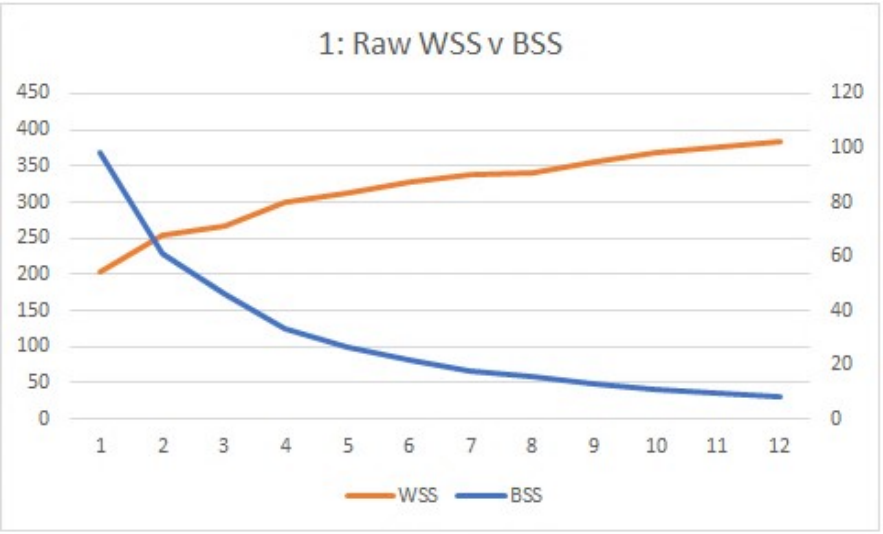
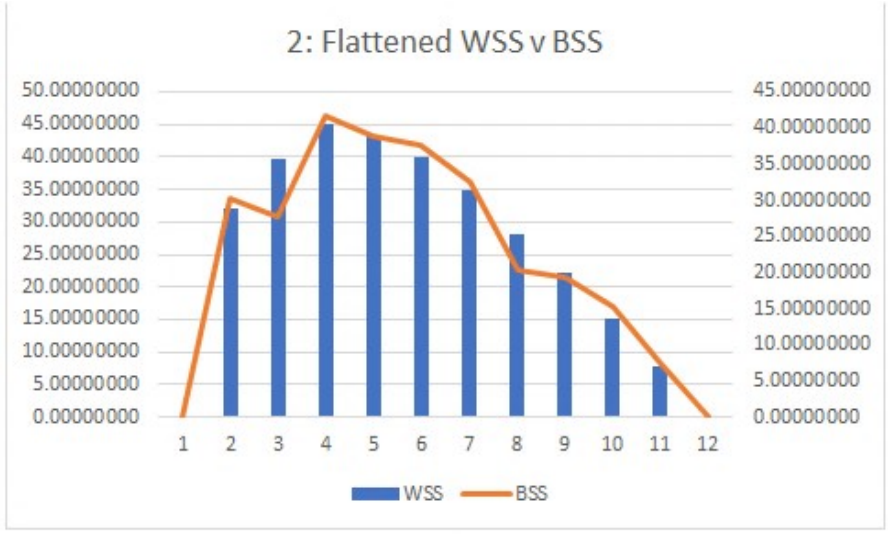
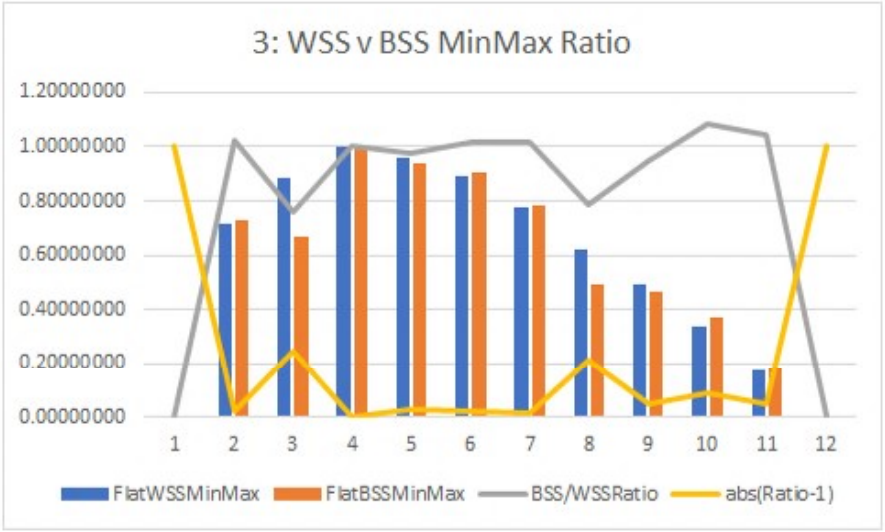






ANOVA (WSS v BSS)
meets
findknee algorithm

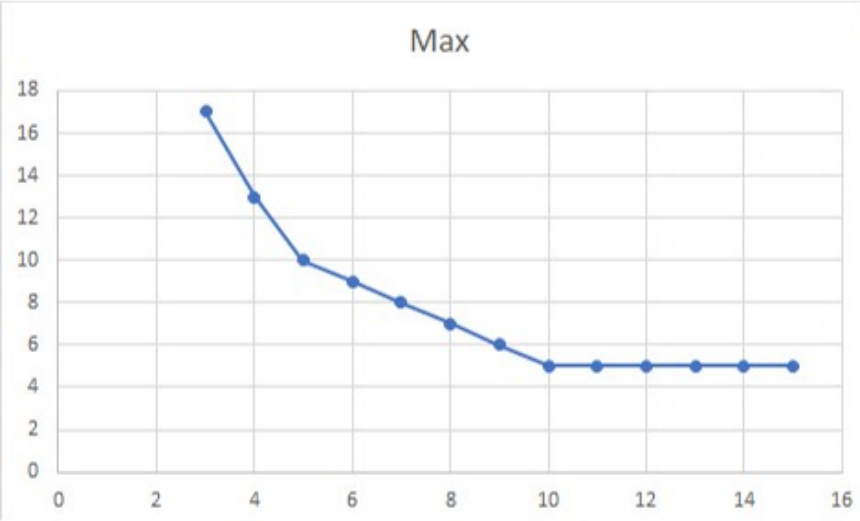
set_ = np.abs(temp_df-1)

optimal_k = np.nanargmin(set_)+mink

Solution is 4th position
(python index starts at
0, so 3 in python's
case)

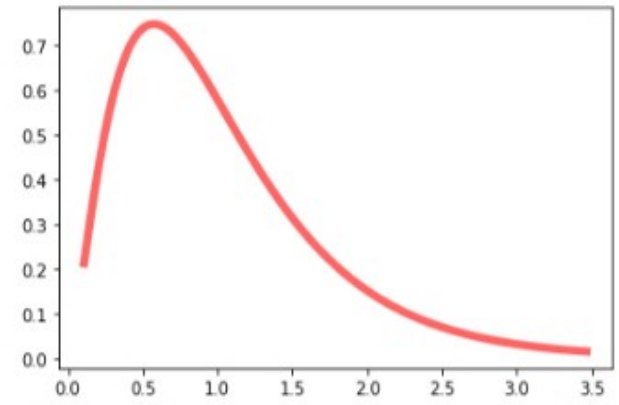
+ mink

mink is 3, so 3+3 = 6
(but excel index starts at 1)




```
dfn = (optimal_k-1)
dfd = len(df)-(optimal_k)

F_scores = (BSS/dfn)/(within_ss/(dfd))

x = np.linspace(f.ppf(0.01, dfn, dfd),
                f.ppf(0.99, dfn, dfd), 100)
plt.plot(x, f.pdf(x, dfn, dfd),
       'r-', lw=5, alpha=0.6, label='f pdf')

print("F-Scores:", F_scores)

print("P-Scores:", 1-f.cdf(F_scores, dfn, dfd))
```

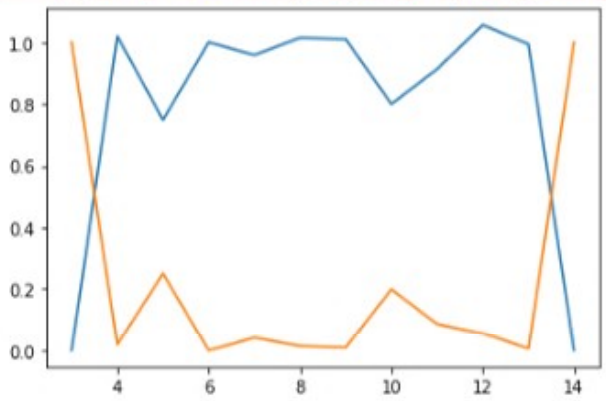
```
F-Scores: [ 78.52330489  66.43137145  60.12298189 152.1336309   81.82322516
  79.09952175]
P-Scores: [1.11022302e-16 1.11022302e-16 1.11022302e-16 1.11022302e-16
 1.11022302e-16 1.11022302e-16]
```



[9]:

```
bss_ = findknee(bss)
bss_ = bss_/np.max(bss_)

wss_ = findknee(np.array(pd.DataFrame(wss).mean(1)))
wss_ = wss_/np.max(wss_)

temp_df = pd.DataFrame(bss_/wss_).replace([np.inf, -np.i
temp_df.index = np.array(range(0,len(wss_)))+mink
plt.plot(temp_df)

set_ = np.abs(temp_df-1)
plt.plot(set_)
#plt.plot(set_)
optimal_k = np.nanargmin(set_)+mink

print(optimal_k)
```

```
6
C:\Users\User\AppData\Local\Temp/ipykernel_10152/4072231
  temp_df = pd.DataFrame(bss_/wss_).replace([np.inf, -np
```




```
clf = KMeansConstrained(n_clusters=optimal_k, size_min=i
clf.fit_predict(X_pca)

labels = clf.labels_
clusters = clf.n_clusters
centers = clf.cluster_centers_

tot_ss, BSS, within_ss = deriveANOVA(clf, X_pca)
```

```
wss 200.1585375370103
tot_ss 500.00000000000006
bss 299.8414624629898
```

```
tss, bss, wss = findOptimalK_ANOVA(X_pca)
```

```
3
wss 295.33541317305645
tot_ss 500.00000000000006
bss 204.6645868269436
4
wss 245.29889619365105
tot_ss 500.00000000000006
bss 254.7011038063491
5
wss 231.80753075697675
tot_ss 500.00000000000006
bss 268.1924692430233
6
wss 200.15853753701026
tot_ss 500.00000000000006
bss 299.8414624629898
7
wss 186.7875232985095
tot_ss 500.00000000000006
bss 313.2124767014906
8
wss 171.9166763698156
tot_ss 500.00000000000006
bss 328.08332363018457
9
wss 161.1570544501567
tot_ss 500.00000000000006
bss 338.8429455498434
10
wss 158.4622626120873
tot_ss 500.00000000000006
bss 341.5377373879128
11
wss 143.324547683454
tot_ss 500.00000000000006
bss 356.67545231654606
12
wss 131.4983010746918
tot_ss 500.00000000000006
bss 368.50169892530835
13
wss 123.91356506793547
tot_ss 500.00000000000006
bss 376.0864349320646
14
wss 115.88086086487225
tot_ss 500.00000000000006
bss 384.11913913512785
```