STAT5003

Week 7 : Bootstrapping, missing data, and class imbalance

Jaslene Lin
The University of Sydney

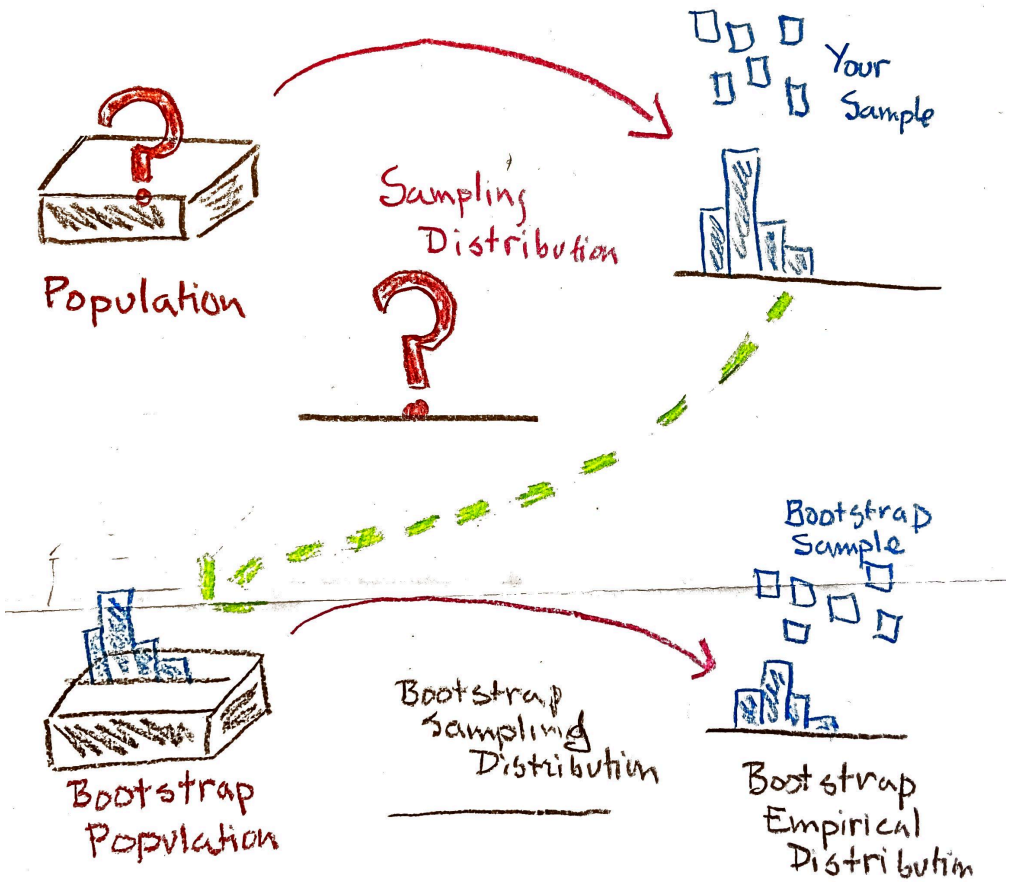
THE UNIVERSITY OF SYDNEY

Readings and R functions covered

Important

- **Introduction to Statistical Learning**
  - Bootstrap covered in Chapter 5
- **R** functions
  - `naniar:: any_na`
  - `naniar:: are_na`
  - `visdat:: vis_dat`
  - `vidat:: vis_miss`
  - `mice::md.pattern`
  - `mice::marginplot`
  - `ROSE::ovun.sample`
  - `themis::step_smote`

Bootstrap



Source

Bootstrap

- The bootstrap is a flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method.
- For example, it can provide an estimate of the standard error of a coefficient, or a confidence interval for that coefficient.

Simple example

- Suppose that we wish to invest a fixed sum of money in two financial assets that yield returns of X and Y where X and Y are random quantities
- The goal is to create a portfolio by investing fraction α of our wealth in X and $(1-\alpha)$ in Y
- Want to choose to minimise the total risk of the investment. Mathematically, this involves minimising the variance of our investment portfolio $\mathbf{Var}(\alpha X + (1-\alpha)Y)$
- The solution to this problem (*via calculus*) is

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}} \nonumber \tag{1}$$

  - $\sigma_X^2 = \mathbf{Var}(X)$, $\sigma_Y^2 = \mathbf{Var}(Y)$, and $\sigma_{XY} = \mathbf{Cov}(X, Y)$

A Thought Experiment

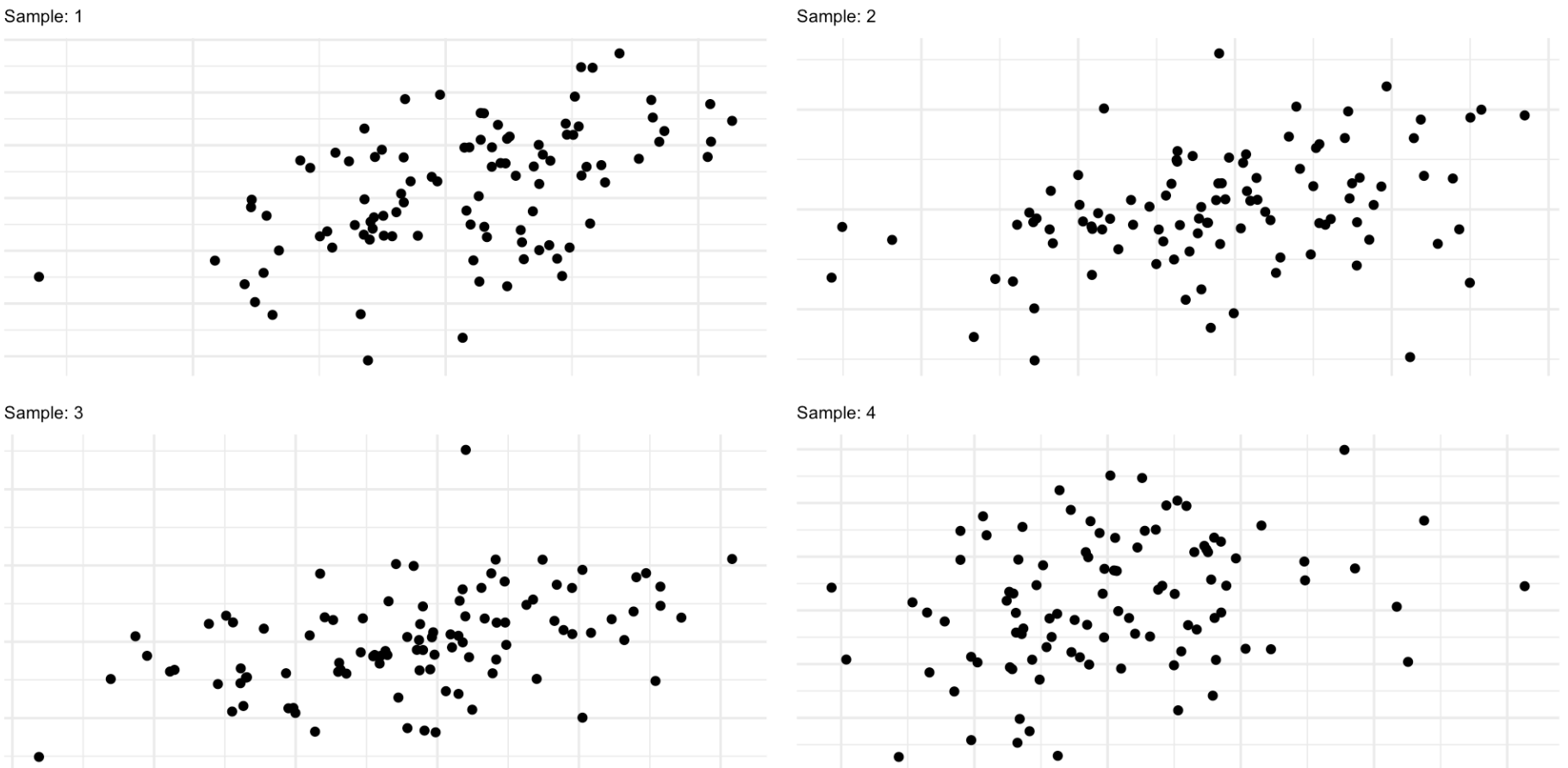
- The values of σ_X^2, σ_Y^2, and σ_{XY} are unknown but estimates can be computed from the data
- The estimate of α that minimises the variance of the investment can then be computed with

$$\widehat{\alpha} = \frac{\widehat{\sigma}_Y^2 - \widehat{\sigma}_{XY}}{\widehat{\sigma}_X^2 + \widehat{\sigma}_Y^2 - 2\widehat{\sigma}_{XY}} \backslash \text{nonumber} \tag{2}$$

- Suppose that X and Y can be sampled from the population repeatedly
- You sample **100 pairs of** (X, Y) from the population — this gives you one dataset.
- You calculate an estimate, say $\widehat{\alpha}$, from this sample.
- Now, repeat this **sampling process 1000 times**, each time drawing a new random sample of 100 pairs from the population and computing $\widehat{\alpha}$.
- You will end up with $\widehat{\alpha}_1, \widehat{\alpha}_2, \ldots, \widehat{\alpha}_{1000}$

Simulations in R

- Consider a population with $\sigma_X^2 = 1$, $\sigma_Y^2 = 1.5$, and $\sigma_{XY} = 0.5 \Rightarrow \alpha = 2/3$.



From left to right, top to bottom, the estimates for α are 0.659, 0.683, 0.726, 0.68.

Sampling Variability and the Variance of an Estimator

> **Note**
>
> **Key Idea**: When we estimate a parameter, the result depends on the sample. If we collecte a different sample, the estimate would likely be different.

- This phenomenon is called **sampling variability**.
- The **variance of an estimator** captures this variability:
  - If an estimator has **low variance**, repeated samples yield similar results.
  - If it has **high variance**, the estimate fluctuates a lot between samples.
- Ideally, we want estimators that are:
  - **Unbiased** (on average, close to the true value)
  - **Low variance** (stable across samples)
- Estimating the variance of an estimator requires repeatedly sample from the population.

Esmating the Variability of $\widehat{\alpha}$

- Consider the mean of all the parameter estimates

$$\overline{\widehat{\alpha}} = \frac{1}{1000}\sum_{k=1}^{1000}\widehat{\alpha}_k = 0.6662595$$

  - ➟ This is close to the true value of 0.6666667
- Estimate of the standard error using the standard deviation of all the estimates

$$\sqrt{\frac{1}{1000-1}\sum_{k=1}^{1000}\left(\widehat{\alpha}_k - \overline{\widehat{\alpha}}\right)^2} = 0.0760217$$

- This gives an intuitive description of the reliability of the estimator
  - ➟ For a random sample the estimate would vary around the true value by 0.0760217

Application in Reality

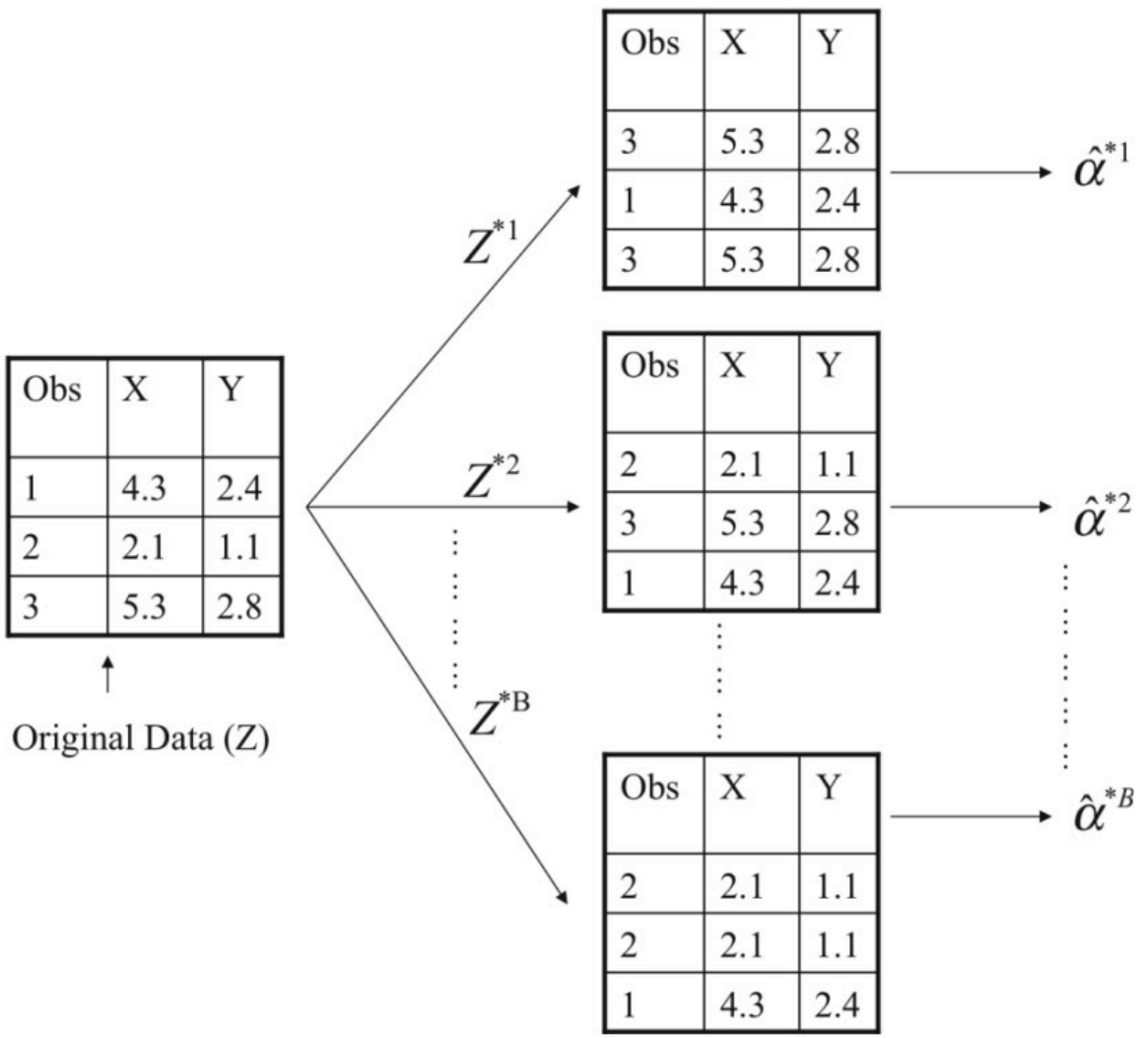
- We can't actually sample repeatedly from the population
  - ➟ In reality, we **only have one dataset**, not the entire population to draw from.

Bootstrap attempts to mimic this process

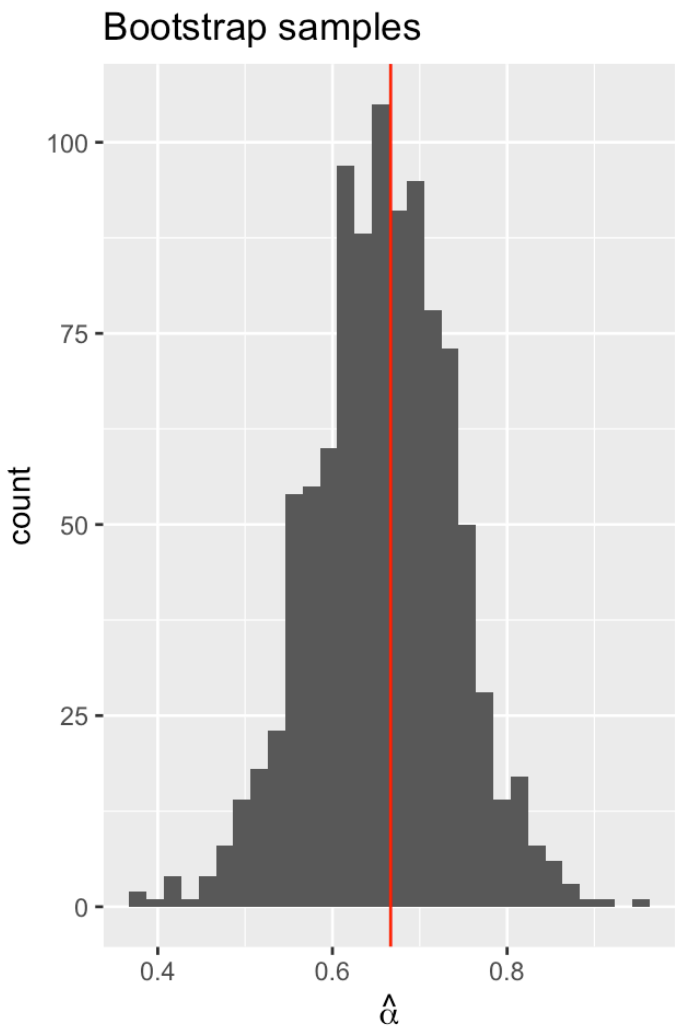
- Instead of sampling new independent observations from the population
  - ➟ Re-sample observations from the data *with replacement*
  - ➟ Some observations may appear more than once while others not at all

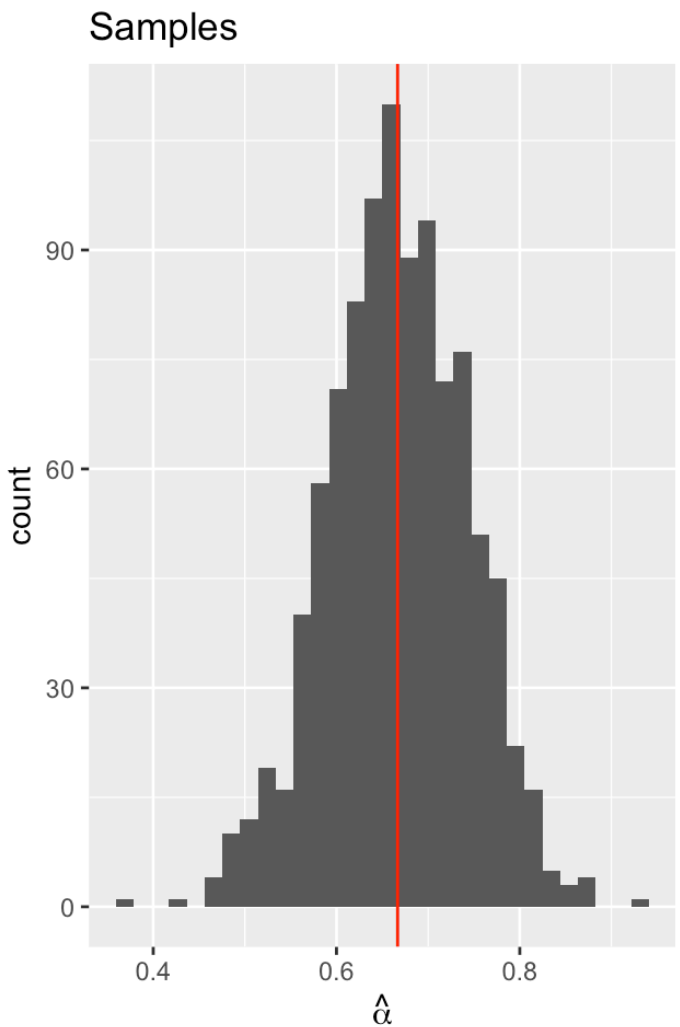
Bootstrap resampling algorithm

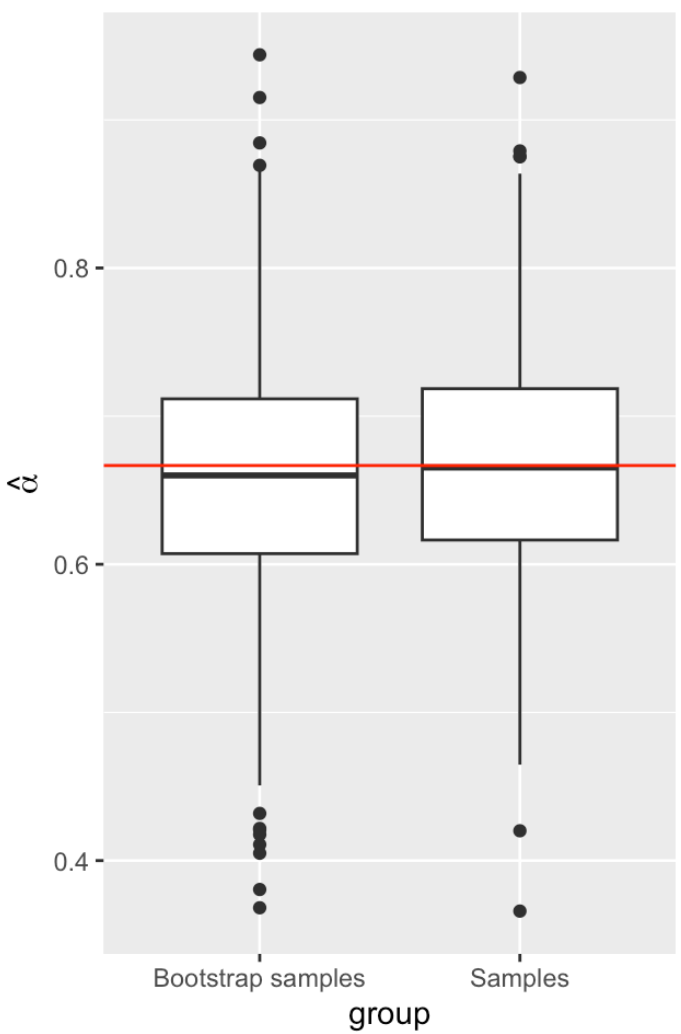
- Essentially sampling with replacement from the original sample



Results bootstrap vs population







Missing data

```
The best thing to do with missing data is to not have any.

    --- Gertrude Mary Cox (statistician)
```

- Working with real-world data = working with missing data

Any Missing Data?

```
1 library(naniar)
2
3 glimpse(airquality)
```

```
Rows: 153
Columns: 6
$ Ozone   <int> 41, 36, 12, 18, NA, 28, 23, 19, 8, NA, 7, 16, 11, 14, 18, 14, …
$ Solar.R <int> 190, 118, 149, 313, NA, NA, 299, 99, 19, 194, NA, 256, 290, 27…
$ Wind    <dbl> 7.4, 8.0, 12.6, 11.5, 14.3, 14.9, 8.6, 13.8, 20.1, 8.6, 6.9, 9…
$ Temp    <int> 67, 72, 74, 62, 56, 66, 65, 59, 61, 69, 74, 69, 66, 68, 58, 64…
$ Month   <int> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,…
$ Day     <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,…
```

```
1 # str(airquality)
2
3 any_na(airquality)
```

```
[1] TRUE
```

```
1 head(are_na(airquality$Solar.R))
```

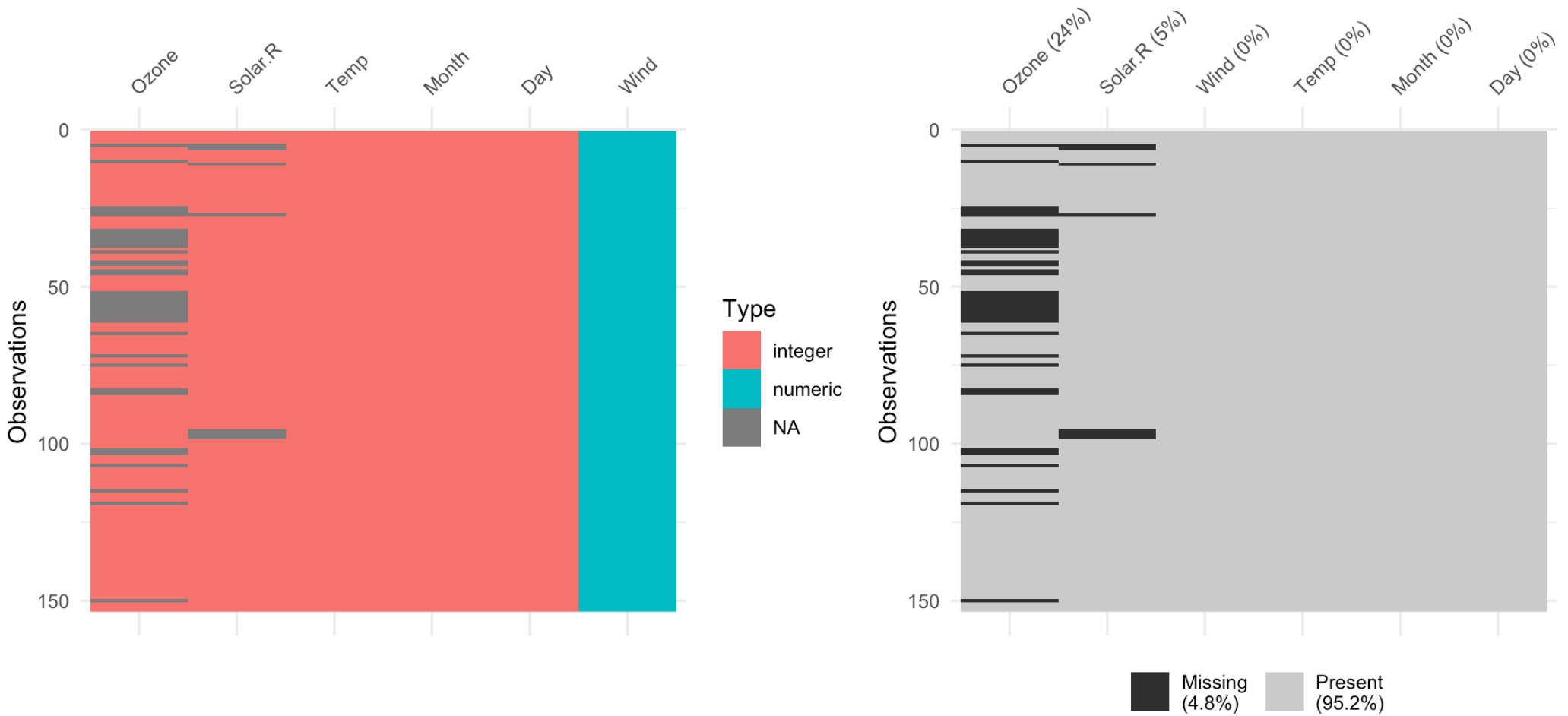
```
[1] FALSE FALSE FALSE FALSE  TRUE  TRUE
```

```
1 any_na(iris)
```

```
[1] FALSE
```

Visualising Missingness

▶ Code



Missing Mechanisms

1. Missing Completely at Random (MCAR)

- Missingness has no association with any data you have **observed**, or **not observed**. Missingness occurs **purely by chance**.
- **Example:**

Test	Special Consideration
65	FALSE
NA	TRUE
80	FALSE
70	FALSE
NA	FALSE

- **Handling**
  - ➟ Imputation is advisable
  - ➟ Deleting observations will not bias the results, but may reduce the model power due to the reduced sample size thus limiting inference.

Missing Mechanisms

2. Missing at Random (MAR)

- Missingness depends on **data observed**, but **not data unobserved**.
- **Example**:

Test	Special Consideration	Health
65	FALSE	Good
NA	TRUE	Poor
80	FALSE	Good
70	FALSE	Good
NA	FALSE	Poor

- **Handling:**
  - ➟ Use **imputation** to adjust for missing data.

Mechanism for missing data

3. Missing Not at Random (MNAR)

- Missingness is related to unobserved data
- **Example**:

Test	Special Consideration	Health
65	FALSE	Good
NA	TRUE	NA
80	FALSE	Good
70	FALSE	Good
NA	FALSE	NA

- **Handling**
  - ➟ Requires **modeling the missingness mechanism** to correct biases.

Identifying different types of missingness

- It's often **difficult to determine the missingness mechanism** using statistical tests alone.
- Instead, rely on:
  - **Domain knowledge** and understanding of the **data collection process**
  - **Predictive models**: Fit a classification model to predict missingness—if successful, data may be MAR.
- For testing **MCAR**:
  - Use tests like `mcar_test()` from the `naniar` package.
  - Based on Little's MCAR test (Little, 1988), which assumes multivariate normality.

Dealing with missing values

- For categorical data, “missing” can be a category
  - For example, in a survey poll, if someone does not want to disclose who they want to vote for, it then can be in the category “undecided”
- Delete data with missing value. Two options:
  - Omit the variable with missing data
  - Omit the observation with missing data
  - Drawbacks are that you might be throwing away valuable information, or inadvertently introduce bias into the data
- **Impute: fill in the missingness**

Single imputation

- Single imputation replaces the missing value with a single value
- Examples:
  - ➟ Replace the missing values of a feature with the mean/median value of that feature
  - ➟ Use a predictive method for filling in the missing values, e.g., regression trees, kNN
  - ➟ Replace the missing value with the last observed value for that feature
- **Drawbacks**:

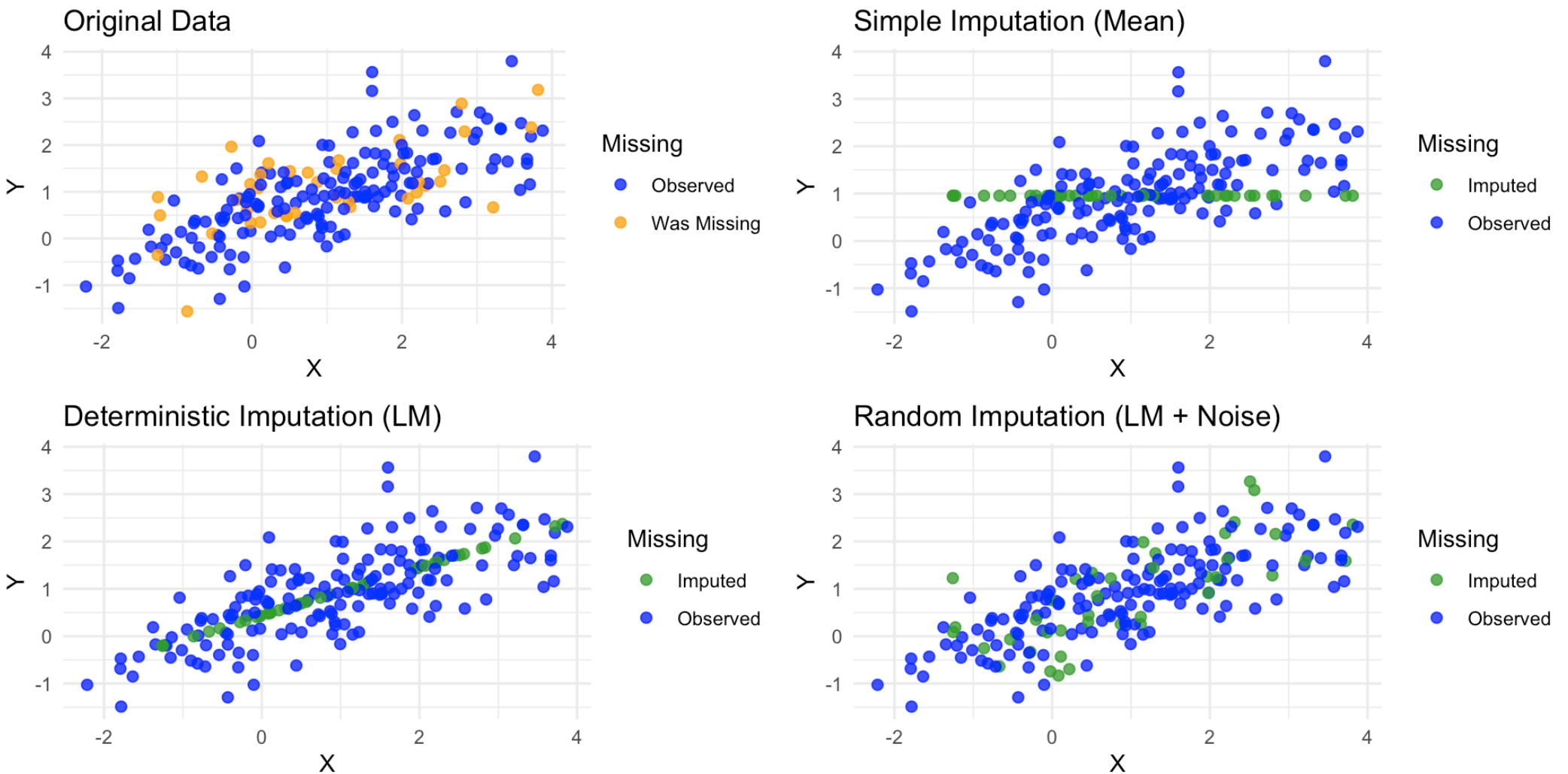
  With single imputation, once the missing data are added back, it is treated as valid observed data, hence the uncertainty in the missing value data is lost

Multiple imputation

- Multiple imputation accounts for uncertainty in the imputation process
- Generally follows three steps:
  - ➟ Impute the data k times (this can be done using a single imputation method)
  - ➟ Perform analysis (e.g., regression/classification) on each of the k imputed data sets
  - ➟ Pool the k results together
- Multiple Imputation by Chained Equation (MICE) is a popular method
  - ➟ See mice: Multivariate Imputation by Chained Equations in R

Basic impute, deterministic imputation, random imputatation

▶ Code



Other practical suggestions

- It is highly recommend that you visualize your data to look for patterns of missingness
- Be wary of variables with high proportion of missing data. However, this might not be a problem if imputation is applicable and performs well
- Some algorithms can cope with missingness (e.g., decision trees) and so you may not need to do imputation
- If you believe the pattern of missingness is informative, you can include it as a dummy variable

R packages for dealing with missing values

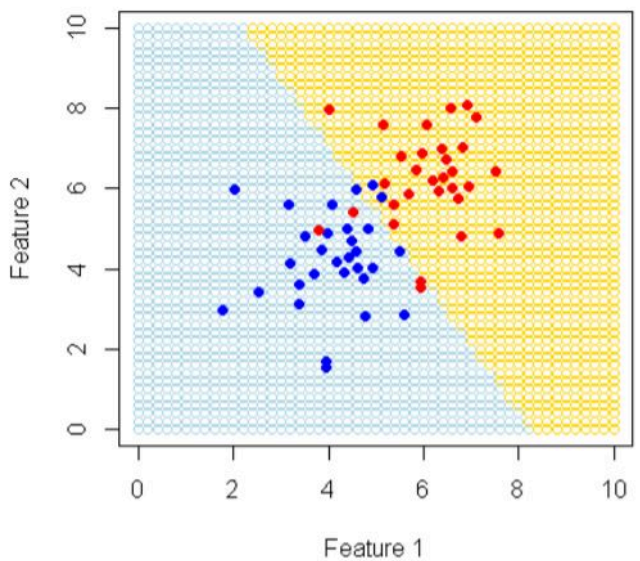
- Impute (Bioconductor package)
  - ➟ KNN imputation, written for microarray data
- MICE
  - ➟ Multiple imputation
- missForest
  - ➟ Uses Random Forest (in upcoming tree-based module) to predict the missing values
  - ➟ Can be used for continuous and categorical data
- Amelia II
  - ➟ Multiple imputation

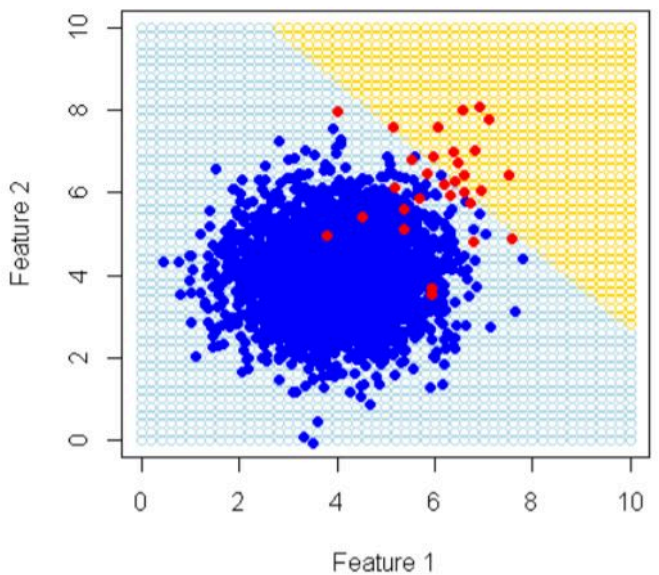
Class imbalance

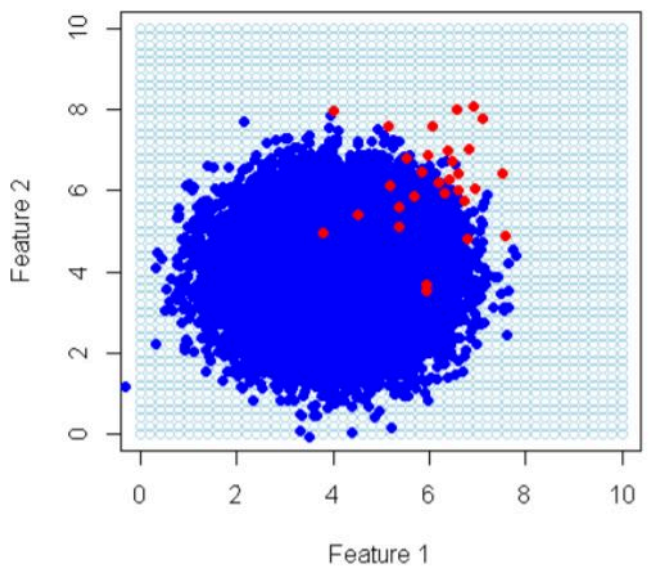
Class imbalance

- Let's say we have a classification problem to detect credit card fraud, but only 1% of transactions are fraud.
- If you use accuracy as the metric to optimise, then just classifying every transaction as not-fraud will get you to 99% accuracy!

Inspect a SVM







- Notice the negative (blue) class swamps the classifier
- Boundary moves and disappears favouring only the negative class

Handelling Class Imbalance

1. Use different metrics

- F_1 score

$$F_1 = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

- Area Under Curve (AUC) for the ROC
- **Cohen's Kappa**

What is Cohen's Kappa metric?

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

Where:

- P_o= Observed accuracy
- P_e = Expected accuracy generated simply by chance

Interpretation Guide

Kappa Value	Strength of Agreement
< 0	Less than chance
0.01–0.20	Slight
0.21–0.40	Fair
0.41–0.60	Moderate
0.61–0.80	Substantial
0.81–1.00	Almost perfect

Example: Binary Classification – Spam Detection

	Actual: Spam	Actual: Not Spam
Predicted: Spam	30	15
Predicted: Not Spam	10	45

- Observed accuracy:

$$P_o = \frac{30 + 45}{100} = 0.75$$

- Class proportions:
  - Predicted Spam: (45%); Predicted Not Spam: (55%)
  - Actual Spam: (40%); Actual Not Spam: (60%)
- Expected accuracy:

$$P_e = (0.4 \times 0.45) + (0.6 \times 0.55) = 0.18 + 0.33 = 0.51$$

- Kappa:

$$\kappa = \frac{0.75 - 0.51}{1 - 0.51} = \frac{0.24}{0.49} \approx 0.49$$

Cohen's Kappa for Class Imbalance Classification

- Example
  - 100 samples with 2% positive class and 98% negative class
  - A classifier classifies all samples as negative class
  - Lead to accuracy 98%, but what about other performance measures?
  - Sensitivity: 0; Specificity: 1; Cohen's kappa: 0

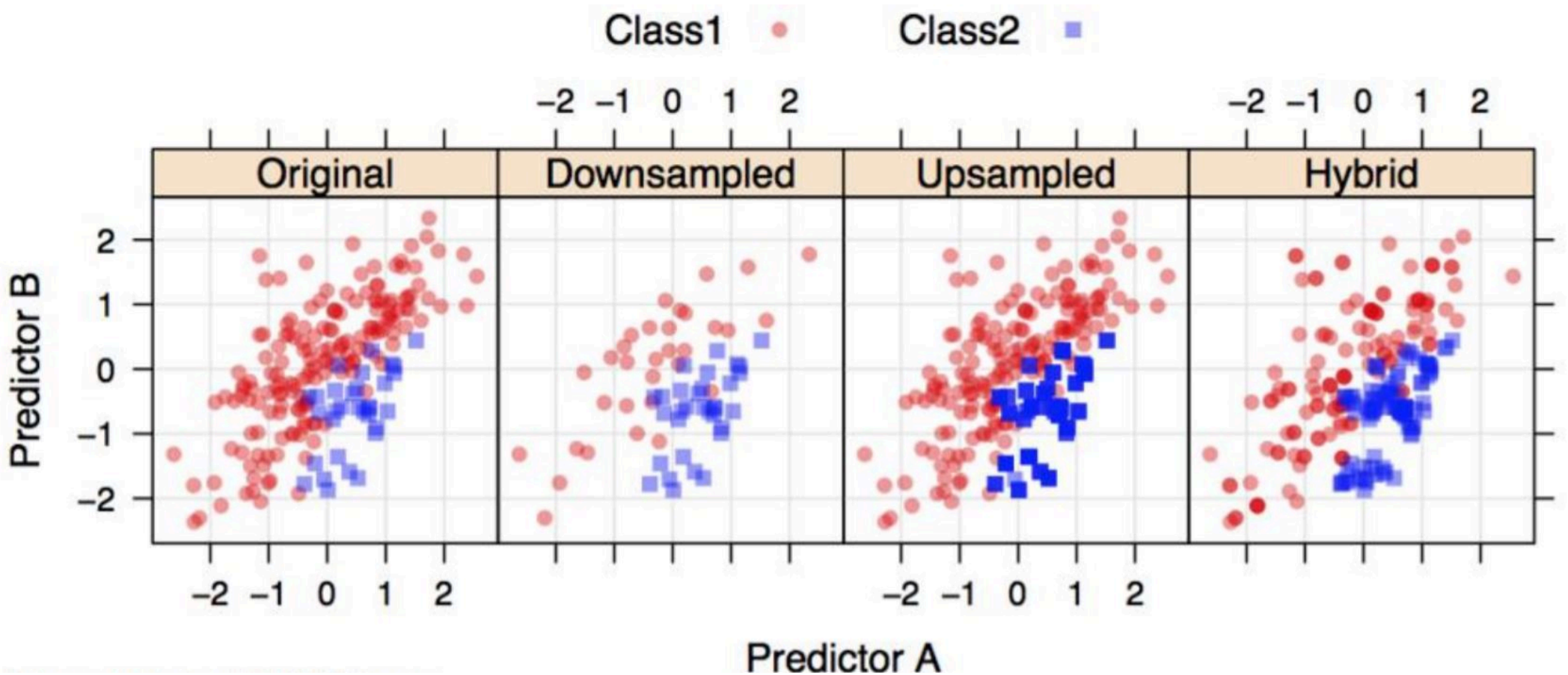
You can compute Kappa in R using `caret::confusionMatrix`

```
Confusion Matrix and Statistics

          Reference
Prediction positive negative
  positive        0        0
  negative        2       98

               Accuracy : 0.98
                 95% CI : (0.9296, 0.9976)
    No Information Rate : 0.98
    P-Value [Acc > NIR] : 0.6767

                  Kappa : 0

 Mcnemar's Test P-Value : 0.4795

            Sensitivity : 0.00
            Specificity : 1.00
         Pos Pred Value :  NaN
         Neg Pred Value : 0.98
```

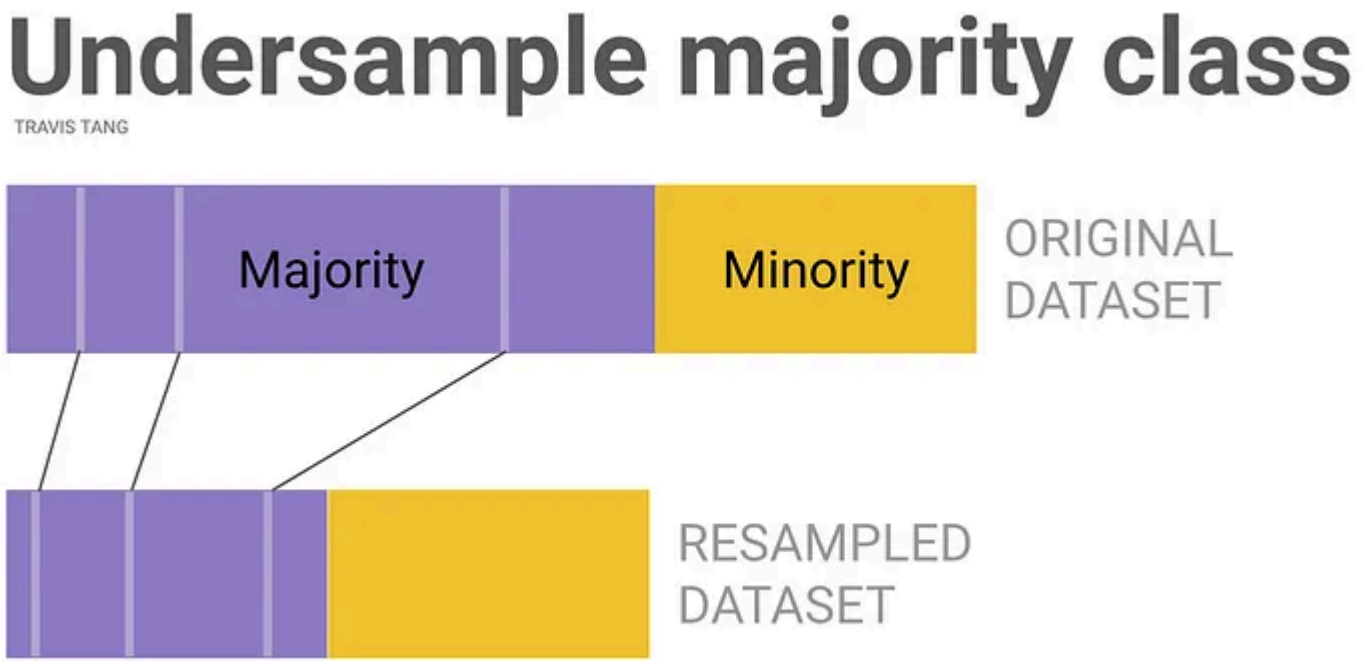
```
            Prevalence : 0.02
        Detection Rate : 0.00
Kappa
    0
```

Handelling Class Imbalance

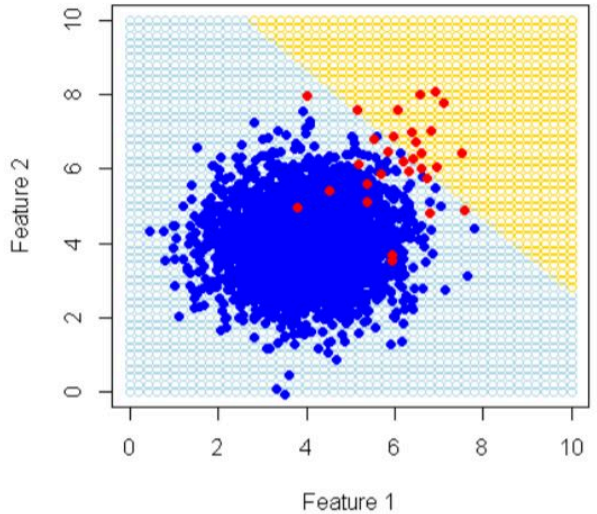
2. Modify the Data

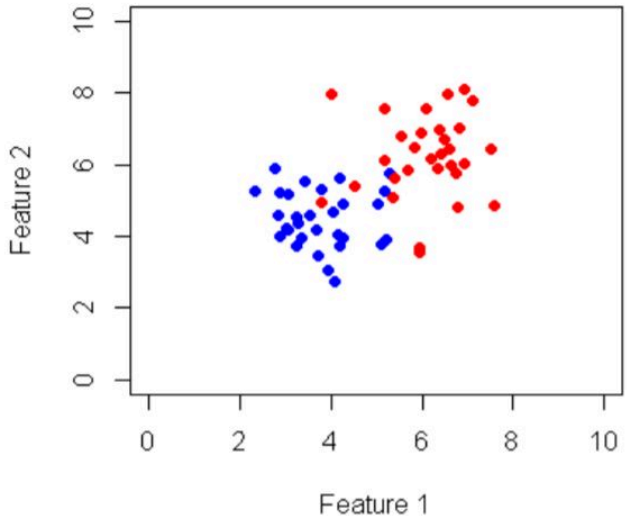


Down-sampling



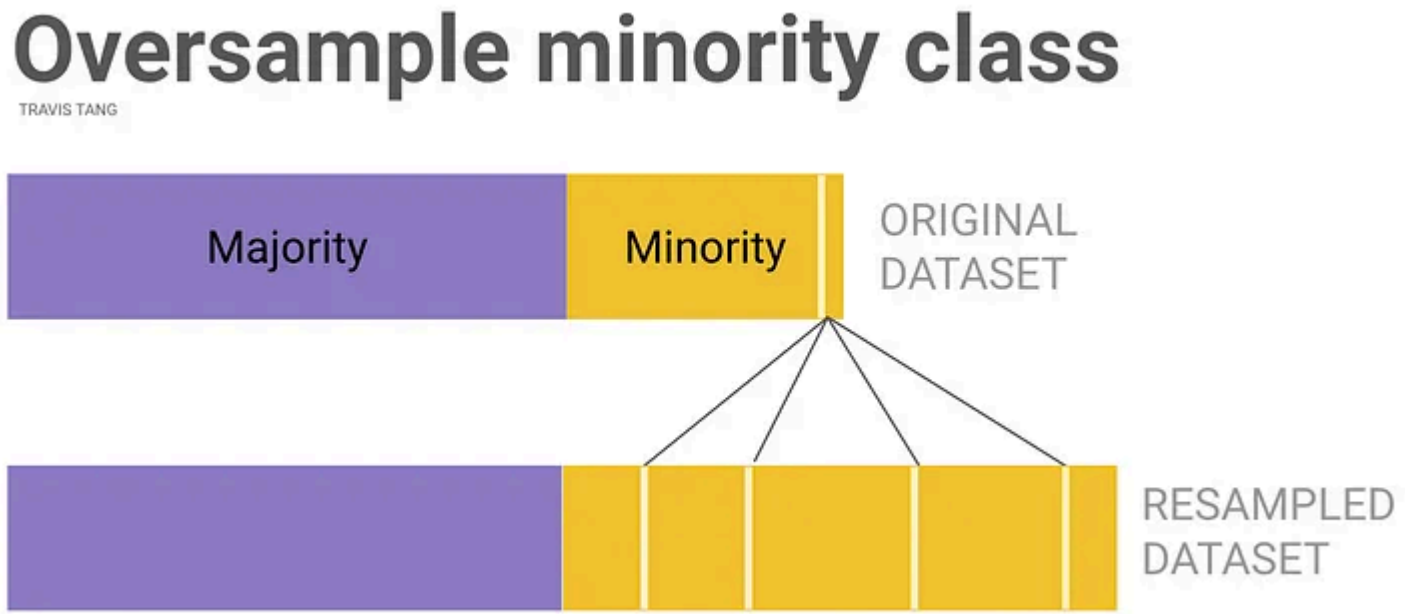
Travis Tang from Medium





- Advantage: it does not introduce duplicates and/or artificial instances.
- Disadvantages:
  - Not all data points are used.
  - Potentially removing useful information.
- Better choice for data with very high class imbalance.

Up-sampling



Travis Tang from Medium

- Random up-sampling
- Disadvantages:
  - creates duplicated and/or artificial instances
  - can introduce bias and/or noise to the original data

Create synthetic samples of the minority class

- Synthetic Minority Over-sampling Technique (SMOTE) is a popular algorithm
- It creates synthetic samples from the minority class by:
  - Finding the k-nearest-neighbours for minority class observations
  - Randomly choosing one of the k-nearest-neighbours, then using it to create a similar but random new observation
- Be careful that you split your data into training/test sets before doing any oversample/SMOTE. Otherwise, you will leak information from training to test data set.

Handelling Class Imbalance via R

- Balanced Resampling with `ROSE::ovun.sample()` can perform **under/over-sampling** or **hybrid** combines over- and under-sampling for balanced output.
- `themis::step_smote` helps balance binary or multiclass classification problems by generating synthetic examples (via knn) of the minority class.