Online Store Data Case Study

Hayden Lynch

2023-02-13

Data Cleaning Process

Step 1: Install/Import Libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## v purrr   1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(dplyr)
library(readr)
library(scales)
```

```
##
## Attaching package: ’scales’
##
## The following object is masked from ’package:purrr’:
##
##     discard
##
## The following object is masked from ’package:readr’:
##
##     col_factor
```

Step 2: Import Unclean Data

```
df = read_csv("online_store_customer_data_copy.csv")
```

```
## Rows: 2512 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): Transaction_date, Gender, Marital_status, State_names, Segment, Emp...
## dbl (4): Transaction_ID, Age, Referal, Amount_spent
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
df
```

```
## # A tibble: 2,512 x 11
##    Transa~1 Trans~2 Gender   Age Marit~3 State~4 Segment Emplo~5 Payme~6 Referal
##    <chr>      <dbl> <chr>  <dbl> <chr>   <chr>   <chr>   <chr>   <chr>     <dbl>
##  1 1/1/2019  151200 Female    19 Single  Kansas  Basic   Unempl~ Other         1
##  2 1/1/2019  151201 Male      49 Single  Illino~ Basic   self-e~ Card          0
##  3 1/1/2019  151202 Male      63 Married New Me~ Basic   workers PayPal        1
##  4 1/1/2019  151203 <NA>      18 Single  Virgin~ Platin~ workers Card          1
##  5 1/1/2019  151204 Male      27 Single  Connec~ Basic   self-e~ Card          0
##  6 1/3/2019  151205 Male      71 Single  Hawaii  Basic   Employ~ PayPal        1
##  7 1/3/2019  151206 Female    34 Married New Me~ Platin~ Employ~ PayPal        1
##  8 1/3/2019  151207 Male      37 Married Connec~ Basic   workers PayPal        1
##  9 1/4/2019  151208 Male      75 Married Florida Silver  Employ~ Card          0
## 10 1/4/2019  151209 Female    41 Married Vermont Gold    Unempl~ Card          1
## # ... with 2,502 more rows, 1 more variable: Amount_spent <dbl>, and
## #   abbreviated variable names 1: Transaction_date, 2: Transaction_ID,
## #   3: Marital_status, 4: State_names, 5: Employees_status, 6: Payment_method
```

Step 3: Remove "NA" rows

Table 1: A knitr kable

Transaction_date	Transaction_ID	Gender	Age	Marital_status	State_names	Segment	Employees_status	Payment_method	Referal	Amount_spent
1/1/2019	151200	Female	19	Single	Kansas	Basic	Unemployment	Other	1	2051.36
1/1/2019	151201	Male	49	Single	Illinois	Basic	self-employed	Card	0	544.04
1/1/2019	151202	Male	63	Married	New Mexico	Basic	workers	PayPal	1	1572.60
1/1/2019	151203	NA	18	Single	Virginia	Platinum	workers	Card	1	1199.79
1/1/2019	151204	Male	27	Single	Connecticut	Basic	self-employed	Card	0	NA
1/3/2019	151205	Male	71	Single	Hawaii	Basic	Employees	PayPal	1	2922.66
1/3/2019	151206	Female	34	Married	New Mexico	Platinum	Employees	PayPal	1	1481.42
1/3/2019	151207	Male	37	Married	Connecticut	Basic	workers	PayPal	1	1149.55
1/4/2019	151208	Male	75	Married	Florida	Silver	Employees	Card	0	1046.20
1/4/2019	151209	Female	41	Married	Vermont	Gold	Unemployment	Card	1	2730.60

- Using knitr kable's allow you to make nice tables in RMarkdown.

Step 4: Remove all 2021 Rows

```
new_df = mutate(new_df, Transaction_date = as.Date(Transaction_date, "%m/%d/%Y"))
class(new_df$Transaction_date)
```

The "Transaction_date" needs to be in 'date format'.

```
## [1] "Date"
```

```
new_df
```

```
## # A tibble: 2,044 x 11
##    Transaction_date Trans~1 Gender   Age Marit~2 State~3 Segment Emplo~4 Payme~5
##    <date>             <dbl> <chr>  <dbl> <chr>   <chr>   <chr>   <chr>   <chr>
##  1 2019-01-01        151200 Female    19 Single  Kansas  Basic   Unempl~ Other
##  2 2019-01-01        151201 Male      49 Single  Illino~ Basic   self-e~ Card
##  3 2019-01-01        151202 Male      63 Married New Me~ Basic   workers PayPal
##  4 2019-01-03        151205 Male      71 Single  Hawaii  Basic   Employ~ PayPal
##  5 2019-01-03        151206 Female    34 Married New Me~ Platin~ Employ~ PayPal
##  6 2019-01-03        151207 Male      37 Married Connec~ Basic   workers PayPal
##  7 2019-01-04        151208 Male      75 Married Florida Silver  Employ~ Card
##  8 2019-01-04        151209 Female    41 Married Vermont Gold    Unempl~ Card
##  9 2019-01-04        151210 Female    56 Married Califo~ Basic   Employ~ PayPal
## 10 2019-01-05        151211 Female    63 Married Colora~ Basic   workers Other
## # ... with 2,034 more rows, 2 more variables: Referal <dbl>,
## #   Amount_spent <dbl>, and abbreviated variable names 1: Transaction_ID,
## #   2: Marital_status, 3: State_names, 4: Employees_status, 5: Payment_method
```

```
new_df = new_df %>%
  filter(Transaction_date < "2021-01-01")
new_df
```

Now, rows including 2021 can be removed.

```
## # A tibble: 1,743 x 11
##    Transaction_date Trans~1 Gender   Age Marit~2 State~3 Segment Emplo~4 Payme~5
##    <date>             <dbl> <chr>  <dbl> <chr>   <chr>   <chr>   <chr>   <chr>
##  1 2019-01-01        151200 Female    19 Single  Kansas  Basic   Unempl~ Other
##  2 2019-01-01        151201 Male      49 Single  Illino~ Basic   self-e~ Card
##  3 2019-01-01        151202 Male      63 Married New Me~ Basic   workers PayPal
##  4 2019-01-03        151205 Male      71 Single  Hawaii  Basic   Employ~ PayPal
##  5 2019-01-03        151206 Female    34 Married New Me~ Platin~ Employ~ PayPal
##  6 2019-01-03        151207 Male      37 Married Connec~ Basic   workers PayPal
##  7 2019-01-04        151208 Male      75 Married Florida Silver  Employ~ Card
##  8 2019-01-04        151209 Female    41 Married Vermont Gold    Unempl~ Card
##  9 2019-01-04        151210 Female    56 Married Califo~ Basic   Employ~ PayPal
## 10 2019-01-05        151211 Female    63 Married Colora~ Basic   workers Other
## # ... with 1,733 more rows, 2 more variables: Referal <dbl>,
## #   Amount_spent <dbl>, and abbreviated variable names 1: Transaction_ID,
## #   2: Marital_status, 3: State_names, 4: Employees_status, 5: Payment_method
```

- The tibble now has 1743 rows of data that can be analyzed.

Step 5: Rename values in Referal column

```
new_df$Referal[new_df$Referal == 1] = 'Reffered'
new_df$Referal[new_df$Referal == 0] = 'Not Referred'
new_df
```

```
## # A tibble: 1,743 x 11
##    Transaction_date Trans~1 Gender   Age Marit~2 State~3 Segment Emplo~4 Payme~5
##    <date>             <dbl> <chr>  <dbl> <chr>   <chr>   <chr>   <chr>   <chr>
##  1 2019-01-01        151200 Female    19 Single  Kansas  Basic   Unempl~ Other
##  2 2019-01-01        151201 Male      49 Single  Illino~ Basic   self-e~ Card
##  3 2019-01-01        151202 Male      63 Married New Me~ Basic   workers PayPal
##  4 2019-01-03        151205 Male      71 Single  Hawaii  Basic   Employ~ PayPal
##  5 2019-01-03        151206 Female    34 Married New Me~ Platin~ Employ~ PayPal
##  6 2019-01-03        151207 Male      37 Married Connec~ Basic   workers PayPal
##  7 2019-01-04        151208 Male      75 Married Florida Silver  Employ~ Card
##  8 2019-01-04        151209 Female    41 Married Vermont Gold    Unempl~ Card
##  9 2019-01-04        151210 Female    56 Married Califo~ Basic   Employ~ PayPal
## 10 2019-01-05        151211 Female    63 Married Colora~ Basic   workers Other
## # ... with 1,733 more rows, 2 more variables: Referal <chr>,
## #   Amount_spent <dbl>, and abbreviated variable names 1: Transaction_ID,
## #   2: Marital_status, 3: State_names, 4: Employees_status, 5: Payment_method
```

Step 6: Data Cleaning Process: Save cleaned data file

```
save(new_df, file = "cleaned_data.csv")
write.csv(new_df, file = "cleaned_data.csv")
```

Data Analysis

Data Overview

```
df <- read_csv("cleaned_data.csv")
```

```
## New names:
## Rows: 1743 Columns: 12
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (7): Gender, Marital_status, State_names, Segment, Employees_status, Pa... dbl
## (4): ...1, Transaction_ID, Age, Amount_spent date (1): Transaction_date
## i Use ‘spec()‘ to retrieve the full column specification for this data. i
## Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## * ‘‘ -> ‘...1‘
```

```r
kable(df[1:10,], caption="Table Layout")
```

Table 2: Table Layout

...1	Transaction_date	Transaction_ID	Gender	Age	Marital_status	State_names	Segment	Employees_status	Payment_method	Referal	Amount_spent
1	2019-01-01	151200	Female	19	Single	Kansas	Basic	Unemployment	Other	Reffered	2051.36
2	2019-01-01	151201	Male	49	Single	Illinois	Basic	self-employed	Card	Not Referred	544.04
3	2019-01-01	151202	Male	63	Married	New Mexico	Basic	workers	PayPal	Reffered	1572.60
4	2019-01-03	151205	Male	71	Single	Hawaii	Basic	Employees	PayPal	Reffered	2922.66
5	2019-01-03	151206	Female	34	Married	New Mexico	Platinum	Employees	PayPal	Reffered	1481.42
6	2019-01-03	151207	Male	37	Married	Connecticut	Basic	workers	PayPal	Reffered	1149.55
7	2019-01-04	151208	Male	75	Married	Florida	Silver	Employees	Card	Not Referred	1046.20
8	2019-01-04	151209	Female	41	Married	Vermont	Gold	Unemployment	Card	Reffered	2730.60
9	2019-01-04	151210	Female	56	Married	California	Basic	Employees	PayPal	Not Referred	1712.82
10	2019-01-05	151211	Female	63	Married	Colorado	Basic	workers	Other	Reffered	154.31

```r
colnames(df)
```

```
##  [1] "...1"             "Transaction_date" "Transaction_ID"   "Gender"
##  [5] "Age"              "Marital_status"   "State_names"      "Segment"
##  [9] "Employees_status" "Payment_method"   "Referal"          "Amount_spent"
```

```r
str(df)
```

```
## spc_tbl_ [1,743 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ...1            : num [1:1743] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Transaction_date: Date[1:1743], format: "2019-01-01" "2019-01-01" ...
##  $ Transaction_ID  : num [1:1743] 151200 151201 151202 151205 151206 ...
##  $ Gender          : chr [1:1743] "Female" "Male" "Male" "Male" ...
##  $ Age             : num [1:1743] 19 49 63 71 34 37 75 41 56 63 ...
##  $ Marital_status  : chr [1:1743] "Single" "Single" "Married" "Single" ...
##  $ State_names     : chr [1:1743] "Kansas" "Illinois" "New Mexico" "Hawaii" ...
##  $ Segment         : chr [1:1743] "Basic" "Basic" "Basic" "Basic" ...
##  $ Employees_status: chr [1:1743] "Unemployment" "self-employed" "workers" "Employees" ...
##  $ Payment_method  : chr [1:1743] "Other" "Card" "PayPal" "PayPal" ...
##  $ Referal         : chr [1:1743] "Reffered" "Not Referred" "Reffered" "Reffered" ...
##  $ Amount_spent    : num [1:1743] 2051 544 1573 2923 1481 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ...1 = col_double(),
##   ..   Transaction_date = col_date(format = ""),
##   ..   Transaction_ID = col_double(),
##   ..   Gender = col_character(),
##   ..   Age = col_double(),
##   ..   Marital_status = col_character(),
##   ..   State_names = col_character(),
##   ..   Segment = col_character(),
##   ..   Employees_status = col_character(),
##   ..   Payment_method = col_character(),
##   ..   Referal = col_character(),
##   ..   Amount_spent = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
glimpse(df)
```

```
## Rows: 1,743
## Columns: 12
## $ ...1             <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16~
## $ Transaction_date <date> 2019-01-01, 2019-01-01, 2019-01-01, 2019-01-03, 2019~
## $ Transaction_ID   <dbl> 151200, 151201, 151202, 151205, 151206, 151207, 15120~
## $ Gender           <chr> "Female", "Male", "Male", "Male", "Female", "Male", "~
## $ Age              <dbl> 19, 49, 63, 71, 34, 37, 75, 41, 56, 63, 60, 47, 24, 1~
## $ Marital_status   <chr> "Single", "Single", "Married", "Single", "Married", "~
## $ State_names      <chr> "Kansas", "Illinois", "New Mexico", "Hawaii", "New Me~
## $ Segment          <chr> "Basic", "Basic", "Basic", "Basic", "Platinum", "Basi~
## $ Employees_status <chr> "Unemployment", "self-employed", "workers", "Employee~
## $ Payment_method   <chr> "Other", "Card", "PayPal", "PayPal", "PayPal", "PayPa~
## $ Referal          <chr> "Reffered", "Not Referred", "Reffered", "Reffered", "~
## $ Amount_spent     <dbl> 2051.36, 544.04, 1572.60, 2922.66, 1481.42, 1149.55, ~
```

Data Analysis Process

Question 1: Does the date influence spending amount (Year/Month)?

```
yearly_differences_amount_spent = df %>%
  mutate(Year = format(Transaction_date, "%Y")) %>%
  group_by(Year) %>%
  summarise(Total_Amount_Spent = sum(Amount_spent))
kable(yearly_differences_amount_spent[1:2,])
```

Year	Total_Amount_Spent
2019	1247650
2020	1237644

- Per Year

```
monthly_differences_amount_spent = df %>%
  mutate(Month = format(Transaction_date, "%m")) %>%
  group_by(Month) %>%
  summarise(Total_Amount_Spent = sum(Amount_spent))
kable(monthly_differences_amount_spent[1:12,])
```

Month	Total_Amount_Spent
01	262925.1
02	210744.9
03	199478.7
04	201662.5
05	211077.4
06	207129.8
07	199257.6
08	213644.6
09	192004.4
10	215111.4
11	160991.5
12	211266.0

- Per Month

```
date_differences_amount_spent = df %>%
  mutate(Month = format(Transaction_date, "%m"), Year = format(Transaction_date, "%Y")) %>%
  group_by(Month,Year) %>%
  summarise(Total_Amount_Spent = sum(Amount_spent))
```

```
## ‘summarise()‘ has grouped output by ’Month’. You can override using the
## ‘.groups‘ argument.
```

```
kable(date_differences_amount_spent[1:12,])
```

Month	Year	Total_Amount_Spent
01	2019	109263.95
01	2020	153661.16
02	2019	105064.02
02	2020	105680.87
03	2019	107089.68
03	2020	92388.98
04	2019	117837.08
04	2020	83825.46
05	2019	114877.42
05	2020	96200.01
06	2019	108504.29
06	2020	98625.54

- Per both year and month

Question 2: Do certain states spend more than others?

```r
# Total amount spent per state
state_differences_amount_spent = df %>%
  group_by(State_names) %>%
  summarise(Total_Amount_Spent = sum(Amount_spent),
            Avg_Amount_Spent = mean(Amount_spent))
# Top performing states to the top of dataset
state_differences_amount_spent = state_differences_amount_spent[rev(order(state_differences_amount_spent
# Rounding to the second decimal
state_differences_amount_spent$Total_Amount_Spent = round(state_differences_amount_spent$Total_Amount_Sp
state_differences_amount_spent$Avg_Amount_Spent = round(state_differences_amount_spent$Avg_Amount_Spent
# View new dataset
kable(state_differences_amount_spent[1:5,])
```

State_names	Total_Amount_Spent	Avg_Amount_Spent
Arizona	70769.26	1645.80
Massachusetts	67080.16	1490.67
Illinois	66729.45	1390.20
Missouri	63834.37	1679.85
New Jersey	61035.85	1606.21

Question 3: Does marital status dictate membership segments?

```r
# Total and average amount spent per marital status per membership
marital_status_dictate_membership = df %>%
  group_by(Marital_status,Segment) %>%
  summarise(Total_Amount_Spent = sum(Amount_spent),Avg_Amount_Spent = mean(Amount_spent))
```

```
## ‘summarise()‘ has grouped output by ’Marital_status’. You can override using
## the ‘.groups‘ argument.
```

```r
marital_status_dictate_membership = marital_status_dictate_membership[order(marital_status_dictate_membe
kable(marital_status_dictate_membership[1:10,])
```

Marital_status	Segment	Total_Amount_Spent	Avg_Amount_Spent
Married	Basic	673206.05	1423.269
Single	Basic	465573.31	1423.772
Married	Gold	142760.13	1456.736
Single	Gold	95452.28	1289.896
Married	Missing	105413.90	1369.012
Single	Missing	96590.12	1509.221
Married	Platinum	238962.31	1413.978
Single	Platinum	197428.95	1462.437
Married	Silver	262528.03	1396.426
Single	Silver	207378.96	1502.746

Question 4: What is the percentage breakdown between employee status?

```
employee_status_percent = df %>%
  group_by(Employees_status) %>%
  count(Employees_status)
# Turning total count to percent of people (using scales package)
employee_status_percent$Percent = percent(employee_status_percent$n/1743)
#employee_status_percent$Percent = percent(employee_status_percent$Percent)
# Changing the name of column
names(employee_status_percent)[names(employee_status_percent) == 'n'] = "Count"
kable(employee_status_percent[1:4,])
```

Employees_status	Count	Percent
Employees	676	38.8%
Unemployment	177	10.2%
self-employed	339	19.4%
workers	551	31.6%

Question 5: What age group spends more than others, how does the payment method influence age group spending?

```
# Making Age_Group Column and new dataframe for the question
age_group_df = df %>%
  mutate(
    Age_Group = dplyr::case_when(
      Age < 25 ~ "15-24",
      Age >= 25 & Age < 40 ~ "25-39",
      Age >= 40 & Age < 55 ~ "40-54",
      Age >= 55 ~ "55+"
      )
    )
age_group_differences_amount_spent = age_group_df %>%
  group_by(Age_Group) %>%
  summarise(Total_Amount_Spent = sum(Amount_spent), Avg_Amount_Spent = mean(Amount_spent))
# Get count of dataset
example = age_group_df %>%
  count(Age_Group)
names(example)[names(example) == 'n'] = "Total People"
# Combine both tables
age_group_differences_amount_spent$Total_People = example$`Total People`
kable(age_group_differences_amount_spent[1:4,])
```

Age_Group	Total_Amount_Spent	Avg_Amount_Spent	Total_People
15-24	372376.5	1489.506	250
25-39	548819.2	1385.907	396
40-54	662596.0	1462.684	453
55+	901502.3	1399.848	644

Question 6: Are referrals worth investing into?

```
referal_amount_spent = df %>%
  group_by(Referal) %>%
  summarise(Total_Amount_Spent = sum(Amount_spent), Avg_Amount_Spent = mean(Amount_spent))
kable(referal_amount_spent[1:2,])
```

Referal	Total_Amount_Spent	Avg_Amount_Spent
Not Referred	871763.8	1426.782
Reffered	1613530.2	1425.380

Question 7: Should other payment methods be targeted and influenced?

```
payment_method_targeting = df %>%
  group_by(Payment_method) %>%
  summarise(Total_Amount_Spent = sum(Amount_spent), Avg_Amount_Spent = mean(Amount_spent))
kable(payment_method_targeting[1:3,])
```

Payment_method	Total_Amount_Spent	Avg_Amount_Spent
Card	725668.1	1425.674
Other	619987.8	1476.161
PayPal	1139638.1	1400.047

Question 8: How much of a difference are the different segments making?

```
# Creating table with total/avg amount spent per segment
segment_influence = df %>%
  group_by(Segment) %>%
  summarise(Total_Amount_Spent = sum(Amount_spent), Avg_Amount_Spent = mean(Amount_spent))
# Adding total people column
segment_count = df %>%
  count(Segment)
segment_influence$Total_People = segment_count$n
# Adding percent column
segment_count$n = segment_count$n / 1743
segment_influence$Percent = segment_count$n
segment_influence$Percent = percent(segment_influence$Percent)
kable(segment_influence[1:5,])
```

Segment	Total_Amount_Spent	Avg_Amount_Spent	Total_People	Percent
Basic	1138779.4	1423.474	800	45.9%
Gold	238212.4	1384.956	172	9.9%
Missing	202004.0	1432.653	141	8.1%
Platinum	436391.3	1435.498	304	17.4%

Segment	Total_Amount_Spent	Avg_Amount_Spent	Total_People	Percent
Silver	469907.0	1441.432	326	18.7%

Question 9: In the varying states, which age group should be targeted, what percent do they make up in the states?

```
# Creating base table
state_targeting = age_group_df %>%
  group_by(State_names, Age_Group) %>%
  summarise(Total_Amount_Spent = sum(Amount_spent), Avg_Amount_Spent = mean(Amount_spent))
```

```
## 'summarise()' has grouped output by 'State_names'. You can override using the
## '.groups' argument.
```

```
# Adding total people using count
state_agegroup_count = age_group_df %>%
  group_by(State_names,Age_Group) %>%
  count(Age_Group)
state_targeting$Total_People = state_agegroup_count$n
state_targeting = state_targeting[,c(1,2,5,3,4)]
# Sort by largest amount spent per state per age group
state_targeting = state_targeting %>%
  arrange(desc(Total_Amount_Spent))
kable(state_targeting[1:10,])
```

State_names	Age_Group	Total_People	Total_Amount_Spent	Avg_Amount_Spent
Massachusetts	55+	21	33588.92	1599.472
Arizona	55+	19	30372.38	1598.546
Georgia	55+	23	26846.50	1167.239
Maine	55+	15	25393.08	1692.872
California	55+	19	25131.57	1322.714
South Dakota	55+	13	24624.65	1894.204
Missouri	40-54	15	24148.52	1609.901
Montana	55+	14	24071.42	1719.387
Delaware	55+	17	23116.05	1359.768
Minnesota	55+	13	22634.89	1741.145

Question 10: Should we influence a gender for a specific segment?

```
# Creating table for total/avg amount spent
gender_segment_influence = df %>%
  group_by(Gender,Segment) %>%
  summarise(Total_Amount_Spent = sum(Amount_spent), Avg_Amount_Spent = mean(Amount_spent))
```

```
## 'summarise()' has grouped output by 'Gender'. You can override using the
## '.groups' argument.
```

```
# Re-ordered the values based on segment
gender_segment_influence = gender_segment_influence[order(gender_segment_influence$Segment),]
# Created total people from different table using count
gender_segment_count = df %>%
  group_by(Gender,Segment) %>%
  count(Segment)
gender_segment_count = gender_segment_count[order(gender_segment_count$Segment),]
# After re-ordering the count, I merged both of them together
gender_segment_influence$Total_People = gender_segment_count$n
gender_segment_influence = gender_segment_influence[,c(1,2,5,3,4)]
kable(gender_segment_influence[1:10,])
```

Gender	Segment	Total_People	Total_Amount_Spent	Avg_Amount_Spent
Female	Basic	427	594996.78	1393.435
Male	Basic	373	543782.58	1457.862
Female	Gold	94	135382.43	1440.239
Male	Gold	78	102829.98	1318.333
Female	Missing	74	115132.55	1555.845
Male	Missing	67	86871.47	1296.589
Female	Platinum	169	230832.40	1365.872
Male	Platinum	135	205558.86	1522.658
Female	Silver	171	248562.60	1453.582
Male	Silver	155	221344.39	1428.028

Question 11: What age group is worth referring to the online environment?

```
# Creating table for age_groups referred total/avg spending
age_group_online_experience = age_group_df %>%
  group_by(Age_Group,Referal) %>%
  summarise(Total_Amount_Spent = sum(Amount_spent), Avg_Amount_Spent = mean(Amount_spent))
```

```
## ‘summarise()‘ has grouped output by ’Age_Group’. You can override using the
## ‘.groups‘ argument.
```

```
# Separate table for total people
age_group_online_count = age_group_df %>%
  group_by(Age_Group,Referal) %>%
  count(Referal)
# Add column from count to main table
age_group_online_experience$Total_People = age_group_online_count$n
# Re-ordered table
age_group_online_experience = age_group_online_experience[,c(1,2,5,3,4)]
kable(age_group_online_experience[1:8,])
```

Age_Group	Referal	Total_People	Total_Amount_Spent	Avg_Amount_Spent
15-24	Not Referred	87	130305.7	1497.767
15-24	Reffered	163	242070.8	1485.097
25-39	Not Referred	156	218461.9	1400.397

Age_Group	Referal	Total_People	Total_Amount_Spent	Avg_Amount_Spent
25-39	Reffered	240	330357.3	1376.489
40-54	Not Referred	150	219728.3	1464.855
40-54	Reffered	303	442867.8	1461.610
55+	Not Referred	218	303268.0	1391.137
55+	Reffered	426	598234.3	1404.306