Alice Wu Lim, PhD

Data Science Fellow

City of Hope

Genomic cluster project

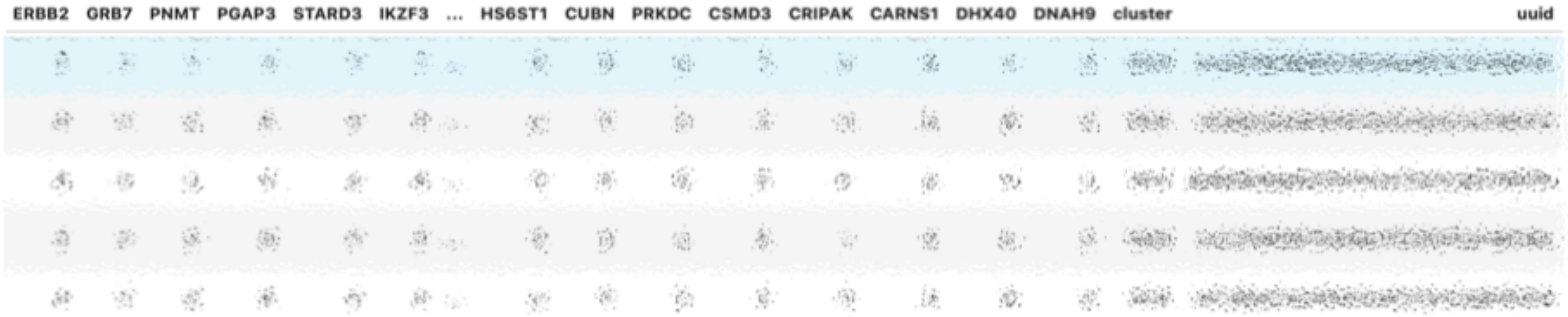
What is Patient-similarity?

- Oftentimes in healthcare, predictions are made about the “average patient”.
- Patient-similarity models make predictions about individual patients by identifying patients which are like an index patient and tailoring the predictions based on these sets of similar patients.
- At City of Hope, satellite hospitals can use patient-similarity to match their patients to patient data from the main hospital

- See https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5357318/

Goal:

Understand what genomic clusters mean with respect to patient variables. Specifically, find patient variables which correlate to genomic data.



ERBB2	GRB7	PNMT	PGAP3	STARD3	IKZF3	...	HS6ST1	CUBN	PRKDC	CSMD3	CRIPAK	CARNS1	DHX40	DNAH9	cluster	uuid
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Here, the first 14 columns shown are the genomic data. The 'cluster' column was created by k-means clustering the data. There are 12 possible 'clusters'. The 'uuid' is the individual patient's unique id.

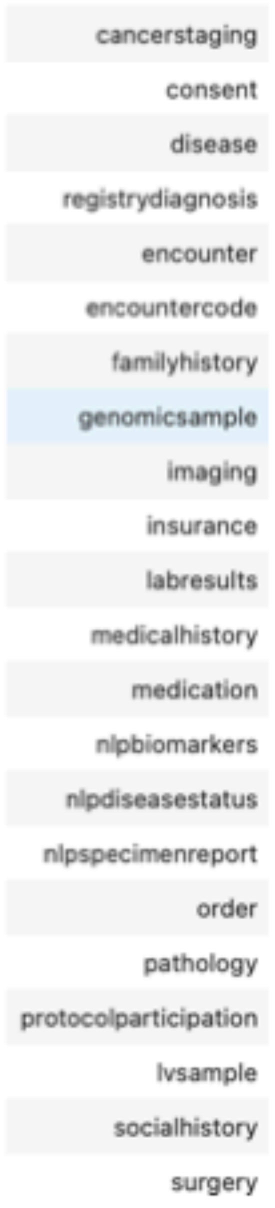
Initial Steps:

Step 1 Use genomic data to create clusters (eg Cluster 102, 103, etc)

Step 2 Attach patient variables to cluster data (eg bmi, tumor size)

Step 3 Evaluate whether patient variable (eg bmi, tumor size) correlates to cluster data

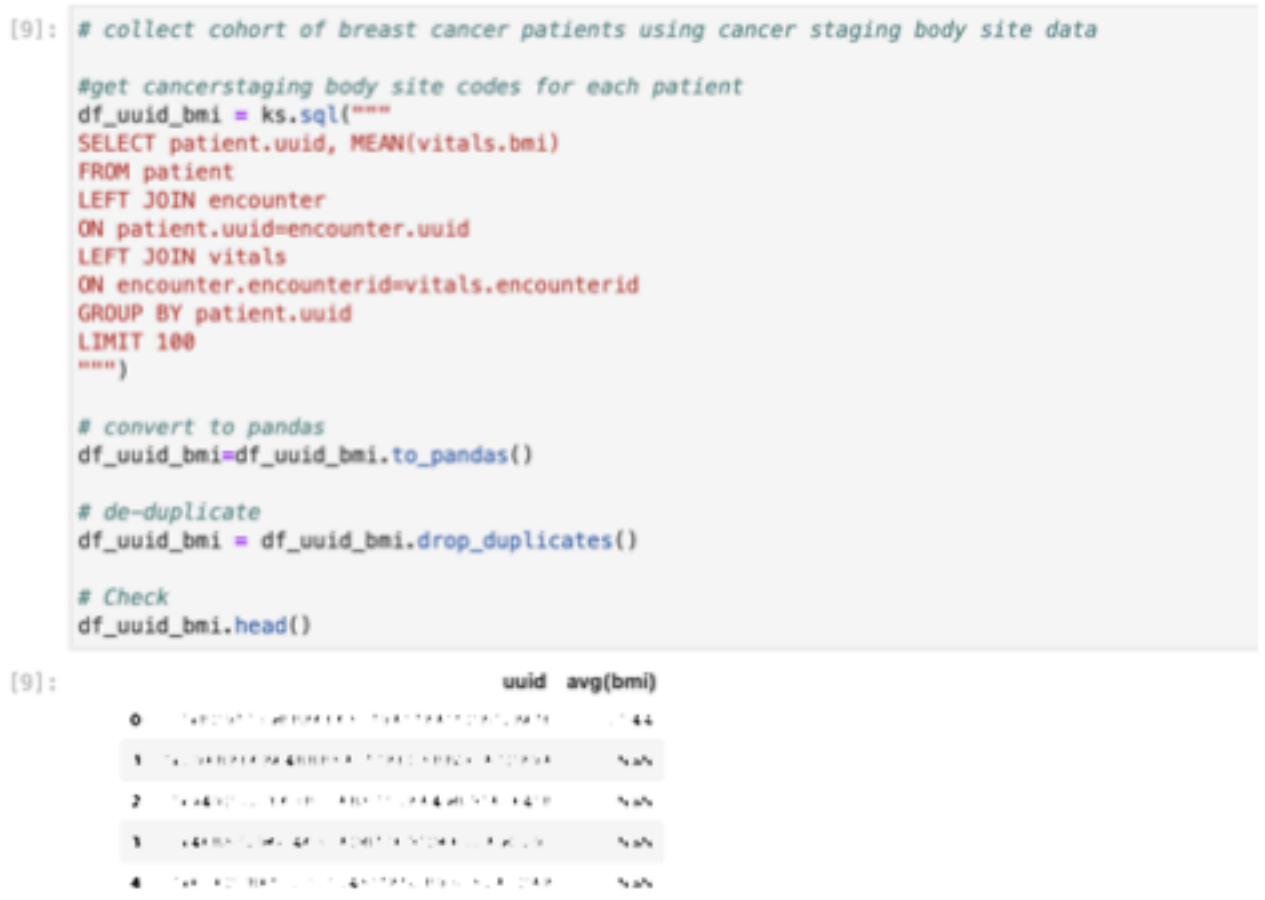
Types of patient data



cancerstaging
consent
disease
registrydiagnosis
encounter
encountercode
familyhistory
genomicsample
imaging
insurance
labresults
medicalhistory
medication
nlpbiomarkers
nlpdiseasestatus
nlpspecimenreport
order
pathology
protocolparticipation
lvsample
socialhistory
surgery

Example of getting bmi data from City of Hope's Poseidon

- Poseidon: Precision Oncology Software Environment Interoperable Data Ontologies Network



```
[9]: # collect cohort of breast cancer patients using cancer staging body site data

     #get cancerstaging body site codes for each patient
     df_uuid_bmi = ks.sql("""
     SELECT patient.uuid, MEAN(vitals.bmi)
     FROM patient
     LEFT JOIN encounter
     ON patient.uuid=encounter.uuid
     LEFT JOIN vitals
     ON encounter.encounterid=vitals.encounterid
     GROUP BY patient.uuid
     LIMIT 100
     """)

     # convert to pandas
     df_uuid_bmi=df_uuid_bmi.to_pandas()

     # de-duplicate
     df_uuid_bmi = df_uuid_bmi.drop_duplicates()

     # Check
     df_uuid_bmi.head()
```

```
[9]:                              uuid   avg(bmi)
     0                    [illegible] [illegible]
     1                    [illegible]       NaN
     2                    [illegible]       NaN
     3                    [illegible]       NaN
     4                    [illegible]       NaN
```

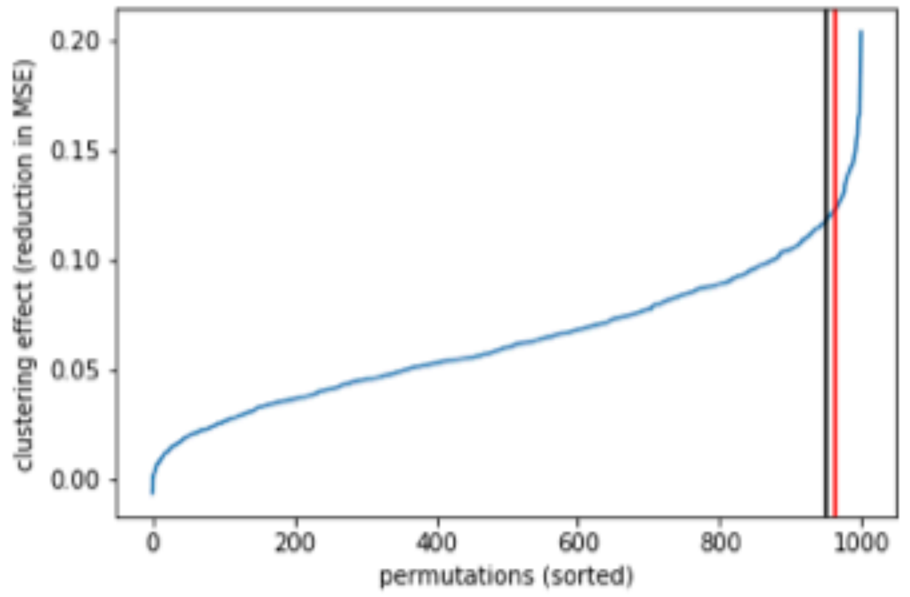
Step 2 example: Attach alcoholusecode to cluster

uuid	cluster	alcoholusecode
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]

The 'uuid' refers to the patient's unique id. The 'cluster' is the genomic cluster. The alcoholusecode is a code which tells us how much alcohol the patient consumes per week. We would like to see if there is correlation between the genomic cluster and the alcoholusecode.

Step 3 example: Evaluate whether ageofonset correlates to cluster data

```
permutation p-value of the effect of clustering on improving prediction is: 0.036

[14]: 0.036
```



Some Initial results

$p>0.05$

- Alcoholousecode (in socialhistory)
- Alcohol_oz_week (in socialhistory)
- Alcoholhowoftencode (in socialhistory)
- Tobaccopackperday (in socialhistory)
- Ethniccode (in patientethnic)
- Relationshipcode (in familyhistory)
- Tumorsizemm (in cancerstaging)
- Classificationcode (in cancerstaging)
- Histologycode (in registrydiagnosis)

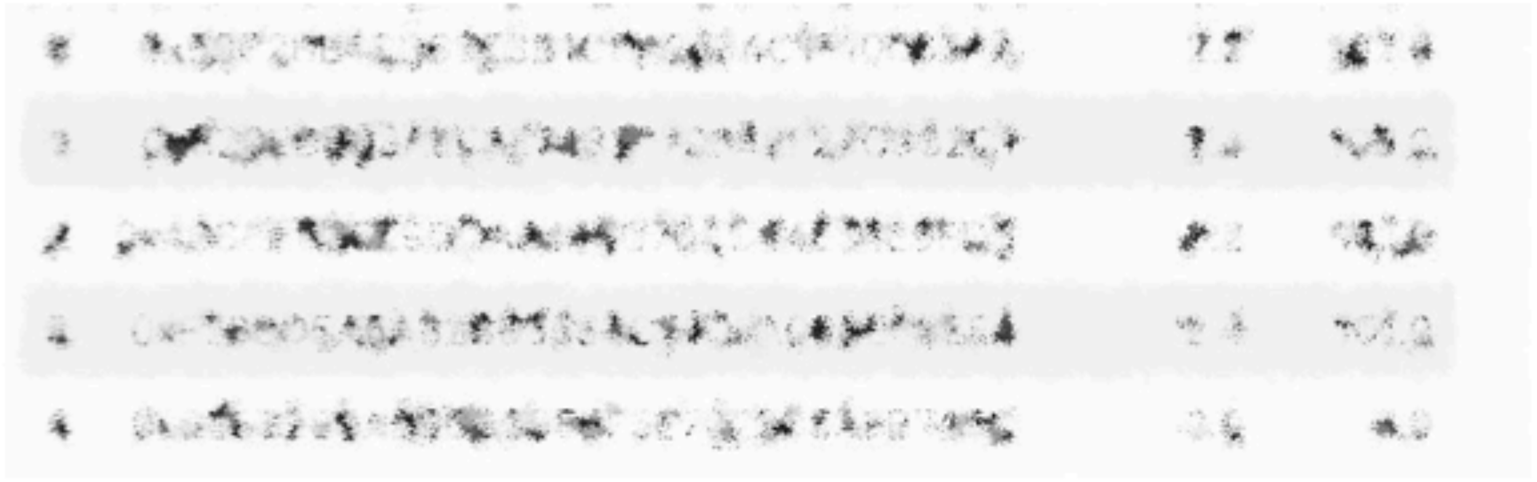
$p<0.05$

- Ageofonset (in familyhistory)

Promising patient variable: T-stage

- From the NIH website:
  - Stage refers to the extent of your cancer, such as how large the tumor is, and if it has spread.
  - The TNM system is the most widely used cancer staging system.
  - The T refers to the size and extent of the main tumor. The main tumor is usually called the primary tumor.
    - TX: Main tumor cannot be measured.
    - T0: Main tumor cannot be found.
    - T1, T2, T3, T4: Refers to the size and/or extent of the main tumor. The higher the number after the T, the larger the tumor or the more it has grown into nearby tissues. T's may be further divided to provide more detail, such as T3a and T3b.

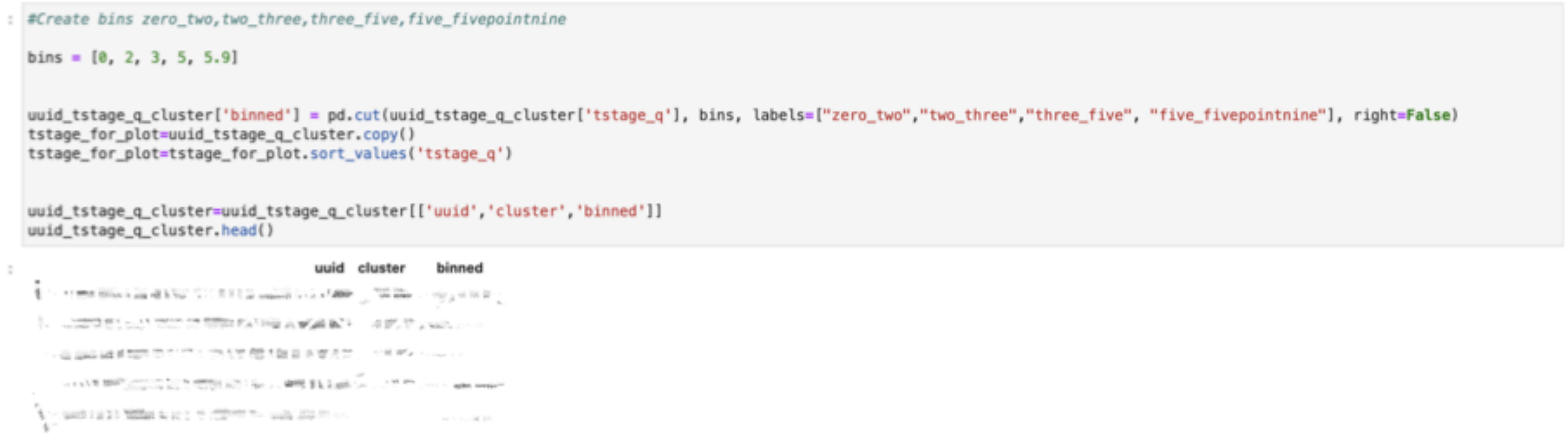
T-stage + cluster



	uuid	tstage_q	cluster
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]

This is our most promising variable. We attached the tstage_q variable with the genomic cluster variable. Next, we will work with the tstage_q variable and the cluster variable to see if certain tstage-q's and clusters have low p-value.

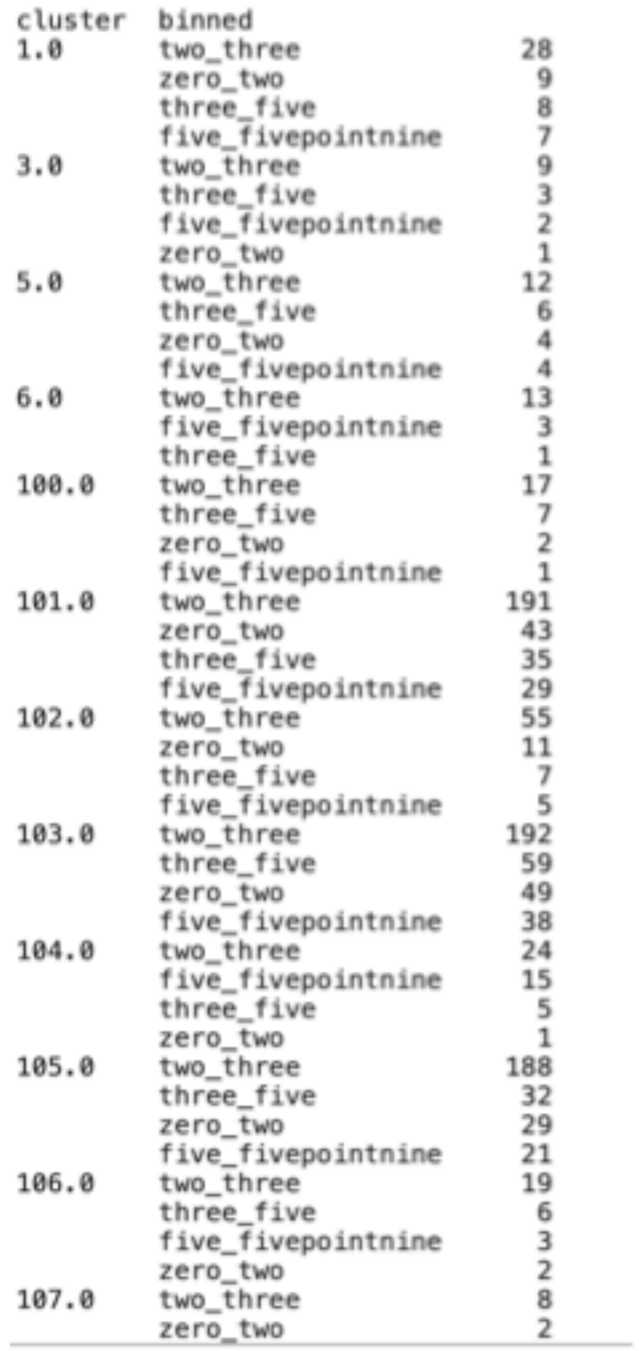
Initial binning technique for t-stage:



```
#Create bins zero_two,two_three,three_five,five_fivepointnine

bins = [0, 2, 3, 5, 5.9]


uuid_tstage_q_cluster['binned'] = pd.cut(uuid_tstage_q_cluster['tstage_q'], bins, labels=["zero_two","two_three","three_five", "five_fivepointnine"], right=False)
tstage_for_plot=uuid_tstage_q_cluster.copy()
tstage_for_plot=tstage_for_plot.sort_values('tstage_q')


uuid_tstage_q_cluster=uuid_tstage_q_cluster[['uuid','cluster','binned']]
uuid_tstage_q_cluster.head()
```

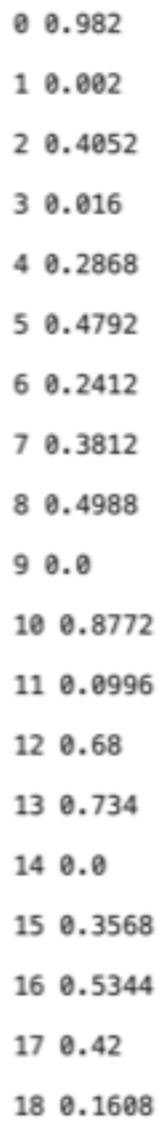
	uuid	cluster	binned
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]

Value counts for some of the cluster-bin combos:

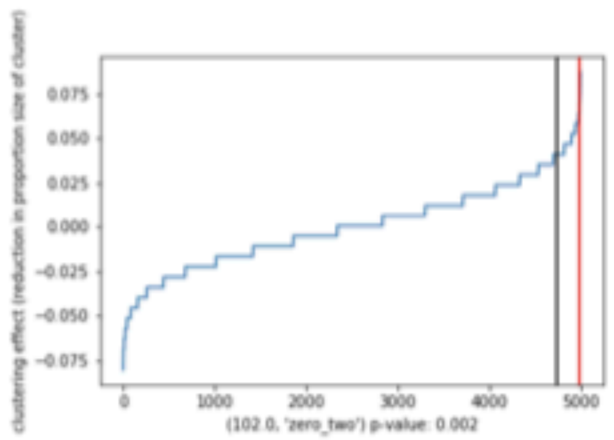


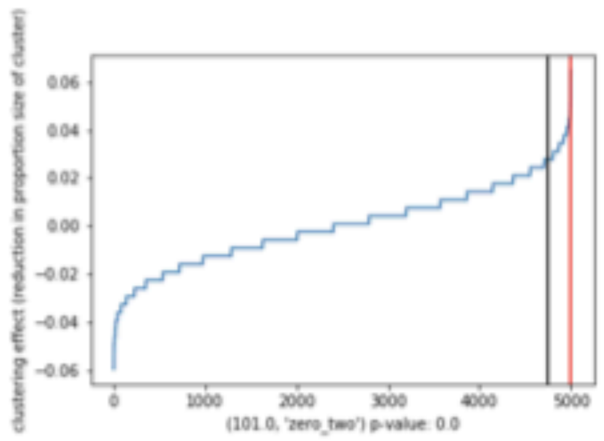
```
cluster  binned
1.0      two_three              28
         zero_two                9
         three_five              8
         five_fivepointnine      7
3.0      two_three               9
         three_five              3
         five_fivepointnine      2
         zero_two                1
5.0      two_three              12
         three_five              6
         zero_two                4
         five_fivepointnine      4
6.0      two_three              13
         five_fivepointnine      3
         three_five              1
100.0    two_three              17
         three_five              7
         zero_two                2
         five_fivepointnine      1
101.0    two_three             191
         zero_two               43
         three_five             35
         five_fivepointnine     29
102.0    two_three              55
         zero_two               11
         three_five              7
         five_fivepointnine      5
103.0    two_three             192
         three_five             59
         zero_two               49
         five_fivepointnine     38
104.0    two_three              24
         five_fivepointnine     15
         three_five              5
         zero_two                1
105.0    two_three             188
         three_five             32
         zero_two               29
         five_fivepointnine     21
106.0    two_three              19
         three_five              6
         five_fivepointnine      3
         zero_two                2
107.0    two_three               8
         zero_two                2
```

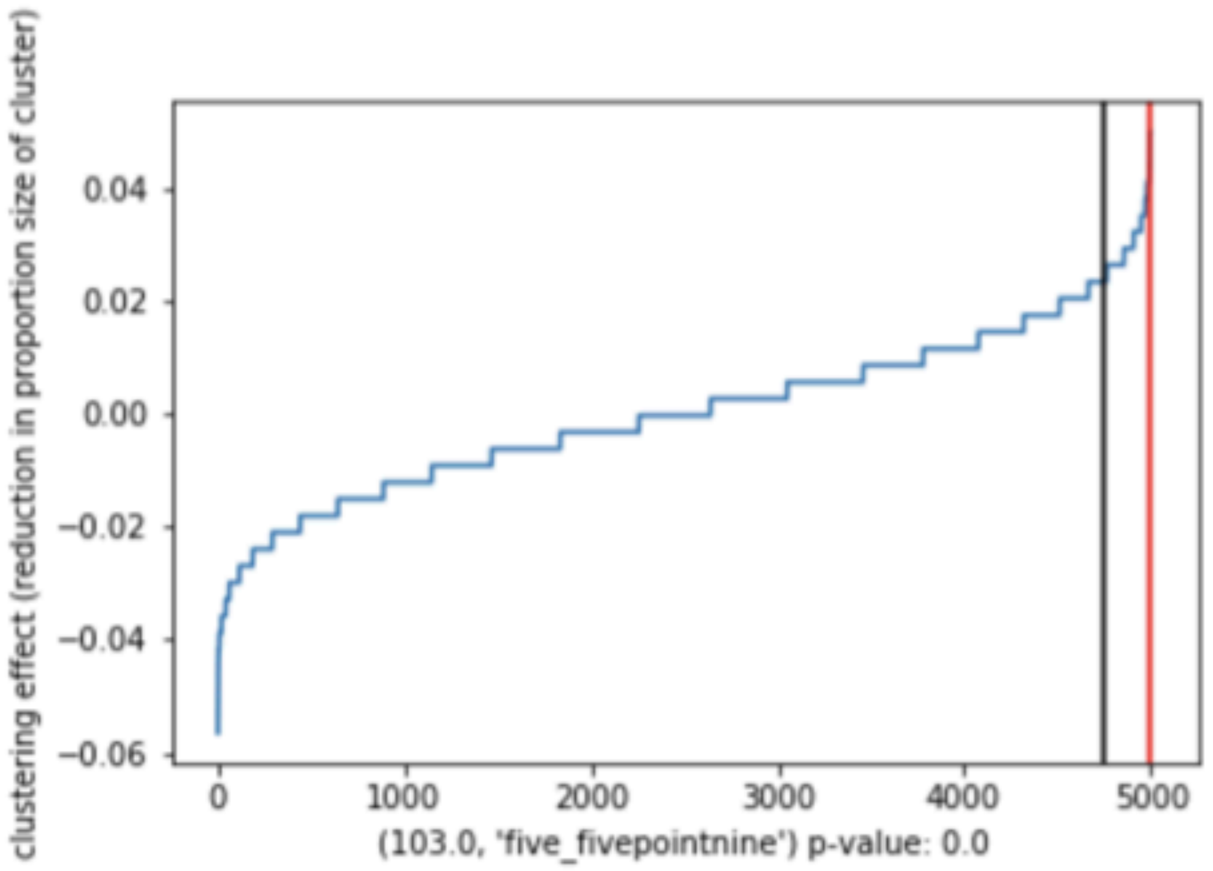
P-values of combinations of cluster-bin:



```
0 0.982

1 0.002

2 0.4052

3 0.016

4 0.2868

5 0.4792

6 0.2412

7 0.3812

8 0.4988

9 0.0

10 0.8772

11 0.0996

12 0.68

13 0.734

14 0.0

15 0.3568

16 0.5344

17 0.42

18 0.1608
```


clustering effect (reduction in proportion size of cluster)
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
0
1000
2000
3000
4000
5000
(102.0, 'zero_two') p-value: 0.002

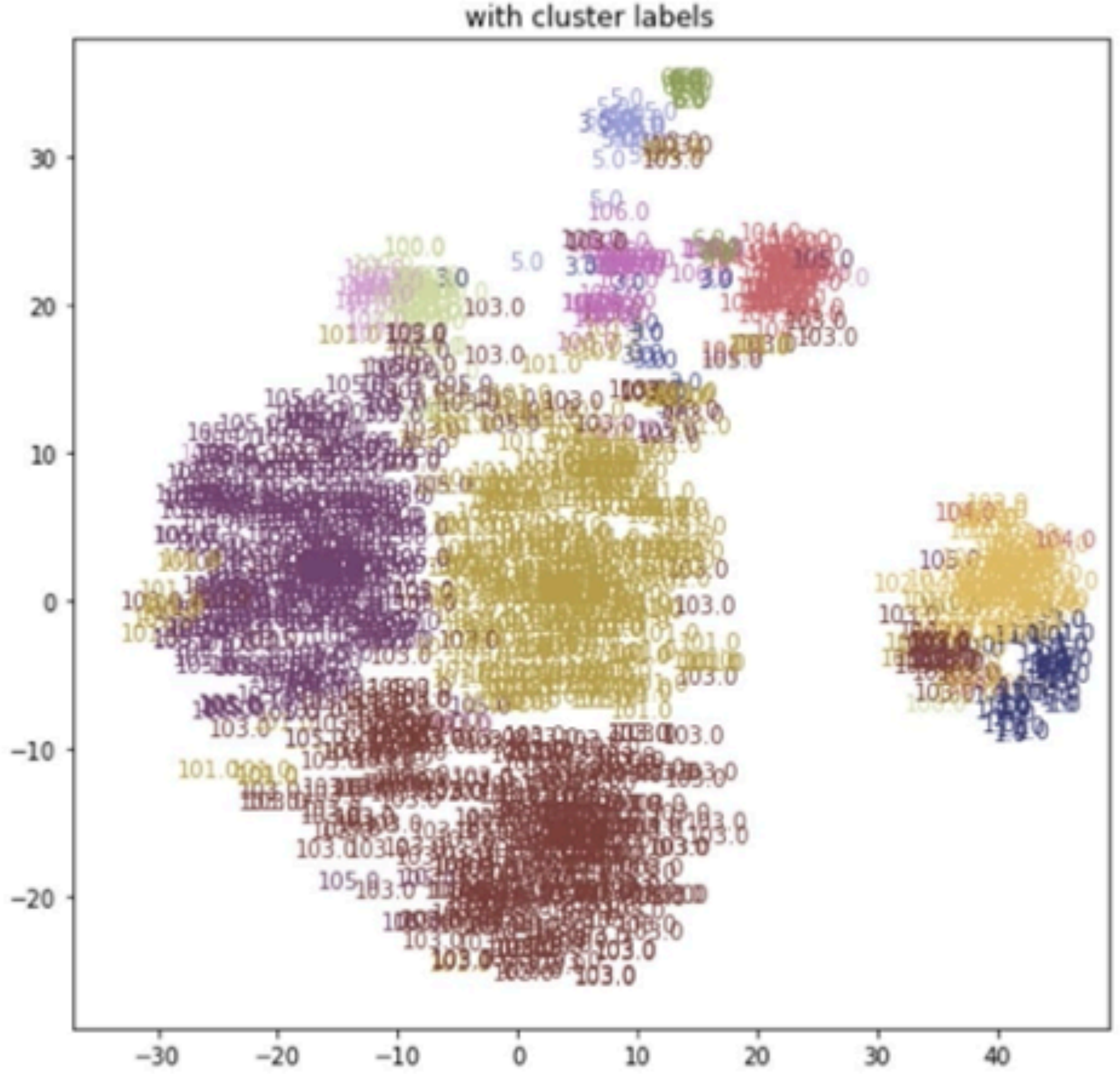

clustering effect (reduction in proportion size of cluster)
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
0
1000
2000
3000
4000
5000
(101.0, 'zero_two') p-value: 0.0


clustering effect (reduction in proportion size of cluster)
0.04
0.02
0.00
−0.02
−0.04
−0.06
0
1000
2000
3000
4000
5000
(103.0, 'five_fivepointnine') p-value: 0.0

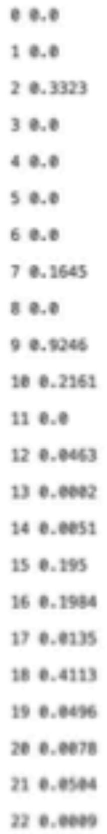
P-values:

t-sne visualizations of the clusters

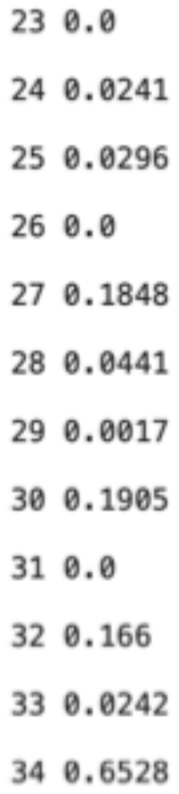
es using t-SNE with perplexity: 30 and seed: 2



We duplicated each of the rows 10x in order to get less sparse value counts in our cluster-bin combos. New p-values:



```
0 0.0
1 0.0
2 0.3323
3 0.0
4 0.0
5 0.0
6 0.0
7 0.1645
8 0.0
9 0.9246
10 0.2161
11 0.0
12 0.0463
13 0.0002
14 0.0051
15 0.195
16 0.1984
17 0.0135
18 0.4113
19 0.0496
20 0.0078
21 0.0504
22 0.0009
```



```
23 0.0
24 0.0241
25 0.0296
26 0.0
27 0.1848
28 0.0441
29 0.0017
30 0.1905
31 0.0
32 0.166
33 0.0242
34 0.6528
```