

MA_HW03_data_preprocess2

November 27, 2020

0.0.1 Marketing Analytics HW03

0.0.2 Group: zj440 + xh2102 + elm454 + xr2008

First, we import all needed packages and the two csv files, replacing NA values as “None”, and extract some information from the two files.

```
[1]: import pandas as pd
import json
import numpy as np
```

```
[2]: df = pd.read_csv("attribution_allocation_student_data.csv")
df.head()
```

```
[2]:   convert_TF  touch1  touch2  touch3  touch4 touch5  tier
0         True   email  referral  referral      NaN   NaN    1
1         True  referral  referral    email      NaN   NaN    1
2         True  referral      NaN      NaN      NaN   NaN    1
3         True   email      NaN      NaN      NaN   NaN    1
4         True  referral  referral    social  referral   NaN    1
```

```
[3]: df=df.fillna('None')
df.head()
```

```
[3]:   convert_TF  touch1  touch2  touch3  touch4 touch5  tier
0         True   email  referral  referral    None   None    1
1         True  referral  referral    email    None   None    1
2         True  referral    None    None    None   None    1
3         True   email    None    None    None   None    1
4         True  referral  referral    social  referral   None    1
```

```
[4]: df_channel = pd.read_csv("channel_spend_student_data.csv")
df_channel.describe()
```

```
[4]:   Unnamed: 0      0
count      3      3
unique      3      3
top         tier1  {'email': 1000.0, 'social': 1000.0000000000001...
```

freq	1	1
------	---	---

```
[5]: df_channel_renamed=df_channel.rename(columns={"Unnamed: 0": "Tier", "0": "Details"})
df_channel_renamed
```

```
[5]:      Tier      Details
0  tier1  {'email': 1000.0, 'social': 1000.0000000000001...
1  tier2  {'email': 2000.0, 'social': 2000.0, 'display':...
2  tier3  {'email': 3000.0, 'social': 3000.0, 'display':...
```

We noticed that “Details” column contains useful information, so we use for loop to get things out.

```
[6]: allocation_method = list()
for i in range(len(df_channel_renamed)):
    r = json.loads(df_channel_renamed["Details"][i].replace("\'", "\""))
    r['tier'] = i+1
    allocation_method.append(r)

df_current_allocation = pd.DataFrame(allocation_method)
df_current_allocation
```

```
[6]:      email  social  display  paid_search  referral  organic_search  direct  \
0  1000.0  1000.0   1000.0      1000.0    1000.0           0.0     0.0
1  2000.0  2000.0   2000.0      2000.0    2000.0           0.0     0.0
2  3000.0  3000.0   3000.0      3000.0    3000.0           0.0     0.0

      tier
0       1
1       2
2       3
```

```
[7]: df_ttl = pd.merge(df, df_current_allocation,how='left', on='tier')
df_ttl
```

```
[7]:      convert_TF  touch1  touch2  touch3  touch4 touch5  tier  \
0             True   email  referral  referral    None    None    1
1             True  referral  referral   email    None    None    1
2             True  referral    None    None    None    None    1
3             True   email    None    None    None    None    1
4             True  referral  referral   social  referral    None    1
...
68119        False   direct    None    None    None    None    3
68120        False   direct   email  referral    None    None    3
68121        False   direct  referral    None    None    None    3
68122        False   email   direct    None    None    None    3
68123        False   direct  referral   social    None    None    3
```

	email	social	display	paid_search	referral	organic_search	direct
0	1000.0	1000.0	1000.0	1000.0	1000.0	0.0	0.0
1	1000.0	1000.0	1000.0	1000.0	1000.0	0.0	0.0
2	1000.0	1000.0	1000.0	1000.0	1000.0	0.0	0.0
3	1000.0	1000.0	1000.0	1000.0	1000.0	0.0	0.0
4	1000.0	1000.0	1000.0	1000.0	1000.0	0.0	0.0
...
68119	3000.0	3000.0	3000.0	3000.0	3000.0	0.0	0.0
68120	3000.0	3000.0	3000.0	3000.0	3000.0	0.0	0.0
68121	3000.0	3000.0	3000.0	3000.0	3000.0	0.0	0.0
68122	3000.0	3000.0	3000.0	3000.0	3000.0	0.0	0.0
68123	3000.0	3000.0	3000.0	3000.0	3000.0	0.0	0.0

[68124 rows x 14 columns]

Here, we have a dataframe containing integrated information.

Part 1 attribution: Allocate conversions by channel (social, organic_search, referral, email, paid_search, display, direct) and evaluate effectiveness

- Test 3 methods for allocation
- Calculate average CAC for each of the channels
- Discuss observations and potential conclusions from CAC calculations

Method 01: First interaction

```
[8]: new_df = df[df["convert_TF"]==True]
      new_df['touch1'].unique()
```

```
[8]: array(['email', 'referral', 'paid_search', 'direct', 'display', 'social',
           'organic_search'], dtype=object)
```

```
[9]: df_1 = new_df['touch1']
      df_1.value_counts()
```

```
[9]: referral      7480
      social       2257
      display     2059
      email       1265
      paid_search   697
      direct        11
      organic_search 10
      Name: touch1, dtype: int64
```

Method 02: Linear

```
[10]: df_ttl_T = df_ttl[df_ttl["convert_TF"]==True]
df_ttl_T
```

```
[10]:
```

	convert_TF	touch1	touch2	touch3	touch4	touch5	tier \
0	True	email	referral	referral	None	None	1
1	True	referral	referral	email	None	None	1
2	True	referral	None	None	None	None	1
3	True	email	None	None	None	None	1
4	True	referral	referral	social	referral	None	1
...
60648	True	referral	referral	None	None	None	3
60649	True	referral	referral	referral	referral	None	3
68103	True	organic_search	referral	referral	referral	None	3
68104	True	referral	referral	None	None	None	3
68117	True	direct	referral	None	None	None	3

	email	social	display	paid_search	referral	organic_search	direct
0	1000.0	1000.0	1000.0	1000.0	1000.0	0.0	0.0
1	1000.0	1000.0	1000.0	1000.0	1000.0	0.0	0.0
2	1000.0	1000.0	1000.0	1000.0	1000.0	0.0	0.0
3	1000.0	1000.0	1000.0	1000.0	1000.0	0.0	0.0
4	1000.0	1000.0	1000.0	1000.0	1000.0	0.0	0.0
...
60648	3000.0	3000.0	3000.0	3000.0	3000.0	0.0	0.0
60649	3000.0	3000.0	3000.0	3000.0	3000.0	0.0	0.0
68103	3000.0	3000.0	3000.0	3000.0	3000.0	0.0	0.0
68104	3000.0	3000.0	3000.0	3000.0	3000.0	0.0	0.0
68117	3000.0	3000.0	3000.0	3000.0	3000.0	0.0	0.0

[13779 rows x 14 columns]

```
[11]: import warnings

#To speed things up
with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    df_ttl_T["t1"] = np.where(df_ttl_T['touch1'] == "None", 0, 1)
    df_ttl_T["t2"] = np.where(df_ttl_T['touch2'] == "None", 0, 1)
    df_ttl_T["t3"] = np.where(df_ttl_T['touch3'] == "None", 0, 1)
    df_ttl_T["t4"] = np.where(df_ttl_T['touch4'] == "None", 0, 1)
    df_ttl_T["t5"] = np.where(df_ttl_T['touch5'] == "None", 0, 1)
    df_ttl_T["ttotal"] = df_ttl_T["t1"] + df_ttl_T["t2"] + df_ttl_T["t3"] +
    ↪df_ttl_T["t4"] + df_ttl_T["t5"]
    df_ttl_T["touch_email_1"] = np.where(df_ttl_T['touch1'] == "email", 1, 0)
    df_ttl_T["touch_email_2"] = np.where(df_ttl_T['touch2'] == "email", 1, 0)
    df_ttl_T["touch_email_3"] = np.where(df_ttl_T['touch3'] == "email", 1, 0)
    df_ttl_T["touch_email_4"] = np.where(df_ttl_T['touch4'] == "email", 1, 0)
```

```

df_ttl_T["touch_email_5"] = np.where(df_ttl_T['touch5'] == "email", 1, 0)
df_ttl_T["email_total"] = df_ttl_T["touch_email_1"] +
↳df_ttl_T["touch_email_2"] + df_ttl_T["touch_email_3"] +
↳df_ttl_T["touch_email_4"] + df_ttl_T["touch_email_5"]
df_ttl_T["touch_referral_1"] = np.where(df_ttl_T['touch1'] == "referral",
↳1, 0)
df_ttl_T["touch_referral_2"] = np.where(df_ttl_T['touch2'] == "referral",
↳1, 0)
df_ttl_T["touch_referral_3"] = np.where(df_ttl_T['touch3'] == "referral",
↳1, 0)
df_ttl_T["touch_referral_4"] = np.where(df_ttl_T['touch4'] == "referral",
↳1, 0)
df_ttl_T["touch_referral_5"] = np.where(df_ttl_T['touch5'] == "referral",
↳1, 0)
df_ttl_T["referral_total"] = df_ttl_T["touch_referral_1"] +
↳df_ttl_T["touch_referral_2"] + df_ttl_T["touch_referral_3"] +
↳df_ttl_T["touch_referral_4"] + df_ttl_T["touch_referral_5"]
df_ttl_T["touch_paid_search_1"] = np.where(df_ttl_T['touch1'] ==
↳"paid_search", 1, 0)
df_ttl_T["touch_paid_search_2"] = np.where(df_ttl_T['touch2'] ==
↳"paid_search", 1, 0)
df_ttl_T["touch_paid_search_3"] = np.where(df_ttl_T['touch3'] ==
↳"paid_search", 1, 0)
df_ttl_T["touch_paid_search_4"] = np.where(df_ttl_T['touch4'] ==
↳"paid_search", 1, 0)
df_ttl_T["touch_paid_search_5"] = np.where(df_ttl_T['touch5'] ==
↳"paid_search", 1, 0)
df_ttl_T["paid_search_total"] = df_ttl_T["touch_paid_search_1"] +
↳df_ttl_T["touch_paid_search_2"] + df_ttl_T["touch_paid_search_3"] +
↳df_ttl_T["touch_paid_search_4"] + df_ttl_T["touch_paid_search_5"]
df_ttl_T["touch_direct_1"] = np.where(df_ttl_T['touch1'] == "direct", 1, 0)
df_ttl_T["touch_direct_2"] = np.where(df_ttl_T['touch2'] == "direct", 1, 0)
df_ttl_T["touch_direct_3"] = np.where(df_ttl_T['touch3'] == "direct", 1, 0)
df_ttl_T["touch_direct_4"] = np.where(df_ttl_T['touch4'] == "direct", 1, 0)
df_ttl_T["touch_direct_5"] = np.where(df_ttl_T['touch5'] == "direct", 1, 0)
df_ttl_T["direct_total"] = df_ttl_T["touch_direct_1"] +
↳df_ttl_T["touch_direct_2"] + df_ttl_T["touch_direct_3"] +
↳df_ttl_T["touch_direct_4"] + df_ttl_T["touch_direct_5"]
df_ttl_T["touch_display_1"] = np.where(df_ttl_T['touch1'] == "display", 1,
↳0)
df_ttl_T["touch_display_2"] = np.where(df_ttl_T['touch2'] == "display", 1,
↳0)
df_ttl_T["touch_display_3"] = np.where(df_ttl_T['touch3'] == "display", 1,
↳0)
df_ttl_T["touch_display_4"] = np.where(df_ttl_T['touch4'] == "display", 1,
↳0)

```

```

df_ttl_T["touch_display_5"] = np.where(df_ttl_T['touch5'] == "display", 1, 0)
df_ttl_T["display_total"] = df_ttl_T["touch_display_1"] +
df_ttl_T["touch_display_2"] + df_ttl_T["touch_display_3"] +
df_ttl_T["touch_display_4"] + df_ttl_T["touch_display_5"]
df_ttl_T["touch_social_1"] = np.where(df_ttl_T['touch1'] == "social", 1, 0)
df_ttl_T["touch_social_2"] = np.where(df_ttl_T['touch2'] == "social", 1, 0)
df_ttl_T["touch_social_3"] = np.where(df_ttl_T['touch3'] == "social", 1, 0)
df_ttl_T["touch_social_4"] = np.where(df_ttl_T['touch4'] == "social", 1, 0)
df_ttl_T["touch_social_5"] = np.where(df_ttl_T['touch5'] == "social", 1, 0)
df_ttl_T["social_total"] = df_ttl_T["touch_social_1"] +
df_ttl_T["touch_social_2"] + df_ttl_T["touch_social_3"] +
df_ttl_T["touch_social_4"] + df_ttl_T["touch_social_5"]
df_ttl_T["touch_organic_search_1"] = np.where(df_ttl_T['touch1'] ==
"organic_search", 1, 0)
df_ttl_T["touch_organic_search_2"] = np.where(df_ttl_T['touch2'] ==
"organic_search", 1, 0)
df_ttl_T["touch_organic_search_3"] = np.where(df_ttl_T['touch3'] ==
"organic_search", 1, 0)
df_ttl_T["touch_organic_search_4"] = np.where(df_ttl_T['touch4'] ==
"organic_search", 1, 0)
df_ttl_T["touch_organic_search_5"] = np.where(df_ttl_T['touch5'] ==
"organic_search", 1, 0)
df_ttl_T["organic_search_total"] = df_ttl_T["touch_organic_search_1"] +
df_ttl_T["touch_organic_search_2"] + df_ttl_T["touch_organic_search_3"] +
df_ttl_T["touch_organic_search_4"] + df_ttl_T["touch_organic_search_5"]
df_ttl_T["email_total%"] = df_ttl_T["email_total"] / df_ttl_T["tttotal"]
df_ttl_T["referral_total%"] = df_ttl_T["referral_total"] /
df_ttl_T["tttotal"]
df_ttl_T["paid_search_total%"] = df_ttl_T["paid_search_total"] /
df_ttl_T["tttotal"]
df_ttl_T["direct_total%"] = df_ttl_T["direct_total"] / df_ttl_T["tttotal"]
df_ttl_T["display_total%"] = df_ttl_T["display_total"] / df_ttl_T["tttotal"]
df_ttl_T["social_total%"] = df_ttl_T["social_total"] / df_ttl_T["tttotal"]
df_ttl_T["organic_search_total%"] = df_ttl_T["organic_search_total"] /
df_ttl_T["tttotal"]

df_ttl_T

```

```

[11]:
convert_TF      touch1  touch2  touch3  touch4 touch5  tier  \
0          True    email  referral  referral    None    None    1
1          True  referral  referral    email    None    None    1
2          True  referral    None    None    None    None    1
3          True    email    None    None    None    None    1
4          True  referral  referral    social  referral    None    1
...          ...      ...      ...      ...      ...      ...

```

60648	True	referral	referral	None	None	None	3
60649	True	referral	referral	referral	referral	None	3
68103	True	organic_search	referral	referral	referral	None	3
68104	True	referral	referral	None	None	None	3
68117	True	direct	referral	None	None	None	3

	email	social	display	...	touch_organic_search_4	\
0	1000.0	1000.0	1000.0	...	0	
1	1000.0	1000.0	1000.0	...	0	
2	1000.0	1000.0	1000.0	...	0	
3	1000.0	1000.0	1000.0	...	0	
4	1000.0	1000.0	1000.0	...	0	
...	
60648	3000.0	3000.0	3000.0	...	0	
60649	3000.0	3000.0	3000.0	...	0	
68103	3000.0	3000.0	3000.0	...	0	
68104	3000.0	3000.0	3000.0	...	0	
68117	3000.0	3000.0	3000.0	...	0	

	touch_organic_search_5	organic_search_total	email_total%	\
0	0	0	0.333333	
1	0	0	0.333333	
2	0	0	0.000000	
3	0	0	1.000000	
4	0	0	0.000000	
...	
60648	0	0	0.000000	
60649	0	0	0.000000	
68103	0	1	0.000000	
68104	0	0	0.000000	
68117	0	0	0.000000	

	referral_total%	paid_search_total%	direct_total%	display_total%	\
0	0.666667	0.0	0.0	0.0	
1	0.666667	0.0	0.0	0.0	
2	1.000000	0.0	0.0	0.0	
3	0.000000	0.0	0.0	0.0	
4	0.750000	0.0	0.0	0.0	
...	
60648	1.000000	0.0	0.0	0.0	
60649	1.000000	0.0	0.0	0.0	
68103	0.750000	0.0	0.0	0.0	
68104	1.000000	0.0	0.0	0.0	
68117	0.500000	0.0	0.5	0.0	

	social_total%	organic_search_total%
0	0.00	0.00

1	0.00	0.00
2	0.00	0.00
3	0.00	0.00
4	0.25	0.00
...
60648	0.00	0.00
60649	0.00	0.00
68103	0.00	0.25
68104	0.00	0.00
68117	0.00	0.00

[13779 rows x 69 columns]

```
[12]: email_total_sum = df_ttl_T["email_total%"].sum()
referral_total_sum = df_ttl_T["referral_total%"].sum()
paid_search_total_sum = df_ttl_T["paid_search_total%"].sum()
direct_total_sum = df_ttl_T["direct_total%"].sum()
display_total_sum = df_ttl_T["display_total%"].sum()
social_total_sum = df_ttl_T["social_total%"].sum()
organic_search_total_sum = df_ttl_T["organic_search_total%"].sum()

total_sums = [{"Email total", email_total_sum}, {"Referral total",
↪ referral_total_sum}, {"Paid Search total", paid_search_total_sum}, {"Direct
↪ total", direct_total_sum}, {"Social total", social_total_sum}, {"Organic
↪ Search total", organic_search_total_sum}]

total_sums.sort(key=lambda x:x[1])
total_sums.reverse()
total_sums
```

```
[12]: [['Referral total', 7470.566666666667],
['Social total', 2291.483333333333],
['Email total', 1257.8500000000001],
['Paid Search total', 707.9833333333333],
['Organic Search total', 10.833333333333332],
['Direct total', 8.95]]
```

```
[13]: for i in total_sums:
        print(str(i[0]) + ": " + str(i[1]))
```

```
Referral total: 7470.566666666667
Social total: 2291.483333333333
Email total: 1257.8500000000001
Paid Search total: 707.9833333333333
Organic Search total: 10.833333333333332
Direct total: 8.95
```



```
[14]: linear_total_sums_df = pd.DataFrame.from_records(total_sums)
linear_total_sums_df
```

```
[14]:
```

		0	1
0	Referral total	7470.566667	
1	Social total	2291.483333	
2	Email total	1257.850000	
3	Paid Search total	707.983333	
4	Organic Search total	10.833333	
5	Direct total	8.950000	

Method 03: Position Based

```
[15]: df_ttl_T2 = df_ttl_T.copy()
df_ttl_T2
```

```
[15]:
```

	convert_TF	touch1	touch2	touch3	touch4	touch5	tier	\
0	True	email	referral	referral	None	None	1	
1	True	referral	referral	email	None	None	1	
2	True	referral	None	None	None	None	1	
3	True	email	None	None	None	None	1	
4	True	referral	referral	social	referral	None	1	
...		
60648	True	referral	referral	None	None	None	3	
60649	True	referral	referral	referral	referral	None	3	
68103	True	organic_search	referral	referral	referral	None	3	
68104	True	referral	referral	None	None	None	3	
68117	True	direct	referral	None	None	None	3	

	email	social	display	...	touch_organic_search_4	\
0	1000.0	1000.0	1000.0	...	0	
1	1000.0	1000.0	1000.0	...	0	
2	1000.0	1000.0	1000.0	...	0	
3	1000.0	1000.0	1000.0	...	0	
4	1000.0	1000.0	1000.0	...	0	
...	
60648	3000.0	3000.0	3000.0	...	0	
60649	3000.0	3000.0	3000.0	...	0	
68103	3000.0	3000.0	3000.0	...	0	
68104	3000.0	3000.0	3000.0	...	0	
68117	3000.0	3000.0	3000.0	...	0	

	touch_organic_search_5	organic_search_total	email_total%	\
0	0	0	0.333333	
1	0	0	0.333333	
2	0	0	0.000000	
3	0	0	1.000000	

4	0	0	0.000000
...
60648	0	0	0.000000
60649	0	0	0.000000
68103	0	1	0.000000
68104	0	0	0.000000
68117	0	0	0.000000

	referral_total%	paid_search_total%	direct_total%	display_total%	\
0	0.666667	0.0	0.0	0.0	
1	0.666667	0.0	0.0	0.0	
2	1.000000	0.0	0.0	0.0	
3	0.000000	0.0	0.0	0.0	
4	0.750000	0.0	0.0	0.0	
...	
60648	1.000000	0.0	0.0	0.0	
60649	1.000000	0.0	0.0	0.0	
68103	0.750000	0.0	0.0	0.0	
68104	1.000000	0.0	0.0	0.0	
68117	0.500000	0.0	0.5	0.0	

	social_total%	organic_search_total%
0	0.00	0.00
1	0.00	0.00
2	0.00	0.00
3	0.00	0.00
4	0.25	0.00
...
60648	0.00	0.00
60649	0.00	0.00
68103	0.00	0.25
68104	0.00	0.00
68117	0.00	0.00

[13779 rows x 69 columns]

```
[16]: #To speed things up. And I get how this is overkill, but I really really wanted
      ↪to vectorize it and so yeah
with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    df_ttl_T2["count1"] = np.where(df_ttl_T2['tttotal'] == 1, 1, 0)
    df_ttl_T2["count2"] = np.where(df_ttl_T2['tttotal'] == 2, 1, 0)
    df_ttl_T2["count3"] = np.where(df_ttl_T2['tttotal'] == 3, 1, 0)
    df_ttl_T2["count4"] = np.where(df_ttl_T2['tttotal'] == 4, 1, 0)
    df_ttl_T2["count5"] = np.where(df_ttl_T2['tttotal'] == 5, 1, 0)
    #If there is one touch
```

```

df_ttl_T2["touch_email_1"] = np.where(df_ttl_T2['touch1'] == "email", np.
↳where(df_ttl_T2["count1"] == 1, 1, df_ttl_T2["touch_email_1"]),␣
↳df_ttl_T2["touch_email_1"])
df_ttl_T2["touch_referral_1"] = np.where(df_ttl_T2['touch1'] == "referral",␣
↳np.where(df_ttl_T2["count1"] == 1, 1, df_ttl_T2["touch_referral_1"]),␣
↳df_ttl_T2["touch_referral_1"])
df_ttl_T2["touch_paid_search_1"] = np.where(df_ttl_T2['touch1'] ==␣
↳"paid_search", np.where(df_ttl_T2["count1"] == 1, 1,␣
↳df_ttl_T2["touch_paid_search_1"]), df_ttl_T2["touch_paid_search_1"])
df_ttl_T2["touch_direct_1"] = np.where(df_ttl_T2['touch1'] == "direct", np.
↳where(df_ttl_T2["count1"] == 1, 1, df_ttl_T2["touch_direct_1"]),␣
↳df_ttl_T2["touch_direct_1"])
df_ttl_T2["touch_display_1"] = np.where(df_ttl_T2['touch1'] == "display",␣
↳np.where(df_ttl_T2["count1"] == 1, 1, df_ttl_T2["touch_display_1"]),␣
↳df_ttl_T2["touch_display_1"])
df_ttl_T2["touch_social_1"] = np.where(df_ttl_T2['touch1'] == "social", np.
↳where(df_ttl_T2["count1"] == 1, 1, df_ttl_T2["touch_social_1"]),␣
↳df_ttl_T2["touch_social_1"])
df_ttl_T2["touch_organic_search_1"] = np.where(df_ttl_T2['touch1'] ==␣
↳"organic_search", np.where(df_ttl_T2["count1"] == 1, 1,␣
↳df_ttl_T2["touch_organic_search_1"]), df_ttl_T2["touch_organic_search_1"])
#If there are two touches
df_ttl_T2["touch_email_1"] = np.where(df_ttl_T2['touch1'] == "email", np.
↳where(df_ttl_T2["count2"] == 1, 0.5, df_ttl_T2["touch_email_1"]),␣
↳df_ttl_T2["touch_email_1"])
df_ttl_T2["touch_referral_1"] = np.where(df_ttl_T2['touch1'] == "referral",␣
↳np.where(df_ttl_T2["count2"] == 1, 0.5, df_ttl_T2["touch_referral_1"]),␣
↳df_ttl_T2["touch_referral_1"])
df_ttl_T2["touch_paid_search_1"] = np.where(df_ttl_T2['touch1'] ==␣
↳"paid_search", np.where(df_ttl_T2["count2"] == 1, 0.5,␣
↳df_ttl_T2["touch_paid_search_1"]), df_ttl_T2["touch_paid_search_1"])
df_ttl_T2["touch_direct_1"] = np.where(df_ttl_T2['touch1'] == "direct", np.
↳where(df_ttl_T2["count2"] == 1, 0.5, df_ttl_T2["touch_direct_1"]),␣
↳df_ttl_T2["touch_direct_1"])
df_ttl_T2["touch_display_1"] = np.where(df_ttl_T2['touch1'] == "display",␣
↳np.where(df_ttl_T2["count2"] == 1, 0.5, df_ttl_T2["touch_display_1"]),␣
↳df_ttl_T2["touch_display_1"])
df_ttl_T2["touch_social_1"] = np.where(df_ttl_T2['touch1'] == "social", np.
↳where(df_ttl_T2["count2"] == 1, 0.5, df_ttl_T2["touch_social_1"]),␣
↳df_ttl_T2["touch_social_1"])
df_ttl_T2["touch_organic_search_1"] = np.where(df_ttl_T2['touch1'] ==␣
↳"organic_search", np.where(df_ttl_T2["count2"] == 1, 0.5,␣
↳df_ttl_T2["touch_organic_search_1"]), df_ttl_T2["touch_organic_search_1"])
df_ttl_T2["touch_email_2"] = np.where(df_ttl_T2['touch2'] == "email", np.
↳where(df_ttl_T2["count2"] == 1, 0.5, df_ttl_T2["touch_email_2"]),␣
↳df_ttl_T2["touch_email_2"])

```

```

df_ttl_T2["touch_referral_2"] = np.where(df_ttl_T2['touch2'] == "referral",
↳np.where(df_ttl_T2["count2"] == 1, 0.5, df_ttl_T2["touch_referral_2"]),
↳df_ttl_T2["touch_referral_2"])
df_ttl_T2["touch_paid_search_2"] = np.where(df_ttl_T2['touch2'] ==
↳"paid_search", np.where(df_ttl_T2["count2"] == 1, 0.5,
↳df_ttl_T2["touch_paid_search_2"]), df_ttl_T2["touch_paid_search_2"])
df_ttl_T2["touch_direct_2"] = np.where(df_ttl_T2['touch2'] == "direct", np.
↳where(df_ttl_T2["count2"] == 1, 0.5, df_ttl_T2["touch_direct_2"]),
↳df_ttl_T2["touch_direct_2"])
df_ttl_T2["touch_display_2"] = np.where(df_ttl_T2['touch2'] == "display",
↳np.where(df_ttl_T2["count2"] == 1, 0.5, df_ttl_T2["touch_display_2"]),
↳df_ttl_T2["touch_display_2"])
df_ttl_T2["touch_social_2"] = np.where(df_ttl_T2['touch2'] == "social", np.
↳where(df_ttl_T2["count2"] == 1, 0.5, df_ttl_T2["touch_social_2"]),
↳df_ttl_T2["touch_social_2"])
df_ttl_T2["touch_organic_search_2"] = np.where(df_ttl_T2['touch2'] ==
↳"organic_search", np.where(df_ttl_T2["count2"] == 1, 0.5,
↳df_ttl_T2["touch_organic_search_2"]), df_ttl_T2["touch_organic_search_2"])
#If there are three touches
df_ttl_T2["touch_email_1"] = np.where(df_ttl_T2['touch1'] == "email", np.
↳where(df_ttl_T2["count3"] == 1, 0.4, df_ttl_T2["touch_email_1"]),
↳df_ttl_T2["touch_email_1"])
df_ttl_T2["touch_referral_1"] = np.where(df_ttl_T2['touch1'] == "referral",
↳np.where(df_ttl_T2["count3"] == 1, 0.4, df_ttl_T2["touch_referral_1"]),
↳df_ttl_T2["touch_referral_1"])
df_ttl_T2["touch_paid_search_1"] = np.where(df_ttl_T2['touch1'] ==
↳"paid_search", np.where(df_ttl_T2["count3"] == 1, 0.4,
↳df_ttl_T2["touch_paid_search_1"]), df_ttl_T2["touch_paid_search_1"])
df_ttl_T2["touch_direct_1"] = np.where(df_ttl_T2['touch1'] == "direct", np.
↳where(df_ttl_T2["count3"] == 1, 0.4, df_ttl_T2["touch_direct_1"]),
↳df_ttl_T2["touch_direct_1"])
df_ttl_T2["touch_display_1"] = np.where(df_ttl_T2['touch1'] == "display",
↳np.where(df_ttl_T2["count3"] == 1, 0.4, df_ttl_T2["touch_display_1"]),
↳df_ttl_T2["touch_display_1"])
df_ttl_T2["touch_social_1"] = np.where(df_ttl_T2['touch1'] == "social", np.
↳where(df_ttl_T2["count3"] == 1, 0.4, df_ttl_T2["touch_social_1"]),
↳df_ttl_T2["touch_social_1"])
df_ttl_T2["touch_organic_search_1"] = np.where(df_ttl_T2['touch1'] ==
↳"organic_search", np.where(df_ttl_T2["count3"] == 1, 0.4,
↳df_ttl_T2["touch_organic_search_1"]), df_ttl_T2["touch_organic_search_1"])
df_ttl_T2["touch_email_2"] = np.where(df_ttl_T2['touch2'] == "email", np.
↳where(df_ttl_T2["count3"] == 1, 0.2, df_ttl_T2["touch_email_2"]),
↳df_ttl_T2["touch_email_2"])
df_ttl_T2["touch_referral_2"] = np.where(df_ttl_T2['touch2'] == "referral",
↳np.where(df_ttl_T2["count3"] == 1, 0.2, df_ttl_T2["touch_referral_2"]),
↳df_ttl_T2["touch_referral_2"])

```

```

df_ttl_T2["touch_paid_search_2"] = np.where(df_ttl_T2['touch2'] ==
↳"paid_search", np.where(df_ttl_T2["count3"] == 1, 0.2,
↳df_ttl_T2["touch_paid_search_2"]), df_ttl_T2["touch_paid_search_2"])
df_ttl_T2["touch_direct_2"] = np.where(df_ttl_T2['touch2'] == "direct", np.
↳where(df_ttl_T2["count3"] == 1, 0.2, df_ttl_T2["touch_direct_2"]),
↳df_ttl_T2["touch_direct_2"])
df_ttl_T2["touch_display_2"] = np.where(df_ttl_T2['touch2'] == "display",
↳np.where(df_ttl_T2["count3"] == 1, 0.2, df_ttl_T2["touch_display_2"]),
↳df_ttl_T2["touch_display_2"])
df_ttl_T2["touch_social_2"] = np.where(df_ttl_T2['touch2'] == "social", np.
↳where(df_ttl_T2["count3"] == 1, 0.2, df_ttl_T2["touch_social_2"]),
↳df_ttl_T2["touch_social_2"])
df_ttl_T2["touch_organic_search_2"] = np.where(df_ttl_T2['touch2'] ==
↳"organic_search", np.where(df_ttl_T2["count3"] == 1, 0.2,
↳df_ttl_T2["touch_organic_search_2"]), df_ttl_T2["touch_organic_search_2"])
df_ttl_T2["touch_email_3"] = np.where(df_ttl_T2['touch3'] == "email", np.
↳where(df_ttl_T2["count3"] == 1, 0.4, df_ttl_T2["touch_email_3"]),
↳df_ttl_T2["touch_email_3"])
df_ttl_T2["touch_referral_3"] = np.where(df_ttl_T2['touch3'] == "referral",
↳np.where(df_ttl_T2["count3"] == 1, 0.4, df_ttl_T2["touch_referral_3"]),
↳df_ttl_T2["touch_referral_3"])
df_ttl_T2["touch_paid_search_3"] = np.where(df_ttl_T2['touch3'] ==
↳"paid_search", np.where(df_ttl_T2["count3"] == 1, 0.4,
↳df_ttl_T2["touch_paid_search_3"]), df_ttl_T2["touch_paid_search_3"])
df_ttl_T2["touch_direct_3"] = np.where(df_ttl_T2['touch3'] == "direct", np.
↳where(df_ttl_T2["count3"] == 1, 0.4, df_ttl_T2["touch_direct_3"]),
↳df_ttl_T2["touch_direct_3"])
df_ttl_T2["touch_display_3"] = np.where(df_ttl_T2['touch3'] == "display",
↳np.where(df_ttl_T2["count3"] == 1, 0.4, df_ttl_T2["touch_display_3"]),
↳df_ttl_T2["touch_display_3"])
df_ttl_T2["touch_social_3"] = np.where(df_ttl_T2['touch3'] == "social", np.
↳where(df_ttl_T2["count3"] == 1, 0.4, df_ttl_T2["touch_social_3"]),
↳df_ttl_T2["touch_social_3"])
df_ttl_T2["touch_organic_search_3"] = np.where(df_ttl_T2['touch3'] ==
↳"organic_search", np.where(df_ttl_T2["count3"] == 1, 0.4,
↳df_ttl_T2["touch_organic_search_3"]), df_ttl_T2["touch_organic_search_3"])
#If there are four touches
df_ttl_T2["touch_email_1"] = np.where(df_ttl_T2['touch1'] == "email", np.
↳where(df_ttl_T2["count4"] == 1, 0.4, df_ttl_T2["touch_email_1"]),
↳df_ttl_T2["touch_email_1"])
df_ttl_T2["touch_referral_1"] = np.where(df_ttl_T2['touch1'] == "referral",
↳np.where(df_ttl_T2["count4"] == 1, 0.4, df_ttl_T2["touch_referral_1"]),
↳df_ttl_T2["touch_referral_1"])
df_ttl_T2["touch_paid_search_1"] = np.where(df_ttl_T2['touch1'] ==
↳"paid_search", np.where(df_ttl_T2["count4"] == 1, 0.4,
↳df_ttl_T2["touch_paid_search_1"]), df_ttl_T2["touch_paid_search_1"])

```

```

df_ttl_T2["touch_direct_1"] = np.where(df_ttl_T2['touch1'] == "direct", np.
↳where(df_ttl_T2["count4"] == 1, 0.4, df_ttl_T2["touch_direct_1"]),␣
↳df_ttl_T2["touch_direct_1"])
df_ttl_T2["touch_display_1"] = np.where(df_ttl_T2['touch1'] == "display",␣
↳np.where(df_ttl_T2["count4"] == 1, 0.4, df_ttl_T2["touch_display_1"]),␣
↳df_ttl_T2["touch_display_1"])
df_ttl_T2["touch_social_1"] = np.where(df_ttl_T2['touch1'] == "social", np.
↳where(df_ttl_T2["count4"] == 1, 0.4, df_ttl_T2["touch_social_1"]),␣
↳df_ttl_T2["touch_social_1"])
df_ttl_T2["touch_organic_search_1"] = np.where(df_ttl_T2['touch1'] ==␣
↳"organic_search", np.where(df_ttl_T2["count4"] == 1, 0.4,␣
↳df_ttl_T2["touch_organic_search_1"]), df_ttl_T2["touch_organic_search_1"])
df_ttl_T2["touch_email_2"] = np.where(df_ttl_T2['touch2'] == "email", np.
↳where(df_ttl_T2["count4"] == 1, 0.1, df_ttl_T2["touch_email_2"]),␣
↳df_ttl_T2["touch_email_2"])
df_ttl_T2["touch_referral_2"] = np.where(df_ttl_T2['touch2'] == "referral",␣
↳np.where(df_ttl_T2["count4"] == 1, 0.1, df_ttl_T2["touch_referral_2"]),␣
↳df_ttl_T2["touch_referral_2"])
df_ttl_T2["touch_paid_search_2"] = np.where(df_ttl_T2['touch2'] ==␣
↳"paid_search", np.where(df_ttl_T2["count4"] == 1, 0.1,␣
↳df_ttl_T2["touch_paid_search_2"]), df_ttl_T2["touch_paid_search_2"])
df_ttl_T2["touch_direct_2"] = np.where(df_ttl_T2['touch2'] == "direct", np.
↳where(df_ttl_T2["count4"] == 1, 0.1, df_ttl_T2["touch_direct_2"]),␣
↳df_ttl_T2["touch_direct_2"])
df_ttl_T2["touch_display_2"] = np.where(df_ttl_T2['touch2'] == "display",␣
↳np.where(df_ttl_T2["count4"] == 1, 0.1, df_ttl_T2["touch_display_2"]),␣
↳df_ttl_T2["touch_display_2"])
df_ttl_T2["touch_social_2"] = np.where(df_ttl_T2['touch2'] == "social", np.
↳where(df_ttl_T2["count4"] == 1, 0.1, df_ttl_T2["touch_social_2"]),␣
↳df_ttl_T2["touch_social_2"])
df_ttl_T2["touch_organic_search_2"] = np.where(df_ttl_T2['touch2'] ==␣
↳"organic_search", np.where(df_ttl_T2["count4"] == 1, 0.1,␣
↳df_ttl_T2["touch_organic_search_2"]), df_ttl_T2["touch_organic_search_2"])
df_ttl_T2["touch_email_3"] = np.where(df_ttl_T2['touch3'] == "email", np.
↳where(df_ttl_T2["count4"] == 1, 0.1, df_ttl_T2["touch_email_3"]),␣
↳df_ttl_T2["touch_email_3"])
df_ttl_T2["touch_referral_3"] = np.where(df_ttl_T2['touch3'] == "referral",␣
↳np.where(df_ttl_T2["count4"] == 1, 0.1, df_ttl_T2["touch_referral_3"]),␣
↳df_ttl_T2["touch_referral_3"])
df_ttl_T2["touch_paid_search_3"] = np.where(df_ttl_T2['touch3'] ==␣
↳"paid_search", np.where(df_ttl_T2["count4"] == 1, 0.1,␣
↳df_ttl_T2["touch_paid_search_3"]), df_ttl_T2["touch_paid_search_3"])
df_ttl_T2["touch_direct_3"] = np.where(df_ttl_T2['touch3'] == "direct", np.
↳where(df_ttl_T2["count4"] == 1, 0.1, df_ttl_T2["touch_direct_3"]),␣
↳df_ttl_T2["touch_direct_3"])

```

```

df_ttl_T2["touch_display_3"] = np.where(df_ttl_T2['touch3'] == "display",
↳np.where(df_ttl_T2["count4"] == 1, 0.1, df_ttl_T2["touch_display_3"]),
↳df_ttl_T2["touch_display_3"])
df_ttl_T2["touch_social_3"] = np.where(df_ttl_T2['touch3'] == "social", np.
↳where(df_ttl_T2["count4"] == 1, 0.1, df_ttl_T2["touch_social_3"]),
↳df_ttl_T2["touch_social_3"])
df_ttl_T2["touch_organic_search_3"] = np.where(df_ttl_T2['touch3'] ==
↳"organic_search", np.where(df_ttl_T2["count4"] == 1, 0.1,
↳df_ttl_T2["touch_organic_search_3"]), df_ttl_T2["touch_organic_search_3"])
df_ttl_T2["touch_email_4"] = np.where(df_ttl_T2['touch4'] == "email", np.
↳where(df_ttl_T2["count4"] == 1, 0.4, df_ttl_T2["touch_email_4"]),
↳df_ttl_T2["touch_email_4"])
df_ttl_T2["touch_referral_4"] = np.where(df_ttl_T2['touch4'] == "referral",
↳np.where(df_ttl_T2["count4"] == 1, 0.4, df_ttl_T2["touch_referral_4"]),
↳df_ttl_T2["touch_referral_4"])
df_ttl_T2["touch_paid_search_4"] = np.where(df_ttl_T2['touch4'] ==
↳"paid_search", np.where(df_ttl_T2["count4"] == 1, 0.4,
↳df_ttl_T2["touch_paid_search_4"]), df_ttl_T2["touch_paid_search_4"])
df_ttl_T2["touch_direct_4"] = np.where(df_ttl_T2['touch4'] == "direct", np.
↳where(df_ttl_T2["count4"] == 1, 0.4, df_ttl_T2["touch_direct_4"]),
↳df_ttl_T2["touch_direct_4"])
df_ttl_T2["touch_display_4"] = np.where(df_ttl_T2['touch4'] == "display",
↳np.where(df_ttl_T2["count4"] == 1, 0.4, df_ttl_T2["touch_display_4"]),
↳df_ttl_T2["touch_display_4"])
df_ttl_T2["touch_social_4"] = np.where(df_ttl_T2['touch4'] == "social", np.
↳where(df_ttl_T2["count4"] == 1, 0.4, df_ttl_T2["touch_social_4"]),
↳df_ttl_T2["touch_social_4"])
df_ttl_T2["touch_organic_search_4"] = np.where(df_ttl_T2['touch4'] ==
↳"organic_search", np.where(df_ttl_T2["count4"] == 1, 0.4,
↳df_ttl_T2["touch_organic_search_4"]), df_ttl_T2["touch_organic_search_4"])
#If there are five touches
df_ttl_T2["touch_email_1"] = np.where(df_ttl_T2['touch1'] == "email", np.
↳where(df_ttl_T2["count5"] == 1, 0.4, df_ttl_T2["touch_email_1"]),
↳df_ttl_T2["touch_email_1"])
df_ttl_T2["touch_referral_1"] = np.where(df_ttl_T2['touch1'] == "referral",
↳np.where(df_ttl_T2["count5"] == 1, 0.4, df_ttl_T2["touch_referral_1"]),
↳df_ttl_T2["touch_referral_1"])
df_ttl_T2["touch_paid_search_1"] = np.where(df_ttl_T2['touch1'] ==
↳"paid_search", np.where(df_ttl_T2["count5"] == 1, 0.4,
↳df_ttl_T2["touch_paid_search_1"]), df_ttl_T2["touch_paid_search_1"])
df_ttl_T2["touch_direct_1"] = np.where(df_ttl_T2['touch1'] == "direct", np.
↳where(df_ttl_T2["count5"] == 1, 0.4, df_ttl_T2["touch_direct_1"]),
↳df_ttl_T2["touch_direct_1"])
df_ttl_T2["touch_display_1"] = np.where(df_ttl_T2['touch1'] == "display",
↳np.where(df_ttl_T2["count5"] == 1, 0.4, df_ttl_T2["touch_display_1"]),
↳df_ttl_T2["touch_display_1"])

```



```

df_ttl_T2["touch_social_1"] = np.where(df_ttl_T2['touch1'] == "social", np.
↳where(df_ttl_T2["count5"] == 1, 0.4, df_ttl_T2["touch_social_1"]),
↳df_ttl_T2["touch_social_1"])
df_ttl_T2["touch_organic_search_1"] = np.where(df_ttl_T2['touch1'] ==
↳"organic_search", np.where(df_ttl_T2["count5"] == 1, 0.4,
↳df_ttl_T2["touch_organic_search_1"]), df_ttl_T2["touch_organic_search_1"])
df_ttl_T2["touch_email_2"] = np.where(df_ttl_T2['touch2'] == "email", np.
↳where(df_ttl_T2["count5"] == 1, 0.2/3, df_ttl_T2["touch_email_2"]),
↳df_ttl_T2["touch_email_2"])
df_ttl_T2["touch_referral_2"] = np.where(df_ttl_T2['touch2'] == "referral",
↳np.where(df_ttl_T2["count5"] == 1, 0.2/3, df_ttl_T2["touch_referral_2"]),
↳df_ttl_T2["touch_referral_2"])
df_ttl_T2["touch_paid_search_2"] = np.where(df_ttl_T2['touch2'] ==
↳"paid_search", np.where(df_ttl_T2["count5"] == 1, 0.2/3,
↳df_ttl_T2["touch_paid_search_2"]), df_ttl_T2["touch_paid_search_2"])
df_ttl_T2["touch_direct_2"] = np.where(df_ttl_T2['touch2'] == "direct", np.
↳where(df_ttl_T2["count5"] == 1, 0.2/3, df_ttl_T2["touch_direct_2"]),
↳df_ttl_T2["touch_direct_2"])
df_ttl_T2["touch_display_2"] = np.where(df_ttl_T2['touch2'] == "display",
↳np.where(df_ttl_T2["count5"] == 1, 0.2/3, df_ttl_T2["touch_display_2"]),
↳df_ttl_T2["touch_display_2"])
df_ttl_T2["touch_social_2"] = np.where(df_ttl_T2['touch2'] == "social", np.
↳where(df_ttl_T2["count5"] == 1, 0.2/3, df_ttl_T2["touch_social_2"]),
↳df_ttl_T2["touch_social_2"])
df_ttl_T2["touch_organic_search_2"] = np.where(df_ttl_T2['touch2'] ==
↳"organic_search", np.where(df_ttl_T2["count5"] == 1, 0.2/3,
↳df_ttl_T2["touch_organic_search_2"]), df_ttl_T2["touch_organic_search_2"])
df_ttl_T2["touch_email_3"] = np.where(df_ttl_T2['touch3'] == "email", np.
↳where(df_ttl_T2["count5"] == 1, 0.2/3, df_ttl_T2["touch_email_3"]),
↳df_ttl_T2["touch_email_3"])
df_ttl_T2["touch_referral_3"] = np.where(df_ttl_T2['touch3'] == "referral",
↳np.where(df_ttl_T2["count5"] == 1, 0.2/3, df_ttl_T2["touch_referral_3"]),
↳df_ttl_T2["touch_referral_3"])
df_ttl_T2["touch_paid_search_3"] = np.where(df_ttl_T2['touch3'] ==
↳"paid_search", np.where(df_ttl_T2["count5"] == 1, 0.2/3,
↳df_ttl_T2["touch_paid_search_3"]), df_ttl_T2["touch_paid_search_3"])
df_ttl_T2["touch_direct_3"] = np.where(df_ttl_T2['touch3'] == "direct", np.
↳where(df_ttl_T2["count5"] == 1, 0.2/3, df_ttl_T2["touch_direct_3"]),
↳df_ttl_T2["touch_direct_3"])
df_ttl_T2["touch_display_3"] = np.where(df_ttl_T2['touch3'] == "display",
↳np.where(df_ttl_T2["count5"] == 1, 0.2/3, df_ttl_T2["touch_display_3"]),
↳df_ttl_T2["touch_display_3"])
df_ttl_T2["touch_social_3"] = np.where(df_ttl_T2['touch3'] == "social", np.
↳where(df_ttl_T2["count5"] == 1, 0.2/3, df_ttl_T2["touch_social_3"]),
↳df_ttl_T2["touch_social_3"])

```



```

df_ttl_T2["touch_organic_search_3"] = np.where(df_ttl_T2['touch3'] ==
↳"organic_search", np.where(df_ttl_T2["count5"] == 1, 0.2/3,
↳df_ttl_T2["touch_organic_search_3"]), df_ttl_T2["touch_organic_search_3"])
df_ttl_T2["touch_email_4"] = np.where(df_ttl_T2['touch4'] == "email", np.
↳where(df_ttl_T2["count5"] == 1, 0.2/3, df_ttl_T2["touch_email_4"]),
↳df_ttl_T2["touch_email_4"])
df_ttl_T2["touch_referral_4"] = np.where(df_ttl_T2['touch4'] == "referral",
↳np.where(df_ttl_T2["count5"] == 1, 0.2/3, df_ttl_T2["touch_referral_4"]),
↳df_ttl_T2["touch_referral_4"])
df_ttl_T2["touch_paid_search_4"] = np.where(df_ttl_T2['touch4'] ==
↳"paid_search", np.where(df_ttl_T2["count5"] == 1, 0.2/3,
↳df_ttl_T2["touch_paid_search_4"]), df_ttl_T2["touch_paid_search_4"])
df_ttl_T2["touch_direct_4"] = np.where(df_ttl_T2['touch4'] == "direct", np.
↳where(df_ttl_T2["count5"] == 1, 0.2/3, df_ttl_T2["touch_direct_4"]),
↳df_ttl_T2["touch_direct_4"])
df_ttl_T2["touch_display_4"] = np.where(df_ttl_T2['touch4'] == "display",
↳np.where(df_ttl_T2["count5"] == 1, 0.2/3, df_ttl_T2["touch_display_4"]),
↳df_ttl_T2["touch_display_4"])
df_ttl_T2["touch_social_4"] = np.where(df_ttl_T2['touch4'] == "social", np.
↳where(df_ttl_T2["count5"] == 1, 0.2/3, df_ttl_T2["touch_social_4"]),
↳df_ttl_T2["touch_social_4"])
df_ttl_T2["touch_organic_search_4"] = np.where(df_ttl_T2['touch4'] ==
↳"organic_search", np.where(df_ttl_T2["count5"] == 1, 0.2/3,
↳df_ttl_T2["touch_organic_search_4"]), df_ttl_T2["touch_organic_search_4"])
df_ttl_T2["touch_email_5"] = np.where(df_ttl_T2['touch5'] == "email", np.
↳where(df_ttl_T2["count5"] == 1, 0.4, df_ttl_T2["touch_email_5"]),
↳df_ttl_T2["touch_email_5"])
df_ttl_T2["touch_referral_5"] = np.where(df_ttl_T2['touch5'] == "referral",
↳np.where(df_ttl_T2["count5"] == 1, 0.4, df_ttl_T2["touch_referral_5"]),
↳df_ttl_T2["touch_referral_5"])
df_ttl_T2["touch_paid_search_5"] = np.where(df_ttl_T2['touch5'] ==
↳"paid_search", np.where(df_ttl_T2["count5"] == 1, 0.4,
↳df_ttl_T2["touch_paid_search_5"]), df_ttl_T2["touch_paid_search_5"])
df_ttl_T2["touch_direct_5"] = np.where(df_ttl_T2['touch5'] == "direct", np.
↳where(df_ttl_T2["count5"] == 1, 0.4, df_ttl_T2["touch_direct_5"]),
↳df_ttl_T2["touch_direct_5"])
df_ttl_T2["touch_display_5"] = np.where(df_ttl_T2['touch5'] == "display",
↳np.where(df_ttl_T2["count5"] == 1, 0.4, df_ttl_T2["touch_display_5"]),
↳df_ttl_T2["touch_display_5"])
df_ttl_T2["touch_social_5"] = np.where(df_ttl_T2['touch5'] == "social", np.
↳where(df_ttl_T2["count5"] == 1, 0.4, df_ttl_T2["touch_social_5"]),
↳df_ttl_T2["touch_social_5"])
df_ttl_T2["touch_organic_search_5"] = np.where(df_ttl_T2['touch5'] ==
↳"organic_search", np.where(df_ttl_T2["count5"] == 1, 0.4,
↳df_ttl_T2["touch_organic_search_5"]), df_ttl_T2["touch_organic_search_5"])
#Calculate position based channel effectiveness

```

```

df_ttl_T2["email_total"] = df_ttl_T2["touch_email_1"] +_
↳df_ttl_T2["touch_email_2"] + df_ttl_T2["touch_email_3"] +_
↳df_ttl_T2["touch_email_4"] + df_ttl_T2["touch_email_5"]
df_ttl_T2["referral_total"] = df_ttl_T2["touch_referral_1"] +_
↳df_ttl_T2["touch_referral_2"] + df_ttl_T2["touch_referral_3"] +_
↳df_ttl_T2["touch_referral_4"] + df_ttl_T2["touch_referral_5"]
df_ttl_T2["paid_search_total"] = df_ttl_T2["touch_paid_search_1"] +_
↳df_ttl_T2["touch_paid_search_2"] + df_ttl_T2["touch_paid_search_3"] +_
↳df_ttl_T2["touch_paid_search_4"] + df_ttl_T2["touch_paid_search_5"]
df_ttl_T2["direct_total"] = df_ttl_T2["touch_direct_1"] +_
↳df_ttl_T2["touch_direct_2"] + df_ttl_T2["touch_direct_3"] +_
↳df_ttl_T2["touch_direct_4"] + df_ttl_T2["touch_direct_5"]
df_ttl_T2["display_total"] = df_ttl_T2["touch_display_1"] +_
↳df_ttl_T2["touch_display_2"] + df_ttl_T2["touch_display_3"] +_
↳df_ttl_T2["touch_display_4"] + df_ttl_T2["touch_display_5"]
df_ttl_T2["social_total"] = df_ttl_T2["touch_social_1"] +_
↳df_ttl_T2["touch_social_2"] + df_ttl_T2["touch_social_3"] +_
↳df_ttl_T2["touch_social_4"] + df_ttl_T2["touch_social_5"]
df_ttl_T2["organic_search_total"] = df_ttl_T2["touch_organic_search_1"] +_
↳df_ttl_T2["touch_organic_search_2"] + df_ttl_T2["touch_organic_search_3"] +_
↳df_ttl_T2["touch_organic_search_4"] + df_ttl_T2["touch_organic_search_5"]
# df_ttl_T2["email_total%"] = df_ttl_T2["email_total"] / df_ttl_T2["tttotal"]
# df_ttl_T2["referral_total%"] = df_ttl_T2["referral_total"] /_
↳df_ttl_T2["tttotal"]
# df_ttl_T2["paid_search_total%"] = df_ttl_T2["paid_search_total"] /_
↳df_ttl_T2["tttotal"]
# df_ttl_T2["direct_total%"] = df_ttl_T2["direct_total"] /_
↳df_ttl_T2["tttotal"]
# df_ttl_T2["display_total%"] = df_ttl_T2["display_total"] /_
↳df_ttl_T2["tttotal"]
# df_ttl_T2["social_total%"] = df_ttl_T2["social_total"] /_
↳df_ttl_T2["tttotal"]
# df_ttl_T2["organic_search_total%"] = df_ttl_T2["organic_search_total"] /_
↳df_ttl_T2["tttotal"]
del df_ttl_T2["email_total%"]
del df_ttl_T2["referral_total%"]
del df_ttl_T2["paid_search_total%"]
del df_ttl_T2["direct_total%"]
del df_ttl_T2["display_total%"]
del df_ttl_T2["social_total%"]
del df_ttl_T2["organic_search_total%"]

df_ttl_T2

```

```

[16]:      convert_TF      touch1      touch2      touch3      touch4 touch5      tier \
0          True      email      referral      referral      None      None      1

```

1	True	referral	referral	email	None	None	1
2	True	referral	None	None	None	None	1
3	True	email	None	None	None	None	1
4	True	referral	referral	social	referral	None	1
...
60648	True	referral	referral	None	None	None	3
60649	True	referral	referral	referral	referral	None	3
68103	True	organic_search	referral	referral	referral	None	3
68104	True	referral	referral	None	None	None	3
68117	True	direct	referral	None	None	None	3

	email	social	display	...	touch_organic_search_2	\
0	1000.0	1000.0	1000.0	...	0.0	
1	1000.0	1000.0	1000.0	...	0.0	
2	1000.0	1000.0	1000.0	...	0.0	
3	1000.0	1000.0	1000.0	...	0.0	
4	1000.0	1000.0	1000.0	...	0.0	
...	
60648	3000.0	3000.0	3000.0	...	0.0	
60649	3000.0	3000.0	3000.0	...	0.0	
68103	3000.0	3000.0	3000.0	...	0.0	
68104	3000.0	3000.0	3000.0	...	0.0	
68117	3000.0	3000.0	3000.0	...	0.0	

	touch_organic_search_3	touch_organic_search_4	touch_organic_search_5	\
0	0.0	0.0	0.0	
1	0.0	0.0	0.0	
2	0.0	0.0	0.0	
3	0.0	0.0	0.0	
4	0.0	0.0	0.0	
...	
60648	0.0	0.0	0.0	
60649	0.0	0.0	0.0	
68103	0.0	0.0	0.0	
68104	0.0	0.0	0.0	
68117	0.0	0.0	0.0	

	organic_search_total	count1	count2	count3	count4	count5
0	0.0	0	0	1	0	0
1	0.0	0	0	1	0	0
2	0.0	1	0	0	0	0
3	0.0	1	0	0	0	0
4	0.0	0	0	0	1	0
...
60648	0.0	0	1	0	0	0
60649	0.0	0	0	0	1	0
68103	0.4	0	0	0	1	0

68104	0.0	0	1	0	0	0
68117	0.0	0	1	0	0	0

[13779 rows x 67 columns]

```
[17]: email_total_sum = df_ttl_T2["email_total"].sum()
referral_total_sum = df_ttl_T2["referral_total"].sum()
paid_search_total_sum = df_ttl_T2["paid_search_total"].sum()
direct_total_sum = df_ttl_T2["direct_total"].sum()
display_total_sum = df_ttl_T2["display_total"].sum()
social_total_sum = df_ttl_T2["social_total"].sum()
organic_search_total_sum = df_ttl_T2["organic_search_total"].sum()

total_sums = [{"Email total", email_total_sum}, {"Referral total",
→ referral_total_sum}, {"Paid Search total", paid_search_total_sum}, {"Direct
→ total", direct_total_sum}, {"Social total", social_total_sum}, {"Organic
→ Search total", organic_search_total_sum}]

total_sums.sort(key=lambda x:x[1])
total_sums.reverse()
total_sums
```

```
[17]: [['Referral total', 7085.366666666668],
['Social total', 2452.5333333333338],
['Email total', 1326.4333333333334],
['Paid Search total', 781.9333333333334],
['Organic Search total', 13.1],
['Direct total', 8.8]]
```

```
[18]: for i in total_sums:
    print(str(i[0]) + ": " + str(i[1]))
```

```
Referral total: 7085.366666666668
Social total: 2452.5333333333338
Email total: 1326.4333333333334
Paid Search total: 781.9333333333334
Organic Search total: 13.1
Direct total: 8.8
```

```
[19]: position_based_total_sums_df = pd.DataFrame.from_records(total_sums)
position_based_total_sums_df
```

```
[19]:
```

		0	1
0	Referral total	7085.366667	
1	Social total	2452.533333	
2	Email total	1326.433333	
3	Paid Search total	781.933333	

```

4 Organic Search total    13.100000
5           Direct total    8.800000

```

0.0.3 Time to create a table with the actual values separated by tiers (so 3 tiers each, for each allocation system, across all 7 channels)

```

[20]: #First method: first interaction
new_df_tier1 = new_df[new_df['tier']==1]
new_df_tier2 = new_df[new_df['tier']==2]
new_df_tier3 = new_df[new_df['tier']==3]

```

```

[21]: df_t1 = new_df_tier1['touch1']
df_t1.value_counts()

```

```

[21]: referral      1493
display      484
social      471
email      246
paid_search   128
direct         2
Name: touch1, dtype: int64

```

```

[22]: df_t2 = new_df_tier2['touch1']
df_t2.value_counts()

```

```

[22]: referral      2674
social      798
display      735
email      430
paid_search   247
organic_search    5
direct         3
Name: touch1, dtype: int64

```

```

[23]: df_t3 = new_df_tier3['touch1']
df_t3.value_counts()

```

```

[23]: referral      3313
social      988
display      840
email      589
paid_search   322
direct         6
organic_search    5
Name: touch1, dtype: int64

```

```
[24]: #Referral / display / social / email / paid_search / organic_search / direct
first_t1_list = [1493, 484, 471, 246, 128, 0, 2]
first_t2_list = [2674, 735, 798, 430, 247, 5, 3]
first_t3_list = [3313, 840, 988, 589, 322, 5, 6]
```

```
[25]: #Second method: Linear
df_ttl_T_tier1 = df_ttl_T[df_ttl_T['tier']==1]
df_ttl_T_tier2 = df_ttl_T[df_ttl_T['tier']==2]
df_ttl_T_tier3 = df_ttl_T[df_ttl_T['tier']==3]

email_total_sum_tier1 = df_ttl_T_tier1["email_total%"].sum()
referral_total_sum_tier1 = df_ttl_T_tier1["referral_total%"].sum()
paid_search_total_sum_tier1 = df_ttl_T_tier1["paid_search_total%"].sum()
direct_total_sum_tier1 = df_ttl_T_tier1["direct_total%"].sum()
display_total_sum_tier1 = df_ttl_T_tier1["display_total%"].sum()
social_total_sum_tier1 = df_ttl_T_tier1["social_total%"].sum()
organic_search_total_sum_tier1 = df_ttl_T_tier1["organic_search_total%"].sum()

total_sums_tier1 = [{"Email total", email_total_sum_tier1}, {"Referral total",
→referral_total_sum_tier1}, {"Paid Search total",
→paid_search_total_sum_tier1}, {"Direct total", direct_total_sum_tier1},
→{"Social total", social_total_sum_tier1}, {"Organic Search total",
→organic_search_total_sum_tier1}]

total_sums_tier1.sort(key=lambda x:x[1])
total_sums_tier1.reverse()
total_sums_tier1
```

```
[25]: [['Referral total', 1503.1166666666668],
['Social total', 479.01666666666665],
['Email total', 243.66666666666669],
['Paid Search total', 127.16666666666666],
['Organic Search total', 2.3499999999999996],
['Direct total', 2.25]]
```

```
[26]: email_total_sum_tier2 = df_ttl_T_tier2["email_total%"].sum()
referral_total_sum_tier2 = df_ttl_T_tier2["referral_total%"].sum()
paid_search_total_sum_tier2 = df_ttl_T_tier2["paid_search_total%"].sum()
direct_total_sum_tier2 = df_ttl_T_tier2["direct_total%"].sum()
display_total_sum_tier2 = df_ttl_T_tier2["display_total%"].sum()
social_total_sum_tier2 = df_ttl_T_tier2["social_total%"].sum()
organic_search_total_sum_tier2 = df_ttl_T_tier2["organic_search_total%"].sum()
```

```
total_sums_tier2 = [{"Email total", email_total_sum_tier2}, {"Referral total",
↳ referral_total_sum_tier2}, {"Paid Search total",
↳ paid_search_total_sum_tier2}, {"Direct total", direct_total_sum_tier2},
↳ {"Social total", social_total_sum_tier2}, {"Organic Search total",
↳ organic_search_total_sum_tier2}]

total_sums_tier2.sort(key=lambda x:x[1])
total_sums_tier2.reverse()
total_sums_tier2
```

```
[26]: [['Referral total', 2682.1666666666665],
      ['Social total', 824.3833333333333],
      ['Email total', 419.05],
      ['Paid Search total', 246.63333333333333],
      ['Organic Search total', 3.25],
      ['Direct total', 1.3333333333333333]]
```

```
[27]: email_total_sum_tier3 = df_ttl_T_tier3["email_total%"].sum()
referral_total_sum_tier3 = df_ttl_T_tier3["referral_total%"].sum()
paid_search_total_sum_tier3 = df_ttl_T_tier3["paid_search_total%"].sum()
direct_total_sum_tier3 = df_ttl_T_tier3["direct_total%"].sum()
display_total_sum_tier3 = df_ttl_T_tier3["display_total%"].sum()
social_total_sum_tier3 = df_ttl_T_tier3["social_total%"].sum()
organic_search_total_sum_tier3 = df_ttl_T_tier3["organic_search_total%"].sum()

total_sums_tier3 = [{"Email total", email_total_sum_tier3}, {"Referral total",
↳ referral_total_sum_tier3}, {"Paid Search total",
↳ paid_search_total_sum_tier3}, {"Direct total", direct_total_sum_tier3},
↳ {"Social total", social_total_sum_tier3}, {"Organic Search total",
↳ organic_search_total_sum_tier3}]

total_sums_tier3.sort(key=lambda x:x[1])
total_sums_tier3.reverse()
total_sums_tier3
```

```
[27]: [['Referral total', 3285.2833333333333],
      ['Social total', 988.0833333333333],
      ['Email total', 595.1333333333333],
      ['Paid Search total', 334.18333333333334],
      ['Direct total', 5.366666666666666],
      ['Organic Search total', 5.2333333333333325]]
```

```
[28]: total_sums_tier1_list = [referral_total_sum_tier1, display_total_sum_tier1,
↳ social_total_sum_tier1, email_total_sum_tier1, paid_search_total_sum_tier1,
↳ organic_search_total_sum_tier1, direct_total_sum_tier1]
```

```
total_sums_tier2_list = [referral_total_sum_tier2, display_total_sum_tier2,
↳social_total_sum_tier2, email_total_sum_tier2, paid_search_total_sum_tier2,
↳organic_search_total_sum_tier2, direct_total_sum_tier2]
total_sums_tier3_list = [referral_total_sum_tier3, display_total_sum_tier3,
↳social_total_sum_tier3, email_total_sum_tier3, paid_search_total_sum_tier3,
↳organic_search_total_sum_tier3, direct_total_sum_tier3]
```

```
[29]: linear_tier1_sums_df = pd.DataFrame.from_records(total_sums_tier1)
linear_tier2_sums_df = pd.DataFrame.from_records(total_sums_tier2)
linear_tier3_sums_df = pd.DataFrame.from_records(total_sums_tier3)
linear_tier1_sums_df
```

```
[29]:
```

	0	1
0	Referral total	1503.116667
1	Social total	479.016667
2	Email total	243.666667
3	Paid Search total	127.166667
4	Organic Search total	2.350000
5	Direct total	2.250000

```
[30]: linear_tier2_sums_df
```

```
[30]:
```

	0	1
0	Referral total	2682.166667
1	Social total	824.383333
2	Email total	419.050000
3	Paid Search total	246.633333
4	Organic Search total	3.250000
5	Direct total	1.333333

```
[31]: linear_tier3_sums_df
```

```
[31]:
```

	0	1
0	Referral total	3285.283333
1	Social total	988.083333
2	Email total	595.133333
3	Paid Search total	334.183333
4	Direct total	5.366667
5	Organic Search total	5.233333

```
[32]: #Method 3: Position based
df_ttl_T2_tier1 = df_ttl_T2[df_ttl_T2['tier']==1]
df_ttl_T2_tier2 = df_ttl_T2[df_ttl_T2['tier']==2]
df_ttl_T2_tier3 = df_ttl_T2[df_ttl_T2['tier']==3]

email_total_sum_tier1 = df_ttl_T2_tier1["email_total"].sum()
referral_total_sum_tier1 = df_ttl_T2_tier1["referral_total"].sum()
```



```

paid_search_total_sum_tier1 = df_ttl_T2_tier1["paid_search_total"].sum()
direct_total_sum_tier1 = df_ttl_T2_tier1["direct_total"].sum()
display_total_sum_tier1 = df_ttl_T2_tier1["display_total"].sum()
social_total_sum_tier1 = df_ttl_T2_tier1["social_total"].sum()
organic_search_total_sum_tier1 = df_ttl_T2_tier1["organic_search_total"].sum()

total_sums2_tier1 = [{"Email total", email_total_sum_tier1}, {"Referral total",
→referral_total_sum_tier1}, {"Paid Search total",
→paid_search_total_sum_tier1}, {"Direct total", direct_total_sum_tier1},
→{"Social total", social_total_sum_tier1}, {"Organic Search total",
→organic_search_total_sum_tier1}]

total_sums2_tier1.sort(key=lambda x:x[1])
total_sums2_tier1.reverse()
total_sums2_tier1

```

```

[32]: [['Referral total', 1419.9],
      ['Social total', 512.7333333333333],
      ['Email total', 258.4],
      ['Paid Search total', 141.5],
      ['Organic Search total', 3.433333333333333],
      ['Direct total', 2.4]]

```

```

[33]: email_total_sum_tier2 = df_ttl_T2_tier2["email_total"].sum()
referral_total_sum_tier2 = df_ttl_T2_tier2["referral_total"].sum()
paid_search_total_sum_tier2 = df_ttl_T2_tier2["paid_search_total"].sum()
direct_total_sum_tier2 = df_ttl_T2_tier2["direct_total"].sum()
display_total_sum_tier2 = df_ttl_T2_tier2["display_total"].sum()
social_total_sum_tier2 = df_ttl_T2_tier2["social_total"].sum()
organic_search_total_sum_tier2 = df_ttl_T2_tier2["organic_search_total"].sum()

total_sums2_tier2 = [{"Email total", email_total_sum_tier2}, {"Referral total",
→referral_total_sum_tier2}, {"Paid Search total",
→paid_search_total_sum_tier2}, {"Direct total", direct_total_sum_tier2},
→{"Social total", social_total_sum_tier2}, {"Organic Search total",
→organic_search_total_sum_tier2}]

total_sums2_tier2.sort(key=lambda x:x[1])
total_sums2_tier2.reverse()
total_sums2_tier2

```

```

[33]: [['Referral total', 2533.4666666666662],
      ['Social total', 881.2666666666667],
      ['Email total', 451.6333333333333],
      ['Paid Search total', 270.5666666666666],
      ['Organic Search total', 3.6000000000000005],
      ['Direct total', 1.4000000000000001]]

```

```
[34]: email_total_sum_tier3 = df_ttl_T2_tier3["email_total"].sum()
referral_total_sum_tier3 = df_ttl_T2_tier3["referral_total"].sum()
paid_search_total_sum_tier3 = df_ttl_T2_tier3["paid_search_total"].sum()
direct_total_sum_tier3 = df_ttl_T2_tier3["direct_total"].sum()
display_total_sum_tier3 = df_ttl_T2_tier3["display_total"].sum()
social_total_sum_tier3 = df_ttl_T2_tier3["social_total"].sum()
organic_search_total_sum_tier3 = df_ttl_T2_tier3["organic_search_total"].sum()

total_sums2_tier3 = [{"Email total", email_total_sum_tier3}, {"Referral total",
→ referral_total_sum_tier3}, {"Paid Search total",
→ paid_search_total_sum_tier3}, {"Direct total", direct_total_sum_tier3},
→ {"Social total", social_total_sum_tier3}, {"Organic Search total",
→ organic_search_total_sum_tier3}]

total_sums2_tier3.sort(key=lambda x:x[1])
total_sums2_tier3.reverse()
total_sums2_tier3
```

```
[34]: [['Referral total', 3132.0],
['Social total', 1058.5333333333333],
['Email total', 616.40000000000001],
['Paid Search total', 369.86666666666667],
['Organic Search total', 6.066666666666666],
['Direct total', 5.0]]
```

```
[35]: total_sums2_tier1_list = [referral_total_sum_tier1, display_total_sum_tier1,
→ social_total_sum_tier1, email_total_sum_tier1, paid_search_total_sum_tier1,
→ organic_search_total_sum_tier1, direct_total_sum_tier1]
total_sums2_tier2_list = [referral_total_sum_tier2, display_total_sum_tier2,
→ social_total_sum_tier2, email_total_sum_tier2, paid_search_total_sum_tier2,
→ organic_search_total_sum_tier2, direct_total_sum_tier2]
total_sums2_tier3_list = [referral_total_sum_tier3, display_total_sum_tier3,
→ social_total_sum_tier3, email_total_sum_tier3, paid_search_total_sum_tier3,
→ organic_search_total_sum_tier3, direct_total_sum_tier3]
```

```
[36]: linear_tier1_sums2_df = pd.DataFrame.from_records(total_sums2_tier1)
linear_tier2_sums2_df = pd.DataFrame.from_records(total_sums2_tier2)
linear_tier3_sums2_df = pd.DataFrame.from_records(total_sums2_tier3)
linear_tier1_sums2_df
```

```
[36]:
```

	0	1
0	Referral total	1419.900000
1	Social total	512.733333
2	Email total	258.400000
3	Paid Search total	141.500000
4	Organic Search total	3.433333
5	Direct total	2.400000

```
[37]: linear_tier2_sums2_df
```

```
[37]:
```

	0	1
0	Referral total	2533.466667
1	Social total	881.266667
2	Email total	451.633333
3	Paid Search total	270.566667
4	Organic Search total	3.600000
5	Direct total	1.400000

```
[38]: linear_tier3_sums2_df
```

```
[38]:
```

	0	1
0	Referral total	3132.000000
1	Social total	1058.533333
2	Email total	616.400000
3	Paid Search total	369.866667
4	Organic Search total	6.066667
5	Direct total	5.000000

```
[39]: data = {'Channel': ["Referral total", "Display total", "Social total", "Email_↪total", "Paid Search total", "Organic Search total", "Direct total"]}

final_df = pd.DataFrame(data)
```

```
[40]: #'Linear Tier 1', 'Linear Tier 2', 'Linear Tier 3', 'Position based Tier 1', ↪
↪'Position based Tier 2', 'Position based Tier 3']

final_df["First Interaction Tier 1"] = first_t1_list
final_df["First Interaction Tier 2"] = first_t2_list
final_df["First Interaction Tier 3"] = first_t3_list
final_df["Linear Tier 1"] = total_sums_tier1_list
final_df["Linear Tier 2"] = total_sums_tier2_list
final_df["Linear Tier 3"] = total_sums_tier3_list
final_df["Position-based Tier 1"] = total_sums2_tier1_list
final_df["Position-based Tier 2"] = total_sums2_tier2_list
final_df["Position-based Tier 3"] = total_sums2_tier3_list
final_df
```

```
[40]:
```

	Channel	First Interaction Tier 1	First Interaction Tier 2	\
0	Referral total	1493	2674	
1	Display total	484	735	
2	Social total	471	798	
3	Email total	246	430	
4	Paid Search total	128	247	
5	Organic Search total	0	5	
6	Direct total	2	3	

	First Interaction	Tier 3	Linear Tier 1	Linear Tier 2	Linear Tier 3	\
0		3313	1503.116667	2682.166667	3285.283333	
1		840	466.433333	715.183333	849.716667	
2		988	479.016667	824.383333	988.083333	
3		589	243.666667	419.050000	595.133333	
4		322	127.166667	246.633333	334.183333	
5		5	2.350000	3.250000	5.233333	
6		6	2.250000	1.333333	5.366667	

	Position-based Tier 1	Position-based Tier 2	Position-based Tier 3
0	1419.900000	2533.466667	3132.000000
1	487.433333	750.066667	876.033333
2	512.733333	881.266667	1058.533333
3	258.400000	451.633333	616.400000
4	141.500000	270.566667	369.866667
5	3.433333	3.600000	6.066667
6	2.400000	1.400000	5.000000

```
[42]: final_df.to_csv("final_df_allocation.csv")
```

```
[ ]:
```

```
[41]: selection = df_ttl_T2.iloc[:,15:35]
print(selection)
```

	t2	t3	t4	t5	ttotal	touch_email_1	touch_email_2	touch_email_3	\
0	1	1	0	0	3	0.4	0.0	0.0	
1	1	1	0	0	3	0.0	0.0	0.4	
2	0	0	0	0	1	0.0	0.0	0.0	
3	0	0	0	0	1	1.0	0.0	0.0	
4	1	1	1	0	4	0.0	0.0	0.0	
...	
60648	1	0	0	0	2	0.0	0.0	0.0	
60649	1	1	1	0	4	0.0	0.0	0.0	
68103	1	1	1	0	4	0.0	0.0	0.0	
68104	1	0	0	0	2	0.0	0.0	0.0	
68117	1	0	0	0	2	0.0	0.0	0.0	

	touch_email_4	touch_email_5	email_total	touch_referral_1	\
0	0.0	0.0	0.4	0.0	
1	0.0	0.0	0.4	0.4	
2	0.0	0.0	0.0	1.0	
3	0.0	0.0	1.0	0.0	
4	0.0	0.0	0.0	0.4	
...	
60648	0.0	0.0	0.0	0.5	

60649	0.0	0.0	0.0	0.4
68103	0.0	0.0	0.0	0.0
68104	0.0	0.0	0.0	0.5
68117	0.0	0.0	0.0	0.0

	touch_referral_2	touch_referral_3	touch_referral_4	touch_referral_5	\
0	0.2	0.4	0.0	0.0	
1	0.2	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	
4	0.1	0.0	0.4	0.0	
...	
60648	0.5	0.0	0.0	0.0	
60649	0.1	0.1	0.4	0.0	
68103	0.1	0.1	0.4	0.0	
68104	0.5	0.0	0.0	0.0	
68117	0.5	0.0	0.0	0.0	

	referral_total	touch_paid_search_1	touch_paid_search_2	\
0	0.6	0.0	0.0	
1	0.6	0.0	0.0	
2	1.0	0.0	0.0	
3	0.0	0.0	0.0	
4	0.9	0.0	0.0	
...	
60648	1.0	0.0	0.0	
60649	1.0	0.0	0.0	
68103	0.6	0.0	0.0	
68104	1.0	0.0	0.0	
68117	0.5	0.0	0.0	

	touch_paid_search_3
0	0.0
1	0.0
2	0.0
3	0.0
4	0.0
...	...
60648	0.0
60649	0.0
68103	0.0
68104	0.0
68117	0.0

[13779 rows x 20 columns]

```
[44]: df_ttl_T2["tttotal"].value_counts()
```

```
[44]: 3    5344
      2    2962
      4    2578
      1    1636
      5    1259
      Name: tttotal, dtype: int64
```

```
[ ]:
```

```
[ ]:
```