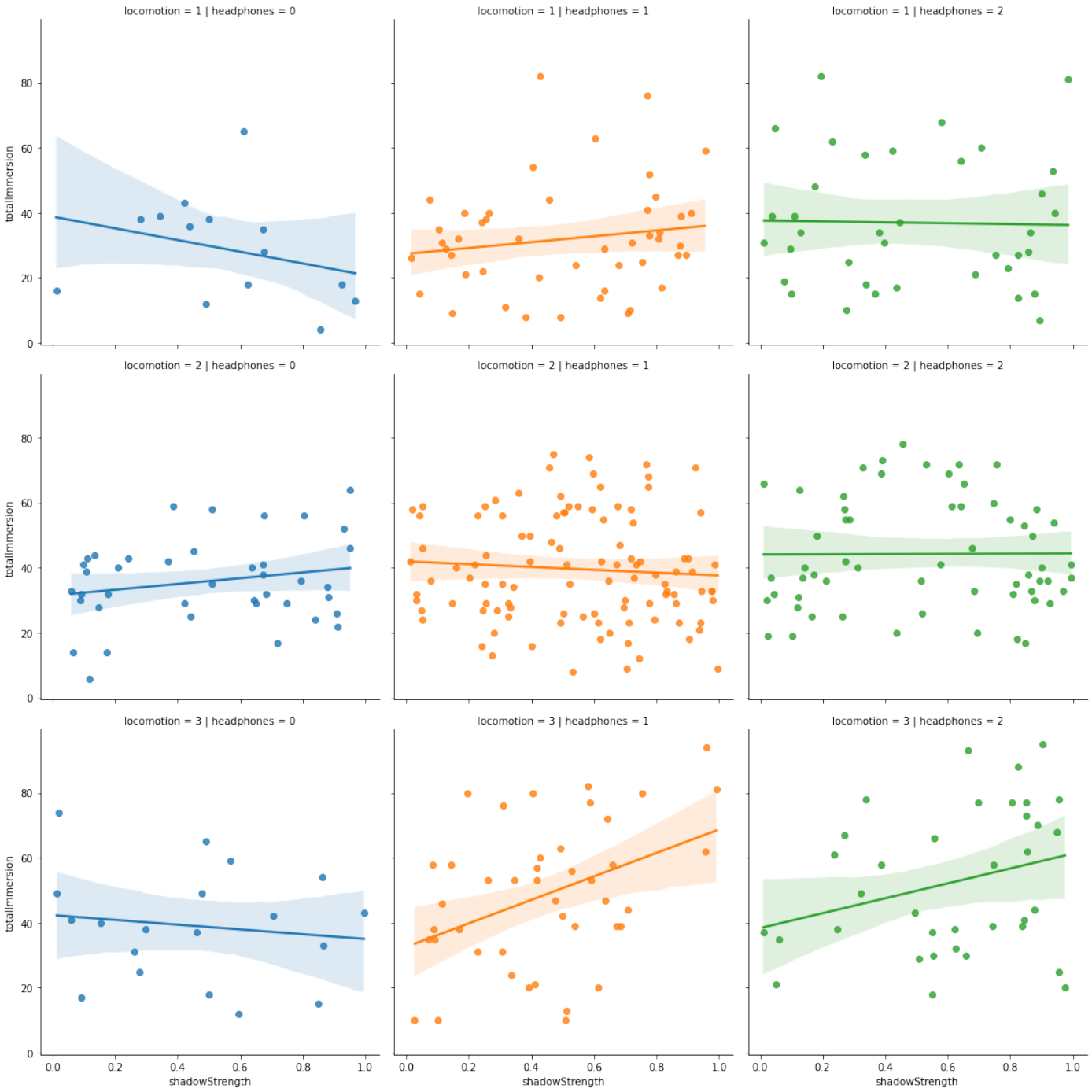





```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='headphones', row = 'locomotion', size = 5, hue=
```

```
3 facet.map(sns.regplot, 'shadowStrength', 'totalImmersion');
```

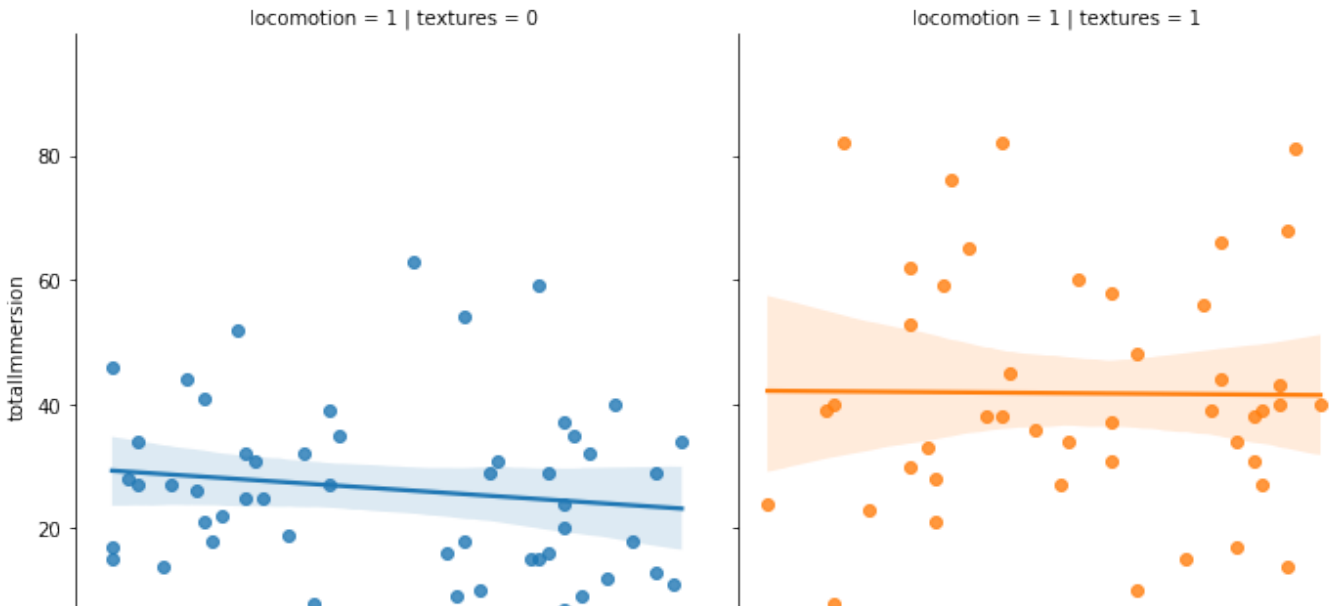
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:
The `size` parameter has been renamed to `height`; please update your code.
```

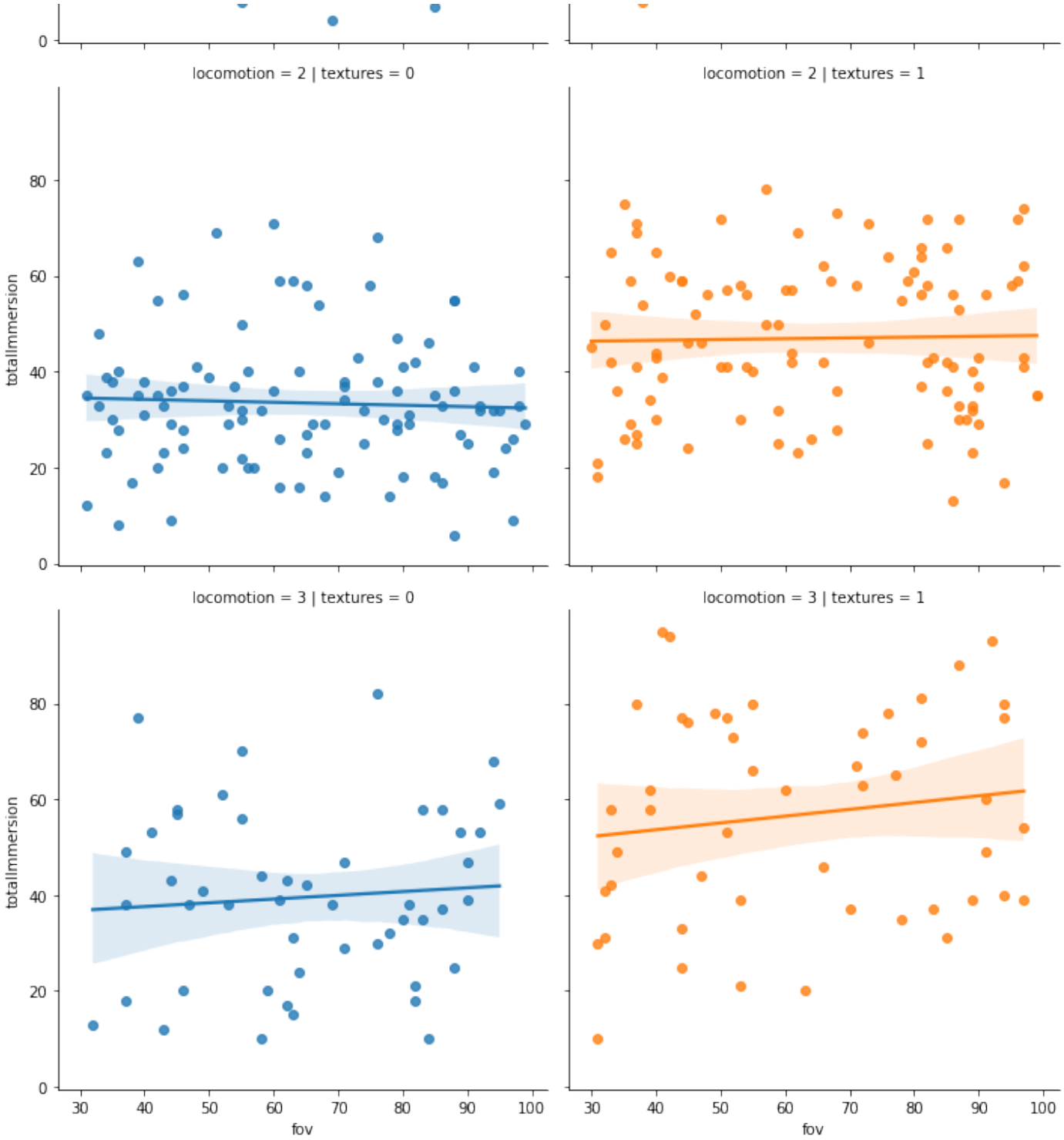


## Locomotion and Textures

```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='textures', row = 'locomotion', size = 5, hue='t
facet.map(sns.regplot, 'fov', 'totalImmersion');
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:
The `size` parameter has been renamed to `height`; please update your code.
```

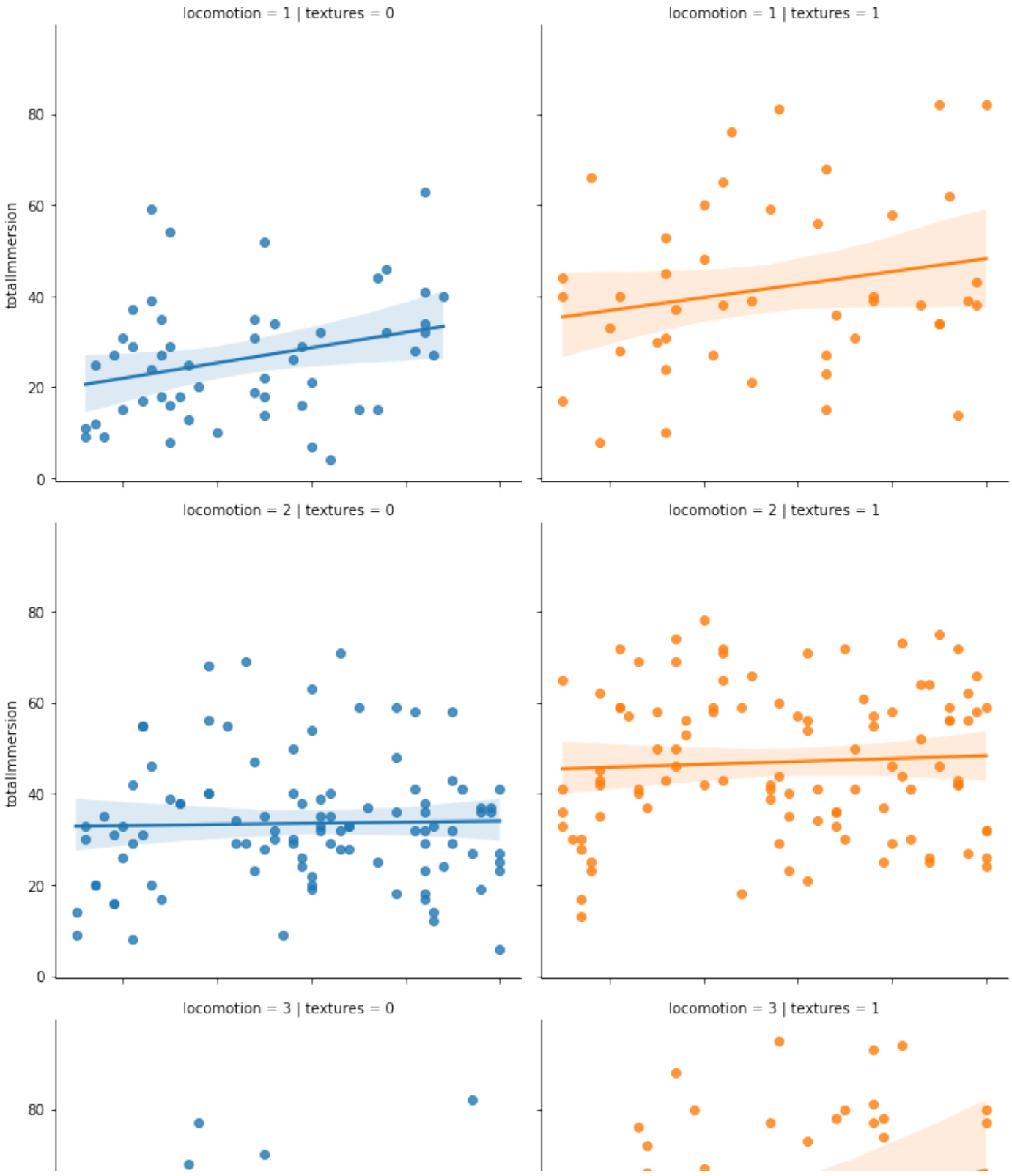


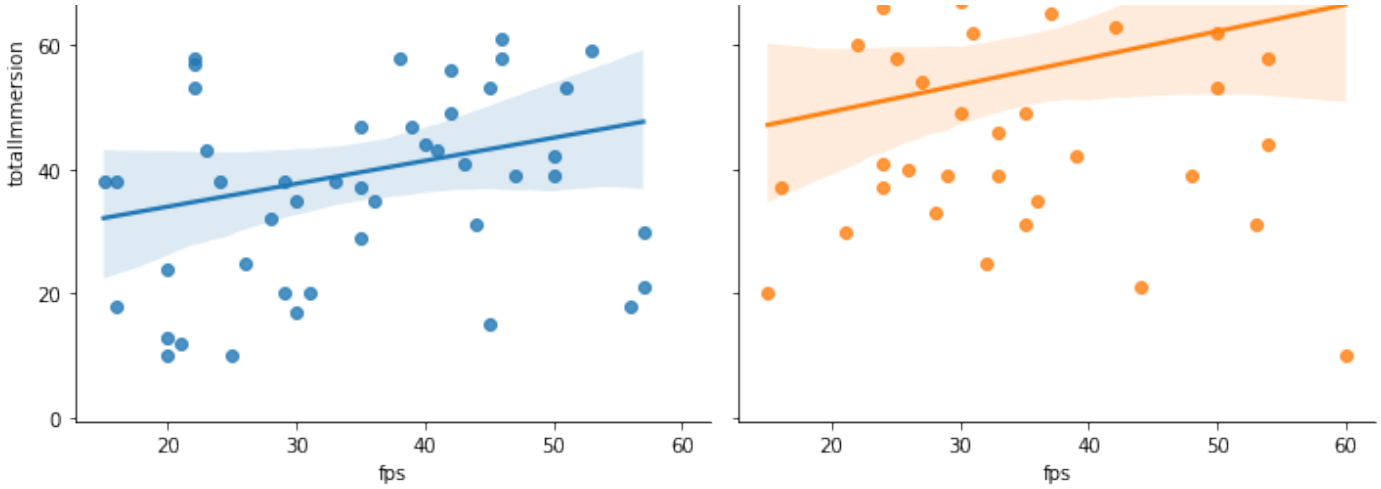



```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='textures', row = 'locomotion', size = 5, hue='t
```

```
facet.map(sns.regplot, 'fps', 'totalImmersion');
```

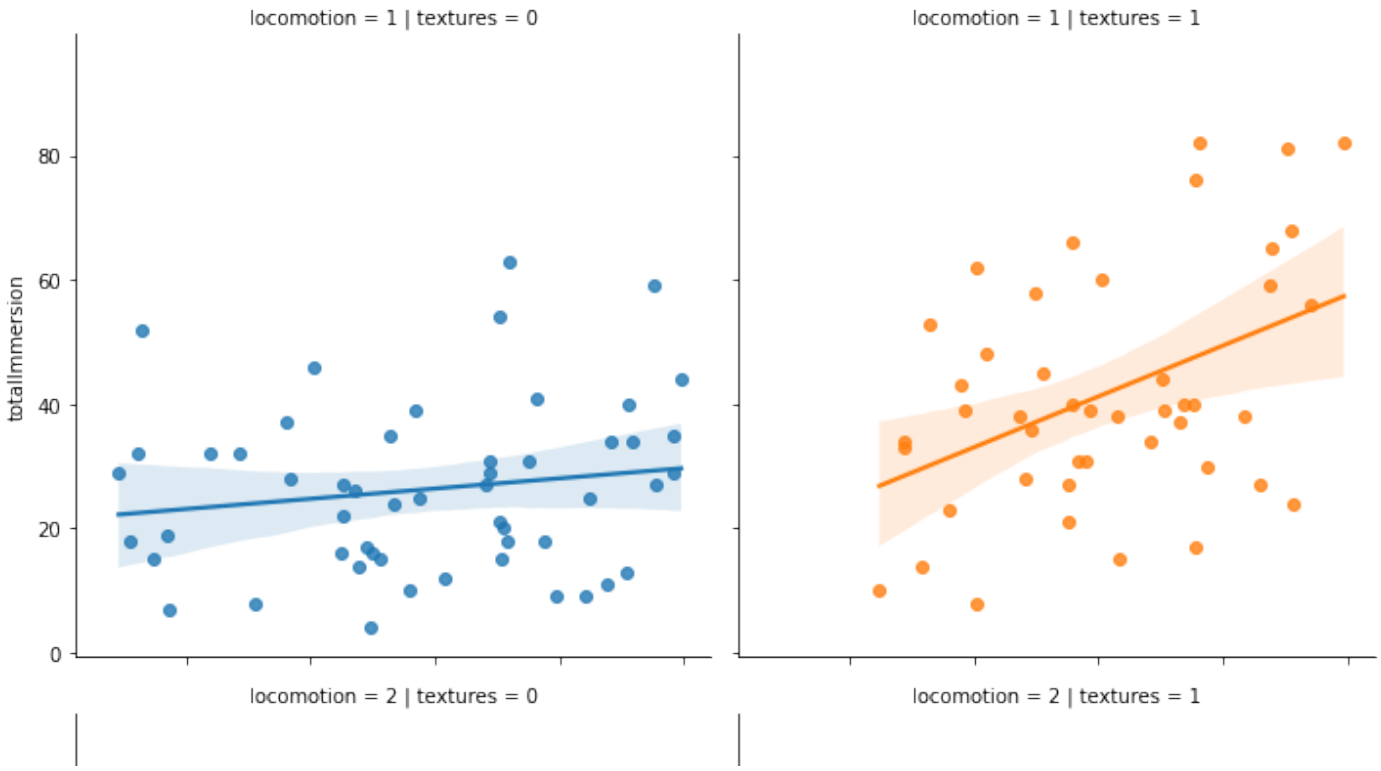
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:
The `size` parameter has been renamed to `height`; please update your code.
```

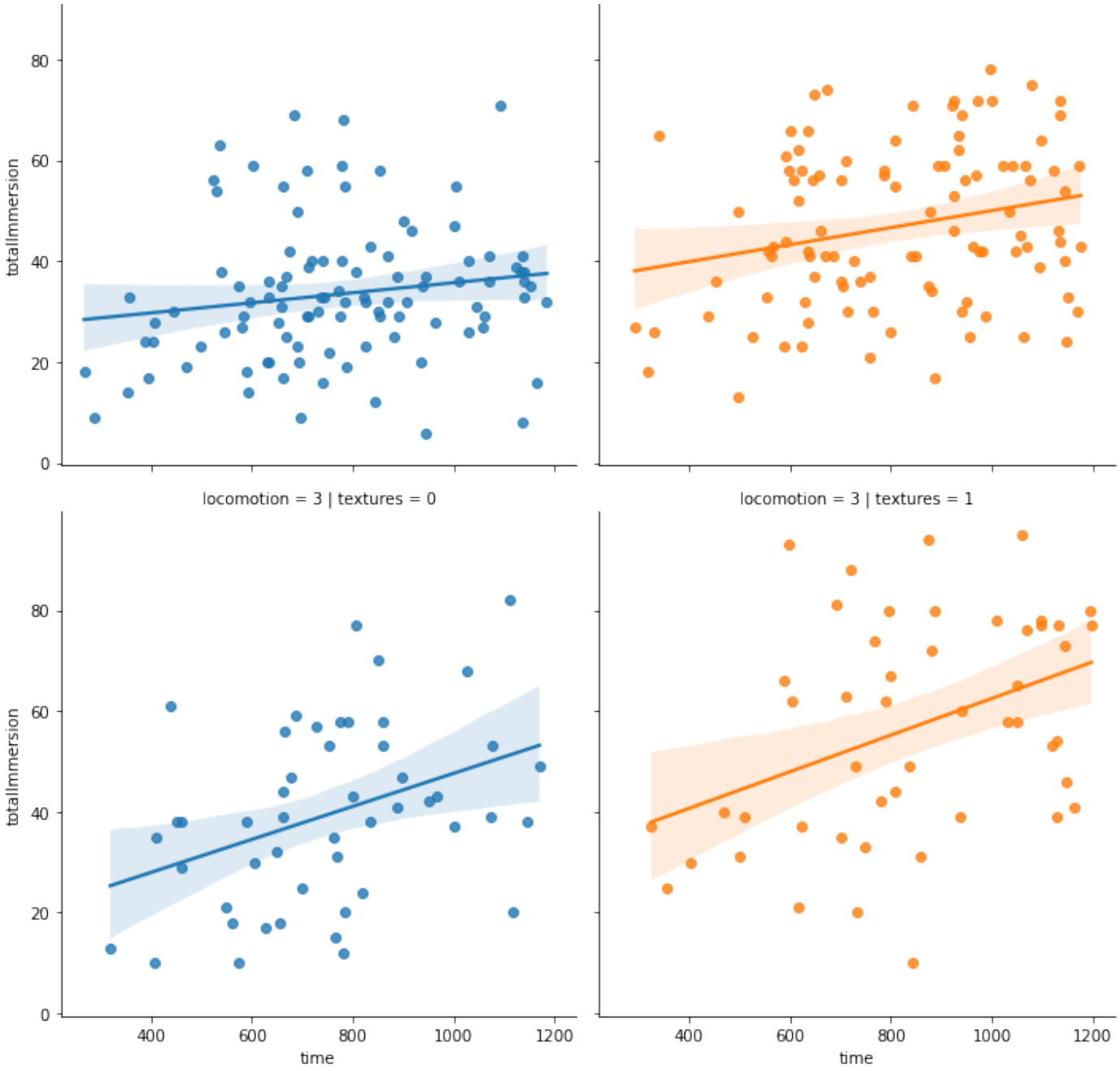





```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='textures', row = 'locomotion', size = 5, hue='t
facet.map(sns.regplot, 'time', 'totalImmersion');
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:

The `size` parameter has been renamed to `height`; please update your code.
```



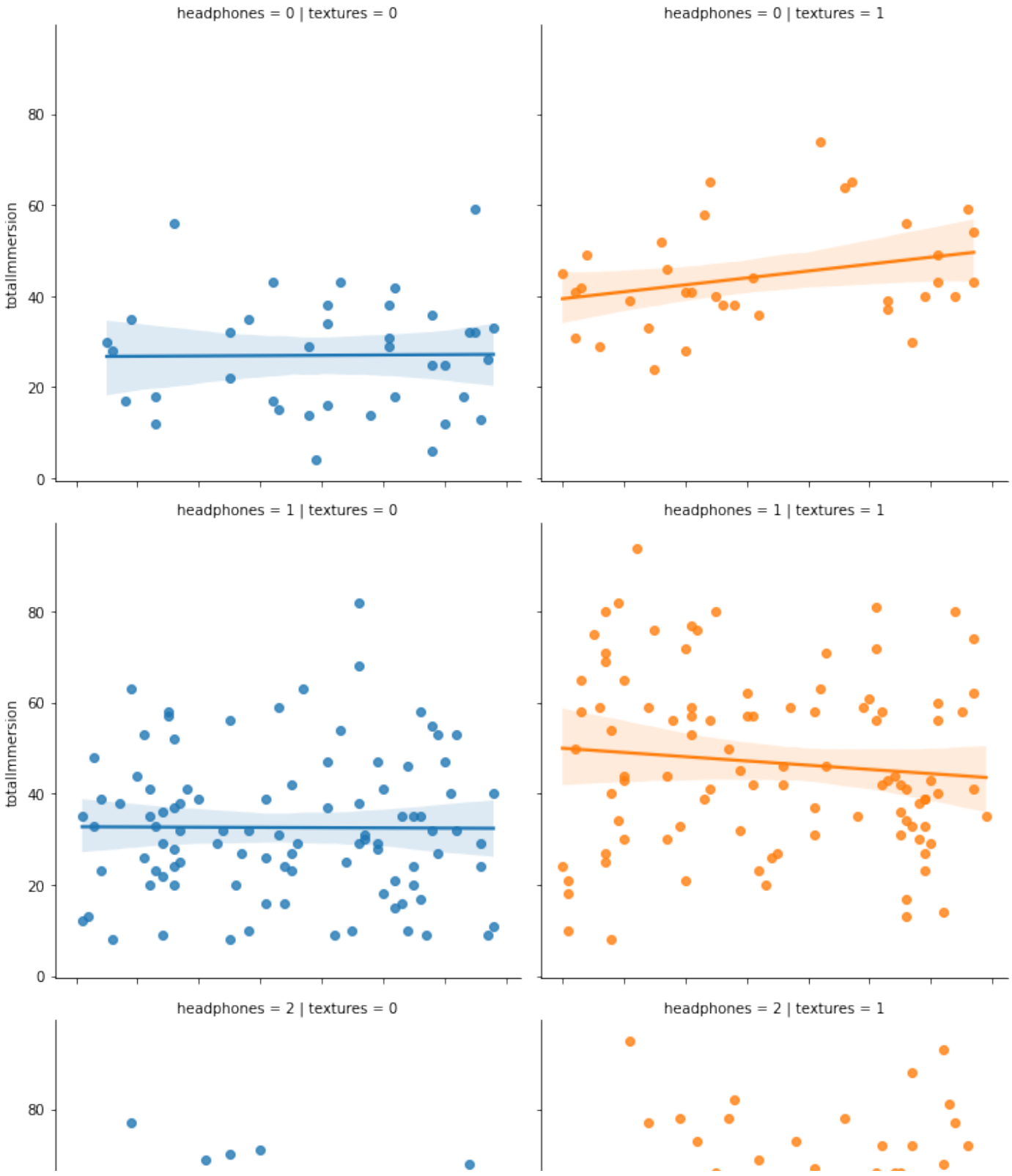


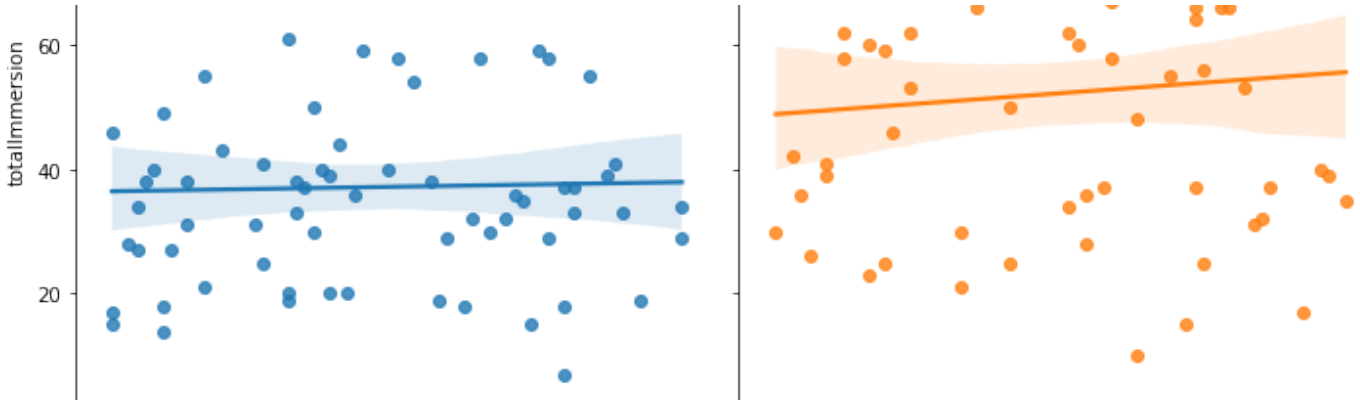

## Sound System and Textures

```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='textures', row = 'headphones', size = 5, hue='t
```

```
facet.map(sns.regplot, 'fov', 'totalImmersion');
```

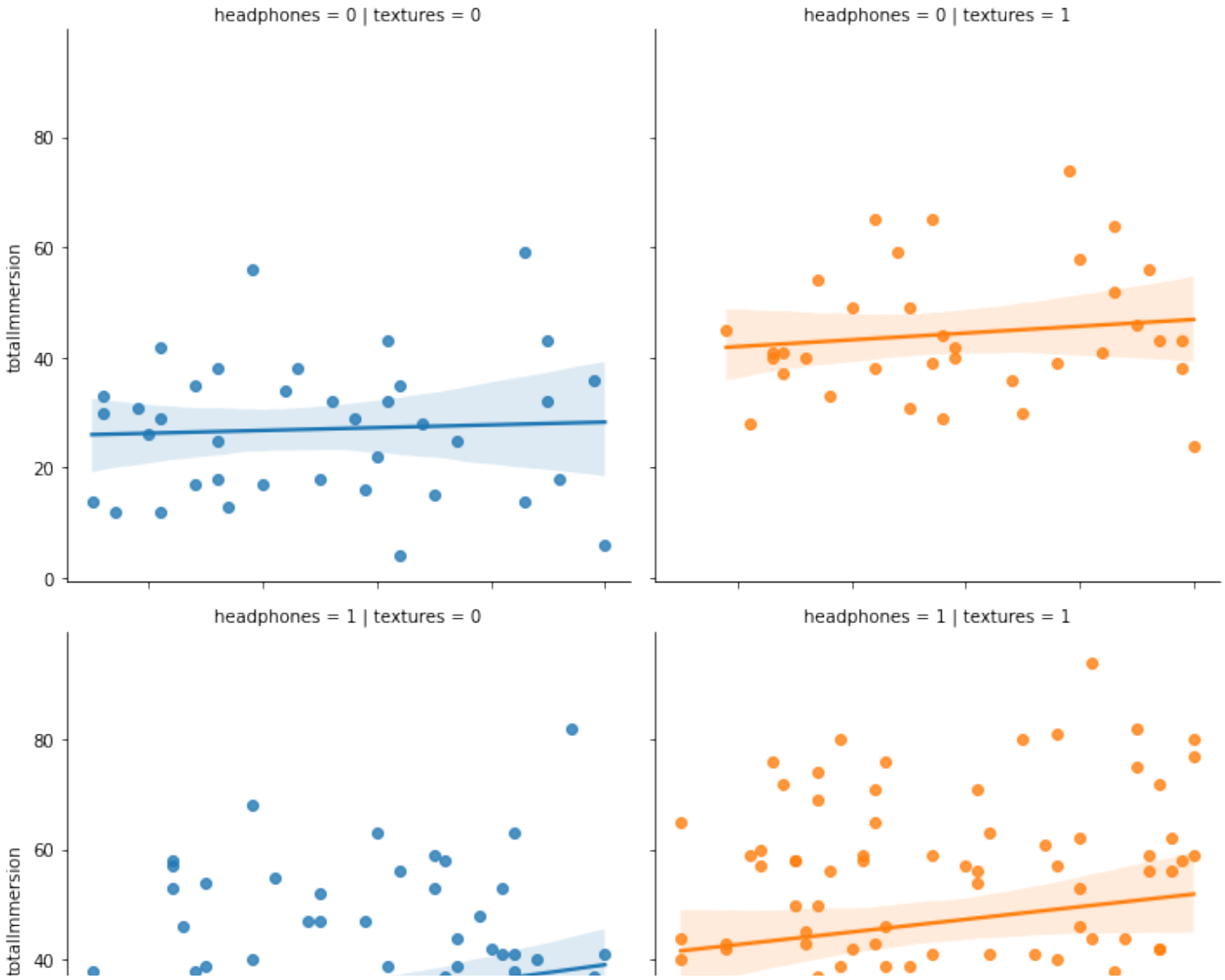
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:
The `size` parameter has been renamed to `height`; please update your code.
```

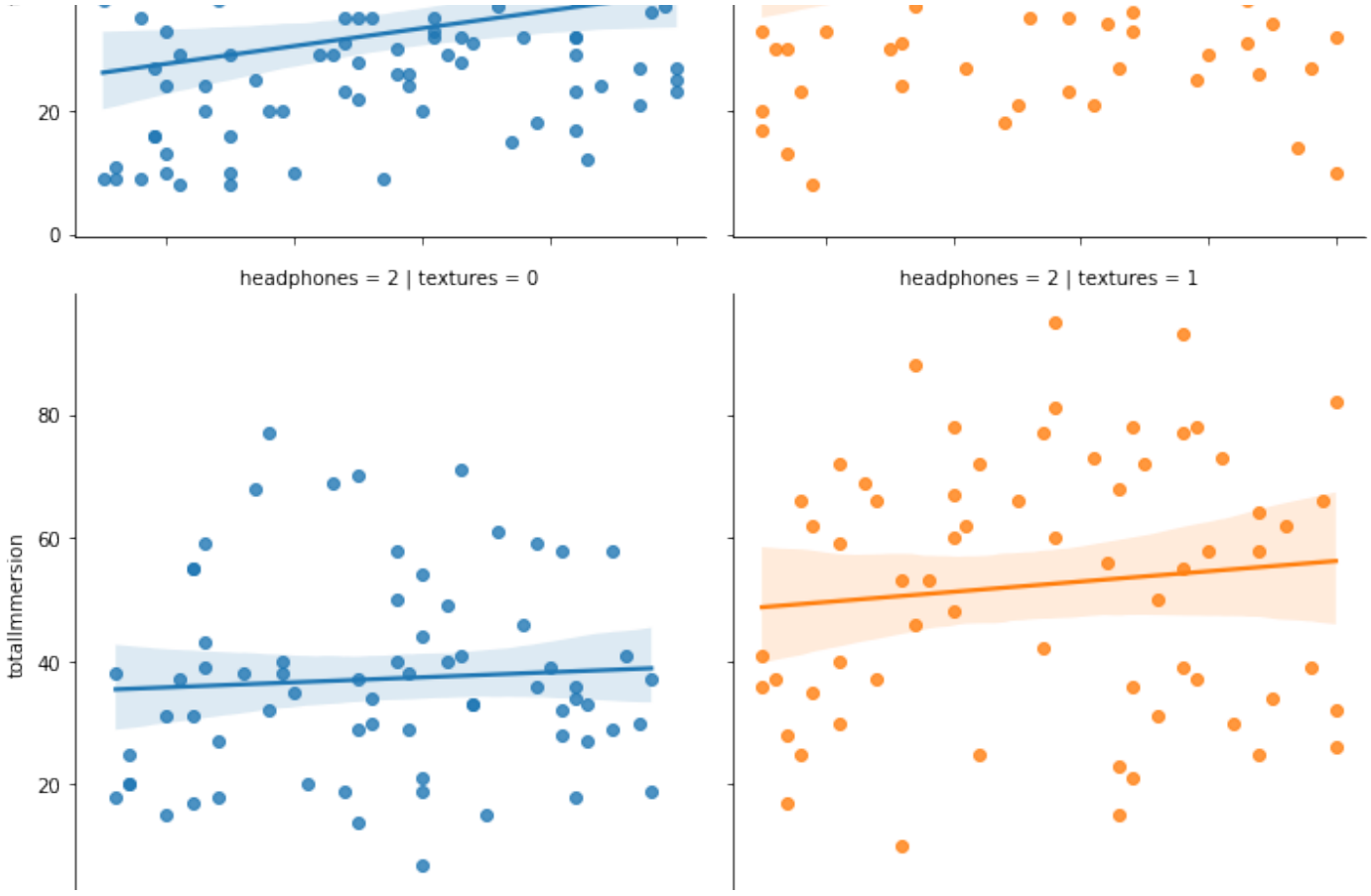





```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='textures', row = 'headphones', size = 5, hue='t
facet.map(sns.regplot, 'fps', 'totalImmersion');
```

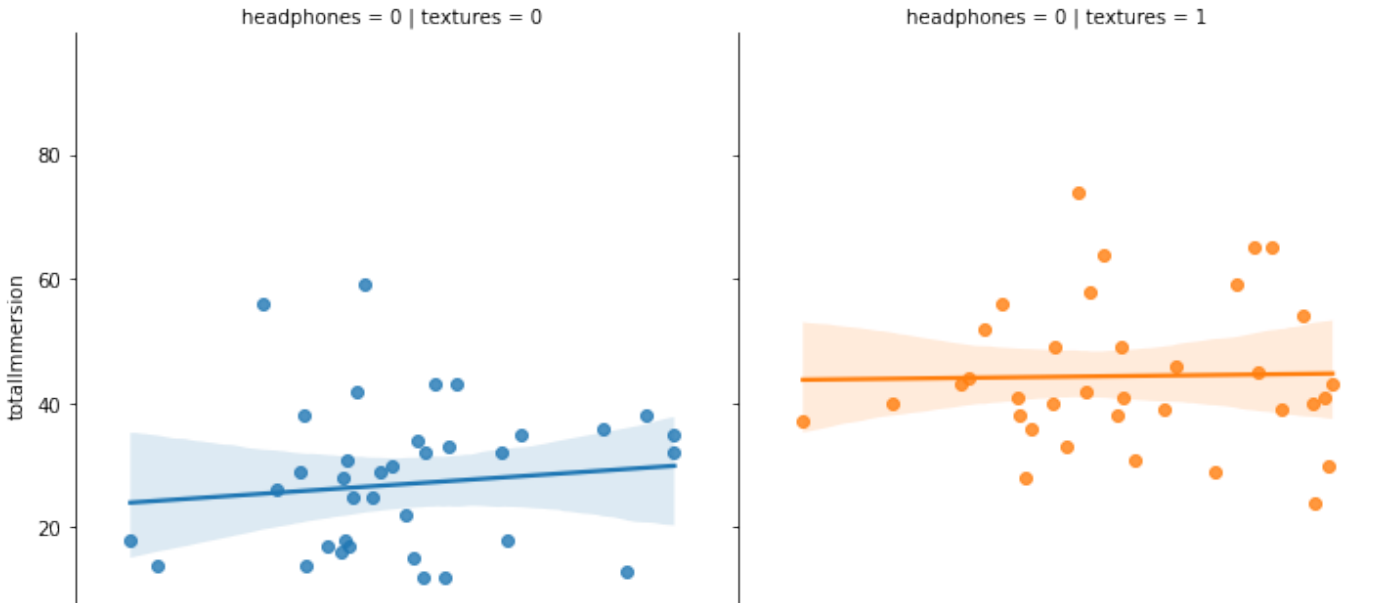
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:

The `size` parameter has been renamed to `height`; please update your code.
```

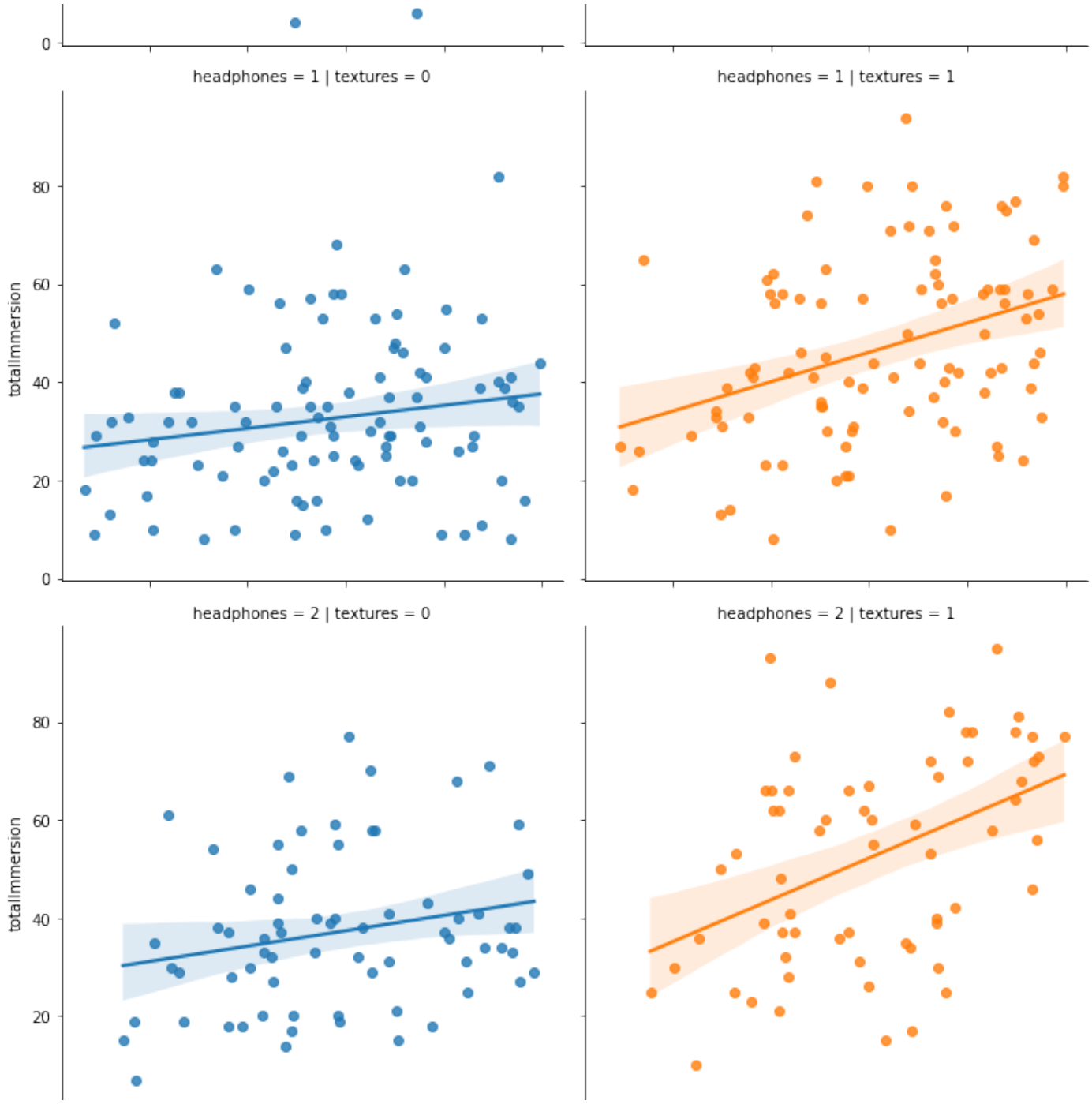





```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='textures', row = 'headphones', size = 5, hue='t
facet.map(sns.regplot, 'time', 'totalImmersion');
```

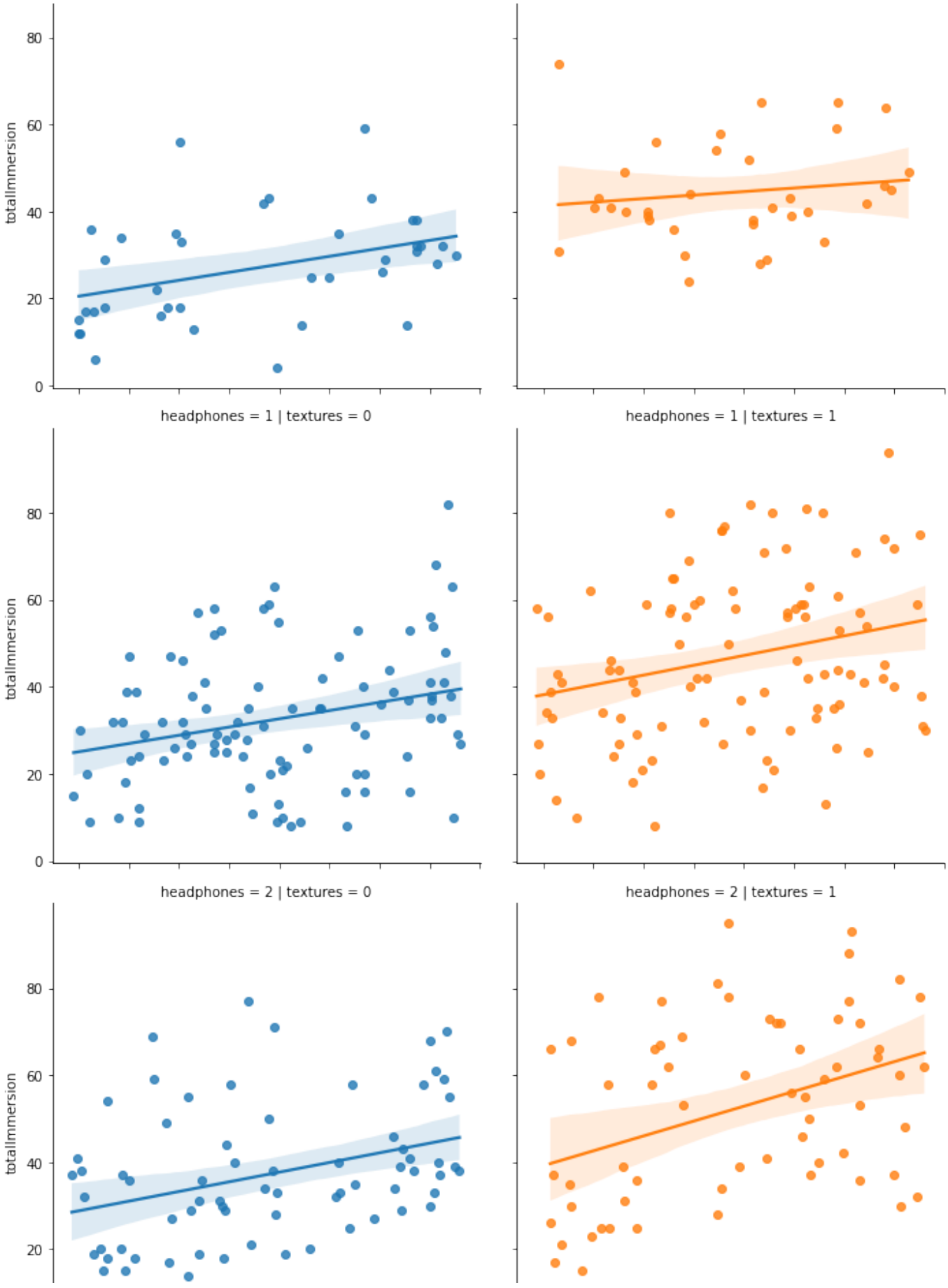
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:

The `size` parameter has been renamed to `height`; please update your code.
```

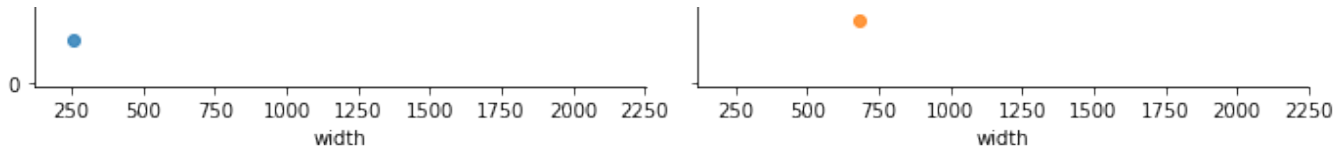





```
# create the FacetGrid
facet = sns.FacetGrid(vr, col='textures', row = 'headphones', size = 5, hue='t
facet.map(sns.regplot, 'width', 'totalImmersion');
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:337: UserWarning:
The `size` parameter has been renamed to `height`; please update your code.
```


totalImmersion
0
20
40
60
80
headphones = 1 | textures = 0
headphones = 1 | textures = 1
headphones = 2 | textures = 0
headphones = 2 | textures = 1


0
250
500
750
1000
1250
1500
1750
2000
2250
width
250
500
750
1000
1250
1500
1750
2000
2250
width