Do You Miss Me?

Tom Ron, July 2023

Tom Ron

Engineering Manager with a soul of a data scientist

Occasionally writing on tomron.net

Mechanism of Missing Data

MCAR - Missing Completely At Random	**The fact that the data are missing is independent of the observed and unobserved data**
MAR - Missing At Random	**The fact that the data are missing is systematically related to the observed but not the unobserved data**
MNAR - Missing Not At Random	**The fact that the data are missing is systematically related to the unobserved data**

https://www.ncbi.nlm.nih.gov/books/NBK493614/

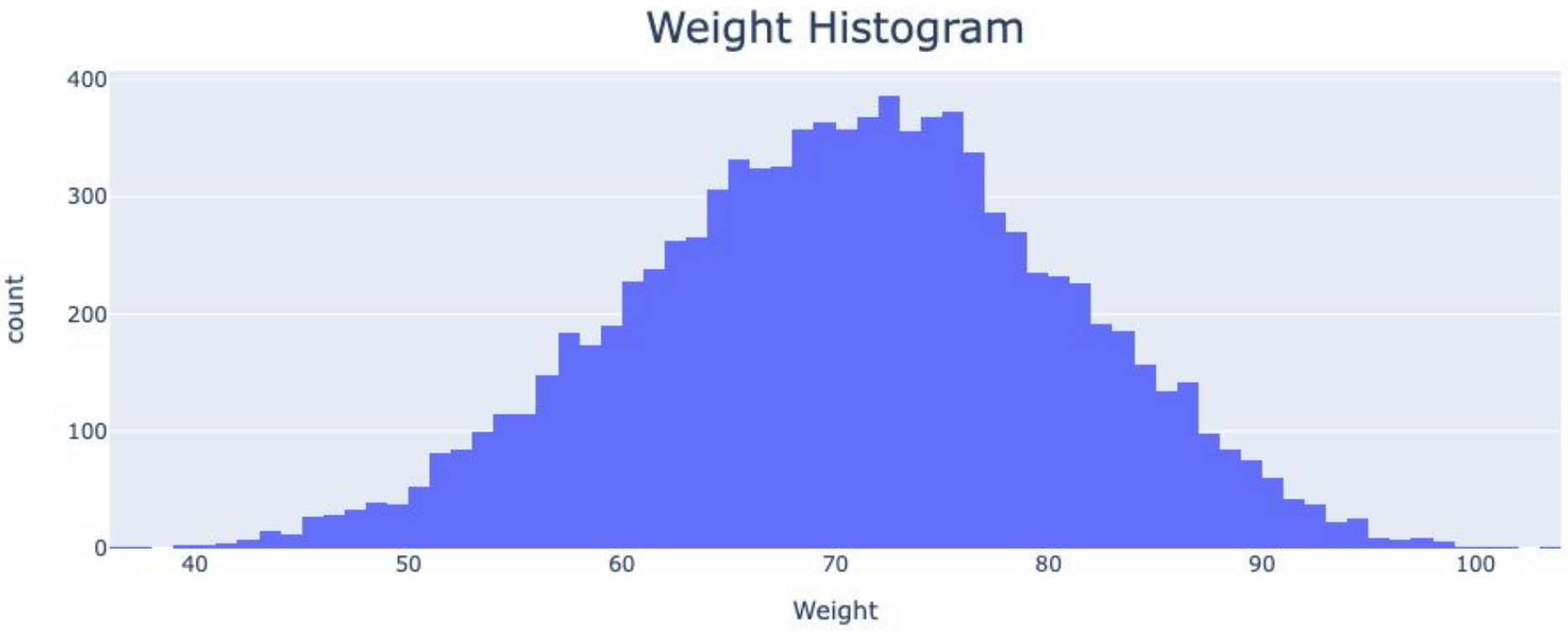
Terminology

Imputation	**the process of replacing missing data with substituted values**

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   gender  10000 non-null  object 
 1   weight  9554 non-null   float64
 2   height  9316 non-null   float64
dtypes: float64(2), object(1)
memory usage: 234.5+ KB
```

```
df['weight'].plot(kind='hist')
```



```
df['weight'].fillna(df['weight'].mean())
```

And we can do the same thing with scikit-learn -

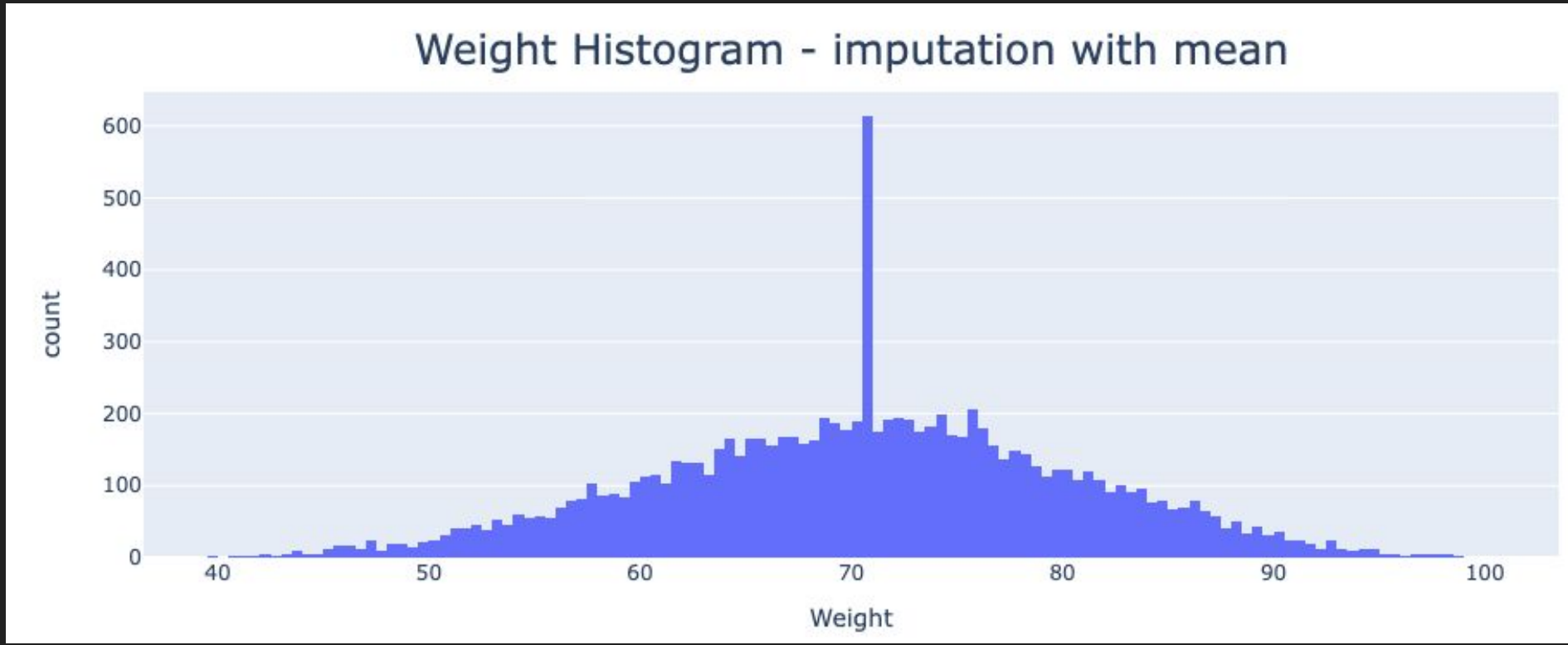
```
from sklearn.impute import SimpleImputer

simple_imputer = SimpleImputer(strategy='mean')

df['weight_average_simple'] =
simple_imputer.fit_transform(df[['weight']])
```

SimpleImputer vs fillna

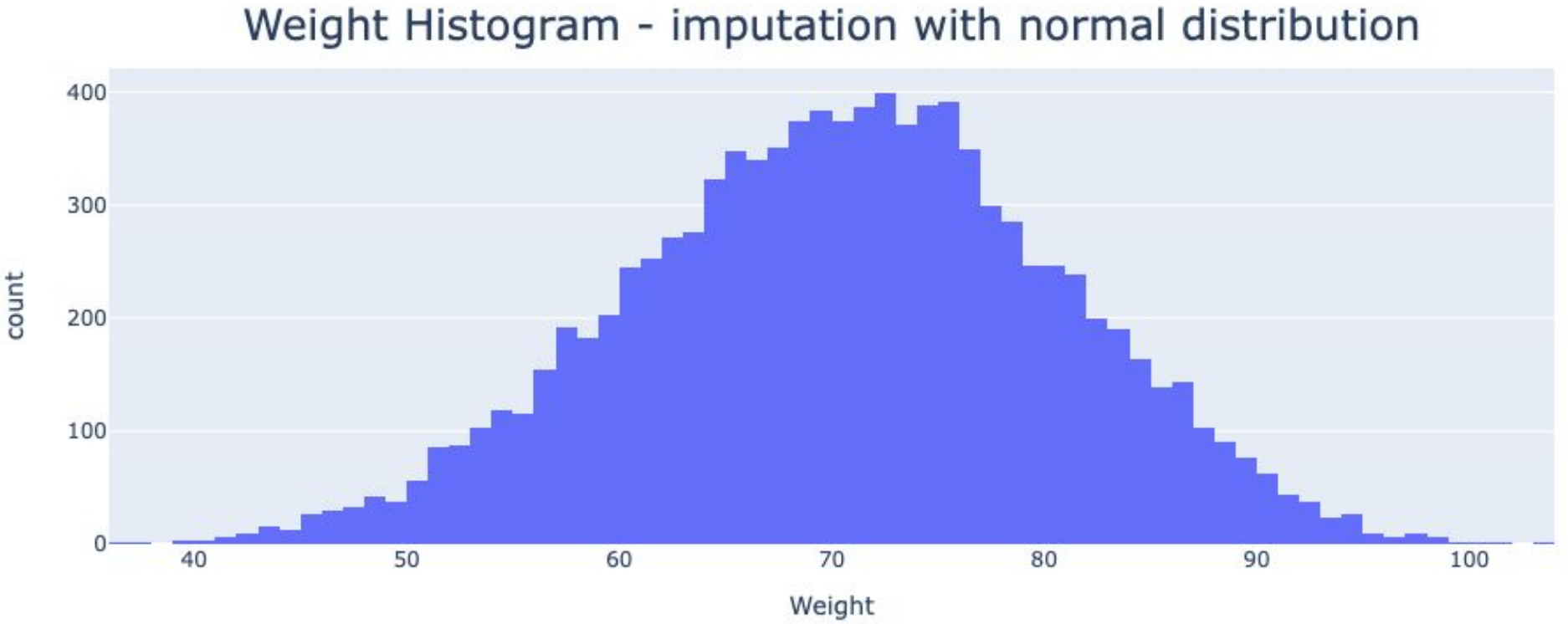
SimpleImputer	fillna
Limited filling options	Flexible filling options
Missing Indicator	Backfill and forward fill

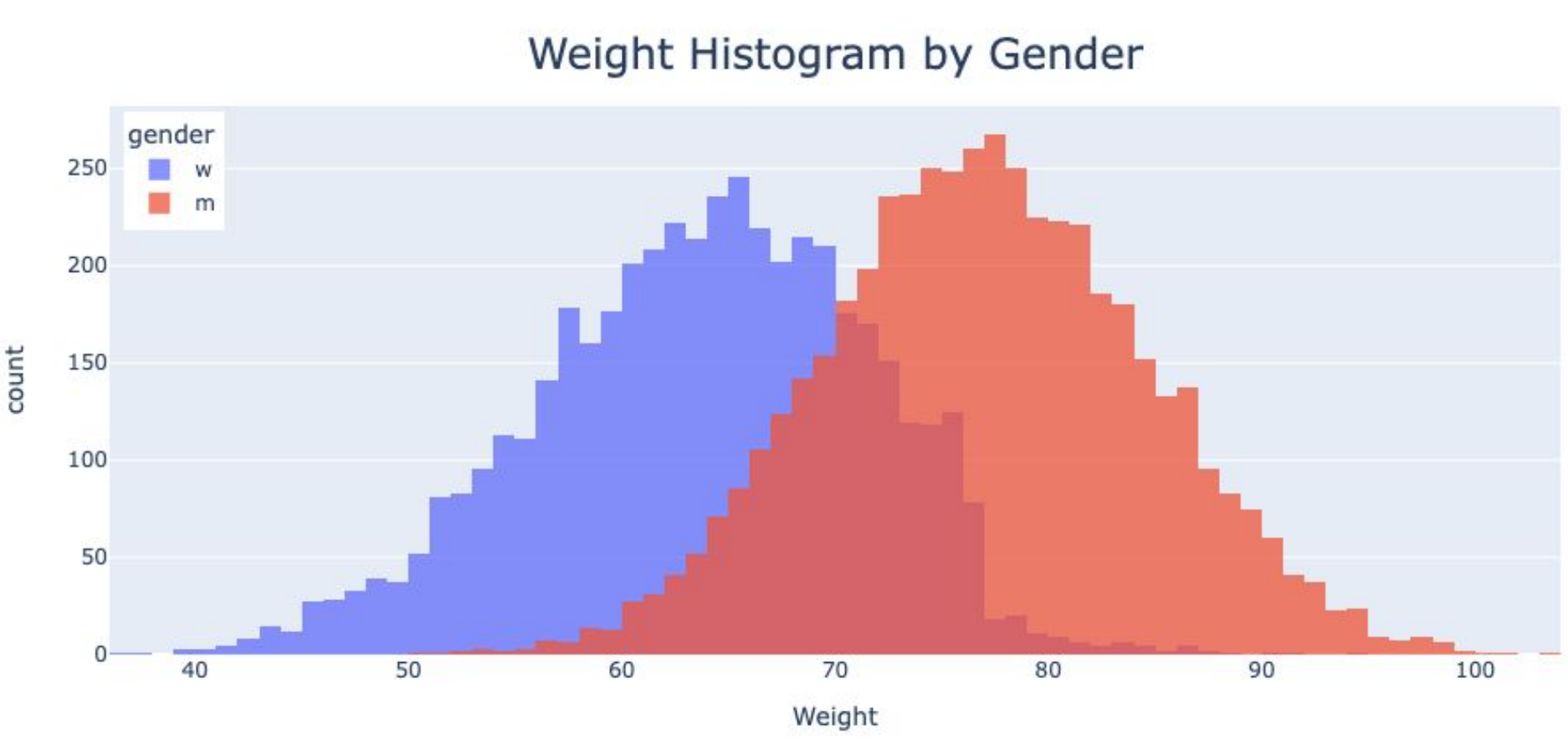
https://tomron.net/2023/06/21/pandas-fillna-vs-scikit-learn-simpleimputer/

```
df['weight'].fillna(df['weight'].mean())
```



```
index = df[df['weight'].isna()].index
mean = df['weight'].mean()
loc = df['weight'].std()
df['weight_normal'] = df['weight'].fillna(pd.Series(np.random.normal(mean,
loc, size=len(index)), index=index))
```




Weight Histogram by Gender
gender
w
m
count
250
200
150
100
50
0
40
50
60
70
80
90
100
Weight

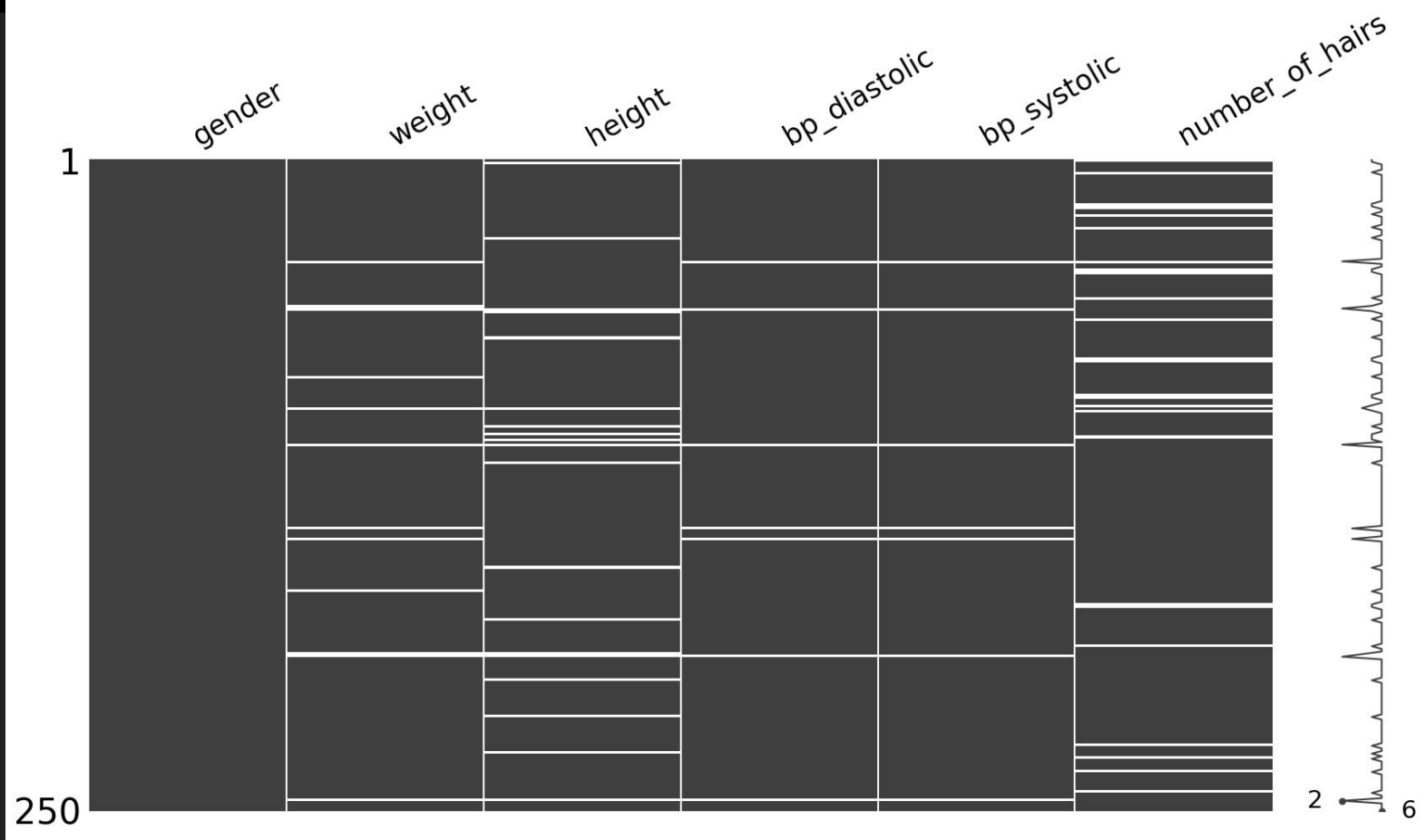
```
df.groupby(['gender', 'missing_weight']).size()
```

```
gender  missing_weight
m       False            4947
        True               53
w       False            4607
        True              393
dtype: int64
```

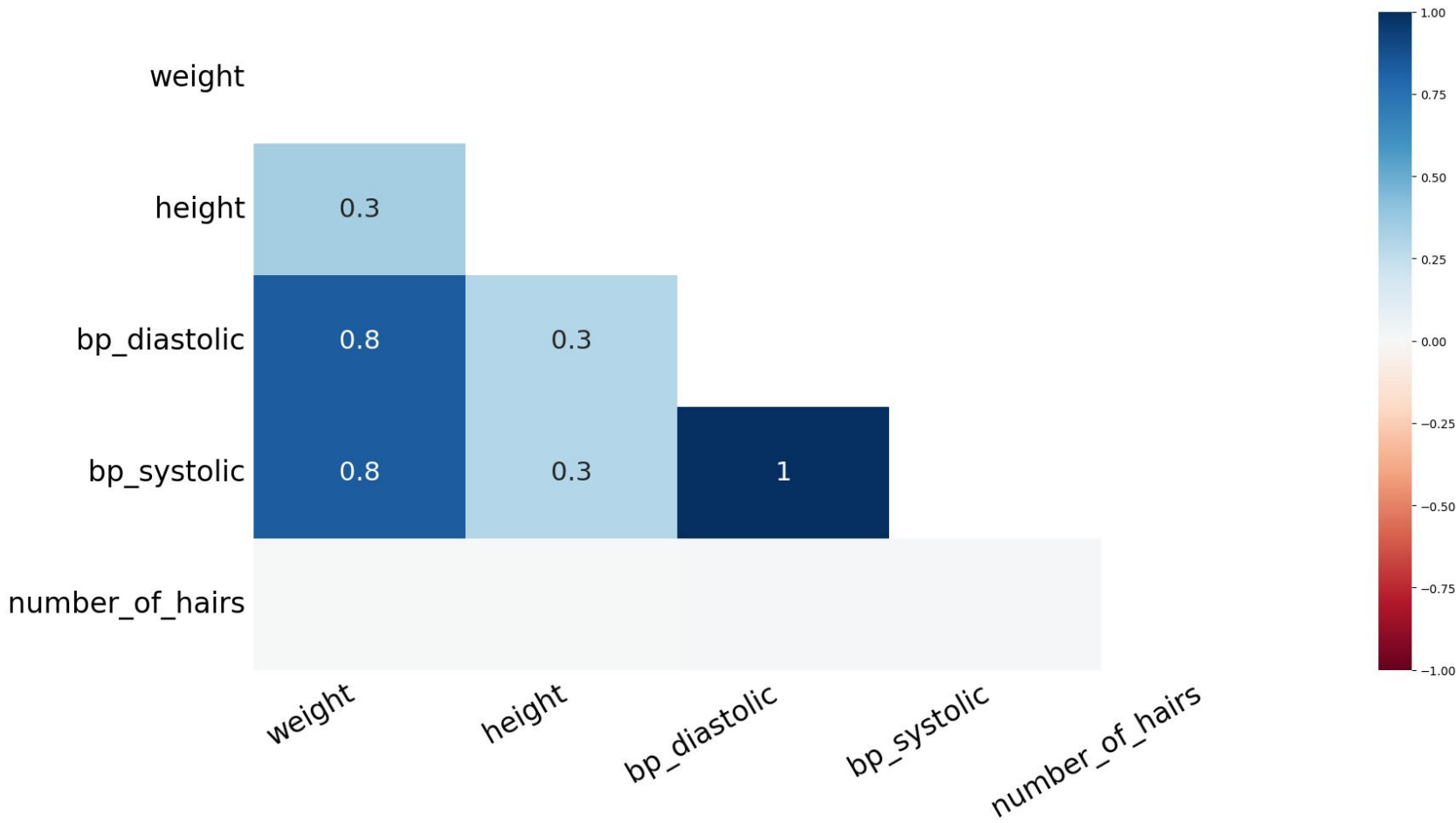

Weight Histogram by Gender
gender
w
m
250
200
150
100
50
0
count
40
50
60
70
80
90
Weight

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   gender           10000 non-null  object 
 1   weight           9554 non-null   float64
 2   height           9316 non-null   float64
 3   bp_diastolic     9689 non-null   float64
 4   bp_systolic      9689 non-null   float64
 5   number_of_hairs  9022 non-null   float64
dtypes: float64(5), object(1)
memory usage: 468.9+ KB
```

```
msno.matrix(df.sample(250))
```

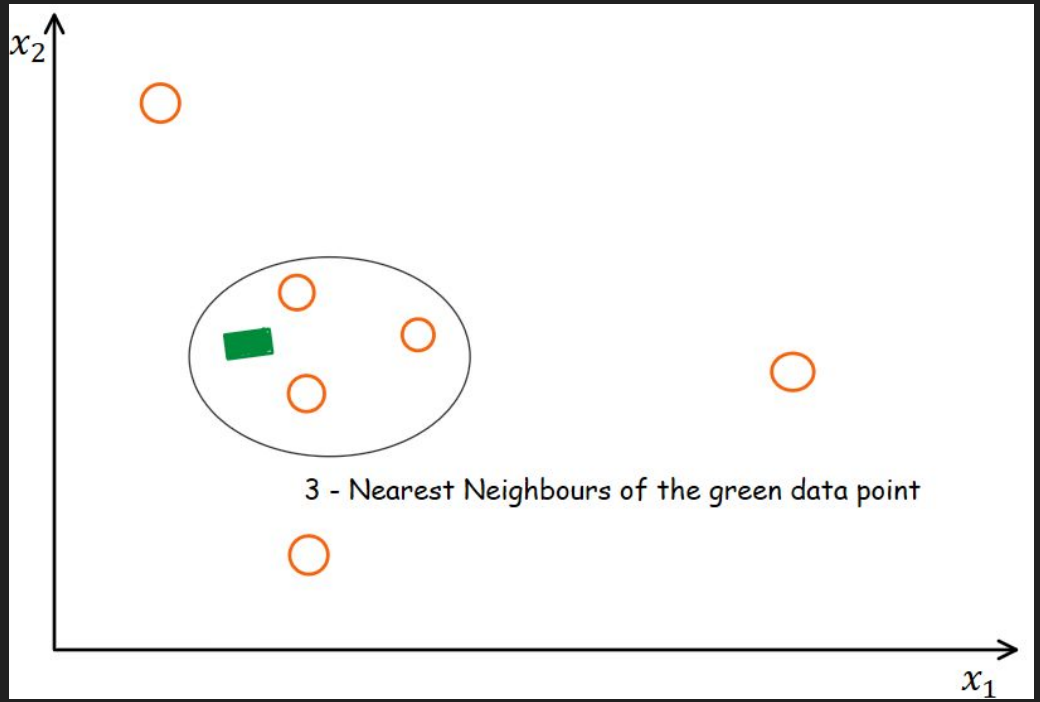


msno.heatmap(df)



KNN imputer

- Mean value of k nearest neighbors



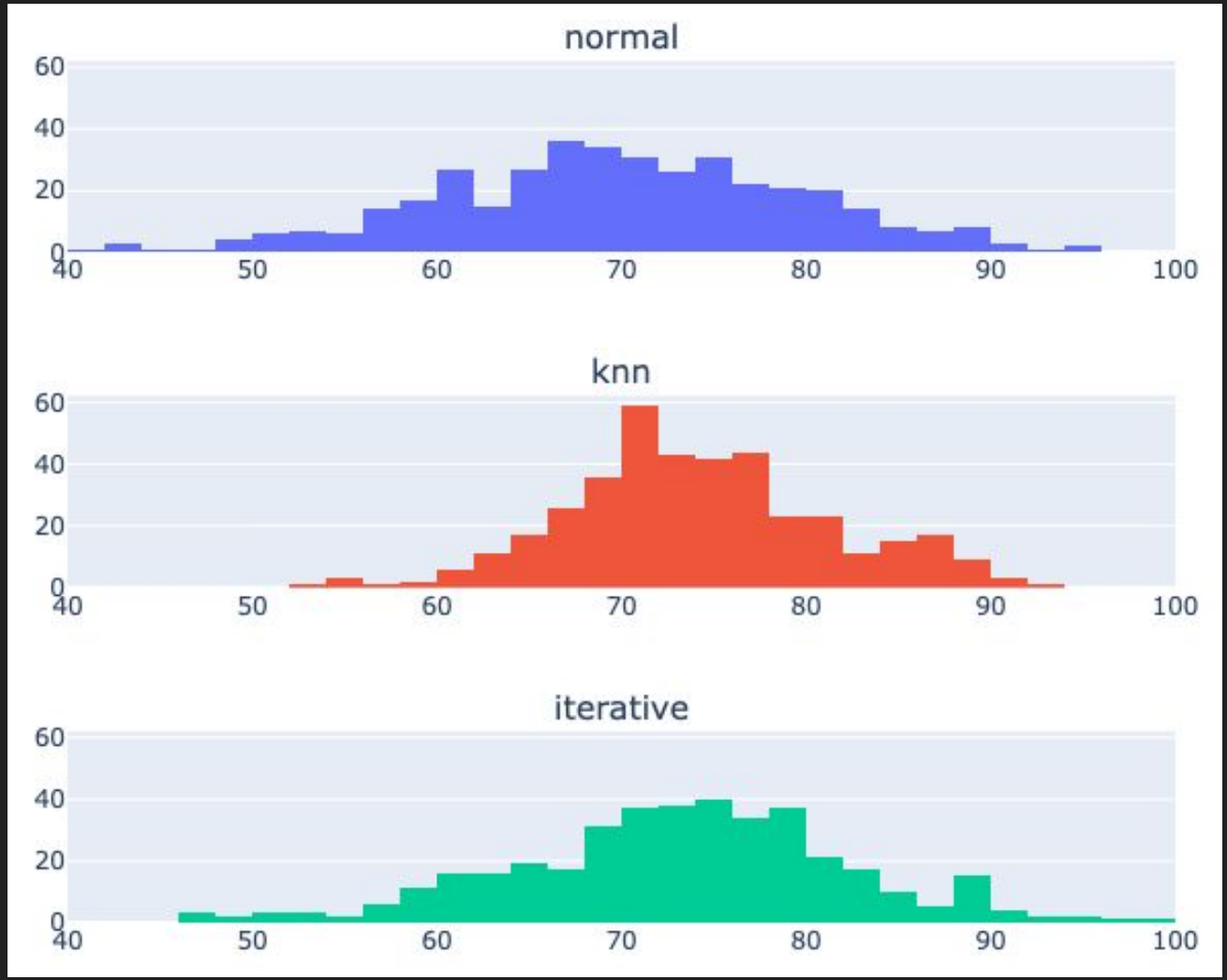
```
from sklearn.impute import KNNImputer

knn_imputer = KNNImputer(n_neighbors=3)

knn_cols = ['weight', 'height', 'number_of_hairs',
'bp_diastolic', 'bp_systolic']

knn_df = pd.DataFrame(knn_imputer.fit_transform(df),
columns=knn_cols)
```

Iterative imputer

- Impute on values on round-robin fashion
- Model each feature as a function of other

- See more here and here

```
from sklearn.experimental import enable_iterative_imputer

from sklearn.impute import IterativeImputer

iterative_imputer = IterativeImputer()

knn_cols = ['weight', 'height', 'number_of_hairs',
'bp_diastolic', 'bp_systolic']

itr_df =
pd.DataFrame(iterative_imputer.fit_transform(df[knn_cols]),
columns=knn_cols)
```


normal
60
40
20
0
40
50
60
70
80
90
100
knn
60
40
20
0
40
50
60
70
80
90
100
iterative
60
40
20
0
40
50
60
70
80
90
100

Summary

- Missing Data is a problem every data scientist and data analyst face
- Data can be missing due to many reasons and can be classified to 3 mechanisms - MCAR, MAR, MNAR
- Who can help you with that? Data Engineer, UX researcher, domain expert

Summary

- Python can help us gain better understanding about missing data and impute values

missingno

Thank you!

Slides and code are available in - here