Personality classification based on Spending Behavior

for Business Implications | *Presented by Team 4*

Gurkirat Singh Sekhon | Jiaqi Wang | Karen Yang | Krunal Patel | Shunran Zhang | Yangming Zhang | Ye Yuan

Summary Report

Overview

1 2 3 4 5 6 7

1 | Problem Statement

1.1 Problem Statement
1.2 Data: Overview

2 | Data Preparation

2. Data Cleaning

- Req. Information > New data frame
- Removed: “Na”, “Savings”
- Separate CTR: area code & categories
- Absolute Spending

3 | Exploratory Data Analysis

3.1 Exploratory Analysis

- State/Spending Histogram

4 | Model Building

4.1 Separating transactions by categories(group by State)
4.2 Normalize Spending (Tax)
4.3 Category-wise spending

5 | Modeling

5.1 K-means Clustering
5.2 Building model by hypothesis (for 3 Ques)
5.3 Logistic Regression
5.4 Linear Regression

6 | Output

6. Personality Estimation

7 | Business Insights

7. Business Insights

1.1 Problem Statement

"Studying consumer behavior is the best way to capture value from your consumer data"

McKinsey

McKinsey & Company

Objective

- Understand transaction pattern of 40 individuals, and classify them into different personality types based on their spending behavior, for business implications.
- Behavioural/Psychographic segmentation

1.2 Data Overview

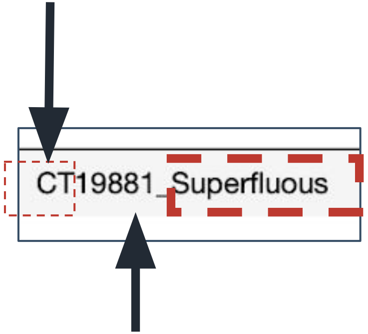
Oper Dictionary

1,2. Checking account debit-main | 3,4. Checking account debit- main
5,6. Checking account debit/credit-linked | 7,8. Checking account debit/credit-ext

	TRANSNumber	TimeStamp	DA	Oper	CTR
0	1	2010-05-11	-588.30	3	CT19881_Superfluous
1	2	2010-05-13	661.03	7	NaN
2	3	2010-05-14	980.57	7	NaN
3	4	2010-05-20	-566.35	1	PA11761_Superfluous
4	5	2010-05-23	-770.32	1	NY22638_Investment
5	6	2010-05-25	974.05	7	NaN

CTR Dictionary

1. First two letters: US State



4. Three Categories:
- **Essentials**(utility bills, food)
- **Superfluous** (expensive items, non-business)
- **Investment** (book, education)

2. Five Digits: Machine number
3. Underscore (separator)

TRANS Number
Time Stamp

2. Data Preparation

- Removing the missing value(“savings” & “Spending without CTR info) and blank columns
- Creating new columns for spreated CTR info
- Absolute value of spending.

```
xls = pd.ExcelFile('Personal Financial Example.xlsx')
wb = openpyxl.load_workbook('Personal Financial Example.xlsx')
sheets = wb.get_sheet_names()
```

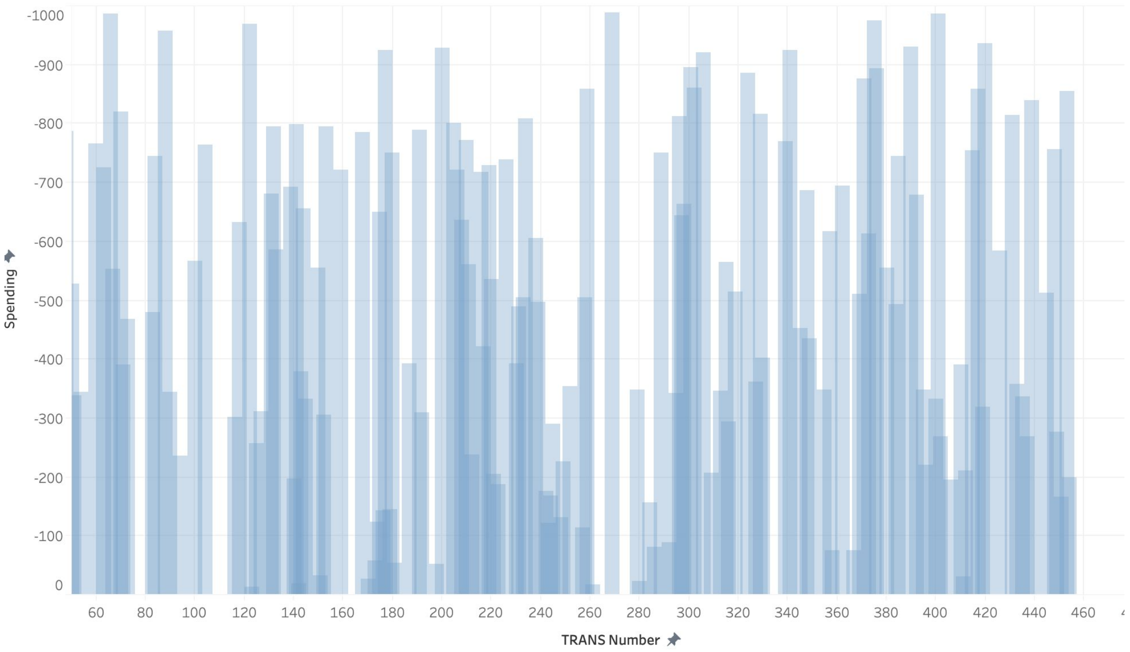
```
for i in range(len(sheets)):
    df = pd.read_excel(xls,sheets[i])
    df = df[['TRANSNumber','TimeStamp','DA','Oper','CTR']]
    df = df.dropna()
    df['CTR_1'] = df['CTR'].str[0:2]
    df['CTR_2'] = df['CTR'].str[8:]
    df['DA'] = abs(df['DA'])

    tem1 = df.groupby(['CTR_1','CTR_2']).describe()
    tem2 = tax_calculate(tem1)
    tem3 = consum_por(tem2)
    portions[i,:] = tem3
```

	TRANSNumber	TimeStamp	DA	Oper	CTR
0	1	2010-05-11	-588.30	3	CT19881_Superfluous
1	2	2010-05-13	661.03	7	NaN
2	3	2010-05-14	980.57	7	NaN
3	4	2010-05-20	-566.35	1	PA11761_Superfluous
4	5	2010-05-23	-770.32	1	NY22638_Investment
5	6	2010-05-25	974.05	7	NaN

	TRANSNumber	TimeStamp	DA	Oper	CTR	CTR_1	CTR_2
8	9	2016-12-10	67.63	1	PA77402_Investment	PA	Investment
9	10	2016-12-14	46.78	5	CT28315_Investment	CT	Investment
10	11	2016-12-15	20.92	5	NJ73559_Essencials	NJ	Essencials
11	12	2016-12-16	37.09	5	NJ13624_Investment	NJ	Investment
15	16	2016-12-17	31.59	1	NY22389_Investment	NY	Investment
17	18	2016-12-22	48.89	1	NY42686_Investment	NY	Investment
19	20	2016-12-25	10.99	5	NY15737_Superfluous	NY	Superfluous
23	24	2017-01-02	23.70	3	NJ53834_Essencials	NJ	Essencials
26	27	2017-01-07	66.17	3	PA26119_Investment	PA	Investment
27	28	2017-01-09	4.95	5	NJ68810_Superfluous	NJ	Superfluous
33	34	2017-01-16	15.21	1	CT19366_Essencials	CT	Essencials

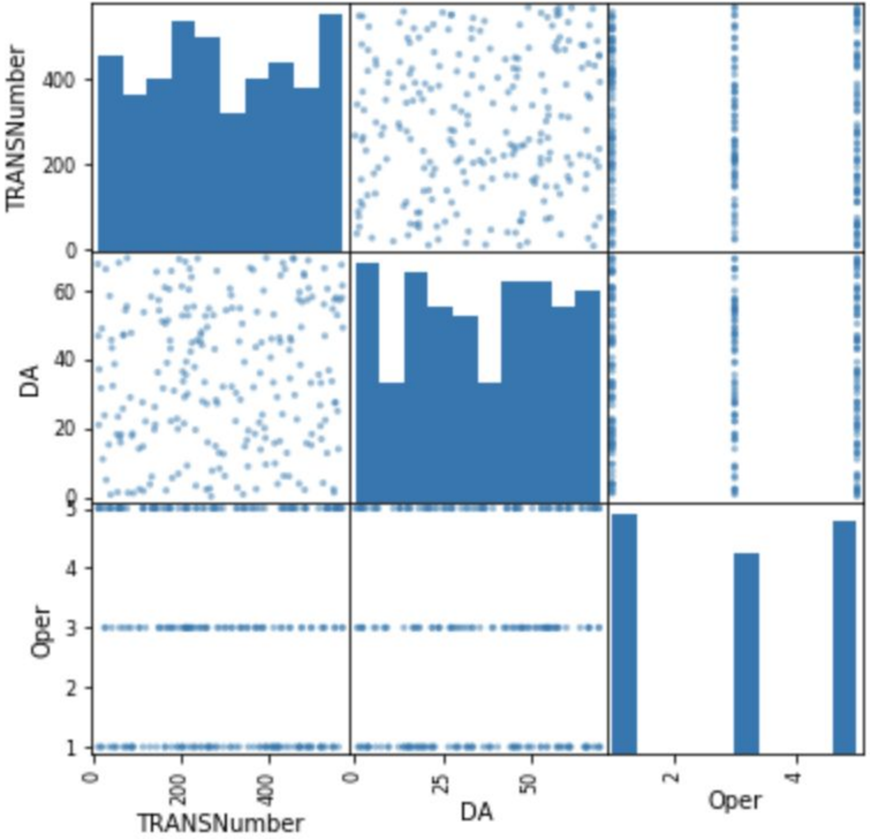
3.1 EDA-Overall Data distribution

	TRANSNumber	TimeStamp	DA	Oper	CTR	CTR_1	CTR_2
8	9	2016-12-10	67.63	1	PA77402_Investment	PA	Investment
9	10	2016-12-14	46.78	5	CT28315_Investment	CT	Investment
10	11	2016-12-15	20.92	5	NJ73559_Essencials	NJ	Essencials
11	12	2016-12-16	37.09	5	NJ13624_Investment	NJ	Investment
15	16	2016-12-17	31.59	1	NY22389_Investment	NY	Investment
17	18	2016-12-22	48.89	1	NY42686_Investment	NY	Investment
19	20	2016-12-25	10.99	5	NY15737_Superfluous	NY	Superfluous
23	24	2017-01-02	23.70	3	NJ53834_Essencials	NJ	Essencials
26	27	2017-01-07	66.17	3	PA26119_Investment	PA	Investment
27	28	2017-01-09	4.95	5	NJ68810_Superfluous	NJ	Superfluous
33	34	2017-01-16	15.21	1	CT19366_Essencials	CT	Essencials



3.3 Variables correlations

```
df.corr()
pd.scatter_matrix(df, figsize=(6, 6))
plt.show()
```



4.1 Separating Transaction by categories

```
for i in range(len(sheets)):
    df = pd.read_excel(xls,sheets[i])
    df = df[['TRANSNumber','TimeStamp','DA','Oper','CTR']]
    df = df.dropna()
    df['CTR_1'] = df['CTR'].str[0:2]
    df['CTR_2'] = df['CTR'].str[8:]
    df['DA'] = abs(df['DA'])

    tem1 = df.groupby(['CTR_1','CTR_2']).describe()
    tem2 = tax_calculate(tem1)
    tem3 = consum_por(tem2)
    portions[i,:] = tem3
```

- Used groupby to separate transactions by different areas and different categories.
- Found the values by location of State and expenditure category.

```
tem1
```

		TRANSNumber								DA					Oper		
		count	mean	std	min	25%	50%	75%	max	count	mean	...	75%	max	count	mean	std
CTR_1	CTR_2																
CT	Essencials	14.0	245.285714	150.457471	34.0	150.00	202.5	359.75	498.0	14.0	30.162857	...	38.8175	60.84	14.0	3.428571	1.785165
	Investment	16.0	282.562500	146.022358	10.0	196.50	293.5	393.75	526.0	16.0	35.284375	...	55.0700	65.95	16.0	3.125000	1.707825
	Superfluous	20.0	339.100000	183.504740	47.0	173.25	387.0	490.00	568.0	20.0	30.107000	...	47.8575	69.21	20.0	3.300000	1.866604
NJ	Essencials	28.0	288.428571	187.162917	11.0	141.00	280.0	452.50	569.0	28.0	33.856071	...	51.1425	69.22	28.0	3.357143	1.725930
	Investment	19.0	280.789474	136.796515	12.0	185.00	296.0	388.00	491.0	19.0	32.962105	...	48.2850	68.30	19.0	3.000000	1.632993
	Superfluous	23.0	314.304348	164.632493	28.0	197.00	371.0	448.00	551.0	23.0	33.490435	...	46.4100	68.68	23.0	2.652174	1.668115
NY	Essencials	17.0	293.352941	172.696606	75.0	172.00	244.0	410.00	554.0	17.0	37.996471	...	54.5900	60.32	17.0	2.764706	1.714986
	Investment	17.0	325.294118	184.205308	16.0	211.00	350.0	468.00	556.0	17.0	36.475882	...	51.0300	60.44	17.0	2.647059	1.617914
	Superfluous	19.0	305.631579	167.211181	20.0	202.50	286.0	440.00	559.0	19.0	34.098947	...	54.4550	69.42	19.0	2.789474	1.750522
PA	Essencials	14.0	272.500000	150.875777	68.0	154.00	255.5	354.00	550.0	14.0	41.303571	...	54.1900	66.54	14.0	3.428571	1.603567
	Investment	23.0	264.347826	167.887031	9.0	145.50	235.0	423.00	548.0	23.0	39.024348	...	57.6400	69.34	23.0	2.652174	1.555305
	Superfluous	18.0	292.500000	159.576738	54.0	206.50	246.5	421.25	566.0	18.0	41.185556	...	59.6800	69.21	18.0	2.777778	1.664705

12 rows × 24 columns

4.2 Normalizing Spending using tax rates

```
for i in range(len(sheets)):
    df = pd.read_excel(xls,sheets[i])
    df = df[['TRANSNumber','TimeStamp','DA','Oper','CTR']]
    df = df.dropna()
    df['CTR_1'] = df['CTR'].str[0:2]
    df['CTR_2'] = df['CTR'].str[8:]
    df['DA'] = abs(df['DA'])

    tem1 = df.groupby(['CTR_1','CTR_2']).describe()
    tem2 = tax_calculate(tem1)
    tem3 = consum_por(tem2)
    portions[i,:] = tem3
```

```
tem2
```

```
array([ 449.09478  ,  600.398925 ,  640.37589  , 1010.7730125,
        667.77105  ,  808.794    ,  703.267175 ,  704.2672175,
        680.274    ,  612.945    ,  969.3648   ,  778.407    ])
```

- Aim to eliminate external influence on spending behaviour.
- Taxes can hinder spending that an individual would actually spend.
- Therefore, there is a need to remove influence of taxes on spending so that actual spending behaviour can be determined.

4.3 Catergory Wise Spending

```
for i in range(len(sheets)):
    df = pd.read_excel(xls,sheets[i])
    df = df[['TRANSNumber','TimeStamp','DA','Oper','CTR']]
    df = df.dropna()
    df['CTR_1'] = df['CTR'].str[0:2]
    df['CTR_2'] = df['CTR'].str[8:]
    df['DA'] = abs(df['DA'])

    tem1 = df.groupby(['CTR_1','CTR_2']).describe()
    tem2 = tax_calculate(tem1)
    tem3 = consum_por(tem2)
    portions[i,:] = tem3
```

```
tem3
```

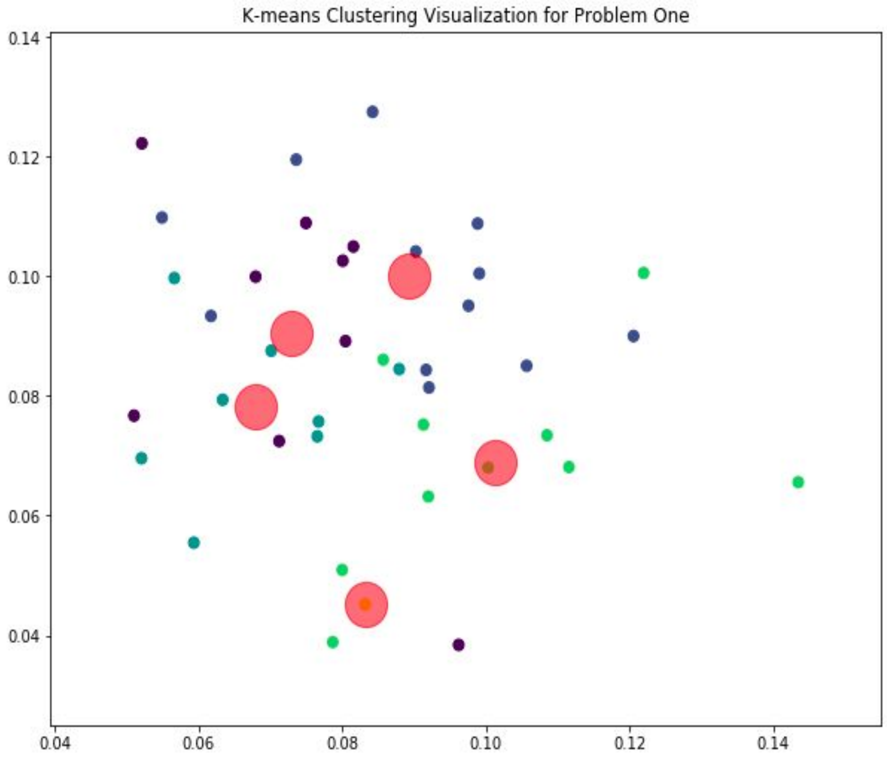
```
array([0.05206454, 0.06960556, 0.07424017, 0.11718112, 0.07741615,
       0.09376525, 0.0815313 , 0.08164723, 0.07886565, 0.07106005,
       0.11238057, 0.09024242])
```

- Spending percentage of each category, which will be later used in modeling.

5.1 K-means clustering

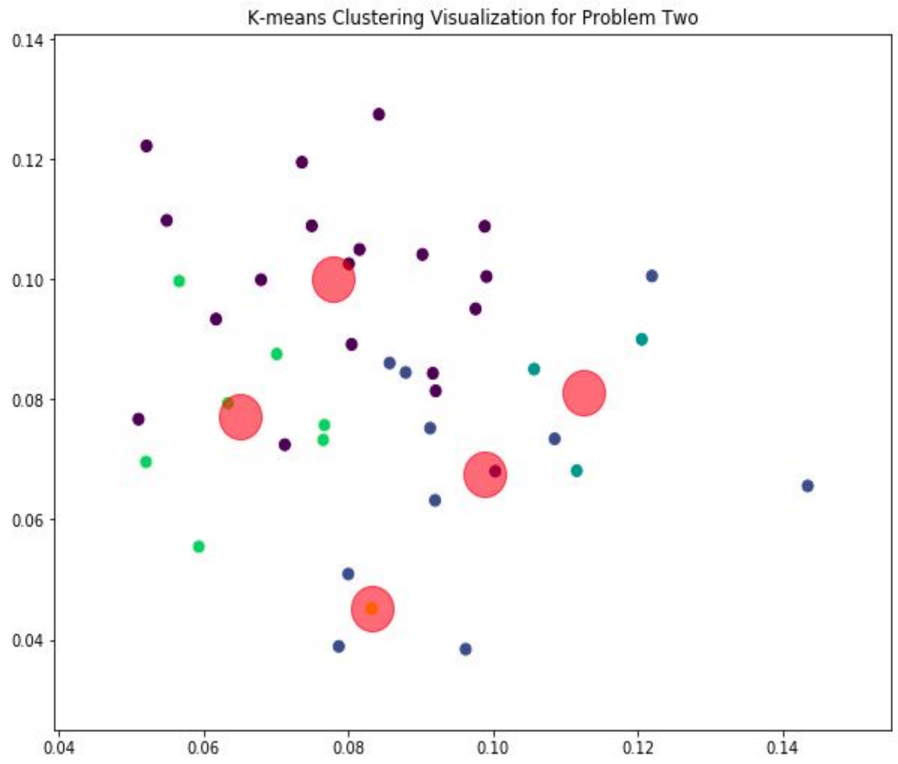
```
from sklearn.cluster import KMeans
kmeans_q1 = KMeans(n_clusters=5, random_state=1).fit(portions)
kmeans_q1.labels_
```

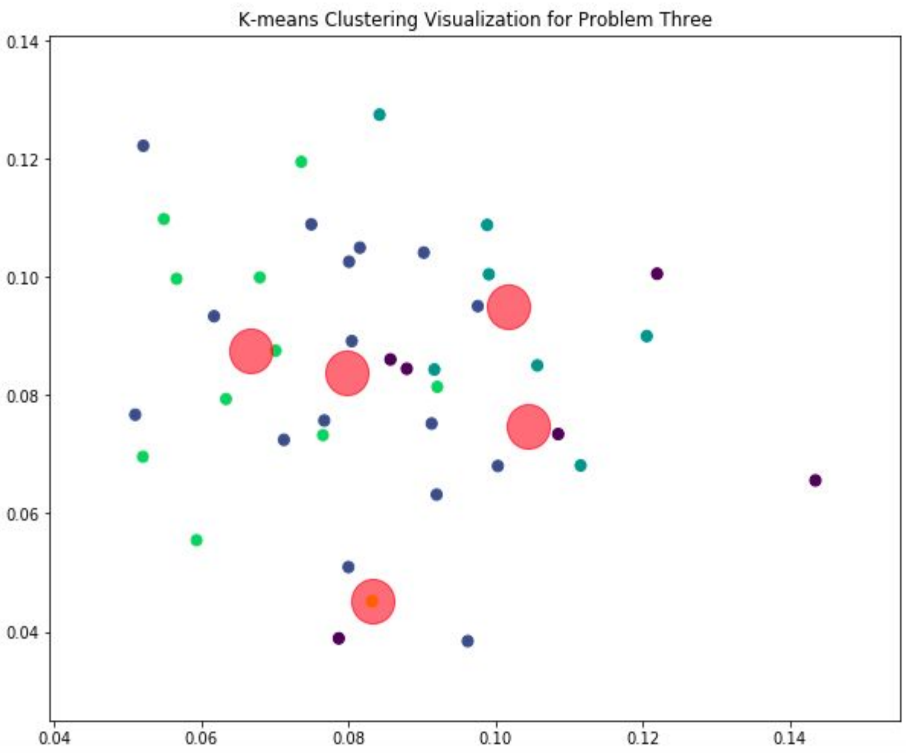
```
plt.figure(figsize=(10,8))
plt.scatter(portions[:, 0], portions[:, 1], c=kmeans_q1.labels_, s=50, cmap='viridis')
centers = kmeans_q1.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='r', s=800, alpha=0.5);
plt.title("K-means Clustering Visualization for Problem One")
```



- "Portions" matrix with 12 columns
- Different random state

5.1 K-means clustering





5.2 Building Model by Hypothesis

This is how each customer spends on each category based on cluster separation. Figure on the right shows the total portions to the first question: **I am a life of party.** The second question is: **I like order**. The third one is : **I have vivid imagination.**

	Category	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
0	Essentials Spending Average Portion	0.338962	0.326538	0.352287	0.333563	0.441820
1	Investment Spending Average Portion	0.349952	0.360666	0.323983	0.305061	0.314466
2	Superfluous Spending Average Portion	0.301064	0.310877	0.358272	0.345600	0.273029

Based on the question, our group thinks people who love party will spend more on superfluous rather than others. So we assign a 0.5 wight to superfluous spending and 0.25 to the others. Figure on the right will show the result of weighted values after assigning weights.

	Category	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
0	Weighted Values for Question 1	0.322760	0.327244	0.348203	0.332456	0.325586
1	Weighted Values for Question 2	0.332234	0.331157	0.346707	0.329446	0.367784
2	Weighted Values for Question 3	0.334982	0.339689	0.339631	0.322321	0.335945

5.2 Building Model by Hypothesis

Based on the weighted values for each question, we have created the answer sheet by sorting out the values, and assigned them with options A, B, C, D, E. The figure on the right shows a quick look of the answer sheet.

Later we will use this answer sheet to find the five attributes in the personality test by SAS

	Q1	Q2	Q3
0	B	D	E
1	E	A	E
2	A	C	A
3	C	E	D
4	D	A	E
5	B	B	E
6	B	D	B
7	B	D	E
8	D	C	D
9	A	C	A
10	B	B	E
11	B	B	A
12	A	C	A
13	E	A	E
14	C	C	D
15	D	A	C
16	E	B	E
17	A	C	A
18	A	A	A
19	A	C	A
20	D	A	D

5.4 Linear Regression

Based on previous assignments, we generate the linear regression model for predicting the outcomes for five personality attributes.

```
** So we have the linear regression model for the five personalities:
**Percentage of Openness = -0.02158 *Answer1 + 0.00724 * Answer2 + 0.00927* Answer2 + 0.52174;
** Percentage of Concientiousness  = -0.00239 *Answer1 - 0.00528 * Answer2 + 0.00644* Answer2 + 0.42196;
** Percentage of Extraversion  = -0.01409 *Answer1 - 0.0001538 * Answer2 - 0.00010867* Answer2 + 0.60706;
** Percentage of Agreebleness  = 0.0077 *Answer1 + 0.00553 * Answer2 - 0.00655* Answer2 + 0.21775;
** Percentage of Neuroticism  = 0.00955 *Answer1 + 0.00589 * Answer2 - 0.01142* Answer2 + 0.59424;
```

5.3 Logistic Regression

B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
0_A	0_B	0_C	0_D	0_E	1_A	1_B	1_C	1_D	1_E	2_A	2_B	2_C	2_D	2_E
0	1	0	0	0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	1	1	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0	0	0	0	0	0	1

```
data new_test;
set new_test;
if Openness < 0.5 and Openness > 0
then op = 0;
if Openness > 0.5 and Openness < 1
then op = 1;
```

5.3 Logistic Regression

We split the outcomes for five personality attributes as binary outcomes and the answers for the three questions as 15 binary options.

```
** Here we genrate the logistic regression for personality tests. beta_n are the intercepts
**logit(Openness) =  -0.0256 * beta1 ;
**logit(Extraversion) = 2.2336 * as05 - 3.3322* beta1;
**logit(Conscientiousness) = -2.9022 * as05 + 4.4427 *beta1;
** logit(Agreebleness) = -1 * as12 -2.2618 * as15 + 3.4657* beta1;
**logit(Neuroticism)  = 0.3943 * as01 + 0.1519 * as02 + 0.6967 * as04 - 0.5485 * as05 + 0.1554 * as11 - 0.2161 * as12 - 0.3924 * as15 -
```

6 Personality type Estimation

Openness	Extraversion	Conscientiousness	Agreebleness	Neuroticism
-0.0256	-3.3322	4.4427	3.4657	0.6053
-0.0256	-1.0986	1.5405	3.4657	0.0603
-0.0256	-3.3322	4.4427	3.4657	-2.2886
-0.0256	-3.3322	4.4427	1.2039	-0.3924
-0.0256	-3.3322	4.4427	3.4657	1.3055
-0.0256	-3.3322	4.4427	1.9924	0.3892
-0.0256	-3.3322	4.4427	3.4657	-2.506
-0.0256	-3.3322	4.4427	3.4657	0.6053
-0.0256	-3.3322	4.4427	3.4657	0.6967
-0.0256	-3.3322	4.4427	3.4657	-2.2886
-0.0256	-3.3322	4.4427	1.9924	0.3892
-0.0256	-3.3322	4.4427	1.9924	-2.7471
-0.0256	-3.3322	4.4427	3.4657	-2.2886
-0.0256	-1.0986	1.5405	3.4657	0.0603
-0.0256	-3.3322	4.4427	3.4657	0
-0.0256	-3.3322	4.4427	3.4657	0.9532
-0.0256	-1.0986	1.5405	1.9924	-0.3112
-0.0256	-3.3322	4.4427	3.4657	-2.2886
-0.0256	-3.3322	4.4427	3.4657	-2.1332
-0.0256	-3.3322	4.4427	3.4657	-2.2886
-0.0256	-3.3322	4.4427	3.4657	0.8521
-0.0256	-3.3322	4.4427	1.2039	-0.3924
-0.0256	-3.3322	4.4427	3.4657	-1.9862
-0.0256	-3.3322	4.4427	3.4657	-2.3756
-0.0256	-3.3322	4.4427	3.4657	0.8521
-0.0256	-3.3322	4.4427	1.9924	0.3892
-0.0256	-3.3322	4.4427	3.4657	0.1519
-0.0256	-3.3322	4.4427	3.4657	-2.506
-0.0256	-3.3322	4.4427	3.4657	-2.531
-0.0256	-3.3322	4.4427	3.4657	-2.2886

OPENNESS	Concientiousness	Extraversion	Agreebleness	Neuroticism
55.39%	36.39%	57.22%	22.25%	57.98%
46.74%	37.25%	53.45%	22.90%	59.08%
53.12%	39.73%	58.82%	23.55%	61.00%
53.03%	36.26%	55.67%	24.23%	60.67%
48.90%	37.49%	54.86%	22.13%	58.12%
53.94%	37.44%	57.53%	21.15%	56.80%
52.61%	38.32%	57.25%	24.22%	61.41%
53.74%	37.56%	57.38%	22.35%	58.53%
49.42%	37.08%	54.57%	23.89%	60.44%
53.12%	39.73%	58.82%	23.55%	61.00%
53.94%	37.44%	57.53%	21.15%	56.80%
50.23%	40.02%	57.57%	23.77%	61.37%
53.12%	39.73%	58.82%	23.55%	61.00%
46.74%	37.25%	53.45%	22.90%	59.08%
51.58%	37.32%	55.97%	23.12%	59.49%
47.05%	38.78%	54.88%	23.44%	60.41%
47.47%	36.73%	53.30%	23.46%	59.67%
53.12%	39.73%	58.82%	23.55%	61.00%
51.67%	40.79%	59.13%	22.44%	59.83%
53.12%	39.73%	58.82%	23.55%	61.00%

7 Business Applications

- **Content Marketing:** Produce more valuable, targeted content by focusing on your audience's unique interests and needs.
- **Display Ads:** Choosing to advertise on the sites you know the target customers visit, based on their behaviour.
- **Real-life example:** This is some real bank data, so when we have each customer's personality, we can focus on recommending him specific products. For instance, if the customer is mainly a investing guy, the bank will recommend more financing products to him.