Handling Missing Data

## Missing Data in Pandas:

## While R has a single integer type, NumPy supports fourteen basic integer types once you account for available precisions, signedness, and endianness of the encoding. Reserving a specific bit pattern in all available NumPy types would lead to an unwieldy amount of overhead in special-casing various operations for various types, likely even requiring a new fork of the NumPy package. Further, for the smaller data types (such as 8-bit integers), sacrificing a bit to use as a mask will significantly reduce the range of values it can represent.

NumPy does have support for masked arrays – that is, arrays that have a separate Boolean mask array attached for marking data as "good" or "bad." Pandas could have derived from this, but the overhead in both storage, computation, and code maintenance makes that an unattractive choice.

Pandas chose to use sentinels for missing data, and further chose to use two already-existing Python null values: the special floating-point NaN value, and the Python None object.

Pythonic missing data:

The first sentinel value used by Pandas is None, a Python singleton object that is often used for missing data in Python code. Because it is a Python object, None cannot be used in any arbitrary NumPy/Pandas array, but only in arrays with data type 'object' (i.e., arrays of Python objects)

This dtype=object means that the best common type representation NumPy could infer for the contents of the array is that they are Python objects.

## Operating on Null Values

As we have seen, Pandas treats None and NaN as essentially interchangeable for indicating missing or null values. To facilitate this convention, there are several useful methods for detecting, removing, and replacing null values in Pandas data structures. They are:

* isnull(): Generate a boolean mask indicating missing values
* notnull(): Opposite of isnull()
* dropna(): Return a filtered version of the data
* fillna(): Return a copy of the data with missing values filled or imputed

### Filling null values

Sometimes rather than dropping NA values, you'd rather replace them with a valid value. This value might be a single number like zero, or it might be some sort of imputation or interpolation from the good values. You could do this in-place using the isnull() method as a mask, but because it is such a common operation Pandas provides the fillna() method, which returns a copy of the array with the null values replaced.

## Operating on Null Values

As we have seen, Pandas treats None and NaN as essentially interchangeable for indicating missing or null values. To facilitate this convention, there are several useful methods for detecting, removing, and replacing null values in Pandas data structures. They are:

* isnull(): Generate a boolean mask indicating missing values
* notnull(): Opposite of isnull()
* dropna(): Return a filtered version of the data
* fillna(): Return a copy of the data with missing values filled or imputed

We will conclude this section with a brief exploration and demonstration of these routines.

# Preprocesamiento de datos con Python Pandas

El agrupamiento de datos (o bucketing) agrupa los datos en bandejas (o buckets), en el sentido de que reemplaza los valores contenidos en un pequeño intervalo con un único valor representativo para ese intervalo. A veces, el binning mejora la precisión en los modelos predictivos.

El agrupamiento de datos es un tipo de preprocesamiento de datos, un mecanismo que incluye también tratar con [valores faltantes](https://towardsdatascience.com/data-preprocessing-with-python-pandas-part-1-missing-data-45e76b781993), [formato](https://towardsdatascience.com/data-processing-with-python-pandas-part-2-data-formatting-710c2eafa426), [normalización](https://towardsdatascience.com/data-preprocessing-with-python-pandas-part-3-normalisation-5b5392d27673) y [estandarización](https://towardsdatascience.com/data-preprocessing-with-python-pandas-part-4-standardization-ccd5b1608f1c).

El binning se puede aplicar para convertir valores numéricos en valores numéricos categóricos o para muestrear (cuantificar).

El binning es una técnica para suavizar datos. El suavizado de datos se emplea para eliminar el ruido de los datos. Tres técnicas para suavizar datos:

* Binning
* regresión
* Análisis de valores atípicos.

# Muestreo

El muestreo es otra técnica de agrupación de datos. Permite reducir el número de muestras, agrupando valores similares o valores contiguos. Existen tres enfoques para realizar el muestreo:

* Por bin significa: cada valor en un contenedor se reemplaza por el valor medio del contenedor.
* Por mediana de bin: cada valor de bin se reemplaza por su valor de bin mediano.
* Por límite de contenedor: cada valor de contenedor se reemplaza por el valor límite más cercano, es decir, el valor máximo o mínimo del contenedor.