

# tu\_10\_multiAxisFigs\_2\_HW

February 19, 2023

## 1 Tutorial 10 homework

In this homework, you'll make a figure containing a scatter plot with marginal histograms similar to what we made earlier. Now, however, we have a few more tools at our disposal so we can probably make a better figure.

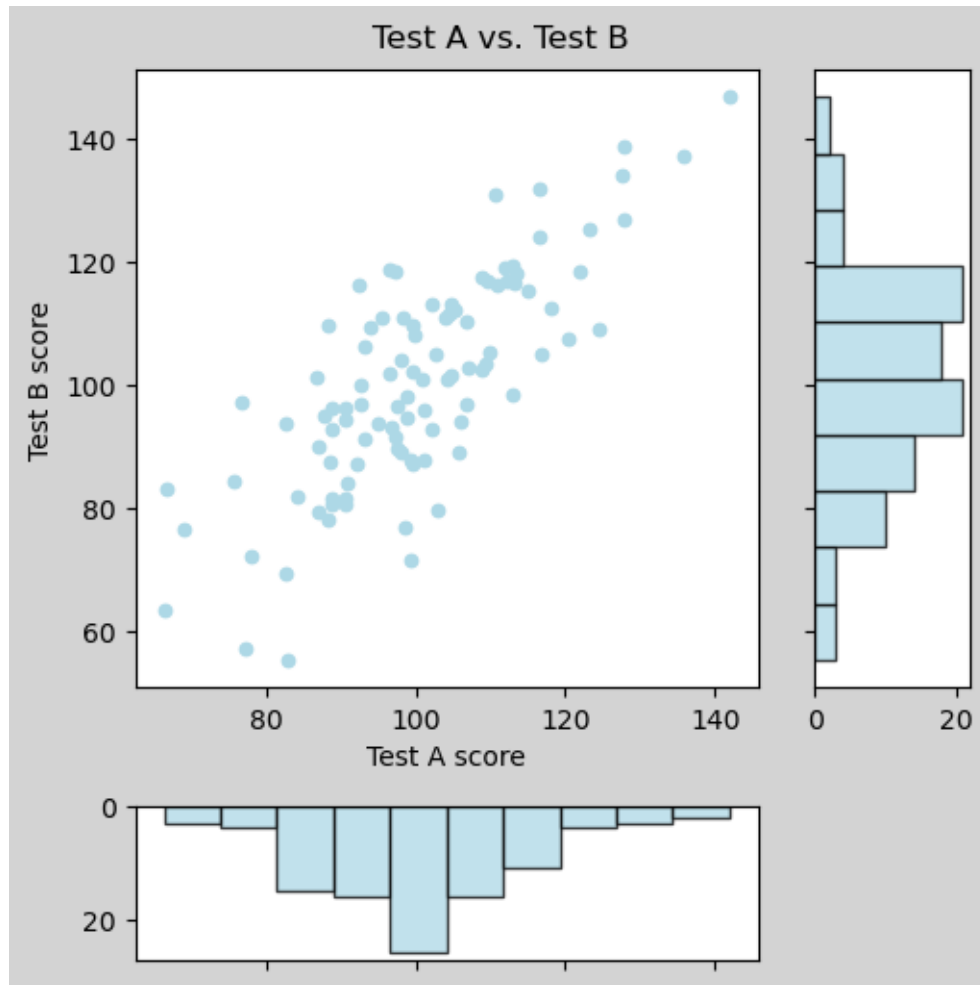
Here is some code to make a simulated data consisting of the scores on two tests (A and B) taken by 100 people.

```
[1]: import numpy as np
import matplotlib.pyplot as plt

my_rng = np.random.default_rng(seed = 42)
test_a = my_rng.normal(100, 15, (100,1))
test_b = test_a + my_rng.normal(0, 10, test_a.shape)
```

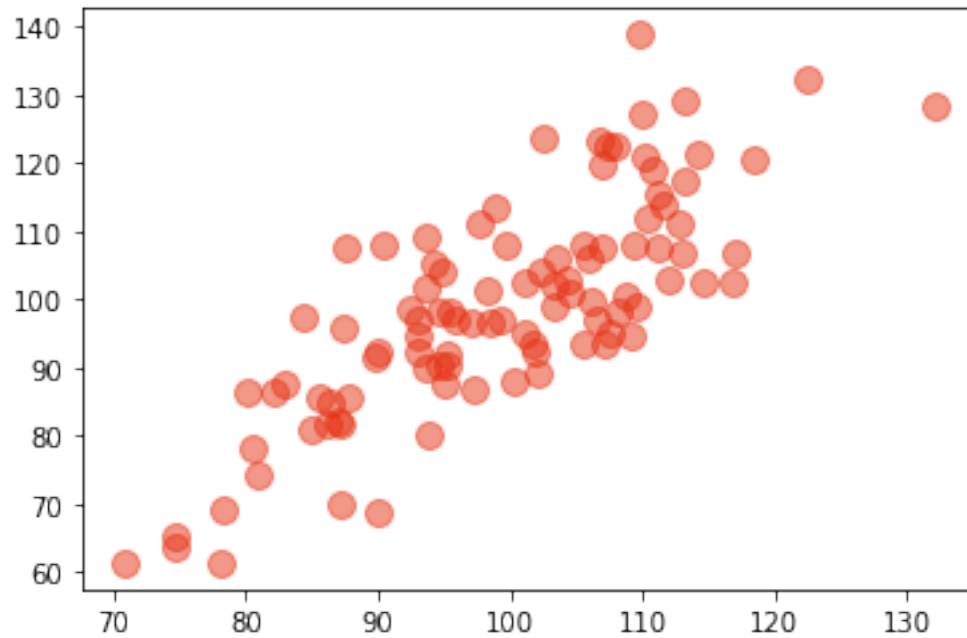
Let's make a figure featuring a scatter plot of the scores against one another, along with two supporting panels showing the histograms of the two test scores.

An example figure would look something like this:

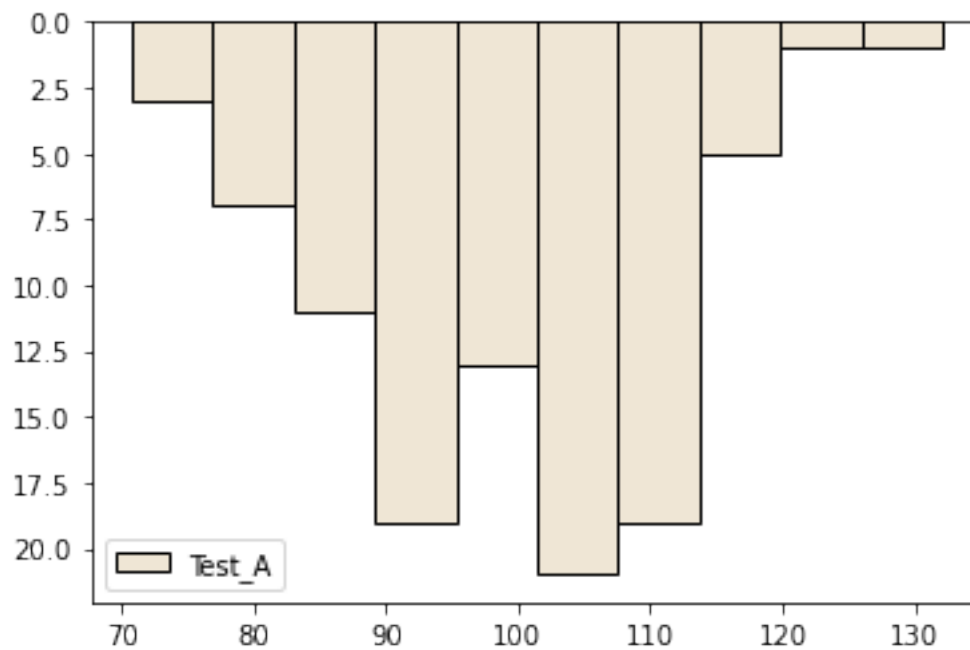


But your goal isn't to make a figure that looks exactly like this; your goal is to make a figure that looks better than this!

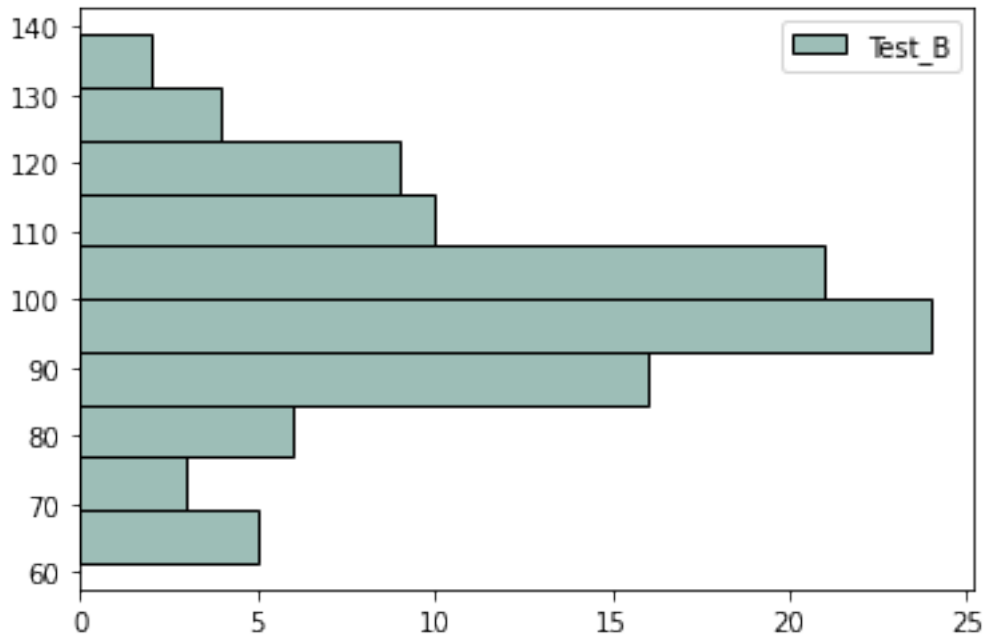
```
[2]: # Scatterplot Alone
plt.scatter(test_a, test_b, color = '#e73213', alpha = 0.5, s = 100);
```



```
[3]: # Histogram of test_a
plt.hist(test_a, bins = 10, color = '#efe6d5', orientation='vertical', edgecolor='black', label = 'Test_A,');
plt.gca().invert_yaxis();
plt.legend();
```



```
[4]: # Histogram of test_b
plt.hist(test_b, bins = 10, color = '#9dbeb7',
        orientation='horizontal', edgecolor = 'black', label = 'Test_B');
plt.legend();
```



```
[5]: # Combining Plot
gs_kw = dict(width_ratios=[4, 2],
              height_ratios=[3, 1],
              wspace=0.1, hspace=0.1) # spacing between the
rows and cols

fig, axd = plt.subplot_mosaic([['scatter', 'barB'],
                               ['barA', 'empty']],
                              gridspec_kw=gs_kw,
                              figsize=(7, 7),
                              layout="constrained")
fig.set_facecolor('#efe6d5')

# Scatter
axd['scatter'].scatter(test_a, test_b, color = '#B5EAD7', alpha = 0.6, s = 120,
                      marker = "X", edgecolors = 'black');
axd['scatter'].set_ylabel('Test B')
axd['scatter'].set_xlabel('Test A')
axd['scatter'].set_facecolor('white')
```

```

# Test A
axd['barA'].hist(test_a, bins = 10, color = '#537c78',
    ↪orientation='vertical',edgecolor = 'black',label = 'Test_A',);
axd['barA'].invert_yaxis();
axd['barA'].legend();
axd['barA'].set_facecolor('white')

# Test B
axd['barB'].hist(test_b, bins = 10, color = '#9dbeb7',
    ↪orientation='horizontal',edgecolor = 'black',label = 'Test_B');
axd['barB'].legend();
axd['barB'].set_facecolor('white')

# No Fig.
axd['empty'].axis('off'); # Turn off the last box ('empty')

fig.suptitle('Test A vs Test B', fontsize = 20);

```

