Titanic Data Analysis

Ahmed Ashraf Omar Gamal Abdelaziz Ahmed Yousri Ahmed Dawood
Alia Medhat

Contents

1 Introduction

In this project we are trying first to explore our data to get a better understanding To answer our questions we first need to have a preliminary look at our data, so that we can get a better a idea what we are dealing with, as well as the possible missing data and relationships that exist

1.1 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 = Queens-town, S = Southampton

1.2 Loading the packages and the Data

```
# Loading Packages
## tidyverse loads dplyr and readr
library(tidyverse)

## To have different color maps
library(viridis)

## ggplot2 to produce different plots
library(ggplot2)

## uses ggplot2 to produce a correlation matrix -- the data must be in the correct form
library(ggcorrplot)

## Gives us better themes
library(hrbrthemes)

## to use skewness fun. to calculate skewness of the distribution
library(e1071)

## Multivariate imputation using chained equations -- to impute the missing values in our data
library(mice)

## Loads different statistical functions
library(statsr)

## To produce interactive plot
library(plotly)

## To use bias function
library(Metrics)


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

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

# Binding them into a full data frame
```

```r
df <- bind_rows(train,test)
```

2 Exploration Of The Data

2.1 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                                        
## 
```

2.2 Categories of Features

Quantitative data are measures of values or counts and are expressed as numbers.

Qualitative data are measures of 'types' and may be represented by a name, symbol, or a number code.

2.2.1 Qualitive

Categorical: Survived, Sex, and Embarked. Ordinal: Pclass. Nominal: Name.

2.2.2 Quantitive

Continuous: Age, Fare. Discrete: SibSp, Parch.

2.2.3 Mix types

Ticket is a mix of numeric and alphanumeric data types Cabin is mix between alpha and numeric

2.3 Exploring Missing Data.

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
```

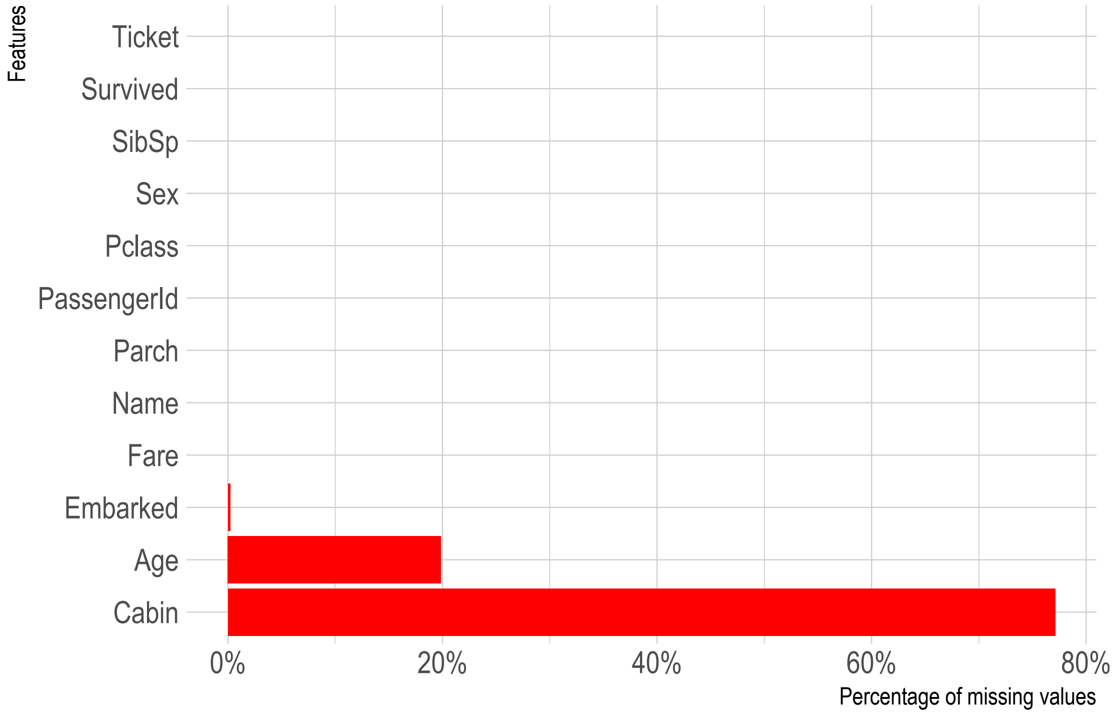
```r
missing_values <- train %>% summarize_all(funs(sum(is.na(.))/n()))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## i Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

```r
missing_values <- gather(missing_values, key="feature", value="missing_pct")

missing_values
```

```
## # A tibble: 12 x 2
##    feature     missing_pct
##    <chr>             <dbl>
##  1 PassengerId     0
##  2 Survived        0
##  3 Pclass          0
##  4 Name            0
##  5 Sex             0
##  6 Age             0.199
##  7 SibSp           0
##  8 Parch           0
##  9 Ticket          0
## 10 Fare            0
## 11 Cabin           0.771
## 12 Embarked        0.00224
```

```r
missing_values %>%
  ggplot(aes(x=reorder(feature,-missing_pct),y=missing_pct)) +
  geom_bar(stat="identity",fill="red") +
  coord_flip() + # to flip the graph
  xlab("Features") +
  ylab("Percentage of missing values")+
  scale_y_continuous(labels=scales::percent) +
  theme_ipsum()
```



2.3.1 Missing data, it is normal?

It is quite normal to see missing data in any data-set as such data is collected by manually which means that there might be some error. Missing data present various problems. First, the absence of data reduces statistical power, which refers to the probability that the test will reject the null hypothesis when it is false. Second, the lost data can cause bias in the estimation of parameters. Third, it can reduce the representativeness of the samples.

2.3.2 The solution to the missing data

2.3.2.1 Removal The removal of the observations that contain missing might cause a bigger issue where it might an even bigger loss of information which cause bias in our estimation. But on close observation of our data we can identify that there are some features that are not extremely useful and contain missing data, so we can drop such features.

```
train <- train %>%
  select(!Cabin)
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  
```

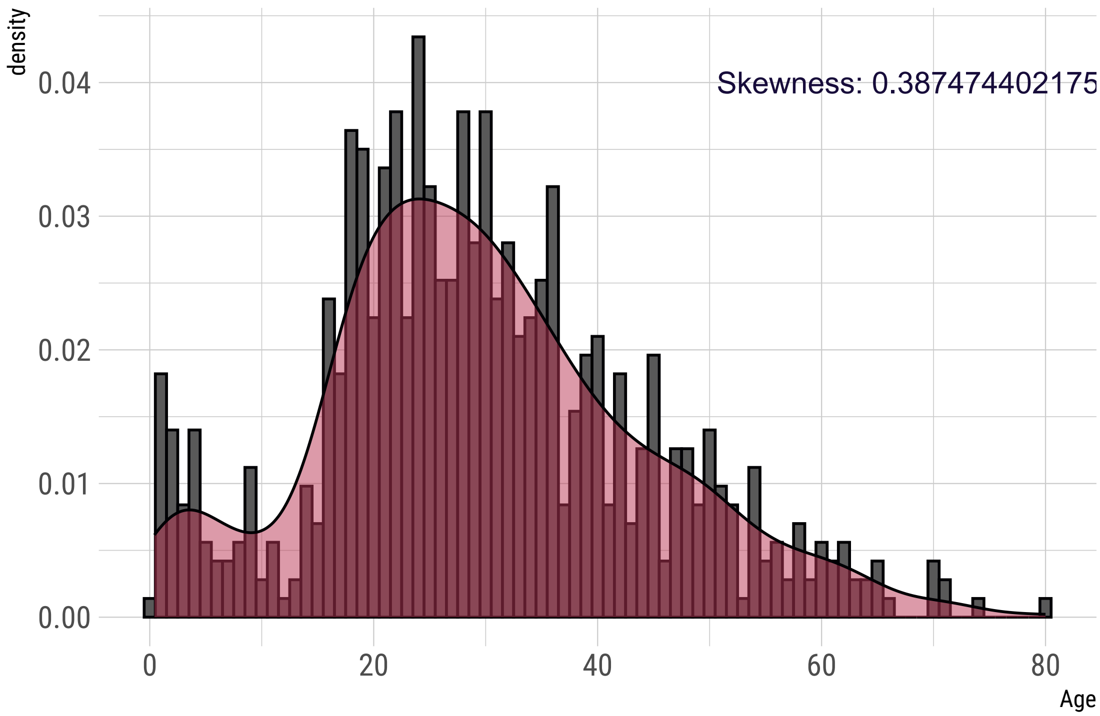
```
##                      NA's   :177
##     Ticket               Fare          Embarked
##  Length:891         Min.   :  0.00   Length:891
##  Class :character   1st Qu.:  7.91   Class :character
##  Mode  :character   Median : 14.45   Mode  :character
##                     Mean   : 32.20
##                     3rd Qu.: 31.00
##                     Max.   :512.33
##
```

2.3.2.2 Imputation Imputing the missing data gives the advantage that we use a learning model to predict such missing data and maintain the distribution of our data.

2.4 Histogram of Age feature.

```
gAgeDensity <- train %>%
  select(Age) %>%
  ggplot(aes(Age, y = ..density..)) +
  geom_histogram(bins = 20,binwidth = 1,color=inferno(1,alpha=1)) +
  geom_density(fill=inferno(1,begin = 0.5,alpha = 0.5),color = inferno(1,begin=0)) +
  annotate(
    "text",
    x = 70,
    y = 0.04,
    label = paste("Skewness:",skewness(train$Age,na.rm = T)),
    colour = inferno(1,begin = 0.1),
    size = 4
  ) +
  theme_ipsum_rc()

gAgeDensity
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.

## Warning: Removed 177 rows containing non-finite values (`stat_bin()`).

## Warning: Removed 177 rows containing non-finite values (`stat_density()`).
```



2.4.1 Population mean and standard deviation of Age feature Before imputation

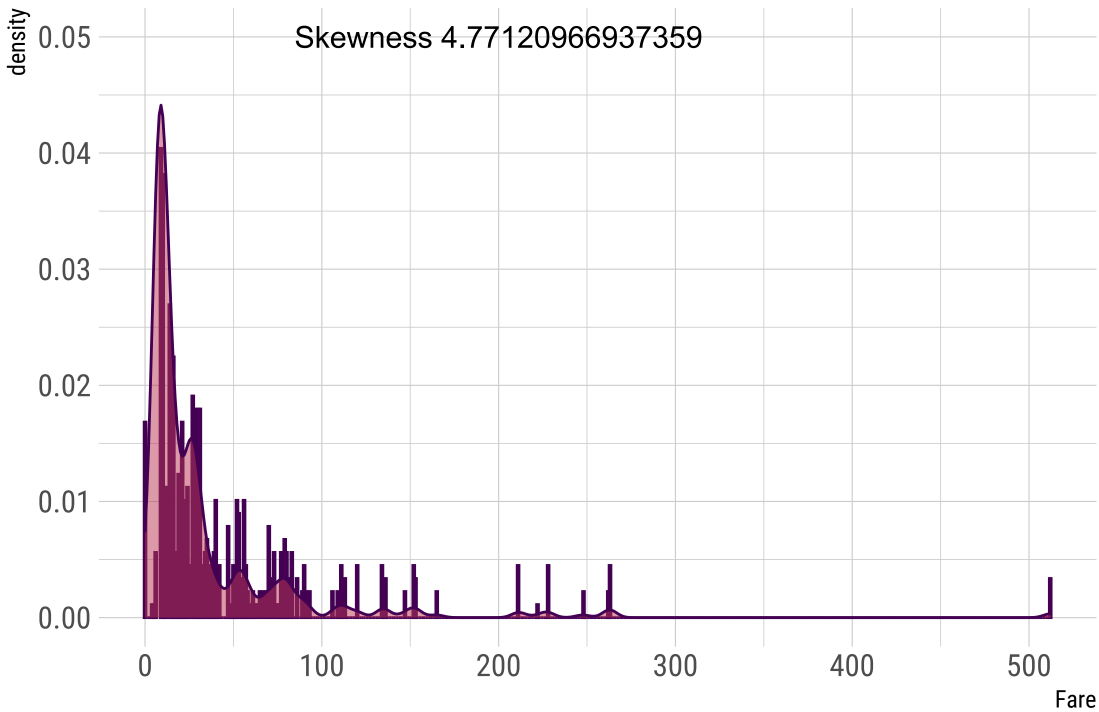
```
train %>%
  summarise(Age_mean = mean(Age,na.rm = T), Age_sd = sd(Age,na.rm = T))
```

```
## # A tibble: 1 x 2
##   Age_mean Age_sd
##      <dbl>  <dbl>
## 1     29.7   14.5
```

2.5 Histogram of Fare feature

```
gFareDensity <- train %>%
  select(Fare) %>%
  ggplot(aes(Fare, y = ..density..)) +
  geom_histogram(bins = 20,binwidth = 1,color=viridis(1,alpha=1)) +
  geom_density(fill=inferno(1,begin = 0.5,alpha = 0.5),color = viridis(1,begin=0)) +
  scale_y_continuous(limits = c(0,0.05))+
  theme_ipsum_rc() +
  annotate(
    "text",
    x = 200,
    y = 0.05,
    label = paste("Skewness",skewness(train$Fare)),
    colour = "black",
    size = 4
  )

gFareDensity
```

```
## Warning: Removed 4 rows containing missing values (`geom_bar()`).
```



2.5.1 Population mean and standard deviaiton of Fare feature.

```
train %>%
  summarise(Fare_mean = mean(Fare,na.rm = T), Fare_sd = sd(Fare,na.rm = T))
```

```
## # A tibble: 1 x 2
##   Fare_mean Fare_sd
##       <dbl>   <dbl>
## 1      32.2    49.7
```

2.6 Imputing the missing age feature

2.6.1 Correlation matrix before imputation

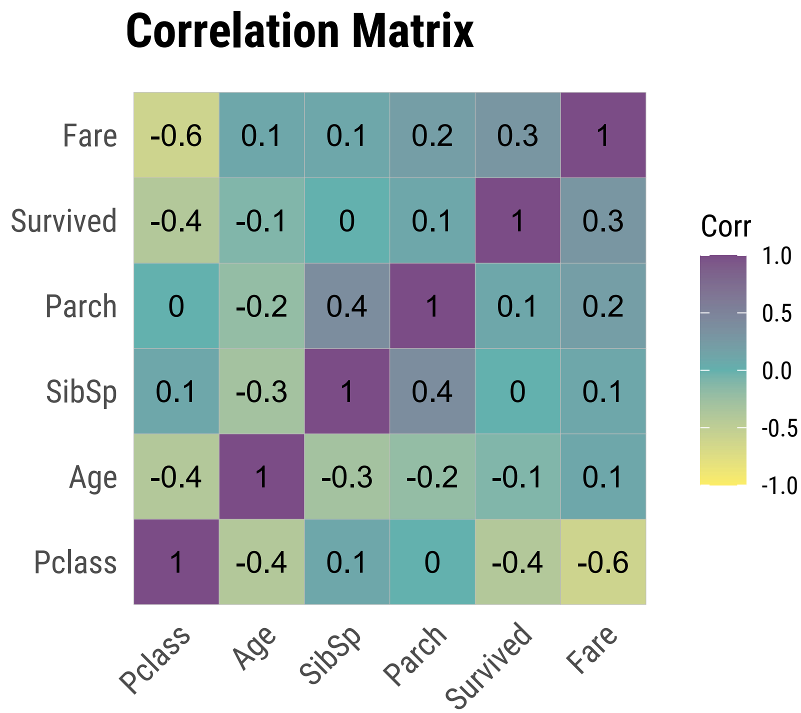
We are going to use correlation matrix of the numerical data to assess the correlation, which might gives a better idea of which feature might be important

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

We will print it once before imputation and the second time after imputation of our data to see an aspect of the data imputation

```
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
```



2.6.2 Imputation

```
#---------------MICE--------------
set.seed(129)
mice_mod <- mice(train[,!names(train) %in% c('PassengerId','Name','Ticket','Cabin','Survived')],method
```

```
##
##  iter imp variable
##   1   1  Age
##   1   2  Age
##   1   3  Age
##   1   4  Age
##   1   5  Age
##   2   1  Age
##   2   2  Age
##   2   3  Age
##   2   4  Age
##   2   5  Age
##   3   1  Age
##   3   2  Age
##   3   3  Age
##   3   4  Age
##   3   5  Age
##   4   1  Age
##   4   2  Age
##   4   3  Age
##   4   4  Age
##   4   5  Age
##   5   1  Age
```

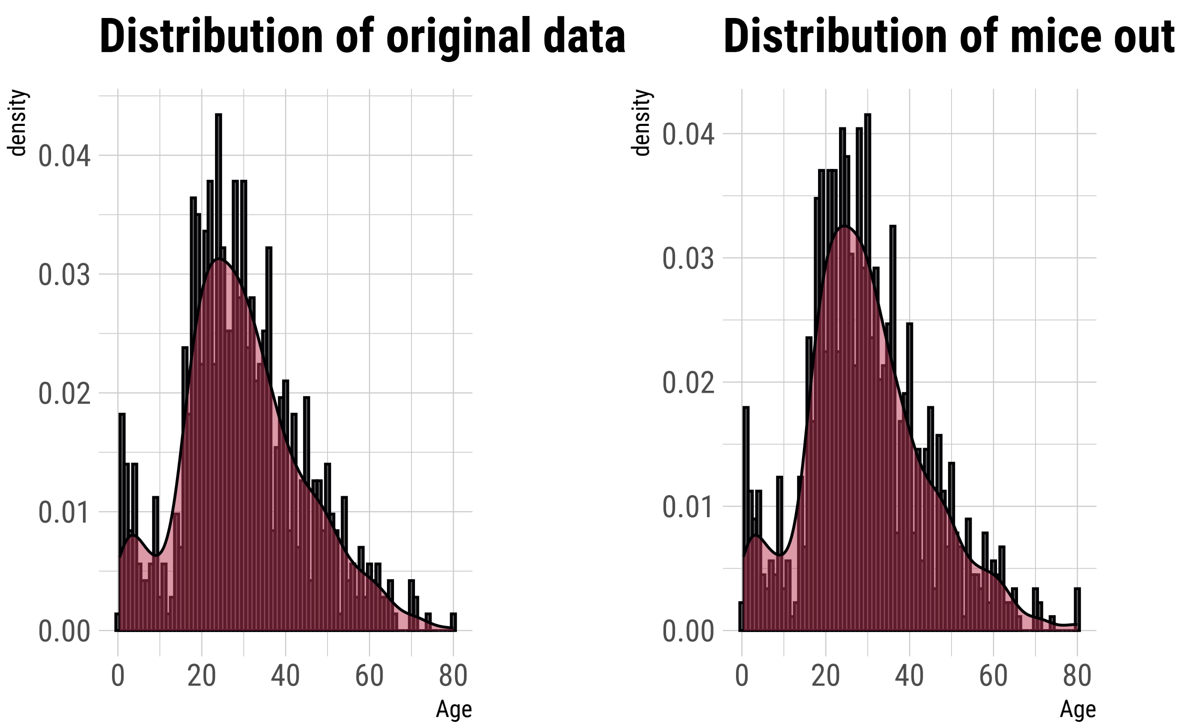
```
##   5   2  Age
##   5   3  Age
##   5   4  Age
##   5   5  Age
```

```
## Warning: Number of logged events: 2
```

```r
mice_output <- complete(mice_mod)

gdistrOriginalData <- train %>%
  select(Age) %>%
  ggplot(aes(Age, y = ..density..)) +
  geom_histogram(bins = 25,binwidth = 1,color=inferno(1,alpha=1)) +
  geom_density(fill=inferno(1,begin = 0.5,alpha = 0.5),color = inferno(1,begin=0)) +
  ggtitle("Distribution of original data") +
  theme_ipsum_rc()
gdistrMICEData <- mice_output %>%
  select(Age) %>%
  ggplot(aes(Age, y = ..density..)) +
  geom_histogram(bins = 25,binwidth = 1,color=inferno(1,alpha=1)) +
  geom_density(fill=inferno(1,begin = 0.5,alpha = 0.5),color = inferno(1,begin=0)) +
  ggtitle("Distribution of mice output") +
  theme_ipsum_rc()

gridExtra::grid.arrange(gdistrOriginalData,gdistrMICEData,nrow = 1)
```

```
## Warning: Removed 177 rows containing non-finite values (`stat_bin()`).
```

```
## Warning: Removed 177 rows containing non-finite values (`stat_density()`).
```

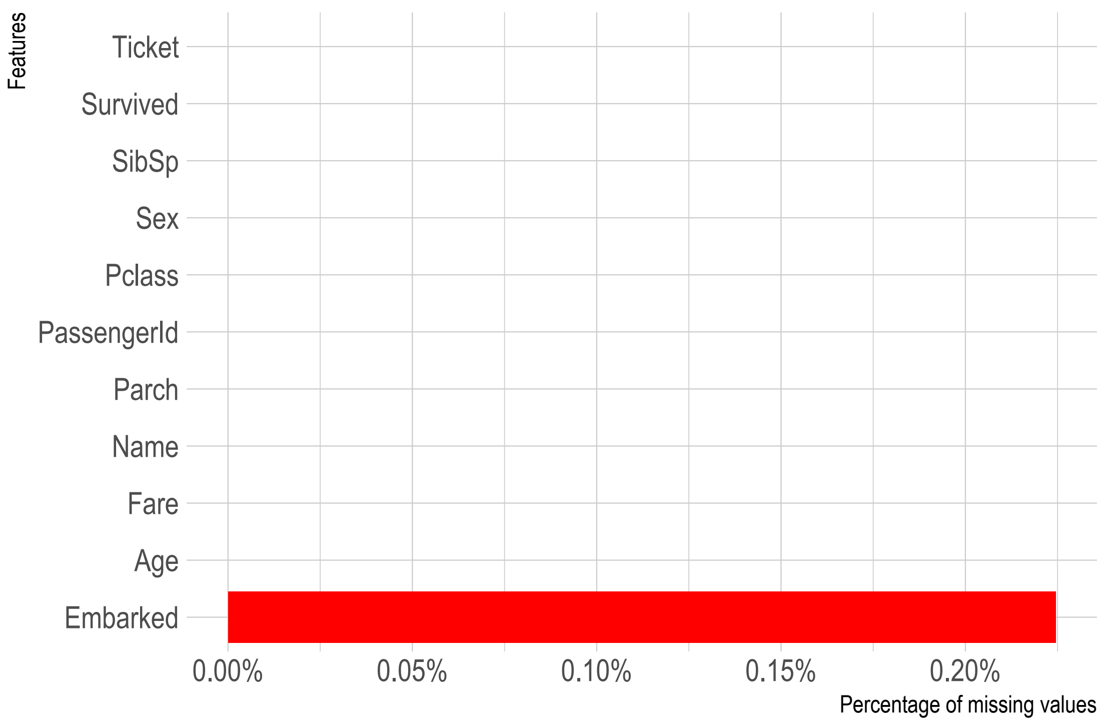



```r
train$Age <- mice_output$Age

missing_values <- train %>% summarize_all(funs(sum(is.na(.))/n()))
```

```r
missing_values <- gather(missing_values, key="feature", value="missing_pct")

missing_values
```

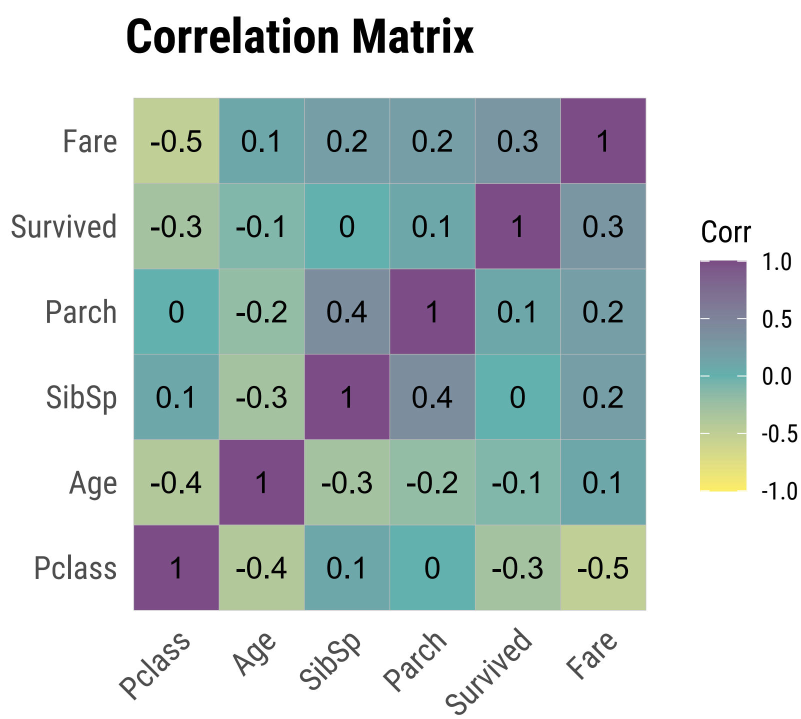
```
## # A tibble: 11 x 2
##    feature     missing_pct
##    <chr>             <dbl>
##  1 PassengerId     0
##  2 Survived        0
##  3 Pclass          0
##  4 Name            0
##  5 Sex             0
##  6 Age             0
##  7 SibSp           0
##  8 Parch           0
##  9 Ticket          0
## 10 Fare            0
## 11 Embarked        0.00224
```

```r
missing_values %>%
  ggplot(aes(x=reorder(feature,-missing_pct),y=missing_pct)) +
  geom_bar(stat="identity",fill="red") +
  coord_flip() + # to flip the graph
  xlab("Features") +
  ylab("Percentage of missing values")+
  scale_y_continuous(labels=scales::percent) +
  theme_ipsum()
```



2.6.3 Correlation matrix after imputation

```
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
```



2.6.4 Population mean and standard deviation

```
train %>%
  summarise(Age_mean = mean(Age,na.rm = T), Age_sd = sd(Age,na.rm = T))

## # A tibble: 1 x 2
##   Age_mean Age_sd
##      <dbl>  <dbl>
## 1     29.6   14.3
```

2.7 Determining the distrubutuion of Age and Fare By inspection

Though determination of the distribution using inspection is likely not going to be effective we are going to the KS-test in a later section

2.7.1 Age

The histogram of the Age feature look very much like a normal distribution, yet it's not a normal distribution itself.

2.7.2 Fare

The histogram of the Fare feature fits the χ^2 distribution.

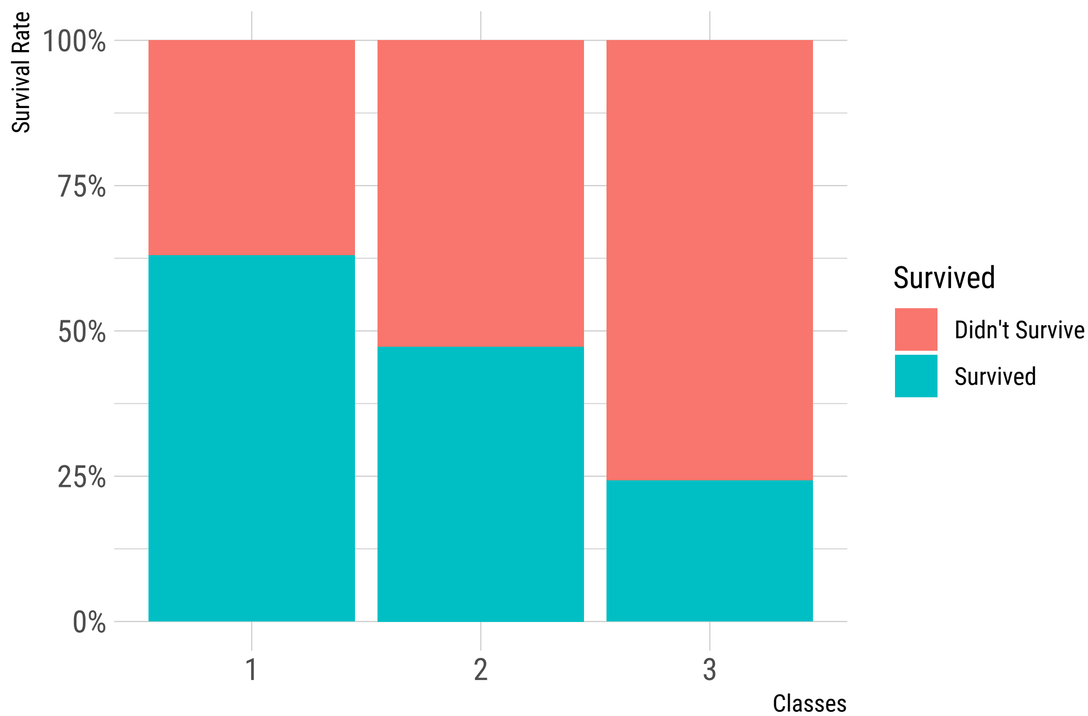
2.8 Plotting The Data

We include even more plots of our data to get a better understanding of it, help us see hidden correlations and ways to facilitate our analysis

2.8.1 Class of Passenger Vs Survived

```
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
```



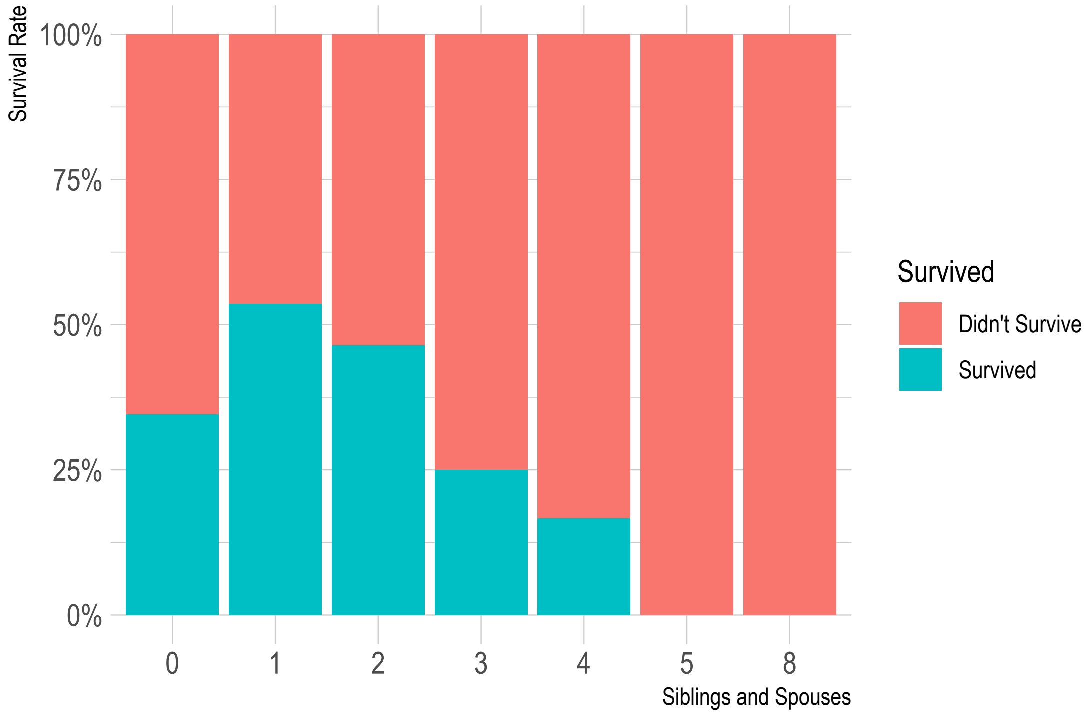
From this Plot, it seems clear that people from the upper classes had higher survival rates, thought this seemed obvious from the beginning.

2.8.2 Siblings and Spouses Vs Survived

```
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()
```

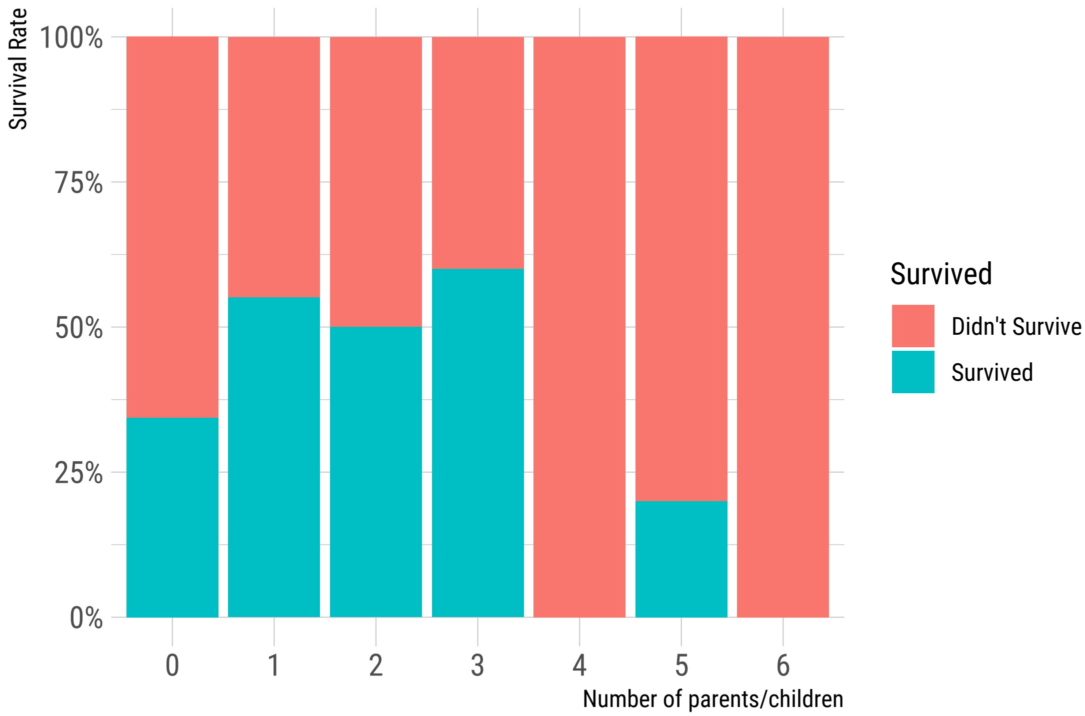
```
gSibSpSurvived
```



This plot highlight a point that might be counter-intuitive that people that have no siblings or spouses had lower survival rates than those with at least one sibling or a spouse

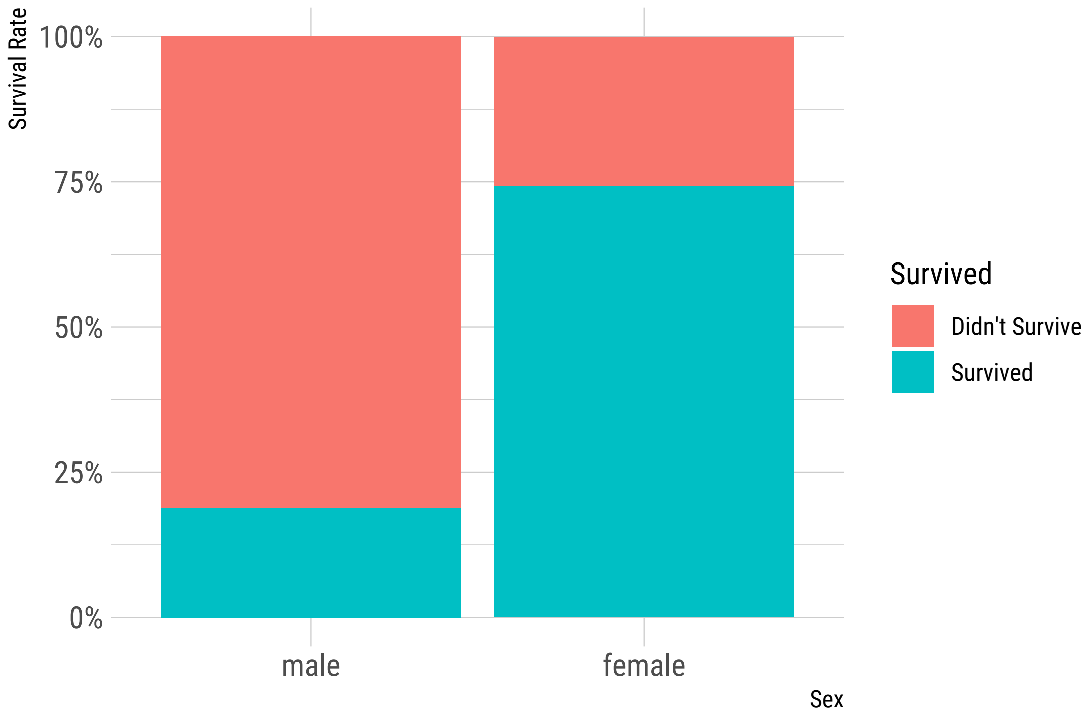
2.8.3 Number of children/parents Vs Survived

```
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
```



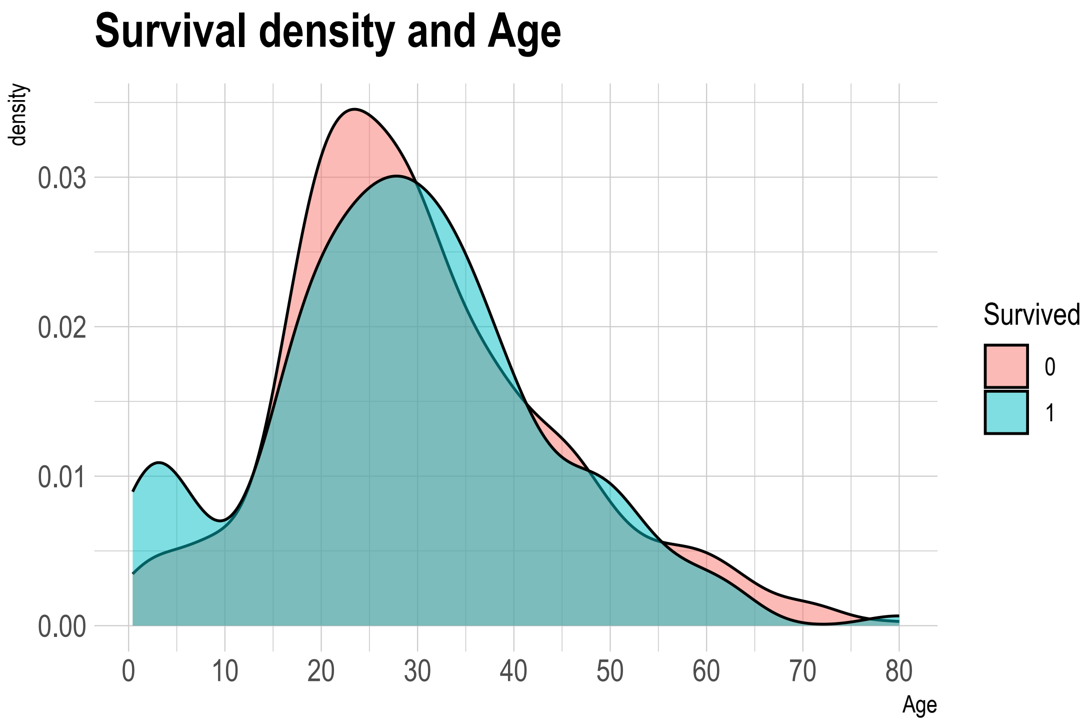
2.8.4 Gender VS Survived

```
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
```



2.8.5 Survival Density and Age

```
gSurvivalAgeDensity <- ggplot(train[(!is.na(train$Survived) & !is.na(train$Age)),],
            aes(x = Age,
                fill = Survived)) +
  geom_density(alpha=0.5,
               aes(fill=as_factor(Survived))) +
  labs(title="Survival density and Age") +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 10)) +
  theme_ipsum()

gSurvivalAgeDensity
```




```
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     27.5   14.0        0.742      0.438
## 2 male       30.7   14.4        0.189      0.392
```

2.9 Sampling

2.9.1 Random sample of size of 50 (Age)

```
sample_50 <- sample_n(train,size = 50,replace = T) %>%
  summarise(mean = mean(Age), sd = sd(Age))
sample_50
```

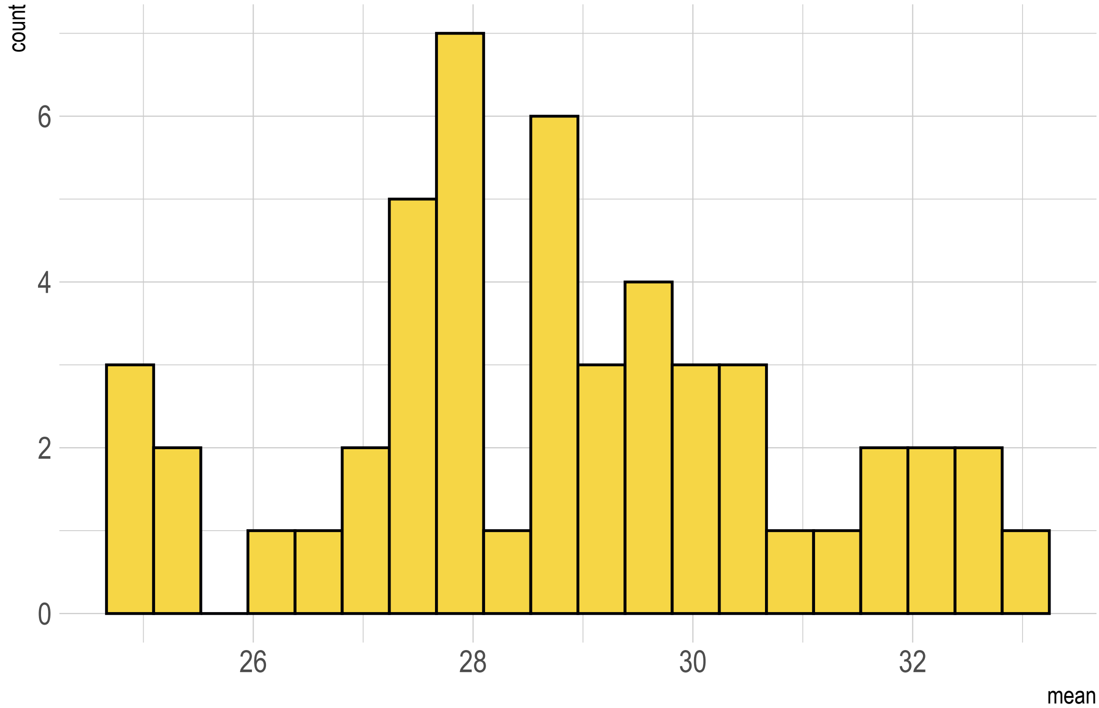
```
## # A tibble: 1 x 2
##    mean    sd
##   <dbl> <dbl>
## 1  29.7  13.3
```

2.9.2 Sampling distrubution with a fixed size (50) (mean)

2.9.2.1 50 Samples

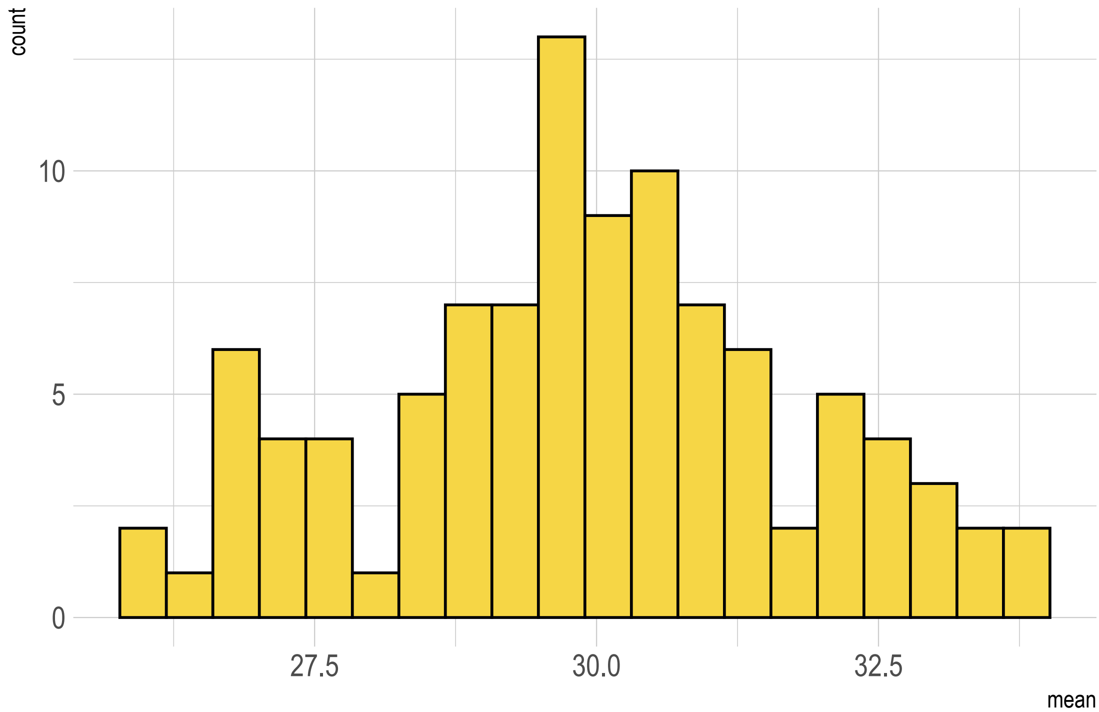
```
sample_means50 <- rep_sample_n(train,size = 50, reps = 50,replace =T) %>%
  summarise(mean = mean(Age))
sample_means50 %>% ggplot(aes(x = mean)) +
```

```
  geom_histogram(bins=20, colour = inferno(1),fill=inferno(1,begin=0.9))+
  theme_ipsum()
```



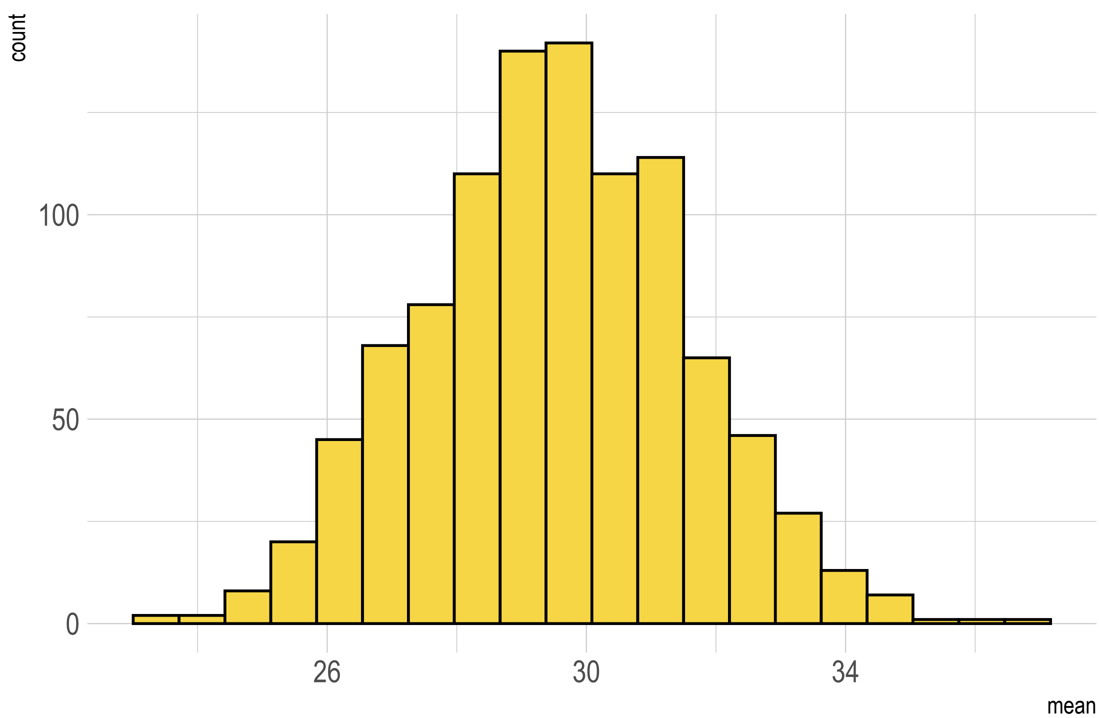
2.9.2.2 100 Samples

```
sample_means100 <- rep_sample_n(train,size = 50, reps = 100,replace =T) %>%
  summarise(mean = mean(Age))
sample_means100 %>% ggplot(aes(x = mean)) +
  geom_histogram(bins=20, colour = inferno(1),fill=inferno(1,begin=0.9))+
  theme_ipsum()
```



2.9.2.3 1000 Samples

```
sample_means1000 <- rep_sample_n(train,size = 50, reps = 1000,replace =T) %>%
  summarise(mean = mean(Age))
sample_means1000 %>% ggplot(aes(x = mean)) +
  geom_histogram(bins=20, colour = inferno(1),fill=inferno(1,begin=0.9))+
  theme_ipsum()
```

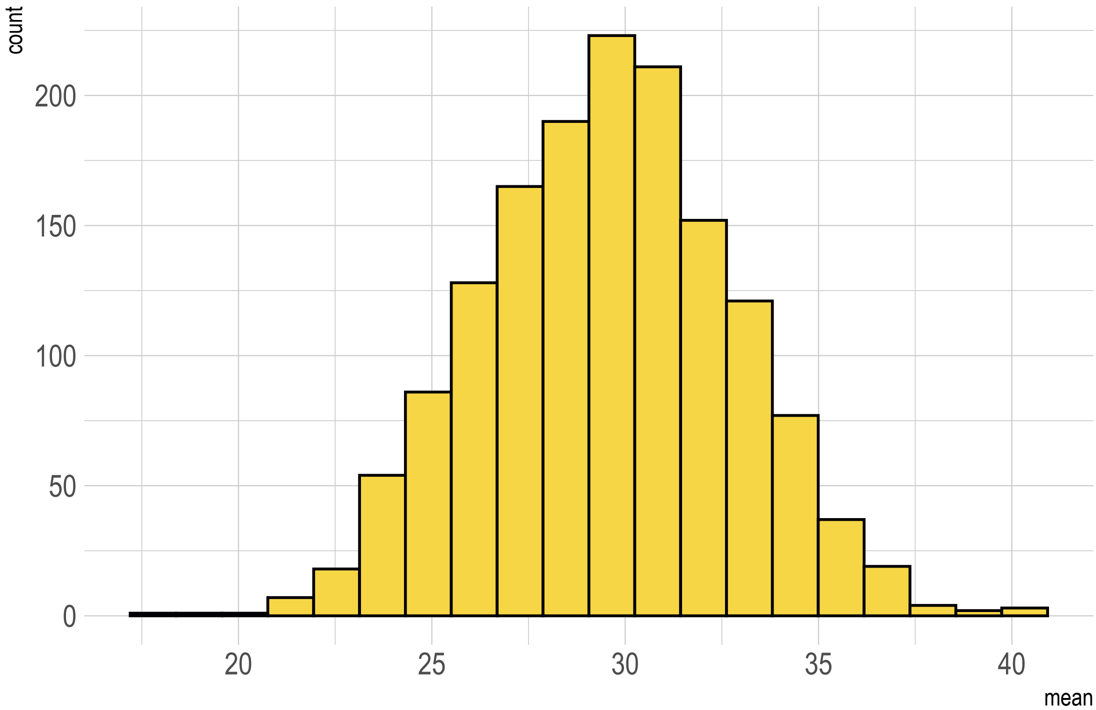


While no. of sample size increase the variability of sampling distribution decrease and the mean increase.

2.9.3 Sampling distribution with a fixed repetitions and different sizes (mean)

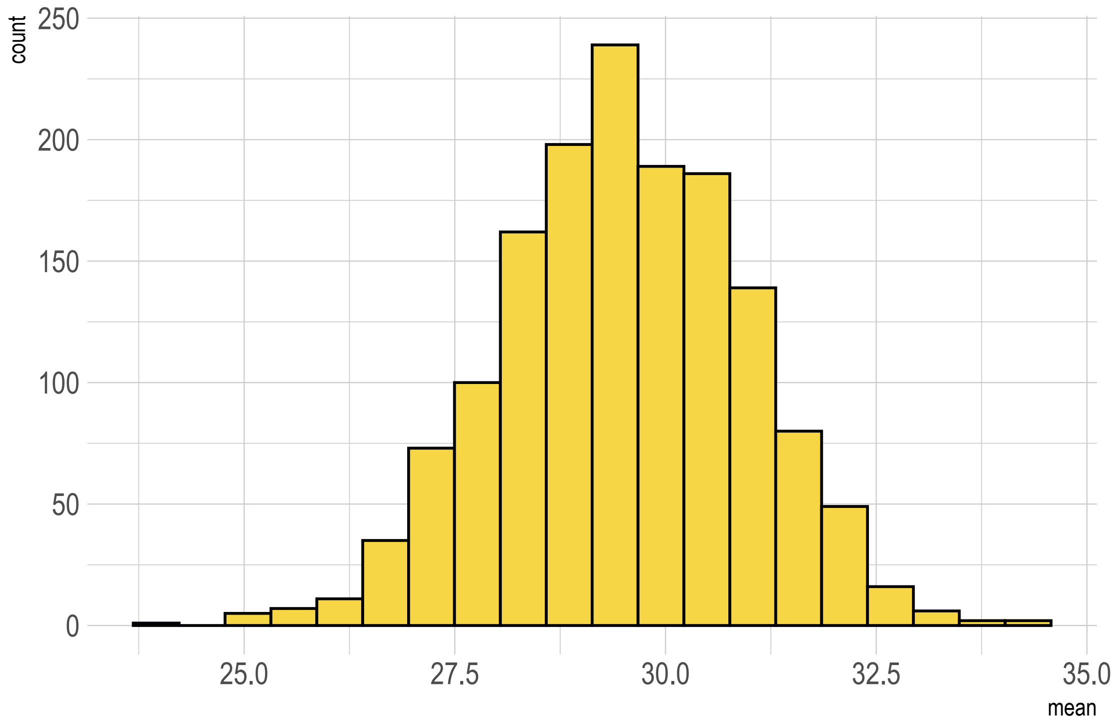
2.9.3.1 Size 20

```
sample_mean_s20 <- rep_sample_n(train,size = 20, reps = 1500,replace =T) %>%
  summarise(mean = mean(Age))
sample_mean_s20 %>% ggplot(aes(x = mean)) +
  geom_histogram(bins=20, colour = inferno(1),fill=inferno(1,begin=0.9))+
  theme_ipsum()
```



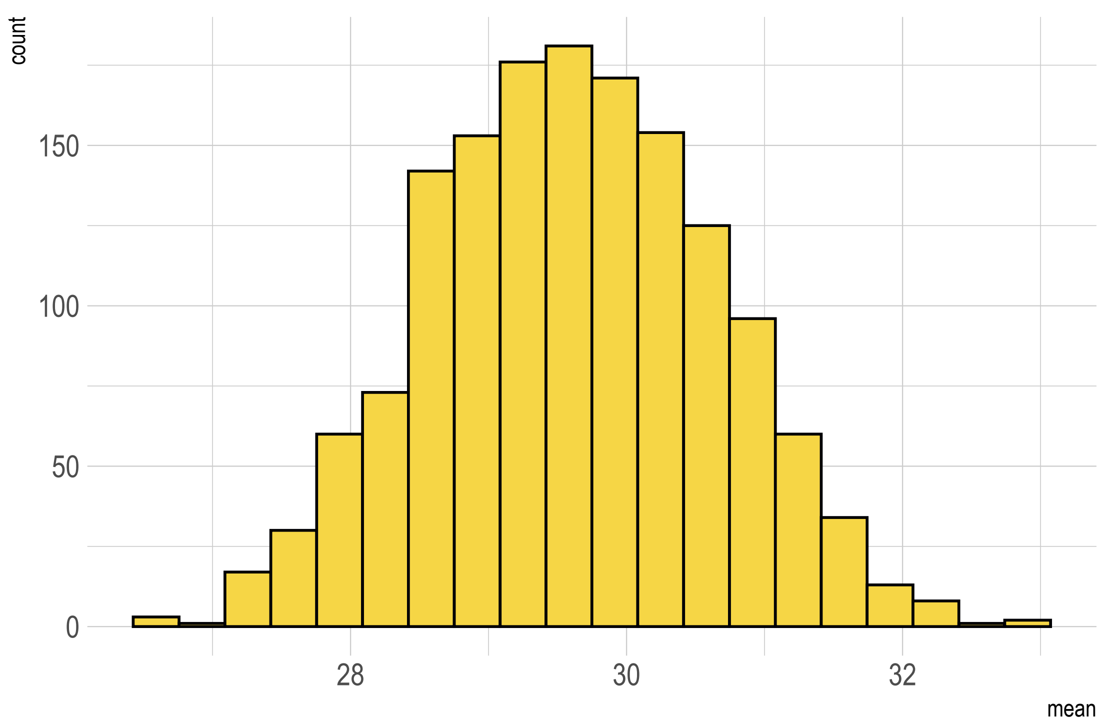
2.9.3.2 Size 100

```
sample_mean_s100 <- rep_sample_n(train,size = 100, reps = 1500,replace =T) %>%
  summarise(mean = mean(Age))
sample_mean_s100 %>% ggplot(aes(x = mean)) +
  geom_histogram(bins=20, colour = inferno(1),fill=inferno(1,begin=0.9))+
  theme_ipsum()
```



2.9.3.3 Size 200

```
sample_mean_s200 <- rep_sample_n(train,size = 200, reps = 1500,replace =T) %>%
  summarise(mean = mean(Age))
sample_mean_s200 %>% ggplot(aes(x = mean)) +
  geom_histogram(bins=20, colour = inferno(1),fill=inferno(1,begin=0.9))+
  theme_ipsum()
```

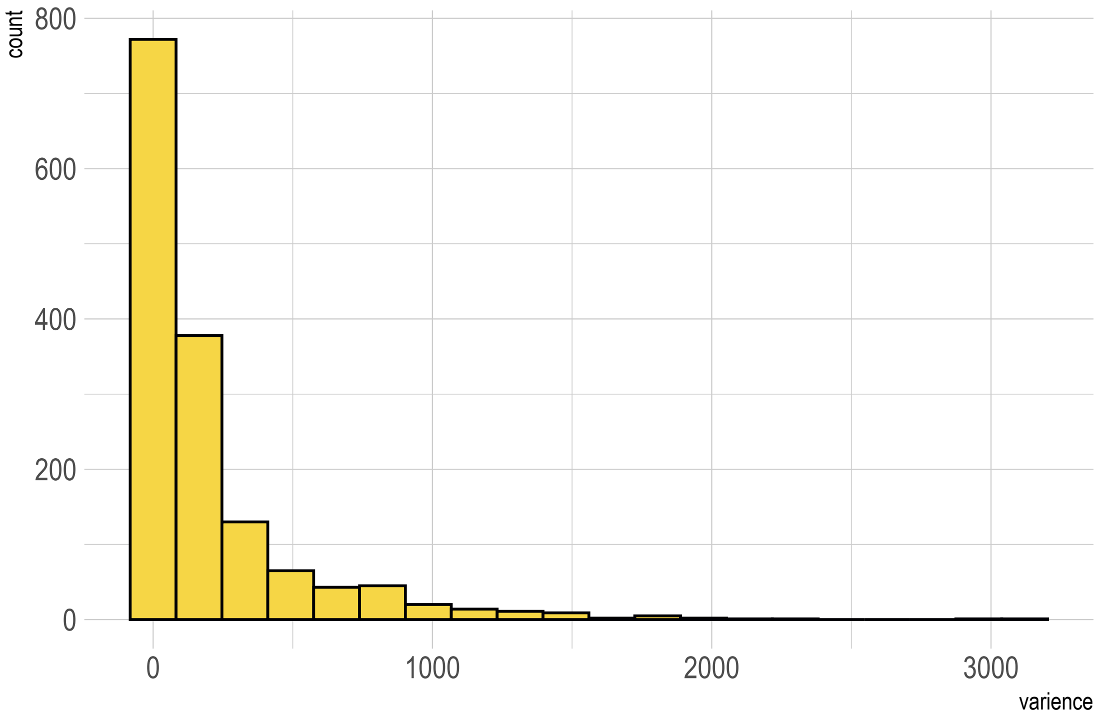


While no. of sample size increase the variability of sampling distribution decrease,and The sample distribution mean will be normally distributed as long as the sample size is more than 30.

2.9.4 sampling distribution of variance with different sizes and fixed repitions.

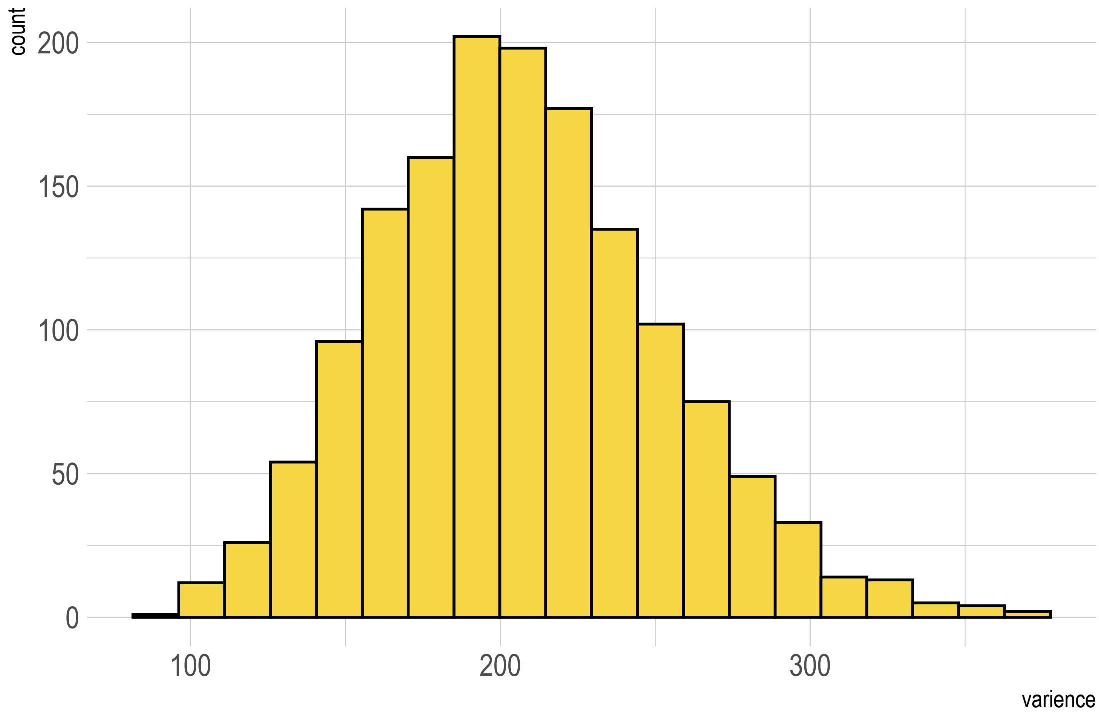
2.9.4.1 Size 2 and 1500 reptitions

```
sample_U1500 <- rep_sample_n(train,size = 2, reps = 1500,replace =T) %>%
  summarise(varience = var(Age))
sample_U1500 %>% ggplot(aes(x = varience)) +
  geom_histogram(bins=20, colour = inferno(1),fill=inferno(1,begin=0.9))+
  theme_ipsum()
```



Size 50 and 1500 reptitions

```
sample_U1500 <- rep_sample_n(train,size = 50, reps = 1500,replace =T) %>%
  summarise(varience = var(Age))
sample_U1500 %>% ggplot(aes(x = varience)) +
  geom_histogram(bins=20, colour = inferno(1),fill=inferno(1,begin=0.9))+
  theme_ipsum()
```



2.10 MME and MLE

$$M_1 = E(x) = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$M_2 = E(x^2) = \sigma^2 + (E(x))^2 = \frac{1}{n}\sum_{i=1}^{n} {x_i}^2$$

$$\mu = \frac{1}{2}\sum_{i=1}^{n} x_i = M_1$$

$$\sigma^2 = E(x^2) - (E(x))^2 = \frac{1}{n}\sum_{i=1}^{n} {x_i}^2 - \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2$$

2.10.1 Size 50

```
# Sampling
n_size = 50
nsample <- sample_n(train,size=n_size,replace=T)
# MME
mean_est1=sum(nsample$Age) / n_size
var_est1 = (sum(nsample$Age^2) / n_size) - (mean_est1)^2
mean_est1
```

```
## [1] 29.23
```

```
var_est1
```

```
## [1] 236.8321
```

```
bias(actual = mean(train$Age,na.rm = T), predicted = mean_est1)
```

```
## [1] 0.3391246
```

```r
# MLE
NLL = function(pars,data){
  # Extract parameters from the vector
  mu = pars[1]
  sigma = pars[2]
  NLL = -sum(dnorm(x = data , mean = mu, sd = sigma , log = TRUE))
  # Log of pdf
  # negative to get max
}
mle <- optim(par = c(mu= .2 , sigma = 1.5),fn=NLL,data=nsample$Age,control = list(parscale=c(mu= .2 , si
mle$par[1]
```

```
##       mu
## 29.23726
```

```r
bias(actual = mean(train$Age,na.rm = T), predicted = mle$par[1])
```

```
## [1] 0.3318609
```

2.10.2 Size 200

```r
# Sampling
n_size = 200
nsample <- sample_n(train,size=n_size,replace=T)
# MME
mean_est1=sum(nsample$Age) / n_size
var_est1 = (sum(nsample$Age^2) / n_size) - (mean_est1)^2
mean_est1
```

```
## [1] 29.1108
```

```r
var_est1
```

```
## [1] 216.4245
```

```r
bias(actual = mean(train$Age,na.rm = T), predicted = mean_est1)
```

```
## [1] 0.4583246
```

```r
# MLE
NLL = function(pars,data){
  # Extract parameters from the vector
  mu = pars[1]
  sigma = pars[2]
  NLL = -sum(dnorm(x = data , mean = mu, sd = sigma , log = TRUE))
  # Log of pdf
  # negative to get max
}
mle <- optim(par = c(mu= .2 , sigma = 1.5),fn=NLL,data=nsample$Age,control = list(parscale=c(mu= .2 , si
mle$par[1]
```

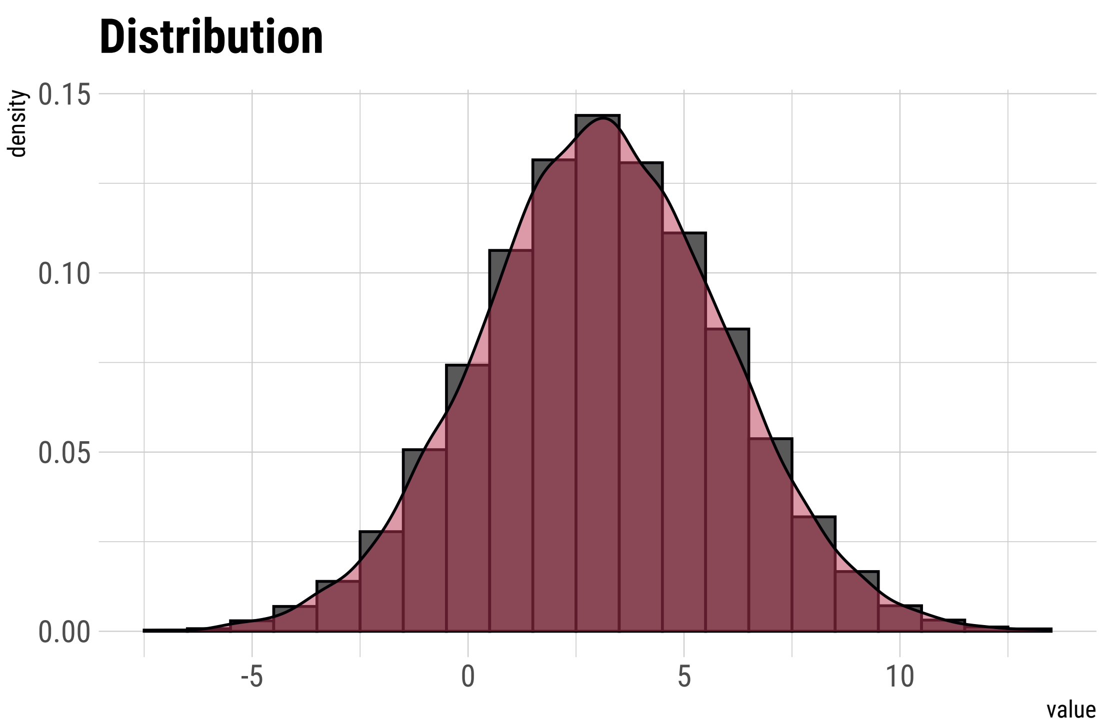
```
##       mu
## 29.10331
```

```r
bias(actual = mean(train$Age,na.rm = T), predicted = mle$par[1])
```

```
## [1] 0.4658188
```

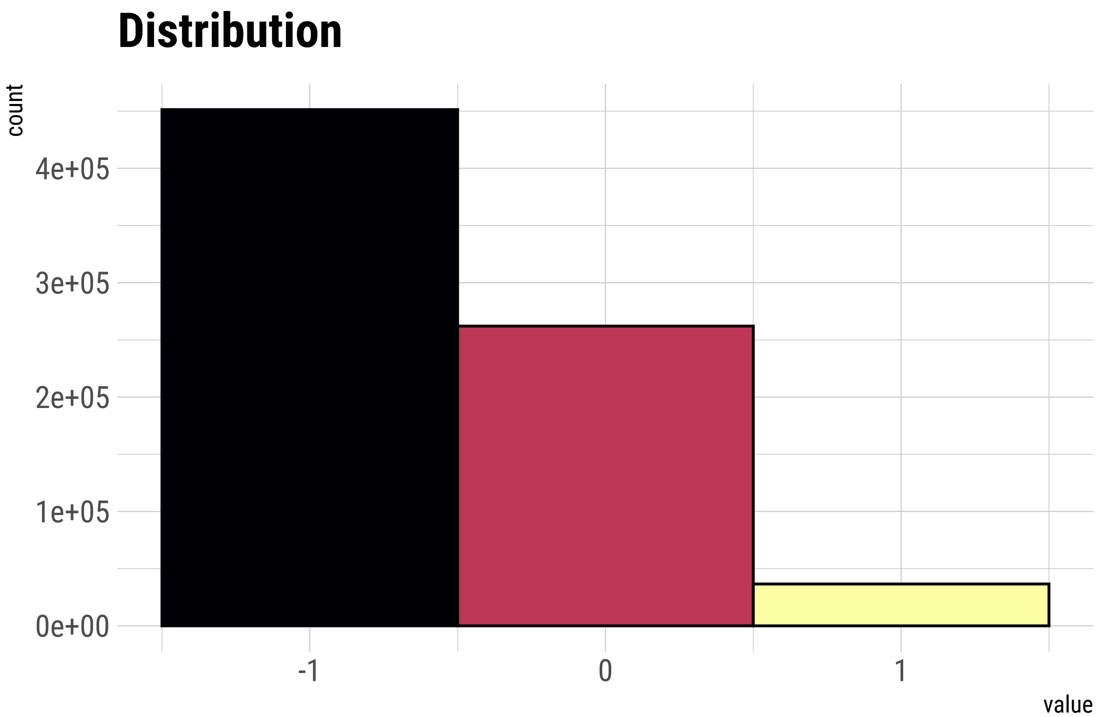
2.11 Male Age - Female Age

```
age_male <- train %>%
  select(Age,Sex) %>%
  mutate(Sex = as_factor(Sex)) %>%
  filter(Sex == "male")
age_female <- train %>%
  select(Age,Sex) %>%
  mutate(Sex = as_factor(Sex)) %>%
  filter(Sex == "female")

sample_age_male_50 <- age_male %>%
  rep_sample_n(size = 50,reps = 15000,replace = T) %>%
  summarise(age_male_bar = mean(Age,na.rm = T))
sample_age_female_50 <- age_female %>%
  rep_sample_n(size = 50,reps = 15000,replace = T) %>%
  summarise(age_female_bar = mean(Age,na.rm = T))

samplediff_means <- sample_age_male_50$age_male_bar -  sample_age_female_50$age_female_bar %>%
  as_tibble()

gsamplediff_means <- samplediff_means %>%
  ggplot(aes(value, y = ..density..)) +
  geom_histogram(bins = 25,binwidth = 1,color=inferno(1,alpha=1)) +
  geom_density(fill=inferno(1,begin = 0.5,alpha = 0.5),color = inferno(1,begin=0)) +
  ggtitle("Distribution") +
  theme_ipsum_rc()
gsamplediff_means
```



2.12 Survived Male - Survived Female

```
survived_male <- train %>%
  select(Survived,Sex) %>%
  filter(Sex == "male")

survived_female <- train %>%
  select(Survived,Sex) %>%
  filter(Sex == "female")

sample_survive_male_50 <- survived_male %>%
  rep_sample_n(size = 50,reps = 15000,replace = T)
sample_survive_female_50 <- survived_female %>%
  rep_sample_n(size = 50,reps = 15000,replace = T)
samplediff_survived <- sample_survive_male_50$Survived - sample_survive_female_50$Survived %>%
  as_tibble()

gsamplediff_survived <- samplediff_survived %>%
  ggplot(aes(value)) +
  geom_histogram(bins = 25,binwidth = 1,color=inferno(1,alpha=1),fill = inferno(3)) +
  ggtitle("Distribution") +
  theme_ipsum_rc()
gsamplediff_survived
```



The plot is a bit hard to understand so it needs a bit of explanation. -1 represents the female survived, 0 represents the neither of them survived, and 1 means that the male survived

After inspecting the plot, it becomes Crystal clear that there was a bias in the rescue process where rescuers preferred to save female more than males.

2.13 Confidence Intervals

```
n_size = 10
nsample <- sample_n(train,size= n_size,replace=T)

# Calculate The Mean
mean_est1=mean(nsample$Age,na.rm = T)

# Calculate The Standard Deviation
sd_est1=sd(nsample$Age,na.rm = T)

# Computing The Error Using the qnorm() Function to Calculate The Normal Distribution
error=qt(0.975,df = 9, lower.tail = T)*(sd_est1/sqrt(n_size)) # at lower.tail = T probability is P[X<=x

#Determining The Mean Interval[]
left <- mean_est1-error
right <- mean_est1+error

print(paste0("[",left,",",right,"]"))
```

```
## [1] "[26.183191484837,51.416808515163]"
```

```
n_size = 50
nsample <- sample_n(train,size=n_size,replace=T)

# Calculate The Mean
mean_est2=mean(nsample$Age,na.rm = T)

# Calculate The Standard Deviation
sd_est2=sd(nsample$Age,na.rm = T)

# Computing The Error Using the qnorm() Function to Calculate The Normal Distribution
error=qnorm(0.975)*(sd_est1/sqrt(n_size))

#Determining The Mean Interval[]
left <- mean_est2-error
right <- mean_est2+error

print(paste0("[",left,",",right,"]"))
```

```
## [1] "[25.0897400803685,34.8670599196315]"
```

2.14 Multiplication and addition of constant to the sample values

2.14.1 Multiplication

```
n_size = 200

nsample <- sample_n(train,size = n_size,replace = T)

nsample %>%
  summarise(Age_mean = mean(Age), Age_sd = sd(Age),Age_var = sd(Age)^2)
```

```
## # A tibble: 1 x 3
##   Age_mean Age_sd Age_var
##      <dbl>  <dbl>   <dbl>
## 1     30.3   13.5    181.
```

```
nsample$Age <- nsample$Age *5
nsample %>%
  summarise(Age_mean = mean(Age), Age_sd = sd(Age),Age_var = sd(Age)^2)
```

```
## # A tibble: 1 x 3
##   Age_mean Age_sd Age_var
##      <dbl>  <dbl>   <dbl>
## 1     151.   67.3   4523.
```

2.14.2 Addition

```
n_size = 200

nsample <- sample_n(train,size = n_size,replace = T)

nsample %>%
  summarise(Age_mean = mean(Age), Age_sd = sd(Age),Age_var = sd(Age)^2)
```

```
## # A tibble: 1 x 3
##   Age_mean Age_sd Age_var
##      <dbl>  <dbl>   <dbl>
## 1     30.5   14.2    200.
```

```
nsample$Age <- nsample$Age + 5
nsample %>%
  summarise(Age_mean = mean(Age), Age_sd = sd(Age),Age_var = sd(Age)^2)
```

```
## # A tibble: 1 x 3
##   Age_mean Age_sd Age_var
##      <dbl>  <dbl>   <dbl>
## 1     35.5   14.2    200.
```

3 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

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

3.1 Rules

3.1.1 Non-weighted Data

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

3.1.2 Weighted Data

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

3.2 Kernel Function

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

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)

3.3 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

3.4 2D

h : could be matrix (different h in different directions)

The choice of norm comes into $d \geq 2$

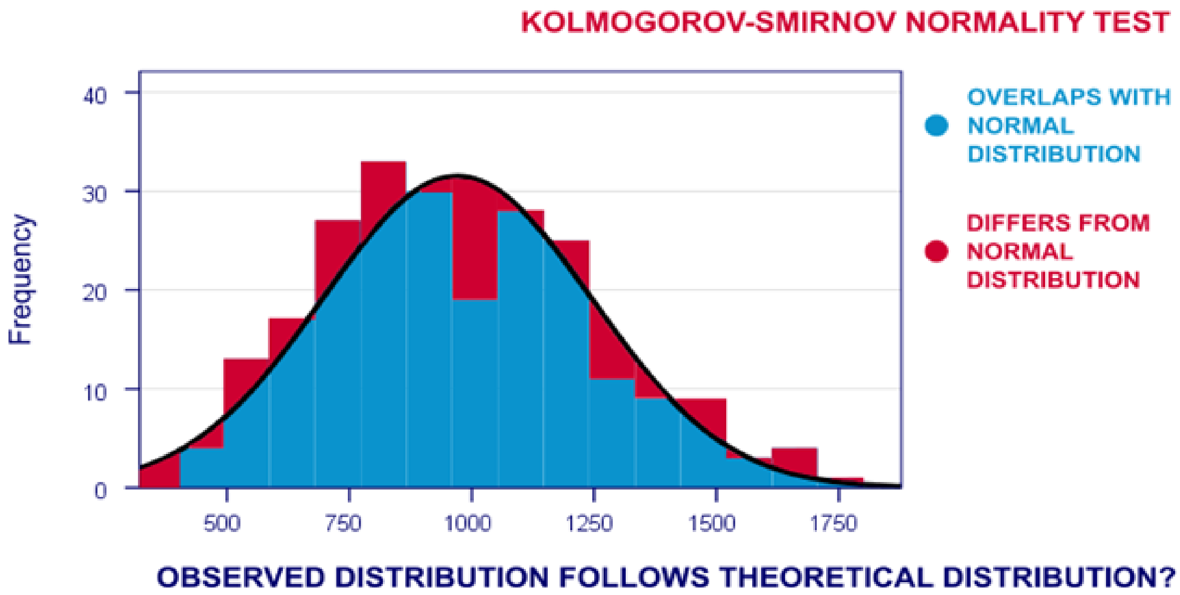
The p-norm is $||x||_p := (\sum_{i=1} |x|^p)^{\frac{1}{p}}$ - norm-p =1 Manhattan distance - norm-p =2 Euclidean norm - norm-p =inf maximum norm (it's not obvious in every case which norm is the correct one)

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

4 Kolmogorov-Smirnov Test

Non parametric test

Kolmogorov-Smirnov Test is used to: 1. Decide if a sample comes from a population with an expected continuous distribution (mostly normal distribution) 2. To test for the difference in the shape of two sample distributions

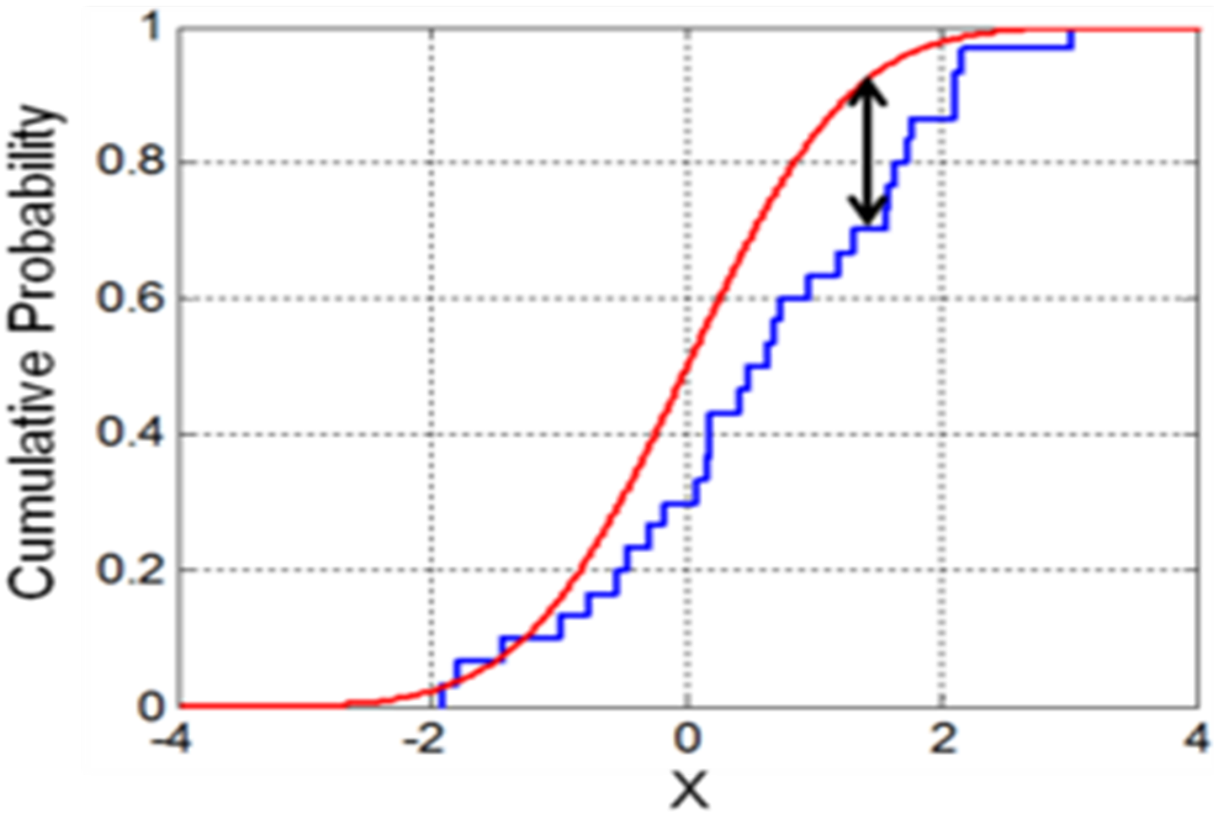


Compare overall shape of distribution, not specifically parameter

Kolmogorov-Smirnov Test is defined by a hypothesis: H_0 : the data follow specific distribution $F(x) = F_T(x)$ H_a : the data don't follow specific distribution $F(x) \neq F_T(x)$

KS-test is made between some theoretical cumulative distribution function ($F_T(x)$), and a sample cumulative distribution function ($F_s(x)$) that measured by the statistic D, which is the greatest vertical distance between them.

$$D = \underbrace{\sup}_{x} |F_s(x) - F_t(x)|$$



1. Determine x values
2. Detemrine frequency of each observation
3. Calculate cumulative frequency
4. Calculate $F_s(x) \rightarrow \frac{\text{cumulative frequence}}{N}$ when $n > 30$ (z-scores), when $n < 30$ (t-scores)
5. calculate $F_t(x)$
6. Calculate $D \rightarrow F_s(x) - F_T(x)$
7. Choose maximum D
8. Calculate P-value
9. Determine Kolmogorov's quartile
10. Compare p-value with Kolmogorove's quartile