יונתן הרשקוביץ –205379373
מתן אשל -203502802

כריית וייצוג מידע – פרויקט

חלק ראשון

הקדמה

שלב 1 - Data information

ראשית, התקנו וייבאנו את כל הספריות הנדרשות לפרויקט, ולאחר מכן טענו את קובץ הcsv בעזרת ספריית panda (pd מעתה) לתוך משתנה df (dataFrame).

הרצנו פקודת print("Shape: ", df.shape) כדי להבין את סדר הגודל של מסד המידע שלנו. וקיבלנו שהוא בגודל של 1680 רשומות מידע, עם 29 פיצ'רים לכל אחד.

ביצענו פקודת head כדי להציג את 5 השורות הראשונות ולהבין קצת איך נראה המידע.

```
Shape:  (1680, 29)
```

Out[6]:

	ID	Year_Birth	Education	Status	Income	Num_of_kids	Num_of_Teen	Registration_date	Recency	Mnt_Fruits	...	Num_Web_Visits	Response_(
0	5376	1979.0	Graduation	Married	NaN	1.0	0.0	06/01/2013	42	1.0	...	2.0	
1	6862	1971.0	Graduation	Divorced	1730.0	0.0	0.0	18/05/2014	65	1.0	...	40.0	
2	10749	1991.0	Graduation	Single	8028.0	0.0	0.0	18/09/2012	62	73.0	...	38.0	
3	238	1967.0	2n Cycle	Together	67309.0	1.0	1.0	23/01/2013	76	515.0	...	14.0	
4	1501	1982.0	PhD	Married	160803.0	0.0	0.0	04/08/2012	21	55.0	...	0.0	

5 rows × 29 columns

כעת ביצענו אתה הפקודה הבאה כדי להבין בראייה מאקרו על המידע כולו:

```
In [9]:  df.info()
```

נשתמש בה פעמים רבות בהמשך בזמן הכנת התרגיל כדי לוודא אילו חוסרים עדיין קיימים:

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1680 entries, 0 to 1679
Data columns (total 30 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   ID                     1680 non-null   int64
 1   Year_Birth             1651 non-null   float64
 2   Education              1672 non-null   object
 3   Status                 1646 non-null   object
 4   Income                 1609 non-null   float64
 5   Num_of_kids            1672 non-null   float64
 6   Num_of_Teen            1660 non-null   float64
 7   Registration_date      1680 non-null   object
 8   Recency                1680 non-null   int64
 9   Mnt_Fruits             1673 non-null   float64
 10  Mnt_Meat               1673 non-null   float64
 11  Mnt_sweet              1659 non-null   float64
 12  Mnt_Wines              1673 non-null   float64
 13  Mnt_Gold_Products      1673 non-null   float64
 14  Mnt_Fish               1673 non-null   float64
 15  Num_Web_Purchases      1651 non-null   float64
 16  Num_Store_Purchases    1673 non-null   float64
 17  Num_Deals_Purchases    1673 non-null   float64
 18  Num_Catalog_Purchases  1673 non-null   float64
 19  Num_Web_Visits         1673 non-null   float64
 20  Response_Campaign_1    1662 non-null   float64
 21  Response_Campaign_2    1673 non-null   float64
 22  Response_Campaign_3    1673 non-null   float64
 23  Response_Campaign_4    1673 non-null   float64
 24  Response_Campaign_5    1673 non-null   float64
 25  Complain               1673 non-null   float64
 26  Cost_Contact           1673 non-null   float64
 27  Revenue                1673 non-null   float64
 28  Response               1680 non-null   int64
 29  Age                    1651 non-null   float64
dtypes: float64(24), int64(3), object(3)
memory usage: 374.1+ KB
```

וכבר ניתן לראות שיש מספר רב של פיצרים חסרים. שבהם נטפל אחכ.

- המרנו את שנת הלידה לגיל (Age) שיהיה מידע יותר נח מול העיניים לניתוח

```
df['Age'] = 2022 - df['Year_Birth'] # add column for convient of the Age
df['Age']
```

- **הצגת המידע הסטטיסטי**

Show the data statistics, e.g., distribution, skewness, median and more. ¶

```
df.describe(include='all')
```

l]:

	ID	Year_Birth	Education	Status	Income	Num_of_kids	Num_of_Teen	Registration_date	Recency	Mnt_Fruits
count	1680.000000	1651.000000	1672	1646	1609.000000	1672.000000	1660.000000	1680	1680.000000	1673.000000
unique	NaN	NaN	5	6	NaN	NaN	NaN	634	NaN	NaN
top	NaN	NaN	Graduation	Married	NaN	NaN	NaN	14/02/2013	NaN	NaN
freq	NaN	NaN	830	653	NaN	NaN	NaN	10	NaN	NaN
mean	5584.735714	1969.047244	NaN	NaN	51983.554382	0.454545	0.503614	NaN	48.890476	303.676031
std	3233.716033	11.937421	NaN	NaN	26567.679664	0.538492	0.544011	NaN	29.091872	340.672889
min	0.000000	1893.000000	NaN	NaN	1730.000000	0.000000	0.000000	NaN	0.000000	0.000000
25%	2862.500000	1959.500000	NaN	NaN	34596.000000	0.000000	0.000000	NaN	24.000000	23.000000
50%	5511.000000	1970.000000	NaN	NaN	50611.000000	0.000000	0.000000	NaN	50.000000	167.000000
75%	8395.500000	1978.000000	NaN	NaN	67716.000000	1.000000	1.000000	NaN	74.000000	508.000000
max	11191.000000	1996.000000	NaN	NaN	666666.000000	2.000000	2.000000	NaN	99.000000	1493.000000

11 rows × 30 columns

שממנו ניתן ללמוד על חלוקת הערכים לפי חציונים, רבעונים ועוד, וכן על הערך שחוזר הכי הרבה ועוד שנראה בהמשך.

יונתן הרשקוביץ –205379373
מתן אשל -203502802

- **נקיון מוקדם**

בשלב נקיון והכנת המידע שיגיע בהמשך גילינו שישנם 7 רשומות מאוד בעיתיות ומלבלבלות, אלו 7 אנשים שאין להם שום מענה ויחס להיענות שלהם לקמפיינים בכלל, ולכן בעצם אין להם ערך, יתרה מכך ,ראינו שאם נסיר אותם- שאר הטיפול במידע נהיה הרבה יותר סביר.

	ID	Year_Birth	Education	Status	Income	Num_of_kids	Num_of_Teen	Registration_date	Recency	Mnt_Fruits	...	Response_Campaign_1	Res
1659	1419	1950.0	Graduation	Together	34026.0	1.0	1.0	05/08/2013	11	NaN	...	NaN	
1662	9284	1958.0	Graduation	Together	53977.0	0.0	1.0	08/06/2013	21	NaN	...	NaN	
1663	3673	1971.0	Graduation	Single	55239.0	0.0	1.0	14/07/2013	59	NaN	...	NaN	
1665	10983	1952.0	Graduation	Together	75278.0	0.0	0.0	29/01/2013	17	NaN	...	NaN	
1666	2611	1959.0	Master	Together	82576.0	0.0	0.0	01/08/2012	66	NaN	...	NaN	
1673	979	1975.0	Graduation	Single	33249.0	1.0	0.0	20/02/2013	11	NaN	...	NaN	
1675	8278	1990.0	PhD	Married	74214.0	0.0	0.0	26/08/2012	3	NaN	...	NaN	

7 rows × 30 columns

```
df[df.isnull().sum(axis=1)>7]
```

]:

ıpaign_1	Response_Campaign_2	Response_Campaign_3	Response_Campaign_4	Response_Campaign_5	Complain	Cost_Contact	Revenue	Response	Age
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	72.0
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	64.0
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	51.0
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	70.0
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	63.0
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	47.0
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	32.0

לכן ביצענו מחיקה של שורות אלו בפקודה הבאה:

```
df=df.drop(df.index[df.isnull().sum(axis=1)>7])
```

ולאחר מכן הצגנו שוב את המידע, ורואים שאכן כעת כמות הפיצרים החסרים יותר סבירה.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1673 entries, 0 to 1679
Data columns (total 30 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   ID                     1673 non-null   int64
 1   Year_Birth             1644 non-null   float64
 2   Education              1665 non-null   object
 3   Status                 1639 non-null   object
 4   Income                 1602 non-null   float64
 5   Num_of_kids            1665 non-null   float64
 6   Num_of_Teen            1653 non-null   float64
 7   Registration_date      1673 non-null   object
 8   Recency                1673 non-null   int64
 9   Mnt_Fruits             1673 non-null   float64
 10  Mnt_Meat               1673 non-null   float64
 11  Mnt_sweet              1659 non-null   float64
 12  Mnt_Wines              1673 non-null   float64
 13  Mnt_Gold_Products      1673 non-null   float64
 14  Mnt_Fish               1673 non-null   float64
 15  Num_Web_Purchases      1651 non-null   float64
 16  Num_Store_Purchases    1673 non-null   float64
 17  Num_Deals_Purchases    1673 non-null   float64
 18  Num_Catalog_Purchases  1673 non-null   float64
 19  Num_Web_Visits         1673 non-null   float64
 20  Response_Campaign_1    1662 non-null   float64
 21  Response_Campaign_2    1673 non-null   float64
 22  Response_Campaign_3    1673 non-null   float64
 23  Response_Campaign_4    1673 non-null   float64
 24  Response_Campaign_5    1673 non-null   float64
 25  Complain               1673 non-null   float64
 26  Cost_Contact           1673 non-null   float64
 27  Revenue                1673 non-null   float64
 28  Response               1673 non-null   int64
 29  Age                    1644 non-null   float64
dtypes: float64(24), int64(3), object(3)
memory usage: 385.6+ KB
```

- לאחר מכן פיצלנו את כל המידע שלנו לערך מטרה בנפרד.
ואת כל שאר המידע בנפרד (בשלב זה כבר הורדנו גם את ID שאין לו משמעות בכלל)

```
target = pd.DataFrame(df['Response']) # line 29 is the target
data = df.drop(['ID','Response'], axis = 1)
```

- כעת, חילצנו את כל העמודות בהם הערכים הם מספריים ולא מילוליים

```
numeric_col = data.describe().columns # to get the numeric column
numeric_col
```

- ואז חילצנו את כל המידע לשני חלקים, מידע נומרי ונומינלי

```
numeric_data = data[numeric_col] # numeric data
nominal_data = data.drop(numeric_col, axis=1) # nominal data
```

יונתן הרשקוביץ –205379373
מתן אשל -203502802

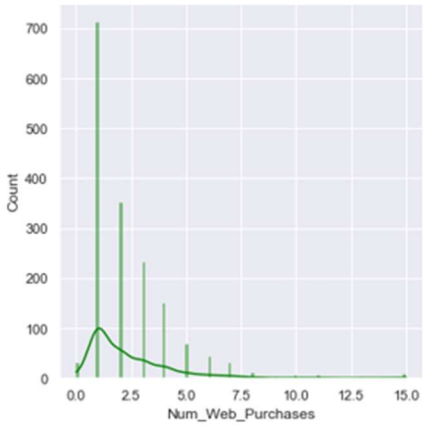
כעת – בשלב זה הצגנו לעצמינו (ונראה בהמשך) את הפלטים בצורת של היסטוגרמה והתפלגויות, והבחנו בכך שה"היענות לקמפיין 1-5" נמצא ביחד כחלק מה"ערכים המספריים" למרות שערכיהם בינארים (0-1) ולכן החלטנו להוסיף מסד נתונים אחר בשם nominal_data2 שהוא יכיל את אותו מידע נומילי לעיל, בנוסף לכל התגובות ל"מענה לקמפיינים" כי הם מתפרשים עבורינו כקטוגריאלים.

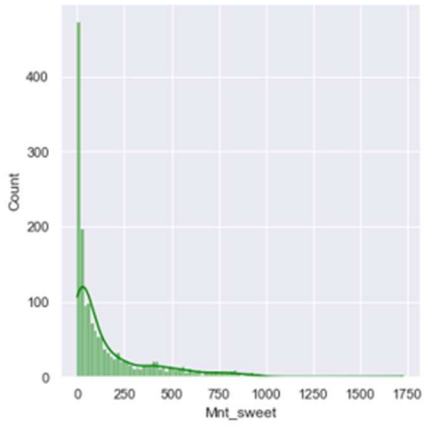
```
numeric_data = data[numeric_col] # numeric data
nominal_data = data.drop(numeric_col, axis=1) # nominal data

nominal_data1 = data.drop(numeric_col, axis=1) # nominal data - origin
nominal_data2 = nominal_data #'wil be the nominal + the answer to campain 1-5'

nominal_data2['Response_Campaign_5'] = data['Response_Campaign_5']
nominal_data2['Response_Campaign_4'] = data['Response_Campaign_4']
nominal_data2['Response_Campaign_3'] = data['Response_Campaign_3']
nominal_data2['Response_Campaign_2'] = data['Response_Campaign_2']
nominal_data2['Response_Campaign_1'] = data['Response_Campaign_1']
nominal_data2['Num_of_kids'] = data['Num_of_kids']
nominal_data2['Num_of_Teen'] = data['Num_of_Teen']

nominal_data1
nominal_data2
```

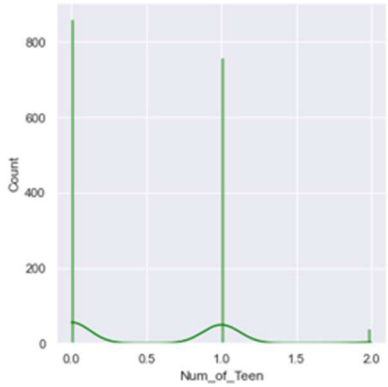
כריית מידע וייצוג מידע-2022

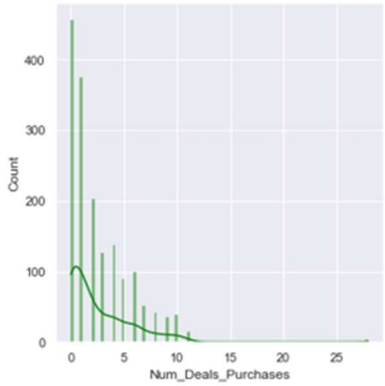
תשפ"ב סמסטר ב'
יונתן הרשקוביץ –205379373
מתן אשל -203502802

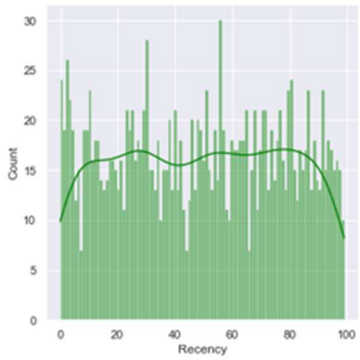
הצגת הגרפים

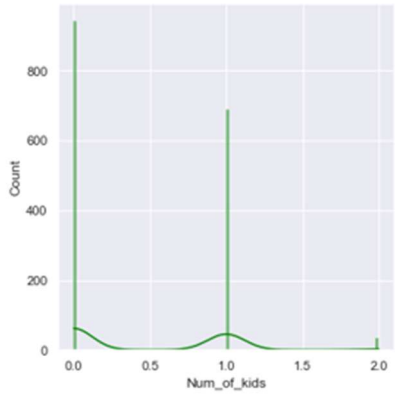








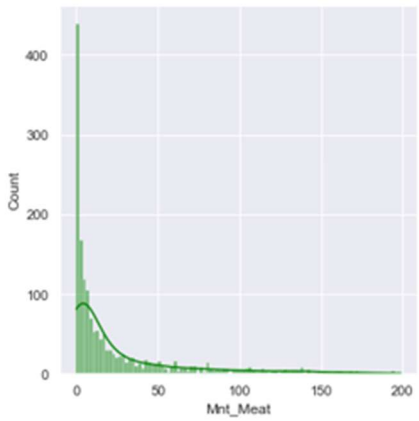


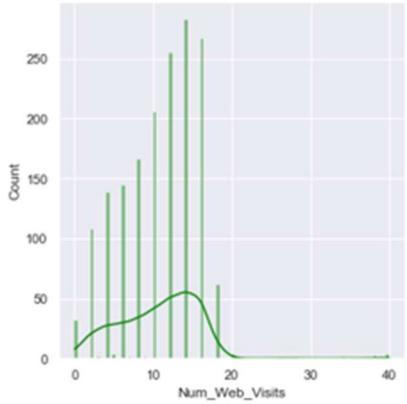


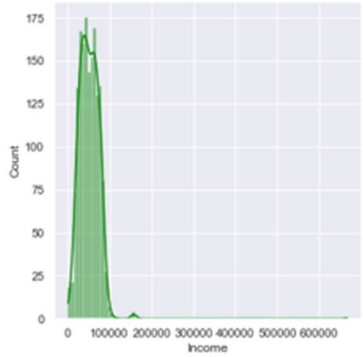
כריית מידע וייצוג מידע-2022

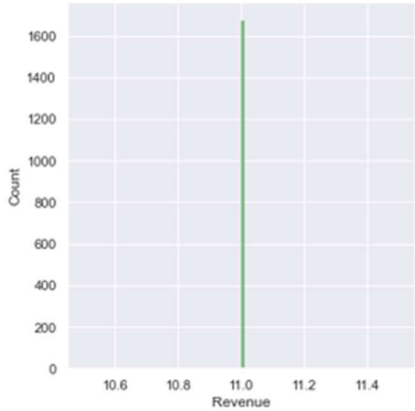
תשפ"ב סמסטר ב'

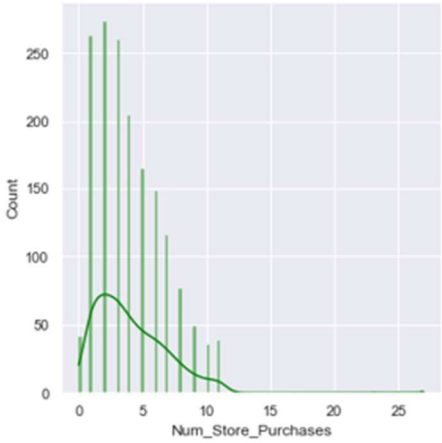
יונתן הרשקוביץ –205379373
מתן אשל -203502802

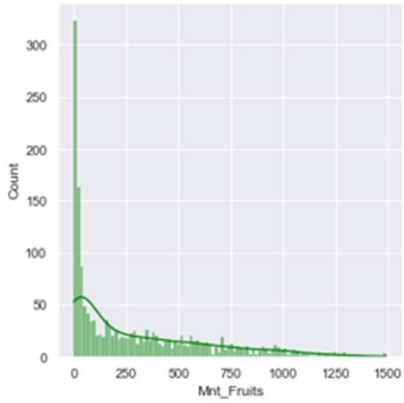








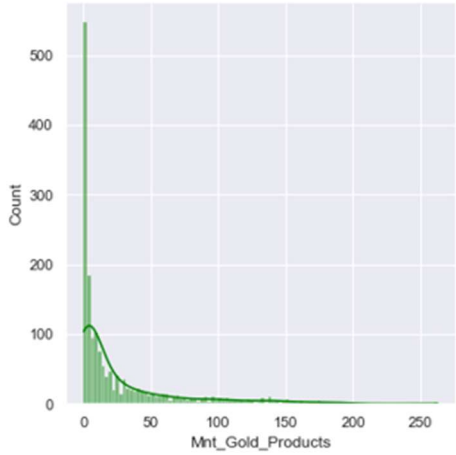


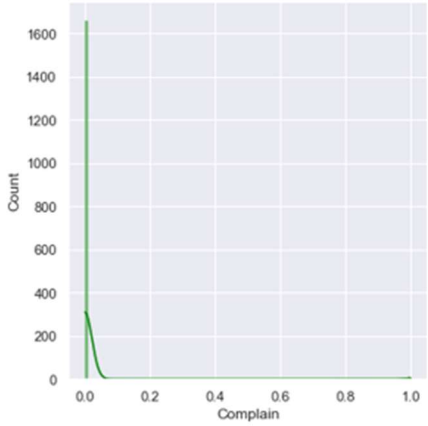


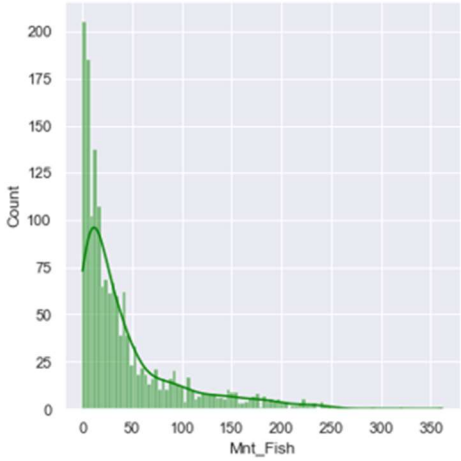
כריית מידע וייצוג מידע-2022

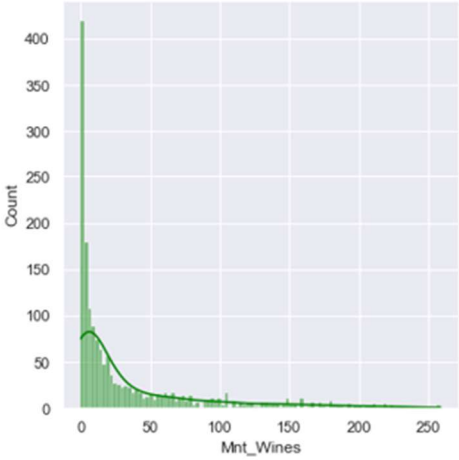
תשפ"ב סמסטר ב'

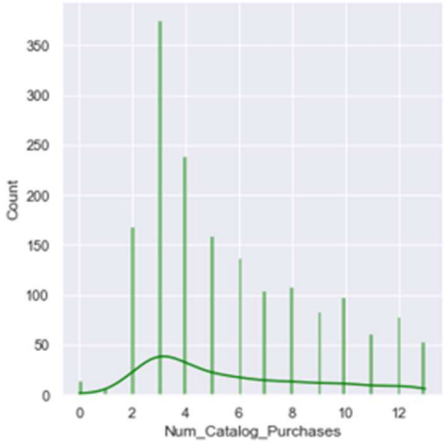
יונתן הרשקוביץ –205379373
מתן אשל -203502802

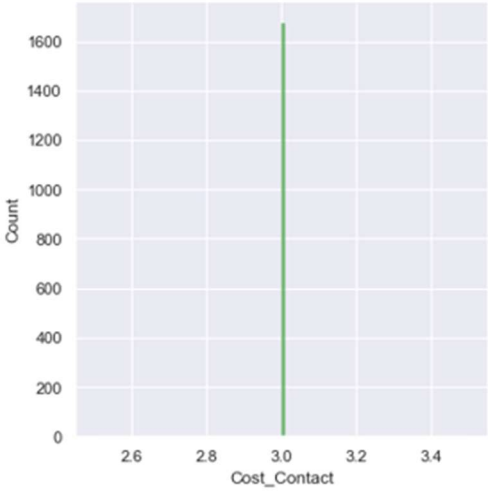




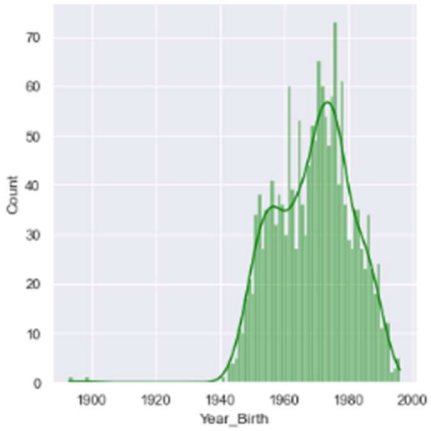


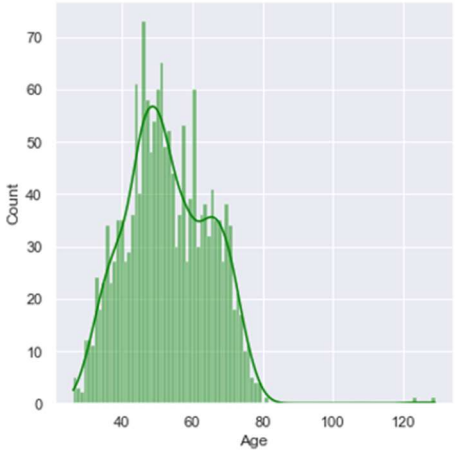






יונתן הרשקוביץ –205379373
מתן אשל -203502802





- שמנו לב שיש שני תמונות נוספות עם ערך קבוע ולא רלוונטי Revenue והCost_contatc לכן העפנו אותם מהמידע בהמשך.

יונתן הרשקוביץ –205379373
מתן אשל -203502802

- חישוב: Skewness

```
for col in numeric_col:
    print(col, ' skewness:', numeric_data[col].skew(axis = 0, skipna = True) )
```

```
Year_Birth  skewness: -0.3570560299294453
Income  skewness: 7.92931885290212
Num_of_kids  skewness: 0.5824618563639308
Num_of_Teen  skewness: 0.41312651801701283
Recency  skewness: -0.01572417501512122
Mnt_Fruits  skewness: 1.1829109635956951
Mnt_Meat  skewness: 2.149518442967082
Mnt_sweet  skewness: 2.1092143754162325
Mnt_Wines  skewness: 1.988342728206097
Mnt_Gold_Products  skewness: 2.218833414968302
Mnt_Fish  skewness: 1.940248617323809
Num_Web_Purchases  skewness: 2.5125546694750898
Num_Store_Purchases  skewness: 1.5552728492914274
Num_Deals_Purchases  skewness: 2.0657617307381697
Num_Catalog_Purchases  skewness: 0.6997937661170337
Num_Web_Visits  skewness: 0.25966407514401624
Response_Campaign_1  skewness: 3.191823311903448
Response_Campaign_2  skewness: 3.3224579772122724
Response_Campaign_3  skewness: 3.305146751217529
Response_Campaign_4  skewness: 3.5271264290087347
Response_Campaign_5  skewness: 8.359327109203889
Complain  skewness: 10.427719201596044
Cost_Contact  skewness: 0
Revenue  skewness: 0
Age  skewness: 0.3570560299294346
```

ניתן לראות שבערכים בהם היינו מצפים לקבל שההתפלגות תהיה נורמלית יחסית בלי הרבה ערכי קיצון הSKEW אכן נמוך כמו למשל בAGE, וכן במספר הילדים והמתבגרים בבית.

לעומת הזנב של ההכנסות, שבו ייתכן ויהיו ערכי קיצון (כמו שנגלה בהמשך שיש ערך אחד קיצוני ביותר שנשמיט אותו בהמשך) וכמובן שהזנב לערכי המענה לקמפיינים הוא לא רלוונטי כי הערכים שם בינארים ואין התפלגות נורמלית כלל. (לכן הסטנו אותם לערכי הנומינלים)

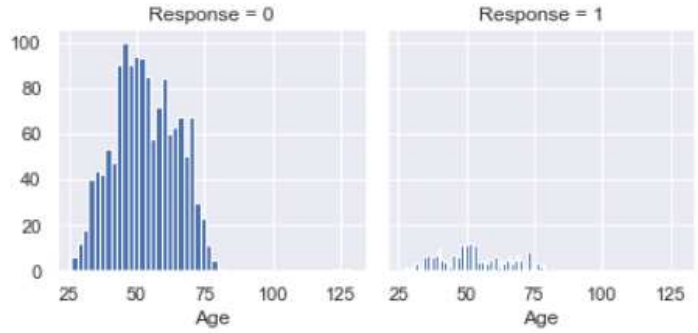
יונתן הרשקוביץ –205379373
מתן אשל -203502802

עצרנו בשלב זה לנסות לנתח ולגלות קולרציה אחת בעצמינו בין ערך המטרה לבין הגילאים:
בהמשך בעזרת מפת החום, נמצא עוד קשרים רבים.
להלן התוצאה:

Categorial attributes

```
g = sns.FacetGrid(df, col='Response')
g.map(plt.hist, 'Age', bins=50)
print("Mean age of Response = False: "+str(df.loc[df['Response']==False,'Age'].mean()))
print("Mean age of Response = True: "+ str(df.loc[df['Response']==True,'Age'].mean()))
print("Median age of Response = False: "+str(df.loc[df['Response']==False,'Age'].median()))
print("Median age of Response = True: "+ str(df.loc[df['Response']==True,'Age'].median()))
```

```
Mean age of Response = False: 53.008528784648185
Mean age of Response = True: 52.631147540983605
Median age of Response = False: 52.0
Median age of Response = True: 51.0
```



יונתן הרשקוביץ –205379373
מתן אשל -203502802

המידע הנומינלי

כעת, התחלנו לנתח ולהציג ויזואליזציה של המידע הנומינלי, וכן מידע סטטיסטי לגביו (לשים לב, בכוונה בהצגת המידע הסטטיסטי הצגנו את nominal_data1 כי אם היינו מציגים את 2 אז היה הולך להציג רק את הערכים המספריים של הקמפיינים, לכן השארנו בשקט את המידע המילולי פה להצגה)

Nominal data

we would like to know how many unique values there are and the distribution.

```
nominal_data2.info()
```

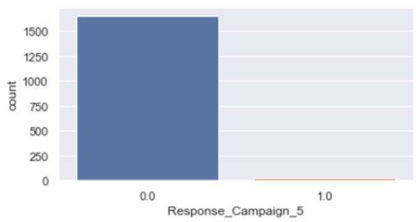
```
describe_df = nominal_data1.describe()
describe_df.loc['%freq of top value'] = describe_df.apply(lambda x: np.round(x.loc['freq']/1680, decimals=3))
describe_df
```

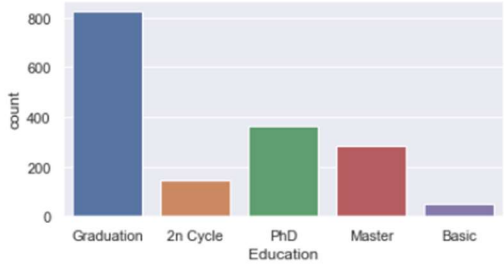
```
describe_df = nominal_data1.describe()
describe_df.loc['%freq of top value'] = describe_df.apply(lambda x: np.round(x.loc['freq']/1680, decimals=3))
describe_df
```

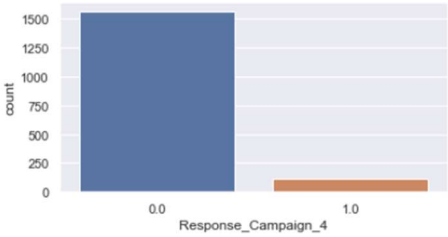
5]:

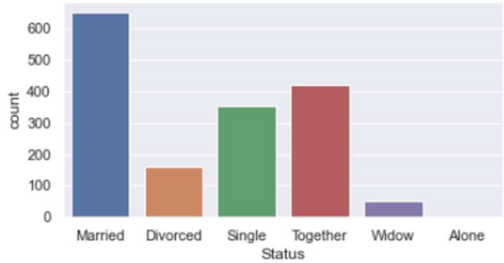
	Education	Status	Registration_date
count	1665	1639	1673
unique	5	6	633
top	Graduation	Married	14/02/2013
freq	825	652	10
%freq of top value	0.491	0.388	0.006

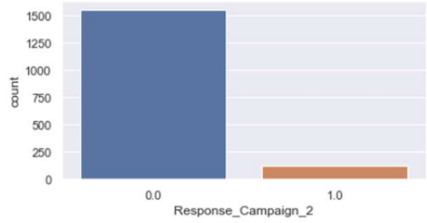
כעת הצגנו את המידע הויזאולי:







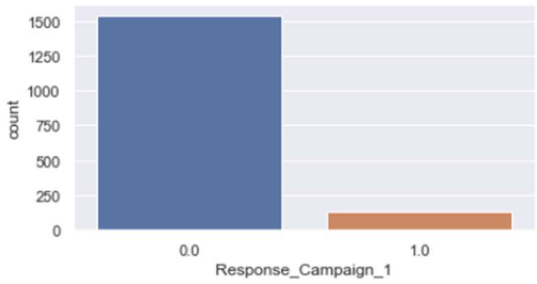


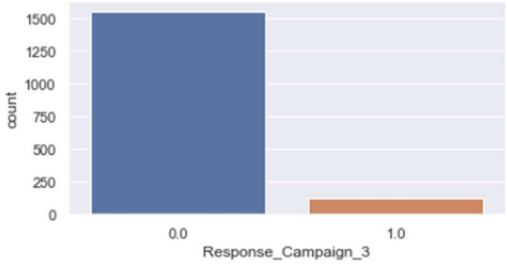


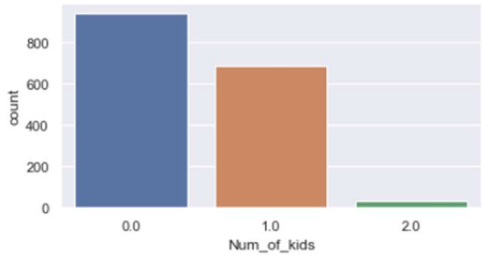
כריית מידע וייצוג מידע-2022

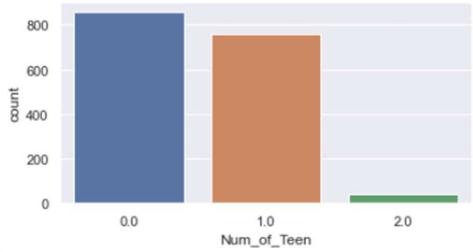
תשפ"ב סמסטר ב'

יונתן הרשקוביץ –205379373
מתן אשל -203502802









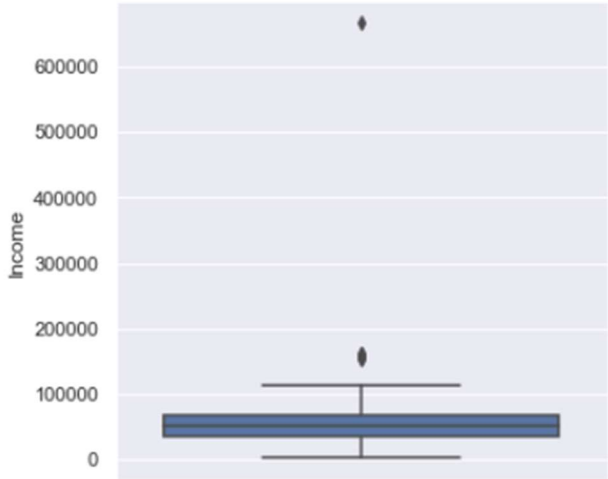
יונתן הרשקוביץ –205379373
מתן אשל -203502802

<u>נציג חלק מהמידע בתצורת BOXPLOT</u>

לדוגמא, המידע על השכר

BOXPLOT

```
In [39]: sns.boxplot(y="Income", data=df)
Out[39]: <AxesSubplot:ylabel='Income'>
```



זיהינו פה חריגה, לכן הלכנו לברר מי מסיט את הגרף והתמונה וגילינו שיש אדם אחד שמרוויח פי כמה יותר מכולם באופן חריג

```
df[df['Income']>600000]
```

:

	ID	Year_Birth	Education	Status	Income	Num_of_kids	Num_of_Teen
276	9432	1977.0	Graduation	Together	666666.0	1.0	0.0

לכן החלטנו להעיף אותו מהמאגר המידע שכן הוא ערך חריג מאוד.
ולאחר מכן יצרנו ערך חדש עם האנשים שמרוויחים פחות מ150 אלף בשנה.
(ישנם אומנם כמה ספורים שמרוויחים יותר אבל לא בצורה קיצונית, השארנו אותם

```
# get out the rich man
df=df.drop(df.index[df['Income']>600000])
```

```
income_under_150k = numeric_data[numeric_data['Income']<=150000]['Income']
```

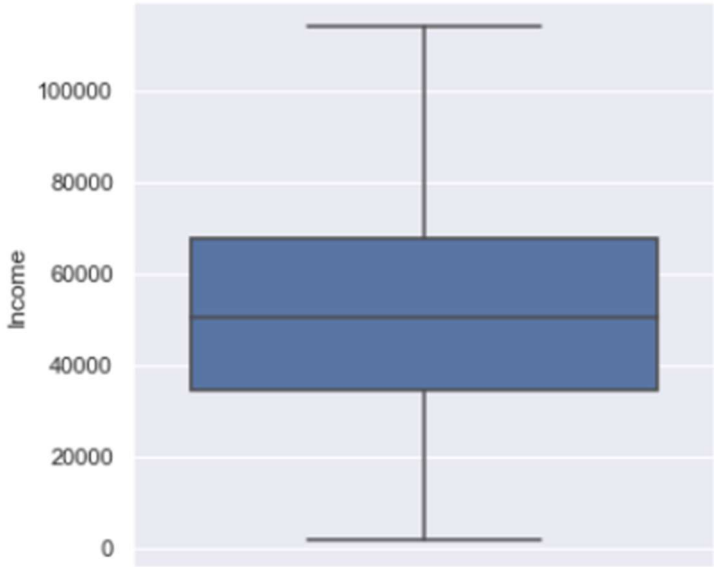
```
income_under_150k.mean()
```

```
: 51171.08275862069
```

```
sns.boxplot(y=income_under_150k)
```

וניתן לראות שכאשר הוצאנו את החריגים אזיי ההתלפגות נראית הגיונית ונורמלית מאוד.

```
In [57]: sns.boxplot(y=income_under_150k)
Out[57]: <AxesSubplot:ylabel='Income'>
```

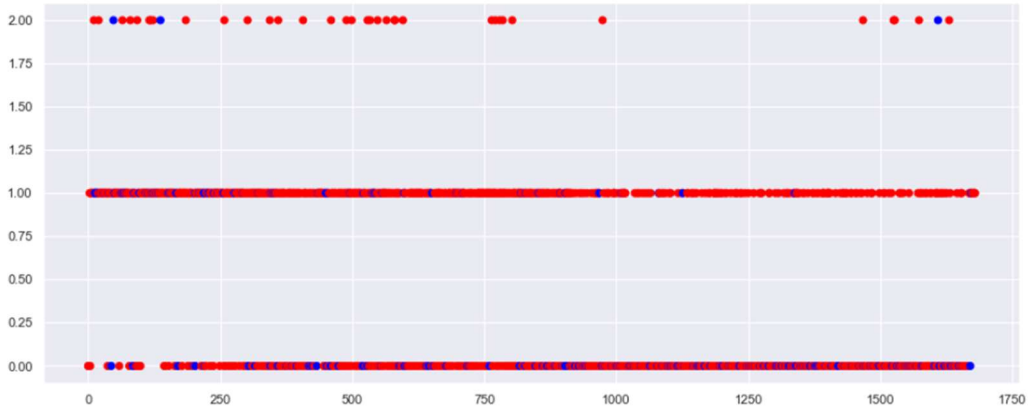


ניסינו לבצע גם הצגה של SCATTER PLOT עם ערכים שונים, אך לא קיבלנו משו משמועתי ללמוד ממנו כמו שראינו בתרגול.
נצרף דוגמא אחת שמצאנו

Scatter Plot -

```
colordict = {0: 'red', 1: 'blue'}

plt.figure(figsize=(15, 6))
plt.scatter(df.index, df['Num_of_Teen'], c = df['Response'].map(colordict))
```

```
8]: <matplotlib.collections.PathCollection at 0x3210a5f8>
```



יונתן הרשקוביץ –205379373
מתן אשל -203502802

קורלציות בין פיצרים והשלמת מידע

כשלב בפני עצמו וכן כהכנה לנקיון המידע, רצינו לבצע פה את שיטת הקורלציה של פירסון ולהציג במפת חום את היחס בין כל המשתנים. גם מעניין לעצמו וגם יעזור בהמשך להשלים מידע עם הגיון.

לצורך כך , קבענו את הערכים המילולים כמו סטטוס וכדומה לערכים מספריים.

והפעלנו את פונקציה שמוצאת קורלציה

2. Filling missing values by exploring correlations. ¶

```
df['Education_cat'] = df['Education'].astype('category').cat.codes
df['Status_cat'] = df['Status'].astype('category').cat.codes
```

```
df.groupby(['Status','Status_cat']).size()
```

```
0]: Status    Status_cat
    Alone     0               2
    Divorced  1             159
    Married   2             652
    Single    3             354
    Together  4             419
    Widow     5              52
    dtype: int64
```

```
df.corr(method = 'pearson')
```

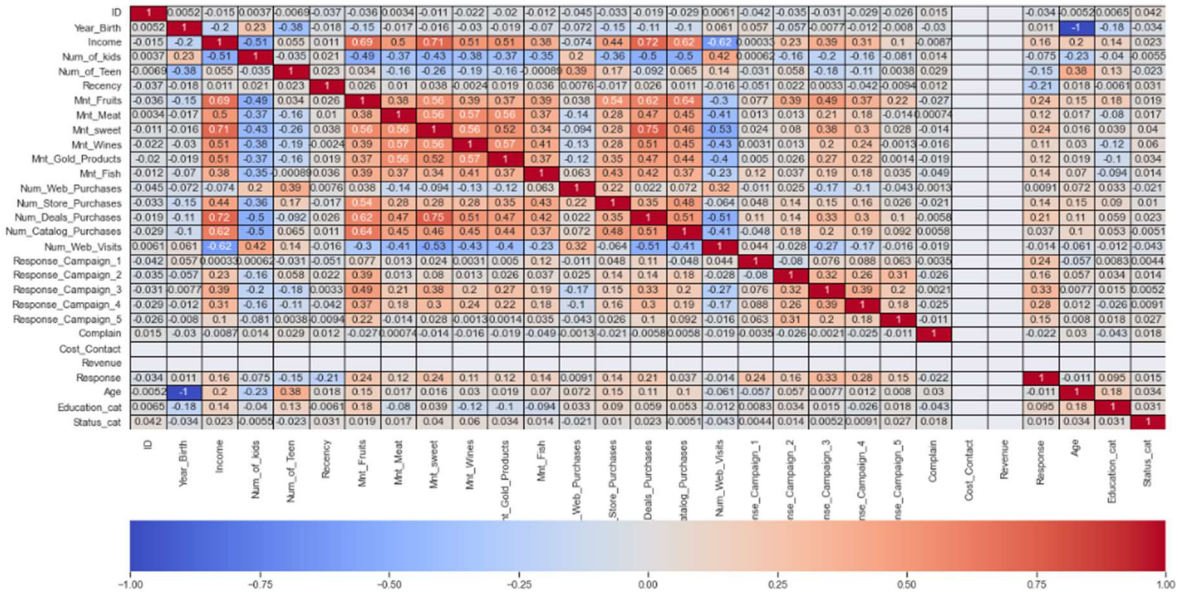
כריית מידע וייצוג מידע-2022

תשפ"ב סמסטר ב'

יונתן הרשקוביץ –205379373
מתן אשל -203502802

מפת החום:

```
# heat map
plt.figure(figsize=(20,12))
sns.heatmap(df.corr(), annot = True, vmin=-1, vmax=1, center= 0,
            cmap= 'coolwarm', linewidths=0.7, linecolor='black', cbar_kws= {'orientation': 'horizontal'})
```



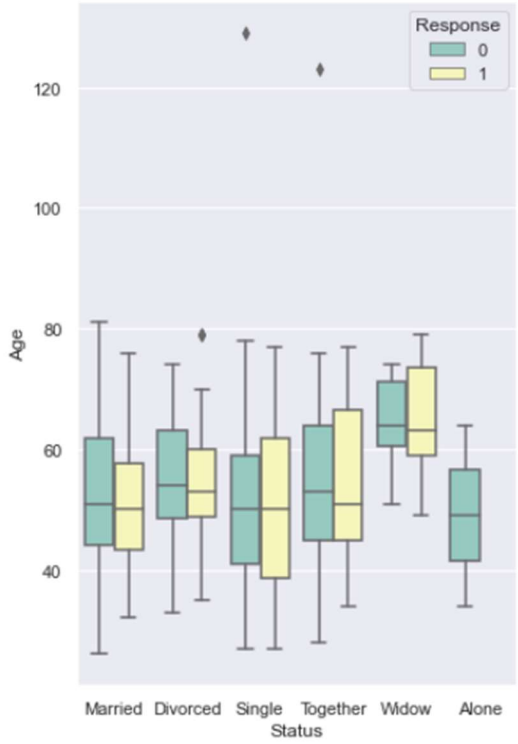
לא לפי הסדר, אבל כך מופיע במחברת, הצגנו עוד כמה boxplot ששופכים מידע מעניין

סטטוס לעומת ערך מטרה

more BOXPLOT ¶

```
sns.set(rc={'figure.figsize':(5,8)})
sns.boxplot(y="Age", x="Status", hue= "Response", data=df, palette="Set3")
```

```
53]: <AxesSubplot:xlabel='Status', ylabel='Age'>
```

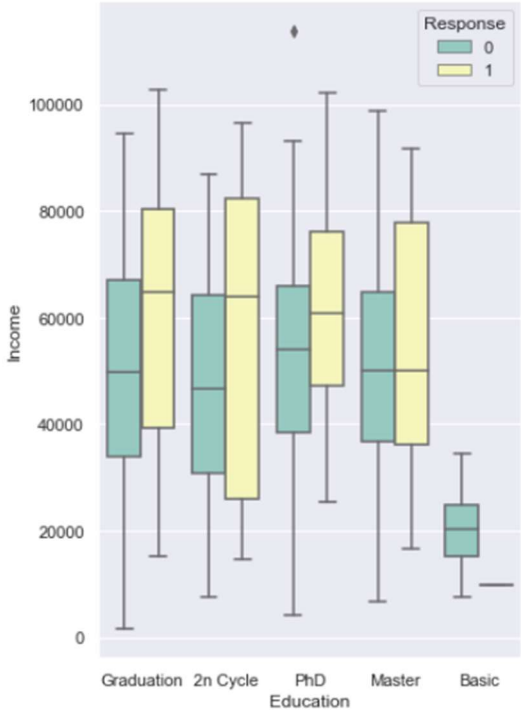


חינוך והשכלה לעומת ערך מטרה

יונתן הרשקוביץ –205379373
מתן אשל -203502802


```
# sns.set(rc={'figure.figsize':(5,8)})
# sns.boxplot(y="Income", x="Education", hue= "Response", data=df, palette="Set3")
sns.set(rc={'figure.figsize':(5,8)})
sns.boxplot(y=income_under_150k, x="Education", hue= "Response", data=df, palette="Set3")
```

```
4]: <AxesSubplot:xlabel='Education', ylabel='Income'>
```



יונתן הרשקוביץ –205379373
מתן אשל -203502802

Data Cleaning

ניגש לשלב נקיון המידע מדברים לא רלנווטים וכן השלמת המידע של דברים חסרים. נעזרנו במפת חום כדי לחפש קולרציה והגיון מושכל להשלמת המידע

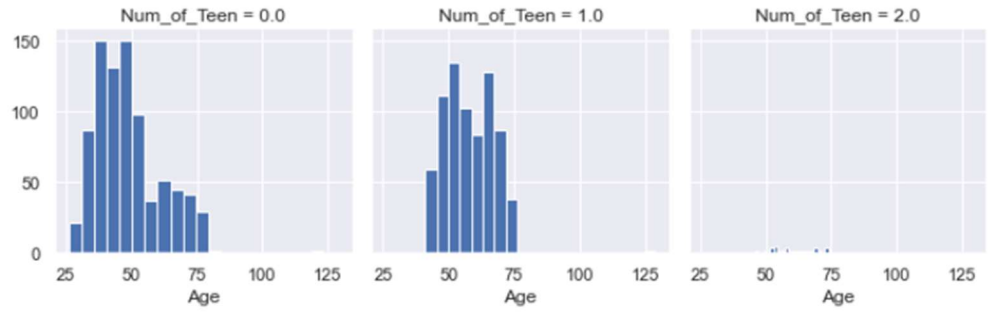
טיפול בחוסר בAGE:

ראינו שיש קשר במפת חום בין הגיל לבין כמות הילדים המתבגרים בבית. החלטנו לבדוק קורלציה ביניהם וכתלות במספר המתבגרים.

ואילו התוצאות:

search for coorrelation to fill the missing age

```
g = sns.FacetGrid(df, col='Num_of_Teen')
g.map(plt.hist, 'Age', bins=20);
```




```
for i in df['Num_of_Teen'].unique():
    df['Age'][df['Num_of_Teen']==i].median()
    print('For class ', i, ' the median is ', df.loc[df['Num_of_Teen']==i , 'Age'].median())
    print('and the number of missing values is ', df.loc[df['Num_of_Teen']==i , 'Age'].isnull().sum(),
          'out of ',(df['Num_of_Teen']==i).sum(), 'rows')
```

```
For class  0.0  the median is  46.0
and the number of missing values is  10 out of  857 rows
For class  1.0  the median is  57.0
and the number of missing values is  11 out of  757 rows
For class  2.0  the median is  58.0
and the number of missing values is  0 out of  38 rows
For class  nan  the median is  nan
and the number of missing values is  0 out of  0 rows
```

יצא בדומה לכיתה, לכן השלמנו את הגילאים החסרים לפי הCLASS שיצא שהם נמצאים בו לפי כמות המתבגרים:

ואת השאר מילאנו לפי הממוצע שמצאנו

```
# נשאר עוד כמה אנשים ללא גיל -נמלא אותם בשכיח\ או ממוצע
mean_age = df['Age'].mean()
mean_age
```

```
]: 52.935523114355234
```

```
df['Age_2'] = df['Age_2'].fillna(mean_age)
```

יונתן הרשקוביץ –205379373
מתן אשל -203502802

טיפול בחוסרים של השכלה.

```
df[df['Education'].isnull()==True]
```

3]:

	ID	Year_Birth	Education	Status	Income	Nu
502	5985	NaN	NaN	NaN	NaN	
509	9699	NaN	NaN	NaN	NaN	
634	2587	NaN	NaN	NaN	NaN	
936	1544	NaN	NaN	NaN	NaN	
1151	2431	NaN	NaN	NaN	NaN	
1464	10451	NaN	NaN	NaN	NaN	
1502	6437	NaN	NaN	NaN	NaN	
1639	7627	NaN	NaN	NaN	NaN	

8 rows × 31 columns

```
mode = df['Education'].mode()[0]
mode
```

9]: 'Graduation'

למרות קשר מעניין שראינו בהמשך בין השכלה לבין צריכת פירות, מכיוון שכמות החוסרים לא גדולה החלטנו למלא את הערכים החסרים פשוט בערך השכיח שזה "בוגר תואר ראשון"

טיפול בחוסרים של סטטוס זוגי

גם חלק זה ניסינו לנתח לפי קורלציות לפיצרים אחרים, אך לא ראינו משהו חזק במיוחד, בין השאר, כמות החוסרים פה קטנה ולכן מילאנו שוב לפי הערך השכיח

```
mode = df['Status'].mode()[0]
mode
df['Status'] = df['Status'].fillna(mode)
```

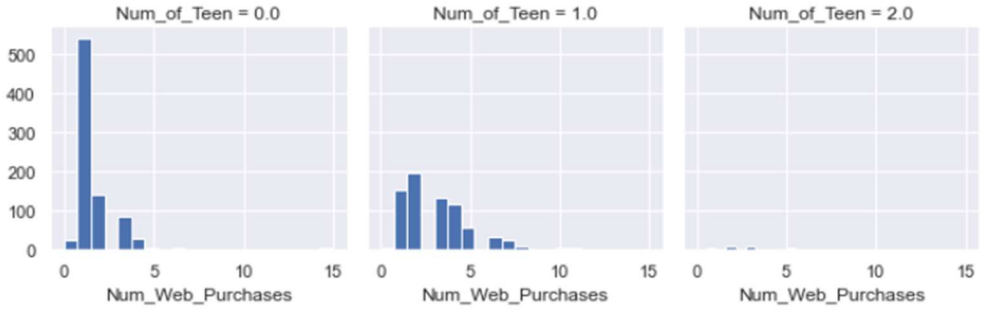
יונתן הרשקוביץ –205379373
מתן אשל -203502802

<u>טיפול בחוסרים של רכישות באינטרנט:</u>

פה שמנו לב ליחס בין כמות הרכישות לבין מספר המתבגרים בבית ,שיש לזה הסבר דווקא מאוד הגיוני😊

ולכן ביצענו בדיקות והשלמות בדומה לחלק לעיל.

```
g = sns.FacetGrid(df, col='Num_of_Teen')
g.map(plt.hist, 'Num_Web_Purchases', bins=20);
```




```
for i in df['Num_of_Teen'].unique():
    df['Num_Web_Purchases'][df['Num_of_Teen']==i].median()
    print('For class ', i, ' the median is ', df.loc[df['Num_of_Teen']==i , 'Num_Web_Purchases'].median())
    print('and the number of missing values is ', df.loc[df['Num_of_Teen']==i , 'Num_Web_Purchases'].isnull().sum(),
          'out of ',(df['Num_of_Teen']==i).sum(), 'rows')
```

```
For class  0.0  the median is  1.0
and the number of missing values is  11 out of  858 rows
For class  1.0  the median is  3.0
and the number of missing values is  11 out of  757 rows
For class  2.0  the median is  3.0
and the number of missing values is  0 out of  38 rows
For class  nan  the median is  nan
and the number of missing values is  0 out of  0 rows
```

את ההשלמות ביצענו בצורה חכמה, לפי הממוצע בכל קבוצה בפני עצמה.

יונתן הרשקוביץ –205379373
מתן אשל -203502802

טיפול בחוסרים של כמות ילדים ומתבגרים בבית

גם פה ראינו שכמות החוסרים לא גדולה, ולכן השלמנו אותם ע"י הערך השכיח שקיים. (כמובן שיכול לצאת מצב להשלים כמות ילדים לבן אדם לא נשוי וכדומה אבל זה ערכים זניחים וגם ייתכן שלאדם לא נשוי יהיו ילדים)

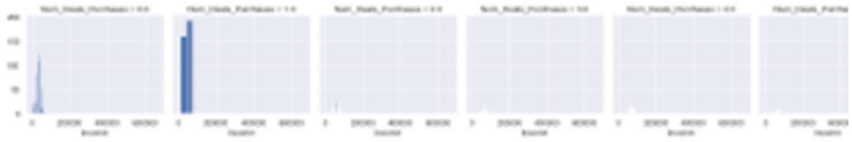
```
# ילדים ומתגברים לא חסר הרבה נשלים לפי הרוב
mode = df['Num_of_kids'].mode()[0]
mode
df['Num_of_kids'] = df['Num_of_kids'].fillna(mode)
```

```
# ילדים ומתגברים לא חסר הרבה נשלים לפי הרוב
mode = df['Num_of_Teen'].mode()[0]
mode
df['Num_of_Teen'] = df['Num_of_Teen'].fillna(mode)
```

טיפול בחוסרים של רמת הכנסה

זיהינו יחס מעניין בין רמת ההכנסה לבין כמות הרכישות שבוצעו עם הנחה. (מעניין אם אולי לאנשים עם משכורת גבוהה יותר יש יותר מודעות כלכלית לקניות מושכלות ובהנחה וכו')

```
g = sns.FacetGrid(df, col='Num_Deals_Purchases')
g.map(plt.hist, 'Income', bins=20);
```



גם פה זיהינו חלוקה למספר מחלקות מסוימות ואת השלמת המידע החסר ביצענו בהתאם.

פיצרים אחרונים חסרים – כמות מתוקים והיענות לקמפיין 1

בשני האחרונים הללו, זיהינו שוב כמות קטנה מאוד של חוסרים ולכן ביצענו השלמה.
במתוקים שזו כמות נומרית ביצענו לפי הממוצע,
ובהיענות לקמפיין שזה ערכים בינארים השלמנו לפי השכיח.

סיימנו את שלב הכנת המידע ונקיון המידע והסרנו עמודות לא רלוונטיות, והשלמנו את החוסרים בעמודות שבהם היה חסר, ככל הניתן לפי הגיון ושכל.

Data transformation And Data reduction

ביצענו שיטות לנורמליזציה שלמדנו בכיתה, הן כדי להתלמד בהן והן כדי להכין את המידע לקראת תהליך הPCA שדורש ערכים מנורמלים.

השתמשנו בשיטת הMAX_MIN לנרמל מספר עמודות בבת אחת שעליהם נרצה לערוך pca בהמשך

Min-Max normalization

Change the data to be with values between 0 to 1.

```
scaler = MinMaxScaler()
for i in ['Response','Age','Income','Mnt_Fruits','Mnt_Meat','Mnt_sweet','Mnt_Fish','Mnt_Gold_Products','Mnt_Wines']:
    i2 = str(i) + "_norm"
    scaler.fit(pd.DataFrame(df[i]))
    df[i2] = scaler.transform(pd.DataFrame(df[i])).astype(np.float64)
    print(scaler.data_min_)
    print(scaler.data_max_)
    df[[i,i2]].head()

df.head()
```

עוד הכנה שלא טופלה לפני הpca היא המרת התאריך שהיה נתון בפורמט מסוים לערכים מחולקים.
(נכון, שבסוף מידע זה לא השתתף בPCA אבל תיקנו אותו בכל זאת)

DATA REDUCTION

PCA

```
# before make sure all value are numeric, notice that the Education_2, Status, go back to Object after we fiilna them
# and to take care of the Registeration date
```

```
df['Education_2'] = df['Education_2'].astype('category').cat.codes
df['Status'] = df['Status'].astype('category').cat.codes
```

```
df[['Registration_date_day', 'Registration_date_month', 'Registration_date_year']] = df['Registration_date'].str.split('/
df['Registration_date_day'] = df['Registration_date_day'].astype(float)
df['Registration_date_month'] = df['Registration_date_month'].astype(float)
df['Registration_date_year'] = df['Registration_date_year'].astype(float)
df= df.drop(['Registration_date'], axis=1)
df.head()
```

נציג את המידע שיצא מהPCA

(הערה: לפני הגשת התרגיל הרצנו הרצה חוזרת של הכל ומשהו לא עבד כשורה, איבדנו את תמונת הPCA, דווקא יצא מידע מעניין וחלוקה יפה

נצרף את הקוד בתקווה שבהרצה נוספת הדברים יסתדרו)

יונתן הרשקוביץ –205379373
מתן אשל -203502802

כעת ביצענו Discretization

ביצענו שני שיטות שלמדנו בכיתה

לחלק את קבוצת הגילאים לפי קבוצות בגודל שווה

או לפי קבוצות שהגדרנו מראש את טווחי הגילאים בכל קבוצה (גם אם גודלם לא שווה