f-16-jupyter-grad-to

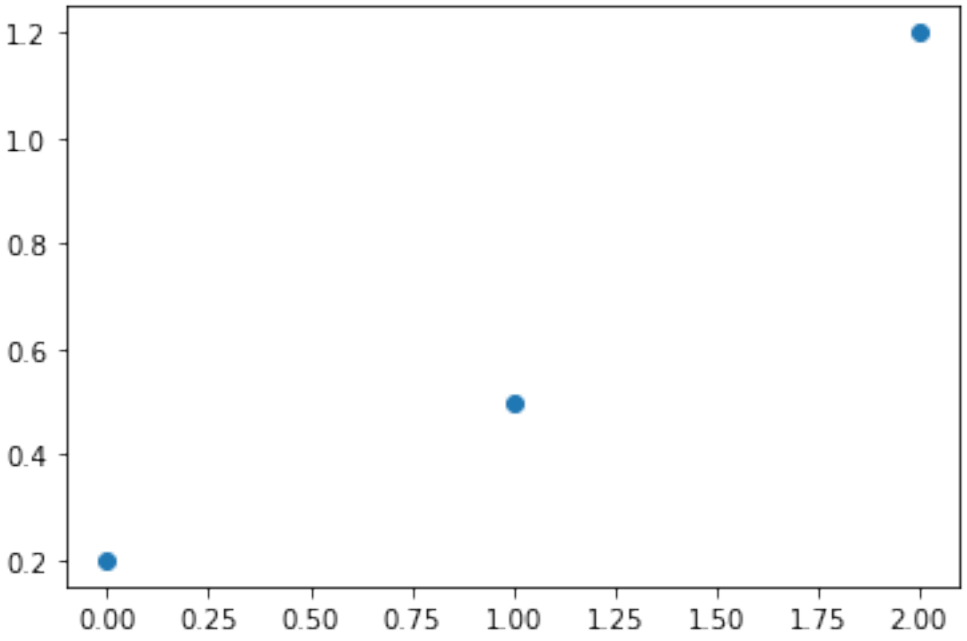
March 25, 2021

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

[2]:
```
x = np.array([0.0, 1.0, 2.0])
y = np.array([0.2, 0.5, 1.2])
```

[3]:
```
fig, ax = plt.subplots()
ax.plot(x, y, 'o')
```

[3]:
```
[<matplotlib.lines.Line2D at 0x10e729520>]
```



[4]:
```
cols = len(x)
```

[5]:
```
cols
```

[5]:
```
3
```

[6]:
```
a = np.vander(x, cols)
```

[7]:
```
a
```

[7]:
```
array([[0., 0., 1.],
       [1., 1., 1.],
       [4., 2., 1.]])
```

[8]:
```
koeffs = np.linalg.solve(a, y[:, np.newaxis])
```

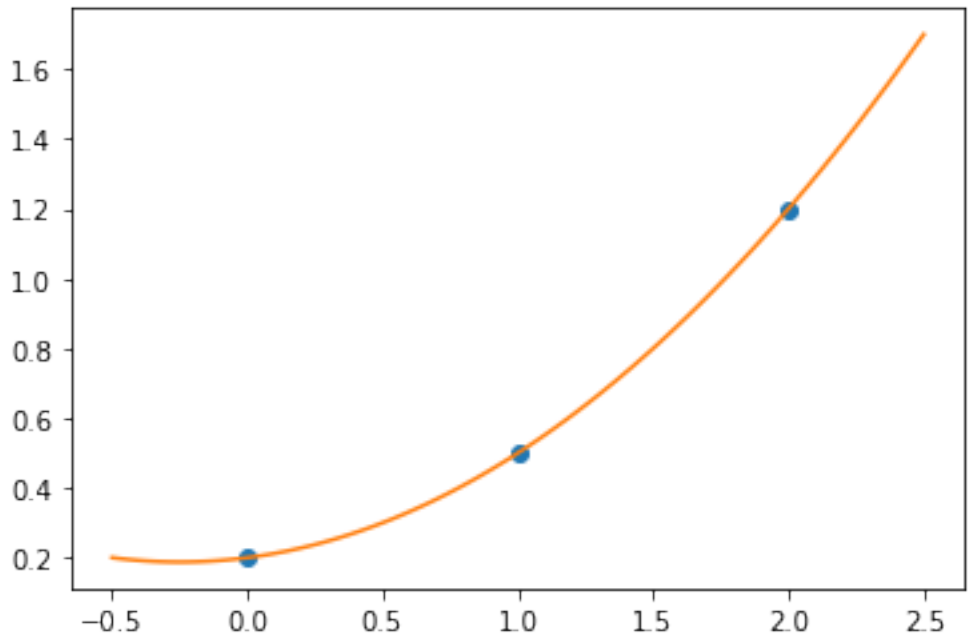
[9]:
```
koeffs
```

[9]:
```
array([[0.2],
       [0.1],
       [0.2]])
```

[10]:
```
t = np.linspace(-0.5, 2.5, 100)

fig, ax = plt.subplots()
ax.plot(x, y, 'o')
ax.plot(t, np.vander(t, cols) @ koeffs)
```

[10]:
```
[<matplotlib.lines.Line2D at 0x10e826460>]
```



[11]:
```
def forbedret_gram_schmidt(a):
    _, k = a.shape
```

```
    q = np.copy(a)
    r = np.zeros((k, k))
    for i in range(k):
        r[i, i] = np.linalg.norm(q[:, i])
        q[:, i] /= r[i, i]
        r[[i], i+1:] = q[:, [i]].T @ q[:, i+1:]
        q[:, i+1:] -= q[:, [i]] @ r[[i], i+1:]
    return q, r
```

[12]:
```
cols = 2
```

[13]:
```
a = np.vander(x, cols)
a
```

[13]:
```
array([[0., 1.],
       [1., 1.],
       [2., 1.]])
```

[14]:
```
q, r = forbedret_gram_schmidt(a)
```

[15]:
```
q
```

[15]:
```
array([[ 0.        ,  0.91287093],
       [ 0.4472136 ,  0.36514837],
       [ 0.89442719, -0.18257419]])
```

[16]:
```
r
```

[16]:
```
array([[2.23606798, 1.34164079],
       [0.        , 1.09544512]])
```

[17]:
```
c = q.T @ y[:, np.newaxis]
```

[18]:
```
c
```

[18]:
```
array([[1.29691943],
       [0.14605935]])
```

[19]:
```
koeffs = np.linalg.solve(r, c)
koeffs
```

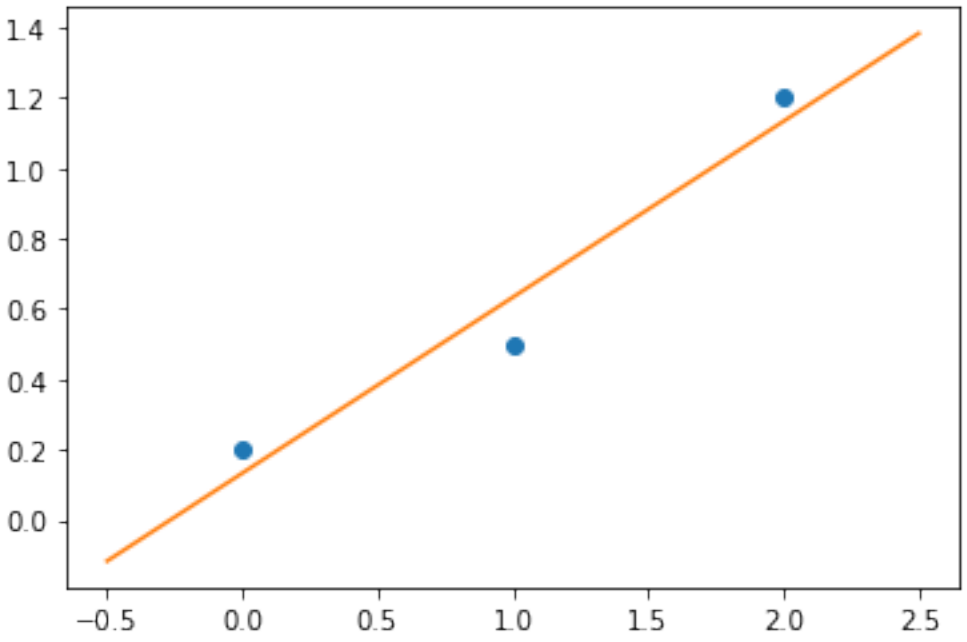
[19]:
```
array([[0.5       ],
       [0.13333333]])
```

[20]:
```
t = np.linspace(-0.5, 2.5, 100)

fig, ax = plt.subplots()
ax.plot(x, y, 'o')
```

```
ax.plot(t, np.vander(t, cols) @ koeffs)
```

[20]: `[<matplotlib.lines.Line2D at 0x10e892a60>]`



[ ]: