



**DigitalHouse** >  
Coding School

# Data Science

MÓDULO 2

Data Wrangling

1

Limpieza y transformación de datos  
Práctica Guiada 1 - Parte I

2

Variables cualitativas y dummies, manejo de strings  
Práctica Guiada 1 - Parte II

3

TimeStamp: Manejos de Fechas y Horas en Pandas  
Práctica Guiada 2

4

Práctica Independiente

# Data Wrangling



- **Data Wrangling**

- Proceso de limpieza y unificación de conjuntos de datos desordenados y complejos para facilitar su acceso, exploración, análisis o modelización posterior.

- **Data munging:** (el arduo) proceso de limpiar, preparar y validar los datos.

- **Extract, Transform and Load (ETL):** extraer, transformar y cargar los datos.

- **Exploratory data analysis (EDA)**

- Resumir sus principales características, a menudo con métodos visuales. Se puede usar o no un modelo estadístico, pero principalmente EDA es para ver lo que los datos pueden decirnos más allá de la tarea de modelado formal o prueba de hipótesis.

# Práctica Guiada - Parte I

## Limpieza y transformación de datos



# Variables categóricas



- Las variables pueden ser caracterizadas como:
  - **cuantitativas.** Estas toman valores numéricos como en el caso de del ingreso de una persona o el precio de una casa.
  - cualitativas.
- Una **variable cualitativa** es una variable que toma valores en una de K diferentes clases o categorías.
- Una variable cualitativa con dos posibles valores se denomina **binaria** o **dicotómica**.

- Tipos de variable cualitativa:
  - Nominal/Categórica. Categorías nombradas.
    - Se suele asignar **valores o rótulos numéricos** a las variables categóricas: Estado civil, 0 si soltero y 1 si casado y 2 si divorciado
    - Los números utilizados para rotular son arbitrarios. En general, el software asume que los valores numéricos reflejan cantidades algebraicas y, por tanto, un orden cierto.
    - La principal medida de posición es la **moda**. La mediana y la media no están definidas (y en general cualquier operación numérica tampoco).



- Tipos de variable cualitativa:
  - Nominal/Categórica. Categorías nombradas.

What is your gender?

- ☒ M - Male
- ☐ F - Female

What is your hair color?

- ☒ 1 - Brown
- ☐ 2 - Black
- ☐ 3 - Blonde
- ☐ 4 - Gray
- ☐ 5 - Other

Where do you live?

- ☒ A - North of the equator
- ☐ B - South of the equator
- ☐ C - Neither: In the international space station

- Tipos de variable cualitativa:
  - Ordinal.
    - Es similar a una categórica pero existe un orden claro.

**How do you feel today?**

- ☒ 1 – Very Unhappy
- ☐ 2 – Unhappy
- ☐ 3 – OK
- ☐ 4 – Happy
- ☐ 5 – Very Happy

**How satisfied are you with our service?**

- ☒ 1 – Very Unsatisfied
- ☐ 2 – Somewhat Unsatisfied
- ☐ 3 – Neutral
- ☐ 4 – Somewhat Satisfied
- ☐ 5 – Very Satisfied

- Una **variable dummy** (variable indicadora) es una variable cualitativa que toma valores 0 o 1 para indicar la ausencia o presencia de algún atributo o efecto categórico.
  - Formalmente una variable dummy puede ser expresada mediante una **función indicadora**:

$$D_i = \mathbb{I}_A(x_i) = \begin{cases} 1 & \text{si } x_i \in A \\ 0 & \text{si } x_i \notin A \end{cases}$$

- ¿Cuál es la relación entre variables categóricas y variables dummies?
  - Una variable categórica con  $N$  categorías puede ser expresada en términos de  $N-1$  variables dummies (**one-hot encoding**).
  - Resuelve el problema de interpretar las etiquetas numéricas como un intervalo.
  - Si las categorías tienen muchos valores aumenta considerablemente la dimensionalidad de los datos.
  - Veamos un ejemplo...

- Supongamos que tenemos una variable categórica, C, que registra la ciudad en la que reside una muestra de habitantes de la Argentina.
  - Asumamos que la variable puede tomar 4 posibles valores: Buenos Aires, Rosario, Córdoba y Mar del Plata.
  - Imaginemos que tenemos las siguiente 5 observaciones:

| Obs. | Ciudad        |
|------|---------------|
| 1    | Rosario       |
| 2    | Buenos Aires  |
| 3    | Rosario       |
| 4    | Mar del Plata |
| 5    | Córdoba       |

- Alternativamente podemos expresar estas observaciones de la variable categórica usando dummies como:

| Obs. | Ciudad        |
|------|---------------|
| 1    | Rosario       |
| 2    | Buenos Aires  |
| 3    | Rosario       |
| 4    | Mar del Plata |
| 5    | Córdoba       |

| Obs. | D_BA | D_C | D_R |
|------|------|-----|-----|
| 1    | 0    | 0   | 1   |
| 2    | 1    | 0   | 0   |
| 3    | 0    | 0   | 1   |
| 4    | 0    | 0   | 0   |
| 5    | 0    | 1   | 0   |

- Es importante notar que si existen  $k$  categorías,  $k-1$  variables Dummies son suficientes para representarlas.

# Práctica Guiada Parte II



- Unix TimeStamp es una manera de “trackear” el tiempo.
- Se considera el tiempo como una suma de segundos, desde el 1 de Enero de 1970 UTC, a las 00:00:00.
- Teniendo un punto de referencia fijo, resulta muy simple el manejo de tiempo y fechas en distintos sistemas y arquitecturas.





Los datos con TimeStamps son la forma más básica de Series de Tiempo. Asocian valores con puntos en el tiempo. En Pandas, podemos instanciar objetos de la clase **Timestamp**

```
In [8]: pd.Timestamp(datetime(2012, 5, 1))
```

```
Out[8]: Timestamp('2012-05-01 00:00:00')
```

```
In [9]: pd.Timestamp('2012-05-01')
```

```
Out[8]: Timestamp('2012-05-01 00:00:00')
```

```
In [10]: pd.Timestamp(2012, 5, 1)
```

```
Out[10]: Timestamp('2012-05-01 00:00:00')
```

En muchos casos es más natural asociar variables con **intervalos de tiempo**.

Representamos un intervalo con un objeto de la clase **Period**, y puede ser instanciado explícitamente o inferido de un **string con cierto formato**.

```
In [11]: pd.Period('2011-01')
```

```
Out[11]: Period('2011-01', 'M')
```

```
In [12]: pd.Period('2012-05', freq='D')
```

```
Out[12]: Period('2012-05-01', 'D')
```

Los objetos de tipo Timestamp y Period pueden ser índices. Las listas que contienen estos objetos son casteadas automáticamente a objetos de tipo **DatetimeIndex** y **PeriodIndex**, respectivamente.

```
In [13]: dates = [pd.Timestamp('2012-05-01'), pd.Timestamp('2012-05-02'), pd.Timestamp('2012-05-03')]
```

```
In [14]: ts = pd.Series(np.random.randn(3), dates)
```

```
In [15]: type(ts.index)
```

```
Out[15]: pandas.core.indexes.datetimes.DatetimeIndex
```

```
In [16]: ts.index
```

```
Out[16]: DatetimeIndex(['2012-05-01', '2012-05-02', '2012-05-03'], dtype='datetime64[ns]', freq=None)
```

A veces necesitamos especificar el formato que ayuda a parsear el string, por ejemplo

```
pd.to_datetime('20170901 100500', format='%Y%m%d %H%M%S')
```

| Code | Meaning   | Example   |
|------|---|-----------|
| %a   | Weekday as locale's abbreviated name.                             | Mon       |
| %A   | Weekday as locale's full name.                                    | Monday    |
| %w   | Weekday as a decimal number, where 0 is Sunday and 6 is Saturday. | 1         |
| %d   | Day of the month as a zero-padded decimal number.                 | 30        |
| %-d  | Day of the month as a decimal number. (Platform specific)         | 30        |
| %b   | Month as locale's abbreviated name.                               | Sep       |
| %B   | Month as locale's full name.                                      | September |
| %m   | Month as a zero-padded decimal number.                            | 09        |
| %-m  | Month as a decimal number. (Platform specific)                    | 9         |
| %y   | Year without century as a zero-padded decimal number.             | 13        |
| %Y   | Year with century as a decimal number.                            | 2013      |
| %H   | Hour (24-hour clock) as a zero-padded decimal number.             | 07        |

# Práctica Guiada - Manejo del tiempo

