1. 1. What is the average number of calls per month regardless of year?

|  |
| --- |
| **1A AVERAGE NUMBER OF CALLS REGARDLESS OF YEAR**  # Grouping by MONTH  concat\_copy = concatenated.copy()  # Retaining SUBID Column only  concat\_copy.drop(['TRANSID', 'TRANS', 'VALIDITY', 'YEAR'], axis = 1, inplace = True)  # Renaming the SUBID to AVERAGE NUMBER OF CALLS PER MONTH  concat\_copy.rename(columns = {'SUBID':'AVERAGE NUMBER OF CALLS PER MONTH'}, inplace = True)  #print(concat\_copy)  # Grouping the dataframe by month  by\_month = concat\_copy.groupby('MONTH')  # Getting the AVERAGE number of calls per month  ave\_calls = by\_month.count()/3  print(ave\_calls) |

Text

Description automatically generated with medium confidence

* 1. Which month has the highest average call volume? Which has the lowest?

March has the highest average call per volume.

July has the lowest average call per volume.

|  |
| --- |
| 1B HIGHEST AND LOWEST AVERAGE CALL VOLUME  max\_ave = ave\_calls.max()  min\_ave = ave\_calls.min()  print(max\_ave, "\n\n", min\_ave) |

Logo

Description automatically generated with low confidence

* 1. What are the total number of calls for each month from 2013 to 2015?

|  |
| --- |
| concat2\_copy = concatenated.copy()  by\_year\_month = concat2\_copy.groupby(['YEAR','MONTH'])  total\_by\_year = by\_year\_month.size()  print(total\_by\_year) |

A picture containing table

Description automatically generated

* 1. Which month and year had the highest traffic of calls?

|  |
| --- |
| max\_by\_year = total\_by\_year.max()  print(max\_by\_year) |

Output: 6948  
Answer: March 2013

* 1. How many invalid calls were made during the 1st and 4th quarters of the 3-year dataset only? How many where made on each type?

|  |
| --- |
| concat3\_copy = concatenated.copy()  concat3\_copy.drop(['SUBID','TRANSID'], axis=1, inplace=True)  print(concat3\_copy)  #%% SETTING MONTH AS THE INDEX - TO REMOVE Q2 AND Q3  index\_month = concat3\_copy.set\_index('MONTH', inplace=False)  # DROPPING Q2-Q3  Q2\_to\_Q3 = [4,5,6,7,8,9]  only\_Q1Q4 = index\_month.drop(Q2\_to\_Q3, axis=0, inplace=True)  #%% REMOVING VALID VALUES  index\_validity = index\_month.set\_index('VALIDITY', inplace=False)  sans\_valid = index\_validity.drop('VALID', axis=0, inplace=False)  #%% DETERMINING INVALID CALLS PER TYPE  by\_validity = sans\_valid.groupby(['VALIDITY','TRANS'])  print(by\_validity.size())  #%% DETERMINING NUMBER OF INVALID CALLS PER MONTH (Q1 AND Q4 ONLY)  # RESETTING THE INDEX OF THE DATAFRAME WITHOUT VALID VALUES  reset\_index = sans\_valid.reset\_index()  print(reset\_index)  #%% GROUPING BY YEAR (Q1 AND Q4 ONLY)  by\_year = reset\_index.groupby('YEAR')  print(by\_year.count()[['VALIDITY']], "\n")  print(by\_year.count().sum()[['VALIDITY']]) |

Text

Description automatically generatedText

Description automatically generated

Figure 1 Number of invalid calls per type

Figure 2 Number of invalid calls in the 1st and 4th quarters of each year, and the total number of invalid calls

There are 21,241 invalid calls made during the 1st and 4th quarters in 2013-2015, as shown in Figure 2. The number of invalid calls per type is shown in Figure 1.

* 1. How many times on average did a subscriber call? Also calculate for the Standard Deviation, Median, Variance, Range, Minimum, Maximum, Sum and Count of this statistic.

|  |
| --- |
| **AVERAGE NUMBER OF CALLS FROM A SUBSCRIBER**  # This uses the SUBID and TRANS Column  # Count the number of times that each SUBID called - group by SUBID  concat4\_copy = concatenated.copy()  # GROUPING THE DATAFRAME BY SUBID  by\_subid = concat4\_copy.groupby('SUBID')  # GETTING THE STATISTICS ABOUT SUBSCRIBER CALLS  df = by\_subid['TRANS']  total\_subcalls = df.count()  ave\_subcalls = total\_subcalls.mean()  std\_subcalls = total\_subcalls.std()  median\_subcalls = total\_subcalls.median()  variance\_subcalls = total\_subcalls.var()  max\_subcalls = total\_subcalls.max()  min\_subcalls = total\_subcalls.min()  range\_subcalls = max\_subcalls - min\_subcalls  sum\_subcalls = total\_subcalls.sum()  count\_subcalls = total\_subcalls.count()  print("A subscriber called", ave\_subcalls, "on average.","\n")  print("Standard deviation:", std\_subcalls,"\n")  print("Median:", median\_subcalls,"\n")  print("Variance:", variance\_subcalls,"\n")  print("Maximum:", max\_subcalls,"\n")  print("Minimum:", min\_subcalls,"\n")  print("Range:", range\_subcalls,"\n")  print("Sum:", sum\_subcalls,"\n")  print("Count:", count\_subcalls) |

Text

Description automatically generated

1. What were the top 5 transactions availed by the subscribers when calling?

|  |
| --- |
| **TOP 5 TRANSACTIONS AVAILED BY SUBSCRIBERS**  concat5\_copy = concatenated.copy()  # RETAINING SUBID AND TRANS  concat5\_copy.drop(['TRANSID','VALIDITY','MONTH','YEAR'], axis=1, inplace=True)  # RENAMING SUBID TO FREQUENCY  concat5\_copy.rename(columns = {'SUBID': 'FREQUENCY'}, inplace=True)  # GROUPING BY TRANS  by\_trans = concat5\_copy.groupby('TRANS')  trans\_frequency = by\_trans.count()  # RANKING THE FREQUENCY OF CALLS PER TRANSACTION IN DESCENDING ORDER  rank\_frequency = trans\_frequency.sort\_values(by='FREQUENCY', ascending=False)  top5 = rank\_frequency.head(5)  print(top5) |

Text

Description automatically generated

1. A repeat call is when a customer calls the hotline again for the same transaction (assume time between calls does not matter). Given that the first call is not a repeat call
   1. What is the average number of repeat calls per subscriber (for all subscribers)?

|  |
| --- |
| 6A AVERAGE OF REPEAT CALLS FOR ALL SUBSCRIBERS  concat6\_copy = concatenated.copy()  # GROUP BY SUBID THEN GET THE FREQUENCY OF CALLS PER SUBID  bysubidtrans = concat6\_copy.groupby(['SUBID','TRANS']).size().to\_frame('COUNT').reset\_index()  descending = bysubidtrans.sort\_values(by='COUNT', ascending=False)  bysubid = descending.groupby('SUBID').count()  repeat\_call = descending[descending['COUNT'] > 1]  repeat\_call = repeat\_call.groupby(['SUBID'])  repeat\_call = repeat\_call.count()  print(repeat\_call.sum()/len(bysubid)) |

A screenshot of a computer

Description automatically generated with low confidence

* 1. What is the average number of repeat calls per subscriber (for subscribers with at least one repeat call only)?

|  |
| --- |
| print(repeat\_call.sum()/len(repeat\_call)) |

