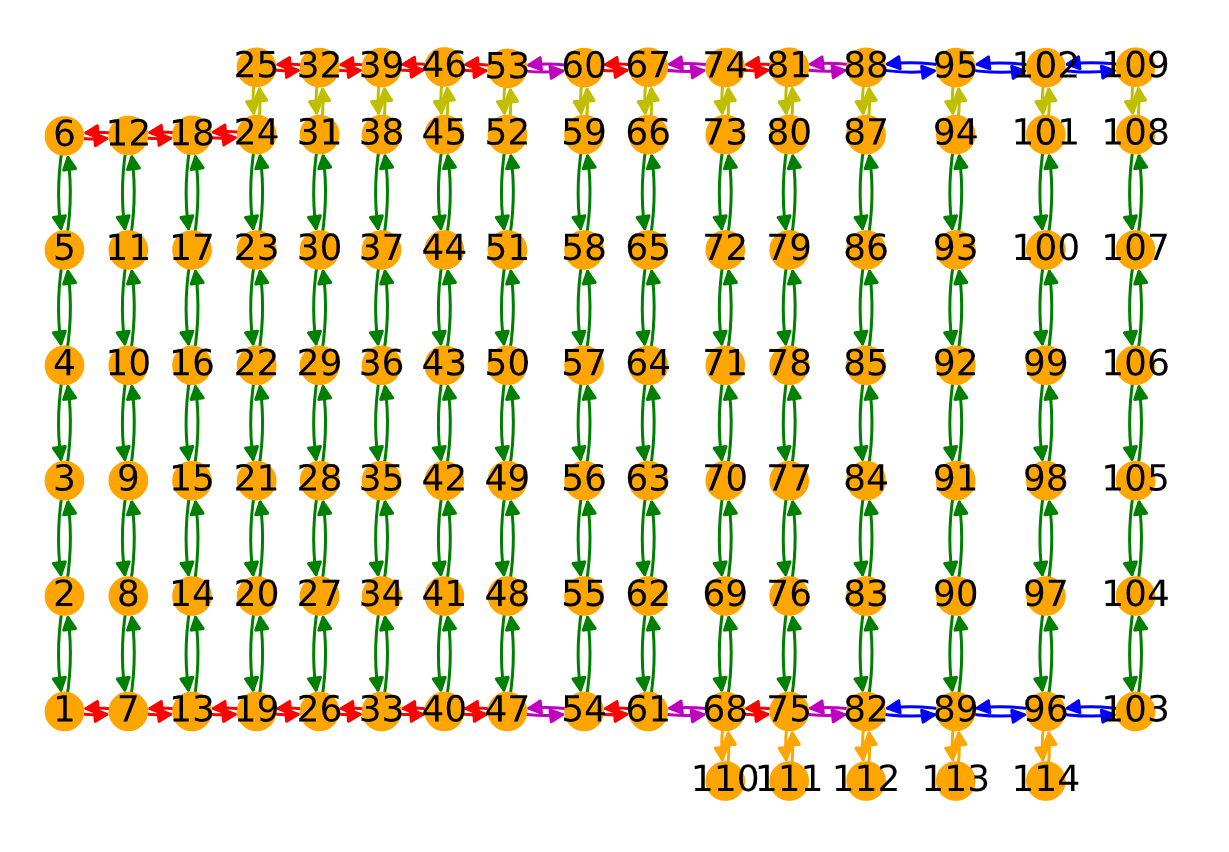
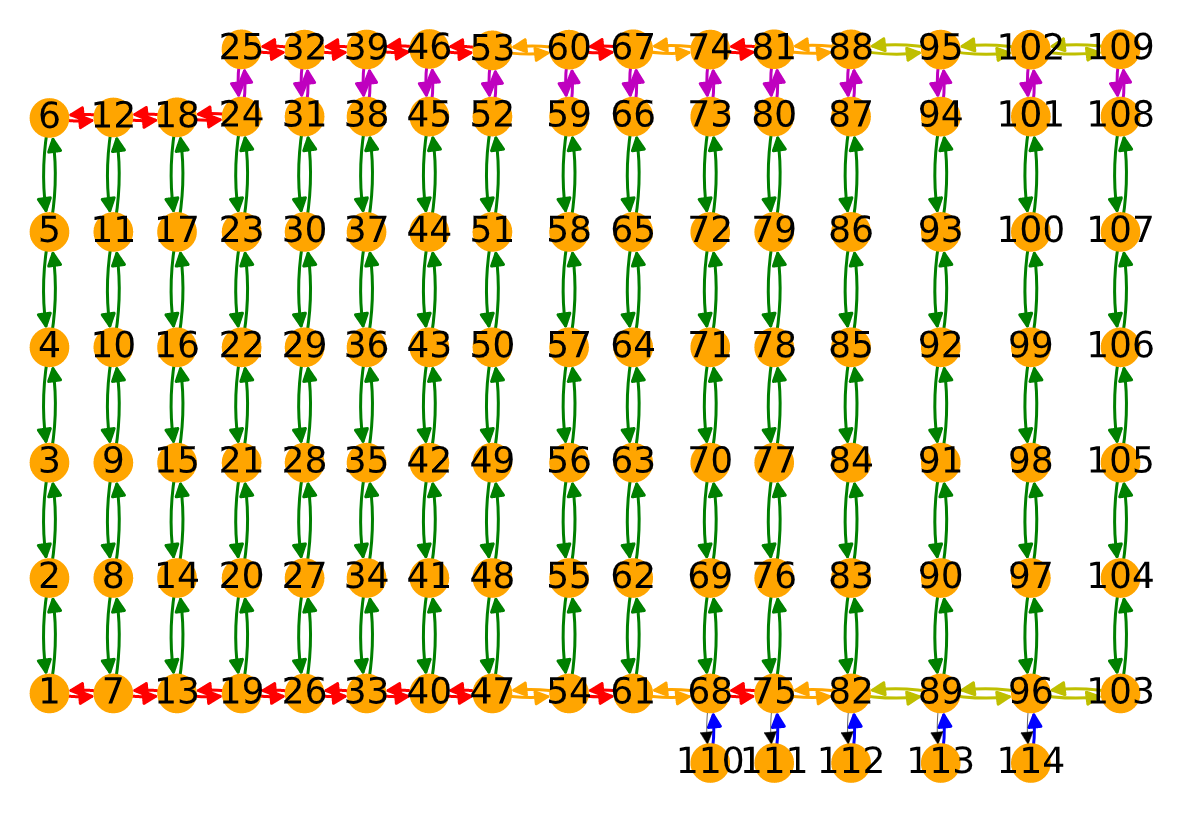
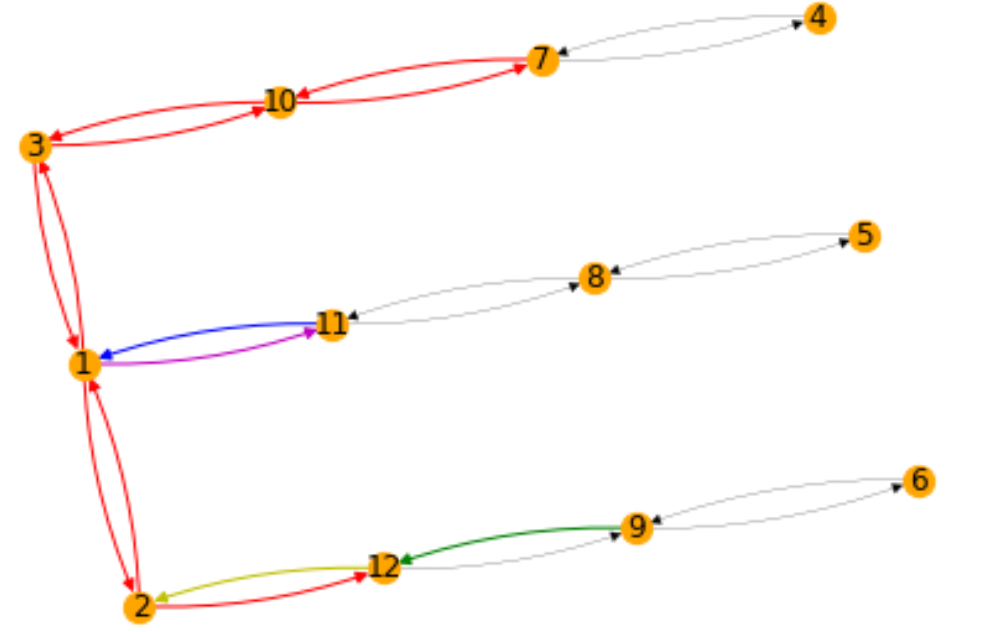
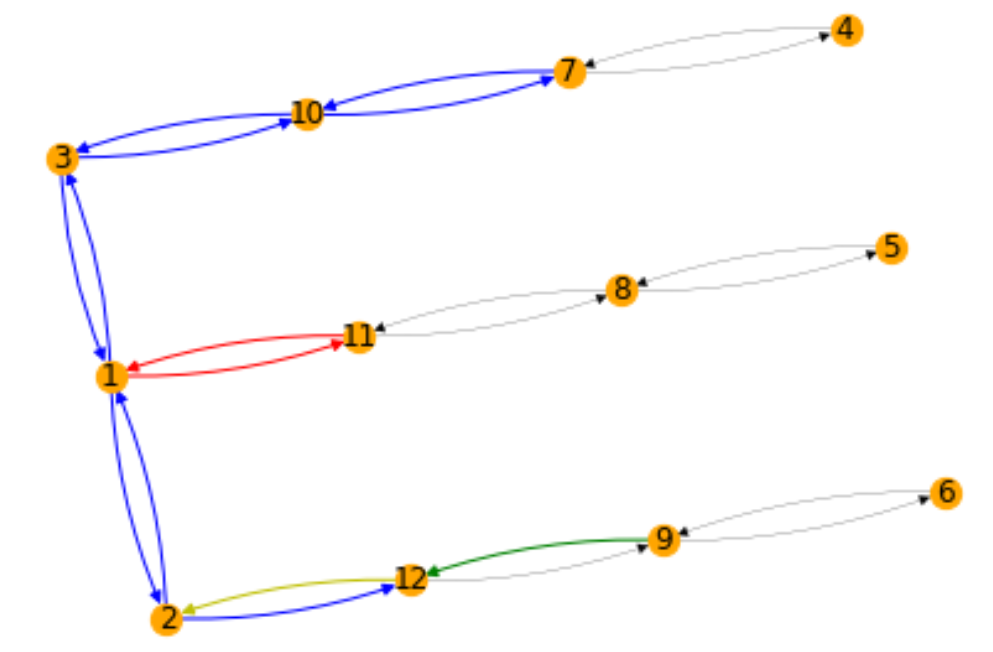
**Week 6 Writeup**

# RECAP: Week 5 Summary

1. **[MON]** – Weekly meeting. Run model-fitting comparison (based on KS & CVM) for 10 edges for 10 models
   1. Note: this used data from “walmart\_targeted” dataset
2. **[TUE]** – Finish PPT
3. **[WED]** – implement CRPS and test against library function. Look at other measures (DIC, WAIC, posterior predictive accuracy)
   1. Allows for an arbitrary cdf as the input
4. **[THU]** – Hierarchical Agglomerative clustering of edges by KS score and spatial similarity for “walmart\_random” dataset
   1. Use Silhouette score, Calinski-Harabasz (CH) Index index, Davies-Bouldin (DB) Index to determine optimum number of clusters
   2. Left – clustering by KS score. Optimum no. of clusters = 5 or 6
   3. Right – clustering by Edge length difference. Optimum no. of clusters = 5 or 6



1. **[FRI]** – Observe clustering as congestion increases for “blenheim\_targeted” dataset.
   1. **1 robot per edge** – 1 cluster
   2. **2 robots per edge** – 1 cluster
   3. **3 robots per edge** – 4 (left) or 5 (right) clusters



# Week 6 Plans

**Goal:** Automatically build edge durations for the whole map based on a subset of the available data. Test against remaining data

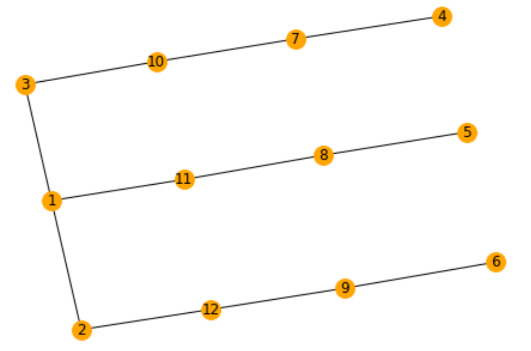
1. **[MON]** – Weekly meeting + updates
   1. Update congestion algorithm – don’t consider robots that are approaching/leaving the origin node.
   2. Try “random” datasets for Blenheim. Are the KS values between 2-12 and 12-2 actually appreciably different? Do the edge connections affect the clustering as the congestion increases?
2. **[TUE]** – Generalise between similar edges
   * 1. Try using one edge in the cluster to fit all other edges
     2. Try using a few datapoints from all edges in cluster to fit all edges in cluster
     3. Use CRPS / KS metric
3. **[WED]** – Is it possible to generalise between dissimilar edges? i.e. is there a mapping between parameters based on edge length, end nodes etc.
   1. Do no. of end nodes matter at all? i.e. is there a difference between robots starting from stationary vs already moving vs turning?
   2. Variable length edges. Can you transform between different edge lengths?
4. **[THU]** – STRANDS / more congestion: does this still work? What map and dataset would be interesting to try for purposes of generalisation? Random data collection with arbitrarily spaced edges – single robot. Have a look at other maps and layouts. Check STRANDS maps
5. **[FRI]** – How many datapoints do you need to get a good representation of an edge? Does this change if you use datapoints from other edges? How do you ensure that datapoints from the true edge count more towards Bayesian updates?
6. **[EXTRA]** – Automatically determine parameters for all edges based on clustering. Use robomaker live in a browser? Betty (STRANDS) vs Jackal (Warehouses)

**Questions:**

1. CRPS: for a vector new observations, are you meant to get a vector of corresponding CRPS scores or just take a mean?
2. How do you ensure that datapoints from the true edge count more towards Bayesian updates?
   1. Each new datapoint count multiple times? Then decay to 1 over time?
3. Maybe need more data for Blenheim? Merge datasets for random & targeted?

# MON: Additional Filtering & Blenheim Clustering

**Additional filtering:** we update the method for counting congestion by counting the number of robots (including the current robot) that are on any edge connected to the “target” node



E.g. for a robot moving from 11 to 1, we count the robots on edges

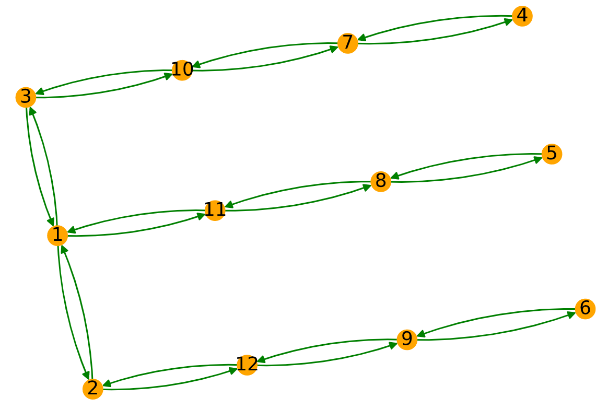
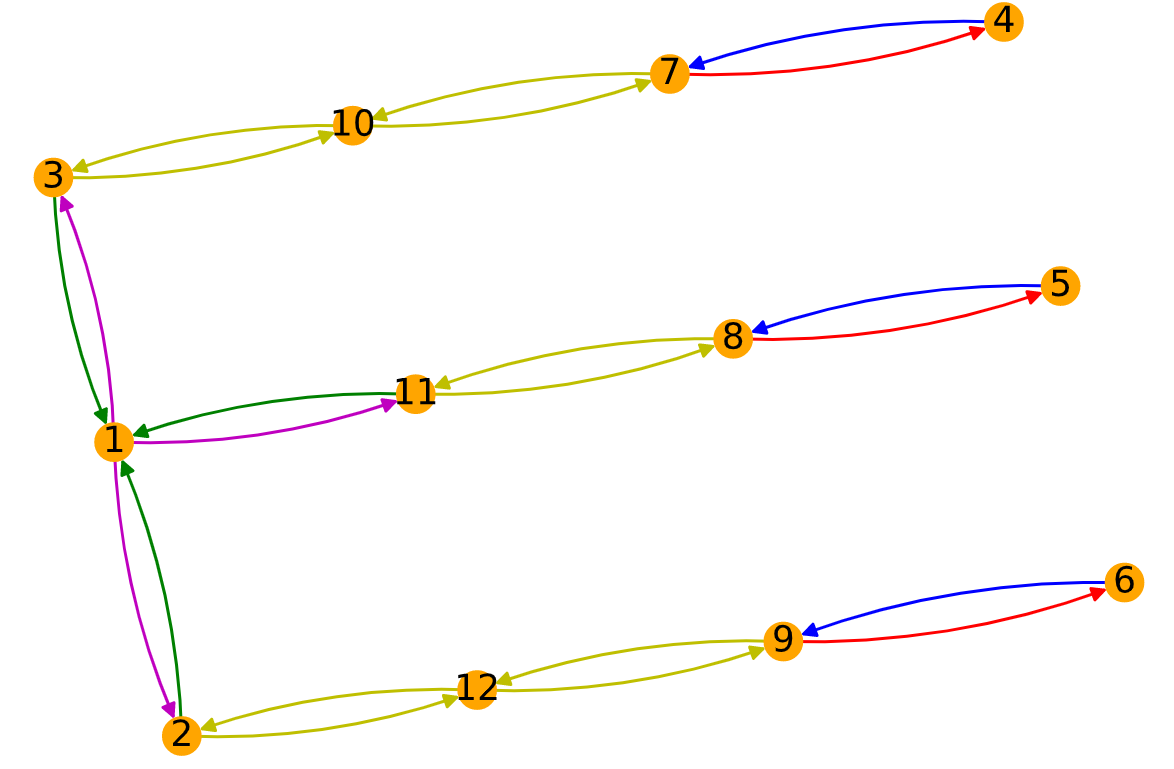
* 11 to 1
* 1 to 3
* 1 to 2

And the reverse edges

**For clustering, the main point is that there is only one cluster in the Blenheim dataset**

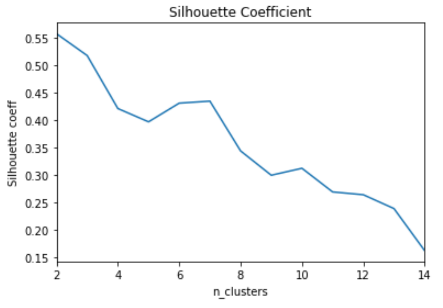
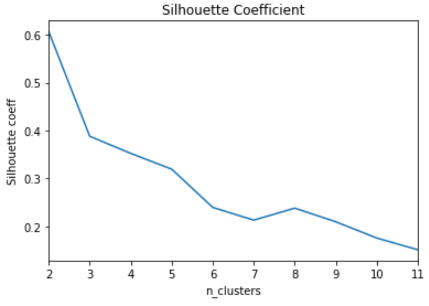
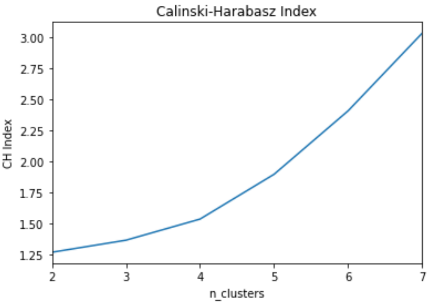
#### Blenheim – clustering by spatial characteristics

By edge length (1 cluster): By connections (5 clusters)

#### Blenheim Random dataset

N\_robots = 1 🡺 1 cluster N\_robots = 2 🡺 1 cluster N\_robots = 3 🡺 1 cluster

Note: high Silhouette coefficient & CH index is good. Low DB index is good

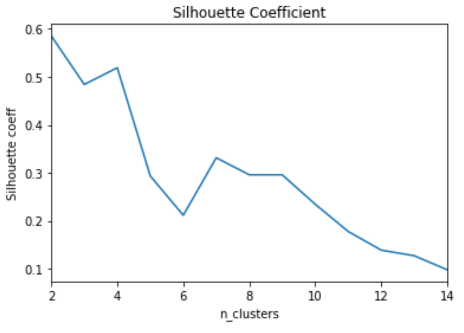
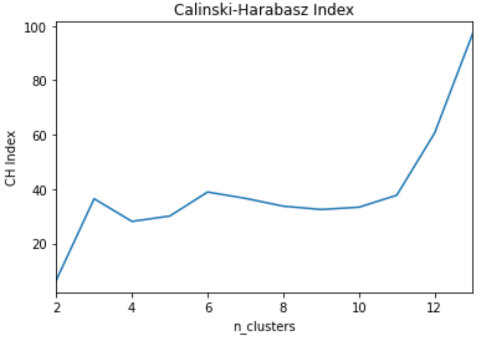
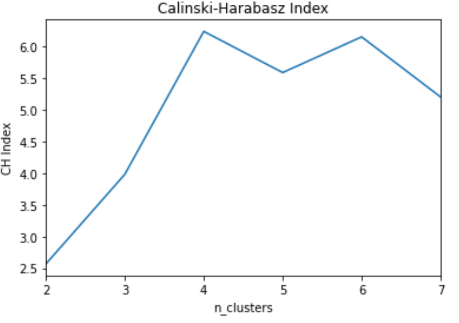
We do not have enough data for N\_robots = 3

We did not have enough data for n\_robots = 4.

* 6712 datapoints for n\_robots = 1
* 744 datapoints for n\_robots = 2
* 22 datapoints for n\_robots = 3
* 0 datapoints for n\_robots = 4

#### Blenheim Targeted dataset

N\_robots = 1 🡺 1 cluster N\_robots = 2 🡺 1 cluster N\_robots = 3 🡺 1 cluster

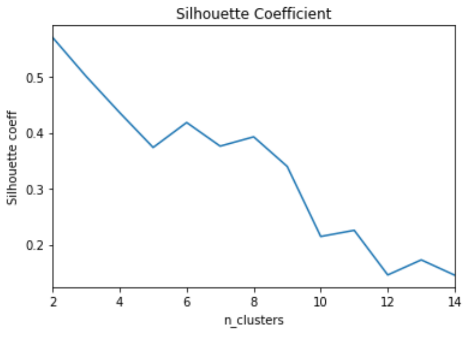
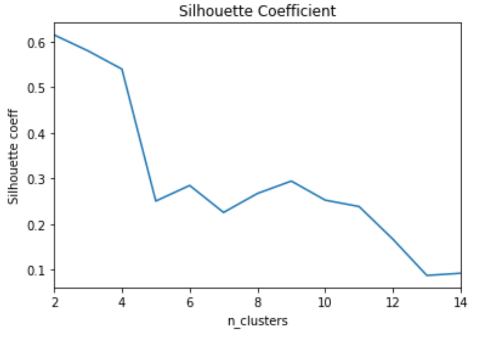
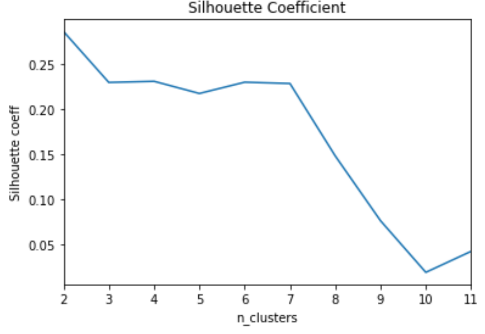
We did not have enough data for n\_robots = 4.

* 8567 datapoints for n\_robots = 1
* 1019 datapoints for n\_robots = 2
* 53 datapoints for n\_robots = 3
* 0 datapoints for n\_robots = 4

#### Blenheim Combined dataset

Use data from both datasets

N\_robots = 1 🡺 1 cluster N\_robots = 2 🡺 1 cluster N\_robots = 3 🡺 1 cluster

We did not have enough data for n\_robots = 4.

* 15279 datapoints for n\_robots = 1
* 1763 datapoints for n\_robots = 2
* 74 datapoints for n\_robots = 3
* 0 datapoints for n\_robots = 4

# TUE: Generalise between edges

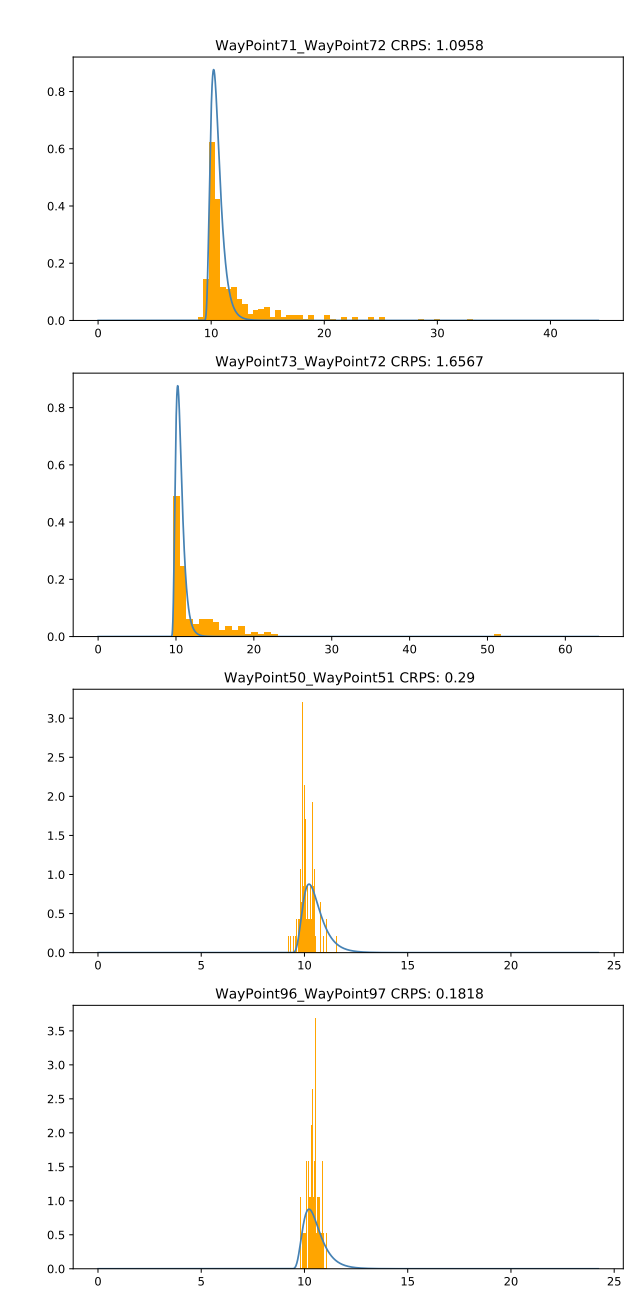
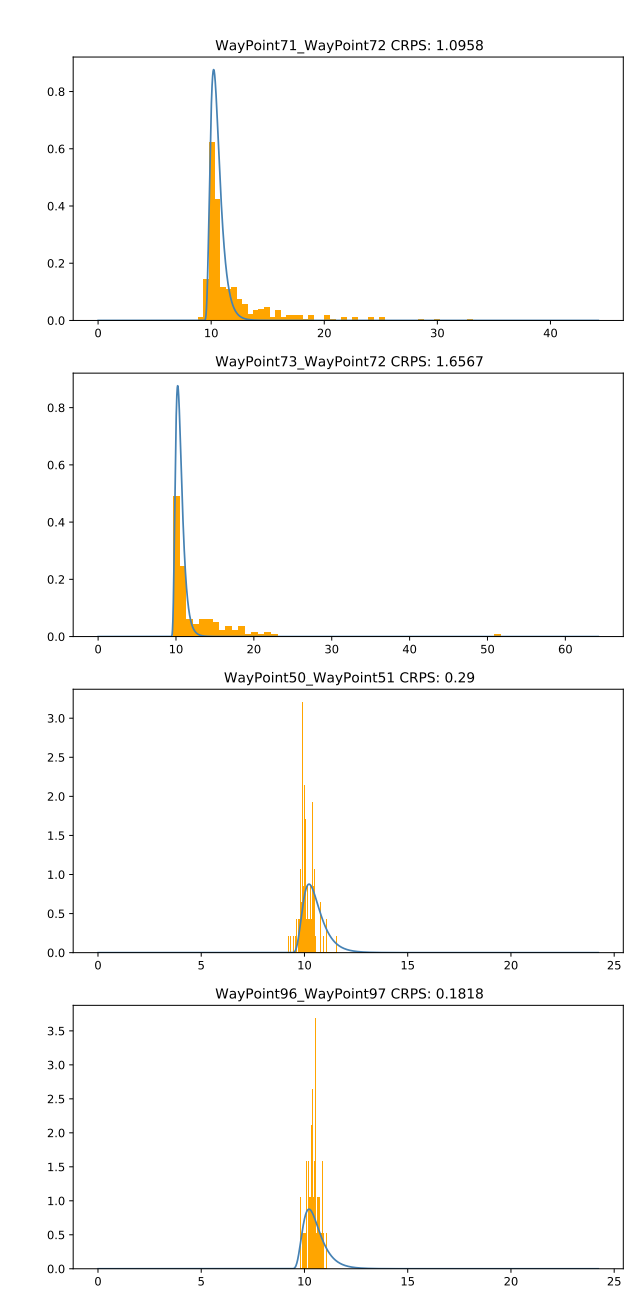
Two methods of generalising between edges within the same cluster for the Walmart map

1. Train lognormal parameters for one edge within the cluster and test using median CRPS score against data from another edge of the same cluster
   1. Stored as excel heatmap (“crps\_median\_clusterX.xlsx”)
2. Train parameters using 10 datapoints from each edge in the cluster and test using median CRPS score against new data from those edges
   1. Plotted the predicted lognormal curve (same for all tests) against histogram of new data for edges in cluster
   2. Stored as PDF (“mergefit\_median\_clusterX.xlsx”)

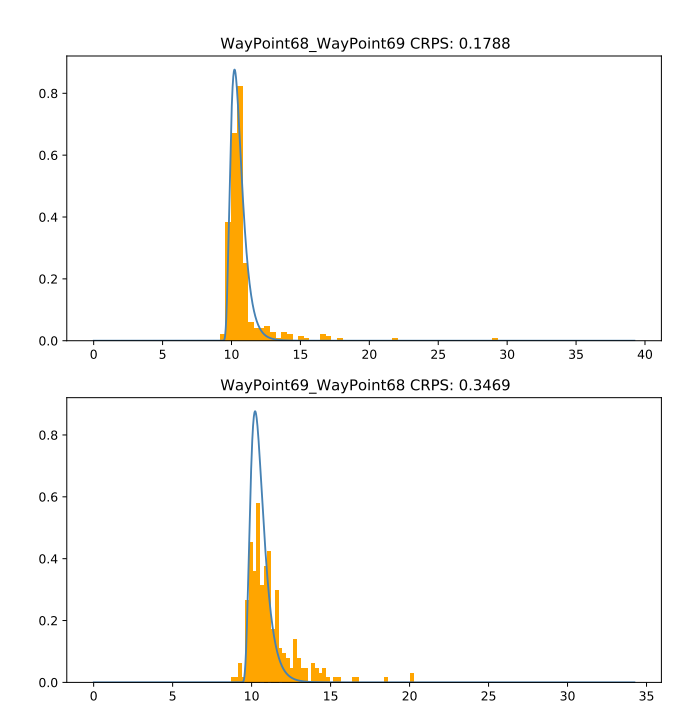
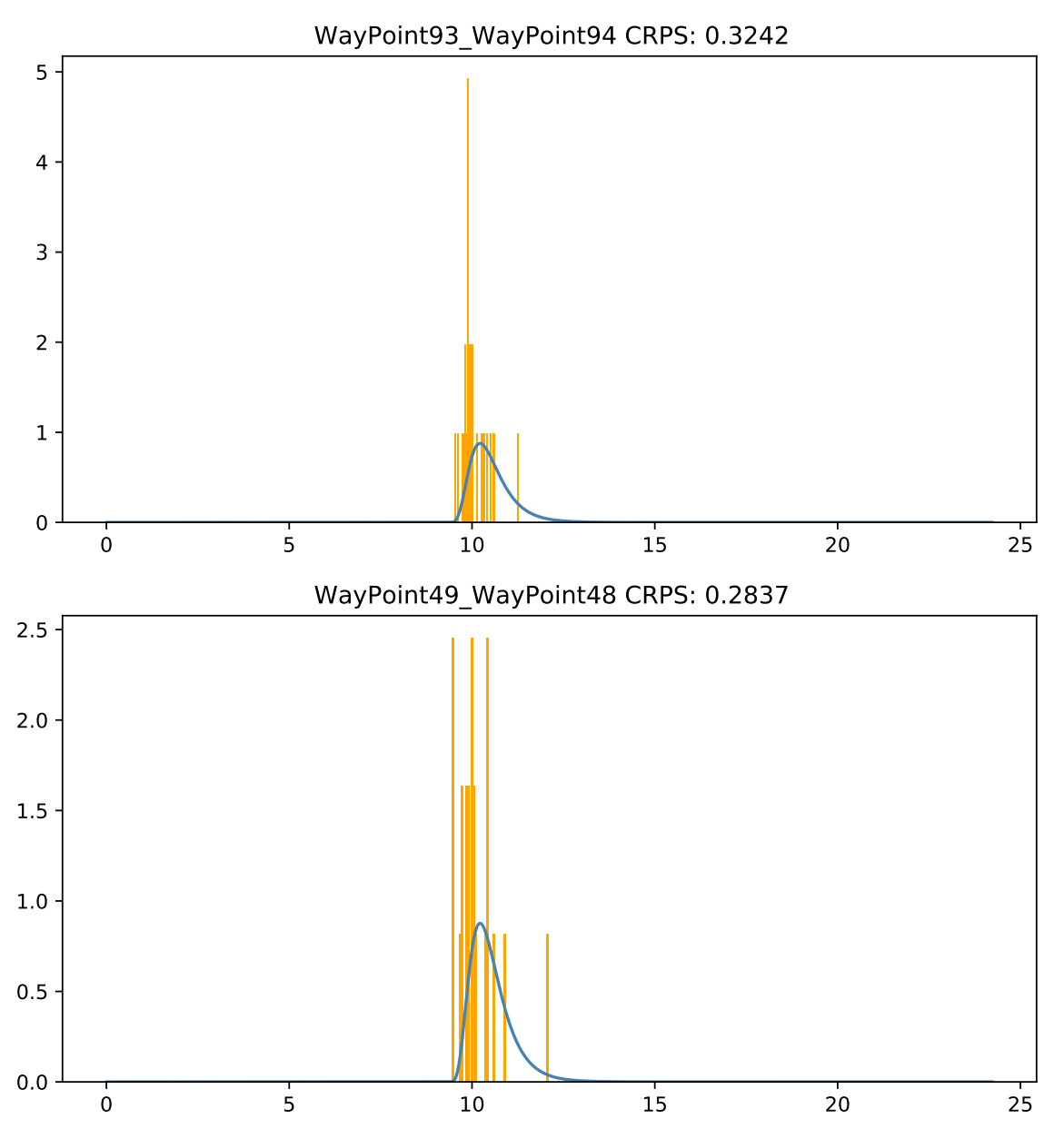
Median is less sensitive to outliers than mean. Below, there are situations where we have many datapoints (left) and situations where we have few datapoints (right), calculating the mean CRPS score. The mean CRPS score is sensitive to the number of observations, with large datasets typically having a bad (high) CRPS score compared to low datasets. This is due to the outliers in large datasets since we are more likely to have extreme observations. Therefore, I used the median CRPS score instead of the mean CRPS score.

**Method 2:**

Mean CRPS score below:

Median CRPS score (large dataset – right, small dataset – left):

# WED: KS metrics & similar edges

#### Clarify parameterisation of lognormal distribution

**My parameterisation (left)** **Scipy Parameterisation (right)**



**Transformation**

* S = SQRT(var)
* Loc = offset
* Scale = EXP(mean)
* where mean & var are the mean & variance of LN(operation\_time)

**We can use in fitting the lognormal distribution and performing KS tests:**

p\_bayes = Lognormal( t\_test-offset, mean\_map,v ar\_map )

ks\_res = sp.stats.kstest( t\_op,

lambda k: sp.stats.lognorm.cdf( k, s = np.sqrt(var\_map), loc = offset, scale = np.exp(mean\_map) ) )

#### Differences between CRPS & KS Scores

To see if there is noticeable difference between the scoring metrics, I looked at cases where there was a large difference between the CRPS and KS score (in the Excel spreadsheet, attached). A few metrics, I considered:

* Rank\_diff: I ranked the edges based on CRPS & KS score from lowest (best) to highest (worst). The rank difference is the np.abs(CRPS\_rank - KS\_rank)
* adjusted\_diff: I transformed both the CRPS & KS scores into the interval between 0 & 1 and took the difference between the adjusted scores
* abs\_diff: the difference between the (unadjusted) CRPS & KS scores

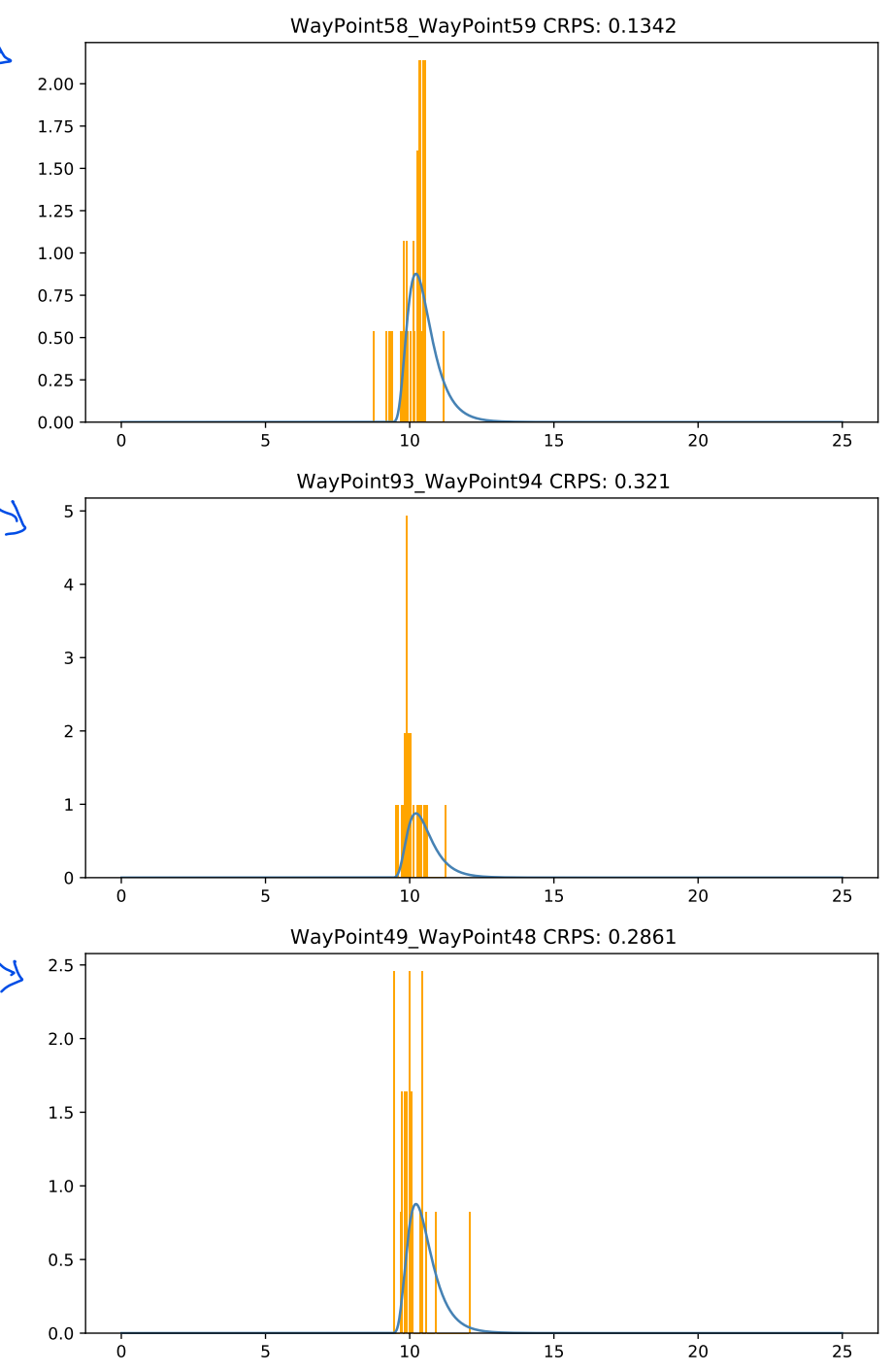
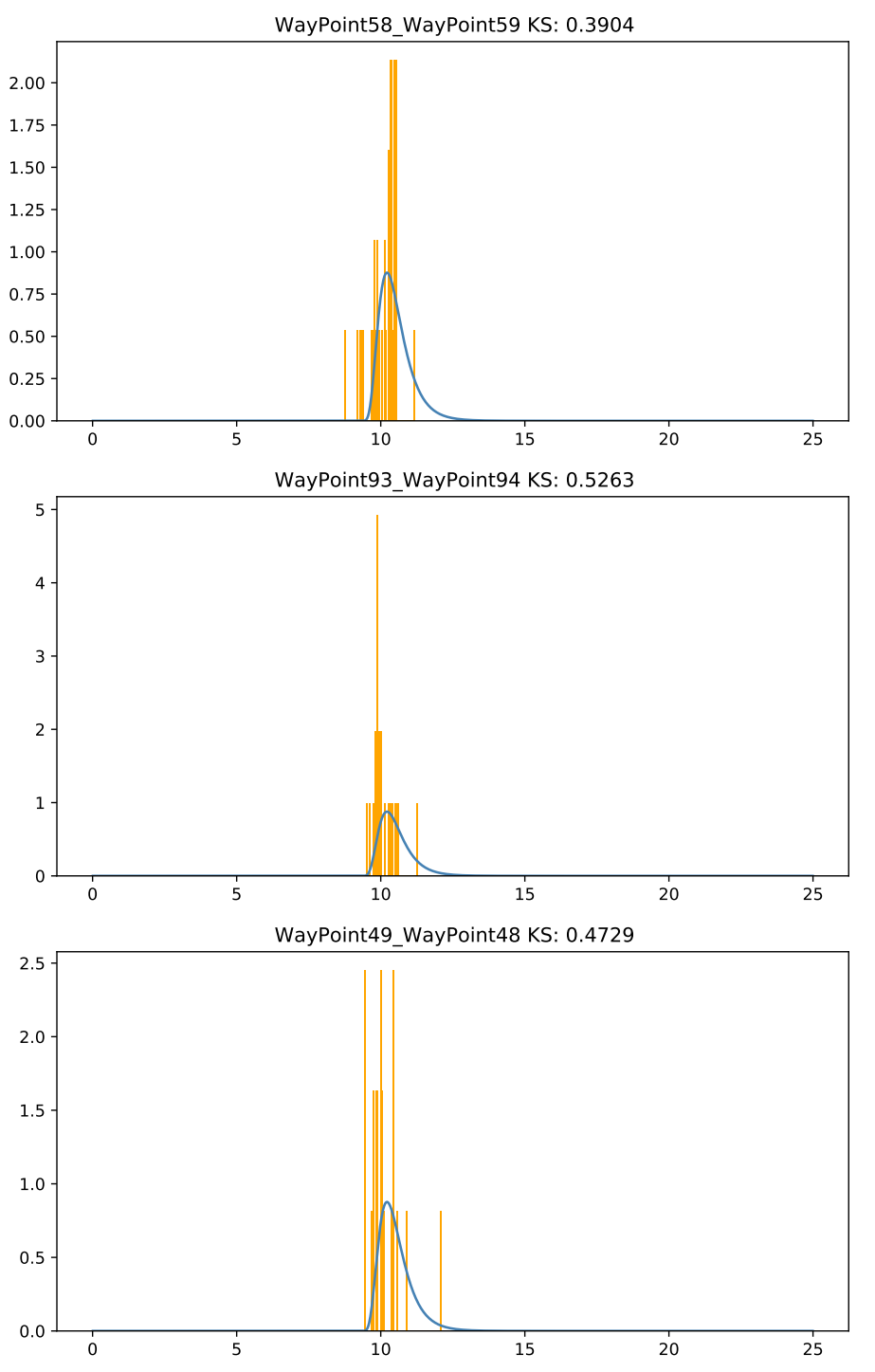
I had a look at some of the cases where there was a high difference (visually more red in the excel heatmap) - these are marked "check" in Excel.

These correspond to certain edges with a small amount of datapoints.

* The median (& mean) CRPS scores are low since the observations occur within the peaked region of the predicted PDF.
* However, the KS score is higher since the empirical CDF shows more difference to the predicted CDF.
* I haven't checked this for all clusters yet though.

**TO DO:**

* **Add original pdf to mergefit plots**
* **Add number of data points to “mergefit\_stats.xlsx” spreadsheet**

**Above:** left is CRPS score, right is KS score. The plots are the same

We see that when there are few datapoints, **the CRPS score tends to be low (good) in comparison to CRPS for other edges, whereas the KS score tends to be high (bad)**

# THU: Transform Lognormal Parameters

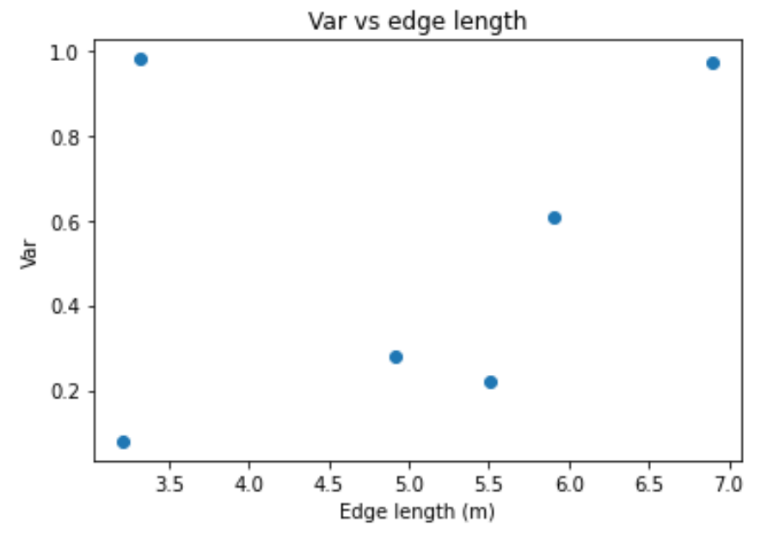
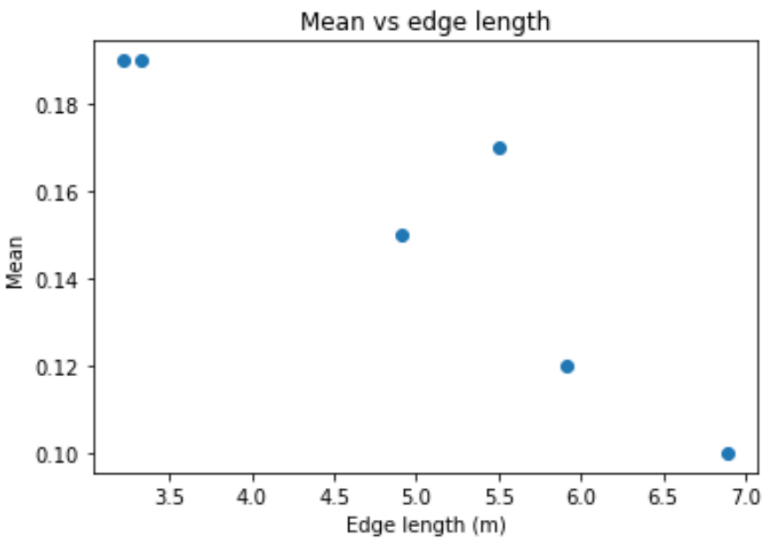
Is there some relationship between edge\_length and the lognormal parameters (mean, variance, offset)?

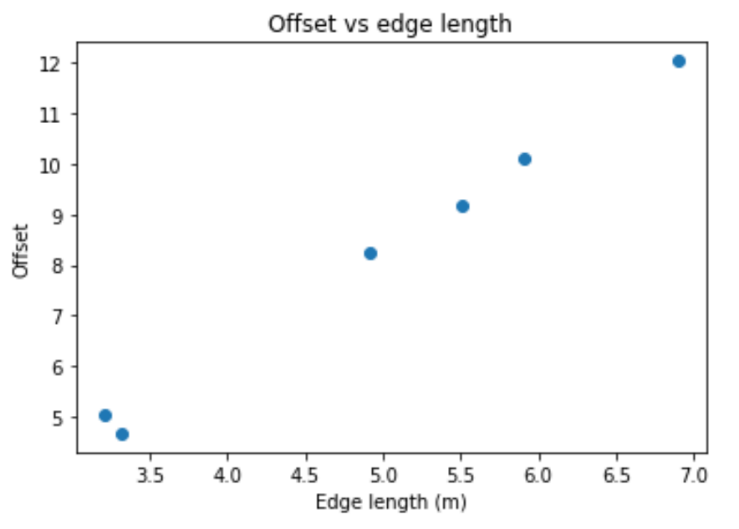
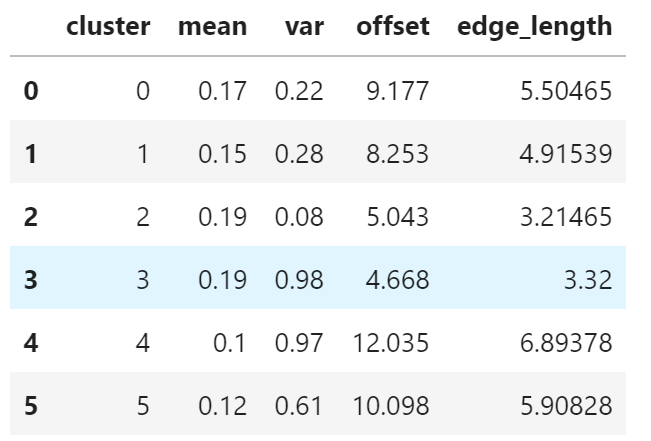
* Offset shows a clear linear relationship, which is expected given an intuition that the robots cannot physically exceed a certain threshold time.
* Mean & variance are less clear.

Try:

* linear/polynomial regression with only edge length
* regression neural network with edge\_length + other possible parameters as the input

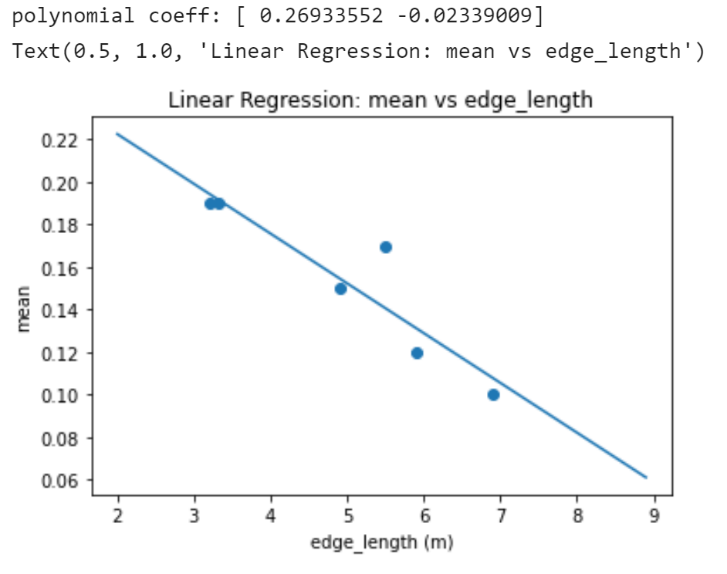
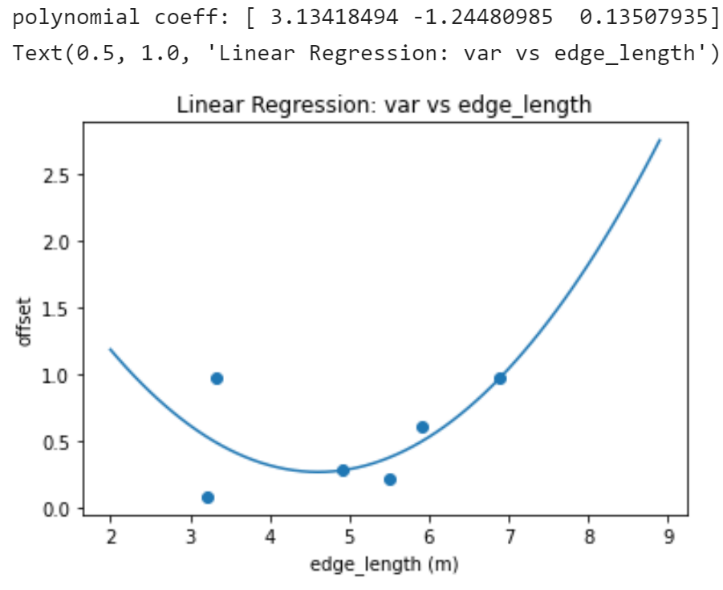
#### Plot lognormal parameters vs edge length

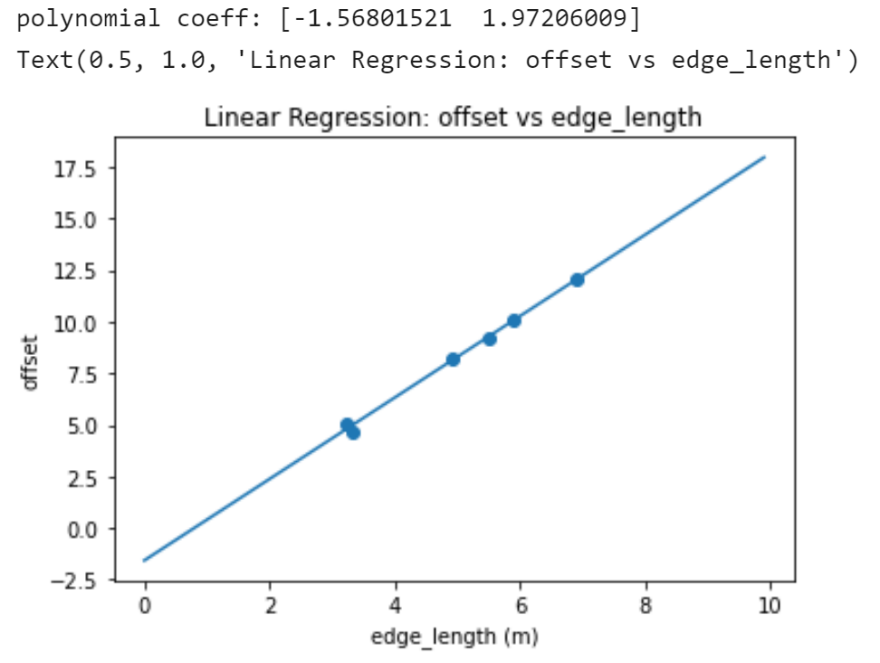


#### Then I tried polynomial regression:

* Offset is linear
* Var is quadratic
* Mean is cubic



# FRI: CRPS & KS for mergefit

I fixed the bugs from yesterday:

**mergefit\_crps\_ks.pdf:** contains plots of the mergefit/original distributions and a histogram of data from the original edge.

* The plots with red titles have a KS\_mergefit which is at least 0.1 higher (worse) than KS\_original

**mergefit\_stats.xlsx** contains the median CRPS & KS score data and the differences:

* The median CRPS is lower (better) for mergefit in 147/150 edges, with the other 3 being only slightly worse. This might be because mergefit has a peak which is close to the median observation for each edge
* For 10 datapoints, KS is higher (worse) for mergefit in 28/150 edges. However, 13/28 of these edges show a significant difference in KS (>0.1). Furthermore, 7 of the 10 edges with most data have this high KS difference (>0.1)

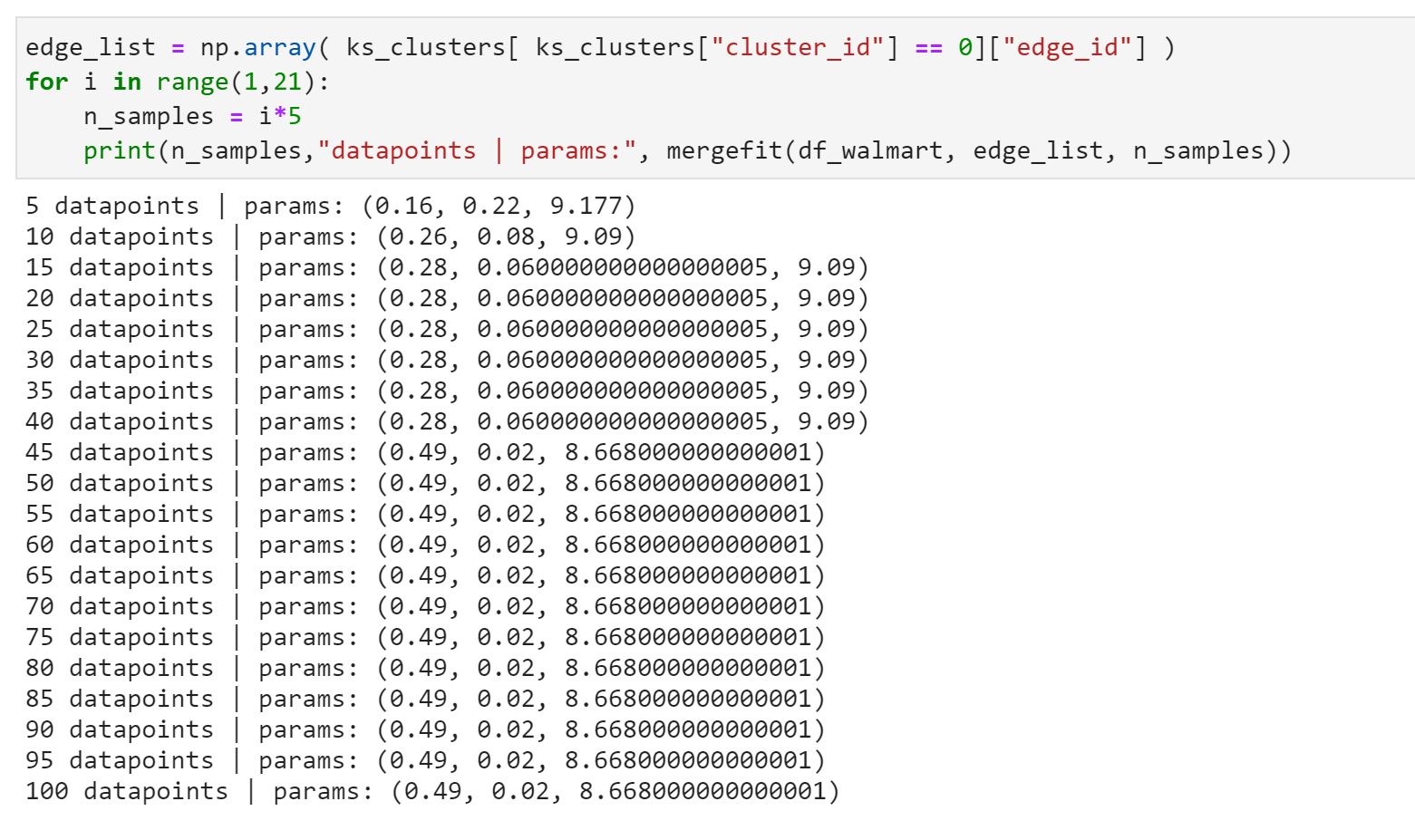
#### Mean or median CRPS:

* Median is more intuitive than mean CRPS since mean CRPS favours edges with low number of observations
* Using mean CRPS (removing outliers exceeding 1 or 2 std) does not appreciably change the difference between the CRPS score for the mergefit and original edges

#### Number of datapoints:

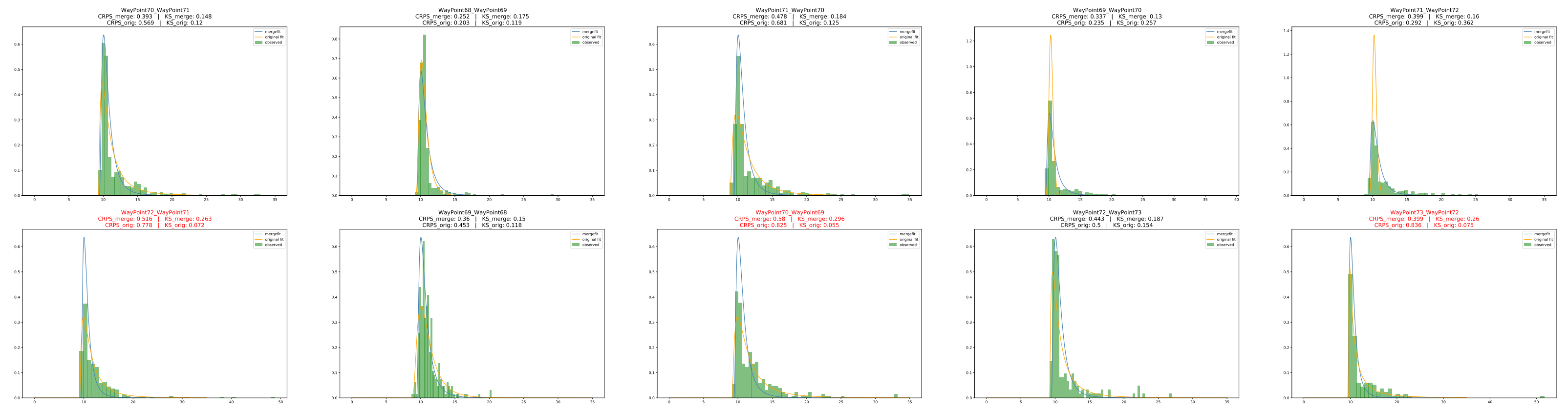
Parameters of lognormal fall into 3 regimes

* <10 datapoints
* 10 to 45 datapoints
* >45 datapoints



Testing the different regimes, **5 datapoints (top)** is much better than 10 or any other choice of datapoints since this models the tail of the distribution well.

* For 10 datapoints, KS is higher (worse) for mergefit in 28/150 edges. However, 13/28 of these edges show a significant difference in KS (>0.1). Furthermore, 7 of the 10 edges with most data have this high KS difference (>0.1)
* For 5 datapoints, only 3 of the 10 edges with the most datapoints have KS difference >0.1. In the cases that do have a >0.1 KS difference, the mergefit KS score is still quite good (0.2-0.3)



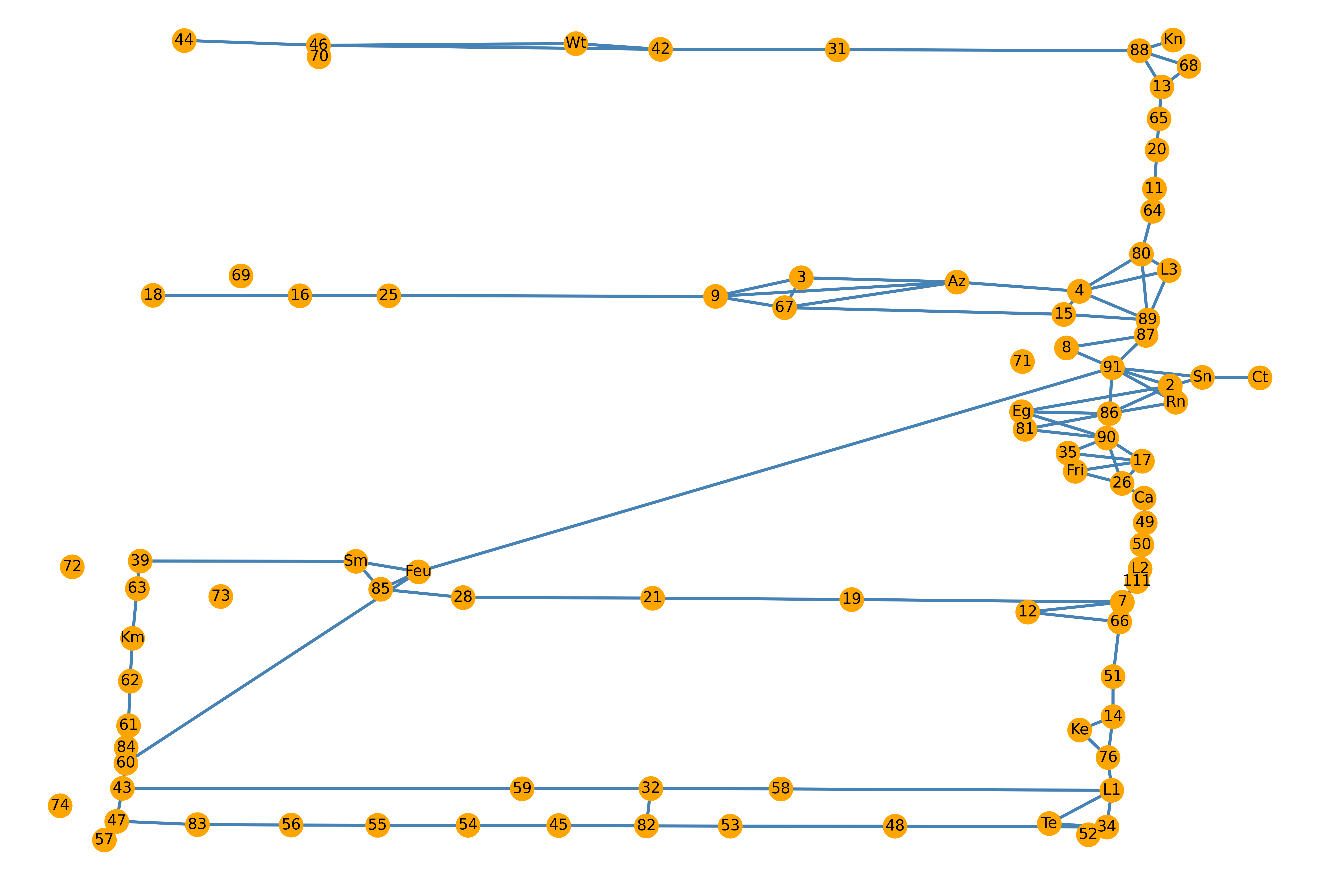
# STRANDS data

**Process**

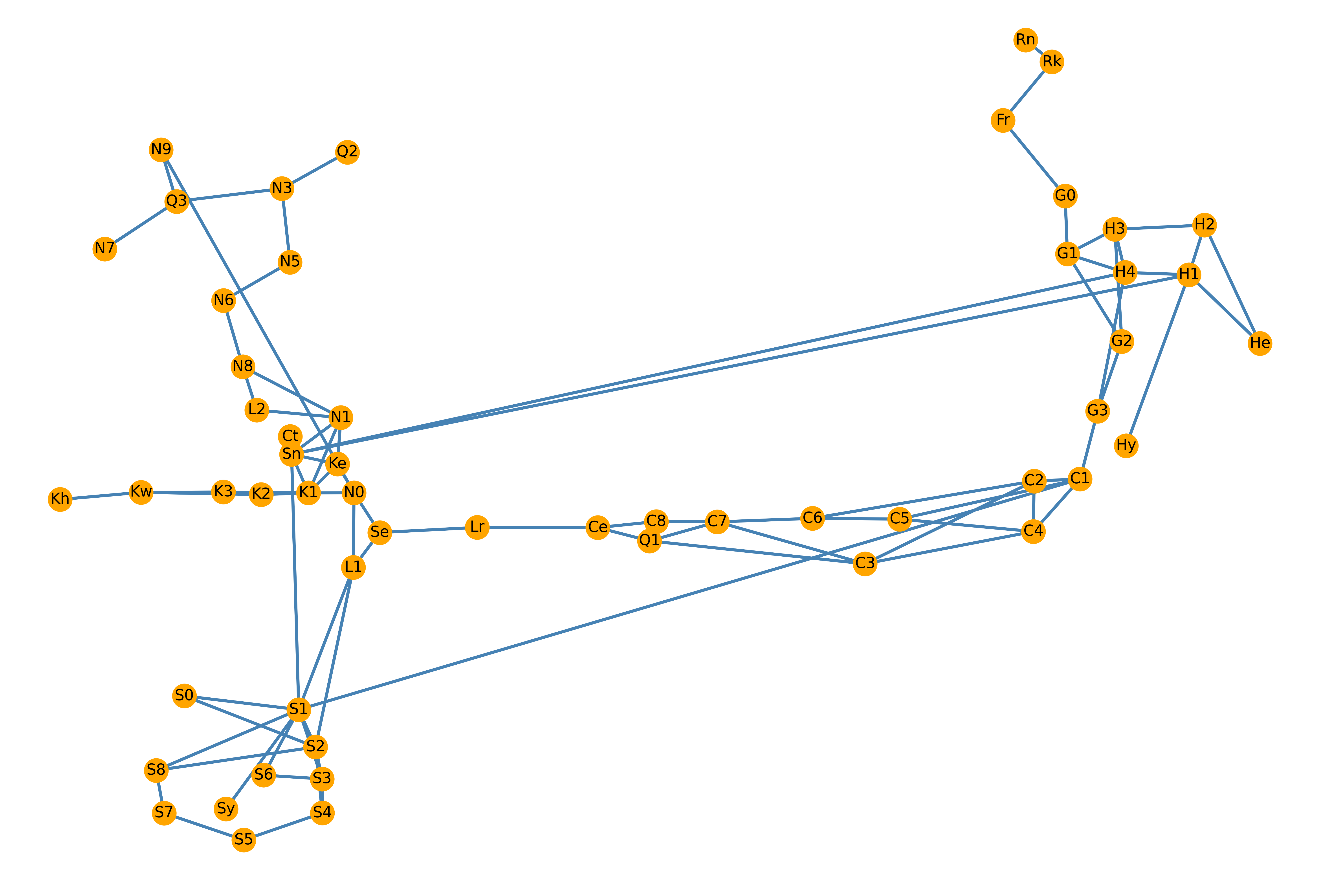
1. What model fits the data well – still lognormal?
2. Find what factors affect clustering in STRANDS data
3. Can you use STRANDS data to learn relationship between parameters and those factors
   1. Use multivariate regression to solve for mean/var/offset using the same model
   2. Possible factors: Edge\_length, maximum turning angle, connections
4. Does this generalise back to Walmart/Blenheim? i.e. can you make predictions between of the cluster parameters
   1. May need to consider robot speed & turning speed

#### Maps of the 3 environments (AAF & TSC look promising):

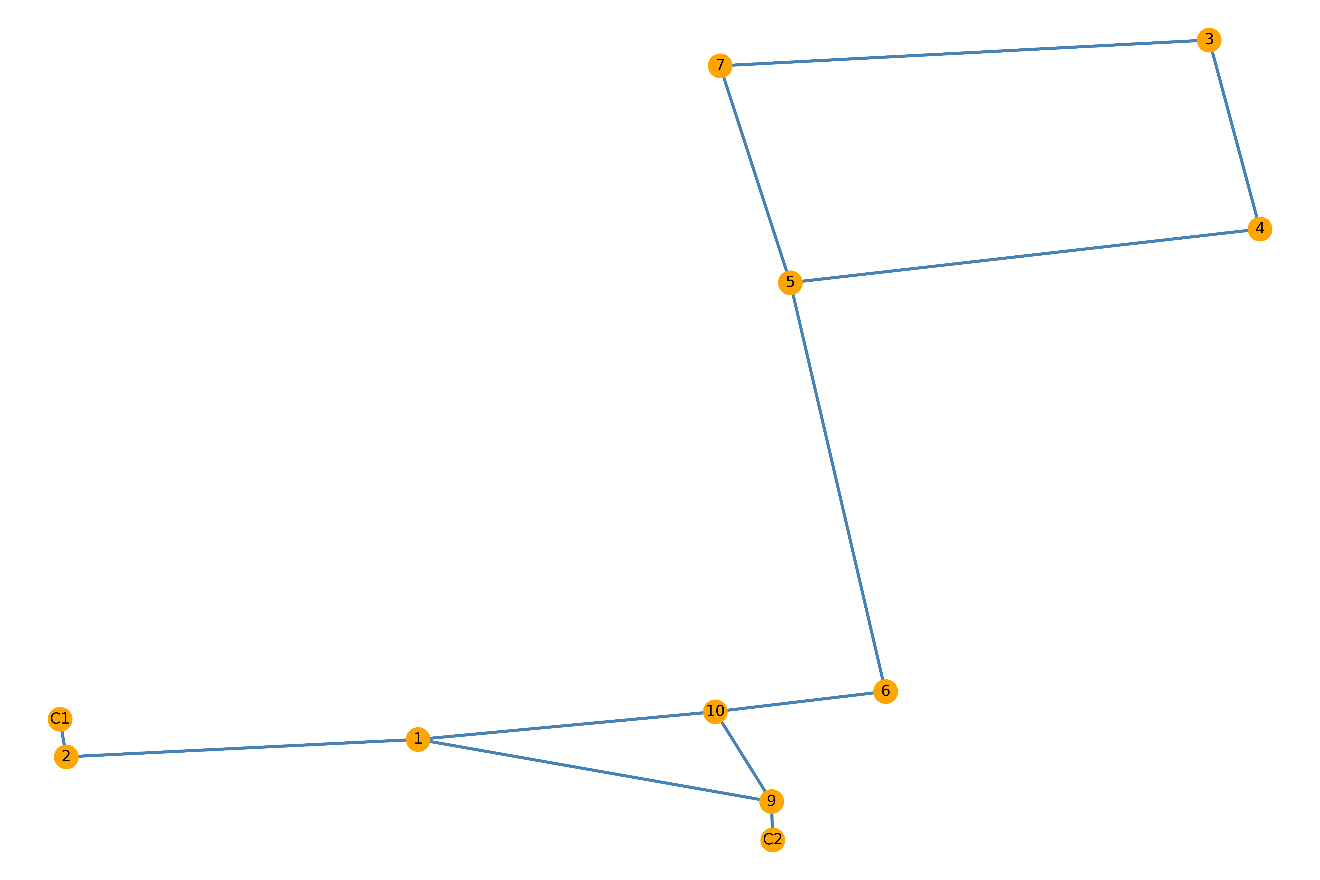
**AAF:**



**TSC:**

****

**LABS:**

****