Titanic Survival Prediction

```
library(data.table)
library(tidyverse)
library(dplyr)
library(stringr)
library(caret)
library(randomForest)
library(e1071)
library(rpart)
```

```
train <- fread("train.csv")%>% data.table()
test <-  fread("test.csv") %>% data.table()
test$Survived <- NA
combi = rbind(train, test)
ntrain <- nrow(train)
```

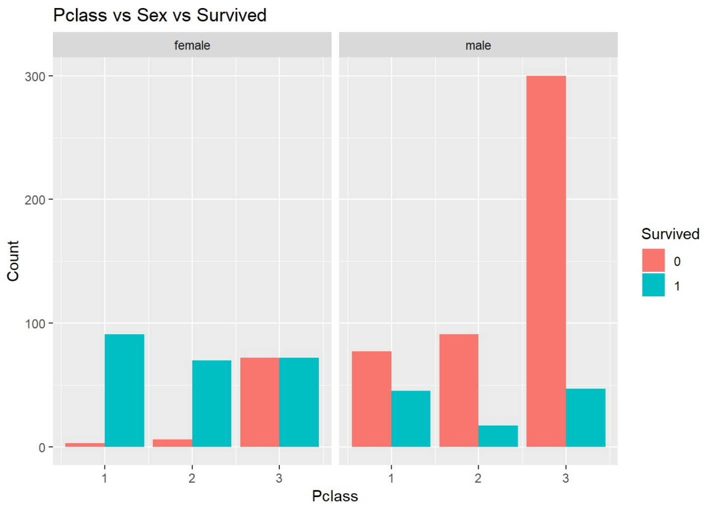
```
str(train)
```

```
## Classes 'data.table' and 'data.frame':   891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heikki
nen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ...
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

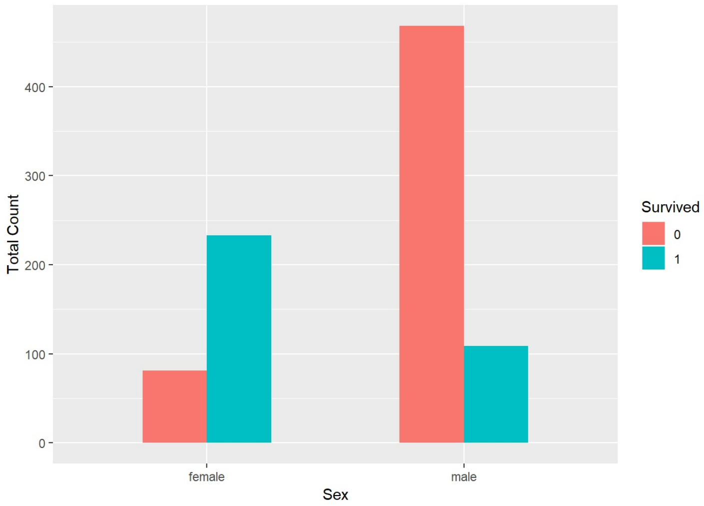
```
str(test)
```

```
## Classes 'data.table' and 'data.frame':   418 obs. of  12 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : chr  "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)" "Myles, Mr. Thomas Francis" "Wirz
, Mr. Albert" ...
##  $ Sex        : chr  "male" "female" "male" "male" ...
##  $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : chr  "330911" "363272" "240276" "315154" ...
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : chr  "" "" "" "" ...
##  $ Embarked   : chr  "Q" "S" "Q" "S" ...
##  $ Survived   : logi  NA NA NA NA NA NA ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

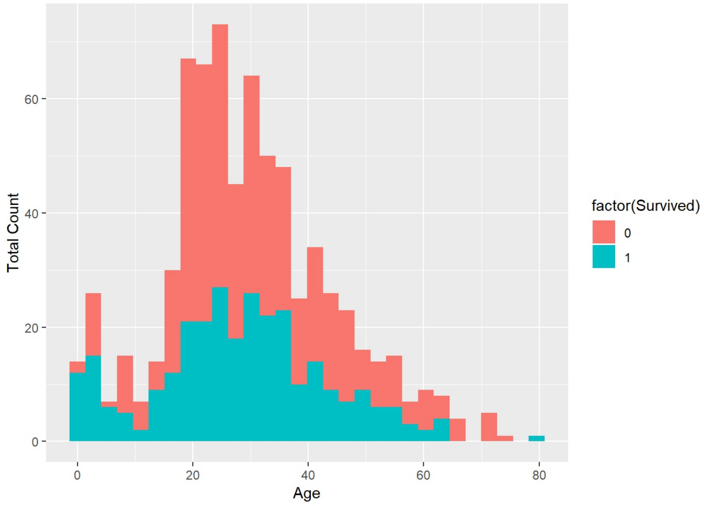
```
ggplot(combi[1:ntrain,], aes(x = factor(Pclass), fill = factor(Survived))) +
  geom_bar(width = 0.5, position="dodge") +
  xlab("Pclass") +
  ylab("Total Count") +
  labs(fill = "Survived")
```



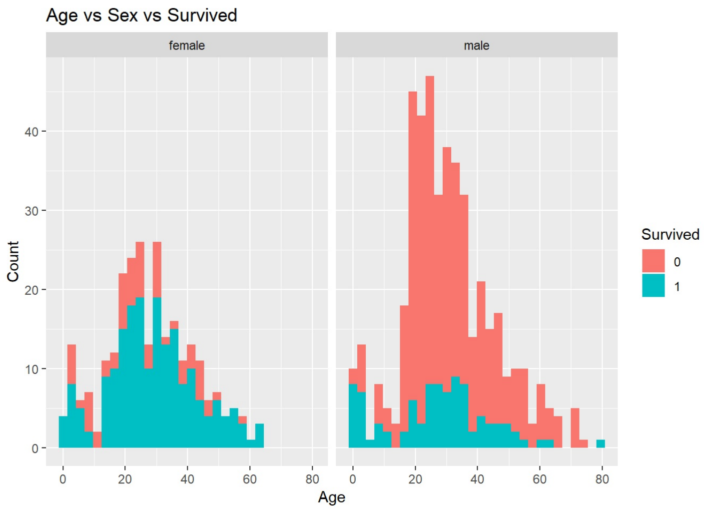

```
ggplot(combi[1:ntrain,], aes(x = factor(Sex), fill = factor(Survived))) +
  geom_bar(width = 0.5, position="dodge") +
  xlab("Sex") +
  ylab("Total Count") +
  labs(fill = "Survived")
```



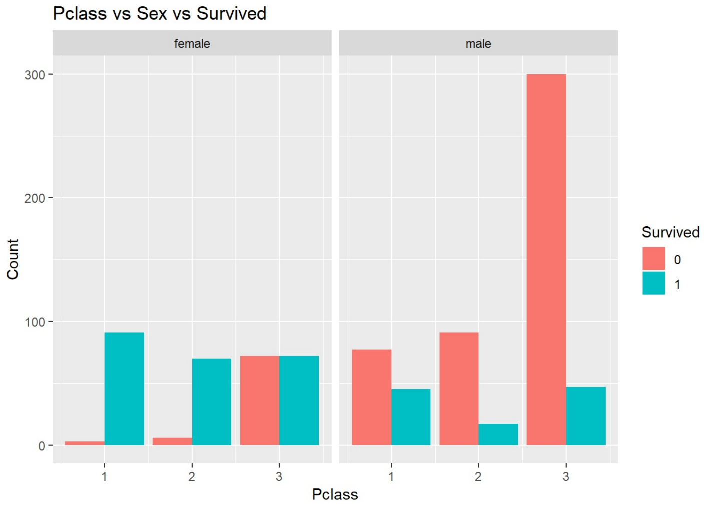

```
ggplot(subset(combi[1:ntrain,],!is.na(Age)), aes(x = Age, fill = factor(Survived))) +
 geom_histogram(bins = 30) +
  xlab("Age") +
  ylab("Total Count")
```




```
ggplot(subset(combi[1:ntrain,],!is.na(Age)), aes(Age, fill = factor(Survived))) +
  geom_histogram(bins=30) +
  xlab("Age") +
  ylab("Count") +
  facet_grid(.~Sex)+
  scale_fill_discrete(name = "Survived") +
  ggtitle("Age vs Sex vs Survived")
```

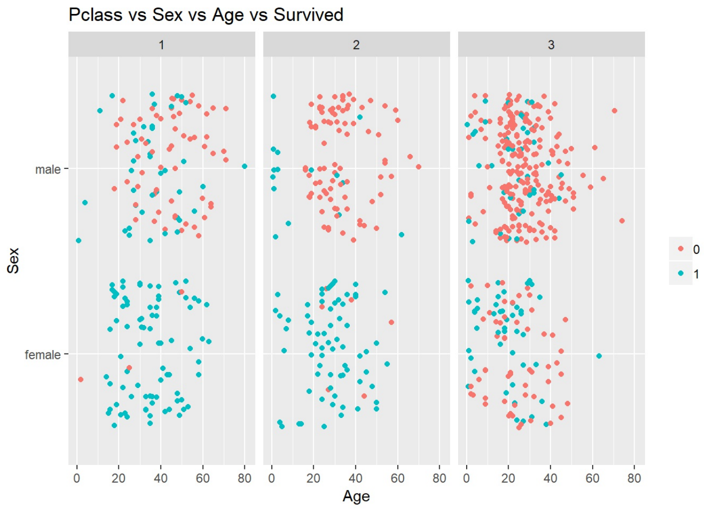



```
ggplot(combi[1:ntrain,], aes(Pclass, fill = factor(Survived))) +
  geom_bar(stat = "count",position = "dodge")+
  xlab("Pclass") +
  facet_grid(.~Sex)+
  ylab("Count") +
  scale_fill_discrete(name = "Survived") +
  ggtitle("Pclass vs Sex vs Survived")
```




```
ggplot(combi[1:ntrain,], aes(x = Age, y = Sex)) +
  geom_jitter(aes(colour = factor(Survived))) +
  theme(legend.title = element_blank())+
  facet_wrap(~Pclass) +
  labs(x = "Age", y = "Sex", title = "Pclass vs Sex vs Age vs Survived")+
  scale_fill_discrete(name = "Survived") +
  scale_x_continuous(name="Age",limits=c(0, 81))
```

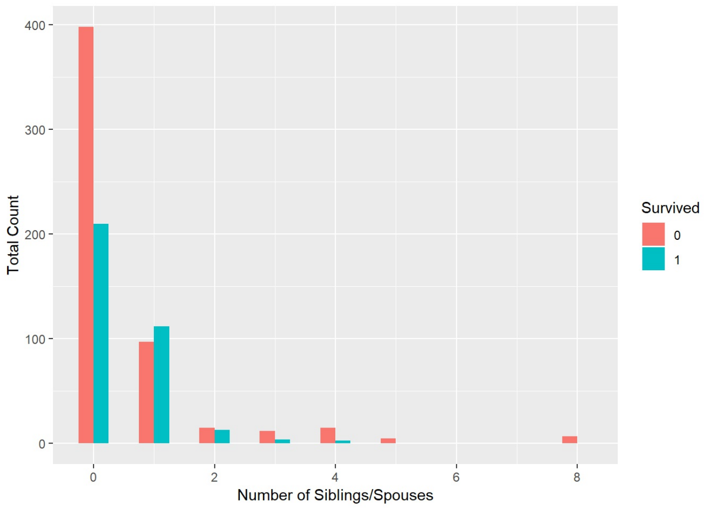
```
## Warning: Removed 177 rows containing missing values (geom_point).
```



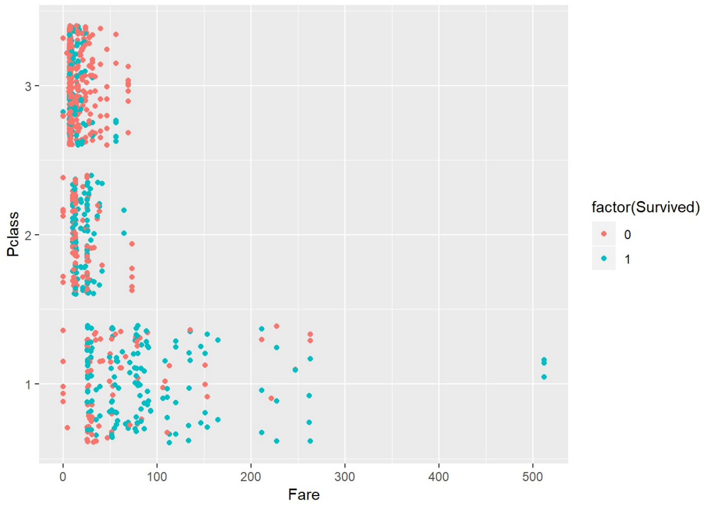

```
table(combi[1:ntrain,]$SibSp)
```

```
##
##   0   1   2   3   4   5   8
## 608 209  28  16  18   5   7
```

```
ggplot(combi[1:ntrain,], aes(x = SibSp, fill = factor(Survived))) +
  geom_histogram(binwidth=0.5, position="dodge") +
  xlab("Number of Siblings/Spouses") +
  ylab("Total Count") +
  labs(fill = "Survived")
```




```
ggplot(combi[1:ntrain,], aes(x=Fare , y = Pclass)) +
  geom_jitter(aes(color = factor(Survived)))
```

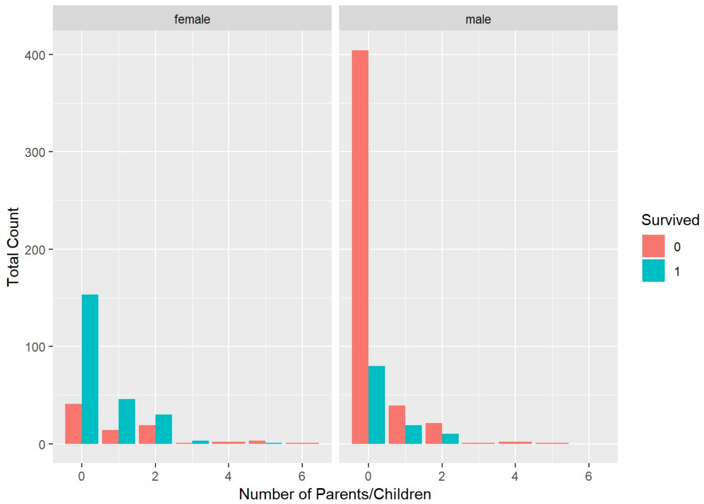



```
table(combi[1:ntrain,]$Parch)
```

```
##
##   0   1   2   3   4   5   6
## 678 118  80   5   4   5   1
```

```
ggplot(combi[1:ntrain,], aes(x = Parch, fill = factor(Survived))) +
  geom_histogram(binwidth=0.5, position="dodge" , stat = "count") +
  facet_grid(.~Sex)+
  xlab("Number of Parents/Children") +
  ylab("Total Count") +
  labs(fill = "Survived")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



finding percentage of missing values in each column

```
sapply(combi, function(x) {ifelse(sum(is.na(x))!=0 , round(sum(is.na(x))*100/nrow(combi),2) , round(sum(x=="")*
100/nrow(combi),2))})
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##        0.00       31.93        0.00        0.00        0.00       20.09
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##        0.00        0.00        0.00        0.08       77.46        0.15
```

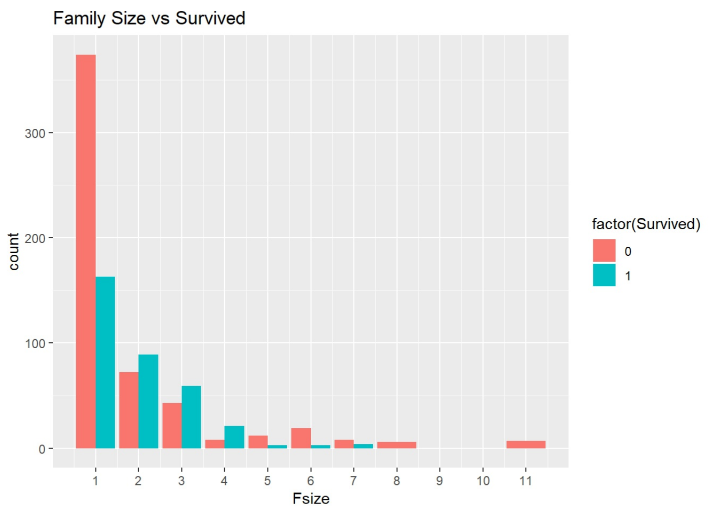
Ignoring attribute Cabin since 77% of data is missing, replacing NA in Embarked to S

```
table(combi$Embarked)
```

```
##
##       C   Q   S
##   2 270 123 914
```

```
combi$Embarked[combi$Embarked ==""] <- 'S'
combi$Embarked[combi$Embarked ==""] <- 'S'
combi$Fsize <- combi$SibSp + combi$Parch + 1
```

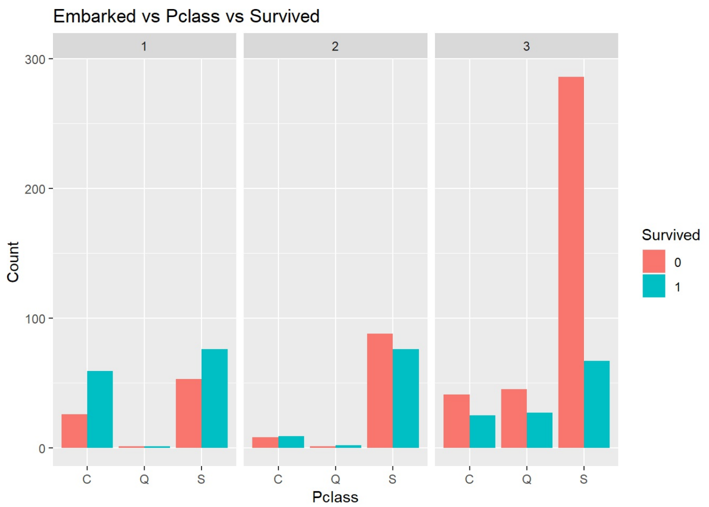
```
ggplot(combi[1:ntrain,] , aes(x=Fsize , fill = factor(Survived)))+geom_bar(stat = "count" , position = "dodge")
+
  scale_x_continuous(breaks = seq(0,max(combi[1:ntrain,]$Fsize),1)) +
  ggtitle("Family Size vs Survived")
```




```
table(combi$Pclass)
```

```
##
##   1   2   3
## 323 277 709
```

```
ggplot(combi[1:ntrain,], aes(Embarked, fill = factor(Survived))) +
  geom_bar(stat = "count",position = "dodge")+
  xlab("Pclass") +
  ylab("Count") +
  facet_grid(.~Pclass) +
  scale_fill_discrete(name = "Survived") +
  ggtitle("Embarked vs Pclass vs Survived")
```



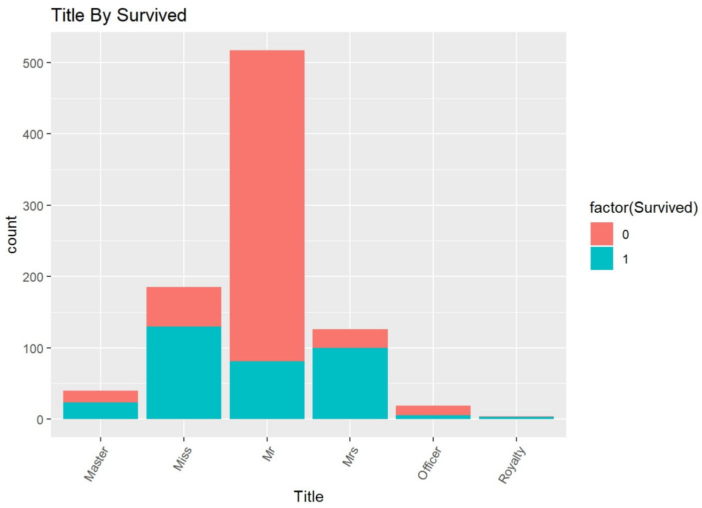

```
combi$Title <- NA
combi$Title <- sapply(combi$Name , function(x) str_trim(str_split(x,"[,.]")[[1]][2],side ="both"))
unique(combi$Title)
```

```
##  [1] "Mr"           "Mrs"          "Miss"         "Master"
##  [5] "Don"          "Rev"          "Dr"           "Mme"
##  [9] "Ms"           "Major"        "Lady"         "Sir"
## [13] "Mlle"         "Col"          "Capt"         "the Countess"
## [17] "Jonkheer"     "Dona"
```

```
combi$Title[combi$Title%in%c("Mme")] <- "Mrs"
combi$Title[combi$Title%in%c("Mlle","Ms")] <- "Miss"
officer <- c('Capt', 'Col', 'Don', 'Dr', 'Major', 'Rev')
royalty <- c('Dona', 'Lady', 'the Countess','Sir', 'Jonkheer')
combi$Title[combi$Title %in% royalty]  <- 'Royalty'
combi$Title[combi$Title %in% officer]  <- 'Officer'
```

```
ggplot(combi[1:ntrain,] , aes(x=Title , fill = factor(Survived)) +
  geom_histogram(bins = 6 , stat = "count") +
  ggtitle("Title By Survived")+
  theme(axis.text.x=element_text(angle=60, hjust=1))
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



imputing missing age by predicting the age based on variables Pclass,Sex,title,SibSp,Parch,fare,Title

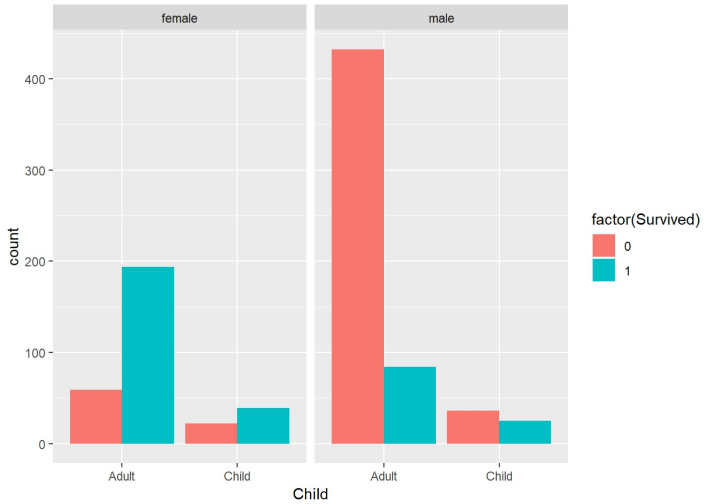
```
agefit <- rpart(Age~Pclass+Sex+Embarked+SibSp+Parch+Fsize + Fare + Title,
                data = combi[!is.na(combi$Age),] , method = "anova")
summary(agefit)
```

```
## Call:
## rpart(formula = Age ~ Pclass + Sex + Embarked + SibSp + Parch +
##     Fsize + Fare + Title, data = combi[!is.na(combi$Age), ],
##     method = "anova")
##   n= 1046
##
##           CP nsplit rel error    xerror       xstd
## 1 0.21028409      0 1.0000000 1.0021794 0.04531621
## 2 0.10512853      1 0.7897159 0.7921058 0.03520690
## 3 0.09537135      2 0.6845874 0.7502329 0.03593644
## 4 0.01436395      3 0.5892160 0.5953120 0.03049945
## 5 0.01266967      4 0.5748521 0.6028027 0.03090432
## 6 0.01056208      5 0.5621824 0.5919662 0.03086995
## 7 0.01000000      6 0.5516203 0.5884412 0.03089030
##
## Variable importance
##    Title     Fare   Pclass    Parch    Fsize    SibSp      Sex Embarked
##       28       16       16       11       11        8        8        3
##
## Node number 1: 1046 observations,    complexity param=0.2102841
##   mean=29.88114, MSE=207.5502
##   left son=2 (266 obs) right son=3 (780 obs)
##   Primary splits:
##       Title  splits as  LLRRRR,       improve=0.21028410, (0 missing)
##       Pclass < 1.5      to the right, improve=0.15460490, (0 missing)
##       SibSp  < 2.5      to the right, improve=0.07107333, (0 missing)
##       Fare   < 49.5021  to the left,  improve=0.05839866, (1 missing)
##       Fsize  < 2.5      to the right, improve=0.05804572, (0 missing)
##   Surrogate splits:
##       Sex      splits as  LR,           agree=0.782, adj=0.143, (0 split)
##       SibSp    < 2.5      to the right, agree=0.773, adj=0.109, (0 split)
##       Fsize    < 4.5      to the right, agree=0.762, adj=0.064, (0 split)
##       Parch    < 1.5      to the right, agree=0.751, adj=0.023, (0 split)
##       Embarked splits as  RLR,          agree=0.748, adj=0.008, (0 split)
##
## Node number 2: 266 observations,    complexity param=0.09537135
##   mean=18.56831, MSE=164.0627
##   left son=4 (128 obs) right son=5 (138 obs)
##   Primary splits:
##       Parch < 0.5      to the right, improve=0.4744399, (0 missing)
##       Fsize < 2.5      to the right, improve=0.3884753, (0 missing)
##       Sex   splits as  RL,          improve=0.2597037, (0 missing)
##       Title splits as  LR----,      improve=0.2597037, (0 missing)
##       SibSp < 0.5      to the right, improve=0.2127207, (0 missing)
##   Surrogate splits:
##       Fsize < 1.5      to the right, agree=0.932, adj=0.859, (0 split)
##       SibSp < 0.5      to the right, agree=0.786, adj=0.555, (0 split)
##       Fare  < 13.20835 to the right, agree=0.744, adj=0.469, (0 split)
##       Sex   splits as  RL,          agree=0.711, adj=0.398, (0 split)
##       Title splits as  LR----,      agree=0.711, adj=0.398, (0 split)
##
## Node number 3: 780 observations,    complexity param=0.1051285
##   mean=33.7391, MSE=163.8521
##   left son=6 (562 obs) right son=7 (218 obs)
##   Primary splits:
##       Pclass < 1.5      to the right, improve=0.17857830, (0 missing)
##       Fare   < 24.86875 to the left,  improve=0.13383420, (1 missing)
##       Title  splits as  --LRRR,       improve=0.03939711, (0 missing)
##       Sex    splits as  RL,           improve=0.01923511, (0 missing)
##       Parch  < 0.5      to the left,  improve=0.01239742, (0 missing)
##   Surrogate splits:
##       Fare     < 26.26875 to the left,  agree=0.908, adj=0.670, (0 split)
##       Embarked splits as  RLL,          agree=0.765, adj=0.161, (0 split)
##       Title    splits as  --LLRR,       agree=0.731, adj=0.037, (0 split)
##
## Node number 4: 128 observations,    complexity param=0.01266967
##   mean=9.407578, MSE=73.70615
##   left son=8 (103 obs) right son=9 (25 obs)
##   Primary splits:
##       Fare   < 48.2     to the left,  improve=0.2915456, (0 missing)
##       Pclass < 1.5      to the right, improve=0.2562854, (0 missing)
##       Sex    splits as  RL,           improve=0.1522839, (0 missing)
##       Title  splits as  LR----,       improve=0.1522839, (0 missing)
##       SibSp  < 0.5      to the right, improve=0.0349171, (0 missing)
##   Surrogate splits:
##       Pclass < 1.5      to the right, agree=0.961, adj=0.8, (0 split)
##
## Node number 5: 138 observations,    complexity param=0.01436395
##   mean=27.06522, MSE=97.83633
```

```
##    left son=10 (69 obs) right son=11 (69 obs)
##    Primary splits:
##        Pclass    < 2.5      to the right, improve=0.2309667000, (0 missing)
##        Fare      < 26.275   to the left,  improve=0.1953148000, (0 missing)
##        Embarked splits as  RLL,           improve=0.0374038300, (0 missing)
##        SibSp     < 0.5      to the right, improve=0.0008847728, (0 missing)
##        Fsize     < 1.5      to the right, improve=0.0008847728, (0 missing)
##    Surrogate splits:
##        Fare      < 10.17085 to the left,  agree=0.935, adj=0.870, (0 split)
##        Embarked splits as  RLL,           agree=0.645, adj=0.290, (0 split)
##        SibSp     < 0.5      to the right, agree=0.514, adj=0.029, (0 split)
##        Fsize     < 1.5      to the right, agree=0.514, adj=0.029, (0 split)
##        Sex       splits as  RL,           agree=0.507, adj=0.014, (0 split)
##
## Node number 6: 562 observations,    complexity param=0.01056208
##    mean=30.37011, MSE=116.7829
##    left son=12 (361 obs) right son=13 (201 obs)
##    Primary splits:
##        Pclass    < 2.5      to the right, improve=0.03493722, (0 missing)
##        Fare      < 9.54375  to the left,  improve=0.03140571, (1 missing)
##        Title     splits as  --LRR-,       improve=0.02300209, (0 missing)
##        Embarked splits as  LRL,           improve=0.01586441, (0 missing)
##        Parch     < 3.5      to the left,  improve=0.01382681, (0 missing)
##    Surrogate splits:
##        Fare  < 10.48125 to the left,  agree=0.835, adj=0.537, (0 split)
##        Title splits as  --LRR-,       agree=0.669, adj=0.075, (0 split)
##        Sex   splits as  RL,           agree=0.651, adj=0.025, (0 split)
##
## Node number 7: 218 observations
##    mean=42.42431, MSE=180.5023
##
## Node number 8: 103 observations
##    mean=7.123786, MSE=43.27704
##
## Node number 9: 25 observations
##    mean=18.8168, MSE=89.05189
##
## Node number 10: 69 observations
##    mean=22.31159, MSE=42.00074
##
## Node number 11: 69 observations
##    mean=31.81884, MSE=108.4781
##
## Node number 12: 361 observations
##    mean=28.86288, MSE=100.2727
##
## Node number 13: 201 observations
##    mean=33.07711, MSE=135.0276
```

```
combi$Age[is.na(combi$Age)] <- predict(agefit , combi[is.na(combi$Age),])
## child or adult based on age
combi$Child[combi$Age < 18] <- 'Child'
combi$Child[combi$Age >= 18] <- 'Adult'
```

```
ggplot(data = combi[1:ntrain,] , aes(x=Child , fill = factor(Survived))) +
  geom_bar(stat = "count", position = "dodge") + facet_grid(.~Sex)
```




```
combi$Pclass <- factor(combi$Pclass)
combi$Sex <- as.integer(combi$Sex=="male")
combi$Child <- as.integer(combi$Child=="Child")
combi$Embarked <- factor(combi$Embarked)
combi$Title <- as.factor(combi$Title)

sapply(combi, function(x) {ifelse(sum(is.na(x))!=0 , round(sum(is.na(x))*100/nrow(combi),2) , round(sum(x=="")*
100/nrow(combi),2))})
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##        0.00       31.93        0.00        0.00        0.00        0.00
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##        0.00        0.00        0.00        0.08       77.46        0.00
##       Fsize       Title       Child
##        0.00        0.00        0.00
```

```
#creating indices
trainIndex <- createDataPartition(combi[1:ntrain,]$Survived,p=1,list=FALSE)

#splitting data into training/testing data using the trainIndex object
train_titanic <- combi[trainIndex,]
test_titanic <- combi[-trainIndex,]

#creating indices to split train into train and validation
Index2 <- createDataPartition(train_titanic$Survived,p=0.8,list=FALSE)
train <- train_titanic[Index2,]
validation <- train_titanic[-Index2,]
```

Logistic Regression

```
model_glm <- glm(Survived ~ Pclass+Sex+Fsize+Child+Fare+Embarked+Title ,data = train, family = binomial(link =
"logit") )
summary(model_glm)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + Fsize + Child + Fare +
##     Embarked + Title, family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1463  -0.5656  -0.3809   0.5329   2.6420
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   19.153678 500.065168   0.038   0.9694
## Pclass2       -1.100031   0.350589  -3.138   0.0017 **
## Pclass3       -2.145112   0.344665  -6.224 4.85e-10 ***
## Sex          -14.758669 500.064477  -0.030   0.9765
## Fsize         -0.539416   0.108508  -4.971 6.65e-07 ***
## Child          0.488847   0.432319   1.131   0.2582
## Fare           0.003918   0.003286   1.192   0.2332
## EmbarkedQ     -0.212344   0.478981  -0.443   0.6575
## EmbarkedS     -0.358904   0.292326  -1.228   0.2195
## TitleMiss    -15.530429 500.064856  -0.031   0.9752
## TitleMr       -3.969396   0.689670  -5.755 8.64e-09 ***
## TitleMrs     -15.032857 500.064975  -0.030   0.9760
## TitleOfficer  -4.292622   0.923427  -4.649 3.34e-06 ***
## TitleRoyalty  -3.517950   1.590117  -2.212   0.0269 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 939.87  on 712  degrees of freedom
## Residual deviance: 568.60  on 699  degrees of freedom
## AIC: 596.6
##
## Number of Fisher Scoring iterations: 13
```

```
# validation
predglm <- predict(model_glm, validation , type ="response" )
logit_survived = as.numeric(predglm >= 0.5)
table(logit_survived)
```

```
## logit_survived
##   0   1
## 105  73
```

```
confusionMatrix(as.factor(validation$Survived) ,as.factor(logit_survived))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 86 14
##          1 19 59
##
##                Accuracy : 0.8146
##                  95% CI : (0.7496, 0.8688)
##     No Information Rate : 0.5899
##     P-Value [Acc > NIR] : 1.274e-10
##
##                   Kappa : 0.6208
##  Mcnemar's Test P-Value : 0.4862
##
##             Sensitivity : 0.8190
##             Specificity : 0.8082
##          Pos Pred Value : 0.8600
##          Neg Pred Value : 0.7564
##              Prevalence : 0.5899
##          Detection Rate : 0.4831
##    Detection Prevalence : 0.5618
##       Balanced Accuracy : 0.8136
##
##        'Positive' Class : 0
##
```

```
# predicting Test
test_glm <- predict(model_glm, test_titanic , type ="response" )
print(RMSE(validation$Survived,logit_survived))
```

```
## [1] 0.4305732
```

Random Forest

```
set.seed(123)
cvCtrl = trainControl(method = "repeatedcv", number = 5, repeats = 5)
mtry <- round(sqrt(ncol(train) -1))

RFgrid <- expand.grid(
 mtry = mtry)

rf_model <-train(Survived ~ Pclass+Sex+Fsize+Child+Fare+Embarked+Title, data=train,
                 tuneGrid = RFgrid,
                 method = "rf" ,
                 trControl = cvCtrl,
                 preProcess = c("center", "scale"))

rf_pred <- predict(rf_model , validation )
rf_pred = as.numeric(rf_pred >= 0.5)
confusionMatrix(as.factor(validation$Survived) ,as.factor(rf_pred))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 91  9
##          1 24 54
##
##                Accuracy : 0.8146
##                  95% CI : (0.7496, 0.8688)
##     No Information Rate : 0.6461
##     P-Value [Acc > NIR] : 6.012e-07
##
##                   Kappa : 0.6153
##  Mcnemar's Test P-Value : 0.01481
##
##             Sensitivity : 0.7913
##             Specificity : 0.8571
##          Pos Pred Value : 0.9100
##          Neg Pred Value : 0.6923
##              Prevalence : 0.6461
##          Detection Rate : 0.5112
##    Detection Prevalence : 0.5618
##       Balanced Accuracy : 0.8242
##
##        'Positive' Class : 0
##
```

```
print(RMSE(validation$Survived,rf_pred))
```

```
## [1] 0.4305732
```

SVM

```
## SVM
# Set up the 5-fold CV
fitControl <- caret::trainControl(method = "repeatedcv",
                                   number = 5,
                                   repeats = 5)

# Define ranges for the two parameters
C_range =     sapply(seq(-1,3,0.0125), function(x) 10^x)
sigma_range = sapply(seq(-3,1,0.0125), function(x) 10^x)

# Create the grid of parameters
fitGrid <- expand.grid(C= C_range,
                       sigma = sigma_range)


Rsvm <- caret::train(Survived ~ Pclass+Sex+Fsize+Child+Fare+Embarked+Title, data=train,
                     method = "svmLinear",
                     trControl = fitControl,
                     preProcess = c("center", "scale"))
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to
## do classification? If so, use a 2 level factor as your outcome column.
```

```
svm_pred <- predict(Rsvm , validation )
svm_pred = as.numeric(svm_pred >= 0.5)
confusionMatrix(as.factor(validation$Survived) ,as.factor(svm_pred))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 80 20
##          1 17 61
##
##                Accuracy : 0.7921
##                  95% CI : (0.7251, 0.8492)
##     No Information Rate : 0.5449
##     P-Value [Acc > NIR] : 5.144e-12
##
##                   Kappa : 0.5796
##  Mcnemar's Test P-Value : 0.7423
##
##             Sensitivity : 0.8247
##             Specificity : 0.7531
##          Pos Pred Value : 0.8000
##          Neg Pred Value : 0.7821
##              Prevalence : 0.5449
##          Detection Rate : 0.4494
##    Detection Prevalence : 0.5618
##       Balanced Accuracy : 0.7889
##
##        'Positive' Class : 0
##
```

```
print(RMSE(validation$Survived,svm_pred))
```

```
## [1] 0.4559223
```

Gradient Boosting

```
#gradient boosting
fitControl <- trainControl(method = 'repeatedcv',
                           number = 5,
                           repeats = 5)
# for caret, there are only four tuning parameters below.

# tune n.trees
newGrid <- expand.grid(n.trees = c(50, 100, 200, 300),
                       interaction.depth = c(6),
                       shrinkage = 0.01,
                       n.minobsinnode = 10
)
fit_gbm <- train(Survived ~ Pclass+Sex+Fsize+Child+Fare+Embarked+Title, data=train,
                 method = 'gbm',
                 trControl = fitControl,
                 tuneGrid =  newGrid,
                 bag.fraction = 0.5,
                 verbose = FALSE,
                 preProcess = c("center", "scale"))
fit_gbm$bestTune
```

```
##   n.trees interaction.depth shrinkage n.minobsinnode
## 4     300                 6      0.01             10
```

```
gbm_pred <- predict(fit_gbm , validation )
gbm_pred = as.numeric(gbm_pred >= 0.5)
confusionMatrix(as.factor(validation$Survived) ,as.factor(gbm_pred))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 95  5
##          1 25 53
##
##                Accuracy : 0.8315
##                  95% CI : (0.7682, 0.8833)
##     No Information Rate : 0.6742
##     P-Value [Acc > NIR] : 1.724e-06
##
##                   Kappa : 0.6478
##  Mcnemar's Test P-Value : 0.0005226
##
##             Sensitivity : 0.7917
##             Specificity : 0.9138
##          Pos Pred Value : 0.9500
##          Neg Pred Value : 0.6795
##              Prevalence : 0.6742
##          Detection Rate : 0.5337
##    Detection Prevalence : 0.5618
##       Balanced Accuracy : 0.8527
##
##        'Positive' Class : 0
##
```

```
print(RMSE(validation$Survived,gbm_pred))
```

```
## [1] 0.4105354
```