

Importación de librerías

```
In [ ]: import pandas as pd
import numpy as np
import datetime as dt
```

```
In [ ]: # Carga del dataframe de ventas depurado

df_ventas = pd.read_csv("df_ventas_1.csv", sep=",")
```

Tratamiento del campo hora para obtener variables de cara a la visualización/modelos

```
In [ ]: # Obtención de distintos campos de fecha

# Campo sin horas
df_ventas["date"] = df_ventas["created_at"].apply(lambda x: x[0:10])

# Campo de año
df_ventas["year"] = df_ventas["created_at"].apply(lambda x: x[0:4])

# Campo de horas por franja
df_ventas["hour"] = df_ventas["created_at"].apply(lambda x: x[11:13])

# Campo de numero de semana (del 1 al 104)
df_ventas["week"] = df_ventas["date"]
df_ventas["week"] = df_ventas["week"].apply(lambda _: dt.datetime.strptime(
df_ventas["week"], "%Y-%m-%d").dt.isocalendar().week
df_ventas["week"][df_ventas["year"] == "2018"] = df_ventas["week"][df_ventas["year"] == "2018"] + 1

# Campo dia de la semana
df_ventas["date"] = pd.to_datetime(df_ventas["date"])
df_ventas["day"] = df_ventas["date"].apply(lambda x: pd.Timestamp(x).day_name())

In [ ]: df_ventas["margin_total"] = (df_ventas["price"] - df_ventas["base_cost"]) *
df_ventas["price_total"] = df_ventas["price"] * df_ventas["qty_ordered"]

In [ ]: df_ventas.head()
```

```
Out[ ]:      Unnamed: 0      item_id      num_order      created_a
```

0	0	000010d95384a6ba3d57dd870e7b337c	65717498f0771a49497d80f11160093c	2017-09-2 15:46:3
1	1	00001a8fb0bd42b1e16ba731e30cc490	09b538e85ce396ecbb70695f91007830	2018-09-1 21:27:0
2	2	0000302bc9b9a670dfcb14381555ff45	bc150db52b5a565d31b1c70969638ca9	2018-11-1 16:36:1
3	3	000039147df4aacf0aa8b3a552e8ecdb	434cf1eaf255b367ce2d3343bb96b1fe	2017-09-0 12:08:4
4	4	000091029a220c2fdf12700f07f70b1d	f268c24275ad1d887925fca2909e2c2d	2018-09-2 09:45:1

5 rows × 22 columns

```
In [ ]: df_ventas.drop("Unnamed: 0", inplace=True, axis = 1)
```

Exportación de resultados a formato CSV

```
In [ ]: df_ventas.to_csv('df_ventas_2.csv')
```