Class Lab 6 –

Visualizing Titanic Data Visualizing Titanic Data with Dave Langer - Comment your code line by line.

The sank of Titanic was known as the deadliest sank of a single ship in 1912. Approximately 1500 passengers out of 2224 died in the tragedy. This project used an incomplete dataset to understand the overall picture.

The dataset had information of 1309 passengers included their ticket-class bought, gender, family size, ticket fare, age, and most importantly, their survivorship. 32% of passengers on the information list did not have their survivorship recorded, there were also many missing values in each of the information group. Data manipulation, imputation, feature engineering, and machine learning algorithms (K-Nearest neightbour, random forest, and extreme-gradient boosting) were applied to clean the dataset. A final, perfectly cleaned dataset was synthesised for data visualisation to understand the trend in the tragedy.

This project concluded that there was 62% of passengers died from the sank, the death rate was the highest in 3rd-class ticket passengers, and the death rate was the highest in adult male. Statistically, 76% of 3rd class ticket passengers, 57% of 2nd class ticket passengers, and 37% of 1st class ticket passengers died from the sank. Among age groups, 47% of children, 57% of teenagers, 65% of adults, and 90% of elders died from the sank of Titanic.

R PACKAGES

INTRODUCTION

RMS Titanic was a British passenger liner that sank in the North Atlantic Ocean on 15 April 1912, after striking an iceberg during her voyage from Southampton to New York City (Wikipedia 2021).

According to Wikipedia, there was an estimate of 2224 passengers and crew aboard, and the sank has caused estimated 1500 of casualty. The sank of Titanic was known at the time one of the deadliest of a single ship.

In this project, I will analyse a Titanic dataset publicly available from Kaggle. The dataset has information of 1309 passengers and their survivorship information (survived, not survived, and missing).

I will apply various data science methodologies include data exploration, data manipulation, feature engineering, algorithmic imputation, and machine learning models to fill up missing values in the dataset including predicting the survival of passengers who had their survivorship data unrecorded. I will then use the final table for data visualisation using graphs and maps to understand the overall trend.

4 DATA PREPARATION

4.1 Data import

Following codes upload the datasets into R.

Code

Following shows a random draw of 10 rows of information from the imported dataset. We can see many information such as Name, Sex, Age, ticket classes, fare and etc.

Code

PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1256	1	Harder, Mrs. George Achilles (Dorothy Annan)	female	25.0	1	0	11765	55.4417	E50	C
397	3	Olsson, Miss. Elina	female	31.0	0	0	350407	7.8542		S
1048	1	Bird, Miss. Ellen	female	29.0	0	0	PC 17483	221.7792	C97	S
836	1	Compton, Miss. Sara Rebecca	female	39.0	1	1	PC 17756	83.1583	E49	C
1207	3	Hagardon, Miss. Kate	female	17.0	0	0	AQ/3. 30631	7.7333		Q

PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
682	1	Hassab, Mr. Hammad	male	27.0	0	0	PC 17572	76.7292	D49	C
705	3	Hansen, Mr. Henrik Juul	male	26.0	1	0	350025	7.8542		S
20	3	Masselmani, Mrs. Fatima	female	NA	0	0	2649	7.2250		C
883	3	Dahlberg, Miss. Gerda Ulrika	female	22.0	0	0	7552	10.5167		S
123	2	Nasser, Mr. Nicholas	male	32.5	1	0	237736	30.0708		C

4.2 Data description

This table is adapted from Kaggle.

Code

Variable	Definition
PassengerId	Id of the passenger
Pclass	Ticket class: 1 = 1st, 2 = 2nd, 3 = 3rd. It is a A proxy for socio-economic status (SES) with 1st = Upper, 2nd = Middle, 3rd = Lower
Name	Name of the passenger
Sex	Sex
Age	Age in years
SibSp	# of siblings / spouses aboard the Titanic

Variable	Definition
Parch	# of parents / children aboard the Titanic
Ticket	Ticket number
Fare	Passenger fare
Cabin	Cabin number
Embarked	Port of Embarkation: C = Cherbourg, Q = Queenstown, S = Southampton
Survived	Survivalship information: 0 = No, 1 = Yes, blank = missing value
Source	train and test. Train has either survived or not survived recorded, whereas test does not. Machine learning will be used to make the prediction.

4.3 Data exploration

There are 1309 rows of observations and 13 columns. There are 6 columns recognised as character type and 7 as numerical type. It is important to change some types into factor during analysis. I will identified columns that need this conversion.

Code

Data summary

Name	titanic
Number of rows	1309
Number of columns	13
_____	
Column type frequency:	
character	6
numeric	7

_____

Data summary

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Name	0	1	12	82	0	1307	0
Sex	0	1	4	6	0	2	0
Ticket	0	1	3	18	0	929	0
Cabin	0	1	0	15	1014	187	0
Embarked	0	1	0	1	2	4	0
source	0	1	4	5	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
PassengerId	0	1.00	655.00	378.02	1.00	328.00	655.00	982.00	1309.00
Pclass	0	1.00	2.29	0.84	1.00	2.00	3.00	3.00	3.00
Age	263	0.80	29.88	14.41	0.17	21.00	28.00	39.00	80.00
SibSp	0	1.00	0.50	1.04	0.00	0.00	0.00	1.00	8.00
Parch	0	1.00	0.39	0.87	0.00	0.00	0.00	0.00	9.00
Fare	1	1.00	33.30	51.76	0.00	7.90	14.45	31.27	512.33
Survived	418	0.68	0.38	0.49	0.00	0.00	0.00	1.00	1.00

I identified that there are 263 missing values from age, 1 from Fare and 418 from survived. However, there are many missing values in “Cabin” as well, the missing values were recorded with a space rather than having a truly blank that would regonised as “NA” in R and be detected.

Code

```
##  [1] ""     "C85"  ""     "C123" ""     ""     "E46"  ""     ""     ""
```

There are 77% of values went missing the column of Cabin, therefore this column will be removed because there are too many missing values in the column. There is a rule of thumb in the market recommending that a column with 60% of missing values and above should be removed during predictive analysis.

Code

```
## # A tibble: 2 x 4
##   value       statistics total percent
##   <chr>            <int> <int> <chr>
## 1 Missing           1014  1309 77%
## 2 Not_Missing        295  1309 23%
```

Following shows another way of looking at the dataset that displays their data types and initial values.

Code

```
## Rows: 1,309
## Columns: 13
## $ PassengerId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,~
## $ Pclass      <int> 3, 1, 3, 1, 3, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2, 3, 2, 3, 3~
## $ Name        <chr> "Braund, Mr. Owen Harris", "Cumings, Mrs. John Bradley (Fl~
## $ Sex         <chr> "male", "female", "female", "female", "male", "male", "mal~
## $ Age         <dbl> 22, 38, 26, 35, 35, NA, 54, 2, 27, 14, 4, 58, 20, 39, 14, ~
## $ SibSp       <int> 1, 1, 0, 1, 0, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0, 4, 0, 1, 0~
## $ Parch       <int> 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0, 1, 0, 0, 0~
## $ Ticket      <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", "113803", "37~
## $ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, 51.8625,~
## $ Cabin       <chr> "", "C85", "", "C123", "", "", "E46", "", "", "", "G6", "C~
## $ Embarked    <chr> "S", "C", "S", "S", "S", "Q", "S", "S", "S", "C", "S", "S"~
## $ Survived    <int> 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1~
## $ source      <chr> "train", "train", "train", "train", "train", "train", "tra~
```

5 DATA CLEANING

Identified cleaning tasks:

- *PasengerId* will be removed, because it adds nothing to the analysis of this project.
- *Name* will be removed, because it adds nothing to the analysis of this project. I am not doing text analytics in this project.
- *Ticket* will be removed, because it adds nothing to the analysis of this project.
- *Cabin* will be removed, because it adds nothing to the analysis of this project.

- Convert all remaining character variables into factor.
- *Pclass* should be converted into factor.
- *Survived* should be converted into factor.

5.1 Variable removal

Following codes remove PassengerId, Name, Ticket, and Cabin.

Code

5.2 Factor conversion

Following codes convert all remaining character variables into factor as well as the numeric column Pclass and Survived.

Code

5.3 Renaming levels

This section renames the levels of many variables. It will not impact on the analysis, instead it will help readers to understand the levels better, especially when these levels are displayed in graphs.

Code

5.4 Renaming variables

Making all variables' name into lower-case format as there are more than 1 form of format. It will not affect the analysis, but helps to make the table looks more clean and tidy.

Code

5.5 Imputation

This section applies imputation model to fill up missing values in the dataset. There are many types of imputation methods including using mean, median, mode (most occurring values, generally applies to categorical data), and machine learning models that make use of the entire dataset to predict the missing values.

Missing values that need imputation are present in following columns:

- Fare (Will be imputed using median)
- Embarked (Will be imputed using mode)
- Age (Will be imputed using imputation model)

Code

```
##       pclass       sex          age            sibsp           parch
##  1st_class:323   Female:466   Min.   : 0.17   Min.   :0.0000   Min.   :0.000
```

```
##  2nd_class:277   Male  :843   1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000
##  3rd_class:709                 Median :28.00   Median :0.0000   Median :0.000
##                                Mean   :29.88   Mean   :0.4989   Mean   :0.385
##                                3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000
##                                Max.   :80.00   Max.   :8.0000   Max.   :9.000
##                                NA's   :263
##       fare              embarked   survived     source
##  Min.   :  0.000              :  2   No  :549   test :418
##  1st Qu.:  7.896   Cherbourg  :270   Yes :342   train:891
##  Median : 14.454   Queenstown :123   NA's:418
##  Mean   : 33.295   Southampton:914
##  3rd Qu.: 31.275
##  Max.   :512.329
##  NA's   :1
```

Following codes will (1) replace the NA's in *Fare* with the overall fare median, and (2) replace 2 of the NA's in *Embarked* with "S", which is the most frequently occurring level within the column.

Code

Following codes complete the imputation of missing values in "Age" using bagged tree algorithm.

Code

All missing values in the dataset have been filled up and left with only the column of "Survived" with 418 missing values. This is actually the responding variable of this analysis, and the survivorship of these missing values will be computed via machine learning algorithm in later section.

Code

```
##       pclass        sex          age            sibsp            parch
##  1st_class:323   Female:466   Min.   : 0.17   Min.   :0.0000   Min.   :0.000
##  2nd_class:277   Male  :843   1st Qu.:21.87   1st Qu.:0.0000   1st Qu.:0.000
##  3rd_class:709                Median :28.38   Median :0.0000   Median :0.000
```

```
##                    Mean   :29.72   Mean   :0.4989   Mean   :0.385
##                    3rd Qu.:36.00   3rd Qu.:1.0000   3rd Qu.:0.000
##                    Max.   :80.00   Max.   :8.0000   Max.   :9.000
##      fare             embarked   survived     source
## Min.   :  0.000   Cherbourg  :270   No  :549   test :418
## 1st Qu.:  7.896   Queenstown :123   Yes :342   train:891
## Median : 14.454   Southampton:916   NA's:418
## Mean   : 33.281
## 3rd Qu.: 31.275
## Max.   :512.329
```

Clean up the age column, the numeric in age column shouldn't has floating numbers and therefore I am rounding up those imputed values since they are just an estimate.

Code

5.6 Round the Fare

Since the unit of fare often comes with 2 floating numbers, I will transform decimal places of "fare" from 4 into 2.

Code

5.7 Feature Engineering

Since "SibSp" (number of siblings or spouses) and "Parch" (parents or children) are the total number of family a passenger was with, and a combination of them would create a new variable "familysize".

Code

Grouping different ranges of age into "age_group" of kid, teenage, adult, and elder.

Code

The dataset has now been cleaned.

6 MACHINE LEARNING

There are 418 passengers do not have their survivorship recorded in the dataset, I will predict their survivorship using relevant data in the dataset with the aid of machine learning algorithms.

Code

```
##   No  Yes NA's
```

```
##  549  342  418
```

There will be 3 different ways in splitting the dataset.

Split the dataset into two datasets, one with survivorship and one without survivorship.

Code

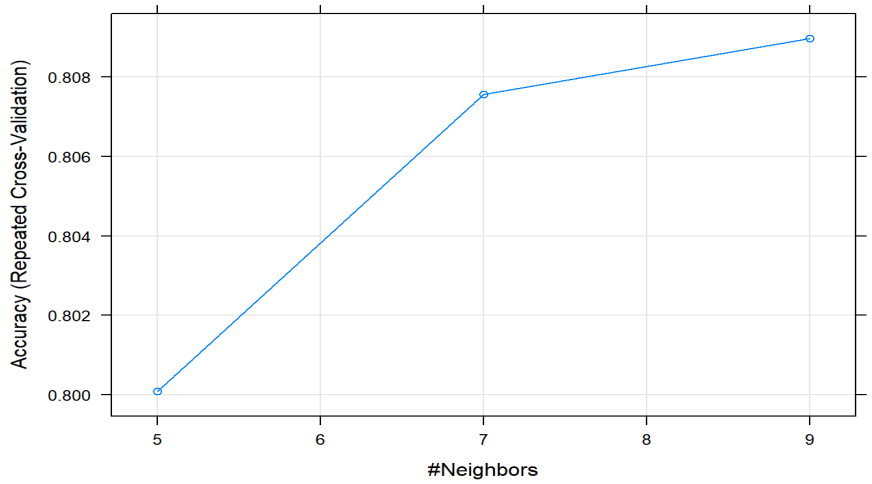
Split out the one with survival information into 80% train set and 20% test set.

Code

6.1 K-Nearest Neightbors (KNN)

This section trains a non-parametric algorithm, KNN, on the train set and make predictions on the test set.

Code



According to graph above and following function, the best K is 9. It will automatically selected as the default K value when this KNN model is used for predictions.

Code

```
##   k
## 3 9
```

Applying the KNN model to make predictions on the test set and evaluate its predictive performance (accuracy at %).

Code

```
## [1] 0.8248588
```

Confusion matrix to check on other performance metrics of this model.

Code

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction  No Yes
##        No  100  22
##        Yes   9  46
##
##               Accuracy : 0.8249
##                 95% CI : (0.7607, 0.8778)
##    No Information Rate : 0.6158
##    P-Value [Acc > NIR] : 1.304e-09
##
##                  Kappa : 0.6161
##
##  Mcnemar's Test P-Value : 0.03114
##
##             Sensitivity : 0.9174
##             Specificity : 0.6765
##          Pos Pred Value : 0.8197
##          Neg Pred Value : 0.8364
##              Prevalence : 0.6158
##          Detection Rate : 0.5650
##    Detection Prevalence : 0.6893
##       Balanced Accuracy : 0.7970
##
##        'Positive' Class : No
##
```

6.2 Random Forest

This section applies random forest algorithm on the train set and make predictions on the test test.

Code

Making the predictions based on random forest model and evaluate its predictive performance (%).

Code

```
## [1] 0.8248588
```

Confusion matrix to check on other performance metrics of this model.

Code

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction No Yes
##        No  98  20
##        Yes 11  48
##
##               Accuracy : 0.8249
##                 95% CI : (0.7607, 0.8778)
##    No Information Rate : 0.6158
##    P-Value [Acc > NIR] : 1.304e-09
##
##                  Kappa : 0.6204
##
##  Mcnemar's Test P-Value : 0.1508
##
##             Sensitivity : 0.8991
##             Specificity : 0.7059
##          Pos Pred Value : 0.8305
##          Neg Pred Value : 0.8136
##              Prevalence : 0.6158
```

```
##          Detection Rate : 0.5537
##    Detection Prevalence : 0.6667
##       Balanced Accuracy : 0.8025
##
##       'Positive' Class : No
##
```

6.3 Xgboosts

This section applies extreme-gradient boosting, which is an alternative to random forest algorithm. Building the model with following codes.

Code

Making the predictions based on random forest model and evaluate its performance.

Code

```
## [1] 0.8022599
```

Confusion matrix to check on other performance metrics of this model.

Code

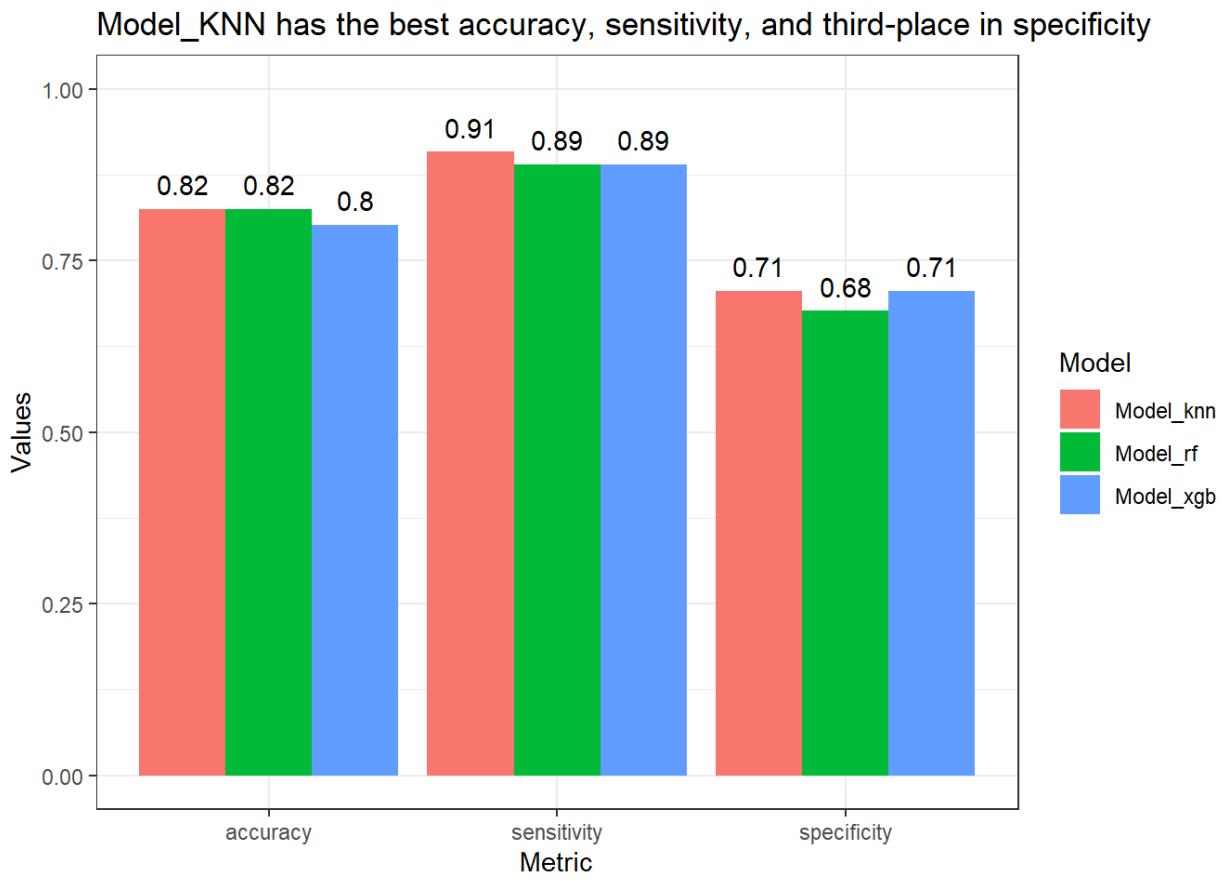
```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction No Yes
##        No  94  20
##        Yes 15  48
##
##               Accuracy : 0.8023
##                 95% CI : (0.7359, 0.8582)
##    No Information Rate : 0.6158
##    P-Value [Acc > NIR] : 7.432e-08
##
##                  Kappa : 0.5762
##
##  Mcnemar's Test P-Value : 0.499
```

```
##
##            Sensitivity : 0.8624
##            Specificity : 0.7059
##         Pos Pred Value : 0.8246
##         Neg Pred Value : 0.7619
##             Prevalence : 0.6158
##         Detection Rate : 0.5311
##   Detection Prevalence : 0.6441
##      Balanced Accuracy : 0.7841
##
##       'Positive' Class : No
##
```

6.4 Model comparison

I will use the KNN model to make prediction on the new dataset that do not have survivorship recorded because KNN model has the highest accuracy, sensitivity, and specificity.

Code



7 PREDICTIONS

A quick recap, my dataset has 1309 rows, and there are 418 passengers do not have their survivorship recorded. Therefore, I trained 3 machine learning algorithms and found that the KNN model had the best predictive power.

Now, I will use the KNN model to predict the survivorship of these 418 passengers to obtain a final cleaned dataset for visualisation.

Making the prediction with following codes.

Code

```
##  [1] No  No  No  No  No  No  Yes No  Yes No  No  No  Yes No  Yes Yes No  No
## [19] No  No  No  Yes Yes No  Yes No  Yes No  Yes No  No  No  No  No  Yes No
## [37] No  No  No  No  No  No  No  Yes Yes No  No  No  Yes No  No  No  Yes Yes
## [55] No  No  No  No  No  Yes No  No  No  Yes Yes Yes Yes No  No  Yes Yes No
## [73] No  No  Yes No  No  Yes No  Yes Yes No  No  No  No  No  Yes Yes Yes Yes
## [91] No  No  Yes No  No  No  No  No  No  No  Yes No  No  No  Yes No  No  No
```

```
## [109] No  No  No  Yes Yes Yes Yes No  No  Yes No  Yes Yes No  Yes No  No  Yes
## [127] No  Yes No  No  No  No  No  No  No  No  No  No  Yes No  No  Yes Yes No
## [145] Yes No  No  No  No  No  Yes No  No  No  No  No  Yes Yes Yes No  Yes Yes
## [163] Yes No  No  Yes No  No  Yes No  No  No  No  No  No  Yes Yes No  Yes Yes
## [181] No  Yes Yes No  Yes No  Yes No  No  No  No  No  Yes No  Yes No  Yes Yes
## [199] No  No  Yes Yes No  Yes No  No  Yes No  Yes No  No  No  No  Yes No  No
## [217] Yes Yes Yes No  Yes No  Yes No  Yes Yes No  Yes No  No  No  Yes No  No
## [235] Yes No  No  No  Yes Yes Yes Yes Yes No  No  No  Yes No  Yes No  Yes No
## [253] Yes No  No  No  No  No  No  No  No  No  Yes Yes No  No  No  No  No  No
## [271] No  No  Yes Yes No  Yes No  No  No  No  No  Yes Yes Yes Yes No  No  No
## [289] No  No  No  Yes No  No  No  No  Yes No  Yes No  No  No  No  No  Yes Yes
## [307] Yes Yes Yes No  No  No  No  Yes Yes Yes Yes No  No  No  No  No  No  Yes
## [325] Yes No  Yes No  No  No  Yes No  No  Yes No  Yes No  No  No  No  No  No
## [343] No  Yes No  Yes No  No  No  Yes Yes No  No  Yes Yes No  Yes No  No  No
## [361] No  Yes Yes No  Yes No  No  No  Yes No  No  Yes No  No  Yes Yes No  No
## [379] No  No  No  No  No  Yes No  Yes No  No  No  No  Yes Yes No  No  No  Yes
## [397] No  Yes No  No  Yes No  Yes No  No  No  No  No  Yes Yes Yes Yes No  No
## [415] Yes No  No  No
## Levels: No Yes
```

Data insert.

Code

Combine both titanic tables.

Code

Final check of the dataset:

Code

```
##       pclass       sex          age          age_group       sibsp
##  1st_class:323   Female:466   Min.   : 0.00   Kid    : 108   Min.   :0.0000
##  2nd_class:277   Male  :843   1st Qu.:22.00   Teenage: 132   1st Qu.:0.0000
##  3rd_class:709                Median :28.00   Adult  :1059   Median :0.0000
##                               Mean   :29.69   Elder  :  10   Mean   :0.4989
```

```
##                      3rd Qu.:36.00              3rd Qu.:1.0000
##                      Max.   :80.00              Max.   :8.0000
##     parch        familysize          fare             embarked    survived
## Min.   :0.000   Min.   : 0.0000   Min.   :  0.00   Cherbourg  :270   No :819
## 1st Qu.:0.000   1st Qu.: 0.0000   1st Qu.:  7.90   Queenstown :123   Yes:490
## Median :0.000   Median : 0.0000   Median : 14.45   Southampton:916
## Mean   :0.385   Mean   : 0.8839   Mean   : 33.28
## 3rd Qu.:0.000   3rd Qu.: 1.0000   3rd Qu.: 31.27
## Max.   :9.000   Max.   :10.0000   Max.   :512.33
```

There are no more missing values from the dataset and is now ready for visualisation.

Saving the file.

Code

8 VISUALISATION

Code

8.1 Passengers across Classes

There are 1309 rows of passengers information, among them, there are 323 passengers bought the first class ticket, 277 for second class ticket, and 709 for third class ticket.

Code

3rd_class54.2%1st_class24.7%2nd_class21.2%

Passenger Counts by Ticket Classes