Lab 8

- Daniel Alfredo Rayo Roldan
- Flavio Galan
- Maria Fernanda
- Irving Fabricio


```
import pandas as pd
from datetime import datetime, timedelta
import matplotlib.pyplot as plt
```

Análisis exploratorio

```
df = pd.read_csv("./data/train.csv")

df['date'] = pd.to_datetime(df['date'])

df = df.sort_values(['date', 'store', 'item']).reset_index(drop=True)
```

```
print("Dimensiones del dataframe:", df.shape)
print("\nPrimeras filas:")
print(df.head())
```

```
Dimensiones del dataframe: (913000, 4)

Primeras filas:
        date  store  item  sales
0 2013-01-01      1     1     13
1 2013-01-01      1     2     33
2 2013-01-01      1     3     15
3 2013-01-01      1     4     10
4 2013-01-01      1     5     11
```

```
print("\nInformacion de Data:")
print(df.info())
```

```
Informacion de Data:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 913000 entries, 0 to 912999
Data columns (total 4 columns):
 #   Column  Non-Null Count   Dtype         
---  ------  --------------   -----         
 0   date    913000 non-null  datetime64[ns]
 1   store   913000 non-null  int64         
 2   item    913000 non-null  int64         
 3   sales   913000 non-null  int64         
dtypes: datetime64[ns](1), int64(3)
memory usage: 27.9 MB
None
```

```
print("\nEstadisticas básicas:")
print(df.describe())
print("\nRango de Fechas:", df['date'].min(), "to", df['date'].max())
print("\nNumero de tiendas:", df['store'].nunique())
print("\nNumero de items:", df['item'].nunique())
```

```
Estadisticas básicas:
                                date          store           item  \
count                         913000  913000.000000  913000.000000   
mean   2015-07-02 12:00:00.000000256       5.500000      25.500000   
min              2013-01-01 00:00:00       1.000000       1.000000   
25%              2014-04-02 00:00:00       3.000000      13.000000   
50%              2015-07-02 12:00:00       5.500000      25.500000   
75%              2016-10-01 00:00:00       8.000000      38.000000   
max              2017-12-31 00:00:00      10.000000      50.000000   
std                              NaN       2.872283      14.430878   

               sales  
count  913000.000000  
mean       52.250287  
min         0.000000  
25%        30.000000  
50%        47.000000  
75%        70.000000  
max       231.000000  
std        28.801144  

Rango de Fechas: 2013-01-01 00:00:00 to 2017-12-31 00:00:00

Numero de tiendas: 10

Numero de items: 50
```

```
print("Valores nulos")
df.isna().sum()
```

```
Valores nulos
date     0
store    0
item     0
sales    0
dtype: int64
```

```
duplicates = df.duplicated(subset=['date', 'store', 'item'])
print(f"\nFilas duplicadas: {duplicates.sum()}")
```

```
Filas duplicadas: 0
```

Limpieza de datos

```
min_date = df['date'].min()
max_date = df['date'].max()
all_dates = pd.date_range(start=min_date, end=max_date, freq='D')

stores = df['store'].unique()
items = df['item'].unique()

complete_index = pd.MultiIndex.from_product(
    [all_dates, stores, items],
    names=['date', 'store', 'item']
)
df_complete = pd.DataFrame(index=complete_index).reset_index()

df_complete = df_complete.merge(df, on=['date', 'store', 'item'], how='left')
```

```
def plot_item_all_stores(df, item_id, max_stores=10, figsize=(15, 10)):
    # Filter data for the item
    item_data = df[df['item'] == item_id]
    stores = sorted(item_data['store'].unique())[:max_stores]

    n_stores = len(stores)
    n_cols = 2
    n_rows = (n_stores + 1) // 2

    fig, axes = plt.subplots(n_rows, n_cols, figsize=figsize)
    axes = axes.flatten() if n_stores > 1 else [axes]

    for idx, store_id in enumerate(stores):
        data = item_data[item_data['store'] == store_id].sort_values('date')

        ax = axes[idx]
        ax.plot(data['date'], data['sales'], linewidth=1, alpha=0.7)

        # Add 7-day moving average
        rolling_mean = data['sales'].rolling(window=7, center=True).mean()
        ax.plot(data['date'], rolling_mean, linewidth=2, color='red', alpha=0.7)

        ax.set_title(f'Store {store_id}', fontsize=11, fontweight='bold')
        ax.set_xlabel('Date', fontsize=9)
        ax.set_ylabel('Sales', fontsize=9)
        ax.grid(True, alpha=0.3)
        ax.tick_params(axis='x', rotation=45, labelsize=8)

    # Hide extra subplots
    for idx in range(n_stores, len(axes)):
        axes[idx].axis('off')
```

```
    fig.suptitle(f'All Stores - Item {item_id}', fontsize=16, fontweight='bold', y=1.
    plt.tight_layout()
    plt.show()
```

Conjuntos de datos

```
max_date = df['date'].max()

# Define split points
test_start = max_date - timedelta(days=90 - 1) # 90 days = 3 months
val_start = test_start - timedelta(days=90) # 90 days for validation

# Split the data
train_df = df[df['date'] < val_start].copy() # rest of the months
val_df = df[(df['date'] >= val_start) & (df['date'] < test_start)].copy()
test_df = df[df['date'] >= test_start].copy()
```