
Strive
SCHOOL

/ Missing Values

In statistics, missing data, or missing values, occur when no data value is stored for the variable in an observation.

Wikipedia

Missing values

/ Missings can appear in different ways. Detect all of them is crucial:

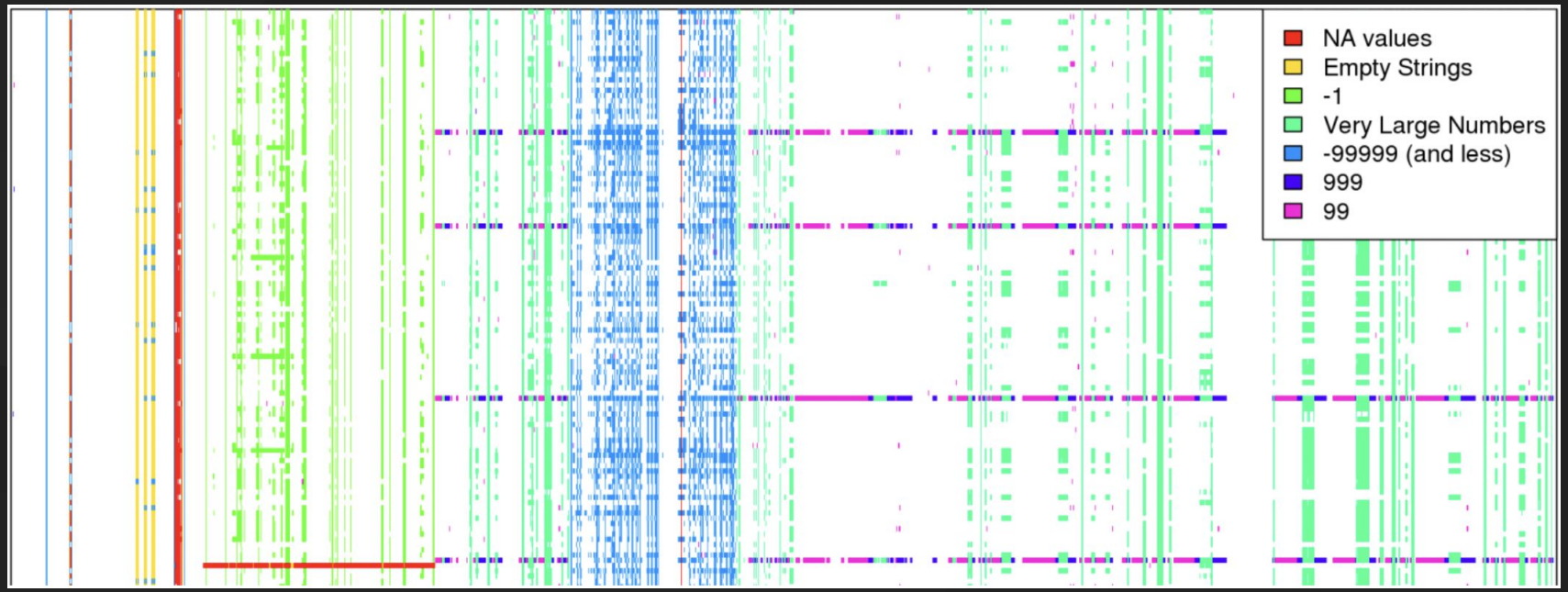
- np.NaN (when CSV file has no data → 2 commas together)
- Empty strings: “”
- Constant
  - -1
  - 0
  - 99
  - 999
  - etc.

Missing values real dataset

/ Train dataset from [Springleaf kaggle competition]

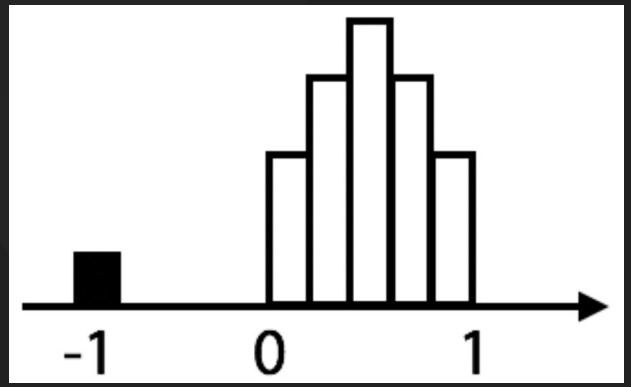
Variables

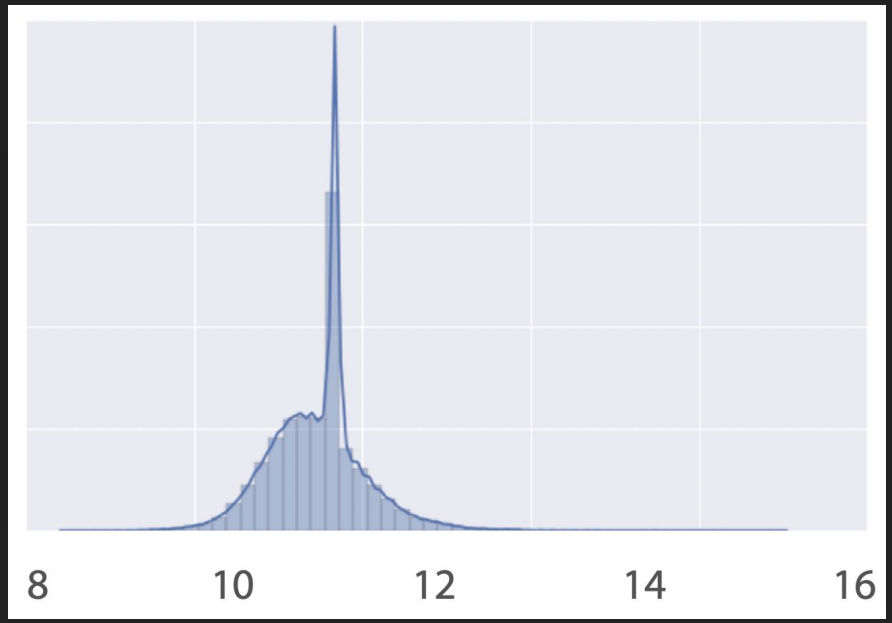
Rows



Hidden Missings

/ Plotting a histogram helps to discover hidden missing values.

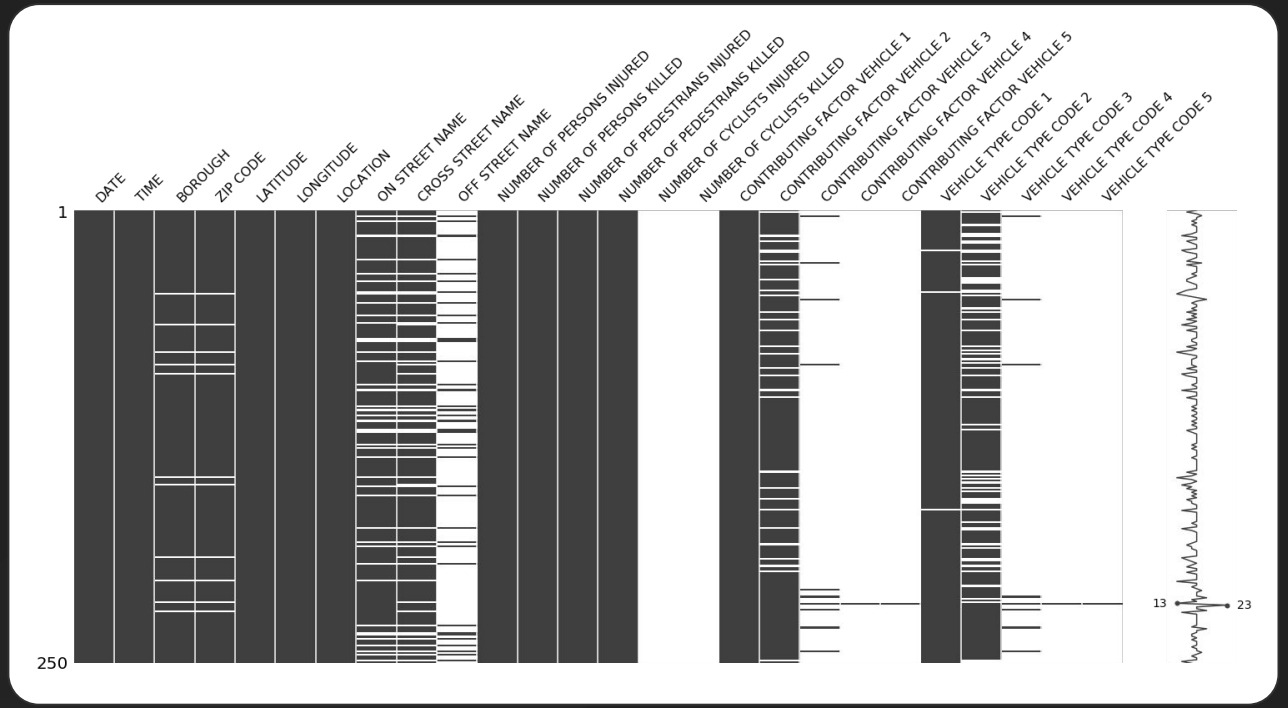




Missings: EDA

/ Once they have been detected you can visualize them with libraries like missingno.

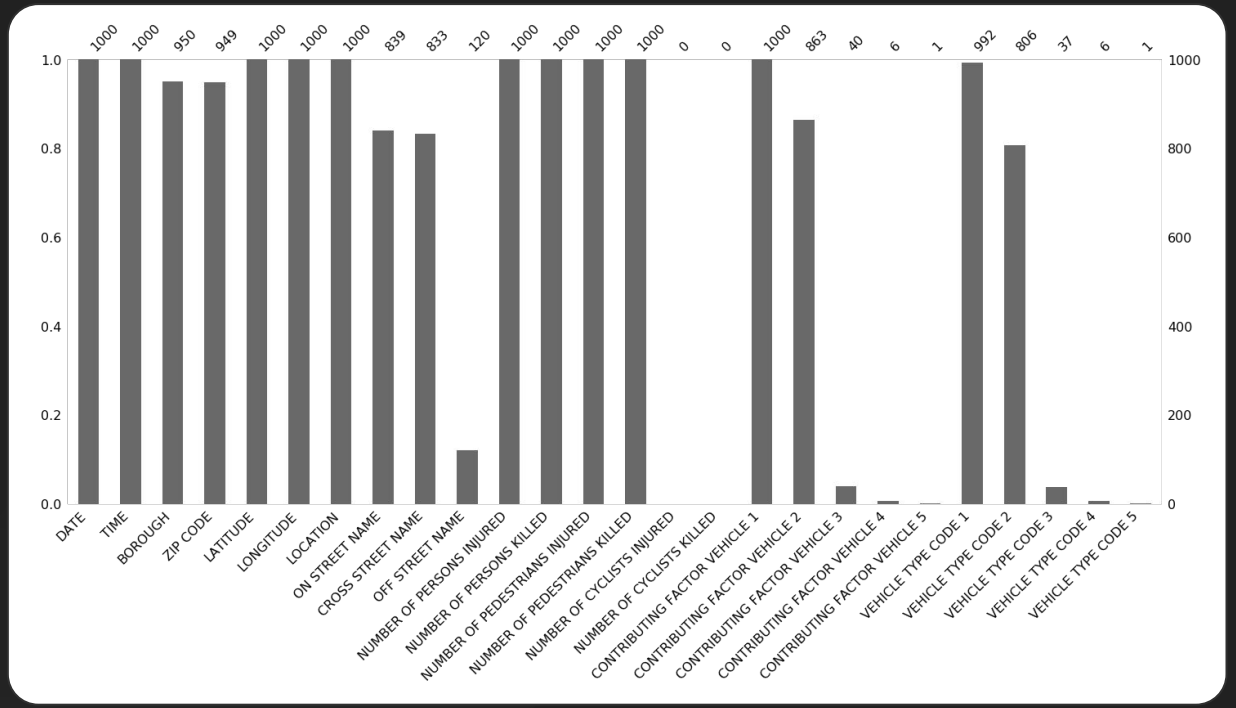
```
import missingno as m
m.matrix(df)
```



Missings: EDA

/ Once they have been detected you can visualize them with libraries like missingno.

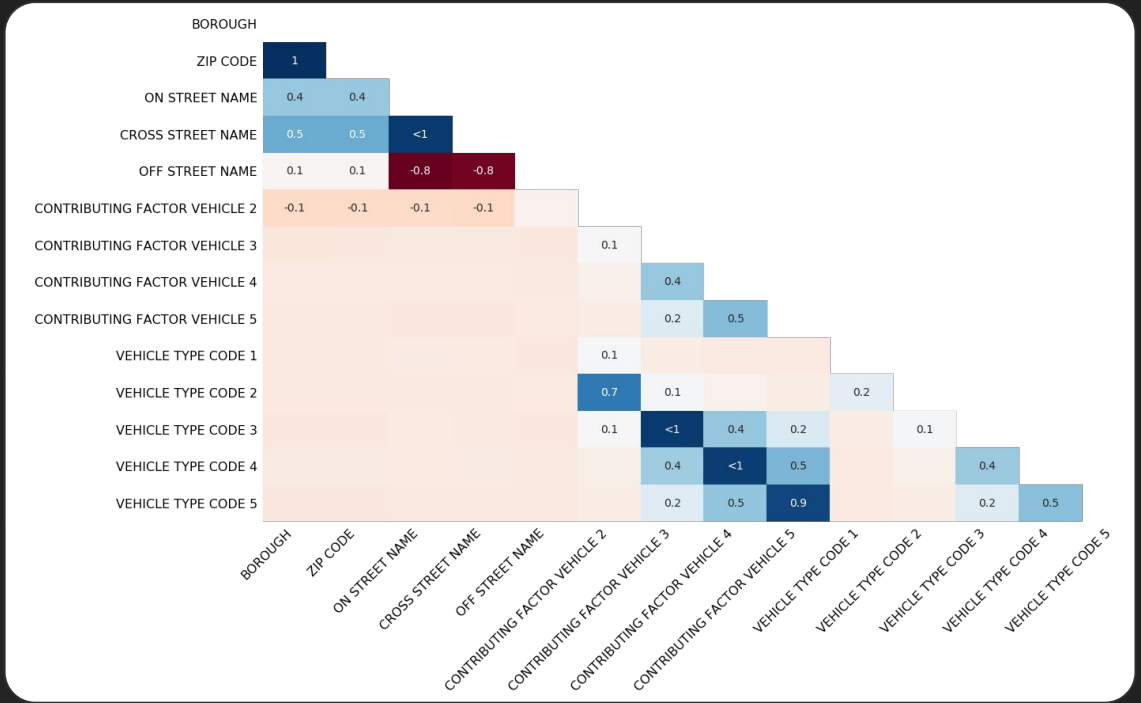
```
import missingno as m
m.bar(df)
```



Missings: EDA

/ Once they have been detected you can visualize them with libraries like missingno.

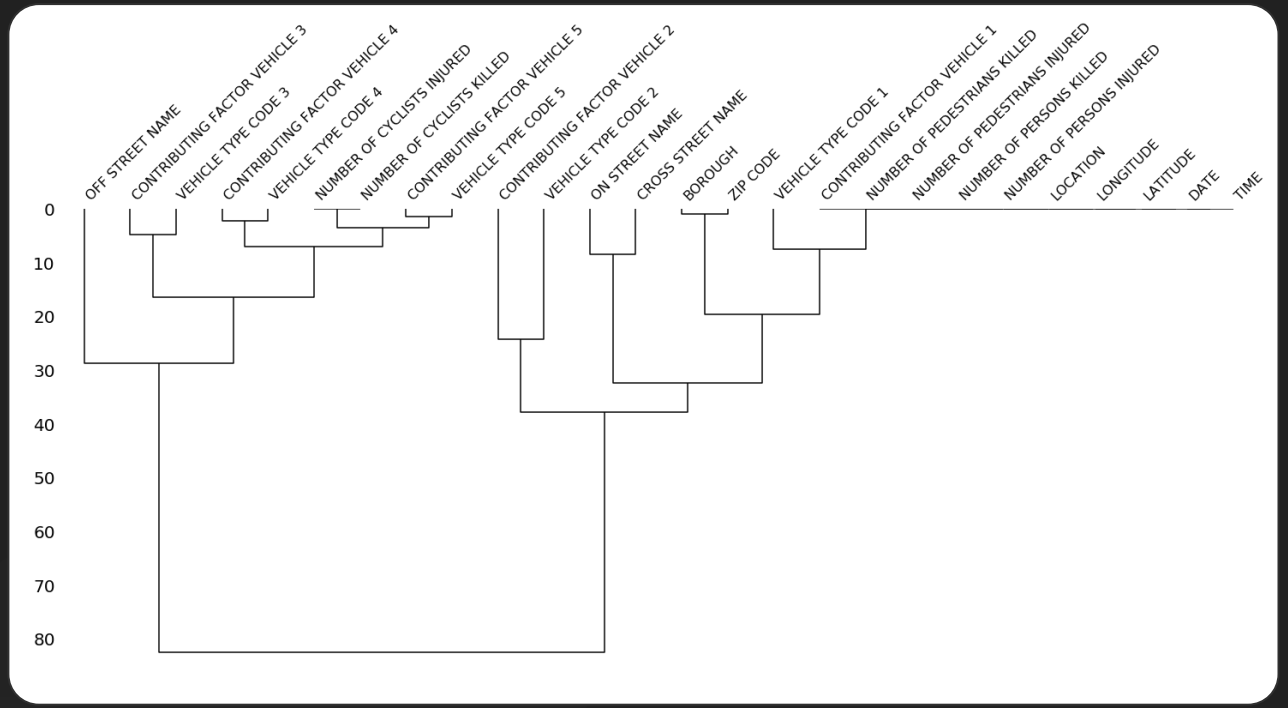
```
import missingno as m
m.heatmap(df)
```



Missings: EDA

/ Once they have been detected you can visualize them with libraries like missingno.

```
import missingno as m
m.dendrogram(df)
```



Handling Missing Values: 3 options

1. Drop them (Not very recommended for important projects)
   - Drop the rows (samples) with missings.
   - Drop the columns (variables) with missings.
2. Imputation: Try to reconstruct the value
   - Univariate Imputation (SimpleImputer in Sklearn) + Add Missing Indicator
     - Constant: For numerical (-999) and categorical (“missing”)
     - Mean: For numerical only
     - Median: For numerical only
     - Most Frequent value: For categorical only
   - Multivariate imputation: We use other variables
     - Grouping by other vars and get the means or medians (In Pandas)
     - Train a ML model to inference the NaNs (IterativeImputer in Sklearn)
3. Use a model that handles missings
   - XGBoost
   - LightGBM

Advice

Develop an intuition. You should not automatically drop or impute missings without thinking about them. You need to be curious and ask yourself: Is this value missing because it wasn't recorded or because it doesn't exist?

If a value is missing because it doesn't exist (like the height of the oldest child of someone who doesn't have any children) then it doesn't make sense to try and guess what it might be. These values you probably do want to keep as NaN.

On the other hand, if a value is missing because it wasn't recorded, then you can try to guess what it might have been based on the other values in that column and row. This is called imputation.

Univariate Imputation: SimpleImputer()

/ Strategy:

- Constant: Replace with a value out of the feature value range.
  - Numerical Feat: -999, -1 (Good for trees models)
  - Categorical Feat: “Missing_value”
- Mean
- Median
- Most Frequent

feature	isnull
0.1	False
0.95	False
NaN	True
-3	False
NaN	True

/ Adding a new column called isNull helps to indicate that the value was imputed.

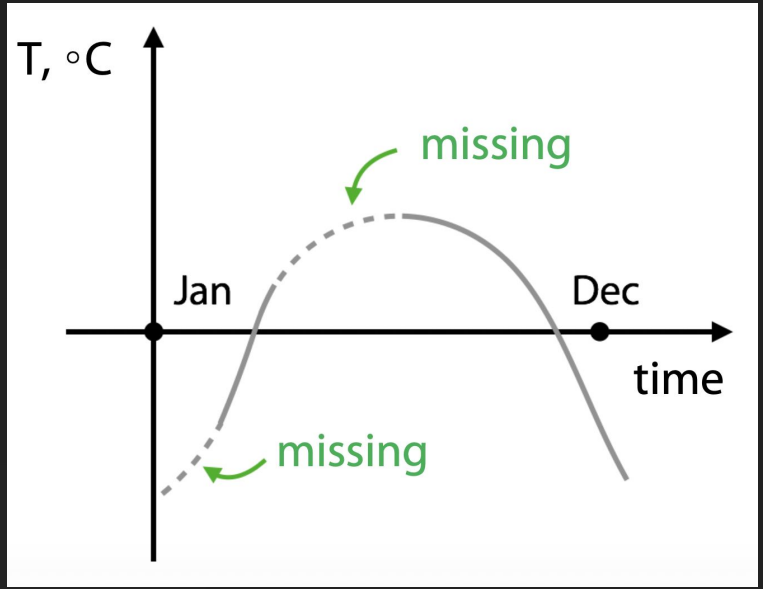
Multivariate Imputation

/ When we try to reconstruct the missing. We make use of other variables (specially the time) to fill the missing.

/ In sklearn we have 2 options:

- impute.IterativeImputer()
- impute.KNNImputer()

An interpolation is also a good idea:



Conclusion

- Missing values can be hidden (replaced with something)
- The choice of method to fill NaN depends on the situation
- Usual way to deal with missing values is to replace them with -999, mean or median
- Binary feature “isnull” can be beneficial
- In general, avoid filling NaNs before feature generation
- Xgboost and LightGBM can handle NaN

/ Q&A

What are your doubts?


Strive
SCHOOL