---

title: 'Pricing Seminar 2017 - Workshop B1: Approaches to building models in R'

output:

 html\_document: default

 html\_notebook: default

---

[Pricing Seminar 2017 Programme](https://www.actuaries.org.uk/learn-develop/attend-event/pricing-seminar-2017)<br>

Author: Simon Tam<br>

Date: 6th June 2017<br>

Abbreviated code from the workshop session. This code has not been independently validated or reviewed.

```{r Load packages and functions, include=FALSE}

#remove all objects

rm(list = ls())

#note that R 3.4.0 was used

#library(checkpoint)

#checkpoint("2017-06-06")

library(broom)

library(dplyr)

library(knitr)

library(govstyle) #devtools::install\_github('ukgovdatascience/govstyle')

library(Hmisc)

library(insuranceData)

library(purrr)

library(tibble)

#functions have been removed as they were included for demonstration only

```

```{r 1. Acquire data, include = FALSE}

# import the data and move into a Tibble

library(insuranceData)

data("SingaporeAuto")

df <- as\_data\_frame(SingaporeAuto)

rm(SingaporeAuto)

```

```{r 2. Process data, include = FALSE}

# SingaporeAuto Data Descriptions

# http://instruction.bus.wisc.edu/jfrees/jfreesbooks/Regression%20Modeling/BookWebDec2010/DataDescriptions.pdf

#

# SexInsured Gender of insured, including male (M), female(F) and unspecified (U)

# Female =1 if female, =0 otherwise

# VehicleType The type of vehicle being insured, such as automobile (A), truck (T), and motorcycle (M)

# PC =1 if private vehicle, =0 otherwise

# Clm Count Number of claims during the year

# Exp weights Exposure weight or the fraction of the year that the policy is in effect

# LNWEIGHT Logarithm of exposure weight

# NCD No Claims Discount. This is based ont he previous accident record of the policyholder.

# The higher the discount, the better is the prior accident record.

# AgeCat The age of the policyholder, in years grouped into seven categories.

# 0-6 indicate age groups 21 and younger, 22-25, 26-35, 36-45, 46-55, 56-65, 66 and over, respectively

# VAgeCat The age of the vehicle, in years, grouped into seven categories.

# 0-6 indicate groups 0, 1, 2, 3-5, 6-10, 11-15, 16 and older, respectively

# AutoAge0 =1 if private vehicle and VAgeCat = 0, =0 otherwise

# AutoAge1 =1 if private vehicle and VAgeCat = 1, =0 otherwise

# AutoAge2 =1 if private vehicle and VAgeCat = 2, =0 otherwise

# AutoAge =1 if Private vehicle and VAgeCat = 0, 1 or 2, =0 otherwise

# VAgecat1 VAgeCat with categories 0, 1, and 2 combined

# process and clean the data

df\_process <- df %>%

 transmute(SexInsured=factor(SexInsured),

 Female=factor(Female),

 Private=factor(PC),

 NCDCat=factor(NCD),

 AgeCat=factor(AgeCat),

 VAgeCat=factor(VAgeCat),

 Exp\_weights,

 Clm\_Count,

 Frequency = Clm\_Count/Exp\_weights

 )

```

## 3. Explore and visualise

### a) Summary of the variables in the dataset

```{r 3a) Summary of the variables in the dataset, echo=TRUE}

summary(df\_process)

```

### b) Graphs for exposures and claims

```{r 3b) Graphs for exposures and claims, echo=FALSE}

# Examples for the govstyle theme

# https://github.com/ukgovdatascience/govstyle/blob/master/vignettes/absence\_statistics.md

#Define the base chart parameters

g <- ggplot(data=df\_process)+scale\_y\_continuous(labels = scales::percent)

#Exposure histogram

g+geom\_histogram(aes\_string(x="Exp\_weights",y="..count../sum(..count..)"),binwidth=.005,fill=unname(gov\_cols)[1]) +ggtitle("Exposure weights")+labs(x=NULL, y="% records")

#Claims bar chart

geom\_claims <- g+geom\_bar(aes\_string(x="Clm\_Count",y="..prop..",weight="Exp\_weights/sum(Exp\_weights)"),fill=unname(gov\_cols)[1])+ggtitle("Claim counts")+labs(x=NULL, y="% exposure")

geom\_claims

#Facet the claims bar chart

geom\_claims+facet\_grid(.~NCDCat)+ggtitle("Claim counts by NCDCat")

```

## 4. Build models

### a) Split training and validation

```{r 4a) Split training and validation, include=FALSE}

df\_sample\_fraction <- 0.80

df\_training\_rows <- sample(1:nrow(df\_process), size = nrow(df\_process) \* df\_sample\_fraction, replace = FALSE)

df\_train <- df\_process [df\_training\_rows,]

df\_validation <- df\_process [-df\_training\_rows,]

```

### b) Fit the intercept model

```{r 4b) Fit the intercept model, echo=TRUE}

model\_intercept <- glm(Clm\_Count~1, family=poisson(link=log), data=df\_train, offset=log(Exp\_weights), y=FALSE, model=FALSE)

tidy(model\_intercept)

paste("Check average model frequency",exp(model\_intercept$coefficients[["(Intercept)"]]),"vs data average frequency",sum(df\_train$Clm\_Count)/sum(df\_train$Exp\_weights))

glance(model\_intercept)

head(augment(model\_intercept))

```

### c) Fit a model with all factors

```{r 4c) Fit a model with all factors, echo=TRUE}

model\_full <- glm(Clm\_Count~. - Frequency - Exp\_weights, family=poisson(link=log), data=df\_train, offset=log(Exp\_weights), y=FALSE, model=FALSE)

tidy(model\_full)

glance(model\_full)

```

### d) Use stepwise regression to add and remove factors

```{r d) Use stepwise regression to add and remove factors, echo=TRUE}

model\_stepwise <- step(model\_intercept, scope=list(lower=model\_intercept, upper=model\_full), direction = "both")

tidy(model\_stepwise)

glance(model\_stepwise)

```

## 5. Validate and score

### a) Declare model objects into a tibble and calculate validation statistics and objects

```{r 5a) Declare model objects into a list and calculate validation statistics and objects, echo=FALSE}

#Declare model objects into a tibble

model\_table <- tibble(description = c("Intercept", "Full", "Stepwise"),

 model = list(model\_intercept, model\_full, model\_stepwise))

#Calculate validation statistics and objects

model\_table <- model\_table %>%

 mutate(Deviance\_train = model %>% map(glance) %>% map\_dbl("deviance"),

 AIC = model %>% map(glance) %>% map\_dbl("AIC"))

model\_table

```