# DS804 - Assignment 1

* By Dikshant Joshi

Q1)

* **Summary column**

Step 1 : Cleansing of Data.

Step 2 : Duplicated summary column.

Step 3: Renamed duplicated Summary Column from Summary-copy to summary\_count.

Step 4: Replaced null values with empty space in summary\_count column.

Step 5: Added custom column with summary\_count\_number\_words name and function : List.Count(List.RemoveItems(Text.SplitAny([#"summary\_count"],"#(tab) #(lf)"),{""})) Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated with medium confidence

Text

Description automatically generated with medium confidence

* **Space column**

Followed the same above steps .

Text

Description automatically generated with medium confidence

* **Description**

Followed the same above steps.

**Graphical user interface, text

Description automatically generated**

* **Neighbourhood Overview**

Followed the same above steps.

Text

Description automatically generated with medium confidence

Q2)

* **Calendar\_updated**

Step 1: Duplicated Calendar\_updated column.

Step 2: Replaced null values with empty space in Calendar\_updated-copy column.

Step 3: Added custom column with Calendar\_updated\_weeks name and function :

if Text.Contains([#"calendar\_updated"], "months")

then

Number.FromText(

Text.Select(

[#"calendar\_updated"],{"0".."9"})

)\*4

else if Text.Contains([#"calendar\_updated"], "weeks")

then Number.FromText(

Text.Select([#"calendar\_updated"],{"0".."9"})

)

else if Text.Contains([#"calendar\_updated"], "a week ago")then 1

else 0

Table

Description automatically generated

Q3)

* **Zipcode**

Step 1: Duplicated Zipcode column.

Step 2: Changed Zipcode data type to text

Step 3: Added custom column with Zipcode\_corrected name and function :

if Text.Length([#"zipcode - Copy"]) = 4

then Text.PadStart([#"zipcode - Copy"],5,"0")

else [#"zipcode - Copy"]

Table

Description automatically generated

Q4)

* **Host Since**

Step 1: Added custom column with host\_since\_day name and function:

Date.DayOfWeekName([#"host\_since"],"en-US")

Step 2: Added custom column with host\_since\_month name and function:

Date.MonthName([#"host\_since"],"en-US")

Step 3: Added custom column with host\_since\_year name and function :

Date.Year([#"host\_since"])

Table

Description automatically generated

* **Host\_Since\_Duration\_Days**

Step 1: Created a new column with function :

Host\_Since\_Duration\_Days = DATEDIFF(listings[host\_since],TODAY(),DAY)

Table

Description automatically generated

Q5)

* **Host Location**

Step 1: Duplicated host\_location column

Step 2: Added custom column with Custom.1 name and function, to complete the given location so that data is clean:

if [#"host\_location - Copy"] = "Boston, MA" then "Boston, Massachusetts, United States"

else if [#"host\_location - Copy"] = "Boston, Massachusetts" then "Boston, Massachusetts, United States"

else if [#"host\_location - Copy"] = "PK" then "Pakistan"

else if [#"host\_location - Copy"] = "ID" then "Indonesia"

else if [#"host\_location - Copy"] = "TN" then "Tennessee, United States"

else if [#"host\_location - Copy"] = "Roslindale (part of Boston), MA" then "Boston, Massachusetts, United States"

else if [#"host\_location - Copy"] = "Massachusetts" then "Massachusetts, United States"

else if [#"host\_location - Copy"] = "Texas" then "Texas, United States"

else if [#"host\_location - Copy"] = "US" then "United States"

else if [#"host\_location - Copy"] = "CN" then "China"

else if [#"host\_location -Copy"] = "Boston, From Jamaica " then "Boston, Massachusetts, United States "

else [#"host\_location - Copy"]

Step 3: Added custom column with Custom.2 name and function, to count number of commas in column Custom.1:

List.Count( Text.ToList(Text.Select([Custom.1], ",")))

Step 4: Added custom column with Custom.3 name and function, to make the number of delimiters equal in column Custom.1 with the help of column Custom.2:

if [Custom.2]=0 then ",,"& [Custom.1]

else if [Custom.2]=1 then ","& [Custom.1]

else [Custom.1]

Step 5: Split the column Custom.3 and rename the columns generated after splitting.

Table

Description automatically generated

Table

Description automatically generated

Text, table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Q6)

* **Host Response Time**

Step 1: Duplicated host\_response\_time column.

Step 2: Removed null values in host\_response\_time – Copy column with Host\_Response\_Time\_Null.

Step 3: Created an Index column

Step 4: Pivot the host\_response\_time – Copy, replaced null values with 0 and renamed all the columns appropriately.

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

Graphical user interface, application, table, Excel

Description automatically generated

* **Host is Superhost**

Followed the same above steps as in Host Response Time

Calendar

Description automatically generated with medium confidence

* **Host Identity Verified**

Followed the same above steps as in Host Response Time

Table

Description automatically generated with medium confidence

* **Host has profile pic**

Followed the same above steps as in Host Response Time

Table, calendar

Description automatically generated with medium confidence

* **Host Verification**

Step 1: Duplicated host\_verification column.

Step 2: Removed [,],’ in host\_verification – Copy column.

Step 3: Replaced null value in host\_verification – Copy column with host\_verification\_null.

Step 4: Created an index column

Step 5 : Pivot the host\_verification – Copy, replaced null values with 0.

Calendar

Description automatically generated

* **Amenities**

Followed same steps as that in Host Verififcation.

Calendar

Description automatically generated

Q7)

* **Boston Neighbourhood Data**

Step 1: Imported the Boston Neighbourhood Data into Listings.

Graphical user interface, table

Description automatically generated with medium confidence

Table

Description automatically generated

Graphical user interface, application

Description automatically generated

Step 2: Cleaned the data

* Removed unwanted top rows and bottom rows from each table i.e.; Age, Per Capita Income, Housing Tenure, Poverty Rates, Race and Ethnicity
* Made the top row as header
* Final cleaned data looked like the following:

Graphical user interface, application, table, Excel

Description automatically generated

Graphical user interface

Description automatically generated with medium confidenceGraphical user interface, application, Word

Description automatically generated

Age

Table

Description automatically generated

Housing Tenure

Table

Description automatically generated

Per Capita Income

Table

Description automatically generated

Poverty Rates

Table

Description automatically generated

Race and Ethnicity

Table

Description automatically generated

Step 3: Resolved the differences between neighbourhood column in the listings table and the Boston data by duplicating the neighbourhood\_cleansed column in listings data and making changes as following:

• Chinatown & Leather District = Downtown

• Bay Village=South End

• Longwood Medical Area=Longwood

Renaming the column as Neighbourhood\_join

Step 4: Creating a fact table to link Boston data to listings data

* Added the Neighbourhood\_join column as a new query.
* Removed duplicates from the column.
* Transformed query into the table using To Table tool.
* Renamed the table as Fact Table with column as Neighbourhood\_join
* Saved everything and closed the power query.

Chart, waterfall chart

Description automatically generatedGraphical user interface, table

Description automatically generated

Graphical user interface, application

Description automatically generated

Final listings workbook will look like the following.

Graphical user interface, text

Description automatically generated

Q8)

* **Creating visualizations in Power BI**

Step 1: Open Powe BI and select model on the left pane.

Graphical user interface, application

Description automatically generated

Step 2: Click on get Data and load the listings workbook into the Power BI.

Step 3: Update the relationship between fact table and all the other tables as following:

* Listings[Neighbourhood\_join] -> FactTable[Neighbourhood\_join]
* Per Capita Income[Neighbourhood] -> FactTable[Neighbourhood\_join]
* Age[Neighbourhood] -> FactTable[Neighbourhood\_join]
* Housing Tenure[Neighbourhood] -> FactTable[Neighbourhood\_join]
* Race and Ethnicity[Neighbourhood] -> FactTable[Neighbourhood\_join]
* Poverty Rates[Neighbourhood] -> FactTable[Neighbourhood\_join]

Step 4: Add reviews data set and create relationship with that as follows:

* Listings[id] -> reviews[listing\_id]

The final model will look like the following:

Graphical user interface, application

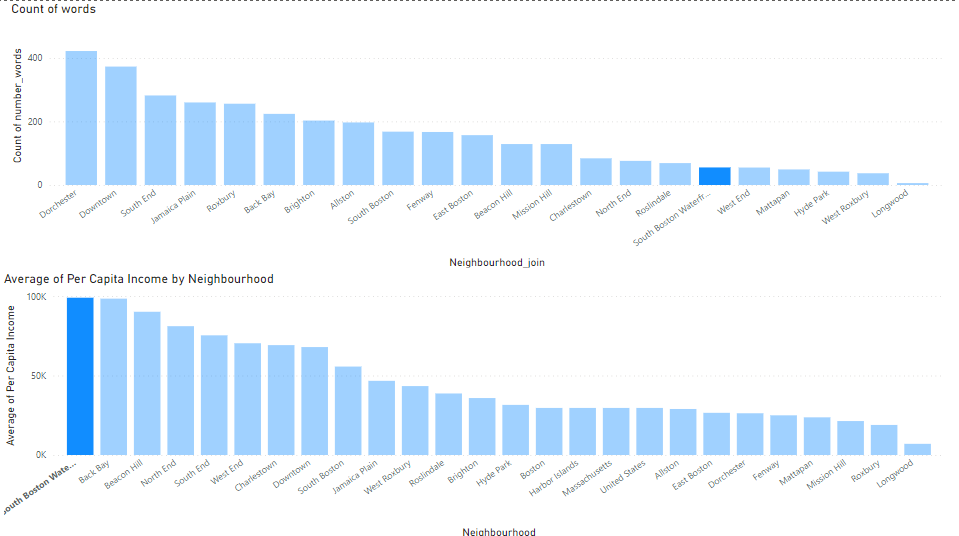
Description automatically generated

1. **Do owners from more affluent neighbourhoods use more words to describe their properties?**

Ans- To answer this question I used two clustered column charts as follows:

Count of words vs Neighbourhood

Average of Per Capita Income vs Neighbourhood



This clearly depicts that more affluent neighbourhood does not use more words to describe their properties.

1. **What percentage of listings were updated between 1-10 weeks?**

Ans – For this I created two tables with columns count of id, calendar\_updated\_weeks. In one table I applied filter for weeks less than equal to 10. In other table I calculated total listings updated.

Table

Description automatically generated with low confidence

So according to this the percentage of listings that were updated between 1-10 weeks is (1729/3440)\*100 = 50.2616%.

1. **Do any zip codes have more Superhosts than non Superhosts? If so, which Zip codes are they?**

Ans – For this I used a line chart with zipcode on X-axis and sum of host\_is\_superhost\_t , sum of host\_is\_superhost\_f on Y-axis.

Chart, line chart

Description automatically generated

It clearly depicts that at 3 points number of Super hosts is greater than number of Non-super hosts. Zip codes for those points are:

02129, 02136, 02199

1. **In which month of the year do the fewest number of people become hosts?**

Ans – For this I used clustered column chart with count of host\_since on Y-axis and host\_since month on X-axis.

Chart, bar chart

Description automatically generated

It clearly shows that in the month of April the fewest number of people became hosts (167).

1. **What proportion of the Superhosts in the United States do host from California constitute?**

Ans – For this I used Donut Chart with legend as host\_location\_state, Values as Count of Host\_is\_superhost\_t, and details as host\_location\_country and specified United States in filter.

Chart, sunburst chart

Description automatically generated

This tells us that 3.39% (97 Superhosts) host from California state.

1. **Is the** **proportion of hosts from Massachusetts who respond within an hour, larger the proportion of hosts from California who respond within an hour? Show** **different host response times across all the states.**

Ans – For this I used donut chart with legend as host\_response\_time, Values as Average of Host\_response\_rate, and details as host\_location\_state and specified California, Massachusetts in filter.

Chart, sunburst chart

Description automatically generated

The chart tells us that proportion of hosts from Massachusetts who respond within an hour (97.42%) is larger the proportion of hosts from California who respond within an hour (95.27%).

Below is the different host response times across all the states

Chart, sunburst chart

Description automatically generated

1. **For listings whose price is between $1 and $500 show the number have pools and hot tubs. Property prices should be grouped shown in increments of $50**

Ans – For this I used clustered column chart with price on X-axis and No. of Pool and No. of Hot tuns on Y-axis. To show the listings whose price is between $1 and $500 I applied a filter on price . To group the Property prices shown in increments of $50 I chose New group and specified the Bin Size as $50.

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, application

Description automatically generated

My final visualization looked like the following:

Chart, bar chart

Description automatically generated

1. **Generate summary statistic(Average, Minimum, Maximum, Standard deviation, variance, median, count) for the following columns and interpret your output [price],[review\_scores\_rating],[description\_numberofWords],[number\_of\_reviews],[host\_total\_listings\_count**

Ans – For this I created different tables for different column Statistics as following visual shows.

Text, email

Description automatically generated

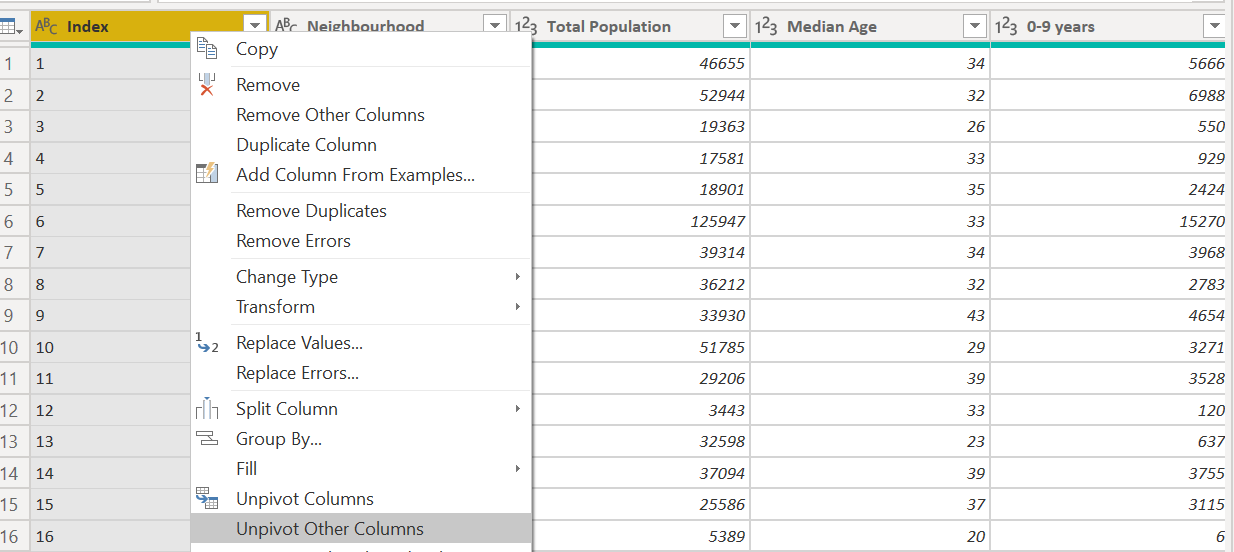
1. **Generate a correlation matrix for each of the following columns and interpret your output [price],[review\_scores\_rating],[description\_numberofWords],[Total Population],[Median Age],[Per Capita Income],[number\_of\_reviews],[host\_total\_listings\_count],[host\_response\_time\_within an hour].**

Ans:

Step 1:Merged Age,listings and Per Capita Income table into one table merge.

Step 2: Created an index for the Merge table.

Step 3: Created a table Merge\_pivot with source = merge and Unpivoted other columns.



Step 4: Filtered the rows for which correlation is required.

Step 5: Created 2 tables Attribute row, Attribute column with source as Merged\_pivot. Select the attribute column and select remove other columns.

Step 6: Remove duplicates .

Step 7: Filtered the rows required for the correlation plot. Close and apply

Step 8: Created measures for correlation(n,X,XY,Y,X2,Y2 and correlation\_coeff). Following are the code:

* XY =

VAR CurrentX = SELECTEDVALUE('Attribute row'[Attribute])

VAR CurrentY = SELECTEDVALUE('Attribute column'[Attribute])

VAR VIRTUAL =

SUMMARIZE(

    'Merge\_pivot',

    'Merge\_pivot'[Index],

    "X", CALCULATE(MAX('Merge\_pivot'[Value]),'Merge\_pivot'[Attribute] = CurrentX),

    "Y", CALCULATE(MAX('Merge\_pivot'[Value]),'Merge\_pivot'[Attribute] = CurrentY)

)

RETURN

SUMX(

    VIRTUAL,

    [X] \* [Y]

)

* X =

VAR CurrentX = SELECTEDVALUE('Attribute row'[Attribute])

VAR CurrentY = SELECTEDVALUE('Attribute column'[Attribute])

VAR VIRTUAL =

SUMMARIZE(

    'Merge\_pivot',

    'Merge\_pivot'[Index],

    "X", CALCULATE(MAX('Merge\_pivot'[Value]),'Merge\_pivot'[Attribute] = CurrentX),

    "Y", CALCULATE(MAX('Merge\_pivot'[Value]),'Merge\_pivot'[Attribute] = CurrentY)

)

RETURN

SUMX(

    VIRTUAL,

    [X]

)

* X^2 =

VAR CurrentX = SELECTEDVALUE('Attribute row'[Attribute])

VAR CurrentY = SELECTEDVALUE('Attribute column'[Attribute])

VAR VIRTUAL =

SUMMARIZE(

    'Merge\_pivot',

    'Merge\_pivot'[Index],

    "X", CALCULATE(MAX('Merge\_pivot'[Value]),'Merge\_pivot'[Attribute] = CurrentX),

    "Y", CALCULATE(MAX('Merge\_pivot'[Value]),'Merge\_pivot'[Attribute] = CurrentY)

)

RETURN

SUMX(

    VIRTUAL,

    [X] \* [X]

)

* Y^2 =

VAR CurrentX = SELECTEDVALUE('Attribute row'[Attribute])

VAR CurrentY = SELECTEDVALUE('Attribute column'[Attribute])

VAR VIRTUAL =

SUMMARIZE(

    'Merge\_pivot',

    'Merge\_pivot'[Index],

    "X", CALCULATE(MAX('Merge\_pivot'[Value]),'Merge\_pivot'[Attribute] = CurrentX),

    "Y", CALCULATE(MAX('Merge\_pivot'[Value]),'Merge\_pivot'[Attribute] = CurrentY)

)

RETURN

SUMX(

    VIRTUAL,

    [Y] \* [Y]

)

* Y =

VAR CurrentX = SELECTEDVALUE('Attribute row'[Attribute])

VAR CurrentY = SELECTEDVALUE('Attribute column'[Attribute])

VAR VIRTUAL =

SUMMARIZE(

    'Merge\_pivot',

    'Merge\_pivot'[Index],

    "X", CALCULATE(MAX('Merge\_pivot'[Value]),'Merge\_pivot'[Attribute] = CurrentX),

    "Y", CALCULATE(MAX('Merge\_pivot'[Value]),'Merge\_pivot'[Attribute] = CurrentY)

)

RETURN

SUMX(

    VIRTUAL,

    [Y]

)

* CORRELATION COEFF =

DIVIDE(

    [N]\*[XY] - [X]\*[Y],

    SQRT(

        ([N]\*[X^2] - [X]^2)\*([N]\*[Y^2]-[Y]^2)

    )

)

* n = DISTINCTCOUNT(Merge\_pivot[Index])

Step 9: Created a matrix with row as Attribute\_row, column as Attribute\_column and values as Correltion\_coeff. Following is the Correlation table which we will get:

A screenshot of a computer

Description automatically generated with medium confidence

Q9)

* **Creating a Dashboard**

1)Superhost: Created a donut chart with **legend** as host\_is\_superhost and **values** as count\_of\_host\_response\_rate. Also selected label contents as Data values, Percent of total.

Graphical user interface, text, application

Description automatically generated

2)Count of comment vs month : Created a clustered column chart with **X-axis** as date-Month and **Y-axis** as Count of comments.

3)Comments :Created a word cloud with **category** as comments Also created a card with **fields** as count of comments.

4)Price by location : Created a Map with **latitude** as latitude and **longitude** as longitude and **Bubble size** as Average of Price. Also created a function for **bubble colors** as shown and created a card with **category** as Average of Review\_Score\_rating.

Table

Description automatically generated

5)Neighbourhood, Review Score and Daily Price: Created a slicer for all three with **field** as Neighbourhood\_join,price,review\_score\_rating respectively. Also added a **filter** for Daily price in price as follows to display only a range of price. For Neighbourhood selected dropdown as the **type** of slicer.

Diagram

Description automatically generatedGraphical user interface, application

Description automatically generated

Finally did some text editing to display the dashboard as was displayed in the question. Final dashboard looked like the following:

