For this Lab, we will use R and KNN to predict survival on the Titanic

1. Download the Titanic dataset from Kaggle: https://www.kaggle.com/c/titanic
2. Open RStudio and load the data:

   2.1. A Open Rstudio on your computer

   Using your Remote Desktop Client log into SoCAppSrv1 & SoCAppSrv2.
   Rstudio is in your applications.

   2.2. B Download and Install R and RStudio

   R: https://cran.r-project.org/bin/windows/base/
   RStudio: https://www.rstudio.com/products/rstudio/download/
   *You cannot use RStudio without a copy of R

   2.3 Open RStudio

   Double click the RStudio logo to start R

   2.4 RStudio Interface

   - Data Pane is Top Left
   - Environment and History Pane is Top Right
   - Command Console is Bottom Left
   - Visualisation, Package and File Pane is Bottom Right

   2.5 Import the data and rename it train (train <- name_of_file)
   ***Use stringsAsFactors = FALSE, this changes continuous to categorical**

3. Create a Data Quality Report:

   https://cran.r-project.org/web/packages/dataQualityR/dataQualityR.pdf

   ```
   install.packages("dataQualityR")
   library(dataQualityR)
   data(train)
   num.file <- paste(tempdir(), "/dq_num.csv", sep= "")
   cat.file <- paste(tempdir(), "/dq_cat.csv", sep= "")
   checkDataQuality(data= crx, out.file.num=num.file, out.file.cat=cat.file)
   ```

   * The file is saved in “/var/folders”, please open it and explore

4. Create visualisations using GGPLOT and other Visualisations tools that:

   Q1 Shows the relationship between continuous variables (Lecture)
   Q2 Shows the relationship between categorical variables (Lecture)
   Q3 Shows the relationship between continuous and categorical variables (Lecture)

5. Install the KNN package - class:

   https://www.rdocumentation.org/packages/DMwR/versions/0.4.1/topics/kNN

   ```
   install.packages(“class”)
   library(class)
   ```

   Remove the first passengerId:

   ```
   train_minus_passengerid <- train[,-1] -> train_minus_passengerid
   ```

   Let’s randomise the data: t

   ```
   train_minus_passengerid <- data[train(1:nrow(train)), ]
   ```

   Split the data into train and test

   ```
   train <- train_minus_passengerid[1:XXXX,]
   ```

test <- train_minus_passengerid[XXXX:XXXXX,]

Train and Test Labels

train_labels <- train_minus_passengerid[1:XXXX, 1]

test_labels <- train_minus_passengerid[XXXX:XXXX, 1]

This implementation of KNN can handle normalisation - the norm arg

pred <- knn(train = train, test = test, cl = train_labels, k=10)

6. Let's evaluate the model: install.packages("gmodels") and use CrossTable

   CrossTable(x = test_labels, y = pred, prop.chisq=FALSE)

   Q4 Paste the results here

7. Let's repeat using Knncat (KNN for categorical) - see below

   Q5 Paste the results here

Some useful bits:

Normalisation Function

normalize <- function(x) { return ((x - min(x)) / (max(x) - min(x))) }

To apply, using lapply on a vector, prc_n <- as.data.frame(lapply(**DATAFRAME**, normalize))

How to randomise a data frame:

data <- data[sample(1:nrow(data)),]

http://www.cookbook-r.com/Manipulating_data/Randomizing_order/

Dealing with categorical features, use Knncat:

https://cran.r-project.org/web/packages/knncat/knncat.pdf