Diabetes Project

Aviv Yaish

General note

The code is found in diabetes_project.py, and is easily run using:

python diabetes_project.py

Running the file will output all graphs and various results presented here. Note that the graphs will be shown only after all the code finished running.

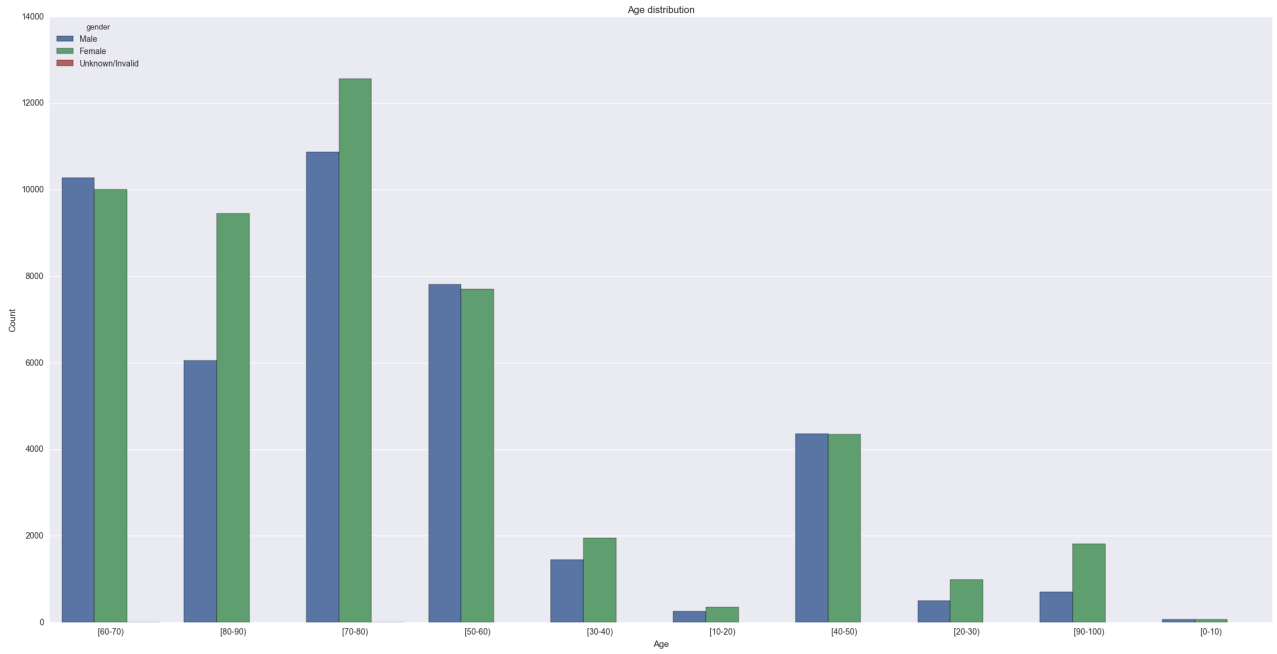
Each part (and subpart) is represented by a different function in the code. The entire code is documented and filled with prints, so that when it is run you will get all answers for the various parts. I've summarized all the answers here.

Part 1

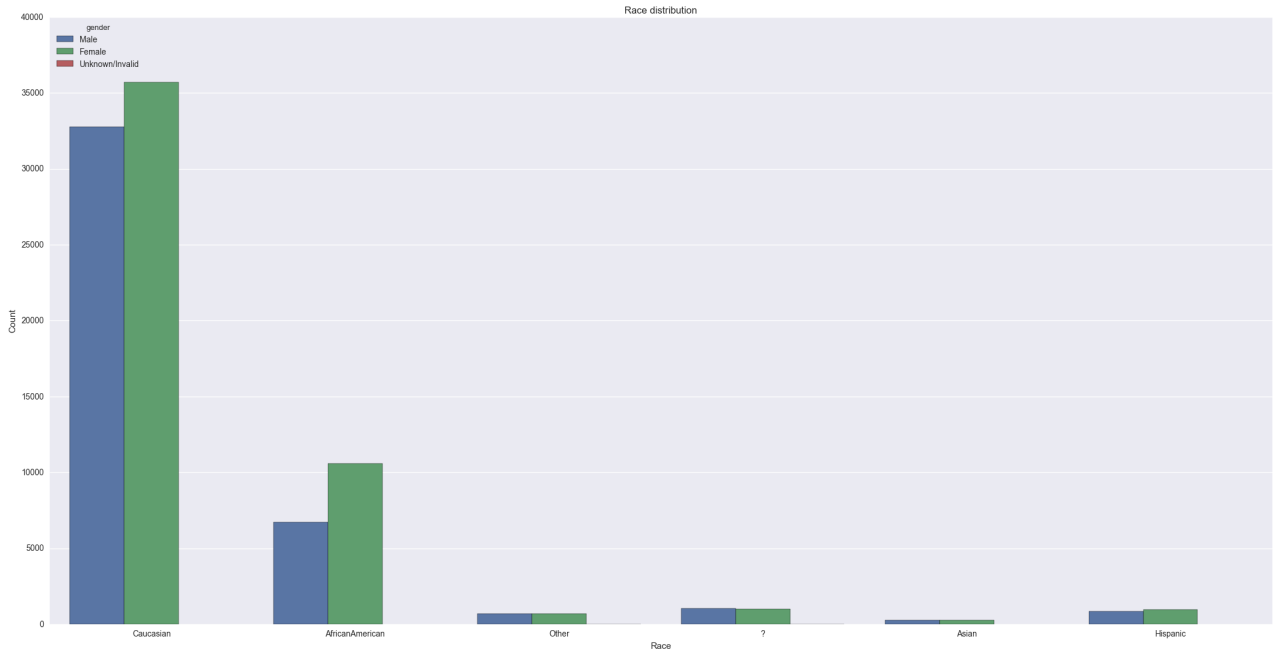
Note: there are more graphs at the bottom of the assignment.

Subpart 1

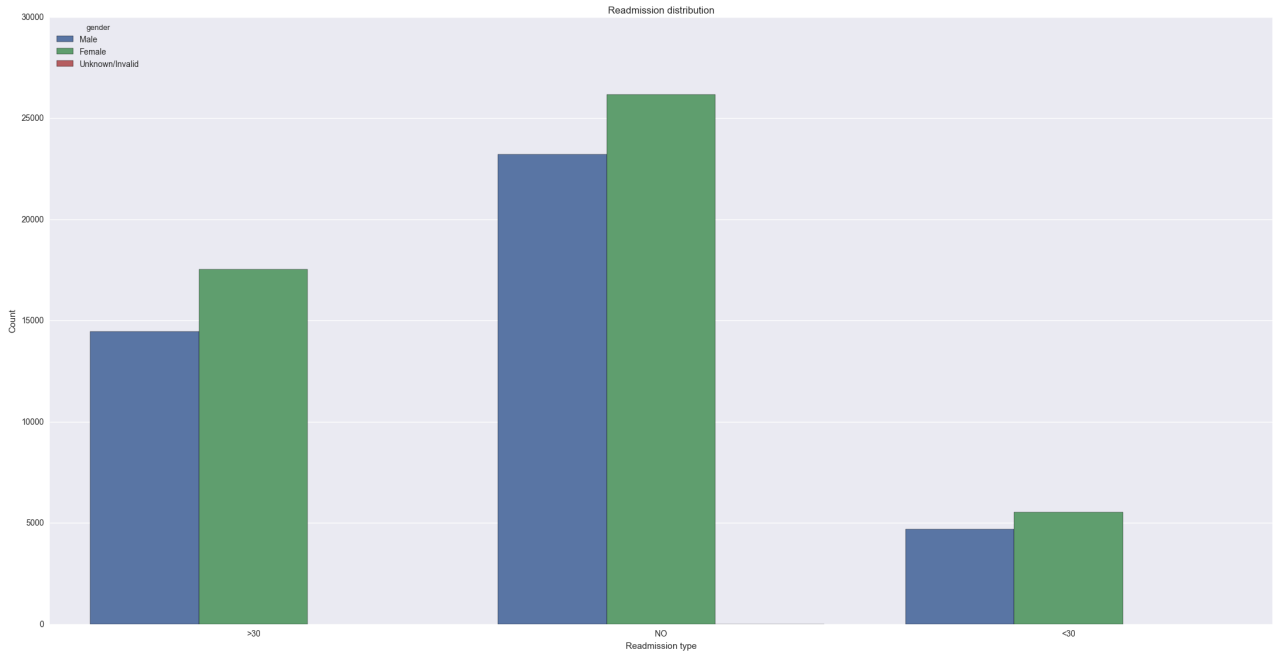
Age distribution



Race distribution

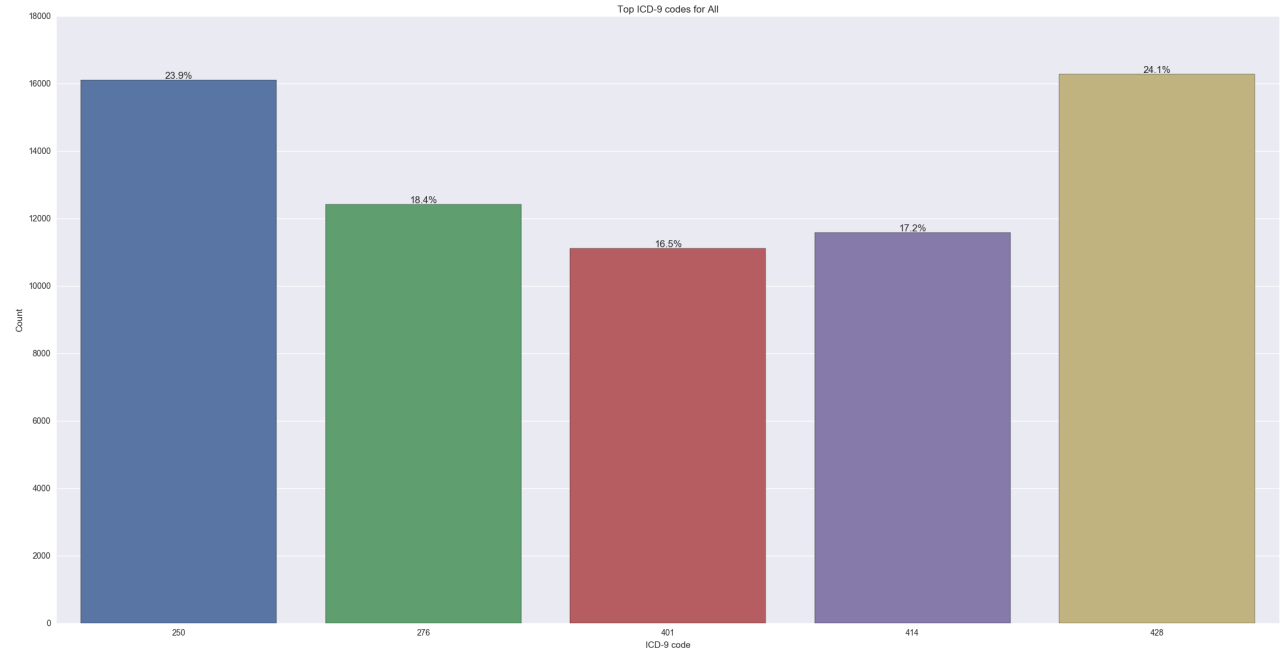


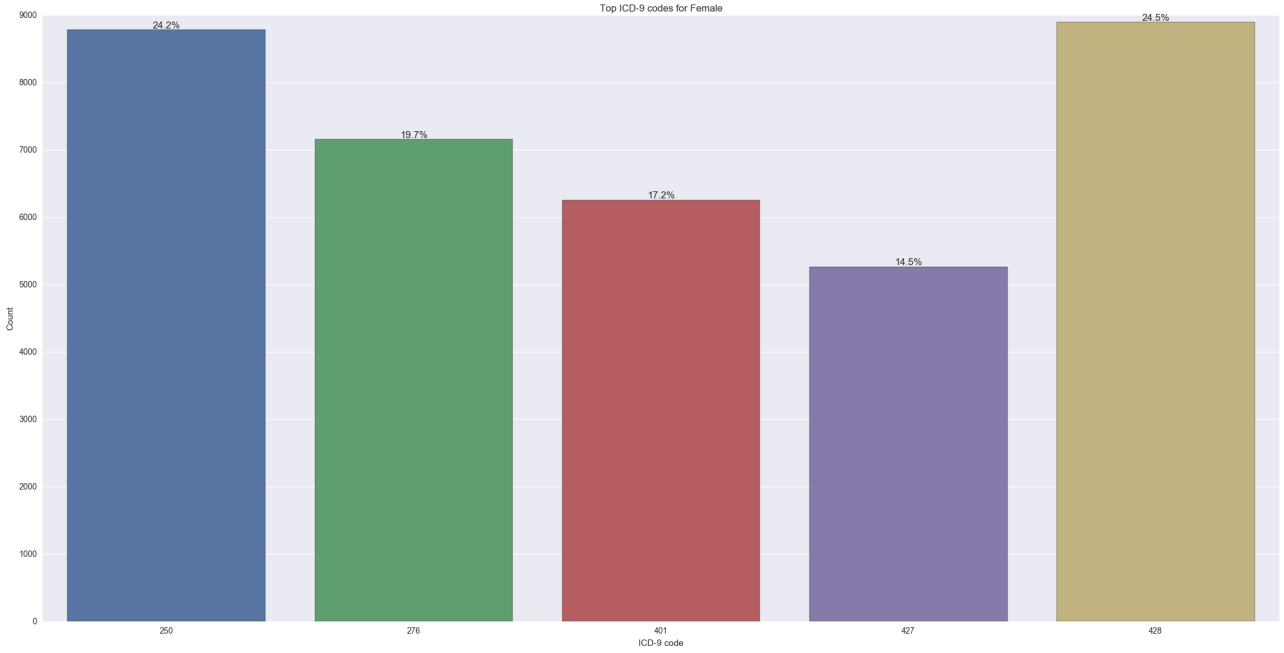
Readmitted distribution

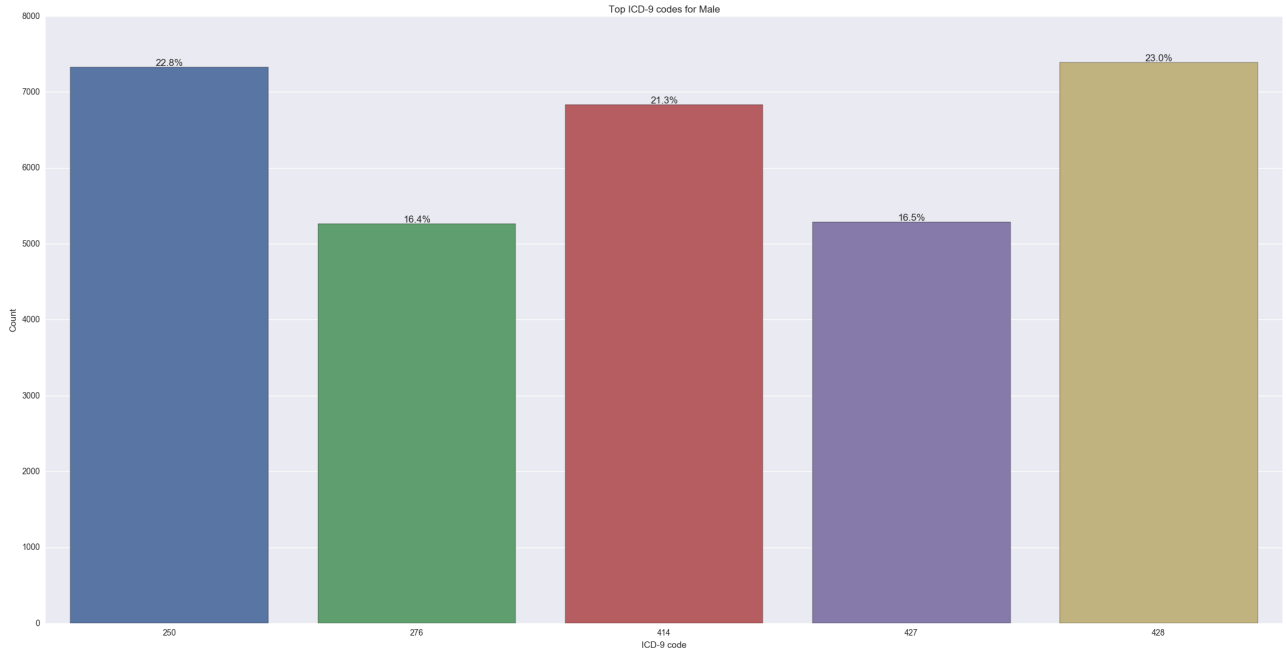


Most frequently used ICD-9 codes distribution

Note: by changing the parameter 'n' in the function part_1_1_helper, you can get the top n ICD-9 codes for each gender. I have chosen n=5 to keep the graphs readable.

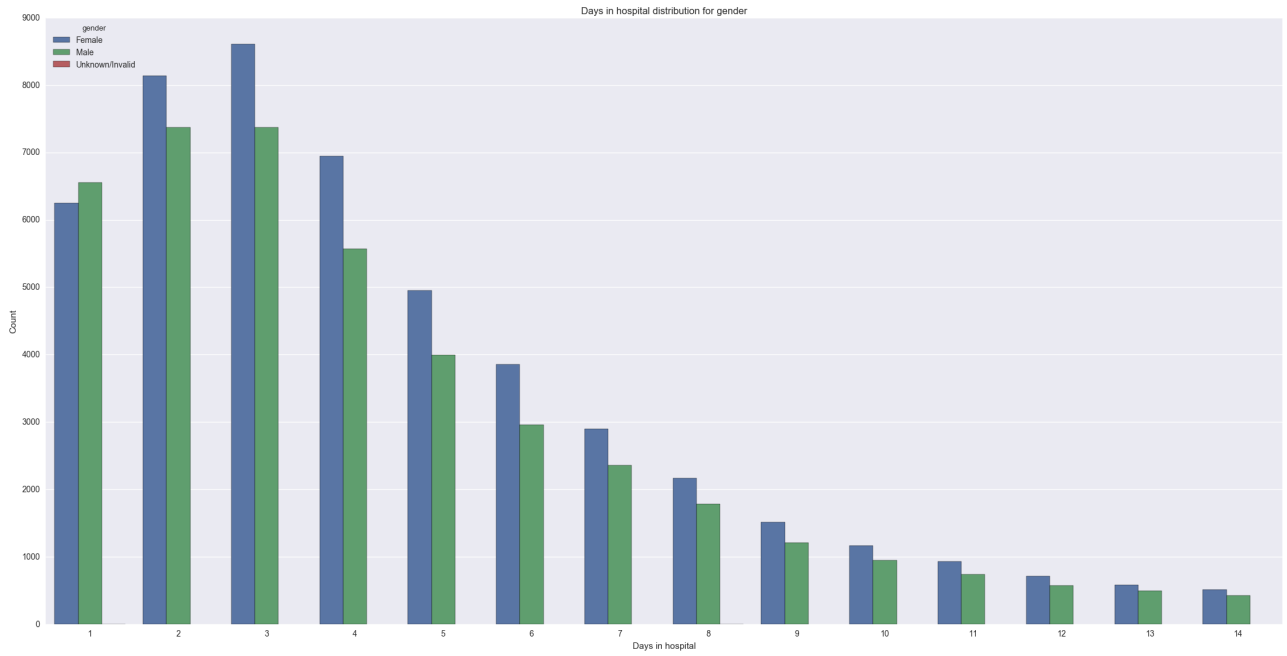

Top ICD-9 codes for All
Count
ICD-9 code
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
23.9%
18.4%
16.5%
17.2%
24.1%
250
276
401
414
428


Top ICD-9 codes for Female
Count
ICD-9 code
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
24.2%
19.7%
17.2%
14.5%
24.5%
250
276
401
427
428

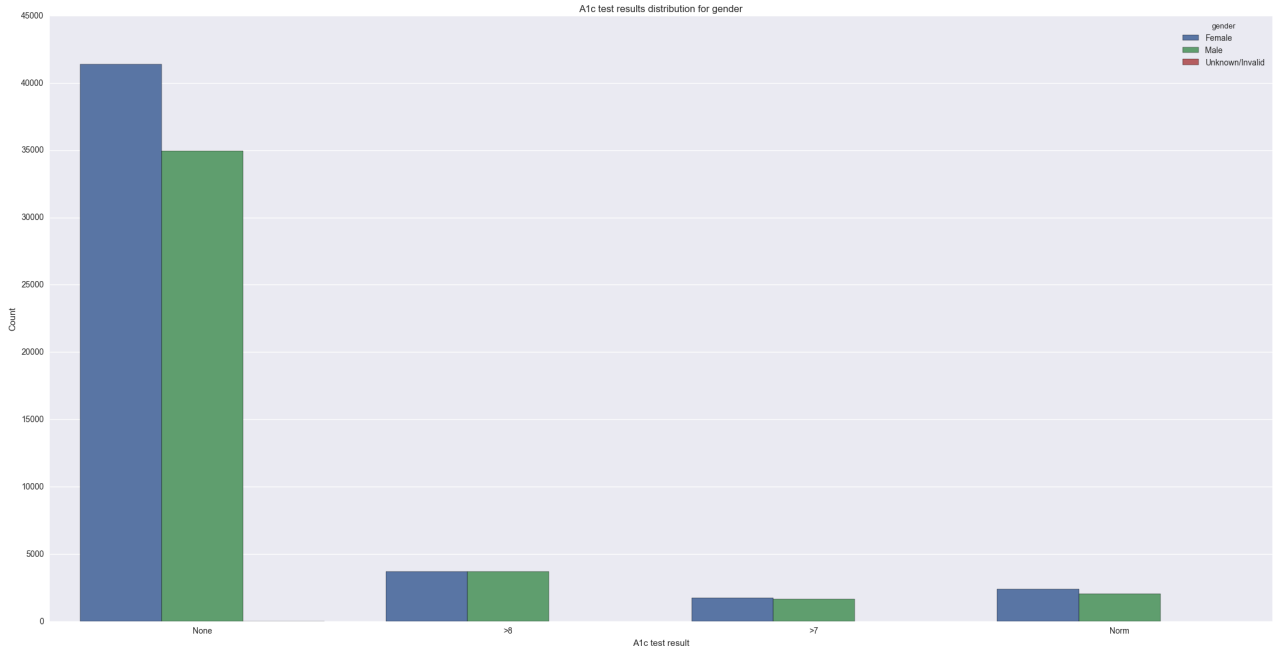


Subpart 2

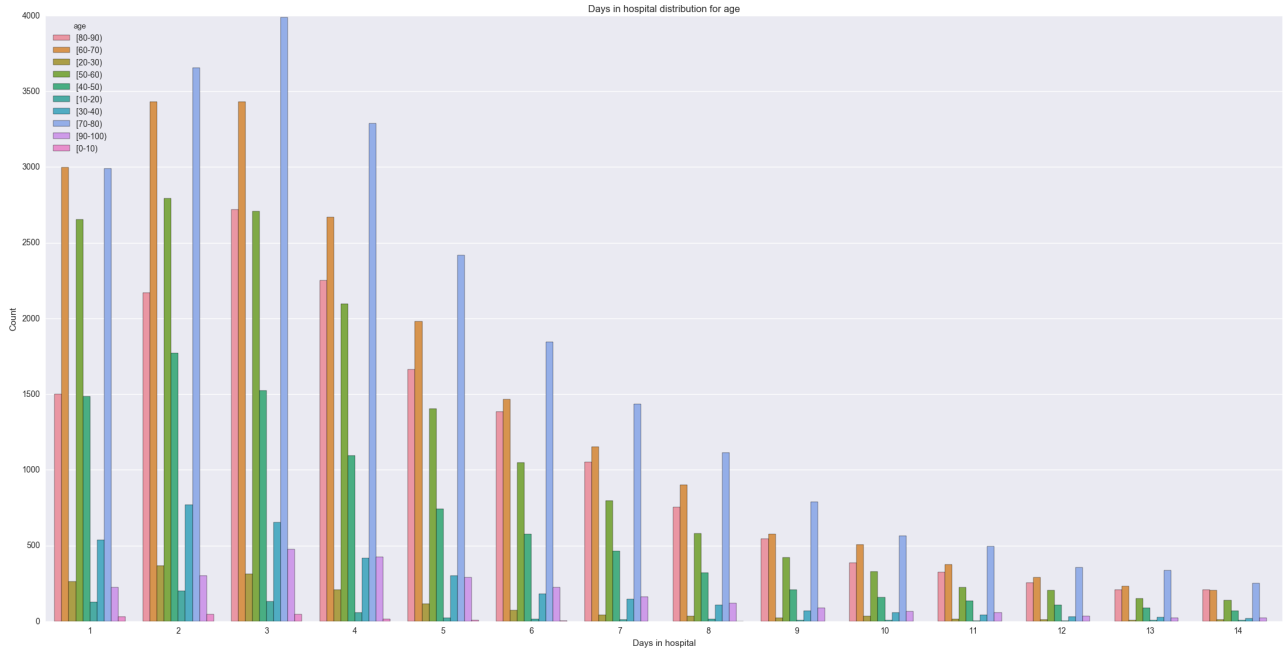
Days in hospital per gender



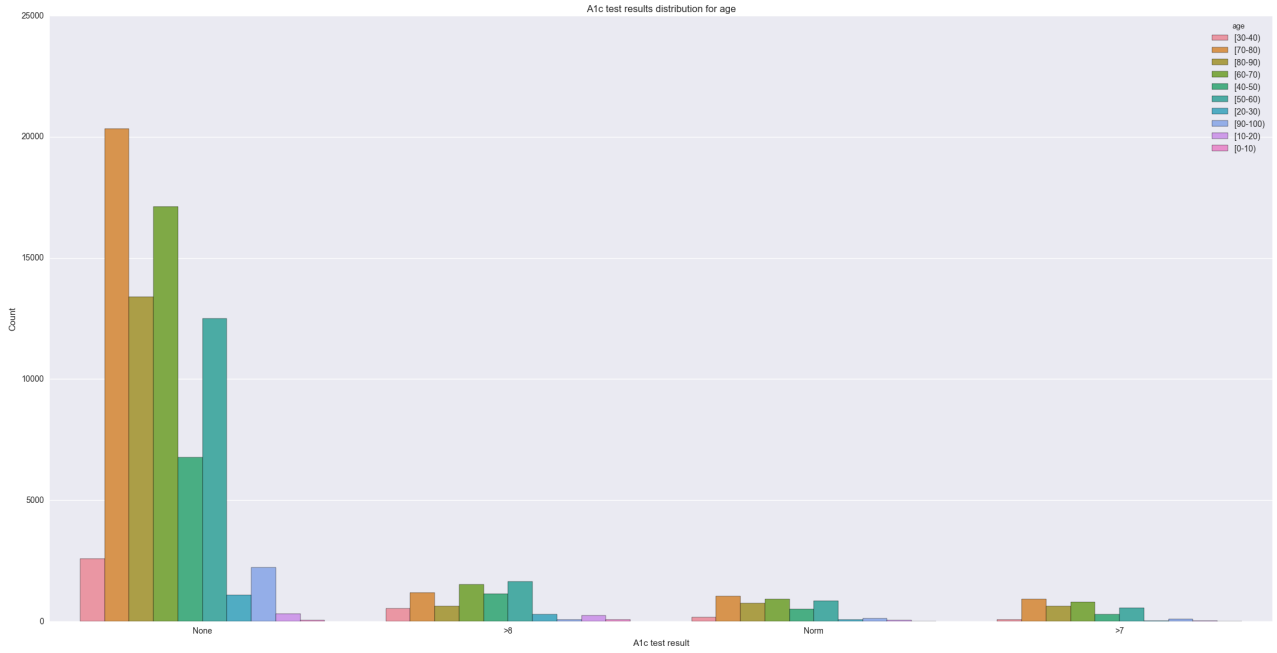
A1C codes per gender



Days in hospital per age group



A1C codes per age group



Part 2

We can regard the hypothesis tests as a form of feature selection - if it is statistically significant that a treatment influences readmission values, we hope that including this treatment as a feature will improve our predictions.

Subpart 1

For this subpart, the treatment is performing a measurement of HbA1c. Let's denote 'failure' of the treatment as readmission within 30 days and denote 'success' as the complimentary. Let's look at the data:

$$
\begin{aligned}
A &= \text{total number of patients who weren't measured for HbA1c (untreated group)} \\
B &= \text{number of patients who were readmitted who weren't measured for HbA1c} \\
C &= \text{number of patients with reduced readmission who weren't measured for HbA1c} \\
D &= \text{total number of patients who were measured for HbA1c (treated group)} \\
E &= \text{number of patients who were readmitted who were measured for HbA1c} \\
F &= \text{number of patients with reduced readmission who were measured for HbA1c}
\end{aligned}
$$

$$
\begin{aligned}
\Pr(failure|\text{no treatment}) &= \frac{B}{\text{A}} = 0.114222 \\
\Pr(success|\text{no treatment}) &= \frac{C}{A} = 1 - \Pr(failure|\text{no treatment}) = 0.885778 \\
\Pr(failure|\text{treatment}) &= \frac{E}{\text{D}} = 0.098 \\
\Pr(success|\text{treatment}) &= \frac{F}{D} = 0.901
\end{aligned}
$$

We want to check for independece between the two variables.

The chi-squared is a test for independece, meaning that given two categorical variables from a single population, we can use it to determine whether there is a significant assosication between the two variables.

Note that the variables under study (readmission category for treated and untreated patients) are categorical - either a patient was readmitted within less than 30 days, or he wasn't readmitted at all/readmitted after 30 days.

Under the assumption that the patients were sampled using a simple random sample, and if the expected value of the number of sample obervations is at least 5 for each category, we can use the chi-squared test. So, let's assume that indeed the sampling was a simple random sample, and check for the number of sample observations condition.

First we need to calculate:

$$\Pr\left(failure|\text{no treatment}\right) = \frac{B}{\text{A}}$$
$$\Pr\left(success|\text{no treatment}\right) = \frac{C}{A}$$
$$\Pr\left(failure|\text{treatment}\right) = \frac{E}{\text{D}}$$
$$\Pr\left(success|\text{treatment}\right) = \frac{F}{D}$$

Now, let's use this data to calculate:

$$D \cdot \Pr\left(success|\text{no treatment}\right),\ D \cdot \Pr\left(failure|\text{no treatment}\right)$$

According to my program, indeed we get that:

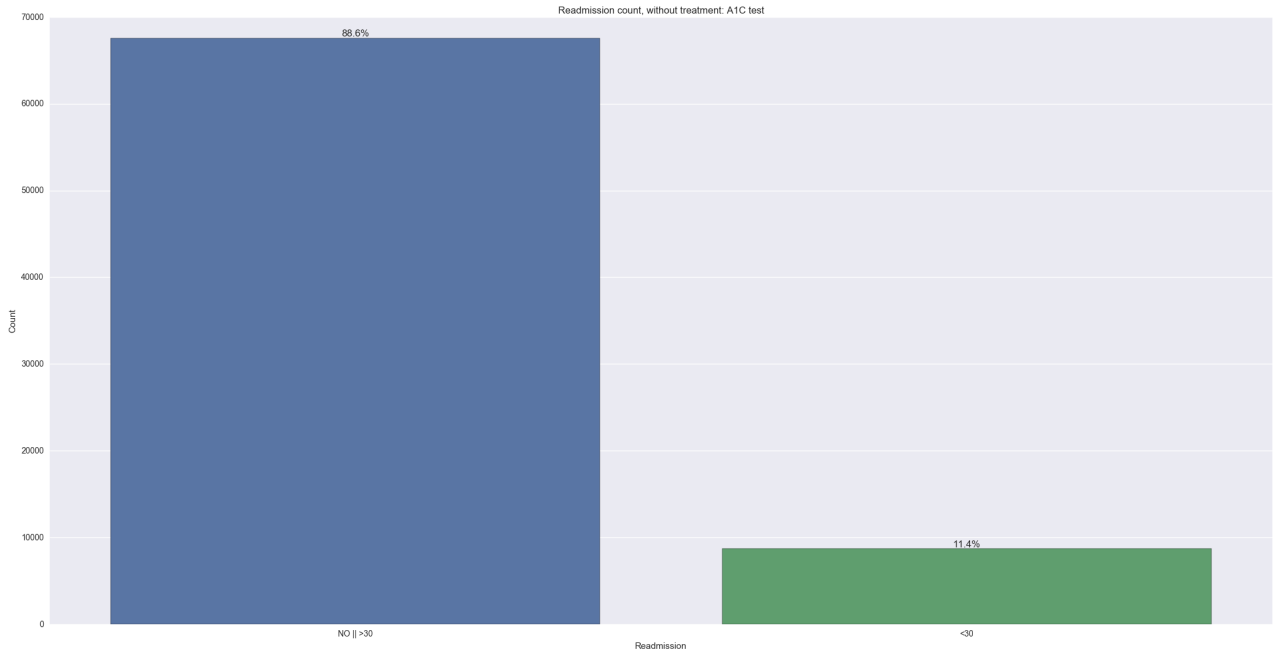
$$D \cdot \Pr\left(success|\text{no treatment}\right) > 5,\ D \cdot \Pr\left(failure|\text{no treatment}\right) > 5$$

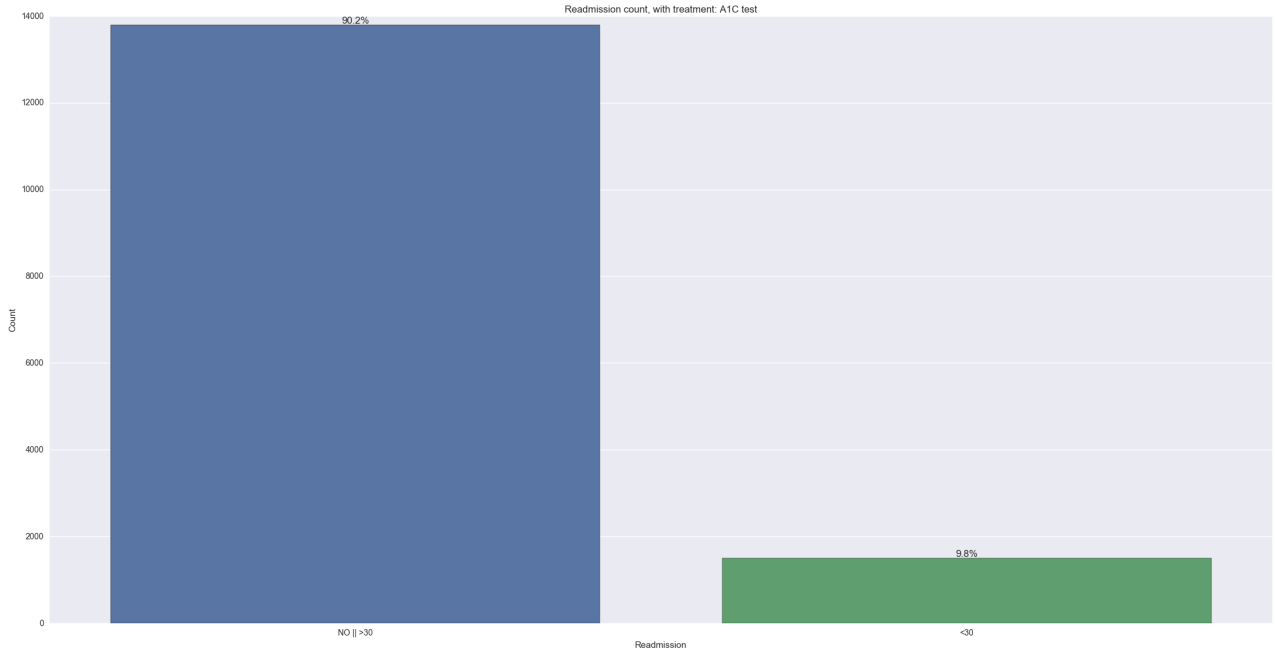
So, the expected value of the number of sample observations is enough, and we can use chi-squared.

Now, we will perform the chi-squared test using $\alpha = 0.05$, and we get:

$$\text{critical-value} = 3.8415$$
$$\chi^2 = 38.93187 > \text{critical-value}$$
$$p = 4.3885521472616642e - 10 < \alpha$$

So, we get that treating indeed changes the distribution of readmission, and from the data we see that the result is a lower readmission rate.





Subpart 2

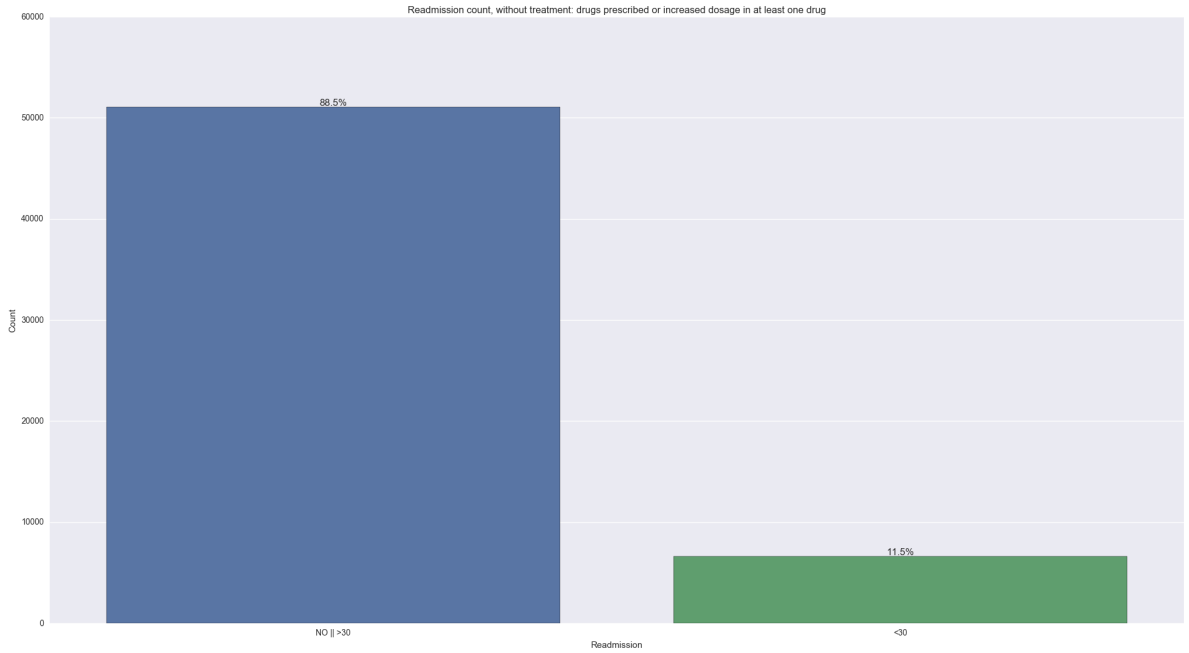
For this subpart, the treatment is prescribing a drug or increasing the dosage of at least one drug.

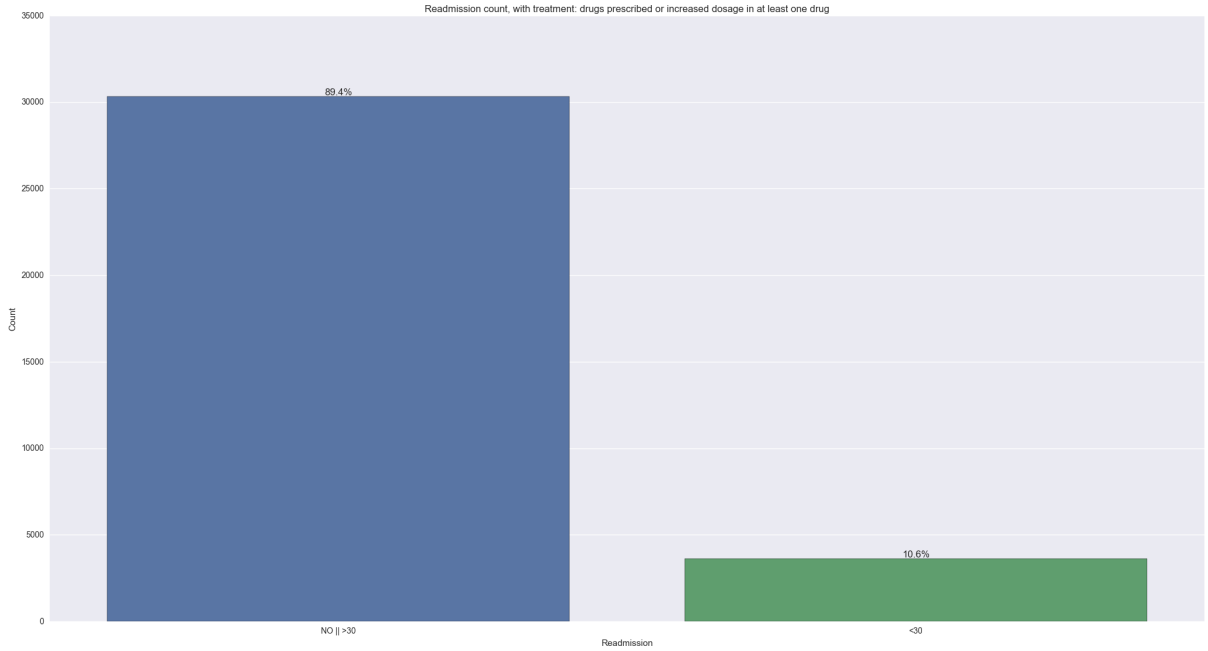
Note that I performed chi-squared here too, for the same reasons, and using the exact same method. You can see all the steps taken to calculate the test by running the code. Here are the results:

$$\begin{aligned} \text{critical-value} &= 3.8415 \\ \chi^2 &= 23.03197 > \text{critical-value} \\ p &= 1.5932960211409451e-06 < \alpha \end{aligned}$$

So, we get that treating indeed changes the distribution of readmission, and from the data we see that the result is a lower readmission rate:

$$\begin{aligned} \Pr\left(failure|\text{no treatment}\right) &= 0.114 \\ \Pr\left(success|\text{no treatment}\right) &= 0.885 \\ \Pr\left(failure|\text{treatment}\right) &= 0.106 \\ \Pr\left(success|\text{treatment}\right) &= 0.893 \end{aligned}$$





Subpart 3

I have performed hypothesis tests on each possible drug, and checked if an increased dosage of each changes the readmission rate.

I've found two nice cases:

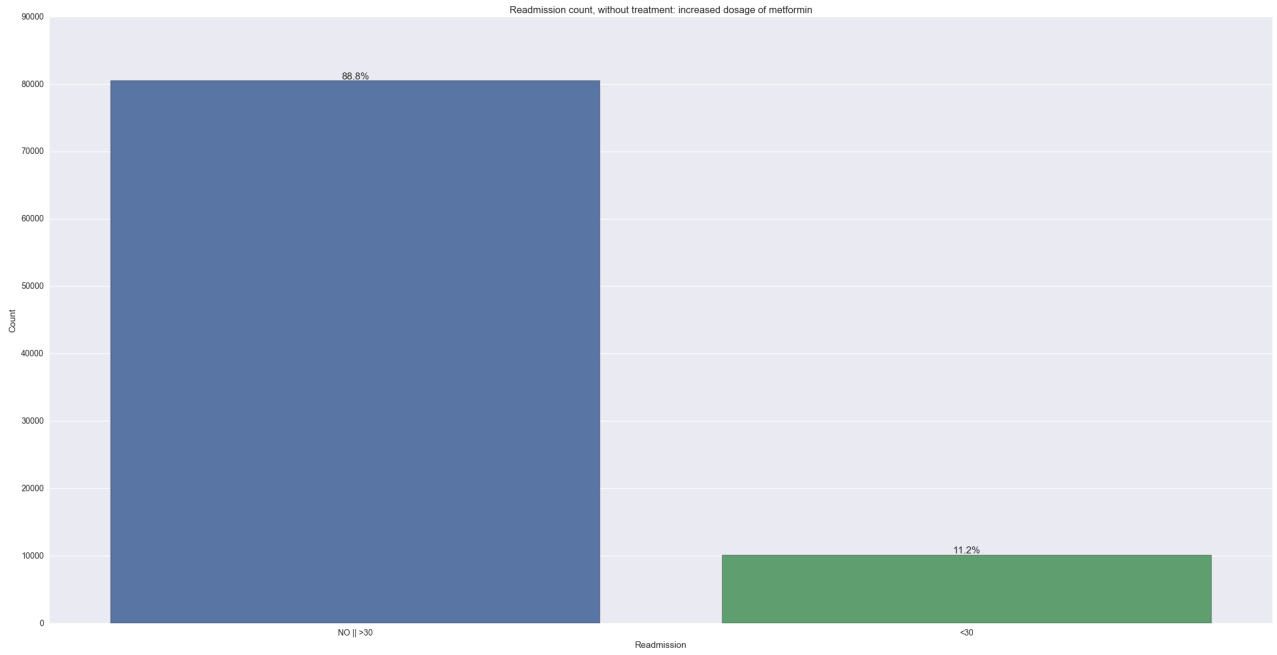
Metformin

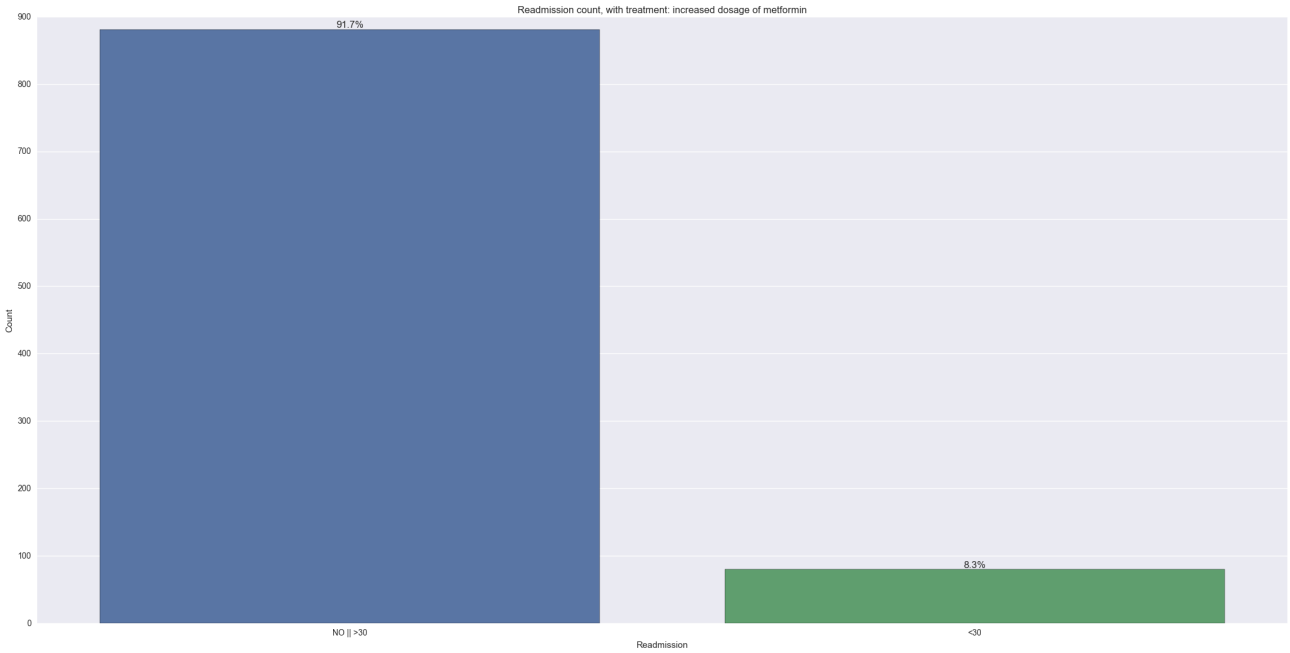
$$\begin{aligned} \text{critical-value} &= 3.8415 \\ \chi^2 &= 7.9107 > \text{critical-value} \\ p &= 0.0049 < \alpha \end{aligned}$$

So, we get that treating indeed changes the distribution of readmission, and from the data we see that the result is a lower readmission rate:

$$\begin{aligned} \Pr\left(failure|\text{no treatment}\right) &= 0.1118 \\ \Pr\left(success|\text{no treatment}\right) &= 0.8881 \\ \Pr\left(failure|\text{treatment}\right) &= 0.0832 \\ \Pr\left(success|\text{treatment}\right) &= 0.9167 \end{aligned}$$

Increased dosage of this drug helped a lot.





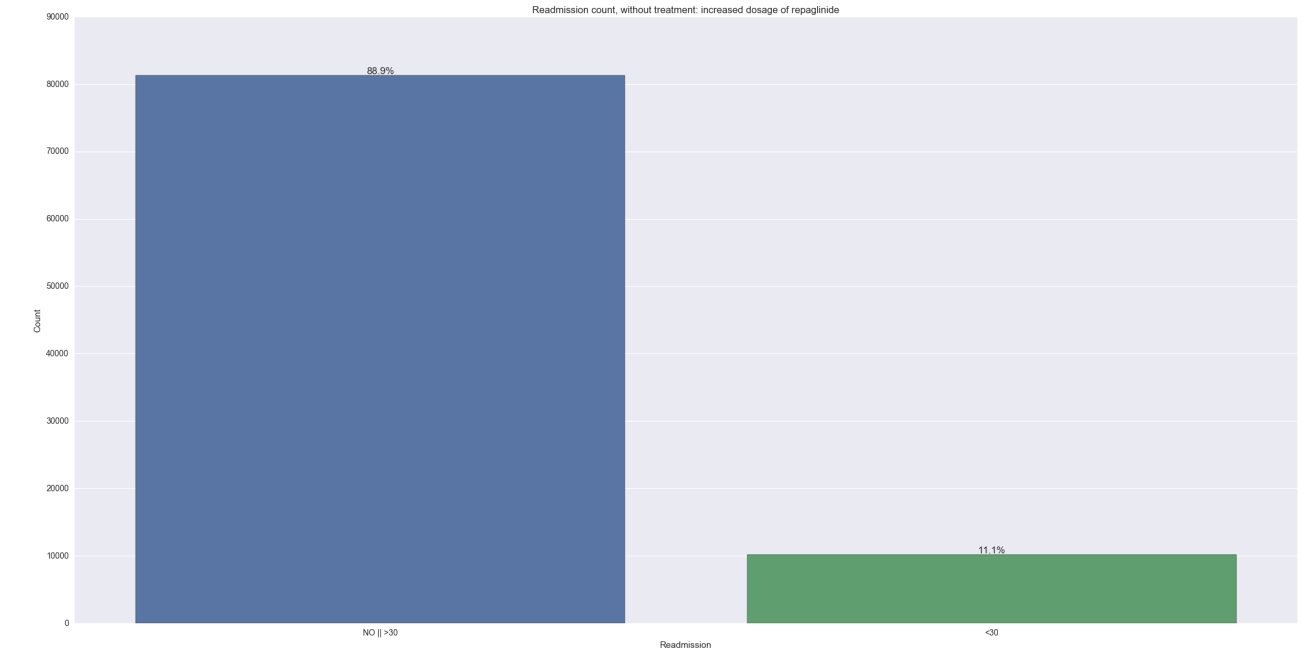
Repaglinide

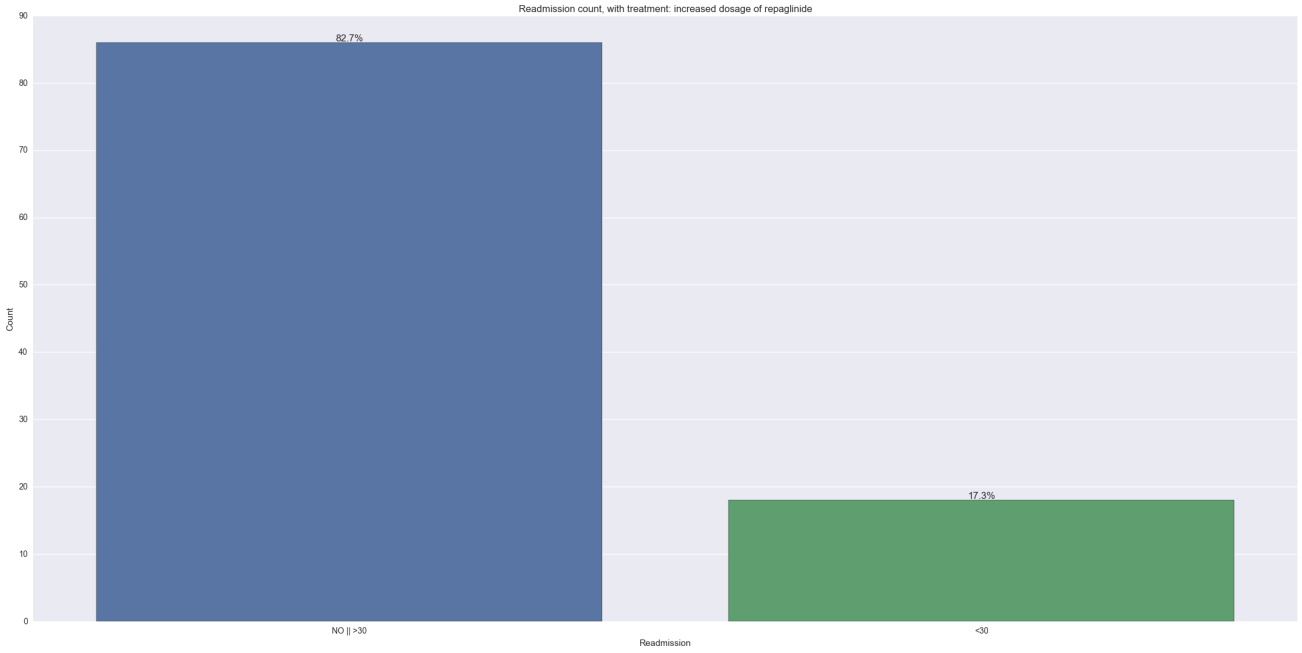
$$\begin{aligned} \text{critical-value} &= 3.8415 \\ \chi^2 &= 3.985 > \text{critical-value} \\ p &= 0.0045 < \alpha \end{aligned}$$

So, we get that treating indeed changes the distribution of readmission, and from the data we see that the result is a lower readmission rate:

$$\begin{aligned} \Pr\left(failure|\text{no treatment}\right) &= 0.1114 \\ \Pr\left(success|\text{no treatment}\right) &= 0.8885 \\ \Pr\left(failure|\text{treatment}\right) &= 0.1730 \\ \Pr\left(success|\text{treatment}\right) &= 0.8269 \end{aligned}$$

Increased dosage of this drug made things worse by a lot.





Thoughts

Note that I've used the chi-squared test in this part, but another possible test is the Kolmogorov–Smirnov test between the treated group and the bernoulli distribution dictated by the untreated group, where

$$\Pr(success) = \frac{\text{number of patients with reduced readmission}}{\text{total number of patients}}$$

Part 3

Defining a metric space on the patients will allow us to later on use standard machine learning methods on our data. Also, performing KMeans will allow us to see if certain features allow easy separation of the data according to readmission values. Of course, if we find such features we will also need to perform hypothesis tests to see if there is stastitical significance between these features and readmission values.

Subpart 1

Note that given two points, x and x', K Means regards $similarity\,(x, x')$ (or $metric\,(x, x')$) as the ℓ_2 difference between the two points:

$$|x - x'|_2$$

So, by defining a vector representation of the data, we are also defining the similarity (metric) for the data. I will explore different vector representations and answer part 3.2 on each.

Initial vectorizer

This vectorizer creates a one hot representation for each of the columns diabetesMed and readmitted, and a0/1 representation for the various disease diagnoses, so we get these columns:

['162', '197', '198', '244', '250', '250.01', '250.02', '250.03', '250.11', '250.13', '250.4', '250.6', '250.7', '250.8', '250.82', '272', '276', '278', '280', '285', '287', '294', '295', '296', '303', '305', '38', '401', '402', '403', '404', '41', '410', '411', '413', '414', '415', '424', '425', '426', '427', '428', '433', '434', '435', '438', '440', '453', '458', '486', '491', '493', '496', '507', '511', '518', '530', '535', '536', '558', '560', '562', '571', '574', '577', '578', '584', '585', '593', '599', '648', '682', '70', '707', '715', '722', '724', '730', '733', '780', '784', '785', '786', '787', '788', '789', '790', '799', '8', '820', '996', '997', '998', '?', 'V45', 'V57', 'V58', 'diabetesMed=No', 'diabetesMed=Yes', 'readmitted=<30', 'readmitted=>30', 'readmitted=NO']

Where the numbers represent ICD-9 codes.

Note that this indicates that if x and x' are different on one diagnoses then:

$$|x - x'|_2 = 1$$

But if x and x' are different only in the readmitted value or only in the diabetesMed value, then:

$$|x - x'|_2 = \sqrt{2}$$

This is something we want, in order to separate patients more based on readmitted and diabetesMed values, because:

1. We saw in part 2 that the diabetesMed column influences the readmitted column.
2. We want patients with the same readmitted value to cluster together. Then, we could look at other shared features they have, and if there is a certain feature that seems to cluster together with the same readmitted value, it might be useful later, for prediction.

More features vectorizer

This vectorizer is similar to the previous one, but instead of the diabetesMed feature there is a feature for "either a drug was perscribed or there was an increase of dosage for at least one drug".

Subpart 2

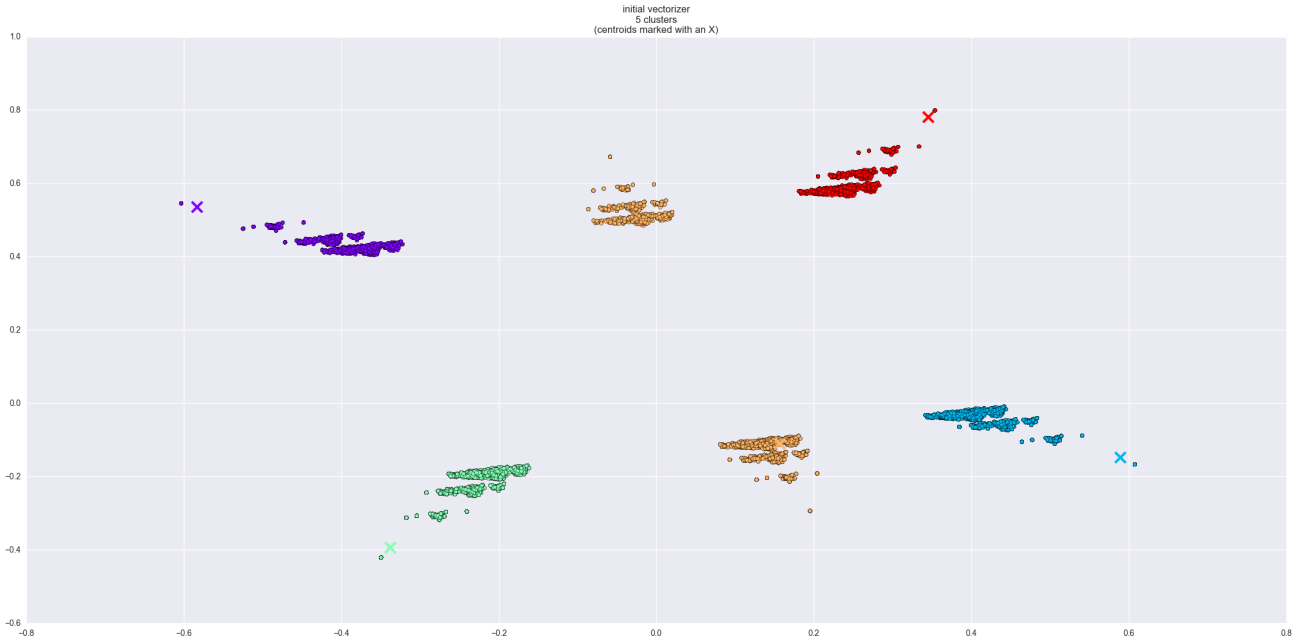
I've run the KMeans algorithm using both vectorizers twice: once with 5 clusters, and once with 10. After receiving the clusters, I've normalized the data and performed PCA to bring the results down to 2 dimensions, in order to make it more understandable.

The resulting clusters are:

Initial vectorizer

5 clusters

After the normalization and the dimensionality reduction, the data looks like a magen david:



The score for 'initial vectorizer' based clustering on the data is

$$-209425.30462155314$$

There are 5 clusters. The centroids are:

1. [{'diabetesMed=No': 1.0, 'readmitted=NO': 1.0}]
2. [{'diabetesMed=Yes': 1.0, 'readmitted=>30': 1.0}]
3. [{'diabetesMed=Yes': 1.0, 'readmitted=NO': 1.0}]
4. [{'readmitted=<30': 1.0, 'diabetesMed=Yes': 1.0}]
5. [{'diabetesMed=No': 1.0, 'readmitted=>30': 1.0}]

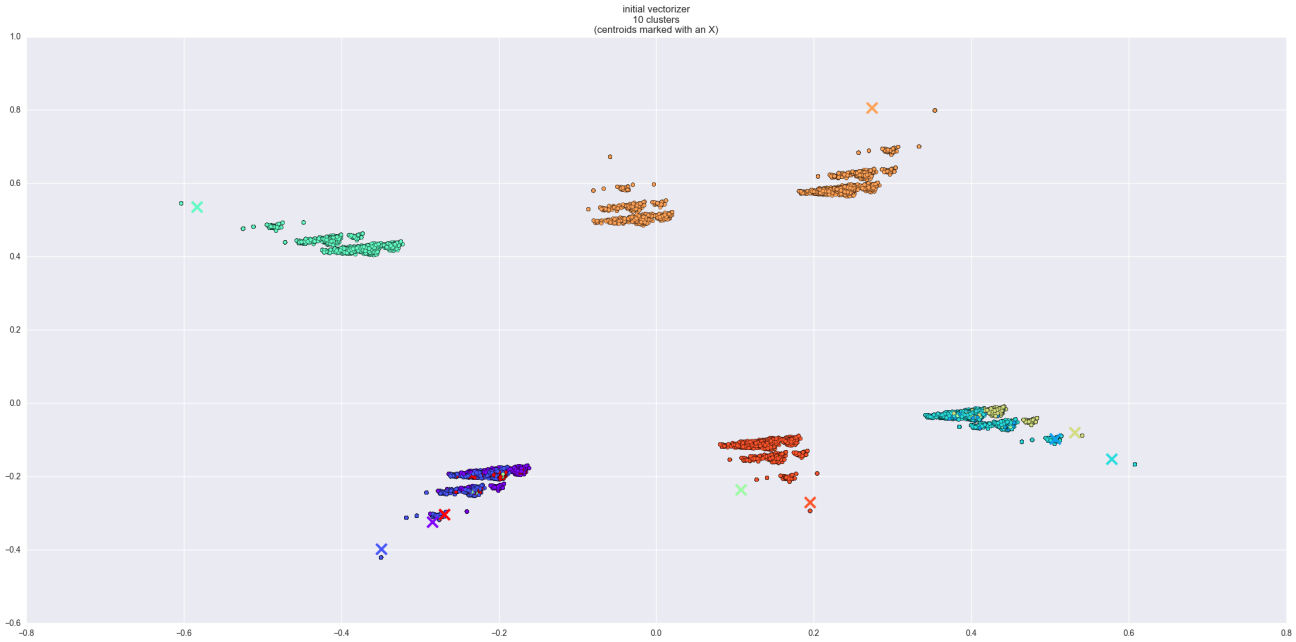
Note: no values (for example, for drugs) means that the centroid had a 0 there.

The percentage of patients allocated to each centroid is:

1. 13.657754
2. 27.840680
3. 40.240640
4. 11.126882
5. 7.134044

Note that the data clusters very nicely, and that no diagnosis feature found itself into any of the centroids. Not a lot can be learnt from these clusters.

10 clusters



The score for 'initial vectorizer' based clustering on the data is

$$-192528.0869948132$$

There are 10 clusters. The centroids are:

1. [{'428': 1.0, 'diabetesMed=Yes': 1.0, '414': 1.0, 'readmitted=NO': 1.0}]
2. [{'diabetesMed=Yes': 1.0, 'readmitted=NO': 1.0}]
3. [{'276': 1.0, 'diabetesMed=Yes': 1.0, 'readmitted=>30': 1.0}]
4. [{'diabetesMed=Yes': 1.0, 'readmitted=>30': 1.0}]
5. [{'diabetesMed=No': 1.0, 'readmitted=NO': 1.0}]
6. [{'707': 1.0, 'diabetesMed=Yes': 1.0, 'readmitted=NO': 1.0}]
7. [{'428': 1.0, 'diabetesMed=Yes': 1.0, 'readmitted=>30': 1.0}]
8. [{'diabetesMed=No': 1.0, 'readmitted=>30': 1.0}]
9. [{'readmitted=<30': 1.0, 'diabetesMed=Yes': 1.0}]
10. [{'276': 1.0, 'diabetesMed=Yes': 1.0, 'readmitted=NO': 1.0}]

Note: no values (for example, for drugs) means that the centroid had a 0 there.
The percentage of patients allocated to each centroid is:

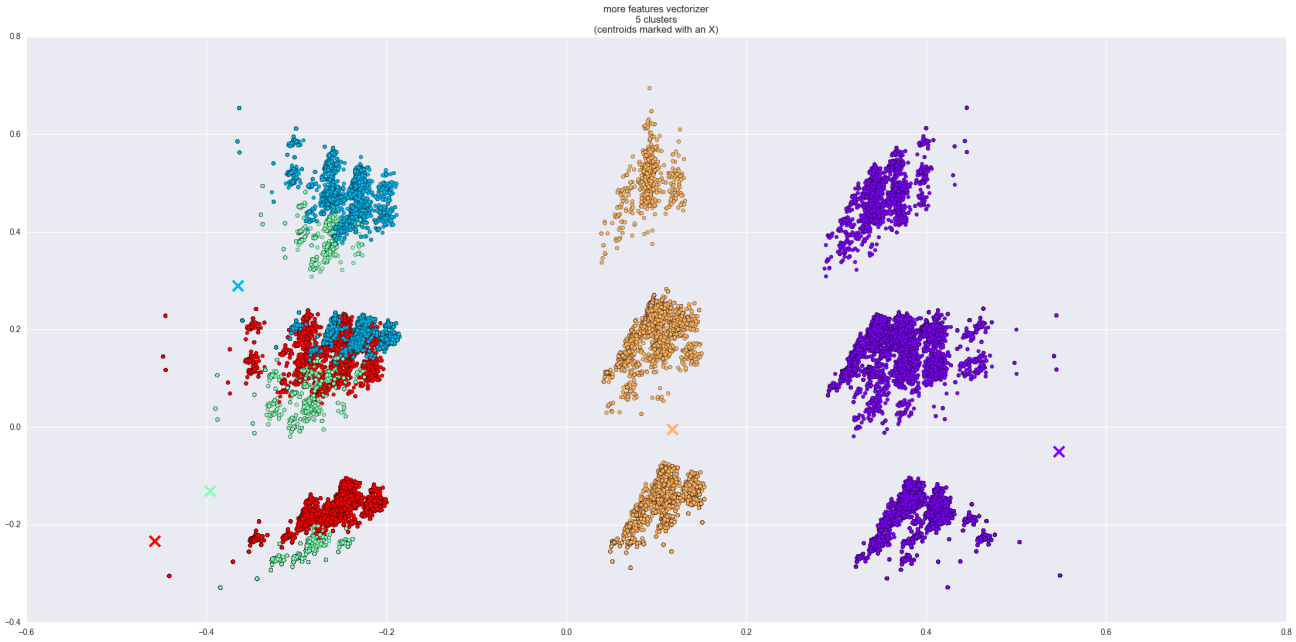
1. 9.754446
2. 24.375198
3. 3.594318
4. 17.855856
5. 13.657754
6. 2.115975

7. 5.423140
8. 9.332999
9. 8.852591
10. 5.037723

Here we can already learn that maybe certain diagnoses might reduce the readmission rate, for example from clusters $1, 3, 6, 7, 10$ we can learn that ICD-9 diagnoses $428, 276$ might be correlated with reduced readmission. Hypothesis testing should be done to reaffirm this.

More features vectorizer

5 clusters



The score for 'more features vectorizer' based clustering on the data is

$$-236611.98502714749$$

There are 5 clusters. The centroids are:

1. [{'A1Cresult=None': 1.0, 'readmitted=>30': 1.0}]
2. [{'medicine prescribed or increased dosage': 1.0, 'A1Cresult=None': 1.0, 'readmitted=NO': 1.0}]
3. [{'250': 1.0, 'A1Cresult=None': 1.0, 'readmitted=NO': 1.0}]
4. [{'readmitted=<30': 1.0, 'A1Cresult=None': 1.0}]
5. [{'A1Cresult=None': 1.0, 'readmitted=NO': 1.0}]

Note: no values (for example, for drugs) means that the centroid had a 0 there.
The percentage of patients allocated to each centroid is:

1. 34.974724
2. 16.829532
3. 10.918342
4. 11.126882
5. 26.150520

10 clusters

The score for 'more features vectorizer' based clustering on the data is

$$-215534.97350704888$$

There are 10 clusters. The centroids are:

1. [{'medicine prescribed or increased dosage': 1.0, 'A1Cresult=None': 1.0, 'readmitted=NO': 1.0}]
2. [{'medicine prescribed or increased dosage': 1.0, 'A1Cresult=None': 1.0, 'readmitted=>30': 1.0}]
3. [{'A1Cresult=None': 1.0, 'readmitted=>30': 1.0}]
4. [{'A1Cresult=None': 1.0, 'readmitted=NO': 1.0}]
5. [{'401': 1.0, '427': 1.0, 'A1Cresult=None': 1.0, 'readmitted=>30': 1.0}]
6. [{'276': 1.0, '428': 1.0, 'A1Cresult=None': 1.0, 'readmitted=NO': 1.0}]
7. [{'readmitted=<30': 1.0, 'A1Cresult=None': 1.0}]
8. [{'A1Cresult=Norm': 1.0, 'readmitted=NO': 1.0}]
9. [{'A1Cresult=>8': 1.0, 'readmitted=NO': 1.0}]
10. [{'A1Cresult=None': 1.0, '414': 1.0, 'readmitted=NO': 1.0}]

Note: no values (for example, for drugs) means that the centroid had a 0 there.

The percentage of patients allocated to each centroid is:

1. 15.518239
2. 10.641016
3. 15.433076
4. 18.347181
5. 4.461234
6. 6.683117

7. 9.862538
8. 4.888142
9. 8.074114
10. 6.091343

The results reaffirm our conclusion from part 2 that if a medicine was prescribed or dosage was increased for at least one drug, there is a correlation with less readmission.

Also, because $401, 414, 427, 428$ appear a lot with either no readmission or readmission after 30 days, the results suggest we should perform an hypothesis test on diseases from the group $390 - 459$ to check if indeed they are correlated with less readmission.

Thoughts

Note that it's very easy to change the code to add more metrics. For example, if you want to add a certain feature to the metric, simply add the column name of the feature to the appropriate part_3_1 function.

Part 4

All methods produced bad results - binary classification never goes above 90% success rate (even a dummy model which always says "no readmission or readmission after 30 days" gets 90%), and ternary classification never goes above 57.5% .

Subpart 1

Gaussian Naive Bayes performs ternary classification better when used together with 'initial vectorizer' than with 'more features vectorizer', but performs binary classification worse.

'initial vectorizer'

The mean test error of the binary classification is:

0.67612916923390975

The median test error of the binary classification is:

0.67551100628930816

The mean test error of the ternary classification is:

0.55553615531644751.

The median test error of the ternary classification is:

0.55242212832858406

The mean test confusion matrix of the ternary classification is:

	NO	>30	<30
NO	2553	2007.55	937
>30	1197.66	1693.11	658.11
<30	359	494.11	277

The median test confusion matrix of the ternary classification is:

	NO	>30	<30
NO	2558	2004	943
>30	1210	1697	654
<30	351	487	280

The validation error of the binary classification is:

0.88346271003242605

The validation error of the ternary classification is:

0.47076741672398548

The validation confusion matrix of the ternary classification is:

	NO	>30	<30
NO	5386	0	0
>30	3605	0	0
<30	1186	0	0

'more features vectorizer'

The mean test error of the binary classification is:

0.67354147109783657

The median test error of the binary classification is:

0.67406897907045293

The mean test error of the ternary classification is:

0.56215268507892491

The median test error of the ternary classification is:

0.56191037735849059

The mean test confusion matrix of the ternary classification is:

	NO	>30	<30
NO	2469.33	2055.88	965.77
>30	1172	1696.55	677.44
<30	348.44	501.22	289.88

The median test confusion matrix of the ternary classification is:

	NO	>30	<30
NO	2488	2040	948
>30	1180	1696	670
<30	351	502	287

The validation error of the binary classification is:

0.89181487668271597

The validation error of the ternary classification is:

0.46497003046084306

The validation confusion matrix of the ternary classification is:

	NO	>30	<30
NO	5445	0	0
>30	3631	0	0
<30	1101	0	0

Subpart 2

I have performed KNN with the following K values:

$$1, 2, 3, 4$$

It seems that $K = 4$ is already too much for my computer, so I didn't try any other K values.

$K = 3$ together with 'more features vectorizer' produces the best validation results for binary classification of all classifiers

The mean test error of the binary classification is:

0.85932801326891384

The median test error of the binary classification is:

0.86075078616352196

The mean test error of the ternary classification is:

0.50988655475752132

The median test error of the ternary classification is:

0.51061320754716977

The mean test confusion matrix of the ternary classification is:

	NO	>30	<30
NO	3861.33	1433	196.66
>30	2324.33	1066.77	154.88
<30	746.33	333.66	59.55

The median test confusion matrix of the ternary classification is:

	NO	>30	<30
NO	3845	1452	196
>30	2337	1076	161
<30	735	338	61

The validation error of the binary classification is:

0.89181487668271597

The validation error of the ternary classification is:

0.49071435590056006

The validation confusion matrix of the ternary classification is:

	NO	>30	<30
NO	4712	733	0
>30	3160	471	0
<30	978	123	0

$K = 1$ together with 'initial vectorizer' produces the best validation results for ternary classification of all classifiers

The mean test error of the binary classification is:

0.80504212514832685

The median test error of the binary classification is:

0.80554190822442762

The mean test error of the ternary classification is:

0.56256767552716636

The median test error of the ternary classification is:

0.56067603419475287

The mean test confusion matrix of the ternary classification is:

	NO	>30	<30
NO	2964.66	1951.77	571.44
>30	1789.22	1357.11	408.22
<30	578.55	425.77	129.77

The median test confusion matrix of the ternary classification is:

	NO	>30	<30
NO	2946	1967	562
>30	1790	1354	403
<30	578	426	132

The validation error of the binary classification is:

0.88700009826078408

The validation error of the ternary classification is:

0.57580819494939572

The validation confusion matrix of the ternary classification is:

	NO	>30	<30
NO	1457	4016	0
>30	694	2860	0
<30	220	930	0

Subpart 3

Note: I have used many machine learning for predictions, including:

1. Quadriatic Discriminant Analysis
2. Multi-Layer Perceptron
3. Decision Trees
4. Random Forest

None of them produce better results than the Naive Bayes and KNN, but both decision trees and random forests are nice to look at because they allow us to extract features that they believe that are better to separate the data. Let's look at decision trees when used with the initial vectorizer:

The mean test error of the binary classification is:

0.87431901534317069

The median test error of the binary classification is:

0.87470518867924529

The mean test error of the ternary classification is:

0.48214303693081095

The median test error of the ternary classification is:

0.48137958140905962

The mean test confusion matrix of the ternary classification is:

	NO	>30	<30
NO	4259.88	1113.77	111.33
>30	2498.55	975.88	84.77
<30	788.11	310	34.22

The median test confusion matrix of the ternary classification is:

	NO	>30	<30
NO	4272	1106	112
>30	2493	979	82
<30	785	309	35

The validation error of the binary classification is:

0.88542792571484719

The validation error of the ternary classification is:

0.45966394811830597

The validation confusion matrix of the ternary classification is:

	NO	>30	<30
NO	5499	0	0
>30	3512	0	0
<30	1166	0	0

Feature ranking for 'Decision Tree with representation: initial vectorizer' for binary classification (in parenthesis - the "importance"):

1. feature '276' (0.037438)
2. feature '427' (0.036107)
3. feature '414' (0.032409)
4. feature '599' (0.032252)
5. feature '780' (0.023436)
6. feature '496' (0.022327)

Feature ranking for 'Decision Tree with representation: initial vectorizer' for ternary classification (in parenthesis - the "importance"):

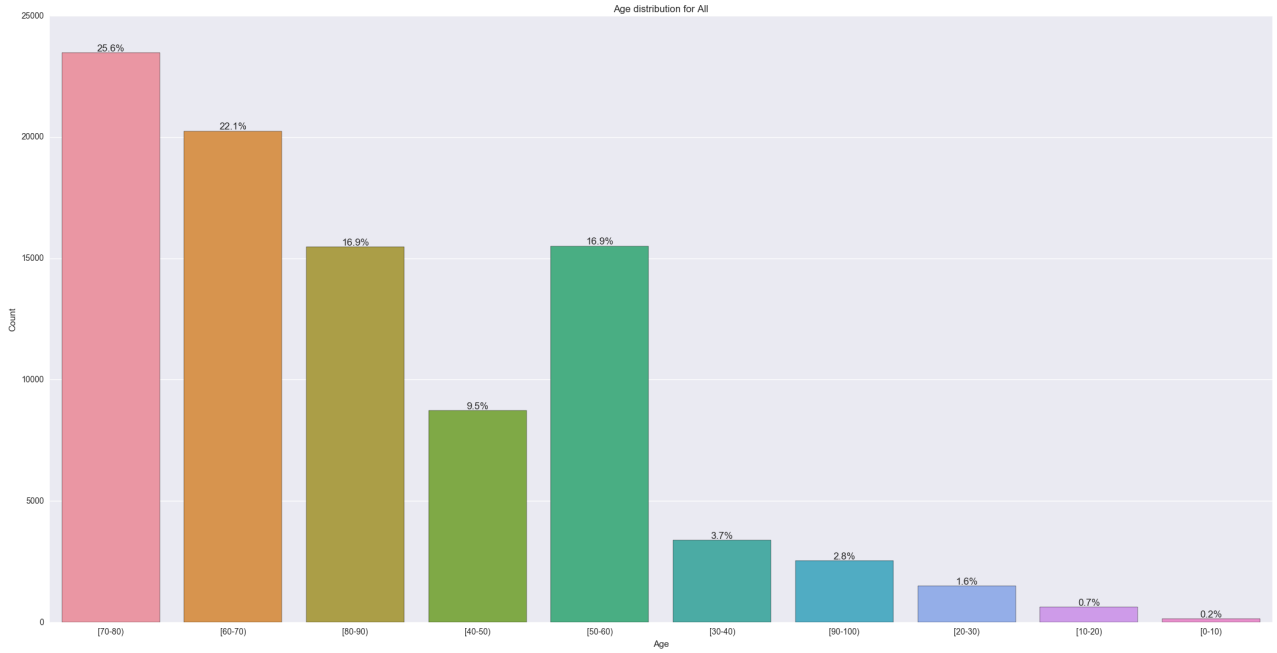
1. feature '276' (0.030455)
2. feature '427' (0.027460)
3. feature '414' (0.026725)
4. feature '599' (0.026156)
5. feature '780' (0.023915)
6. feature '786' (0.021213)

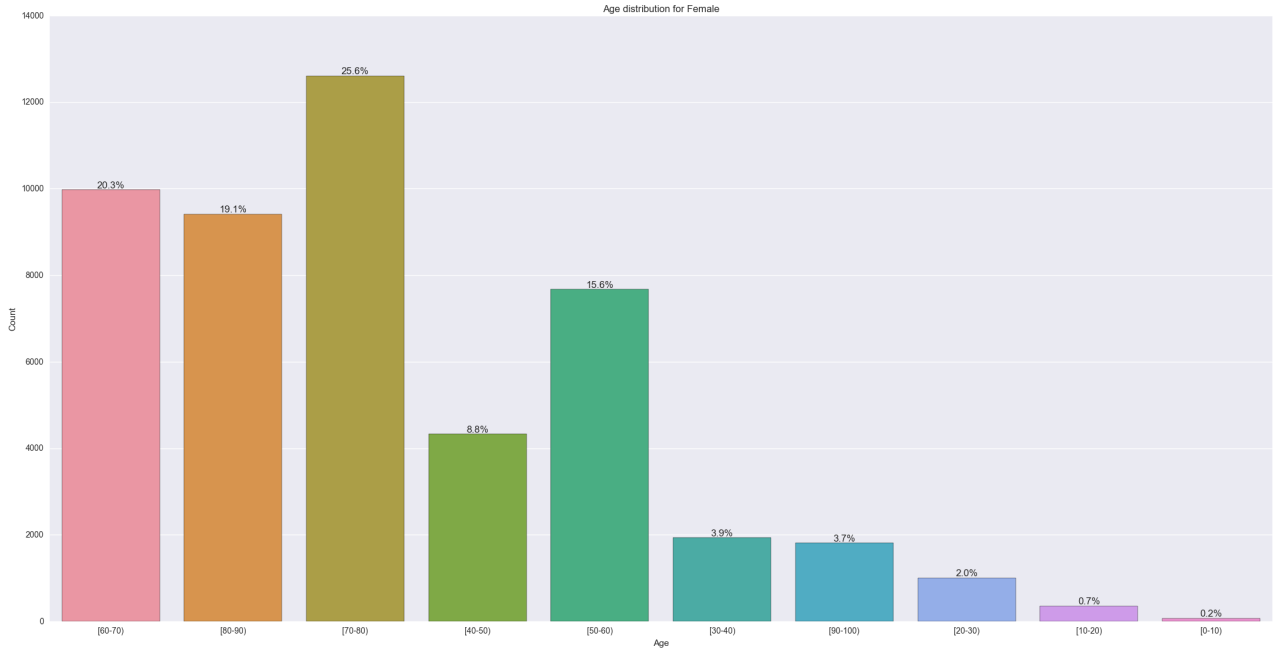
It's interesting to note that the classifier thinks that different features are important for the two classification tasks. It would be nice to perform an hypothesis test on each such feature.

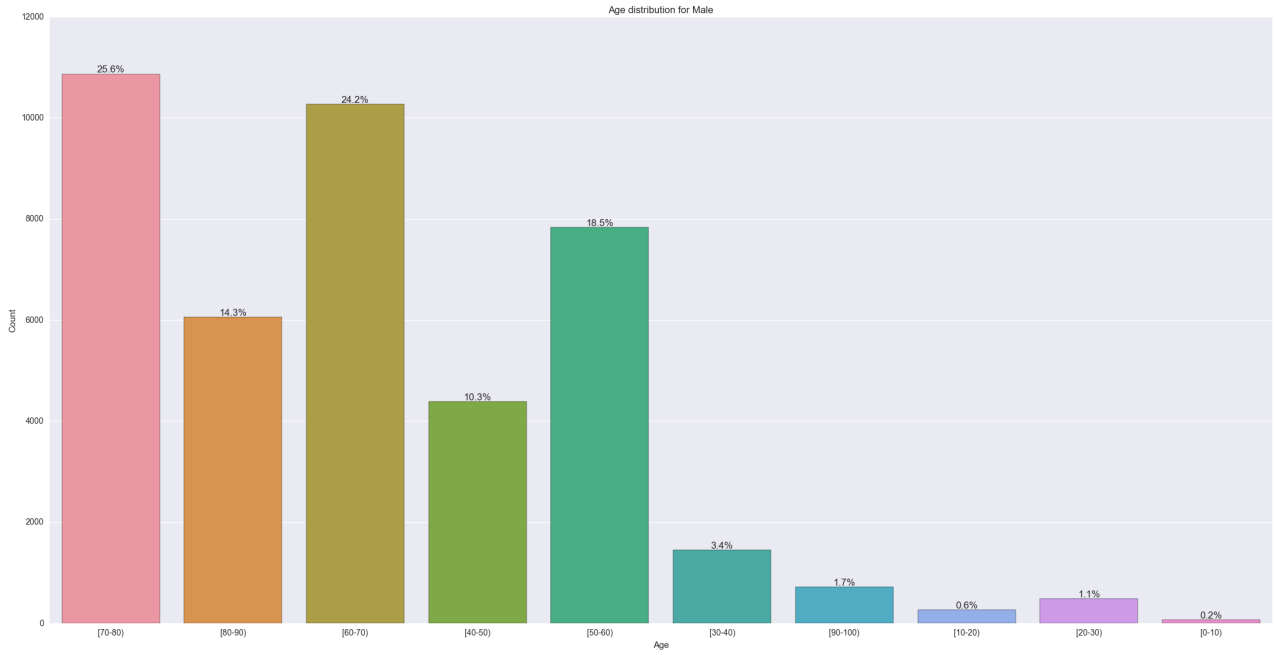
More Graphs

Subpart 1

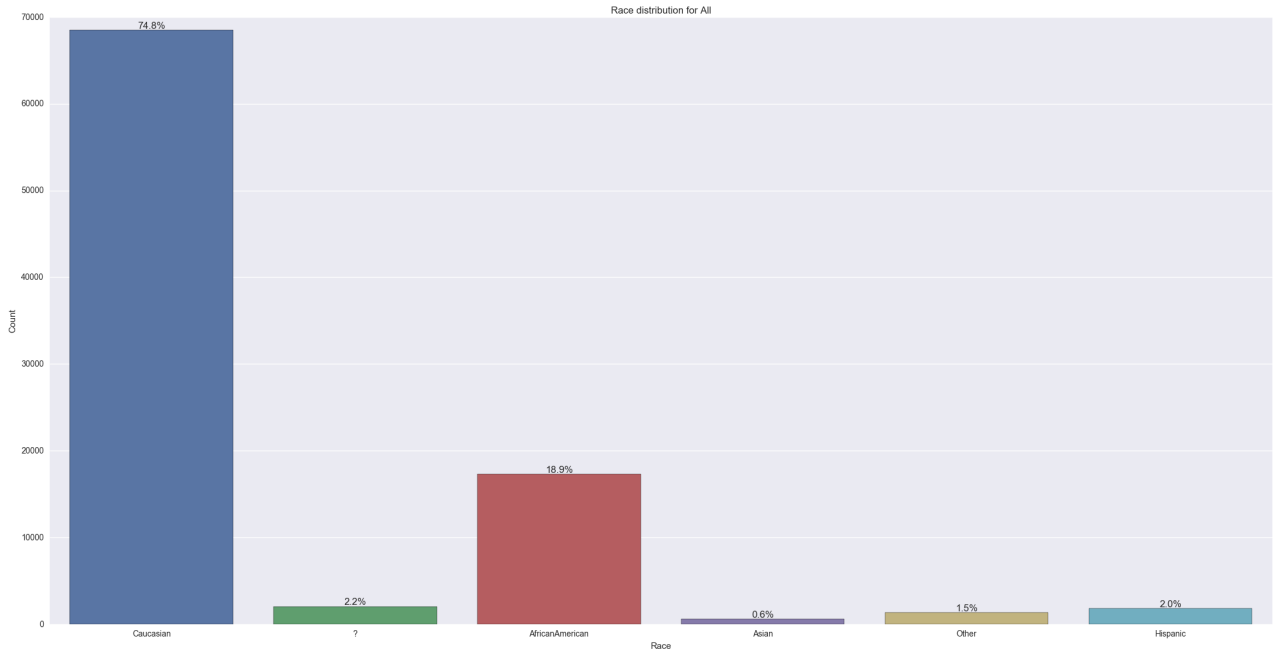
Age distribution

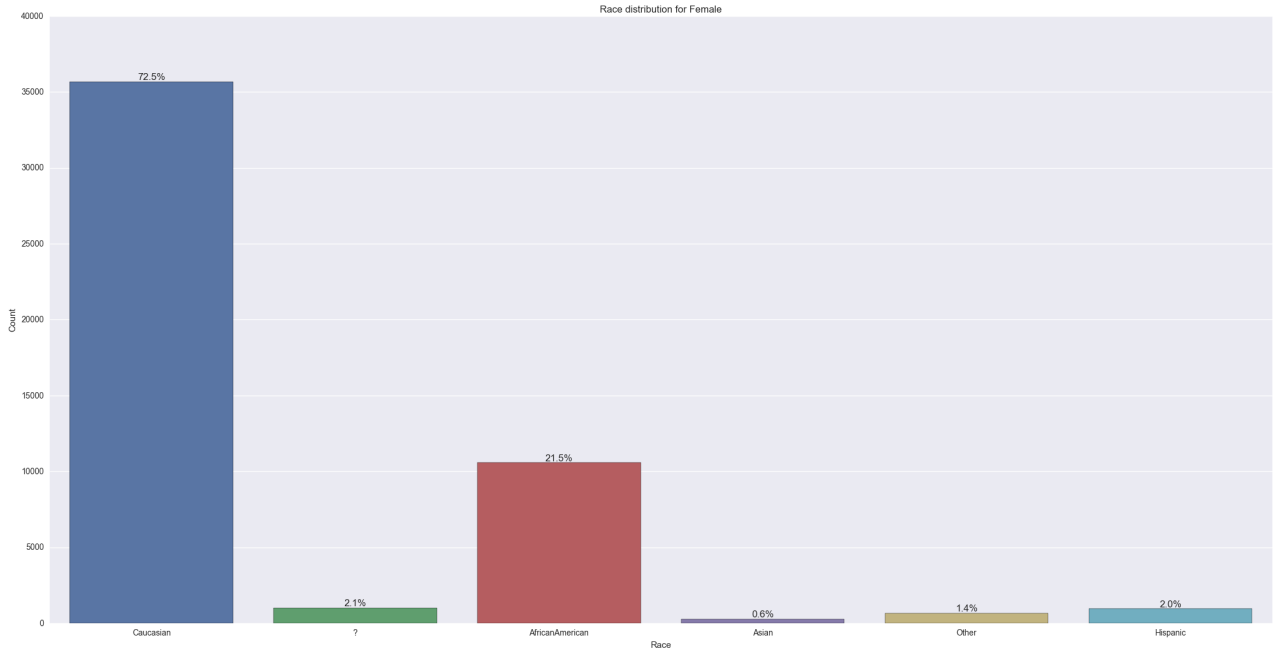


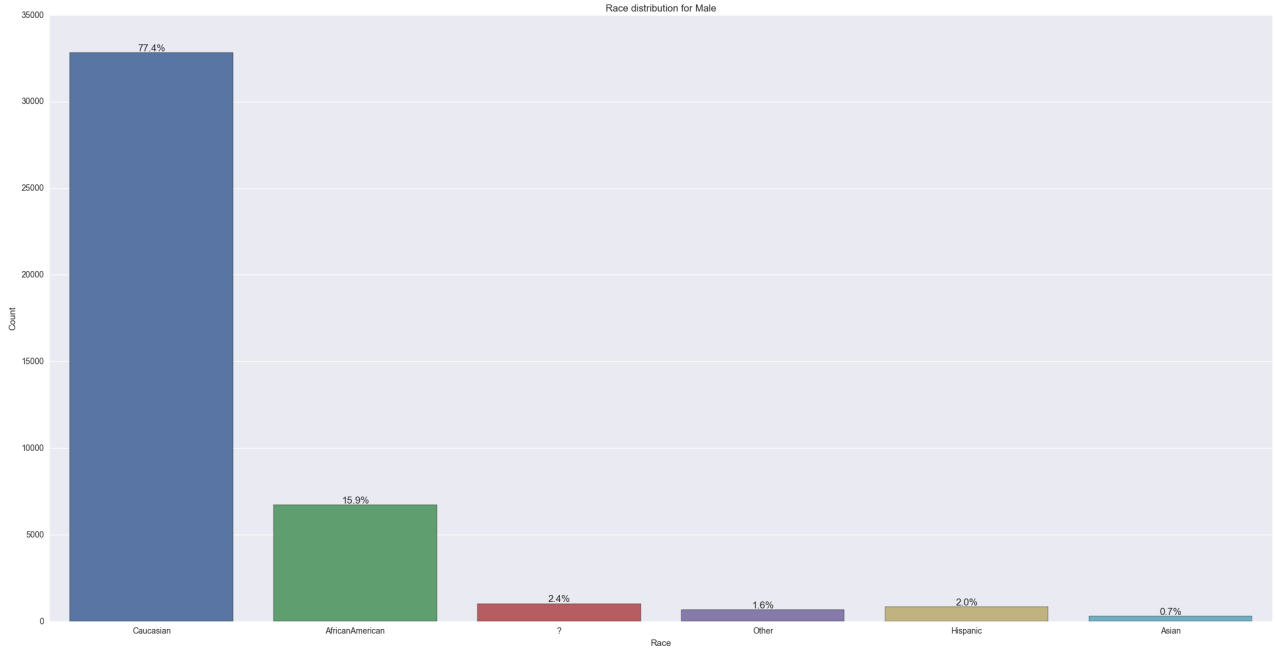




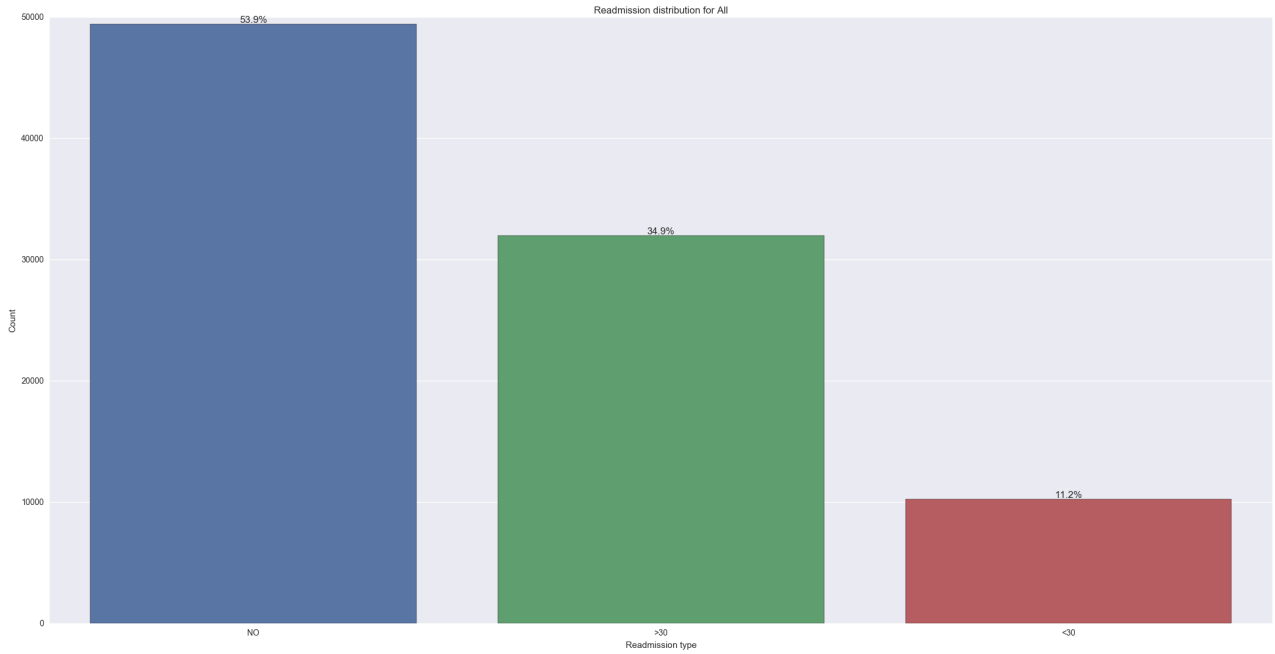
Race distribution

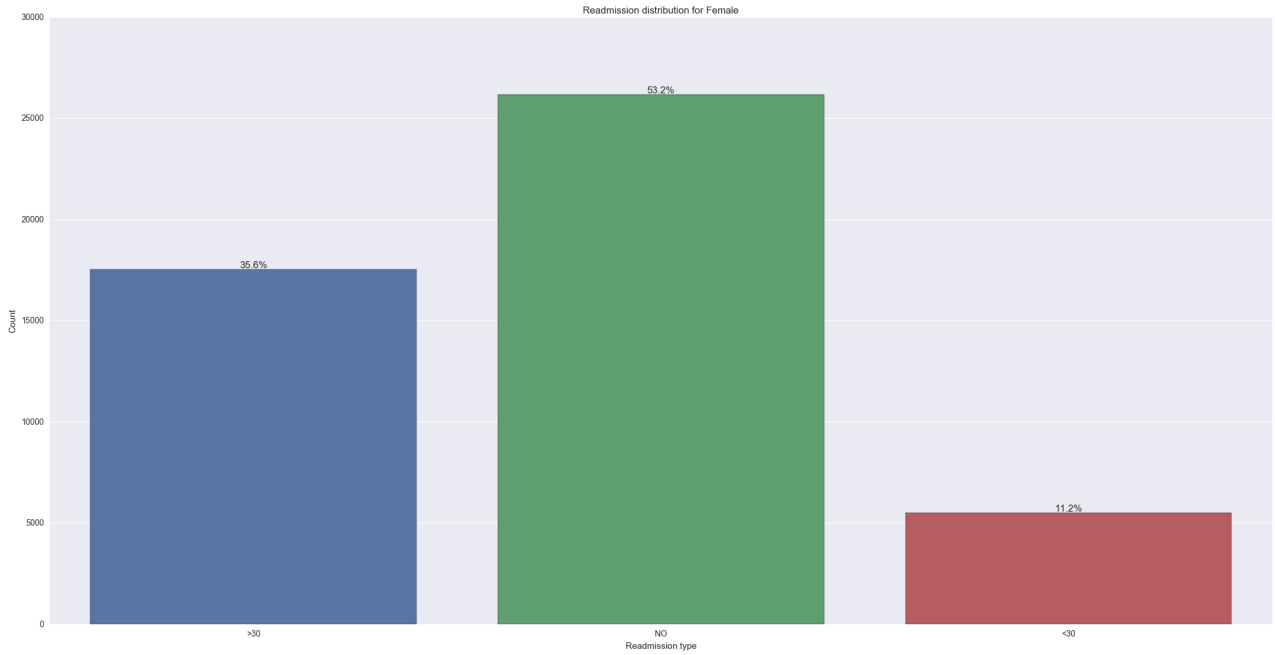


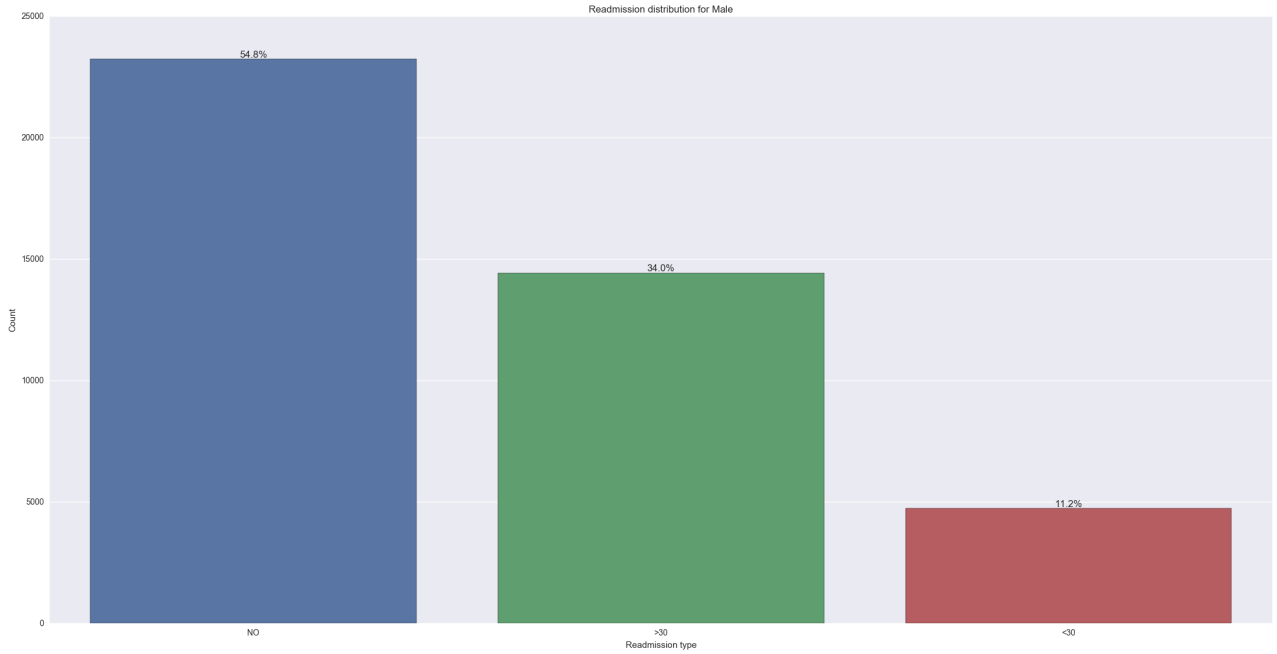

Race distribution for Female
Count
Race
Caucasian
?
AfricanAmerican
Asian
Other
Hispanic
72.5%
2.1%
21.5%
0.6%
1.4%
2.0%


Race distribution for Male
Count
Race
Caucasian
AfricanAmerican
?
Other
Hispanic
Asian
77.4%
15.9%
2.4%
1.6%
2.0%
0.7%

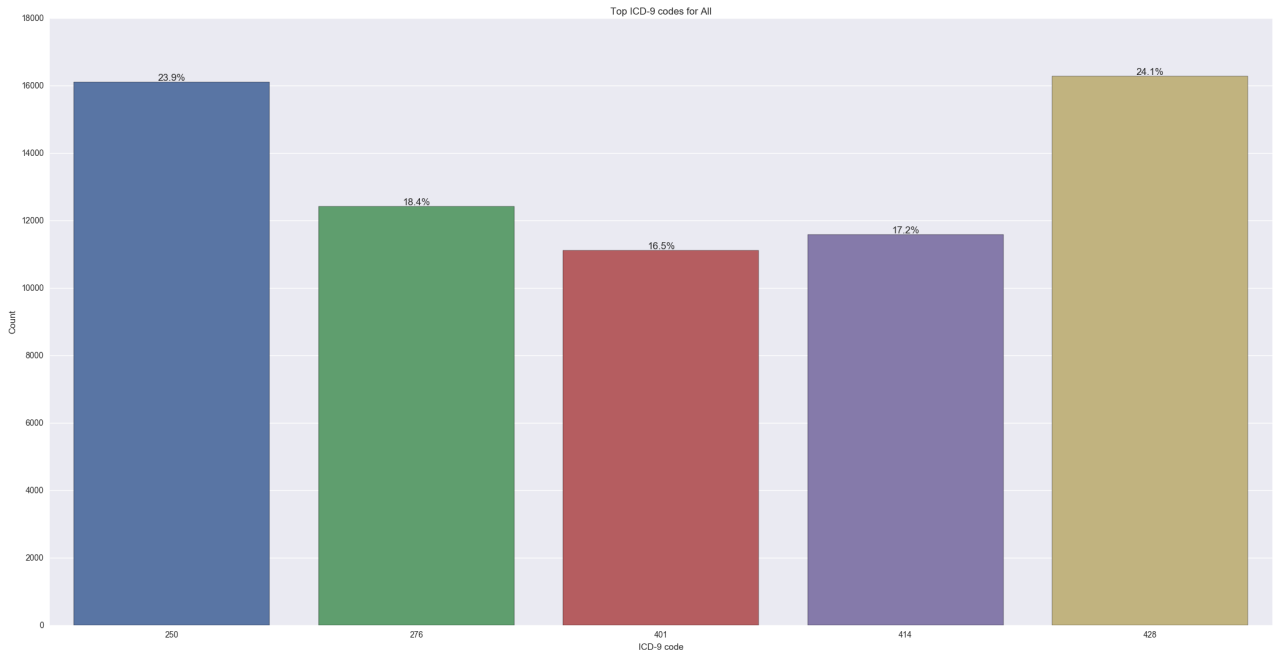
Readmitted distribution

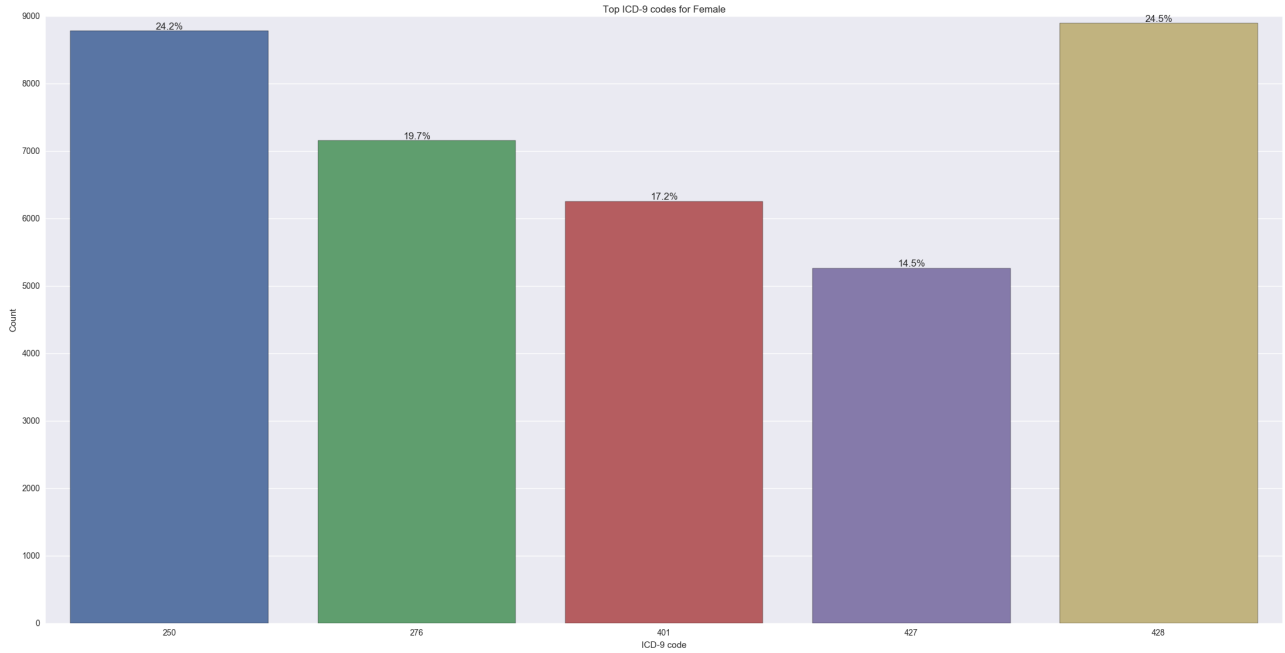


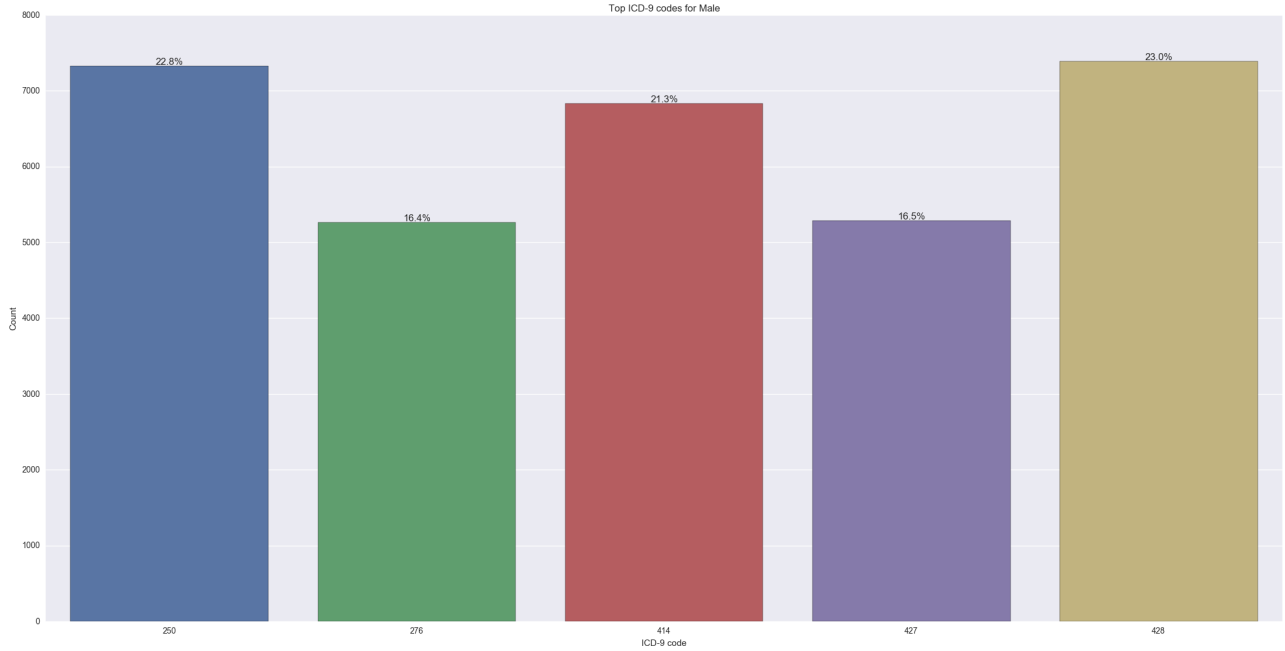




Most frequently used ICD-9 codes distribution

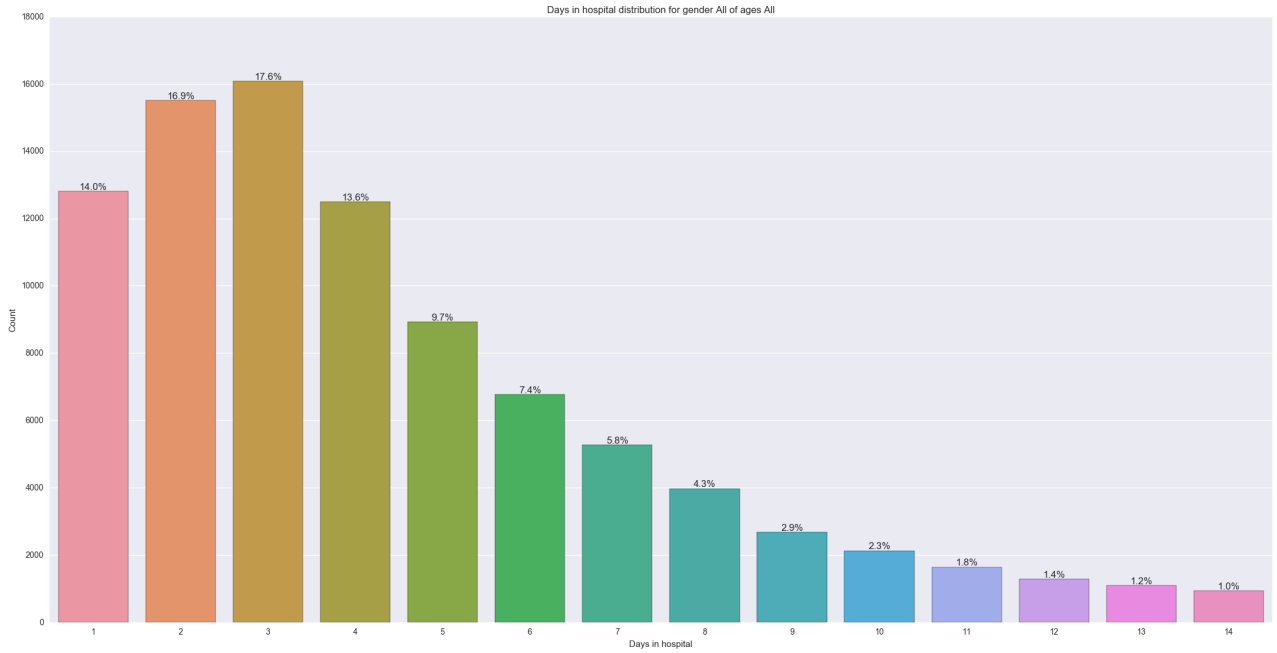


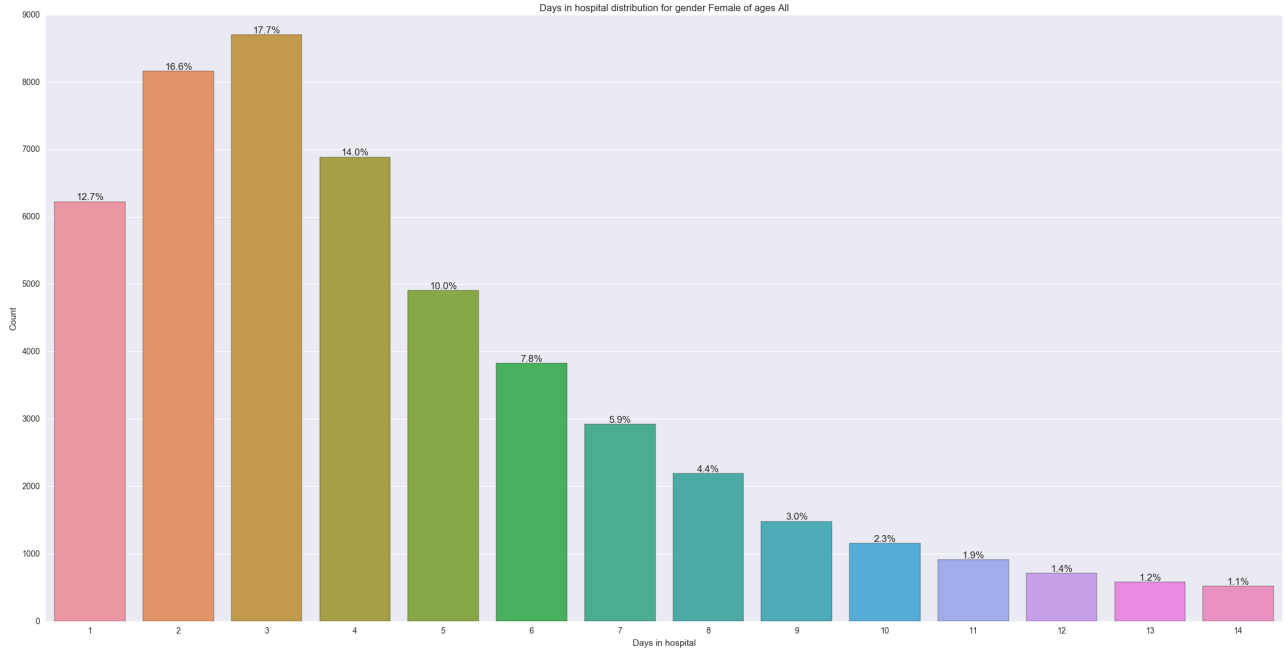

Top ICD-9 codes for Female
Count
ICD-9 code
24.2%
19.7%
17.2%
14.5%
24.5%
250
276
401
427
428

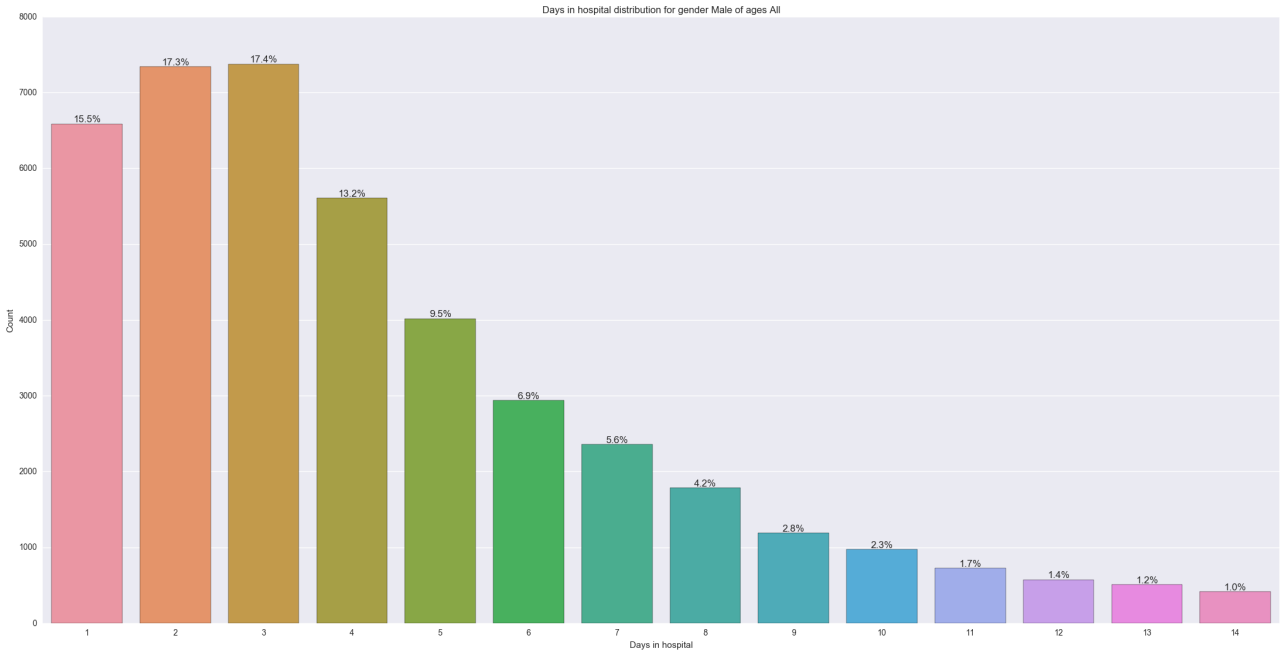

Top ICD-9 codes for Male
Count
ICD-9 code
22.8%
16.4%
21.3%
16.5%
23.0%
250
276
414
427
428

Subpart 2

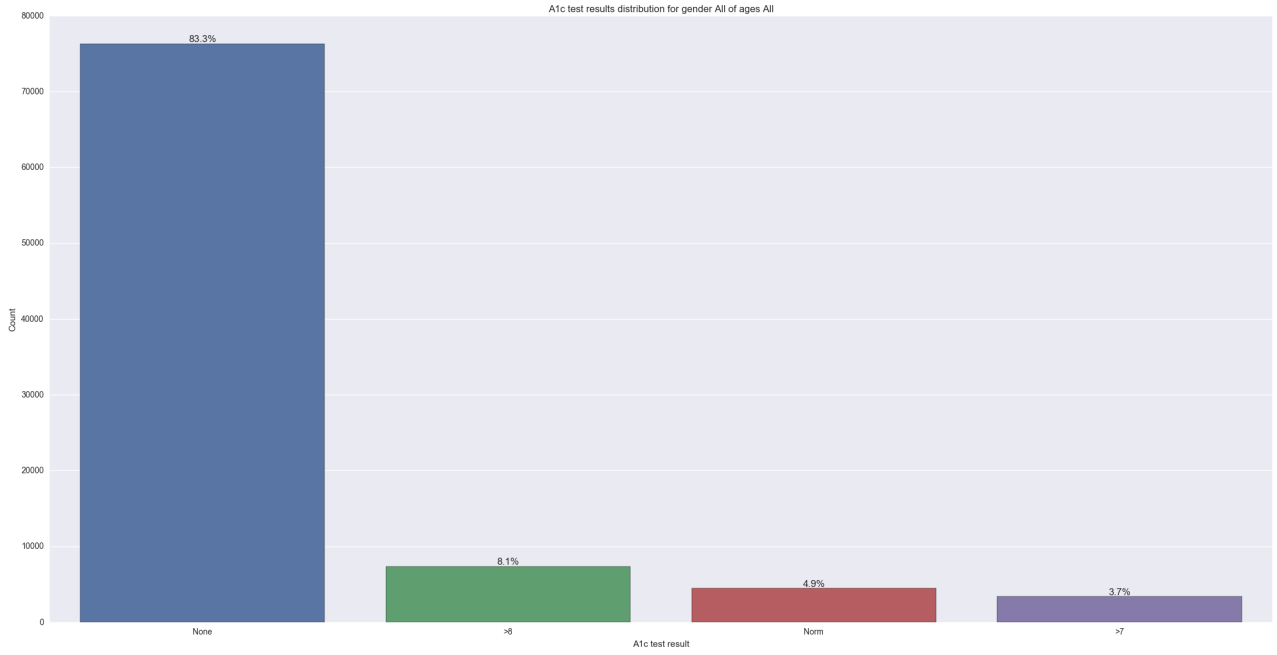
Days in hospital per gender

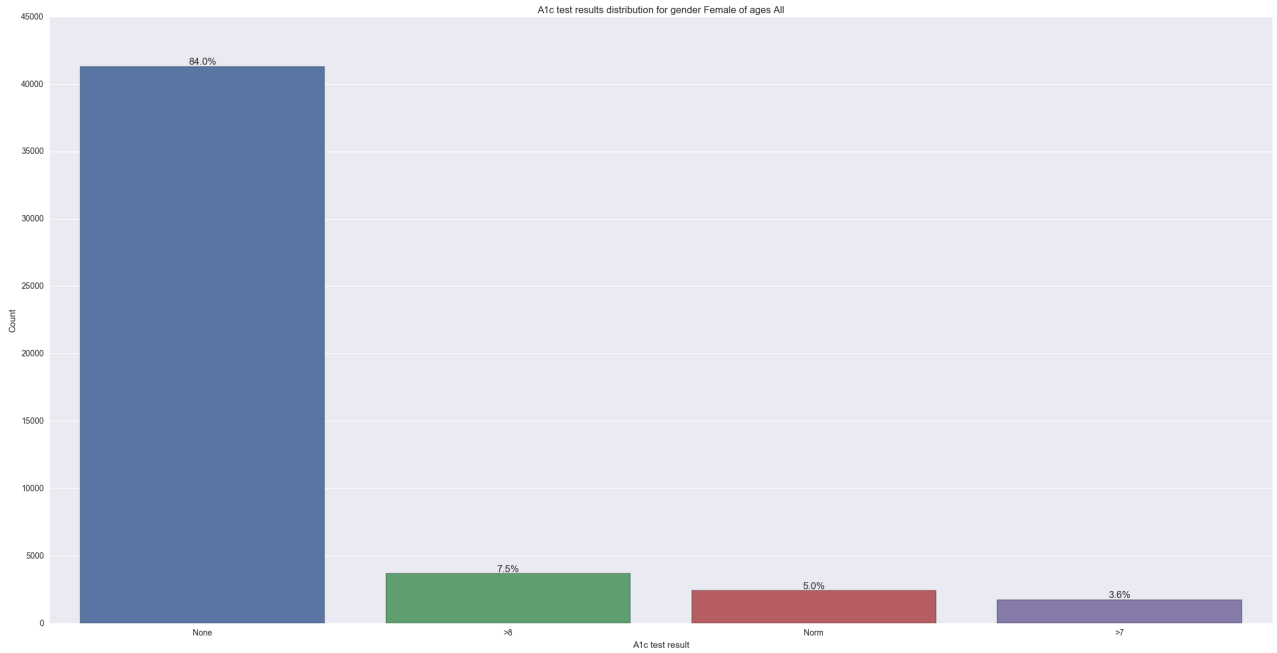


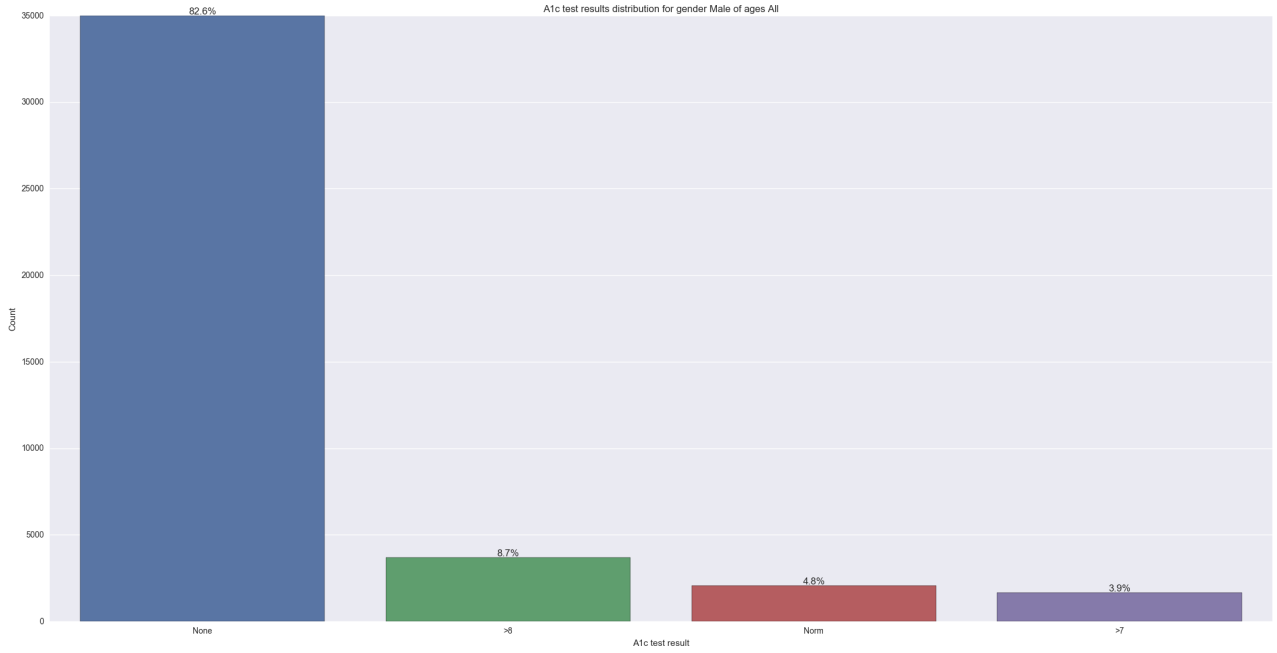




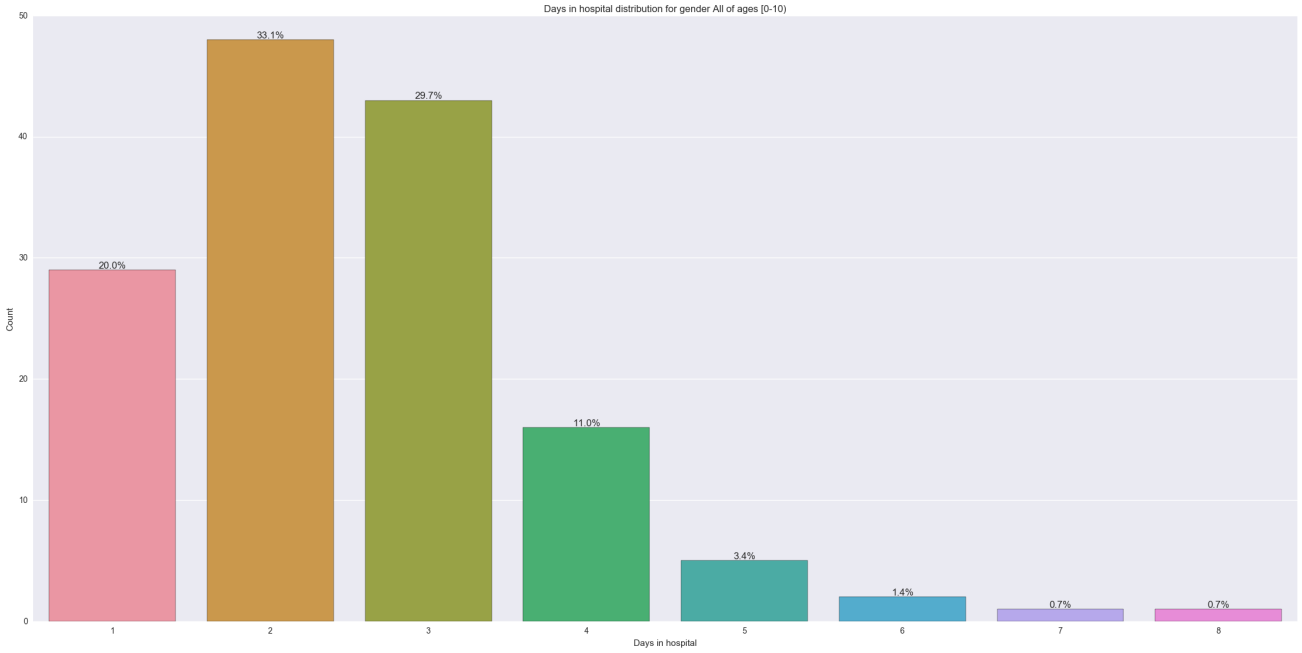
A1C codes per gender

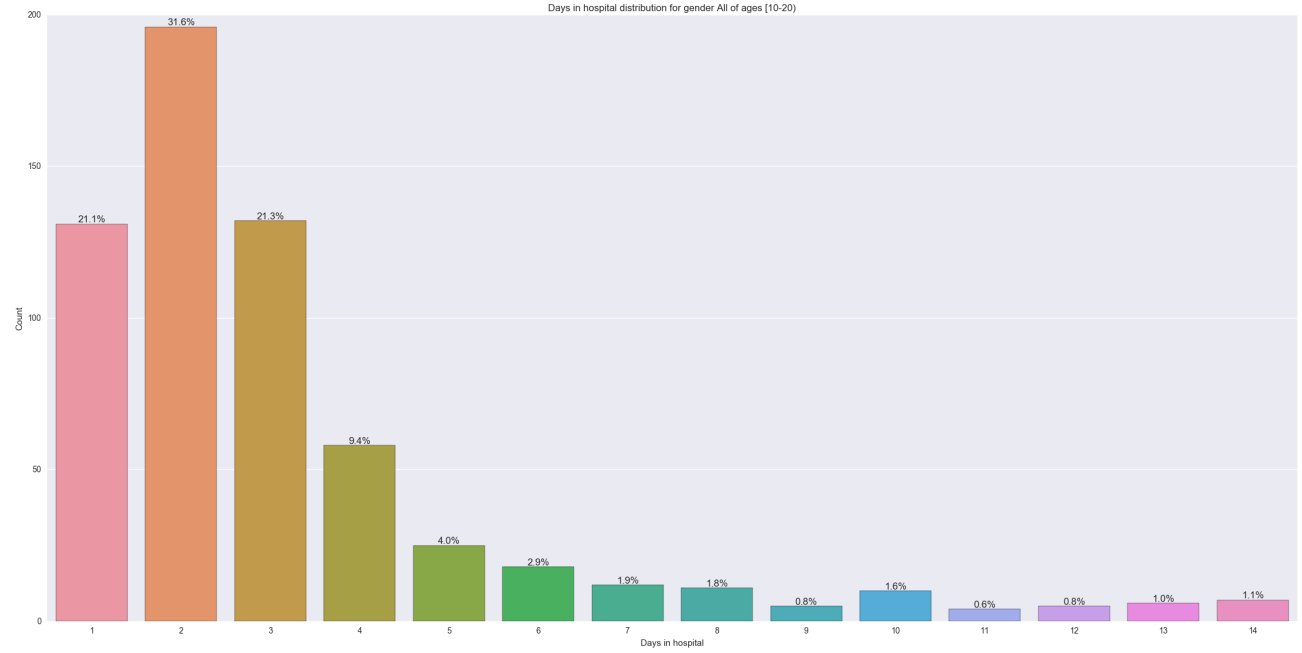


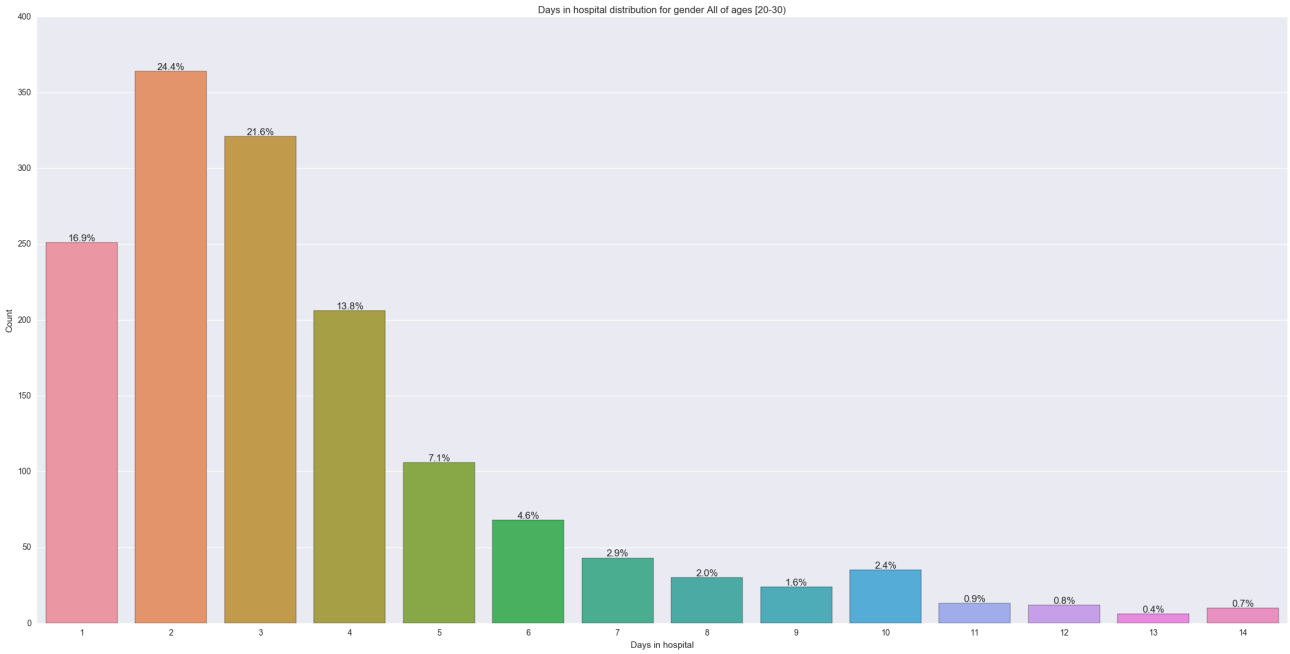

A1c test results distribution for gender Female of ages All
Count
A1c test result
None
>8
Norm
>7
84.0%
7.5%
5.0%
3.6%
0
5000
10000
15000
20000
25000
30000
35000
40000
45000

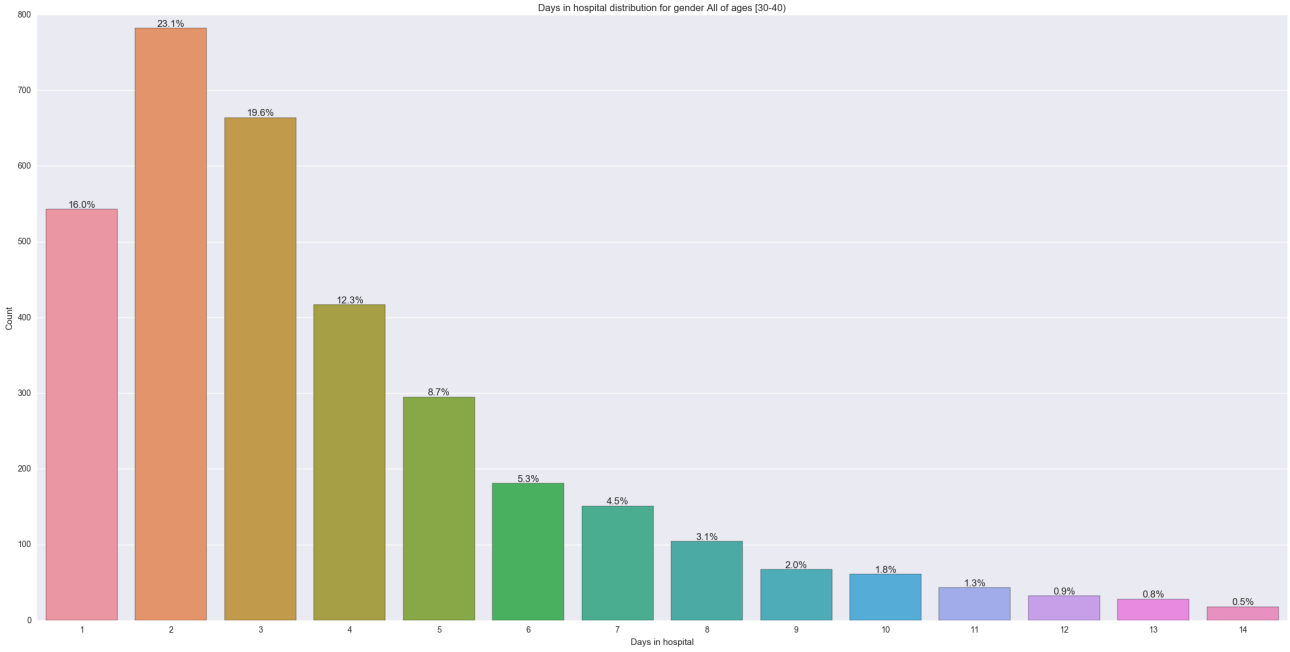

A1c test results distribution for gender Male of ages All
Count
A1c test result
None
>8
Norm
>7
82.6%
8.7%
4.8%
3.9%
0
5000
10000
15000
20000
25000
30000
35000

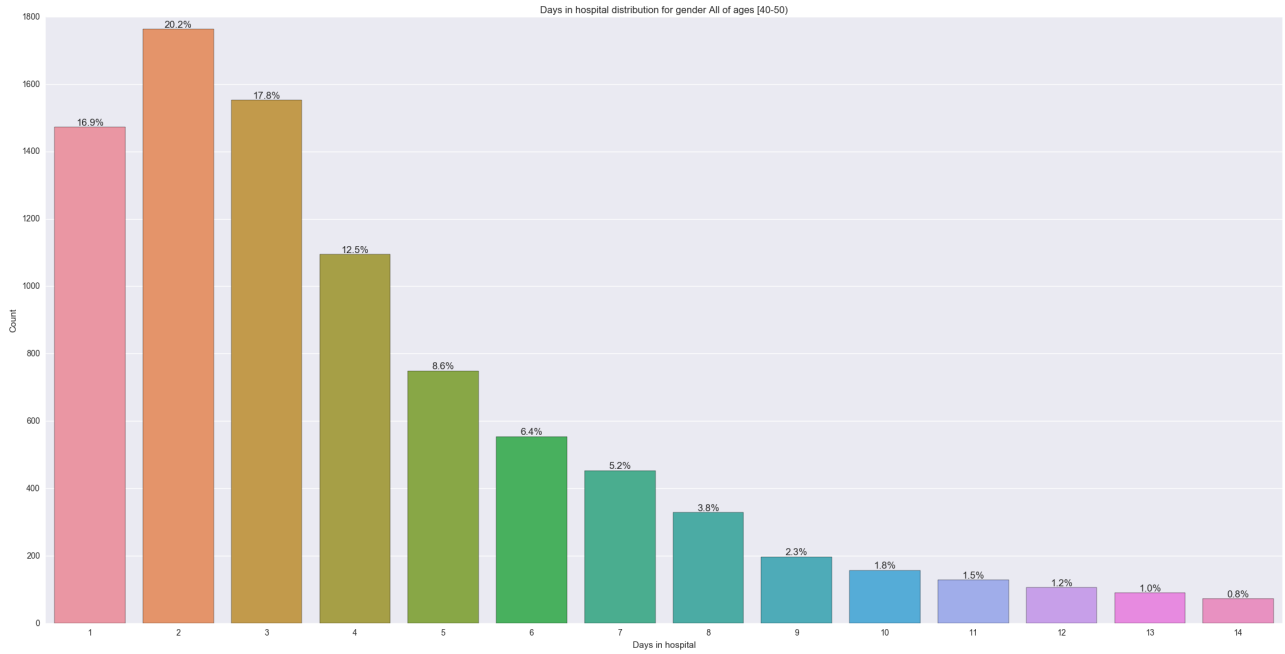
Days in hospital per age group

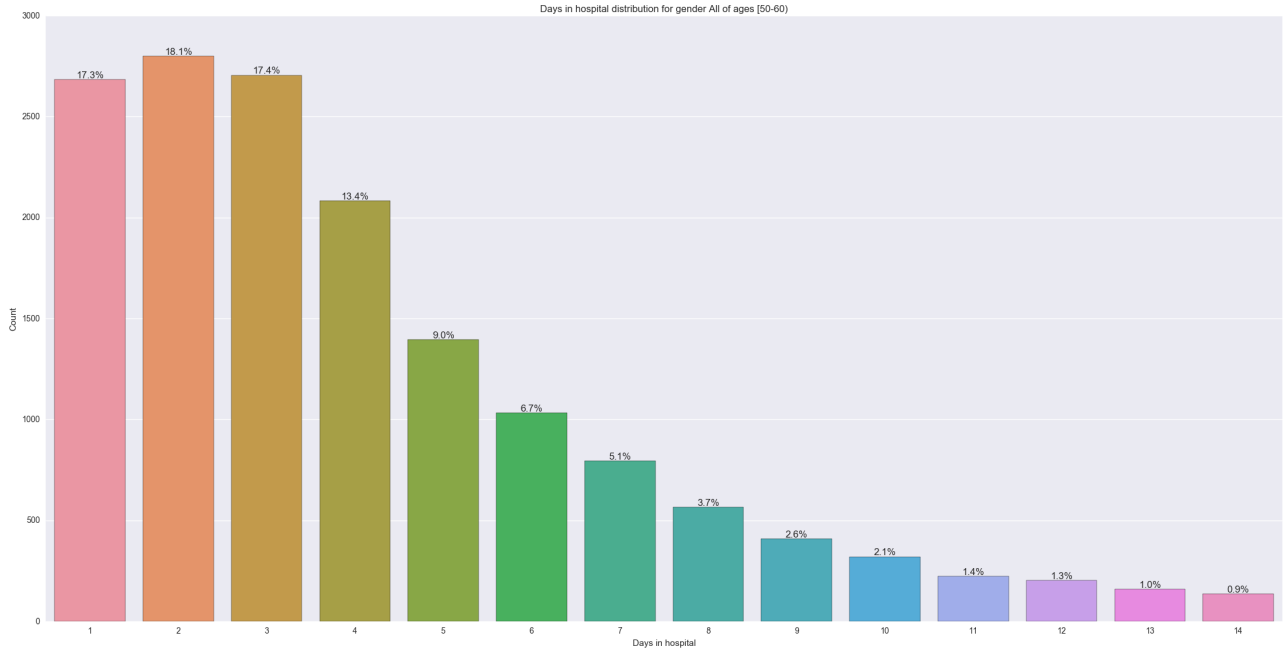


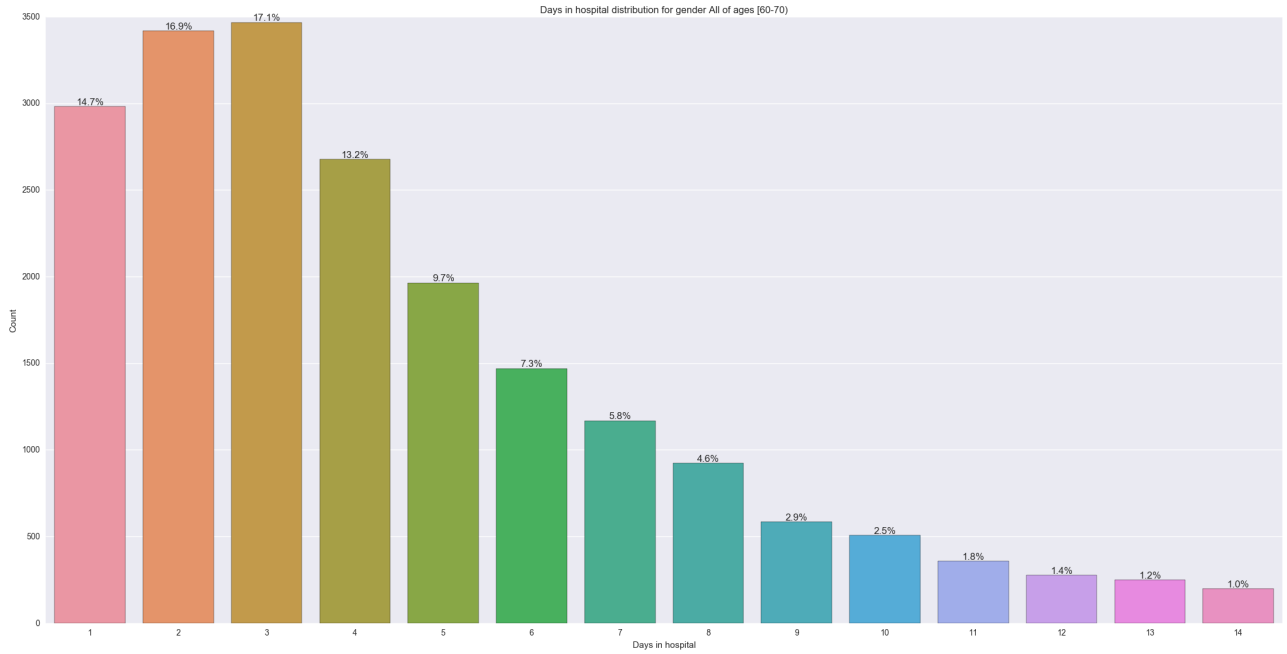


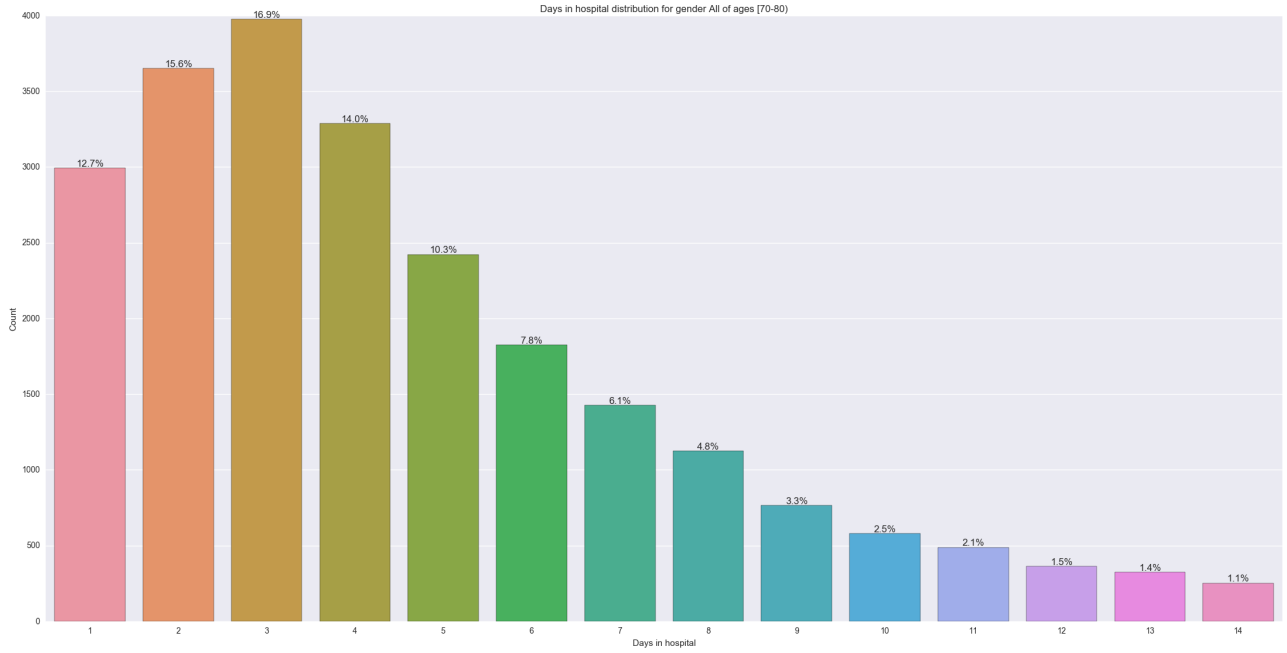

Days in hospital distribution for gender All of ages [20-30)
Count
Days in hospital
16.9%
24.4%
21.6%
13.8%
7.1%
4.6%
2.9%
2.0%
1.6%
2.4%
0.9%
0.8%
0.4%
0.7%

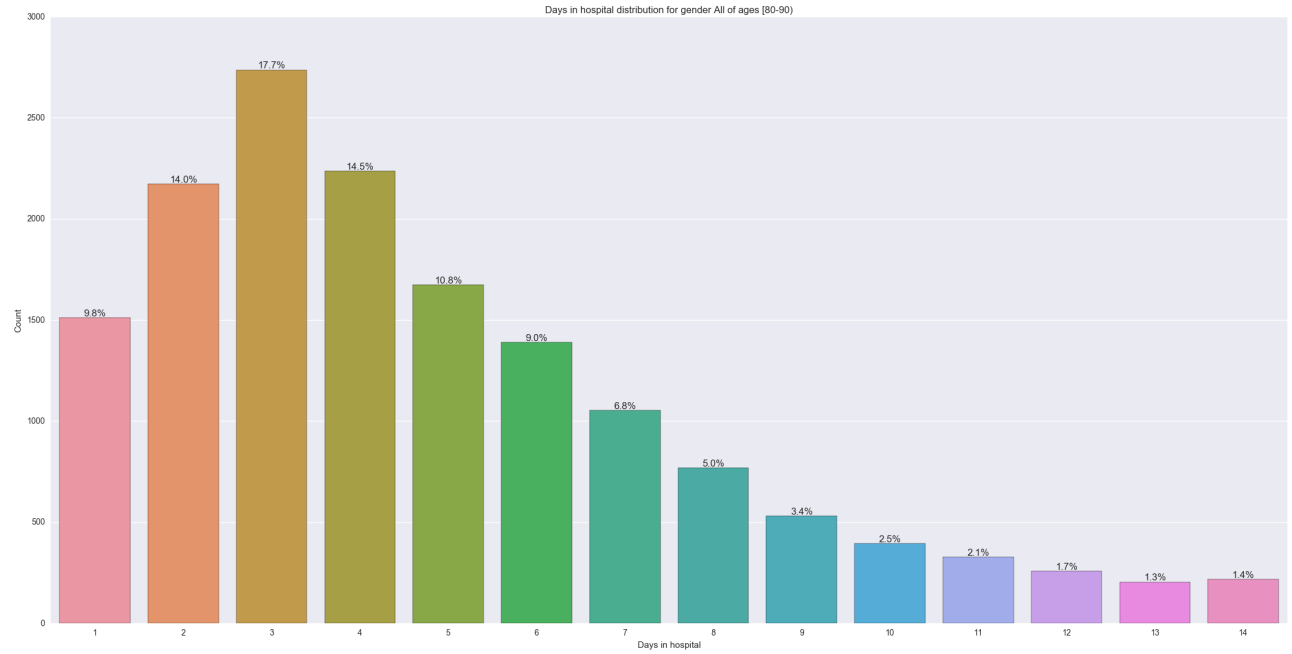

Days in hospital distribution for gender All of ages [30-40)
Count
Days in hospital
16.0%
23.1%
19.6%
12.3%
8.7%
5.3%
4.5%
3.1%
2.0%
1.8%
1.3%
0.9%
0.8%
0.5%

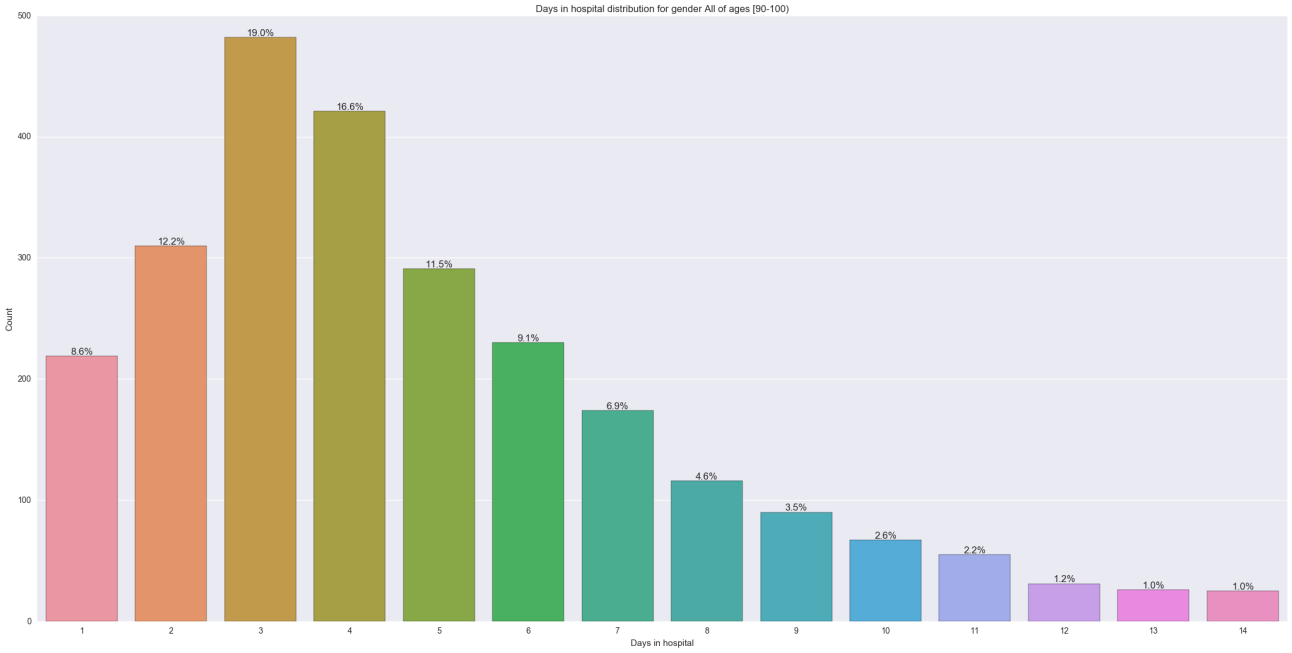

Days in hospital distribution for gender All of ages [40-50)
Count
Days in hospital
16.9%
20.2%
17.8%
12.5%
8.6%
6.4%
5.2%
3.8%
2.3%
1.8%
1.5%
1.2%
1.0%
0.8%


Days in hospital distribution for gender All of ages [50-60)
Count
Days in hospital
17.3%
18.1%
17.4%
13.4%
9.0%
6.7%
5.1%
3.7%
2.6%
2.1%
1.4%
1.3%
1.0%
0.9%

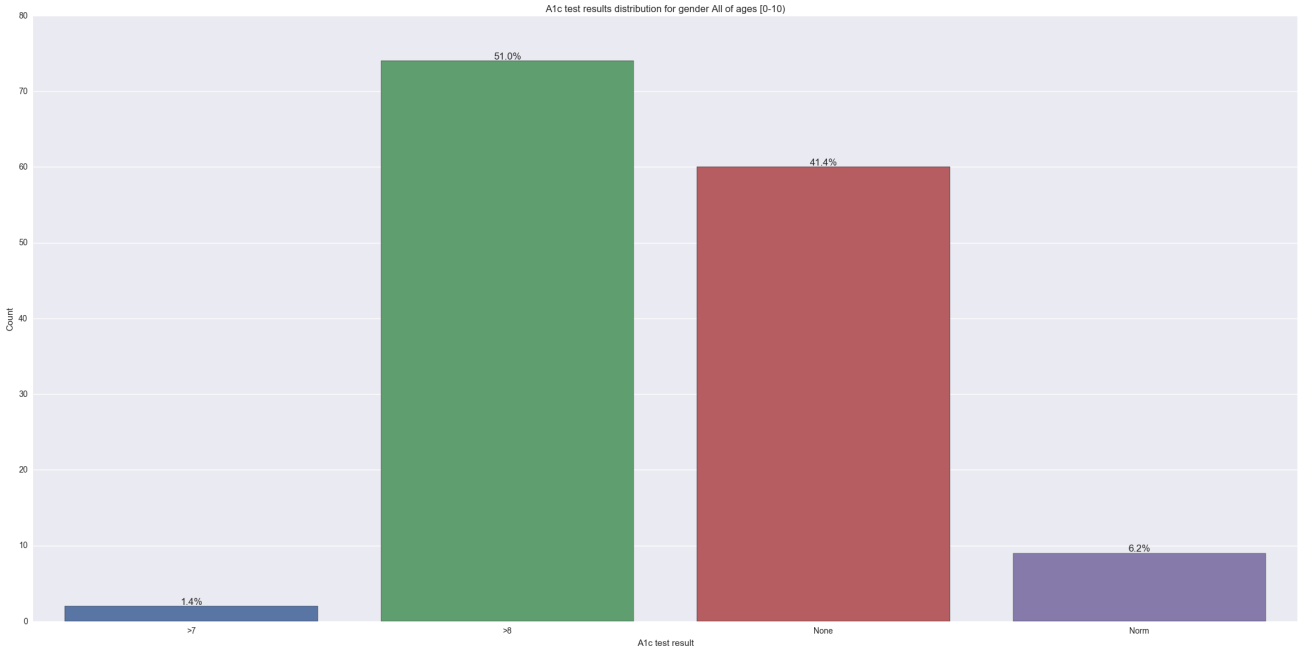

Days in hospital distribution for gender All of ages [60-70)
Count
Days in hospital
14.7%
16.9%
17.1%
13.2%
9.7%
7.3%
5.8%
4.6%
2.9%
2.5%
1.8%
1.4%
1.2%
1.0%

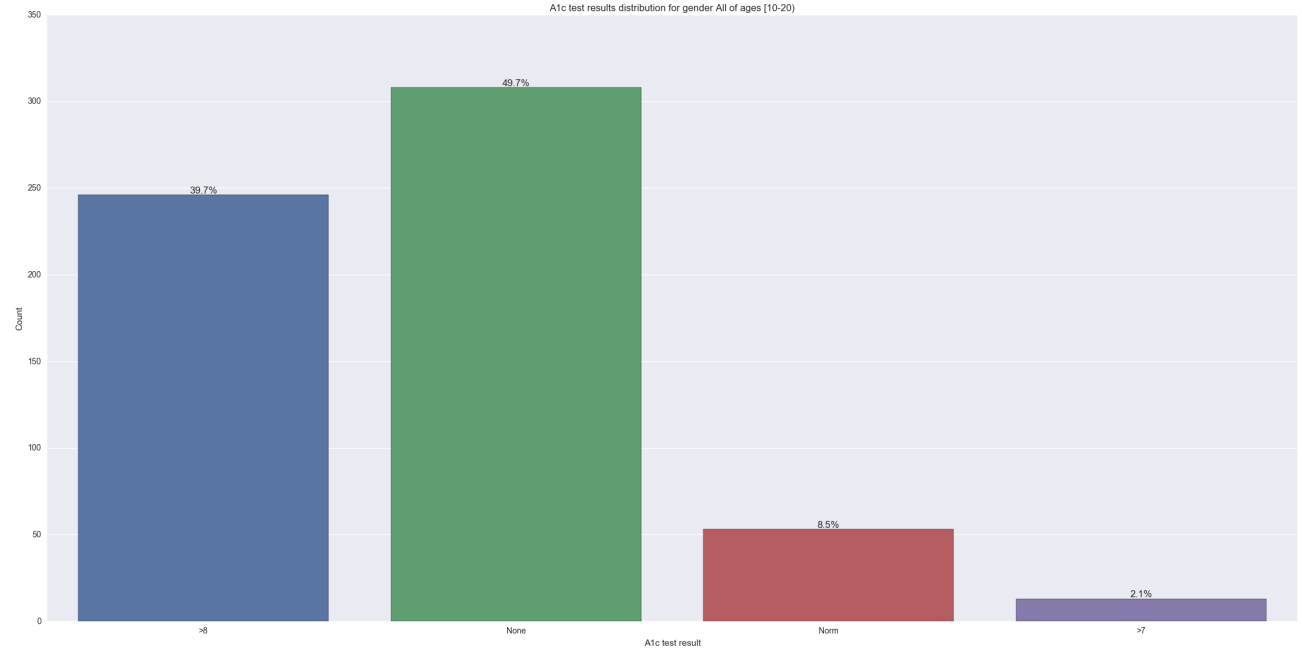

Days in hospital distribution for gender All of ages [70-80)
Count
Days in hospital
12.7%
15.6%
16.9%
14.0%
10.3%
7.8%
6.1%
4.8%
3.3%
2.5%
2.1%
1.5%
1.4%
1.1%

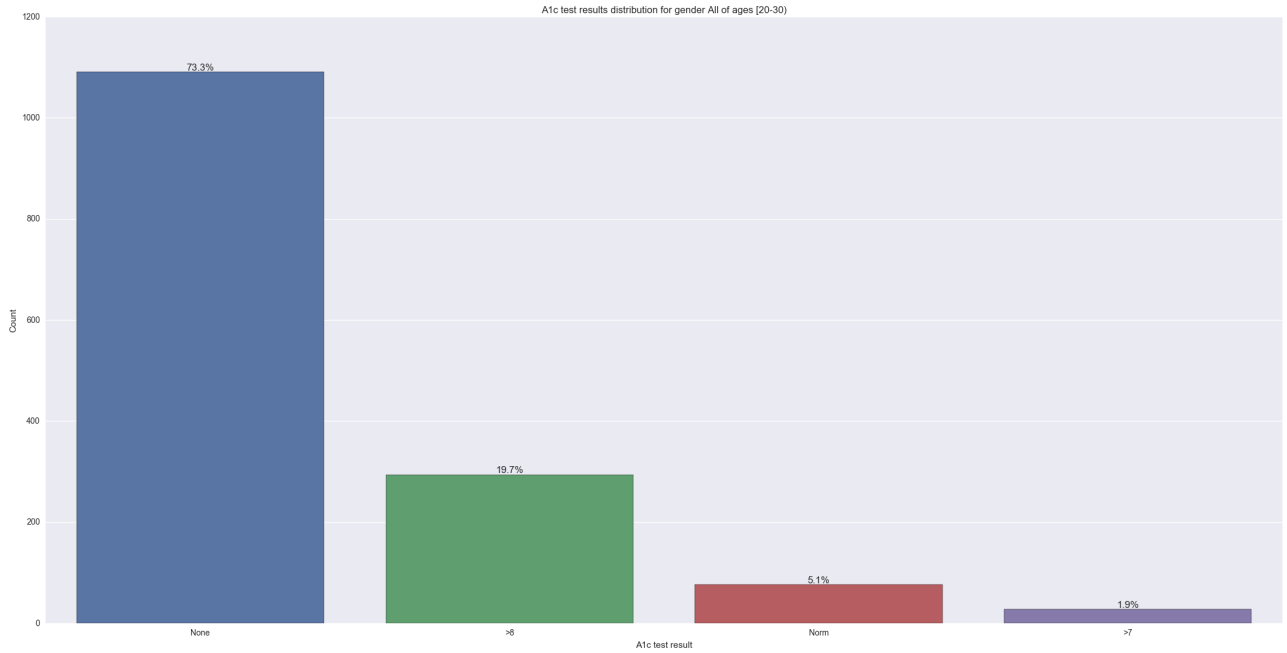

Days in hospital distribution for gender All of ages [80-90)
Count
Days in hospital
9.8%
14.0%
17.7%
14.5%
10.8%
9.0%
6.8%
5.0%
3.4%
2.5%
2.1%
1.7%
1.3%
1.4%

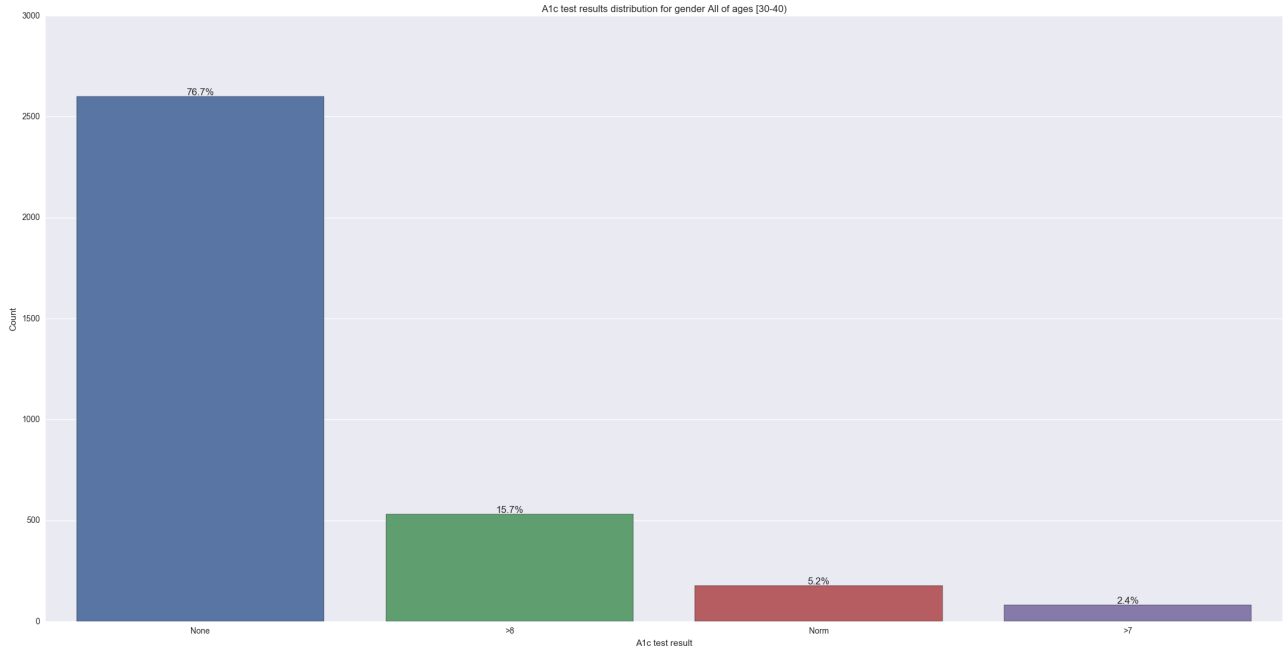

Days in hospital distribution for gender All of ages [90-100)
Count
Days in hospital
8.6%
12.2%
19.0%
16.6%
11.5%
9.1%
6.9%
4.6%
3.5%
2.6%
2.2%
1.2%
1.0%
1.0%

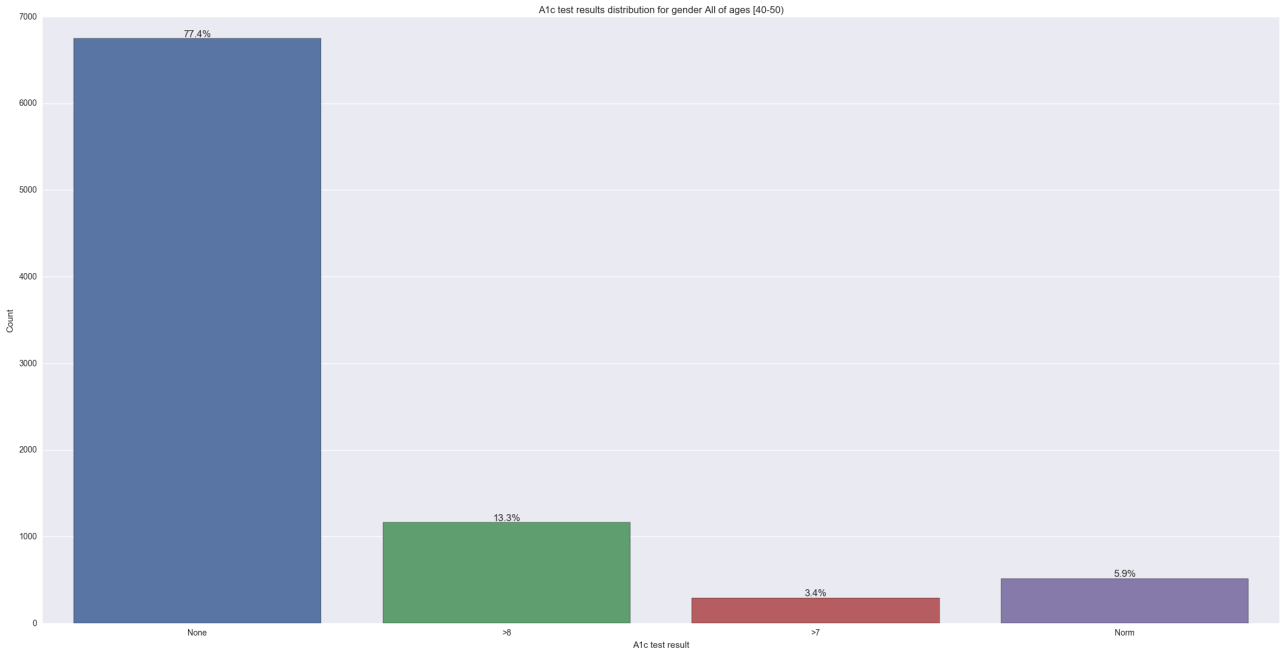
A1C codes per age group

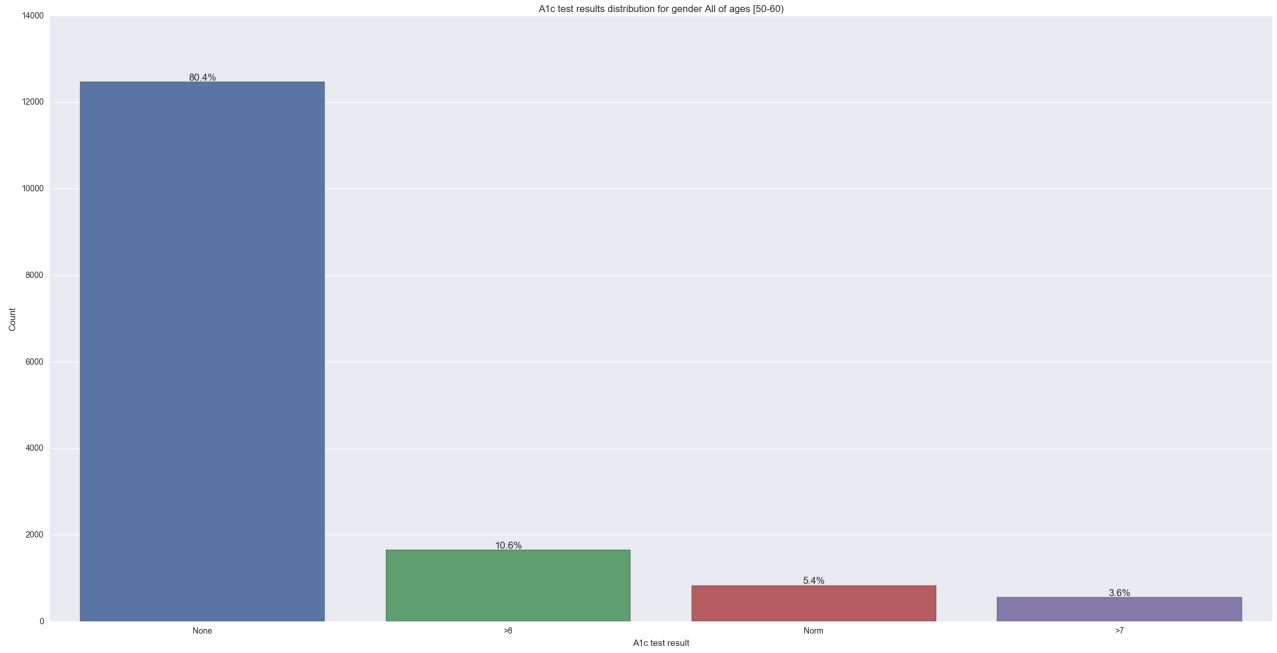


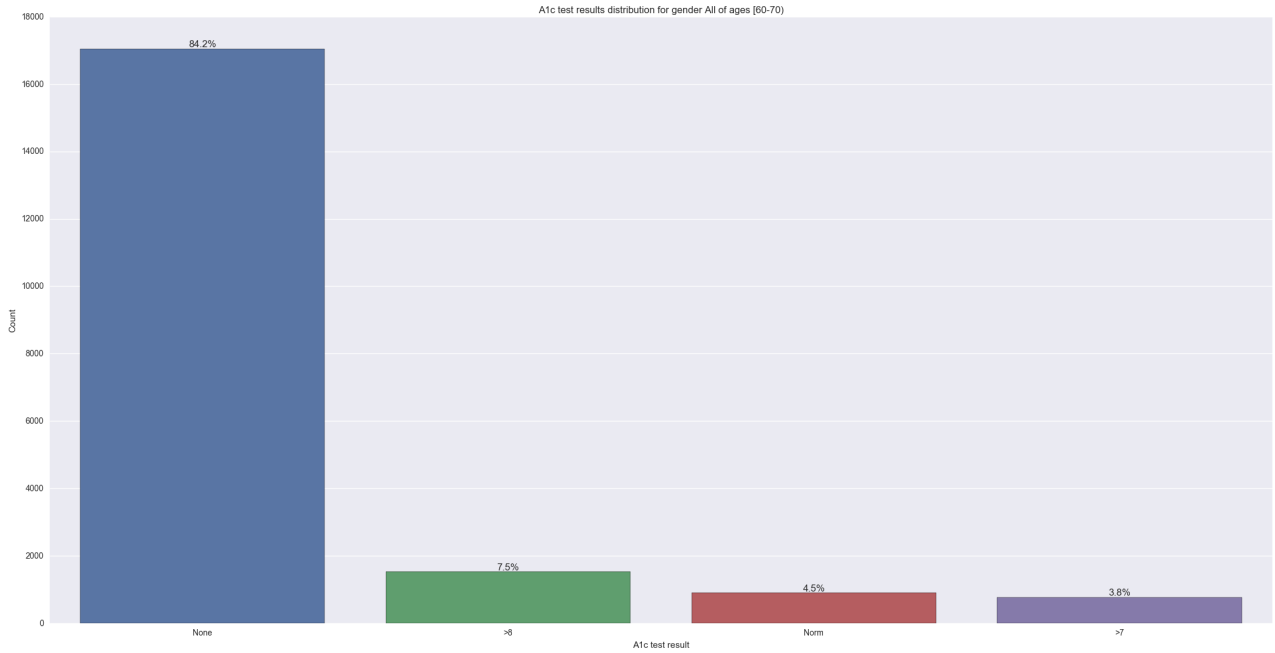


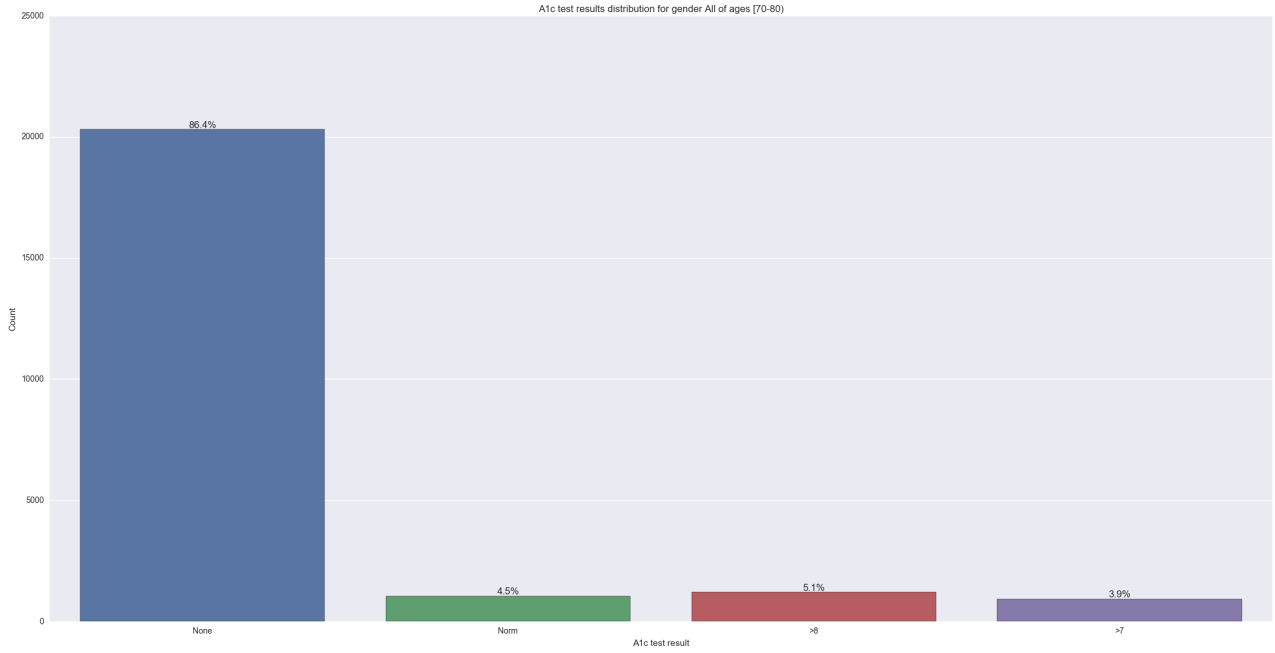

A1c test results distribution for gender All of ages [20-30)
1200
1000
800
600
400
200
0
Count
73.3%
19.7%
5.1%
1.9%
None
>8
Norm
>7
A1c test result

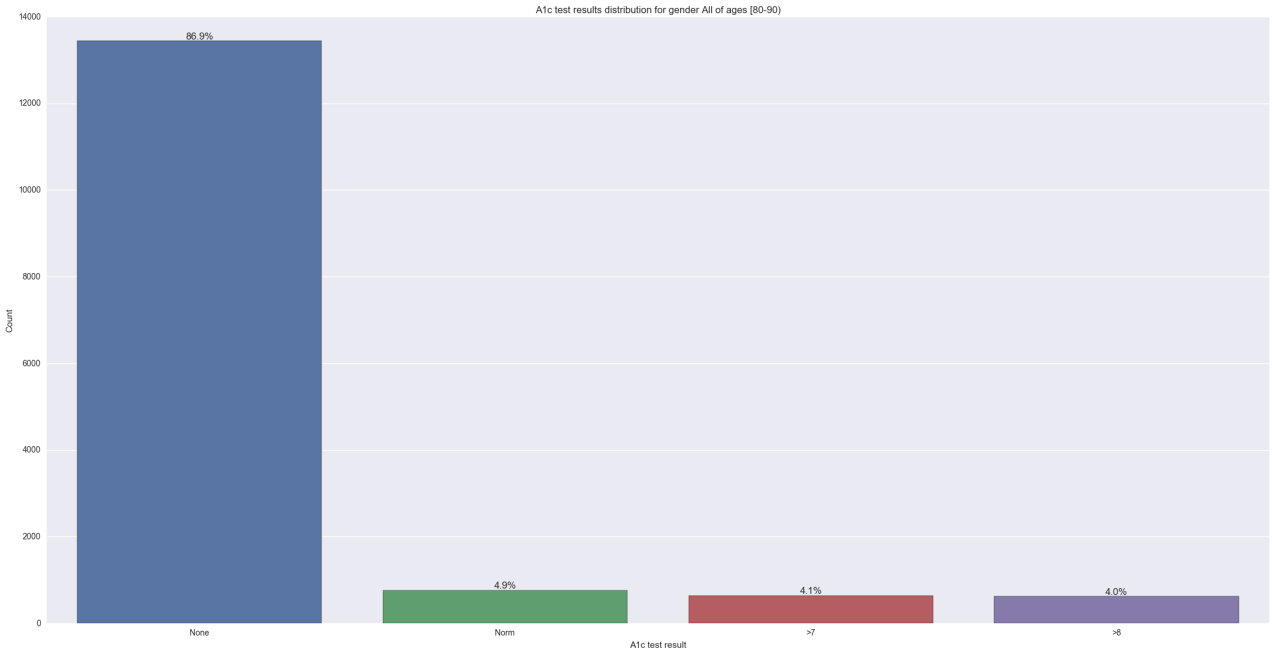

A1c test results distribution for gender All of ages [30-40)
3000
2500
2000
1500
1000
500
0
Count
76.7%
15.7%
5.2%
2.4%
None
>8
Norm
>7
A1c test result


A1c test results distribution for gender All of ages [40-50)
Count
A1c test result
None
>8
>7
Norm
77.4%
13.3%
3.4%
5.9%
0
1000
2000
3000
4000
5000
6000
7000


A1c test results distribution for gender All of ages [50-60)
Count
A1c test result
None
>8
Norm
>7
80.4%
10.6%
5.4%
3.6%
0
2000
4000
6000
8000
10000
12000
14000


A1c test results distribution for gender All of ages [60-70)
Count
A1c test result
None
>8
Norm
>7
84.2%
7.5%
4.5%
3.8%
0
2000
4000
6000
8000
10000
12000
14000
16000
18000


A1c test results distribution for gender All of ages [70-80)
Count
A1c test result
None
Norm
>8
>7
86.4%
4.5%
5.1%
3.9%
0
5000
10000
15000
20000
25000


A1c test results distribution for gender All of ages [80-90)
14000
12000
10000
8000
6000
4000
2000
0
Count
86.9%
4.9%
4.1%
4.0%
None
Norm
>7
>8
A1c test result