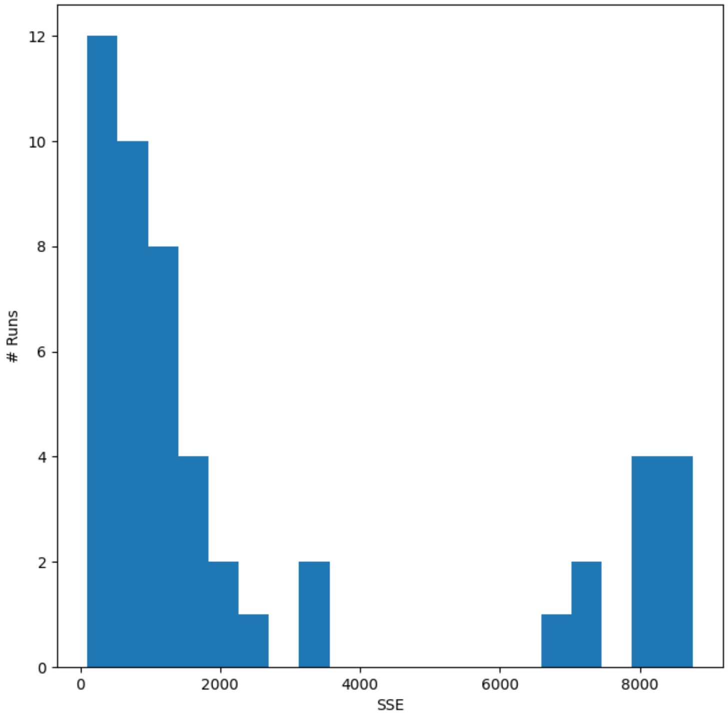
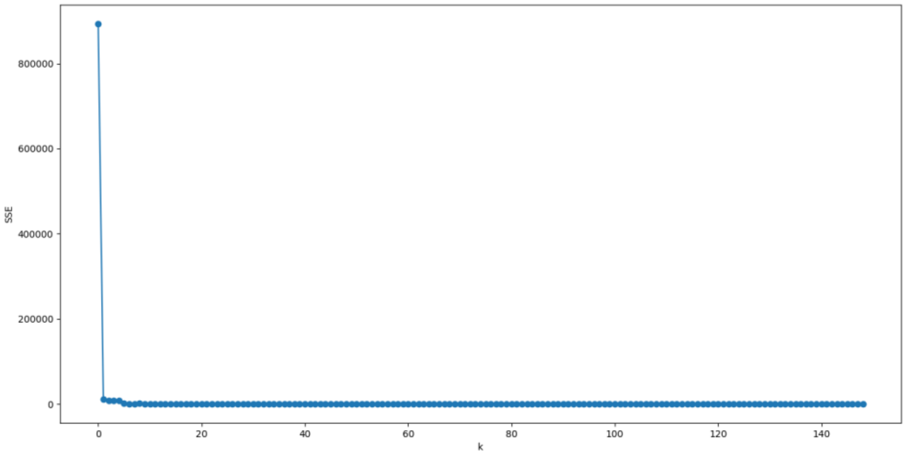
Q5:

k means when implemented the way I did has a low SSE for most of the runs but on a few has a very high SSE. There is also a large gap in between the two clusters of lines, meaning that when there is an error there is a large error. Therefore, if I was applying this to a real dataset, I would run it multiple times, because every now and then there is an outlier run that ruins the efficiency of the algorithm.



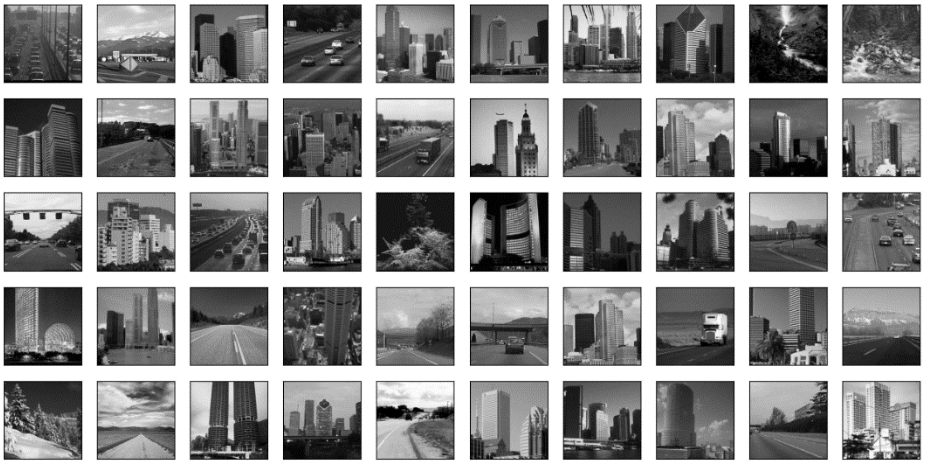
Q6:

It doesn't make sense because k doesn't seem to matter at all. The only time it spikes up is when k = 0 which will never be an option. I would say if k > 5 the curve for sse seems flat and low, so it doesn't really matter.

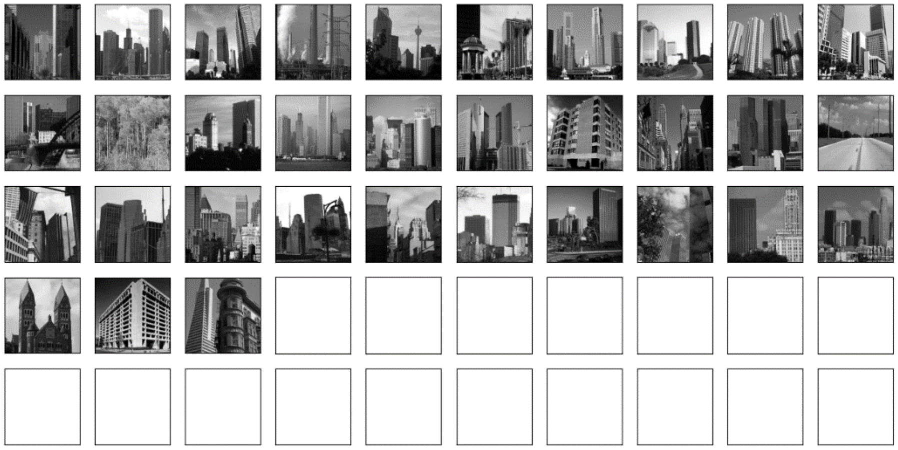


Q7:

a) The images in the dataset seem to be images that have to do with cities. Some I saw were tall skyscrapers and some were freeways with cars. There are also some trees and I'm pretty sure I saw a cute little panda waving. k =10 does a pretty good job, however it is not perfect, so I would say it is a little too low.
b) K = 10 – decent but there are errors



K = 15 – pretty good but still a few errors



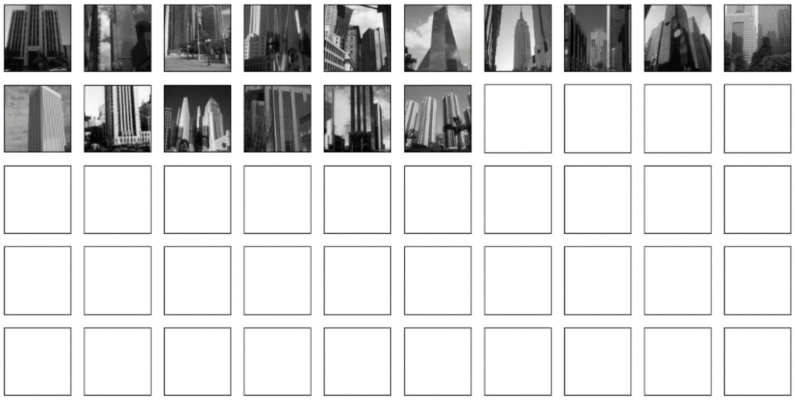
K = 30

Best so far

however it does

not produce many

images. I would



say somewhere between 20 – 30 would be perfect based off my examples.

c) K = 10 SSE = 4789 | K = 15 SSE = 2028 | K = 30 SSE = 880
I would say this is a pretty good indicator of clustering quality.

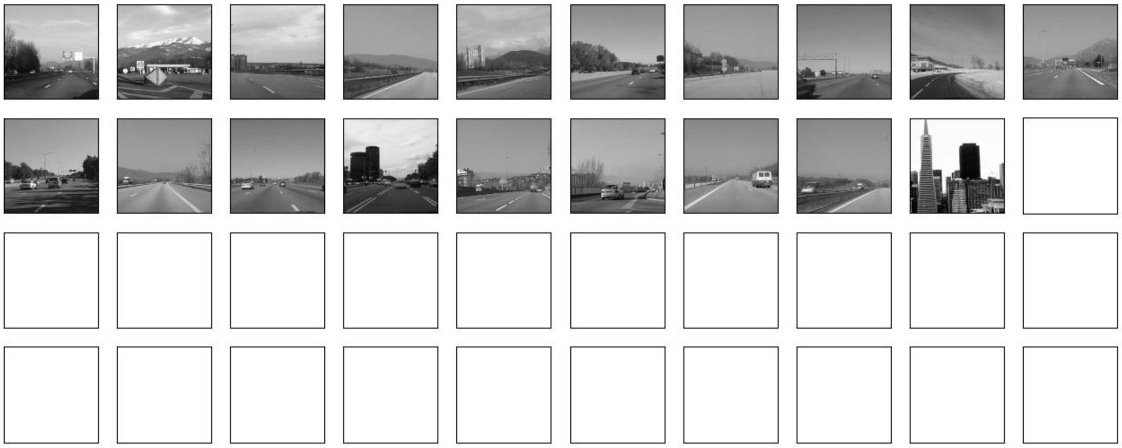
Q8:

From Q7: when k = 10 skyscrapers (31/50)

when k = 15 skyscrapers (31/33)

when k = 30 skyscrapers (16/16)

Others:



Streets (18/19)

-I hope this is what you wanted from this question :)

DEBRIEF:

1) At least 20
2) Difficult
3) Alone
4) 75%
5) Nope, just very difficult and I wasn't expecting that. Cute panda too.