

# 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.

```
In [1]: import numpy as np

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!

```
In [6]: import matplotlib.pyplot as plt

fig = plt.figure(figsize=(8, 8))
gs = fig.add_gridspec(2, 2, width_ratios=(4, 1), height_ratios=(1, 4), wspace=0.5)

ax_scatter = fig.add_subplot(gs[1, 0])
ax_histx = fig.add_subplot(gs[0, 0], sharex=ax_scatter)
ax_histy = fig.add_subplot(gs[1, 1], sharey=ax_scatter)

ax_scatter.scatter(test_a[:, 0], test_b[:, 0], alpha=0.7, color='b')
ax_scatter.set_xlabel("Test A Scores")
ax_scatter.set_ylabel("Test B Scores")
ax_scatter.grid(True)

ax_histx.hist(test_a[:, 0], bins=15, color='b', edgecolor='k')
ax_histx.set_ylabel("Count")
ax_histx.tick_params(labelbottom=False)

ax_histy.hist(test_b[:, 0], bins=15, orientation='horizontal', color='b', edgecolor='k')
ax_histy.set_xlabel("Count")
ax_histy.tick_params(labelleft=False)

plt.show()
```

