Rahul Chamarthi

CS373

11/13/19

HW4

1. THEORY (10 points)

1.1. (2 points) Limitations of K-Means: **K-means is a great algorithm but certainly has its drawbacks. In particular, k-means is extremely sensitive to outliers, and tends to be sensitive to complex data patterns. An example of the aforementioned insensitivity to complex patterns can be seen by the circular clusters that form as a result of the k-means algorithm. It's a disadvantage because it clearly may miss data that does not fit into this specific pattern. In terms of its sensitivity to outliers, this can be explained by the k-means algorithm itself. Everytime a point is encountered, it is added to the nearest cluster. Once added to the nearest cluster, the centroid for the cluster is recalculated. If the outlier is significant this can have a huge impact on our predictions.**

1.2. (2 points) Discovering data structures: **Most of the time k-means will generate clusters. Whether they are meaningful for the data set is not guaranteed. An easy counterexample is non spherical data. This type of data would require a transformation of the coordinates to polar coordinates to produce a proper result. Thus, the first time k-means is applied a result will be produced, however, the result is not a true reflection of the dataset.**

1.3. (2 points) theoretical time complexity of the algorithm?:

- **Time Complexity:** O(dn)
- **Key:**
  - **n = # of training examples**
  - **d = # of attributes n contains**
- **Possible Speedup:**
  - **We could parallelize the algorithm by selecting data points and clusters in parallel.**

1.4. (2 points) Consider dataset X with p discrete, q continuous attributes and 1 binary class label: **Approach A utilizes solely NBC on the p discrete features, the end result, as described in the handout, would be a 0/1 loss associated with X. Approach B is far more complex. We are utilizing NBC on discrete features in**

the set and k-means on the continuous features in the set. I am comfortable in stating that Approach B will produce the more accurate result. The primary reason for Approach B's superiority stems from the fact that more information is available to us in making our prediction. At the end of the day Approach A tries to predict the class label using only a subset of the features. Approach B, cleverly uses k-means and NBC to use all of the features to make a class label prediction.

1.5. (2 points) Improve the scoring function: **To have a metric that gauges not only within cluster distance, but also cluster to cluster distance. Within clusters distance takes the difference between each point and the centroid it belongs to. I would propose that we also take the difference between each cluster as well. To implement this we would simply pick two clusters with minimal distance in the first step. In the second step we would fuse them together. We will start with n clusters and work our way down to k clusters. n is the number of datapoints in the dataset. In this way we not only take into account the compactness of the clusters, but the distance between the clusters as well.**

2. CODE (15 points)

2.1. SEE CODE

3. ANALYSIS (30 points)

3.1. **(5 points)** Cluster the Yelp data using k-means

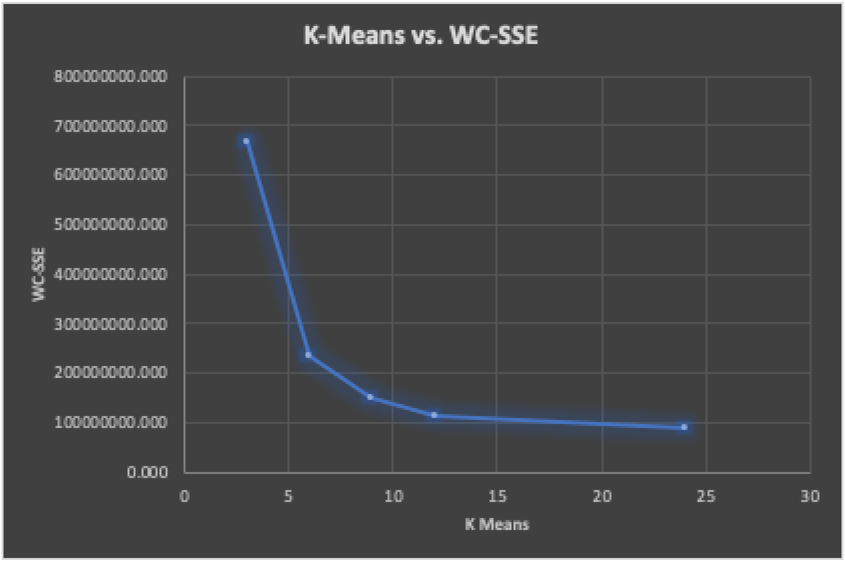
a. Use a random set of examples as the initial centroids.

b. Use values of K = [3,6,9,12,24].

c. Plot the within-cluster sum of squares (wc) as a function of K.

d. Choose an appropriate K from the plot and argue why you chose this particular K.

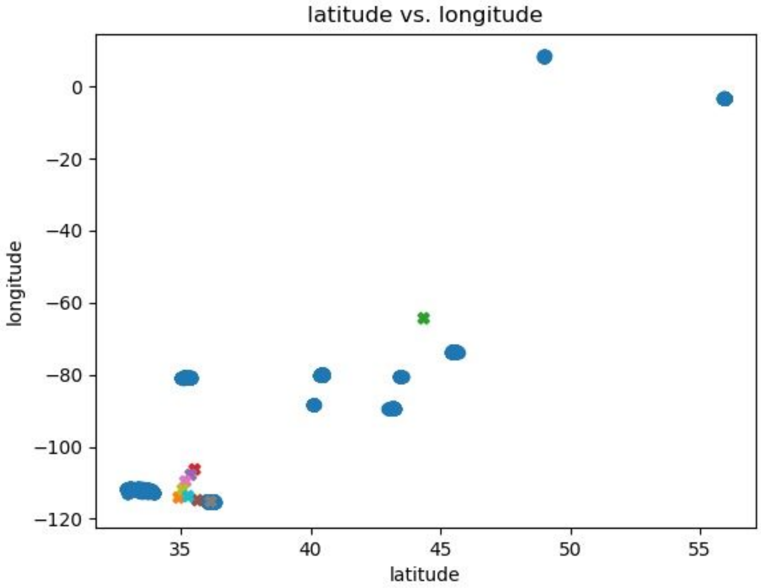
i. **GRAPH:**

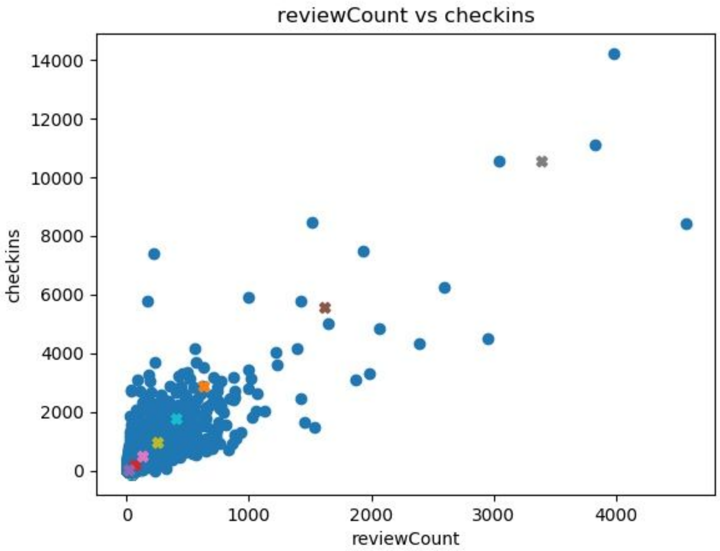


ii. **DISCUSSION: We choose the value k based on the concept of elbow point. The elbow point is the point in the graph where we can see the accuracy gain, we achieve by increasing k, start to slow. We can see the aforementioned trend in the slope between the points. Before the elbow point we can justify each incremental increase in k. We can state that each incremental gain in k leads to significantly better accuracy. Once we reach the elbow point the runtime cost starts to outweigh the benefit of another increase in k. In this graph I would say that the elbow point exists at: k = 9. An argument could be made for k = 6 if one was feeling a little risky. At k = 6 it looks like there is still a significant gain to be found by moving to k = 9.**

e. For the chosen value of K, plot the clusters with their centroids in two ways: first using latitude vs. longitude and second using reviewCount, checkins. Discuss whether any patterns are visible.

   i. **GRAPH:**

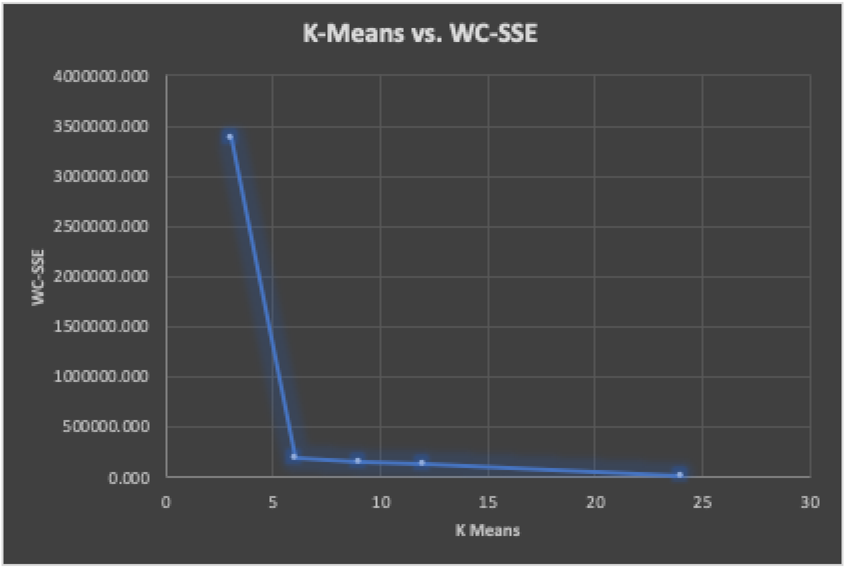




   ii. **DISCUSSION: In the latitude - longitude plot the centroids seemed to be isolated into the left hand corner of the plot. This skewness does not bode well considering the lack of centroids assigned to the farther out values. In the reviewCount - checkins scatterplot the centroids seem to be spread out in a more linear fashion. I am still concerned about the** distribution of the centroids across the data. However, my concern is eased a little bit by the concentration of data, around the concentration of clusters.

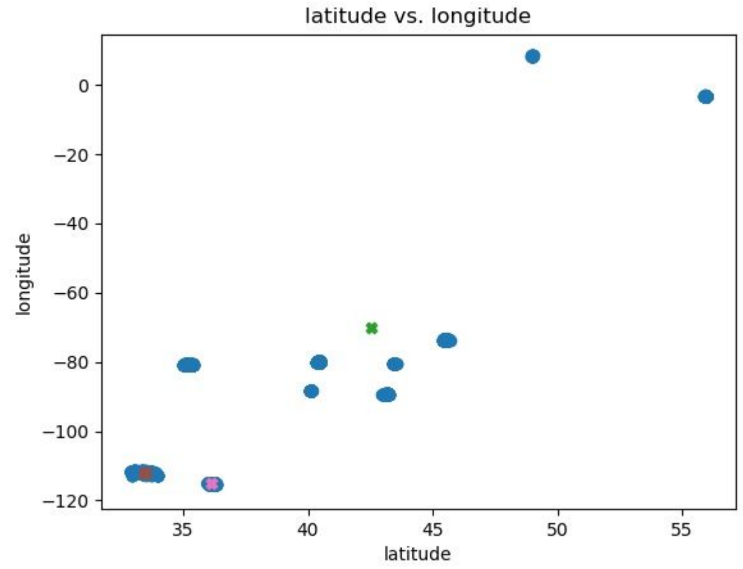
3.2. **(5 points)** Do a log transform of reviewCount, checkins. Describe how you expect the transformation to change the clustering results. Then repeat the analysis (1). Discuss any differences in the results.

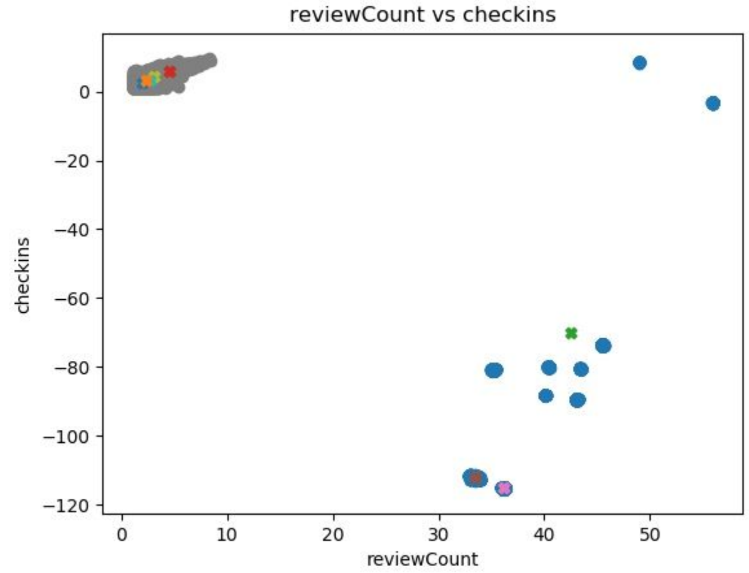
- **PRE-RUN EXPECTATIONS:**
  - I think that latitude and longitude's respective clusters will stay the same because of a lack of transformation. I expect the bigger impact will be on reviewCount, and checkins. The aforementioned attributes being the ones that we actually applied transformations to.
- **ANALYSIS (1):**
  - **WC as a function of K plot:**
    - **GRAPH:**



    - **INTERPRETATION:** I chose k = 6 because it clearly demonstrated that it was the elbow point. We could see that after k = 6, the Information Gained by increasing k dropped off.

- **Chosen K Cluster Plots:**
  - **GRAPH:**



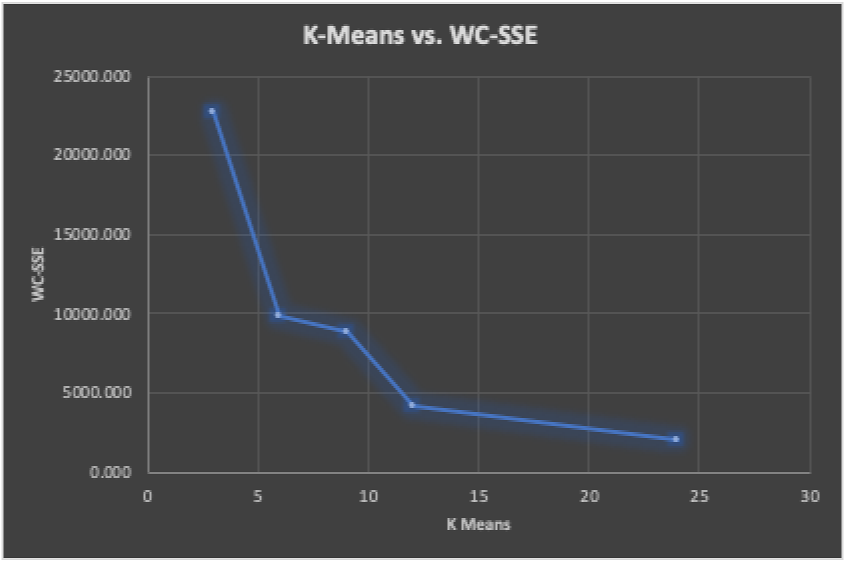


  - **INTERPRETATION:** **We can clearly see a few interesting and promising trends from the scatterplots. In the reviewCount and checkins plot we can clearly see that the centroids are much more evenly distributed. The more encouraging trend is that we now have one cluster per each “group” of data points. In the transformed latitude and longitude plots we can see a similar trend as** described in the reviewCount and check ins plot. The centroids are evenly distributed, and now also identifying the outlying data groups as well.

- **POST-RUN ANALYSIS:**
    - Largely speaking the results of the K Cluster Plots and WC vs K plot lined up with my PRE-RUN analysis. I did think it was cool to see how significant the transformation of the impact was visually.

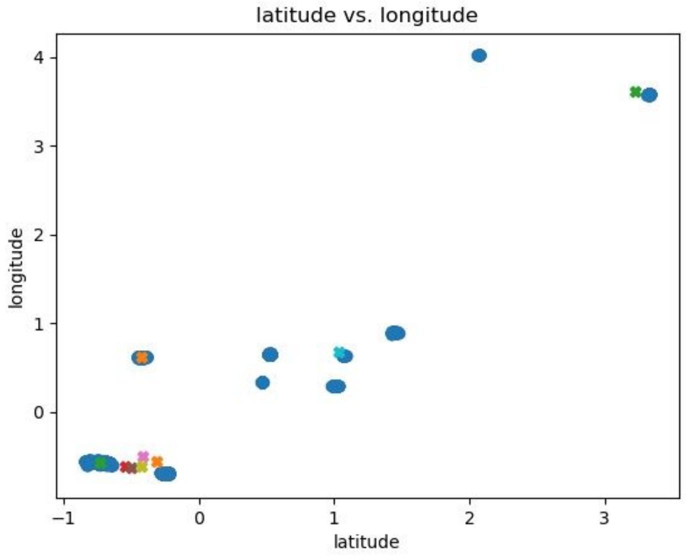
3.3. **(5 points)** Transform the four original attributes so that each attribute has mean = 0 and stdev = 1. You can do this with the numpy functions, numpy.mean() and numpy.std() (i.e., subtract mean, divide by stdev). Describe how you expect the transformation to change the clustering results. Then repeat the analysis (1). Discuss any differences in the results.

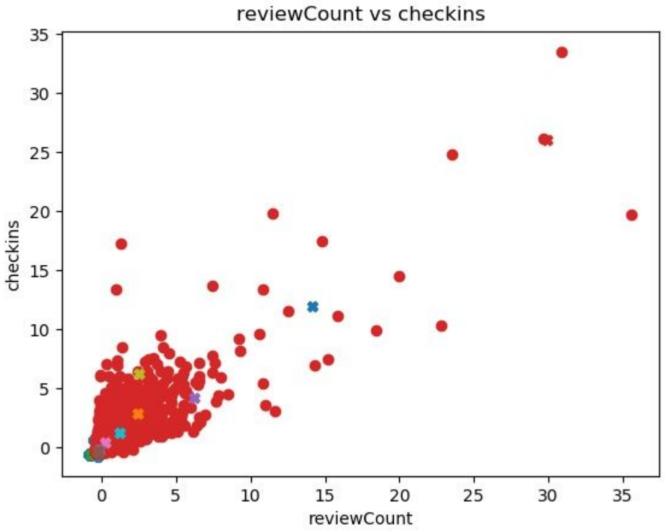
- **PRE-RUN EXPECTATIONS:** I would say that my expectation for the result of transforming all four attributes will be along the lines of the resulting transformation from part 2. In part 2 we simply took the log of only two of the attributes. In the graphs I would expect to see spread out clusters, each of which I expect to be close to the mean.
- **ANALYSIS(1):**
    - **WC as function of K plot:**
        - **GRAPH:**



        - **INTERPRETATION:** I chose 12 because up until 12 there is a pronounced benefit to increasing the K value. We can see, however, that at 12 this drops off. The runtime hit going to 24 is significant compared to the minimal increase in accuracy we will gain.

- **Chosen K Cluster Plots:**
  - **GRAPH:**



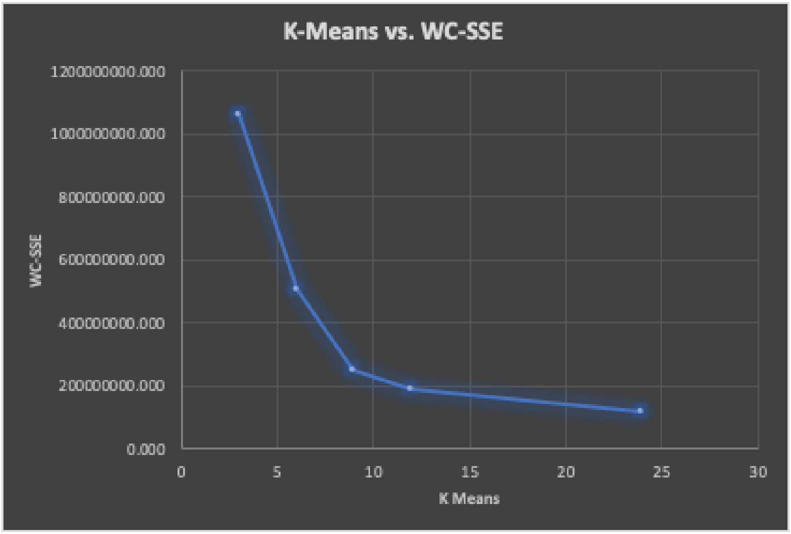


  - **INTERPRETATION:** The latitude and longitude plot roughly reflects the same behavior in question 2. That is to say that we capture and cover most of the data with our centroids. The reviewCount - checkins scatter plot is a little bit more wild. The plot on first glance keeps a largely linear distribution, however, we can see that the centroids are not very evenly distributed. Specifically, there is only one centroid that seems to be catching most of the outlying data points.

- **POST-RUN ANALYSIS:**
  - I would say that I was pretty surprised with this run. 2 of the attributes (the same ones transformed in question 2) demonstrated the same behavior as expected. reviewCount and checkins, however, did not demonstrate such behavior.

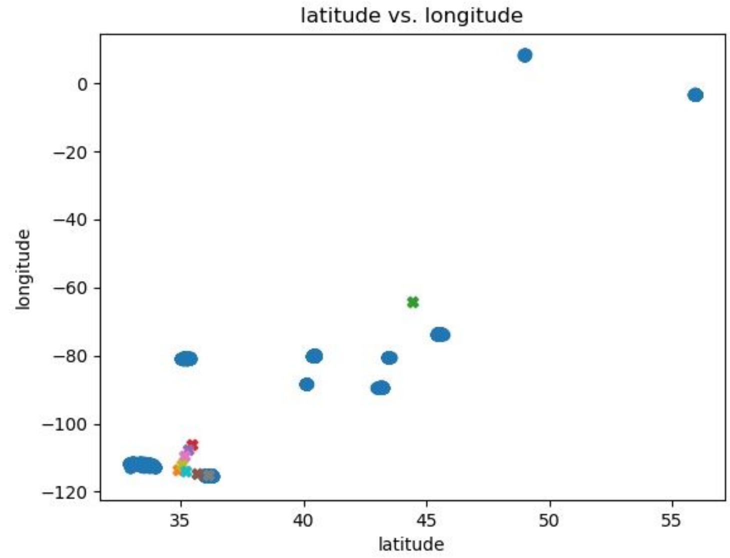
3.4. **(5 points)** Use Manhattan distance instead of Euclidean distance in the algorithm. Describe how you expect the change in the clustering results. Then repeat the analysis (1). Discuss any differences in the results.

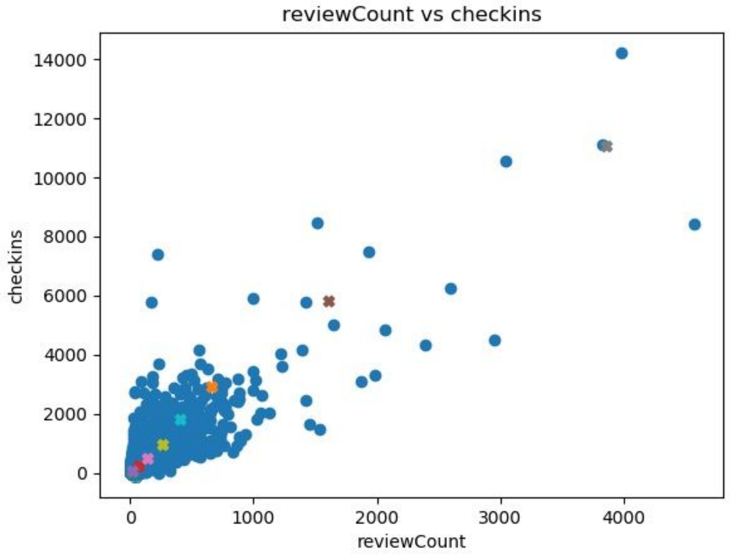
- **PRE-RUN EXPECTATIONS:** I would say that I wouldn't expect too much of a change. I think obviously the scaling of our plots will change but I expect this may be the extent of the change. Both Manhattan and Euclidean are often used for heuristics, with roughly interchangeable effects.
- **ANALYSIS(1):**
  - **WC as function of K plot:**
    - **GRAPH:**



    - **INTERPRETATION:** I chose K = 9 as the K for this graph. The next K forward (12) does not seem to have enough of an information gain to justify increasing to it.

- **Chosen K Cluster plots:**
  - **GRAPH:**





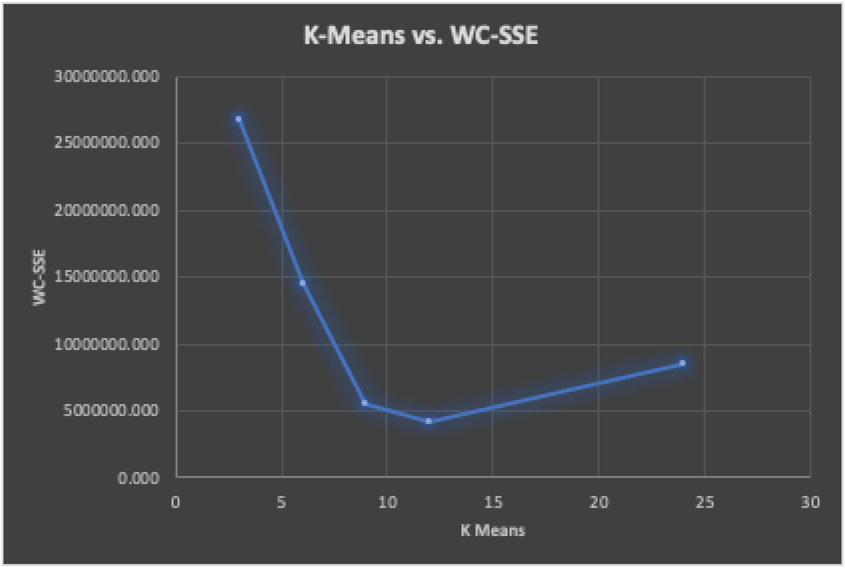
  - **INTERPRETATION:** In the latitude - longitude graph the points stayed largely the same as Euclidean distance. The only noticeable difference was with reviewCount - checkins graph. The centroids were a little bit more spread out for this graph vs. the one utilizing Euclidean distance.

- **POST-RUN ANALYSIS:**
    - **I picked the same K point in both the Euclidean and Manhattan versions of my implementation. The scatter plots did not have much variance either. Sure, there was some variance in the reviewCount - checkins graphs, but not enough to say they have significant differences. So my initial prediction held true.**

3.5. **(5 points)** Take a 6% random sample of the data. Describe how you expect the downsampling to change the clustering results. Then run the analysis (i) five times and report the average performance. Especially, you should use a single random 6% sample of the data. Then run 5 trials where you start k-means from different random choices of the initial centroids. Report the average wc when you plot wc vs. K. For your chosen K, determine which trial had performance closest to the reported average. Plot the centroids from that trial. Discuss any differences in the results and comment on the variability you observe.

- **PRE-RUN EXPECTATIONS:**
    - **Taking a 6% random sample of the data I believe would have to lead to the desirability of having a fewer number of clusters. If we have too many clusters with too few data points we may see several meaningless clusters. Meaningless clusters being those that get assigned a few, or simply no data points. To conclude, I expect that once we hit the knee point we will start to see the WC - SSE begin to increase as K continues to increase.**
- **GRAPHS:**

- **AVERAGE OF 6% SAMPLE (5 TRIALS):**



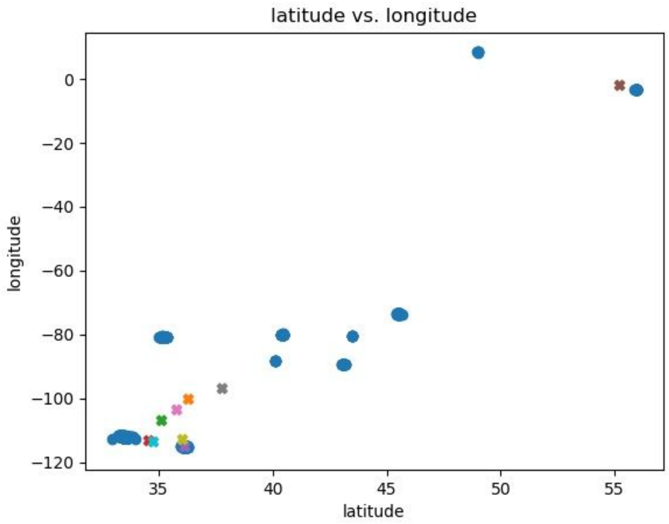
- **AVERAGE WC - SSE FOR ALL TRIALS:**
  -

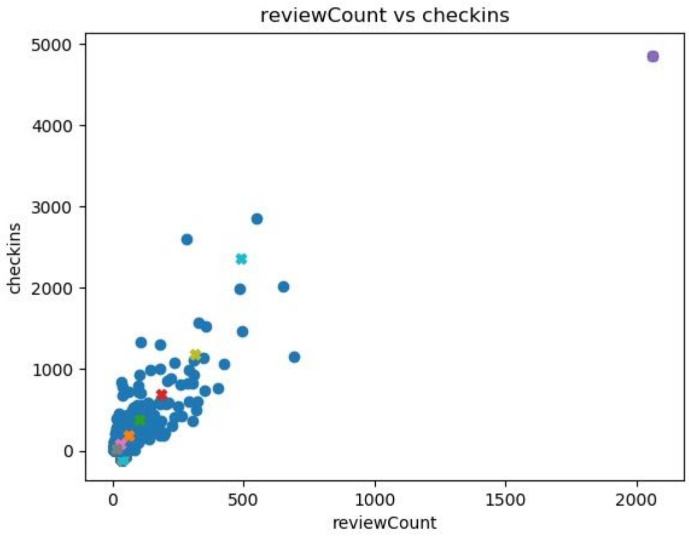
K (CLUSTER NUMBER) :	AVERAGE WC - SSE
3	26746623.163
6	14554849.421
9	5544210.087
12	4123662.066
24	8504923.124

- **WC - SSE FOR ALL 5 K = 9 RUNS:**
  -

K (CLUSTER NUMBER) = 9 :	WC - SSE FOR EACH TRIAL
#1	6135874.174907
#2	5400086.818758
#3	3452982.534696
#4	8630549.190706
#5	4101557.717133

- **CLUSTER GRAPH OF BEST K (K = 9) FROM BEST TRIAL (TRIAL #2) :**





- **POST-RUN EXPECTATIONS:** **I selected K = 9 as the optimal chosen K. We could see in the average run graph that after K = 9, K =12 made marginal improvement and then K = 24 actually got worse. For K = 9, trial #2 had the closest value to the average. The average value for K = 9, as seen in the table above, was 5544210.087. Trial #2 had a WC - SSE value of 5400086.818758. We ended up selecting the same k value as we did when we sampled the full set of data. The aforementioned behavior should not be taken to say that downsampling had no result. While the effect may not have been as pronounced as we would have hoped, downsampling did lead to the increase in K value that we expected once**

we hit the elbow joint. K = 24 has a considerable increase over K =12 and K = 9.

3.6. **(5 points)** Improved score function. In this case, you will use the score function you proposed in Theory (2.1) question 5. Using the best configuration from Questions 1-5, plot the results of your score function for K = [3, 6, 9, 12, 24], and compare the results to the appropriate algorithm from Question 1-5.