Programmieren mit R: Seminararbeit_1

Daniyar Akhmetov (5127348)
Marcelo Rainho Avila (4679876)
Xuan Son Le (4669361)

Abgabedatum: 14/11/2017

Contents

1 Part I (3 Points)

1.1 Task

- *What are the atomic vector types in R? Explain which value they can take and give an example!*
- *What is the difference between generic and atomic vectors?*
- *Explain the following statement: A data frame is a list, but not every list is a data frame.*

1.2 Answer

1. Atomic vectors are linear vectors (one-dimensional), which consists of values of the same type.

Atomic vectors	**Values**	**Examples**
logical	TRUE/FALSE	c(TRUE,FALSE)
integer	integer numbers	c(1L, 10L, 100L)
numeric	real numbers	c(-26, 1.3, -0.25)
complex	complex numbers a+bi	c(-2+3i, -4i, 0i)
character	strings (letters or words)	c(“Hello”,“2222”,“How are you”)
raw	raw bytes (as pairs of hex digits)	as.raw(255) = ff

2. Generic vectors (lists) are also linear vectors (one-dimensional), but can contain objects of different types. Example of a list:

```r
x <- list(1:10,"a", c(TRUE,FALSE), c(1 + 2i,3i))
str(x)
```

```
## List of 4
##  $ : int [1:10] 1 2 3 4 5 6 7 8 9 10
##  $ : chr "a"
##  $ : logi [1:2] TRUE FALSE
##  $ : cplx [1:2] 1+2i 0+3i
```

3. A data frame is a list of atomic vectors of the same length and therefore has a two dimensional structure based on rows and columns. Example of a data frame:

```r
df <- data.frame(nickName = c(1, 'B','X'),
                 hasGlasses = c(FALSE,"No","Yes"),
                 bankAmount = c(800,1.2,-20))
df # the different data types are coerced into one unique data type.
```

```
##   nickName hasGlasses bankAmount
## 1        1      FALSE      800.0
## 2        B         No        1.2
## 3        X        Yes      -20.0
```

Data frame is a list of atomic vectors with the same length. Each vector represents a column or a row of a data frame. But a list can not be a data frame because lists are linear vectors in one dimensional space. That’s why we have the statement.

2 Part II (7 Points)

2.1 Task

- *Explain each line of the following code. In addition, discuss what the output of identical(a, b) will be. Check the help files for the functions set.seed, identical, rnorm and cumsum.*
- *system.time returns the time elapsed for the computation. Explain the differences:*

```
  # ensure that results from a random generator are reproducible
set.seed(1)
  # generate a vector of normally distributed random numbers
largeVector <- rnorm(100, mean = 5, sd = 10)
  # define a and b
a <- cumsum(largeVector)[1:100]
b <- cumsum(largeVector[1:100])
  # is a equal to b?
identical(a, b)

system.time(cumsum(largeVector)[1:100])
system.time(cumsum(largeVector)[1:100])
```

2.2 Answer

- Because we ran the function set.seed before, the result of randomly chosen largeVector is reproducible for a and b. This means we are now working with the only one largeVector.
- A *cumulative sum (cumsum)* is a sequence of partial sums of a given sequence.
  For instance:
  cumsum(largeVector)
  = cumsum(c(k1,k2,k3,. . . ,kn))
  = k1 k1+k2 k1+k2+k3 . . . k1+k2+k3+..+kn
- In a we firstly have a cumulative sum of largeVector and then take the first 100 elements from it:
  a = cumsum(largeVector)[1:100]
  a = (k1 k1+k2 k1+k2+k3 . . . k1+k2+k3+. . . +kn)[1:100]
  a = k1 k1+k2 k1+k2+k3 . . . k1+k2+k3+. . . +k100
- In b we firstly take the first 100 elements from largeVector and then calculate the cumulative sum of these 100 elements.
  b = cumsum(largeVector[1:100])
  b = cumsum(k1 k2 k3 . . . k100)
  b = k1 k1+k2 k1+k2+k3 . . . k1+k2+k3+. . . +k100
- Now let see why we get different results when applying the function system.time to a and b. On the one hand, *system.time(cumsum(largeVector)[1:100])* firstly try to calculate the cumulative sum of the whole largeVector, which contains 10ˆ8 elements. This would take R a while to finish. On the other hand, *system.time(cumsum(largeVector[1:100]))* firstly get the first 100 elements of largeVector and then calculate the cumulative sum of these 100 elements, which would be much faster to run. That's why we have the difference.

3 Part III (20 Points)

In this section, we aim to construct a model that explains the main factors affecting the probability of survival of the Titanic passengers based on a logistic regression. Alongside the model construction and interpretation, We also aim to document the steps necessary to conduct such an analysis, such as importing, cleaning and validating the data, as explored in the following sections. Let's summarize what we are going to do:

1. **data import**: We import the data set and all necessary libraries
2. **descriptive statistics and data validation**: We try to understand the data set based on some plots and statistics core values.
3. **identification of relevant regressors**: We decide which features to be kept based on their relevance to the target feature and also create some interesting new features based on the given ones.
4. **fitting a regression model**: We create a logit model based on backward selection.
5. **discussion of model fit**: We discuss the significance of the features in the model based on different criteria.
6. **interpreting the model**: We calculate the model accuracy and introduce some different approaches to probably improve the performance of the model.

3.1 Data import

Firstly we need to import all necessary libraries. For this analysis, we used ggplot2, gridExtra, Hmisc and mice libraries, which are not available in the base R packages might have to be installed via the `install.packages()` function.

```r
library(ggplot2)
library(gridExtra)
library(Hmisc)
library(mice)
library(knitr)
```

Firstly we need to import the data. In this case, the file was saved in a "Datasets" sub-folder, relative to the current working directory.

```r
load(file = "/Users/XuanSon/Desktop/titanic.Rdata")
```

3.2 Descriptive statistics and data validation

With the data loaded, we can get an overview of of the data with the `describe()` function from the package *Hmisc*.

```r
describe(titanic)
```

Due to its length, the output is hidden in this report but a few key descriptive statistics can be summarized as following:

- Regarding the given values:
  - So we have 1309 observations with 14 features in total.
  - Only 38,2% of 1309 passengers survived. There are more male (843) than female (466) in this data set
  - The average age of these passengers is about 30.
  - The fare varies between 0 and 512 Pre-1970 British Pound, wit mean equals to about 33 and a median of around 14.5 Pounds.
  - The large majority of passengers embarks on the titanic in Southampton.
  - The fare varies between 0 and 512 Pre-1970 British Pound.
  - A large majority of passengers embarks on the titanic in Southampton.

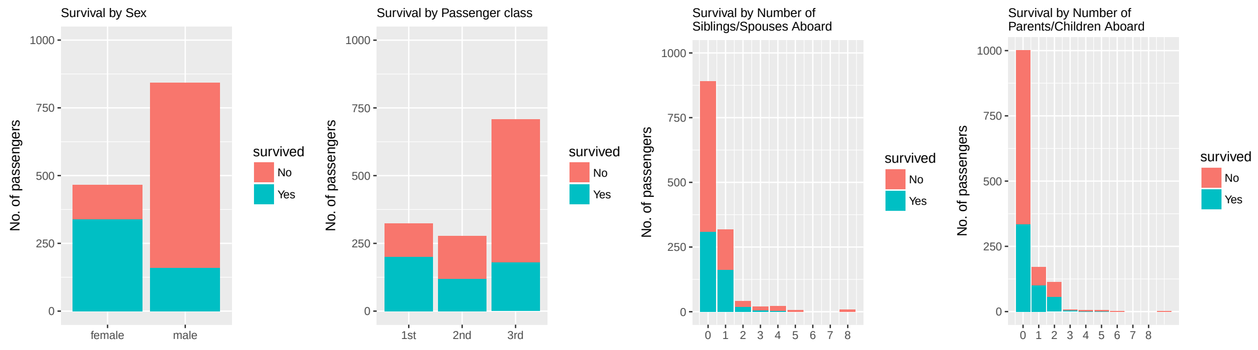
- Regarding missing values:
  - The feature *age* contains about 20% missing values. We assume that Age does have influence on predicting the survival, so we will try to handle these missing values later on.
  - The feature *body* contains about 91% missing values. It is quite a high amount. This variable has to be dropped by the model in any case, since it describes the identification number of bodies found after the accident, which is 100 percent correlated with a victim (as long as the data is not missing).
  - The amounts of missing vales of *fare* and *embarked* are very small. Maybe we would just drop the concerned passengers from the data set or replace the missing values by one certain value.
  - The feature *cabin* also contain a large amount of missing values (1014). While it might contain useful information about the survival probability, it is not trivial to deal with such a high percentage of missing data and we will drop this variable from our model.

In order to get a better overview of the data and support the variable selection, in the next following segments, we conduct an explorative analysis through a few plots in order to explore common pattern between the exploratory variables and the target feature *survived*. Firstly, we need to transform the the data type for easier plotting capabilities, as in the following segment.

```r
  # Transform some features, which are given as a vector
titanic$age <- as.numeric(titanic$age)
titanic$fare <- as.numeric(titanic$fare)
titanic$sibsp <- as.numeric(titanic$sibsp)
titanic$parch <- as.numeric(titanic$parch)

  # Some features also need to be factorized, so that numeric values
  # (for example 1,2,3) should not be treated as number but as category.
titanic$pclass <- factor(titanic$pclass)
titanic$survived <- factor(titanic$survived,levels = c(0,1),labels = c("No","Yes"))
titanic$sex <- factor(titanic$sex)
titanic$embarked <- factor(titanic$embarked)
```

We use bar plots for categorical features:

```r
p1 <- ggplot(titanic, aes(x = sex, fill = survived)) +
    geom_bar() +
    labs(subtitle = "Survival by Sex", y = "No. of passengers", x = " ") +
    ylim(0,1000)

p2 <- ggplot(titanic, aes(x = pclass,fill = survived)) +
    geom_bar() +
    labs(subtitle = "Survival by Passenger class", y = "No. of passengers", x = " ") +
    ylim(0,1000)

p3 <- ggplot(titanic, aes(x = sibsp, fill = survived)) +
    geom_bar() +
    labs(subtitle = "Survival by Number of\nSiblings/Spouses Aboard ",
         y = "No. of passengers", x = " ") +
    ylim(0,1000) +
    scale_x_continuous(breaks = c(0:8))

p4 <- ggplot(titanic, aes(x = parch, fill = survived)) + geom_bar() +
    labs(subtitle = "Survival by Number of\nParents/Children Aboard ",
         y = "No. of passengers", x = " ") +
    scale_x_continuous(breaks = c(0:8))

grid.arrange(p1,p2,p3,p4,ncol = 4)
```

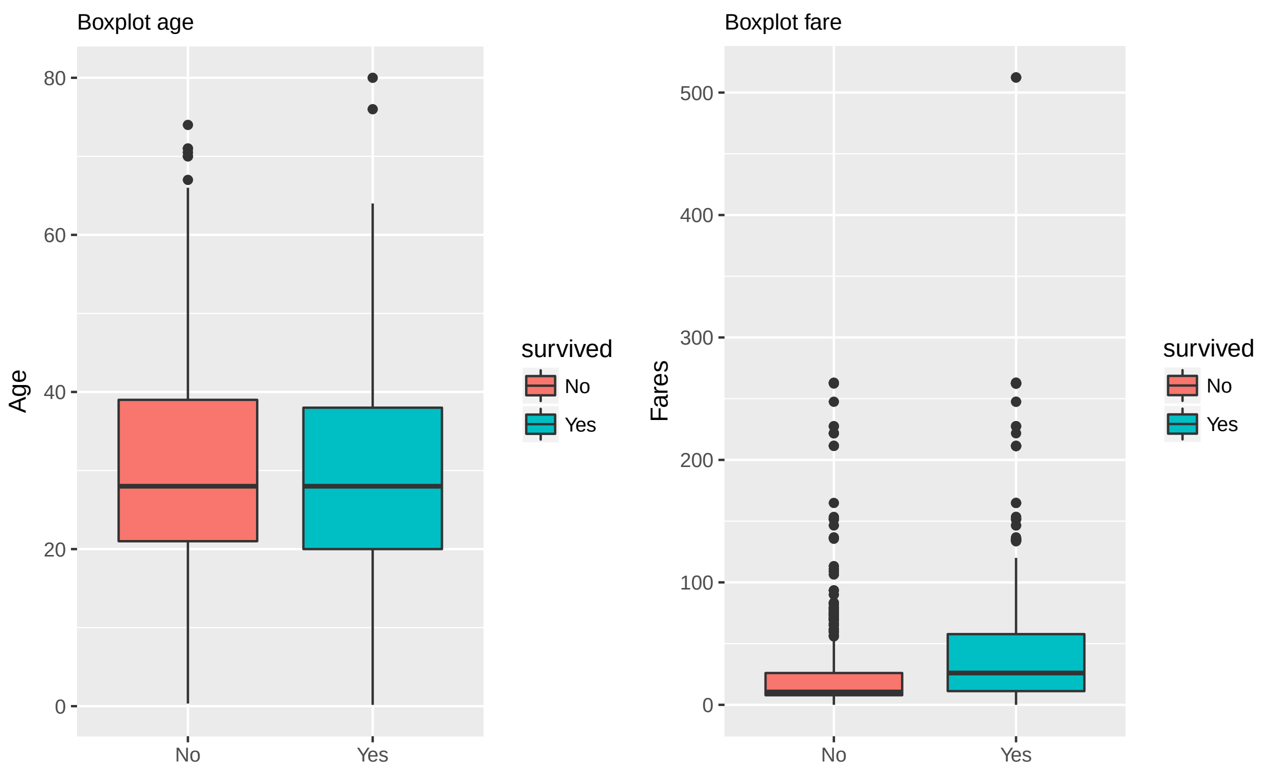


We learn the following things from these bar plots:

- *sex*: Females have a much higher chance of survival than males. The *sex* feature is important in our predictions
- *pclass*: Passengers from 3rd class are less likely to survive than those from 1st and 2rd class.
- *sibsp* and *parch*: having too many siblings/spouses/parents/children on board or travelling alone would decrease the chance of surviving. Travelling with one or two family members seems to raise the survival chances of the individuals.

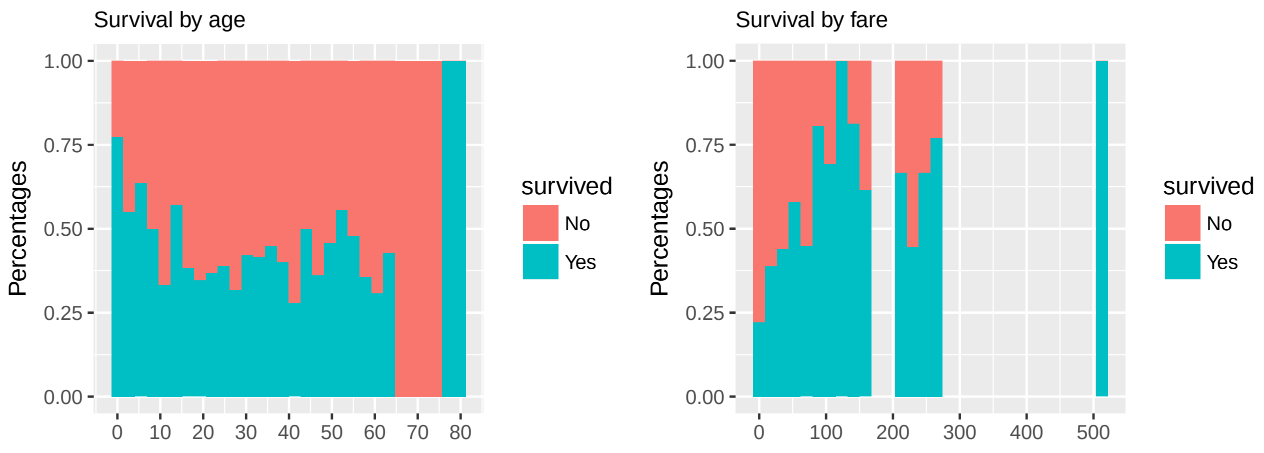
For metric features *age* and *fare* we use box plots.

```
p5 <- ggplot(titanic, aes(y = age, x = survived, fill = survived)) +
    geom_boxplot() +
    labs(subtitle = "Boxplot age", y = "Age", x = " ")

p6 <- ggplot(titanic, aes(y = fare, x = survived, fill = survived)) +
    geom_boxplot() +
    labs(subtitle = "Boxplot fare", y = "Fares", x = " ")

grid.arrange(p5,p6,ncol = 2)
```



It could be also useful to use a bar plot for metric features to see the trend better:

```
p7 <- ggplot(titanic, aes(x = age, fill = survived)) +
    geom_histogram(position = "fill", bins = 30) +
    labs(subtitle = "Survival by age", y = "Percentages", x = " ") +
    scale_x_continuous(breaks = seq(0, 90, by = 10))

p8 <- ggplot(titanic, aes(x = fare, fill = survived)) +
    geom_histogram(position = "fill", bins = 30) +
    labs(subtitle = "Survival by fare",
         y = "Percentages", x = " ")

grid.arrange(p7,p8,ncol = 2)
```



- *fare* and *age* contains some outliers, which should be treated later.
- The chance of survival shows a declining tendency, although all oldest passengers (about 80 years old) had survived. Passengers from 65-75 years old have the lowest chance of survival.
- The higher the fare is, the higher is the chance of survival.

It could be also helpful to take a look at the relationship of explanatory features:

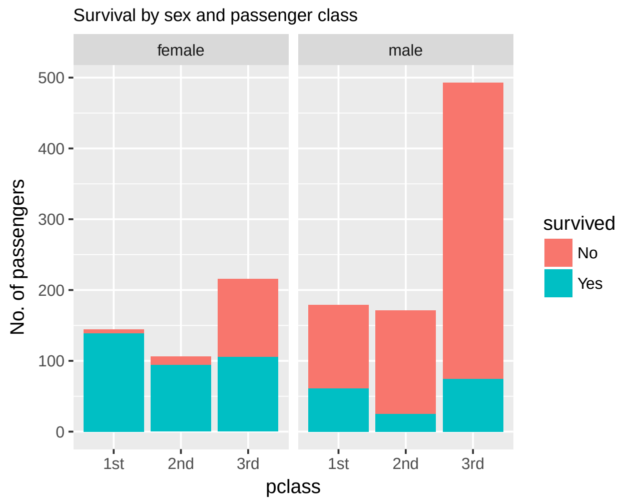
```r
dropValue <- c("name","cabin","ticket","boat","body","home.dest")
corDataframe <- titanic[ , !(names(titanic) %in% dropValue)]
corDataframe <- sapply(corDataframe, function(x) x <- as.integer(x))
round(cor(na.omit(corDataframe)), digits = 2)
```

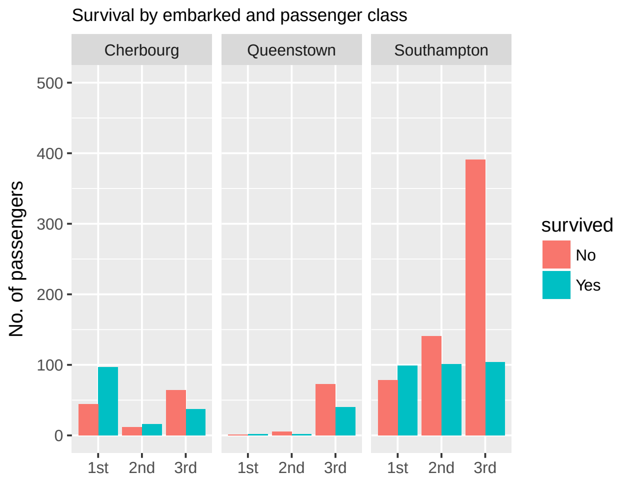
```
##          pclass survived   sex   age sibsp parch  fare embarked
## pclass     1.00    -0.32  0.14 -0.41  0.05  0.02 -0.57     0.28
## survived  -0.32     1.00 -0.54 -0.06 -0.01  0.12  0.25    -0.20
## sex        0.14    -0.54  1.00  0.07 -0.10 -0.22 -0.19     0.11
## age       -0.41    -0.06  0.07  1.00 -0.24 -0.15  0.18    -0.08
## sibsp      0.05    -0.01 -0.10 -0.24  1.00  0.37  0.14     0.05
## parch      0.02     0.12 -0.22 -0.15  0.37  1.00  0.22     0.01
## fare      -0.57     0.25 -0.19  0.18  0.14  0.22  1.00    -0.30
## embarked   0.28    -0.20  0.11 -0.08  0.05  0.01 -0.30     1.00
```

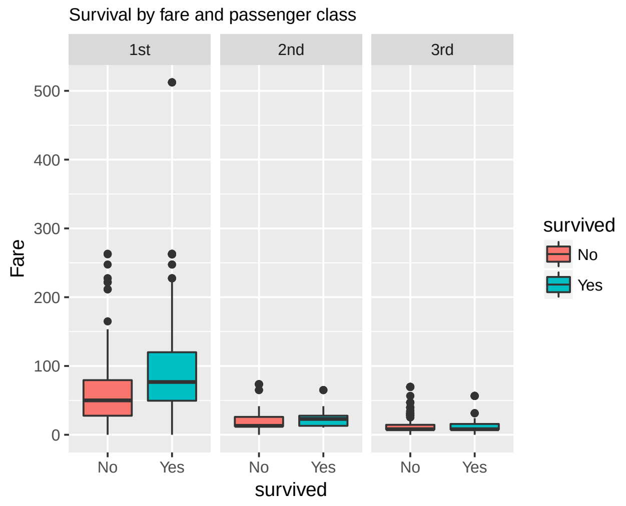
- Our target feature *survived* correlates mostly to *sex* and *pclass*, followed by *fare* and *embarked*.
- *pclass* and *fare* correlates strongly negative to each other. So the first class is more expensive than second and third class.
- *pclass* also correlates quite strongly with *age*. But it could also be a spurious correlation.
- *sibsp* and *parch* also correlate, since both declare family size

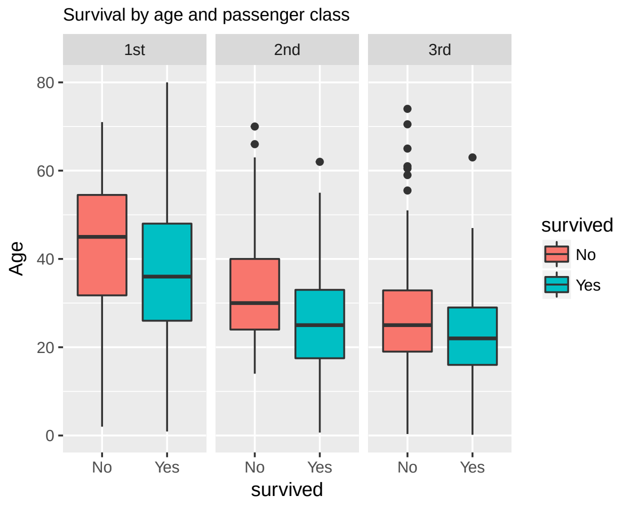
Let us show some multiple relationships graphically:

```r
p9 <- ggplot(titanic,aes(pclass, fill = survived)) +
    facet_grid( ~ sex) +
    geom_bar() + labs(subtitle = "Survival by sex and passenger class",
                      y = "No. of passengers")

p10 <- ggplot(subset(titanic, !is.na(embarked)), aes(x = pclass, fill = survived)) +
    facet_grid( ~ embarked) + geom_bar(position = "dodge") +
    labs(subtitle = "Survival by embarked and passenger class",
         y = "No. of passengers", x = "") + ylim(0,500)

p11 <- ggplot(titanic, aes(y = fare, x = survived, fill = survived)) +
    facet_grid( ~ pclass) +
    geom_boxplot() +
    labs(subtitle = "Survival by fare and passenger class",
         y = "Fare")

p12 <- ggplot(titanic, aes(y = age, x = survived, fill = survived)) +
    facet_grid( ~ pclass) + geom_boxplot() +
    labs(subtitle = "Survival by age and passenger class",
         y = "Age")

grid.arrange(p9, p10, p11, p12, ncol = 2)
```









- Female passengers in the first and second class are mostly survived, meanwhile men in all classes have a similar chance of survival, which is quite low.
- A major of passenger embarked in Southampton. Most of them did not survive, especially those in second and third classes.
- Unsurprisingly the first class is the most expensive one. The average fare of survived passengers is also higher in the first and second classes. This information could be better explained if we have more information about the relationship between *pclass* and *cabin*.
- The average age of first class passenger (regardless of survival) is higher than the rest.

That might be enough with plots. Let's move to the next step to choose the most important features for our model.

3.3 Identification of relevant regressors

Since some features do not to contain a lot useful information to our model - or indicates a nearly one to one relationship with the target feature - we drop them from our model. These are: *name, cabin, ticket, boat, body, home.dest*

- We assume that the name of a passenger does not have any influence on his/her chance of survival. (Although it might be possible to extract some useful information from this variable, such as social class. But for simplicity purposes, we decided to leave it out of our model)
- We also ignore cabin, since its amount of missing values is too big.

- Apparently the chance of survival does not depend on ticket and home-destination. The boat number and body identification number are available only after the Titanic had sunk. So they should not be involved in the model as well.

Let's drop these irrelevant features.

```
dropValue <- c("name","cabin","ticket","boat","body","home.dest")
titanic <-  titanic[ , !(names(titanic) %in% dropValue)]

# source: https://stackoverflow.com/questions/4605206
```

Missing values:
In the following segment, we check for the missing values pattern with help from the library *mice*. The output describes with 1's the available and with 0's the missing values for each variable in an aggregated manner. That is, on the leftmost column we obtain the sum of observations matching the available/missing values of the relative row. Further, on the last row, we obtain the sum of missing values for the relevant variable. With that, we can have a good overview not only about how much data went missing, but also some pattern among missing observations.

```
md.pattern(titanic)
```

```
##      pclass survived sex sibsp parch fare embarked age
## 1043      1        1   1     1     1    1        1   1   0
##  263      1        1   1     1     1    1        1   0   1
##    1      1        1   1     1     1    0        1   1   1
##    2      1        1   1     1     1    1        0   1   1
##           0        0   0     0     0    1        2 263 266
```

From the output, we can see that there are 1306 complete observations (within the subset data frame created above, since in the full data frame there are more missing values), one observation where only the variable *fare* is missing, two observations where *embarked* is missing and 263 where *age* is missing. Differently to the *fare* and *embarked* variables, 263 (~20%) is a significant amount and care is advised when dealing with such a high proportion of missing data. Even more, since it is very plausible that the data did not go randomly missing, but a more systematic cause can be underlying the problem. For instance, it is reasonable that the observations on the deceased passengers is more likely to go missing that on those who survived. Nonetheless, dropping those observations completely and neglecting the remaining information that these data points would convey, also affecting negatively our model.

Let's see how the variable *age* varies in relation to gender and class.

```
kable(
  aggregate(titanic[c(2,4:7)],
          by = list(titanic$pclass, titanic$sex),
          FUN = mean,
          na.rm = TRUE,
          na.action = na.pass),
  digits = 3
)
```

Group.1	Group.2	survived	age	sibsp	parch	fare
1st	female	NA	37.038	0.556	0.472	109.412
2nd	female	NA	27.499	0.500	0.651	23.235
3rd	female	NA	22.185	0.792	0.731	15.324
1st	male	NA	41.029	0.341	0.279	69.888
2nd	male	NA	30.815	0.327	0.193	19.905
3rd	male	NA	25.962	0.471	0.256	12.415

```
# source: https://stackoverflow.com/questions/16844613
```

Due to the great age variation among male female passengers in different classes, we replace the `NA` values conditional on their respective class and sex. This should perform better than naively using the `mean()` or `median()` values.

```
titanic[is.na(titanic$age) &
        titanic$pclass == "1st" &
        titanic$sex == "female", "age"] <- 37.03759

titanic[is.na(titanic$age) &
        titanic$pclass == "2nd" &
        titanic$sex == "female", "age"] <- 27.49919

titanic[is.na(titanic$age) &
        titanic$pclass == "3rd" &
        titanic$sex == "female", "age"] <- 22.18531

titanic[is.na(titanic$age) &
        titanic$pclass == "1st" &
        titanic$sex == "male", "age"] <- 41.02925

titanic[is.na(titanic$age) &
        titanic$pclass == "2nd" &
        titanic$sex == "male", "age"] <- 30.81540

titanic[is.na(titanic$age) &
        titanic$pclass == "3rd" &
        titanic$sex == "male", "age"] <- 25.96227
```

- **fare**: Replace 1 missing value by the mean *fare* in the concerned *pclass*.

For the variable *fare*, only the following observation is missing:

```
titanic[is.na(titanic$fare), ]
```

```
##      pclass survived  sex  age sibsp parch fare    embarked
## 1226    3rd       No male 60.5     0     0   NA Southampton
```

```
  # Replace the missing values by the median fare of the third class
medianFare <- median(titanic$fare[titanic$pclass == "3rd"], na.rm = TRUE)
titanic$fare[is.na(titanic$fare)] <- medianFare
```

- **embarked**: Replace missing value by the most frequent value.

```
summary(titanic$embarked)
```

```
##   Cherbourg  Queenstown Southampton        NA's
##         270         123         914           2
```

```
titanic$embarked[is.na(titanic$embarked)] <- "Southampton"
```

Let’s check once again, whether all missing values are handled.

```
any(is.na(titanic))
```

```
## [1] FALSE
```

So that is enough with cleaning the features. We are going to create the model in the next step.

3.4 Fitting a regression model

Let's create a train and test data set. Now we still have 1309 passengers in total.

```
set.seed(1)
sampleTitanic <- titanic[sample(nrow(titanic), replace = FALSE),]
head(sampleTitanic, 10)
```

```
##      pclass survived    sex      age sibsp parch    fare    embarked
## 348     2nd       No   male 42.00000     0     0 13.0000 Southampton
## 487     2nd       No   male 24.00000     0     0 10.5000 Southampton
## 749     3rd       No   male 34.00000     1     1 14.4000 Southampton
## 1187    3rd       No   male 25.96227     2     0 21.6792   Cherbourg
## 264     1st      Yes female 39.00000     1     0 55.9000 Southampton
## 1172    3rd       No   male 14.50000     8     2 69.5500 Southampton
## 1231    3rd       No female  2.00000     0     1 10.4625 Southampton
## 861     3rd      Yes female 26.00000     0     0  7.9250 Southampton
## 819     3rd      Yes female 16.00000     0     0  7.7333  Queenstown
## 81      1st       No   male 41.02925     0     0 26.5500 Southampton
```

```
# separate train and test data
train <- sampleTitanic[1:1100,]
test <- sampleTitanic[1101:1309,]
```

Let's fit the logistic regression. Here we use the backward selection to keep only relevant regressors. A backward selection removes every regressor from a model step by step and compare the performance of model with and without this regressor. The criterion used in this case is the Akaike Information Criterion (AIC). In general, the AIC rewards a model with high degree of fit, while including a penalty term for each added parameter.

```
  # Training a model with all filtered features. Here we use the training data.
modelFull <- glm(survived ~., family = binomial, data = train)
  # Run a backward selection
modelAIC <- step(modelFull, direction = "backward")
```

```
## Start:  AIC=1023.16
## survived ~ pclass + sex + age + sibsp + parch + fare + embarked
##
##            Df Deviance    AIC
## - fare      1   1003.3 1021.3
## - parch     1   1003.8 1021.8
## <none>          1003.2 1023.2
## - embarked  2   1013.3 1029.3
## - sibsp     1   1015.8 1033.8
## - age       1   1042.9 1060.9
## - pclass    2   1069.5 1085.5
## - sex       1   1267.7 1285.7
##
## Step:  AIC=1021.32
## survived ~ pclass + sex + age + sibsp + parch + embarked
##
##            Df Deviance    AIC
## - parch     1   1003.9 1019.9
## <none>          1003.3 1021.3
## - embarked  2   1014.1 1028.2
## - sibsp     1   1015.8 1031.8
## - age       1   1043.2 1059.2
## - pclass    2   1101.0 1115.0
## - sex       1   1271.4 1287.4
##
## Step:  AIC=1019.85
## survived ~ pclass + sex + age + sibsp + embarked
##
##            Df Deviance    AIC
## <none>          1003.9 1019.9
## - embarked  2   1014.6 1026.6
## - sibsp     1   1019.0 1033.0
## - age       1   1043.3 1057.3
## - pclass    2   1101.5 1113.5
## - sex       1   1279.1 1293.1
```

Based on the AIC, the selected model includes the *pclass, sex, age, sibsp and embarked* features. We were expecting that *fare* would remain part of the selected model, based on the plots created above. In any case, it might be due to the fact that the fare price highly correlates with the ticket classes. In that sense, the actual price might not contain much *added* information to the model. Surprisingly, the city where the passengers embarked remains in the selected model. We expect that this might be due to some correlation between the port of embarking and the social status of the passenger, that is not completely captured in the ticket class varible. For the subsequent analysis, we use the model selected by the AIC. But, for comparison purposes, we will discuss and compare the results of the small model (*modelAIC*) - with the ones for the full model (*modelFull*) in the section 3.7, when we discuss model accuracy.

3.5 Discussion of model fit

Let's take a look at the significance of regressors in our model.

```r
summary(modelAIC)
```

```
##
## Call:
## glm(formula = survived ~ pclass + sex + age + sibsp + embarked,
##     family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5705  -0.6690  -0.4341   0.6571   2.6029
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          4.369486   0.393924  11.092  < 2e-16 ***
## pclass2nd           -1.081288   0.244310  -4.426 9.61e-06 ***
## pclass3rd           -2.190450   0.239827  -9.133  < 2e-16 ***
## sexmale             -2.566980   0.171936 -14.930  < 2e-16 ***
## age                 -0.043334   0.007186  -6.030 1.64e-09 ***
## sibsp               -0.346029   0.097965  -3.532 0.000412 ***
## embarkedQueenstown  -0.629794   0.328419  -1.918 0.055155 .
## embarkedSouthampton -0.670480   0.204961  -3.271 0.001071 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 1459.0  on 1099  degrees of freedom
## Residual deviance: 1003.8  on 1092  degrees of freedom
## AIC: 1019.8
##
## Number of Fisher Scoring iterations: 5
```

One way of identifying relevant regressors in via the p-value. The p-value is the result of a test, which under its Null Hypothesis the variable coefficient is equal to zero, i.e. irrelevant to explaining the predicted variable. A low p-value indicates that a low probability of gathering such an *extreme* estimated value, given the hypothesis that the *real* value is actually zero. What "low" indeed means is somewhat arbitrary and depends on the model application. In the selected model, all variables has a p-value lower than 0.05 and is, therefore, said to be statistically significant, with exception of *embarkedQueenstown*, which lies marginally above this cut-off point.

```r
anova(modelAIC, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: survived
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                      1099     1459.0
## pclass    2   99.341      1097     1359.6 < 2.2e-16 ***
## sex       1  293.263      1096     1066.4 < 2.2e-16 ***
## age       1   34.842      1095     1031.5 3.576e-09 ***
## sibsp     1   16.940      1094     1014.6 3.859e-05 ***
## embarked  2   10.733      1092     1003.9   0.00467 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The anova() function allows us to see the difference between the null and residuals deviances. The null deviances is 1497.9, which indicates the performance of the model with only the intercept based on the Chi-Square values. The difference between the null deviance and the residual deviance shows how well the actual model performs against the null model. Again, feature *sex* performs the best regarding the significant improvement of the residual deviance, following by *pclass* and *age*. Other features also reduce the residual deviance, even though just a bit, and the significance are way worse.

3.6 Interpreting the model

The interpretation of the parameters in a logistic regression is not as straightforward as in a (strictly) linear regression. This is due to the restriction imposed on the dependent variable, which can only take values between zero and one. This restriction is observed by constructing a model[1], such as

$$\pi_i = P(Y_i = 1 \mid \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_k x_{ik}) = F(\eta_i) \tag{1}$$

[1]This model derivation is based on the lectures notes and slides from *Statistische Modellierung*, given by Prof. Dr. Schmid last Fall. Another useful resource was the *Lecture Notes on Generalized Linear Models* by by Germán Rodríguez. Available on http://data.princeton.edu/wws509/notes/. Accessed on 13.11.2017.

where π_i is the probability of a *desired* event happening, $\eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_k x_{ik}$, and $F()$ is the logit-function defined as

$$logit(\eta_i) = \frac{exp(\eta_i)}{1 + exp(\eta_i)} = \pi_i. \tag{2}$$

With that, one cannot directly interpret the variable coefficients. To circumvent this problem, one can refer to the *odds* of the occurrence of an event. This is defined as the probability of the event occurring against the probability that it does not. Formally,

$$Odds(\pi_i) := \frac{P(Y=1)}{P(Y=0)} = \frac{P(Y=1)}{1 - P(Y=1)} = \frac{\pi_i}{1-\pi_i}. \tag{3}$$

This term is often used in gambling scenarios. That is, if the odds of winning in a hand of poker is 3 to 1 (i.e. 3/1 or 3:1), it is expected that the player wins 3 times out of every 4 games, which equals to 75 percent. And, as expected $3 = \frac{0.75}{1-0.75}$.

From Equations (2) and (3), one can show that

$$\frac{\pi_i}{1-\pi_i} = \frac{\frac{exp(\eta_i)}{1+exp(\eta_i)}}{\frac{1}{1+exp(\eta_i)}} = exp(\eta_i). \tag{4}$$

Moreover, taking the log on both sides of equation (4), one gets

$$log\left(\frac{\pi_i}{1-\pi_i}\right) = \eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_k x_{ik}. \tag{5}$$

Furthermore, due to the product rule of exponentials, one can show that

$$Odds_1(\pi \mid x_{i1}) = \frac{P(Y=1 \mid x_{i1})}{P(Y=0 \mid x_{i1})} = exp(\beta_0) \cdot exp(\beta_1 x_{i1}) \cdot \ ... \ \cdot exp(\beta_k x_{ik}). \tag{6}$$

And, if one increases x_{i1} by one unit, while keeping other variables constant, one gets

$$Odds_2(\pi \mid x_{i1}+1) = \frac{P(Y=1 \mid x_{i1}+1)}{P(Y=0 \mid x_{i1}+1)} = exp(\beta_0) \cdot exp(\beta_1(x_{i1}+1)) \cdot \ ... \ \cdot exp(\beta_k x_{ik}). \tag{7}$$

Finally, if we divide $Odds_2$ by $Odds_1$, what is defined as the *Odds Ratio* between the two scenarios, we see that all terms cancel themselves out, leaving only the term with the added unit, as in

$$Odds\ Ratio_{(2,1)} = \frac{Odds_2(\pi, x_{i1}+1)}{Odds_1(\pi, x_{i1})} = exp(\beta_1). \tag{8}$$

For instance, if $\beta_1 = 0.3$ and x_1, $exp(\beta_1) \approx 1.35$, which means that - for a continuous x - the **odds** (not the probability!) of the event occurring increases by about 35 percent, that means, it gets multiplied by 1.35, as x increases by one unit. One can observe that while the Odds Ratio remains constant for any x, the probability itself depends on the base value one starts with. In case x_1 was a dummy variable (such as $x_1 = 0$, for male and $x_1 = 1$ for female) and keeping the same coefficient as illustrated above, than the odds of the event occurring is 35 percent higher for a female, when comparing to a male with, otherwise, same characteristics.

In general, if:

- $\beta_i > 0 \Rightarrow exp(\beta_i) > 1 \Rightarrow Odds\ increases$

- $\beta_i < 0 \Rightarrow 0 < exp(\beta_i) < 1 \Rightarrow Odds\ decreases$
- $\beta_i = 0 \Rightarrow exp(\beta_i) = 1 \Rightarrow Odds\ remains\ constant$

With that in mind, let's have a look at the each coefficient, its exponential alongside their 95% confidence intervals.

```r
kable(data.frame(
  coef = round(modelAIC$coef, 2),
  "exp_coef" = round(exp(modelAIC$coef),2),
  confint_low = round(exp(confint(modelAIC)[1:8]),2),
  confint_high = round(exp(confint(modelAIC)[9:16]),2))
)
```

	coef	exp_coef	confint_low	confint_high
(Intercept)	4.37	79.00	37.23	174.67
pclass2nd	-1.08	0.34	0.21	0.55
pclass3rd	-2.19	0.11	0.07	0.18
sexmale	-2.57	0.08	0.05	0.11
age	-0.04	0.96	0.94	0.97
sibsp	-0.35	0.71	0.58	0.85
embarkedQueenstown	-0.63	0.53	0.28	1.01
embarkedSouthampton	-0.67	0.51	0.34	0.76

From the above table, one can observe that all variables, due to the way the model was specified, has a negative impact in the probability of survival. More specifically, the odds of an individual from the 2nd class surviving - while keeping other variables constant - is about a third of an *similar* individual in the 1st class. A passenger from the third class has its odds of survival decreased to about a tenth of one from the 1st class.

The strongest predictor of survival is the variable *sex*. A male passenger has his odds of survival diminished to about eight percent of that of a female passenger. Also an increasing *age* has a negative effect on the chances of surviving the tragedy, but it is not as strong as the already mentioned variables. We expected a stronger effect of age, to be inline with the social rule of providing safety and protection to women and children first. While this is true with women, the weak effect of age might be due to the fact that the effect is not linear throughout ones lifetime. As we could see from the plots discussed above, children had a higher rate of survival, with younger adults having actually a lower rate and older adults and elderly have a quite high rate of survival. This may be due to the fact that the ticket class is correlated with older passengers, and thus, diminishes the negative effect of survival of an adult when compared to a child. And that it is also a social rule to protect the elderly first in such situations.

Surprisingly, the number of siblings or spouses aboard decreases the odds of survival quite significantly. We suspect that this is due to people in the lower classes travelling in bigger families, but further analysis is required in order to test this hypothesis.

Moreover, when compared to the passengers that embarked in Cherbourg, those from Queenstown are technically (to the statistical significance of 95%) not necessarily negatively affect by the chosen city of embarkment. Those that embarked in Southampton, however, had their odds of survival decreased by about half.

3.7 Model Accuracy

Let's see how good our model performs regarding the accuracy and compare to the full model, as specified above.

```r
  # Predict the test data. The column survived is hidden during the predicting.
testPredict <- predict(modelAIC,type = 'response', newdata = test[-2])

  # Assign the results into 2 categories of survival.
survivedPredict <- ifelse(testPredict > 0.5,"Yes","No")

  # Create a confusion matrix and print it.
confusionMatrix <- table(test[,2],survivedPredict)
confusionMatrix
```

```
##      survivedPredict
##         No Yes
##   No   108  17
##   Yes   22  62
```

```r
  # Calculate the accuracy and print it.
accuracy <- mean(survivedPredict == test$survived)*100
print(paste("The (restricted) model accuracy is", round(accuracy,digits = 2), "%"))
```

```
## [1] "The (restricted) model accuracy is 81.34 %"
```

In order to test if the classification is indeed better than randomly picking values, we conducted the Press's Q Test, which consist of the following test equation.

$$Press's\ Q = \frac{(n - (n \cdot G \cdot \alpha))^2}{n \cdot (G - 1)} \sim \chi^2_1$$

```r
n <- length(titanic$survived)
pq <- (n - n * 2 * accuracy)^2/n
1 - pchisq(pq,1)
```

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

As one can see, the test indicates that is extremely improbable that the model randomly assigned the correct values around 81 percent of the time. In any case, how does the restricted model compares to the full one, when only considering their predictive power.

For that, we calculate the predictive accuracy for the full model.

```r
  # Predict the test data. The column survived is hidden during the predicting.
testPredict2 <- predict(modelFull,type = 'response', newdata = test[-2])

  # Assign the results into 2 categories of survival.
survivedPredict2 <- ifelse(testPredict2 > 0.5,"Yes","No")

  # Create a confusion matrix and print it.
confusionMatrix <- table(test[,2],survivedPredict2)
confusionMatrix
```

```
##      survivedPredict2
##         No Yes
##   No   111  14
##   Yes   23  61
```

```r
  # Calculate the accuracy and print it.
accuracy <- mean(survivedPredict2 == test$survived)*100
print(paste("The (full) model accuracy is", round(accuracy,digits = 2), "%"))
```

```
## [1] "The (full) model accuracy is 82.3 %"
```

One can observe that the accuracy of the unrestricted model is slightly superior to that of the restricted one. This model, although includes variables that are statistically insignificant performs better, when the application only concerns better classification accuracy. This exemplifies that a "best" model is not a robust concept, since it always depends on the application intended for the model. Econometricians usually aspire to construct models with great explanatory power for each variable. This points towards more parsimonious models. If, on the other hand, the intend is to create a model, where the main objective is to achieve predictions as accurate as possible - as it is the case in machine learning classifiers, for instance - it would point us towards a bigger model, including variables that might be statistically insignificant, but increases prediction accuracy.

There are still several ways to improve the quality of the data set as well as the performance of our predicted model. There is a method called k-fold cross validation, which is used to limit problems like over-fitting (over-fitting: a model which contains too many explanatory variables than necessary). This method partitioned randomly the data set into k equal sized subsets. One of these subsets serves as test data and the rest (k-1) as training data. The cross-validation repeats k-times and returns k different results, which could be averaged to one final result.

So that is. We know that there are still several things need to be improved in our model. But we hope that we did it in the right direction and got the right idea of analysing a logistic regression from a given data set.

Thank you in advance.