1. a. Data.

We get the diabetes dataset from UCI machine learning repository. Data can be obtained from the following link.

https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008#

Citation:

Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore, "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records," BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014.

b. Variables.

encounter_id = Unique case id
patient_nbr = Unique patient id
race = Categorical variable describing race.
gender = Categorical variable describing gender.
race = Categorical variable describing race.
age = Quantitative variable giving age.
time_in_hospital = Time spent in the hospital.
num_lab_procedures = Number of lab procedures.
num_procedures = Number of other procedures.
num_outpatient = Number of outpatient visits.
num_emergency = Number of emergency visits.
num_inpatient = Number of inpatient visits.
diag_1, diag_2, diag_3 – Numerical representations of patient diagnosis.

c. Sample size.

101766 observations of 50 variables.

d. Missing values.

Yes, there are missing values. They will be cleaned up before consuming the data.

2. Question to be answered.

Do exploratory data analysis figure out the relationship between the variables in the data and to reveal these relationships.

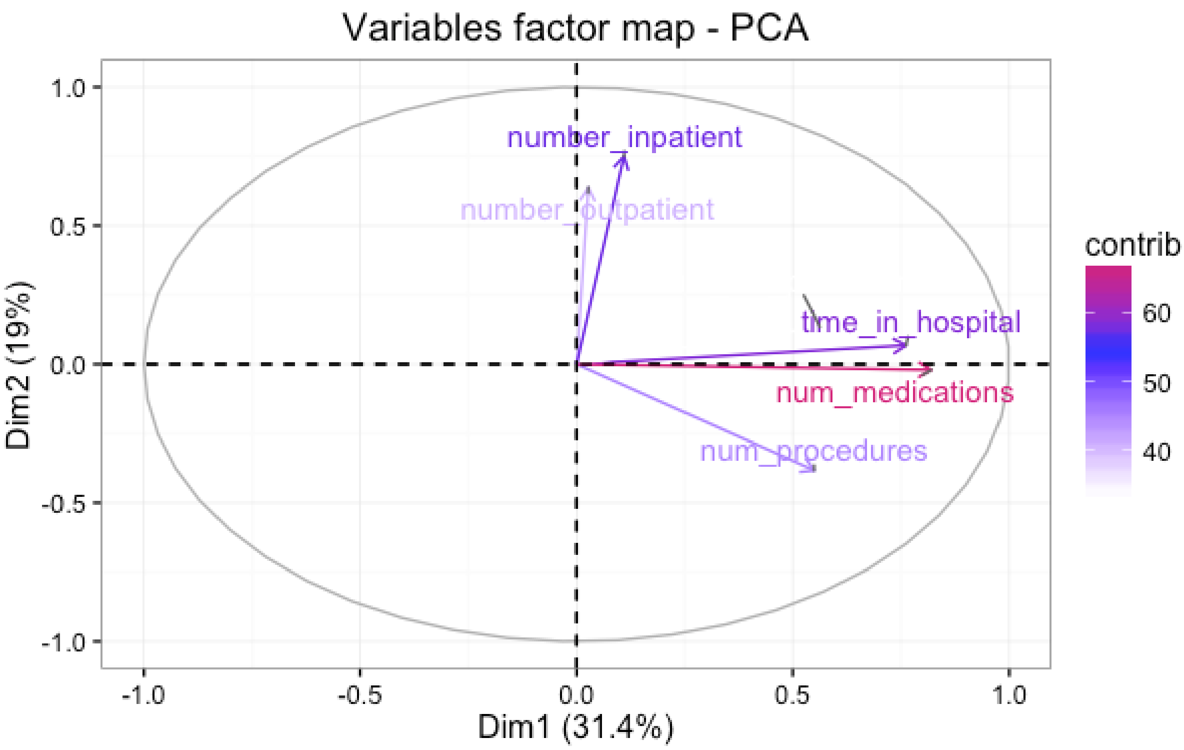
3. Explain technique.

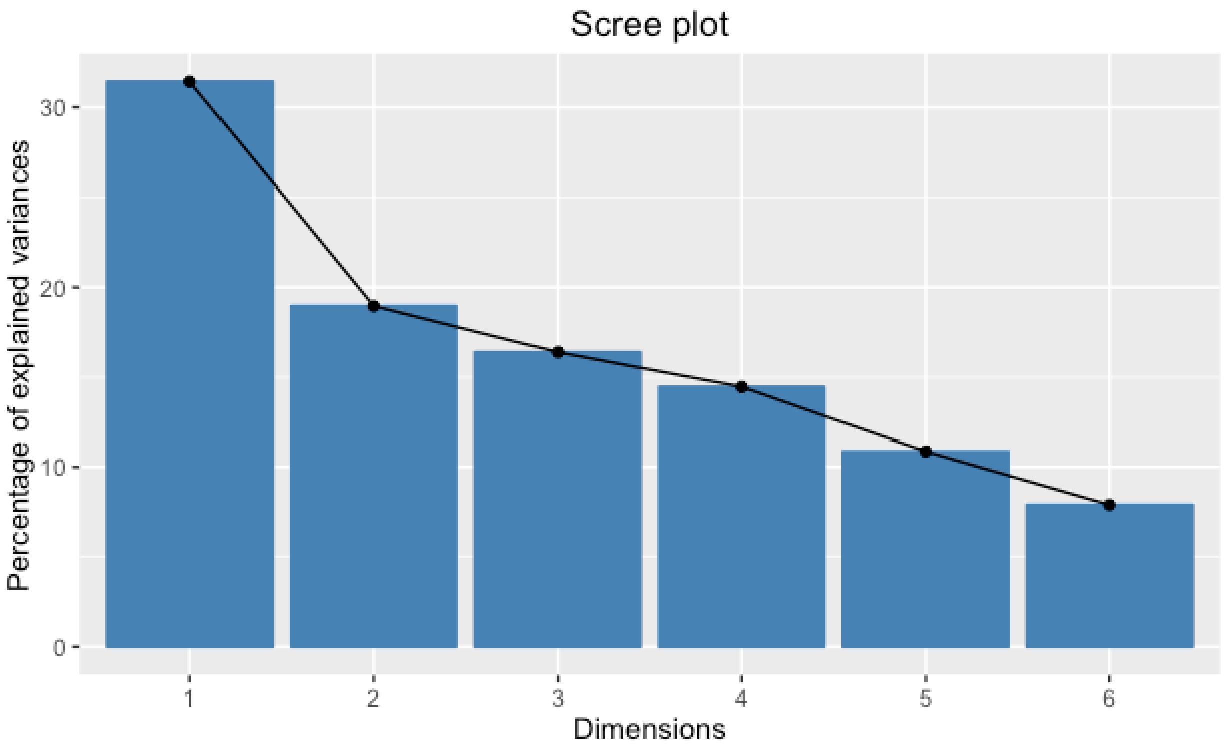
We will be running principal component analysis to reduce the dimensions of our dataset, and to find patterns in dataset. We will then run k-means clustering to analyze for clusters in the data set.

Principal Component Analysis.

We run Principal Component Analysis to reduce the dimensionality of the dataset.

```
> pca$var$contrib
                        Dim.1       Dim.2      Dim.3       Dim.4        Dim.5
time_in_hospital   30.7850808  0.41129871  3.7570252  0.08710701 38.780520533
num_lab_procedures 16.8094277  1.60038410 28.9494835 17.20714143 35.310859658
num_procedures     16.0382975 12.90762329 30.0473608  6.32748113 20.221541646
num_medications    35.6871785  0.03811681  3.0573512  0.50761317  2.400809292
number_outpatient   0.0394338 35.68130245 33.3553498 30.23793317  0.004438583
number_inpatient    0.6405817 49.36127464  0.8334295 45.63272409  3.281830288
```





Scree Plot

We should be able to get away with the first 2 principal components.

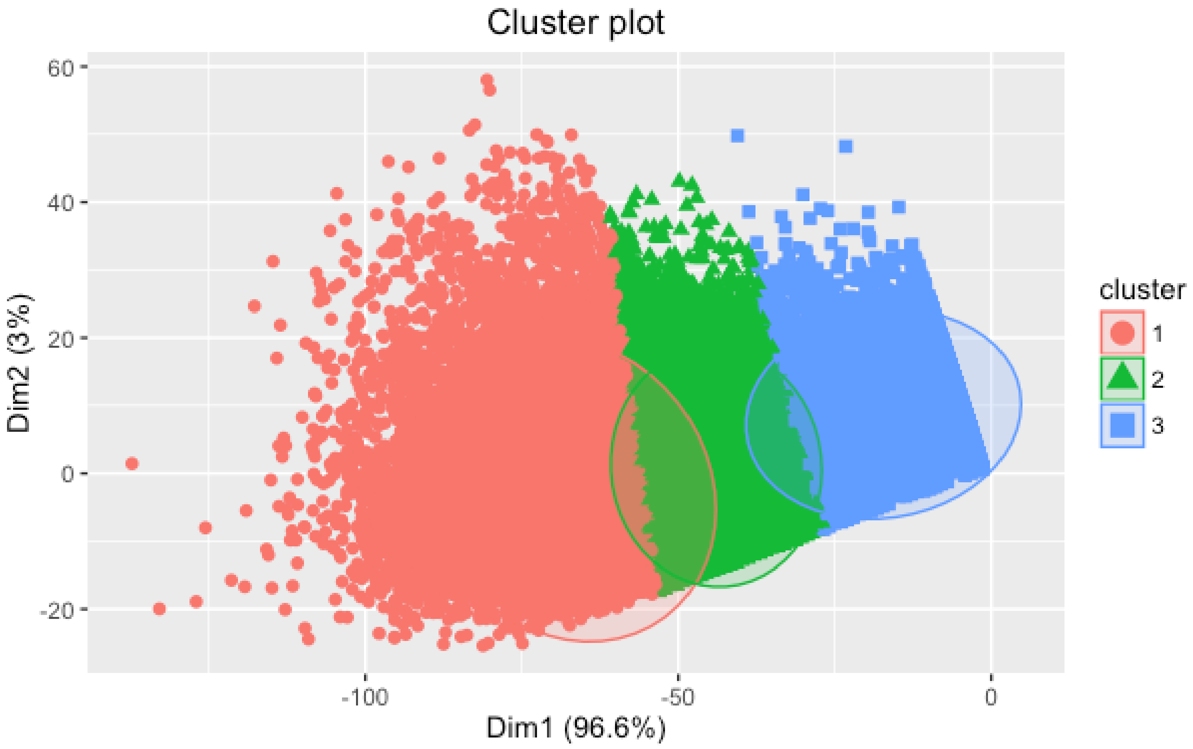
```
> dimdesc(pca)
$Dim.1
$Dim.1$quanti
                   correlation       p.value
num_medications     0.82033603  0.000000e+00
time_in_hospital    0.76191378  0.000000e+00
num_lab_procedures  0.56300486  0.000000e+00
num_procedures      0.54993937  0.000000e+00
number_inpatient    0.10990639 6.358207e-271
number_outpatient   0.02726905  3.302765e-18

$Dim.2
$Dim.2$quanti
                   correlation       p.value
number_inpatient    0.74920119  0.000000e+00
number_outpatient   0.63697974  0.000000e+00
num_lab_procedures  0.13490169  0.000000e+00
time_in_hospital    0.06838864 9.298384e-106
num_medications    -0.02081918  3.092054e-11
num_procedures     -0.38311435  0.000000e+00

$Dim.3
$Dim.3$quanti
                   correlation      p.value
number_outpatient   0.57267934  0.00000e+00
num_procedures      0.54354053  0.00000e+00
num_medications     0.17338097  0.00000e+00
number_inpatient   -0.09052389 4.10215e-184
time_in_hospital   -0.19219887  0.00000e+00
num_lab_procedures -0.53351813  0.00000e+00
```

Clustering (k-means).

Next we run k-means clustering with k=3 (we first tried it with k=5, but the data was not separated properly)



(note: that we could get "Knee plot" to determine the optimal clusters, but my laptop does not have the memory to process this large dataset.)

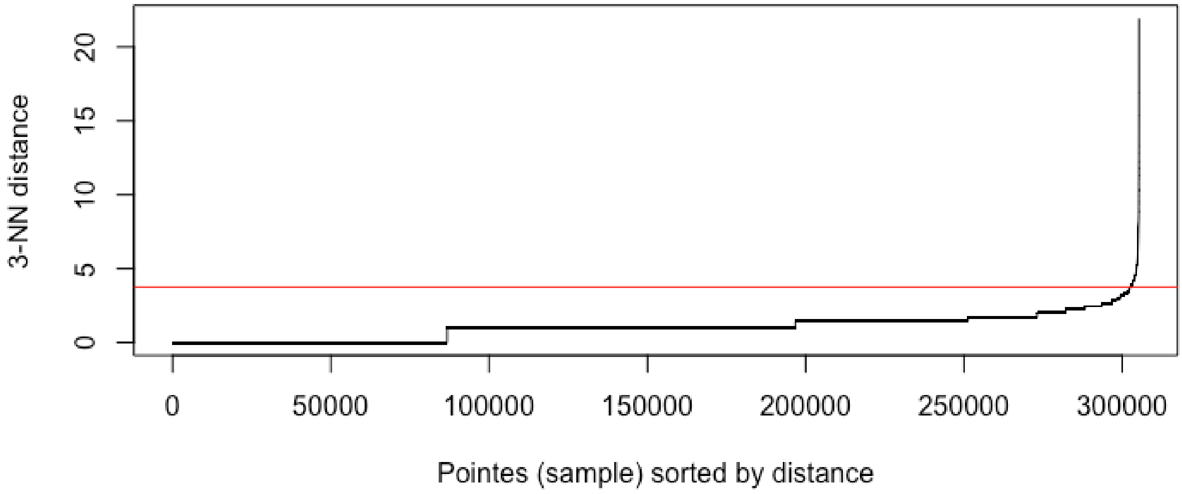
```
> table(km$cluster)

    1     2     3 
20295 48498 32973 
```

This gives us elements in each cluster.

Clustering (DBScan).

Let's run a DBScan clustering.
It takes 2 parameters, eps = The maximum radius of the neighborhood; and minPts = The minimum number of points within the eps neighborhood. Ideally minPts is chosen based on the domain knowledge; but since we do not know much about the domain, we will set minPts to dimensionality of data plus one. (Note that same goes for the distance function. We will just use the normal Euclidean distance function.)
We run the knn distance plot to get the value for eps.



Let’s run a DBScan clustering.

```
> db <- dbscan(dia_data_nomed[c(-1,-2,-3,-4)], eps=3.75, minPts=7)
> db
DBSCAN clustering for 101766 objects.
Parameters: eps = 3.75, minPts = 7
The clustering contains 2 cluster(s).
Available fields: cluster, eps, minPts
```

The DBScan clustering shows that there are 2 clusters. (remember that k-means showed that there were 3)