Capstone 02: the booking company.
Updated on 11/02/2020.

1-) Purpose.

We are going to work with data provided by a company that connects different clients to different firms. More specifically, the company allows clients to book (schedule) visits to firms in order to learn more about their production processes and final products. Each booking has an unique identification which is used as index in the data structure. For confidentiality reason, some information cannot and will not be made public. We have the following main objectives:

- Organize the data in a single dataframe format.
- Create a dictionary to help preserve the meaning of more obscure variables.
- Save all the newly organized information in the format of csv files as a backup.
- Analyze the dataframe looking for missing data or mistakes.
- Select a balanced subset of firms to be included in a more focused analysis approach.
- Analyze the network between clients booking visits and firms being visited.
- Use supervised classification to predict relevant missing data.
- Use unsupervised learning techniques to identify potential groups (clusters) of bookings based on the information available about each individual booking.

Please, go to https://github.com/tmivanus/Springboard/tree/master/Capstone_02 to see all data and codes.

2-) Data.

The original data was stored in four different sheets inside a microsoft excel file named 'data_original'. That excel file was lightly organized and saved as a microsoft excel file named 'data_organized'. From 'data_organized' came three csv files: 'bookings_questions_data', 'bookings_addons_data' and 'main_data'. From the internet came other three csv files with necessary additional information: 'countries_data', 'us_regions_data' and 'us_zipcodes_data'. In particular, the source of 'us_zipcodes_data.csv' is https://simple.wikipedia.org/wiki/List_of_ZIP_Code_prefixes. In summary, we are going to work with the following six csv files in the 'data' directory:

```
- countries_data.csv
- us_regions_data.csv
- us_zipcodes_data.csv
- bookings_questions_data.csv
- bookings_addons_data.csv
- main_data.csv
```

Some of those files are not in fact made publicly available for confidentiality reason. All those files are handled by the 'data_preparation' code file. Please, check the 'data_preparation' code file for detailed information.

- 'countries_data.csv' contains the abbreviation code and the full name of several countries. From it comes 'country_dic', which is a dictionary with abbreviation code (key) and full name (value) for each country, and 'country_indic', which is the inverted 'country_dic' dictionary.
- 'us_regions_data.csv' contains the abbreviation code and the full name of all US states and territories. From it comes 'us_region_dic', which is a dictionary with abbreviation code (key) and full name (value) for each US state or territory, and 'us_region_indic', which is the inverted 'us_region_dic' dictionary.
- 'us_zipcodes_data.csv' contains the zipcode prefixes and the abbreviation code of US states and territories. From it comes 'us_zipcode_dic', which is a dictionary with zipcode prefixes (key) and abbreviation code (value) of US states or territories.
- 'bookings_questions_data.csv' contains extra four categorical questions associated with some bookings (named as questions 'q4' to 'q7'). From it comes questions_df, which is a dataframe with booking id and columns for questions 'q4' to 'q7', and four dictionaries describing those questions (named as dictionaries 'q4_dic' to 'q7_dic').
- 'bookings_addons_data.csv' contains add-ons purchases associated with some bookings. It informs price, quantity and revenue (price x quantity) of 27 different add-ons. Add-ons can be, for example, souvenirs or special additions to the booking. From that file comes 'addons_df', which is a dataframe with booking id and columns related to 'addon_01' through 'addon_27', and 'addons_dic', which is a dictionary with 'addon_01' through 'addon_27' as keys, and their descriptions as values.
- 'main_data.csv', as the name suggests, is the main file with several different information associate with all bookings between 2016-08-31 and 2017-12-31. There is information about the client booking a visit, the booking itself, and the firm to be visited. From it comes 'main_df', which is a dataframe with booking id and several columns of information, and many dictionaries describing some of the columns (named as dictionaries 'firm_id_dic', 'firm_type_dic', 'status_cancel_dic', 'source_online_dic', 'payment_cc_dic', 'discount_dic', 'q1_bef_dic', 'q1_aft_dic', 'q2_dic' and 'q3_dic')

'data_preparation' code file organize all those csv files in a single dataframe called 'final_df'. In addition, all dictionaries are preserved in 'final_dic' (a dictionary of dictionaries) and all publicly available dictionaries are preserved in 'final_dic_pub' (also a dictionary of dictionaries). 'data_preparation' code file also saves all this information in the format of four csv files located in a newly created 'data_new' directory:

```
- final_df.csv
- final_df_dtypelist.csv
- final_dic_df.csv
- final_dic_pub_df.csv
```

'final_df.csv' and 'final_df_dtypelist.csv' are both necessary to recover 'final_df'. 'final_dic_df.csv' is necessary to recover 'final_dic'. Finally, 'final_dic_pub_df.csv' is necessary to recover 'final_dic_pub'. Please, check the last code cells in 'data_preparation' code file for a demonstration of how to recover 'final_df', 'final_dic' and 'final_dic_pub' using the csv files in the 'data_new' directory. In reality, for confidentiality reason, 'final_dic' is not made publicly available. Instead, only its censored version 'final_dic_pub' is made publicly available.

3-) Data analysis.

3.1-) Getting to know the dataframe and initial trimming.

We are going to use the 'data_analysis_and_results' code file to analyze the data stored in the dataframe 'final_df', which is located in the directory 'data_new'. The reader can check the 'data_preparation' code file to find out how 'final_df' was built. Many columns in that dataframe have self-explanatory titles. Other columns have descriptions located in the nested dictionary 'final_dic' (a dictionary of dictionaries). In reality, for confidentiality reason, 'final_dic' is not made publicly available. Instead, only its censored version 'final_dic_pub' is made publicly available.

The files stored in the directory 'data_new' are actually csv files. We use them to recover 'final_df' and 'final_dic_pub' (the recovering of 'final_dic' is for private use only). In order to keep things safe and simple, we actually work with a copy of 'final_df' called just 'df' and a copy of 'final_dic_pub' called just 'df_dic_pub' (there is also a copy of 'final_dic' called 'df_dic' but it is for private use only). Below is a quick overview of 'df' and 'df_dic_pub'.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 56640 entries, 132755 to 295082
Data columns (total 112 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   age               9051 non-null   Int64
 1   age_range         9051 non-null   object
 2   country           41135 non-null  object
 3   us_region         40599 non-null  object
 4   us_zipcode        40529 non-null  object
 5   firm_id           56425 non-null  float64
 6   firm_type         56425 non-null  float64
 7   firm_loc          56425 non-null  object
 8   contact_date      56425 non-null  datetime64[ns]
 9   visit_datetime    56425 non-null  datetime64[ns]
 10  status_cancel     56425 non-null  float64
 11  source_online     56425 non-null  float64
 12  payment_cc        56425 non-null  float64
 13  guests            56425 non-null  float64
 14  guests_in         34521 non-null  Int64
 15  fee               56425 non-null  float64
 16  discount          56425 non-null  float64
 17  tot_quant_addons  56425 non-null  float64
 18  tot_rev_addons    56425 non-null  float64
 19  tot_rev           56425 non-null  float64
 20  q1_bef            35154 non-null  Int64
 21  q1_aft            16412 non-null  Int64
 22  q2                35437 non-null  Int64
 23  q3                10561 non-null  Int64
 24  feedback          5313 non-null   object
 25  q4                1355 non-null   Int64
 26  q5                5978 non-null   Int64
 27  q6                5961 non-null   Int64
 28  q7                247 non-null    Int64
 29  merge_questions   56640 non-null  object
 30  addon_01_price    1624 non-null   float64
```

```
31  addon_02_price    545 non-null    float64
32  addon_03_price    55 non-null     float64
33  addon_04_price    291 non-null    float64
34  addon_05_price    226 non-null    float64
35  addon_06_price    169 non-null    float64
36  addon_07_price    17 non-null     float64
37  addon_08_price    93 non-null     float64
38  addon_09_price    14 non-null     float64
39  addon_10_price    98 non-null     float64
40  addon_11_price    37 non-null     float64
41  addon_12_price    50 non-null     float64
42  addon_13_price    88 non-null     float64
43  addon_14_price    90 non-null     float64
44  addon_15_price    18 non-null     float64
45  addon_16_price    62 non-null     float64
46  addon_17_price    26 non-null     float64
47  addon_18_price    33 non-null     float64
48  addon_19_price    33 non-null     float64
49  addon_20_price    13 non-null     float64
50  addon_21_price    8 non-null      float64
51  addon_22_price    20 non-null     float64
52  addon_23_price    13 non-null     float64
53  addon_24_price    6 non-null      float64
54  addon_25_price    12 non-null     float64
55  addon_26_price    6 non-null      float64
56  addon_27_price    1 non-null      float64
57  addon_01_quant    1624 non-null   float64
58  addon_02_quant    545 non-null    float64
59  addon_03_quant    55 non-null     float64
60  addon_04_quant    291 non-null    float64
61  addon_05_quant    226 non-null    float64
62  addon_06_quant    169 non-null    float64
63  addon_07_quant    17 non-null     float64
64  addon_08_quant    93 non-null     float64
65  addon_09_quant    14 non-null     float64
66  addon_10_quant    98 non-null     float64
67  addon_11_quant    37 non-null     float64
68  addon_12_quant    50 non-null     float64
69  addon_13_quant    88 non-null     float64
70  addon_14_quant    90 non-null     float64
71  addon_15_quant    18 non-null     float64
72  addon_16_quant    62 non-null     float64
73  addon_17_quant    26 non-null     float64
74  addon_18_quant    33 non-null     float64
75  addon_19_quant    33 non-null     float64
76  addon_20_quant    13 non-null     float64
77  addon_21_quant    8 non-null      float64
78  addon_22_quant    20 non-null     float64
79  addon_23_quant    13 non-null     float64
80  addon_24_quant    6 non-null      float64
81  addon_25_quant    12 non-null     float64
82  addon_26_quant    6 non-null      float64
83  addon_27_quant    1 non-null      float64
84  addon_01_rev      1624 non-null   float64
85  addon_02_rev      545 non-null    float64
86  addon_03_rev      55 non-null     float64
87  addon_04_rev      291 non-null    float64
```

```
 88  addon_05_rev      226 non-null    float64
 89  addon_06_rev      169 non-null    float64
 90  addon_07_rev      17 non-null     float64
 91  addon_08_rev      93 non-null     float64
 92  addon_09_rev      14 non-null     float64
 93  addon_10_rev      98 non-null     float64
 94  addon_11_rev      37 non-null     float64
 95  addon_12_rev      50 non-null     float64
 96  addon_13_rev      88 non-null     float64
 97  addon_14_rev      90 non-null     float64
 98  addon_15_rev      18 non-null     float64
 99  addon_16_rev      62 non-null     float64
 100 addon_17_rev      26 non-null     float64
 101 addon_18_rev      33 non-null     float64
 102 addon_19_rev      33 non-null     float64
 103 addon_20_rev      13 non-null     float64
 104 addon_21_rev      8 non-null      float64
 105 addon_22_rev      20 non-null     float64
 106 addon_23_rev      13 non-null     float64
 107 addon_24_rev      6 non-null      float64
 108 addon_25_rev      12 non-null     float64
 109 addon_26_rev      6 non-null      float64
 110 addon_27_rev      1 non-null      float64
 111 merge_addons      56640 non-null  object
dtypes: Int64(10), datetime64[ns](2), float64(92), object(8)
memory usage: 49.4+ MB
None

'df_dic_pub' keys:
dict_keys(['country', 'us_region', 'us_zipcode', 'firm_id', 'firm_type',
'status_cancel', 'source_online', 'payment_cc', 'discount', 'q1_bef', 'q1_aft',
'q2', 'q3', 'q4', 'q5', 'q6', 'q7', 'addons'])
```

We first take a closer look at some variables to better understand them and perhaps detect possible problems. There are price, quantity and revenue for 27 addons (addons 01 to 27). Indeed, revenue equals to price multiplied by quantity for every single addon. No problems there.

'tot_rev_addons' should supposedly be equal to the horizontal sum of revenues for addons 01 to 27. That is true for 56320 out of 56640 entries, but that is not true for 320 out of 56640 entries. All those 320 entries show 'tot_rev_addons' larger than the horizontal sum of revenues (minimum difference is 3.0 and maximum difference is 300.0).

'tot_quant_addons' should supposedly be equal to the horizontal sum of quantities for addons 01 to 27. Again, that is true for 56320 out of 56640 entries, but that is not true for 320 out of 56640 entries. All those 320 entries show 'tot_quant_addons' larger than the horizontal sum of quantities (minimum difference is 1.0 and maximum difference is 22.0).

Therefore, the reason 320 entries of 'tot_rev_addons' are larger than expected is because 320 entries of 'tot_quant_addons' are larger than expected. This can be a mistake or an indication that other types of addons exist but were not specified in the original database. We will assume the latter and maintain those 320 entries given that 320 is a small portion of 56640: a little more than 0.5% of the whole sample.

Finally, 'tot_rev' should supposedly be equal to 'fee' · 'guests' + 'tot_rev_addons'. That is true for 56638 out of 56640 entries, but that is not true for 2 out of 56640 entries, just a tiny fraction of the total. Those 2 entries show 'tot_rev' larger than expected, but the differences are less than a dollar, so their removal is not necessary.

There are 56425 non-null entries for 'firm_id', which is the variable that identifies the 11 firms in our sample. However, those 56426 entries are not equally distributed among the 11 firms. For example, 'firm_id' 38850 has 15374 entries while 'firm_id' 38433 has only 113 entries. Moreover, some variables expected to have the same number of non-null entries as 'firm_id' actually have less than 56425 non-null entries. They are showed in the table below. Variables about feedback, questions (those starting with 'q') and addons (those starting with 'addons') are not expected to have the same number of non-null entries as 'firm_id', and so are not included in the table below.

Amount of non-null entries in selected columns for each firm_id.

firm_id	firm_id_count	age	country	us_region	us_zipcode	guests_in
38850.0	15374	0	6239	6201	6200	10171
40638.0	11738	8612	9374	9016	9005	1382
26742.0	10266	0	8082	8014	7983	9655
36146.0	8986	0	8067	8054	8043	8084
34700.0	3736	0	3722	3698	3689	1
3254.0	2455	0	2259	2255	2254	2106
38204.0	1457	0	1256	1240	1239	1234
38852.0	1263	0	1214	1207	1204	1024
41065.0	558	439	445	442	441	413
40664.0	479	0	373	368	368	349
38433.0	113	0	104	104	103	102

The biggest problem is 'age' (and the correlated 'age_range'), which has 9051 entries unequally divided between only two 'firm_id'. Therefore, if we wanted to include 'age' in our analysis, we would have information available only for 'firm_id' 40638 and 41065, and the data would be distributed very unevenly between the two.

Another problematic variable in terms of non-null entries available is 'guests_in', which shows the number of guests (i.e., clients with booked visits) that actually checked in. The information availability changes a lot among firms. For example, 'firm_id' 34700, with 3736 non-null entries, only has 1 non-null entry for 'guests_in'.

If we ditch 'age' and 'guests_in', and then eliminate rows with null entries, we have the following result in terms of non-null entries in the selected columns (see the table below). We can now observe that 'firm_id' 40638, 36146 and 26742 have a high and similar amount of data. We are going to work only with these three firms in a more focused analysis approach. In other words, we are going to work only with 'firm_id' 40638, 36146 and 26742 so as to have a high and somewhat balanced amount of non-null entries among the firms.

Amount of non-null entries in selected columns for each firm_id.
Remaining number after excluding age, guests_in and rows with null entries.

firm_id	firm_id_count	country	us_region	us_zipcode
40638.0	9005	9005	9005	9005
36146.0	8043	8043	8043	8043
26742.0	7983	7983	7983	7983
38850.0	6200	6200	6200	6200
34700.0	3689	3689	3689	3689
3254.0	2254	2254	2254	2254
38204.0	1239	1239	1239	1239
38852.0	1204	1204	1204	1204
41065.0	441	441	441	441
40664.0	368	368	368	368
38433.0	103	103	103	103

Firms 40638 and 36146 are type 1 whereas firm 26742 is type 2. Firm 40638 is located in Saint Louis, MO. Firm 36146 is located in San Francisco, CA. Firm 26742 is located in Denver, CO. After having selected these three firms, we trimmed the dataframe 'df' by:

- ditching rows related to 'firm_id' other than 40638, 36146 and 26742.
- ditching rows with null entries based on columns expected to have equal number of non-null entries ('firm_id', 'country', 'us_region', 'us_zipcode').
- ditching 'age', 'age_range' and 'guests_in' columns due to the reasons explained before.
- ditching the unnecessary 'merge_questions' and 'merge_addons' columns.

It is worth noting that the original dataframe 'df' can be recovered any time if necessary by using the command df = final_df.copy(). After this first trimming process, the resulting dataframe 'df' is as follows, with firms 40638, 36146 and 26742 having respectively 9005, 8043 and 7983 non-null entries (25031 non-null entries in total).

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 25031 entries, 132755 to 187154
Data columns (total 107 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   country           25031 non-null  object
 1   us_region         25031 non-null  object
 2   us_zipcode        25031 non-null  object
 3   firm_id           25031 non-null  float64
 4   firm_type         25031 non-null  float64
 5   firm_loc          25031 non-null  object
 6   contact_date      25031 non-null  datetime64[ns]
 7   visit_datetime    25031 non-null  datetime64[ns]
 8   status_cancel     25031 non-null  float64
 9   source_online     25031 non-null  float64
 10  payment_cc        25031 non-null  float64
 11  guests            25031 non-null  float64
 12  fee               25031 non-null  float64
 13  discount          25031 non-null  float64
 14  tot_quant_addons  25031 non-null  float64
 15  tot_rev_addons    25031 non-null  float64
 16  tot_rev           25031 non-null  float64
```

```
17  q1_bef            24010 non-null  Int64
18  q1_aft            9191 non-null   Int64
19  q2                24136 non-null  Int64
20  q3                9001 non-null   Int64
21  feedback          4291 non-null   object
22  q4                0 non-null      Int64
23  q5                0 non-null      Int64
24  q6                0 non-null      Int64
25  q7                0 non-null      Int64
26  addon_01_price    1566 non-null   float64
27  addon_02_price    522 non-null    float64
28  addon_03_price    0 non-null      float64
29  addon_04_price    0 non-null      float64
30  addon_05_price    0 non-null      float64
31  addon_06_price    0 non-null      float64
32  addon_07_price    0 non-null      float64
33  addon_08_price    0 non-null      float64
34  addon_09_price    0 non-null      float64
35  addon_10_price    70 non-null     float64
36  addon_11_price    14 non-null     float64
37  addon_12_price    0 non-null      float64
38  addon_13_price    83 non-null     float64
39  addon_14_price    88 non-null     float64
40  addon_15_price    0 non-null      float64
41  addon_16_price    61 non-null     float64
42  addon_17_price    0 non-null      float64
43  addon_18_price    0 non-null      float64
44  addon_19_price    0 non-null      float64
45  addon_20_price    0 non-null      float64
46  addon_21_price    0 non-null      float64
47  addon_22_price    0 non-null      float64
48  addon_23_price    0 non-null      float64
49  addon_24_price    5 non-null      float64
50  addon_25_price    0 non-null      float64
51  addon_26_price    0 non-null      float64
52  addon_27_price    0 non-null      float64
53  addon_01_quant    1566 non-null   float64
54  addon_02_quant    522 non-null    float64
55  addon_03_quant    0 non-null      float64
56  addon_04_quant    0 non-null      float64
57  addon_05_quant    0 non-null      float64
58  addon_06_quant    0 non-null      float64
59  addon_07_quant    0 non-null      float64
60  addon_08_quant    0 non-null      float64
61  addon_09_quant    0 non-null      float64
62  addon_10_quant    70 non-null     float64
63  addon_11_quant    14 non-null     float64
64  addon_12_quant    0 non-null      float64
65  addon_13_quant    83 non-null     float64
66  addon_14_quant    88 non-null     float64
67  addon_15_quant    0 non-null      float64
68  addon_16_quant    61 non-null     float64
69  addon_17_quant    0 non-null      float64
70  addon_18_quant    0 non-null      float64
71  addon_19_quant    0 non-null      float64
72  addon_20_quant    0 non-null      float64
73  addon_21_quant    0 non-null      float64
```

```
 74  addon_22_quant    0 non-null      float64
 75  addon_23_quant    0 non-null      float64
 76  addon_24_quant    5 non-null      float64
 77  addon_25_quant    0 non-null      float64
 78  addon_26_quant    0 non-null      float64
 79  addon_27_quant    0 non-null      float64
 80  addon_01_rev      1566 non-null   float64
 81  addon_02_rev      522 non-null    float64
 82  addon_03_rev      0 non-null      float64
 83  addon_04_rev      0 non-null      float64
 84  addon_05_rev      0 non-null      float64
 85  addon_06_rev      0 non-null      float64
 86  addon_07_rev      0 non-null      float64
 87  addon_08_rev      0 non-null      float64
 88  addon_09_rev      0 non-null      float64
 89  addon_10_rev      70 non-null     float64
 90  addon_11_rev      14 non-null     float64
 91  addon_12_rev      0 non-null      float64
 92  addon_13_rev      83 non-null     float64
 93  addon_14_rev      88 non-null     float64
 94  addon_15_rev      0 non-null      float64
 95  addon_16_rev      61 non-null     float64
 96  addon_17_rev      0 non-null      float64
 97  addon_18_rev      0 non-null      float64
 98  addon_19_rev      0 non-null      float64
 99  addon_20_rev      0 non-null      float64
 100 addon_21_rev      0 non-null      float64
 101 addon_22_rev      0 non-null      float64
 102 addon_23_rev      0 non-null      float64
 103 addon_24_rev      5 non-null      float64
 104 addon_25_rev      0 non-null      float64
 105 addon_26_rev      0 non-null      float64
 106 addon_27_rev      0 non-null      float64
dtypes: Int64(8), datetime64[ns](2), float64(92), object(5)
memory usage: 21.4+ MB
None
```

It is easy to notice now that some columns related to questions (those starting with 'q') and addons (those starting with 'addons') only have null values. They refer to questions and addons that are not relevant to the three firms we are currently investigating. Thus, they can be deleted.

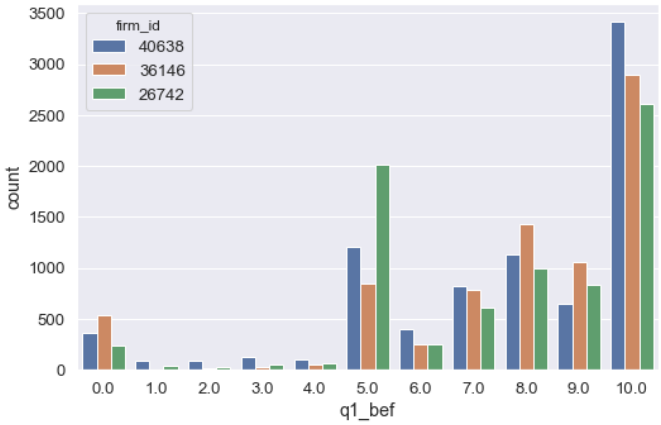
Among the columns related to questions and feedback with data available, the most interesting in terms of amount of non-null entries are 'q1_bef' and 'q2' (see the table below). The column 'q1_aft' is also interesting as a follow-up of column 'q1_bef', despite its high number of null entries.

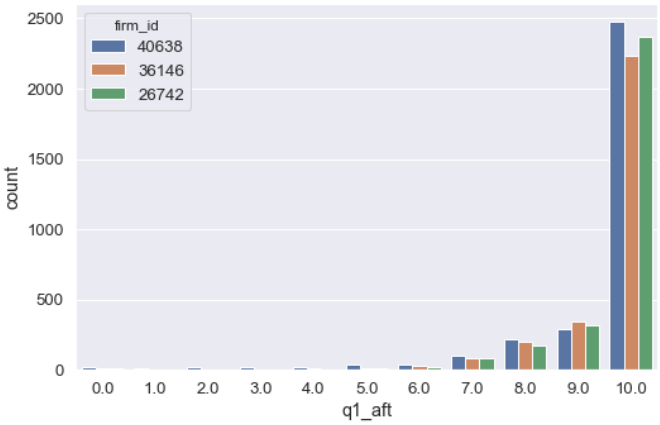
Amount of non-null entries in selected columns for each firm_id.

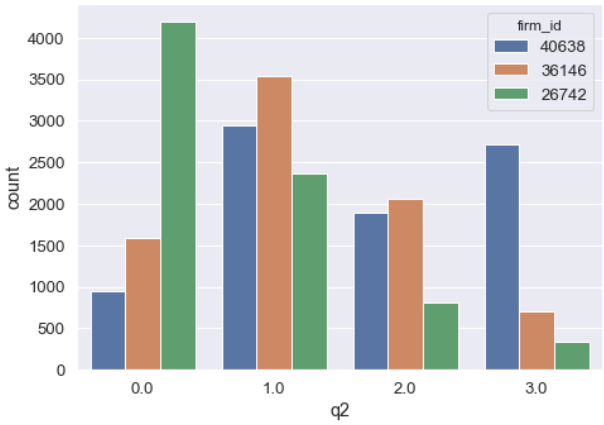
firm_id	firm_id_count	q1_bef	q1_aft	q2	q3	feedback
40638.0	9005	8394	3261	8520	9001	2227
36146.0	8043	7899	2934	7899	0	1679
26742.0	7983	7717	2996	7717	0	385

'q1_bef' is the likelihood to recommend our product before visit on scale 0-10. 'q1_aft' is the likelihood to recommend our product after visit on scale 0-10. 'q2' asks “how often did you use our product in the

past year?", being the possible answers 0 never, 1 occasionally, 2 at least once a month and 3 at least once a week. When comparing each variable between firms, the distributions of 'q1_bef' and 'q1_aft' are not equal but somehow similar whereas the distributions of 'q2' are quite different. In particular, when contrasting 'q1_bef' and 'q1_aft', it is possible to notice that they don't mirror each other, meaning that visits may have an impact on clients.







Between 6.79% and 1.79% of the entries for 'q1_bef' are null depending on the 'firm_id' (in total, 4.08% of the entries for 'q1_bef' are null). Between 63.79% and 62.47% of the entries for 'q1_aft' are null depending on the 'firm_id' (in total, 63.28% of the entries for 'q1_aft' are null). Between 5.39% and 1.79% of the entries for 'q2' are null depending on the 'firm_id' (in total, 3.58% of the entries for 'q2' are null).

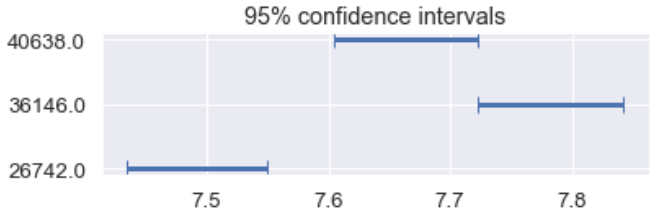
We will take a quick closer look at 'q1_bef' and 'q2' since they have the most amount of non-null entries. We will deal with 'q1_aft' later on given its high number of null entries, which is a problem. In regard to question 'q1_bef', the mean answers for each firm don't come from the same distribution, but they are close to each other between 7 and 8 in a scale between 0 and 10. In regard to question 'q2', the mean answers for each firm don't come from the same distribution and are clearly different from each other taking into account the scale between 0 and 3.

Comparing means of 'q1_bef' between firms.
First comparison of means.
H0: difference in means of 'q1_bef' for different 'firm_id' is by chance.
The meaning of H0 is that the means actually come from the same distribution.

- p-value for H0 in the case of 26742.0 vs. 36146.0 is 0.0: H0 is rejected.
- p-value for H0 in the case of 26742.0 vs. 40638.0 is 0.0: H0 is rejected.
- p-value for H0 in the case of 36146.0 vs. 40638.0 is 0.002: H0 is rejected.

Second comparison of means.
H0: there is no difference in means of 'q1_bef' for different 'firm_id'.

- 95% c. i. if 26742.0 = [7.4342361 7.54930997]
- 95% c. i. if 36146.0 = [7.72261995 7.84238828]
- 95% c. i. if 40638.0 = [7.60519419 7.72206338]

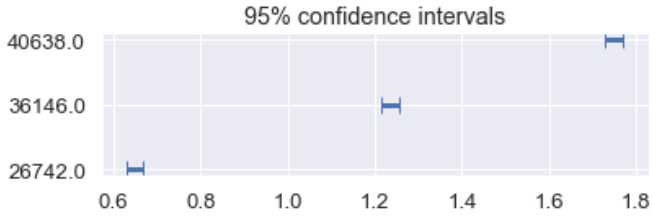


Comparing means of 'q2' between firms.
First comparison of means.
H0: difference in means of 'q2' for different 'firm_id' is by chance.
The meaning of H0 is that the means actually come from the same distribution.

- p-value for H0 in the case of 26742.0 vs. 36146.0 is 0.0: H0 is rejected.
- p-value for H0 in the case of 26742.0 vs. 40638.0 is 0.0: H0 is rejected.
- p-value for H0 in the case of 36146.0 vs. 40638.0 is 0.0: H0 is rejected.

Second comparison of means.
H0: there is no difference in means of 'q2' for different 'firm_id'.

- 95% c. i. if 26742.0 = [0.62964883 0.6672282]
- 95% c. i. if 36146.0 = [1.2163565 1.25471895]
- 95% c. i. if 40638.0 = [1.72769953 1.77089202]



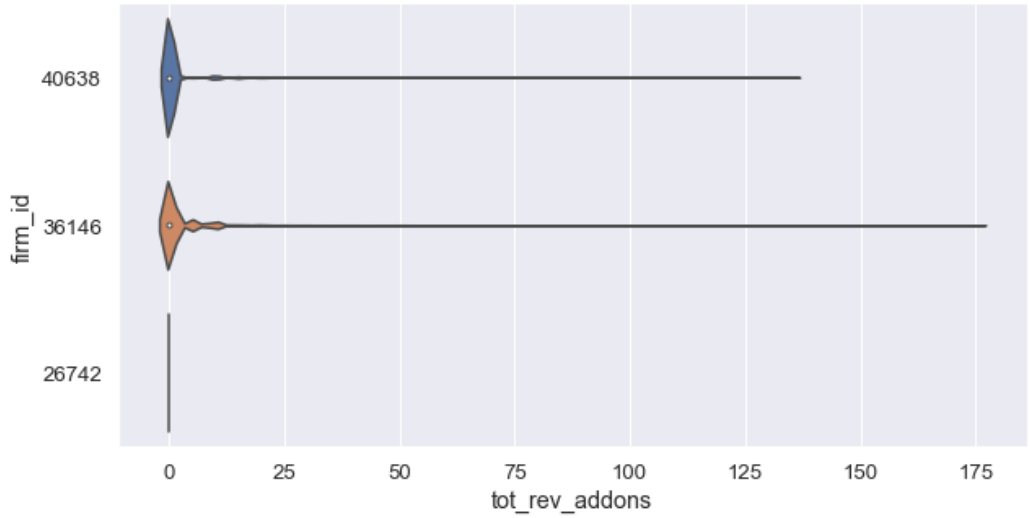
Addons are specific to each 'firm_id'. The table below shows the number of non-null entries for each addon revenue and firm. Addons 02, 10, 11 and 24 are specific to 'firm_id' 40638. Addons 01, 13, 14 and 16 are specific to 'firm_id' 36146. There is no addons for 'firm_id' 26742 and, as a result, 'firm_id' 26742 does not get revenues from addons, meaning that its 'tot_rev_addons' is zero. In regard to the other two 'firm_id', most entries of 'tot_rev_addons' are zero or small values, with the mean not far from zero, but there are few entries with values surpassing $100.

Amount of non-null entries in selected columns for each firm_id.

firm_id	addon_01_rev	addon_02_rev	addon_10_rev	addon_11_rev	addon_13_rev
40638.0	0	522	70	14	0
36146.0	1566	0	0	0	83
26742.0	0	0	0	0	0

firm_id	addon_14_rev	addon_16_rev	addon_24_rev
40638.0	0	0	5
36146.0	88	61	0
26742.0	0	0	0

firm_id and tot_rev_addons.



Regarding the variables 'fee' and 'tot_rev', there is some rare extreme values. In the case of 'tot_rev', those outliers may have a reasonable explanation (e.g., booking for many individuals or the inclusion of many addons). In the case of 'fee', they seem to be mistakes and perhaps they should be excluded. A possible exclusion rule is to ditch rows with fees above the value of the 3rd quartile (i.e., the 75th percentile). That would eliminate less than 4% of the entries for each 'firm_id' (less than 2.5% of all entries for fee in total).

firm_id 40638		
	fee	tot_rev
count	9005.000000	9005.000000
mean	23.583554	57.952027
std	48.003607	75.440967
min	0.000000	0.000000
25%	8.000000	20.000000
50%	10.000000	35.000000
75%	35.000000	70.000000
max	350.000000	2100.000000

341 entries for fee above the 75th percentile.

3.79% of the entries for fee above the 75th percentile.

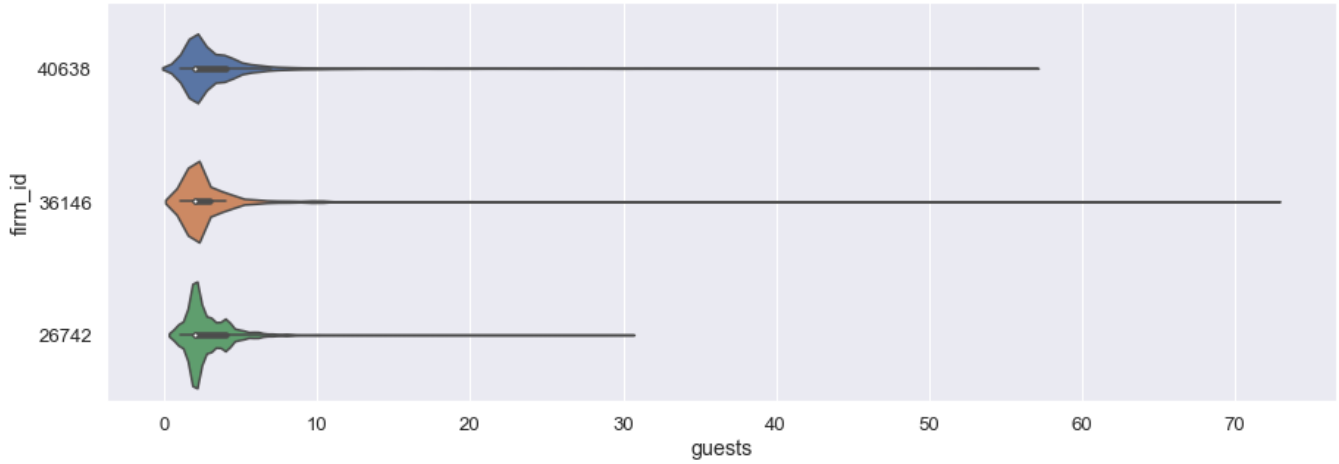
firm_id 36146 fee tot_rev count 8043.000000 8043.000000 mean 24.150111 72.176302 std 15.921995 109.017858 min 0.000000 0.000000 25% 25.000000 40.000000 50% 25.000000 50.000000 75% 25.000000 75.000000 max 1200.000000 3600.000000	60 entries for fee above the 75th percentile. 0.75% of the entries for fee above the 75th percentile.

firm_id 26742 fee tot_rev count 7983.000000 7983.000000 mean 11.383569 33.939622 std 29.092336 93.538286 min 0.000000 0.000000 25% 10.000000 20.000000 50% 10.000000 20.000000 75% 10.000000 40.000000 max 2500.000000 4000.000000	204 entries for fee above the 75th percentile. 2.56% of the entries for fee above the 75th percentile.

In total
605 entries for fee above the 75th percentile.
2.42% of the entries for fee above the 75th percentile.

With respect to the columns 'status_cancel' and 'payment_cc', the data available for them practically doesn't vary and, as a result, they will not be considered in further analysis. We will keep the columns 'source_online' and 'discount', although their data doesn't vary that much either. The column 'guests' is certainly more interesting in terms of data variation. The mean values of 'guests' for each firm don't come from the same distribution, but they are close to each other. Firms 26742 and 36146 have mean values of 'guests' between 2 and 3 (they may actually have the same mean value), while firm 40638 has mean value of 'guests' above 3, but probably not above 3.5.

firm_id and guests.

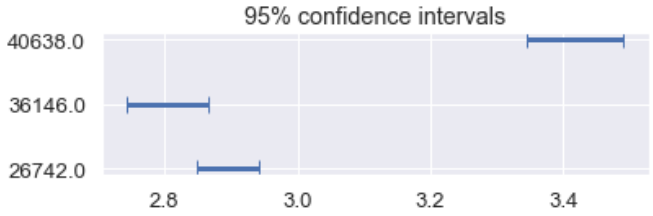


Comparing means of 'guests' between firms.
First comparison of means.
H0: difference in means of 'guests' for different 'firm_id' is by chance.
The meaning of H0 is that the means actually come from the same distribution.

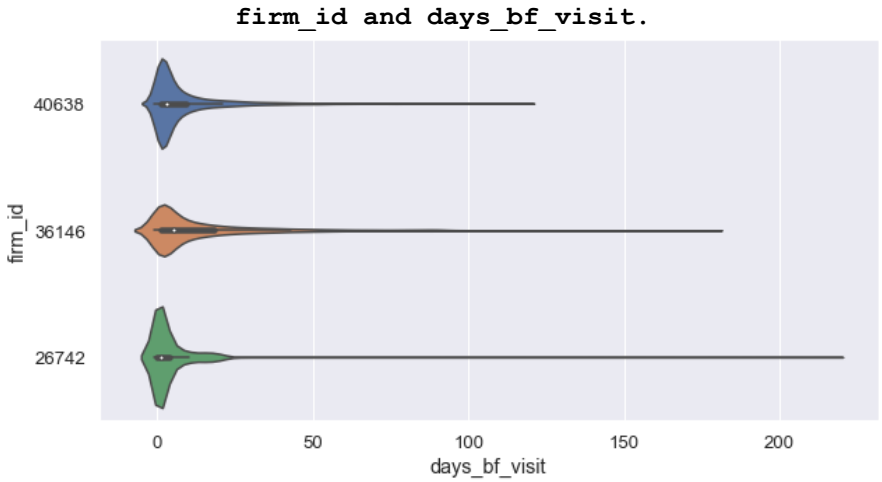
- p-value for H0 in the case of 26742.0 vs. 36146.0 is 0.01: H0 is rejected.
- p-value for H0 in the case of 26742.0 vs. 40638.0 is 0.0: H0 is rejected.
- p-value for H0 in the case of 36146.0 vs. 40638.0 is 0.0: H0 is rejected.

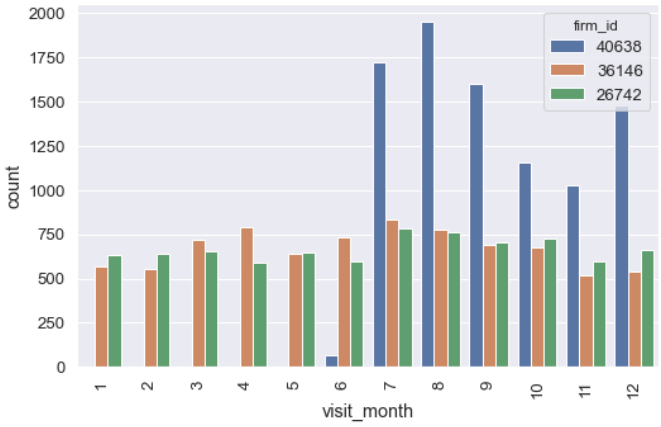
Second comparison of means.
H0: there is no difference in means of 'guests' for different 'firm_id'.

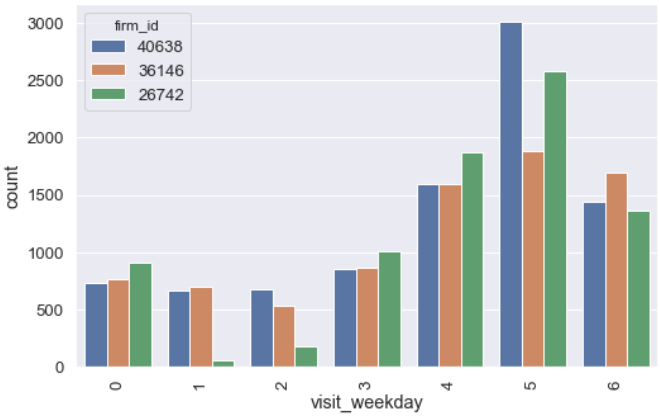
- 95% c. i. if 26742.0 = [2.84842791 2.94137542]
- 95% c. i. if 36146.0 = [2.74250901 2.86622218]
- 95% c. i. if 40638.0 = [3.34491671 3.49128262]



Both variables 'contact_date' and 'visit_datetime' are datetime type. As a result, we can create the variable 'days_bf_visit' to count the days between the contact day and the visit day. We can also analyze the frequence of visits among the 12 months of a year and the 7 days of a week. All entries for visit day are related to the year 2017. The distribution of 'days_bf_visit' are somewhat similar. More about that later. For firms 36146 and 26742, visits are not spread so unevenly among months, although summer months are noticeably busier. For firm 40638, there is no visits in the five first months. That's because contacts booking visits to firm 40638 also started to show up in the dataset only later, as we are going to see further ahead. Firm 40638 probably became a partner of the booking company only later in 2017. As expected, the volume of visits increases during the end of the week ('visit_weekday' 0 means Monday and 6 means Sunday).







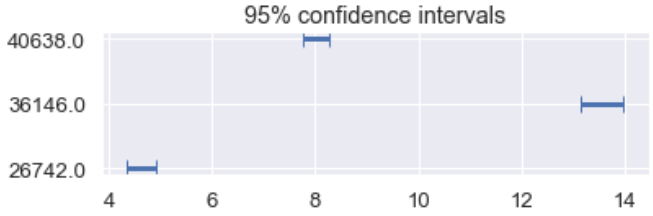
Despite some visual similarity between the distributions of 'days_bf_visit' for each firm, the statistical tests below show that 1-) the means for each firm don't come from the same distribution, and 2-) the means are different from each other. The mean number of days from contact to visit for firm 40638 is probably between 7 and 8, for firm 36146 is about 13, and for firm 26742 is about 4.

Comparing means of 'days_bf_visit' between firms.
First comparison of means.
H0: difference in means of 'days_bf_visit' for different 'firm_id' is by chance.
The meaning of H0 is that the means actually come from the same distribution.

- p-value for H0 in the case of 26742.0 vs. 36146.0 is 0.0: H0 is rejected.
- p-value for H0 in the case of 26742.0 vs. 40638.0 is 0.0: H0 is rejected.
- p-value for H0 in the case of 36146.0 vs. 40638.0 is 0.0: H0 is rejected.

Second comparison of means.
H0: there is no difference in means of 'days_bf_visit' for different 'firm_id'.

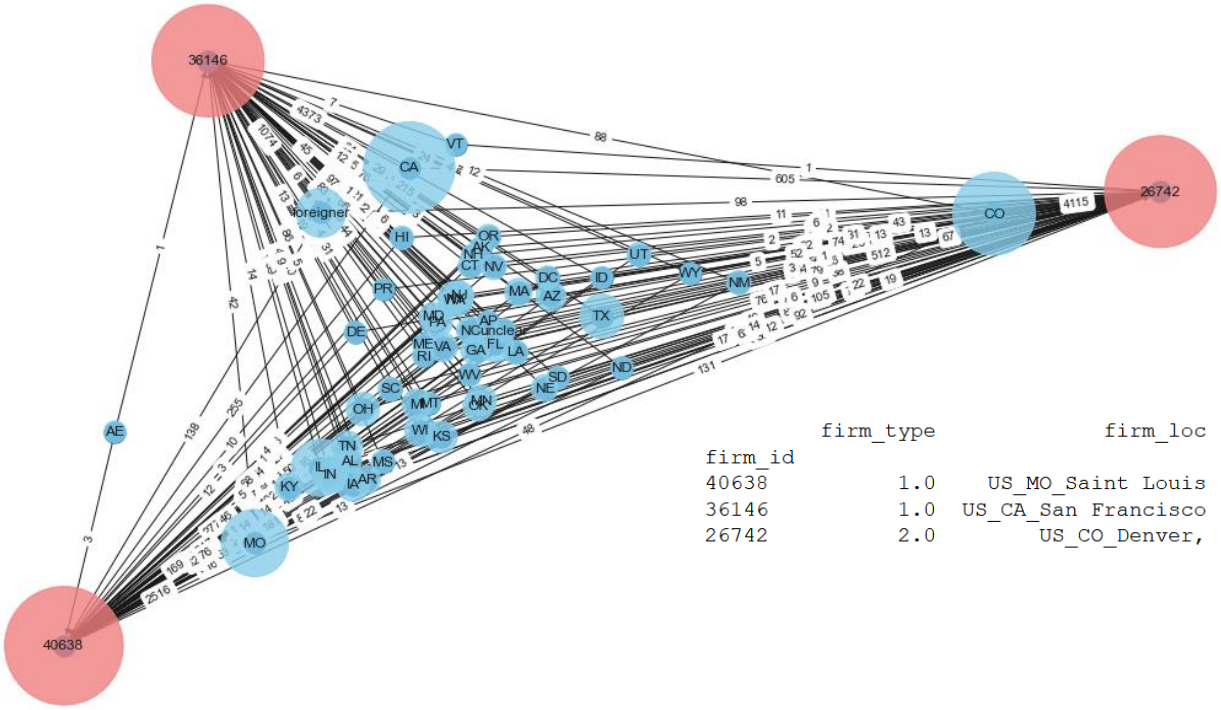
- 95% c. i. if 26742.0 = [4.3433515 4.90367343]
- 95% c. i. if 36146.0 = [13.14333893 13.98048303]
- 95% c. i. if 40638.0 = [7.75346197 8.25930594]



3.2-) Network analysis.

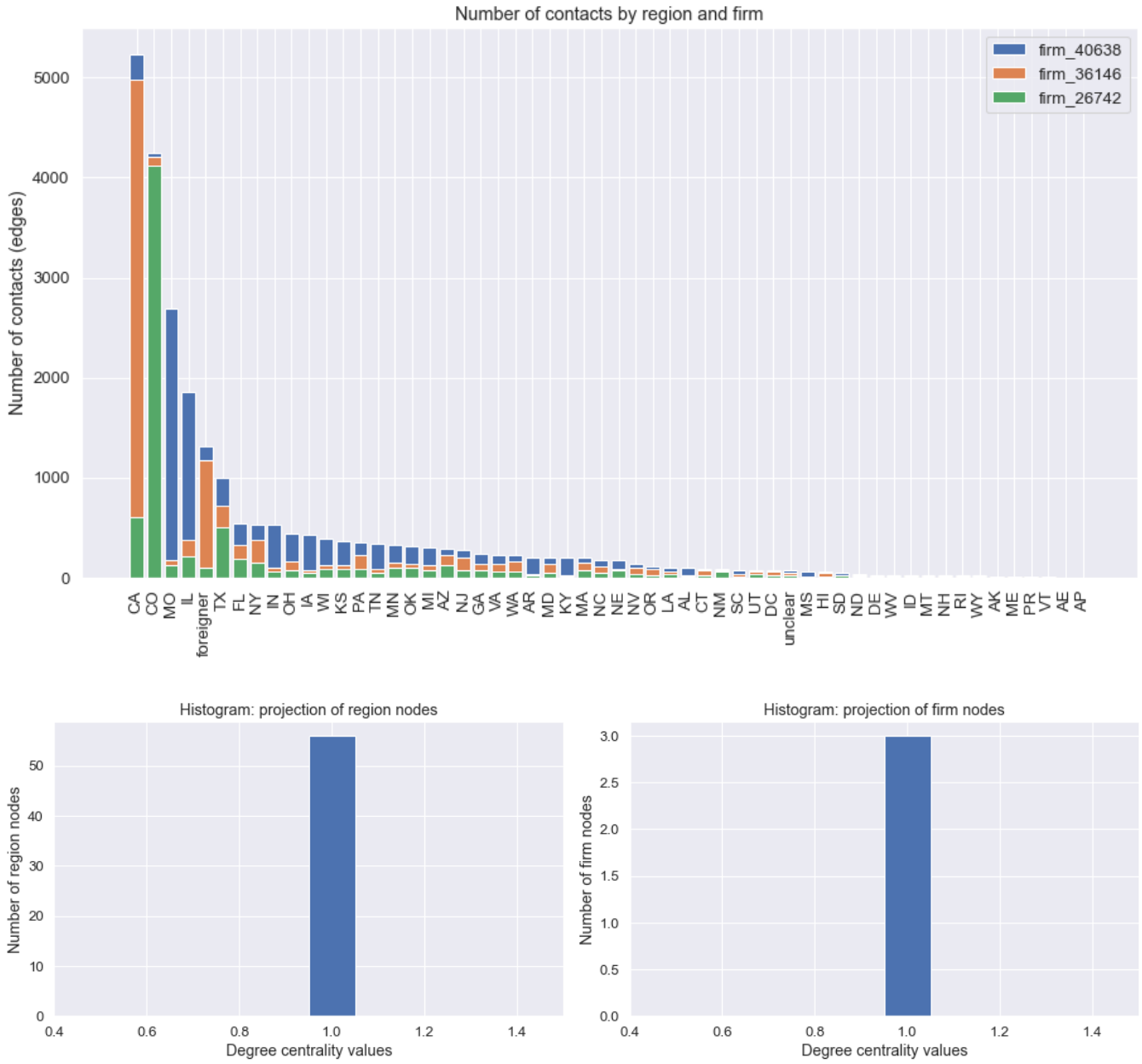
It is possible to use network analysis to see how different firms attract different clients from different regions (US states or foreign countries). The directed network below shows regions in blue connecting to firms in red. It uses booking data from 2016-08-31 to 2017-12-31 ('contact_date'). The size of each node is proportional to its degree (number of edges connecting it). The number on each edge informs its weight in terms of number of connections that edge actually represents. Technically speaking, we have a

Weighted DiGraph representing a MultiDiGraph. Higher weights at the edges pull the regions closer to the firms as if firms more connected to a certain region exert more gravitational force on that region. It is a nice way to visualize competing firms in a market, although a more complete description would require the inclusion of all competing firms in that market, not just the three we are analysing. The resulting layout shows that locality is an important attractiveness factor. Firm 40638 from MO attracts a comparatively larger number of clients from MO. Between the three firms, 40638 appears to have the greatest appeal among American clients from different states. On the other hand, firm 36146 appears to have the greatest appeal among foreigner clients. Being from CA, 36146 also attracts a comparatively larger number of clients from that state. Finally, firm 26742 has the greatest appeal among clients in its home state CO.

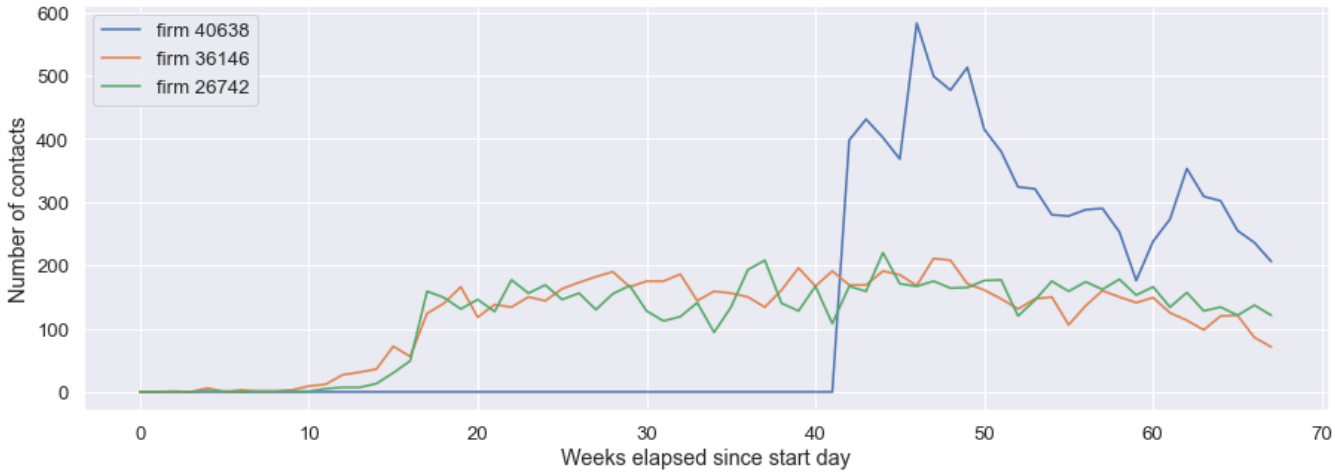


firm_id	firm_type	firm_loc
40638	1.0	US_MO_Saint Louis
36146	1.0	US_CA_San Francisco
26742	2.0	US_CO_Denver,

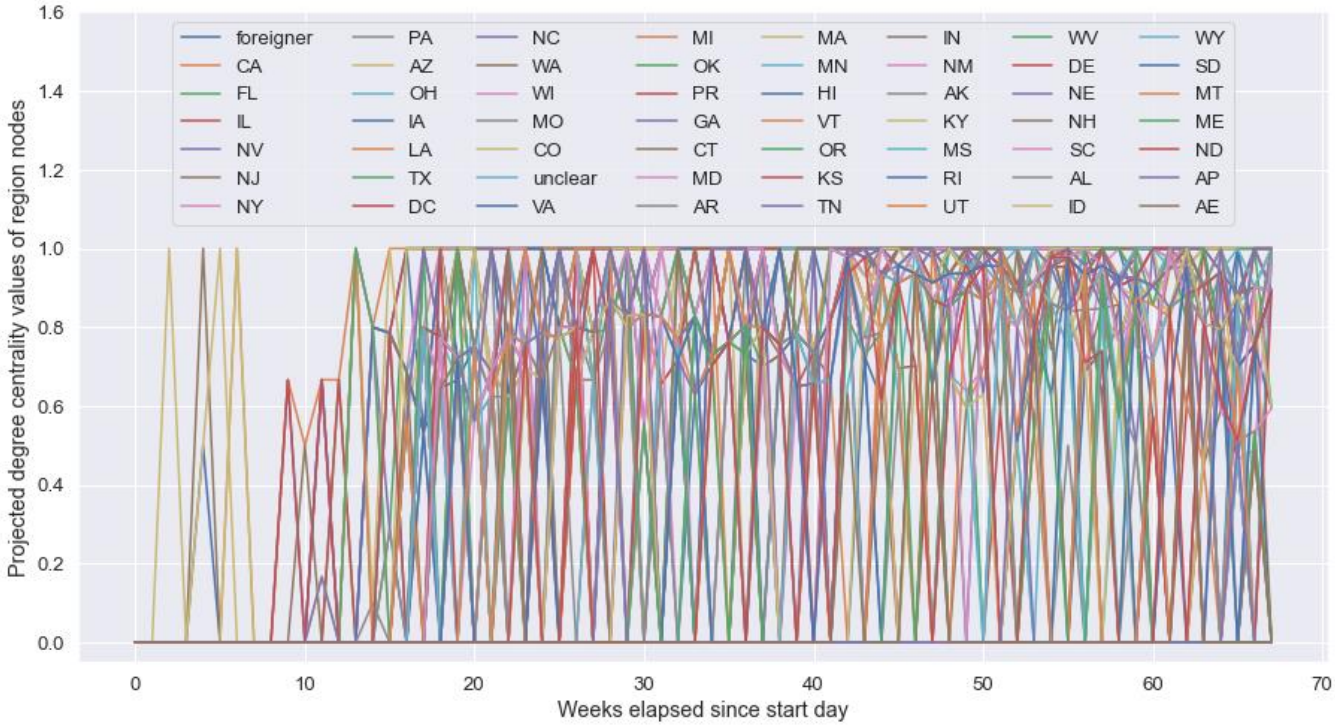
The stacked barplot below is another way to show the number of contacts by region and firm. All the information was extracted from the network above. It basically confirms the observations made before: most visitors to a firm are from the firm's home state, firm 40638 has the greatest appeal among American clients from different states (the blue color is overall more present among the bars) and firm 36146 has the greatest appeal among foreigner clients. We also present histograms for degree centrality values of unipartite unweighted projections of our bipartite network above. One histogram is for region nodes and the other is for firm nodes. Because of the nature of our network and the large period of time represented, those histograms don't inform much. Given enough time, every region will develop connections to enough firms so as to be extended connected to each other in their projected network, and every firm will be connected to enough regions so as to be extended connected to each other in their projected network. That explain why the histograms have only one bar at degree centrality 1.



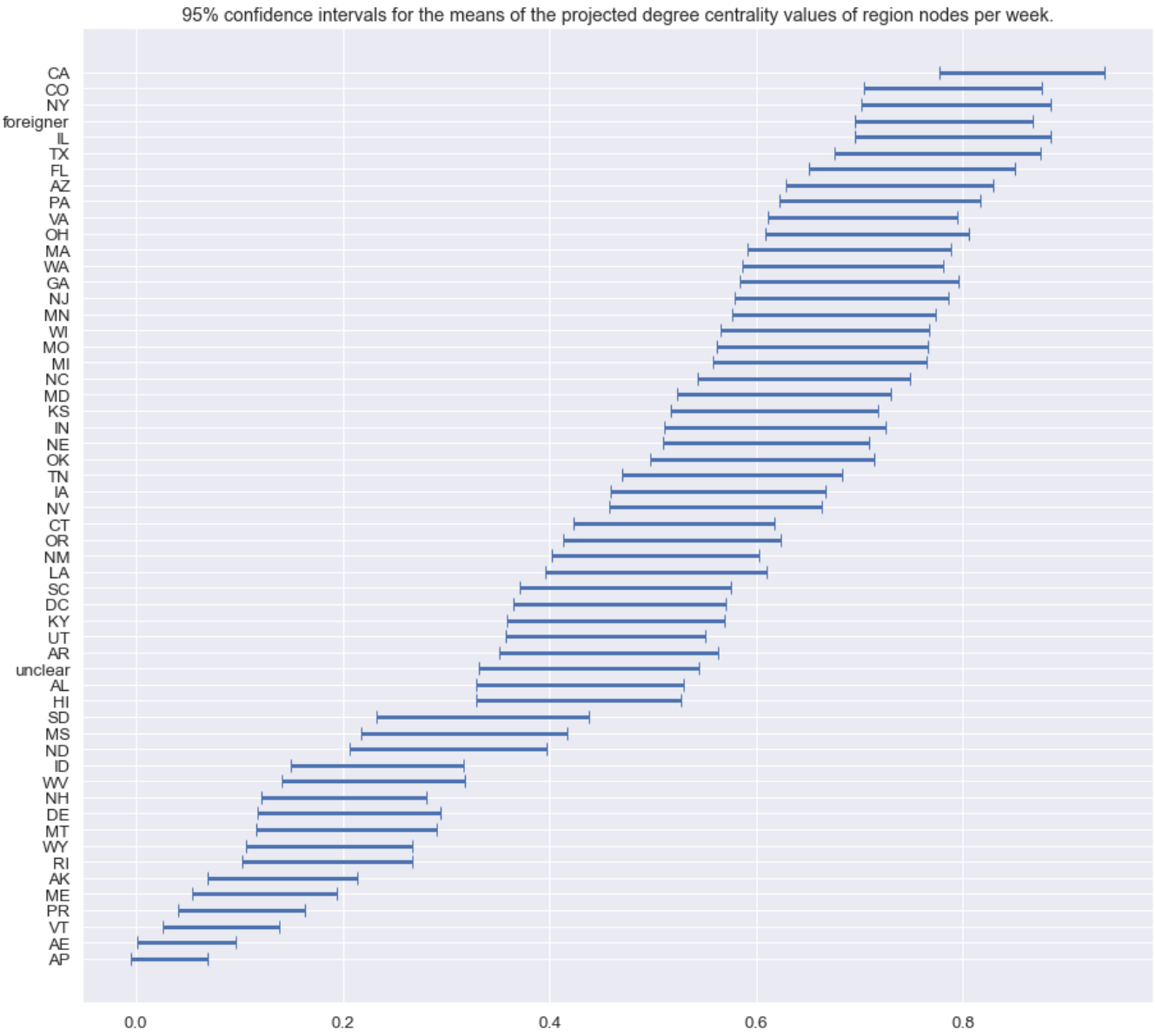
Next, the graph below shows the number of contacts being made as the time pass in weeks. The contacts come from the column 'contact_date' in our dataframe. They are clients entering in contact with the booking company to schedule visits to firms. The start date is Monday 2016-09-05 and the end date is Sunday 2017-12-24. This graph clarifies why firm 40638 had no visits in the first five months of 2017, as we saw before. Up to 40 weeks from the start date, there was no contacts scheduling visits for that firm. As we said earlier, firm 40638 probably became a partner of the booking company only later in 2017. However, once the contacts started for that firm, the volume was high. The other two firms present a similar evolution of contacts along time in relation to each other, depite being of different types (36146 is type 1.0 whereas 26742 is type 2.0).



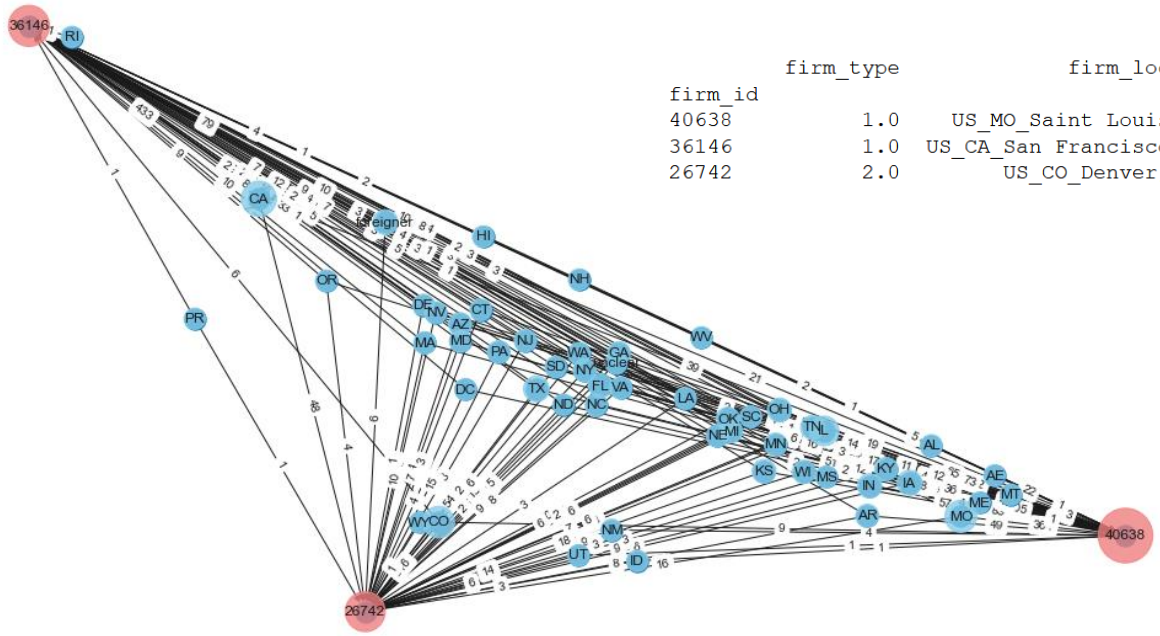
The following graph shows the projected degree centrality values of region nodes per week. In a short period of time like a week, it is much more likely that the network formed will contain region nodes with projected degree centrality value lower than 1. The motivation behind the second graph is to see if we can build a meaningful numerical scale to represent the categorical variable 'us_region' based on connectivity. By meaningful we mean a numerical scale where distances between points actually mean something. A meaningful numerical scale would avoid the use of many dummies to represent regions and would incorporate information about the connections among them that could be used in other quantitative analyses.



Finally, the following graph is derived from the previous above and shows for each region the 95% confidence interval (95% c. i.) for the mean of the weekly projected degree centrality values. Given a long period of weeks, it is possible to observe which regions are statistically similar or distinct in terms of projected conectivity among them. Most importantly, we now have the regions being represented on a connectivity scale between 0 and 1. We can use the minimum, maximum and middle of these confidence intervals in other quantitative analyses, assuming that those intervals will remain stable as time (weeks) continous to pass. We have added that information to our dataframe under the columns 'region_dc_95min' (minimum value of the 95% c. i.), 'region_dc_95max' (maximum value of the 95% c. i.), 'region_dc_mean' (middle value of the 95% c. i., which is the mean of the weekly projected degree centrality values) and 'region_dc_half_size' (half the size of the 95% c. i.).

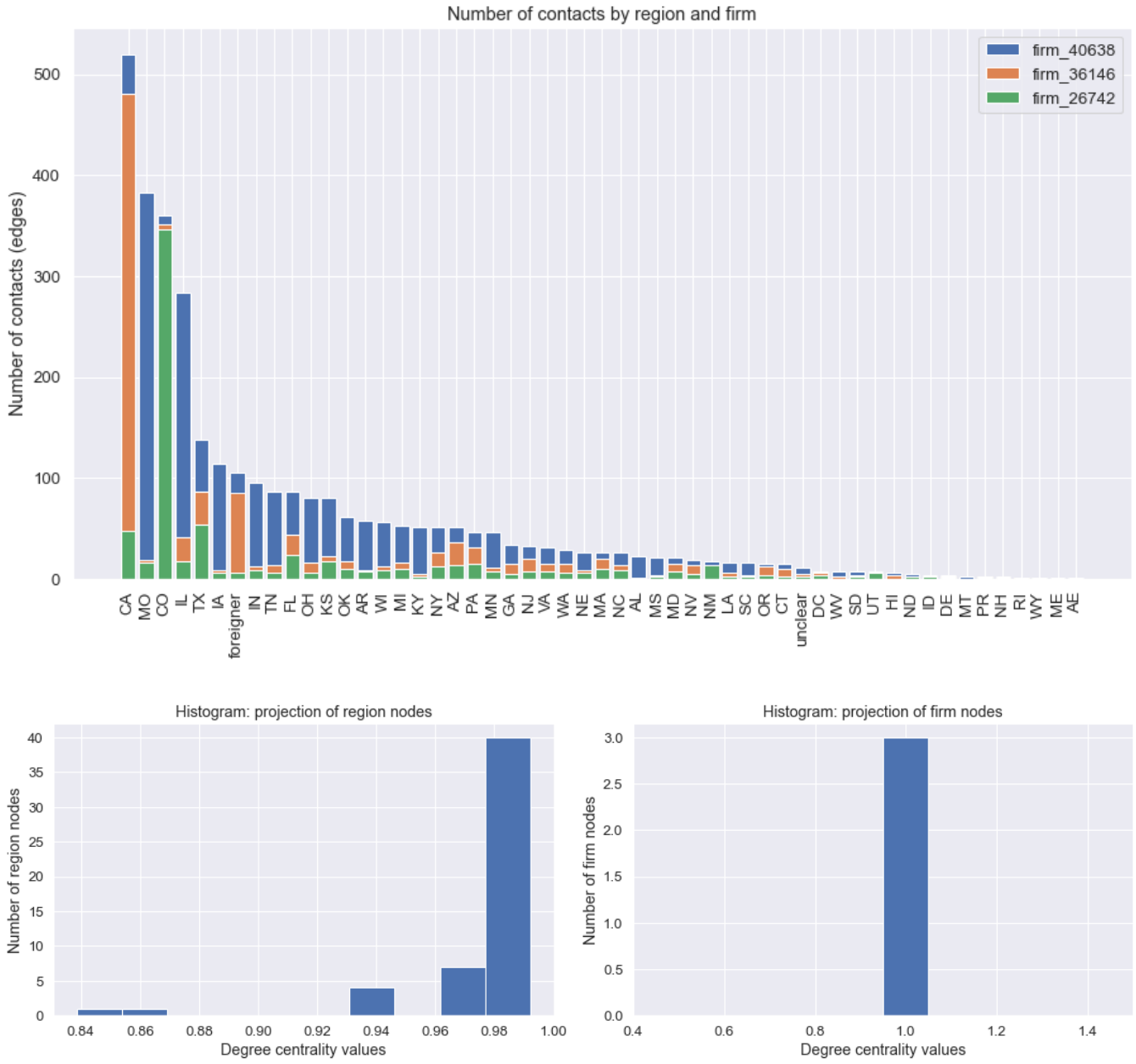


As we noticed earlier, summer months are noticeably busier. We took a quick look in the network during the month of July. Below we can see the weighted directed network filtered for date of visit. We only added entries with date of visit between 2017-07-01 and 2017-07-31. The conclusions are basically the same.



firm_id	firm_type	firm_loc
40638	1.0	US_MO_Saint Louis
36146	1.0	US_CA_San Francisco
26742	2.0	US_CO_Denver,

We again present the stacked barplot for the number of contacts by region and firm, but now considering only the month of July 2017. Nothing particularly new here. The month of July seems to be a good proxy for what happens during a whole year of visits. A highligth would be the clearer overall dominance of the blue color among the bars, reinforcing the appeal of firm 40638 among American clients from different states (taking into comparison only the three firms we are analysing). As for the histograms for degree centrality values of unipartite unweighted projections, because the period of time is shorter, we can see some variance between region nodes in their projected network, meaning that in July some regions didn't develop connections to enough firms so as to be extended connected to all other regions.



3.2-) Final trimming, supervised classification and unsupervised clustering.

Now that we understand better our dataset, let's trim it a little more and proceed with further analysis:

- Columns 'q3' and 'feedback' will be ditched. Among the columns related to questions and feedback, only 'q1_bef' and 'q2' have a good amount of non-null entries. The column 'q1_aft' will be kept in spite of its high number of null entries because it is an interesting follow-up of column 'q1_bef'.

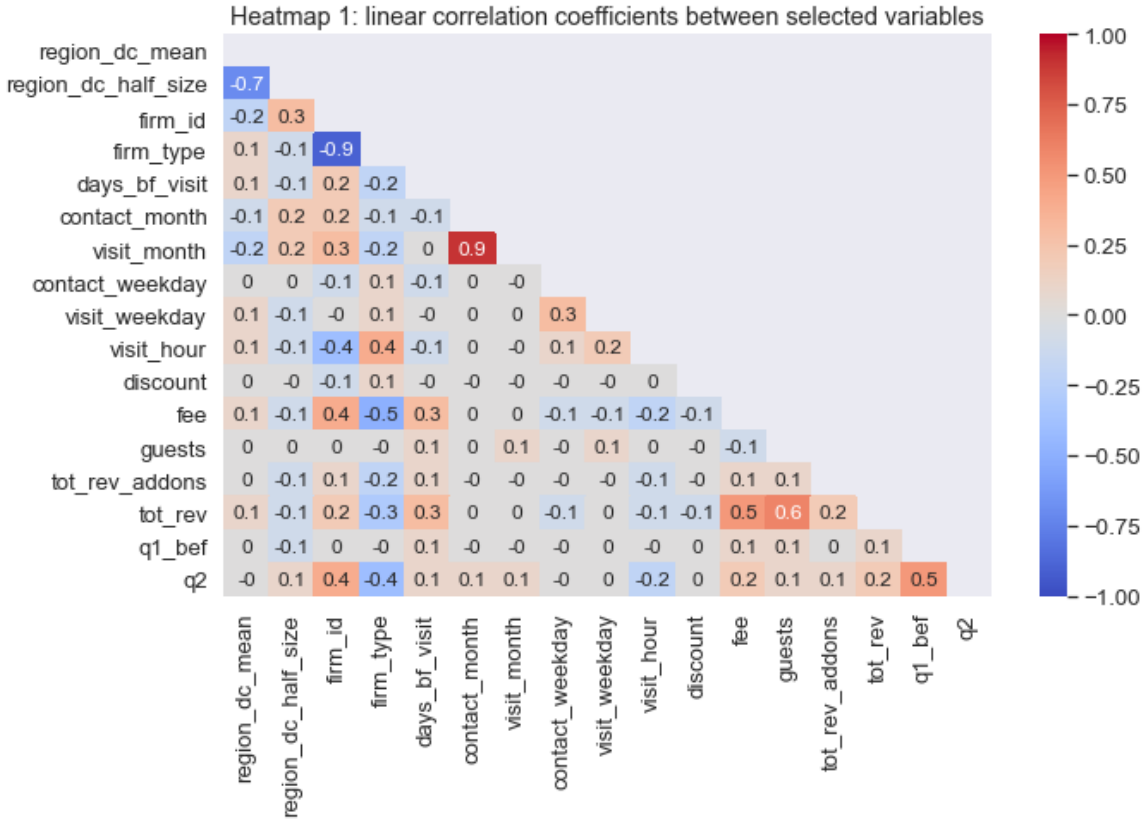
- Columns related to addons will be ditched. They are specific to each firm. Addons 02, 10, 11 and 24 are specific to firm 40638. Addons 01, 13, 14 and 16 are specific to firm 36146. There is no addons for firm 26742. In each case, the amount of non-null entries is small compared to the size of the sample. On the other hand, the columns 'tot_rev_addons' and 'tot_rev' will be kept, the latter being 'fee' · 'guests' + 'tot_rev_addons'.
- Column 'tot_quant_addons' will be ditched. It does not make sense to sum quantities of addons from different varieties or qualities.
- The columns 'status_cancel' and 'payment_cc' will be ditched. The data available for them practically doesn't vary. On the other hand, the columns 'source_online' and 'guests' will be kept.
- Rows where column 'fee' have values above the threshold of the 3rd quartile (i.e., the 75th percentile) will be ditched just to play safe. They represent less than 4% of the entries for each 'firm_id' (less than 2.5% of all entries in total).
- Rows where the columns 'q1_bef' and 'q2' are null will be ditched. In regard to 'q1_bef', they represent between 6.79% and 1.79% of the entries depending on the 'firm_id' (4.08% of all entries in total). In regard to 'q2', they represent between 5.39% and 1.79% of the entries depending on the 'firm_id' (3.58% of all entries in total).

In summary, 29 columns and 1421 rows will be lost (regarding the rows, they represent 5.68% of the total rows in the dataframe before the trimming process). Some columns were also removed after the trimming process due to lack of variation (namely 'visit_year' and 'source_online'). The resulting dataframe contains 29 columns and 23610 entries. It is leaner, easier to read and has no null entries except in 'q1_aft', which is a problem that we will try to overcome later.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 23610 entries, 132755 to 181698
Data columns (total 29 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   country              23610 non-null  object
 1   us_region            23610 non-null  object
 2   region_dc_95min      23610 non-null  float64
 3   region_dc_95max      23610 non-null  float64
 4   region_dc_mean       23610 non-null  float64
 5   region_dc_half_size  23610 non-null  float64
 6   us_zipcode           23610 non-null  object
 7   firm_id              23610 non-null  float64
 8   firm_type            23610 non-null  float64
 9   firm_loc             23610 non-null  object
 10  contact_date         23610 non-null  datetime64[ns]
 11  contact_year         23610 non-null  int64
 12  contact_month        23610 non-null  int64
 13  contact_day          23610 non-null  int64
 14  contact_weekday      23610 non-null  int64
 15  days_bf_visit        23610 non-null  int64
 16  visit_datetime       23610 non-null  datetime64[ns]
 17  visit_month          23610 non-null  int64
 18  visit_day            23610 non-null  int64
 19  visit_weekday        23610 non-null  int64
 20  visit_hour           23610 non-null  int64
 21  guests               23610 non-null  float64
 22  fee                  23610 non-null  float64
```

```
 23  discount              23610 non-null  float64
 24  tot_rev_addons        23610 non-null  float64
 25  tot_rev               23610 non-null  float64
 26  q1_bef                23610 non-null  Int64
 27  q1_aft                8759 non-null   Int64
 28  q2                    23610 non-null  Int64
dtypes: Int64(3), datetime64[ns](2), float64(11), int64(9), object(4)
memory usage: 5.5+ MB
None
```

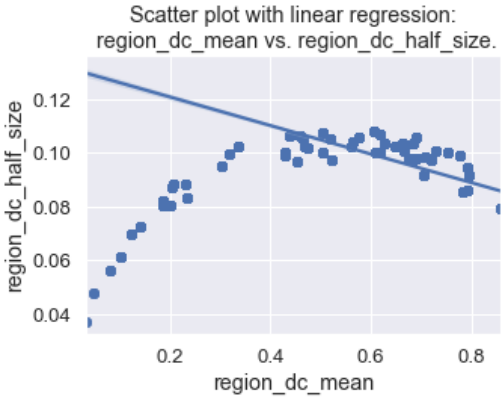
As we can see, 'q1_aft' is the only column with null entries. In fact, it contains 14,851 null entries (about 63% of the entries). However, there is enough non-null entries to allow for "guessing" the null entries by using supervised classification. Basically, we want to predict the target variable 'q1_aft' using a set of attributes from the dataframe df. We can start selecting those attributes by analyzing the linear correlation coefficients between some canditates, and that analysis can take the form of a heatmap, keeping in mind that linear correlation coefficients can go from 1 (perfect positive linear correlation) to 0 (no linear correlation) and then to -1 (perfect negative linear correlation).



According to the first heatmap above:

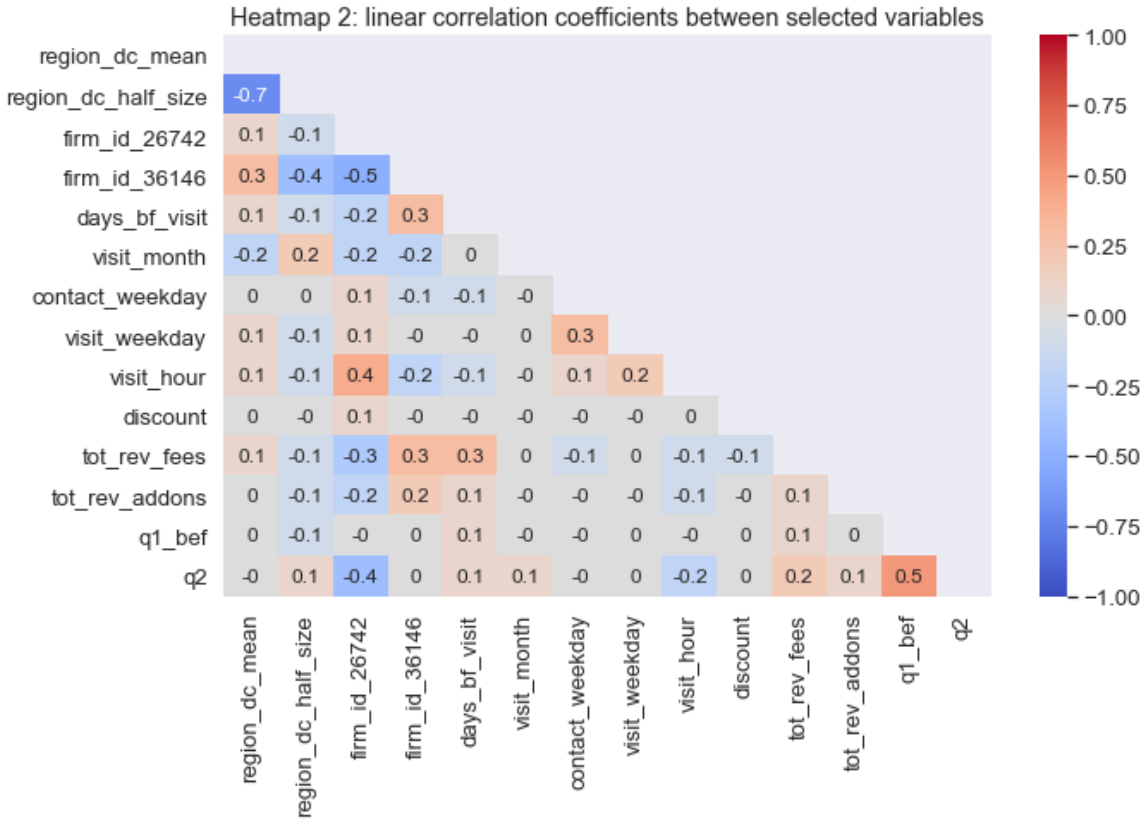
- Many variables don't seem to have strong linear correlation between them, which is good in the sense of avoiding problems with multicollinearity.

- 'region_dc_mean' and 'region_dc_half_size' would have a strong negative linear correlation, but a non-linear correlation would better fit the data and tell a more complete and slightly different story, as illustrated by their scatter plot (on the right). It goes to show that linear correlation coefficients must be taken with a "grain of salt". Both variables are important because they report different things: one reports average value while the other reports uncertainty.



- 'firm_id' and 'firm_type' seem to be strongly correlated. In order to avoid problems with multicollinearity, it makes more sense to substitute them for dummies identifying each firm.
- 'contact_month' and 'visit_month' also seem to be strongly correlated. That is probably because there is a certain pattern in the number of days between contact and visit. In order to avoid problems with multicollinearity, it is better to keep just one of them (for example, 'visit_month') together with 'days_bf_visit'.
- 'fee', 'guests', 'tot_rev_addons' and 'tot_rev' show some degree of correlation. That shouldn't be a surprise given that 'tot_rev' is 'fee' * 'guests' + 'tot_rev_addons'. In order to avoid problems with multicollinearity and simplify the analysis, it may be better to create 'tot_rev_fees' = 'fee' * 'guests' and keep it together with 'tot_rev_addons', not including 'tot_rev' as attribute.

After the changes, the result is the second heatmap below.

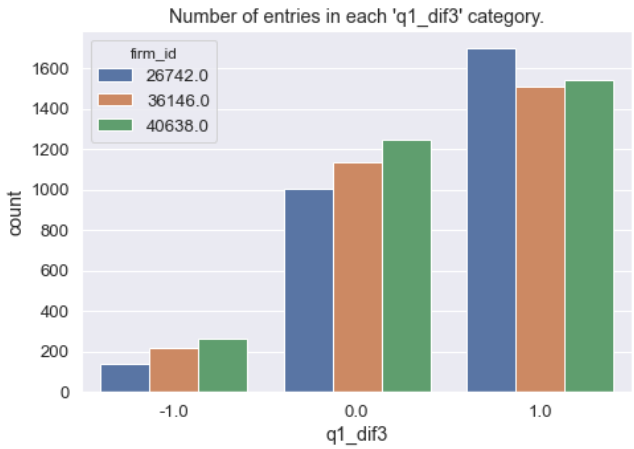


Only three correlations present coefficient around or above 0.5 in absolute value: 'region_dc_mean' vs 'region_dc_half_size', which we have already discussed; 'firm_id_26742' vs 'firm_id_36146', which is understandable (when one dummy is 1, the other is necessarily 0, which results in correlation); and 'q1_bef' vs 'q2', suggesting that people who uses more the product is also more likely to recommend it. Overall, the coefficients are not so high and multicolinearity shouldn't be a problem. The variables in the second heatmap seem to be good candidates to attributes.

Ideally, we would like to have attributes that are not strongly correlated to each other, but are strongly correlated with the target varible, which in our case would be 'q1_aft'. Unfortunately, when we analyze the linear correlation coefficients between 'q1_aft' and the selected attributes (the ones represented in the second heatmap above), the results are not great (see below): the highest coefficient, the one for 'q1_aft' vs 'q1_bef', is only 0.16 .

```
Linear correlation coefficients in relation to q1_aft
-0.01 between q1_aft and region_dc_mean
-0.01 between q1_aft and region_dc_half_size
+0.05 between q1_aft and firm_id_26742
+0.02 between q1_aft and firm_id_36146
+0.01 between q1_aft and days_bf_visit
-0.08 between q1_aft and visit_month
-0.01 between q1_aft and contact_weekday
-0.04 between q1_aft and visit_weekday
-0.07 between q1_aft and visit_hour
+0.02 between q1_aft and discount
+0.02 between q1_aft and tot_rev_fees
-0.01 between q1_aft and tot_rev_addons
+0.16 between q1_aft and q1_bef
+0.04 between q1_aft and q2
```

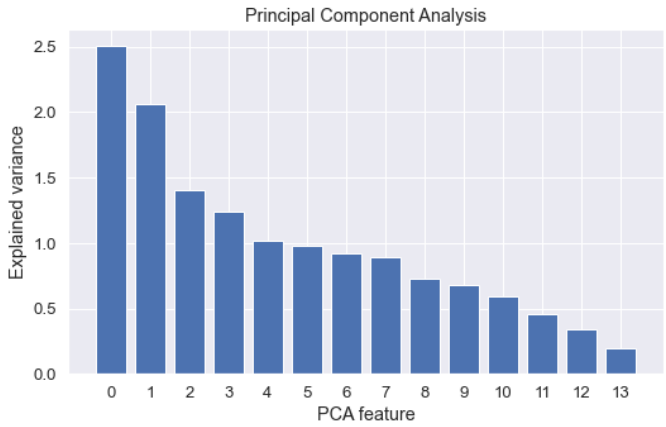
In order to improve that, we decide to use a target variable still connected to 'q1_aft' but expressing a reduced number of categories. The original plan would include 'q1_aft' directly. However, by doing so, we would have to deal with 11 categories: scores from 0 to 10. Instead, we decided to take the difference between 'q1_aft' and 'q1_bef' by making 'q1_dif' = 'q1_aft' - 'q1_bef'. We then created a new variable called 'q1_dif3' which associates the value -1 for 'q1_dif' < 0, 0 for 'q1_dif' = 0, and 1 for 'q1_dif' > 0. In short, the new target variable 'q1_dif3' has only three categories: -1 if the visit had a negative impact on the client, 0 if the visit was neutral, and 1 if the visit had a positive impact on the client.

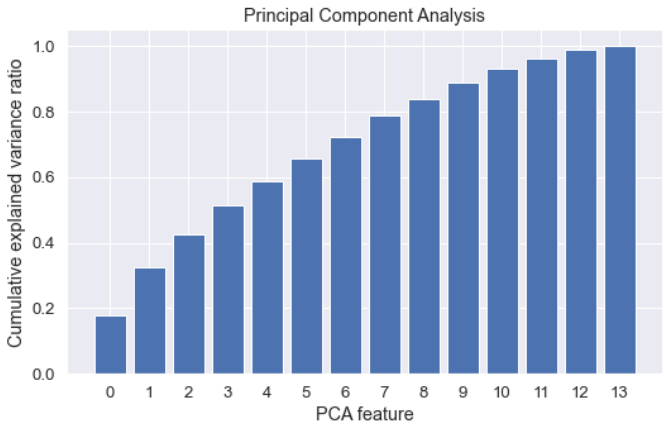


When we analyze the linear correlation coefficients between 'q1_dif3' and the selected attributes, we can see that the highest coefficient is now 0.6 for q1_dif3 vs q1_bef, and the second highest is 0.35 in absolute value for q1_dif3 vs q2. It is important to remember that the analysis below is just a first approach that provides some clues but doesn't tell the whole story because linear correlation coefficients don't capture possible non-linear correlations. That's why we don't simply discharge attributes with low linear correlation coefficients.

```
Linear correlation coefficients in relation to q1_dif3
-0.02 between q1_dif3 and region_dc_mean
+0.01 between q1_dif3 and region_dc_half_size
+0.09 between q1_dif3 and firm_id_26742
-0.02 between q1_dif3 and firm_id_36146
-0.07 between q1_dif3 and days_bf_visit
-0.04 between q1_dif3 and visit_month
+0.02 between q1_dif3 and contact_weekday
-0.02 between q1_dif3 and visit_weekday
-0.02 between q1_dif3 and visit_hour
-0.00 between q1_dif3 and discount
-0.08 between q1_dif3 and tot_rev_fees
-0.03 between q1_dif3 and tot_rev_addons
-0.60 between q1_dif3 and q1_bef
-0.35 between q1_dif3 and q2
```

Once defined the target variable as 'q1_dif3' and the list of attributes as the one illustrated in Heatmap 2, we can proceed with the objective of using supervised classification to predict the 14,851 null entries of the target variable. Before starting to implement supervised classification, we will make a quick Principal Component Analysis (PCA) of the attributes to check the potential for dimensionality reduction. The graphs below representing explained variance and cumulative explained variance ratio vis-à-vis PCA features does not show a clear intrinsic dimension for our list of attributes. Four dimensions can explain 51% while nine dimensions can explain 84% of the variance. We will leave the decision regarding dimensionality for the optimization algorithm (coming next).





There are different possible methods to implement supervised classification: k-nearest neighbors, logistic regression, decision tree and support vector machine. For the sake of simplicity, and because the target variable has more than two categories, we decided to use k-nearest neighbors. Data is

preprocessed using StandardScaler to standardize features and PCA to address any remaining multicollinearity and perhaps reduce dimensionality (depending on the optimization algorithm).

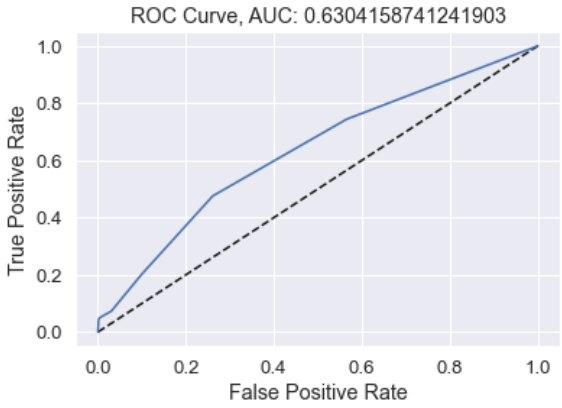
The optimized k-NN classifier uses 13 pca components and 18 neighbors. It seems to be good predicting 'q1_dif3' equal to 1 or 0, but it shows deficiency predicting 'q1_dif3' equal to -1. For that category, precision is 0.67 but recall is only 0.04. In dealing with visits causing negative impact, the classifier is decent in the sense that it does not produce many false positives, but it fails in sense that it does produce many false negatives. In other words, when it predicts a negative impact, it is correct 67% of the time, but it correctly identifies only 4% of all negative impacts. The new variable 'q1_dif3_pr' contains the predicted values for the missing data in 'q1_dif3'.

```
Target variable: q1_dif3
Optimizing k-NN: k-nearest neighbors.
Tuned Model Parameters: {'knn__n_neighbors': 18, 'pca__n_components': 13}

Confusion matrix:
[[  6 109  49]
 [  3 728 112]
 [  0 200 983]]

Accuracy: 0.7840182648401827
Classification report:
              precision    recall  f1-score   support
        -1.0       0.67      0.04      0.07       164
         0.0       0.70      0.86      0.77       843
         1.0       0.86      0.83      0.84      1183
    accuracy                           0.78      2190
   macro avg       0.74      0.58      0.56      2190
weighted avg       0.78      0.78      0.76      2190

Target variable with the inclusion of predicted values:
count    23610.000000
mean         0.526175
std          0.555540
min         -1.000000
25%          0.000000
50%          1.000000
75%          1.000000
max          1.000000
Name: q1_dif3_pr, dtype: float64
```

Because the above k-NN classifier is deficient predicting 'q1_dif3' equal to -1, we decided to try a different approach. We create three binary target variables: 'q1_dif_neg' (for negative impact) which is 1 if 'q1_dif' < 0 and zero otherwise, 'q1_dif_neu' (for neutral impact) which is 1 if 'q1_dif' = 0 and zero otherwise, and 'q1_dif_pos' (for positive impact) which is 1 if 'q1_dif' > 0 and zero otherwise. Then we optimized a k-NN classifier for each of the three binary target variables and use it to predict the missing data in them. The results can be seen below. The new binary variables with the predicted values are 'q1_dif_neg_pr', 'q1_dif_neu_pr' and 'q1_dif_pos_pr'. Unsurprisingly, as the results show, the classifier for 'q1_dif_neg' is not reliable to predict when it should be 1 (precision is 0.67 but recall is only 0.02). The classifiers for 'q1_dif_neu' and 'q1_dif_pos' are fine. More about the new approach coming next.

Target variable: **q1_dif_neg**
Optimizing k-NN: k-nearest neighbors.
Tuned Model Parameters: {'knn__n_neighbors': 14, 'pca__n_components': 7}
Accuracy: 0.9260273972602739
Classification report:

	precision	recall	f1-score	support
0.0	0.93	1.00	0.96	2026
1.0	0.67	0.02	0.05	164
accuracy			0.93	2190
macro avg	0.80	0.51	0.50	2190
weighted avg	0.91	0.93	0.89	2190



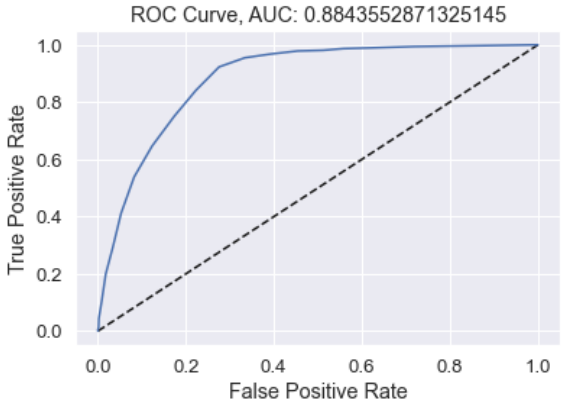
Target variable with the inclusion of predicted values:

count	23610.000000
mean	0.027446
std	0.163383
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

Name: **q1_dif_neg_pr**, dtype: float64

Target variable: **q1_dif_neu**
Optimizing k-NN: k-nearest neighbors.
Tuned Model Parameters: {'knn__n_neighbors': 19, 'pca__n_components': 13}
Accuracy: 0.7972602739726027
Classification report:

	precision	recall	f1-score	support
0.0	0.84	0.83	0.83	1347
1.0	0.73	0.75	0.74	843
accuracy			0.80	2190
macro avg	0.79	0.79	0.79	2190
weighted avg	0.80	0.80	0.80	2190

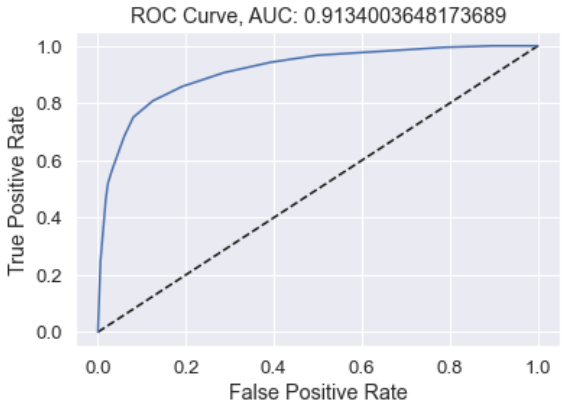


Target variable with the inclusion of predicted values:

count	23610.000000
mean	0.372342
std	0.483439
min	0.000000
25%	0.000000
50%	0.000000
75%	1.000000
max	1.000000

Name: **q1_dif_neu_pr**, dtype: float64

```
Target variable: q1_dif_pos
Optimizing k-NN: k-nearest neighbors.
Tuned Model Parameters: {'knn__n_neighbors': 17, 'pca__n_components': 13}
Accuracy: 0.8388127853881279
Classification report:
              precision    recall  f1-score   support

         0.0       0.80      0.87      0.83      1007
         1.0       0.88      0.81      0.84      1183

    accuracy                           0.84      2190
   macro avg       0.84      0.84      0.84      2190
weighted avg       0.84      0.84      0.84      2190
```




```
Target variable with the inclusion of
predicted values:
count    23610.000000
mean         0.539263
std          0.498467
min          0.000000
25%          0.000000
50%          1.000000
75%          1.000000
max          1.000000
Name: q1_dif_pos_pr, dtype: float64
```

The new binary variables with the predicted values allow us to reconstruct an alternative to 'q1_dif3_pr', which we will call 'q1_dif3_pr_alt'. It is easy to see how:

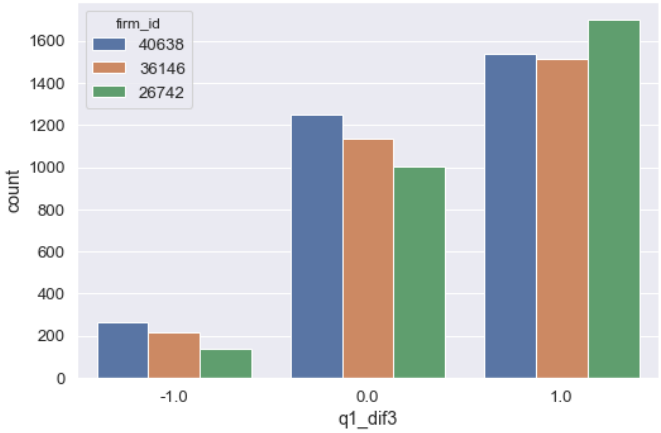
- if 'q1_dif_neg_pr' = 0 and 'q1_dif_neu_pr' = 0 and 'q1_dif_pos_pr' = 1, then 'q1_dif3_pr_alt' = 1 (visit had positive impact).
- if 'q1_dif_neg_pr' = 0 and 'q1_dif_neu_pr' = 1 and 'q1_dif_pos_pr' = 0, then 'q1_dif3_pr_alt' = 0 (visit had neutral impact).
- if 'q1_dif_neg_pr' = 1 and 'q1_dif_neu_pr' = 0 and 'q1_dif_pos_pr' = 0, then 'q1_dif3_pr_alt' = -1 (visit had negative impact).

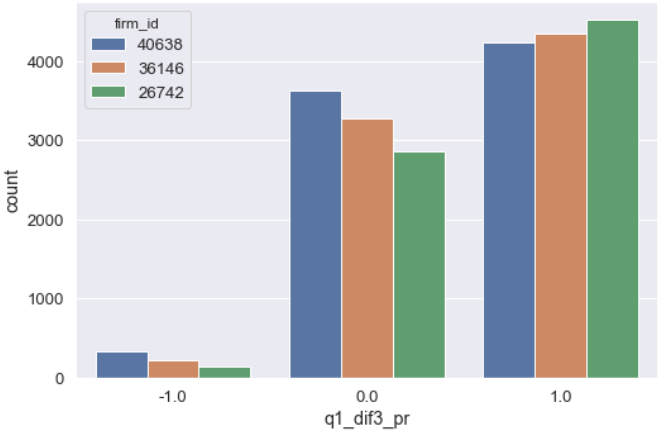
However, because we know that the classifier for 'q1_dif_neg' is not reliable to predict when it should be 1 (more precisely, because we know that the classifier for 'q1_dif_neg' underpredicts 1), we added the following rule:

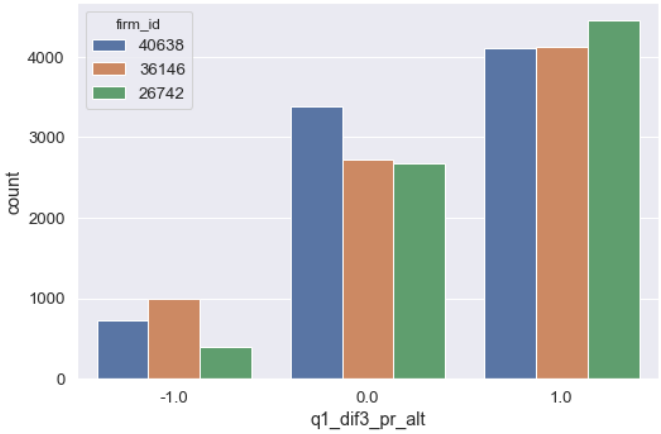
- if 'q1_dif_neg_pr' = 0 and 'q1_dif_neu_pr' = 0 and 'q1_dif_pos_pr' = 0, then 'q1_dif3_pr_alt' = -1 (visit had negative impact).

In other words, instead of only relying on 'q1_dif_neg_pr' to decide when 'q1_dif3_pr_alt' should be -1, we are also relying on the other two binary variables: when 'q1_dif_neu_pr' says that there was no neutral impact and 'q1_dif_pos_pr' also says that there was no positive impact, then 'q1_dif3_pr_alt' should register that there was negative impact no matter what 'q1_dif_neg_pr' is saying. With the inclusion of that simple rule, we can make predictions for 99.75% of the missing data (only 37 out of the initially 14851 null entries remain missing in 'q1_dif3_pr_alt', or 0.25% of the initial missing data).

Because the remaining missing data is so small, we could just remove those 37 rows, but we opted to maintain them and complete the predictions by just adding the previously predicted information from 'q1_dif3_pr'. In other words, for the remaining missing data, we just did 'q1_dif3_pr_alt' = 'q1_dif3_pr'. The results can been seen below. Only 1473 entries changed value between 'q1_dif3_pr' and 'q1_dif3_pr_alt', with most of them becoming -1 in 'q1_dif3_pr_alt'. Comparing the original distribution of 'q1_dif3', which has 14851 null entries, with the distributions of 'q1_dif3_pr' and 'q1_dif3_pr_alt', both of which having 0 null entries, we can clearly observe that 'q1_dif3_pr' was probably underpredicting -1.







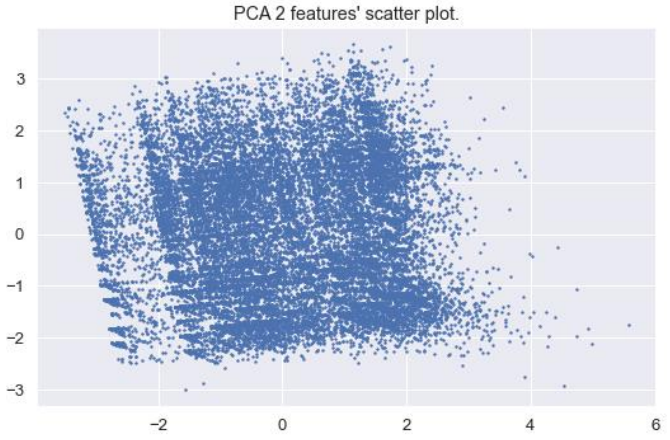
Different non-null entries between 'q1_dif3_pr' and 'q1_dif3_pr_alt':

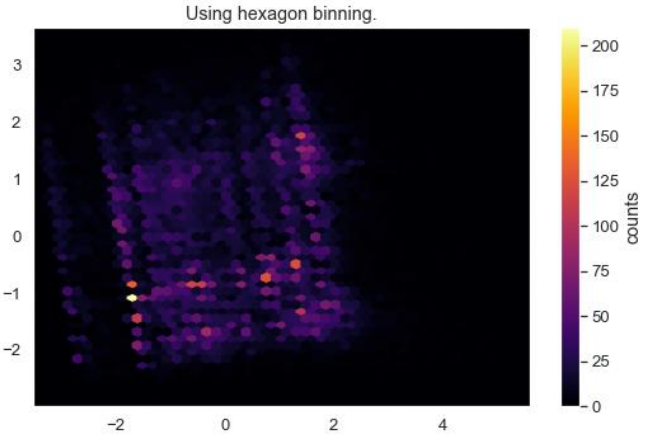
	q1_dif3_pr	q1_dif3_pr_alt
count	1473.000000	1473.000000
mean	0.323829	-0.932111
std	0.468095	0.362295
min	0.000000	-1.000000
25%	0.000000	-1.000000
50%	0.000000	-1.000000
75%	1.000000	-1.000000
max	1.000000	1.000000

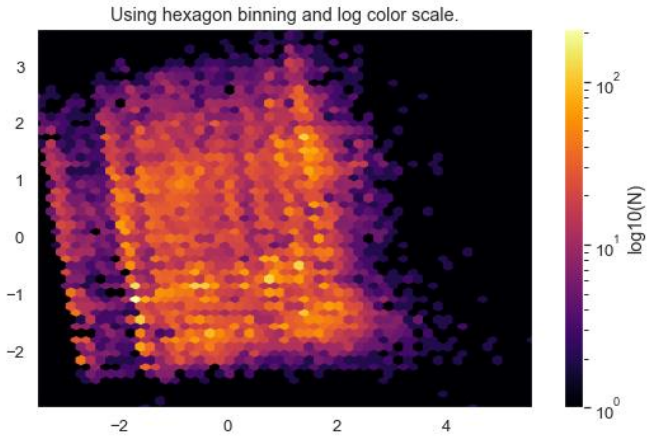
Now that we have completed the task of predicting the missing values in the variable 'q1_dif3' by building the variable 'q1_dif3_pr_alt', we can proceed to the next and last task: the identification of groups (clusters) of clients based on the information available about each individual client. In fact, to be more precise, because each entry in our database is actually a booking (that is, a client booking a visit to a firm), we will try to identify groups (clusters) of bookings based on the information available about each individual booking. To carry out this job, we will use unsupervised learning techniques. More specifically, we will use principal component analysis (PCA) and k-means clustering. Because we will be using k-means, care is necessary when selecting the list of attributes to be included. Basically, they should be on a meaningful numerical scale where distances between observations are not meaningless. Ordinal categorical variables may be included, but nominal categorical variables should not. As a result, we decided to include the following variables as attributes:

- 'region_dc_mean' and 'region_dc_half_size': these two attributes represent regions (US state, foreign country or unclear) in terms of weekly projected degree centrality value, with the first being the mean and the second being half the size of the 95% confidence interval.
- 'visit_season3': this attribute refers to time, giving the value 0 to winter months, 1 to spring or autumn months, and 2 to summer months. It preserves the cyclical nature of a year's time without creating a artificial distance between, for example, december and january (as we would have if we were to use 12 to represent december and 1 to represent january).
- 'days_bf_visit', 'guests' and 'tot_rev_addons': they are all numeric variables by nature, no issues here.
- 'q1_bef', 'q1_dif3_pr_alt' and 'q2': these are ordinal categorical variables. 'q1_bef' shows the likelihood to recommend the firm's product before visit on a scale 0 to 10. 'q1_dif3_pr_alt' shows if the visit had a negative (-1), neutral (0) or positive (1) impact on the client. 'q2' shows how often the client has used the firm's product in the past year on a scale 0 (never) to 3 (at least once a week).

Having chosen those 9 attributes, we used PCA to reduce the dimensionality from nine to two and plot the result. It is important to mention that the attributes were standardized prior to PCA using Sklearn's StandardScaler, which standardizes attributes by removing the mean and scaling to unit variance. According to the results, two dimensions account for 41.8% of the variance (the cumulative explained variance ratio). The scatter plot doesn't show clear distinct clusters, but a case could be made for four clusters.

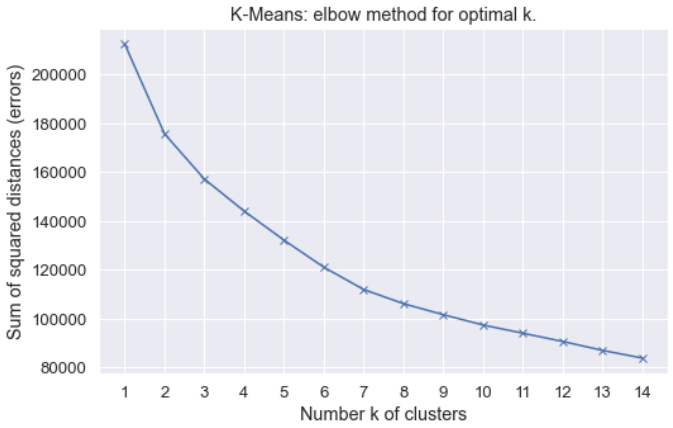







```
PCA 2 features' explained
variance: [2.04303589 1.71943262]

PCA 2 features' cumulative explained
variance ratio: [0.22699437 0.41803435]
```

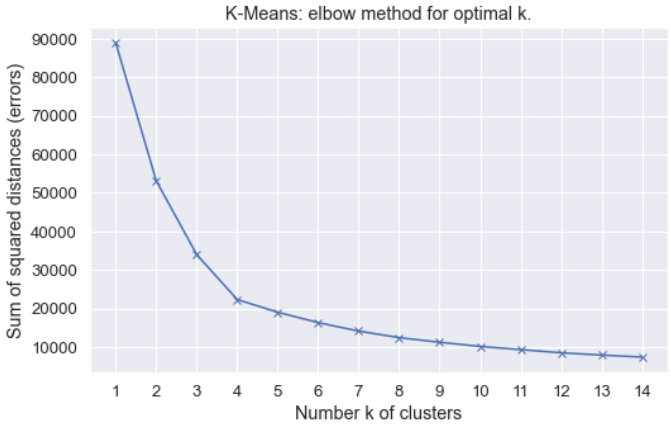
After PCA, the other unsupervised learning technique we implemented was k-means clustering. Again, prior to k-means, the attributes were standardized using Sklearn's StandardScaler. However, because the PCA analysis showed that two dimensions accounts for only 41.8% of the variance (with not much more gain by adding a third or even a forth dimension), we first opted to implement k-means without reducing dimensionality. Unfortunately, the results this time were even less clear: as we increase the number of clusters in order to find the optimal number of clusters, the sum of squared distances (errors) didn't show any evident "elbow", and the plot of the (mean) silhouette scores looked like a roller coast, which is not helpful.

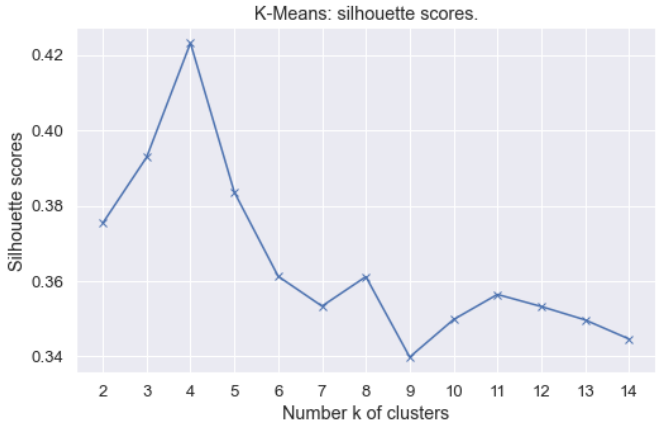


Since the k-means results without reducing dimensionality were not clear, we decided to implement k-means again but using PCA after the standardization of the attributes in order to reduce the dimensionality to two, despite the fact that two dimensions accounts for only 41.8% of the variance. This time the results were much clearer: for increasing number of clusters, the sum of squared distances (errors) show an "elbow" with 4 clusters, and the plot of the (mean) silhouette scores also shows a maximum with 4 clusters, although the value of 0.42 for average silhouette score (closer to 0 than 1) suggests a weak cluster structure (clusters close to each other that could be artificial). With 4 clusters, there is very few negative silhoute scores and the number of observations in each cluster also seems similar and, therefore, reasonable.

It is important to keep in mind two observations:

- The cluster structure with 4 clusters that we found accounts for only 41.8% of the variance. It may be possible to obtain better results if outliers are first identified and discharged before the k-means implementation.
- Regardless of the possible influence of outliers, the cluster structure with 4 clusters is weak because the suggested clusters are close to each other and the result does not rule out the possibility that there is actually only one cluster.


K-Means: elbow method for optimal k.
Sum of squared distances (errors)
Number k of clusters

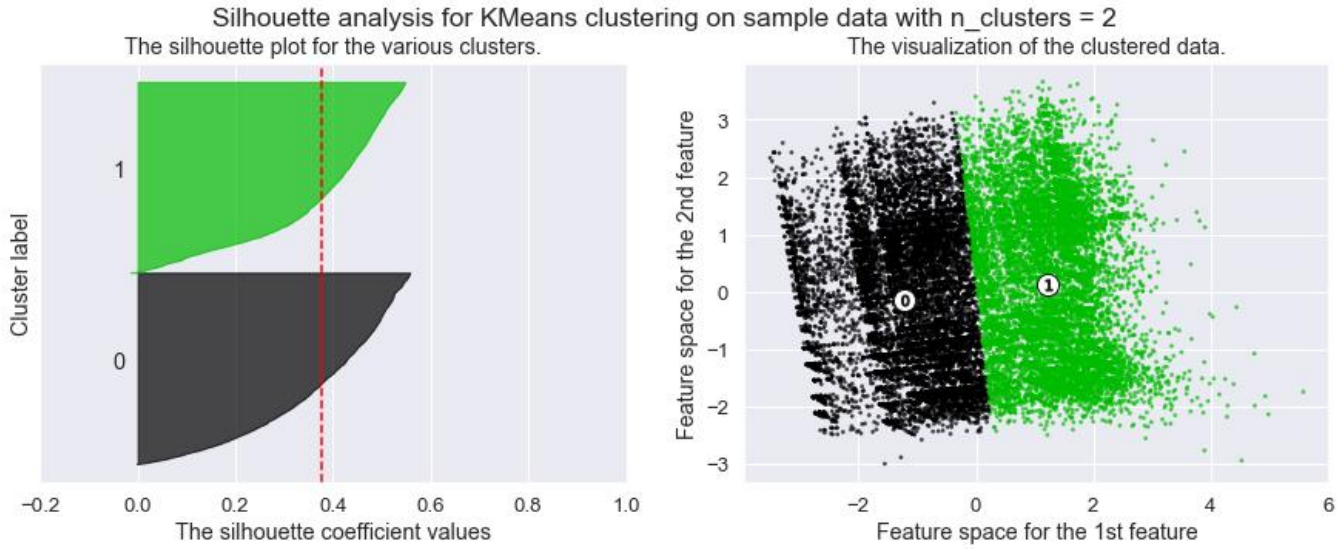

K-Means: silhouette scores.
Silhouette scores
Number k of clusters

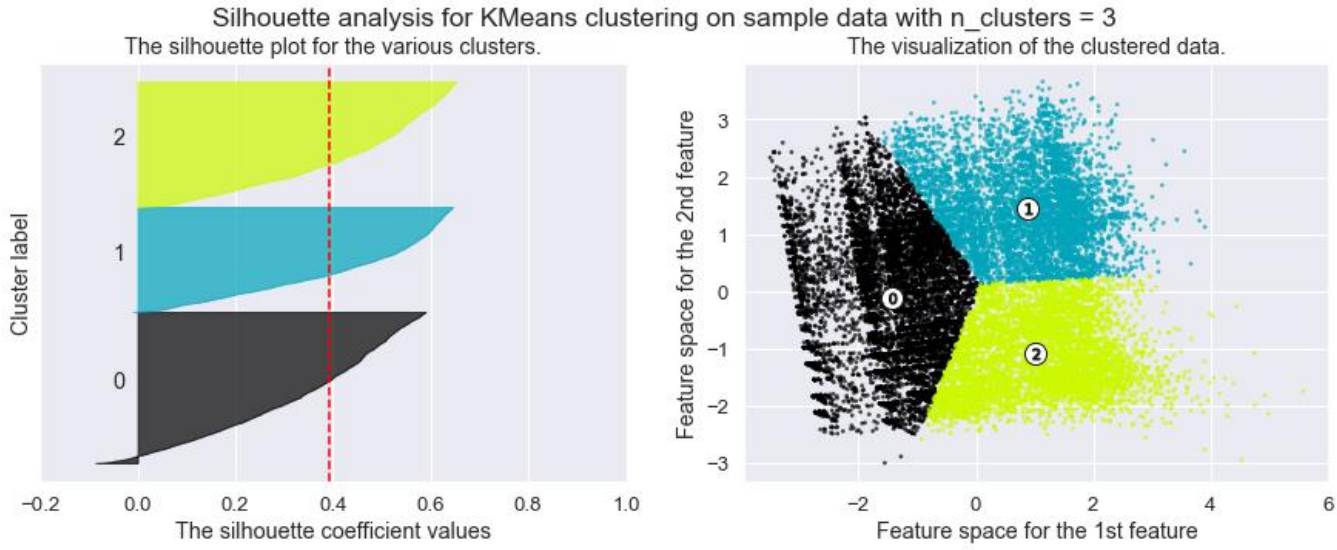

K-Means clustering, k = 2.
Number of observations
Clusters

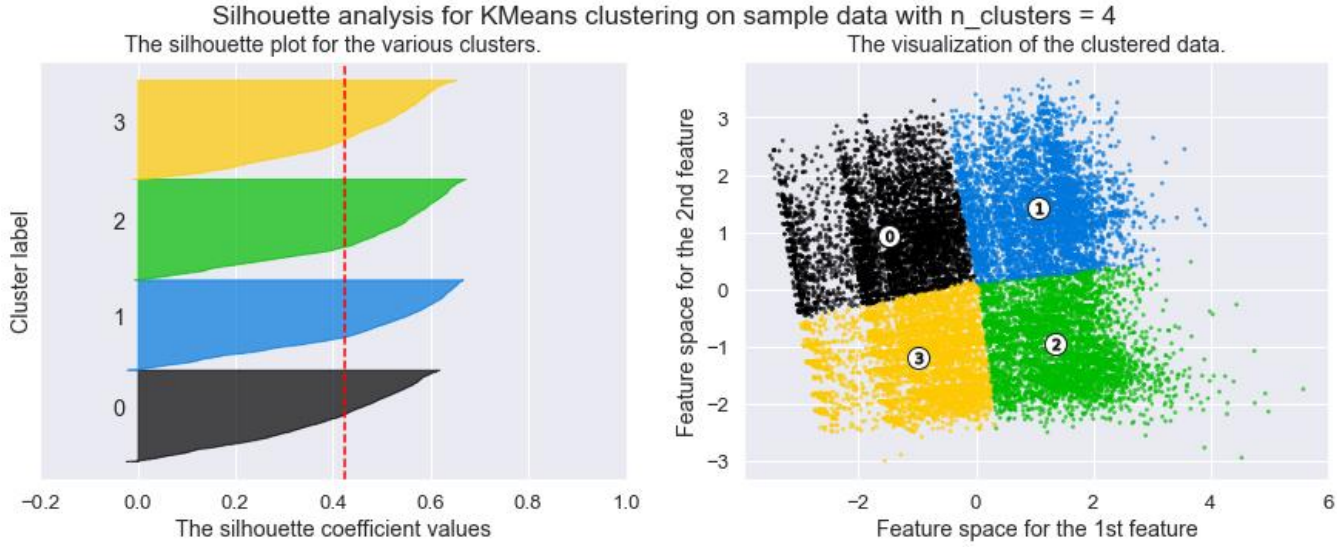

K-Means clustering, k = 3.
Number of observations
Clusters


K-Means clustering, k = 4.
Number of observations
Clusters


K-Means clustering, k = 5.
Number of observations
Clusters


Silhouette analysis for KMeans clustering on sample data with n_clusters = 2
The silhouette plot for the various clusters.
The visualization of the clustered data.
Cluster label
The silhouette coefficient values
Feature space for the 2nd feature
Feature space for the 1st feature


Silhouette analysis for KMeans clustering on sample data with n_clusters = 3
The silhouette plot for the various clusters.
The visualization of the clustered data.
Cluster label
The silhouette coefficient values
Feature space for the 2nd feature
Feature space for the 1st feature


Silhouette analysis for KMeans clustering on sample data with n_clusters = 4
The silhouette plot for the various clusters.
The visualization of the clustered data.
Cluster label
The silhouette coefficient values
Feature space for the 2nd feature
Feature space for the 1st feature

Finally, we close this report characterizing the four suggested clusters with regard to some attributes.

```
Most frequent values:
+-----------------+-----------+-------------+-----------------+---------------+
|   kmeans_labels |   firm_id | us_region   |   contact_month |   visit_month |
|-----------------+-----------+-------------+-----------------+---------------|
|               0 |     40638 | MO          |               7 |             8 |
|               1 |     40638 | MO          |               7 |             8 |
|               2 |     36146 | CA          |               8 |             9 |
|               3 |     26742 | CA          |               8 |             9 |
+-----------------+-----------+-------------+-----------------+---------------+
Most frequent values:
+-----------------+----------+------------------+------+
|   kmeans_labels |   q1_bef |   q1_dif3_pr_alt |   q2 |
|-----------------+----------+------------------+------|
|               0 |        5 |                1 |    0 |
|               1 |       10 |                0 |    3 |
|               2 |       10 |                0 |    2 |
|               3 |        5 |                1 |    1 |
+-----------------+----------+------------------+------+
Mean values:
+-----------------+-----------------+----------------+------------------+
|   kmeans_labels |   days_bf_visit |   tot_rev_fees |   tot_rev_addons |
|-----------------+-----------------+----------------+------------------|
|               0 |            5.13 |          37.1  |             0.58 |
|               1 |            7.95 |          48.66 |             1    |
|               2 |           11.79 |          55.86 |             1.72 |
|               3 |            6.59 |          41.78 |             0.62 |
+-----------------+-----------------+----------------+------------------+
```

firm_id	26742.0	36146.0	40638.0
kmeans_labels			
0	0.294000	0.195000	0.511000
1	0.159000	0.148000	0.693000
2	0.353000	0.534000	0.113000
3	0.453000	0.424000	0.123000
All	0.319000	0.333000	0.348000

q2	0	1	2	3
kmeans_labels				
0	0.445000	0.439000	0.104000	0.012000
1	0.106000	0.208000	0.256000	0.430000
2	0.132000	0.342000	0.346000	0.179000
3	0.447000	0.472000	0.076000	0.004000
All	0.283000	0.367000	0.197000	0.153000

- In terms of most often value (mode), the four clusters seem to distinguish the four categories in question 'q2', which asks how often the client used the product in the past year:
  - Cluster 0 is mainly composed by 'q2' = 0, clients who never used the product in the past year (closely followed by 'q2' = 1, those who used it only occasionaly, see last table with frequencies).
  - Cluster 1 is mainly composed by 'q2' = 3, clients who used the product at least once a week (they can be said to be fans).
  - Cluster 2 is mainly composed by 'q2' = 2, clients who used the product at least once a month (closely followed by 'q2' = 1, those who used it only occasionaly).
  - Cluster 3 is mainly composed by 'q2' = 1, clients who used the product only occasionaly (followed by 'q2' = 0, those who never used it, so clusters 0 and 3 are similar in that regard).

- Clusters 0 and 1 are mainly associated with 'firm_id' 40638 and 'us_region' MO (where the firm is located). Cluster 0 mostly contains clients who would give the product grade 5 out of 10 (question 'q1_bef') before the visit, with that grade improving after the visit (variable 'q1_dif3_pr_alt'). Cluster 1 mostly contains clients who would give the product grade 10 out of 10 before the visit, with that grade remaining the same after the visit.
- Cluster 2 is mainly associated with 'firm_id' 36146 and 'us_region' CA (where the firm is located). Cluster 3 is mainly associated with 'firm_id' 26742 but also with 'us_region' CA (instead of CO where the firm is located). That happens because cluster 3 is weakly associated with 'firm_id' 26742 and could also be associated with 'firm_id' 36146 (see the table with frequencies at the bottom). Therefore, cluster 3 is also connected to 'firm_id 36146'.
- Cluster 2, like cluster 1, mostly contains clients who would give the product grade 10 out of 10 before the visit, with that grade remaining the same after the visit. Cluster 3, like cluster 0, mostly contains clients who would give the product grade 5 out of 10 before the visit, with that grade improving after the visit.

In summary, there is evidence to say that, in general, customers who used and valued the product poorly before the visit improved their evaluation of the product after the visit. Customers who already used and valued the product well before the visit maintained their evaluation of the product after the visit.