SOUTHERN ILLINOIS UNIVERSITY EDWARDSVILLE

CMIS 427 Security Analytics

Assignment 4

Analyzing Data with R and Python

Author:
Luis Barrios
MS CMIS Student
Professor:
Dr. Joseph Vithayathil

March 5, 2022

Table of Contents

Analyzing Data with R code

Listing 3-19: It consist in creating a contingency table to show the relationship between risk and reliability from the previous dataset in assignment 2. In this case, we must initiate the .data file and assign its columns.

```
av <- read.csv("C:/Users/luisa/Documents/SIUE/Courses/CMIS 427
                  /assignment/as2/reputation.data", sep="#", header=FALSE)

# assign more readable column names since we didn't pick

colnames(av) <- c("IP", "Reliability", "Risk", "Type","Country", "Locale", "Coords", "x")
```

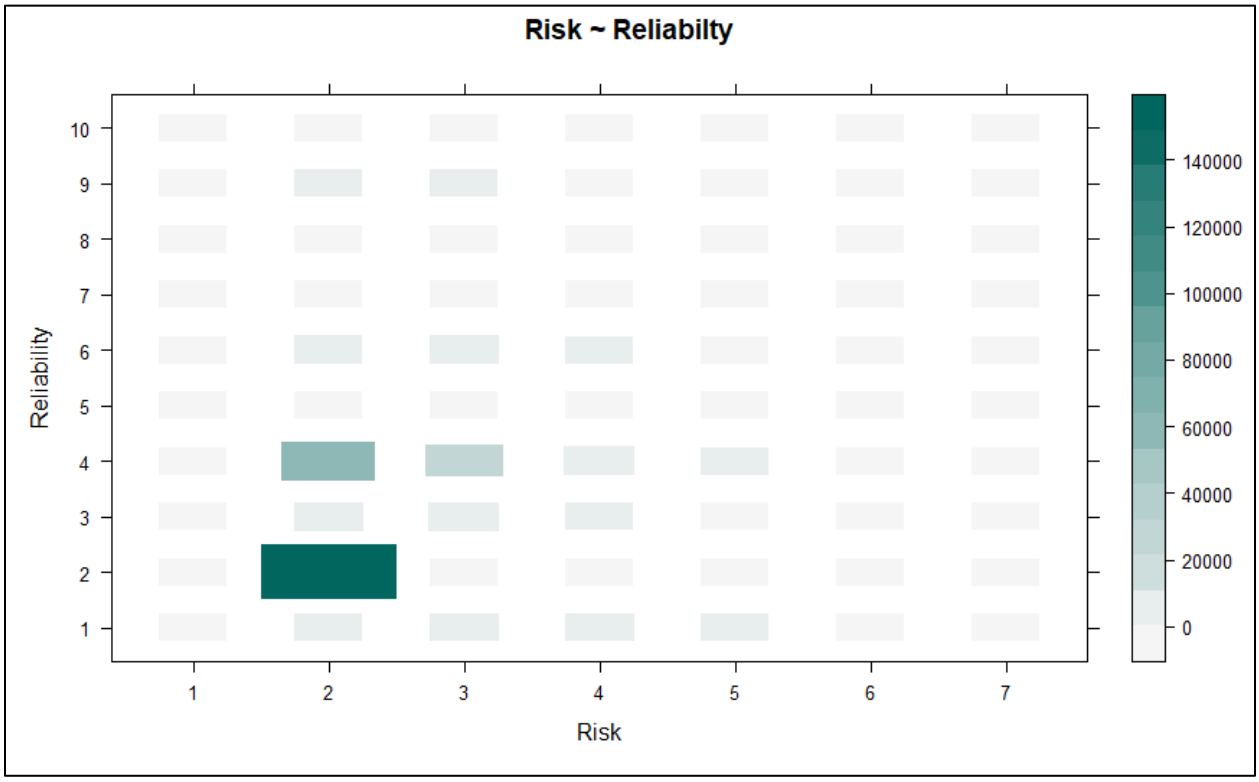
Compute a contingency table for risk and reliability factors at (x ,y) location. Then, graph a heat map based on the contingency table to visualize the most significant relations. See figure 1 for "ftable" reference and figure 2 for listing 3-19 output.

```
#listing 3-19

# compute contingency table for Risk/Reliability factors which
# produces a matrix of counts of rows that have attributes at
# each (x, y) location


rr.tab <- xtabs(~Risk+Reliability, data=av)

ftable(rr.tab) # print table


# graphical view of levelplot

# need to use levelplot function from lattice package

library(lattice)


# cast the table into a data frame

rr.df = data.frame(table(av$Risk, av$Reliability))


# set the column names since table uses "Var1" and "Var2"

colnames(rr.df) <- c("Risk", "Reliability", "Freq")


# now create a level plot with readable labels

levelplot(Freq~Risk*Reliability, data=rr.df, main="Risk ~ Reliabilty",

ylab="Reliability", xlab = "Risk", shrink = c(0.5, 1),

col.regions = colorRampPalette(c("#F5F5F5", "#01665E"))(20))
```

Figure 1: output for ftable

```
> ftable(rr.tab) # print table
     Reliability      1      2      3      4      5      6      7      8      9     10
Risk
1                     0      0     16      7      0      8      8      0      0      0
2                   804 149114   3670  57653      4   2084     85     11    345     82
3                  2225      3   6668  22168      2   2151    156      7    260     79
4                  2129      0    481   6447      0    404     43      2     58     24
5                   432      0     55    700      1    103      5      1     20     11
6                    19      0      2     60      0      8      0      0      1      0
7                     3      0      0      5      0      0      0      0      2      0
>
```

Figure 2: Listing 3-19 output | Heat map for contingency table for risk-reliability

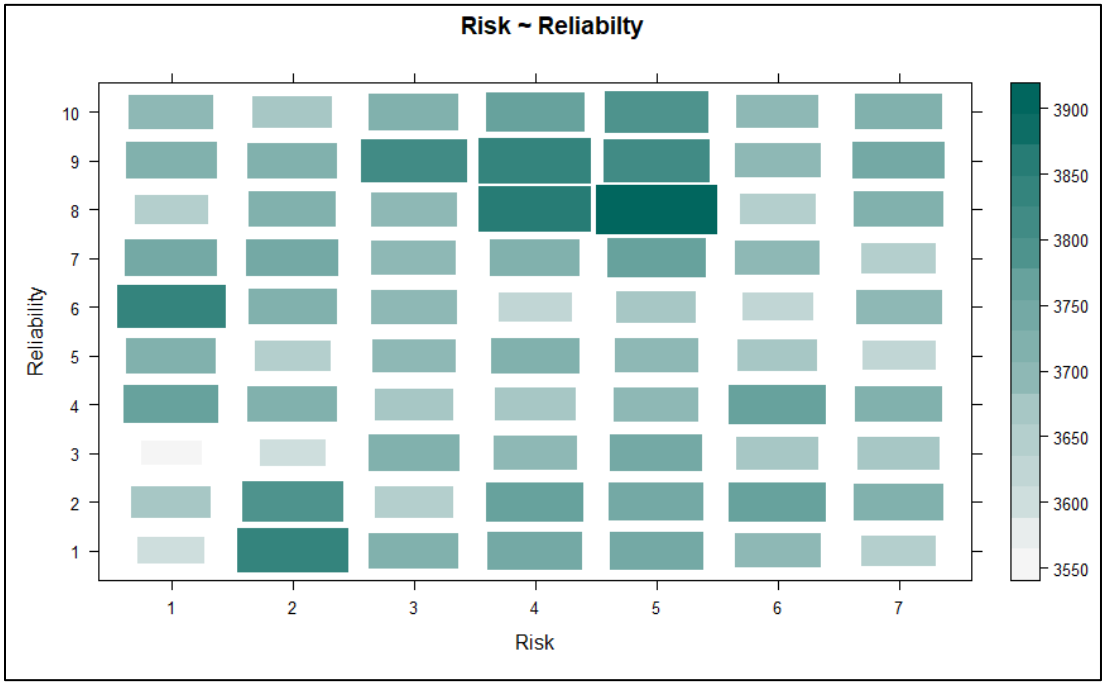


Listing 3-21: The R code produces the levelplot in figure 3 and shows two things. First, generating sample random data from the data set we can see how the plot reach to different data. Second, show how helpful are the different color to identify values.

```
#Listing 3-21

# require object: av (3-4), lattice (3-19)

# generate random samples for risk & reliability and re-run xtab
# starting PRNG from reproducable point
```

```
set.seed(1492) # as it leads to discovery

# generate 260,000 random samples
rel=sample(1:7, 260000, replace=T)
rsk=sample(1:10, 260000, replace=T)

# cast table into data frame
tmp.df = data.frame(table(factor(rsk), factor(rel)))
colnames(tmp.df) <- c("Risk", "Reliability", "Freq")
levelplot(Freq~Reliability*Risk, data=tmp.df, main="Risk ~ Reliabilty",
      ylab="Reliability", xlab = "Risk", shrink = c(0.5, 1),
      col.regions = colorRampPalette(c("#F5F5F5", "#01665E"))(20))
```

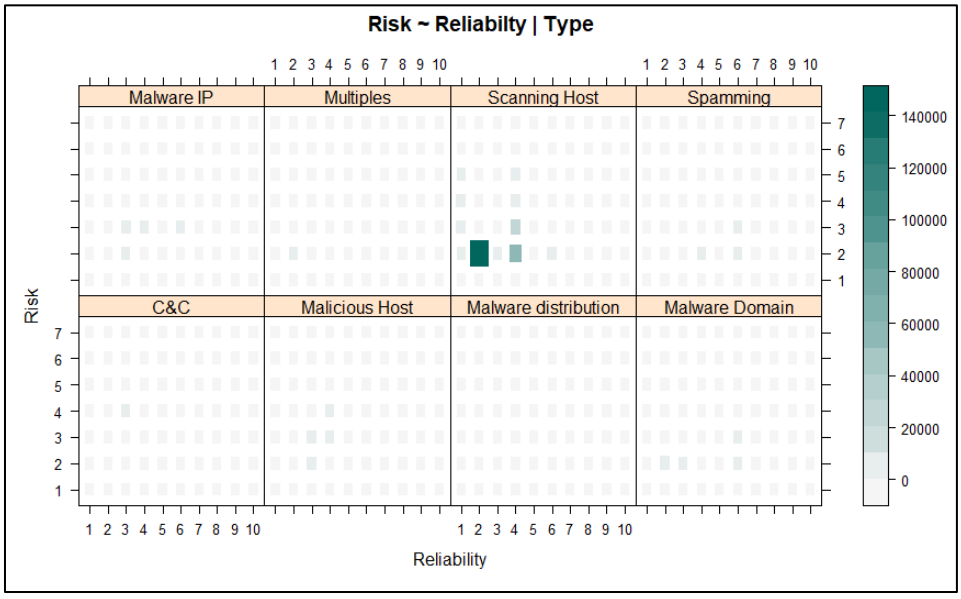
Figure 3: Listing 3-21 output | Heat map for contingency table for risk-reliability sample



Listing 3-22: The R code produces the three-way contingency table lattice graph in Figure 4, enabling you to visually compare the amount of impact Type has on the Risk and Reliability classifications.

```
# require object: av (3-4), lattice (3-19)

# Create a new varible called "simpletype"
# replacing mutiple categories with label of "Multiples"

av$simpletype <- as.character(av$Type)


# Group all nodes with mutiple categories into a new category

av$simpletype[grep(';', av$simpletype)] <- "Multiples"


# Turn it into a factor again

av$simpletype <- factor(av$simpletype)

rrt.df = data.frame(table(av$Risk, av$Reliability, av$simpletype))

colnames(rrt.df) <- c("Risk", "Reliability", "simpletype", "Freq")

levelplot(Freq ~ Reliability*Risk|simpletype, data =rrt.df,

      main="Risk ~ Reliabilty | Type", ylab = "Risk",

      xlab = "Reliability", shrink = c(0.5, 1),

      col.regions = colorRampPalette(c("#F5F5F5","#01665E"))(20))
```

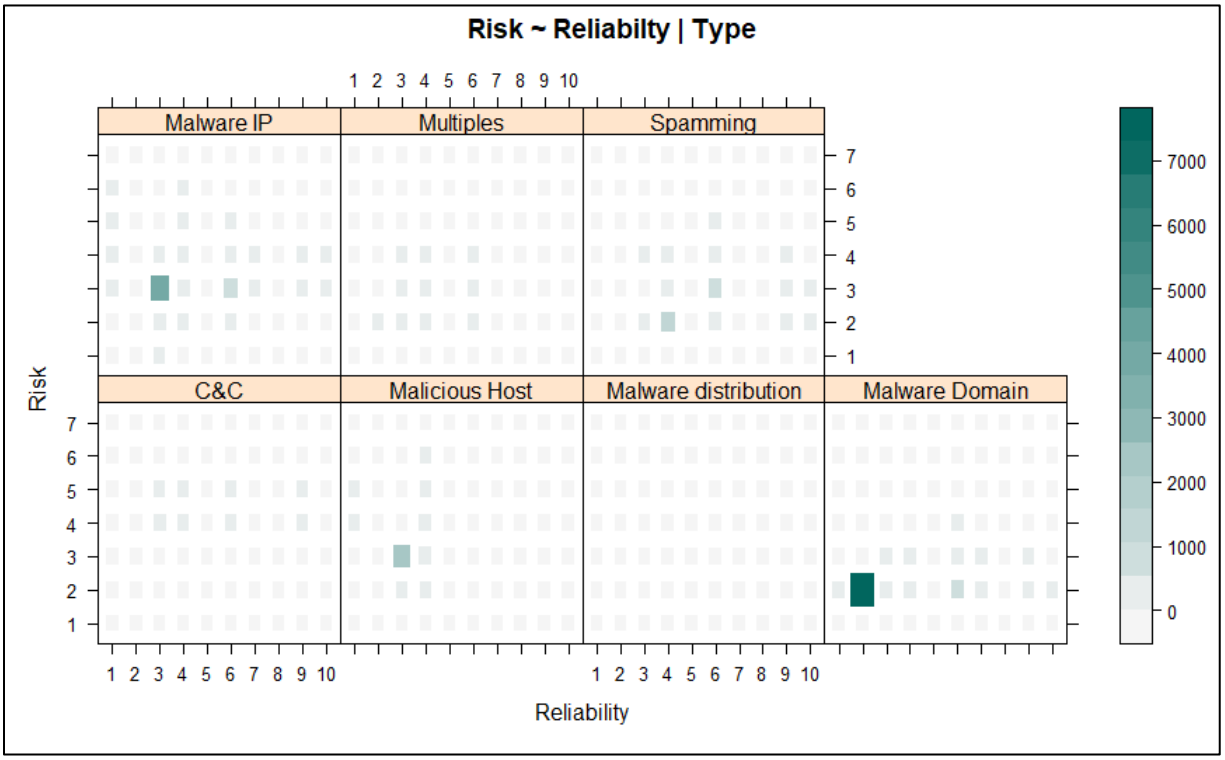
Figure 4: Listing 3-22 | Heat map for contingency table for risk-reliability within multiple factors



Listing 3-24: The main idea is to filter out “Scanning Host” because it has most of the outliers in the dataset. Thus, taking out the attribute will show new nodes outliers in other categories. See figure 5 for reference.

```
#Listing 3-24

# require object: av (3-4), lattice (3-19)

# from the existing rrt.df, filter out 'Scanning Host'


rrt.df <- subset(rrt.df, simpletype != "Scanning Host")

levelplot(Freq ~ Reliability*Risk|simpletype, data =rrt.df,

      main="Risk ~ Reliabilty | Type", ylab = "Risk",

      xlab = "Reliability", shrink = c(0.5, 1),

      col.regions = colorRampPalette(c("#F5F5F5","#01665E"))(20))
```

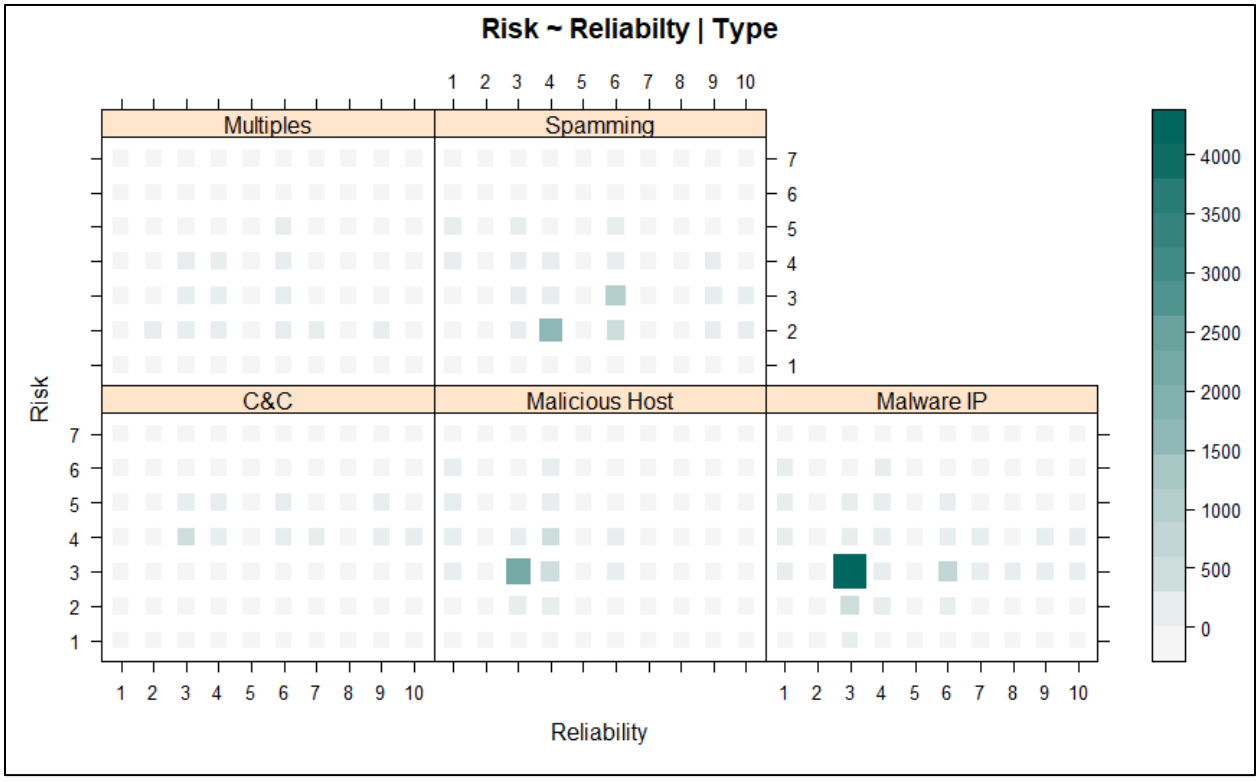
Figure 5: Listing 3-24 | Heat map for contingency table for risk-reliability within multiple factors excluding “Scanning Host"



Listing 3-26: The main idea is to filter out “Malware distribution” and “Malware Domain” because it has most of the outliers in the dataset. Thus, taking out the attribute will show new nodes outliers in other categories. See figure 6 for reference.

```
# require object: av (3-4), lattice (3-19), rrt.df (3-24)

rrt.df = subset(rrt.df,

            !(simpletype %in% c("Malware distribution",

                                "Malware Domain")))

sprintf("Count: %d; Percent: %2.1f%%",

      sum(rrt.df$Freq),

      100*sum(rrt.df$Freq)/nrow(av))


levelplot(Freq ~ Reliability*Risk|simpletype, data =rrt.df,

      main="Risk ~ Reliabilty | Type", ylab = "Risk",

      xlab = "Reliability", shrink = c(0.5, 1),

      col.regions = colorRampPalette(c("#F5F5F5","#01665E"))(20))
```

Figure 6: Listing 3-26 | Heat map for contingency table for risk-reliability within multiple factors excluding “Scanning Host", “Malware distribution”, and “Malware Domain.”



Analyzing Data with Python code

Listing 3-20: It consist in creating a contingency table to show the relationship between risk and reliability from the previous dataset in assignment 2. In this case, we must initiate the .data file and assign its columns.

```
import pandas as pd
import matplotlib.pyplot as plt


#open the file

av = pd.read_csv("C:\\Users\\luisa\\Downloads\\ch03\\ch03\\data\\reputation.data",sep="#")

# make smarter column names

av.columns = ["IP","Reliability","Risk","Type","Country",

"Locale","Coords","x"]
```

Compute a contingency table for risk and reliability factors at (x ,y) location. Then, graph a heat map based on the contingency table to visualize the most significant relations. See figure 7 for contingency table reference and figure 8 for listing 3-20 output.

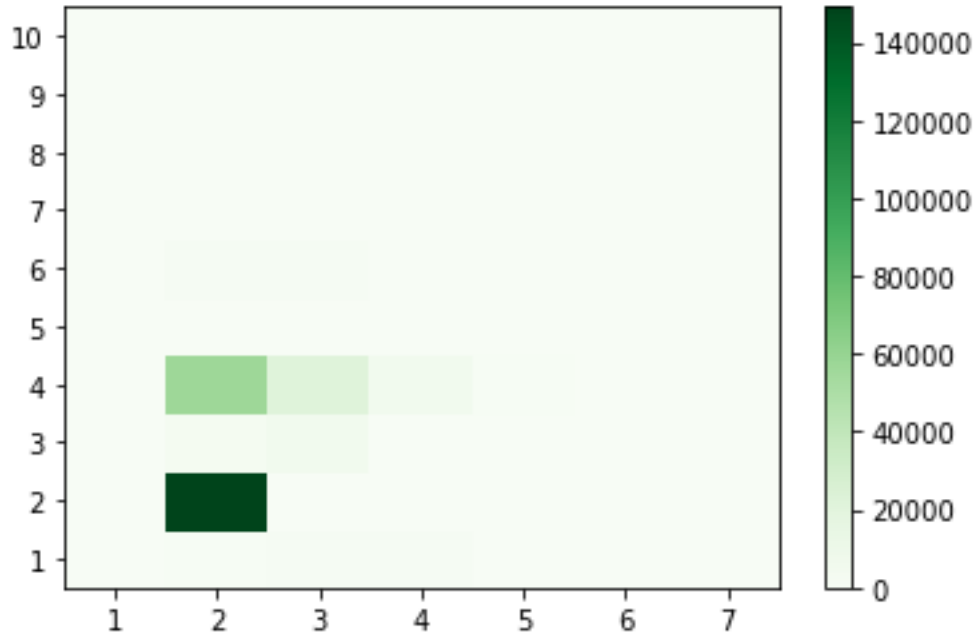
```
from matplotlib import cm

from numpy import arange

pd.crosstab(av['Risk'], av['Reliability'])

print(pd.crosstab(av['Risk'], av['Reliability']))
```

Figure 7: contingency table raw data

```
258622   216.99.159.169            3  ...  34.0115013123,-117.853500366  11
258623   216.99.159.176            3  ...  34.0115013123,-117.853500366  11
258624   216.99.159.117            3  ...  34.0115013123,-117.853500366  11

[258625 rows x 8 columns]
Reliability    1       2     3      4   5     6    7   8    9   10
Risk
1              0       0    16      7   0     8    8   0    0   0
2            804  149114  3670  57652   4  2084   85  11  345  82
3           2225       3  6668  22168   2  2151  156   7  260  79
4           2129       0   481   6447   0   404   43   2   58  24
5            432       0    55    700   1   103    5   1   20  11
6             19       0     2     60   0     8    0   0    1   0
7              3       0     0      5   0     0    0   0    2   0
rel                    1                              2      3                    4
5       6                        7                 8           9                   10
Multiples
rsk                    2     3     4    5   6  7       2  3   1     2     3    4   5  6  1      2
3     4    5   6  7  2  3  5  1    2    3    4   5  6  1   2    3   4  5  2  3  4  5    2    3   4
```

```
# graphical view of contingency table (swapping risk/reliability)

xtab = pd.crosstab(av['Reliability'], av['Risk'],)

plt.pcolor(xtab,cmap=cm.Greens)

plt.yticks(arange(0.5,len(xtab.index), 1),xtab.index)

plt.xticks(arange(0.5,len(xtab.columns), 1),xtab.columns)

plt.colorbar()
```

Figure 8: Listing 3-20 output | Heat map for contingency table for risk-reliability



Listing 3-23: The python code produces the three-way contingency table lattice graph in figure 10, enabling you to visually compare the amount of impact Type has on the Risk and Reliability classifications in a simple bar chart. See figure 11 for reference.

```
# compute contingency table for Risk/Reliability factors which
# produces a matrix of counts of rows that have attributes at
# create new column as a copy of Type column

av['newtype'] = av['Type']


# replace multi-Type entries with Multiples

av[av['newtype'].str.contains(";")] = "Multiples"
```

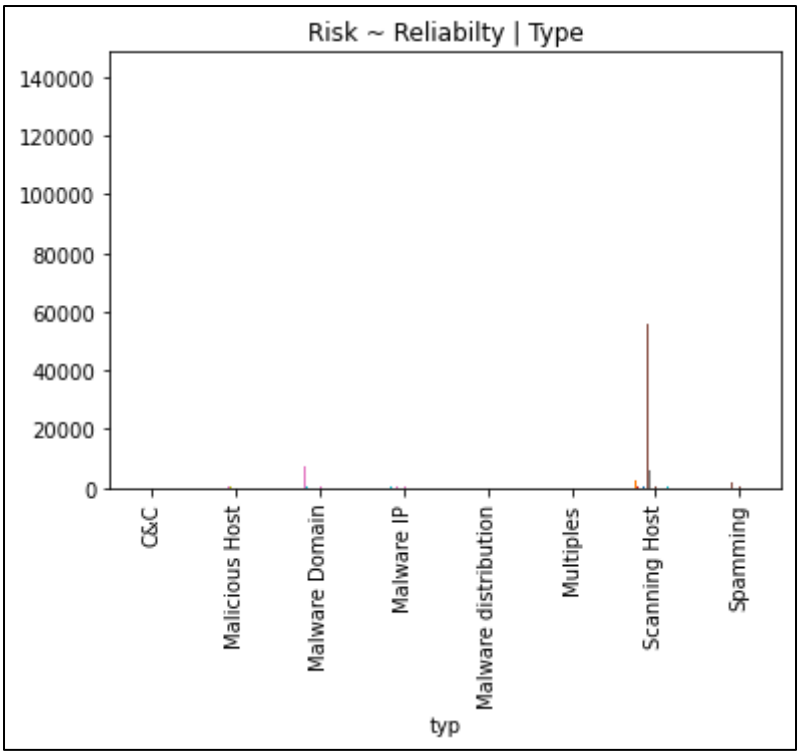
```
# setup new crosstab structures

typ = av['newtype']

rel = av['Reliability']

rsk = av['Risk']


# compute crosstab making it split on the
# new type column

xtab = pd.crosstab(typ, [ rel, rsk ],

rownames=['typ'], colnames=['rel', 'rsk'])


# the following print statement will show a huge text
# representation of the contingency table. Then, we graph
# the plot

print (xtab.to_string())
```

Figure 10: contingency table raw data for listing 3-23

```
Console 3/A
rel                   1                          2     3                      4
5       6                    7              8           9                   10
Multiples
rsk                   2     3     4    5   6  7        2  3   1    2     3    4   5  6  1      2
3    4    5   6  7  2  3  5  1    2    3    4   5  6  1   2    3   4  5  2  3  4  5    2    3   4   5
6  7   2   3   4  5 Multiples
typ
C&C                   0     0     1    2   1  0        0  0   0    0     0  313  22  2  0      0
0    15   22   4  1  0  0  1  0    0    0   98  60  5  0   0    0   7  3  0  0  1  1    0    0  19  16
1  1   0   1   8  5        0
Malicious Host        0     6    51   41   8  1        0  0   1  206  2250    7   2  0  0    152
512   336  138  30  2  1  0  0  1    3    8    8   4  0  0   0    0   0  0  0  0  0  0    0    2   0   0
0  0   0   0   0  0        0
Malware Domain       12     1     0    0   0  0     7309  0   2  246    55    2   1  0  0     60
18     2    0   0  0  2  1  0  2  921  273   26   2  0  3  72   13   0  0  7  1  1  0  135   38   6   0
0  0  54   7   2  0        0
Malware IP            0    23    11   15  10  2        0  3  12  415  4091   71   6  0  1    132
205   122   45  13  2  0  1  0  3   10  793  133  11  3  5   0  140  35  0  0  6  0  0    1   74  10   0
0  0   0  53  11  2        0
Malware distribution  0     0     0    0   0  0        0  0   0    0     1    0   0  0  0      0
0     0    0   0  0  0  0  0  0    0    0    0   0  0  0   0    0   0  0  0  0  0  0    0    0   0   0
0  0   0   0   0  0        0
Multiples             0     0     0    0   0  0        0  0   0    0     0    0   0  0  0      0
0     0    0   0  0  0  0  0  0    0    0    0   0  0  0   0    0   0  0  0  0  0  0    0    0   0   0
0  0   0   0   0  0      834
Scanning Host       790  2189  2056  366   0  0   141543  0   1 2685   159   35  13  0  6  55653
21325  5931  488  13  0  1  0  0  2  611  107   23   1  0  0   0    0   0  0  2  0  0  0  150   22   7
0  0  0   0   0   0  0        0
Spamming              1     2     9    7   0  0        1  0   0   22     9   17   6  0  0   1536
40    21    4   0  0  0  0  0  0  512  931  106  17  0  0   4    1   0  2  1  0  0  0   52  120  15   3
0  0  24  17   3  4        0
Count: 15171; Percent: 5.9%
I Luis Barrios certify that I did this work by myself
```

```
xtab.plot(kind='bar',legend=False,

title="Risk ~ Reliabilty | Type").grid(False)
```

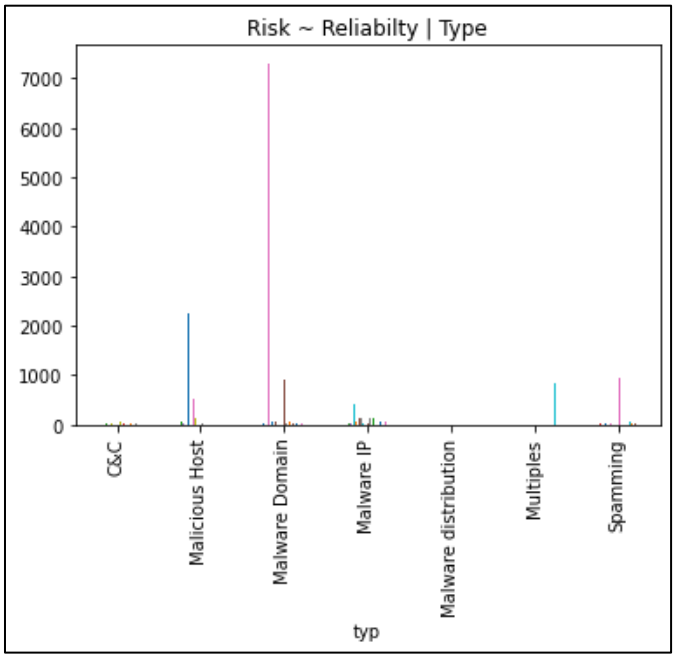
Figure 11: Listing 3-23 | line chart for contingency table for risk-reliability within multiple factors.



Listing 3-25: The main idea is to filter out “Scanning Host” because it has most of the outliers in the dataset. Thus, taking out the attribute will show new nodes outliers in other categories. See figure 12 for reference.

```
# filter out all "Scanning Hosts"

rrt_df = av[av['newtype'] != "Scanning Host"]

typ = rrt_df['newtype']

rel = rrt_df['Reliability']

rsk = rrt_df['Risk']

xtab = pd.crosstab(typ, [ rel, rsk ],

rownames=['typ'], colnames=['rel', 'rsk'])

xtab.plot(kind='bar',legend=False,

title="Risk ~ Reliabilty | Type").grid(False)
```

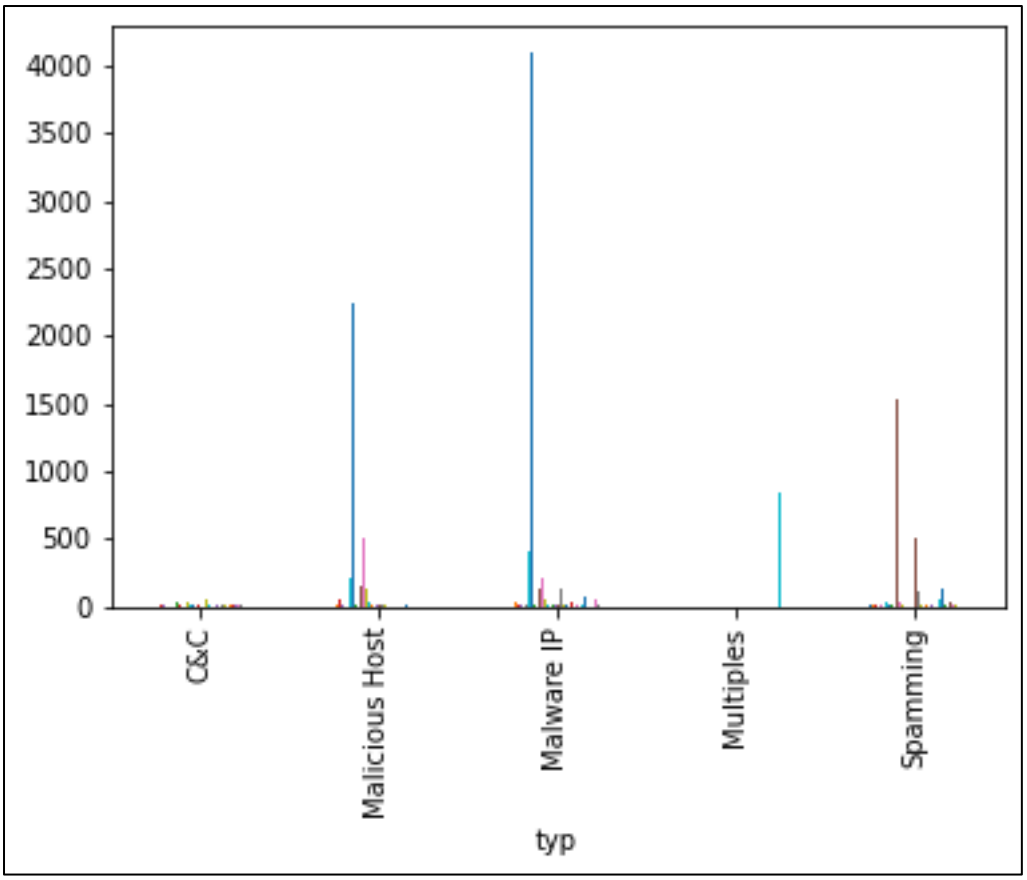
Figure 12: Listing 3-25 | bar chart for contingency table for risk-reliability within multiple factors excluding “Scanning Host



Listing 3-27: The main idea is to filter out “Malware distribution” and “Malware Domain” because it has most of the outliers in the dataset. Thus, taking out the attribute will show new nodes outliers in other categories. See figure 13 for reference.

```
# filter out all "Malware distribution" & "Malware Domain" ]

rrt_df = rrt_df[rrt_df['newtype'] != "Malware distribution" ]

rrt_df = rrt_df[rrt_df['newtype'] != "Malware Domain" ]

typ = rrt_df['newtype']

rel = rrt_df['Reliability']

rsk = rrt_df['Risk']

xtab = pd.crosstab(typ, [ rel, rsk ],

rownames=['typ'], colnames=['rel', 'rsk'])

print ("Count: %d; Percent: %2.1f%%" % (len(rrt_df), (float(len(rrt_df))

/ len(av)) * 100))

xtab.plot(kind='bar',legend=False)
```

Figure 13: Listing 3-27 | Heat map for contingency table for risk-reliability within multiple factors excluding “Scanning Host", “Malware distribution”, and “Malware Domain.”



Appendix

Figure 14: Work signature in Rstudio

```
> # require object: av (3-4), lattice (3-19), rrt.df (3-24)
>
>
> rrt.df = subset(rrt.df,
+
+                 !(simpletype %in% c("Malware distribution",
+
+                                     "Malware Domain")))
>
> sprintf("Count: %d; Percent: %2.1f%%",
+
+         sum(rrt.df$Freq),
+
+         100*sum(rrt.df$Freq)/nrow(av))
[1] "Count: 15171; Percent: 5.9%"
>
>
> levelplot(Freq ~ Reliability*Risk|simpletype, data =rrt.df,
+
+           main="Risk ~ Reliabilty | Type", ylab = "Risk",
+
+           xlab = "Reliability", shrink = c(0.5, 1),
+
+           col.regions = colorRampPalette(c("#F5F5F5","#01665E"))(20))
>
> luis = 'I Luis Barrios certify that I did this work on assigment 4 by myself'
> print(luis)
[1] "I Luis Barrios certify that I did this work on assigment 4 by myself"
>
```

Figure 15: Work signature in Python using Spyder