

UNIVERSIDAD POLITÉCNICA DE PUEBLA

Inteligencia de Negocios

Tablero de Pandas

Nava Dionicio Gerardo - 191801029
Cuatrimestre: Mayo - Agosto del 2024
Profesor: Joel Herrera Cruz



Contenido

1.-Tema	2
2.- ¿Por qué elegí este tema?	2
3.- Tablero SQL.....	3
4.- Tablero CSV	6

1.-Tema

Este reporte va dedicado a todas esas empresas que se dedican a la gestión y producciones de eventos, ya sen producciones de audio, visual etc.

2.- ¿Por qué elegí este tema?

Este tema fue elegido ya que yo me dedico a este tipo de trabajos y al tener una herramienta donde se pueda llevar a cabo una mejor logística de los eventos a realizar garantiza un mejor trabajo de una manera más eficiente y eficaz.

3.- Tablero SQL

La siguiente consulta ya contiene un “order by” y “group by”

```

1  SELECT
2      c.nombre AS Cliente,
3      pro.nombreproduccion AS Produccion,
4      pro.costo,
5      ev.fecha,
6      MAX(u.nombreubicacion) AS Ubicacion,
7      MAX(te.nombretipoevento) AS Fiesta
8  FROM
9      clientes AS c
10 INNER JOIN
11     eventos AS ev ON ev.idcliente = c.idcliente
12 INNER JOIN
13     producciones AS pro ON pro.idproduccion = ev.idproduccion
14 INNER JOIN
15     ubicaciones AS u ON u.idubicacion = ev.idubicacion
16 INNER JOIN
17     tipoeventos AS te ON te.idtipoevento = ev.idtipoevento
18 GROUP BY
19     c.nombre,
20     pro.nombreproduccion,
21     pro.costo,
22     ev.fecha
23 ORDER BY
24     c.nombre;
25 """

```

Ilustración 1 consulta SQL

dfconsulta						
	Cliente	Produccion	costo	fecha	Ubicacion	Fiesta
0	carla	dj	12000.0	2024-08-24	cholula	graduaciones
1	carla	dj	12000.0	2024-10-26	puebla	boda
2	carla	pista	8000.0	2024-10-19	cholula	xv años
3	juan	dj	12000.0	2024-08-24	puebla	boda
4	laura	grupo	35000.0	2024-08-31	cholula	xv años
5	liz	dj	12000.0	2024-09-12	atlixco	graduaciones
6	liz	pista	8000.0	2024-09-07	atlixco	graduaciones
7	pepe	grupo	35000.0	2024-09-14	cholula	xv años
8	pepe	grupo	35000.0	2024-12-27	atlixco	boda

Ilustración 2 resultado de consulta

La siguiente consulta ya contiene un “order by” y “group by” y 3 “where”

```

1  SELECT
2      c.nombre AS Cliente,
3      pro.nombreproduccion AS Produccion,
4      pro.costo,
5      ev.fecha,
6      MAX(u.nombreubicacion) AS Ubicacion,
7      MAX(te.nombretipoevento) AS Fiesta
8  FROM
9      clientes AS c
10 INNER JOIN
11     eventos AS ev ON ev.idcliente = c.idcliente
12 INNER JOIN
13     producciones AS pro ON pro.idproduccion = ev.idproduccion
14 INNER JOIN
15     ubicaciones AS u ON u.idubicacion = ev.idubicacion
16 INNER JOIN
17     tipoeventos AS te ON te.idtipoevento = ev.idtipoevento
18 GROUP BY
19     c.nombre,
20     pro.nombreproduccion,
21     pro.costo,
22     ev.fecha
23 ORDER BY
24     c.nombre;

```

Ilustración 3 consulta SQL con where

dfwhere					
	Cliente	Produccion	fecha	Ubicacion	Fiesta
0	carla	pista	2024-10-19	cholula	xv años

Ilustración 4 resultado de consulta

Estas son las líneas necesarias para crear una gráfica de pastel, teniendo como etiquetas producción y ubicación y como datos numéricos la cantidad que hay de cada uno

```
dfconsulta["Etiqueta"] = dfconsulta["Produccion"] + ' - ' + dfconsulta["Ubicacion"]
data = dfconsulta.groupby('Etiqueta')['costo'].sum()
explode = [0.1] * len(data)

plt.pie(data, labels=data.index, autopct="%0.1f %%", explode=explode)
plt.axis("equal")
plt.show()
```

Ilustración 5 líneas para creación de gráfica

Se observa de mejor manera en la siguiente gráfica

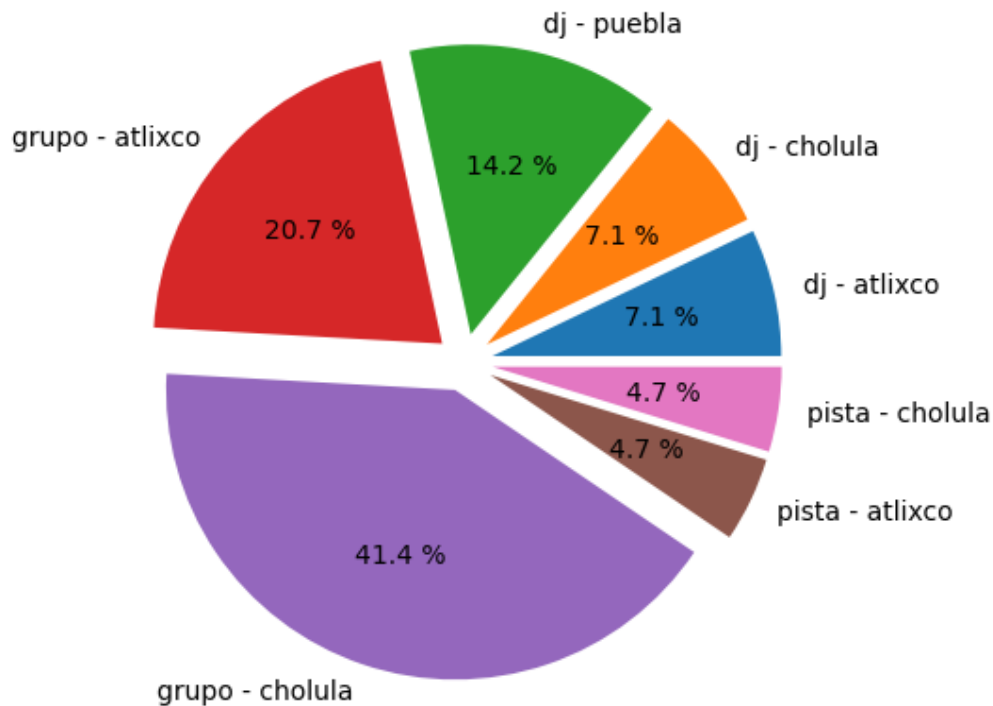


Ilustración 6 gráfica de pastel

4.- Tablero CSV

```
dfeventos=pd.read_csv(r'C:\Users\Genady\Documents\eventos\eventos.csv',encoding='utf-8',delimiter=',',nrows=10)
dfclientes=pd.read_csv(r'C:\Users\Genady\Documents\eventos\clientes.csv',encoding='utf-8',delimiter=',',nrows=10)
```

Ilustración 7 importación de archivos csv

	idevento	evento	costo	status
0	1	boda	35000	pagado
1	2	xv años	25000	adeudo
2	3	cumpleaños	20000	adueudo
3	4	graduacion	45000	pagado
4	5	bautizo	12000	adeudo

Ilustración 8 contenido de dfeventos

	idclientes	idevento	cantidad	nombre
0	1	1	1	juan
1	2	2	2	liz
2	3	3	4	lau
3	4	4	2	juan
4	5	5	5	liz
5	6	1	1	lau
6	7	2	1	juan
7	8	3	1	liz
8	9	3	1	lau
9	10	3	1	liz

Ilustración 9 contenido de dfclientes

Aquí se renombran algunos atributos de nuestro df mediante la instrucción "rename"

```
dfeventos.rename(columns={"costo":"precio","status":"estado"},inplace = True)
dfeventos
```

	idevento	evento	precio	estado
0	1	boda	35000	pagado
1	2	xv años	25000	adeudo
2	3	cumpleaños	20000	adeudo
3	4	graduacion	45000	pagado
4	5	bautizo	12000	adeudo

Ilustración 10 Renombramiento de atributos de dfeventos

```
dfclientes.rename(columns={"nombre":"razonsocial","cantidad":"cantidades"},inplace = True)
dfclientes
```

	idclientes	idevento	cantidades	razonsocial
0	1	1	1	juan
1	2	2	2	liz
2	3	3	4	lau
3	4	4	2	juan
4	5	5	5	liz
5	6	1	1	lau
6	7	2	1	juan
7	8	3	1	liz
8	9	3	1	lau
9	10	3	1	liz

Ilustración 11 Renombramiento de atributos de dfclientes

Aquí se unen ambos df mediante la instrucción “Merge”

```
dfdetalleventa = dfeventos.merge(dfclientes, on = "idevento", how = "inner")
dfdetalleventa
```

	idevento	evento	precio	estado	idclientes	cantidades	razonsocial
0	1	boda	35000	pagado	1	1	juan
1	1	boda	35000	pagado	6	1	lau
2	2	xv años	25000	adeudo	2	2	liz
3	2	xv años	25000	adeudo	7	1	juan
4	3	cumpleaños	20000	adueudo	3	4	lau
5	3	cumpleaños	20000	adueudo	8	1	liz
6	3	cumpleaños	20000	adueudo	9	1	lau
7	3	cumpleaños	20000	adueudo	10	1	liz
8	4	graduacion	45000	pagado	4	2	juan
9	5	bautizo	12000	adeudo	5	5	liz

Ilustración 12 Merge de ambos data frames

Aquí se usa la instrucción “groupby” y se puede observar que eventos hay y su estado, pagado o adeudo

```
dfdetalleventa.groupby("evento")["estado"].agg("sum")
```

```
evento
bautizo          adeudo
boda             pagadopagado
cumpleaños    adueudoadueudoadueudoadueudo
graduacion             pagado
xv años          adeudoadeudo
Name: estado, dtype: object
```

Ilustración 13 instrucción “group by”

Se crea una gráfica de barras con los atributos “evento” y “estado”
(se trabaja con el df que une a dos df mediante la instrucción “merge”)

Se puede observar el tipo de evento y su estado, de esta manera podemos determinar cuántos eventos están siendo pagados.

```
plt.bar(dfdetalleventa["evento"],dfdetalleventa["estado"])  
plt.show()
```

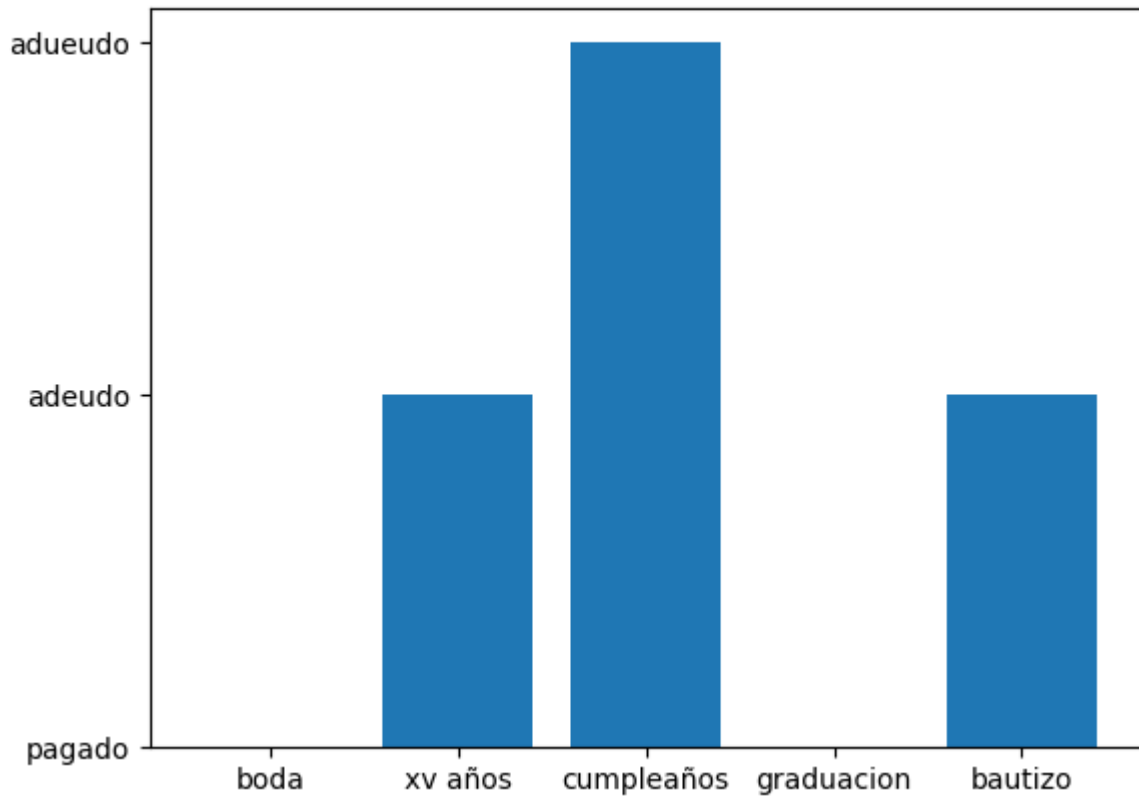


Ilustración 14 gráfica de barras

En esta parte se trato de darles un orden por el atributo “razónsocial”, no existe como la instrucción “orderby”, pero se encontró su equivalencia

```
dfdetalleventa_sorted = dfdetalleventa.sort_values(by="razonsocial")
dfdetalleventa_sorted
```

	idevento	evento	precio	estado	idclientes	cantidades	razonsocial
0	1	boda	35000	pagado	1	1	juan
3	2	xv años	25000	adeudo	7	1	juan
8	4	graduacion	45000	pagado	4	2	juan
1	1	boda	35000	pagado	6	1	lau
4	3	cumpleaños	20000	adueudo	3	4	lau
6	3	cumpleaños	20000	adueudo	9	1	lau
2	2	xv años	25000	adeudo	2	2	liz
5	3	cumpleaños	20000	adueudo	8	1	liz
7	3	cumpleaños	20000	adueudo	10	1	liz
9	5	bautizo	12000	adeudo	5	5	liz

Ilustración 15 df ordenado

Aquí tampoco existe como tal la instrucción ‘Where’, pero se encontró su equivalencia.

```
df_filtrado = dfdetalleventa[
    (dfdetalleventa['evento'] == 'bautizo') &
    (dfdetalleventa['estado'] == 'adeudo') &
    (dfdetalleventa['razonsocial'] == 'liz')
]
df_filtrado
```

	idevento	evento	precio	estado	idclientes	cantidades	razonsocial
9	5	bautizo	12000	adeudo	5	5	liz

Ilustración 16 df filtrado