

```
#!pip install psycopg2
!pip install --user psycopg2-binary
!pip install --user seaborn
!pip install --user pandas
```

```
import psycopg2
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
/usr/local/lib/python3.6/dist-packages/psycopg2/__init__.py:144: UserWarning: The psy
    """)
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarnin
    import pandas.util.testing as tm
```

```
conn = psycopg2.connect(host="tuffi.db.elephantsql.com", database="vzgqijrz", user="vzgqijr
```

## ▼ Relatório 1

**Objetivo: Obter os dados dos usuario, as datas em que foram compradas e os valores das compras.**

```
compras_usuario = pd.read_sql_query("""
    Select usuario.*,
    date_part('month', compra.data_compra)::int as mes,
    produto.preco*compra.quantidade as valor_compra
    From usuario
    Inner Join compra
        On usuario.id_usuario = compra.id_usuario
    Inner Join produto
        On produto.id_produto = compra.id_produto
    Order By mes, usuario.nome

    """, conn)
```

```
compras_usuario
```



```
(select avg(preco*quantidade) from produto inner join compra on produto.id_produto = compra.id_produto  
(select ROUND(stddev(preco*quantidade)::numeric,2) from produto inner join compra on produto.id_produto = compra.id_produto  
from compra  
inner join produto p  
on p.id_produto = compra.id_produto  
order by data_compra  
""",conn)
```

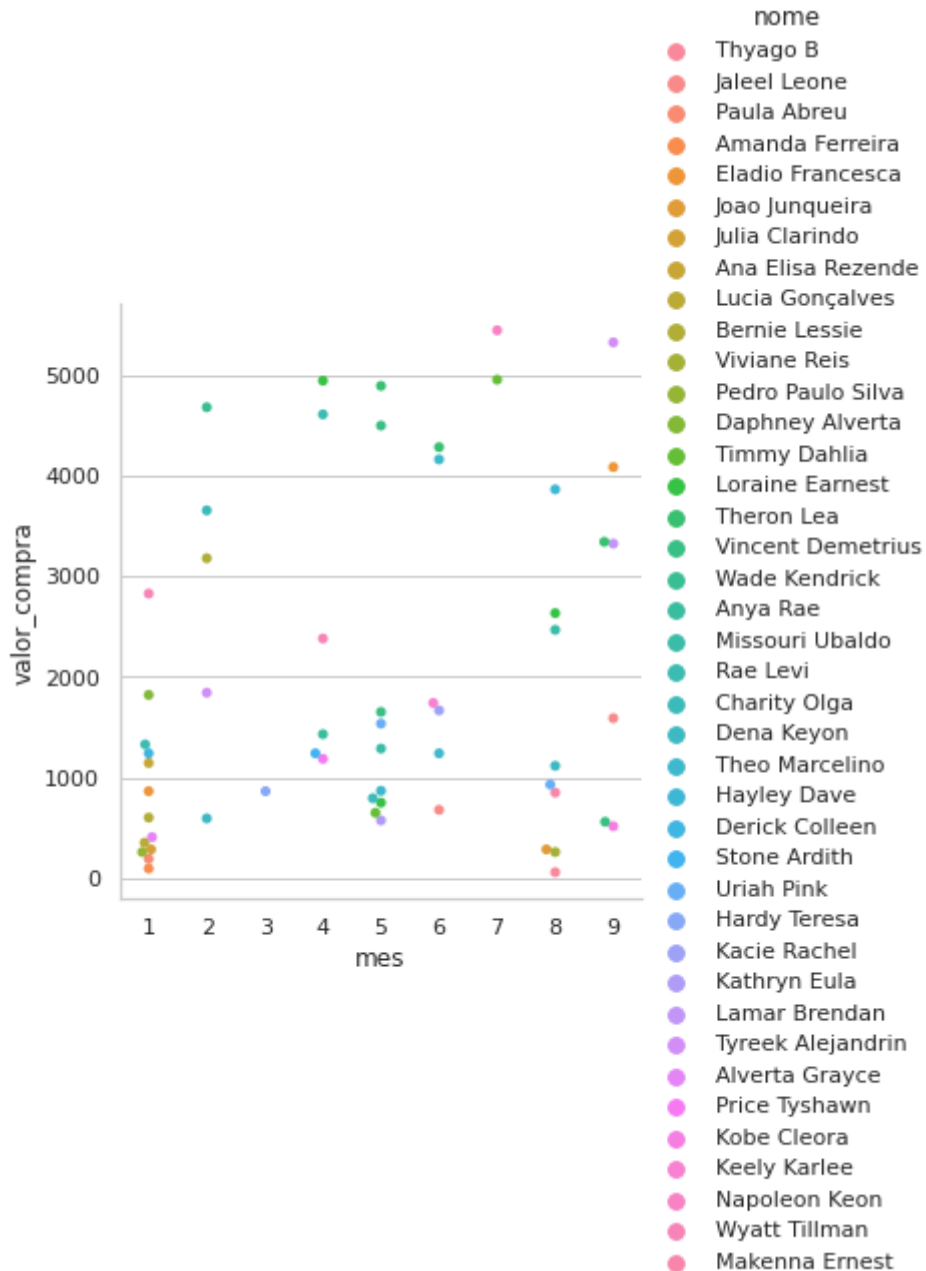
analise\_compras

	id_usuario	nome	cpf	email	senha	est:
0	27	Alverta Grayce	16126	lemke.oma@nolan.com	feil.roberta	
1	51	Amanda Ferreira	12345	amanda@gmail.com	123@456	
2	52	Ana Elisa Rezende	12120	anaelisa@gmail.com	123@456	
3	18	Bernie Lessie	64482	tschumm@wintheiser.com	jaylan.cassin	
4	39	Daphney Alverta	49501	willow.spinka@hotmail.com	hkeebler	
5	17	Derick Colleen	64537	ferne51@hotmail.com	harrison.ryan	
6	16	Eladio Francesca	24148	qanderson@doyle.com	martina99	
7	55	Julia Clarindo	81254	julia@gmail.com	123@456	
8	54	Lucia Gonçalves	78789	lucia@gmail.com	123@456	
9	56	Paula Abreu	96547	paula@gmail.com	123@456	
10	53	Pedro Paulo Silva	99999	pedro@gmail.com	123@456	
11	20	Rae Levi	59368	pgerlach@sawayn.com	flittle	
12	3	Wyatt Tillman	69557	cleve03@robel.com	jayda.collier	
13	18	Bernie Lessie	64482	tschumm@wintheiser.com	jaylan.cassin	
14	37	Charity Olga	26320	shayna.zulauf@yahoo.com	iframi	
15	14	Dena Keyon	38438	rosalind03@gmail.com	christelle.roberts	
16	45	Tyreek Alejandrin	30196	cristian87@bergstrom.com	litzy.collins	
17	11	Wade Kendrick	39256	rframi@tromp.net	kiana88	
18	1	Hardy Teresa	48173	rollin.bradtke@witting.com	baumbach.imogene	
19	31	Anya Rae	24310	orosenbaum@hotmail.com	eduardo.oconnell	
20	48	Loraine	38376	dhowe@ratke.biz	lerdman	

		Earnest			
21	50	Price Tyshawn	54557	elena.ortiz@gmail.com	cale52
22	20	Rae Levi	59368	pgerlach@sawayn.com	flittle
23	49	Stone Ardith	17522	acrona@jaskolski.com	summer28
24	3	Wyatt Tillman	69557	cleve03@robels.com	jayda.collier
25	37	Charity Olga	26320	shayna.zulauf@yahoo.com	iframi
26	24	Kathryn Eula	24430	cummings.berneice@jenkins.com	ikutch
27	48	Loraine Earnest	38376	dhowe@ratke.biz	lerdman
28	7	Missouri Ubaldo	18277	thompson.emma@yahoo.com	mandy02
29	35	Theo Marcelino	30095	muller.destiney@gmail.com	davis.roman
30	13	Theron Lea	34951	margarita.quitzon@murphy.com	armstrong.vicenta
31	44	Timmy Dahlia	27062	ruthe39@schuppe.com	aleannon
32	30	Uriah Pink	50452	yfay@hotmail.com	erling.moore
33	34	Vincent Demetrius	12503	shakira.gerlach@gmail.com	carole.kessler
34	34	Vincent Demetrius	12503	shakira.gerlach@gmail.com	carole.kessler
35	5	Jaleel Leone	25282	andreane15@hotmail.com	brooklyn94
36	33	Kacie Rachel	33472	q Dickinson@paucek.com	diego.stanton
37	2	Keely Karlee	24836	bosco.lukas@koelpin.com	zion96
38	35	Theo Marcelino	30095	muller.destiney@gmail.com	davis.roman
39	35	Theo Marcelino	30095	muller.destiney@gmail.com	davis.roman
40	13	Theron Lea	34951	margarita.quitzon@murphy.com	armstrong.vicenta
41	23	Napoleon Keon	63237	daron73@gmail.com	lwiegand
42	44	Timmy	67888		

42	44	Dahlia	27062	ruthe39@schuppe.com	aleannon
43	14	Dena Keyon	38438	rosalind03@gmail.com	christelle.roberts
44	46	Hayley Dave	64529	sam61@yahoo.com	bkutch

```
g = sns.catplot(x='mes', y='valor_compra', hue='nome', kind='swarm', dodge=False, data=com
```



## ▼ Relatório 2

**Objetivo: obter a distribuição dos usuário por município.**

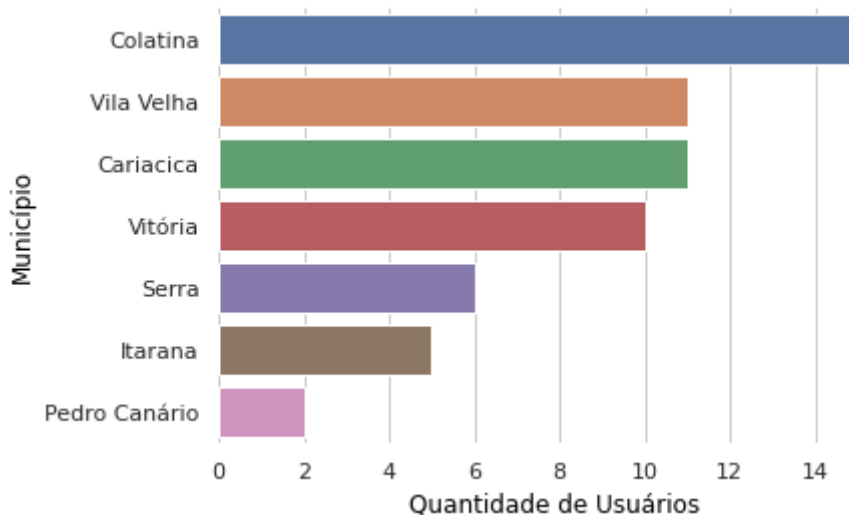
```
usuario_por_municipio = pd.read_sql_query("""
    Select municipio, count(id_usuario) as qnt_usuarios
    From usuario
    Group by municipio
```

```
order by qnt_usuarios desc
""",conn)
```

```
usuario_por_municipio
```

	municipio	qnt_usuarios
0	Colatina	15
1	Vila Velha	11
2	Cariacica	11
3	Vitória	10
4	Serra	6
5	Itarana	5
6	Pedro Canário	2

```
sns.set(style="whitegrid")
sns.set_color_codes("pastel")
g = sns.barplot(x='qnt_usuarios', y='municipio', data=usuario_por_municipio)
g.set_xlabel("Quantidade de Usuários", size = 12,color="black")
g.set_ylabel("Município", size = 12,color="black")
sns.despine(left=True, bottom=True)
```



## ▼ Relatório 3

**Objetivo: obter a receita mensal das vendas da loja.**

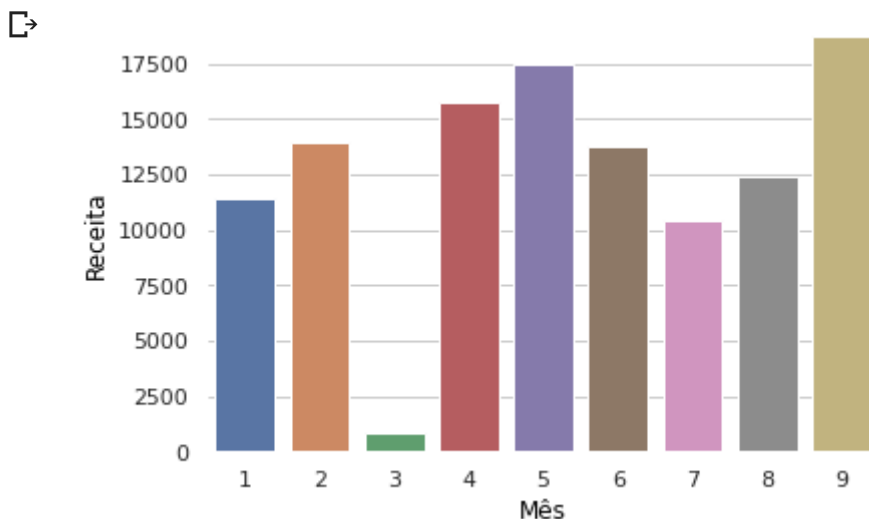
```
receita_por_mes = pd.read_sql_query("""
Select date_part('month',compra.data_compra)::int as mes, SUM (compra.quantidade * produ
From compra
Inner Join produto
On compra.id_produto = produto.id_produto
```

```
Group by mes
Order by mes asc
""",conn)
```

receita\_por\_mes

	mes	receita
0	1	11419.0
1	2	13941.0
2	3	864.0
3	4	15780.0
4	5	17491.0
5	6	13763.0
6	7	10390.0
7	8	12461.0
8	9	18729.0

```
g = sns.barplot(x='mes',y='receita',data=receita_por_mes)
g.set_xlabel("Mês", size = 12,color="black")
g.set_ylabel("Receita", size = 12,color="black")
sns.despine(left=True, bottom=True)
```



## ▼ Relatório 4

**Objetivo: obter os valores das compras e compará-las com a média geral de compras**

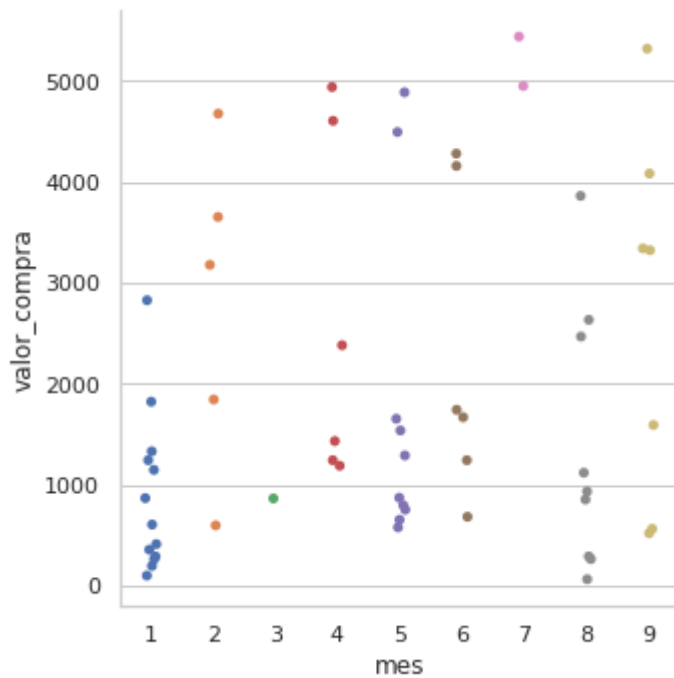
```
analise_compras = pd.read_sql_query("""
select date_part('month', compra.data_compra)::int as mes,
preco*quantidade as valor_compra,
(select avg(preco*quantidade) from produto inner join compra on produto.id_produto = com
https://colab.research.google.com/drive/1msz4l6z71WKYy4OJb8KywoxL5IVRRjOm#scrollTo=5xgJwj2jMVKX&printMode=true
```

	mes	valor_compra	media_compras	desvio_padrao_compras
0	1	864.0	1946.40678	1620.05
1	1	1328.0	1946.40678	1620.05
2	1	96.0	1946.40678	1620.05
3	1	260.0	1946.40678	1620.05
4	1	192.0	1946.40678	1620.05
5	1	286.0	1946.40678	1620.05
6	1	1144.0	1946.40678	1620.05
7	1	353.0	1946.40678	1620.05
8	1	408.0	1946.40678	1620.05
9	1	1820.0	1946.40678	1620.05
10	1	1240.0	1946.40678	1620.05
11	1	2826.0	1946.40678	1620.05
12	1	602.0	1946.40678	1620.05
13	2	1842.0	1946.40678	1620.05
14	2	3177.0	1946.40678	1620.05
15	2	594.0	1946.40678	1620.05
16	2	4676.0	1946.40678	1620.05
17	2	3652.0	1946.40678	1620.05
18	3	864.0	1946.40678	1620.05
19	4	4940.0	1946.40678	1620.05
20	4	1185.0	1946.40678	1620.05
21	4	1430.0	1946.40678	1620.05
22	4	4605.0	1946.40678	1620.05
23	4	2380.0	1946.40678	1620.05
24	4	1240.0	1946.40678	1620.05
25	5	1651.0	1946.40678	1620.05
26	5	574.0	1946.40678	1620.05
27	5	750.0	1946.40678	1620.05
28	5	4494.0	1946.40678	1620.05
29	5	794.0	1946.40678	1620.05

```
tips = sns.load_dataset("tips")
sns.catplot(x="mes", y="valor_compra", data=analise_compras)
```



&lt;seaborn.axisgrid.FacetGrid at 0x7f1825603470&gt;



## ▼ Relatório 5

**Objetivo: Obter a valor total proveniente das compras realizadas por cada município.**

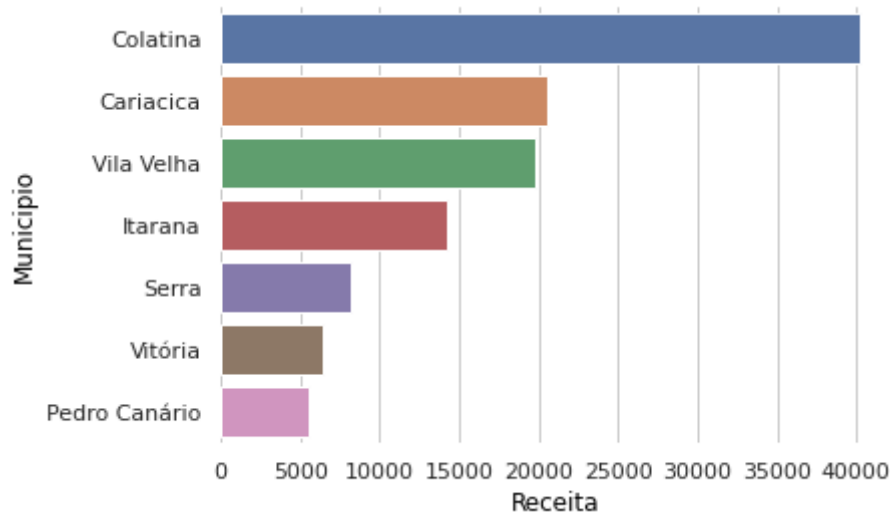
```
pesquisa = pd.read_sql_query("""
select usuario.municipio, count(compra.id_usuario) as qtde_compras, sum(produto.preco * co
from usuario
inner join compra
on (usuario.id_usuario = compra.id_usuario)
inner join produto
on (compra.id_produto = produto.id_produto)
group by usuario.municipio
order by receita desc
""",conn)
```

```
55      0      2240.0      4040.40670      4000.05
pesquisa
```

	municipio	qtde_compras	receita
0	Colatina	14	40207.0
1	Cariacica	12	20584.0
2	Vila Velha	13	19784.0
3	Itarana	7	14248.0
4	Serra	5	8150.0
5	Vitória	4	6376.0
6	Pedro Canário	4	5489.0

```
sns.set(style="whitegrid")
```

```
sns.set_color_codes("pastel")
g = sns.barplot(x='receita', y='municipio', data=pesquisa)
g.set_xlabel("Receita", size = 12,color="black")
g.set_ylabel("Município", size = 12,color="black")
sns.despine(left=True, bottom=True)
```



## ▼ Relatório 6

**Objetivo: Obter a quantidade de vezes que cada produto foi comprado.**

```
quantidade = pd.read_sql_query("""
select nome,preco,count(*) as quantas_vezes_comprado
from produto
inner join compra
on compra.id_produto = produto.id_produto
group by nome,preco
order by quantas_vezes_comprado
""",conn)
```

quantidade



	nome	preco	quantas_vezes_comprado
0	Calça	302.0	1
1	Camalha	237.0	1
2	Collant	145.0	1
3	Anoraque	60.0	1
4	Bustiê	41.0	1
5	Smoking	97.0	1
6	Avacote	227.0	1
7	Conjunto	274.0	1
8	Camal	34.0	1
9	Burca	238.0	1
10	Calção	275.0	1
11	Blazer	125.0	1
12	Pantalonas	70.0	1
13	Mantilha	321.0	1
14	Blusa de lã	127.0	1
15	Brial	216.0	1
16	Blusa	214.0	1
17	Pano	397.0	1
18	Mantô	143.0	2
19	Meia-calça	332.0	2
20	Soquete	314.0	2
21	Calções	380.0	2
22	Boina	334.0	2
23	Cueca	124.0	2

```
g = sns.catplot(x='quantas_vezes_comprado', y='preco', hue='nome', kind='point', dodge=False)
```



## ▼ Relatório 7

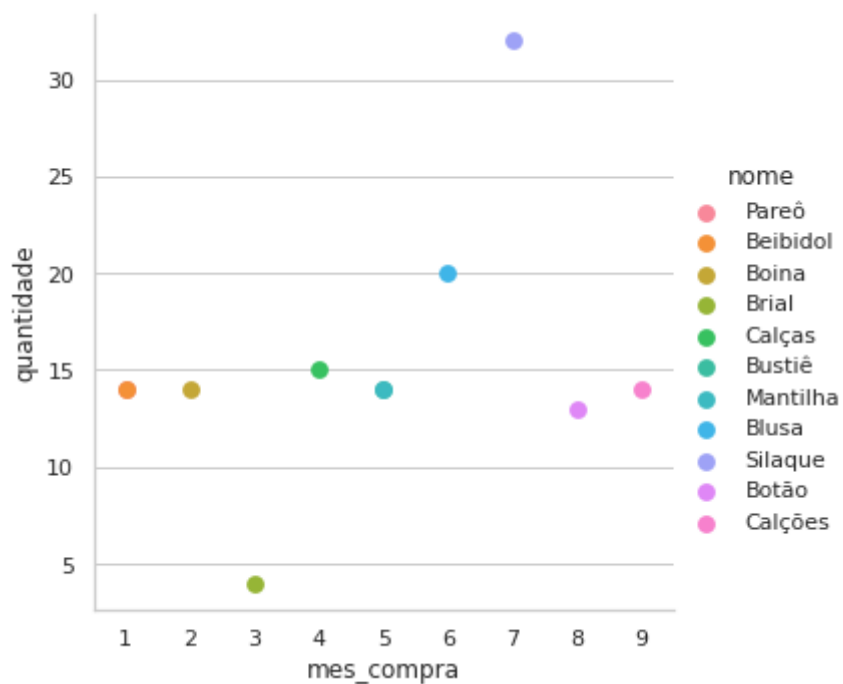
**Objetivo: obter os produtos mais vendidos de cada mês**

```
produtos_mais_vendidos = pd.read_sql_query("""
Select date_part('month',compra.data_compra)::int mes_compra,
       produto.nome,
       compra.quantidade,
       compra.quantidade * produto.preco as valor_total
From compra
Inner Join produto
  On compra.id_produto = produto.id_produto
Inner Join (
  Select date_part('month',b.data_compra)::int datab, MAX(quantidade) quantidade
  From compra As b
  Group By date_part('month',b.data_compra)::int
) As b
  On compra.quantidade = b.quantidade and b.datab = date_part('month',compra.data_compra):
Order By mes_compra
""",conn)

produtos_mais_vendidos
```

	mes_compra	nome	quantidade	valor_total
0	1	Pareô	14	602.0
1	1	Beibidol	14	1820.0
2	2	Boina	14	4676.0
3	3	Brial	4	864.0
4	4	Calças	15	4605.0

```
g = sns.catplot(x='mes_compra', y='quantidade', hue='nome', kind='point', dodge=False, dat
```



```
conn.close()
```