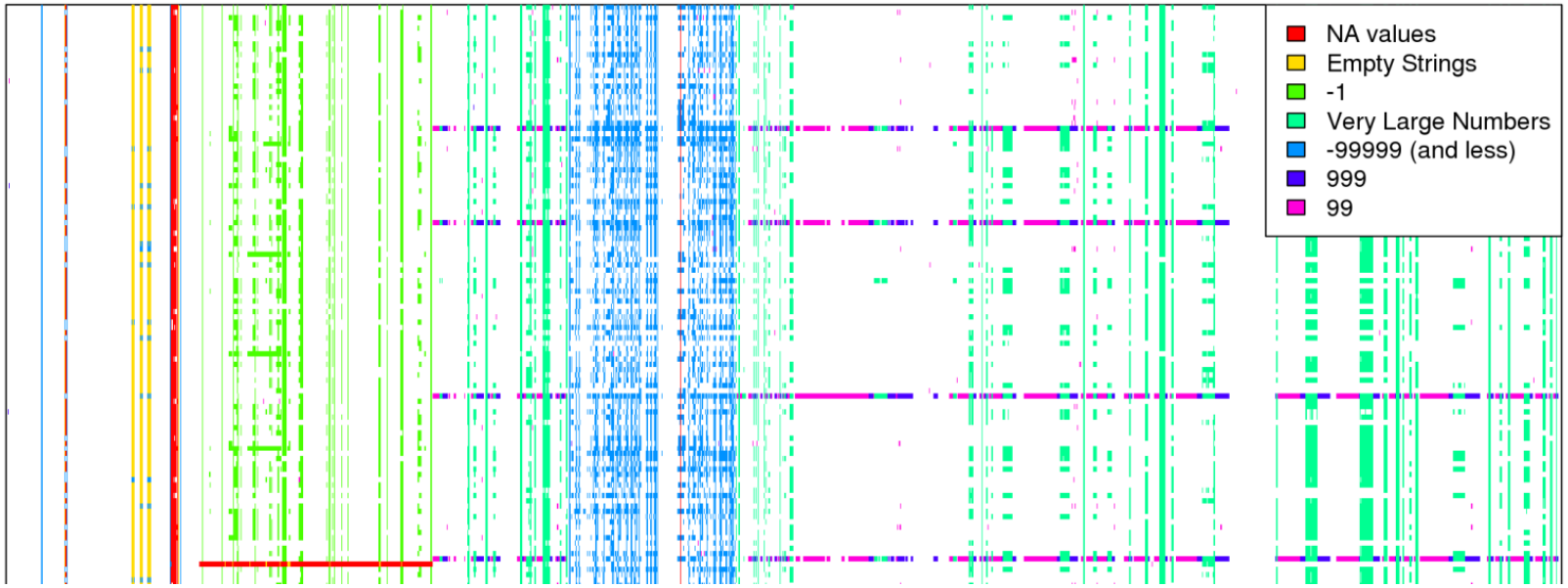
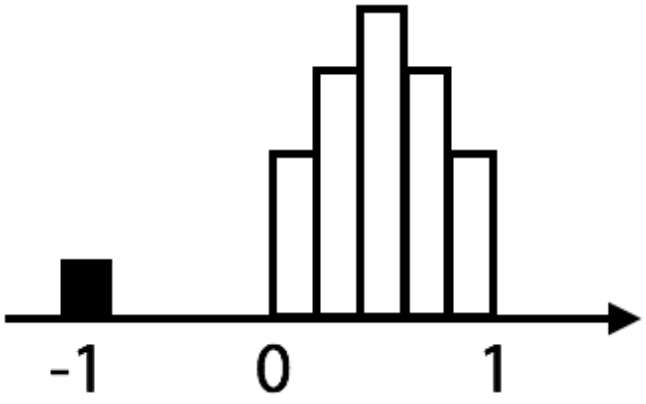
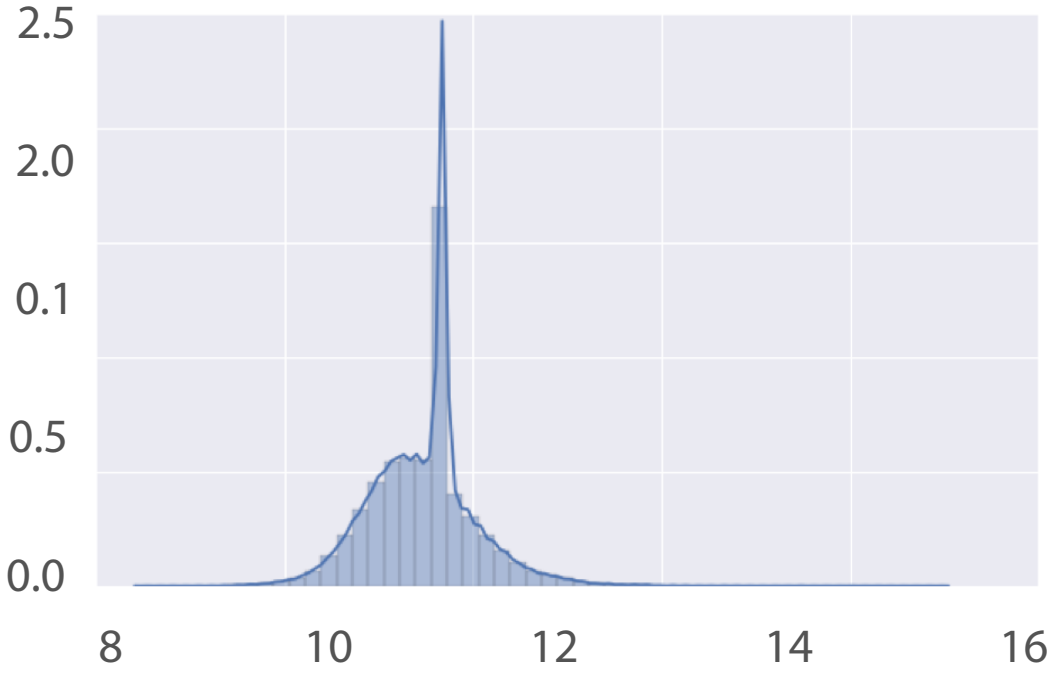
# Missing values

# Missing data, numeric



# Hidden NaNs



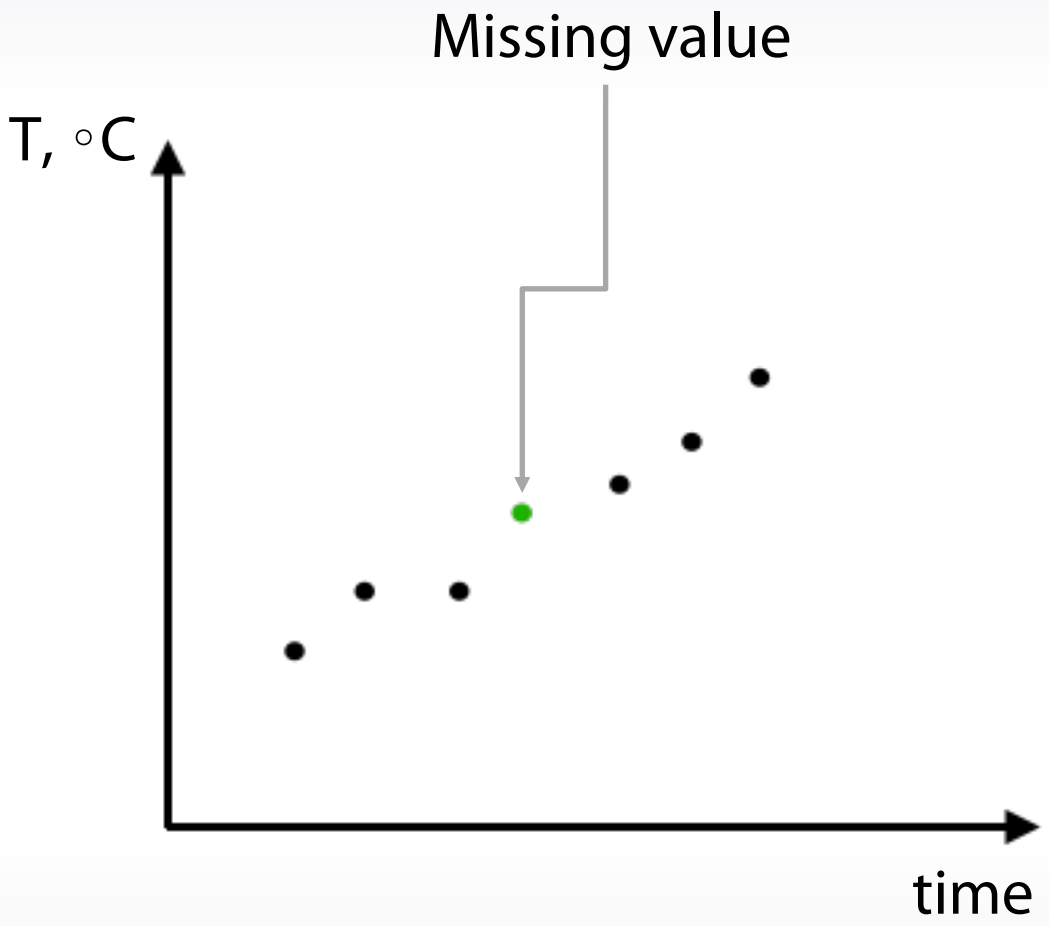
# Fillna approaches

1. -999, -1, etc
2. mean, median
3. Reconstruct value

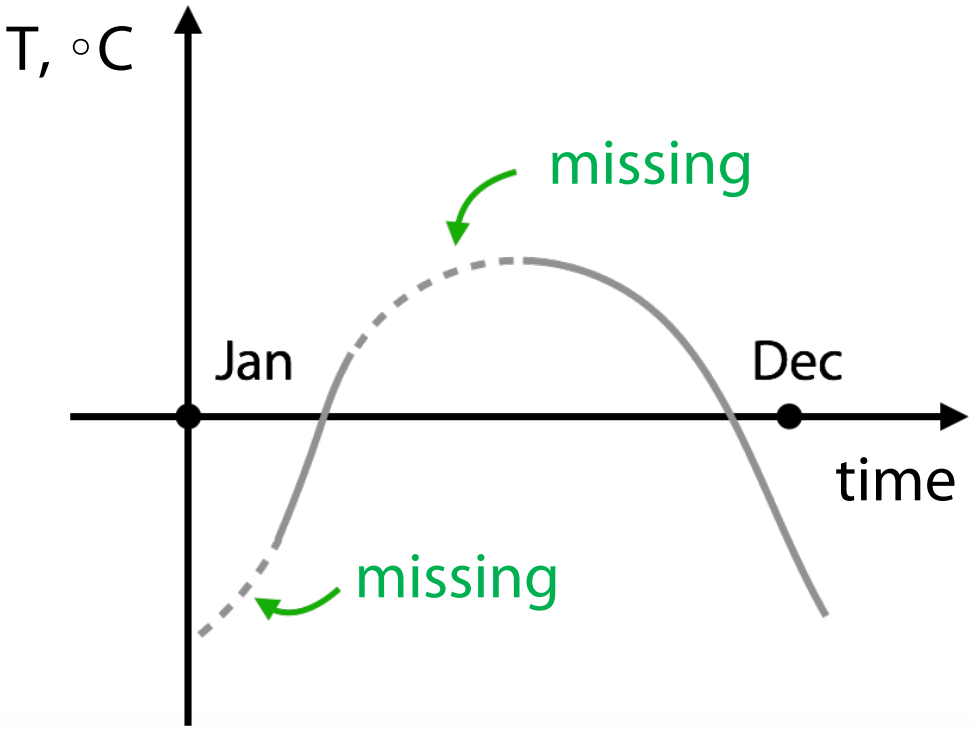
# “Isnull” feature

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

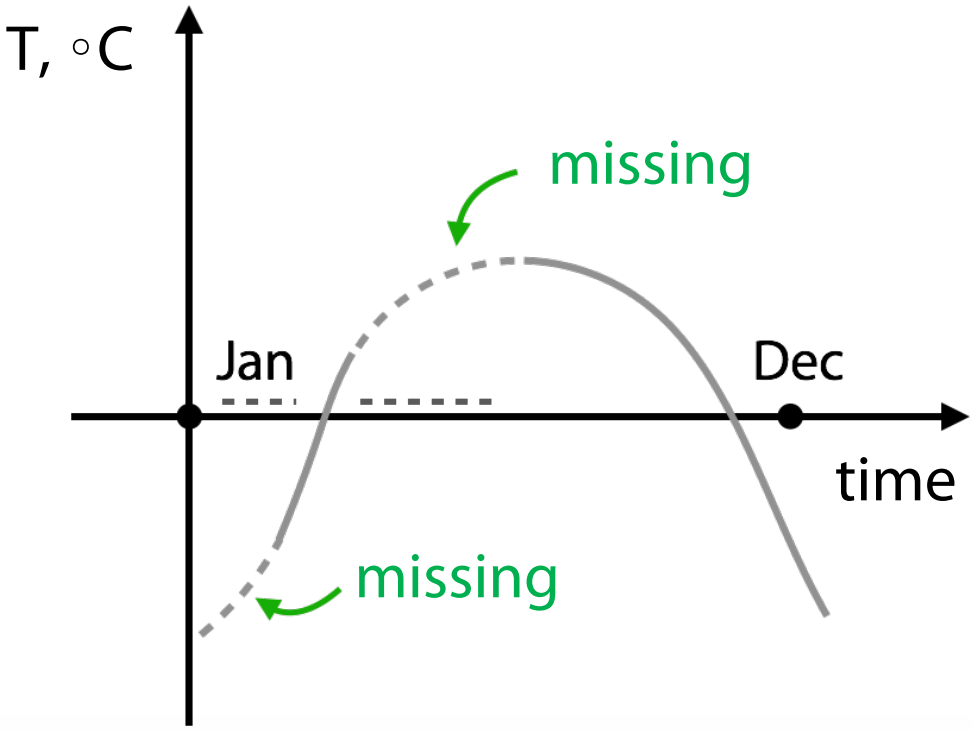
# Missing values reconstruction



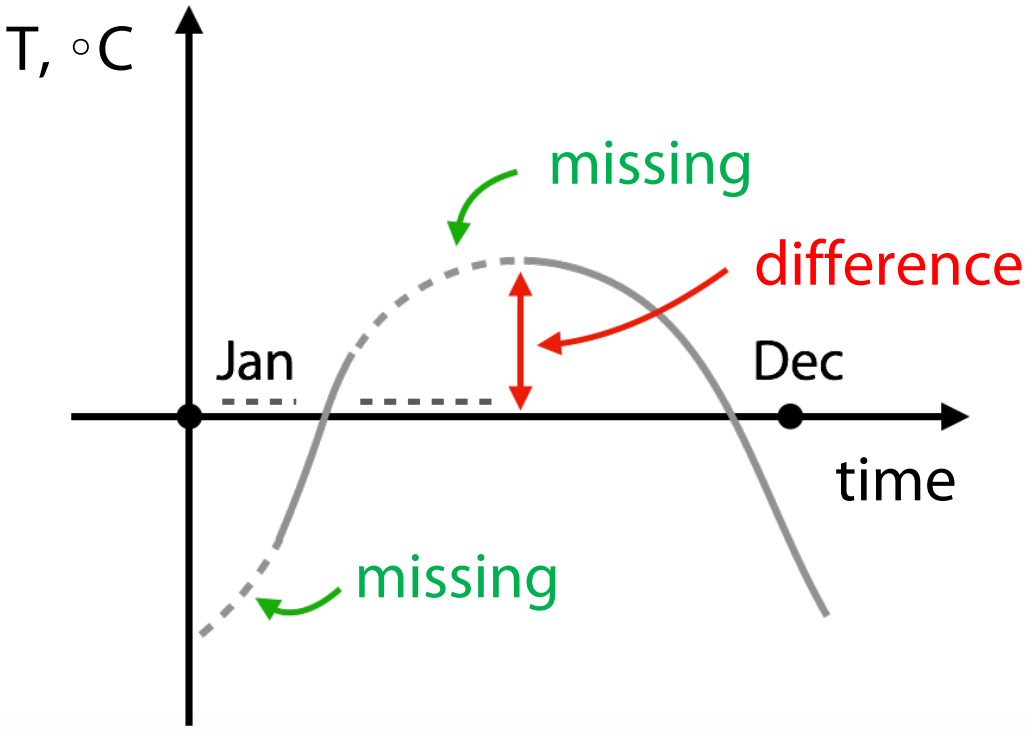
# Feature generation with missing values



# Feature generation with missing values



# Feature generation with missing values



# Feature generation with missing values

| categorical_ feature | numeric _feature |
|---|---|
| A | 1 |
| A | 4 |
| A | 2 |
| A | -1 |
| B | 9 |
| B | NaN |

# Feature generation with missing values

| categorical_ feature | numeric _feature | numeric_ feature_filled |
|---|---|---|
| A | 1 | 1 |
| A | 4 | 4 |
| A | 2 | 2 |
| A | -1 | -1 |
| B | 9 | 9 |
| B | NaN | -999 |

# Feature generation with missing values

You should be very careful with early none importation if you want to generate new features. There's one more interesting thing about missing values.
6:43
[INAUDIBLE] boost can handle a lot of numbers and sometimes using this approach can change score drastically.
6:51
Besides common approaches we have discussed, sometimes we can treat outliers as missing values.

| categorical_feature | numeric_feature | numeric_feature_filled | categorical_encoded |
|---|---|---|---|
| A | 1 | 1 | 1.5 |
| A | 4 | 4 | 1.5 |
| A | 2 | 2 | 1.5 |
| A | -1 | -1 | 1.5 |
| B | 9 | 9 | -495 |
| B | NaN | -999 | -495 |

# Treating values which do not present in train data

Train:

| categorical_feature | target |
|---|---|
| A | 0 |
| A | 1 |
| A | 1 |
| A | 1 |
| B | 0 |
| B | 0 |
| D | 1 |

Test:

| categorical_feature | target |
|---|---|
| A | ? |
| A | ? |
| B | ? |
| C | ? |

If you have categorical features, sometimes it can be beneficial to change the missing values or categories which present in the test data but do not present in the train data. The intention for doing so appeals to the fact that the model which didn't have that category in the train data will eventually treat it randomly.

# Treating values which do not present in train data

Train:

| categorical _feature | categorical _encoded | target |
|---|---|---|
| A | 6 | 0 |
| A | 6 | 1 |
| A | 6 | 1 |
| A | 6 | 1 |
| B | 3 | 0 |
| B | 3 | 0 |
| D | 1 | 1 |

Test:

| categorical _feature | categorical _encoded | target |
|---|---|---|
| A | 6 | ? |
| A | 6 | ? |
| B | 3 | ? |
| C | 1 | ? |

# Treating values which do not present in train data

1. The choice of method to fill NaN depends on the situation
2. Usual way to deal with missing values is to replace them with -999, mean or median
3. Missing values already can be replaced with something by organizers
4. Binary feature “isnull” can be beneficial
5. In general, avoid filling nans before feature generation
6. Xgboost can handle NaN