23S-EC ENGR-232E-LEC-1 Project 4: Graph Algorithms

SARAH WILEN, YAMAN YUCEL, ROBERT OZTURK

TOTAL POINTS

200 / 215

QUESTION 1

1 Question 1 **2 / 2**

✓ **- 0 pts** *Correct*

QUESTION 2

2 Question 2 **3 / 3**

✓ **- 0 pts** *Correct*

QUESTION 3

3 Question 3 **7 / 7**

✓ **- 0 pts** *Correct*

QUESTION 4

4 Question 4 **5 / 5**

✓ **- 0 pts** *Correct*

QUESTION 5

5 Question 5 **5 / 5**

✓ **- 0 pts** *Correct*

QUESTION 6

6 Question 6 **10 / 10**

✓ **- 0 pts** *Correct*

QUESTION 7

7 Question 7 **10 / 10**

✓ **- 0 pts** *Correct*

QUESTION 8

8 Question 8 **10 / 10**

✓ **- 0 pts** *Correct*

QUESTION 9

9 Question 9 **4 / 4**

✓ **- 0 pts** *Correct*

QUESTION 10

10 Question 10 **5 / 5**

✓ **- 0 pts** *Correct*

QUESTION 11

11 Question 11 **5 / 5**

✓ **- 0 pts** *Correct*

QUESTION 12

12 Question 12 **2 / 2**

✓ **- 0 pts** *Correct*

QUESTION 13

13 Question 13 **5 / 5**

✓ **- 0 pts** *Correct*

QUESTION 14

14 Question 14 **5 / 5**

✓ **- 0 pts** *Correct*

QUESTION 15

15 Question 15 **5 / 5**

✓ **- 0 pts** *Correct*

QUESTION 16

16 Question 16 **5 / 5**

✓ **- 0 pts** *Correct*

QUESTION 17

17 Question 17 **5 / 5**

✓ **- 0 pts** *Correct*

QUESTION 18

18 Question 18 **3 / 3**

✓ **- 0 pts** *Correct*

QUESTION 19

19 Question 19 **14 / 15**

✓ **- 1 pts** *time complexity slighltly wrong*

QUESTION 20

20 Question 20 **5 / 5**

✓ **- 0 pts** *Correct*

QUESTION 21

21 Question 21 **8 / 10**

✓ **- 2 pts** *Partial credit for time complexity*

QUESTION 22

22 Question 22 **9 / 10**

✓ **- 1 pts** *Partially wrong time complexity*

QUESTION 23

23 Question 23 **9 / 10**

✓ **- 1 pts** *time complexity slighltly wrong*

QUESTION 24

24 Question 24 **20 / 24**

✓ **- 4 pts** *dynamic is better but should not be optimal*

QUESTION 25

25 Define Your Own Task **39 / 45**

✓ **- 0 pts** *Correct*

- 6 *Point adjustment*

Task Definition: 4/5, Creativity: 5/10, Success: 10/10, Methodology: 10/10, Completeness: 10/10

ECE 232E Project 4:
Graph Algorithms

Question 1

$Let\ r_i = c * r_j$ (correlated)

$$\rho_{ij} = \frac{c\langle r_j^2\rangle - c\langle r_i\rangle^2}{\sqrt{c^2(\langle r_j^2\rangle - \langle r_i\rangle^2)^2}} = \frac{c(\langle r_j^2\rangle - \langle r_i\rangle^2)}{\sqrt{c^2}(\langle r_j^2\rangle - \langle r_i\rangle^2)} = \frac{c}{\sqrt{c^2}}$$

- When c is negative:
  - $\rho_{ij} = -1$ (lower bound)
- When c is positive:
  - $\rho_{ij} = 1$

Interpretation of the upper bound is when the correlation coefficient is 1, it indicates that stocks i and j move in perfect synchronization. Every increase or decrease in stock i's return corresponds to an equal increase or decrease in stock j's return. When the correlation coefficient is -1, this indicates perfect negative correlation where stocks i and j move in perfect opposition. When stock i increases, this corresponds to an equal decrease in stock j's returns (and vice versa).

There are a few benefits of using log returns, which make them more advantageous for financial calculations and analysis.

For one, log returns are normally distributed (if the prices are log normally distributed). This provides mathematical convenience. Prices are lower bounded at zero, so the log normal distribution has the correct domain support.

Next, log returns are more symmetric than raw returns, which is a feature of logarithms. This simplifies many stat models, such as linear regression, which assumes the data follows a symmetric distribution. Symmetric distributions also help provide us with interpretability such as the average or median. It helps us interpret the central tendency of log returns allowing analysis of comparing investments or evaluating the performance of the stock.

Another benefit is that log returns can be interpreted as continuously compounded returns, which helps us understand the growth or decay rate of the stock.

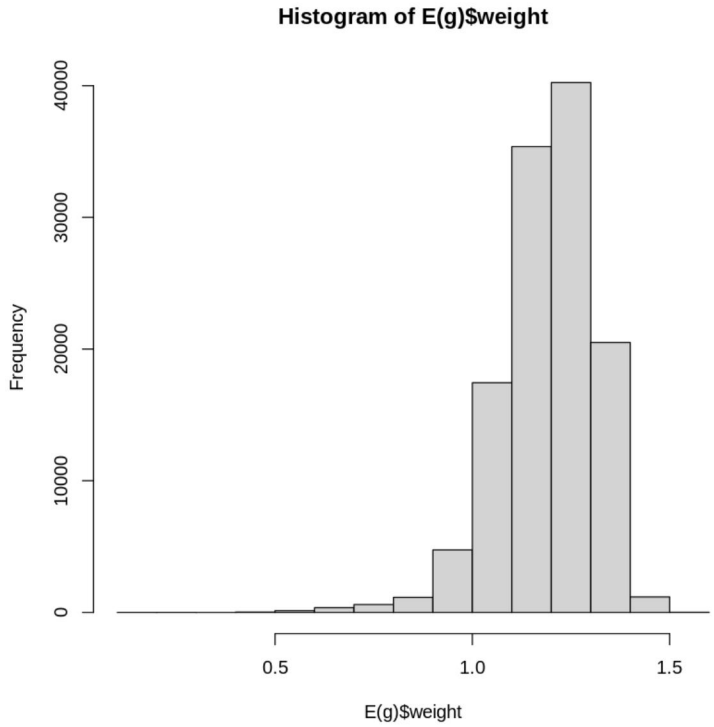
The next benefit is a feature of log transformation of multiplicative changes into additive changes. Therefore, the log return of an overall period is equal to the sum of the log returns of its sub-periods. This simplifies compounding return calculation (which is the multiplication of raw returns).

1 Question 1 **2 / 2**

✓ **- 0 pts** *Correct*

Question 2

Un-normalized distribution of edge weights:



If the stocks were positively correlated, then $\rho_{ij} \to 1$, so $w_{ij} \to 0$. if the stocks are negatively correlated, then $\rho_{ij} \to -1$, so $w_{ij} \to 2$. If the stocks are not correlated (i.e. independent), then $\rho_{ij} \to 0$, so $w_{ij} \to \sqrt{2} = 1.41$

From this distribution, we see that most of the edge weights are around 1.3-1.5, therefore, this indicates that the stocks are uncorrelated, which makes sense considering the sectors that the stocks come from are pretty distinct and independent of each other.

Question 3

The minimum spanning tree is a tree that spans all the nodes with the minimum total edge weight. In other words, it is a tree that is formed in an attempt to minimize the distance between the nodes. In our case, the smaller the edge weight, that means the closer (positively) correlated the stocks.

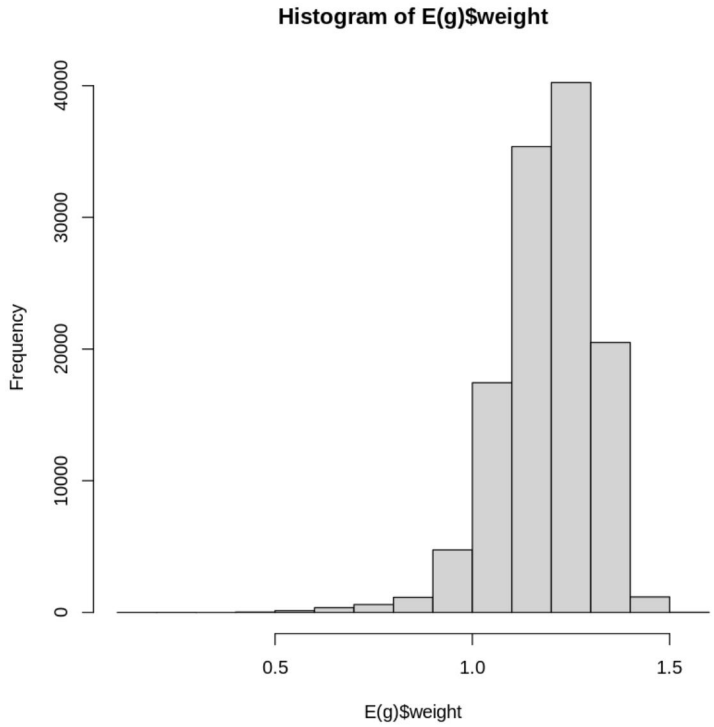
Minimum spanning tree with colors correlated to sector:

2 Question 2 **3 / 3**

✓ **- 0 pts** *Correct*

Question 2

Un-normalized distribution of edge weights:



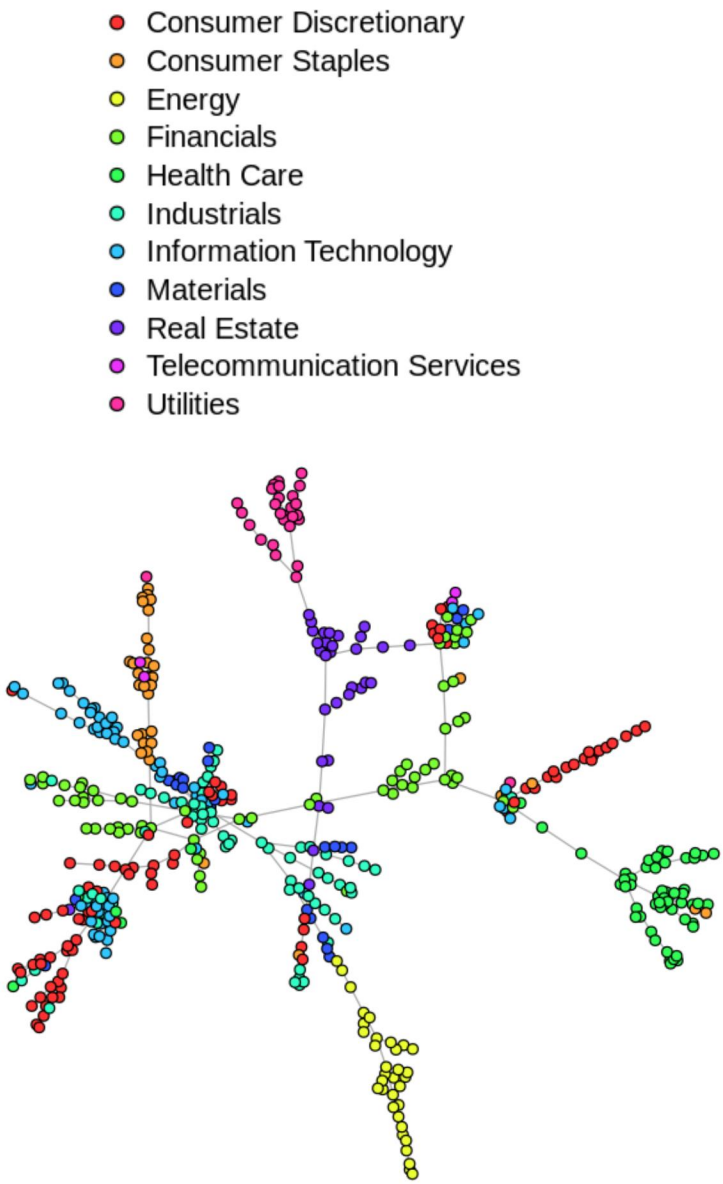
If the stocks were positively correlated, then $\rho_{ij} \to 1$, so $w_{ij} \to 0$. if the stocks are negatively correlated, then $\rho_{ij} \to -1$, so $w_{ij} \to 2$. If the stocks are not correlated (i.e. independent), then $\rho_{ij} \to 0$, so $w_{ij} \to \sqrt{2} = 1.41$

From this distribution, we see that most of the edge weights are around 1.3-1.5, therefore, this indicates that the stocks are uncorrelated, which makes sense considering the sectors that the stocks come from are pretty distinct and independent of each other.

Question 3

The minimum spanning tree is a tree that spans all the nodes with the minimum total edge weight. In other words, it is a tree that is formed in an attempt to minimize the distance between the nodes. In our case, the smaller the edge weight, that means the closer (positively) correlated the stocks.

Minimum spanning tree with colors correlated to sector:



As expected, we notice that the “vine clusters” are those stocks within the same cluster. For example, all the energy stocks are in one group towards the bottom on the graph (in yellow). The interpretation of this is that stocks in the energy sector are correlated with each other, which is how it is in real life. Moreover, nodes within one sector are further away from other sectors, implying that nodes in the same sector are more correlated with each other than to the nodes of other industries.

We also notice that the stocks furthest from each other are the stocks we expect to not be very related. For example, energy nodes are far from real estate. This makes intuitive sense because these two sectors are not highly correlated.

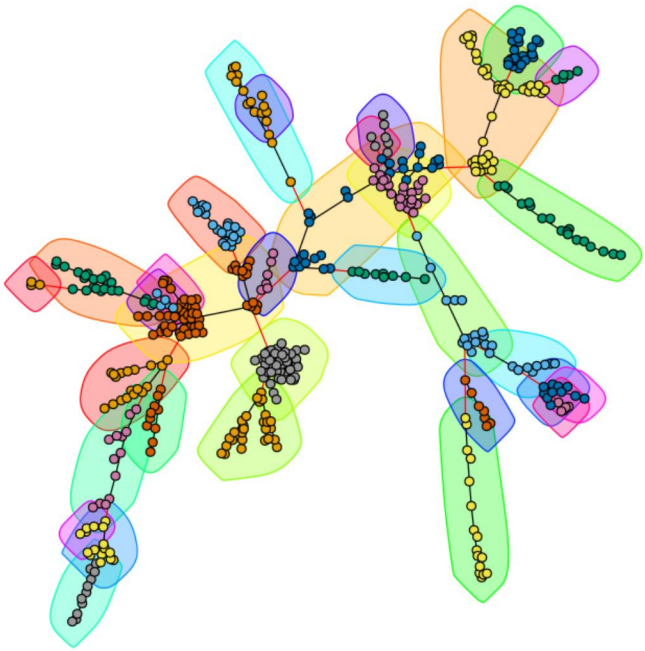
The stock sector groups closer to each other makes sense as well. For example, it looks like health care, industrials, and financial stocks are close to each other. The interpretation for this is that these sectors interact with each other closely and changes to the sector will influence each other.

3 Question 3 **7 / 7**

✓ **- 0 pts** *Correct*

Question 4

Walktrap community detection algorithm:



Homogeneity: 0.682644648161366
Completeness: 0.479284479244588

Question 5

Method 1)
The probability that a node is in a sector for method 1 is the number of neighboring nodes that belong to the same sector as node i divided by the total number of neighbors of nodes i.

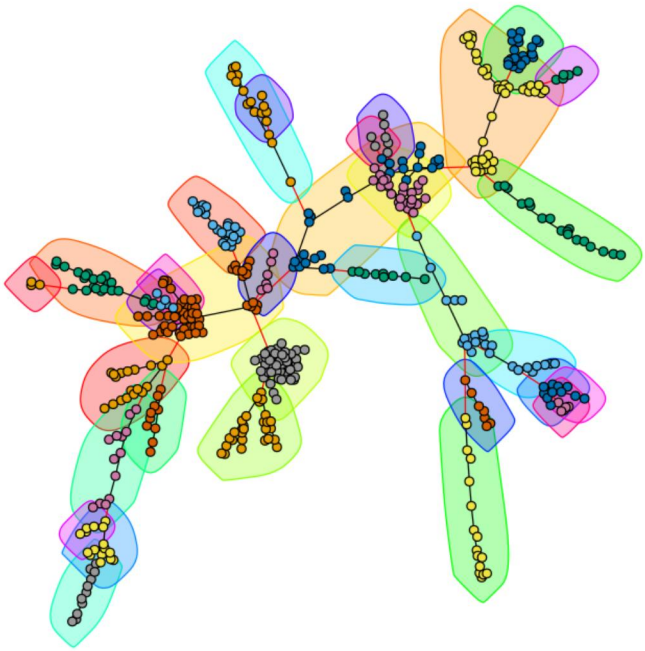
Alpha value for method 1 is: 0.828930077530676"

4 Question 4 **5 / 5**

✓ **- 0 pts** *Correct*

Question 4

Walktrap community detection algorithm:



Homogeneity: 0.682644648161366
Completeness: 0.479284479244588

Question 5

Method 1)
The probability that a node is in a sector for method 1 is the number of neighboring nodes that belong to the same sector as node i divided by the total number of neighbors of nodes i.

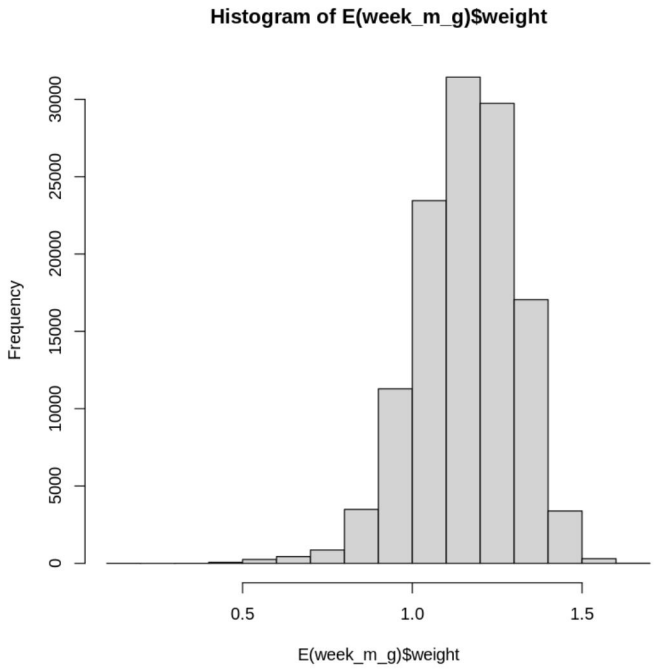
Alpha value for method 1 is: 0.828930077530676"

Method 2)
This is a naive method. The probability that a node is in the sector is the total number of stocks in each sector divided by the total number of stocks.

Alpha value for method 2 is: 0.114188070612533

As expected, the alpha value for method 2 is a lot lower than for method 1 because method 1 uses local neighboring information, whereas method 2 is a naive approach that considers all nodes in a sector without using the local cluster information. It makes more sense that neighboring nodes are more related and, hence, are more probable to be similar stocks, so we favor method 2 more in this case.

Question 6



5 Question 5 **5 / 5**

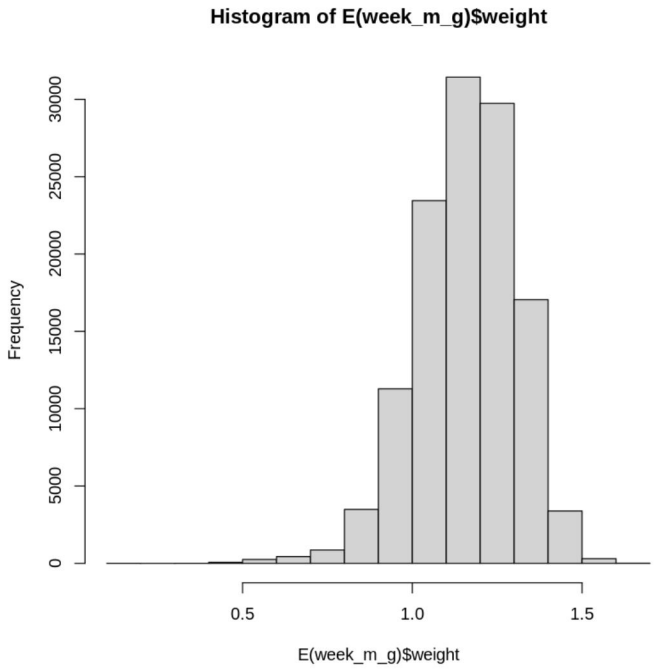
✓ **- 0 pts** *Correct*

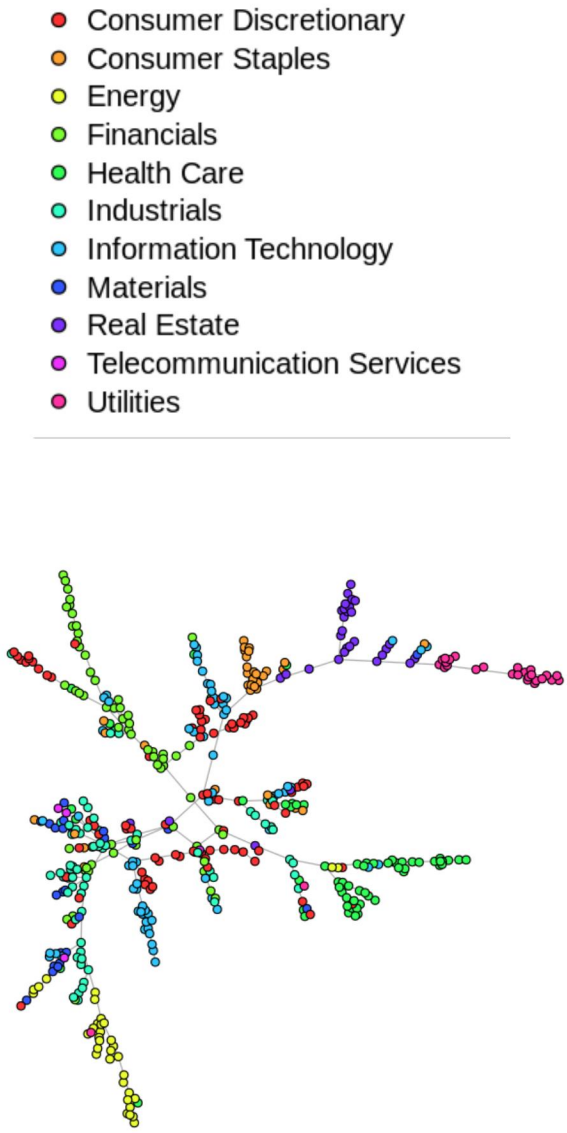
Method 2)
This is a naive method. The probability that a node is in the sector is the total number of stocks in each sector divided by the total number of stocks.

Alpha value for method 2 is: 0.114188070612533

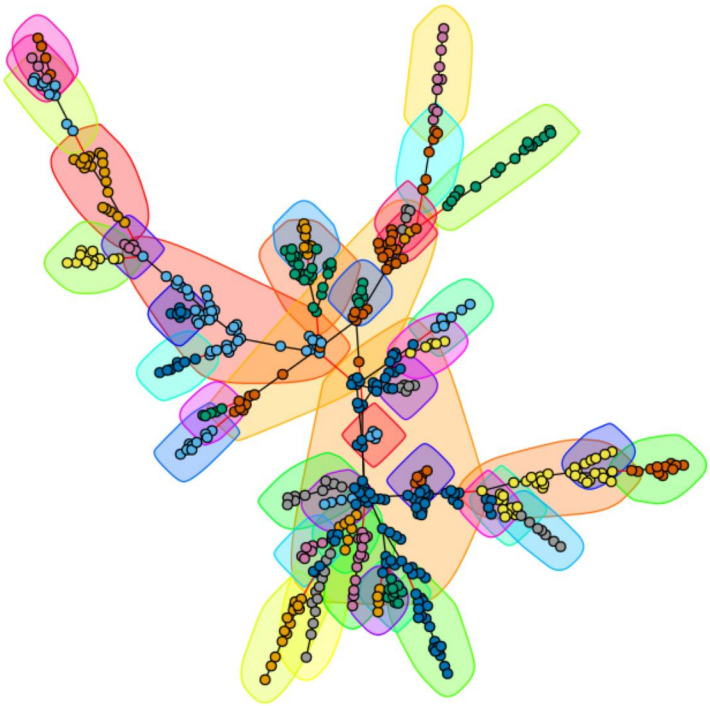
As expected, the alpha value for method 2 is a lot lower than for method 1 because method 1 uses local neighboring information, whereas method 2 is a naive approach that considers all nodes in a sector without using the local cluster information. It makes more sense that neighboring nodes are more related and, hence, are more probable to be similar stocks, so we favor method 2 more in this case.

Question 6





The structure is similar to the daily MST where stocks of the same sector are closest together and are in clusters together. Stocks that are a part of different sectors are apart.



Homogeneity: 0.58200695320238
Completeness: 0.390771399621547

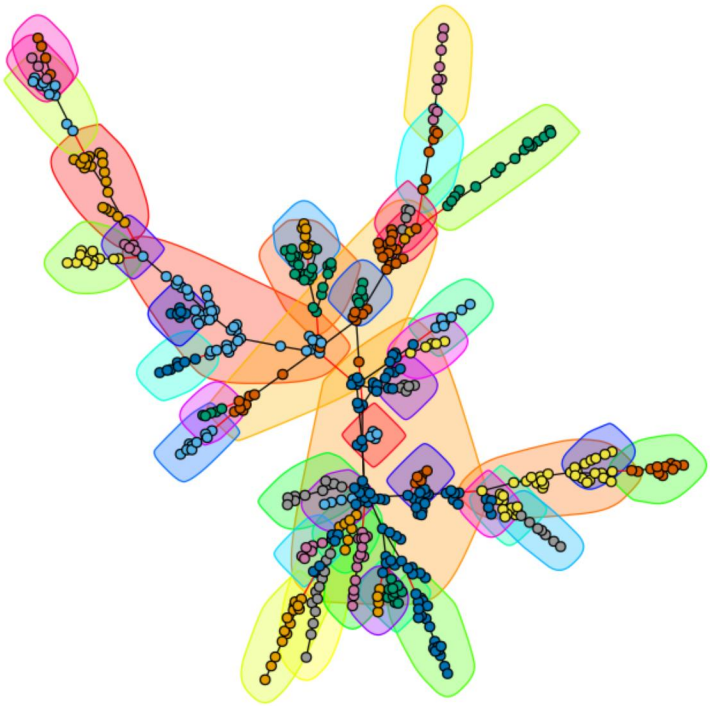
Alpha value for method 1 is: 0.742969603495919
Alpha value for method 2 is: 0.114188070612533

The difference between alpha methods is the same as the daily stocks. Method 1 still performs better because it uses local information rather than global information about stocks and neighboring performance to measure.

Question 7

6 Question 6 **10 / 10**

✓ **- 0 pts** *Correct*

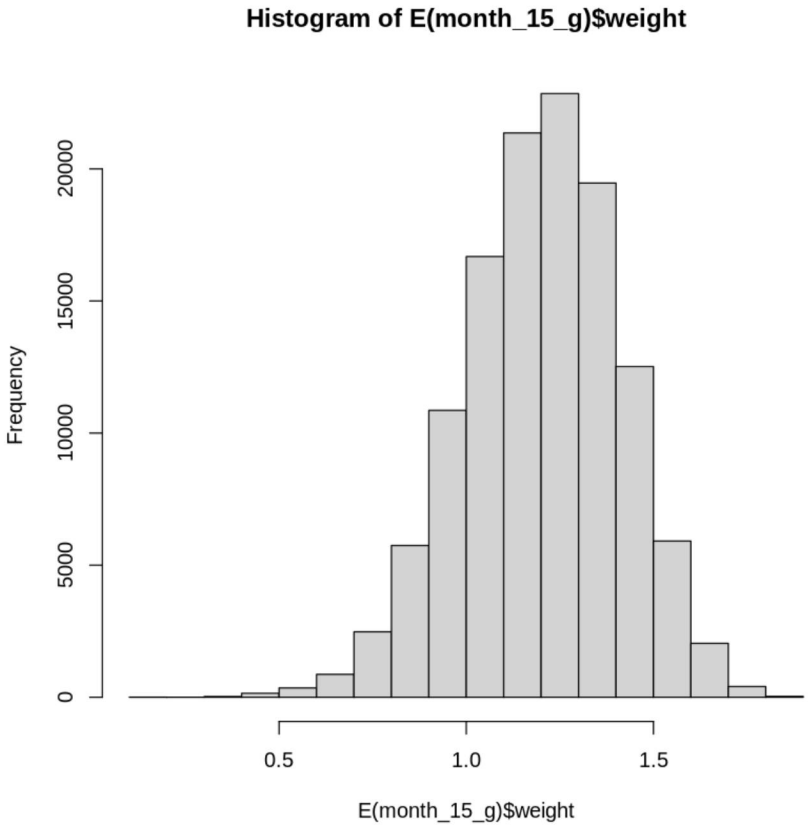


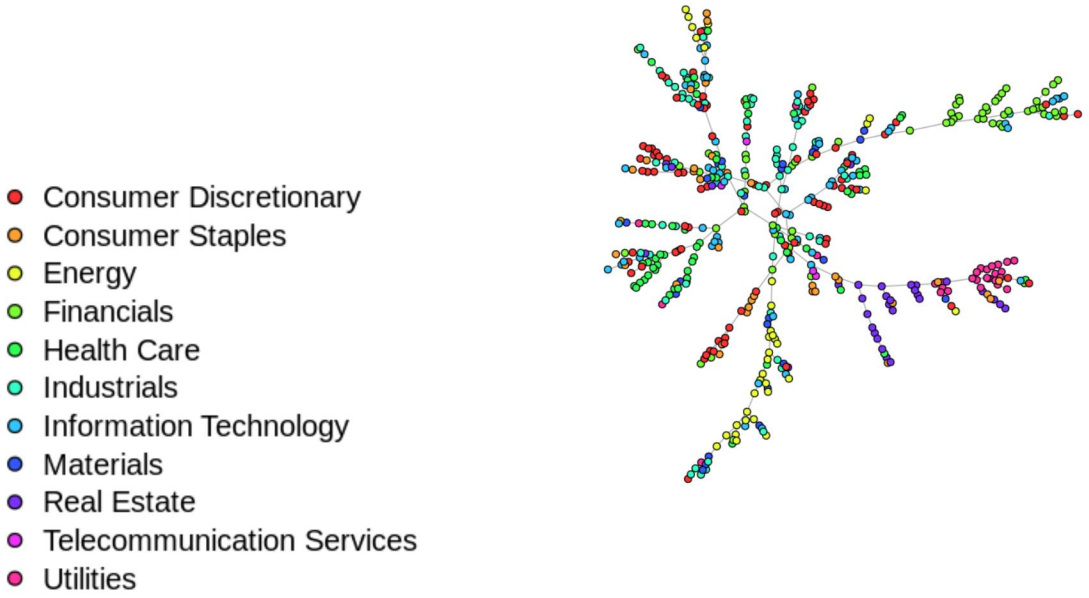
Homogeneity: 0.58200695320238
Completeness: 0.390771399621547

Alpha value for method 1 is: 0.742969603495919
Alpha value for method 2 is: 0.114188070612533

The difference between alpha methods is the same as the daily stocks. Method 1 still performs better because it uses local information rather than global information about stocks and neighboring performance to measure.

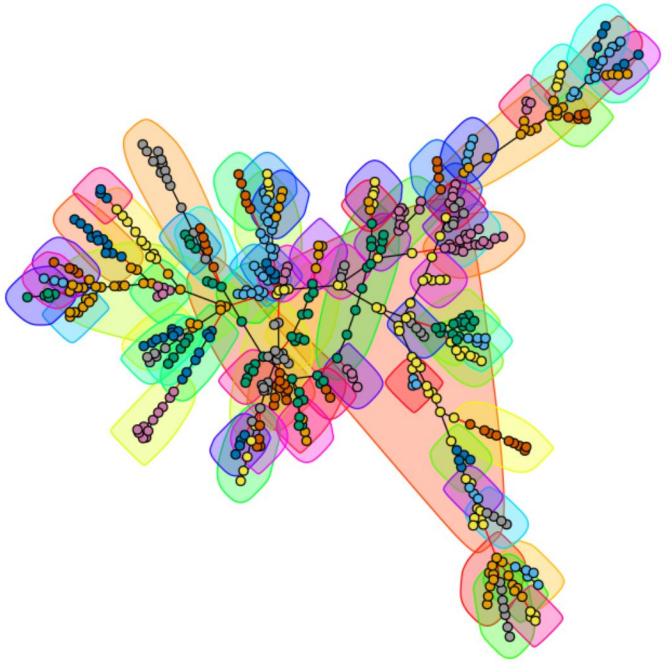
Question 7





Interestingly, for the monthly stock data, the clustering of distinct sectors is not as clear as the daily stocks. The stocks from different sectors are starting to intermingle. This suggests that over a longer

period of time, it is less clear cut how stocks correlate with each other. This makes sense because different companies within the same sector may correlate closely with each other when you look at their stocks daily. However, when you sample specific dates of stocks, these values correlate less closely.



Homogeneity: 0.509509234436883
Completeness: 0.282337925983778

Alpha value for method 1 is: 0.483468286099865
Alpha value for method 2 is: 0.114188070612533

We see the same difference between alpha values for method 1 and 2 and this is for the same reasoning as explained above. However, it is interesting to notice that the alpha value for method 1 is decreasing, which makes sense. The homogeneity of the clusters is decreasing and so is the completeness. It is more difficult to discern which stock belongs to which community and which sector because the clusters are increasingly intermingling since the correlation between stocks within the same sector is less distinct compared to stocks from other sectors.

7 Question 7 **10 / 10**

✓ **- 0 pts** *Correct*

Question 8

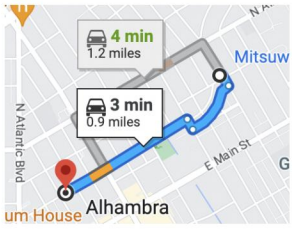
Expanding on explanation from previous questions 6 and 7, as we move from daily to monthly data, the correlation of nodes is less discernible amongst nodes of the same sector. We see from the MST tree that as we move from daily to monthly, the clustering between nodes of the same sector is less distinct. Nodes between different sectors are now close to each other too and it is more muddied to separate the different sectors into distinct clusters.

We see that daily data provides the best granularity to predict the sector of an unknown stock. This is because daily stocks provide the closest correlation within sectors, so it is easier to understand the relationships between stocks. However, as we move to monthly stocks, the stocks in the same sector are not as distinctly correlated (as noticed by the MST and cluster graphs).

Question 9

Number of nodes: 2649
Number of edges: 1003858

Question 10

Coordinates (longitude, latitude)	Distance (miles)	Time (seconds)	Street Addresses (from Google Maps)
Source: [-118.12053321 34.10309557] Target: [-118.13138209 34.09626386]	0.885	129.8	

8 Question 8 **10 / 10**

✓ **- 0 pts** *Correct*

Question 8

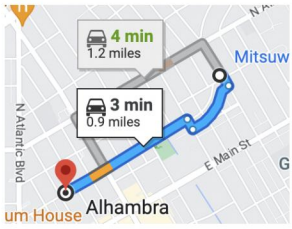
Expanding on explanation from previous questions 6 and 7, as we move from daily to monthly data, the correlation of nodes is less discernible amongst nodes of the same sector. We see from the MST tree that as we move from daily to monthly, the clustering between nodes of the same sector is less distinct. Nodes between different sectors are now close to each other too and it is more muddied to separate the different sectors into distinct clusters.

We see that daily data provides the best granularity to predict the sector of an unknown stock. This is because daily stocks provide the closest correlation within sectors, so it is easier to understand the relationships between stocks. However, as we move to monthly stocks, the stocks in the same sector are not as distinctly correlated (as noticed by the MST and cluster graphs).

Question 9

Number of nodes: 2649
Number of edges: 1003858

Question 10

Coordinates (longitude, latitude)	Distance (miles)	Time (seconds)	Street Addresses (from Google Maps)
Source: [-118.12053321 34.10309557] Target: [-118.13138209 34.09626386]	0.885	129.8	

9 Question 9 4 / 4

✓ **- 0 pts** *Correct*

Question 8

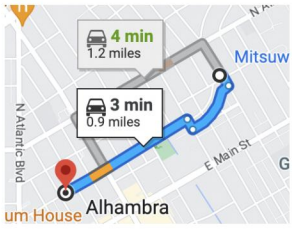
Expanding on explanation from previous questions 6 and 7, as we move from daily to monthly data, the correlation of nodes is less discernible amongst nodes of the same sector. We see from the MST tree that as we move from daily to monthly, the clustering between nodes of the same sector is less distinct. Nodes between different sectors are now close to each other too and it is more muddied to separate the different sectors into distinct clusters.

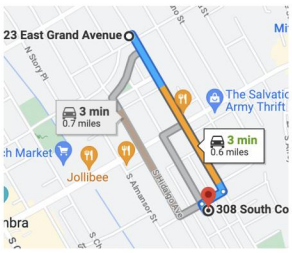
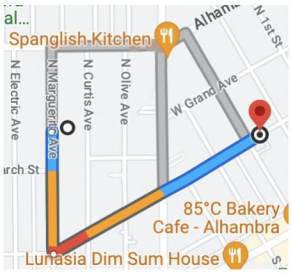
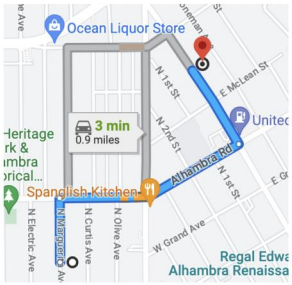
We see that daily data provides the best granularity to predict the sector of an unknown stock. This is because daily stocks provide the closest correlation within sectors, so it is easier to understand the relationships between stocks. However, as we move to monthly stocks, the stocks in the same sector are not as distinctly correlated (as noticed by the MST and cluster graphs).

Question 9

Number of nodes: 2649
Number of edges: 1003858

Question 10

Coordinates (longitude, latitude)	Distance (miles)	Time (seconds)	Street Addresses (from Google Maps)
Source: [-118.12053321 34.10309557] Target: [-118.13138209 34.09626386]	0.885	129.8	

Source: [-118.12053321 34.10309557] Target: [-118.11656383 34.09585388]	0.570	123.8	
Source: [-118.13785063 34.09645121] Target: [-118.13138209 34.09626386]	0.447	90.2	 Distance: 0.7 mile Time: 3 min
Source: [-118.13785063 34.09645121] Target: [-118.13224544 34.10349303]	0.621	126.5	

As seen by the actual Google Maps data, this does make intuitive sense. The sources and destinations created by the MST are very close to each other in distance. Most all the sources and destinations are within 1 mile of each other.

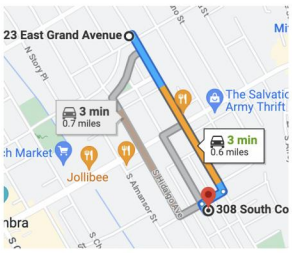
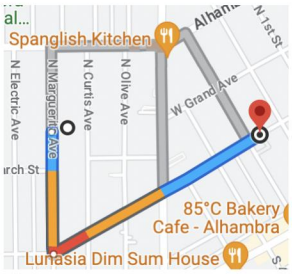
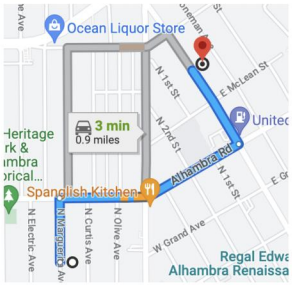
Question 11

Out of 1,000 random samples, the triangle inequality holds for 92.3% of them.

The distance of the three locations on the triangle, in fact, do fit the triangle inequality. This ensures that our MST has valid and optimal connections that minimize the overall traveling time cost.

10 Question 10 **5 / 5**

✓ **- 0 pts** *Correct*

Source: [-118.12053321 34.10309557] Target: [-118.11656383 34.09585388]	0.570	123.8	
Source: [-118.13785063 34.09645121] Target: [-118.13138209 34.09626386]	0.447	90.2	 Distance: 0.7 mile Time: 3 min
Source: [-118.13785063 34.09645121] Target: [-118.13224544 34.10349303]	0.621	126.5	

As seen by the actual Google Maps data, this does make intuitive sense. The sources and destinations created by the MST are very close to each other in distance. Most all the sources and destinations are within 1 mile of each other.

Question 11

Out of 1,000 random samples, the triangle inequality holds for 92.3% of them.

The distance of the three locations on the triangle, in fact, do fit the triangle inequality. This ensures that our MST has valid and optimal connections that minimize the overall traveling time cost.

11 Question 11 **5 / 5**

✓ **- 0 pts** *Correct*

Question 12

From the given approximation algorithm…

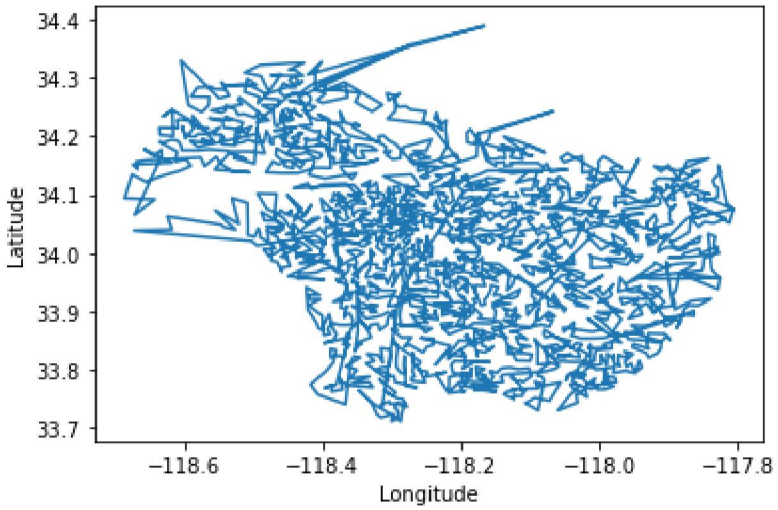
1. Find the MST
2. Create a multi-graph
    a. For each edge, replace by two directed edges
3. Find a Euler tour in the multi-graph
    a. Euler cycle: a cycle that covers every edge only once
4. For each edge in the Eurler tour…
    a. If the edge is not in the original G…
        i. Substitute the path with the shortest path using Dijkstra's algorithm
5. Calculate the TSP cost
    a. $TSP\ cost = \sum weights\ of\ edges\ in\ the\ tour\ sequence$

$$upper\ bound = \frac{minimum\ approx\ cost}{MST\ cost} = \frac{421482.315}{269184.545} = 1.56577$$

The upper bound was calculated by taking the total weight of the path created by the approximate TSP cost and dividing it by the sum of all the weights in the MST.

Question 13

Using a map of Los Angeles (https://www.openstreetmap.org/relation/207359) we superimposed the longitudinal and latitudinal trajectory Santa needs to take.



12 Question 12 **2 / 2**

✓ **- 0 pts** *Correct*

Question 12

From the given approximation algorithm…

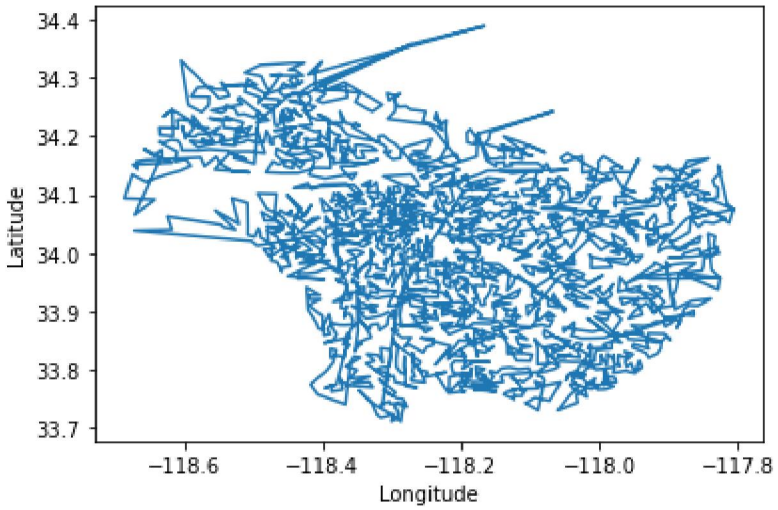
1. Find the MST
2. Create a multi-graph
    a. For each edge, replace by two directed edges
3. Find a Euler tour in the multi-graph
    a. Euler cycle: a cycle that covers every edge only once
4. For each edge in the Eurler tour…
    a. If the edge is not in the original G…
        i. Substitute the path with the shortest path using Dijkstra's algorithm
5. Calculate the TSP cost
    a. $TSP\ cost = \sum weights\ of\ edges\ in\ the\ tour\ sequence$

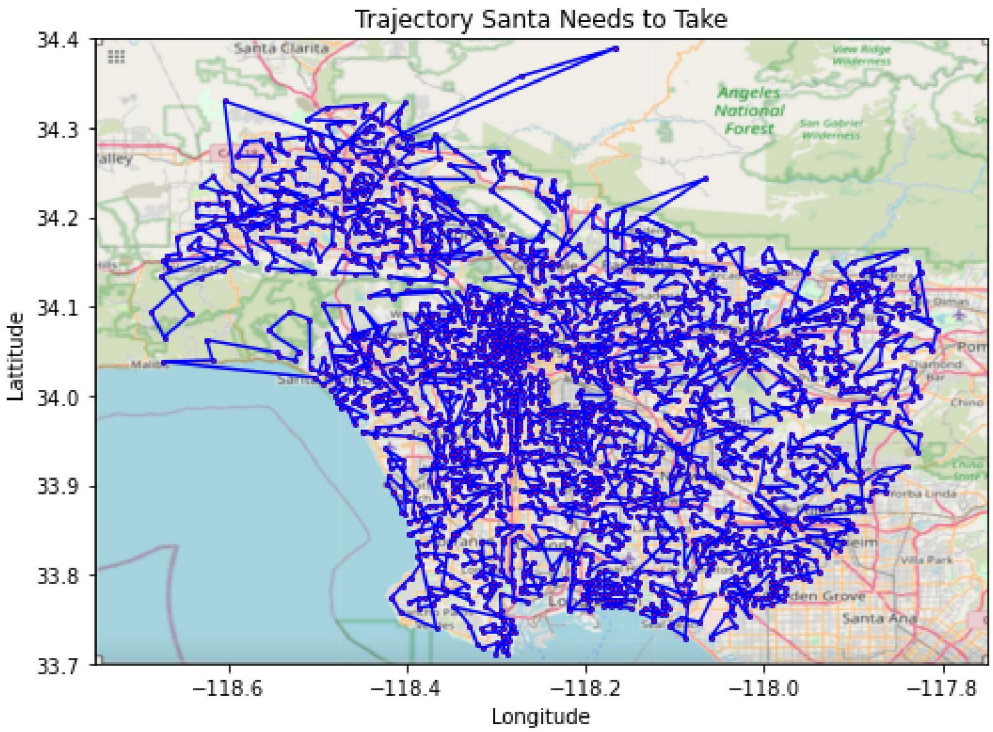
$$upper\ bound = \frac{minimum\ approx\ cost}{MST\ cost} = \frac{421482.315}{269184.545} = 1.56577$$

The upper bound was calculated by taking the total weight of the path created by the approximate TSP cost and dividing it by the sum of all the weights in the MST.

Question 13

Using a map of Los Angeles (https://www.openstreetmap.org/relation/207359) we superimposed the longitudinal and latitudinal trajectory Santa needs to take.





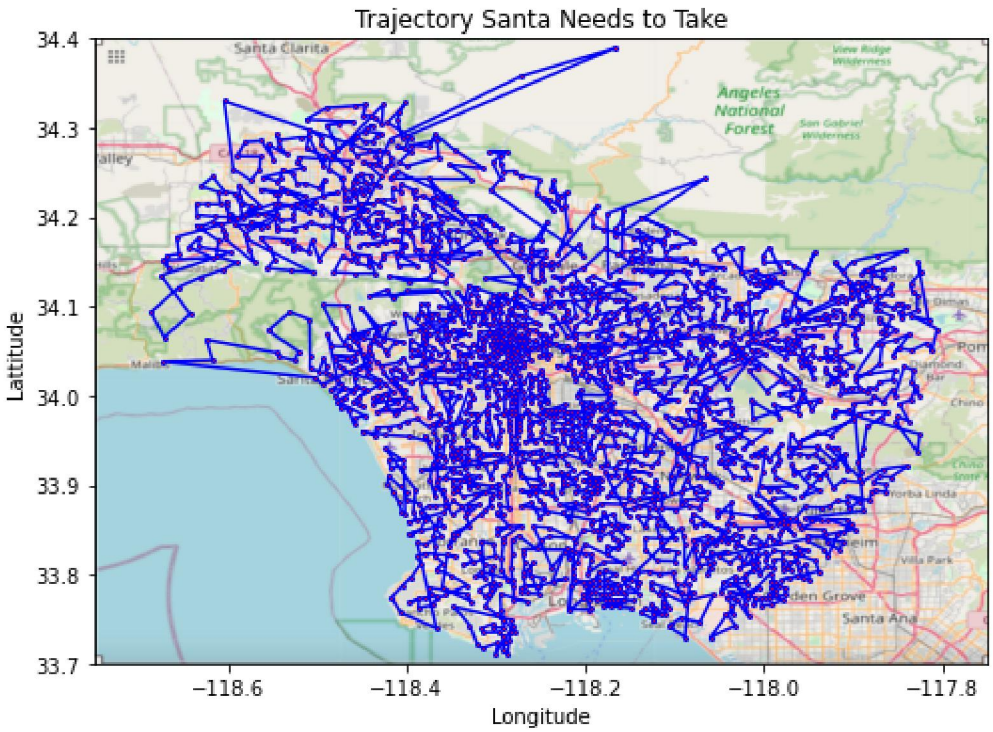
An interesting result is that some of the paths go straight into the Santa Monica mountains and even the Ocean (you'll notice between Santa Monica and Malibu). This is due to the fact that the approximation algorithm does not fully take into account road information. However, this is a fine approximation for Santa's case because he can probably fly. With that said, we see our approximation algorithm does a good job because the path is chosen that minimizes the overall mean travel time for each successive address with few edges to overlap with each other.

Question 14

Delaunay triangulation is used to obtain an estimation of road maps of LA. Delaunay triangulation algorithm connects the node of the graph such that connections of nodes form a triangle and no other node is inside of any triangle. The downside of using the Delaunay triangulation is that it introduces virtual nodes to satisfy triangle properties. Also, Delaunay triangulation tends to avoid silver triangles which are triangles with one or two extremely acute angles. Following road map is obtained.

13 Question 13 **5 / 5**

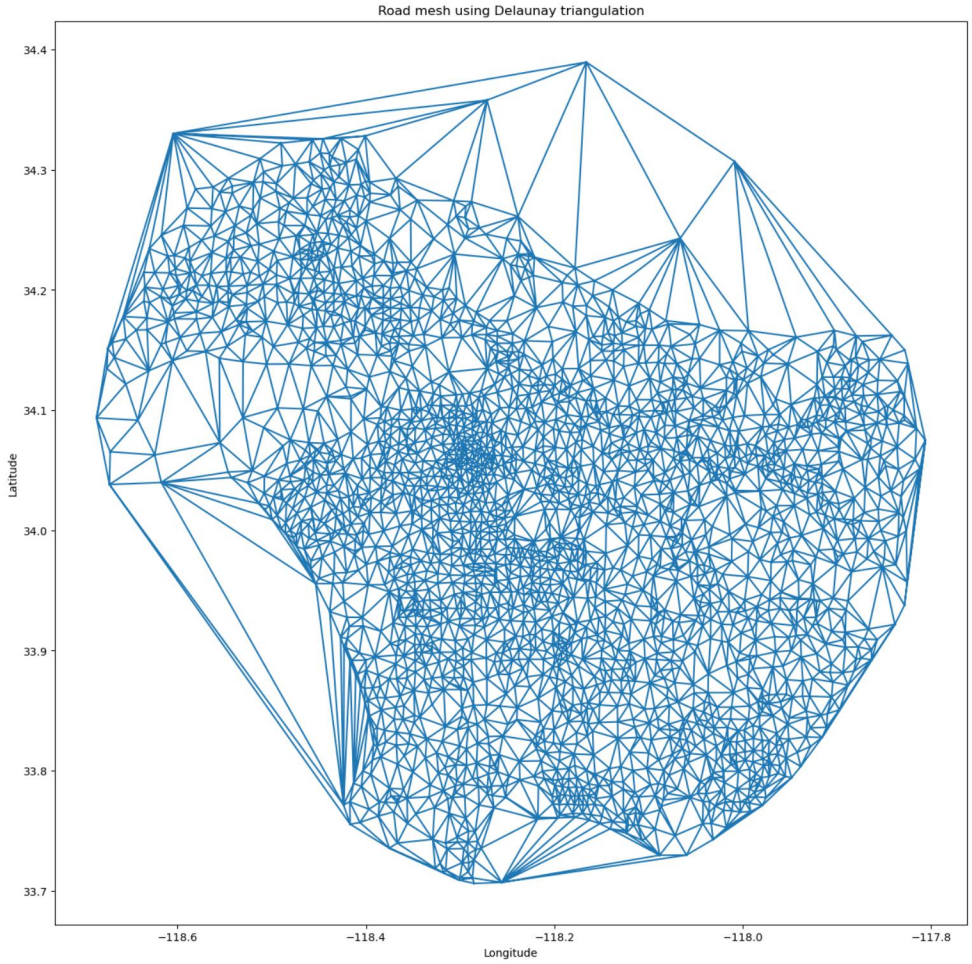
✓ **- 0 pts** *Correct*



An interesting result is that some of the paths go straight into the Santa Monica mountains and even the Ocean (you'll notice between Santa Monica and Malibu). This is due to the fact that the approximation algorithm does not fully take into account road information. However, this is a fine approximation for Santa's case because he can probably fly. With that said, we see our approximation algorithm does a good job because the path is chosen that minimizes the overall mean travel time for each successive address with few edges to overlap with each other.

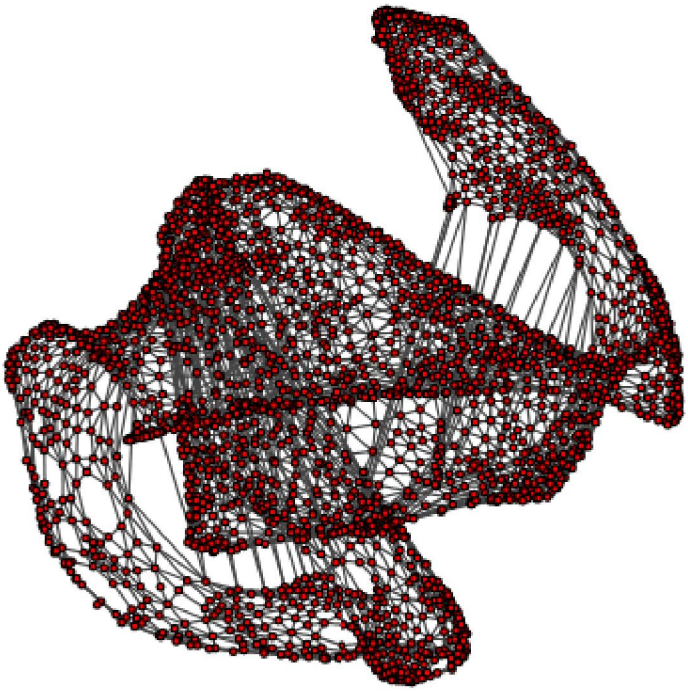
Question 14

Delaunay triangulation is used to obtain an estimation of road maps of LA. Delaunay triangulation algorithm connects the node of the graph such that connections of nodes form a triangle and no other node is inside of any triangle. The downside of using the Delaunay triangulation is that it introduces virtual nodes to satisfy triangle properties. Also, Delaunay triangulation tends to avoid silver triangles which are triangles with one or two extremely acute angles. Following road map is obtained.

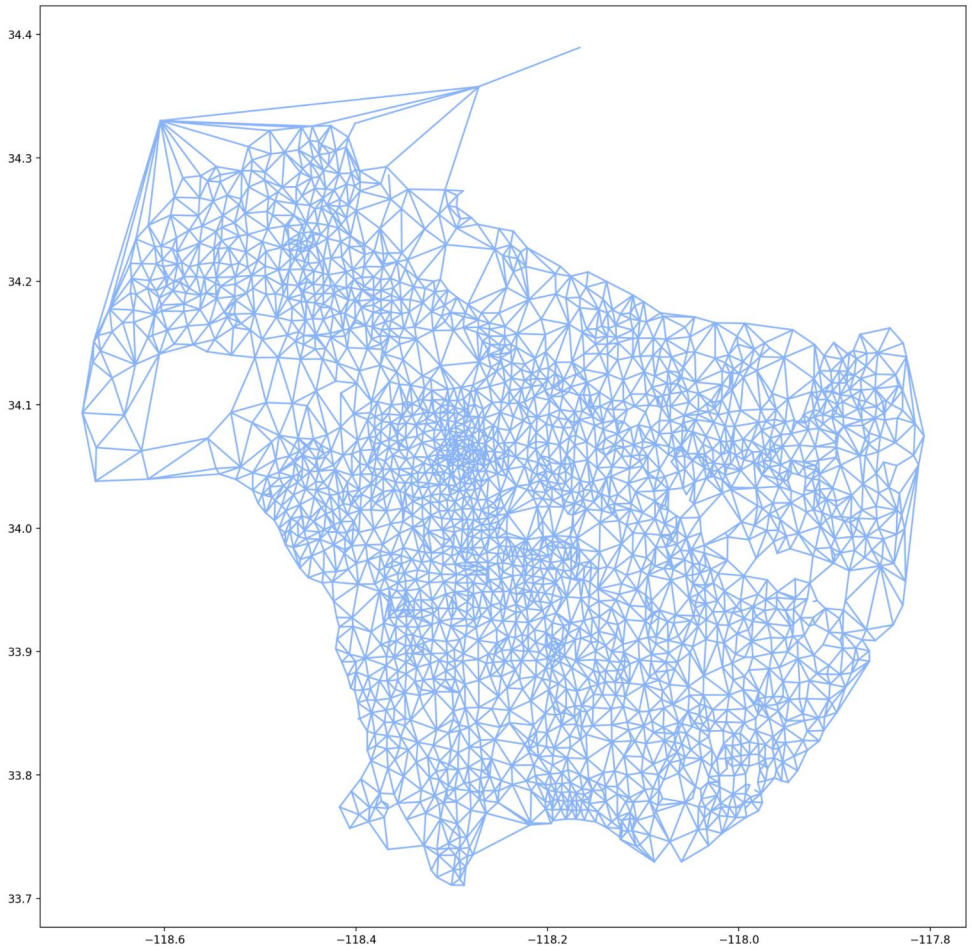


Note that, the road map describes the map of LA with some errors such as the connection between Long beach and Malibu. The road density decreases in the areas of Bel Air since that area consists of mountains and increases in the downtown area. Downtown LA has one of the most complex road maps in the world and this is easily seen from the above road map.

Then, we have found G_Δ using the edges obtained from Delaunay triangularization, we have also discarded edges that are not present in G but present in triangularization. Therefore, G_Δ becomes a subgraph of G. We have plotted G_Δ using the helper code.



However, since it is better to plot the graph using the coordinates of the nodes, we have also plotted the G_{Δ} using the road map technique. Each node is plotted according to the coordinates of that node.



This figure really looks similar to the original map of LA, since we have discarded many virtual roads.

Question 15

Traffic flow for each road in terms of car/hour can be estimated as follows.

1) First, we have found the length of the road using the two end coordinates of the node, then this length is converted to miles by multiplying the length by 69.(miles)

$$d = 69\sqrt{(lat_i - lat_j)^2 + (lon_i - lon_j)^2}$$

2) Second, time is converted to hours.(h)

$$t_h = \frac{t}{3600}$$

3) Average speed of cars on the road is calculated.(mph)

$$s = \frac{d}{t_h}$$

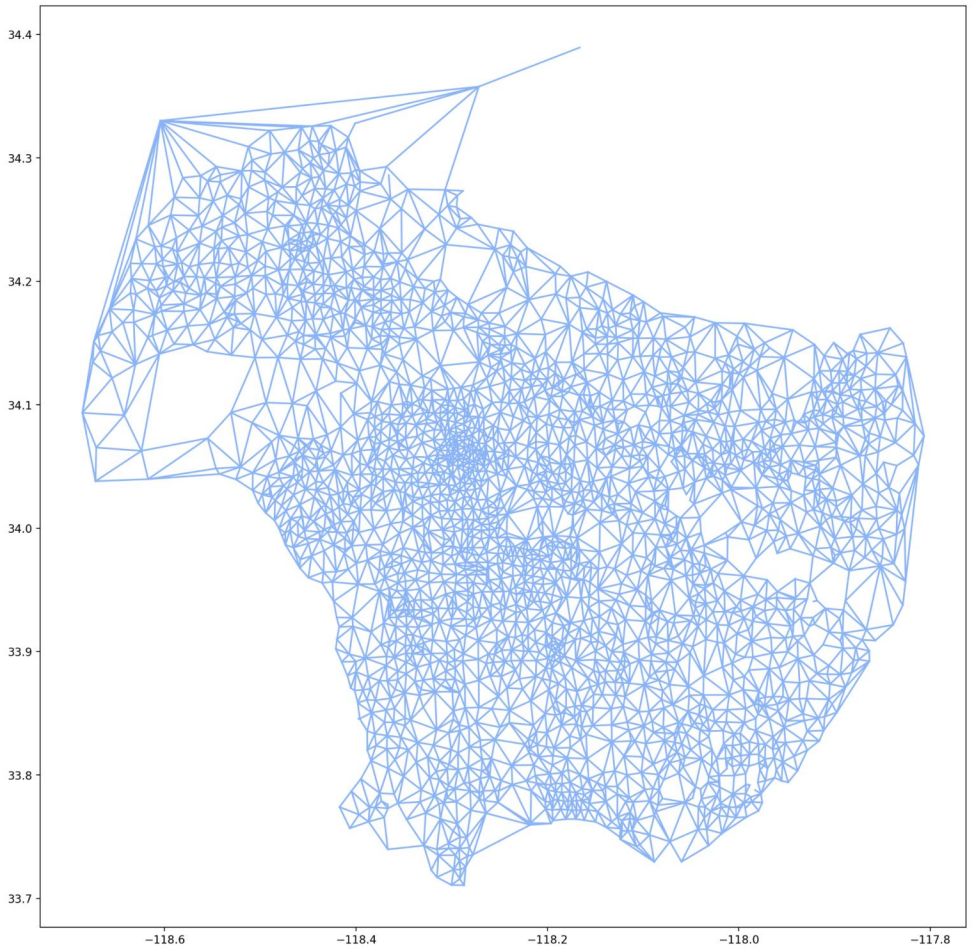
4) Safety distance between cars are calculated.(miles)

$$s_d = s\frac{2}{3600}$$

5) Number of cars on road the can be calculated by the diving the length of the road by (car length + safety distance)

14 Question 14 **5 / 5**

✓ **- 0 pts** *Correct*



This figure really looks similar to the original map of LA, since we have discarded many virtual roads.

Question 15

Traffic flow for each road in terms of car/hour can be estimated as follows.

1) First, we have found the length of the road using the two end coordinates of the node, then this length is converted to miles by multiplying the length by 69.(miles)

$$d = 69\sqrt{(lat_i - lat_j)^2 + (lon_i - lon_j)^2}$$

2) Second, time is converted to hours.(h)

$$t_h = \frac{t}{3600}$$

3) Average speed of cars on the road is calculated.(mph)

$$s = \frac{d}{t_h}$$

4) Safety distance between cars are calculated.(miles)

$$s_d = s\frac{2}{3600}$$

5) Number of cars on road the can be calculated by the diving the length of the road by (car length + safety distance)

$N = c_l + s_d = 0.003 + s\frac{2}{3600}$

6) Then the cars per hour is the number of cars divided by the time. Note, we have multiplied by 2 since there are two lanes.

$flow = \frac{2N}{t_h} = \frac{3600s}{5.4+s}$

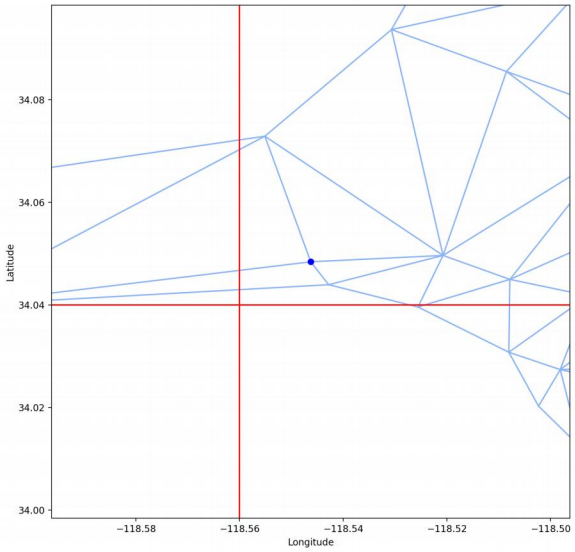
We have calculated the flow of each road using the formula above and assigned them as the max capacity of the road.

Question 16

In this part, using the capacities calculated in the Q15, we have done max-flow analysis to find the maximum flow from Malibu to Long Beach.

Maximum numbers of cars that can commute per hour from Malibu to Long Beach is found as 11069. The exact coordinates were not available in the road map of G_Δ, therefore we have used the closest node to the given coordinates for both locations.

Number of edge-disjoint paths is found to be 4. Degree of node Malibu is 4, degree of node Long Beach is 6. Number of edge-disjoints should be less than or equal to the min(degree Malibu, degree Long Beach) = 4. Therefore, they match. Also,we can conclude that flow is processed through 4 parallel channels.



Malibu is marked with blue and red lines indicating the given coordinates at the assignment. Note that the degree of malibu is 4.

15 Question 15 **5 / 5**

✓ **- 0 pts** *Correct*

$N = c_l + s_d = 0.003 + s\frac{2}{3600}$

6) Then the cars per hour is the number of cars divided by the time. Note, we have multiplied by 2 since there are two lanes.

$flow = \frac{2N}{t_h} = \frac{3600s}{5.4+s}$

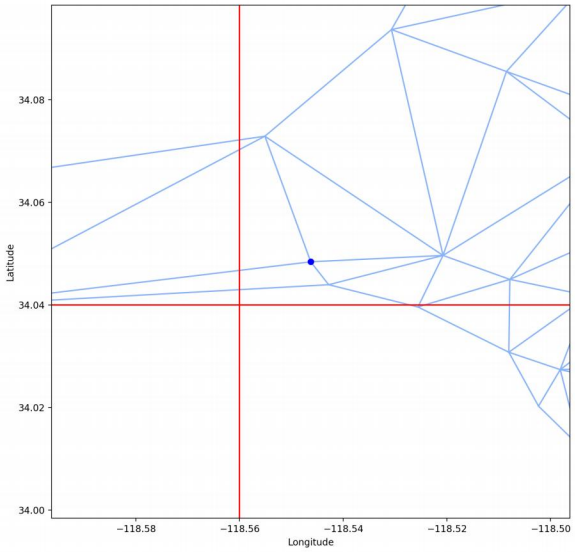
We have calculated the flow of each road using the formula above and assigned them as the max capacity of the road.

Question 16

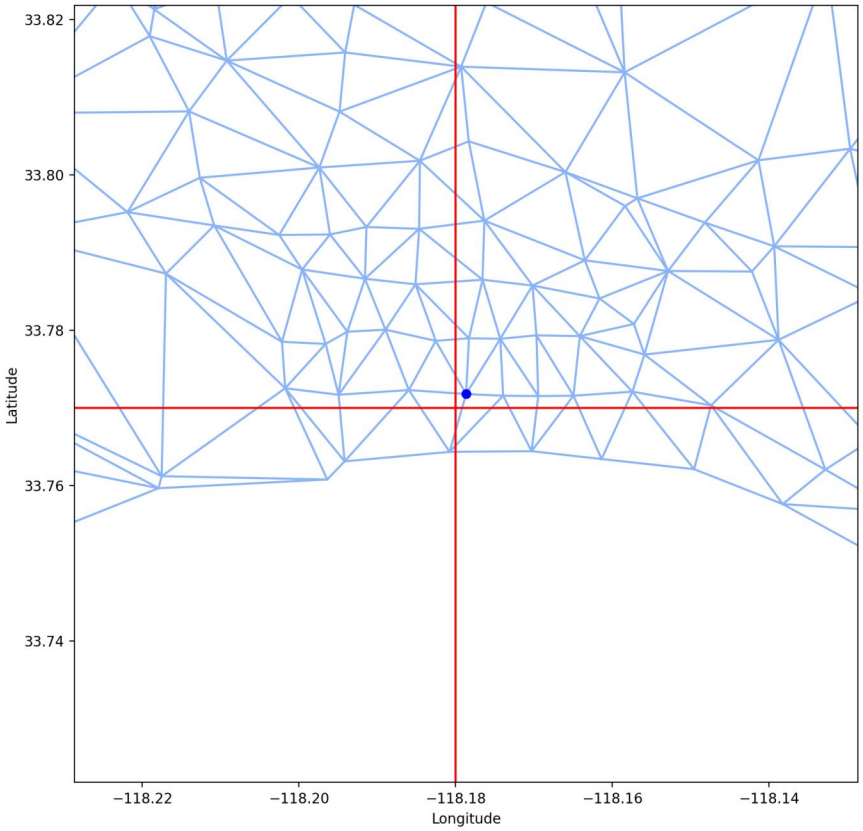
In this part, using the capacities calculated in the Q15, we have done max-flow analysis to find the maximum flow from Malibu to Long Beach.

Maximum numbers of cars that can commute per hour from Malibu to Long Beach is found as 11069. The exact coordinates were not available in the road map of G_Δ, therefore we have used the closest node to the given coordinates for both locations.

Number of edge-disjoint paths is found to be 4. Degree of node Malibu is 4, degree of node Long Beach is 6. Number of edge-disjoints should be less than or equal to the min(degree Malibu, degree Long Beach) = 4. Therefore, they match. Also,we can conclude that flow is processed through 4 parallel channels.

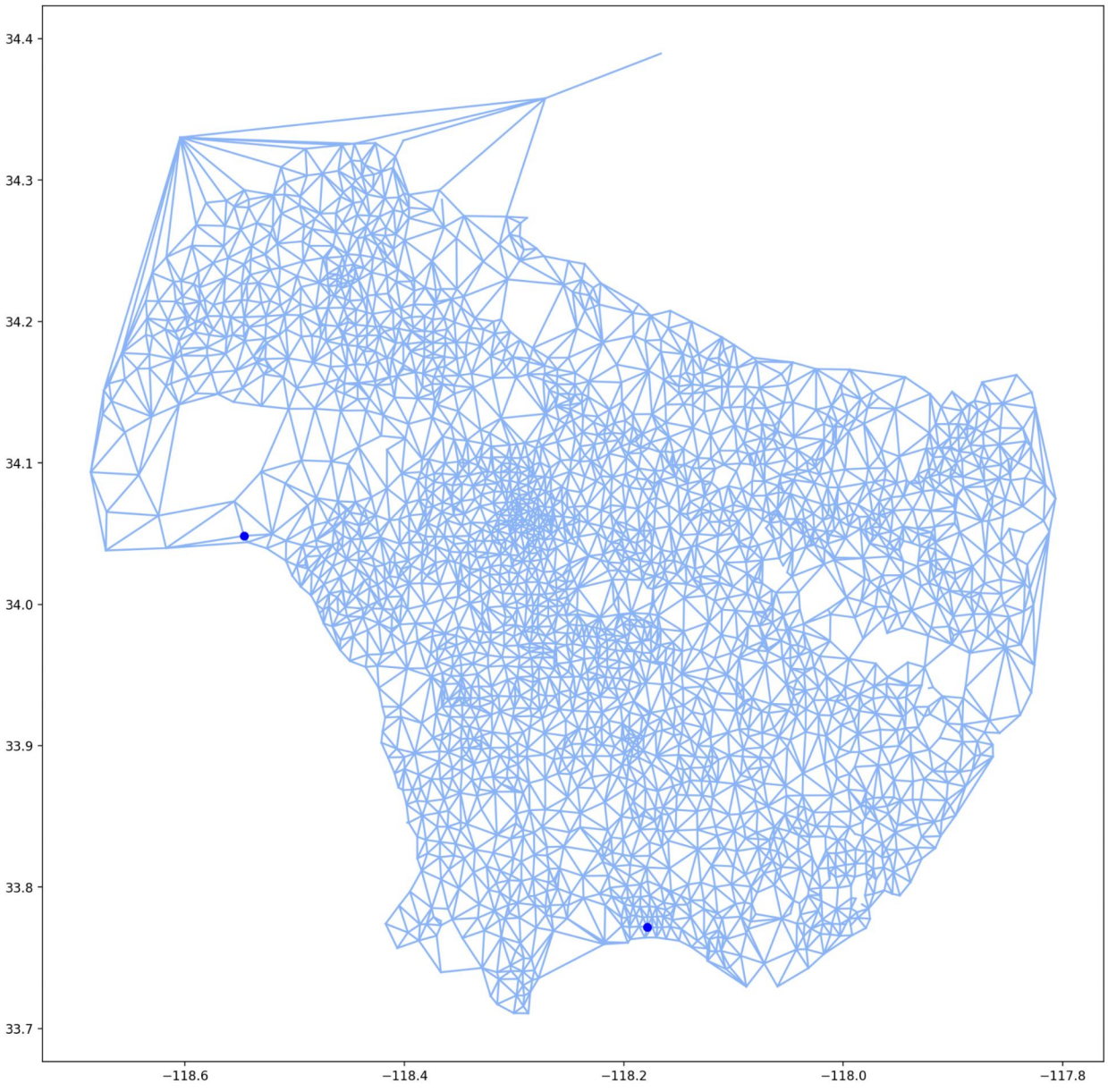


Malibu is marked with blue and red lines indicating the given coordinates at the assignment. Note that the degree of malibu is 4.



Long Beach is marked with blue and red lines indicating the given coordinates at the assignment. Note that the degree of Long Beach is 6.

Unzoomed road map:

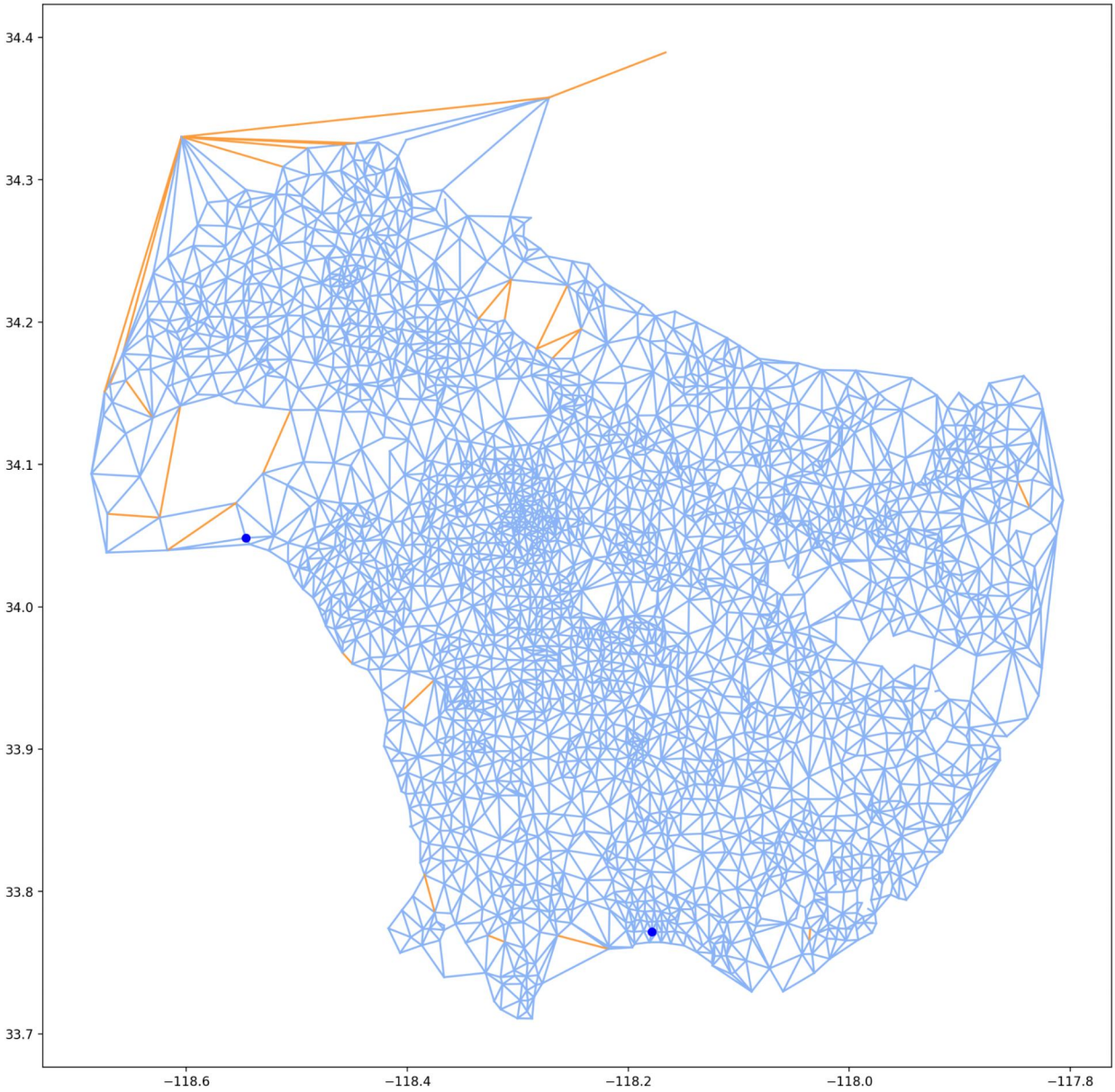

34.4
34.3
34.2
34.1
34.0
33.9
33.8
33.7
−118.6
−118.4
−118.2
−118.0
−117.8

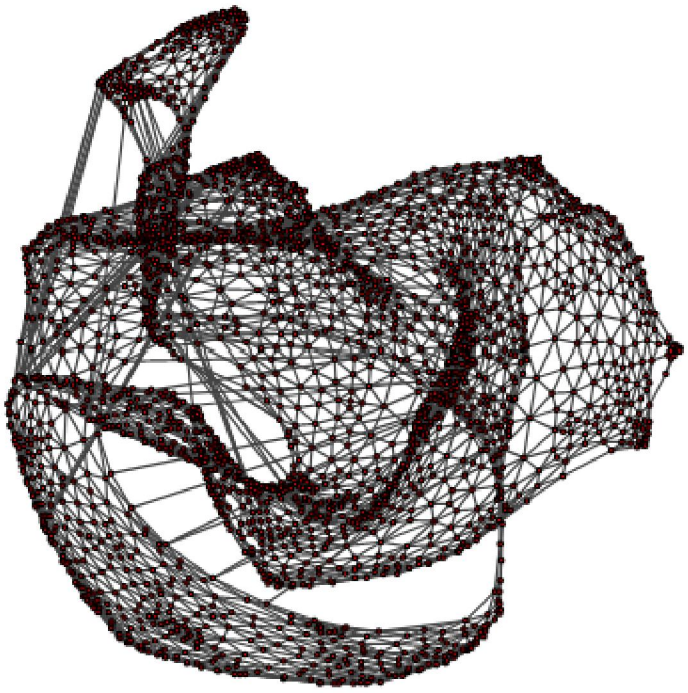
16 Question 16 **5 / 5**

✓ **- 0 pts** *Correct*

Question 17

In this part, we have applied thresholding on time of the road to remove remaining virtual roads. Removed roads are marked with orange. As we can see from the below graph, some of the long roads are removed. It is important to note that all long roads are not virtual roads, however most of the long roads are introduced by the Delaunay triangularization. Therefore, we can say that the process has partially worked to remove bad roads. Also, it is important to choose the threshold value which was set to be 800 seconds.Choosing a low threshold will eliminate most of the edges and choosing a large threshold will not eliminate any edges. Therefore, thresholding has a trade off on removing roads.





We have also plotted the obtained pruned graph using the graph plotting.

Question 18

In this part, using the capacities calculated in the Q15, we have done max-flow analysis to find the maximum flow from Malibu to Long Beach. We have also used the pruned graph to see the difference between the results obtained at Q16.

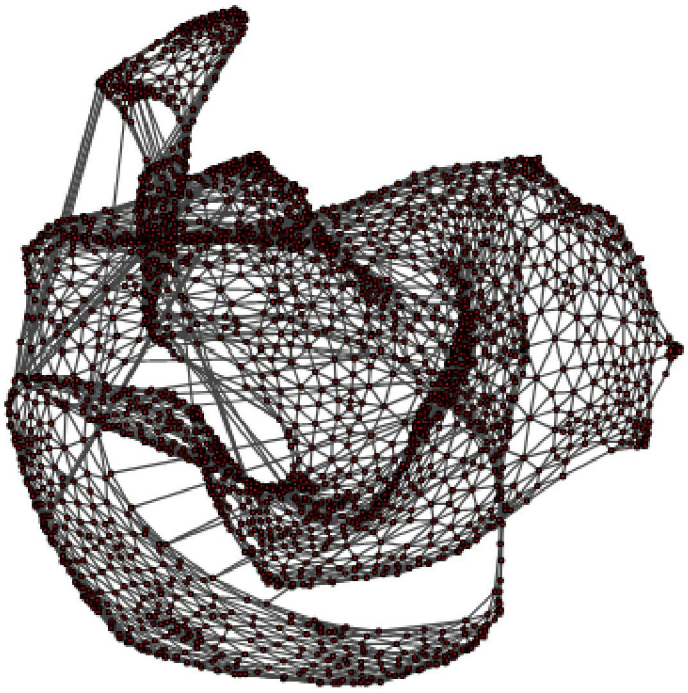
Maximum numbers of cars that can commute per hour from Malibu to Long Beach is found as 10898. The exact coordinates were not available in the road map of $G_{\Delta,pruned}$, therefore we have used the closest node to the given coordinates for both locations.

Number of edge-disjoint paths is found to be 4. Degree of node Malibu is 4, degree of node Long Beach is 6. Number of edge-disjoints should be less than or equal to the min(degree Malibu, degree Long Beach) = 4. Therefore, they match. Also,we can conclude that flow is processed through 4 parallel channels.

There is not much difference between pruned graphs except the max flow has decreased a little due to the fact that edge disjoint paths are slightly modified due to removed edges. Number of edge-disjoint paths and degrees are the same with Q16.

17 Question 17 **5 / 5**

✓ **- 0 pts** *Correct*



We have also plotted the obtained pruned graph using the graph plotting.

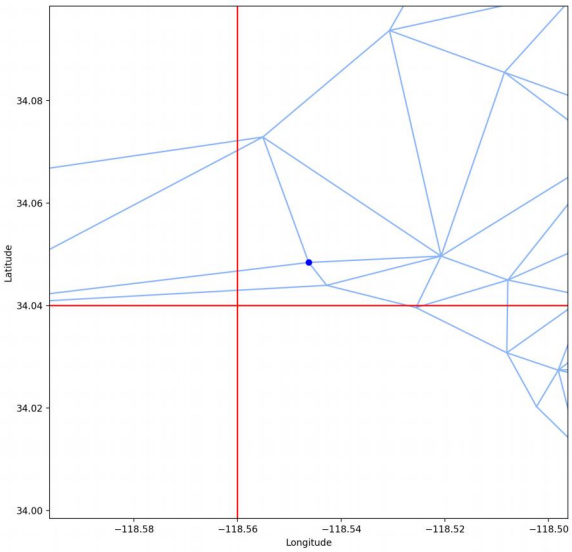
Question 18

In this part, using the capacities calculated in the Q15, we have done max-flow analysis to find the maximum flow from Malibu to Long Beach. We have also used the pruned graph to see the difference between the results obtained at Q16.

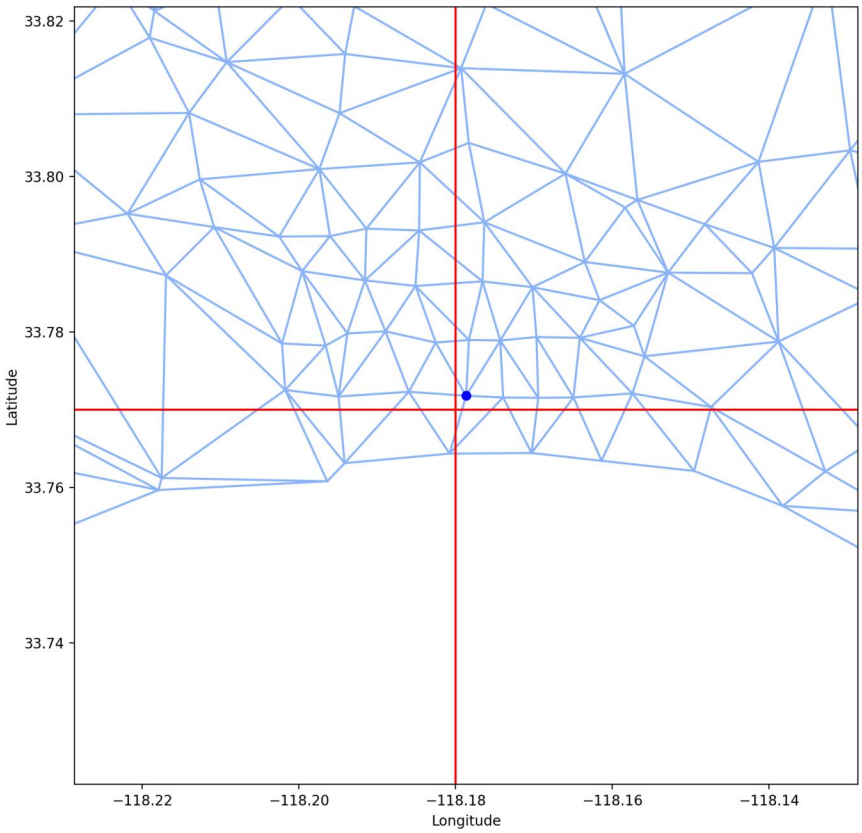
Maximum numbers of cars that can commute per hour from Malibu to Long Beach is found as 10898. The exact coordinates were not available in the road map of $G_{\Delta,pruned}$, therefore we have used the closest node to the given coordinates for both locations.

Number of edge-disjoint paths is found to be 4. Degree of node Malibu is 4, degree of node Long Beach is 6. Number of edge-disjoints should be less than or equal to the min(degree Malibu, degree Long Beach) = 4. Therefore, they match. Also,we can conclude that flow is processed through 4 parallel channels.

There is not much difference between pruned graphs except the max flow has decreased a little due to the fact that edge disjoint paths are slightly modified due to removed edges. Number of edge-disjoint paths and degrees are the same with Q16.



Malibu is marked with blue and red lines indicating the given coordinates at the assignment. Note that the degree of malibu is 4.



Long Beach is marked with blue and red lines indicating the given coordinates at the assignment. Note that the degree of Long Beach is 6.

18 Question 18 **3 / 3**

✓ **- 0 pts** *Correct*

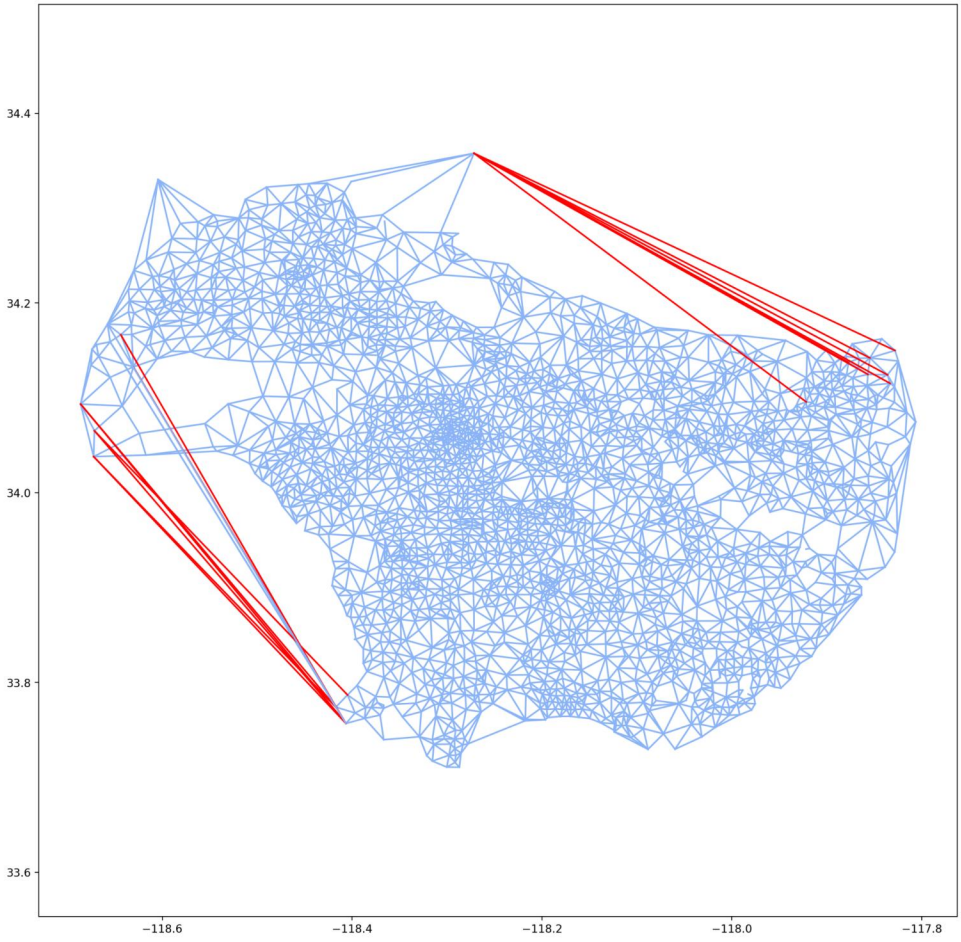
Question 19

Now the graph has been trimmed, we implement the first strategy, that being to construct 20 new roads based on the difference of the shortest traveling distance and the straight line distance between two points, the extra distance. The shortest path between points was calculated via the dijkstra algorithm.

The node pairs with the highest extra distance values are as such:

[[1860, 2416], [1699, 1783], [382, 2419], [1699, 1860], [1783, 2413], [1783, 144], [1783, 430], [1860, 2413], [43, 2419], [144, 1860], [386, 2419], [1860, 430], [1782, 2416], [390, 2419], [391, 2419], [388, 2419], [385, 2419], [1670, 1783], [2171, 2419], [1783, 983]]

A graph highlighting the newly constructed roads is below.



The time complexity of Dijkstra's shortest path algorithm is O(ElogV), where E is the number of edges and V is the number of vertices. But since we do this for every node, the complexity becomes O(VElogV), where V = 2647 and E = 7668.

19 Question 19 **14 / 15**

✓ **- 1 pts** *time complexity slighltly wrong*

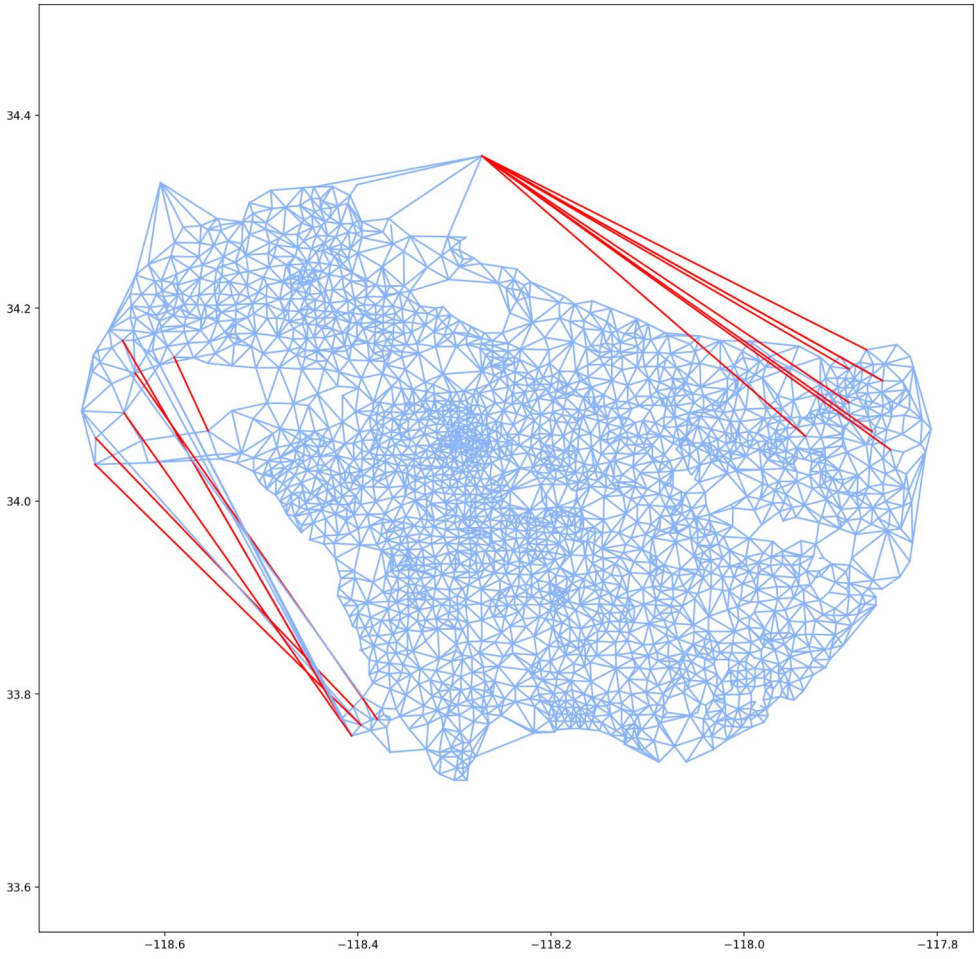
Question 20

We repeat the previous strategy with a slight tweak. In the real world, demand for traveling between some points will be higher than others. To simulate this, we multiply the extra distances found before by a random integer between 1 and 1000 and repick the top 20 pairs.

The node pairs with the highest frequency adjusted extra distance values are as such:

[[383, 2419], [2052, 2419], [1860, 1670], [1782, 2416], [388, 2419], [2057, 2419], [384, 2419], [989, 1510], [231, 2419], [2413, 1859], [1783, 982], [1670, 1783], [44, 2419], [2072, 2419], [144, 1881], [1699, 1859], [1781, 430], [1860, 984], [1860, 2412], [1783, 984]]

A graph highlighting the newly constructed roads is below.



The complexity order remains the same, being O(VElogV), though we have to perform an additional V^2 multiplication.

20 Question 20 **5 / 5**

✓ **- 0 pts** *Correct*

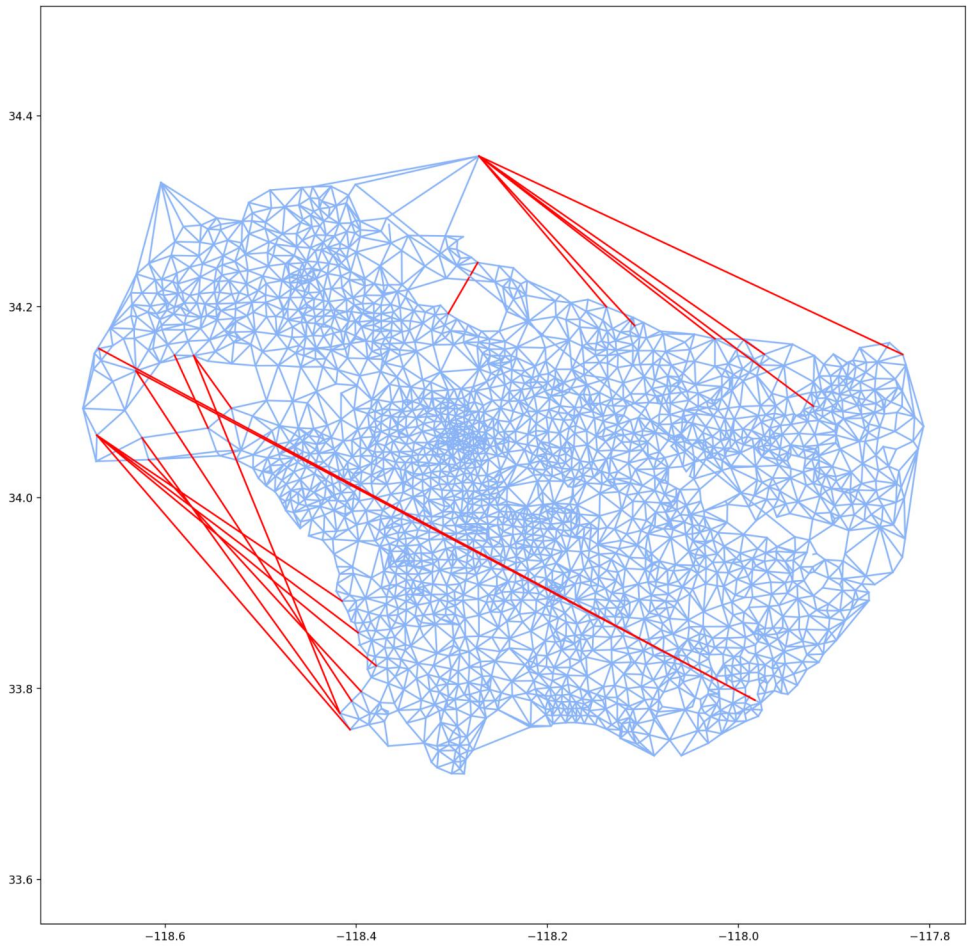
Question 21

For strategy 3, we take an iterative approach. We calculate the extra distances for all the pairs in the graph and construct a one road corresponding to the highest valued pair. We then recalculate the shortest paths and continue in that fashion for a total of 20 iterations.

The node pairs are as such:

[[1860, 2416], [382, 2419], [2171, 2419], [144, 1782], [1783, 2417], [118, 2419], [989, 1510], [22, 2419], [2247, 2419], [2414, 2619], [144, 2619], [1876, 2416], [988, 1783], [429, 2416], [1700, 1781], [121, 689], [2241, 2419], [1704, 2416], [988, 1678]]

A graph highlighting the newly constructed roads is below.



The time complexity of the process is essentially that for strategy 1, but on each new iteration, we add an edge. Thus the time complexity for iteration 1 is O(VElogV), but O(V(E+1)logV) for the second iteration, where E and V are the number of edges and vertices in the base trimmed graph.

21 Question 21 **8 / 10**

✓ **- 2 pts** *Partial credit for time complexity*

Question 22

For strategy 4, rather than creating roads based on extra distance, we create them based on extra time. Using the Dijkstra algorithm, we can find the shortest travel time between every point on the graph. We subtract the time it would take to travel the euclidean distance between the points at the average velocity traveled on the shortest path to get the extra time. We sort for the largest values.

The node pairs generated are as follows:

[2464, 1682], [2527, 2415], [2412, 452], [2578, 2415], [2578, 1682] [968, 2052], [2475, 2414], [2578, 2413], [2527, 2412], [968, 2050], [2578, 2412], [968, 2152], [2475, 772], [968, 2143], [968, 2470], [968, 1854], [968, 2462], [968, 2054], [968, 2051], [968, 2065]

A graph highlighting the newly constructed roads is below.

The time complexity, similar to question 19 is O(ElogV).

22 Question 22 **9 / 10**

✓ **- 1 pts** *Partially wrong time complexity*

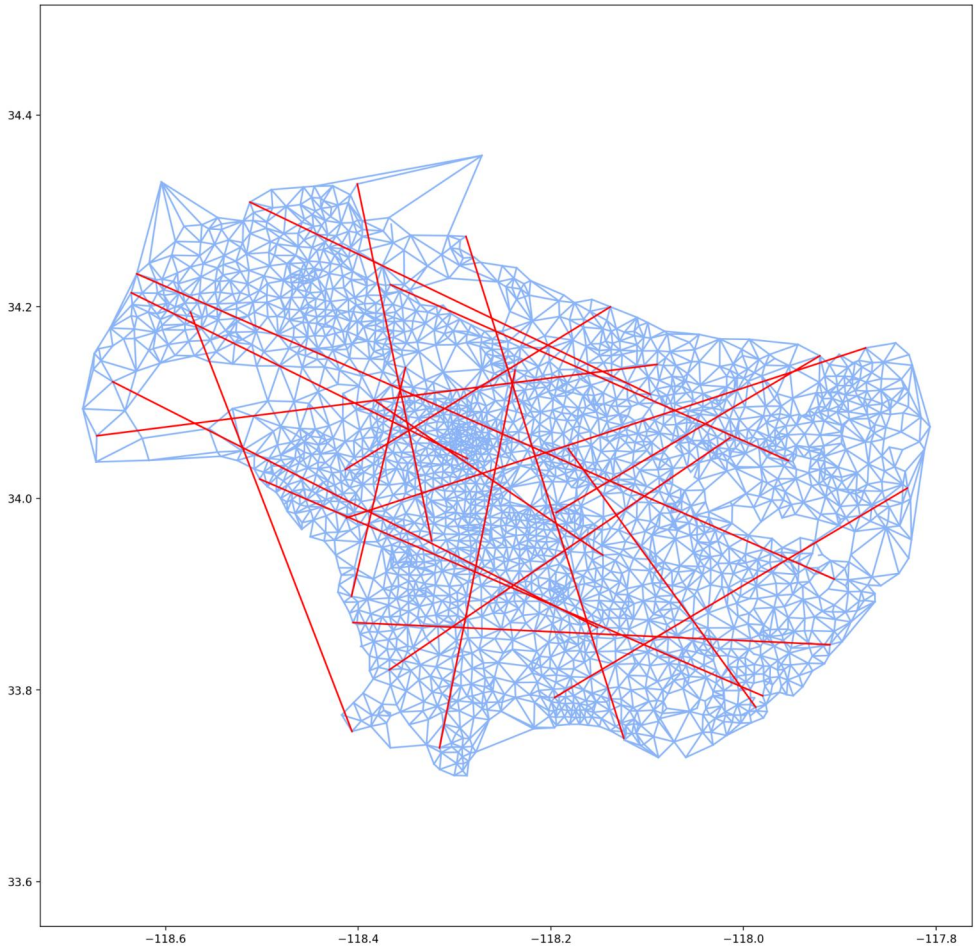
Question 23

For strategy 5, we apply the exact same method as in question 21, replacing extra distance as the metric with extra time.

The node pairs generated are as follows:

[[2464, 1682], [1598, 383], [1827, 2416], [677, 1684], [968, 1860], [2542, 1941], [833, 552], [2037, 305], [1346, 962], [277, 1134], [574, 145], [2425, 426], [476, 2128], [614, 255], [1556, 2241], [42, 447], [369, 1647], [2545, 2303], [1914, 734], [1702, 2139]]

A graph highlighting the newly constructed roads is below.



As it is not addressed in the following analysis section, the results of this strategy are likely the best so far. We have used the ultimate goal metric, reducing travel time, in our assessment of roads to build, while also minimizing redundancy. Seeing the web-like nature of the solution, it appears to be tackling many problems at once, slicing through areas of high traffic both east to west and north to south. The fact that it looks the most similar to the highway system we have currently in place is telling in this aspect.

The time complexity, similar to question 21 increases as edges are added. The time complexity for iteration 1 is O(VElogV), but O(V(E+1)logV) for the second iteration, where E and V are the number of edges and vertices in the base trimmed graph.

Question 24

a) Looking at the results for strategy 1, we see the newly constructed roads created acting as bridges connecting points across areas with no roads. This makes sense considering that extra distance is the metric we are using. It should be noted that an obvious problem is that some roads go over terrain that cannot simply be built on, a clear flaw in this method. Traveling between these points would require a path that significantly deviates from the optimal straight line path, maximizing the extra distance. When we look at strategy 2 we see similar results, expected since they both operate on the same basic algorithm. However, the points adjoined are less clustered together with less redundancy. Some similar extra distance values from strategy one were likely separated via the frequency multiplier, opening up the possibility for other roads to make it to the top 20. Overall it seems that strategy 2 works better and builds somewhat less redundant roads, though it would likely perform much better if an empirical frequency multiplier was used.

b) The main problem with strategy 1 was that the built roads were very redundant. Strategy 3 shows significant improvement on this. Looking at the top half of the graph, we see that the many connections from the singular vertex are now more spread out and provide better coverage of the area. We see similar results in the lower half. With this strategy we also see the emergence of smaller connections across pocket sizes gaps in the road map. Less redundancy has opened up room for these connections. Strategy 3 is clearly superior. It has better coverage and more intuitive connections for what this algorithm is trying to accomplish.

c) Now replacing extra distance with extra time, we expect to see an emphasis on lowering travel time as opposed to travel distance. Average traffic will be heavy in highly populated areas. Looking at the newly constructed roads we see that they seem to run east to west. This makes sense. The roads constructed in strategy 1 provide shortcuts along regions with no roads, but they do so for stretches along the coast that are generally not overly populated and do not experience heavy rush hour traffic, such as near downtown LA and major inland connective highways. It then makes sense that the algorithm would construct roads to bridge the areas that need to otherwise drive through these areas. Although both suffer redundancy issues, strategy 4 targets time spent driving, which is ultimately what we want to minimize, making it the better strategy.

d) i) Dynamic appears to be the optimal strategy. Static makes little to no sense. The resulting roads generated from this road have heavy redundancy. Since resources are limited, we want to maximize the total extra distance minimized and introducing a road right next to another one does not accomplish this. Taking into account a new road's effects on the extra distances in the road map does. Although more time complex, time complexity will likely play a lesser role in the months or years long road planning process.

23 Question 23 **9 / 10**

✓ **- 1 pts** *time complexity slighltly wrong*

The time complexity, similar to question 21 increases as edges are added. The time complexity for iteration 1 is O(VElogV), but O(V(E+1)logV) for the second iteration, where E and V are the number of edges and vertices in the base trimmed graph.

Question 24

a) Looking at the results for strategy 1, we see the newly constructed roads created acting as bridges connecting points across areas with no roads. This makes sense considering that extra distance is the metric we are using. It should be noted that an obvious problem is that some roads go over terrain that cannot simply be built on, a clear flaw in this method. Traveling between these points would require a path that significantly deviates from the optimal straight line path, maximizing the extra distance. When we look at strategy 2 we see similar results, expected since they both operate on the same basic algorithm. However, the points adjoined are less clustered together with less redundancy. Some similar extra distance values from strategy one were likely separated via the frequency multiplier, opening up the possibility for other roads to make it to the top 20. Overall it seems that strategy 2 works better and builds somewhat less redundant roads, though it would likely perform much better if an empirical frequency multiplier was used.

b) The main problem with strategy 1 was that the built roads were very redundant. Strategy 3 shows significant improvement on this. Looking at the top half of the graph, we see that the many connections from the singular vertex are now more spread out and provide better coverage of the area. We see similar results in the lower half. With this strategy we also see the emergence of smaller connections across pocket sizes gaps in the road map. Less redundancy has opened up room for these connections. Strategy 3 is clearly superior. It has better coverage and more intuitive connections for what this algorithm is trying to accomplish.

c) Now replacing extra distance with extra time, we expect to see an emphasis on lowering travel time as opposed to travel distance. Average traffic will be heavy in highly populated areas. Looking at the newly constructed roads we see that they seem to run east to west. This makes sense. The roads constructed in strategy 1 provide shortcuts along regions with no roads, but they do so for stretches along the coast that are generally not overly populated and do not experience heavy rush hour traffic, such as near downtown LA and major inland connective highways. It then makes sense that the algorithm would construct roads to bridge the areas that need to otherwise drive through these areas. Although both suffer redundancy issues, strategy 4 targets time spent driving, which is ultimately what we want to minimize, making it the better strategy.

d) i) Dynamic appears to be the optimal strategy. Static makes little to no sense. The resulting roads generated from this road have heavy redundancy. Since resources are limited, we want to maximize the total extra distance minimized and introducing a road right next to another one does not accomplish this. Taking into account a new road's effects on the extra distances in the road map does. Although more time complex, time complexity will likely play a lesser role in the months or years long road planning process.

e) One problem with the method of strategy five is that many of the roads are extremely long and likely unfeasible for building. We could modify our algorithm to cap the amount of roads that can be taken along a shortest path and rule out road building for vertices separated by more than a given threshold of edges. Another idea could be to simply disqualify algorithm selected connections based on the euclidean distance of the road. Another problem is that suggested roads may breach areas of terrain that cannot be built on. We could trace the line of these roads and if they breach into no-go areas, disqualify them. These are not fundamental changes to the algorithm but postprocessing methods that may need to be used if these ideas were to be realized. Additionally, the idea of straight lines roads is crude and non-realistic. Certain areas may simply be harder to develop over and this would need to be taken into account if the top 20 roads are to be chosen. The average building coefficient could be formulated from the straight line path between two points, perhaps via an integration over a difficult heat map of some sort. This could then be factored into the sorting of the road.

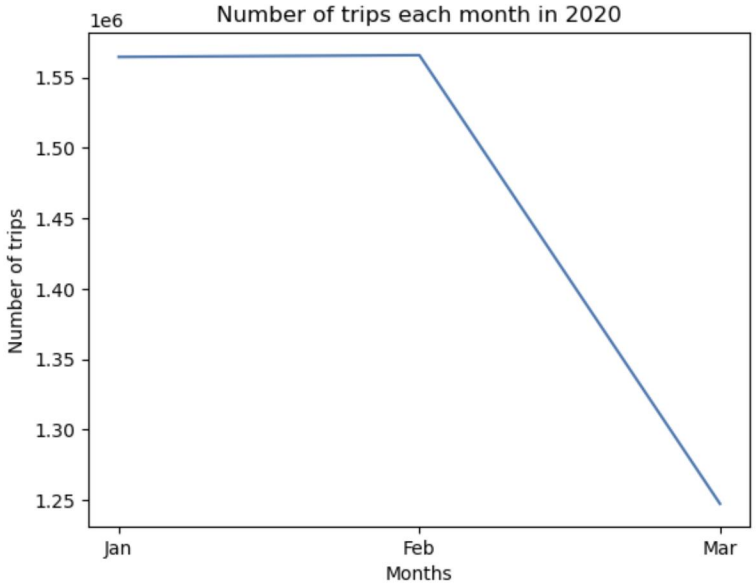
Task

In the task part, we have done three independent tasks to investigate the uber dataset for 2020 Quarter 1. Quarter 1 of 2020 has a drastic change by months due to COVID-19.

These tasks can be summarized as follows:

1) Investigation of number of trips and average time of trips and average speed of trips.
2) Investigation of the most visited places using the pagerank algorithm
3) Investigation of the closest coordinates to UCLA campus by the time criteria.

First, we have investigated the number of trips each month and their average time and speed.



24 Question 24 **20 / 24**

✓ **- 4 pts** *dynamic is better but should not be optimal*

e) One problem with the method of strategy five is that many of the roads are extremely long and likely unfeasible for building. We could modify our algorithm to cap the amount of roads that can be taken along a shortest path and rule out road building for vertices separated by more than a given threshold of edges. Another idea could be to simply disqualify algorithm selected connections based on the euclidean distance of the road. Another problem is that suggested roads may breach areas of terrain that cannot be built on. We could trace the line of these roads and if they breach into no-go areas, disqualify them. These are not fundamental changes to the algorithm but postprocessing methods that may need to be used if these ideas were to be realized. Additionally, the idea of straight lines roads is crude and non-realistic. Certain areas may simply be harder to develop over and this would need to be taken into account if the top 20 roads are to be chosen. The average building coefficient could be formulated from the straight line path between two points, perhaps via an integration over a difficult heat map of some sort. This could then be factored into the sorting of the road.

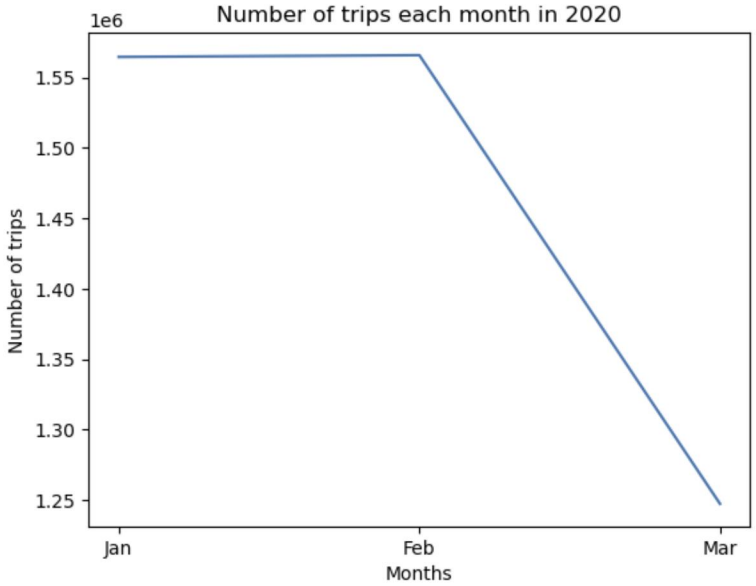
Task

In the task part, we have done three independent tasks to investigate the uber dataset for 2020 Quarter 1. Quarter 1 of 2020 has a drastic change by months due to COVID-19.

These tasks can be summarized as follows:

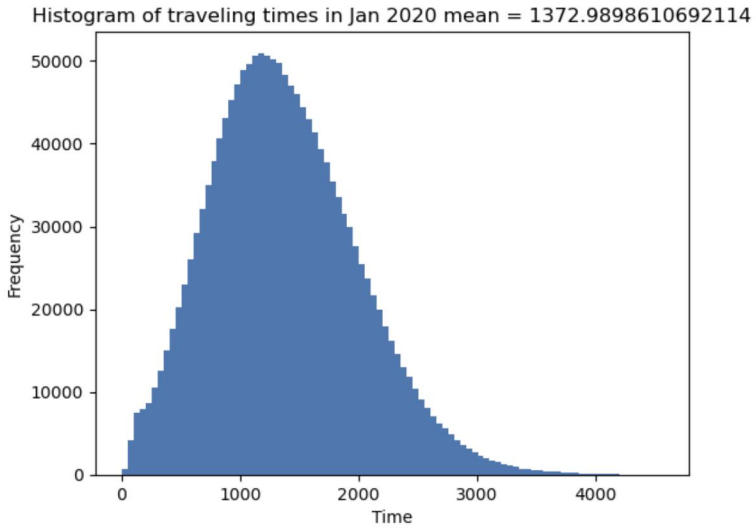
1) Investigation of number of trips and average time of trips and average speed of trips.
2) Investigation of the most visited places using the pagerank algorithm
3) Investigation of the closest coordinates to UCLA campus by the time criteria.

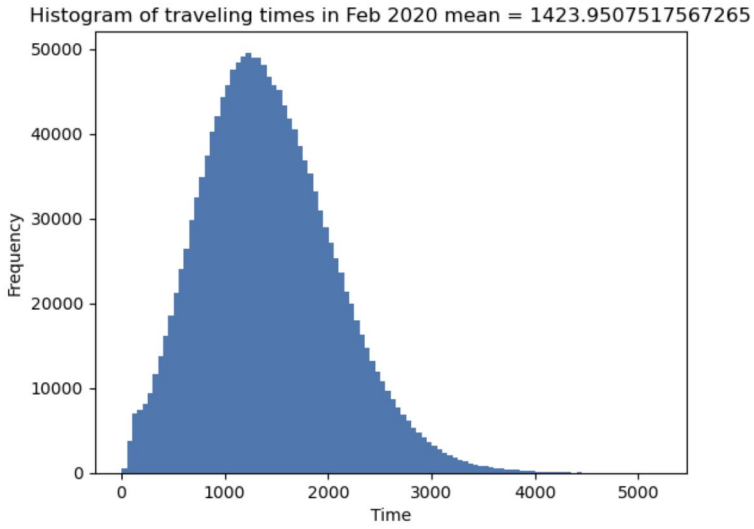
First, we have investigated the number of trips each month and their average time and speed.

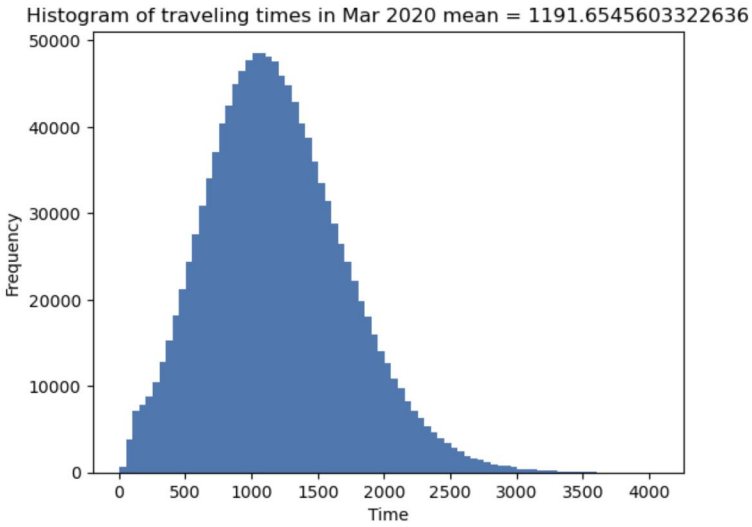


It is easy to observe that there is a drastic decrease in the number of trips in March 2020 due to COVID-19. People did not use Uber to commute. Number of trips can be easily calculated by counting the number of edges of the graph.

Next, we have obtained average trip times for each month.

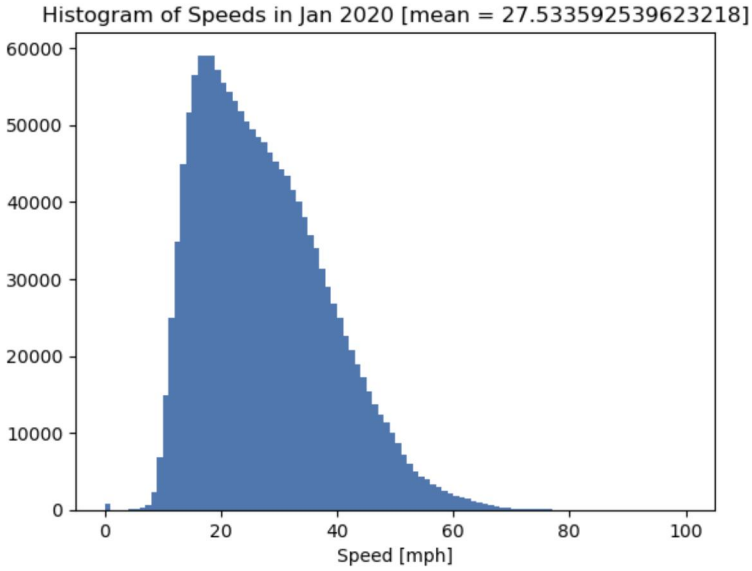


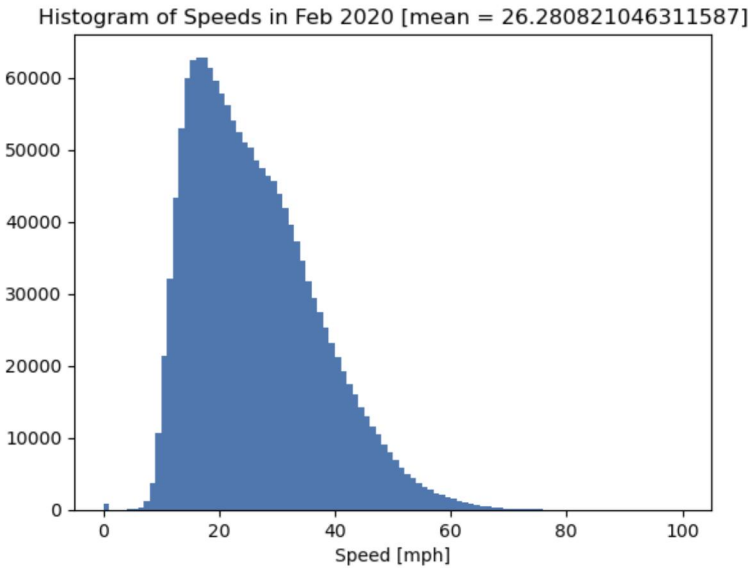


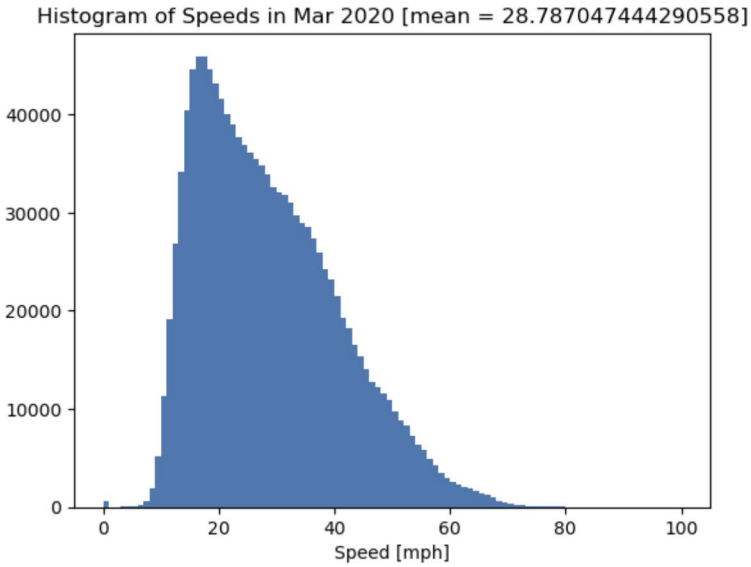


The distribution of time is similar to each graph except the mean of the graph. Average trip time has nearly decreased by 200 seconds from February to March since people did not want to take longer trips to get exposed to COVID-19.

Next, we have also investigated the average speed of trips to understand the dynamics of traffic at Quarter 1 2020.







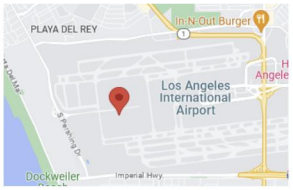
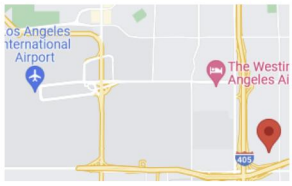
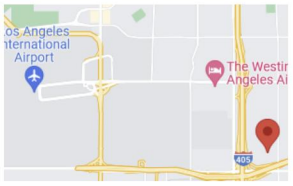
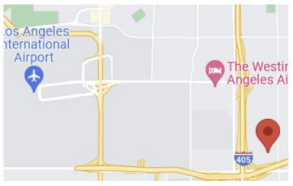
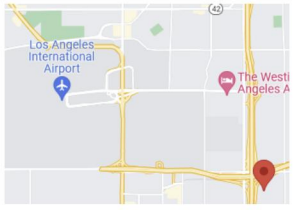
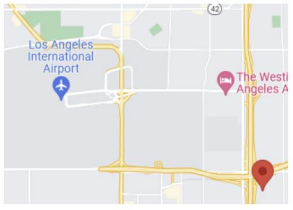
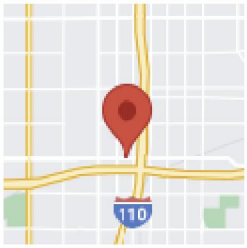
There is not much difference in the average speeds of trips since most of the trips take place in residential areas with the speed limit of 30. This is the main reason behind the skewness of the graph. Skewness is introduced when there is an external force affecting the distribution. In our case, external force is the law. We can slightly conclude that since there is less traffic in March 2020, the average speed of trips has slightly increased.

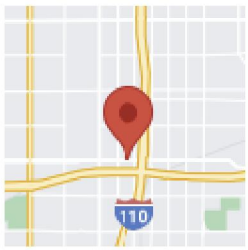
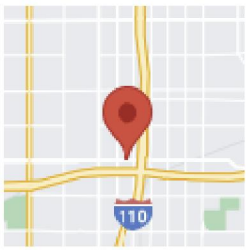
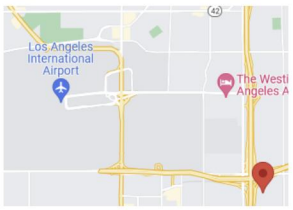
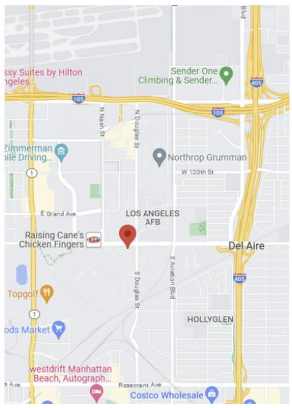
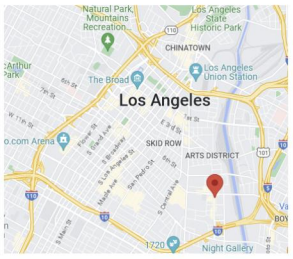
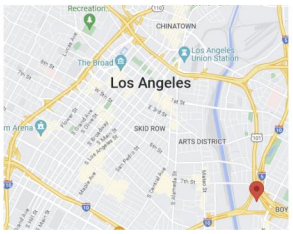
In the second task, we have investigated the most visited places in LA with Uber using the pagerank algorithm. Pagerank algorithm can be easily used to distinguish the most visited places in LA since it brings up the nodes that have the highest number of edges.

Most Visited Places	Jan 2020	Feb 2020	Mar 2020
1	**33.94087088,**	**33.94087088**	**33.94087088**

	-118.42336888	**-118.42336888**	**-118.42336888**
2	**33.93421103** **-118.36347711**	**33.93421103** **-118.36347711**	**33.93421103** **-118.36347711**
3	33.92776102 -118.36585774	33.92776102 -118.36585774	33.931878 -118.28427826
4	33.931878 -118.28427826	33.931878 -118.28427826	33.92776102 -118.36585774
5	33.91599813 -118.38430965	34.03412762 -118.23218029	34.03138969 -118.22114822

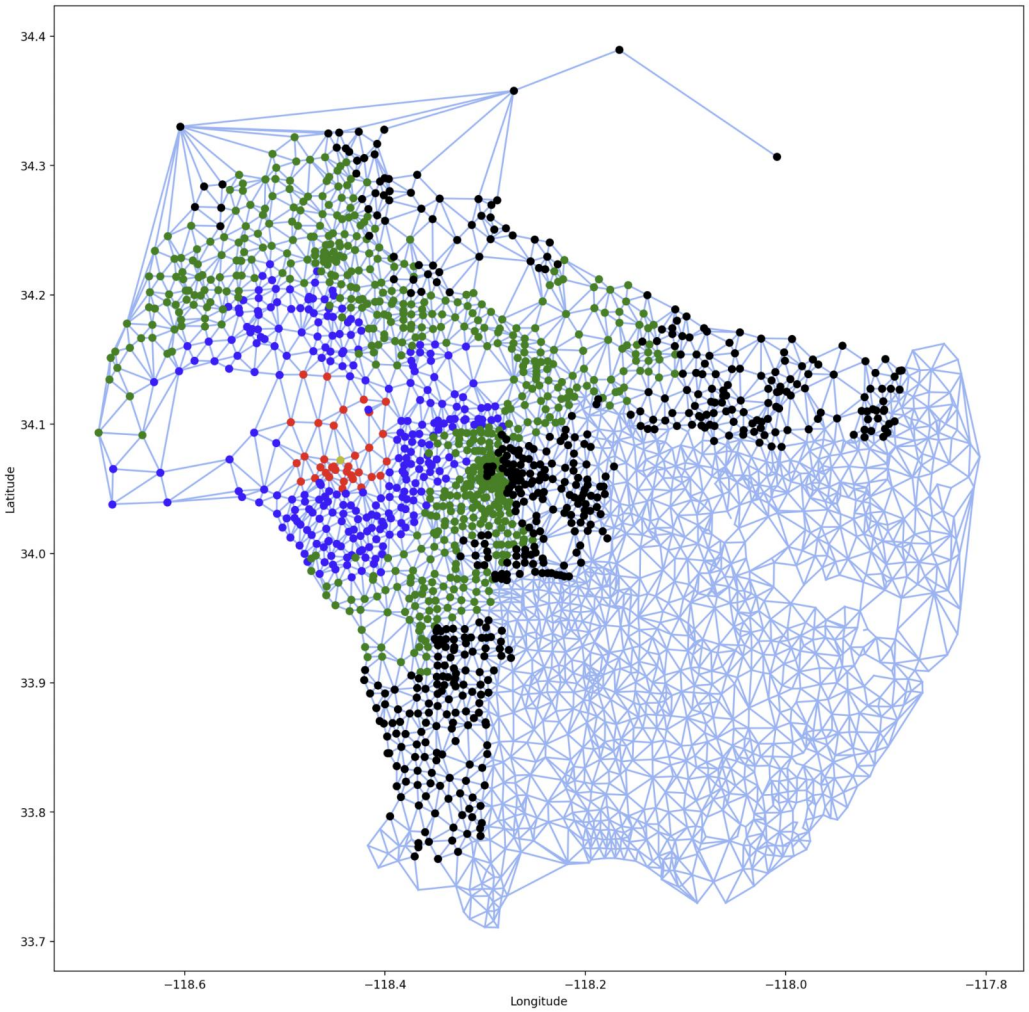
This table gives the coordinates of the most visited places, however it is wise to see the location of the coordinates using Google Maps.

Most Visited Places	Jan 2020	Feb 2020	Mar 2020
1			
2			
3			

4			
5			

From the results, we can see that Uber is mainly used for places that parking is not available such as LAX airport and LA downtown. Another important point to notice would be that most trips pass from the interstate highways which was expected. Also, most visited places did not change even though the number of trips has significantly decreased, since people are mandatory to take Uber to go to the airport.

In the last task, we thought that we would have benefited from a map where we can see the commute times to UCLA. Therefore, we have calculated the time of commute using the shortest path algorithm. Our model assumes that a commuter uses the shortest path UCLA which is the case for us. We have divided commute times to 4 categories which are. Locations that have commute time less than 10 minutes, 30 minutes, 60 minutes and 90 minutes. The graph can be improved by also selecting the time of starting to commute since commute times are highly affected by the time of the day and day of the month, however we have used the average of the month for the commute time.



- We have marked the coordinates of UCLA with yellow marker.
- Red marker indicates that the commute time from that location to UCLA is less than 10 minutes.
- Blue marker indicates that commute time from that location to UCLA is less than 30 minutes.
- Green marker indicates that commute time from that location to UCLA is less than 60 minutes.
- Black marker indicates that commute time from that location to UCLA is less than 90 minutes.
- If a position is not marked, the commute time from that location to UCLA is more than 90 minutes.

We can easily observe that in order to minimize commute time, one needs to stay in the Westwood area, Bel Air area or Beverly Hills area which are all expensive so blue markers should be also considered. Sawtelle and Santa Monica region relies on blue areas indicating that commuting will be less than 30 minutes. Therefore, it is wise to rent or buy a place in those areas, if you are a regular commuter to UCLA. The times are averages therefore does not reflect the commute time precisely, however they can show the comparison of regions.

25 Define Your Own Task **39 / 45**

✓ **- 0 pts** *Correct*

- 6 *Point adjustment*

Task Definition: 4/5, Creativity: 5/10, Success: 10/10, Methodology: 10/10, Completeness: 10/10