SQL EDA: customers & marketing campaigns (EN)

Introduction

In this project we'll be analyzing a table containing data related to the customers and their reaction to the 5 marketing campaigns. The data in the dataset has been generated for the purpose of learning and is mock.

Using SQL (and a tiny bit of Python) we'll perfrom an exploratory data analysis and identify groups of clients most and least perceptive to marketing campaigns.

Data Source

The dataset is available publicly on Kaggle.com at this link

Schema Code

▼ Table creation

```
CREATE TABLE campaigns
(user_id    INT PRIMARY KEY,
year_birth  INT,
education   VARCHAR(512),
marital_status  VARCHAR(512),
income  FLOAT,
kidhome INT,
teenhome    INT,
dt_enrolled DATE,
recency INT,
amt_wine    INT,
amt_fruit   INT,
amt_meat    INT,
amt_fish    INT,
amt_sweet   INT,
amt_gold    INT,
discount_purchases  INT,
```

```
web_purchases   INT,
catalog_purchases   INT,
store_purchases INT,
web_visits_mnth INT,
cmp1_accepted   INT,
cmp2_accepted   INT,
cmp3_accepted   INT,
cmp4_accepted   INT,
cmp5_accepted   INT,
complain    INT,
response    INT
);
```

▼ Schema

campaigns	
user_id	integer
year_birth	integer
education	varchar(512)
marital_status	varchar(512)
income	float
kidhome	integer
teenhome	integer
dt_enrolled	date
recency	integer
amt_wine	integer
amt_fruit	integer
amt_meat	integer
amt_fish	integer
amt_sweet	integer
amt_gold	integer
discount_purchases	integer
web_purchases	integer
catalog_purchases	integer
store_purchases	integer
web_visits_mnth	integer
cmp1_accepted	integer
cmp2_accepted	integer
cmp3_accepted	integer
cmp4_accepted	integer
cmp5_accepted	integer
complain	integer
response	integer

Data Description

- **user_id**: unique identifier of the customer
- **year_birth**: customer's birth year
- **education**: customer's education level
- **marital_status**: customer's marital status
- **income**: customer's yearly household income
- **kidhome**: number of children in customer's household
- **teenhome**: number of teenagers in customer's household
- **dt_enrolled**: date of customer's enrollment with the company's loyalty program
- **recency**: number of days since customer's last purchase
- **complain**: 1 if the customer complained in the last 2 years, 0 otherwise
- **amt_wine**: amount spent on wine in last 2 years
- **amt_fruit**: amount spent on fruits in last 2 years
- **amt_meat**: amount spent on meat in last 2 years
- **amt_fish**: amount spent on fish in last 2 years
- **amt_sweet**: amount spent on sweets in last 2 years
- **amt_gold**: amount spent on golden products (premium category) in the last 2 years
- **discount_purchases**: number of purchases made with a discount
- **web_purchases**: number of purchases made through the company's website
- **catalog_purchases**: number of purchases made using a catalogue
- **store_purchases**: number of purchases made directly in stores
- **web_visits_mnth**: number of visits to company's website in the last month
- **cmp1_accepted**: 1 if customer accepted the offer in the 1st campaign, 0 otherwise
- **cmp2_accepted**: 1 if customer accepted the offer in the 2nd campaign, 0 otherwise
- **cmp3_accepted**: 1 if customer accepted the offer in the 3rd campaign, 0 otherwise
- **cmp4_accepted**: 1 if customer accepted the offer in the 4th campaign, 0 otherwise
- **cmp5_accepted**: 1 if customer accepted the offer in the 5th campaign, 0 otherwise
- **response**: 1 if customer accepted at least 1 offer

Initial Data Assessment

Creating & Editing Fields

First of all, let's create new columns based on the data we have:

1. Create **age** column using **year_birth.**

```
ALTER TABLE campaigns
ADD COLUMN age INTEGER;
```

```
UPDATE campaigns
SET age = DATE_PART('YEAR', NOW()) - year_birth;
```

2. **total_kids** will store the total number of kids.

```
ALTER TABLE campaigns
ADD COLUMN total_kids INTEGER;
```

```
UPDATE campaigns
SET total_kids = kidhome + teenhome;
```

3. **total_spent** column will be storing total spendings.

```
ALTER TABLE campaigns
ADD COLUMN total_spent INT;
```

```
UPDATE campaigns
SET total_spent = amt_wine + amt_fruit + amt_meat + amt_fish + amt_sweet + amt_g
old;
```

4. **total_purchases** will reflect the number of purchases.

```
ALTER TABLE campaigns
ADD COLUMN total_purchases INT;
```

```
UPDATE campaigns
SET total_purchases = discount_purchases + web_purchases + store_purchases;
```

5. **marital_status** contains some ambiguous and/or duplicate values, let's override them.

```
UPDATE campaigns
SET marital_status = 'Single'
WHERE marital_status IN ('YOLO', 'Absurd', 'Alone');
```

7. There are some records with unlikely values related to age, we'll drop them.

```
DELETE FROM campaigns
WHERE age > 100;
```

Analyzing by categories

Average income and total spendings by education level and marital status:

```
SELECT education, marital_status,
ROUND(AVG(income)::NUMERIC, 2) AS avg_income,
SUM(total_spent) AS total_spendings
FROM campaigns
GROUP BY education, marital_status
ORDER BY total_spendings DESC;
```

	education character varying (512)	marital_status character varying (512)	avg_income numeric	total_spendings bigint
1	Graduation	Married	50800.26	258030
2	Graduation	Together	55758.48	188468
3	Graduation	Single	51365.63	155075
4	PhD	Married	58138.03	137439
5	Master	Married	53286.03	78200
6	PhD	Together	55802.37	74146
7	Graduation	Divorced	54526.04	73353
8	PhD	Single	53039.67	61300
9	Master	Together	52109.01	59450
10	Master	Single	53787.14	57754

Clients with the ***highest spendings*** are married people who have already graduated (Graduation)

Clients with the ***highest income*** are people who have already graduated and are in a relationship (not married)

Average income and total spendings by number of children:

```
SELECT total_kids,
    ROUND(AVG(income)::NUMERIC, 2) AS avg_income,
    SUM(total_spent) AS total_spendings
FROM campaigns
GROUP BY total_kids
ORDER BY total_spendings DESC;
```

	total_kids integer	avg_income numeric	total_spendings bigint
1	0	65677.36	703794
2	1	47712.01	533156
3	2	44612.31	103544
4	3	46677.00	14554

Clients with the ***highest average income*** and ***total spendings*** are people without kids

Average income and total spendings by age group:

```
SELECT CASE WHEN age BETWEEN 20 AND 30 THEN '20-30'
    WHEN age BETWEEN 31 AND 40 THEN '30-40'
    WHEN age BETWEEN 41 AND 50 THEN '40-50'
    WHEN age BETWEEN 51 AND 60 THEN '50-60'
    WHEN age BETWEEN 61 AND 70 THEN '60-70'
    ELSE '70+'  END AS age_group,
    ROUND(AVG(income)::NUMERIC, 2) AS avg_income,
    SUM(total_spent) AS total_spendings
FROM campaigns
GROUP BY age_group
ORDER BY total_spendings DESC;
```

	age_group text	avg_income numeric	total_spendings bigint
1	50-60	51441.32	372625
2	60-70	56431.57	327972
3	40-50	49776.88	306047
4	70+	59002.95	198960
5	30-40	44707.29	141128
6	20-30	58295.40	8316

The leader by total spendings is the category 50-60 years, by average income - people of 20-30 years.

Most active web visitors

```
SELECT education, marital_status,
    SUM(web_visits_mnth) AS total_web_visits
FROM campaigns
GROUP BY education, marital_status
ORDER BY total_web_visits DESC;
```

	education character varying (512)	marital_status character varying (512)	total_web_visits bigint
1	Graduation	Married	2334
2	Graduation	Together	1481
3	Graduation	Single	1340
4	PhD	Married	1007
5	Master	Married	721
6	Graduation	Divorced	636
7	PhD	Together	620
8	Master	Together	540
9	PhD	Single	523
10	2n Cycle	Married	432

The ***most active visitors of the website*** in the last month are married people who have already graduated

Most popular shopping method

```
SELECT CASE WHEN age BETWEEN 20 AND 30 THEN '20-30'
    WHEN age BETWEEN 31 AND 40 THEN '30-40'
    WHEN age BETWEEN 41 AND 50 THEN '40-50'
    WHEN age BETWEEN 51 AND 60 THEN '50-60'
    WHEN age BETWEEN 61 AND 70 THEN '60-70'
    ELSE '70+'  END AS age_group,
    SUM(discount_purchases) AS discount_pur_sum,
    SUM(web_purchases) AS web_pur_sum,
    SUM(catalog_purchases) AS catalog_pur_sum,
    SUM(store_purchases) AS store_pur_sum
FROM campaigns
GROUP BY age_group;
```

	age_group text	discount_pur_sum bigint	web_pur_sum bigint	catalog_pur_sum bigint	store_pur_sum bigint
1	70+	561	1229	910	1747
2	30-40	429	824	587	1315
3	60-70	1205	2117	1492	3009
4	40-50	1327	2238	1336	3175
5	20-30	12	33	39	69
6	50-60	1671	2702	1592	3647

The ***most popular shopping method*** is right at the store regardless of the age group

Response rate

```
SELECT education,
    marital_status,
    total_kids,
    ROUND((SUM(response)::NUMERIC /COUNT(user_id)::NUMERIC), 2) AS response_rate
```

```
FROM campaigns
GROUP BY education, marital_status, total_kids
ORDER BY response_rate DESC;
```

	education character varying (512)	marital_status character varying (512)	total_kids integer	response_rate numeric
1	2n Cycle	Divorced	0	0.67
2	Master	Widow	0	0.60
3	PhD	Divorced	0	0.55
4	Master	Single	0	0.50
5	Master	Widow	1	0.50

People in the 2n cycle of education (this may mean either bachelor's or PhD depending on the country) are the ***most active responders to the campaigns***

Most popular product category by total spendings

```
SELECT *
FROM
(SELECT 'amt_wine' AS product_group,
    SUM(amt_wine) AS amt
FROM campaigns
UNION ALL
SELECT 'amt_fruit' AS product_group,
    SUM(amt_fruit) AS amt
FROM campaigns
UNION ALL
SELECT 'amt_meat' AS product_group,
    SUM(amt_meat) AS amt
FROM campaigns
UNION ALL
SELECT 'amt_fish' AS product_group,
    SUM(amt_fish) AS amt
FROM campaigns
UNION ALL
SELECT 'amt_sweet' AS product_group,
    SUM(amt_sweet) AS amt
FROM campaigns
UNION ALL
SELECT 'amt_gold' AS product_group,
    SUM(amt_gold) AS amt
FROM campaigns)
ORDER BY amt DESC;
```

	product_group text	amt bigint
1	amt_wine	680038
2	amt_meat	373393
3	amt_gold	98358
4	amt_fish	83939
5	amt_sweet	60553
6	amt_fruit	58767

The ***most popular category by total spendings*** is wine, the customers spent almost twice as much on wine as they did on the second most populat product - meat

Most successful campaigns

```
SELECT *
FROM
(SELECT 'campaign1' AS campaign,
    SUM(cmp1_accepted) AS cmp_acceptance
FROM campaigns
UNION ALL
SELECT 'campaign2' AS campaign,
    SUM(cmp2_accepted) AS cmp_acceptance
FROM campaigns
UNION ALL
SELECT 'campaign3' AS campaign,
    SUM(cmp3_accepted) AS cmp_acceptance
FROM campaigns
UNION ALL
SELECT 'campaign4' AS campaign,
    SUM(cmp4_accepted) AS cmp_acceptance
FROM campaigns
UNION ALL
SELECT 'campaign5' AS campaign,
    SUM(cmp5_accepted) AS cmp_acceptance
FROM campaigns)
ORDER BY cmp_acceptance DESC;
```

	campaign text	cmp_acceptance bigint
1	campaign4	167
2	campaign3	163
3	campaign5	162
4	campaign1	144
5	campaign2	30

The most successful campaign was the fourth one: 167 clients accepted an offer during this campaign

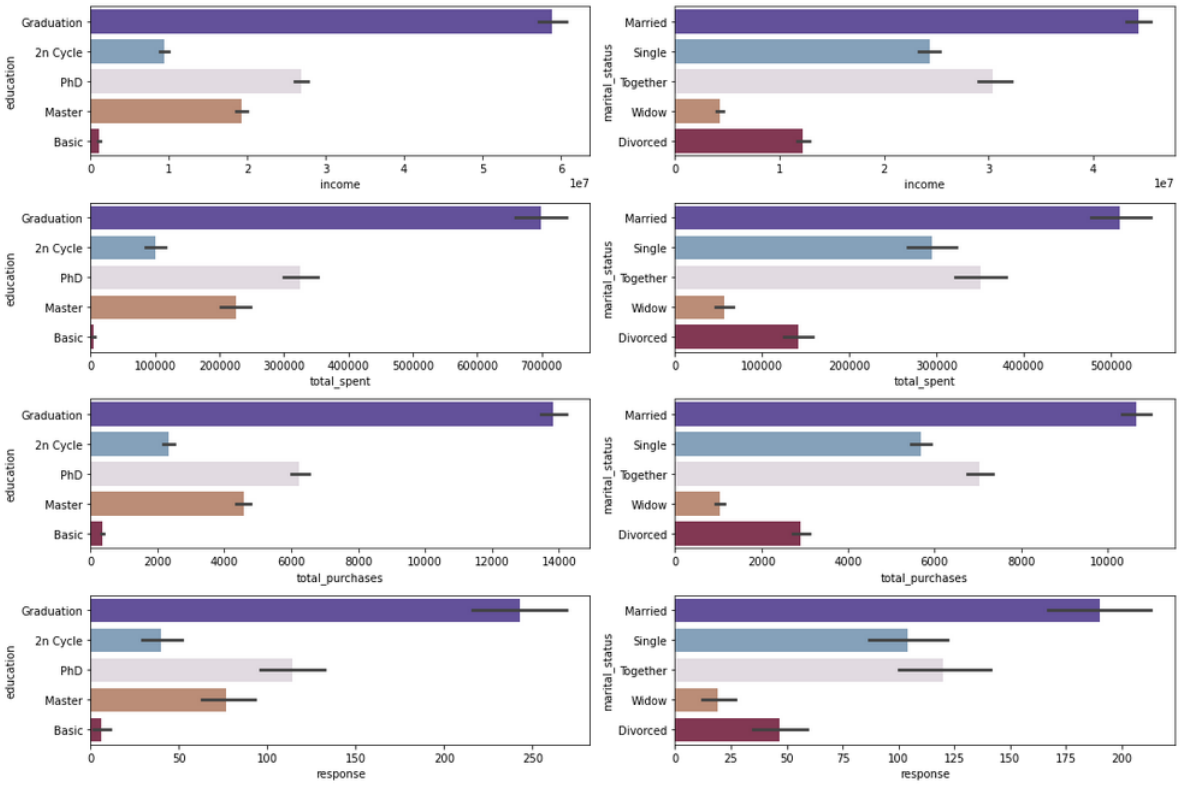
Visualization

To create some visuals we connect the database to Jupyter Notebook and load the data into a dataframe.

1. **Marital status, education vs Income, spendings, purchases and response**

▼ Code

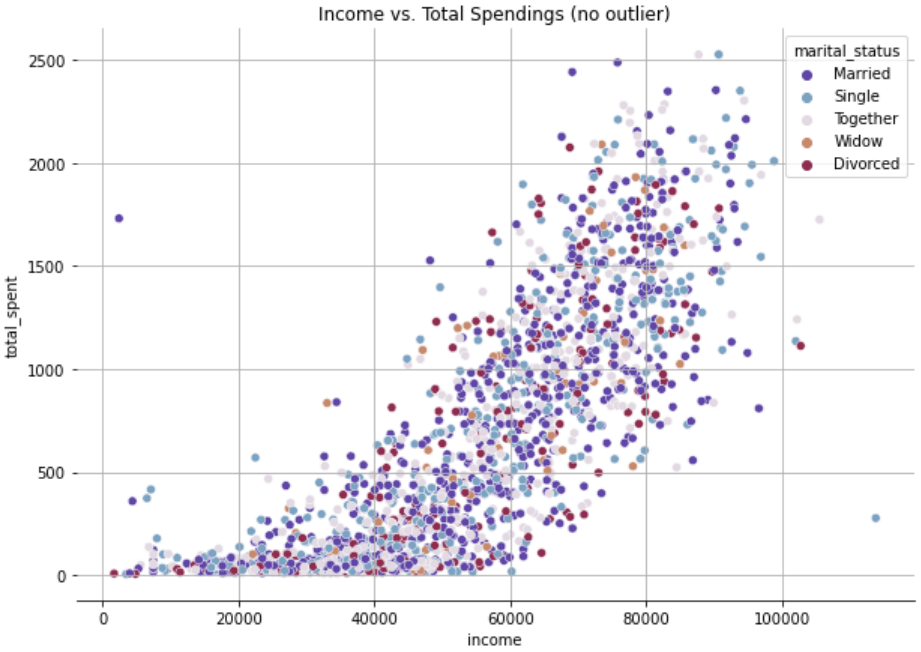
```
metrics = ['income', 'total_spent', 'total_purchases', 'response']
features = ['education', 'marital_status']

fig = plt.figure(figsize=(15, 10))
gs = GridSpec(ncols = 2, nrows = 4, figure = fig)
c = 0
r = 0



for feature in features:
    for i, metric in enumerate(metrics):
        grouped = df.groupby('education', as_index = False)[feature].sum().re
set_index()
        plt.subplot(gs[c, r])
        ax = sns.barplot(data = df, x = df[metric], y = df[feature], estimato
r = np.sum, palette = 'twilight_shifted')
        c+=1
    c=0
    r+=1



plt.tight_layout()
plt.show()
```



2. **Income vs. Total Spendings**

▼ Code

```
plt.figure(figsize=(10,7))
sns.scatterplot(data = df[df['income'] < df['income'].mean() + df['income'].s
td()*3], x = 'income', y = 'total_spent', hue = 'marital_status', palette =
'twilight_shifted')
plt.grid(True)
plt.title
sns.despine(left=True)
```



As income grows so do spendings

3. **Income vs. Total Purchases**

▼ Code

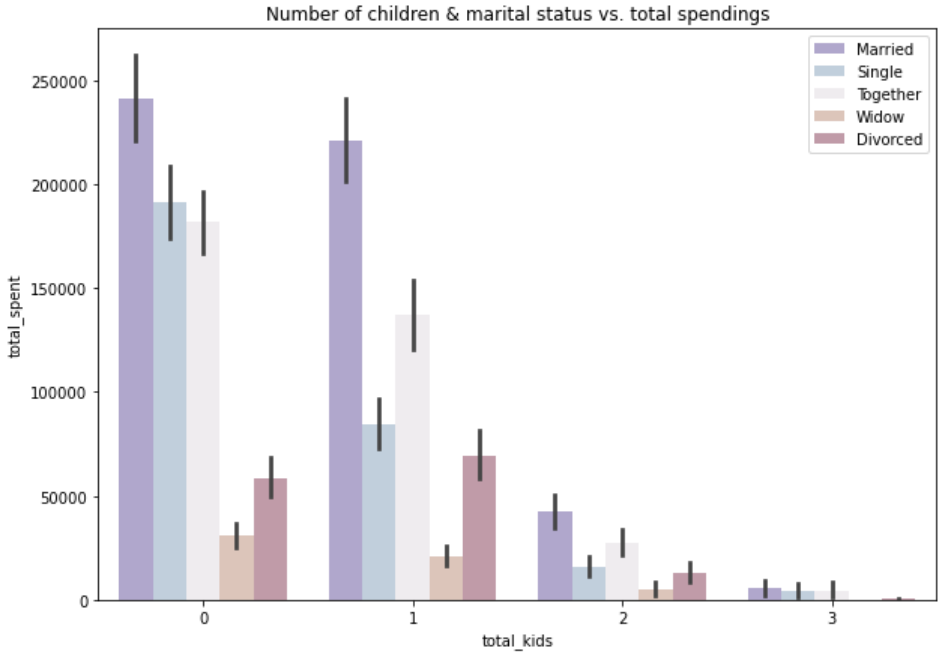
```
fig = plt.figure(figsize=(10,7))
ax = sns.scatterplot(data = df[df['income'] < df['income'].mean() + df['incom
e'].std()*3], x = 'income', y = 'total_purchases', hue = 'marital_status', pa
lette = 'twilight_shifted')
plt.grid(True)
sns.despine(left=True)
plt.title('Income vs. Total Purchases (no outlier)')
```

As income grows so does the number of purchases

4. **Number of children & marital status vs. Total Spendings**

▼ Code

```
fig = plt.figure(figsize = (10,7))
ax = sns.barplot(data = df, x = 'total_kids', y = 'total_spent', hue = 'marit
al_status', estimator = np.sum, palette = 'twilight_shifted', alpha = 0.5)
ax.legend(loc='upper right')
plt.title('Number of children & marital status vs. total spendings')
```

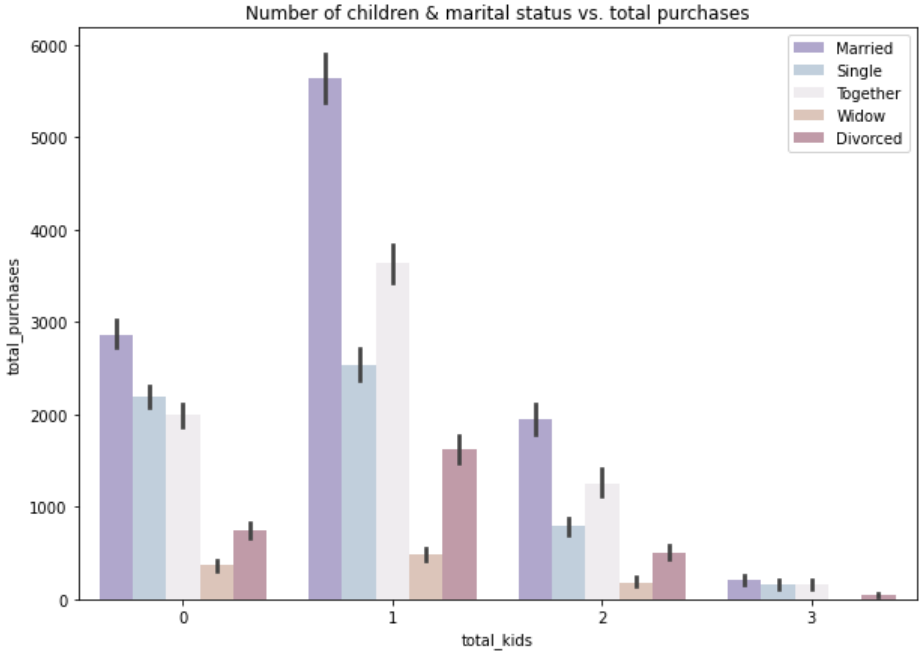


Married people with no children have the highest spendings

5. **Number of children & marital status vs. Total purchases**

▼ Code

```
fig = plt.figure(figsize = (10,7))
ax = sns.barplot(data = df, x = 'total_kids', y = 'total_purchases', hue = 'm
arital_status', estimator = np.sum, palette = 'twilight_shifted', alpha = 0.
5)
ax.legend(loc='upper right')
```



Married people with 1 kid have made the most purchases

Conclusion

We have analyzed the effectiveness of 5 marketing campaigns in regards to different categories of customers and their preferred methods of shopping and product types.

1. Campaign #5 proved to be the most effective with 167 client responses to it.
2. The most actively responding to the campaigns were divorced childless people in the 2n cycle (bachelor's or PhD depending on the country).
3. Client category with the highest average income and spendings are people without kids.
4. The most popular shopping method amongst all age groups is shopping at the store.
5. The most active web visitors in the last 2 months were married people who have already graduated.
6. The product category clients have spent most on in the last 2 years is wine.

It would be great to have more data on customers, their purchases and a detailed marketing campaigns dataset.

Links

Link to the full Jupyter Notebook (EN)

Link to the full Jupyter Notebook (RU)

Contacts

lianazaripovar@gmail.com

tg: @zaripova_liana