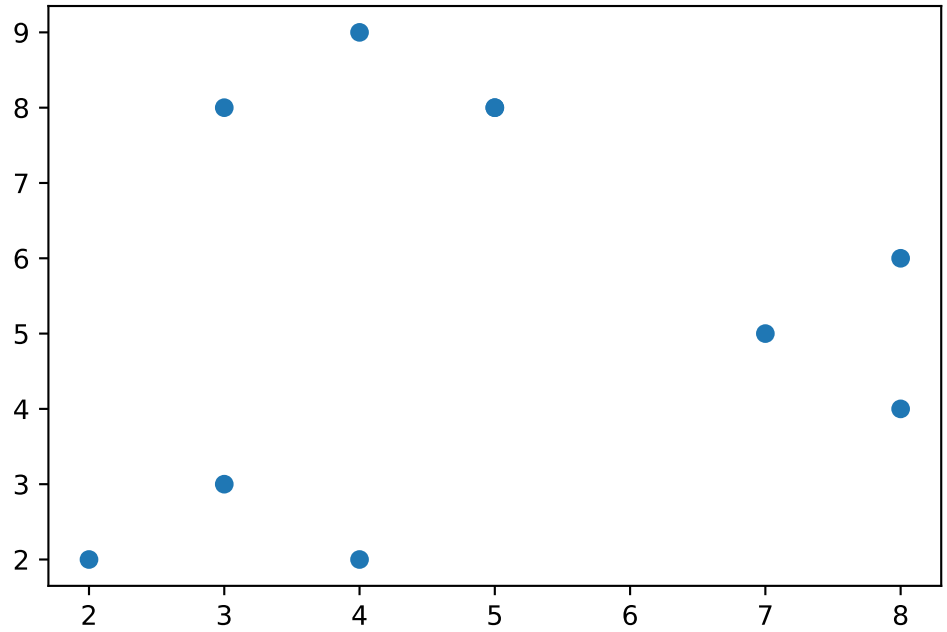
Needell_Coen_HW7

March 14, 2020

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.spatial.distance import cdist
np.random.seed(101)
```

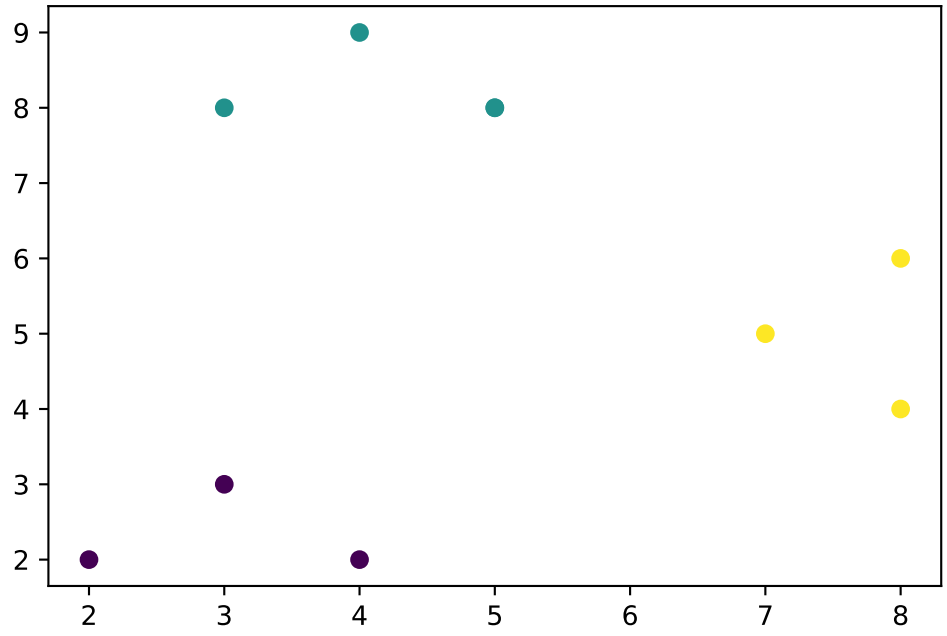
In [2]:
```
x = np.array([(5,8,7,8,3,4,2,3,4,5), (8,6,5,4,3,2,2,8,9,8)]).transpose()
plt.scatter(x[:, 0], x[:, 1])
```

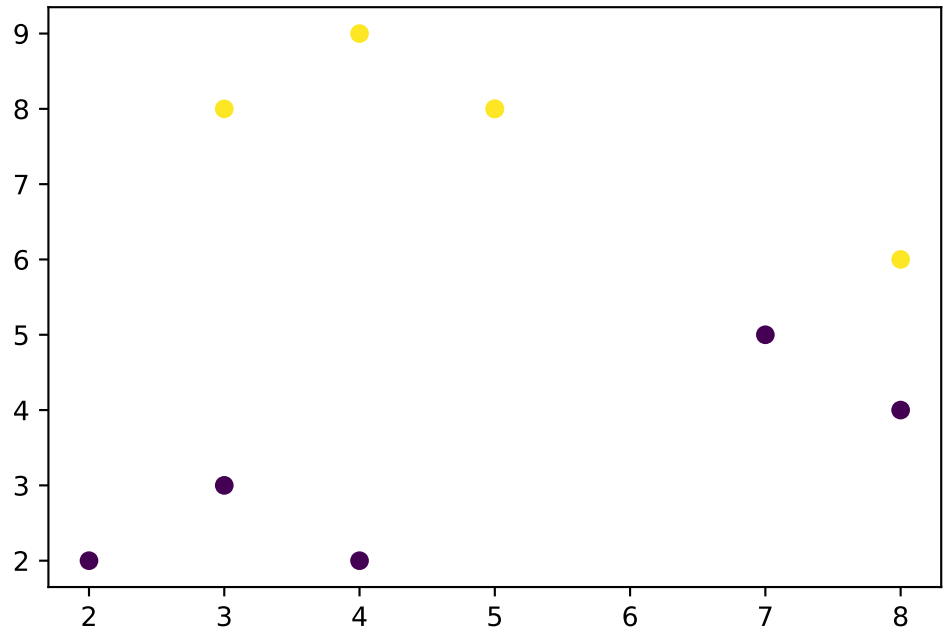
Out[2]: <matplotlib.collections.PathCollection at 0x7f5706a20a90>



In [20]:
```
def k_means(x, k=3):
    clusts = np.random.choice(range(k), size=(x.shape[0]))
    done = False
    while not done:
        cents = []
```

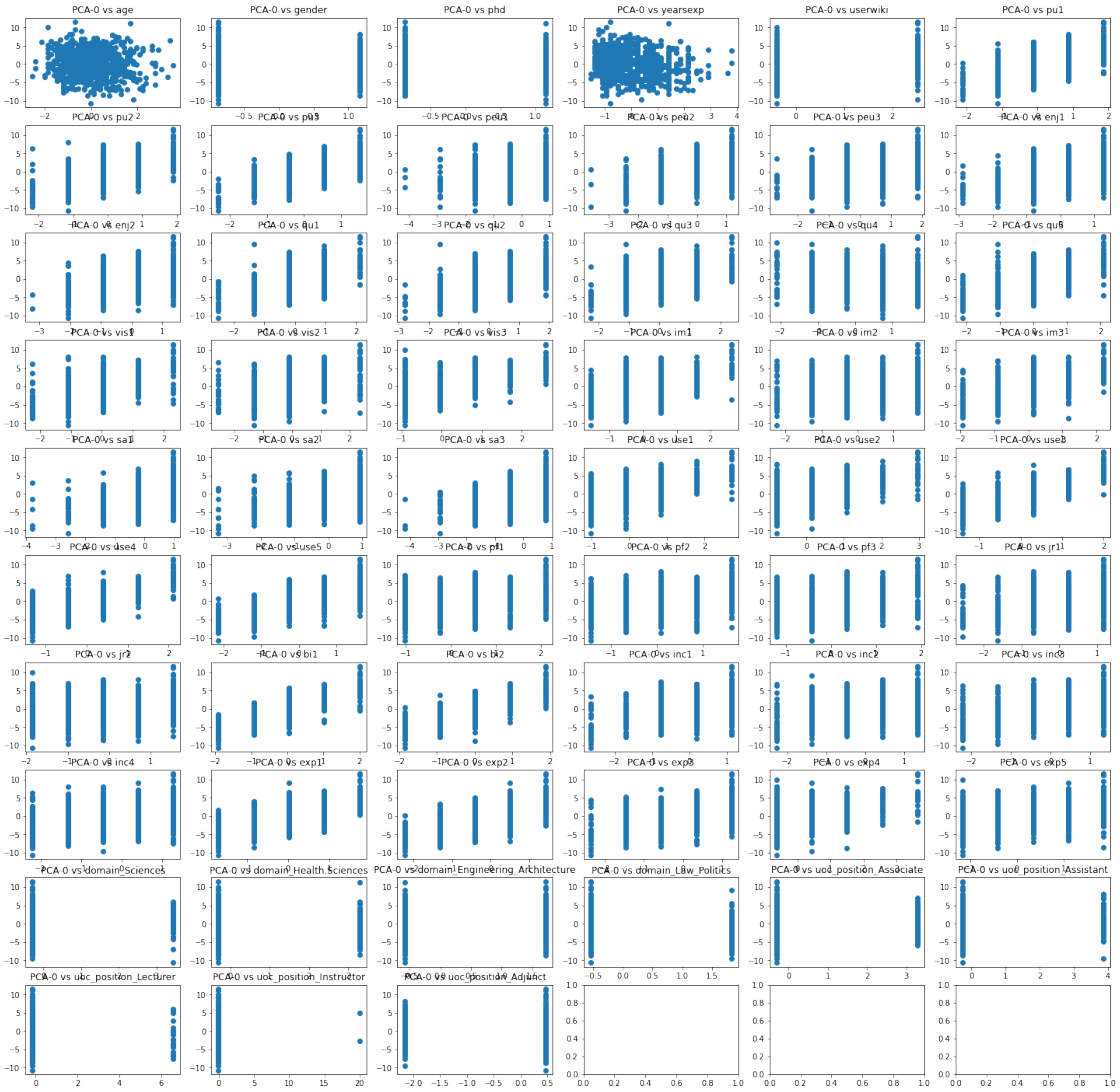
```
        for c in range(k):
            x_c = x[np.where(clusts == c)]
            cents.append(np.mean(x_c, axis=0))
        dists = cdist(x, cents)
        new_clusts = np.argmin(dists, axis=1)
        if (clusts == new_clusts).all():
            done = True
        clusts = new_clusts
    return clusts

plt.scatter(x[:, 0], x[:, 1], c=k_means(x))
plt.show()
plt.scatter(x[:, 0], x[:, 1], c=k_means(x, k=2))
plt.show()
```

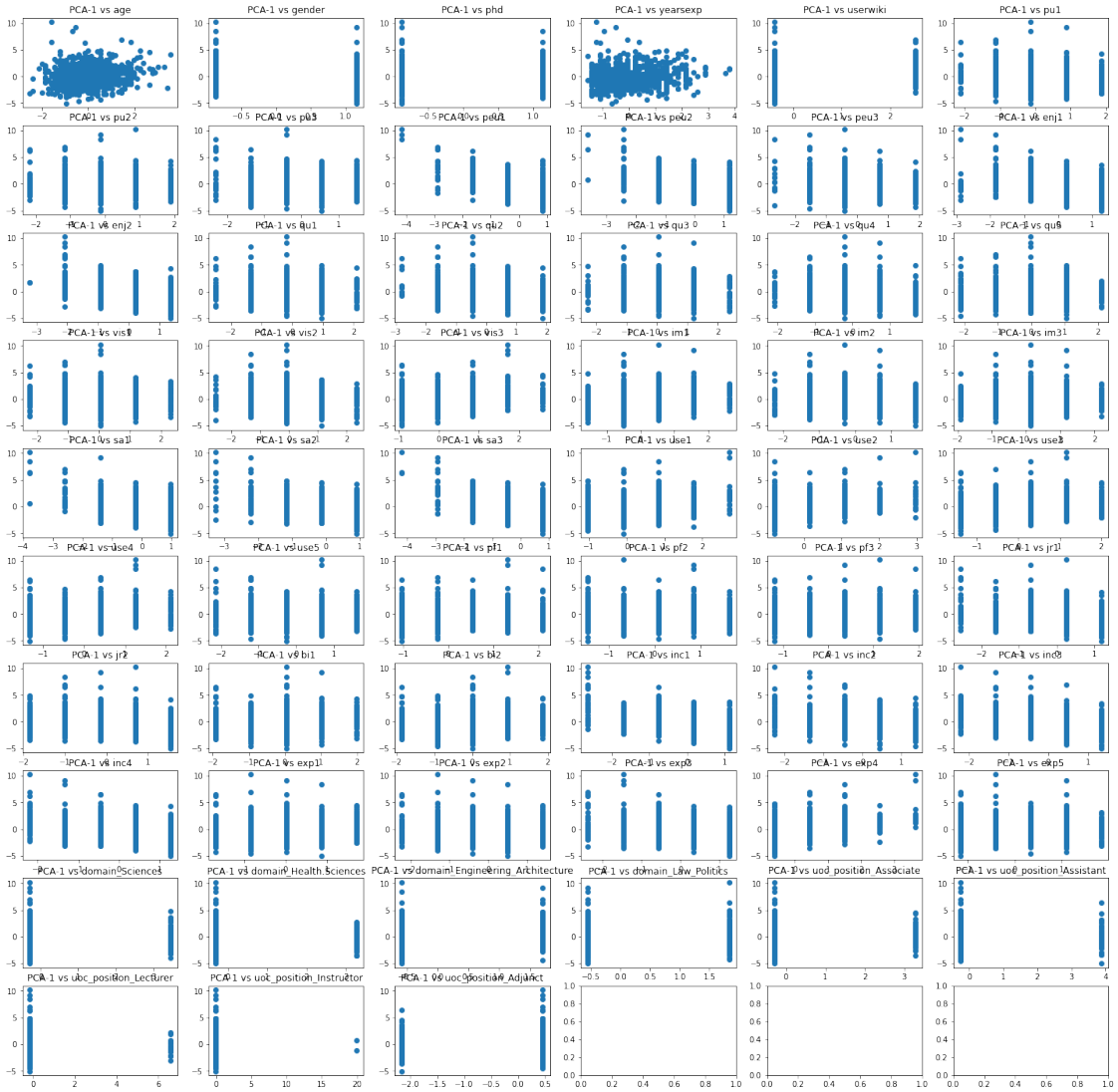




$k = 3$ fits this data best. This makes sense since there are three human-identifiable clusters, and the $k = 2$ case splits one of them in half.

```
In [2]: from sklearn.cluster import KMeans
        from sklearn.decomposition import PCA
        from sklearn.manifold import TSNE
        from sklearn.preprocessing import StandardScaler
        wiki = pd.read_csv('./data/wiki.csv')
        wiki = pd.DataFrame(StandardScaler().fit_transform(wiki), columns=wiki.columns)

In [3]: pca = PCA(n_components=2)
        cats = pca.fit_transform(wiki)
        fig, axs = plt.subplots(10, 6, figsize=(25, 25))
        for ax, col in zip(axs.flat, wiki.columns):
            ax.scatter(wiki[col], cats[:, 0])
            ax.set_title(f'PCA-0 vs {col}')
        plt.show()
        fig, axs = plt.subplots(10, 6, figsize=(25, 25))
        for ax, col in zip(axs.flat, wiki.columns):
            ax.scatter(wiki[col], cats[:, 1])
            ax.set_title(f'PCA-1 vs {col}')
        plt.show()
```

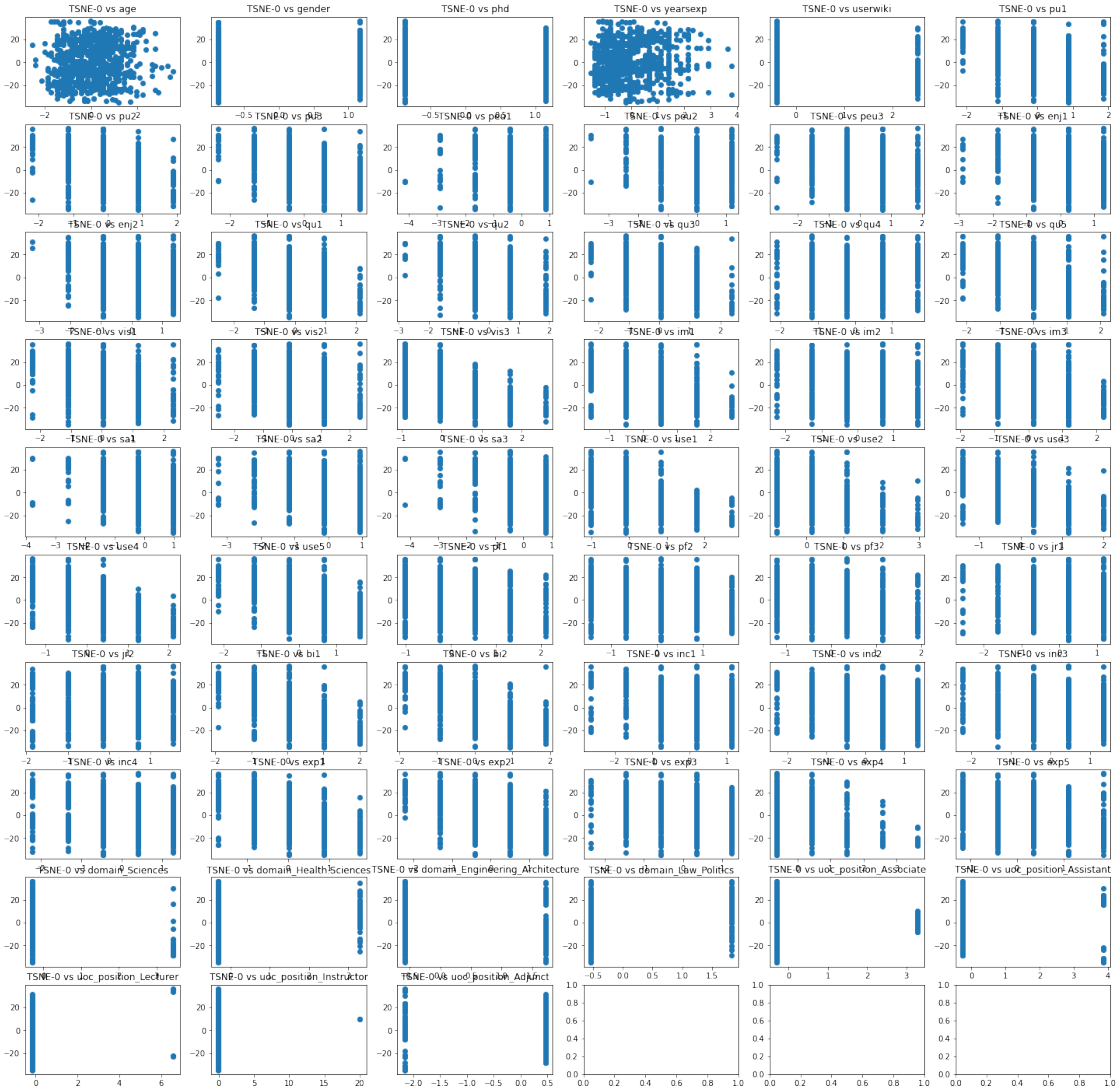

PCA-0 vs age
PCA-0 vs gender
PCA-0 vs phd
PCA-0 vs yearsexp
PCA-0 vs userwiki
PCA-0 vs pu1
PCA-0 vs pu2
PCA-0 vs pu3
PCA-0 vs peu1
PCA-0 vs peu2
PCA-0 vs peu3
PCA-0 vs enj1
PCA-0 vs enj2
PCA-0 vs qu1
PCA-0 vs qu2
PCA-0 vs qu3
PCA-0 vs qu4
PCA-0 vs qu5
PCA-0 vs vis1
PCA-0 vs vis2
PCA-0 vs vis3
PCA-0 vs im1
PCA-0 vs im2
PCA-0 vs im3
PCA-0 vs sa1
PCA-0 vs sa2
PCA-0 vs sa3
PCA-0 vs use1
PCA-0 vs use2
PCA-0 vs use3
PCA-0 vs use4
PCA-0 vs use5
PCA-0 vs pf1
PCA-0 vs pf2
PCA-0 vs pf3
PCA-0 vs jr11
PCA-0 vs jr2
PCA-0 vs bi1
PCA-0 vs bi2
PCA-0 vs inc1
PCA-0 vs inc2
PCA-0 vs inc3
PCA-0 vs inc4
PCA-0 vs exp1
PCA-0 vs exp2
PCA-0 vs exp3
PCA-0 vs exp4
PCA-0 vs exp5
PCA-0 vs domain_Sciences
PCA-0 vs domain_Health_Sciences
PCA-0 vs domain_Engineering_Architecture
PCA-0 vs domain_Law_Politics
PCA-0 vs uoc_position_Associate
PCA-0 vs uoc_position_Assistant
PCA-0 vs uoc_position_Lecturer
PCA-0 vs uoc_position_Instructor
PCA-0 vs uoc_position_Adjunct

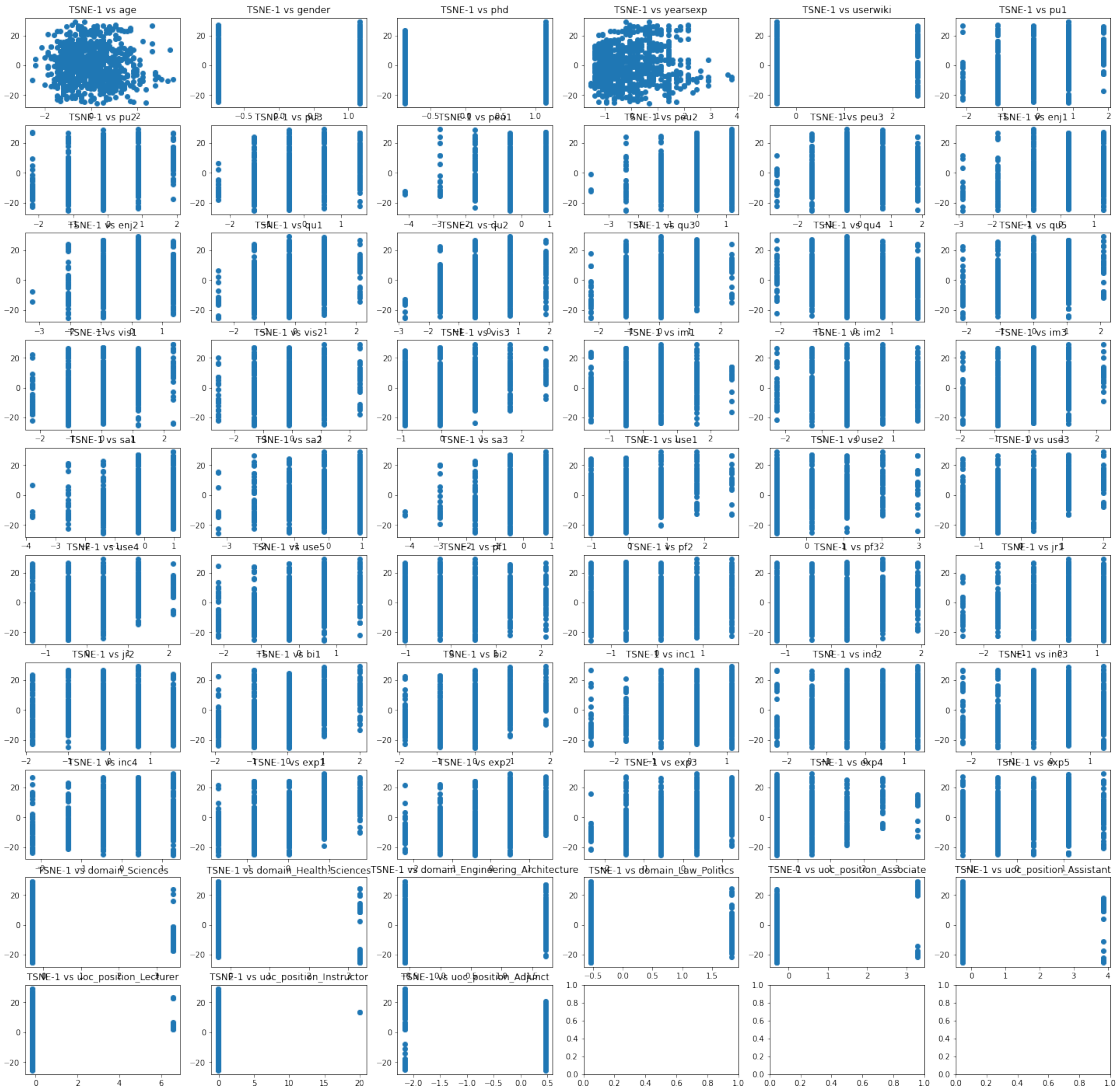


We see that the first component embodies a lot of the things that we would consider to be "response variables". These are things like percieved usefulness and percieved ease of use. The second component is related to a number of things that we would consider to be "input variables" like incentives and experience.

```
In [4]: tsne = TSNE(n_components=2)
        tcats = tsne.fit_transform(wiki)
        fig, axs = plt.subplots(10, 6, figsize=(25, 25))
        for ax, col in zip(axs.flat, wiki.columns):
            ax.scatter(wiki[col], tcats[:, 0])
            ax.set_title(f'TSNE-0 vs {col}')
        plt.show()
        fig, axs = plt.subplots(10, 6, figsize=(25, 25))
        for ax, col in zip(axs.flat, wiki.columns):
            ax.scatter(wiki[col], tcats[:, 1])
```

```
        ax.set_title(f'TSNE-1 vs {col}')
    plt.show()
```

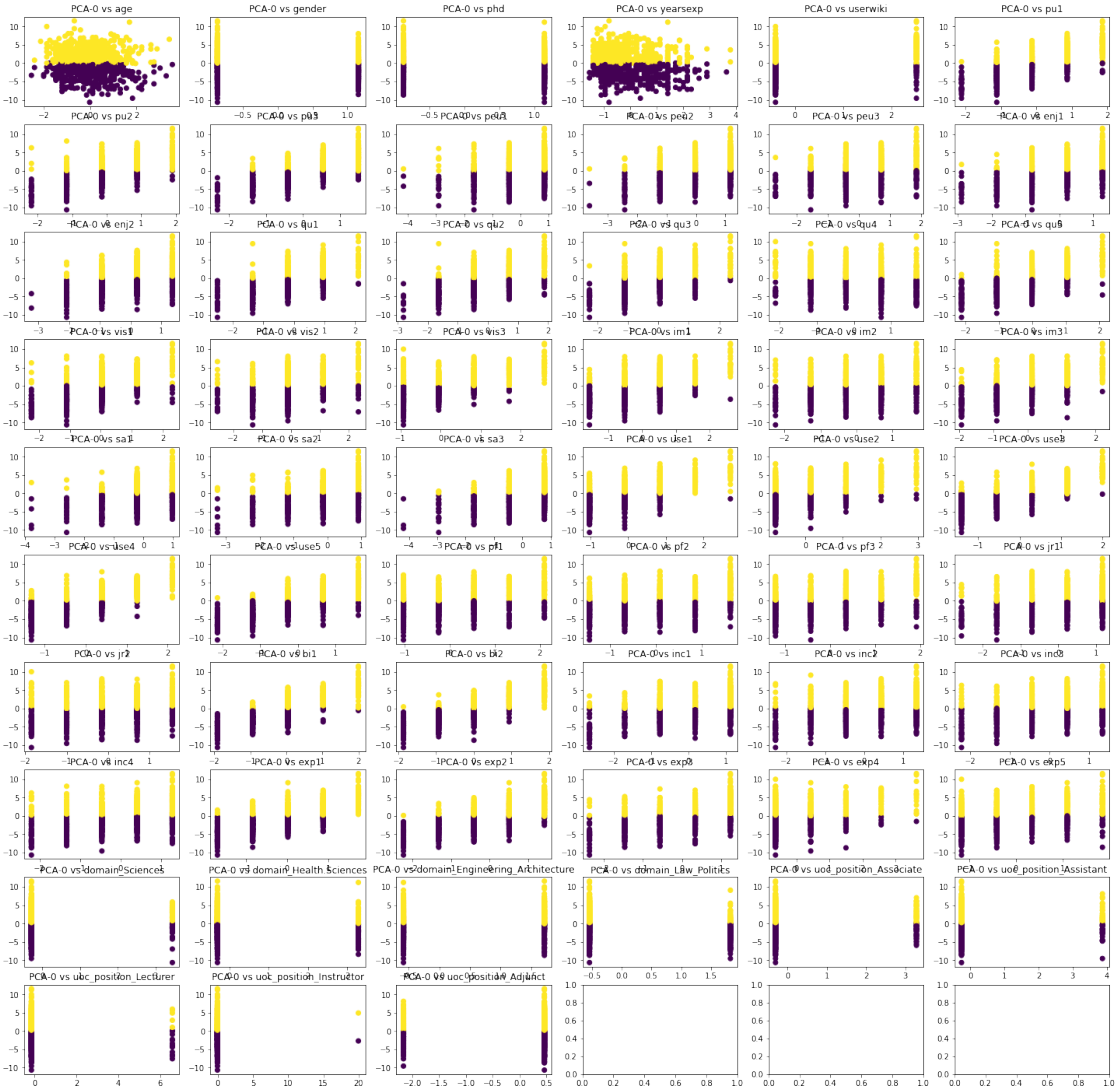




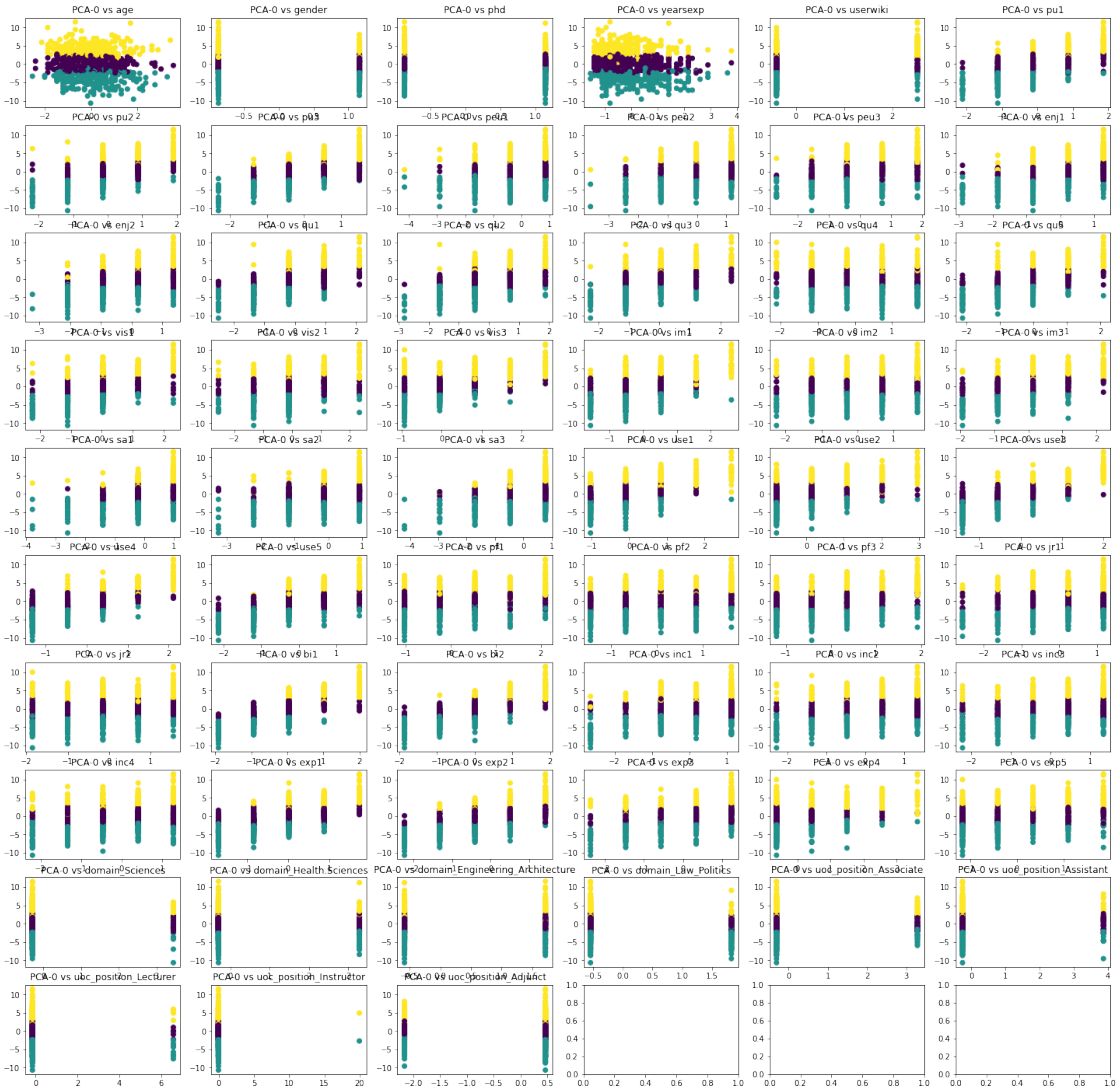
The things that t-SNE latches on to are not as evident from the plots, however it constructs a manifold rather than identifying components, so perhaps that is just hard to see given this visualization.

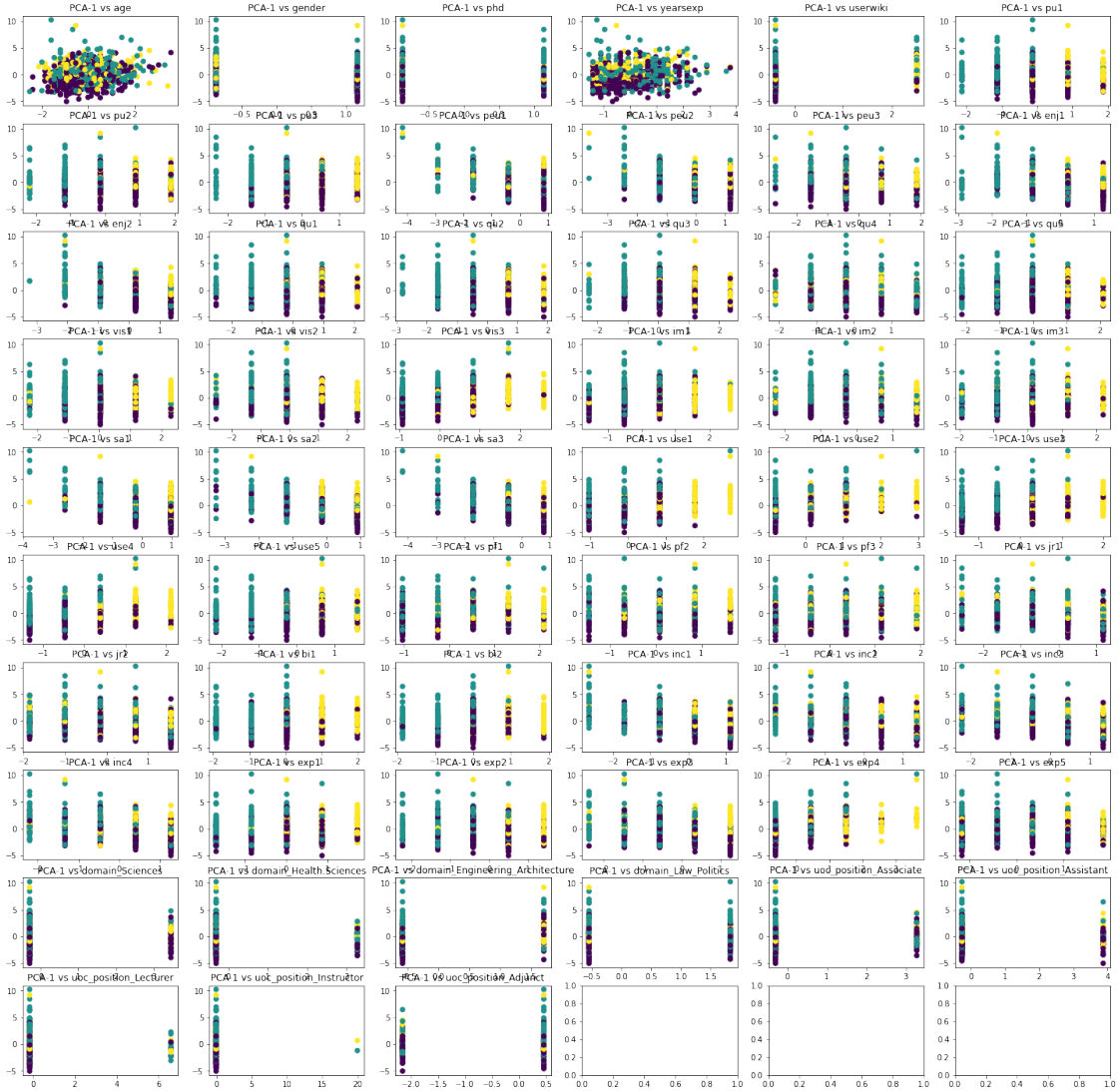
```
In [5]: for k in range(2, 5):
            km = KMeans(n_clusters=k)
            cs = km.fit_predict(wiki)

            fig, axs = plt.subplots(10, 6, figsize=(25, 25))
            for ax, col in zip(axs.flat, wiki.columns):
                ax.scatter(wiki[col], cats[:, 0], c=cs)
                ax.set_title(f'PCA-0 vs {col}')
            plt.show()
            fig, axs = plt.subplots(10, 6, figsize=(25, 25))
            for ax, col in zip(axs.flat, wiki.columns):
```

```
            ax.scatter(wiki[col], cats[:, 1], c=cs)
            ax.set_title(f'PCA-1 vs {col}')
    plt.show()
```

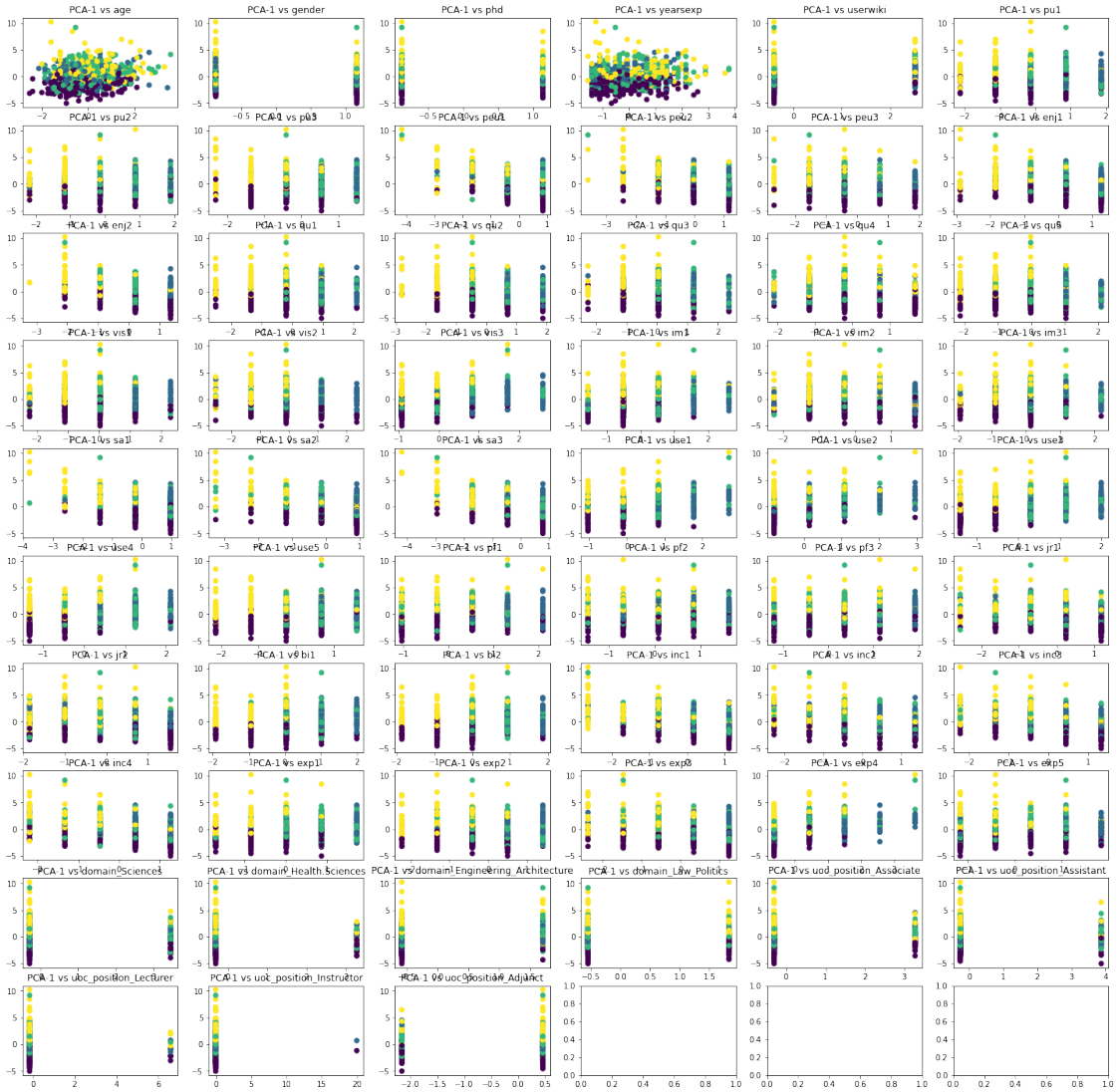



PCA-1 vs age
PCA-1 vs gender
PCA-1 vs phd
PCA-1 vs yearsexp
PCA-1 vs userwiki
PCA-1 vs pu1
PCA-1 vs pu2
PCA-1 vs pu3
PCA-1 vs pu1
PCA-1 vs peu2
PCA-1 vs peu3
PCA-1 vs enj1
PCA-1 vs enj2
PCA-1 vs qu1
PCA-1 vs qu2
PCA-1 vs qu3
PCA-1 vs qu4
PCA-1 vs qu5
PCA-1 vs vis1
PCA-1 vs vis2
PCA-1 vs vis3
PCA-1 vs im1
PCA-1 vs im2
PCA-1 vs im3
PCA-1 vs sa1
PCA-1 vs sa2
PCA-1 vs sa3
PCA-1 vs use1
PCA-1 vs use2
PCA-1 vs use3
PCA-1 vs use4
PCA-1 vs use5
PCA-1 vs pf1
PCA-1 vs pf2
PCA-1 vs pf3
PCA-1 vs jr1
PCA-1 vs jr2
PCA-1 vs bi1
PCA-1 vs bi2
PCA-1 vs inc1
PCA-1 vs inc2
PCA-1 vs inc3
PCA-1 vs inc4
PCA-1 vs exp1
PCA-1 vs exp2
PCA-1 vs exp3
PCA-1 vs exp4
PCA-1 vs exp5
PCA-1 vs domain_Sciences
PCA-1 vs domain_Health_Sciences
PCA-1 vs domain_Engineering_Architecture
PCA-1 vs domain_Law_Politics
PCA-1 vs uoc_position_Associate
PCA-1 vs uoc_position_Assistant
PCA-1 vs uoc_position_Lecturer
PCA-1 vs uoc_position_Instructor
PCA-1 vs uoc_position_Adjunct


PCA-0 vs age
PCA-0 vs gender
PCA-0 vs phd
PCA-0 vs yearsexp
PCA-0 vs userwiki
PCA-0 vs pu1
PCA-0 vs pu2
PCA-0 vs pu3
PCA-0 vs peu1
PCA-0 vs peu2
PCA-0 vs peu3
PCA-0 vs enj1
PCA-0 vs enj2
PCA-0 vs qu1
PCA-0 vs qu2
PCA-0 vs qu3
PCA-0 vs qu4
PCA-0 vs qu5
PCA-0 vs vis1
PCA-0 vs vis2
PCA-0 vs vis3
PCA-0 vs im1
PCA-0 vs im2
PCA-0 vs im3
PCA-0 vs sa1
PCA-0 vs sa2
PCA-0 vs sa3
PCA-0 vs use1
PCA-0 vs use2
PCA-0 vs use3
PCA-0 vs use4
PCA-0 vs use5
PCA-0 vs pf1
PCA-0 vs pf2
PCA-0 vs pf3
PCA-0 vs jr11
PCA-0 vs jr2
PCA-0 vs bi1
PCA-0 vs bi2
PCA-0 vs inc1
PCA-0 vs inc2
PCA-0 vs inc3
PCA-0 vs inc4
PCA-0 vs exp1
PCA-0 vs exp2
PCA-0 vs exp3
PCA-0 vs exp4
PCA-0 vs exp5
PCA-0 vs domain_Sciences
PCA-0 vs domain_Health_Sciences
PCA-0 vs domain_Engineering_Architecture
PCA-0 vs domain_Law_Politics
PCA-0 vs uoc_position_Associate
PCA-0 vs uoc_position_Assistant
PCA-0 vs uoc_position_Lecturer
PCA-0 vs uoc_position_Instructor
PCA-0 vs uoc_position_Adjunct


PCA-1 vs age
PCA-1 vs gender
PCA-1 vs phd
PCA-1 vs yearsexp
PCA-1 vs userwiki
PCA-1 vs pu1
PCA-1 vs pu2
PCA-1 vs pu3
PCA-1 vs pu4
PCA-1 vs peu2
PCA-1 vs peu3
PCA-1 vs enj1
PCA-1 vs enj2
PCA-1 vs qu1
PCA-1 vs qu2
PCA-1 vs qu3
PCA-1 vs qu4
PCA-1 vs qu5
PCA-1 vs vis1
PCA-1 vs vis2
PCA-1 vs vis3
PCA-1 vs im1
PCA-1 vs im2
PCA-1 vs im3
PCA-1 vs sa1
PCA-1 vs sa2
PCA-1 vs sa3
PCA-1 vs use1
PCA-1 vs use2
PCA-1 vs use3
PCA-1 vs use4
PCA-1 vs use5
PCA-1 vs pf1
PCA-1 vs pf2
PCA-1 vs pf3
PCA-1 vs jr1
PCA-1 vs jr2
PCA-1 vs bi1
PCA-1 vs bi2
PCA-1 vs inc1
PCA-1 vs inc2
PCA-1 vs inc3
PCA-1 vs inc4
PCA-1 vs exp1
PCA-1 vs exp2
PCA-1 vs exp3
PCA-1 vs exp4
PCA-1 vs exp5
PCA-1 vs domain_Sciences
PCA-1 vs domain_Health_Sciences
PCA-1 vs domain_Engineering_Architecture
PCA-1 vs domain_Law_Politics
PCA-1 vs uoc_position_Associate
PCA-1 vs uoc_position_Assistant
PCA-1 vs uoc_position_Lecturer
PCA-1 vs uoc_position_Instructor
PCA-1 vs uoc_position_Adjunct

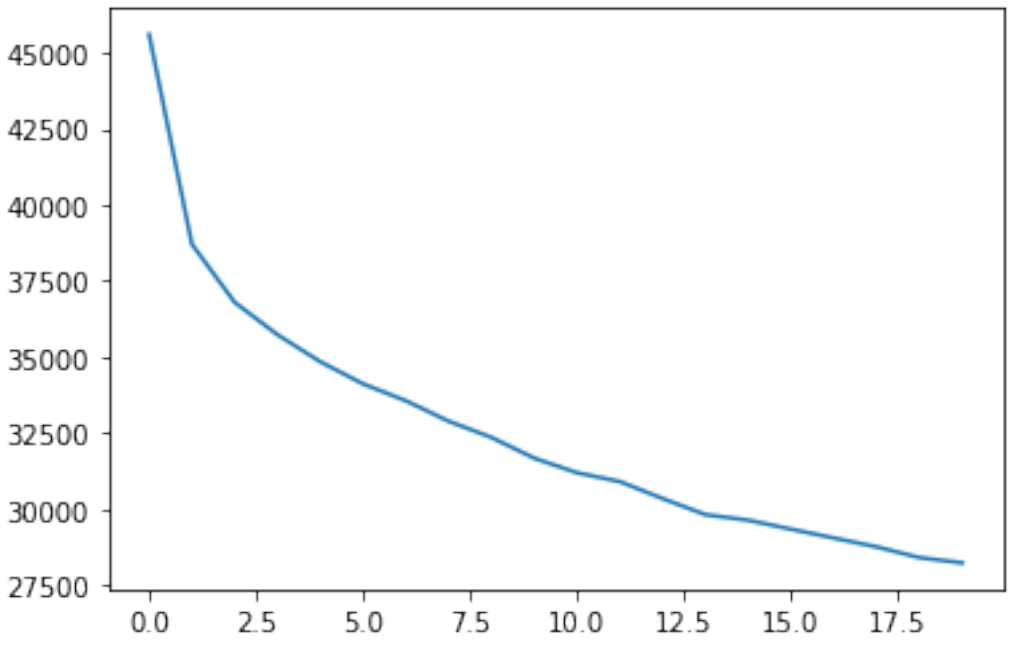

PCA-0 vs age
PCA-0 vs gender
PCA-0 vs phd
PCA-0 vs yearsexp
PCA-0 vs userwiki
PCA-0 vs pu1
PCA-0 vs pu2
PCA-0 vs pu3
PCA-0 vs peu1
PCA-0 vs peu2
PCA-0 vs peu3
PCA-0 vs enj1
PCA-0 vs enj2
PCA-0 vs qu1
PCA-0 vs qu2
PCA-0 vs qu3
PCA-0 vs qu4
PCA-0 vs qu5
PCA-0 vs vis1
PCA-0 vs vis2
PCA-0 vs vis3
PCA-0 vs im1
PCA-0 vs im2
PCA-0 vs im3
PCA-0 vs sa1
PCA-0 vs sa2
PCA-0 vs sa3
PCA-0 vs use1
PCA-0 vs use2
PCA-0 vs use3
PCA-0 vs use4
PCA-0 vs use5
PCA-0 vs pf1
PCA-0 vs pf2
PCA-0 vs pf3
PCA-0 vs jr1
PCA-0 vs jr2
PCA-0 vs bi1
PCA-0 vs bi2
PCA-0 vs inc1
PCA-0 vs inc2
PCA-0 vs inc3
PCA-0 vs inc4
PCA-0 vs exp1
PCA-0 vs exp2
PCA-0 vs exp3
PCA-0 vs exp4
PCA-0 vs exp5
PCA-0 vs domain_Sciences
PCA-0 vs domain_Health_Sciences
PCA-0 vs domain_Engineering_Architecture
PCA-0 vs domain_Law_Politics
PCA-0 vs uoc_position_Associate
PCA-0 vs uoc_position_Assistant
PCA-0 vs uoc_position_Lecturer
PCA-0 vs uoc_position_Instructor
PCA-0 vs uoc_position_Adjunct



The Kmeans algorithm comes to a similar conclusion to the PCA first component, but it's a little more scattered in the second component. It's still somewhat cohesive though, but in a different direction from the PCA second component.

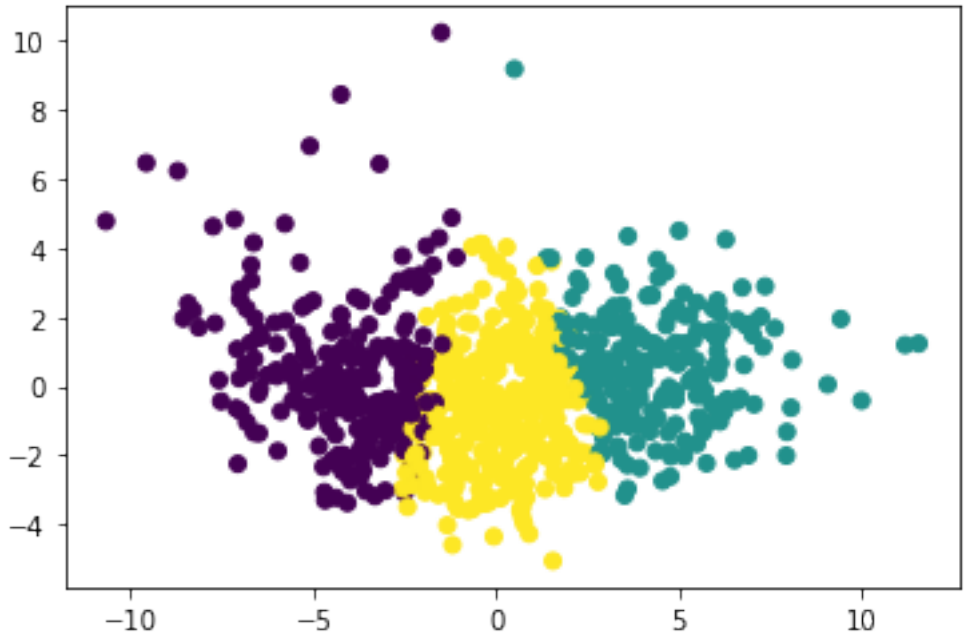
```
In [7]: inerts = []
        for k in range(1, 21):
            km = KMeans(n_clusters=k)
            km.fit(wiki)
            inerts.append(km.inertia_)
        plt.plot(inerts)
```

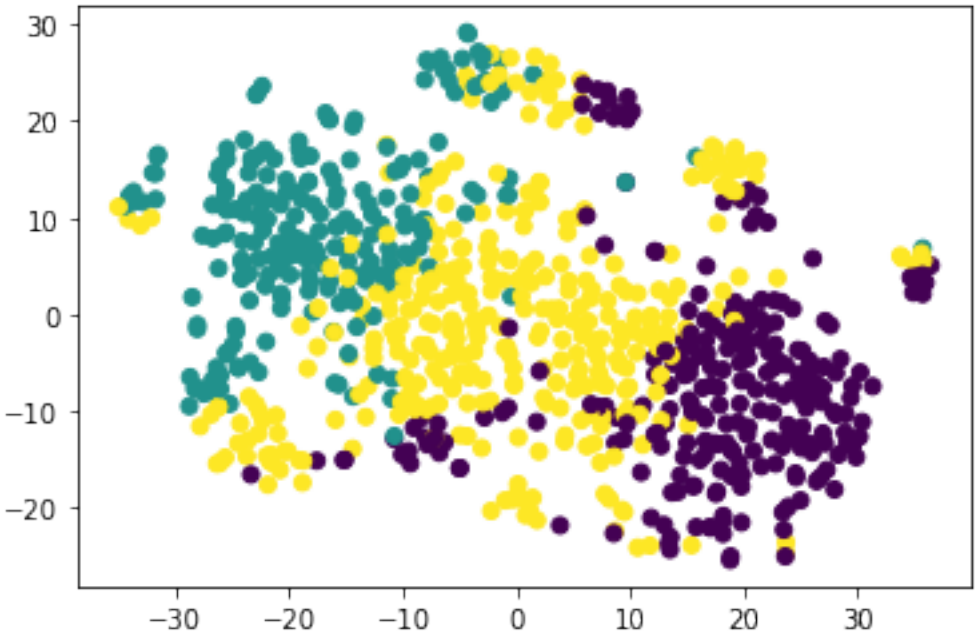
```
Out[7]: [<matplotlib.lines.Line2D at 0x7f243b4365d0>]
```



The elbow is at $k = 3$.

```
In [10]: km = KMeans(n_clusters=3)
         cs = km.fit_predict(wiki)
         plt.figure()
         plt.scatter(cats[:, 0], cats[:, 1], c=cs)
         plt.show()
         plt.figure()
         plt.scatter(tcats[:, 0], tcats[:, 1], c=cs)
         plt.show()
```





We see that the TSNE clustering is a little less coherent than it is in the PCA space, but neither are really distinct. Either way, it's clear that the K-Means solution is really similar to the first component of PCA and fairly similar to the first component of t-SNE.