Is the data missing at random?

DEALING WITH MISSING DATA IN PYTHON

Suraj Donthi
Deep Learning & Computer Vision Consultant

datacamp

Possible reasons for missing data

Note — (variable → data field or column in a DataFrame)

- Values simply missing at random instances or intervals in a variable
- Values missing due to another variable
- Values missing due to the missingness of the same or another variable

Types of missingness

1. Missing Completely at Random (MCAR)
2. Missing at Random (MAR)
3. Missing Not at Random (MNAR)

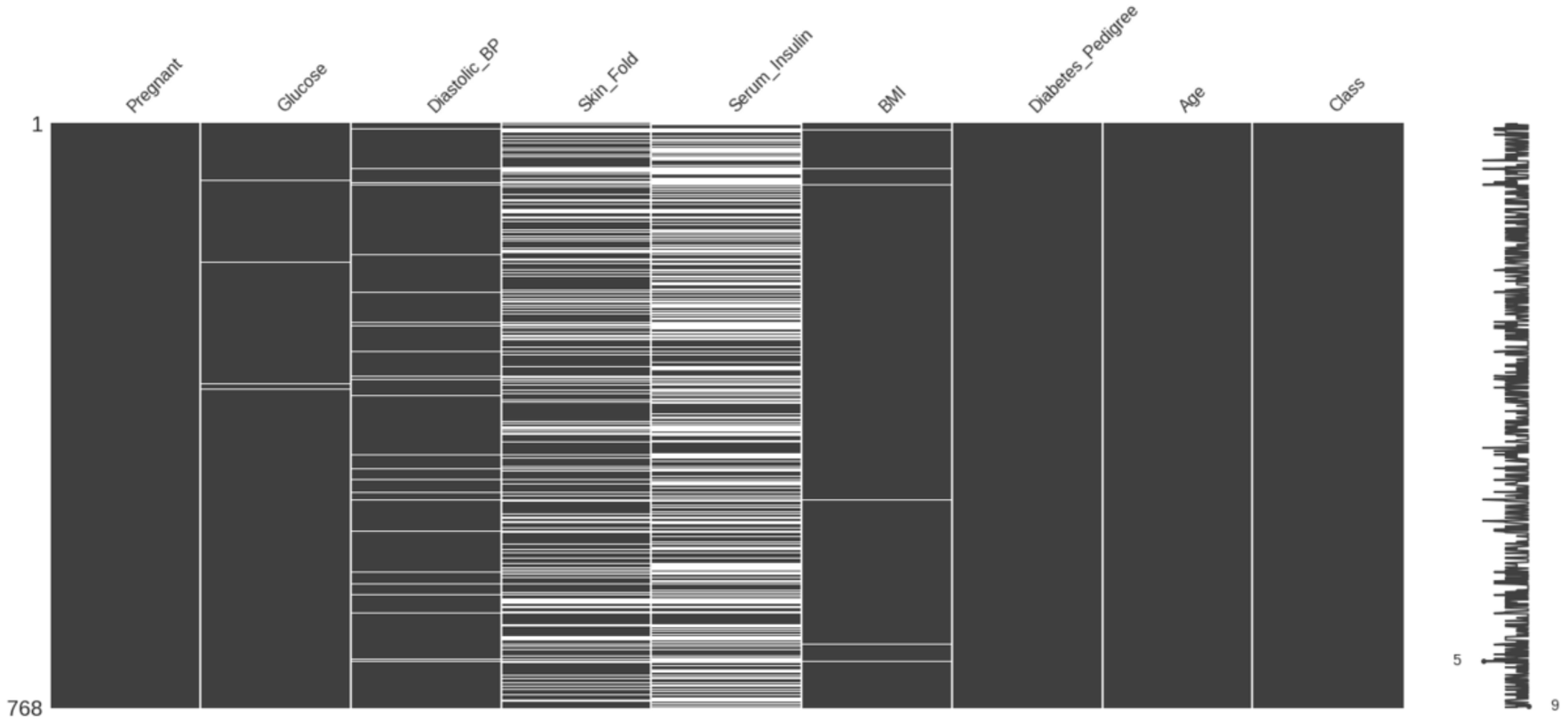
Missing Completely at Random(MCAR)

Definition:

> "Missingness has no relationship between any values, observed or missing"

MCAR - An example

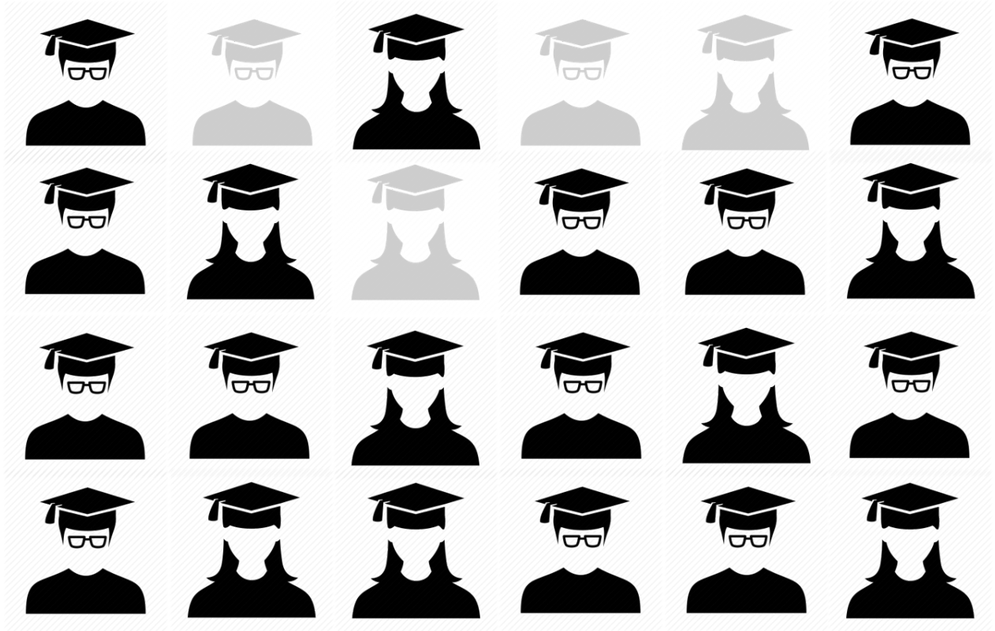
```
msno.matrix(diabetes)
```



Missing at Random(MAR)

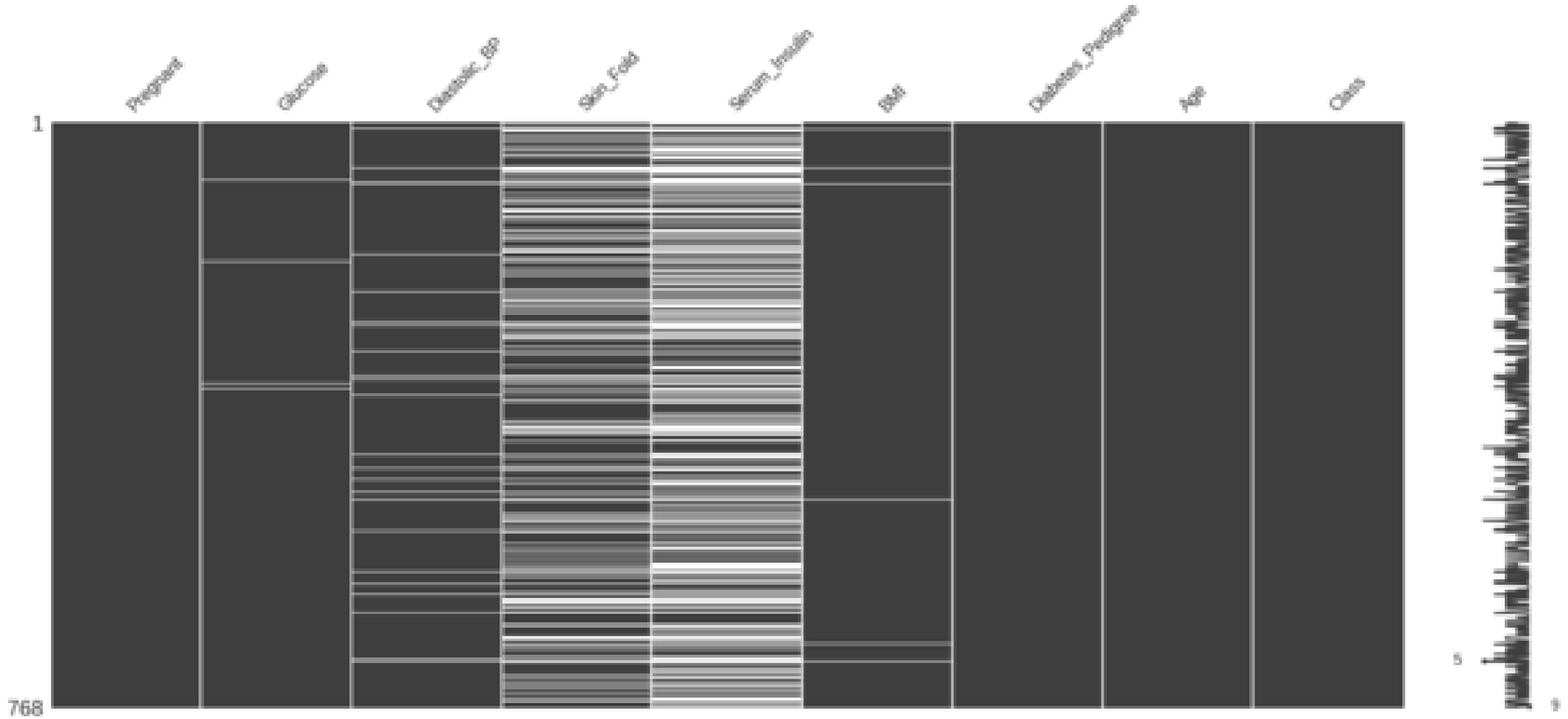
Definition:

> "There is a systematic relationship between missingness and other observed data, but not the missing data"



MAR - An example

```
msno.matrix(diabetes)
```



Missing not at Random(MNAR)

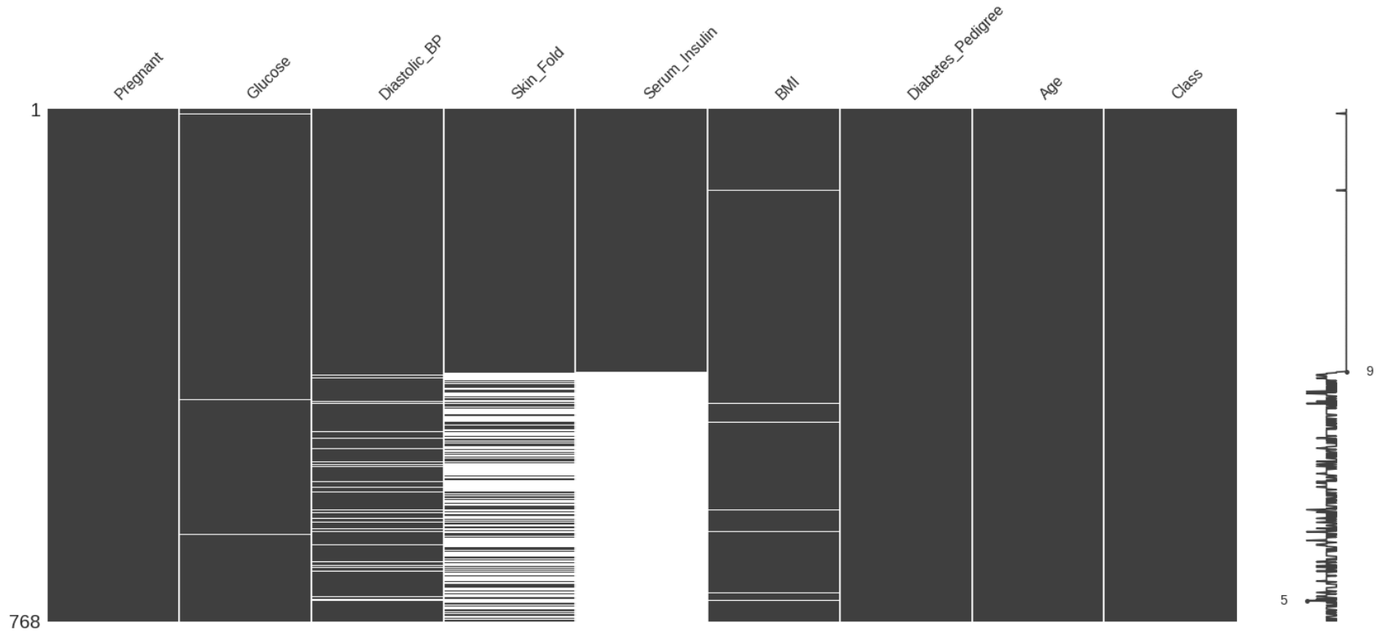
Definition:

> "There is a relationship between missingness and its values, missing or non-missing"

MNAR - An example

- Missingness pattern of the `diabetes` sorted by `Serum_Insulin`

```
sorted = diabetes.sort_values('Serum_Insulin')
msno.matrix(sorted)
```



Summary

- Possible reasons for missingness
  - Missing Completely at Random (MCAR),
  - Missing at Random (MAR) or
  - Missing Not at Random (MNAR)
- Detecting missingness pattern by sorting the variables
- Mapping missingness to MCAR, MAR & MNAR

Let's practice!

DEALING WITH MISSING DATA IN PYTHON

Finding patterns in missing data

DEALING WITH MISSING DATA IN PYTHON

Suraj Donthi

Deep Learning & Computer Vision Consultant

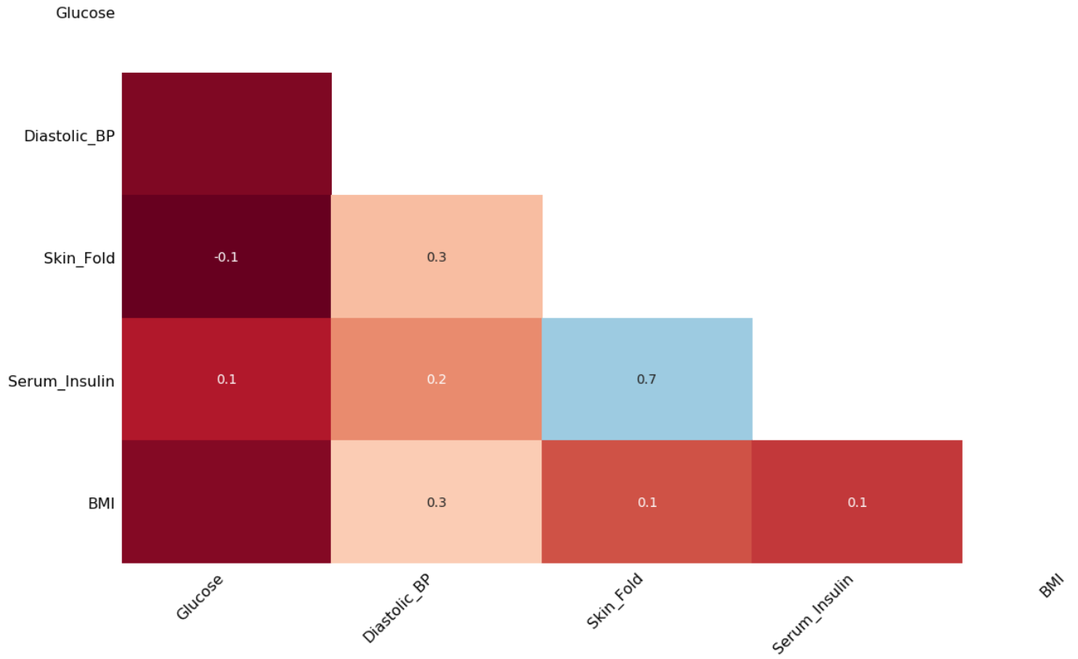
datacamp

Finding correlations between missingness

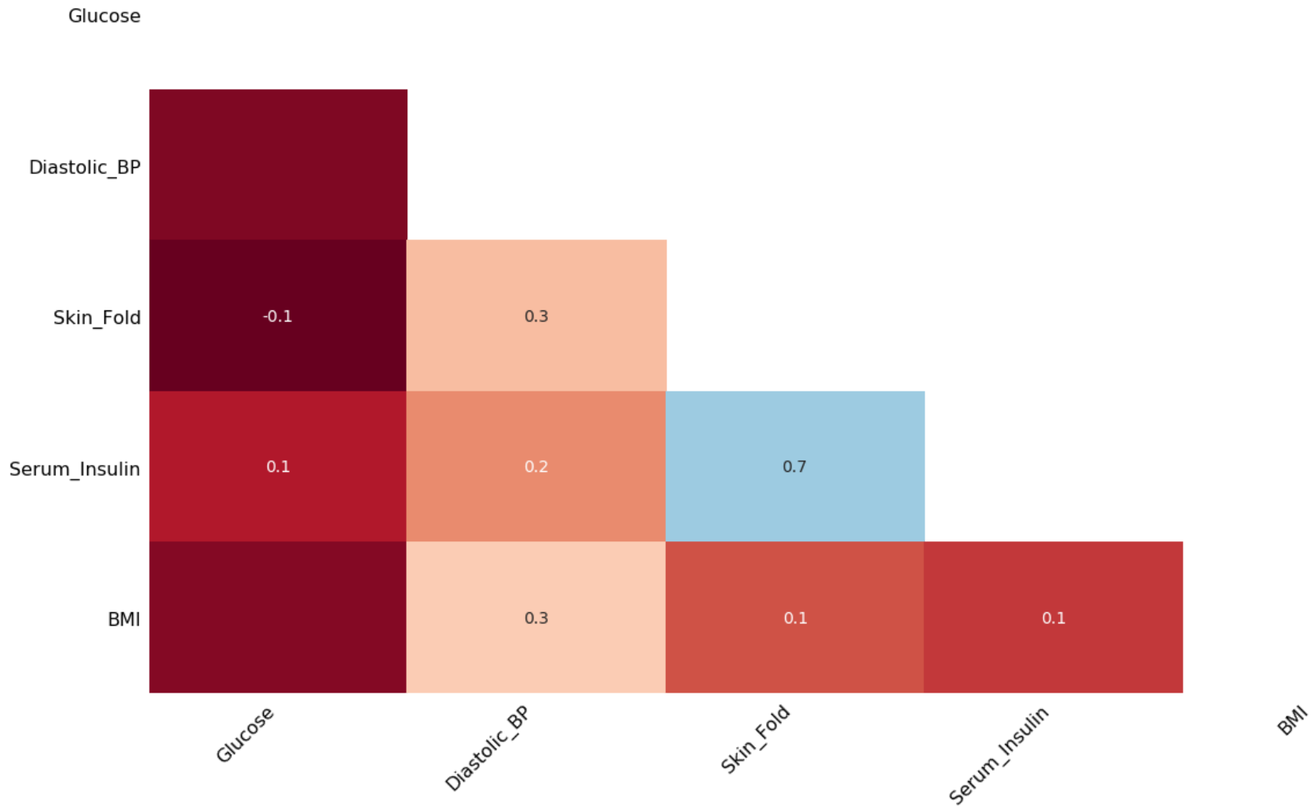
- Missingness heatmap or correlation map
- Missingness dendrogram

Missingness Heatmap

- Graph of correlation of missing values between columns
- Explains the dependencies of missingness between columns



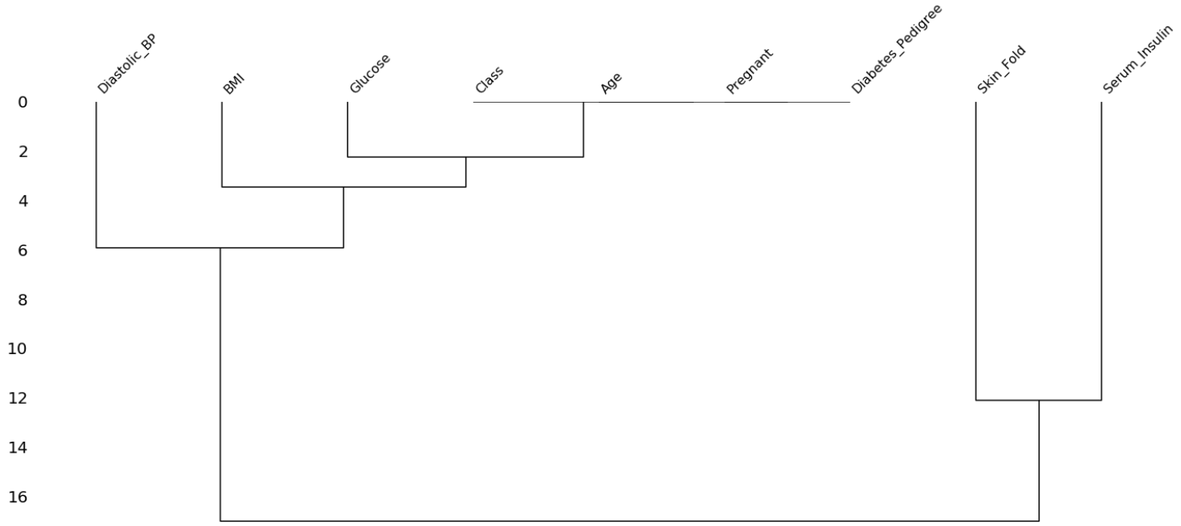
```
import missingno as msno
diabetes = pd.read_csv('pima-indians-diabetes data.csv')
msno.heatmap(diabetes)
```

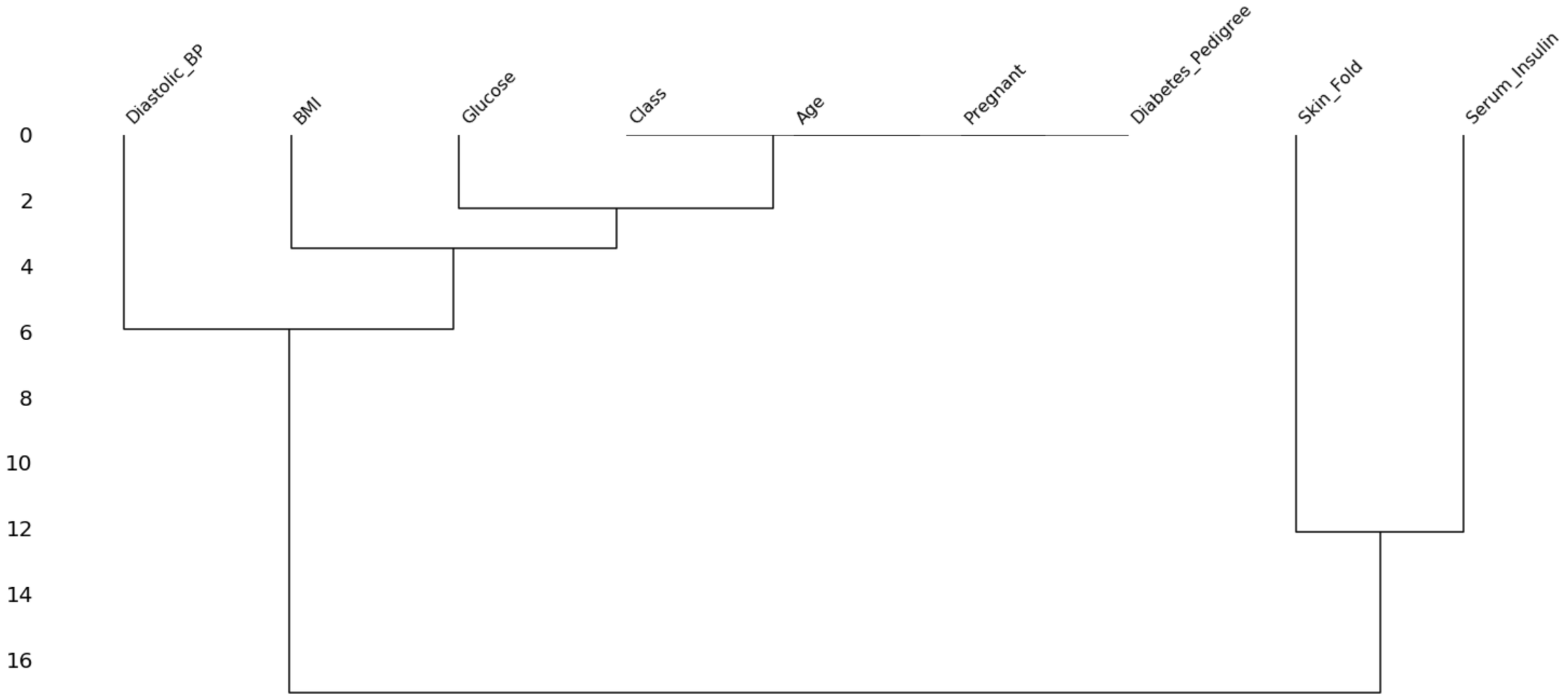


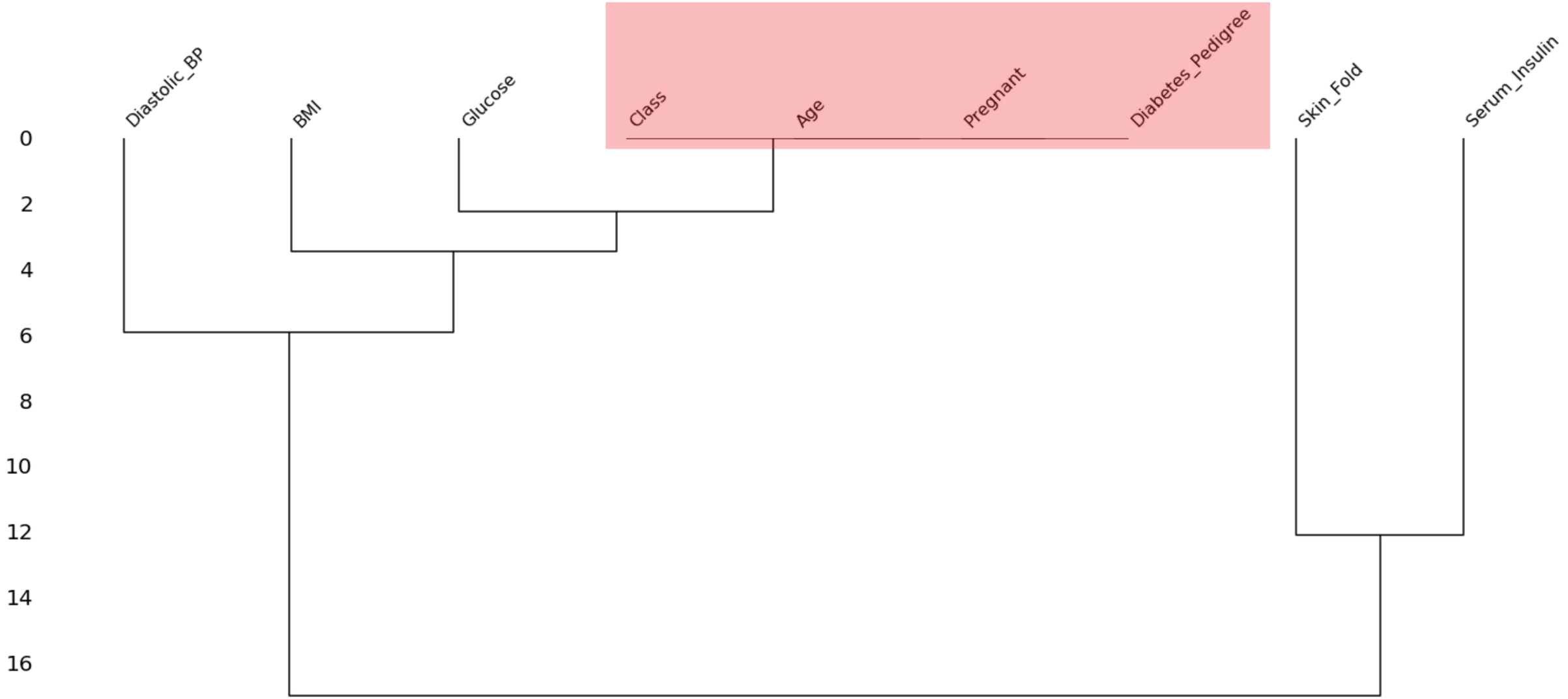
Missingness Dendrogram

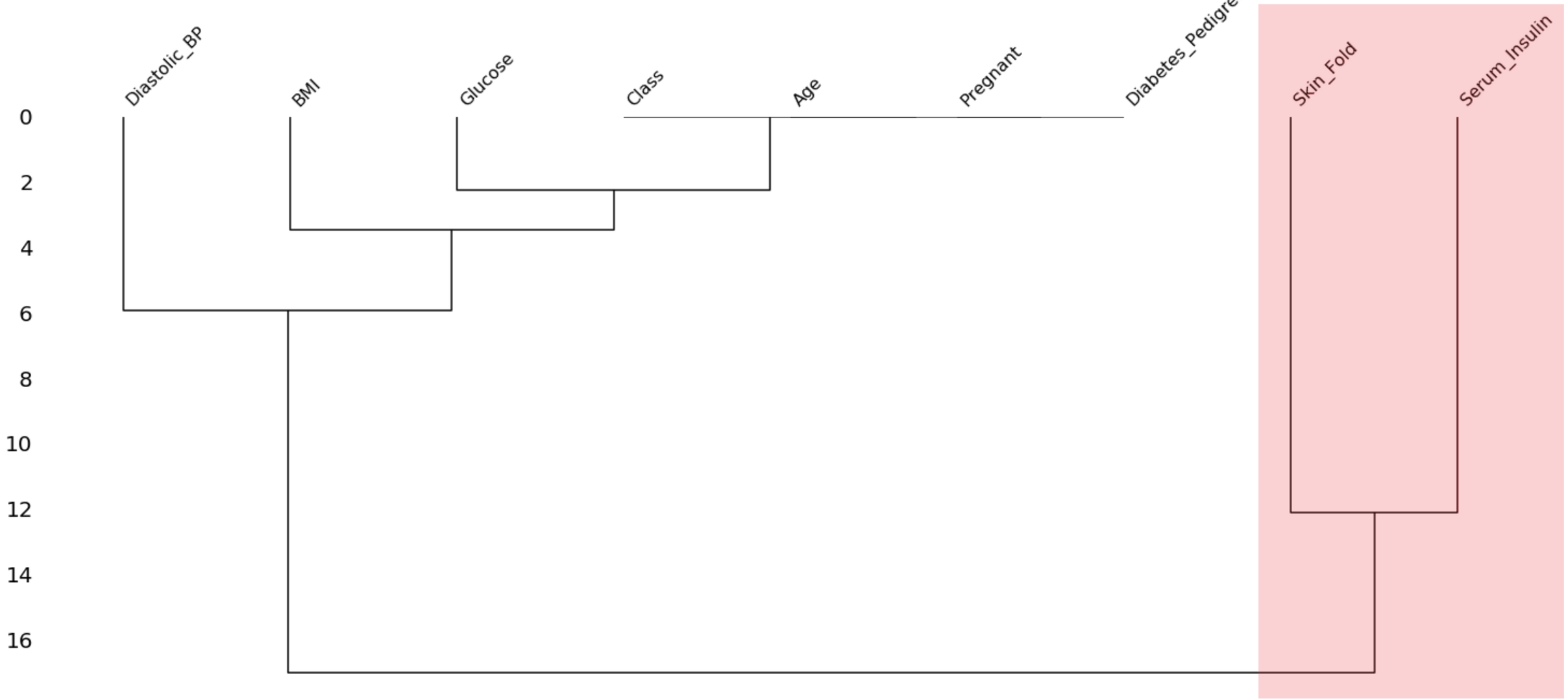
- Tree diagram of missingness
- Describes correlation of variables by grouping them

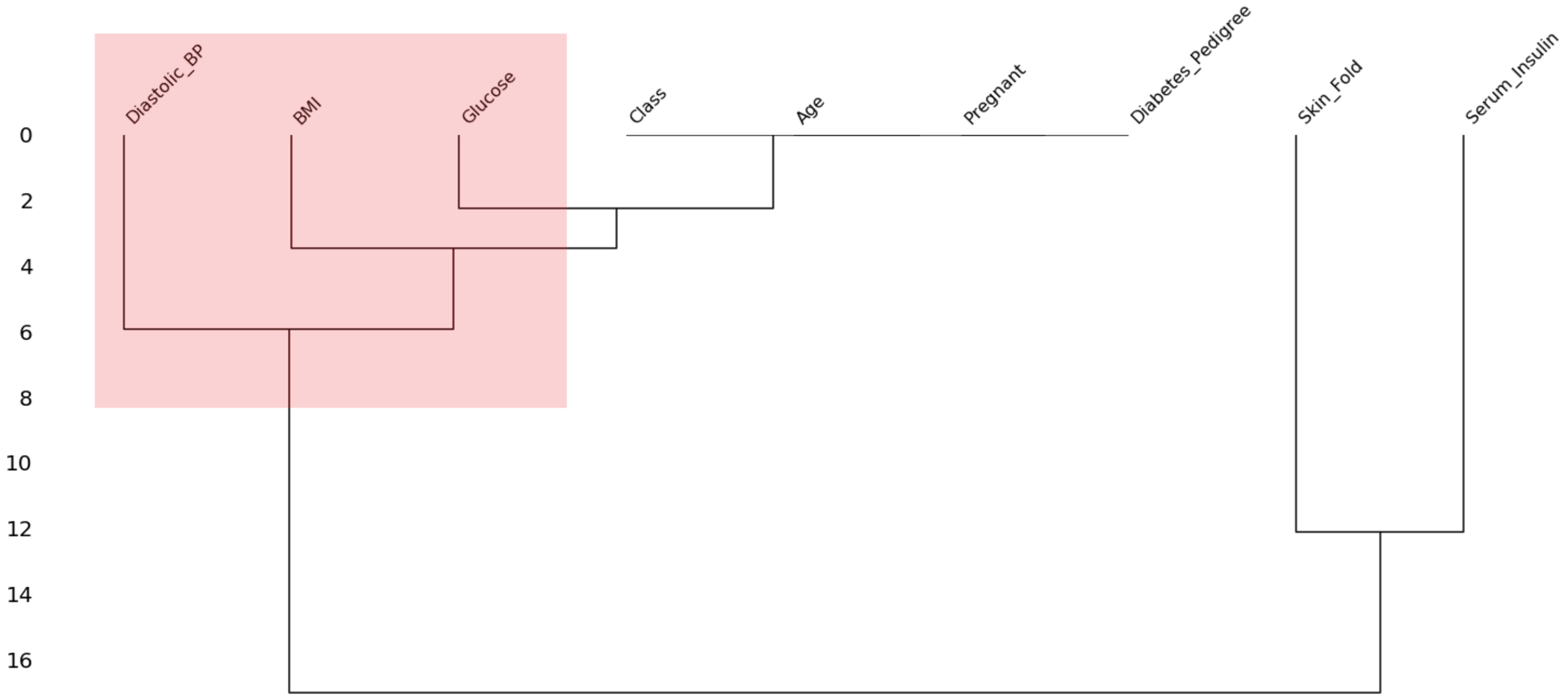
```
msno.dendrogram(diabetes)
```




Diastolic_BP
BMI
Glucose
Class
Age
Pregnant
Diabetes_Pedigree
Skin_Fold
Serum_Insulin
0
2
4
6
8
10
12
14
16


Diastolic_BP
BMI
Glucose
Class
Age
Pregnant
Diabetes_Pedigree
Skin_Fold
Serum_Insulin
0
2
4
6
8
10
12
14
16


Diastolic_BP
BMI
Glucose
Class
Age
Pregnant
Diabetes_Pedigree
Skin_Fold
Serum_Insulin
0
2
4
6
8
10
12
14
16


Diastolic_BP
BMI
Glucose
Class
Age
Pregnant
Diabetes_Pedigree
Skin_Fold
Serum_Insulin
0
2
4
6
8
10
12
14
16

Summary

- Analyze missingness heatmap

```
msno.heatmap(df)
```

- Analayze missingness dendrogram

```
msno.dendrogram(df)
```

Let's practice!

DEALING WITH MISSING DATA IN PYTHON

Visualizing missingness across a variable

DEALING WITH MISSING DATA IN PYTHON

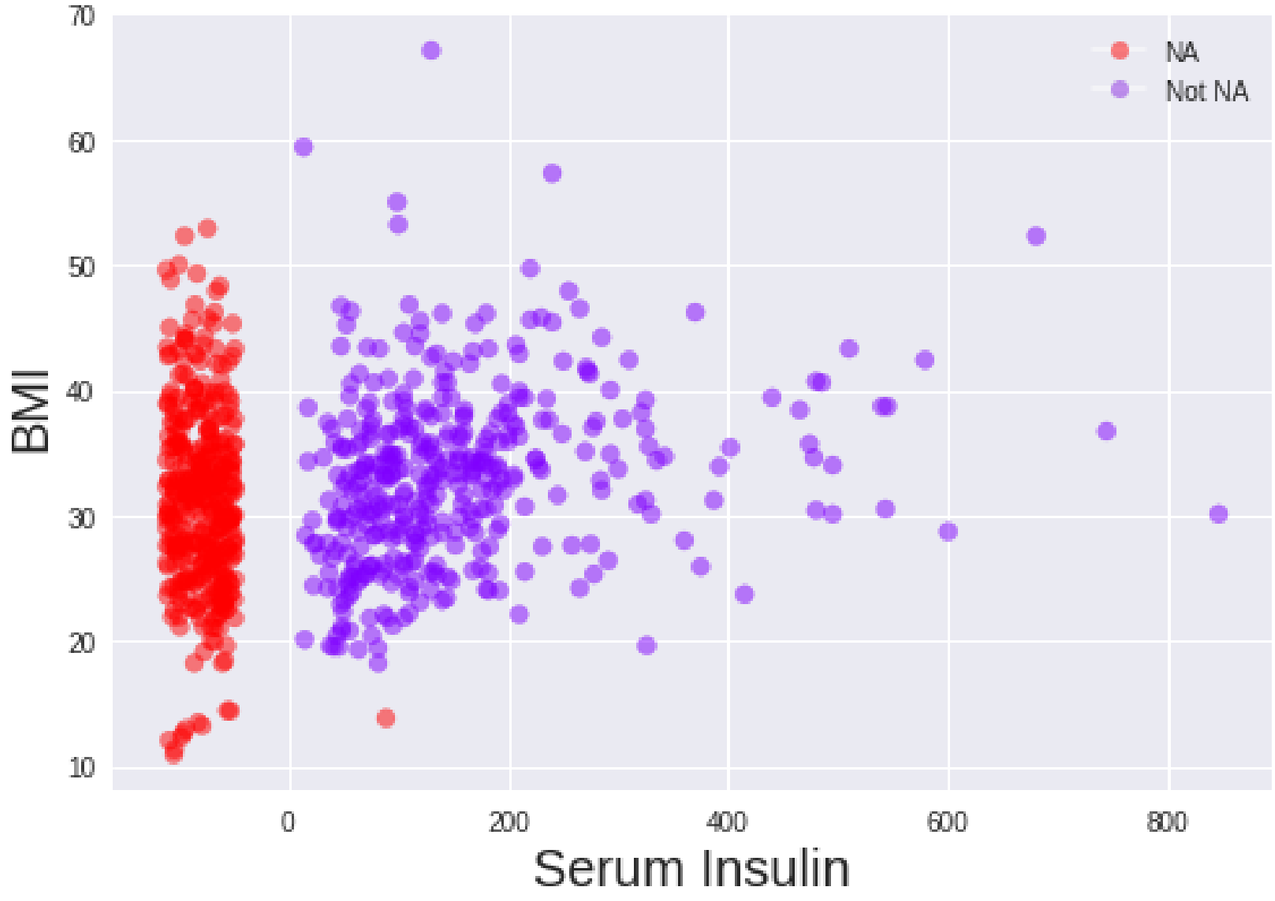
Suraj Donthi

Deep Learning & Computer Vision Consultant

datacamp

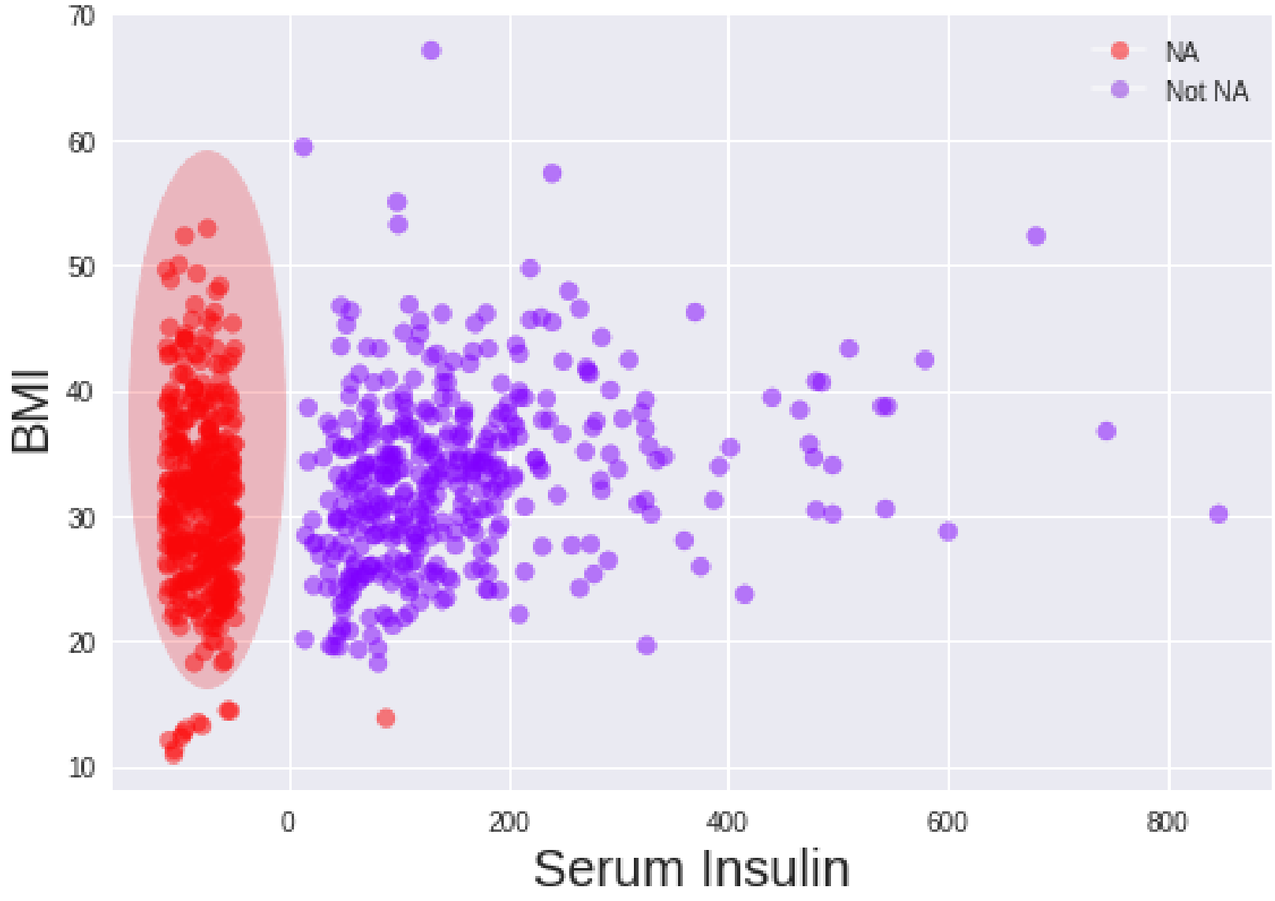
Missingness across a variable

- Visualize how missingness of a variable changes against another variable



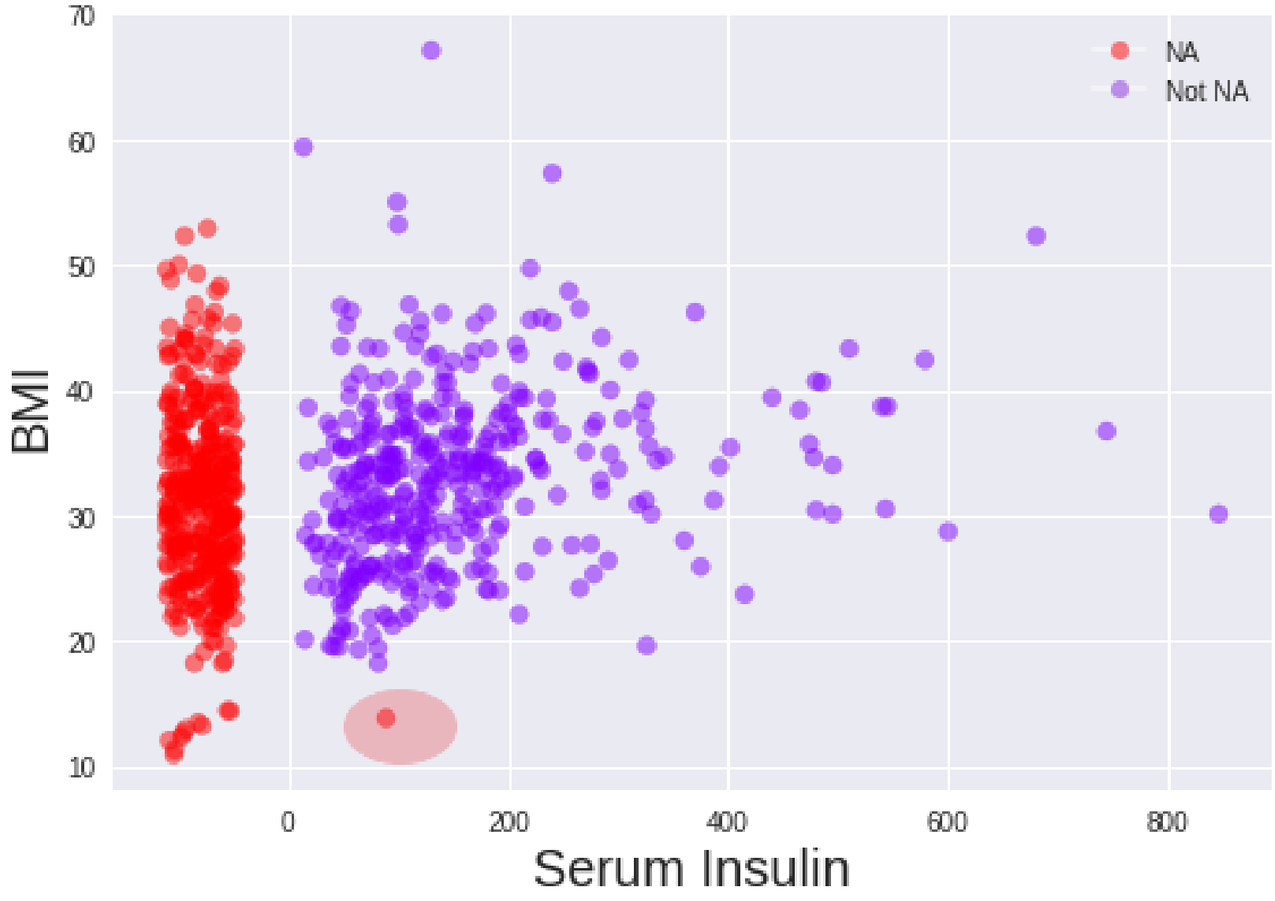
Missingness across a variable

- Visualize how missingness of a variable changes against another variable



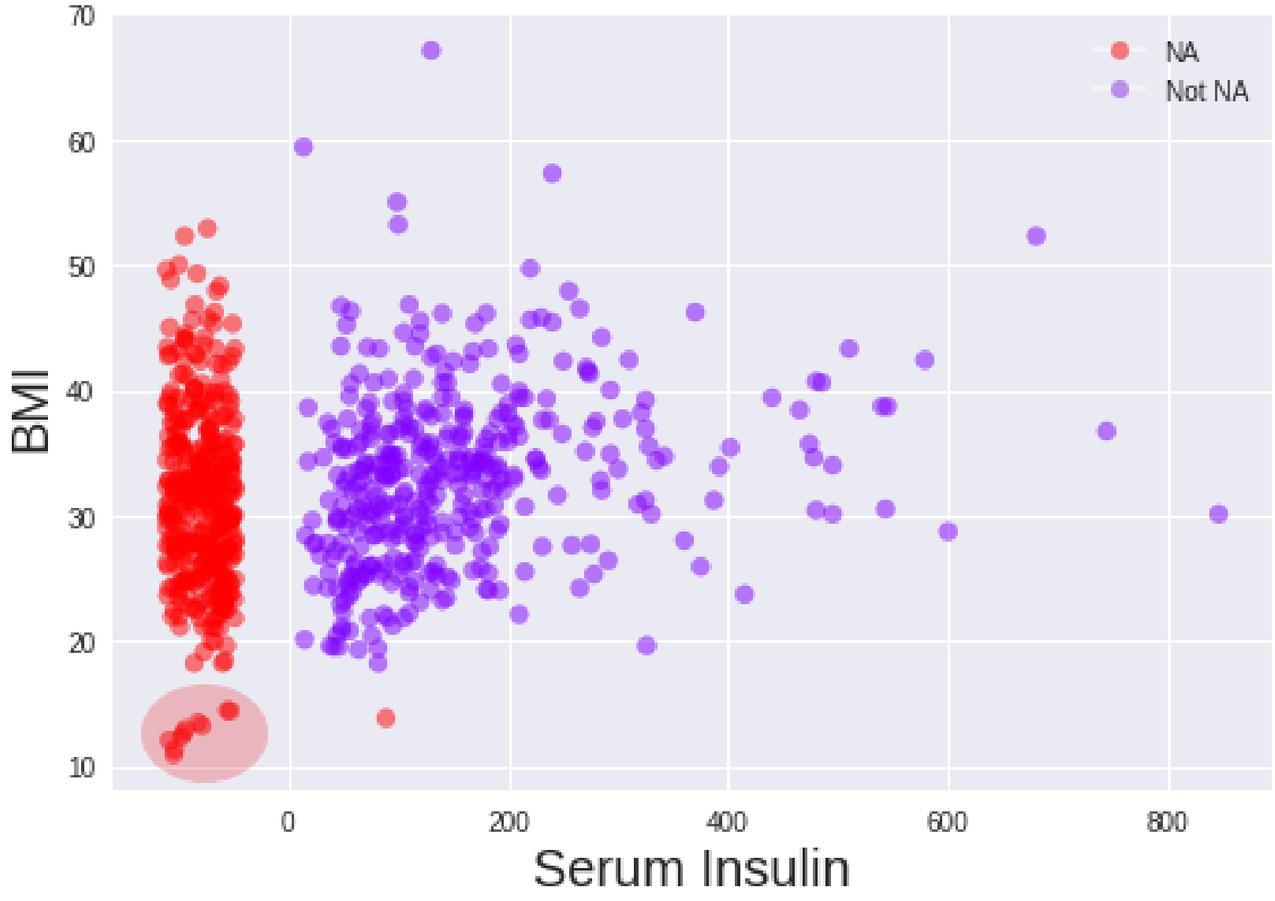
Missingness across a variable

- Visualize how missingness of a variable changes against another variable



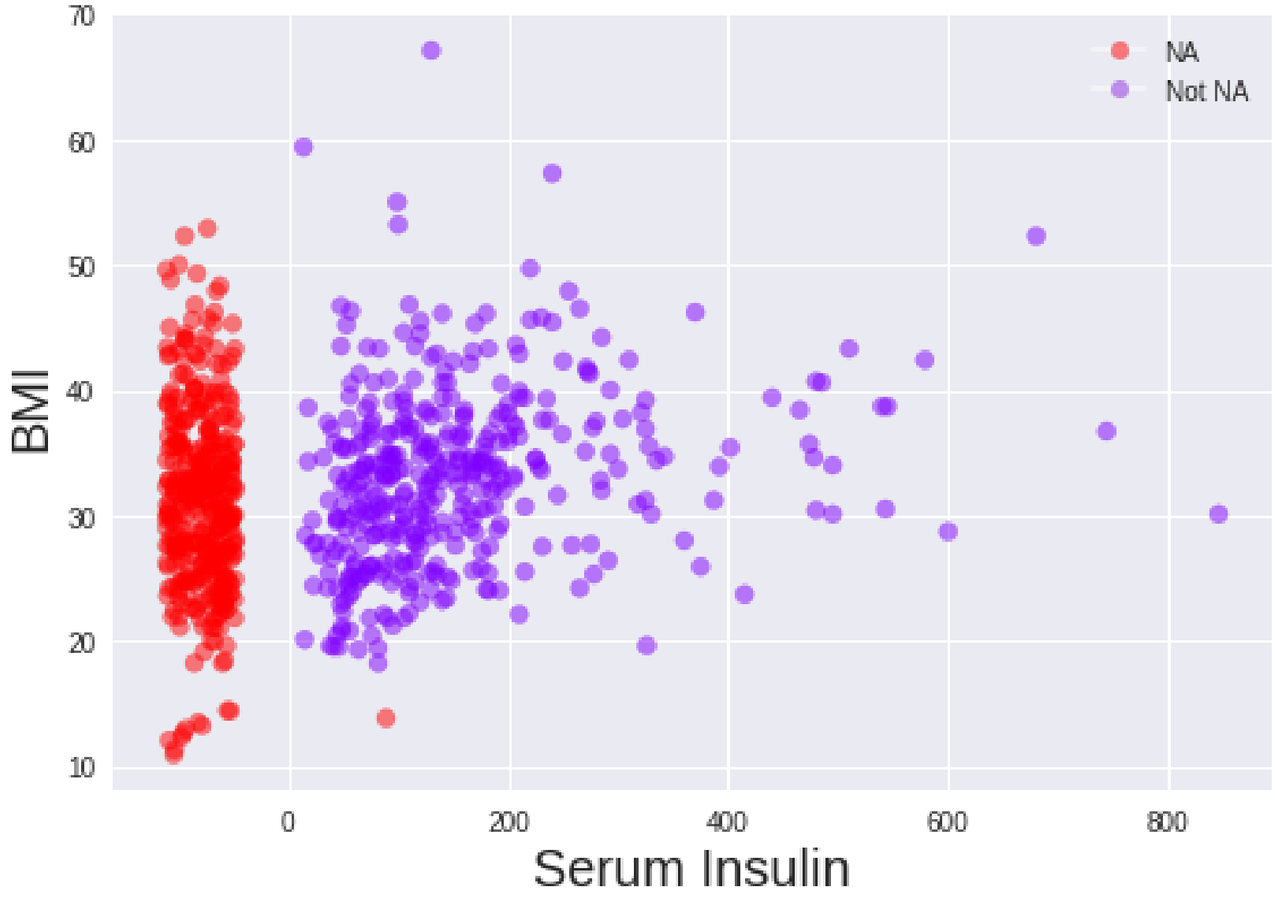
Missingness across a variable

- Visualize how missingness of a variable changes against another variable



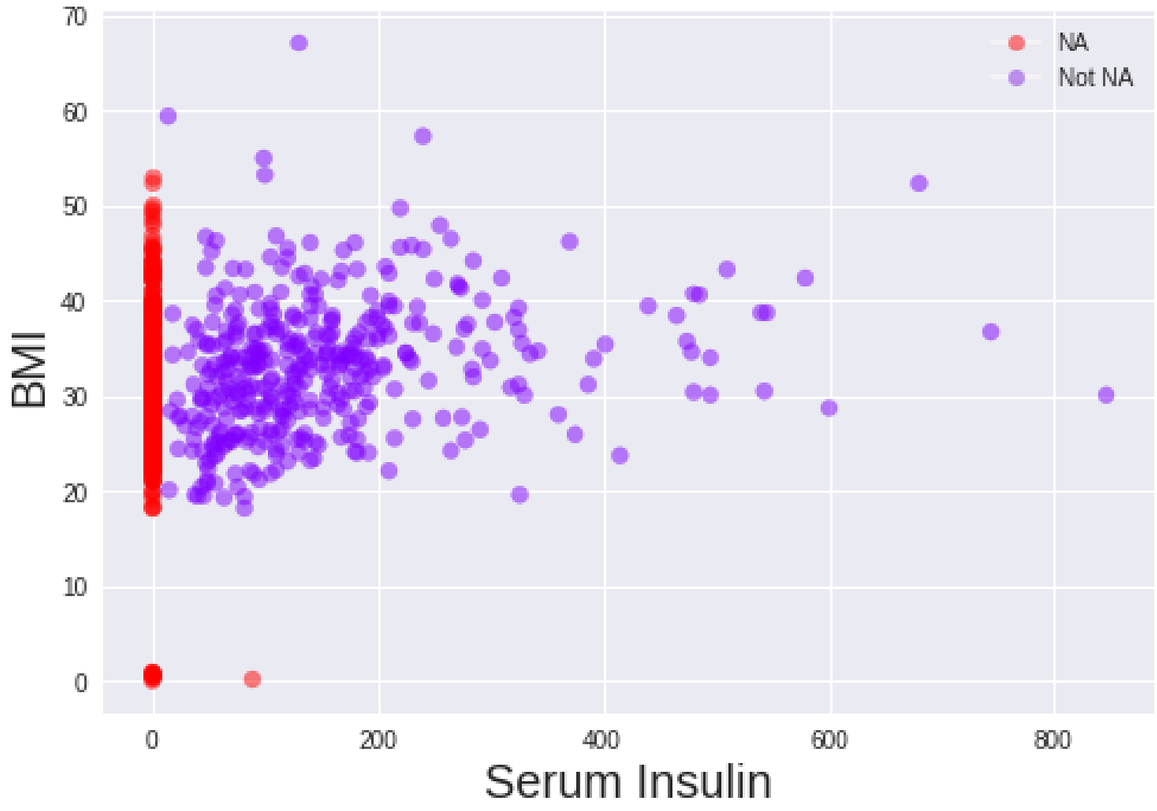
Missingness across a variable

- Visualize how missingness of a variable changes against another variable



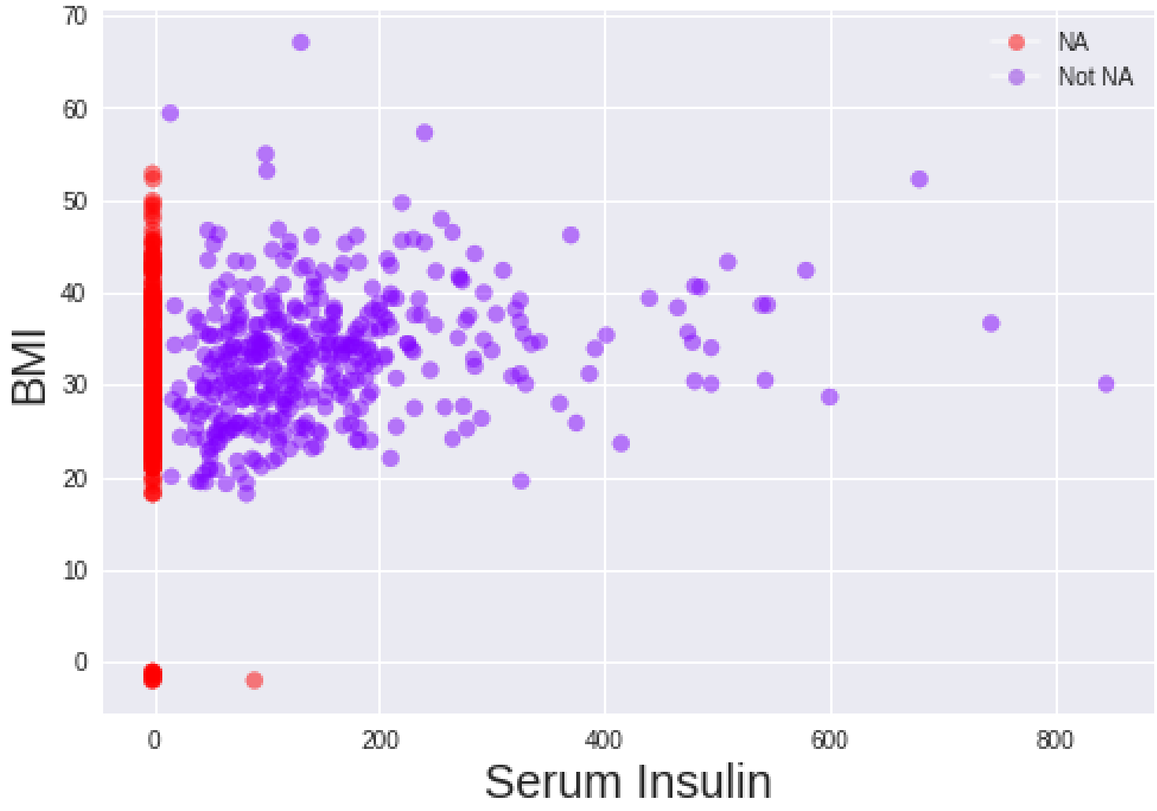
Filling dummy Values

```
from numpy.random import rand

BMI_null = diabetes['BMI'].isnull()
num_nulls = BMI_null.sum()

# Generate random values
dummy_values = rand(num_nulls)
```



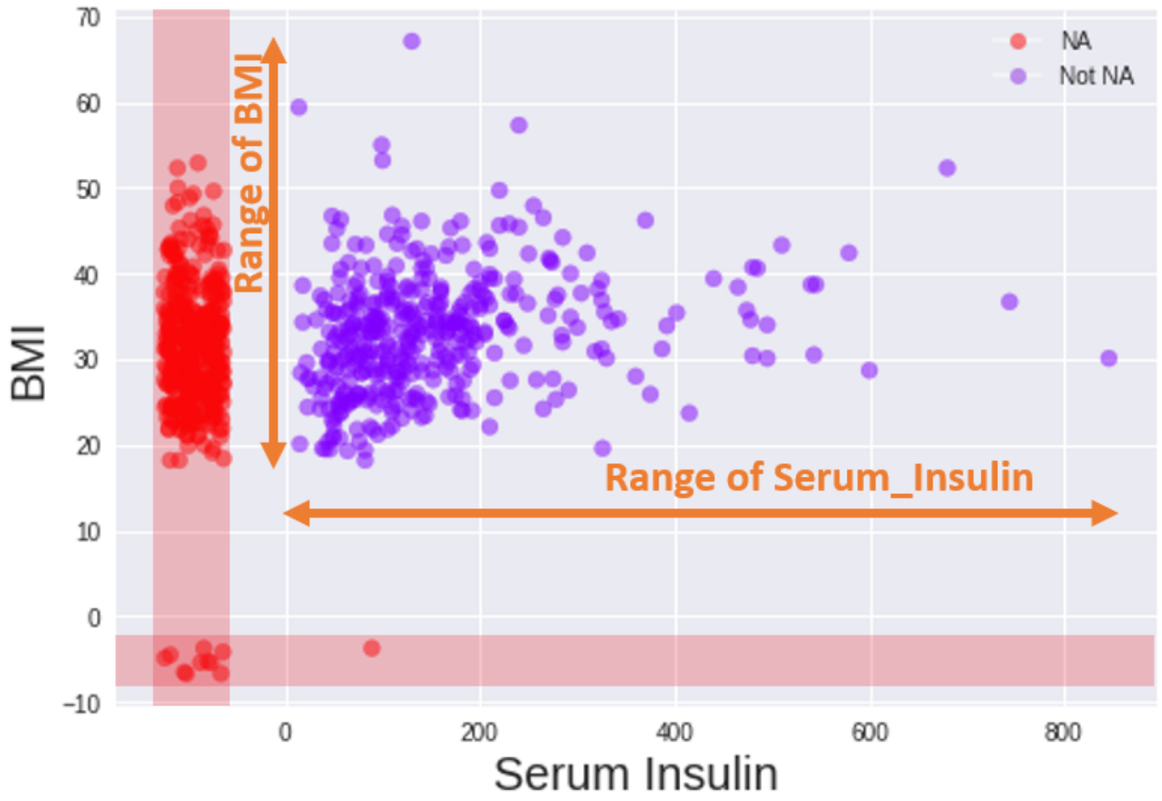
Filling dummy Values

```
from numpy.random import rand

BMI_null = diabetes['BMI'].isnull()
num_nulls = BMI_null.sum()

# Generate random values
dummy_values = rand(num_nulls)
# Shift to -2 & -1
dummy_values = dummy_values - 2
```



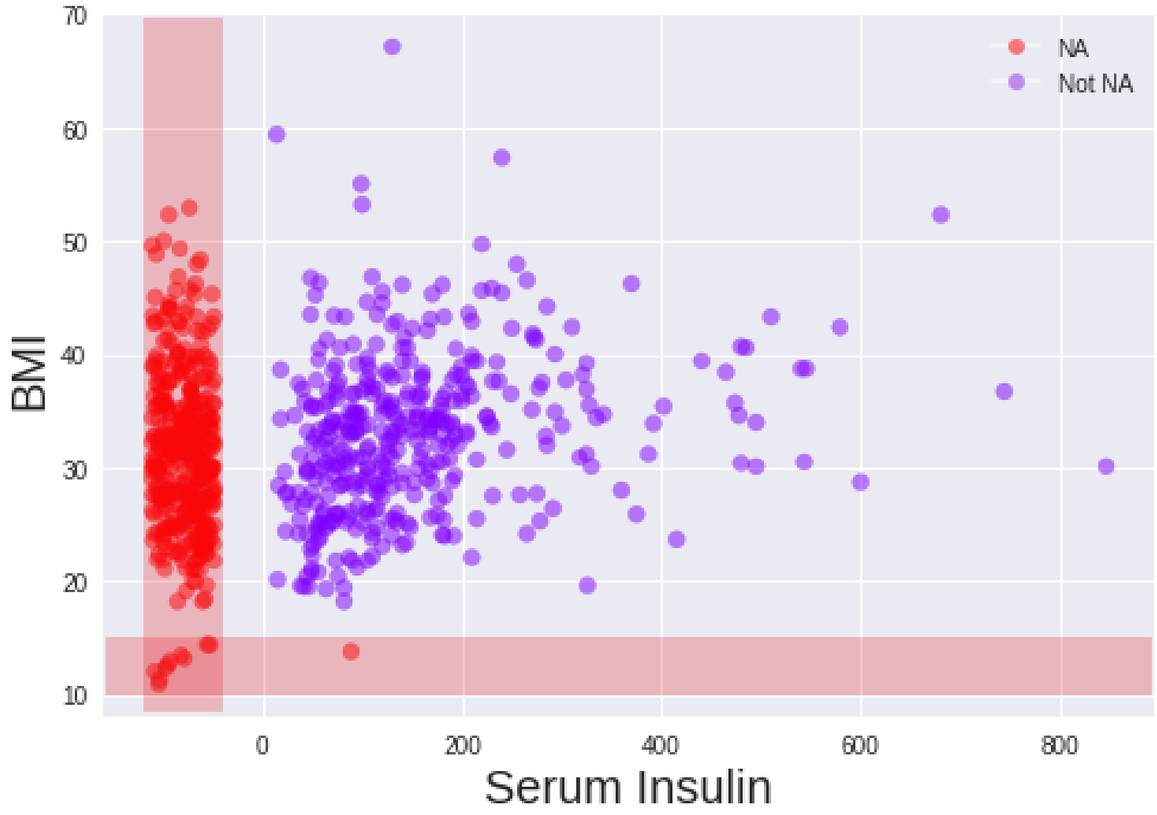
Filling dummy Values

```
from numpy.random import rand

BMI_null = diabetes['BMI'].isnull()
num_nulls = BMI_null.sum()

# Generate random values
dummy_values = rand(num_nulls)
# Shift to -2 & -1
dummy_values = dummy_values - 2
# Scale to 0.075 of Column Range
BMI_range = BMI.max() - BMI.min()
dummy_values = dummy_values * 0.075 * BMI_range
```



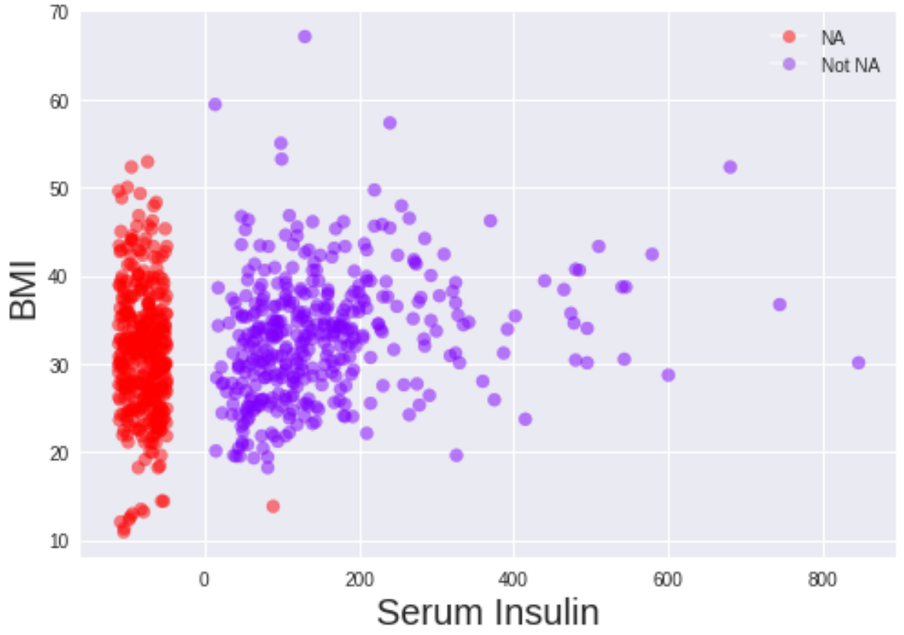
Filling dummy Values

```
from numpy.random import rand

BMI_null = diabetes['BMI'].isnull()
num_nulls = BMI_null.sum()

# Generate random values
dummy_values = rand(num_nulls)
# Shift to -2 & -1
dummy_values = dummy_values - 2
# Scale to 0.075 of Column Range
BMI_range = BMI.max() - BMI.min()
dummy_values = dummy_values * 0.075 * BMI_range
# Shift to Column Minimum
dummy_values = (rand(num_nulls) - 2)
        * 0.075 * BMI_range + BMI.min()
```



```
from numpy.random import rand

def fill_dummy_values(df, scaling_factor):
  # Create copy of dataframe
  df_dummy = df.copy(deep=True)
  # Iterate over each column
  for col in df_dummy:

    # Get column, column missing values and range
    col = df_dummy[col]
    col_null = col.isnull()
    num_nulls = col_null.sum()
    col_range = col.max() - col.min()

    # Shift and scale dummy values
    dummy_values = (rand(num_nulls) - 2)
    dummy_values = dummy_values * scaling_factor * col_range + col.min()

    # Return dummy values
    col[col_null] = dummy_values
  return df_dummy
```

```
# Create dummy dataframe
diabetes_dummy = fill_dummy_values(diabetes)
# Get missing values of both columns for coloring
nullity=diabetes.Serum_Insulin.isnull()+diabetes.BMI.isnull()
# Generate scatter plot
diabetes_dummy.plot(x='Serum_Insulin', y='BMI', kind='scatter', alpha=0.5,
                    c=nullity, cmap='rainbow')
```



Let's practice!

DEALING WITH MISSING DATA IN PYTHON

When and how to delete missing data

DEALING WITH MISSING DATA IN PYTHON

Suraj Donthi

Deep Learning & Computer Vision Consultant

datacamp

Types of deletions

1. Pairwise deletion
2. Listwise deletion

Note: Used when the values are MCAR.

Pairwise Deletion

`diabetes` DataFrame

Pregnant	Glucose	Diastolic_BP	...
6	148	72	...
5	NaN	80	...
1	89	66	...
1	NaN	74	...
...	...	...	...
8	183	64	...
6	NaN	68	...

768 rows × 9 columns

```
diabetes['Glucose'].mean()
```

```
121.687
```

```
diabetes.count()
```

```
763
```

```
diabetes['Glucose'].sum() /
       diabetes['Glucose'].count()
```

```
121.687
```

Listwise Deletion or Complete Case

`diabetes` DataFrame

Pregnant	Glucose	Diastolic_BP	...
6	148	72	...
5	NaN	80	...
1	89	66	...
1	NaN	74	...
...	...	...	...
8	183	64	...
6	NaN	68	...

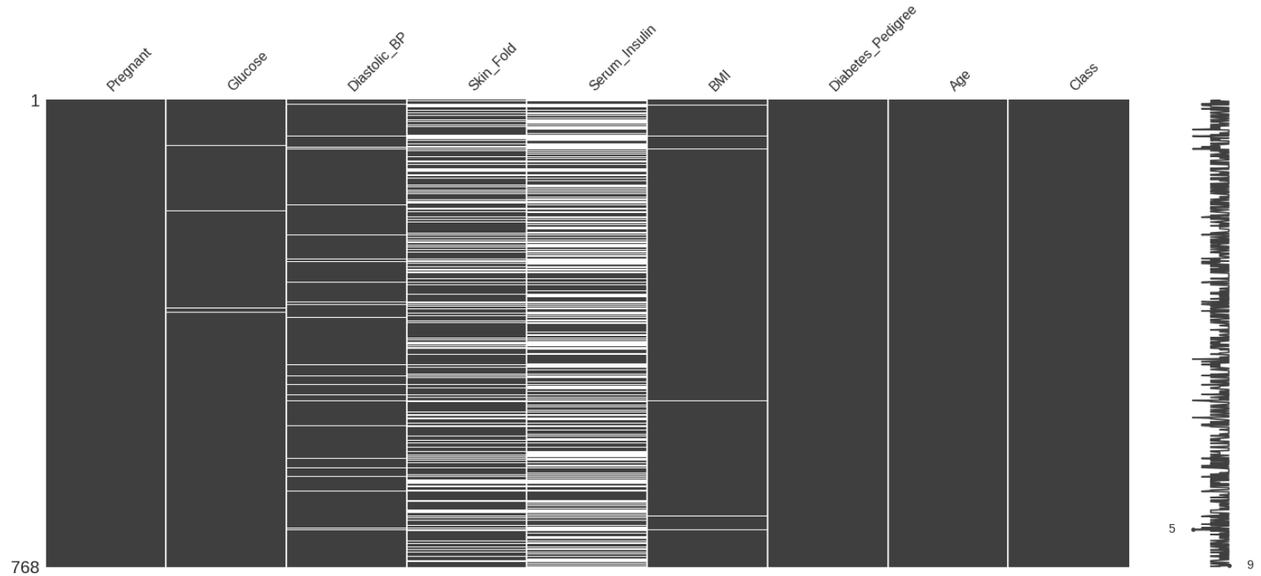
768 rows × 9 columns

```
diabetes.dropna(subset=['Glucose'],
                how='any',
                inplace=True)
```

Deletion in diabetes DataFrame

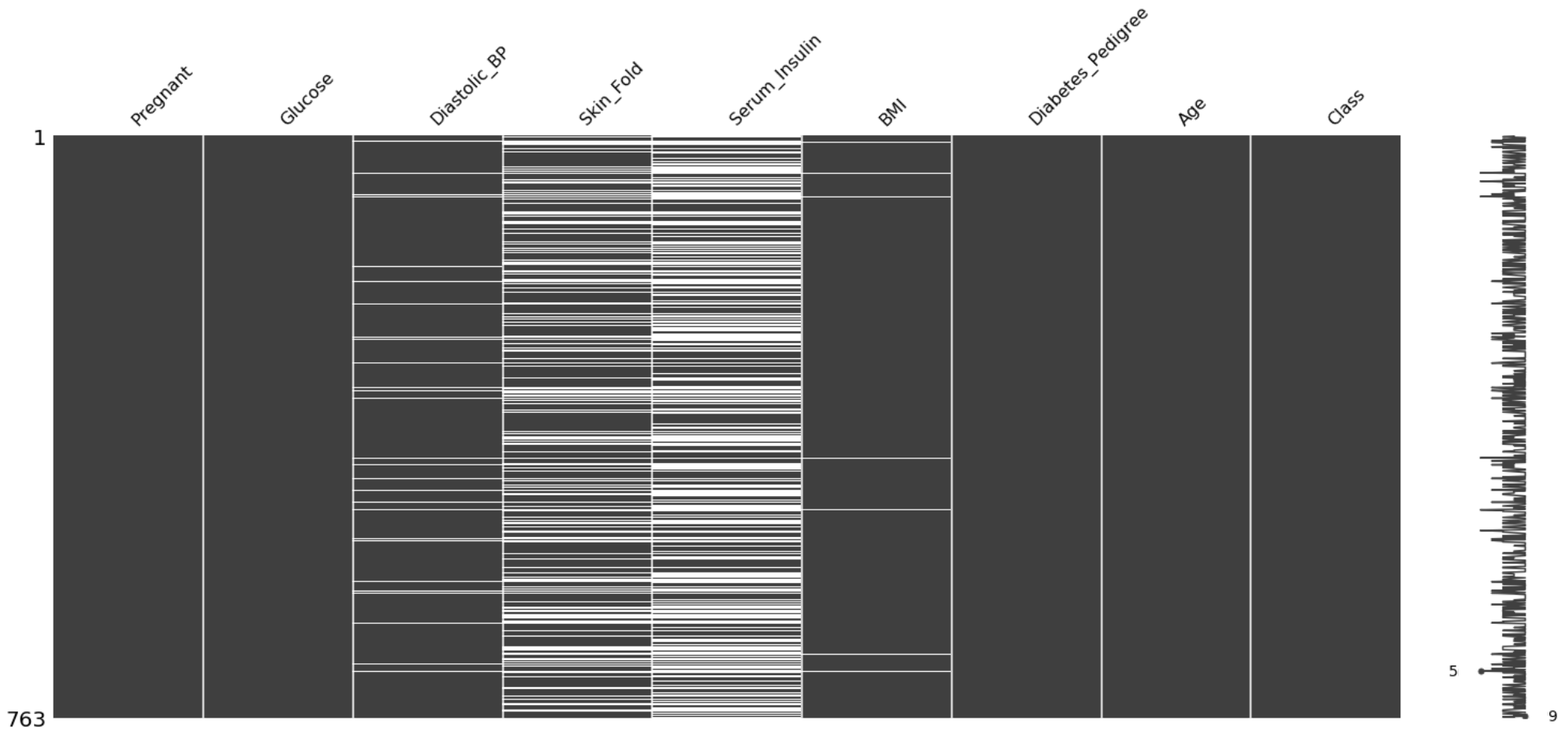
```
msno.matrix(diabetes)
diabetes['Glucose'].isnull().sum()
```

```
5
```



Deletion in diabetes DataFrame

```
diabetes.dropna(subset=["Glucose"], how='any', inplace=True)
msno.matrix(diabetes)
```

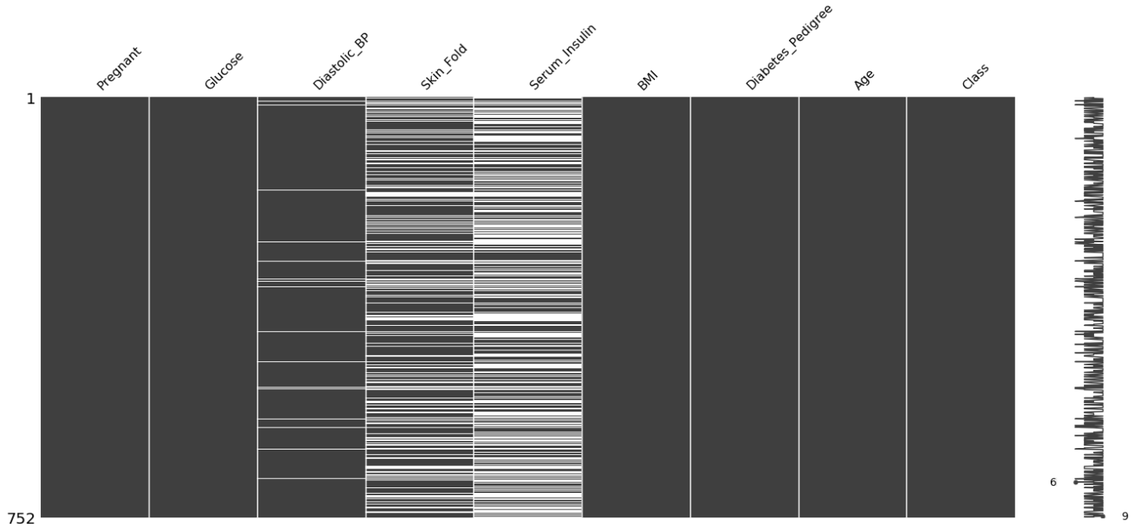


Deletion in diabetes DataFrame

```
diabetes['BMI'].isnull().sum()
```

```
11
```

```
diabetes.dropna(subset=["BMI"], how='any', inplace=True)
msno.matrix(diabetes)
```



Summary

- Pairwise deletion
- Listwise deletion
- Deletion is used only when values are MCAR

Let's practice!

DEALING WITH MISSING DATA IN PYTHON