Analytics Data Science Task Sheet

1 K- anonymity analysis

```
#Loading the relevant libraries
library(matrixStats)
library(tidyverse)
library(plyr)
```

A) Randomly generate a dataset (dataframe) with eight columns and 50,000 rows. Each column should be a categorical variable (of arbitrary name) with three levels (of arbitrary names) in approximately equal proportions.

```
#Used S,T,U,V,W,X,y, and Z as the column headings, and A,B, and C for the levels.
set.seed(117)
column_names <- c(LETTERS[19:26])
df = data.frame(replicate(8,sample(LETTERS[1:3],50000,rep=TRUE)))
colnames(df) <- c(column_names)
head(df)
```

```
##   S T U V W X Y Z
## 1 C A B A C C C A
## 2 B B B C A A C B
## 3 C B B A A A C C
## 4 B A B A B C C A
## 5 B C A B C A A C
## 6 C A C B A C B C
```

B) Verify that the proportions of each value are similar for each of the eight columns.

```
#creating the frequency table:
proportions <- as.data.frame(lapply(df[, column_names], table))
proportions <- select(proportions, -contains("Var")) # removing the duplicated "Var" columns
rownames(proportions) <- LETTERS[1:3]
proportions # look equal from an initial glance, perform formal testing to be sure.
```

```
##   S.Freq T.Freq U.Freq V.Freq W.Freq X.Freq Y.Freq Z.Freq
## A  16648  16738  16560  16766  16607  16541  16637  16698
## B  16694  16634  16712  16647  16788  16674  16606  16630
## C  16658  16628  16728  16587  16605  16785  16757  16672
```

Verifying proportionality: As we have three levels to our categorical variables and expect approximately equal proportions, we should see A, B, and C representing ~33% of values in each column, respectively. Therefore, we can run a chi-square goodness of fit test on each of the columns and determine if there is a statistically significant difference between the actual and expected proportions of A, B, and C. As demonstrated by the output of the code below, none of the columns showed statistically significant differences between the proportions of A, B, and C

```
percentages <- (proportions/50000)*100 # calculating the percentages of A,B,and C in
#each column. All A,B, and C values across all column are ~33%.

#Looping the chi-squared goodness of fit test for each column, none of the p-values are
#significant (p<0.05). Therefore, none of the proportions of A,B, and C are statistically
#significantly different.


for(i in 1:ncol(percentages)){
res <- chisq.test(proportions[,i], p = c(1/3, 1/3, 1/3))
print( res$p.value) # prints the p-values from columns 1 to 8.
#print(res) # uncomment this to print the full stats from each test.
}
```

```
## [1] 0.9654896
## [1] 0.7949151
## [1] 0.5969989
## [1] 0.6077328
## [1] 0.5155402
## [1] 0.4084197
## [1] 0.6827545
## [1] 0.9317973
```

C) How many unique rows (i.e., permutations of category levels) are possible?

```
# 3 Choices (levels) - A,B, and C
# 8 Variables - S,T,U,V,W,X,Y,Z

3^8 #6561 unique rows are possible with 8 columns and 3 variables.
```

```
## [1] 6561
```

D) Produce a table and appropriate graph which show the frequencies (numbers of groups) by permutation group sizes up to group size of 12.

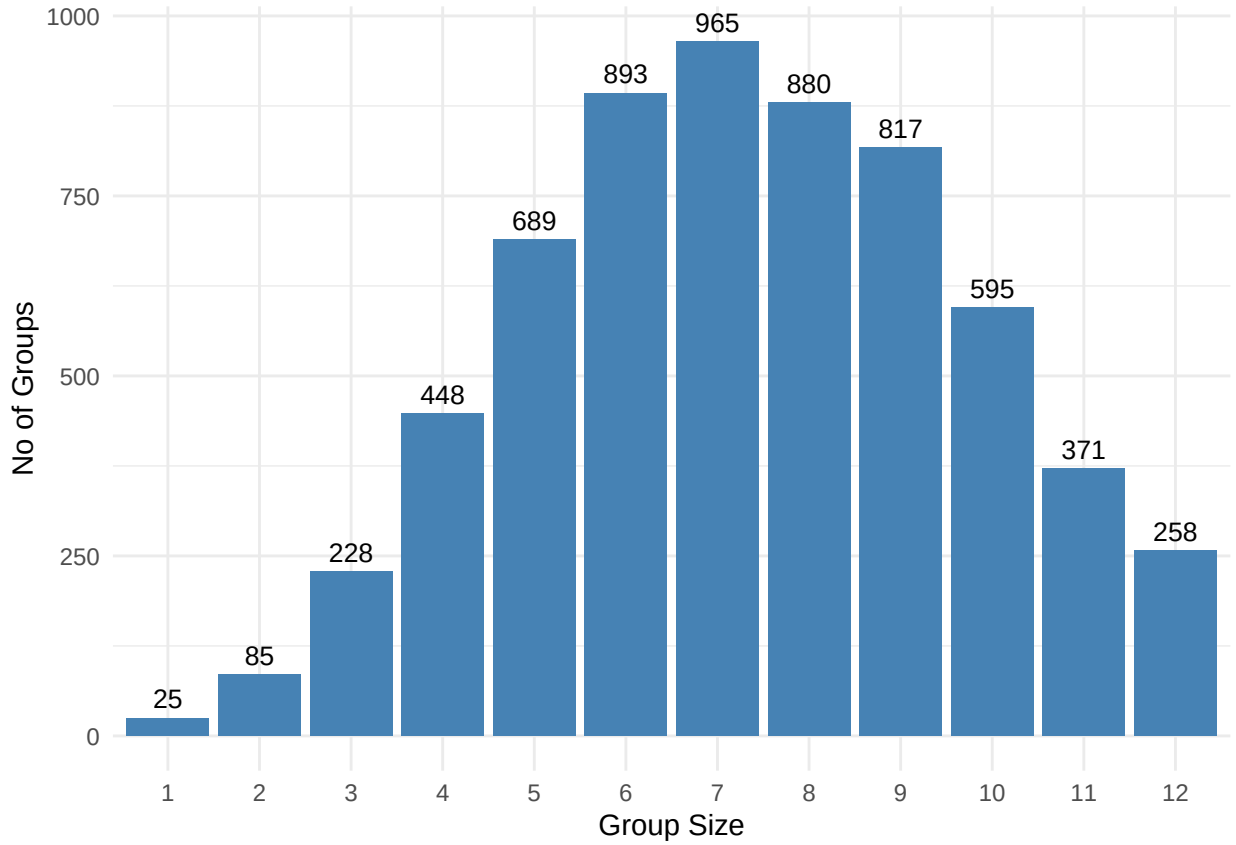
```
#Creating the frequency table
freq_table <- as.data.frame(table(df))
freq_table <- subset(freq_table, Freq > 0 & Freq <=12) # we want a maximum group size of
#12, and no groups where the frequency was 0

final_freq_table <- as.data.frame(table(freq_table$Freq))
colnames(final_freq_table) <- c("Group Size", "No of Groups")
as.data.frame(final_freq_table)
```

```
##    Group Size No of Groups
## 1           1           25
## 2           2           85
## 3           3          228
## 4           4          448
## 5           5          689
## 6           6          893
## 7           7          965
## 8           8          880
## 9           9          817
## 10         10          595
## 11         11          371
## 12         12          258
```

```r
#creating the bar plot to visualize the number of groups with particular group sizes:

ggplot(final_freq_table, aes(x=`Group Size`, y=`No of Groups`)) +
  geom_bar(stat = "identity", fill="steelblue")+
  geom_text(aes(label=`No of Groups`), vjust=-.5, color="black", size=3.5)+ theme_minimal()
```

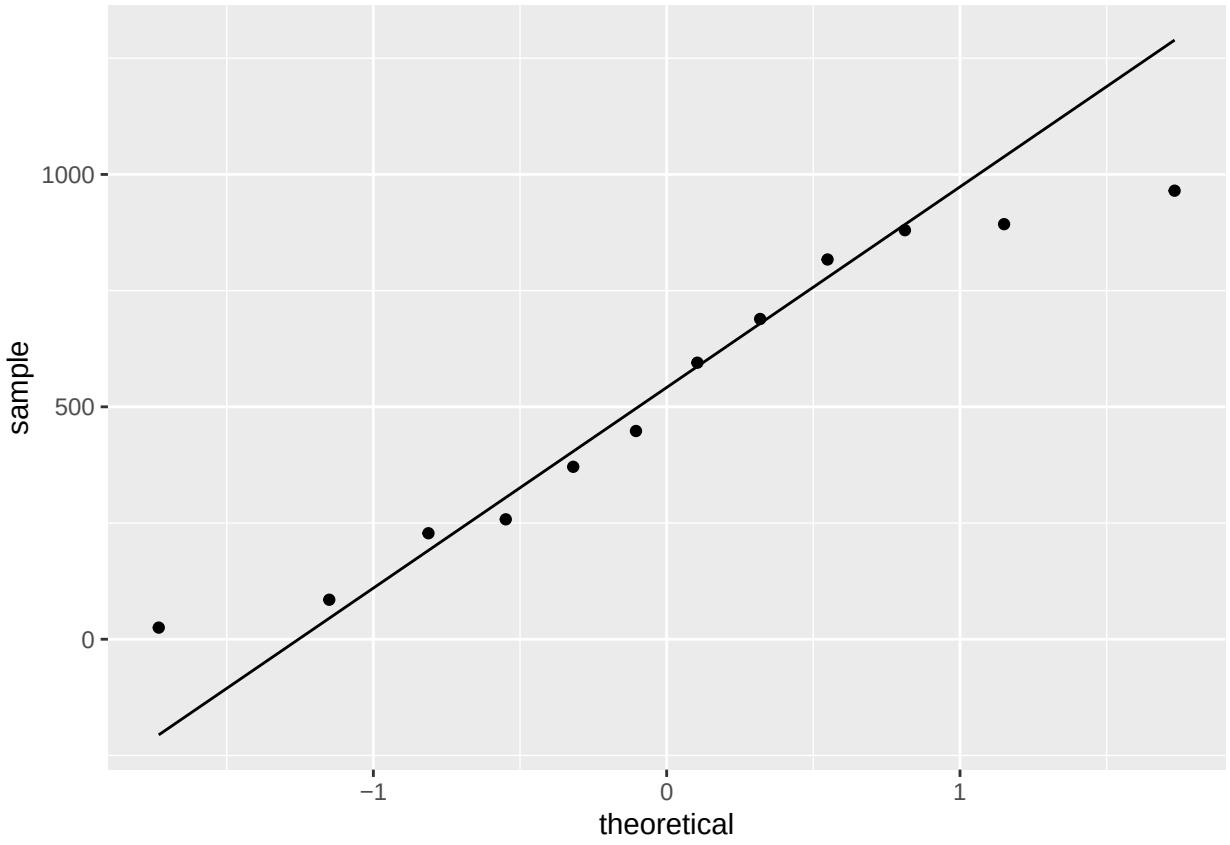


E) Comment upon the distribution of group sizes in d.

The distributions appear to be approximately normal, with a slight left-skew. The normality of the distribution could be further tested with a quantile-quantile plot (qq-plot), which involves randomly generating data from a theoretical, normal distribution with the same mean and standard deviation, by creating a scatterplot

of the two sets of quantiles (ours vs the theoretical one) against one another. If both sets of quantiles are from the same distribution, we should see the points forming a roughly straight line.

```
# can run this to test if the data is normally distributed
ggplot(final_freq_table, aes(sample=`No of Groups`))+stat_qq() +stat_qq_line()
```



F) If your random variables were, in fact, meaningful information on individuals, which group sizes are of most concern from a privacy perspective?

If the random variables were meaningful information on individuals and assuming each row represented an individual, the group sizes of most significant concern are small. For instance, our smallest Group Size of 1 (i.e. only one instance of a row with a particular combination of the random variables) has 25 groups (rows). Because the groups are a unique combination of our 8 bits of meaningful information on our individuals, anyone who obtained these random variable values for Group Size = 1, could in theory, identify the individual it corresponded to. Conversely, even if someone identified the exact combination of the 8 random variables for a sequence in the Group Size = 12, they would still have 12 individuals who this data could potentially correspond to.

G) Consider the effect of missing data in the data set you created in Part a). How might this complicate the production of a frequency table of group sizes in Part d)?

The effect of the missing data would be partially dependent on if rows with missing data were included or excluded. As good practice, any rows with missing data would generally be excluded before analysis.

However, this could lead to an erroneous or skewed frequency table/distribution, especially if the data is missing in a non-random way (e.g. Random Variable 8 is missing more often in rows with particular values for their other random variables). On the other hand, if the data is missing completely at random, then although this will reduce the sample size, it shouldn't fundamentally impact the proportions.

H) Imagine the code that you wrote for Part d) was to be deployed in an automated system that customers could use independently, on potentially large volumes of their own data. Describe how you might deploy the code, and what additional considerations you might have or any changes to the code you might make.

With respect to directly deploying the code in d from this notebook, the code could be directly placed in a separate R notebook file, converted to an R markdown file, and published to a database such as Rstudio Connect. If deploying/ productionizing the code for a customer to use independently and on potentially large volumes of their own data, there are several considerations. Firstly, assuming only the code in part d was deployed, I would include the installation and loading of the libraries used (ggplot2 and dplyr) at the beginning of the code to ensure the relevant packages were ready prior to the code being run. Although the code in part d only worked with a frequency table generated from a data frame with eight columns and 50,000 rows, I would also test the code using a much larger data frame to create a frequency table with larger group sizes to test runtimes and to mimic the large volumes of data X's company may use. I would also tentatively implement some code that would basic clean their data (i.e. removal of data with missing rows) at the beginning of the code, to prevent the effect of missing data touched upon in question 1G. Lastly, and if possible, I would upload the code to an internal GitHub (or equivalent) to allow other staff to run it, helping to expose and thereby solve any bugs or errors.

2 Postcodes and Privacy

```
postcodes <- read.csv("C:\\Users\\Kai\\Downloads\\Postcode_Estimates_Table_1.csv")

summary(postcodes) # quick exploration of the data

mean(postcodes$Total)/mean(postcodes$Occupied_Households) #The average number of people
#per household  is ~2.4. this can be combined with UCL data to determine population
#size dependent on postcode specificity.

one_household <- subset(postcodes, Occupied_Households == 1)
length(unique(one_household$Postcode)) #59,777 houses in which a full postcode only
#contains 1 house. Allowing potentially easy identification a patient based on
#spatial information.

postcodes$Outward_code<- gsub(" .*","", postcodes$Postcode)# only keeps the outward part
#of the postcode, (e.g. LS4 3RL would become LS4).Some postcodes don't have a space between
#them, meaning the entire postcode is kept when the above command is run, preventing group
#of these under a single postal district.*


#Are there any postcodes which are entirely unique up the postal district:

unique_district_codes<-subset(postcodes, nchar(as.character(Outward_code)) <= 4
```

```
                                & Occupied_Households ==1 )
# Only 1 household, and makings sure only postal districts are included given the above issue*

#It doesn't appear there are any cases where a single household corresponds to a unique
#postal district, indicating there are no cases where a household could be directly identifiable
#by anonymization to district level.

district_code_populations<-subset(postcodes, nchar(as.character(Outward_code)) <= 4)
district_code_populations<- district_code_populations[,c(2,6)] #remove specific postcode

district_code_populations<- aggregate(Total ~ Outward_code, district_code_populations, sum)
#shows some district codes for example have very small populations
#(e.g. N1C only has two full postcodes, 3 houses and 7 people total. )

small_district_code_populations <- subset(district_code_populations, Total <= 4000)
#districts with fewer than 4000 people, based on scaling provided below.
```

In the UK, postcodes are either composed of between 6-8 characters (including a space). Each postcode is divided into two parts separated by a single space: the outward code and the inward code, respectively. The outward code includes the postcode area and the postcode district, respectively. The inward code includes the postcode sector and the postcode unit. For instance, with the postcode "SW1W 0NY", SW1W is the outward code, SW is the postcode area and 1W postcode district. The inward code is 0NY, with 0 as the postcode sector and NY as the postcode unit. When considering the relative number of households represented by the inclusion of each postcode character, the University of College London Center for advanced spatial analysis predicts the number of properties identifiable with differing postcode specificity as follows:

- **full postcode = approx 15 households**
- **postcode minus the last digit = approx 120/200 households**
- **postal sector = 4 outward digits + 1 inbound gives approx 2,600 households**
- **postal district = 4 outward digits approx 8,600 households**
- **postal area = 2 outward digits approx 194,000 households**

From the initial data analysis, 59,777 postcodes only contain one household, meaning individuals at these addresses would be particularly vulnerable to a privacy breach if the postcode was identified. Although the US postal system (ZIP codes) is slightly different, a similar process of removing the inward portion, or last 3-digits (e.g. LS4 3RL -> LS4) still yields no households that could be directly identified.

Although the number of households doesn't necessarily represent the number of people for each postcode level, based on the postcode data provided, the average number of people per household is approximately 2.4 based on postcode data. Multiplying our the number of households by this, we find that postal districts (e.g. SW1W) should contain, on average, approximately 20,640 individuals, with greater detail than this, meaning fewer people per postcode and more specificity. Based on the American system of grouping together codes containing 20,000 people or fewer, an argument could be made for lumping together postcodes more specific than the postal district in the UK. However, an argument could be made that this might reduce the granularity and accuracy of data (in a research context, for example). This is especially true when considering the UK population is approximately 1/5th the size of the US Population (67.2 million vs 329.5 million). In this regard, scaling down the population size from ~20,000 to something closer to ~4000 and combining these under a new postcode may be more realistic. In this regard, anonymization to the postal

sector (SW1W 0XX) may create balance between anonymity and accuracy. However, even if this scaling is applied, 43 districts with fewer than 4000 people would arguably need combining under a shared code.

Beyond implementing the anonymization process, An alternative approach that may produce more useful data and avoid the problems of inaccuracy and misinterpretation created by partial disclosure of postcodes would be to use 'replacement' postcodes in place of real ones. This could allow whoever is accessing the data to retain the accuracy of data whilst reducing identification risk.

Link to UCL advanced spatial analysis centre data: https://ico.org.uk/media/1061/anonymisation-code.pdf