Trabajando con fechas

In [2]:

```
import pandas as pd
import os
import matplotlib.pyplot as plt
```

In [ ]:

```
os.getcwd()
```

Librería Datetime: *https://docs.python.org/3/library/dat*...**Pandas to datetime:** https://pandas.pydata.org/pandas-docs...

In [4]:

```
df = pd.read_excel(os.getcwd() + "\\Global Superstore_2.xls")
```

In [5]:

```
df_c = df.loc[:,["Order Date","Ship Date","Segment","Category","Sales","Quantity"]]
```

In [ ]:

```
df_c.info()
```

Ordenamos por fecha de pedido

In [7]:

```
df_c = df_c.sort_values(by=['Order Date'], ascending = [True])
```

In [ ]:

```
df_c.shape
```

In [ ]:

```
df_c
```

Obtenemos fecha maxima y fecha minima

In [ ]:

```
print(max(df_c["Order Date"]))
print(min(df_c["Order Date"]))
```

Seleccionamos solo los registros del 2014

In [323]:

```
df_2014 = df_c[(df_c["Order Date"] >= "2014-01-01") & (df_c["Order Date"] <= "2014-12-31")]
```

In [ ]:

```
df_2014
```

In [ ]:

```
plt.rcParams["figure.figsize"] = (30,10) # con esta linea predefinimos el tamaño del gráfico
df_gbd = df_2014.groupby("Order Date").sum()["Sales"]
df_gbd.plot.line(x=df_gbd.index,y = df_gbd, fontsize = 20 )
```

Creamos una nueva columna con el formato de fecha d/m/a, pero OJO!! Será una columna tipo object, es decir, de texto.

In [ ]:

```
import datetime
df_2014["Order Date_str"] = df_2014["Order Date"].dt.strftime('%d/%m/%Y')
```

In [ ]:

```
df_2014.head()
```

In [ ]:

```
df_2014.info()
```

Extraemos el mes

In [ ]:

```
df_2014["mes pedido"] =  df_2014['Order Date'].dt.month
```

In [ ]:

```
df_2014.info()
```

In [ ]:

```
df_2014
```

En la documentación de la librería datetime (link al inicio del notebook), al final del todo tenemos una tabla con los diferentes codigos que nos sirven para obtener los diferentes formatos de fecha, nombre mes, dia semana, etc..

In [ ]:

```
#  %b nos devuelve el nombre del mes
df_2014["mes pedido_str"] =  df_2014["Order Date"].dt.strftime('%b')
```

In [ ]:

```
#  %b nos devuelve el día de la semana
df_2014["dia_semana_pedido_str"] =  df_2014["Order Date"].dt.strftime('%A')
```

In [ ]:

```
df_2014
```

Hemos visto como obtner un fragmento del dataframe (2014) filtrando como ya sabíamos hacer con otro tipo de variables. Ahora vamos a obtener el de 2013 con el metodo between:

In [335]:

```
df_2013 =df.loc[df["Order Date"].between('2013-01-01', '2013-12-31')]
```

In [ ]:

```
df_2013.head()
```

Diferencia entre fechas

Crearemos una columna que nos devuelva los días entre la fecha del pedido y la fecha del envío.

In [443]:

```
df_c["Días_transcurridos"] = (df_c["Ship Date"] - df_c["Order Date"]).dt.days
```

In [ ]:

```
df_c
```

Vamos a montar algún gráfico interesante:

In [ ]:

```
# extraemos el año del pedidio del df original
df_c["order_year"] = df_c['Order Date'].dt.year
df_c["order_month"] = df_c['Order Date'].dt.month
df_c["order_w_day"] =  df_c["Order Date"].dt.strftime('%A')
df_c.head()
```

In [ ]:

```
type(groupeddf)
```

In [339]:

```
groupeddf = df_c.groupby(["order_year","order_month"]).mean()["Sales"]
```

In [340]:

```
pd_group = pd.DataFrame(groupeddf).reset_index()
```

In [ ]:

```

```

In [ ]:

```
pd_group.head()
```

In [342]:

```
df_ym = df_c.groupby(["order_year","order_month"]).mean()["Sales"].to_frame()
```

In [ ]:

```
df_ym
```

In [ ]:

```
df_c.columns
```

In [345]:

```
p_table = pd.pivot_table(df_ym, index=['order_month'], columns = ["order_year"])
```

In [ ]:

```
p_table.plot.line()
```

In [ ]:

```
p_table.plot.bar()
```

In [ ]:

```
df_2014["dia_semana_pedido_str"].unique()
```

In [ ]:

```
len(df_2014)
```

Creamos una columna que nos diga si es fin de semana o dia de diario

In [ ]:

```
df_c.columns
```

In [419]:

```
for n in range(len(df_c)):

    df_c.loc[n,"Finde"] = "Finde" if df_c.loc[n, "order_w_day"] in ['Saturday',"Sunday"] else "Diario"
```

In [ ]:

```
df_c
```

In [421]:

```
df_c_sales = df_c.loc[:,['Sales','order_year',"Finde"]]
```

In [465]:

```
#creamos una medida mas manejable dividiendo Slaes entre 1000
df_c_sales["Sales_miles€"] =  round(df_c_sales["Sales"]/1000,2)
```

In [466]:

```
p_table2 = pd.pivot_table(df_c_sales, index=['Finde'], columns = ["order_year"], values = "Sales_miles€"
```

```
, aggfunc="sum")
```

In [ ]:

```
p_table2
```

In [ ]:

```
p_table2.plot.pie(subplots=True, # Parámetro necesario para múltiples quesitos
                  figsize = (20,20), # Tamaño quesito
                  autopct = "%1.2f %%", # Formato
                  legend = False, #Leyenda

                  startangle = 90
                 )
```

In [ ]:

```
p_table2
```

In [470]:

```
p_table3 = pd.pivot_table(df_c_sales, index=['order_year'], columns = ["Finde"], values = "Sales_miles€"
, aggfunc="sum")
```

In [ ]:

```
p_table3.plot.bar(stacked = True)

plt.title("Ventas por años desglosado en fines de semana o dias de diario", fontsize = 35)
plt.xlabel("Años", fontsize = 20)
plt.ylabel("Ventas", fontsize = 20)
plt.legend(fontsize = 30)
plt.xticks(fontsize = 30)
plt.yticks(fontsize = 20)
print(p_table3)
plt.show()
```