Exploratory Data Analysis

Aldo Solari

Outline

1. **Titanic Competition**
2. **The Classification Setting**
3. **Data Science Project**
   - Import
   - Variables
   - Missing Values
   - Exploratory Data Analysis
   - Feature Engineering

Titanic: statistical learning from disaster

On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew

Goal

- One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew
- Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class
- The goal is to predict a 0 or 1 value for the `survived` variable for each passenger in the test set

Adapted from the Kaggle competition "Titanic: Machine Learning from Disaster".

See https://www.kaggle.com/c/titanic

Outline

1. Titanic Competition
2. **The Classification Setting**
3. Data Science Project
    - Import
    - Variables
    - Missing Values
    - Exploratory Data Analysis
    - Feature Engineering

The classification setting

Binary response

$Y \in \{0, 1\}$

Regression function

$f(x) = \mathbb{E}(Y|X = x) = \Pr(Y = 1|X = x)$

Bayes' classification rule

$$C(x) = \begin{cases} 1 & \text{if } f(x) > 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Bayes error rate

- A classification rule is any function $\hat{C} : x \mapsto \{0, 1\}$
- For example, the plug-in rule

$$\hat{C}(x) = \begin{cases} 1 & \text{if } \hat{f}(x) > 1/2 \\ 0 & \text{otherwise} \end{cases}$$

  where $\hat{f}(x)$ is an estimate of $f(x)$ based on training data
- The Bayes classifier is optimal because it has the smallest error rate:

$$\mathbb{E}\left[\Pr(Y \neq C(x))\right] \leq \mathbb{E}\left[\Pr(Y \neq \hat{C}(x))\right] \quad \forall \hat{C}$$

  where the expectation averages the probability over all possible values of X
- The Bayes error rate $\mathbb{E}\left[\Pr(Y \neq C(x))\right]$ is analogous to the irreducible error

Missclassifications

- Training set: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

$$\mathrm{Err}_{\mathrm{Tr}} = \frac{1}{n} \sum_{i=1}^{n} I\{y_i \neq \hat{c}(x_i)\}$$

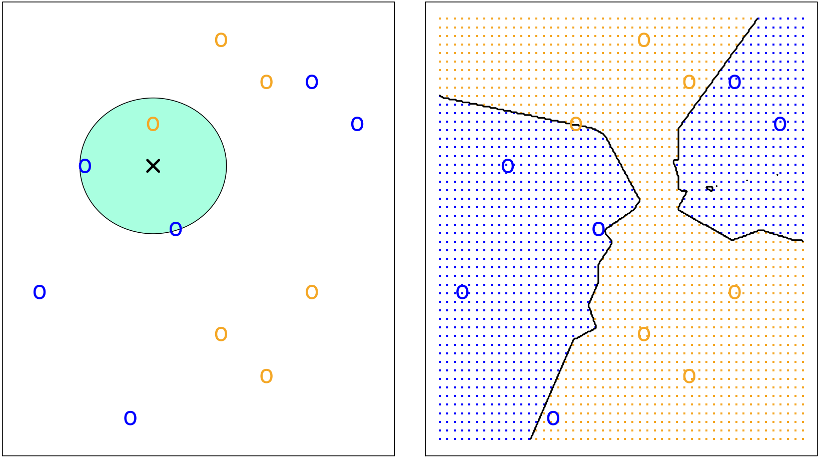
- Test set: $(x_1^*, y_1^*), (x_2^*, y_2^*), \ldots, (x_m^*, y_m^*)$

$$\mathrm{Err}_{\mathrm{Te}} = \frac{1}{m} \sum_{i=1}^{m} I\{y_i^* \neq \hat{c}(x_i^*)\}$$

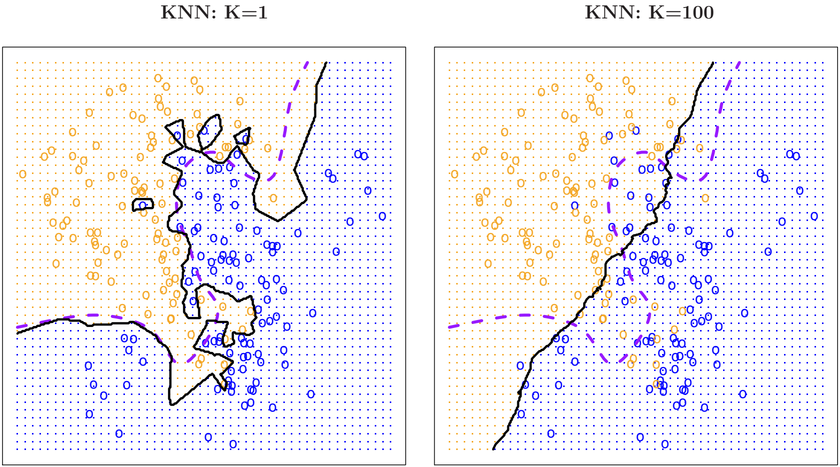
- Accuracy

$$\mathrm{Acc}_{\mathrm{Te}} = 1 - \mathrm{Err}_{\mathrm{Te}}$$

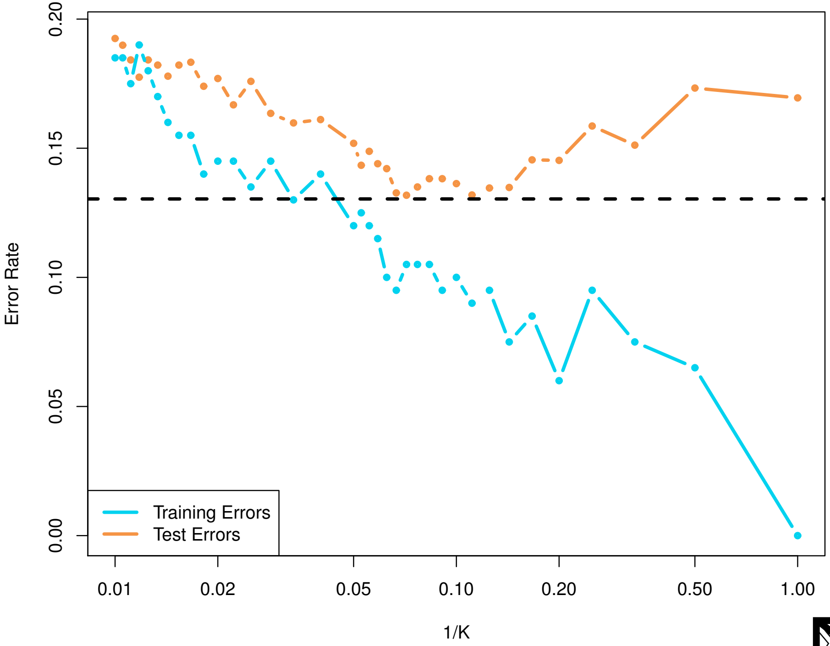
k-nearest-neighbor classifier



Source: ISL p. 40



Source: ISL p. 41



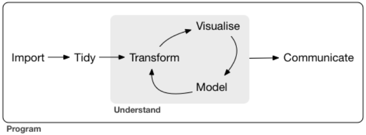
Source: ISL p. 42

Outline

1. Titanic Competition
2. The Classification Setting
3. **Data Science Project**
   - Import
   - Variables
   - Missing Values
   - Exploratory Data Analysis
   - Feature Engineering

Data science project

- A typical data science project looks something like this:



- First you must import your data into R
- Once you've imported your data, it is a good idea to tidy it
- Then you have to understand your data by exploratory data analysis (visualisation and transformation) and modelling
- The last step is communication

Source: http://r4ds.had.co.nz/introduction.html

Import data

```
train <- read.csv("titanic_tr.csv",
         header = TRUE,
         stringsAsFactors = FALSE)

test <- read.csv("titanic_te.csv",
         header = TRUE,
         stringsAsFactors = FALSE)
```

For more advanced functions:

http://r4ds.had.co.nz/data-import.html

Variable descriptions

http://biostat.mc.vanderbilt.edu/twiki/pub/Main/DataSets/titanic3info.txt

```
pclass          Passenger Class
                (1 = 1st; 2 = 2nd; 3 = 3rd)
survived       Survival
                (0 = No; 1 = Yes)
name            Name
sex             Sex
age             Age
sibsp           Number of Siblings/Spouses Aboard
parch           Number of Parents/Children Aboard
ticket          Ticket Number
fare            Passenger Fare
cabin           Cabin
embarked        Port of Embarkation
                (C = Cherbourg; Q = Queenstown; S = Southampton)
```

Type of variables

```
# combine data sets
combi <- rbind(train, test)
# check type of variables
str(combi)

# convert pclass, sex, embarked to factors
combi$pclass <- as.factor(combi$pclass)
combi$sex <- as.factor(combi$sex)
combi$embarked <- as.factor(combi$embarked)

# copy of the response as a factor for better readability
combi$survived01 <- combi$survived
combi$survived <- as.factor(combi$survived01)
levels(combi$survived) = c("Death","Alive")
```

Missing values

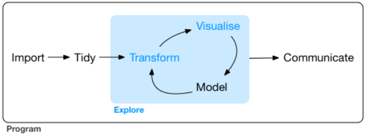
```
# cabin has missing values coded as "" instead of NA
combi$cabin[combi$cabin==""] <- NA

# where are the missing values?
summary(combi)

# fare: 1 missing value
# embarked: 2 missing values
# age: 20% missing values
# cabin: 77% missing values
```

Imputing missing values

```
# embarked
combi[which(is.na(combi$embarked)), ]
boxplot(fare~ pclass + embarked, data=combi); abline(h=80)
combi$embarked[which(is.na(combi$embarked))] <- c("C","C")

# fare
combi[which(is.na(combi$fare)), ]
aggregate(fare ~ pclass + embarked, combi, FUN=median)
combi$fare[which(is.na(combi$fare))] <- 8.0500

# age
aggregate(age ~ pclass + sex, combi, FUN=mean)
fit.age <- lm(age ~ sex + pclass,
          data = combi[!is.na(combi$age),])
combi$age[is.na(combi$age)] <- predict(fit.age,
          newdata=combi[is.na(combi$age),])
```

Exploratory data analysis

EDA is an iterative cycle:

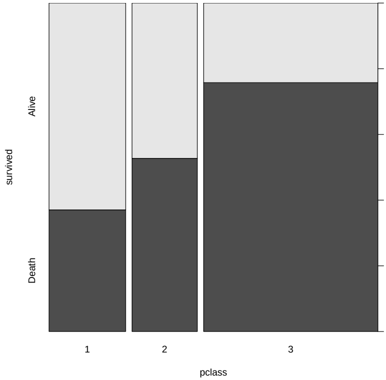


1. Generate questions about your data
2. Search for answers by visualising, transforming, and modelling your data
3. Use what you learn to refine your questions and or generate new questions

Source: http://r4ds.had.co.nz/explore-intro.html

Survived $\sim$ pclass

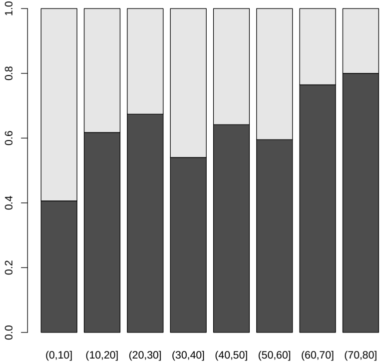
Rich people survived at a higher rate?

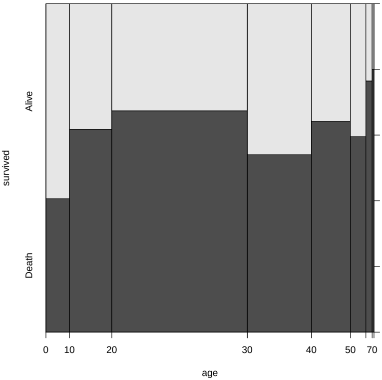



```
plot(survived ~ pclass, train)
```

Survived $\sim$ age

What is the relationship between age and survival?

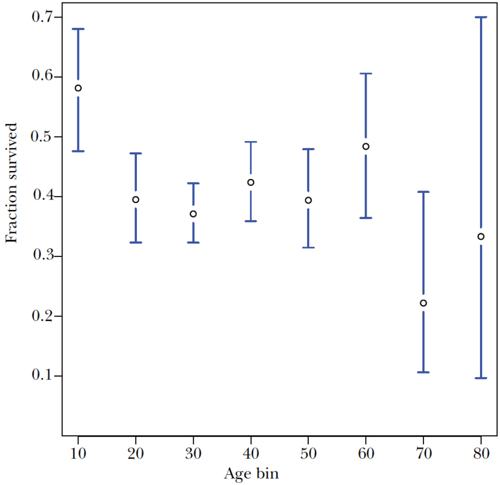




Better visualization

```
plot(survived ~ age, train)
```

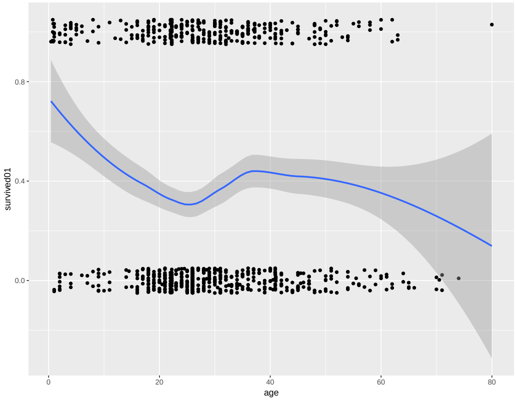
Figure 3
***Titanic* Survival Rates by Age Group**



Notes: The figure shows the mean survival rates for different age groups along with confidence intervals. The age bin 10 means "10 and younger," the next age bin is "older than 10 through 20," and so on.

Source: Varian (2014)

ggplot




```
ggplot(train, aes(x=age, y=survived01)) + geom_smooth()
```

Logistic model

Table 3

Logistic Regression of Survival versus Age

Coefficient	*Estimate*	*Standard error*	*t value*	*p value*
Intercept	0.465	0.0350	13.291	0.000
Age	−0.002	0.001	−1.796	0.072

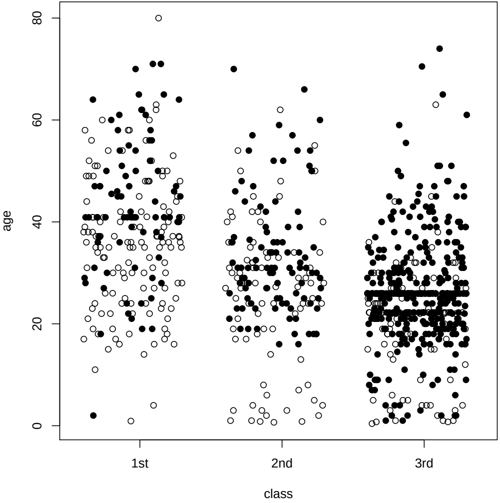
Note: Logistic regression relating survival (0 or 1) to age in years.

- the logistic model seems to suggest that age is not an important predictor of survival
- however, the relationship between age and survival is not linear

Source: Varian (2014)

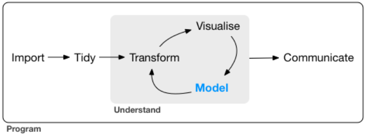
Survived $\sim$ pclass + age

What about class and age combined?



Modelling (basic)

Let's use what we've learned to build a basic model



Hal Varian (2014) Big Data: New Tricks for Econometrics, *Journal of Economic Perspectives* 28:3-28 illustrates the use of classification trees with the R package `rpart` to predict `survived` as a function of `pclass` and `age`

Null model

- The null model uses only the information of the response
- Training data: 38.38% of passengers survived, 61.62% died
- The null model prediction for all the passengers in the test set is "death" (the mode of the response)

```
yhat <- rep("Death",m)

# confusion matrix
table(yhat, test$survived)
         true
predicted Death Alive
    Death   260   158

# accuracy
mean(yhat == test$survived)
0.622
```

Classification trees

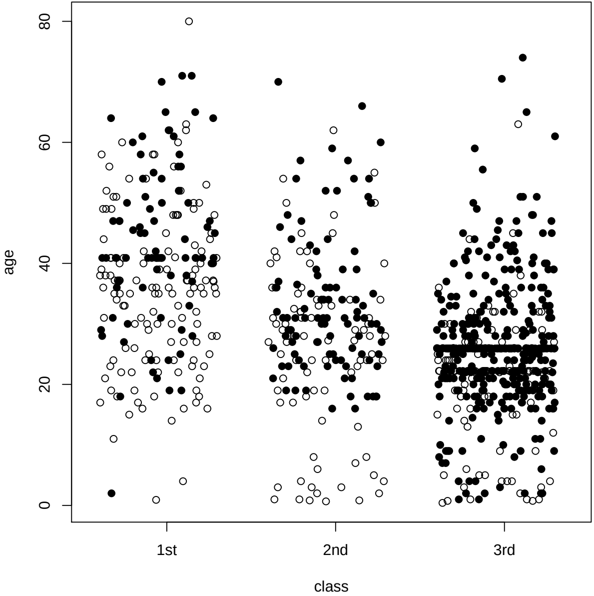
- Classification trees recursively partition the sample space into smaller and smaller rectangles
- To see how this works, consider the response $Y = \texttt{survived}$ and two predictors $X_1 = \texttt{pclass}$ and $X_2 = \texttt{age}$
- Begin by splitting the predictor space into two regions on the basis of a rule of the form

$$X_1 \leq x_1, \quad x_1 \in \{1, 2, 3\}$$

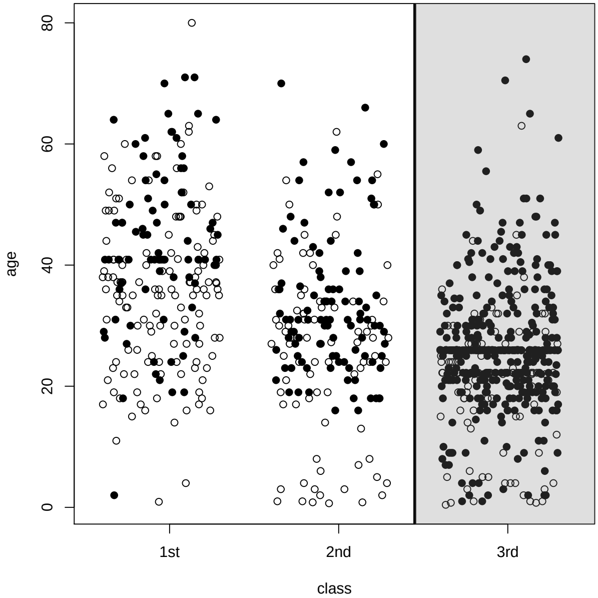
$$X_2 \leq x_2, \quad x_2 \in [0.42, 80]$$

- The optimal split, in terms of reducing the missclassification error (or the Gini index or the Deviance) is found over all variables and all possible split points
- The process is then repeated in a recursive fashion for each of the two sub-regions

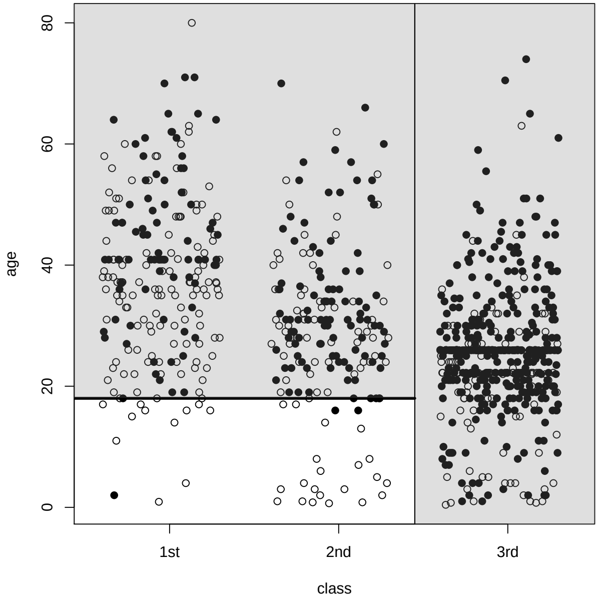
Where is the 1st split?



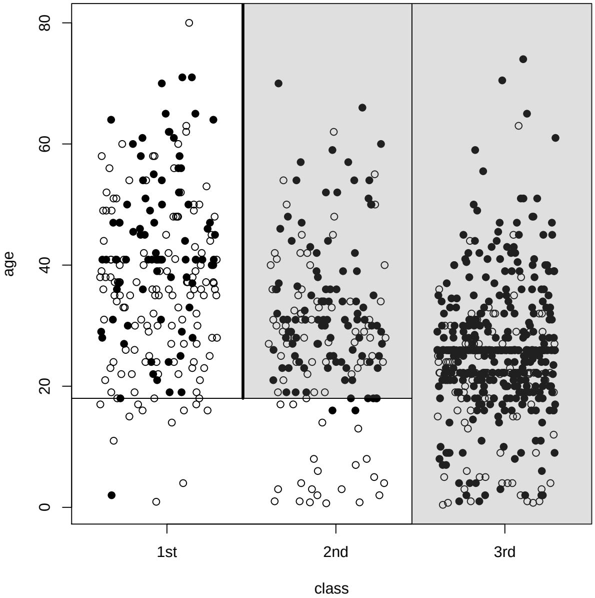
1st split



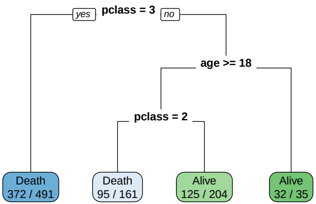
2nd split



3rd split



Classification rule

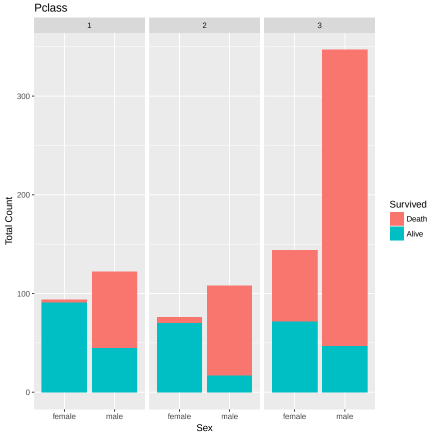


Class 3	Death	76%
Class 1-2, younger than 18	Alive	91%
Class 2, older than 18	Death	56%
Class 1, older than 18	Alive	61%

rpart

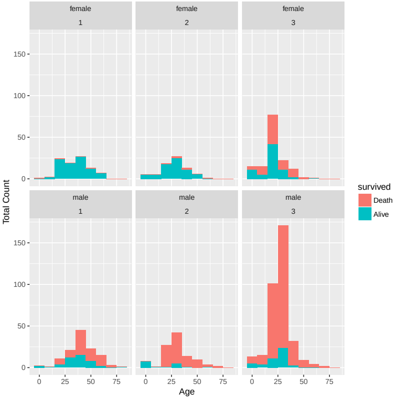
```
library(rpart)
fit.rpart <- rpart(survived ~ pclass + age, train,
               control=rpart.control(maxdepth =  3))

library(rpart.plot)
rpart.plot(fit.rpart, type=0, extra=2)
yhat <- predict(fit.rpart, newdata=test, type="class")

# confusion matrix
table(yhat, test$survived)
           true
 predicted Death Alive
     Death   215    86
     Alive    45    72
# accuracy
mean(yhat == test$survived)
0.6866029
```

Back to data visualization: $\sim$ pclass + sex




```
# visualize the relationship of sex and pclass with surviva
ggplot(train, aes(x = sex, fill = survived))
  + geom_bar()
  + facet_wrap(~ pclass)
```

Survived ~ pclass + sex + age

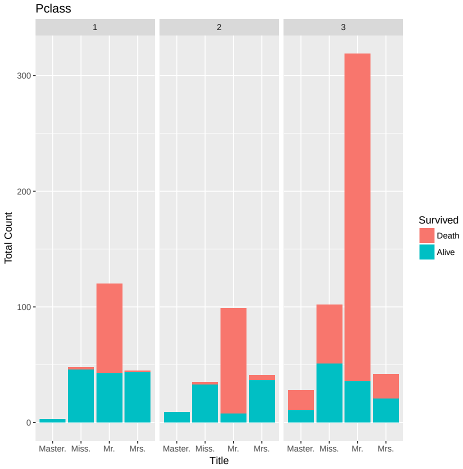



```
# visualize the relationship of sex, pclass, age with survi
ggplot(train, aes(x = age, fill = survived))
+ facet_wrap(~sex + pclass)
+ geom_histogram(binwidth = 10)
```

Data transformation (feature engineering)

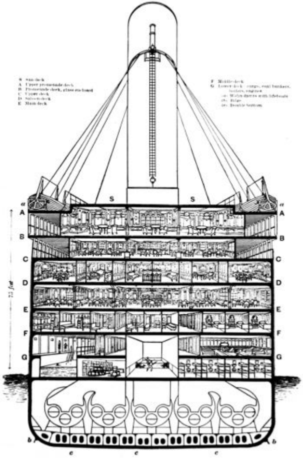
```
# passenger title is contained within the passenger name
combi$name[1]
#D.Langer Data Wrangling & Feature Engineering with dplyr
library(dplyr)
library(stringr)
combi <- combi %>%
     mutate(title = str_extract(name, "[a-zA-Z]+\\."))
table(combi$title)

    Capt.      Col. Countess.       Don.      Dona.
        1         4         1          1          1
      Dr. Jonkheer.     Lady.     Major.    Master.
        8         1         1          2         61
    Miss.     Mlle.      Mme.        Mr.       Mrs.
      260         2         1        757        197
      Ms.      Rev.      Sir.
        2         8         1
```

Survived $\sim$ pclass + title




```
ggplot(combi[1:n, ], aes(x = title2, fill = survived))
  + geom_bar()
  + facet_wrap(~pclass)
```

Use of external informations



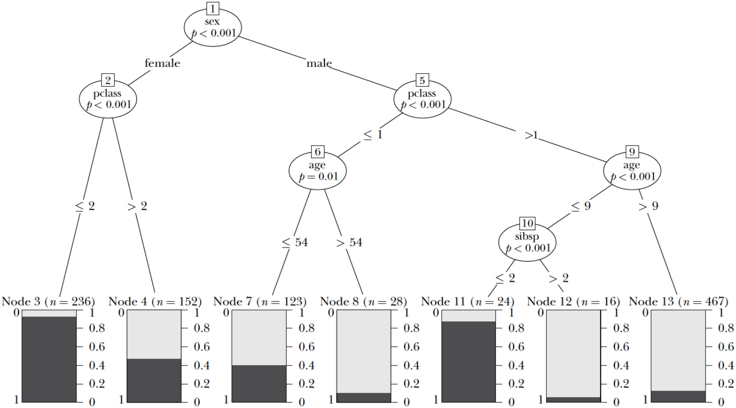
```
# the first character of cabin is the deck
table(substr(combi$cabin, 1, 1))
 A  B  C  D  E  F  G  T
22 65 94 46 41 21  5  1
```

Conditional inference trees

- One problem with trees is that they tend to overfit the data
- The most common solution to this problem is to prune the tree by imposing a cost for complexity (e.g. number of terminal nodes)
- Conditional inference trees (`ctree`) choose the structure of the tree using a sequence of hypothesis tests
- The resulting trees tend to need very little pruning
- From Hal Varian (2014) code

```
library(party)
ctree(survived ~ pclass + sex + age + sibsp, data = train)
```

Figure 4
A ctree for Survivors of the *Titanic*
(black bars indicate fraction of the group that survived)



Source: Hal Varian (2014) p. 12

Interpretation of Figure 4

- The first node divides by gender
- The second node then divides by class.
- In the right-hand branches, the third node divides by age, and a fourth node divides by the number of siblings plus spouse aboard
- One might summarize this tree by the following principle: "women and children first ... particularly if they were traveling first class."
- This simple example again illustrates that classification trees can be helpful in summarizing relationships in data, as well as predicting outcomes

Source: Hal Varian (2014) p. 12

`ctree` **survived ~ pclass + title + fsize**