Ag20

January 20, 2024

[98]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, r2_score,␣
 ↪mean_absolute_error,mean_squared_error
sns.set_style("whitegrid")
```

[99]:
```
import pandas as pd
df = pd.read_csv('/Users/thutranghoa/Code/Data_analysis/Data/marketing_data.
 ↪csv')
df
```

[99]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	\
0	5524	1957	Graduation	Single	58138.0	0	
1	2174	1954	Graduation	Single	46344.0	1	
2	4141	1965	Graduation	Together	71613.0	0	
3	6182	1984	Graduation	Together	26646.0	1	
4	5324	1981	PhD	Married	58293.0	1	
…	…	…	…	…	…	…	
2235	10870	1967	Graduation	Married	61223.0	0	
2236	4001	1946	PhD	Together	64014.0	2	
2237	7270	1981	Graduation	Divorced	56981.0	0	
2238	8235	1956	Master	Together	69245.0	0	
2239	9405	1954	PhD	Married	52869.0	1	

	Teenhome	Dt_Customer	Recency	MntWines	…	NumWebVisitsMonth	\
0	0	04-09-2012	58	635	…	7	
1	1	08-03-2014	38	11	…	5	
2	0	21-08-2013	26	426	…	4	
3	0	10-02-2014	26	11	…	6	
4	0	19-01-2014	94	173	…	5	
…	…	…	…	…	…	…	
2235	1	13-06-2013	46	709	…	5	
2236	1	10-06-2014	56	406	…	7	

```
2237         0  25-01-2014       91       908  …                  6
2238         1  24-01-2014        8       428  …                  3
2239         1  15-10-2012       40        84  …                  7

      AcceptedCmp3  AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  \
0                0             0             0             0             0
1                0             0             0             0             0
2                0             0             0             0             0
3                0             0             0             0             0
4                0             0             0             0             0
…              …             …             …             …             …
2235             0             0             0             0             0
2236             0             0             0             1             0
2237             0             1             0             0             0
2238             0             0             0             0             0
2239             0             0             0             0             0

      Complain  Z_CostContact  Z_Revenue  Response
0            0              3         11         1
1            0              3         11         0
2            0              3         11         0
3            0              3         11         0
4            0              3         11         0
…          …              …         …          …
2235         0              3         11         0
2236         0              3         11         0
2237         0              3         11         0
2238         0              3         11         0
2239         0              3         11         1

[2240 rows x 29 columns]
```

[100]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   ID                   2240 non-null   int64
 1   Year_Birth           2240 non-null   int64
 2   Education            2240 non-null   object
 3   Marital_Status       2240 non-null   object
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64
 6   Teenhome             2240 non-null   int64
 7   Dt_Customer          2240 non-null   object
 8   Recency              2240 non-null   int64
```

```
 9   MntWines              2240 non-null   int64
 10  MntFruits             2240 non-null   int64
 11  MntMeatProducts       2240 non-null   int64
 12  MntFishProducts       2240 non-null   int64
 13  MntSweetProducts      2240 non-null   int64
 14  MntGoldProds          2240 non-null   int64
 15  NumDealsPurchases     2240 non-null   int64
 16  NumWebPurchases       2240 non-null   int64
 17  NumCatalogPurchases   2240 non-null   int64
 18  NumStorePurchases     2240 non-null   int64
 19  NumWebVisitsMonth     2240 non-null   int64
 20  AcceptedCmp3          2240 non-null   int64
 21  AcceptedCmp4          2240 non-null   int64
 22  AcceptedCmp5          2240 non-null   int64
 23  AcceptedCmp1          2240 non-null   int64
 24  AcceptedCmp2          2240 non-null   int64
 25  Complain              2240 non-null   int64
 26  Z_CostContact         2240 non-null   int64
 27  Z_Revenue             2240 non-null   int64
 28  Response              2240 non-null   int64
dtypes: float64(1), int64(25), object(3)
memory usage: 507.6+ KB
```

[101]:
```
df = df.drop(['ID','Z_CostContact', 'Z_Revenue'], axis=1)
```

[102]:
```
df.columns
```

[102]:
```
Index(['Year_Birth', 'Education', 'Marital_Status', 'Income', 'Kidhome',
       'Teenhome', 'Dt_Customer', 'Recency', 'MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
       'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
       'AcceptedCmp2', 'Complain', 'Response'],
      dtype='object')
```

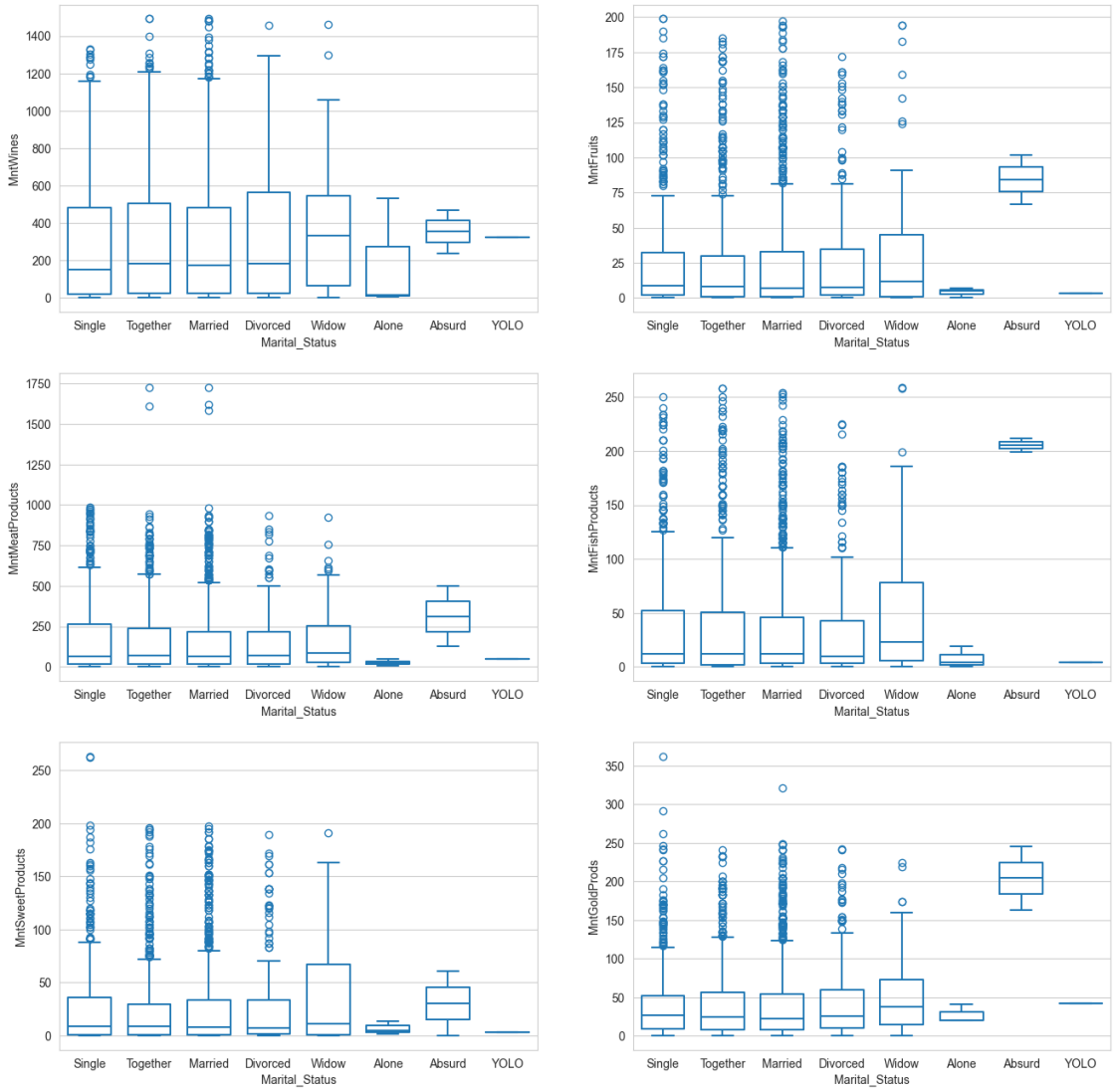
0.1 Spend category

'MntWines', 'MntFruits','MntMeatProducts', 'MntFishProducts', 'MntSweetProducts','MntGoldProds'

[103]:
```
'Marital situation'

fig = plt.gcf()
# ax = f.add_subplot(111)
# ax.yaxis.tick_right()
fig.set_size_inches(16, 16)
```

```
fig.suptitle('Spend category by marital situation', fontsize=20, weight='bold',␣
 ↪color = 'r')


plt.subplot(3,2,1)
sns.boxplot(data= df,x='Marital_Status',y='MntWines', fill=False, gap=.1)

plt.subplot(3,2,2)
sns.boxplot(data= df,x='Marital_Status',y='MntFruits', fill=False, gap=.1)

plt.subplot(3,2,3)
sns.boxplot(data= df,x='Marital_Status',y='MntMeatProducts', fill=False, gap=.1)

plt.subplot(3,2,4)
sns.boxplot(data= df,x='Marital_Status',y='MntFishProducts', fill=False, gap=.1)

plt.subplot(3,2,5)
sns.boxplot(data= df,x='Marital_Status',y='MntSweetProducts', fill=False, gap=.
 ↪1)

plt.subplot(3,2,6)
sns.boxplot(data= df,x='Marital_Status',y='MntGoldProds', fill=False, gap=.1)
```

[103]: <Axes: xlabel='Marital_Status', ylabel='MntGoldProds'>

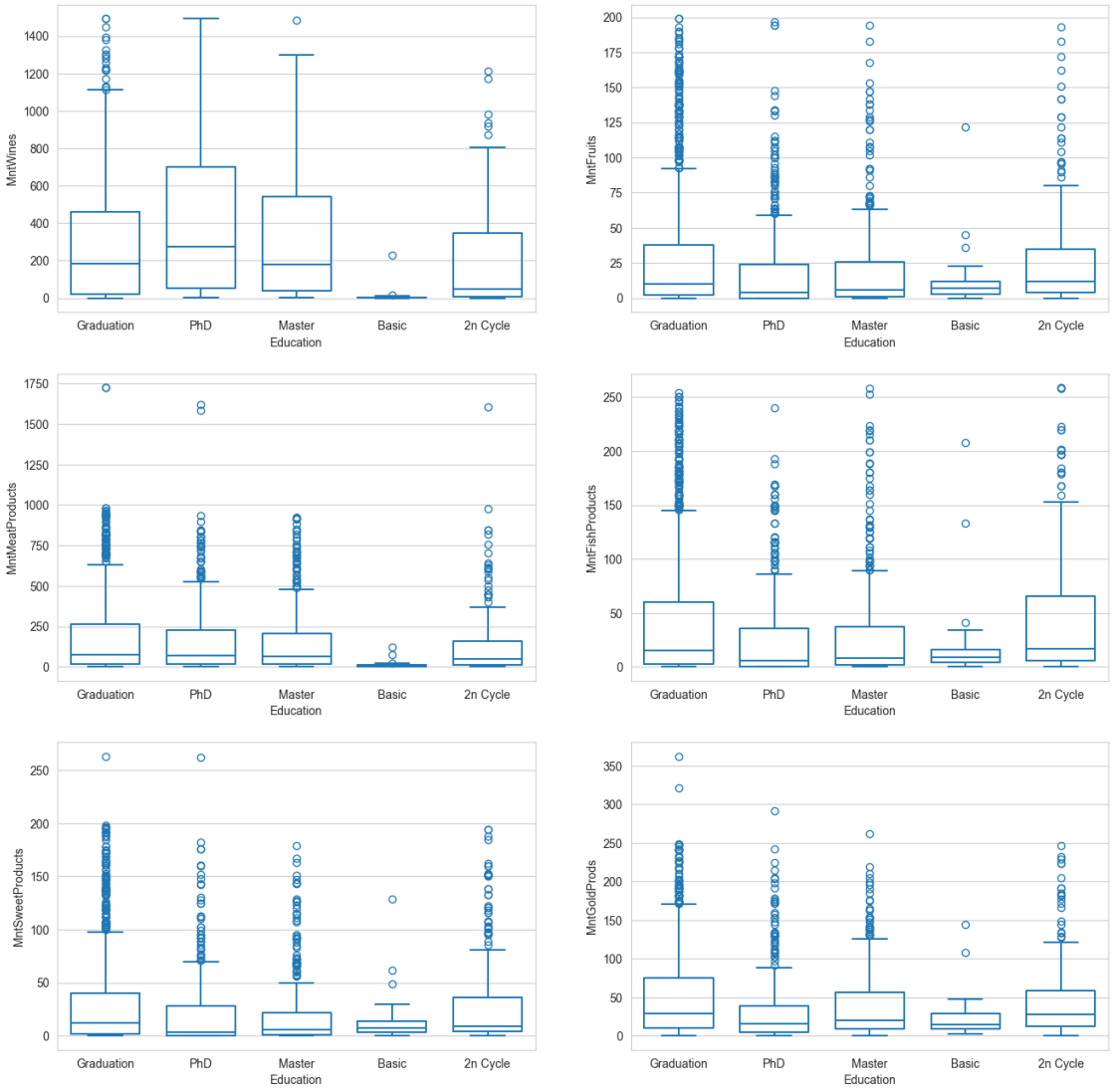


[104]:
```
'Spend category by education '

fig = plt.gcf()
# ax = f.add_subplot(111)
# ax.yaxis.tick_right()
fig.set_size_inches(16, 16)
fig.suptitle('Spend category by education', fontsize=20, weight='bold', color =␣
 ↪'r')
```

```
plt.subplot(3,2,1)
sns.boxplot(data= df,x='Education',y='MntWines', fill=False, gap=.1)

plt.subplot(3,2,2)
sns.boxplot(data= df,x='Education',y='MntFruits', fill=False, gap=.1)

plt.subplot(3,2,3)
sns.boxplot(data= df,x='Education',y='MntMeatProducts', fill=False, gap=.1)

plt.subplot(3,2,4)
sns.boxplot(data= df,x='Education',y='MntFishProducts', fill=False, gap=.1)

plt.subplot(3,2,5)
sns.boxplot(data= df,x='Education',y='MntSweetProducts', fill=False, gap=.1)

plt.subplot(3,2,6)
sns.boxplot(data= df,x='Education',y='MntGoldProds', fill=False, gap=.1)
```

[104]: <Axes: xlabel='Education', ylabel='MntGoldProds'>

Spend category by education



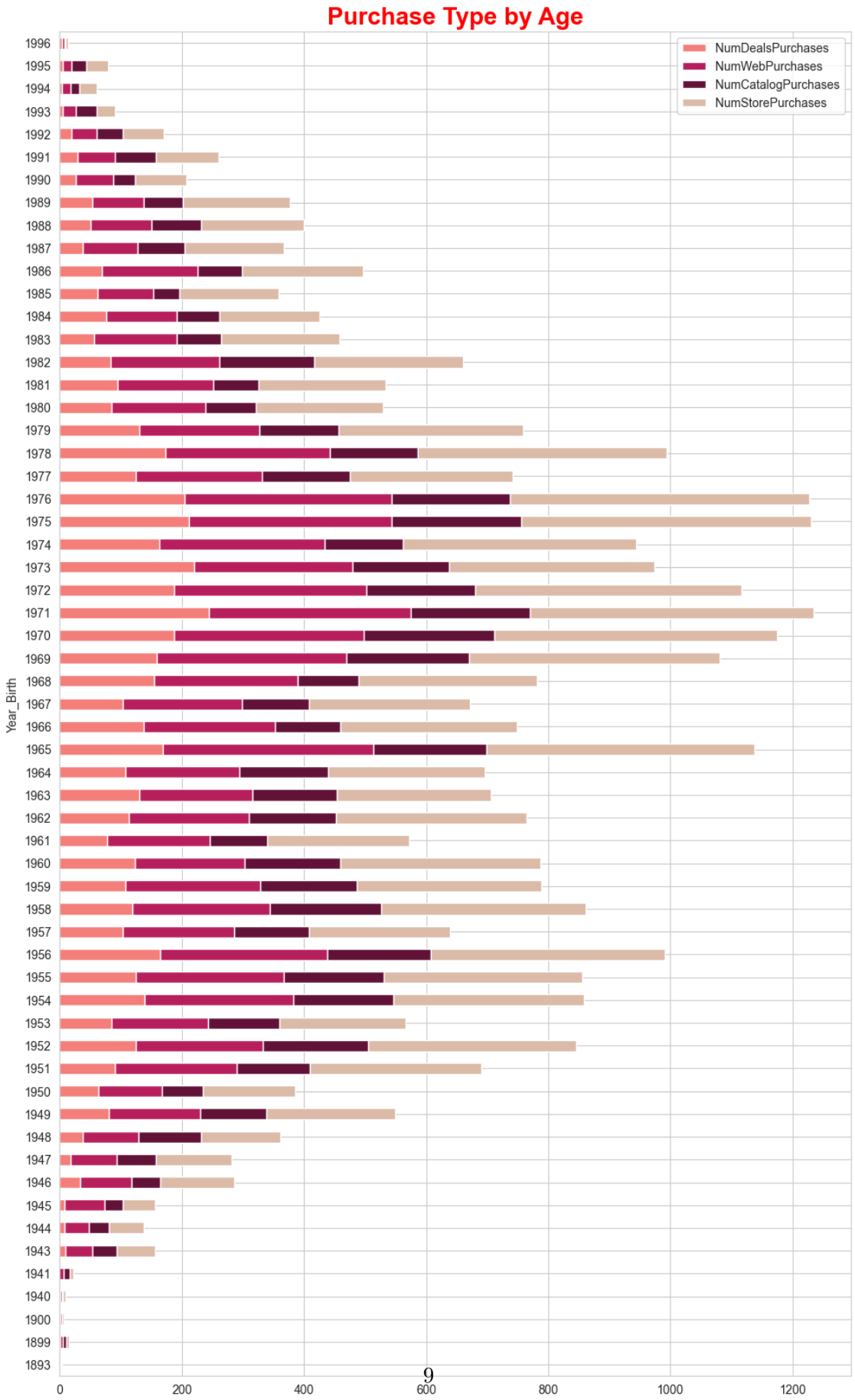
0.2 purchase category

'NumDealsPurchases', 'NumWebPurchases','NumCatalogPurchases', 'NumStorePurchases'

[105]:
```
age = df.loc[: , ['Year_Birth', 'NumDealsPurchases', 'NumWebPurchases',␣
 ↪'NumCatalogPurchases', 'NumStorePurchases']]
# age = df.loc[: , ['Year_Birth', 'NumDealsPurchases', 'NumWebPurchases']]

age = age.groupby(by='Year_Birth').sum()
font_color = '#525252'
```

```
csfont = {'fontname':'Georgia'} # title font
hfont = {'fontname':'Calibri'} # main font
colors = ['#f47e7a', '#b71f5c', '#621237', '#dbbaa7']

ax = age.plot.barh(align='center', stacked=True, figsize=(10, 16), color=colors)
plt.tight_layout()
plt.title('Purchase Type by Age',fontsize=20, weight='bold', color = 'r')
```

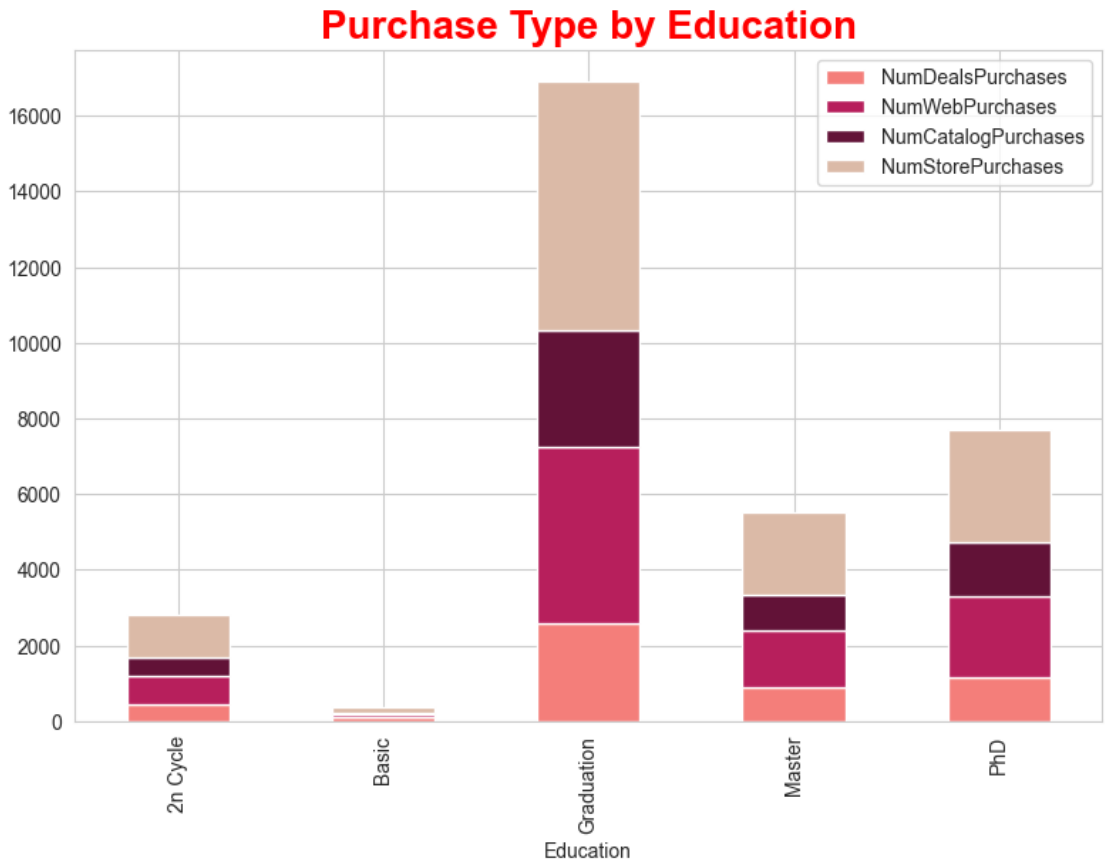
[105]: Text(0.5, 1.0, 'Purchase Type by Age')


Purchase Type by Age
NumDealsPurchases
NumWebPurchases
NumCatalogPurchases
NumStorePurchases
Year_Birth
1996
1995
1994
1993
1992
1991
1990
1989
1988
1987
1986
1985
1984
1983
1982
1981
1980
1979
1978
1977
1976
1975
1974
1973
1972
1971
1970
1969
1968
1967
1966
1965
1964
1963
1962
1961
1960
1959
1958
1957
1956
1955
1954
1953
1952
1951
1950
1949
1948
1947
1946
1945
1944
1943
1941
1940
1900
1899
1893
0
200
400
600
800
1000
1200

Purchase Type by Education

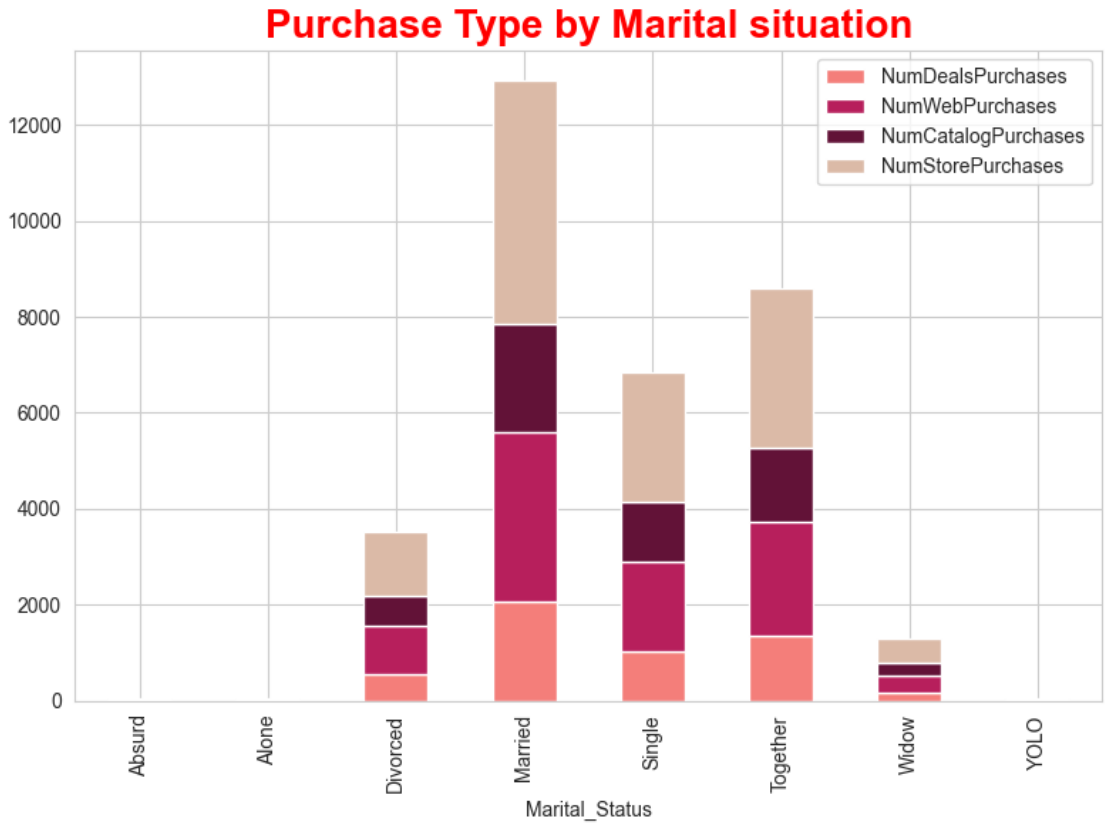
[106]:
```
education = df.loc[: , ['Education', 'NumDealsPurchases', 'NumWebPurchases',␣
 ↪'NumCatalogPurchases', 'NumStorePurchases']]
# age = df.loc[: , ['Year_Birth', 'NumDealsPurchases', 'NumWebPurchases']]

education = education.groupby(by='Education').sum()
font_color = '#525252'
csfont = {'fontname':'Georgia'} # title font
hfont = {'fontname':'Calibri'} # main font
colors = ['#f47e7a', '#b71f5c', '#621237', '#dbbaa7']

ax = education.plot.bar(align='center', stacked=True, figsize=(8, 6),␣
 ↪color=colors)

plt.tight_layout()
plt.title('Purchase Type by Education',fontsize=20, weight='bold', color = 'r')
```

[106]:
```
Text(0.5, 1.0, 'Purchase Type by Education')
```



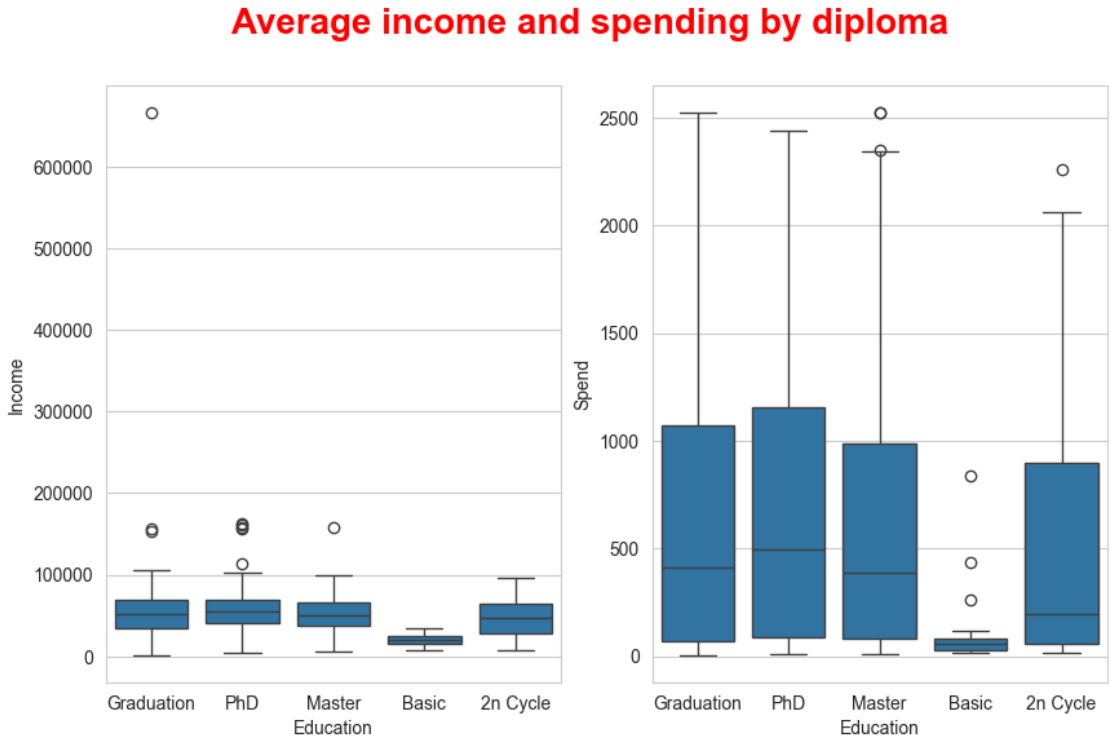
Purchase Type by Marital_Situation

[107]:
```
marry = df.loc[: , ['Marital_Status', 'NumDealsPurchases', 'NumWebPurchases',␣
 ↪'NumCatalogPurchases', 'NumStorePurchases']]
# age = df.loc[: , ['Year_Birth', 'NumDealsPurchases', 'NumWebPurchases']]

marry = marry.groupby(by='Marital_Status').sum()
font_color = '#525252'
csfont = {'fontname':'Georgia'} # title font
hfont = {'fontname':'Calibri'} # main font
colors = ['#f47e7a', '#b71f5c', '#621237', '#dbbaa7']

ax = marry.plot.bar(align='center', stacked=True, figsize=(8, 6), color=colors)
plt.title('Purchase Type by Marital situation',fontsize=20, weight='bold',␣
 ↪color = 'r')
plt.tight_layout()
```



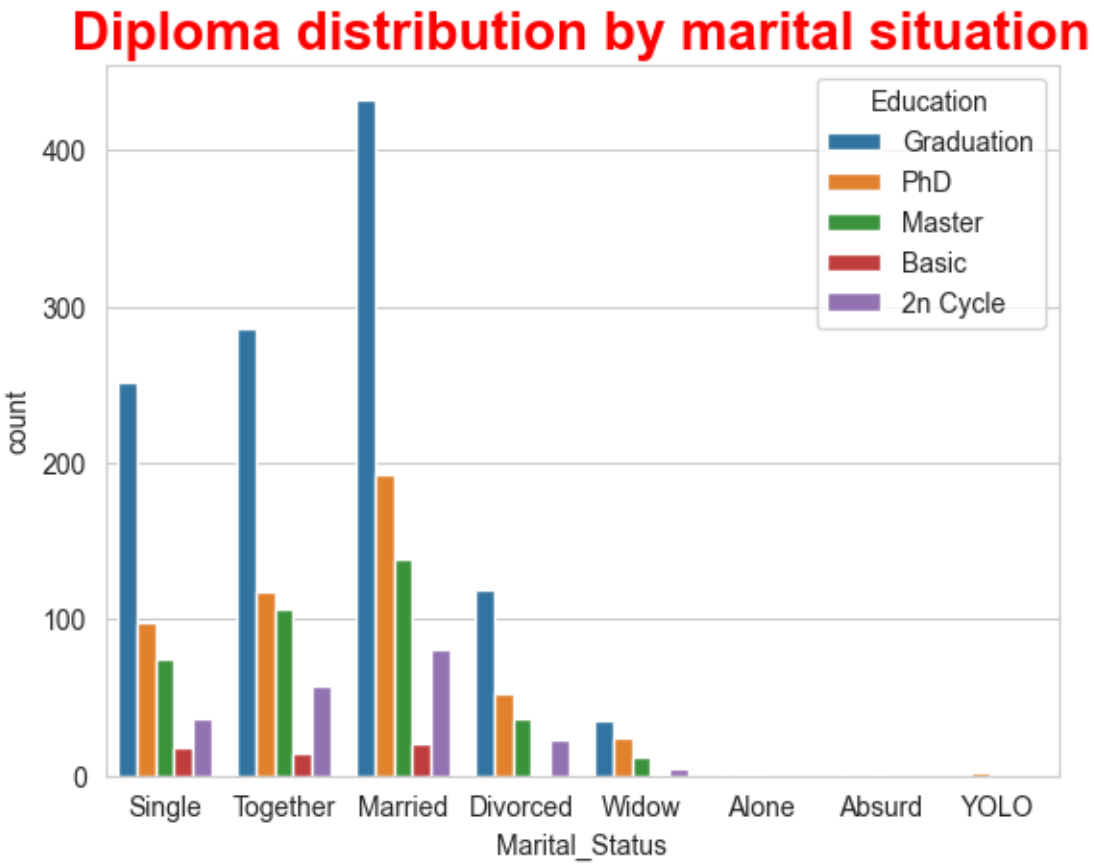
[108]:
```
spend_education = df.loc[: , ['Education',␣
 ↪'MntWines','MntMeatProducts','MntFruits','MntFishProducts',␣
 ↪'MntSweetProducts', 'MntGoldProds']]
spend_education['Spend'] =␣
 ↪df[['MntWines','MntMeatProducts','MntFruits','MntFishProducts',␣
 ↪'MntSweetProducts', 'MntGoldProds']].sum(axis=1)
spend_education

fig = plt.gcf()
# ax = f.add_subplot(111)
# ax.yaxis.tick_right()
fig.suptitle('Average income and spending by diploma', fontsize=20,␣
 ↪weight='bold', color = 'r')

fig.set_size_inches(10, 6)
plt.subplot(1,2,1)
sns.boxplot(data= df,x='Education',y='Income')
plt.subplot(1,2,2)
sns.boxplot(data=spend_education,x='Education',y='Spend')
```

[108]:
```
<Axes: xlabel='Education', ylabel='Spend'>
```



[109]:
```
t = df.loc[: , ['Marital_Status', 'Education']]
sns.countplot(x='Marital_Status', hue= 'Education',data=t)
plt.title('Diploma distribution by marital situation',fontsize=20,␣
 ↪weight='bold', color = 'r')
```

[109]:
```
Text(0.5, 1.0, 'Diploma distribution by marital situation')
```



[110]:
```
income = df.loc[: , ['Income','Teenhome', 'Kidhome']]
income = pd.melt(income, id_vars='Income', value_vars = ['Teenhome', 'Kidhome'])
income
```

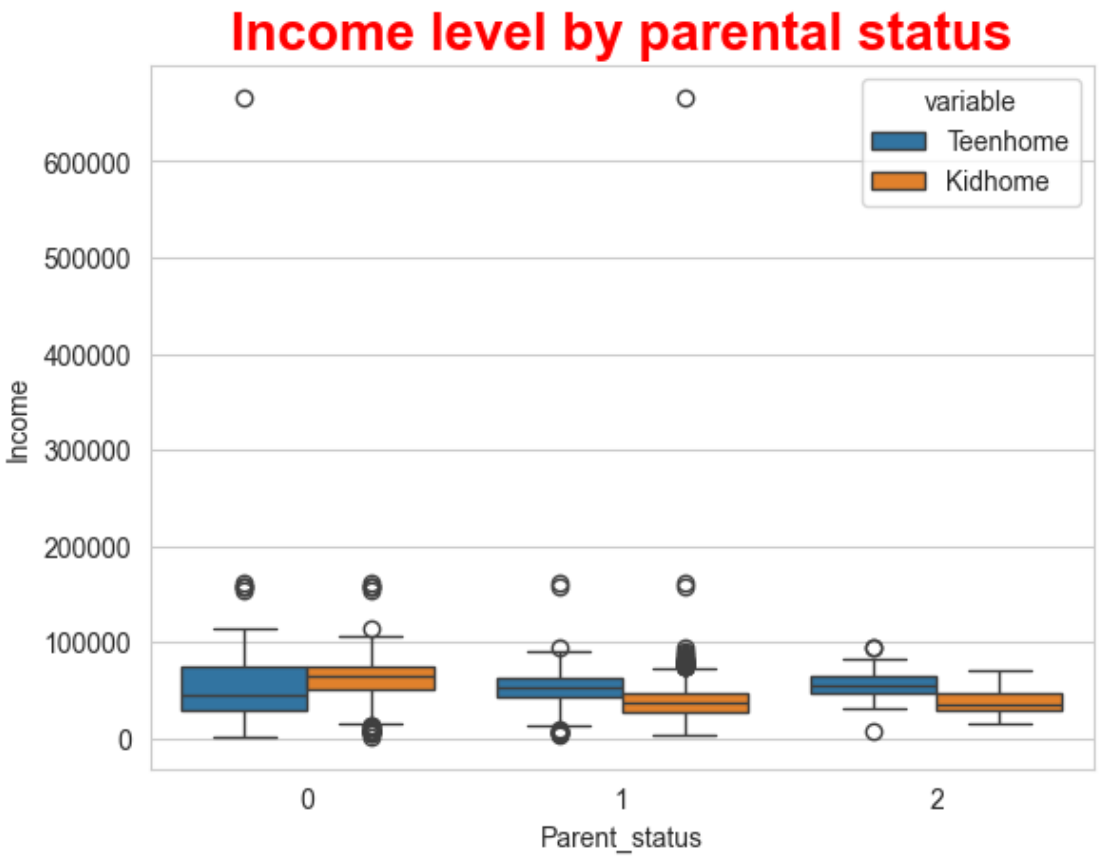
[110]:

	Income	variable	value
0	58138.0	Teenhome	0
1	46344.0	Teenhome	1
2	71613.0	Teenhome	0
3	26646.0	Teenhome	0
4	58293.0	Teenhome	0
…	…	…	…
4475	61223.0	Kidhome	0

```
4476  64014.0   Kidhome      2
4477  56981.0   Kidhome      0
4478  69245.0   Kidhome      0
4479  52869.0   Kidhome      1

[4480 rows x 3 columns]
```

[111]:
```
sns.boxplot(data= income,x='value',y='Income', hue = 'variable')
plt.xlabel('Parent_status')
plt.title('Income level by parental status',fontsize=20, weight='bold', color =␣
↪'r')
```

[111]:
```
Text(0.5, 1.0, 'Income level by parental status')
```



[112]:
```
spend_parent = df.loc[: , ['Teenhome', 'Kidhome']]
spend_parent['Spend'] = spend_education['Spend']
spend_parent = pd.melt(spend_parent, id_vars='Spend', value_vars = ['Teenhome',␣
↪'Kidhome'])
spend_parent
```

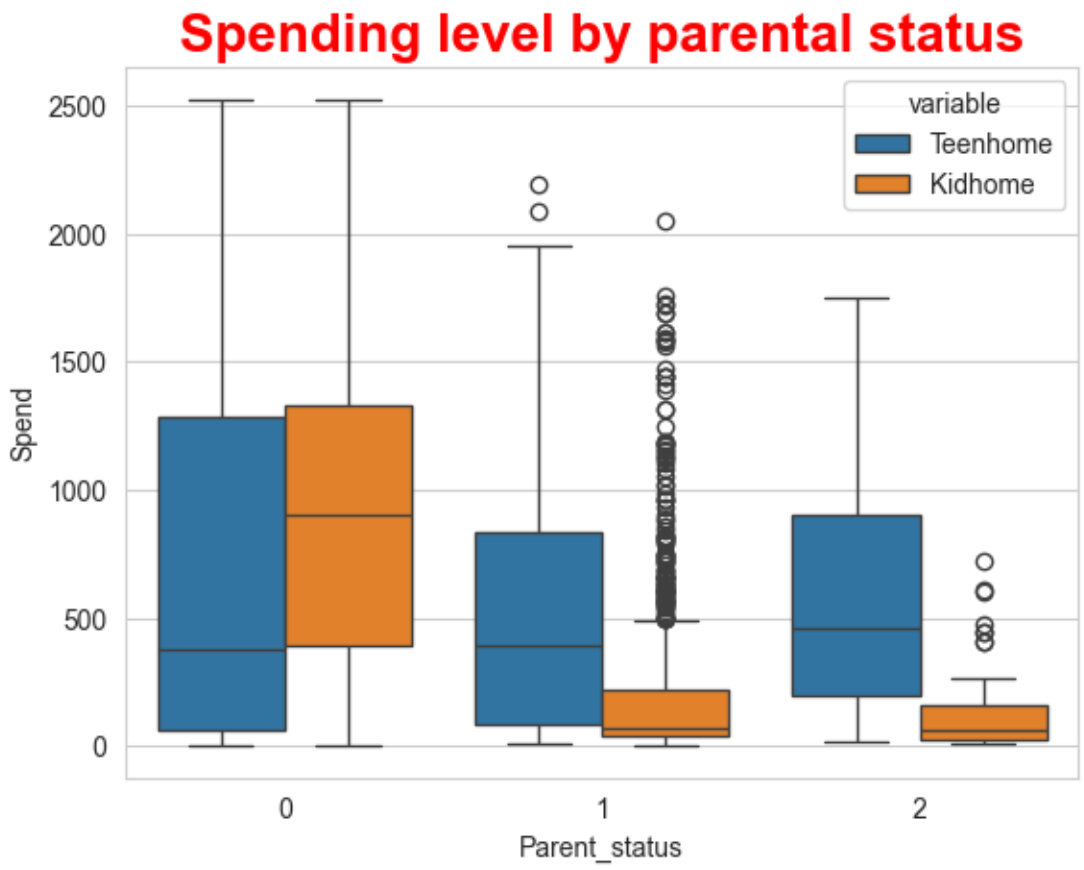
[112]:

	Spend	variable	value
0	1617	Teenhome	0
1	27	Teenhome	1
2	776	Teenhome	0
3	53	Teenhome	0
4	422	Teenhome	0
…	…	…	…
4475	1341	Kidhome	0
4476	444	Kidhome	2
4477	1241	Kidhome	0
4478	843	Kidhome	0
4479	172	Kidhome	1

[4480 rows x 3 columns]

[113]:
```
sns.boxplot(data= spend_parent,x='value',y='Spend', hue = 'variable')
plt.xlabel('Parent_status')

plt.title('Spending level by parental status',fontsize=20, weight='bold', color␣
↪= 'r')
```

[113]: Text(0.5, 1.0, 'Spending level by parental status')



[114]:
```
spend_income = df.loc[: , ['Income',␣
 ↪'MntWines','MntMeatProducts','MntFruits','MntFishProducts',␣
 ↪'MntSweetProducts', 'MntGoldProds']]
# spend_income = pd.melt(spend_income, id_vars='Income', value_vars =␣
 ↪['MntWines','MntMeatProducts','MntFruits','MntFishProducts',␣
 ↪'MntSweetProducts', 'MntGoldProds'])
spend_income
```

[114]:

	Income	MntWines	MntMeatProducts	MntFruits	MntFishProducts	\
0	58138.0	635	546	88	172	
1	46344.0	11	6	1	2	
2	71613.0	426	127	49	111	
3	26646.0	11	20	4	10	
4	58293.0	173	118	43	46	
…	…	…	…	…	…	
2235	61223.0	709	182	43	42	
2236	64014.0	406	30	0	0	
2237	56981.0	908	217	48	32	
2238	69245.0	428	214	30	80	
2239	52869.0	84	61	3	2	

	MntSweetProducts	MntGoldProds
0	88	88
1	1	6
2	21	42
3	3	5
4	27	15
…	…	…
2235	118	247
2236	0	8
2237	12	24
2238	30	61
2239	1	21

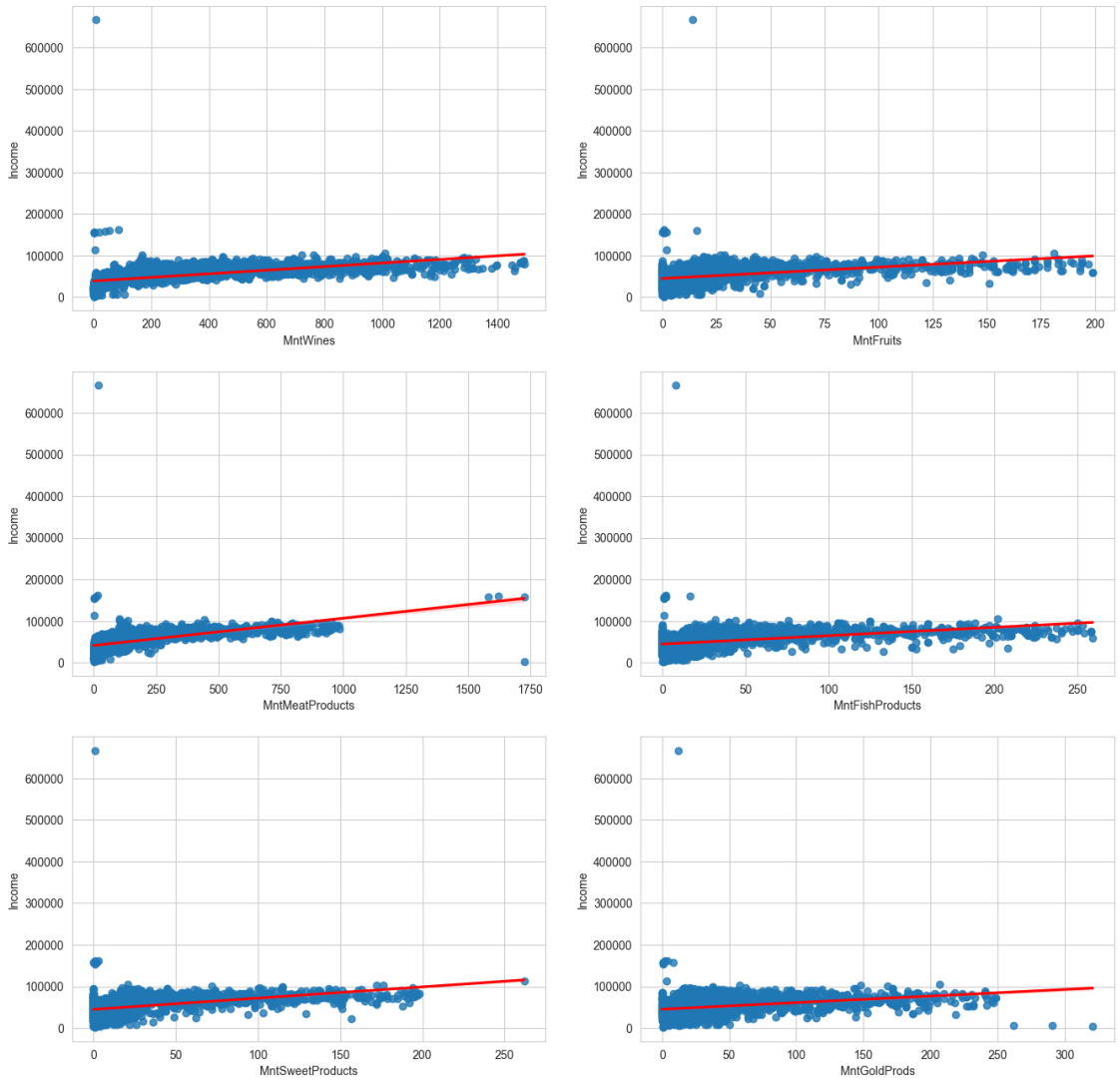
[2240 rows x 7 columns]

[115]:
```
fig = plt.gcf()

fig.set_size_inches(16, 16)
fig.suptitle('Spend category by income', fontsize=20, weight='bold', color =␣
 ↪'r')

plt.subplot(3,2,1)
```

```
sns.regplot(data = spend_income,x = 'MntWines', y = 'Income',line_kws =␣
 ↪{"color": "red"})
plt.subplot(3,2,2)
sns.regplot(data = spend_income,x = 'MntFruits', y = 'Income',line_kws =␣
 ↪{"color": "red"})
plt.subplot(3,2,3)
sns.regplot(data = spend_income,x = 'MntMeatProducts', y = 'Income',line_kws =␣
 ↪{"color": "red"})
plt.subplot(3,2,4)
sns.regplot(data = spend_income,x = 'MntFishProducts', y = 'Income',line_kws =␣
 ↪{"color": "red"})
plt.subplot(3,2,5)
sns.regplot(data = spend_income,x = 'MntSweetProducts', y = 'Income',line_kws =␣
 ↪{"color": "red"})
plt.subplot(3,2,6)
sns.regplot(data = spend_income,x = 'MntGoldProds', y = 'Income',line_kws =␣
 ↪{"color": "red"})
```

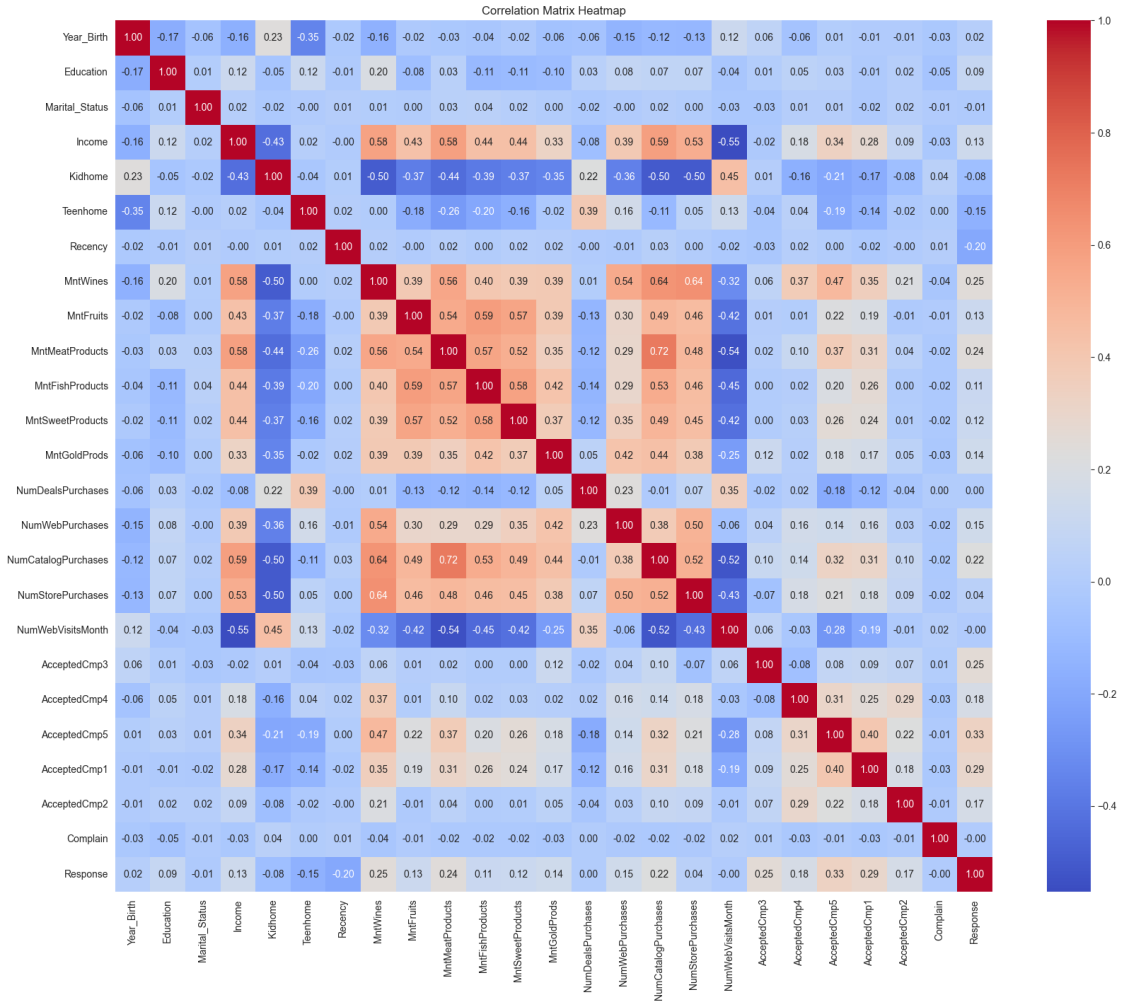
[115]: <Axes: xlabel='MntGoldProds', ylabel='Income'>

Spend category by income



[116]:
```
from sklearn.preprocessing import LabelEncoder
e=LabelEncoder()

df['Education'] = e.fit_transform(df['Education'])
df['Marital_Status'] = e.fit_transform(df['Marital_Status'])

' Heat map'
corr_matrix = df.drop(['Dt_Customer'], axis=1).corr()

# Set up the matplotlib figure
```

```
plt.figure(figsize=(20, 16))

# Draw the heatmap
sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap='coolwarm')

# Add title
plt.title('Correlation Matrix Heatmap')

# Show the plot
plt.show()
```



[117]:
```
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import accuracy_score
df = df.drop(['Dt_Customer'], axis= 1)
df = df.dropna()
```

[118]:
```
X=df.drop(['Response'],axis=1)
Y=df['Response']
```

[119]:
```
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2,␣
↪random_state=42)
x_train
```

[119]:

	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	\
209	1954	2	2	64497.0	0	1	
53	1977	2	2	82582.0	0	0	
184	1961	1	3	28249.0	0	0	
2115	1969	3	5	66476.0	0	1	
728	1965	3	3	56962.0	2	1	
…	…	…	…	…	…	…	
1655	1978	3	4	35544.0	1	0	
1108	1974	0	3	65463.0	1	0	
1143	1962	4	4	33419.0	0	1	
1307	1965	4	3	81051.0	0	0	
873	1981	0	4	42395.0	1	1	

	Recency	MntWines	MntFruits	MntMeatProducts	…	NumWebPurchases	\
209	17	1170	48	320	…	11	
53	54	510	120	550	…	4	
184	80	1	9	7	…	2	
2115	80	742	28	152	…	6	
728	60	292	3	77	…	6	
…	…	…	…	…	…	…	
1655	77	30	5	23	…	2	
1108	17	391	32	70	…	6	
1143	76	56	0	12	…	2	
1307	43	1142	29	249	…	5	
873	35	48	13	57	…	3	

	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth	AcceptedCmp3	\
209	4	9	8	1	
53	9	7	1	1	
184	0	3	6	0	
2115	8	10	4	0	
728	3	5	7	0	
…	…	…	…	…	
1655	0	3	7	0	
1108	2	9	5	0	
1143	0	4	7	0	
1307	5	12	2	0	
873	1	4	7	0	

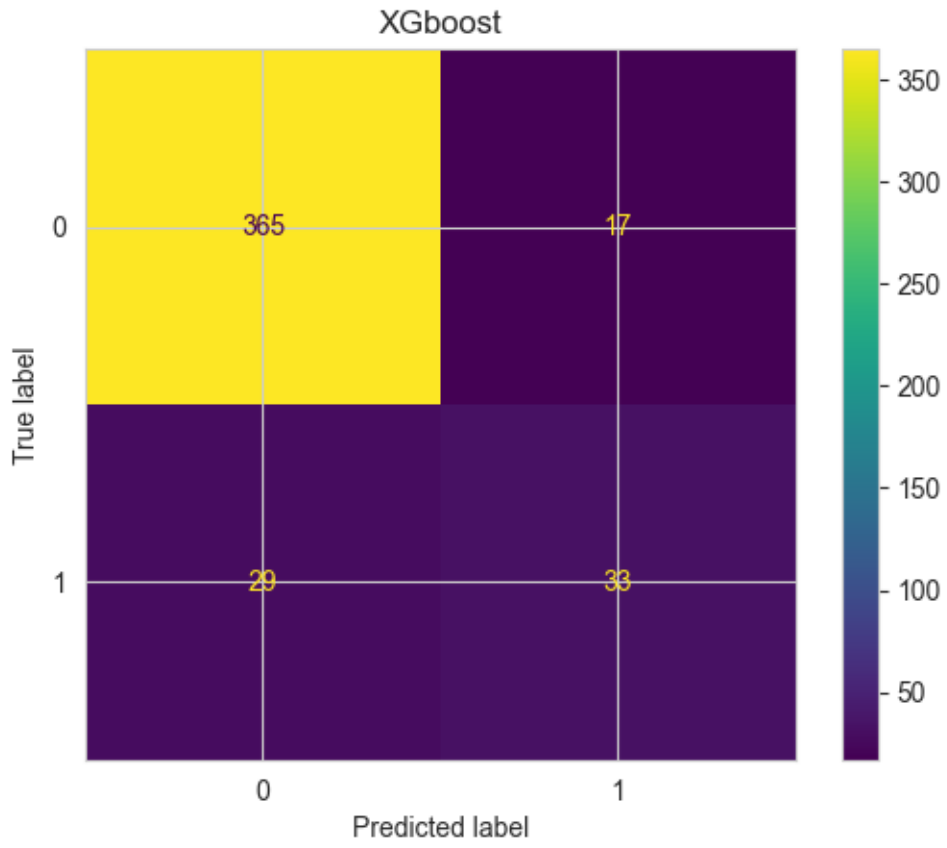
	AcceptedCmp4	AcceptedCmp5	AcceptedCmp1	AcceptedCmp2	Complain

```
209     0     0     0     0     0
53      0     0     1     0     0
184     0     0     0     0     0
2115    0     0     0     0     0
728     0     0     0     0     0
…       …     …     …     …     …
1655    0     0     0     0     0
1108    1     0     0     0     0
1143    0     0     0     0     0
1307    1     1     0     0     0
873     0     0     0     0     0

[1772 rows x 24 columns]
```

[120]:
```
from xgboost import XGBClassifier
model = XGBClassifier()
model.fit(x_train, y_train)
y_pred_xg = model.predict(x_test)
#Score/Accuracy
acc_xg=model.score(x_test, y_test)*100
print ('Train : ', accuracy_score(y_train,model.predict(x_train))*100)
print ('Test : ', acc_xg)
```

```
Train :  99.20993227990971
Test :  89.63963963963964
```

[121]:
```
from sklearn.metrics import ConfusionMatrixDisplay,confusion_matrix,␣
 ↪classification_report

disp = ConfusionMatrixDisplay.from_predictions(y_test, y_pred_xg)
plt.title ('XGboost')
plt.show()
```


XGboost
365
17
29
33
0
1
True label
0
1
Predicted label
350
300
250
200
150
100
50