Titanic Research

Dambaev B

Содержание

1 Загрузка и обзор данных

Загружаем данные. А также объединим тренировочный и проверочный датасеты в общий для дальнейшей подготовки данных к лучшему виду.

```
train <- read.csv('datasets/train.csv', na.strings = c("NA", ""), stringsAsFactors = F)
test <- read.csv('datasets/test.csv', na.strings = c("NA", ""), stringsAsFactors = F)
test$Survived <- NA
full <- rbind(train, test)
```

Теперь посмотрим на структуру наших загруженных данных.

```
str(full)
```

```
## 'data.frame':    1309 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)" "Heikkinen, Miss. L
##  $ 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  NA "C85" NA "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

Всего 12 переменных 1309 объектов. Ниже представлено подробное описание каждой переменной.

Переменная	Описание
Survived	Выжил (1) или Погиб (0)
Pclass	Класс пассажира
Name	Имя пассажира
Sex	Пол пассажира
Age	Возраст пассажира
SibSp	Количество братьев и сестер/супругов на борту
Parch	Количество родителей/детей на борту
Ticket	Номер билета
Fare	Плата за проезд
Cabin	Номер каюты
Embarked	Порт отправления

2 Обработка отсутствующих значений

Прежде всего необходимо решить вопрос с отсутствующими значениями. Есть несколько вариантов:

- игнорировать признаки, имеющие недостающие значения;
- игнорировать наблюдения с отсутствующими значениями;
- заполнить недостающие значения (средним значением, медианой или модой);
- предсказать недостающие значения на основе других известных признаков;

Но сначала узнаем у каких признаков есть отсутствующие значения и сколько в каждом из них отсутствующих значений.

```
kable(miss_var_summary(full, order = F))
```

variable	n_miss	pct_miss
PassengerId	0	0.0000000
Survived	418	31.9327731
Pclass	0	0.0000000
Name	0	0.0000000
Sex	0	0.0000000
Age	263	20.0916730
SibSp	0	0.0000000

variable	n_miss	pct_miss
Parch	0	0.0000000
Ticket	0	0.0000000
Fare	1	0.0763942
Cabin	1014	77.4637128
Embarked	2	0.1527884

Учитывая небольшой размер набора данных, нецелесообразно избавляться ни от целых признаков, ни от наблюдений, имеющих отсутствующие значения. Поэтому остается заполнить значения либо одним и тем же значением (средним, медианой или модой), либо попробовать предсказать их.

2.1 Возраст - Age

Ввиду достаточно большого количества пропущенных значений (263 пассажира, что составляет около 20% от общего числа пассажиров), следует спрогнозировать недостающие значения на основе других признаков. Если внимательно изучить строковую переменную Name, то из нее можно выделить такой признак, как титул, который тесно коррелирует с нашим искомым возрастом.

```
head(full$Name)
```

```
## [1] "Braund, Mr. Owen Harris"
## [2] "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## [3] "Heikkinen, Miss. Laina"
## [4] "Futrelle, Mrs. Jacques Heath (Lily May Peel)"
## [5] "Allen, Mr. William Henry"
## [6] "Moran, Mr. James"
```

Выделим из Name новый признак Title. Далее найдем титулы, у которых пропущено хотя бы одно значение. И заменим недостающие значения на их средний возраст.

```
full$Title <- sapply(full$Name, function(x) {strsplit(x, split='[,.]')[[1]][2]})
full$Title <- trimws(full$Title, which = c("left"))
```

```
kable(full %>% group_by(Title) %>%
      summarise(Count = n(), MissingAge = sum(is.na(Age)), Mean = round(mean(Age, na.rm = T), 2)))
```

Title	Count	MissingAge	Mean
Capt	1	0	70.00
Col	4	0	54.00
Don	1	0	40.00
Dona	1	0	39.00
Dr	8	1	43.57
Jonkheer	1	0	38.00
Lady	1	0	48.00
Major	2	0	48.50
Master	61	8	5.48
Miss	260	50	21.77
Mlle	2	0	24.00
Mme	1	0	24.00
Mr	757	176	32.25
Mrs	197	27	36.99
Ms	2	1	28.00
Rev	8	0	41.25

Title	Count	MissingAge	Mean
Sir	1	0	49.00
the Countess	1	0	33.00

```
for(title in c("Dr","Master","Miss","Mr","Mrs","Ms")){
  full$Age[is.na(full$Age) & full$Title == title] <-
    mean(full$Age[full$Title == title], na.rm = T)
}
# Проверим количество отсутствующих значений
n_miss(full$Age)
```

```
## [1] 0
```

Новосозданный признак Title также стоит сгруппировать и профакторизовать.

```
full$Title[full$Title %in% c("Ms")] <- "Mrs"
full$Title[full$Title %in% c("Mlle", "Mme")] <- "Miss"
full$Title[!(full$Title %in% c("Master", "Miss", "Mr", "Mrs"))] <- "Rare Title"
full$Title <- as.factor(full$Title)
kable(table(full$Sex, full$Title))
```

	Master	Miss	Mr	Mrs	Rare Title
female	0	263	0	199	4
male	61	0	757	0	25

2.2 Плата за проезд- Fare

Fare имеет одно пропущенное значение. Заполним его медианой.

```
full$Fare[is.na(full$Fare)] = median(full$Fare, na.rm = T)
# Проверим количество отсутствующих значений
n_miss(full$Fare)
```

```
## [1] 0
```

2.3 Номер каюты - Cabin

Cabin имеет 1014 недостающих значений. В процентном соотношении пропущена очень огромная часть - 77.46% . Поэтому разумно не использовать этот признак во избежание добавления шума.

2.4 Порт отправления - Embarked

В переменной Embark пропущено два значения. Заполним их модой.

```
#Так как в R нет функции вычисления моды, создадим ее сами
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}

full$Embarked[is.na(full$Embarked)] = getmode(full$Embarked)
# Проверим количество отсутствующих значений
n_miss(full$Embarked)
```

```
## [1] 0
```

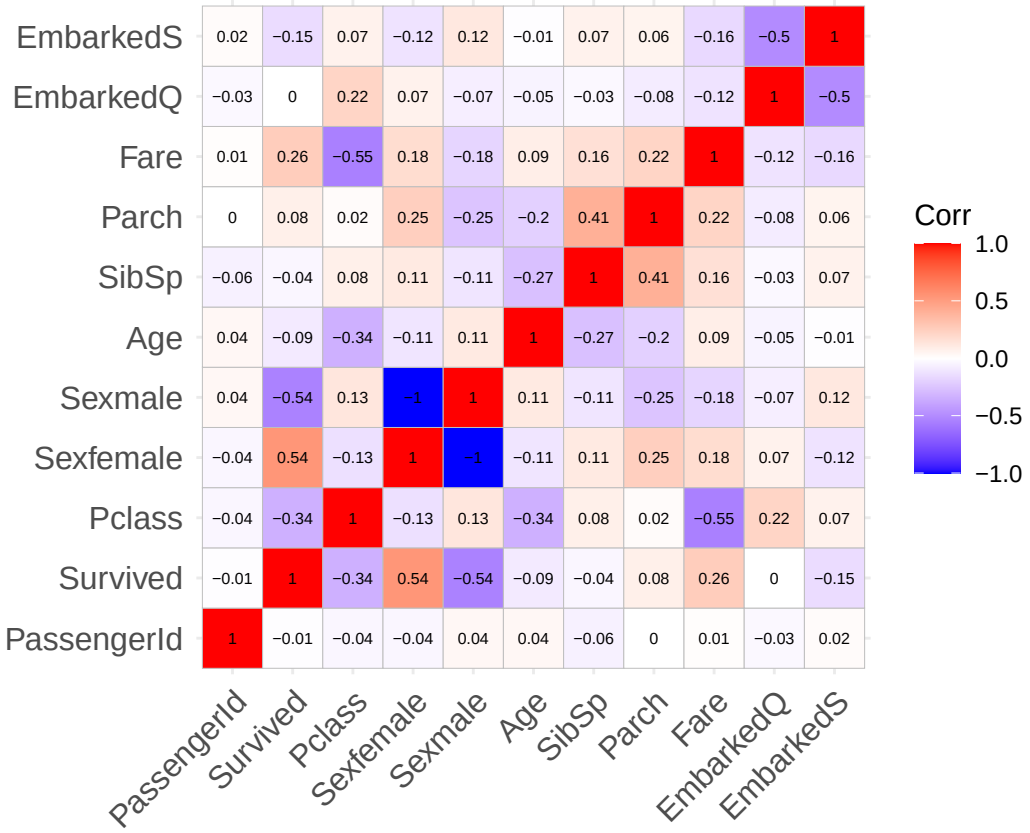
Также факторизируем этот категориальный признак.

```
full$Embarked = factor(full$Embarked)
```

3 Корреляционная матрица

Для начала попробуем найти главные признаки, влияющие на искомую вероятность выживаемости. Для этого составим корреляционную матрицу и посмотрим на зависимости между имеющимися переменными.

```
full_dropped = full[, !(names(full) %in% c("Name","Cabin","Ticket", "Title"))]
model.matrix(~0+., data=full_dropped) %>%
  cor(use="complete.obs") %>%
  ggcorrplot(show.diag = F, type = "full", lab=TRUE, lab_size=2)
```



Из матрицы видно, что непосредственное влияние имеют такие признаки, как Pclass, Sex и немного Fare (хотя Fare сильно связан с классом пассажира).

4 Исследование наиболее важных признаков

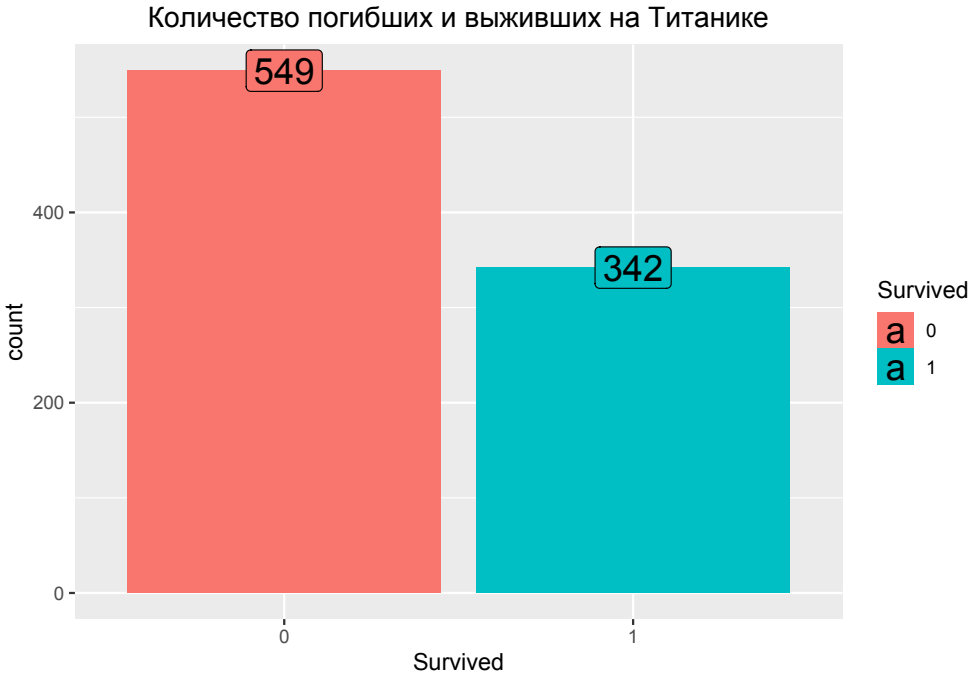
Для нахождения зависимости или независимости между некоторыми признаками и показателем выживаемости построим соответствующие графики.

Но прежде всего, факторизируем категориальные полноценные признаки в тренировочном наборе для удобства последующей визуализации.

```
cat_vars = c("Survived", "Pclass", "Sex");
full[cat_vars] = lapply(full[cat_vars], function(x) as.factor(x));
```

4.1 Искомая переменная - Survived

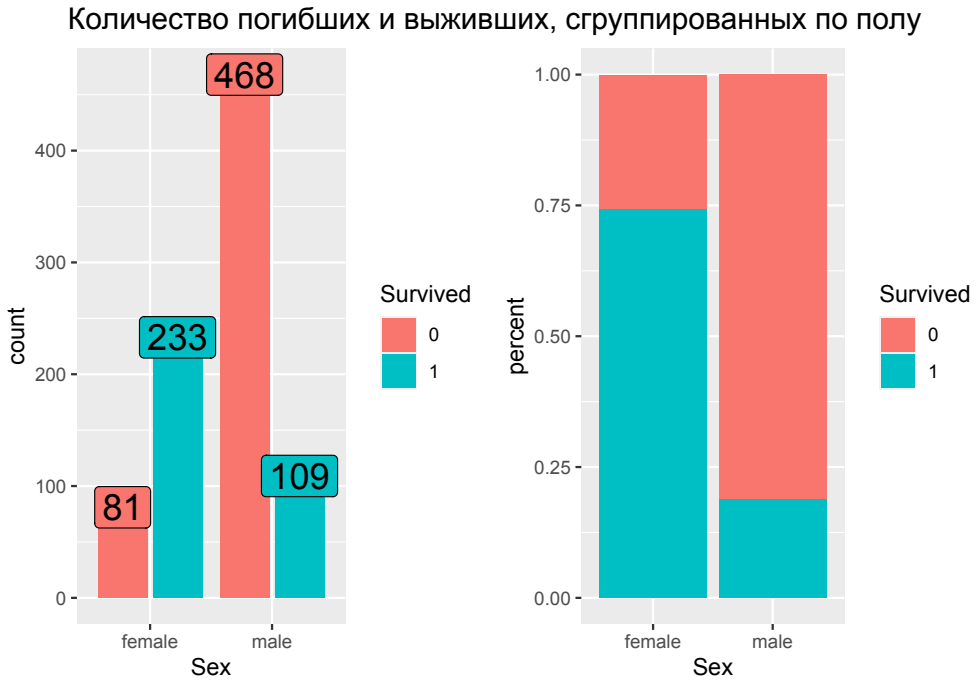
```
ggplot(full[!is.na(full$Survived),], aes(Survived, fill = Survived)) +
  geom_bar(stat = "count") +
  geom_label(stat='count', aes(label=..count..), size = 6) +
  labs(title = "Количество погибших и выживших на Титанике", x = "Survived") +
  theme(plot.title = element_text(hjust = 0.5));
```



Из 891 пассажира 342 выжило и 549 погибло, т.е. 38.4% и 61.6% соответственно. Исходя из этого соотношения, можно сказать в дальнейшем будет легче предсказать гибель чем его спасение.

4.2 Пол - Sex

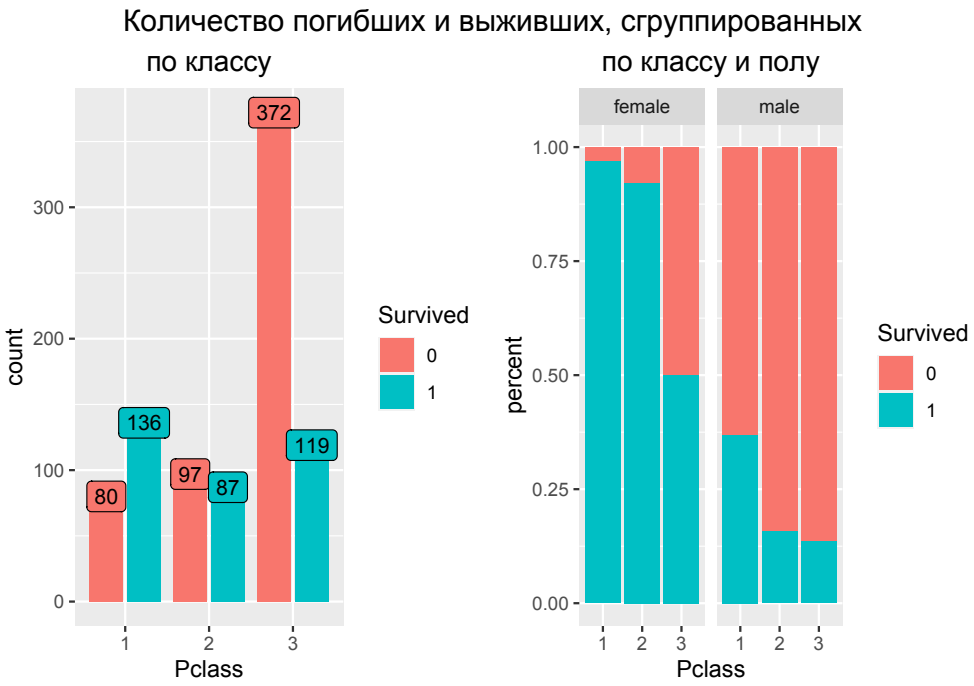
```
plot1 = ggplot(full[!is.na(full$Survived),], aes(Sex, fill = Survived)) +
  geom_bar(stat = "count", position = position_dodge2()) +
  geom_label(stat='count', aes(label=..count..), size = 6, position = position_dodge(0.9), show.legend = F) +
  labs(x = "Sex") +
  theme(plot.title = element_text(hjust = 0.5));
plot2 = ggplot(full[!is.na(full$Survived),], aes(Sex, fill = Survived)) +
  geom_bar(stat = "count", position = "fill") +
  labs(x = "Sex", y = "percent") +
  theme(plot.title = element_text(hjust = 0.5));
grid.arrange(plot1, plot2, ncol = 2, top = textGrob("Количество погибших и выживших, сгруппированных по полу",gp=
```



В рамках данных обучения выжили 74.2% женщин и 18.9% мужчин. Благодаря этой существенной разнице половой признак можно считать одним из важнейших факторов будущего предсказания.

4.3 Класс пассажира - Pclass

```
plot1 = ggplot(full[!is.na(full$Survived),], aes(Pclass, fill = Survived)) +
  geom_bar(stat = "count", position = position_dodge2()) +
  geom_label(stat='count', aes(label=..count..), size = 3.5, position = position_dodge(0.9), show.legend = F) +
  labs(title = "по классу", x = "Pclass") +
  theme(plot.title = element_text(hjust = 0.5));
plot2 = ggplot(full[!is.na(full$Survived),], aes(Pclass, fill = Survived)) +
  geom_bar(stat = "count", position = "fill") +
  labs(title = "по классу и полу", x = "Pclass", y = "percent") +
  facet_grid(.~Sex) +
  theme(plot.title = element_text(hjust = 0.5));
grid.arrange(plot1, plot2, ncol = 2,
             top = textGrob("Количество погибших и выживших, сгруппированных",gp=gpar(fontsize=14,font=1)))
```



По первому графику видно, что большинство людей было из 3-го класса. Более того, как и ожидалось, большинство смертей приходится на 3-ий класс.

Также для полноты понимания добавим группирование по половому признаку.

Теперь можно заметить, что:

- женщины из 1-го и 2-го классов почти гарантированно выживают
- мужчины как из 2-го, так и из 3-го класса имеют очень плохие шансы на выживание
- у женщин из 3-го класса шансы на спасение и гибель равны
- доля выживших мужчин из 1-го класса также мала относительно женщин, но гораздо больше остальных мужчин

5 Исследование признаков и выделение новых

Попробуем извлечь полезную информацию из разных признаков.

5.1 Номер билета - Ticket

Переменная Ticket имеет огромный диапазон разнообразных значений, которые невозможно как-либо сгруппировать. Использование такой переменной несомненно не принесет какой-либо пользы.

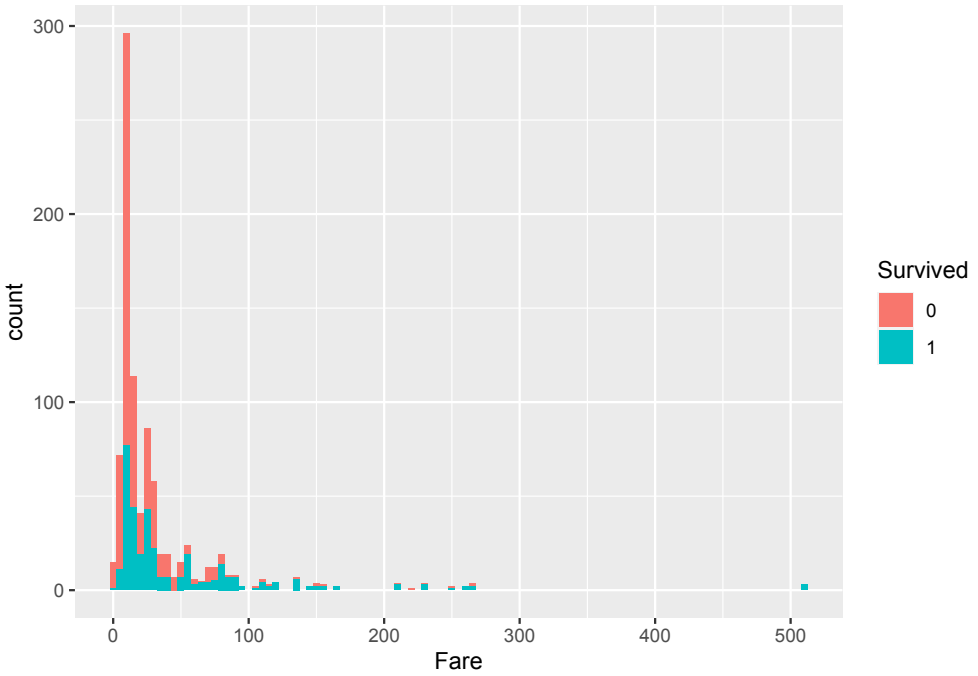
```
length(unique(full$Ticket)) / length(full$Ticket)
```

```
## [1] 0.7097021
```

5.2 Стоимость билета - Fare

Посмотрим на график зависимости Survived от переменной Fare, построив соответствующие плотности.

```
ggplot(full[!is.na(full$Survived),], aes(Fare)) +
  geom_histogram(aes(fill = Survived), binwidth = 5)
```

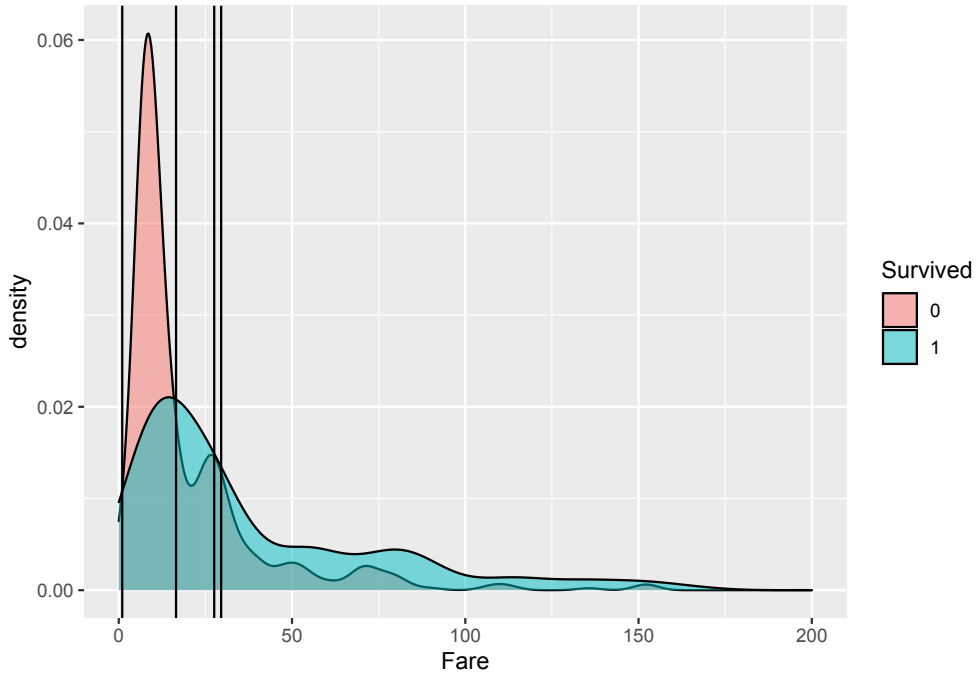


Разобъем переменную Fare на выделяющиеся подгруппы. Для этого найдем точки пересечения плотностей.

```
lower.limit <- min(full$Fare)
upper.limit <- max(full$Fare)
fare_survived_dens <- density(subset(full[!is.na(full$Survived),], Survived == "1")$Fare, from = lower.limit, to = upper.li
fare_dead_dens <- density(subset(full[!is.na(full$Survived),], Survived == "0")$Fare, from = lower.limit, to = upper.limit,

density.difference <- fare_survived_dens$y - fare_dead_dens$y
points <- fare_survived_dens$x[which(diff(density.difference > 0) != 0) + 1]
```

Снова посмотрим на график, но предварительно его сузим и добавим точки пересечения.

```
ggplot(full[!is.na(full$Survived),], aes(Fare)) +
  geom_density(alpha = 0.5, aes(fill = Survived)) +
  geom_vline(xintercept = points) +
  xlim(0, 200)
```

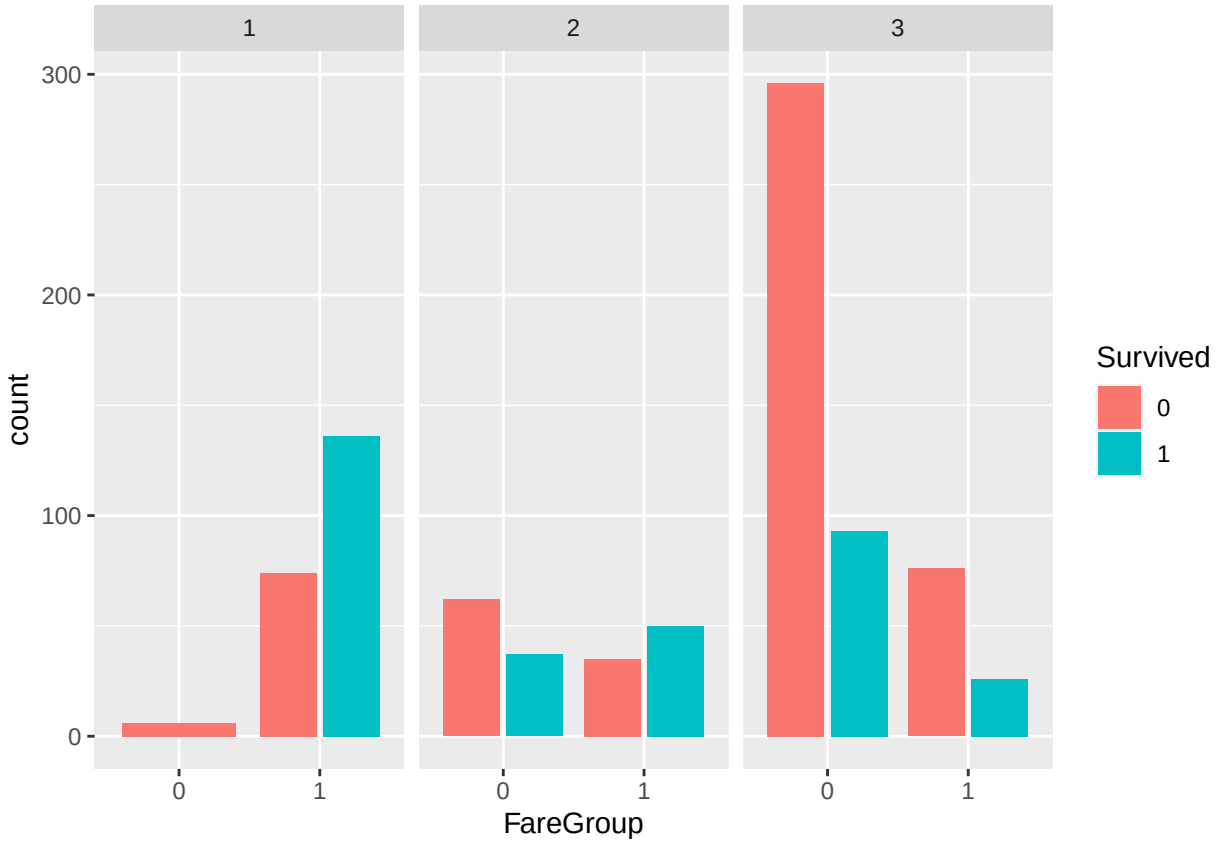


Разобъем Fare на две подгруппы. Выберем в качестве разделителя подгрупп - 2-ую точку слева.

```
full$FareGroup[full$Fare <= points[2]] <- "0"
full$FareGroup[full$Fare > points[2]] <- "1"
full$FareGroup = factor(full$FareGroup)
```

Далее построим график зависимость Survived по переменным FareGroup и Pclass.

```
ggplot(full[!is.na(full$Survived),], aes(FareGroup, fill = Survived)) +
  geom_bar(stat = "count", position = position_dodge2()) +
  facet_grid(.~ Pclass)
```



Исходя из графика, можно сказать, что выделенный нами признак FareGroup вносит дополнительную информацию в разделении признака Pclass (1 и 2 классы), но при этом также имеет с ним некоторую зависимость (3 класс).

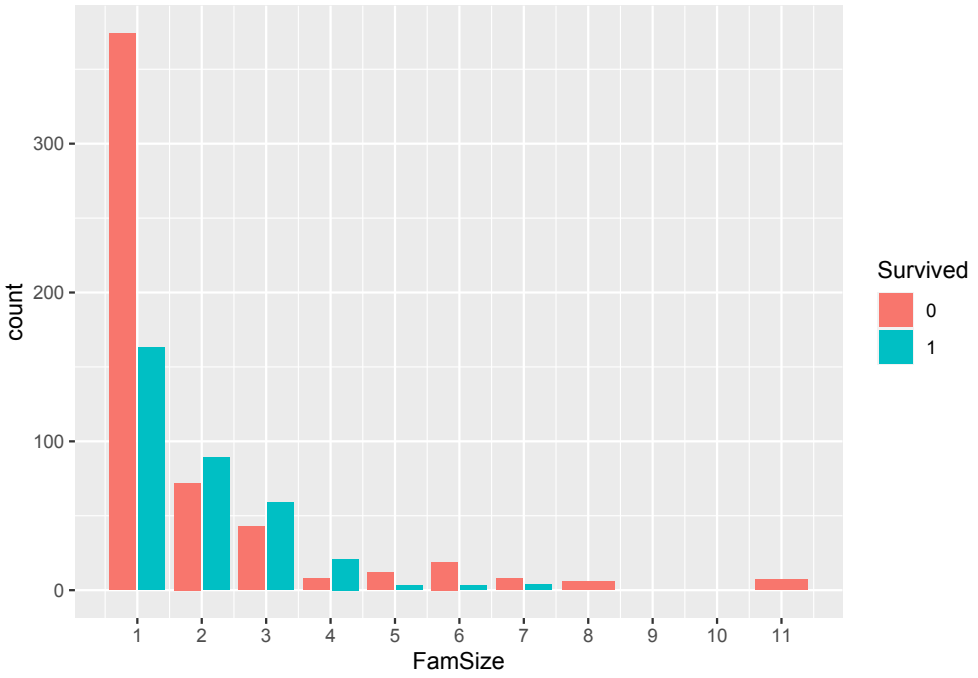
5.3 Размер семьи - FamSize

Создадим новый признак FamilySize, соединив два изначальных SibSp и Parch, обозначающих количество братьев и сестер/супругов на борту и количество родителей/детей на борту соотвественно.

```
full$FamSize <- full$SibSp + full$Parch + 1
```

Далее построим график влияния нашей новой переменной на показатель выживаемости.

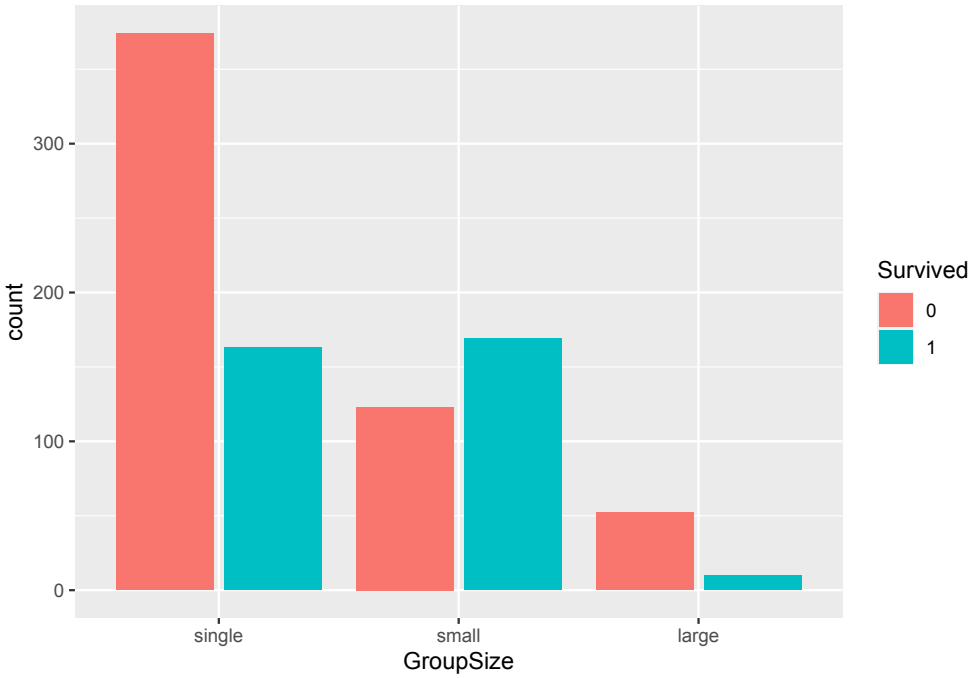
```
ggplot(full[!is.na(full$Survived),], aes(FamSize, fill = Survived)) +
  geom_bar(stat = "count", position = position_dodge2()) +
  scale_x_continuous(breaks = c(1:max(full$FamSize))) +
  labs(x = "FamSize") +
  theme(plot.title = element_text(hjust = 0.5));
```



По вышерасположенному графику можно заметить, что у людей, которые путешествовали в одиночку, шансы погибнуть были гораздо выше чем выжить. Также невезло и большим семьям. А люди, путешествовавшие небольшой семьей, имели относительно высокую вероятность спастись.

В силу относительно малого количества больших семей, будет логично разбить переменную на три разные группы.

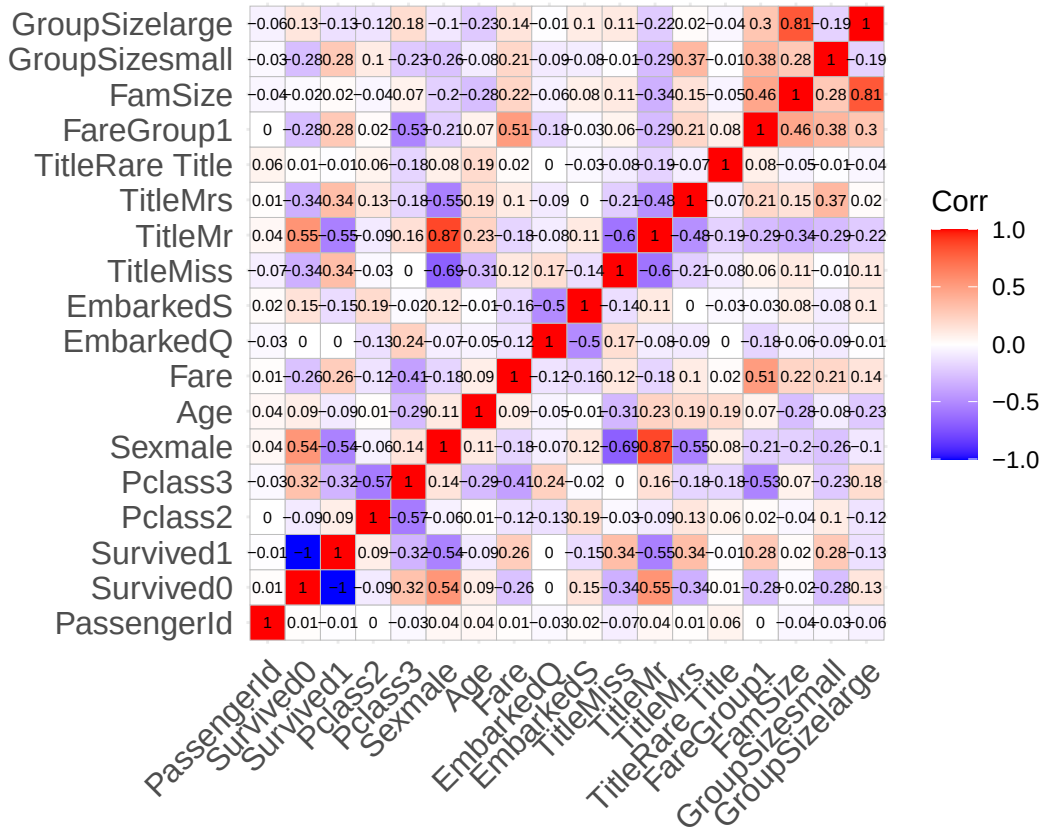
```
full$GroupSize[full$FamSize == 1] <- "single"
full$GroupSize[full$FamSize > 1 & full$FamSize <= 4] <- "small"
full$GroupSize[full$FamSize > 4] <- "large"
full$GroupSize = factor(full$GroupSize, levels = c("single", "small", "large"))

ggplot(full[!is.na(full$Survived),], aes(GroupSize, fill = Survived)) +
  geom_bar(stat = "count", position = position_dodge2()) +
  labs(x = "GroupSize") +
  theme(plot.title = element_text(hjust = 0.5));
```



6 Корреляционная матрица (обновленная)

Снова посмотрим на корреляционную матрицу, но уже с новыми переменными.

```
full_dropped = full[, !(names(full) %in% c("Name","Cabin","Ticket", "SibSp", "Parch"))]
model.matrix(~0+., data=full_dropped) %>%
  cor(use="complete.obs") %>%
  ggcorrplot(show.diag = F, type = "full", lab=TRUE, lab_size=2)
```



Из представленной выше матрицы получаем, что новый признак GroupSize оказывает примерно такое же влияние на выживаемость, что и признак Fare, но при этом первый в два раза меньше зависит от Pclass чем второй.

Также стоит отказаться от признака Title, так как он слишком сильно коррелирует с переменными Age и Sex, которые мы уже используем, что безусловно приведет к переобучению модели.

7 Прогнозирование

7.1 Разделение данных на обучающий и тестовый наборы

Для будущей оценки модели разобъем наши данные на обучающий и тестовый наборы в соотношении 8 к 2.

```
set.seed(2021)
train <- full[!is.na(full$Survived),]
split <- createDataPartition(train$Survived, p = 0.8, list = FALSE)
train_clean <- slice(train, split)
test_clean <- slice(train, -split)
```

7.2 Построение моделей

В качестве основных предикторов возьмем такие переменные, как Pclass, Sex, Age и GroupSize.

В дальнейшем для решения нашей задачи попробуем три разные модели - метод случайного леса, метод опорных векторов и метод k ближайших соседей.

Для получения более точных оценок используем пятикратную перекрестную проверку с разбиением выборки на 7 частей.

```
set.seed(2021)
trCtrl <- trainControl(method="repeatedcv", number=7, repeats = 5)
```

7.2.1 модель Random Forest

```
set.seed(2021)
rf.grid <- data.frame(.mtry = c(2:4))

rf_model.1 <- train(Survived ~ Pclass+Sex+Age+GroupSize,
                    data=train_clean,
                    method='rf',
                    tuneGrid = rf.grid,
                    trControl=trCtrl)

kable(rf_model.1$results)
```

mtry	Accuracy	Kappa	AccuracySD	KappaSD
2	0.8307805	0.6309150	0.0405537	0.0912862
3	0.8288114	0.6268893	0.0403106	0.0907257
4	0.8265785	0.6249744	0.0370430	0.0823853

```
set.seed(2021)
rf_model.2 <- train(Survived ~ Pclass+Sex+Age+GroupSize+Fare+Embarked,
                    data=train_clean,
                    method='rf',
                    tuneGrid = rf.grid,
                    trControl=trCtrl)

kable(rf_model.2$results)
```

mtry	Accuracy	Kappa	AccuracySD	KappaSD
2	0.8332849	0.6305789	0.0379093	0.0879545
3	0.8388737	0.6480765	0.0353362	0.0789968
4	0.8355259	0.6439369	0.0366335	0.0812133

7.2.2 модель Support Vector Machine (SVM)

```
set.seed(2021)
svm_model.1 <- train(Survived ~ Pclass+Sex+Age+GroupSize,
                    data=train_clean, method='svmRadial',
                    tuneLength = 9,
                    preProcess = c("center", "scale"),
                    trControl=trCtrl)

kable(svm_model.1$results)
```

sigma	C	Accuracy	Kappa	AccuracySD	KappaSD
0.1471344	0.25	0.8243294	0.6192902	0.0353922	0.0782596
0.1471344	0.50	0.8307667	0.6303436	0.0416661	0.0929389
0.1471344	1.00	0.8321756	0.6319302	0.0393653	0.0883978
0.1471344	2.00	0.8355398	0.6389600	0.0405728	0.0917869
0.1471344	4.00	0.8338646	0.6348999	0.0424498	0.0956950
0.1471344	8.00	0.8324612	0.6315456	0.0417489	0.0941456
0.1471344	16.00	0.8307833	0.6280578	0.0425379	0.0959177
0.1471344	32.00	0.8296601	0.6255594	0.0429893	0.0970532
0.1471344	64.00	0.8282594	0.6225730	0.0414776	0.0937021

```
set.seed(2021)
svm_model.2 <- train(Survived ~ Pclass+Sex+Age+GroupSize+Fare+Embarked,
                     data=train_clean, method='svmRadial',
                     tuneLength = 9,
                     preProcess = c("center", "scale"),
                     trControl=trCtrl)

kable(svm_model.2$results)
```

sigma	C	Accuracy	Kappa	AccuracySD	KappaSD
0.1166408	0.25	0.8192653	0.6060606	0.0388261	0.0878932
0.1166408	0.50	0.8287949	0.6239966	0.0392420	0.0887987
0.1166408	1.00	0.8315850	0.6291348	0.0401193	0.0916481
0.1166408	2.00	0.8287811	0.6221141	0.0398880	0.0917447
0.1166408	4.00	0.8296241	0.6242148	0.0387254	0.0887044
0.1166408	8.00	0.8265290	0.6180455	0.0414248	0.0937391
0.1166408	16.00	0.8206684	0.6039713	0.0415517	0.0943872
0.1166408	32.00	0.8145112	0.5922043	0.0421320	0.0943251
0.1166408	64.00	0.8061295	0.5747233	0.0425681	0.0947615

7.2.3 модель k ближайших соседей (KNN)

```
set.seed(2021)
knn_model.1 <- train(Survived ~ Pclass+Sex+Age+GroupSize,
                     data=train_clean, method='knn',
                     trControl=trCtrl)

kable(knn_model.1$results)
```

k	Accuracy	Kappa	AccuracySD	KappaSD
5	0.7554117	0.4636945	0.0342209	0.0767826
7	0.7607173	0.4767982	0.0385017	0.0833237
9	0.7542910	0.4597192	0.0355642	0.0770317

```
set.seed(2021)
knn_model.2 <- train(Survived ~ Pclass+GroupSize+Title,
                     data=train_clean, method='knn',
                     trControl=trCtrl)
```

```
kable(knn_model.2$results)
```

k	Accuracy	Kappa	AccuracySD	KappaSD
5	0.8251726	0.6186149	0.0365926	0.0822373
7	0.8167608	0.5986232	0.0354602	0.0812834
9	0.8125590	0.5886667	0.0349097	0.0799286

7.3 Оценка моделей

Для всех моделей проведём оценку при помощи пересечения предсказанных на тестовой выборке и реальных значений целевого признака.

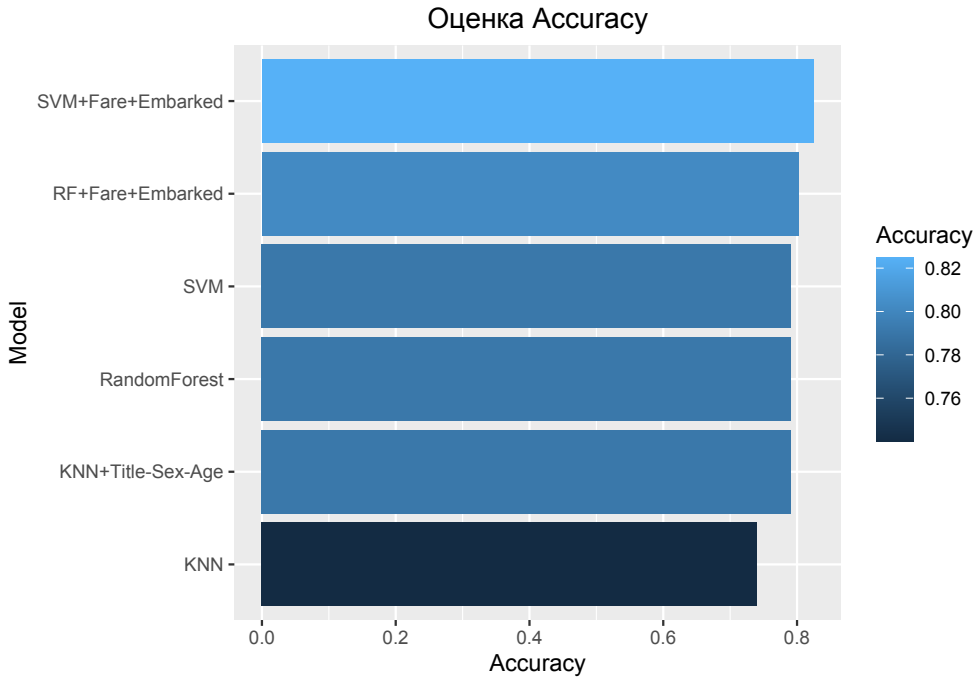
```
metrics <- data.frame(Model = numeric(), Accuracy = numeric(),
                      Sensitivity = numeric(), Specificity = numeric())
models <- list(rf_model.1, rf_model.2, svm_model.1, svm_model.2, knn_model.1, knn_model.2)
for (model in models) {
  prediction <- predict(model, test_clean)
  results <- confusionMatrix(prediction, test_clean$Survived)
  metrics[nrow(metrics)+1,] <-
    c(NA,
    results$overall[1],
    results$byClass["Sensitivity"],
    results$byClass["Specificity"])
}
metrics$Model <- c("RandomForest",
                   "RF+Fare+Embarked",
                   "SVM",
                   "SVM+Fare+Embarked",
                   "KNN",
                   "KNN+Title-Sex-Age")
kable(metrics)
```

Model	Accuracy	Sensitivity	Specificity
RandomForest	0.7909605	0.9174312	0.5882353
RF+Fare+Embarked	0.8022599	0.9541284	0.5588235
SVM	0.7909605	0.9174312	0.5882353
SVM+Fare+Embarked	0.8248588	0.9724771	0.5882353
KNN	0.7401130	0.8532110	0.5588235
KNN+Title-Sex-Age	0.7909605	0.9174312	0.5882353

В качестве главного параметра оценивания будем использовать точность измерений (Accuracy), а именно процент правильно предсказанных пассажиров.

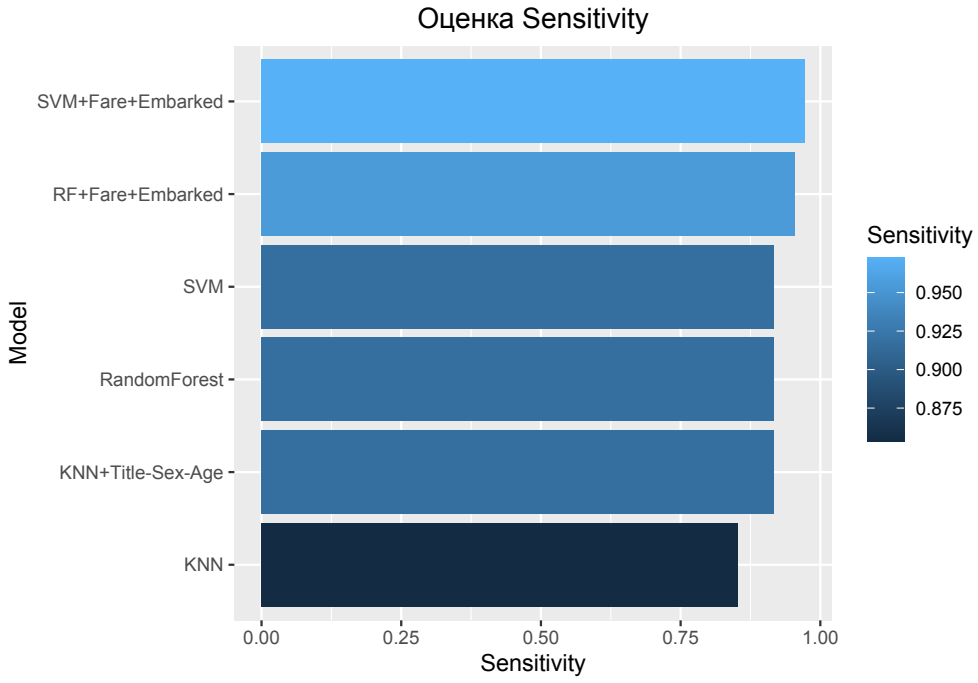
Также будем учитывать параметры чувствительность (Sensitivity) и специфичность (Specificity), означающих истинно положительную пропорцию и истинно отрицательную пропорцию соотвественно.

```
ggplot(metrics, aes(x = reorder(Model, Accuracy),
    y = Accuracy, fill = Accuracy)) +
  geom_bar(stat='identity') +
  coord_flip() +
  labs(title = 'Оценка Accuracy', x = 'Model') +
  theme(plot.title = element_text(hjust = 0.5))
```

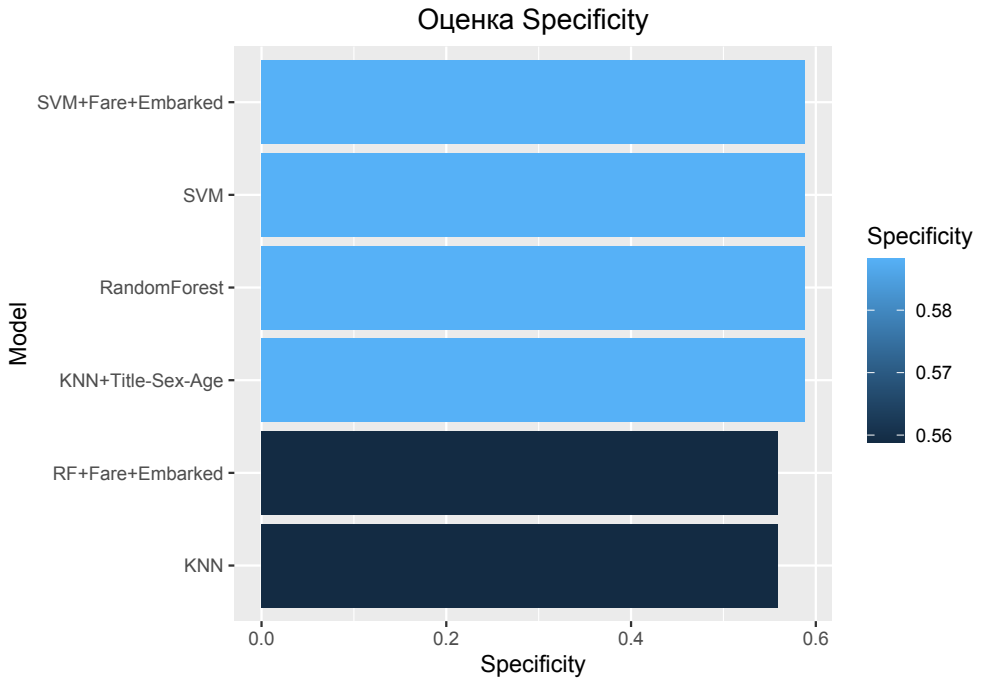


Модель SVM, с дополнительно учтенными признаками Fare и Embarked, показывает лучший результат - 0.8248% точности.

```
ggplot(metrics, aes(x = reorder(Model, Sensitivity),
    y = Sensitivity, fill = Sensitivity)) +
  geom_bar(stat='identity') +
  coord_flip() +
  labs(title = 'Оценка Sensitivity', x = 'Model') +
  theme(plot.title = element_text(hjust = 0.5))
```



```
ggplot(metrics, aes(x = reorder(Model, Specificity),
    y = Specificity, fill = Specificity)) +
  geom_bar(stat='identity') +
  coord_flip() +
  labs(title = 'Оценка Specificity', x = 'Model') +
  theme(plot.title = element_text(hjust = 0.5))
```



По предсказанию погибших (Sensitivity) также лидирует SVM+Fare+Embarked, имея небольшой отрыв от Random Forest с теми же признаками. Но уже в предсказании выживших (Specificity) все три метода имеют равный показатель.

7.4 Выгрузка результатов для Kaggle

По итогу оценивания лучшей моделью должна быть SVM+Fare+Embarked.

После загрузки результатов применения моделей на Kaggle наилучшие результаты показали модели SVM и RF с дополнительными Fare и Embarked, получив одинаковую оценку - 0.78229.

```
test <- full[is.na(full$Survived),]
y_predict <- predict(svm_model.2, test)
submission_df <- data.frame(PassengerId = test$PassengerId, Survived = y_predict)
write.csv(submission_df, file = "titanic_predictions.csv", row.names = F)
```