Some heuristics for **missingness**

Quinn Dougherty
Lambda School

About me

Quinn Dougherty

Philadelphia

Section Lead - Lambda School

Table of contents

Assumptions

I'm assuming you know what this means 👉

```
(titanic_.isna().sum() / titanic_.shape[0])
```

```
missing rate by feature:
PassengerId      0.0 %
Survived         0.0 %
Pclass           0.0 %
Name             0.0 %
Sex              0.0 %
Age            19.87 %
SibSp            0.0 %
Parch            0.0 %
Ticket           0.0 %
Fare             0.0 %
Cabin           77.1 %
Embarked      0.2245 %
dtype: object
```

I'm assuming you've filled

with column-mean at least once

I'm assuming some Pandas syntax

What the talk is not

- Advanced imputation workshop
- Intermediate imputation workshop

What the talk is

- Overview of strategies that might point you toward wise experimentation

Imputation

*In statistics, **imputation** is the process of replacing missing data with substituted values.*

- Wikipedia

```
titanic_.fillna('mean')
```

Missingness regimes

We classify missingness into three different *regimes*

1. Missingness of a feature is a function of itself
2. Missingness of a feature is a function of all other features
3. Missingness of a feature is "truly random"

Missingness regimes - technical terms

We classify missingness into three different *regimes*

1. Missingness of a feature is a function of itself
    a. *MNAR: Missing Not At Random*
2. Missingness of a feature is a function of other features
    a. *MAR: Missing At Random*
3. Missingness of a feature is “truly random”
    a. *MCAR: Missing Completely At Random*

Missingness regimes - how do I know?

We classify missingness into three different *regimes*

1. Missingness of a feature is a function of itself
    a. *MNAR: Missing Not At Random*
    b. Domain knowledge >> statistical tests
2. Missingness of a feature is a function of other features
    a. *MAR: Missing At Random*
    b. Domain knowledge >> statistical tests
3. Missingness of a feature is "truly random"
    a. *MCAR: Missing Completely At Random*
    b. Domain knowledge >> statistical tests

MNAR: Missing Not at Random

Missingness of a feature is a function of itself

- Domain knowledge, data collection methodology
- Absent a common-sensical insight, you can't prove MNAR, but you can show that it's not MAR or MCAR (Eekhout, 2014).

- *Example:* Someone is collecting height data by having people write down their height on a piece of paper and dropping it in a box, which is 9 feet above the ground

MAR: Missing at Random

Missingness of a feature is a function of other features.

find some `f_i := (X_k | k in 1..n) ↦ filled(X_i)`

- *Example:* Someone magically has height data already, and they're collecting movie preference data by having people write down their preferences on a piece of paper and dropping it in a box, which is 9 feet above the ground

MAR: Missing at Random

Missingness of a feature is a function of other features.

- Observe correlations of the indicator matrix - strong correlations is one reason to suspect MAR

```
titanic_.isna().corr()
```

- You can also target a feature's indicator function in a logistic regression, strong coefficients suggest MAR.

```
df.Xi.isna() ≅ sigmoid(X β)
```

MAR: Missing at Random -- ok, then what?

Missingness of a feature is a function of other features.

MAR suggests that you should try **multivariate imputation**

- Python: `sklearn.impute.IterativeImputer` (experimental)
- R: `MICE` (Multivariate Imputation by Chained Equations)

But again-- experiment and commonsense prevail over heuristics

MCAR: Missing Completely at Random

Missingness of a feature is “truly random”

It’s seductive to fall back on this, but not as likely as you think (Niederhut, 2018)

- *Example*: A goblin has set upon stealing your data, because it amuses him and gets on your nerves

MCAR: Missing Completely at Random -- a brief study

Missingness of a feature is “truly random”

We can easily simulate this case with a bernoulli variable

```
def mcar_goblin(dat: DataFrame, ratio: float) -> DataFrame:
    ''' Simulate MCAR with bernoulli '''
    def ident_or_nan(x: float) -> float:
        ''' if heads, replace value with nan. if tails, identity '''
        coin = bernoulli(ratio)
        if coin.rvs()==1:
            return nan
        else:
            return x

    return dat.assign(**{feat: [ident_or_nan(x)
                                for x in dat[feat].values]
                          for feat in dat.columns if feat!='y'})
```

MCAR: Missing Completely at Random -- a brief study

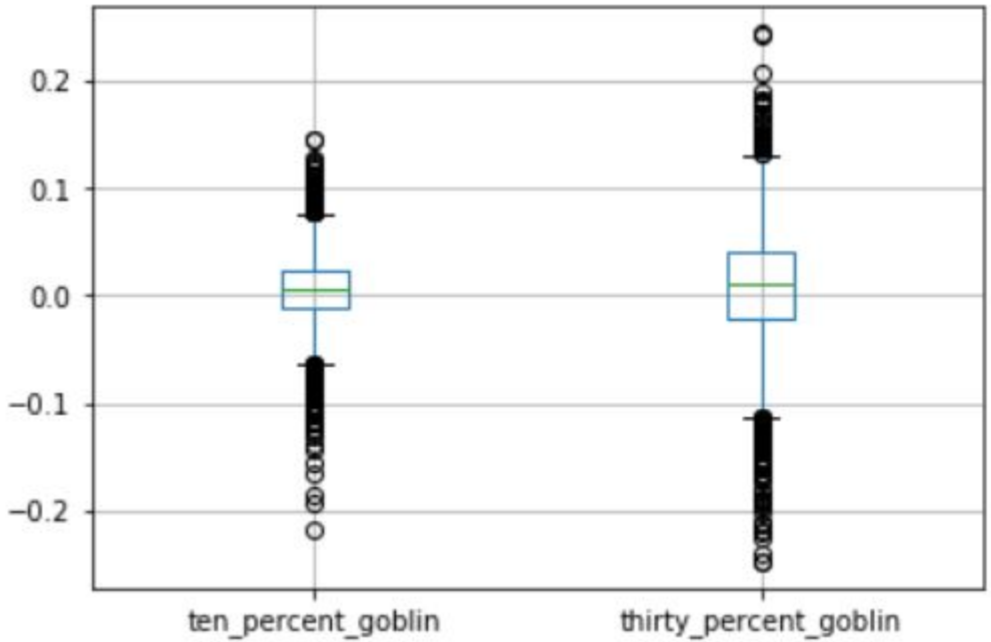
Missingness of a feature is “truly random”

Train a model on complete data and then train a pipeline with an imputer on data that the goblin has interfered with.

Observe the *differences* in coefficients to see the **bias** of imputation and missingness, over several imputation methods.

Dataset is DrivenData’s Tanzanian Waterpoints competition

MCAR: Missing Completely at Random -- a brief study



Summary

Imputation is finding values to fill in missingness with

It introduces bias

There are different ways it can be missing (MNAR, MAR, and MCAR)

Experiment, commonsense, and domain knowledge >> statistical tests

Sources

https://github.com/deniederhut/safe-handling-instructions-for-missing-data

https://www.iriseekhout.com/missing-data/

https://pypi.org/project/fancyimpute/

Thank you

Quinn Dougherty
quinndougherty92@gmail.com
/in/quinn-dougherty/