# SmartFly: Prepare Data For Modeling

Cindy Lamm

January 14, 2015

Load preprocessed data from the previous step "Exploratory Data Analysis"

```
rm(list=ls())   #clear memory
load("../01_exploratory_data_analysis/trainDataTyped.Rdata")
```

## 0.1 Analyse & deal with missing values

The variable `cancellation_code` has the most missing values since it is only filled if the flight was not cancelled - which is the case for most flights. I would expect that all non-cancelled flights don't have a `cancellation_code`, however I found 168 non-cancelled flights that do have an entry as `cancellation_code`, I'll exclude those observations from the data set in the following.

```
weird_idx <- which(!trainDataTyped$cancelled & !is.na(trainDataTyped$cancellation_code))
trainDataTyped <- trainDataTyped[-weird_idx,]
n_obs <- dim(trainDataTyped)[1]
n_obs

## [1] 7374197
```

Often observations with any missing value are excluded from the modeling stage. There would by 7270012 observations left if I would exclude observations that have `NA` for any of the variables (excluding `cancellation_code`). I do not remove the 104185 rows with `NA`s yet since I don't know which variables I will use for the modeling.

Split the train data based on simple bootstrap resampling into a series of train and test sets

```
library(caret)

## Loading required package:  lattice
## Loading required package:  ggplot2
## Loading required package:  methods

set.seed(998)
#use simple bootstrap resampling to split data into a series of train and test set
inTraining <- createDataPartition(trainDataTyped$is_delayed, p = .5, list = FALSE)

## Error in createDataPartition(trainDataTyped$is_delayed, p = 0.5, list = FALSE): y must have
at least 2 data points

training <- trainDataTyped[ inTraining,]

## Error in `[.data.frame`(trainDataTyped, inTraining, ):  object 'inTraining' not found

testing  <- trainDataTyped[-inTraining,]

## Error in `[.data.frame`(trainDataTyped, -inTraining, ):  object 'inTraining' not found
```

Create custom function to specifiy the type of resampling and a grid for tuning parameters[1]

```
cctrl4 <- trainControl(method = "cv", number = 3, classProbs = TRUE)
eGrid <- expand.grid(.alpha = seq(.05, 1, length = 15), .lambda = c((1:5)/10))
```

Save R environment

```
save.image(file="prepared_data.Rdata")
```

[1] both taken from `https://github.com/topepo/caret/blob/master/RegressionTests/Code/glmnet.R`