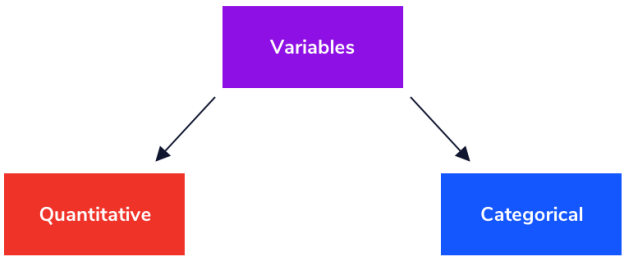
Variable Types for Data Science

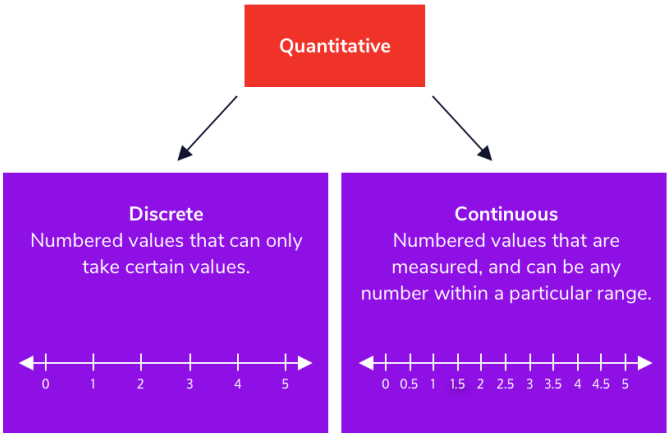
Quantitative Vs. Categorical Variables

Variables can be either quantitative or categorical. Quantitative variables are amounts or counts; for example, age, number of children, and income are all quantitative variables. Categorical variables represent groupings; for example, type of pet, agreement rating, and brand of shoes are all categorical variables.



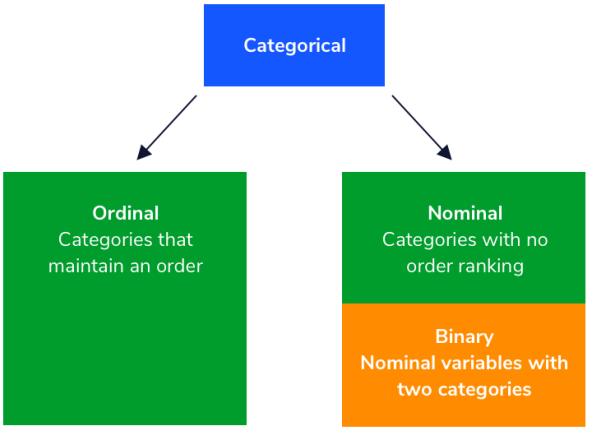
Quantitative Variables

Quantitative variables are numeric in nature and can be either continuous or discrete. Continuous variables contain measurements with decimal precision, for example the height or weight of a person. Discrete variables contain counts that must be whole integer values, such as the number of members in a person's family, or the number of goals a basketball team scored in a game.



Categorical Variables

Categorical variables consist of data that can be grouped into distinct categories, and are ordinal or nominal. Ordinal categorical variables which are groups that contain an inherent ranking, such as ratings of plays or responses to a survey question with a point scale e.g., on a scale from 1-7, how happy are you right now? Nominal categorical variables are made of categories without an inherent order, examples of nominal variables are species of ants, or people's hair color.



Ordinal Vs. Discrete Variables

A key distinction between ordinal categorical variables and discrete quantitative variables is that there is a uniform degree of difference within discrete quantitative variables. The difference between one and two kittens is the same as the difference between five and six kittens. With ordinal categorical variables however, the difference between categories can vary greatly. The difference between a one star rating and a two star rating for example can be different than a three star rating and a four star rating.

Binary Categorical Variables

Categorical variables can also be binary or dichotomous variables. Binary variables are nominal categorical variables that contain only two, mutually exclusive categories. Examples of binary variables are if a person is pregnant, or if a house's price is above or below a particular price.

Inspecting Variable Types

One of the most important first steps when working with a dataset is to inspect the variable types, and identify relevant variables. An efficient method to use when inspecting variables is the `.head()` method which will return the first rows of a dataset.

```
print(df.head())
```

Matching Variable Types and Data Types

Ensuring that variables within your dataset are expressed with the appropriate data type will help you manage your data effectively, and allow you to

```
print(df.dtypes)
```

perform any necessary operations on your variables. When using Python, the data types of pandas dataframes can be inspected with the `.dtypes` accessor. Usually, continuous quantitative variables are represented as floats, discrete quantitative variables as integers, binary variables as booleans, nominal categorical variables as strings, and ordinal categorical variables as integers or strings.

Storing Ordinal Categories

It is often useful to store ordinal categorical variables as both strings and integers. For example, suppose there is a variable named `response` that contains responses to the question "Rate your agreement with the statement: the wealthy should pay higher taxes," where the response options are "strongly disagree", "disagree", "neutral", "agree" and "strongly agree". Then in order to do future calculations, we may want to also store those categories as numerical values such as 0, 1, 2, 3, and 4.

The Pandas Category Data Type

When working with categorical variables in Python, especially ordinal categorical variables, it can often be advantageous to use the Pandas specific `category` datatype, which allows you to store category names with associated values and rankings.

```
df['column_cat'] =
pd.Categorical(df['column'], ['cat1',
'cat2', 'cat3'], ordered = True)
```

Altering a Variable's Data Type

Often when working with datasets, variables will be assigned an inappropriate datatype. For example you may have a continuous variable which is assigned the `str` datatype. In this scenario, it will not be possible to perform numerical operations on that variable. In this case it will be necessary to alter the datatype to something more appropriate, such as `float`.

Methods for Altering Data Types

In the event that you need to alter the datatype of a variable in Python, you can use the `.astype()` method, which allows you to assign a new datatype to a variable in your dataset. However there may be cases where certain values do not allow you to implement `.astype()` for example a `missing` value in a discrete variable that was assigned the `str` datatype. To change this variable's datatype to `int` you would need to can use the `.replace()` method to

```
df['column'] = df['column'].astype('int')
df['column'] =
df['column'].replace(['missing'], None)
```

change missing values to something more appropriate, then alter the data type of the variable.

One-Hot Encoding with Python

When working with nominal categorical variables in Python, it can be useful to use One-Hot Encoding, which is a technique that will effectively create binary variables for each of the nominal categories. This encodes the variable without creating an order among the categories. To one-hot encode a variable in a pandas dataframe, we can use the `.get_dummies()`.

```
df = pd.get_dummies(data = df, columns=
['column1', 'column2')
```