1. In the context of car insurances, image recognition can be used help to assess car damages and estimate costs of repair. This is useful for a company selling said insurances to easily and quickly receive information on the payout regarding the damage done on a policyholder's car without having to send over an employee to take a look at the damages. The costs of repair can also be estimated through the recognition of the car and model using available collected data of damage done and respective costs.

   On the other hand, text recognition helps both customers and the company selling car insurances when personal information is being collected. For the customers, a simple photo of their ID and the driver's license already fills up most of the information needed with less possibility of making mistakes if entered manually. For the company, they benefit from the standardised information collected.

   Combining both technologies, it eases the process of confirming the identity of customers by having customers upload a video of them saying a distinct line. Overall, the tech will significantly improve the communication between the company and the customers directly.

2. Before applying the K-means algorithm to see if it is possible to split the 500 policyholders into subgroups, it is very useful to visualise the health check data available through a scatterplot. Further, the visual will be of use in choosing K number of clusters for the analysis. The data points in the plots of Figure 1 is scaled using Principle Component Analysis (PCA) to reduce the dimensionality of the data.

```
data = as.matrix(read.table("39542q2.txt",header=FALSE,sep=" "))
scaled.data = scale(data[,1:3], scale = T)
pairs(scaled.data)
```

   It is very clear from all the plots in Figure 1 that there doesn't seem to be any grouping that the policyholders can be put into. However, we can still apply the K-means analysis to see if this observation holds true, choosing $K = 3$. An analysis on the total within sum of squares will further contribute in seeing whether the choice of K is appropriate for the dataset. The following R codes are used to produce the plots in Figure 2 of the groups obtained in by the `kmeans()` command in R and the total within group variation values as the number of clusters are increased.

```
par(mfrow=c(2,1), mai = c(0.8, 0.8, 0.5, 0.3))
kmc = kmeans(scaled.data, 3, nstart = 20, algorithm ="Lloyd")
plot(scaled.data, col = kmc$cluster, main="")

maxk = 10
wssvec = numeric(maxk)
for (k in 1:maxk) {
  kmcn = kmeans(scaled.data, k, nstart = 20, algorithm ="Lloyd")
  wssvec[k] = kmcn$tot.withinss
}
plot(1:maxk, wssvec, type="b", xlab="No of clusters, K",
    ylab="Within Group SS", main="")
```

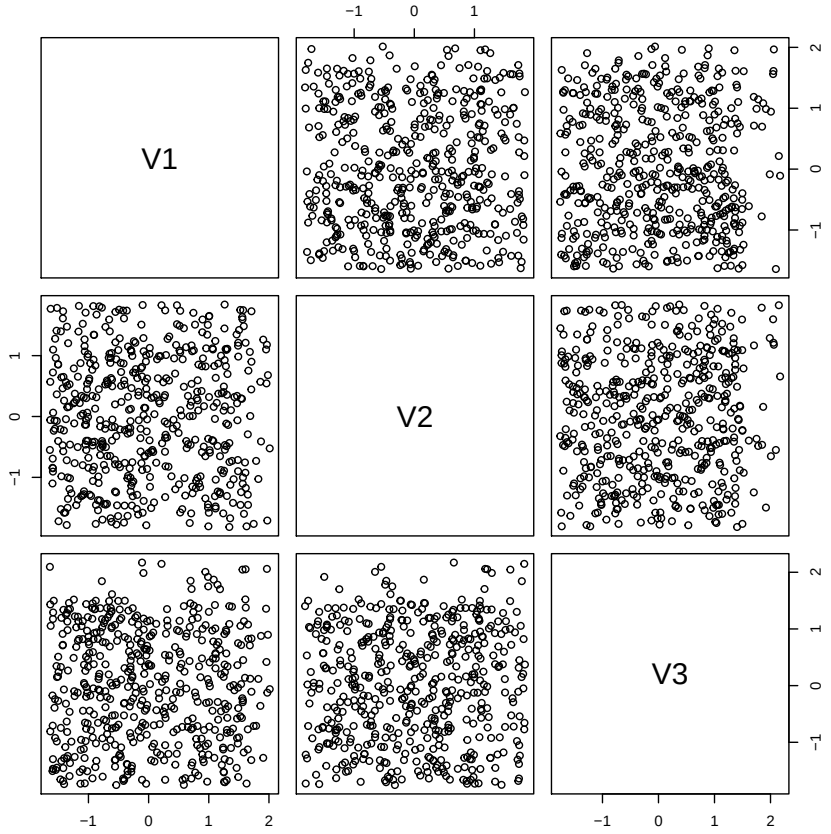

Figure 1: Scatterplot of each variable in dataset against each other

Looking at the first plot in Figure 2, there aren't any clear separations in the three clusters which are categorised by colour, possibly confirming the observation made in the scatterplot earlier. We see that there are many overlaps between the red and the green group as well as the red and the black group. However, it does seem as though there is a clear distinction between the green and the black group, with very few data points being classified into the wrong group. Therefore, it is possible that the K that was chosen for this analysis is not suitable for the dataset provided. To examine this observation, we expect an elbow at $K = 2$ in the second plot of Figure 2. However, the plot has no elbow, it is a smooth curve instead. This tells us that there is most likely no grouping that can be done for the dataset by the K-means analysis. Therefore, this health check data does not give reason to believe that the 500 policyholders can be categorised into groups with distinct health profiles.

3. We know that K-means clustering is a random process that aims to increase the between group variations as well as decrease the within group sum of squares. In theory, the total within sum of squares decrease as the number of clusters are increased, eventually reaching 0 when $K = n$ trivially. However, noting that it is a random process, it is possible that the set

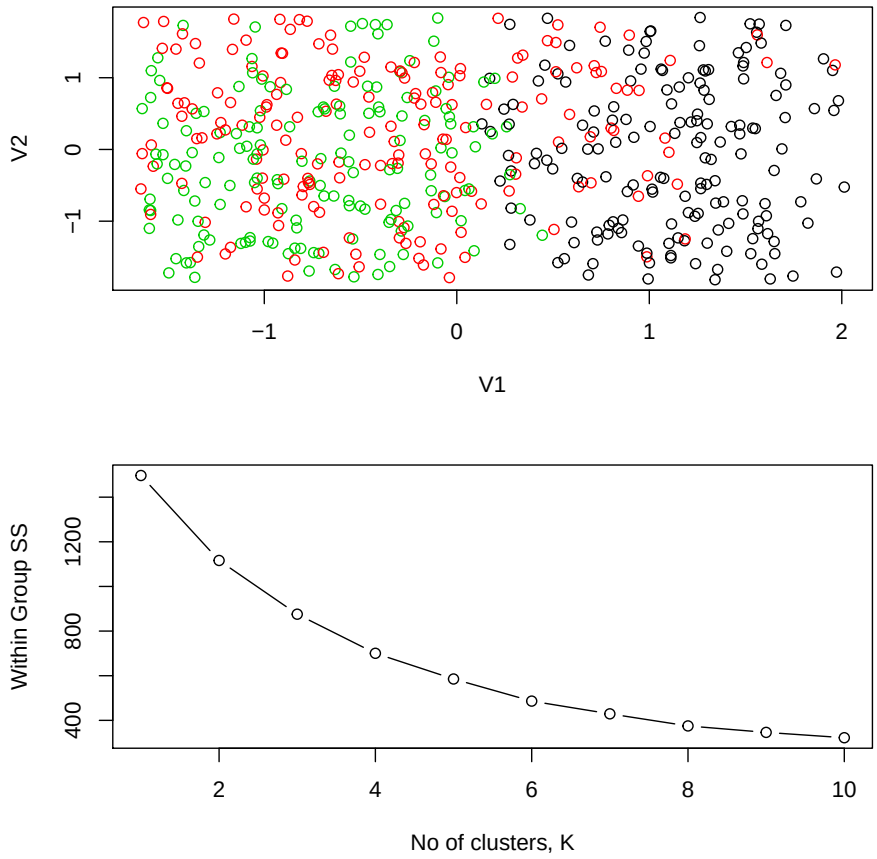

Figure 2: Scatterplot the first two variables in the dataset (top) and plot of within group variation against number of clusters (bottom)

of clusters in the first step of the algorithm for $K = 4$ is poorly chosen, causing the irregularity observed in the figure. A possible cause of this is that the colleague left the argument `nstart` in command `kmeans()` as the default, `nstart` $= 1$. The argument is important in decreasing the likelihood of finding a local minimum instead of a global minimum for the objective function, $J(C_1, C_2, ..., C_K)$ in (2).

$$W(C) = \frac{1}{\text{no of points in C}} \sum_{i,i \in C} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \tag{1}$$

$$J(C_1, C_2, ..., C_K) = \sum_{k=1}^{K} W(C_k) \tag{2}$$

In (1), $W(C)$ the squared Euclidean distance multiplied by a factor of 1 over number of points in cluster C as a correction term for the size C. In `kmeans()`, `nstart` represents how many

times the algorithm is repeated with different random initial clusters to average out the errors due to said randomness. The larger the value that the argument is set at, the lower the aforementioned likelihood. For a balance between allowing an appropriate computation time and lowering the likelihood, `nstart` $= 25$ should sufficient.

4. (a) To produce the plots in Figure 3, the following R codes have been used.

```
par(mfrow = c(2,2), mai = c(0.8, 0.8, 0.5, 0.3))
theta = c(-700, -0.01, 0.01, 700)
n = 5000
for (i in theta) {
  cop = frankCopula(i)
  distr = mvdc(cop, margins = c("gamma","gamma"),
               paramMargins = list(list(2,2), list(2,2))
  )
  samples = rMvdc(n, distr)
  plot(samples ,xlab="x",ylab="y")
}
```

   (b) Based on the visuals in Figure 3,

      i. As $\theta \to -\infty$,
      The relationship between X and Y approaches

$$y = \frac{1}{x}$$

      where as X increases, Y decreases. This shows the negative relationship between the two variables, therefore the limiting copula in this case would be the countermonotonicity copula.

$$C_{\theta \to -\infty}(u, v) = max\{u + v - 1, 0\}$$

      ii. $\theta \to 0$, the points seem to be scattered randomly in both $\theta = -0.01$ and $\theta = 0.01$, implying that the limiting copula should the independence copula.

$$C_{\theta \to 0}(u, v) = uv$$

      iii. As $\theta \to \infty$,
      The relationship between X and Y approaches

$$y = x$$

      which is a perfect dependence. The limiting copula is most likely the comonotonicity copula.

$$C_{\theta \to \infty}(u, v) = min\{u, v\}$$

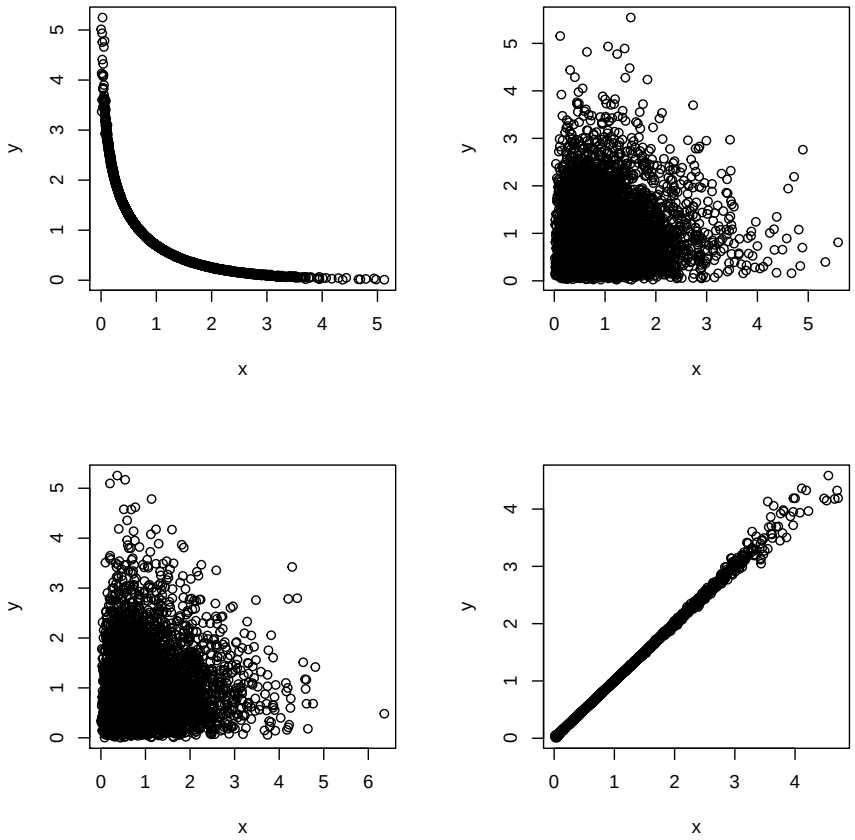

Figure 3: Scatterplots of variable Y against X. Topleft: $\theta = -700$, Topright: $\theta = -0.01$, Bottom left: $\theta = 0.01$, Bottom right: $\theta = 700$