Titanic Data Analysis

Ahmed Ashraf Mohamed Omar Gamal Abdelaziz Ahmed Yousri Ahmed Dawood
Alia Medhat

Contents

1 Preliminary Look at the data **1**
1.1 Loading the packages and the Data . . . 1

2 Exploration Of The Data **2**
2.1 Description of the Data . . . 2
2.2 Summary of Data . . . 2
2.3 Plotting The Data . . . 3

3 Description of the data **6**
3.1 Categorical Features . . . 6
3.2 Numerical Features . . . 6

4 kernel distribution **6**
4.1 Rules . . . 7
4.2 Kernel Function . . . 8

1 Preliminary Look at the data

We need first to define the data we have.

Variable	Definition	Key
survival	Survival	0 = No, 1 = yes
pclass	ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	sex	
age	Age in year	
sibsp	Number of siblings/spouses aboard the titanic	
parch	Number of parents/children aboard the Titanic	
ticket	ticket number(unique)	
fare	Passenger fare	
cabin	Cabin number	
embarked	port of embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

1.1 Loading the packages and the Data

```
# Loading Packages
library(tidyverse)
```

```r
library(viridis)
library(ggplot2)
library(ggcorrplot)
library(ggthemes)
library(hrbrthemes)
library(e1071)
library(mice)
library(statsr)


# Loading Data
train <- read_csv("data/train.csv")
test <- read_csv("data/test.csv")
```

2 Exploration Of The Data

2.1 Description of the Data

2.2 Summary of Data

```r
summary(train)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##      Sex                 Age            SibSp           Parch
##  Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##  Mode  :character   Median :28.00   Median :0.000   Median :0.0000
##                     Mean   :29.70   Mean   :0.523   Mean   :0.3816
##                     3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                     Max.   :80.00   Max.   :8.000   Max.   :6.0000
##                     NA's   :177
##     Ticket               Fare           Cabin             Embarked
##  Length:891         Min.   :  0.00   Length:891         Length:891
##  Class :character   1st Qu.:  7.91   Class :character   Class :character
##  Mode  :character   Median : 14.45   Mode  :character   Mode  :character
##                     Mean   : 32.20
##                     3rd Qu.: 31.00
##                     Max.   :512.33
##
```

Checking for Missing values in each feature

```r
colSums(is.na(train))
```

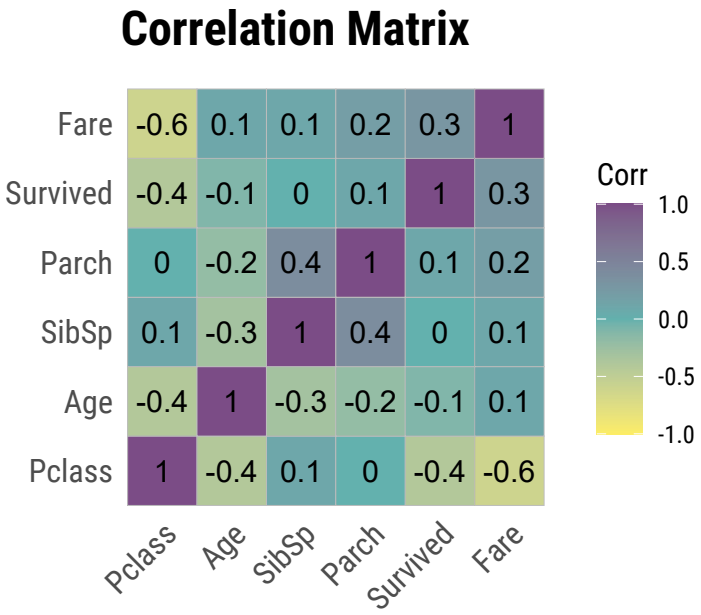
```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           2
```

2.3 Plotting The Data

2.3.1 Correlation Matrix

We are going to use correlation matrix of the numerical data to assess the correlation, which might gives a better idea of which feature might be important

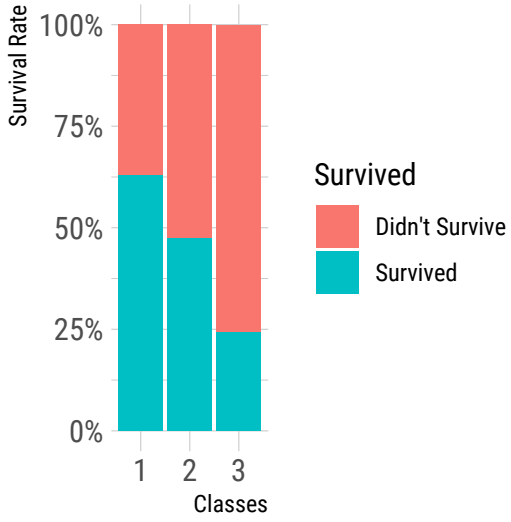
```
correlationMatrix <- train %>%
  filter(!is.na(Age)) %>%
  select(Survived, Pclass,Age,SibSp,Parch,Fare) %>%
  cor() %>%
  ggcorrplot(lab = T,
             ggtheme =theme_ipsum_rc(grid = F),
             title="Correlation Matrix",hc.order=T,
             colors =rev(viridis(3,alpha=0.7)),
             digits = 1)

correlationMatrix
```



The fare features seems to be the most correlated feature to survival of the passengers, but it doesn't negate the importance of the other features in the data. Which means that we will start by comparing the each that we consider to be important against survival feature

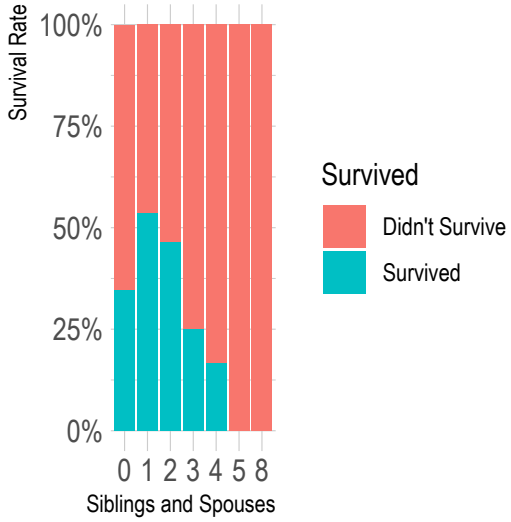
2.3.2 Class of Passenger

```
gPclassSurvived <- train %>%
  select(Pclass,Survived) %>%
  ggplot(aes(as_factor(Pclass),fill=as_factor(Survived))) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels=scales::percent) +
  theme_ipsum_rc() +
  labs(x = "Classes",y = "Survival Rate")+
  scale_fill_discrete(name = "Survived", labels = c("Didn't Survive","Survived"))

gPclassSurvived
```



2.3.3 Siblings and Spouses

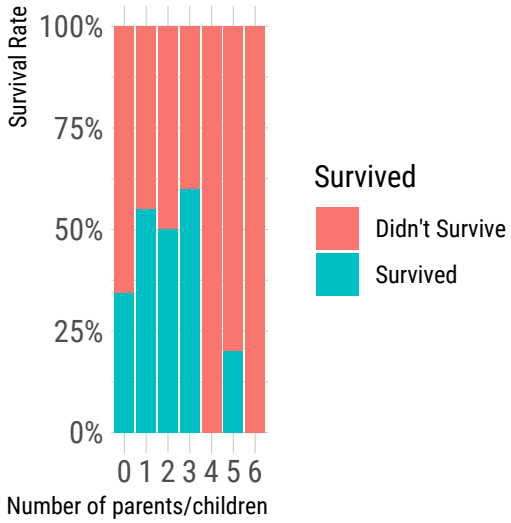
```
gSibSpSurvived <- train %>%
  select(SibSp,Survived) %>%
  ggplot(aes(as_factor(SibSp),fill=as_factor(Survived))) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  labs(x = "Siblings and Spouses",y = "Survival Rate")+
  scale_fill_discrete(name = "Survived", labels = c("Didn't Survive","Survived")) +
  theme_ipsum()

gSibSpSurvived
```



2.3.4 Number of children/parents

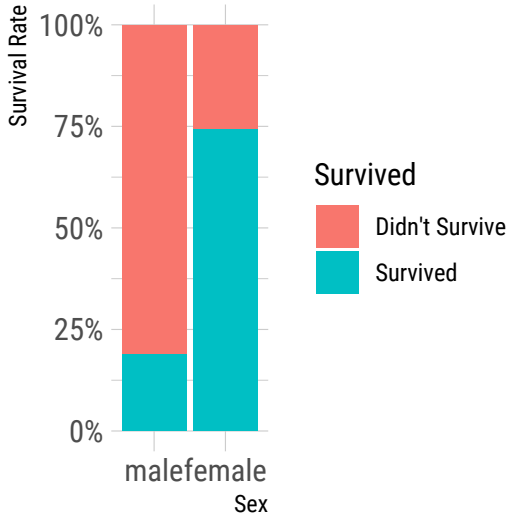
```
gParchSurvived <- train %>%
  select(Parch,Survived) %>%
  ggplot(aes(as_factor(Parch),fill=as_factor(Survived))) +
  geom_bar(position = "fill") +
  scale_y_continuous(label = scales::percent)+
```

```
  labs(x = "Number of parents/children",y = "Survival Rate")+
  scale_fill_discrete(name = "Survived", labels = c("Didn't Survive","Survived")) +
  theme_ipsum_rc()
gParchSurvived
```



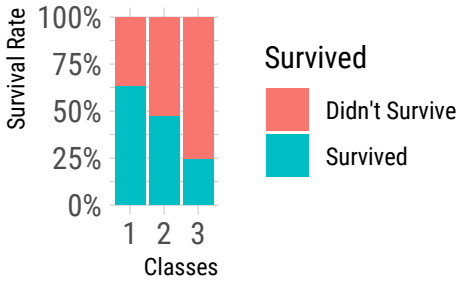
2.3.5 Gender VS Survived

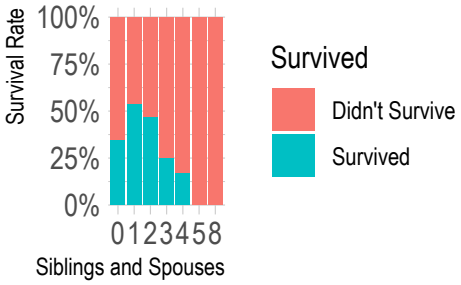
```
gSexSurvived <- train %>%
  select(Sex,Survived) %>%
  ggplot(aes(as_factor(Sex),fill = as_factor(Survived))) +
  geom_bar(position = "fill") +
  scale_y_continuous(label = scales::percent) +
  labs(x = "Sex",y = "Survival Rate")+
  scale_fill_discrete(name = "Survived", labels = c("Didn't Survive","Survived")) +
  theme_ipsum_rc()
gSexSurvived
```

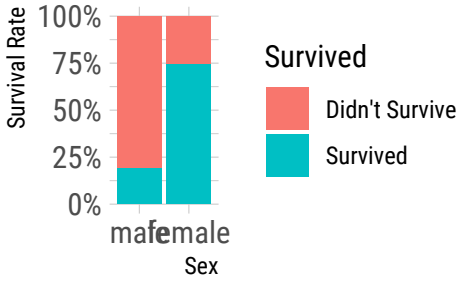


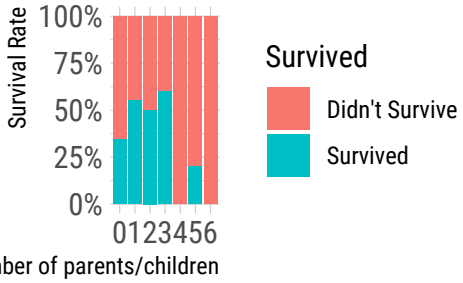

```
gridExtra::grid.arrange(gPclassSurvived,
                        gSibSpSurvived,
                        gSexSurvived,
                        gParchSurvived,
```

```
nrow=2)
```










```
train %>%
  group_by(Sex) %>%
  summarise(Age_mean = mean(Age,na.rm=TRUE),
            age_sd = sd(Age,na.rm=T),
            surival_mean = mean(Survived,na.rm =T),
            surival_sd = sd(Survived,na.rm = T))
```

```
## # A tibble: 2 x 5
##   Sex    Age_mean age_sd surival_mean surival_sd
##   <chr>     <dbl>  <dbl>        <dbl>      <dbl>
## 1 female     27.9   14.1        0.742      0.438
## 2 male       30.7   14.7        0.189      0.392
```

3 Description of the data

3.1 Categorical Features

3.2 Numerical Features

4 kernel distribution

A kernel distribution is a nonparametric representation of the probability density function (pdf) of a random variable in any population

The kernel smoothing function defines the shape of the curve used to generate the pdf Kernel distribution is Quote from histogram in other word (smooth representation of a histogram) That the integral =1 There is a benefit of smooth representation of a histogram like Ignores irregularities and outliers , more efficient in approximation so it deals better with large data than small data

$$\hat{f}_h = \frac{1}{n} = \sum_{i=1}^{n} K(x - x_i) = \frac{1}{nh} K\left(\frac{x - x_i}{h}\right)$$

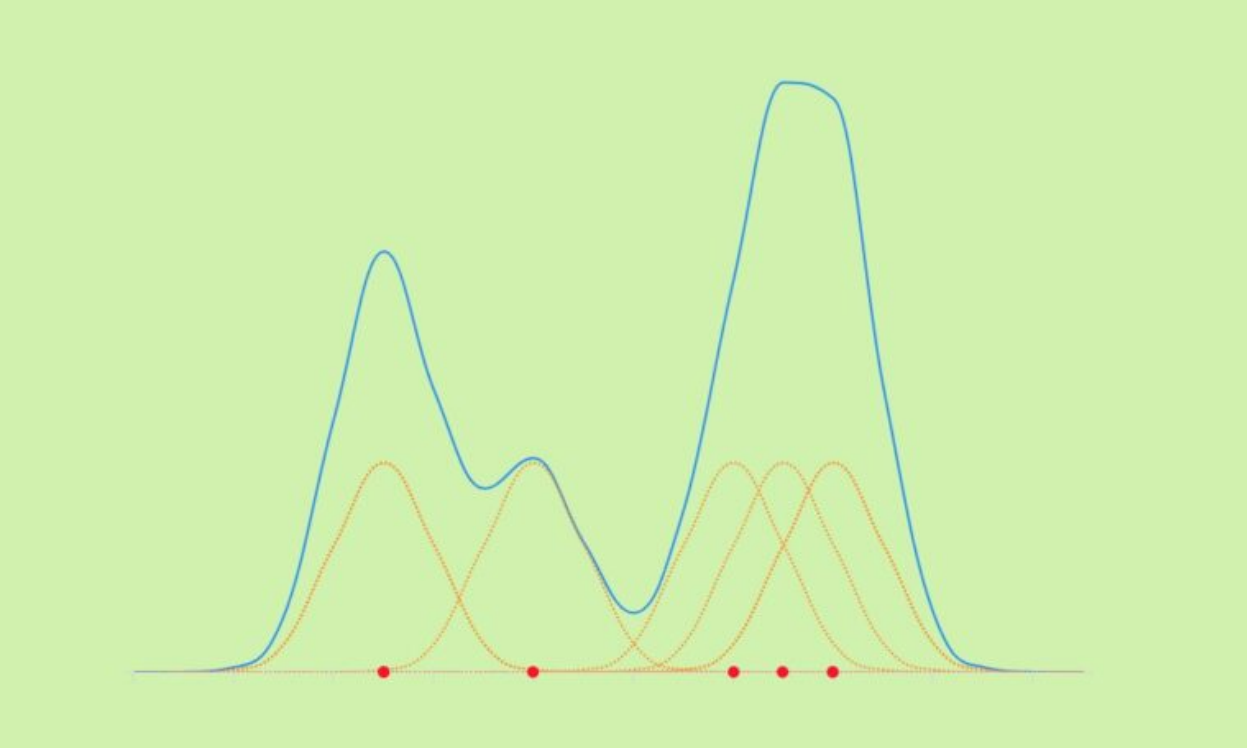
4.1 Rules

4.1.1 Non-weighted Data

$$\hat{f}_h = \frac{1}{n} \sum_{i=1}^{n} K(x - x_i) = \frac{1}{nh} K\left(\frac{x - x_i}{h}\right)$$

4.1.2 Weighted Data

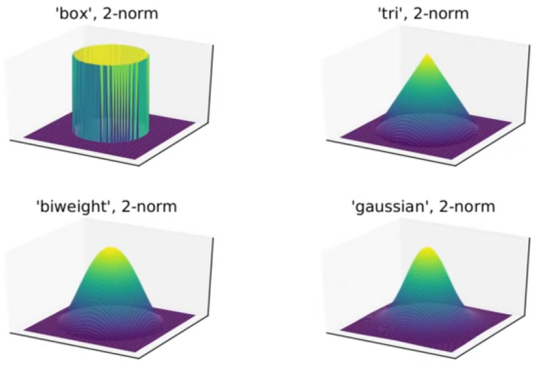
$$\hat{f}_h = \frac{1}{h} \sum_{i=1}^{N} w_i K\left(\frac{x - x_i}{h}\right), \qquad \text{where} \sum_{i=1}^{N} w_i = 1$$



Kernels in 2D

An approach to d-dimensional estimates is to write

$$\hat{f}(x) = \frac{1}{h^d} \sum_{i=1}^{N} w_i K\left(\frac{\|x - x_i\|_p}{h}\right), \text{ where } \sum_{i=1}^{N} w_i = 1.$$

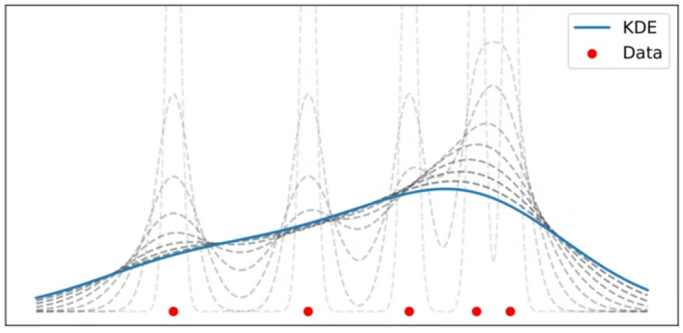


61 / 105

Choice of bandwidth

We use h to control for the *bandwidth* of $\hat{f}(x)$ by writing

$$\hat{f}(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right).$$



34 / 105

4.2 Kernel Function

1. Box
2. Triangle
3. Normal
4. Epanechnikov

Each density curve uses the same input data, but applies a different kernel smoothing function to generate the pdf. The density estimates are roughly comparable, but the shape of each curve varies slightly. For example, the box kernel

produces a density curve that is less smooth than the others.

The choice of bandwidth value controls the smoothness of the resulting probability density curve (higher value of h more smoothing)

Specifying a smaller bandwidth produces a very rough curve, but reveals that there might be two major peaks in the data. Specifying a larger bandwidth produces a curve nearly identical to the kernel function Choosing the optimal (h) bandwidth methods : 1- Silverman's rule of thump that computes an optimal h by assuming that data is normally distributed 2- Improved Sheather jones (ISJ) an algorithm is more robust with multimodality data or a lot of data (one disadvantage is it needs to large data)

Bounded domains data : have a constrains like data couldn't be negative (-ve lead to probability = 0) Mirror method 1- Mirror the data 2- Sum the original and mirrored kernel density estimate 3- Chop it so that zero at the boundary side

2D :

h : could be matrix (different h in different directions) the choice of norm comes into d>=2 the p-norm is $||x||_p := (\sum_{i=1} |x|^p)^{\frac{1}{p}}$ norm-p =1 manhattan distance norm-p =2 ecludian norm

norm-p =inf maximum norm (it's not abvious in every case which norm is the correct one)

standard euclidean distance is good choice because it invariant under rotation as large data choice of k and p isn't important so