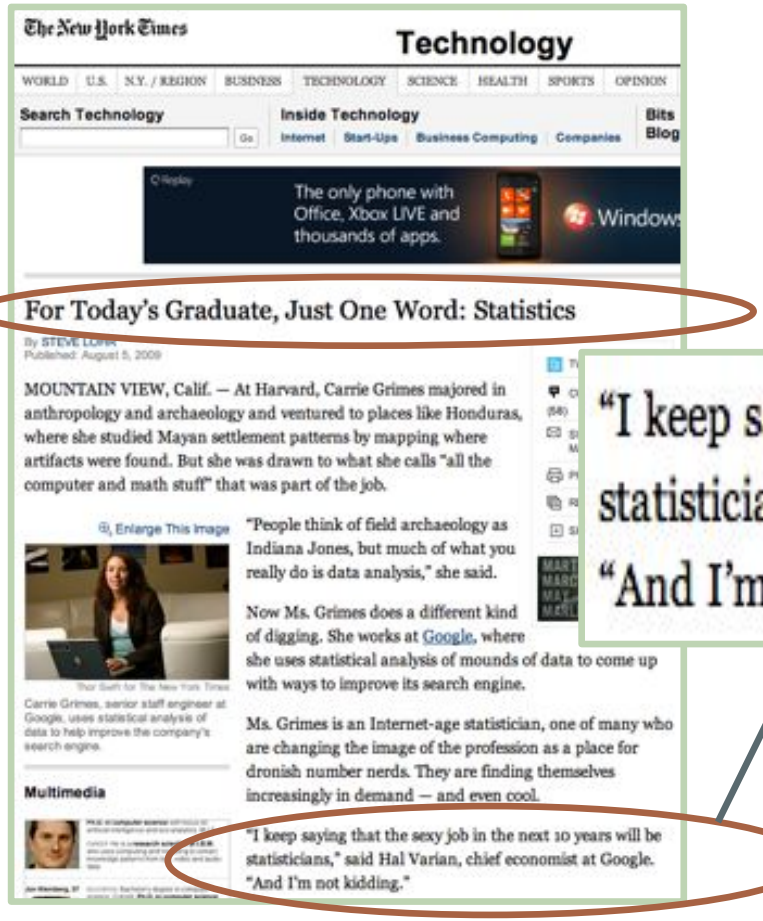
Predictive Modelling in insurance

you ready?

Xavier Conort
SAS Talk – 16 June 2011

Statistics is cool!



The New York Times

Technology

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION

Search Technology

Inside Technology

Internet | Start-Ups | Business Computing | Companies

Bits Blog

For Today's Graduate, Just One Word: Statistics

By STEVE LOHR

Published: August 5, 2009

MOUNTAIN VIEW, Calif. — At Harvard, Carrie Grimes majored in anthropology and archaeology and ventured to places like Honduras, where she studied Mayan settlement patterns by mapping where artifacts were found. But she was drawn to what she calls "all the computer and math stuff" that was part of the job.

Enlarge This Image

Thor Swift for The New York Times

Carrie Grimes, senior staff engineer at Google, uses statistical analysis of data to help improve the company's search engine.

"People think of field archaeology as Indiana Jones, but much of what you really do is data analysis," she said.

Now Ms. Grimes does a different kind of digging. She works at Google, where she uses statistical analysis of mounds of data to come up with ways to improve its search engine.

Ms. Grimes is an Internet-age statistician, one of many who are changing the image of the profession as a place for dronish number nerds. They are finding themselves increasingly in demand — and even cool.

Multimedia

"I keep saying that the sexy job in the next 10 years will be statisticians," said Hal Varian, chief economist at Google. "And I'm not kidding."

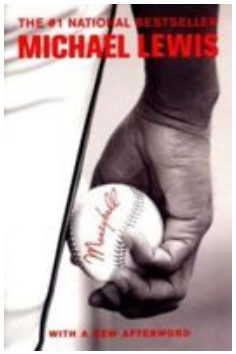
"I keep saying that the sexy job in the next 10 years will be statisticians," said Hal Varian, chief economist at Google. "And I'm not kidding."

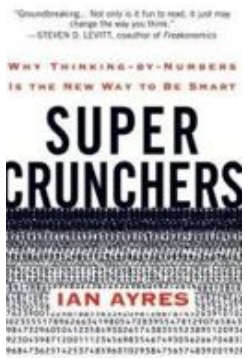
Predictive modelling is everywhere!

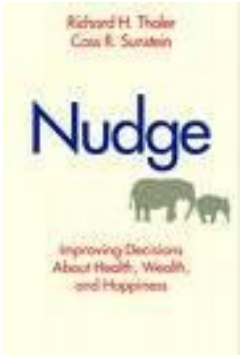
Most industries have taken advantage of increasing computing power and better data:

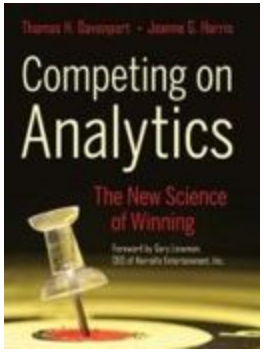
- Insurers use predictive models to underwrite risk
- Financial institutions determine credit score when you want a loan;
- The post office uses them to decipher your handwriting;
- Meteorologists to predict weather;
- Retailers to decide what to put on their shelves;
- Marketers to improve their products;
- They are even used by sports teams to hire players undervalued by the market

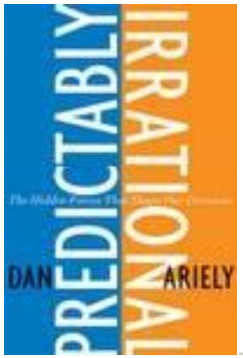
Ideas from other fields can be applied to insurance problems!











What is predictive modelling?

What makes the difference is the buy-in from critical users, the ability to ask the right questions and the ability to take advantage from the insight obtained from data

Building models is only a one phase of a predictive modelling project

Analyse data (improved quality but can still be messy)

Build Model

Predict Future Behaviours and Outcomes

Actionable information

What are the potential benefits in insurance?

Supports better decision making that will result in improved profitability

New Premium Growth
(campaigns, pricing to attract good risks, higher conversion)

Reduced Loss Ratio
(more accurate pricing, non profitable segments detected, reduction of fraud)

Improved Retention
(better customer service or discounts for profitable and more sensitive segments, enhanced agents producitivity)

Increased Underwriting Efficiency
(better focus on what really matters)

Improved Capital management
(quantify risk to retain, avoid or reduce risk optimally)

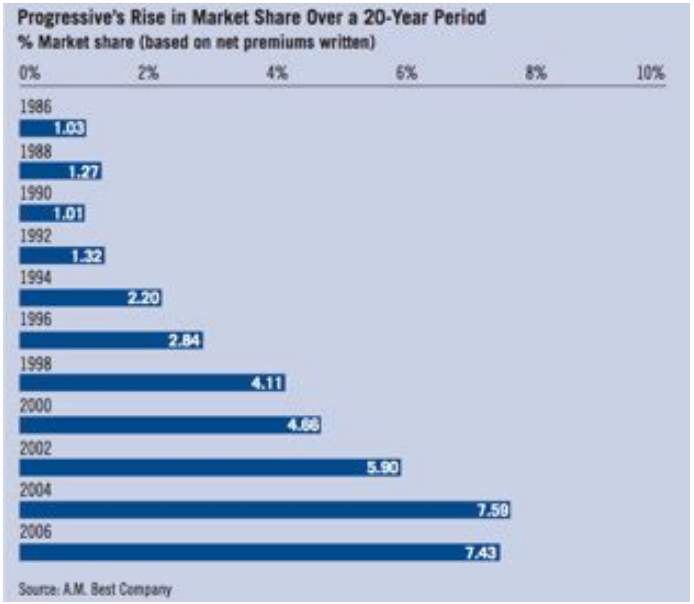
What do companies think about Predictive modelling?

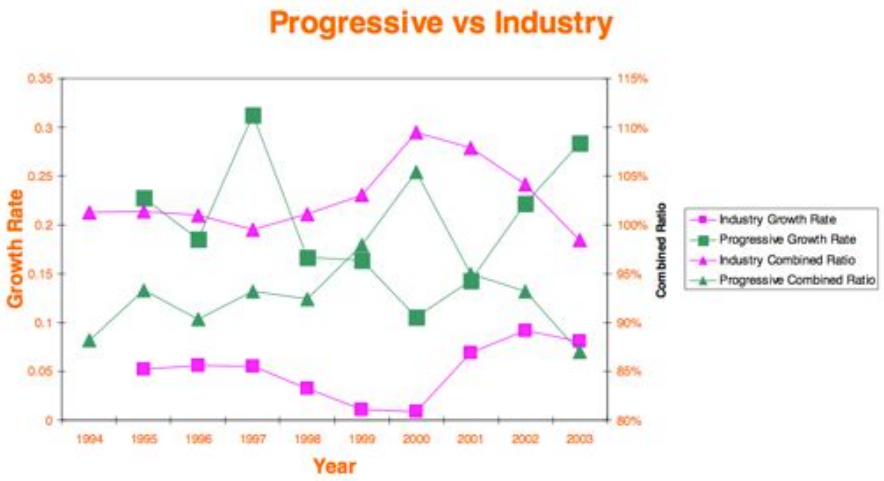
- Of the **43 U.S. GI companies** queried by Towers Watson in 2010,
  - 88% said the use of predictive modeling **enhanced rate accuracy**
  - 76% said they realized an **improvement in loss ratio**
  - 68% said that it **improved profitability**
  - 42% said it has furthered the expansion of their company's underwriting appetite
  - 39% indicated it helped increase market share

One case study
PROGRESSIVE in US

Progressive is a well-known pioneer of the application of predictive modelling in Insurance.

- In 1995, Progressive started to implement a "right price for every risk" approach
- Progressive grew rapidly, moving up 31 places over a 20-year period to become the third-largest auto insurer in the U.S. In the meantime, it got impressive Combined Ratio vs the industry.





Methods used in predictive modelling

- We will discuss mainly today on Generalized Linear Models (**GLMs**) which are the key tool for predictive modelling in the General Insurance industry
- In function of time, we may cover the Bootstrap of Over-dispersed Poisson (**ODP**) model (in appendix) which is one of the widely used methods for stochastic reserving
- We will mention the potential use of other techniques
  - classification and regression trees (CART), clustering, survival modelling, Generalized estimating equations (GEEs), Generalized Linear Mixed Models (GLMMs), Generalized Additive Models for Location Scale and Shape (GAMLSS) and Generalized Non Linear Models (GNM).
  - We will omit Neural Network and Random Forest which are powerful predictive tools but are "black box" and then have no explanatory power.
- **See the application of GLMs using R**, a freely available statistical software.

What is  ?

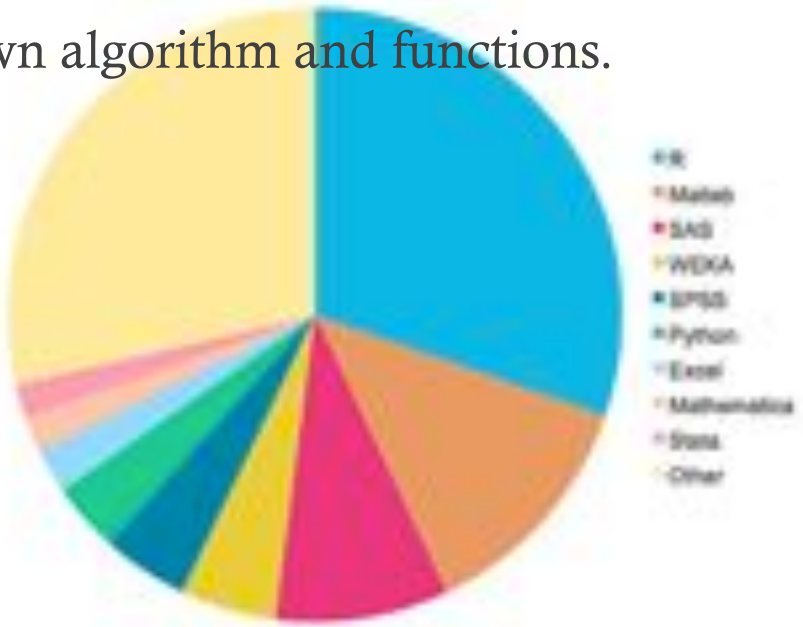
- R is one powerful way to get started with predictive modelling and benefit from a large community of R users (including academics, professional actuaries and actuarial associations)
  - The Actuarial Toolkit Working Party of UK Institute and Faculty of Actuaries suggested in 2006 that *"R, being free, could be the statistical language in which we exchange ideas and present concepts from actuarial papers, in a way that they can be used by others without rewriting code (and making the same mistakes again)".*
- You can download it for **free** @ www.r-project.org/
- It is the most common statistical package used in **universities** and more and more students in Actuarial Science are trained on R
- It is gaining exponential popularity in a **wide variety of industries**, including insurance, pharmaceuticals, finance, telecom, websites and oil and gas. Google, Merck, Shell, Bell, AT&T, Intercontinental, Oxford and Stanford are among R's benefactors and donors.
- It has attracted the attention of key **actuarial institutions** in Europe and North America who have already run and promoted courses on R.

What is  **key drawback**?

- Cannot handle large data.
  - This can be overcome with a commercial version of R, created by Revolution Analytics, owned by Norman Nie, the ex-founder of SPSS (he sold SPSS to IBM for $1.2 Billion…)
- Not so user-friendly (need to code)
  - For actuaries used to Visual Basic, it is not a problem
  - Some see it as an advantage, it offers more flexibility and R script is easier to share.
- **Other powerful tools are available as working tools**. Some are more user-friendly and handle larger data.

Why is R worth **learning**?

- **By learning R, you can learn a lot about other things**. It is
  - Very extensive: thousands of packages are available. Some of these packages represent **cutting-edge statistical research** as a lot of statistical research is first implemented in R.
  - Very flexible: you can write your own algorithm and functions.
- According to Kaggle (a leading platform for data mining competitions), about a third of all competitors report using R.
- **50% of data mining competition winners used R!**



Why should I use multivariate techniques and not Excel?

According to CAS "Basic Ratemaking" paper, multivariate methods

- **Consider all risk factors simultaneously and automatically adjust for exposure correlations**
  - Univariate approaches: results are distorted by distributional biases
  - Minimum bias methods: the iterative calculations are considered computationally inefficient
- **Attempt to remove noise** (unsystematic effects) in the data and **capture only the signal** (specification of how the expectation varies with the risk factors). This is not the case with univariate methods, which include both signal and noise in the results.
- **Produce model diagnostics** about the certainty of results and the appropriateness of the model fitted.
- **Can be refined to incorporate interaction**, or interdependency, between two or more risk factors (Interactions occur when the effect of one variable varies according to the levels of another)

Actuarial Toolkit Working Party

- In 2006, **the Actuarial Toolkit Working Party** of UK Institute and Faculty of Actuaries **likened the Microsoft Office suite for actuaries to a Swiss army knife for a dentist.**
- It can do most of the job but you would rather choose a dentist with a better tool.
  - "An actuarial tookit" by Trevor Maynard and al: http://toolkit.pbworks.com/f/Maynard.pdf
- The paper was also the first to introduce R to the actuarial community

Why GLMs and not the simple regression? more flexible!

$$Y = h(X\beta + \xi) + \varepsilon$$

Response to be modeled

Link function ($g=h^{-1}$) g=identity,log, logit and a few others *Vs idendity in the case of a simple regression*

Model structure = Linear Combination of Factors + Offset

Error structure = exponential family *Vs normal error in the case of a simple regression*

- Model structure: Generalized Linear Models (GLMs) can accommodate discrete variables (also known as categorical factors) and continuous variables (also known as variates).
- Link function (g=h-1) chosen based on how the variables relate to one another to produce the best signal
  - Log: variables relate multiplicatively / Identity: variables relate additively / Logit for response values between 0 and 1
- The error reflects the variability of the underlying process and can be any distribution within the exponential family, for example
  - Binomial is consistent with Lapse / Poisson consistent with claim count modeling / Gamma or Inverse Gaussian consistent with severity modeling / Tweedie consistent with pure premium modeling

A **simple example**

2 objectives: Show that

- the **univariate approach leads to distorted results** and it is more efficient (and easier) to fit a GLM than trying to control the effect of other factors manually
- **Don't be intimidated. Diagnostics tools are available to guide you!**
  - To choose the error structure and link function
  - To select the best model structure
  - To evaluate the certainty of your model

Let's first cheat and have a look at the truth

In this example, **we will work on simulated data** where we know the true underlying model for the response y (can be claims severity in this example)

- y follows a **gamma** distribution with shape =2
- the **mean parameter of y is function of 2 variables** Fac (categorical variable) and x (a continuous variable)
  - E(y)=exp(5+0.2*x) if Fac="A"
  - E(y)=exp(4.5+0.2*x) if Fac="B"

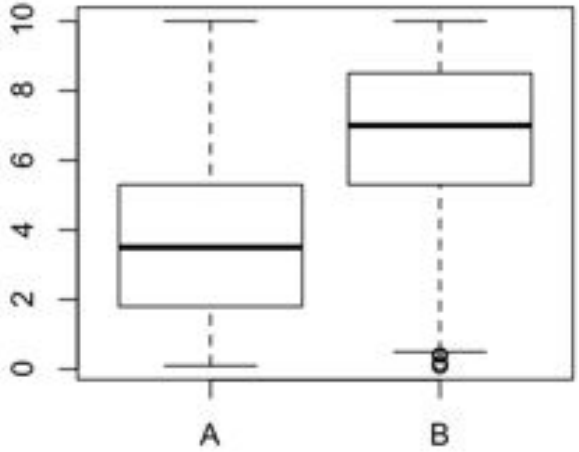
We fix the 2 predictors **Fac** and **x** so that they are strongly **correlated**

In the following, we will pretend that we are not aware of it and check if the GLMs outputs are consistent with the truth

```
N<-1000
set.seed(5)
x0 <- seq(0.1,10,by=.1)
x <- sample(x0,N,replace=T)
temp<-runif(N)
Fac<-as.factor(ifelse(x>5 & temp<0.7,"B",ifelse(temp<0.2,"B","A")))
mu<-exp(5+0.2*x-0.5*(Fac=="B"))
shape<-2
y<-rgamma(N,shape=shape,scale=mu/shape)
```

Univariate approach and distorted results

```
boxplot(x~Fac)
mean(y[Fac=="B"])/mean(y
[Fac=="A"])
exp(-0.5)
regGlog <- glm(y~x+Fac,
family=Gamma(link="log"))
summary(regGlog)
```



We can see here that for Fac="B" x tend to have stronger values. This will produce bias in the univariate analyis

```
> mean(y[Fac=="B"])/mean(y[Fac=="A"])
[1] 0.9658348
> exp(-0.5)
[1] 0.6065307
> regGlog <- glm(y~x+Fac,family=Gamma(link="log"))
> summary(regGlog)

Call:
glm(formula = y ~ x + Fac, family = Gamma(link = "log"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0323  -0.6386  -0.1902   0.2988   2.2673

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.045506   0.045695  110.42   <2e-16 ***
x            0.204437   0.008732   23.41   <2e-16 ***
FacB        -0.546139   0.051374  -10.63   <2e-16 ***
```

To compare with "true model"

- E(y)=exp(5+0.2*x) if Fac="A"
- E(y)=exp(4.5+0.2*x) if Fac="B"

What is the right model???

```
regNId <- glm(y~x+Fac,family=gaussian(link="identity"))
regGId <- glm(y~x+Fac,family=Gamma(link="identity"))
regNlog <- glm(y~x+Fac,family=gaussian(link="log"))
regGlog <- glm(y~x+Fac,family=Gamma(link="log"))
```

1. GLM with a normal error and the link “identity” (equivalent to the classical regression)?
2. GLM with a gamma error and the link “identity”?
3. GLM with a normal error and the link “log”?
4. GLM with a gamma error and the link “log”?

Accepted **standards**

Use accepted standards as a starting point

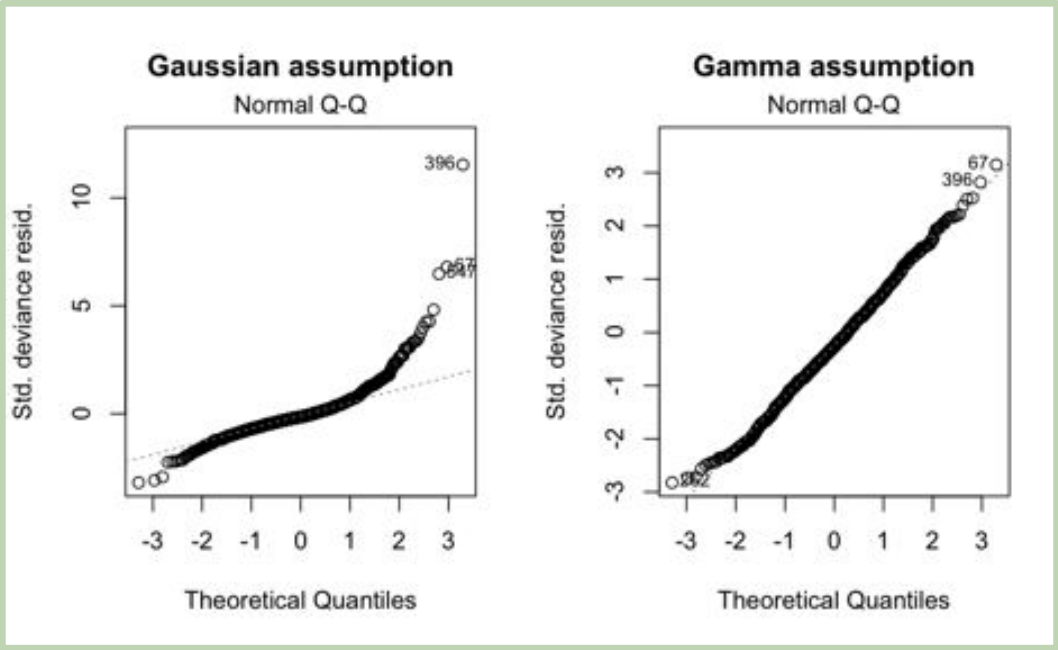
Observed response	Link function	Accepted standards	Comments
Claim count	log	Poisson	Don't forget to include the log of exposure as an offset. Negative binomial to account for heterogeneity is also common
Claim Severity	log	Gamma / Inverse Gaussian	Some recommend the Inverse link for the Inverse Gaussian. This is mostly motivated because it allows analytical calculation. Since you are using a statistical software, the log function makes thing easier and not less acceptable
Risk Premium	log	Tweedie	To model separately claim count and claim severity is more common, gives more insight but requires the 2 process to be independent
Retention / Conversion Rate / Fraud / Large claims	logit	Binomial	Other techniques are also commonly used such as classification trees or survival models

Test error structure

- **Adjust residuals**
  - 2 types of residuals are commonly used (deviance and Pearson residuals). Generally the deviance residuals are closer to being normally distributed than Pearson residuals. For that reason, **deviance residuals are generally preferred in connection with model diagnostics**
- **Test normality of residuals**
- **Observe scatter plots of residuals against fitted values**. It can give an indication of the appropriateness of the error function which has been assumed
  - Elliptical pattern is ideal
  - Fanning out suggests power of variance function is too low

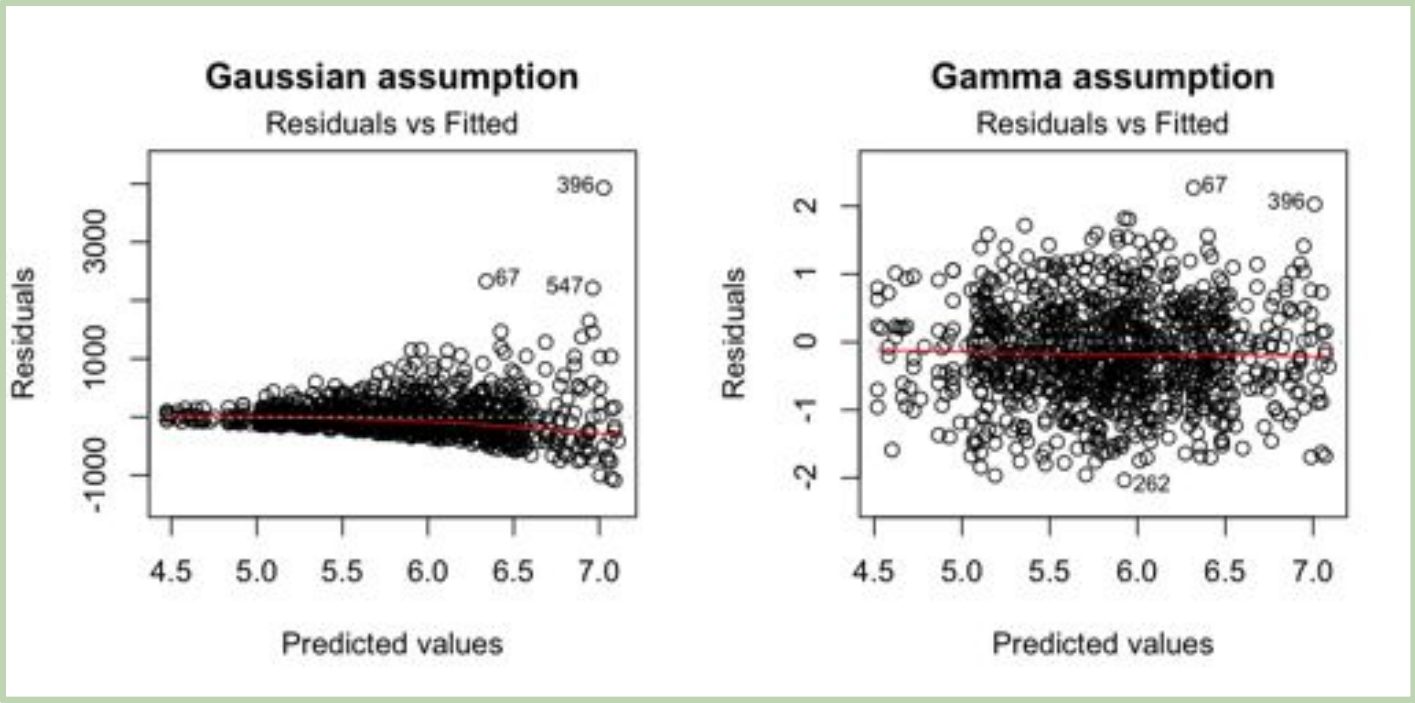
Test normality of deviance residuals to test error structure

```
par(mfrow=c(1,2))
plot(regNlog,2)
title("Gaussian assumption")
plot(regGlog,2)
title("Gamma assumption")
```



Scatterplot of deviance residuals to test error structure

```
par(mfrow=c(1,2))
plot(regNlog,1)
title("Gaussian assumption")
plot(regGlog,1)
title("Gamma assumption")
```



Compare models with **AIC**

- **Akaike information criterion (AIC)** provides a means for comparison among models
  - AIC = −2 log(Likelihood) + 2 · p attempts to optimize two opposing goals
    - make the fit as close as possible to the data: the likelihood should be large,
    - make the model simple: use only a small number p of parameters.
  - **Models with lower AIC can be argued to be "better" than those with higher AIC**
- Another criterion is sometimes used: the Schwarz Bayesian information Criterion (SBC or BIC) where the penalty for each degree of freedom is log n (instead of 2).
- Some consider AIC too generous in model selection, SBC too restrictive and then advice to work with a value of the penalty in the range 2.5 - 3.
- Another approach to avoid over-fitting is to split the data into training, validation and test data sets

```
> AIC(regNId,regGId,regNlog,regGlog)
         df      AIC
regNId    4 14572.81
regGId    4 13527.23
regNlog   4 14522.42
regGlog   4 13458.37
```

Note: Ripley advises to restrict the use of AIC to nested models (models with different sets of predictors, the simpler model being a special case of the more complicated model) and that AIC should not be used to compare models with different link functions and error structures. However, Burnham and Anderson (2002) consider Ripley's advice unfounded and state that only the likelihood-ratio test has such restriction…

How to ensure that my model is the "best" one?

- Ensure you **used all information available**. Consider external information if available. Think about new ones (car usage is one example). And ensure it is clean…
- **Compare** your model with
  - models with more factors and less factors
  - models with interaction
  - models with a more complex non-linear relationship
- **Cross check** your findings with alternative techniques such as Classification and Regression Trees (CART)
- **Use standard errors** of estimates and prediction to have an indication of the certainty of your results

Add or drop factors?

Null hypothesis: models with and without a factor have the same statistical significance

```
> Fac2 <- as.factor(sample(1:5,N,replace=T))
> addterm(regGlog,scope=~.+Fac2,test="Chisq")
Single term additions

Model:
y ~ x + Fac
       Df Deviance   AIC scaled dev. Pr(Chi)
<none>      540.56 13458
Fac2    4   538.88 13463      3.2322  0.5197
> dropterm(regGlog,test="Chisq")
Single term deletions

Model:
y ~ x + Fac
       Df Deviance   AIC scaled dev.   Pr(Chi)
<none>      540.56 13458
x       1   826.87 14008      551.15 < 2.2e-16 ***
Fac     1   597.65 13566      109.89 < 2.2e-16 ***
```

How to read the R output

① to include Fac2 (which is totally unrelated to y) in the model doesn't add any value. There is 52% of chance to wrongly reject the Null hypothesis.

② to keep x and Fac as predictor is definitely a good choice! There is 0% of chance to wrongly reject the Null hypothesis.

③ Note that the criterion AIC will give you the same recommendations

Add interaction?

Interactions occur when the effect of one variable varies according to the levels of another. This is not automatically detected by GLMs.

We have to integrate them manually and use Automated Stepwise Regressions based on AIC criteria.

```
> stepAIC(regGlog,scope=~x*Fac*Fac2,direction="forward")
Start:  AIC=13458.37
y ~ x + Fac

        Df Deviance   AIC
<none>       540.56 13458
+ x:Fac  1   540.26 13460
+ Fac2   4   538.88 13463

Call:  glm(formula = y ~ x + Fac, family = Gamma(link = "log"))

Coefficients:
(Intercept)            x         FacB
     5.0455       0.2044      -0.5461

Degrees of Freedom: 999 Total (i.e. Null);  997 Residual
Null Deviance:        827.2
Residual Deviance: 540.6    AIC: 13460
```

Add **non linear** relationship?

The link-function offer a **limited number of non-linear models**

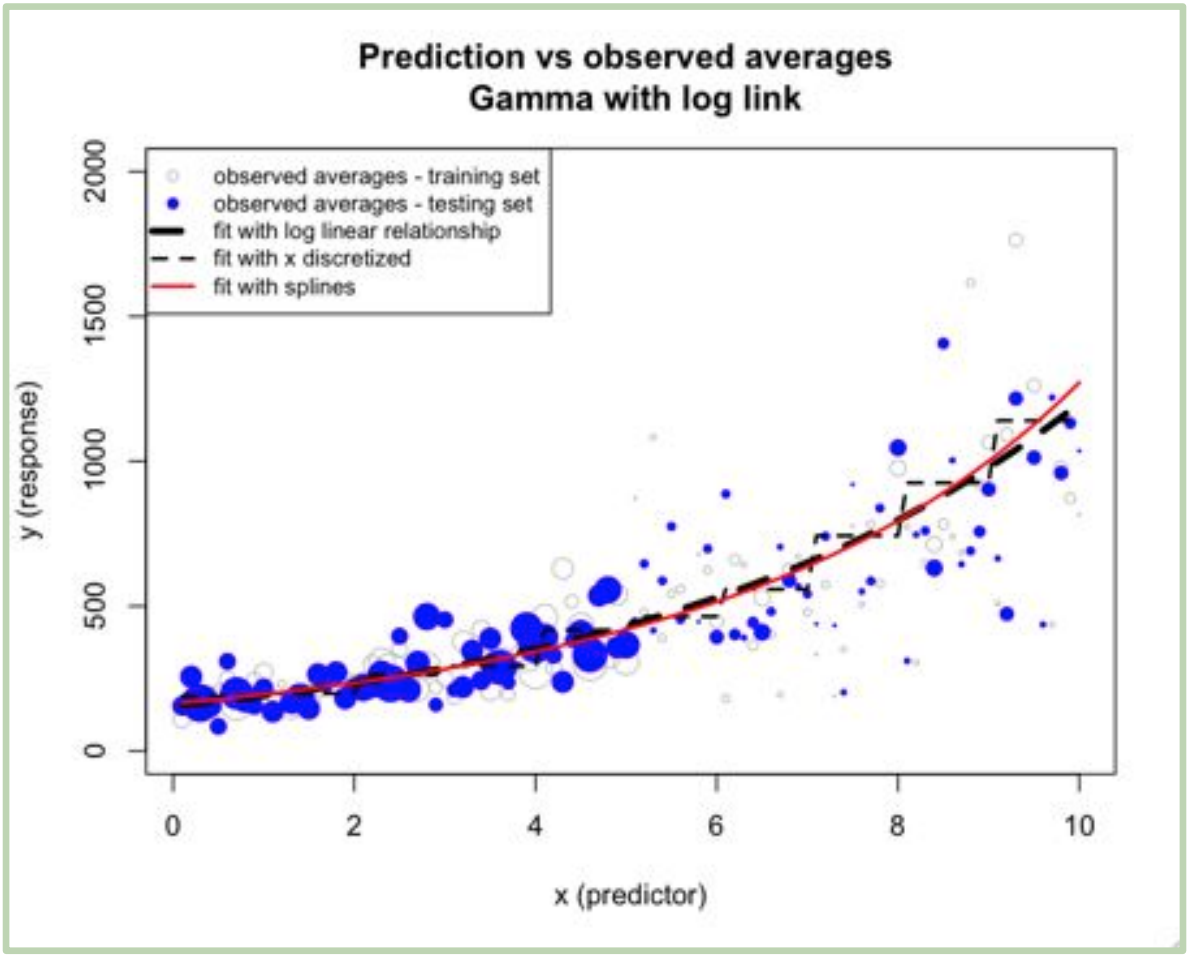
To overcome this, continuous variable (such as age) can :

- be discretized and grouped
- take the form of polynomials or splines

```
> x_group <- cut(x, breaks=0:10)
> regGlog2 <- glm(y~x_group+Fac,family=Gamma(link="log"))
> regGlog3 <- glm(y~x+I(x^2)+I(x^3)+Fac,family=Gamma(link="log"))
> library(splines)
> regGlog4 <- glm(y~bs(x)+Fac,family=Gamma(link="log"))
> AIC(regGlog,regGlog2,regGlog3,regGlog4)
          df      AIC
regGlog    4 13458.37
regGlog2  12 13464.98
regGlog3   6 13461.13
regGlog4   6 13461.13
```

A more advanced alternative is to use Generalized Non Linear Models (GNMs). In actuarial science, such models are used to model longevity.

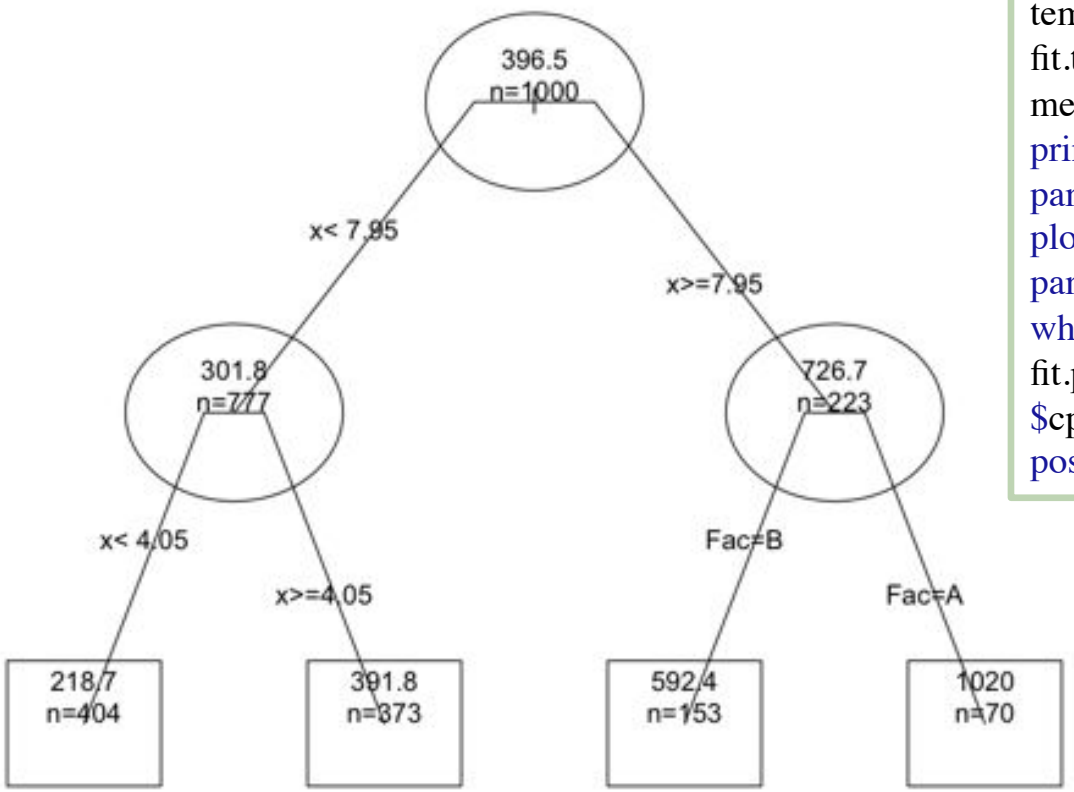
Plot prediction vs observed averages and check consistency over time or data sets



R script for the previous slide

```
par(mfrow=c(1,1))
x1<-sort(unique(x[Fac=="A"]))
plot(x1,tapply(y[Fac=="A"],x[Fac=="A"],mean),col="gray",cex=table(x[Fac=="A"])/5,pch=1,xlab="x
(predictor)",ylab="y (response)",ylim=c(0,2000))
set.seed(20110616)
y_test<-rgamma(N,shape=shape,scale=mu/shape)
points(x1,tapply(y_test[Fac=="A"],x[Fac=="A"],mean),col="blue",cex=table(x[Fac=="A"])/5,pch=19)
yp <- predict(regGlog,newdata=data.frame(x=x0,Fac="A"),  type="response")
lines(x0,yp,lty=2,lwd=4)
yp <- predict(regGlog2,newdata=data.frame(x=x0,x_group=cut(x0, breaks=0:10),Fac="A"),
type="response")
lines(x0,yp,lty=2,lwd=2)
yp <- predict(regGlog4,newdata=data.frame(x=x0,Fac="A"),  type="response")
lines(x0,yp,col="red",lwd=2)
legend("topleft",c("observed averages - training set","observed averages - testing set","fit with log
linear relationship","fit with x discretized","fit with splines"),col=c(8,4,1,1,2),lty=c(NA,NA,
2,2,1),lwd=c(NA,NA,4,2,2),pch=c(1,19,NA,NA,NA),cex=0.8)
title("Prediction vs observed averages \n Gamma with log link")
```

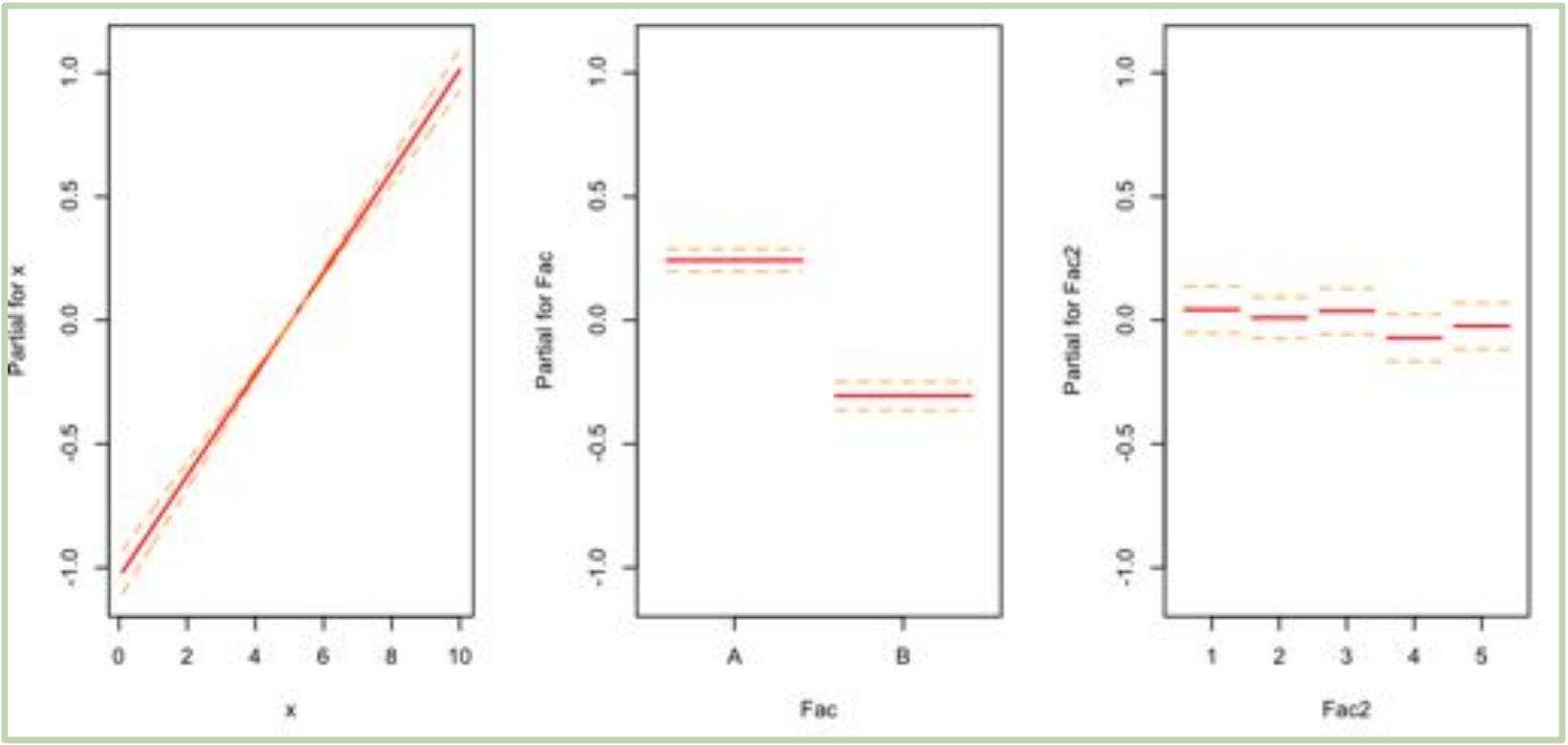
Use CART to **cross check findings**



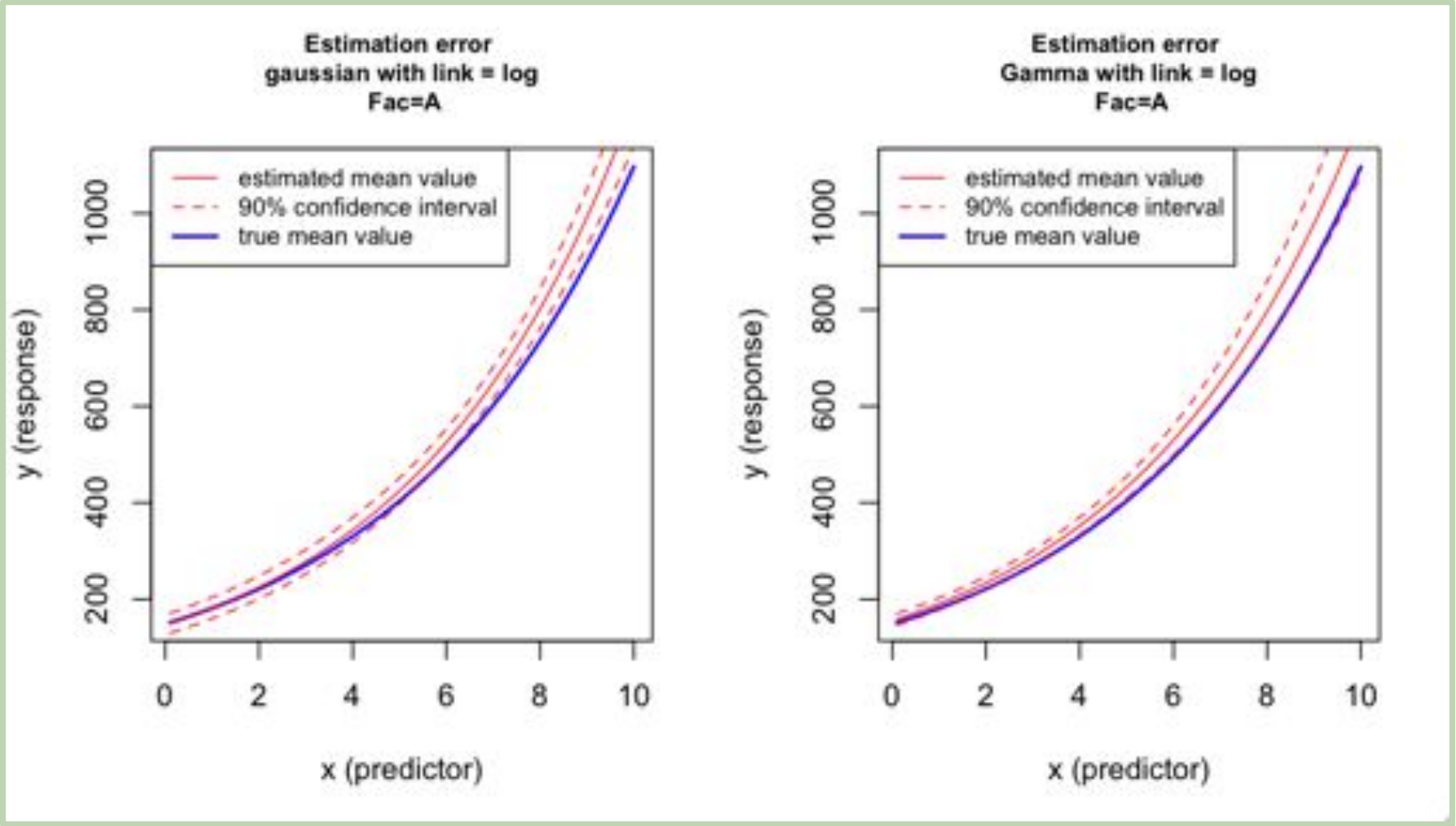

```
library(rpart)
temp<-rpart.control(xval=10,cp=0.0001)
fit.tree<-rpart(y~x + Fac + Fac2,
method='anova', control=temp)
printcp(fit.tree)
par(mfrow=c(1,1))
plotcp(fit.tree)
par(mfrow=c(1,1))
which.min(fit.tree$cptable[,"xerror"])
fit.prune_s1<- prune(fit.tree,cp= fit.tree
$cptable[4,"CP"])
post(fit.prune_s1,file="")
```

Plot **estimates with standard error.** Do they make sense? Statistically significant?

```
regGlog5 <- glm(y~x+Fac+Fac2,family=Gamma(link="log"))
par(mfrow=c(1,3))
termplot(regGlog5,se=TRUE)
```



Quantify the **estimation error** of your prediction



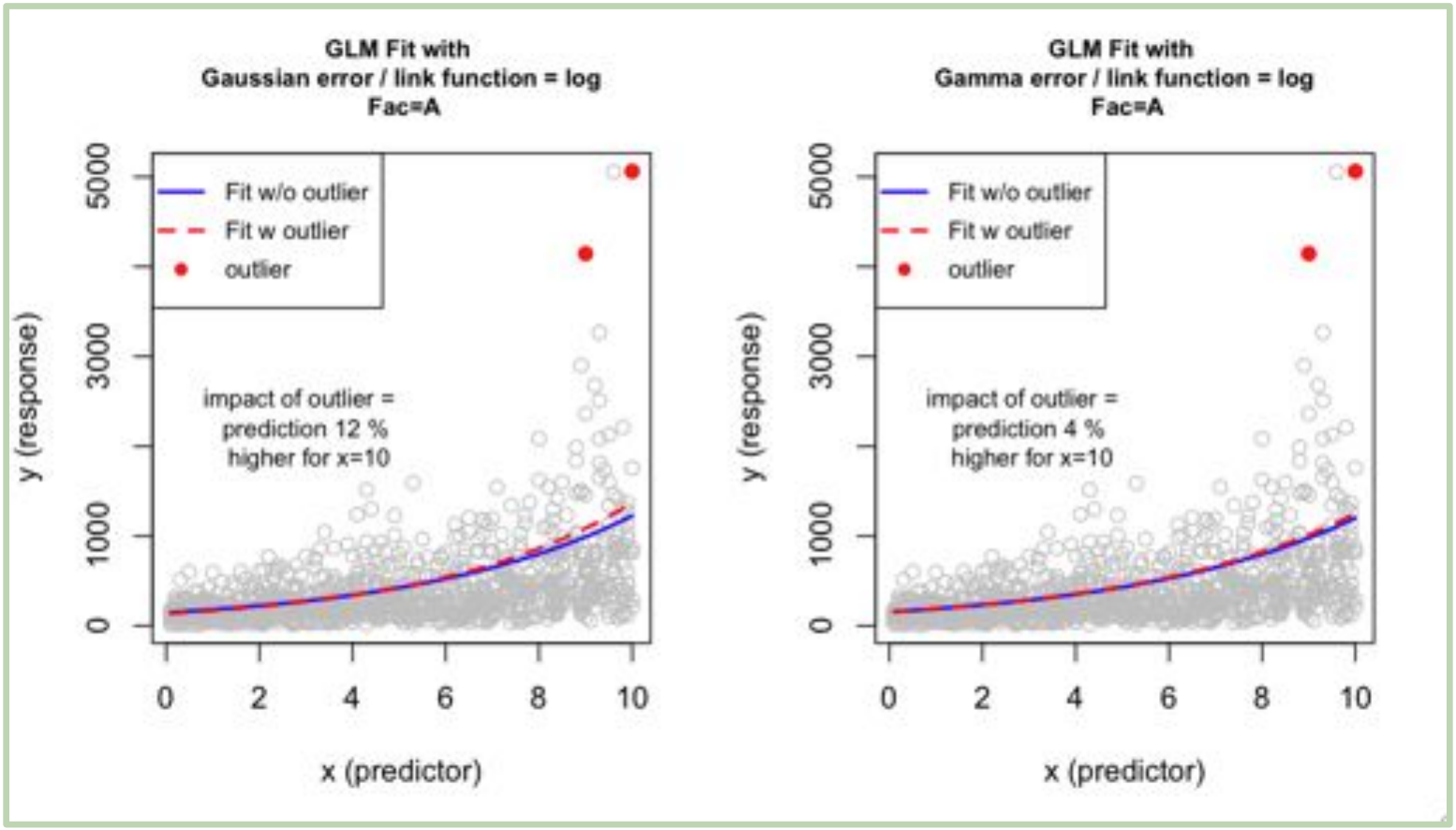
R script for the previous slide

```
func_1<-function(Fit) {
    yp <- predict(Fit,newdata=data.frame(x=x0,Fac="A"),se.fit = TRUE,
    type="response")
    plot(x0,exp(5+0.2*x0) ,lwd=2,col="blue",type="l",xlab="x (predictor)",
    ylab="y (response)")
    lines(x0,yp$fit,col="red")
    lines(x0,yp$fit+qnorm(0.95)*yp$se.fit,lty=2,col="red")
    lines(x0,yp$fit-qnorm(0.95)*yp$se.fit,lty=2,col="red")
    legend("topleft",c("estimated mean value","90% confidence
    interval","true mean value"),col=c(2,2,4),lty=c(1,2,1),lwd=c(1,1,2),
    cex=0.8)
    title(paste("Estimation error \n",Fit$family$family,"with link =",Fit
    $family$link,"\n Fac=A"),cex.main=0.8)
}

par(mfrow=c(1,2))

func_1(regNlog)
func_1(regGlog)
```

What happens if I choose the **wrong error structure**?

- **To pick the wrong one will not have a great impact on the estimation of BEs** (for the same group of predictors) most of the times…
- But **to pick the right one will**
  - **Produce more robust estimations** and
  - **Quantify the standard errors and estimation errors correctly**
- To avoid unrobust outputs, it is common to **cap extreme values or outliers** and work by type of loss with different patterns
- Some techniques such as Generalized Additive Models for Location, Scale and Shape (GAMLSS) allow to relax the exponential family (and reduce the risk of misspecification) but it is more advanced…
- When the independence assumption is violated (longitudinal or clustered data), Generalized estimating equations (GEEs) and Generalized Linear Mixed Models (GLMMs) are useful to correct the bias in the standard errors estimation. But it is also more advanced…
- Finally note that incorrect quantifications of standard errors may lead to wrong conclusions about the statistical significance of predictors and then to different decisions during the model selection process.

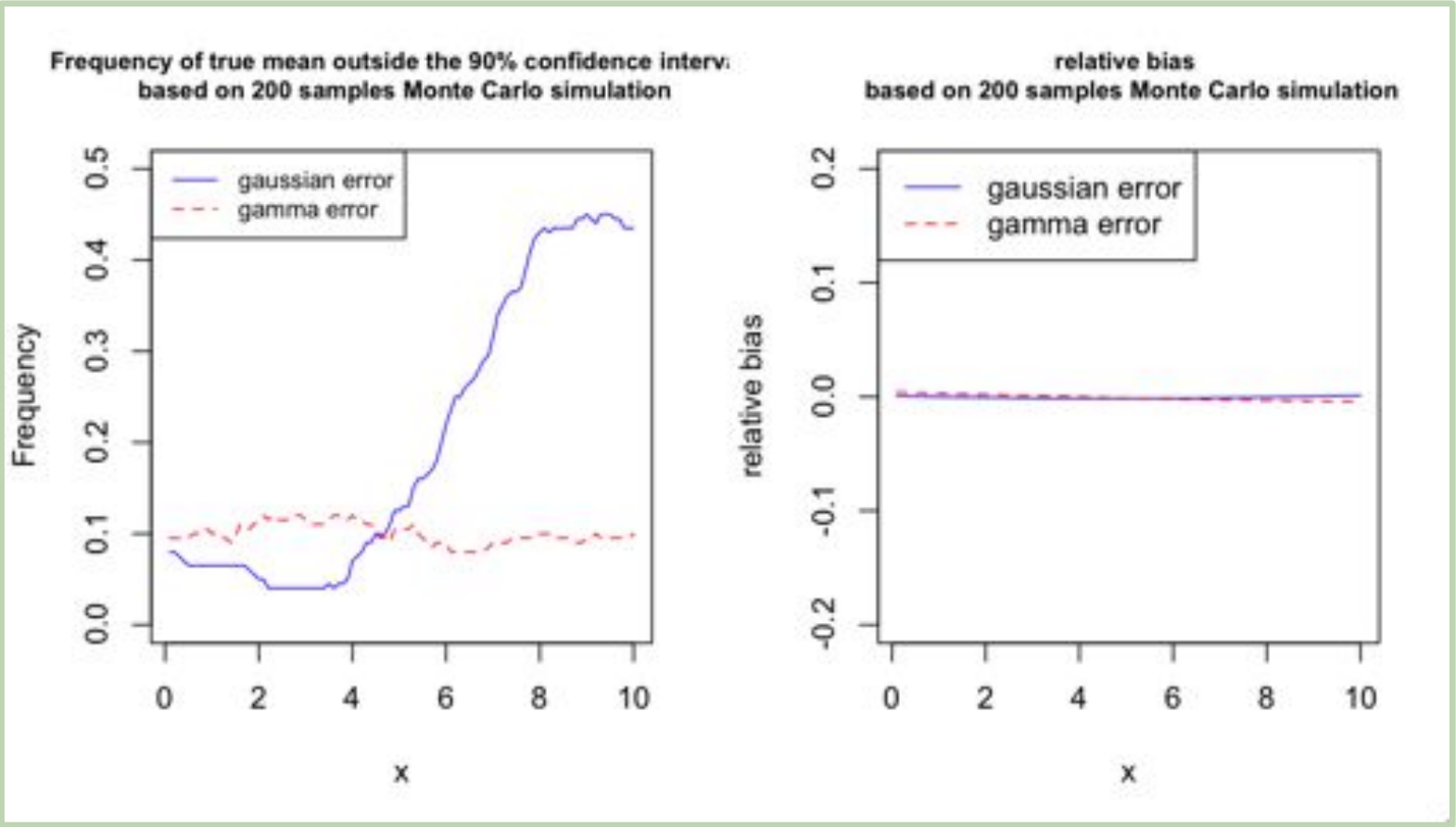
Selecting the right error structure gives **more robust results**



R script for the previous slide

```
outlier<-qgamma(0.999,shape=2,scale=exp(5+0.2*9)/2)
outlier2<-qgamma(0.999,shape=2,scale=exp(5+0.2*10)/2)
x2<-c(x,9,10) ; y2<-c(y,outlier,outlier2) ; Fac2<-as.factor(c(as.character(Fac),"A","A"))

regNlog2 <- glm(y2~x2+Fac2,family=gaussian(link="log"))
regGlog2 <- glm(y2~x2+Fac2,family=Gamma(link="log"))

func2<-function(Fit1,Fit2,err) {
  col<-c(rep(8,N),2,2)
  pch<-c(rep(1,N),19,19)
  plot(x2,y2,col=col,pch=pch,xlab="x (predictor)",ylab="y (response)")
  yp <- predict(Fit1,newdata=data.frame(x=x0,Fac="A"),  type="response")
  lines(x0,yp,col="blue",lwd=2)
  yp2 <- predict(Fit2,newdata=data.frame(x2=x0,Fac2="A"),  type="response")
  lines(x0,yp2,col="red",lty=2,lwd=2)
  legend("topleft",c("Fit w/o outlier","Fit w outlier","outlier"),col=c(4,2,2),lty=c(1,2,NA),lwd=c(2,2,1),pch=c(NA,NA,
19),cex=0.8)
  title(paste("GLM Fit with \n",err,"error / link function = log \n Fac=A"),cex.main=0.8)
  text(3,2200,paste("impact of outlier =  \n prediction",round(yp2[length(yp2)]/yp[length(yp)]*100-100,0),"% \n higher for
x=10"),cex=0.8)
}
par(mfrow=c(1,2))
func2(regNlog,regNlog2,"Gaussian")
func2(regGlog,regGlog2,"Gamma")
```

Selecting the right error structure gives a more **accurate quantification of the estimation error**



Working with the gamma error structure improved dramatically the accuracy of the quantificaiton of the estimation error. The gaussian error gives unbiased estimation of the mean, even if as we saw earlier, the results may be less robust in presence of large values.

R script for the previous slide

```
mu_<-exp(5+0.2*x0)

func_3<-function(i) {
 set.seed(i)
 y<-rgamma(N,shape=shape,scale=mu/shape)
 Fit <- glm(y~x+Fac,family=gaussian(link="log"))
 yp <- predict(Fit,newdata=data.frame(x=x0,Fac="A"), se.fit = TRUE,  type="response")
 Fit <- glm(y~x+Fac,family=Gamma(link="log"))
 yp2 <- predict(Fit,newdata=data.frame(x=x0,Fac="A"), se.fit = TRUE,  type="response")
 out<-out2<-rep(0,length(x0))
 out[mu_>yp$fit+qnorm(0.95)*yp$se.fit | mu_<yp$fit-qnorm(0.95)*yp$se.fit]<-1
 out2[mu_>yp2$fit+qnorm(0.95)*yp2$se.fit | mu_<yp2$fit-qnorm(0.95)*yp2$se.fit]<-1
 list(out,out2,yp$fit,yp2$fit)
}

K<-200
res<-func_3(1) ; out<-res[[1]] ; out2<-res[[2]] ; yp<-res[[3]] ; yp2<-res[[4]] ; se<-(mu_-res[[3]])^2 ; se2<-(mu_-res[[4]])^2
for (i in 2:K) {
 res<-func_3(i) ;  out<-out+res[[1]] ; out2<-out2+res[[2]] ; yp<-yp+res[[3]] ; yp2<-yp2+res[[4]]
}
out<-out/K ; out2<-out2/K ; yp<-yp/K ; yp2<-yp2/K

par(mfrow=c(1,2))

plot(x0,out,type="l",col="blue",ylim=c(0,0.5),xlab="x",ylab="Frequency",cex=0.8)
lines(x0,out2,col="red",lty=2)
legend("topleft",c("gaussian error","gamma error"),col=c("blue","red"),lty=c(1,2),cex=0.8)
title(paste("Frequency of true mean outside the 90% confidence interval \n based on",K,"samples Monte Carlo simulation"),cex.main=0.8)

plot(x0,yp/mu_-1,type="l",col="blue",ylim=c(-0.2,0.2),xlab="x",ylab="relative bias",cex=0.8)
lines(x0,yp2/mu_-1,col="red",lty=2)
legend("topleft",c("gaussian error","gamma error"),col=c("blue","red"),lty=c(1,2))
title(paste("relative bias \n based on",K,"samples Monte Carlo simulation"),cex.main=0.8)
```

Is Predictive modelling only for General Insurance?

- General Insurance is the first to have used predictive modelling and it is now a standard in mature markets
- **In the Life Insurance industry, interest is growing**. A few papers on predictive modelling in Life Insurance are available @
  - http://www.soa.org/research/research-projects/life-insurance/research-pred-mod-life-insurers.aspx
- Some examples, where predictive modelling can be useful to the Life Industry, include
  - Marketing, Underwriting, Lapse, Fraud, Longevity, Health, Disability and Reserving (IBNR) risks

Can GLMs do most of the job?

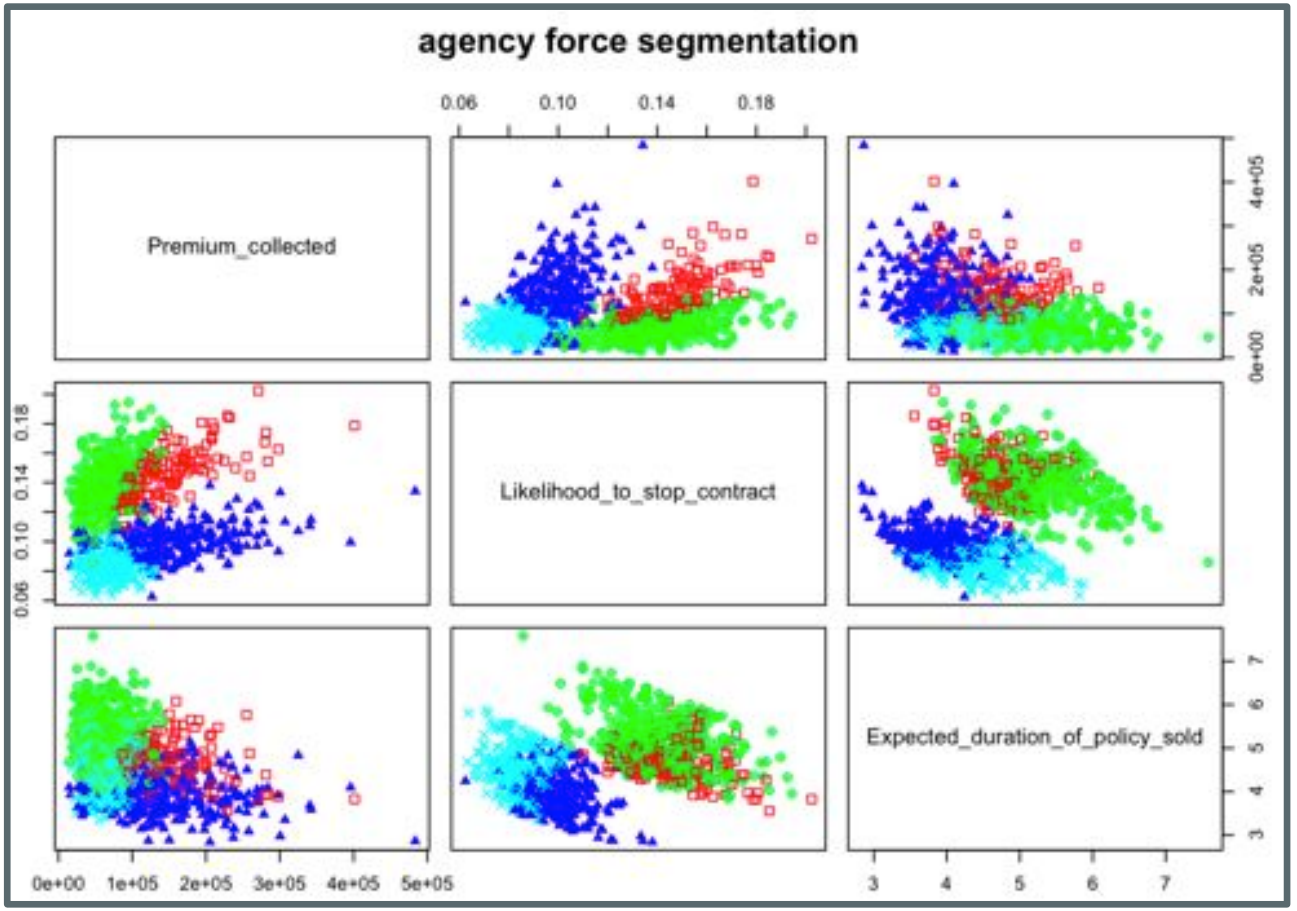
GLMs

Observed response	Accepted standards	More advanced error structures or techniques
Claim count	Poisson	Negative binomial
Claim Severity	Gamma / Inverse Gaussian	Lognormal, truncated, censored skewed distributions
Risk Premium	Tweedie	Zero-adjusted gamma, Zero-adjusted inverse gaussian
Retention / Conversion Rate / Fraud / Large claims	Binomial (+logit link = logistic regression)	Classification trees, Survival models
IBNR	GLM (Over Dispersed Poisson)+Bootstrapping	Bayesian techniques, Micro-level stochastic reserving
Claim duration, report lag	Survival models	
Longevity	Lee Carter (LC)	LC variants, Generalized Non Linear Models w. cohort effects
Segmentation	K-means	Other clustering techniques

R packages of interest to actuaries include

- MASS : many useful functions, data examples and negative binomial linear models
- rpart : Classification and Regression Trees
- survival : Survival analysis
- splines : B-splines and natural cubic splines
- ChainLadder : ChainLadder based Stochastic reserving models
- gamlss : Generalized Additive Models for Location, Scale and Shape
- lme4, gamlss.mx, hglm : packages for Generalized Linear Mixed Models
- gnm : packages for Generalized Non Linear Models
- demography: Lee Carter and its variants
- copula : commonly used copulas including elliptical (normal and t), Archimedean (Clayton, Gumbel, Frank, and Ali-Mikhail-Haq)
- POT, evir : functions related to the Extreme Value Theory
- R2WinBugs : Markov Chain Monte Carlo methods

Predictive Modelling is not only transforming actuarial science but also the way to do business



Questions?

Contact details

Xavier Conort

Email : xavier.conort@gear-analytics.com

Mobile : + 65 9339 8636

Drop me an email if you want a copy of the slides of this presentation.

Appendix

① How to bootstrap the ODP model to obtain a distribution of IBNR potential deviations using R?

② A few slides for new R users on how to

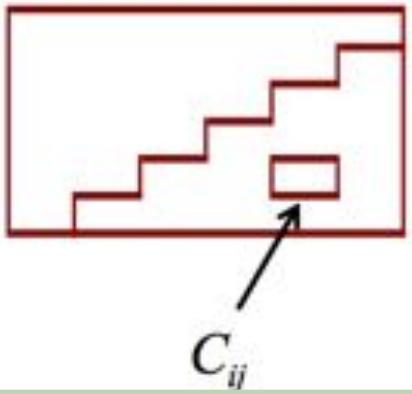
- Install R
- Install additional packages
- Execute commands

How to bootstrap the ODP model to obtain a distribution of IBNR potential deviations using R?

*The example follows closely the algorithm recommended by Peter England in "**Bootstrapping: Lessons Learnt in the Last 10 Years**" - Swiss Association of Actuaries - 10 September 2010 and previous papers from England & Verrall (1999, 2002, 2006)*

Over-Dispersed Poisson (ODP) Model

- C_{ij}=Incremental claims in origin year i and development year j
- $C_{ij} \sim ODP(\mu_{ij}, \phi_j)$
  - $E[C_{ij}] = \mu_{ij}$
  - $V[C_{ij}] = \phi_j \mu_{ij}$
- With $\log(\mu_{ij}) = c + a_i + b_j$



Note: The main justification of the model is that it gives the same forecasts as the chain ladder model

In the next slides, we will bootstrap the ODP model to obtain the distribution of potential deviation from the IBNR Best Estimate and follow closely the algorithm recommended by Peter England in "***Bootstrapping: Lessons Learnt in the Last 10 Years***" - Swiss Association of Actuaries - 10 September 2010 and previous papers from England & Verrall (1999, 2002, 2006)

What is Bootstrapping?

It is a method for producing sampling distributions for statistical quantities of interest by **generating pseudo samples**, which are obtained by randomly drawing, with replacement, from observed data

The simplest bootstrapping process is the following:

- Suppose we have a sample X and we require the distribution of a statistic μ

1. Draw a bootstrap sample $X_1^B = \{x_1^B, x_2^B, \ldots x_n^B\}_1$ from the observed data $X = \{x_1, x_2, \ldots x_n\}$,
2. Calculate the statistic of interest μ_1^B for the first bootstrap sample X_1^B

- By repeating steps 1 and 2 N times, we obtain a sample of unknown statistic $\mu^B = \{\mu_1^B, \mu_2^B, \ldots \mu_N^B\}$, calculated from N pseudo samples. When N > 1000, the empirical distribution constructed from μ^B can be taken as the approximation to the distribution for the statistic of interest μ

Let's consider a coin-flipping experiment

Let $X = x_1, x_2, \ldots, x_{100}$ be 100 observations from the experiment. $x_i = 1$ if the i th flip lands heads, and 0 otherwise

```
n<-100
(X<-sample(c(0,1),n,replace=T))
```

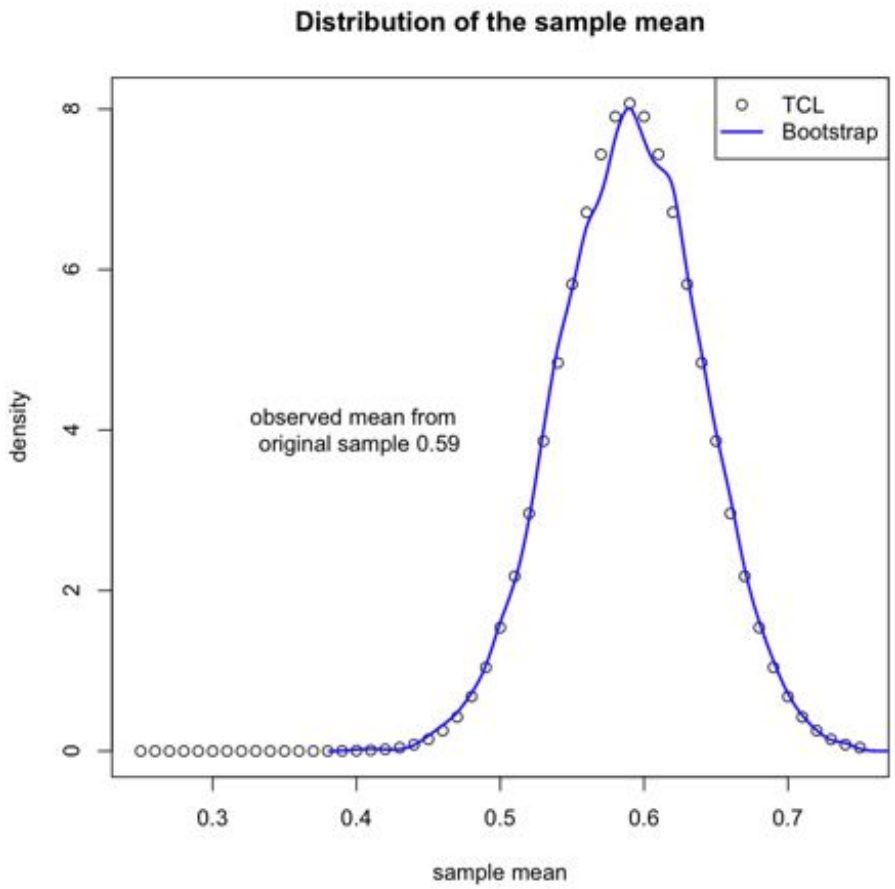
We could have estimated and plotted the distribution of the sample mean μ using the Central Limit Theorem

```
plot(seq(0.25,0.75,by=0.01),dnorm(seq(0.25,0.75,by=0.01),mean=mean(X),sd=sd(X)/n^0.5))
```

Instead, we will use the bootstrap approach to derive the distribution of μ

```
mub<-c()
for (i in 1:10000)  mub[i]=mean(sample(X,n,replace=T))
lines(density(mub),lwd=2,col="blue")
```

Result from the coin-flipping experiment



Bootstraping applied to ODP model

We will here **resample the ODP residuals instead of the original data**. This is necessary because the bootstrap algorithm requires that the response variables are independent and identically distributed.

1. Fit ODP and obtain fitted incremental values
2. Obtain adjusted and scaled residuals
3. Resample residuals
4. Obtain pseudo data from resampled residuals and fitted incremental values
5. Use chain ladder to re-fit model, and estimate future incremental payments
6. Repeat many times

   => ESTIMATION ERROR = standard deviation of forecasts obtained with pseudo data
7. Add random fluctuations to the forecasts in step 5

   => PREDICTION ERROR

Adjusted and scaled residuals and pseudo data with **constant scale parameter**

Adjusted unscaled residuals

$$adj.unscaled.res_{ij} = \frac{C_{ij} - \mu_{ij}}{\sqrt{\mu_{ij}}} \times \left(\frac{N}{N-p} \right)^{1/2}$$

(N: number of observations; p : number of parameters)

Constant scale parameter

$$\phi = \frac{\sum \left(adj.unscaled.res_{ij} \right)^2}{N}$$

Adjusted scaled residuals

$$adj.scaled.res_{ij} = adj.unscaled.res_{ij} \Big/ \phi$$

Obtain pseudo data

$$C_{ij}^{*} = resampled.adj.scaled.res_{ij} \sqrt{\phi \mu_{ij}} + \mu_{ij}$$

Adjusted and scaled residuals and pseudo data with **non-constant scale parameter**

Adjusted unscaled residuals

$$adj.unscaled.res_{ij} = \frac{C_{ij} - \mu_{ij}}{\sqrt{\mu_{ij}}} \times \left(\frac{N}{N-p} \right)^{1/2}$$

(N: number of observations; p : number of parameters)

Non-constant scale parameter

$$\phi_j = \frac{\sum_{i=1}^{n_j} \left(adj.unscaled.res_{ij} \right)^2}{n_j}$$

Adjusted scaled residuals

$$adj.scaled.res_{ij} = adj.unscaled.res_{ij} \Big/ \phi_j$$

Obtain pseudo data

$$C_{ij}^{*} = resampled.adj.scaled.res_{ij} \sqrt{\phi_j \mu_{ij}} + \mu_{ij}$$

Example

	1	2	3	4	5	6	7	8	9	10
1	357,848	766,940	610,542	482,940	527,326	574,398	146,342	139,950	227,229	67,948
2	352,118	884,021	933,894	1,183,289	445,745	320,996	527,804	266,172	425,046	
3	290,507	1,001,799	926,219	1,016,654	750,816	146,923	495,992	280,405		
4	310,608	1,108,250	776,189	1,562,400	272,482	352,053	206,286			
5	443,160	693,190	991,983	769,488	504,851	470,639				
6	396,132	937,085	847,498	805,037	705,960					
7	440,832	847,631	1,131,398	1,063,269						
8	359,480	1,061,648	1,443,370							
9	376,686	986,608								
10	344,014									

Incremental values triangle taken from "***Bootstrapping: Lessons Learnt in the Last 10 Years"*** Peter England - Swiss Association of Actuaries - 10 September 2010

Fit ODP model with R

```
INCR<-matrix(c(357848, 352118, 290507, 310608, 443160, 396132, 440832, 359480, 376686,
344014, 766940, 884021, 1001799, 1108250, 693190, 937085, 847631,1061648, 986608, NA,
610542, 933894, 926219, 776189, 991983, 847498, 1131398, 1443370, NA, NA, 482940,
1183289, 1016654, 1562400, 769488, 805037, 1063269, NA, NA, NA, 527326, 445745,
750816, 272482, 504851, 705960, NA, NA, NA, NA,574398, 320996, 146923, 352053, 470639,
NA, NA, NA, NA, NA, 146342, 527804, 495992, 206286, NA, NA, NA, NA, NA, NA,  139950,
266172, 280405, NA, NA, NA, NA, NA, NA, NA, 227229, 425046, NA, NA, NA, NA, NA,
NA, NA, NA, 67948, NA, NA, NA, NA, NA, NA, NA, NA, NA),nrow=10,ncol=10,byrow=F)

# Fit ODP
n= nrow(INCR)
lig = rep(1:n, each=n); col = rep(1:n, n)
past = (lig + col - 1)<=n;
Y = as.vector(INCR)
base=data.frame(Y,lig,col)
fit=glm(Y~as.factor(lig)+as.factor(col),family=quasipoisson)
```

Fitted incremental values

```
#fitted incremental values
YP=predict(fit,newdata=base,type="response")
matrix(YP,nrow=10,ncol=10,byrow=F)
```

```
> matrix(YP,nrow=10,ncol=10,byrow=F)
          [,1]      [,2]      [,3]      [,4]     [,5]     [,6]     [,7]     [,8]     [,9]     [,10]
 [1,] 270061.4  672616.7  704494.1  753437.8 417350.2 292570.6 268343.5 182034.7 272606.0  67948.00
 [2,] 376125.0  936779.4  981176.3 1049342.0 581259.8 407474.4 373732.4 253526.8 379669.0  94633.81
 [3,] 372325.3  927315.9  971264.3 1038741.3 575387.7 403358.0 369956.9 250965.6 375833.5  93677.80
 [4,] 366724.0  913365.1  956652.3 1023114.2 566731.5 397289.8 364391.2 247190.0 370179.3  92268.49
 [5,] 336287.3  837559.2  877253.8  938199.6 519694.9 364316.2 334148.1 226674.1 339455.9  84610.55
 [6,] 353798.1  881171.9  922933.4  987052.7 546756.0 383286.6 351547.5 238477.3 357131.7  89016.32
 [7,] 391841.7  975923.4 1022175.5 1093189.5 605548.1 424501.0 389349.1 264120.5 395533.7  98588.16
 [8,] 469647.5 1169707.2 1225143.3 1310258.2 725788.5 508791.9 466660.0 316565.5 474072.7 118164.27
 [9,] 390560.8  972733.2 1018834.1 1089616.0 603568.6 423113.4 388076.4 263257.2 394240.8  98265.88
[10,] 344014.0  856803.5  897410.1  959756.3 531635.7 372687.0 341825.7 231882.4 347255.4  86554.62
```

Adjusted and scaled residuals
with a constant scale parameter

```
# adjust and scale residuals
nobs <- sum(1:n)
p<-2 * n – 1
adj.unscaled.resids <-(Y-YP)/(YP)^0.5* sqrt(nobs/(nobs - p))
phi<-sum(adj.unscaled.resids[past]^2)/nobs
adj.resids <- adj.unscaled.resids/(phi)^0.5
matrix(round(adj.resids,3),nrow=10, ncol=10,byrow=F)
```

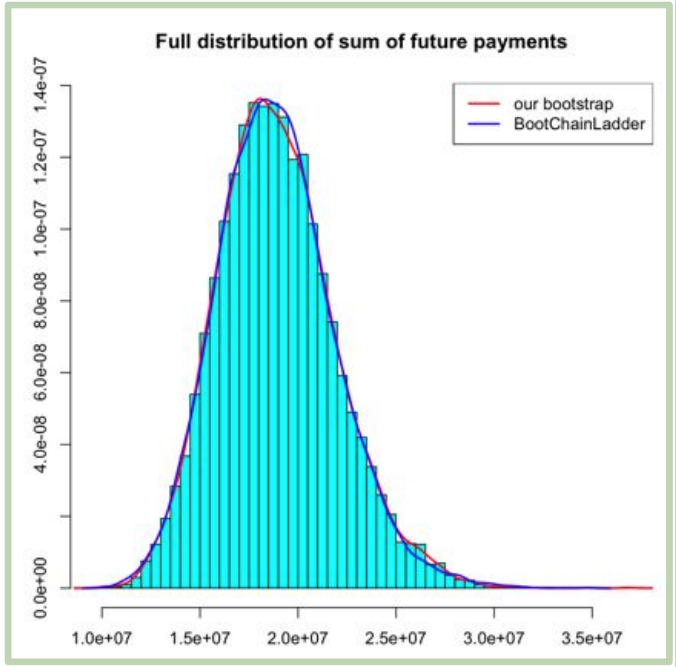
```
> matrix(round(adj.resids,3),nrow=10,ncol=10,byrow=F)
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9] [,10]
 [1,]  0.910  0.620 -0.603 -1.679  0.917  2.808 -1.269 -0.532 -0.468     0
 [2,] -0.211 -0.294 -0.257  0.705 -0.958 -0.730  1.358  0.135  0.397    NA
 [3,] -0.723  0.417 -0.246 -0.117  1.246 -2.176  1.117  0.317     NA    NA
 [4,] -0.499  1.099 -0.994  2.873 -2.106 -0.387 -1.412     NA     NA    NA
 [5,]  0.993 -0.850  0.660 -0.939 -0.111  0.949     NA     NA     NA    NA
 [6,]  0.384  0.321 -0.423 -0.987  1.160     NA     NA     NA     NA    NA
 [7,]  0.422 -0.700  0.582 -0.154     NA     NA     NA     NA     NA    NA
 [8,] -0.866 -0.538  1.063     NA     NA     NA     NA     NA     NA    NA
 [9,] -0.120  0.076     NA     NA     NA     NA     NA     NA     NA    NA
[10,]  0.000     NA     NA     NA     NA     NA     NA     NA     NA    NA
```

Resample residuals/ obtain pseudo data/ estimate future payments / add random fluctuations – **10,000 times**

```
k_simu<-10000 ; set.seed(2)
R1=rep(NA,k_simu)

for (s in 1:k_simu){
  # resample residuals and obtain pseudo data
  Ysim=YP+sqrt(phi*YP)*sample(adj.resids[!is.na(adj.resids)], size=n^2,replace=TRUE)
  Ysim[past==FALSE]=NA
  # Use chain ladder to re-fit model, and estimate future incremental payments
  INCRsim=matrix(Ysim,n,n)
  PAIDsim=INCRsim
  for (j in 2:n) PAIDsim[,j]=PAIDsim[,j-1]+INCRsim[,j]
  lambda <- rep(NA,n-1)
  for(i in 1:(n-1)) lambda[i] <- sum(PAIDsim[1:(n-i),i+1])/ sum(PAIDsim[1:(n-i),i])
  for(i in 1:(n-1)) PAIDsim[(n-i+1):(n),i+1]=lambda[i]* PAIDsim[(n-i+1):(n),i]
  INCRsim <-PAIDsim
  INCRsim[,2:n] <- PAIDsim[,2:n]-PAIDsim[,1:(n-1)]
  # We obtain here the BEs of future incremental payments
  MU=INCRsim[past==FALSE]
  # add random fluctuations in the 2nd part of the triangle. Use a gamma as an approx of a quasi poisson
distribution
  R1[s]=sum(sign(MU) %*% rgamma(length(MU),shape=abs(MU)/phi,scale=phi))
}
```

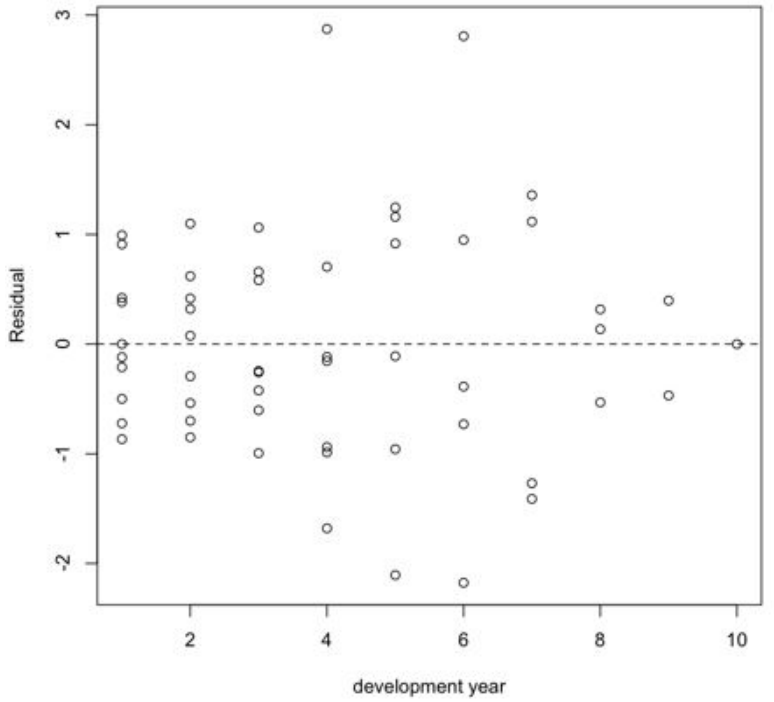
Compare with predefined function from the ChainLadder package

```
# Compare with BootChainLadder function from R package "ChainLadder"
set.seed(2)
n <- nrow(INCR)
PAID=INCR; for (j in 2:n) PAID[,j]=PAID[,j-1]+INCR[,j]
library(ChainLadder)
BCL<-BootChainLadder(PAID, R = k_simu,
process.distr="gamma")

truehist(R1)
lines(density(R1),col=2,lwd=2)
lines(density(BCL$IBNR.Totals),col=4,lwd=2)
legend("topright",c("our bootstrap","BootChainLadder"),
col=c(2,4), lwd=c(2,2))
title("Full distribution of sum of future payments")
```



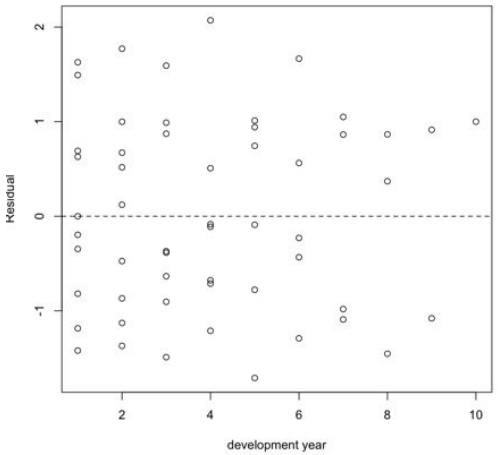
Test constant scale parameter assumption

```
# plot development residuals to test appropriateness of ODP with constant scale parameter
par(mfrow=c(1,1))
plot(lig[past],adj.resids[past],
xlab="development year",ylab="Residual")
abline(h=0,lty=2)
```



Adjust and scale residuals with non constant scale parameters

```
# adjust and scale residuals with non constant scale parameters
phi_j<-c()
for (j in 1:n) phi_j[j] <- sum(adj.unscaled.resids[past & lig==j]
^2)/length(adj.unscaled.resids[past & lig==j])adj.resids2<-
adj.resids
for (j in 1:n) adj.resids2[past & lig==j] <-adj.resids[past &
lig==j]*sqrt(phi)/ sqrt(phi_j[j])

# plot development adjusted residuals with non constant scale
parameters
par(mfrow=c(1,1))
plot(lig[past],adj.resids2[past],xlab="development
year",ylab="Residual")
abline(h=0,lty=2)
```



Note that the residuals are standardised better when using non-constant scale parameters

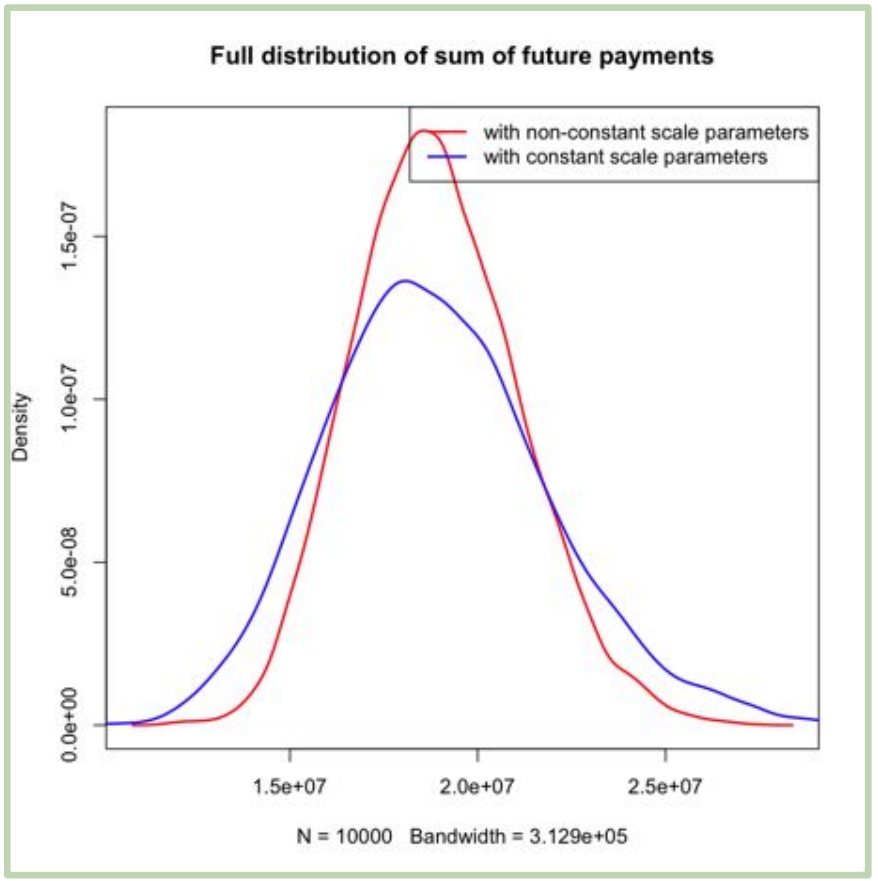
Resample residuals/ obtain pseudo data/ estimate future payments / add random fluctuations – **10,000 times**

```
R2=rep(NA,k_simu) ; set.seed(2)

for (s in 1:k_simu){
 # resample residuals and obtain pseudo data
  Ysim=YP+sqrt(YP*rep(phi_j,each=n))*sample(adj.resids2[past], size=n^2,replace=TRUE)
  Ysim[past==FALSE]=NA
 # Use chain ladder to re-fit model, and estimate future incremental payments
  INCRsim=matrix(Ysim,n,n)
 PAIDsim=INCRsim
 for (j in 2:n) PAIDsim[,j]=PAIDsim[,j-1]+INCRsim[,j]
 lambda <- rep(NA,n-1)
 for(i in 1:(n-1)) lambda[i] <- sum(PAIDsim[1:(n-i),i+1])/ sum(PAIDsim[1:(n-i),i])
 for(i in 1:(n-1)) PAIDsim[(n-i+1):(n),i+1]=lambda[i]* PAIDsim[(n-i+1):(n),i]
 INCRsim <-PAIDsim
 INCRsim[,2:n] <- PAIDsim[,2:n]-PAIDsim[,1:(n-1)]
 # We obtain here the BEs of future incremental payments
 MU=INCRsim[past==FALSE]
 # add random fluctuations in the 2nd part of the triangle. Use a gamma as an approx of a quasi poisson
distribution
 R2[s]=sum(sign(MU) %*% rgamma(length(MU),shape=abs(MU)/rep(phi_j,0:(n-1)),scale=rep(phi_j,0:
(n-1))))
 }
```

Compare with previous results

```
plot(density(R2),col=2,lwd=2,main="Full distribution of sum of future payments")
lines(density(R1),col=4,lwd=2)
legend("topright",c("with non-constant scale parameters","with constant scale parameters"), col=c(2,4), lwd=c(2,2))
```

```
> sd(R2)
[1] 2193731
> sd(R1)
[1] 2989050
```



Bootstraping beyond the ODP model

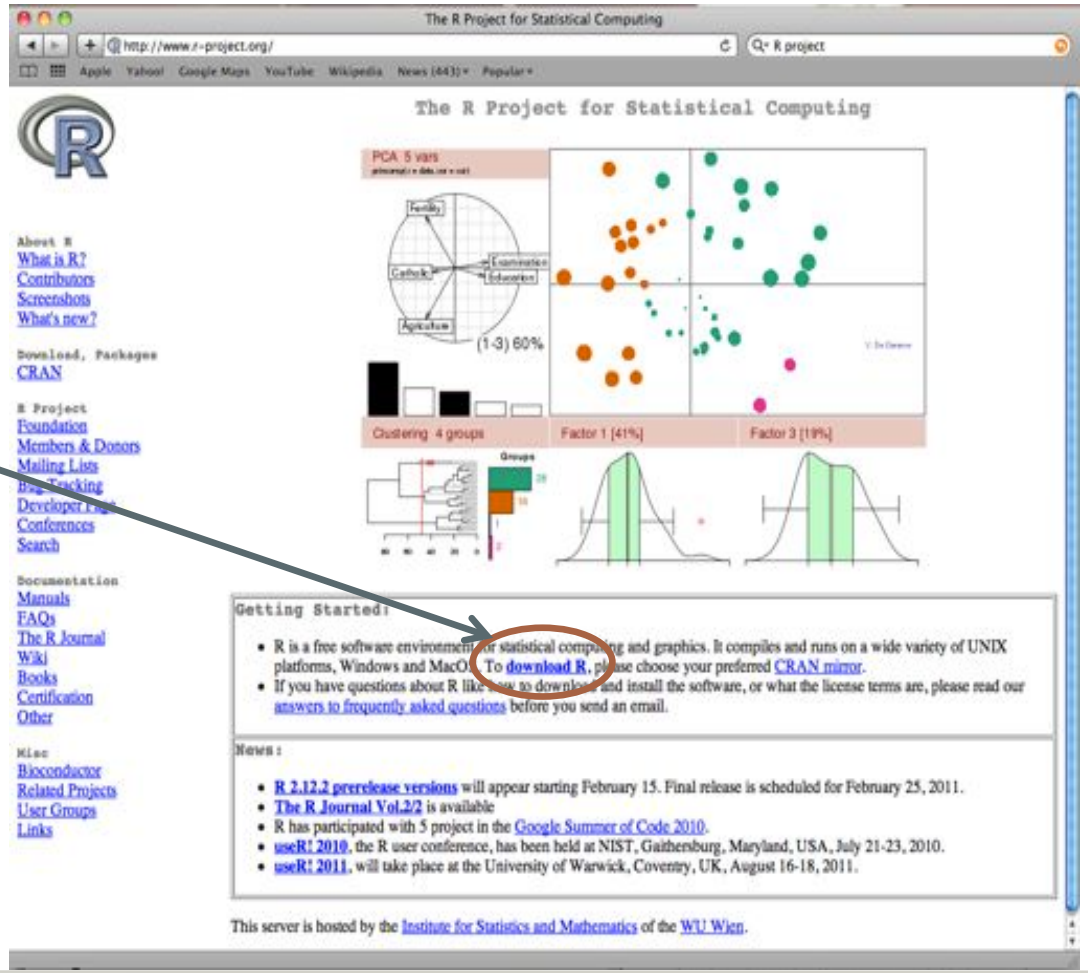
- The bootstraping approach is not only applicable to the ODP model. Here are a few examples:
  - Huijuan Liu and Richard Verrall propose a bootstrap algorithm for the Munich Chain Ladder in "***Bootstrap Estimation of the Predictive Distributions of Reserves Using Paid and Incurred Claims***"
  - Peter England describes a bootstrap of the Mack model in "***Bootstrapping: Lessons Learnt in the Last 10 Years"*** - Swiss Association of Actuaries - 10 September 2010
  - Some research has been done for the Bornhuetter-Ferguson Method but researchers tend to prefer the Bayesian apprach. See "***Bayesian Overdispersed Poisson Model and the Bornhuetter-Ferguson Claims Reserving Method***" by England, Verrall and Wuthrich
- For those who still want to use Excel, Shapland provided Excel files at his presentation "Bootstrat Modeling: Beyond the Basics" at the last GI seminar of Institute of Actuaries of Australia" available @ http://www.actuaries.asn.au/GIS2010/Program/ProgramSnapshot.aspx

A few slides for new R users on how to

- Install R
- Install additional packages
- Execute commands

R installation

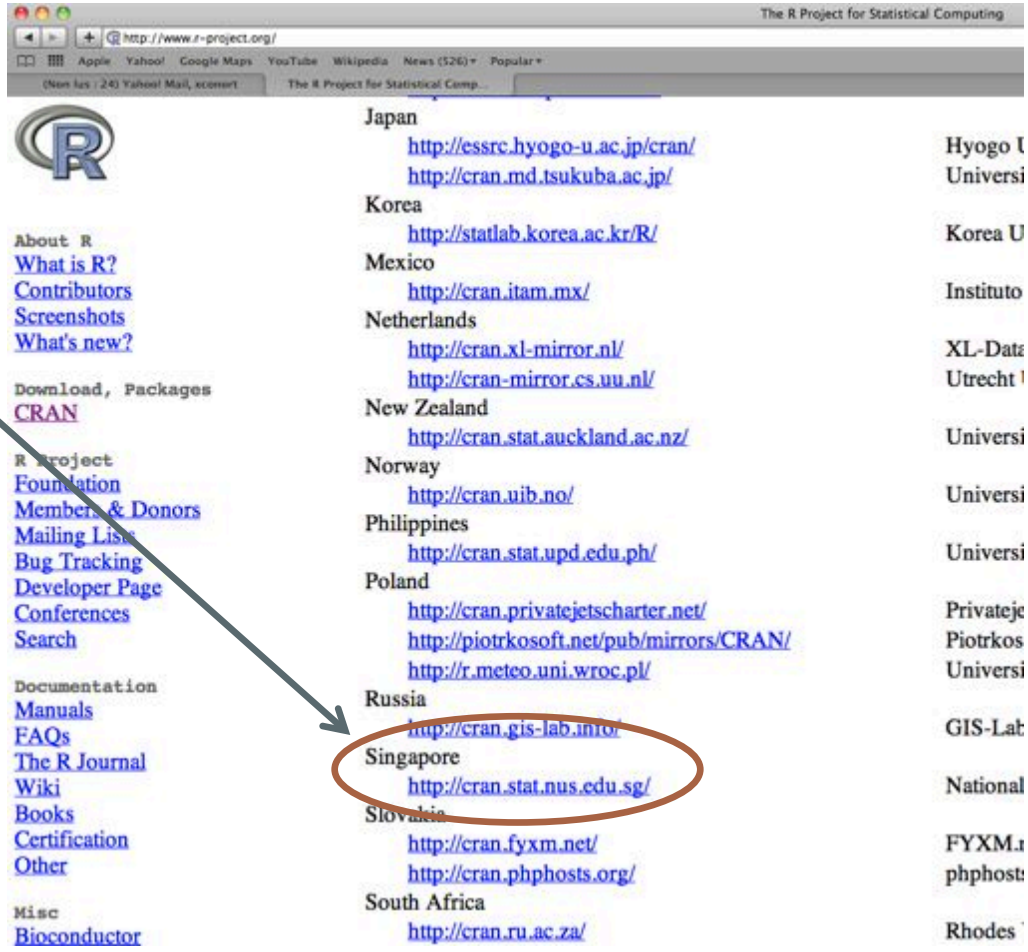
Download R from the website
http://www.r-project.org/



R installation

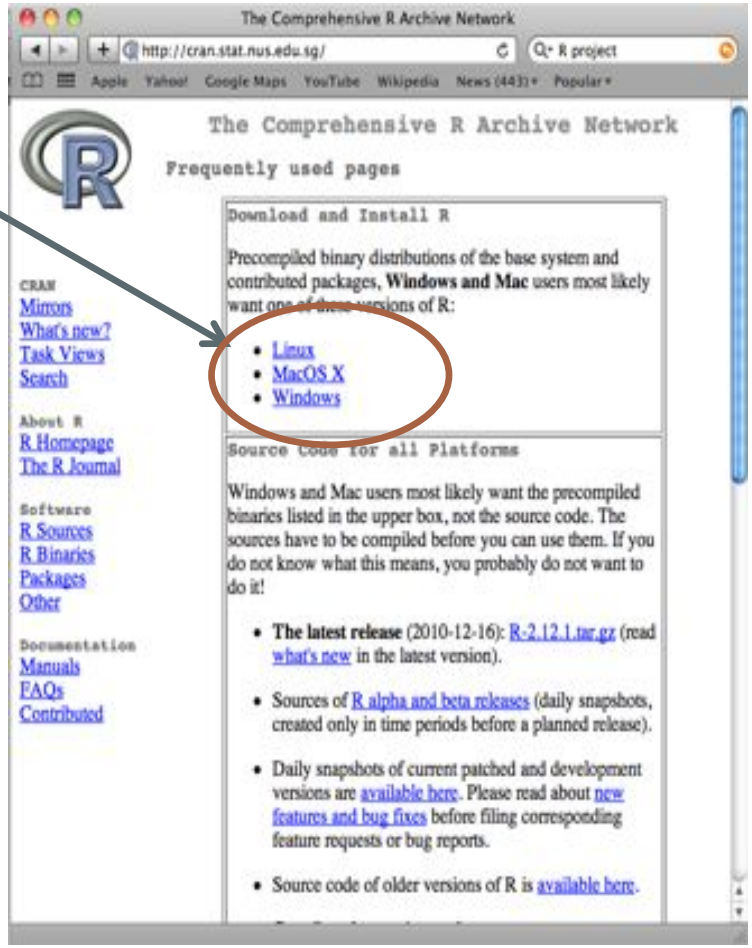
Select NUS as a CRAN mirror (sites around the world that store R software and its packages).

After the click, you shall access to cran.stat.nus.edu.sg.



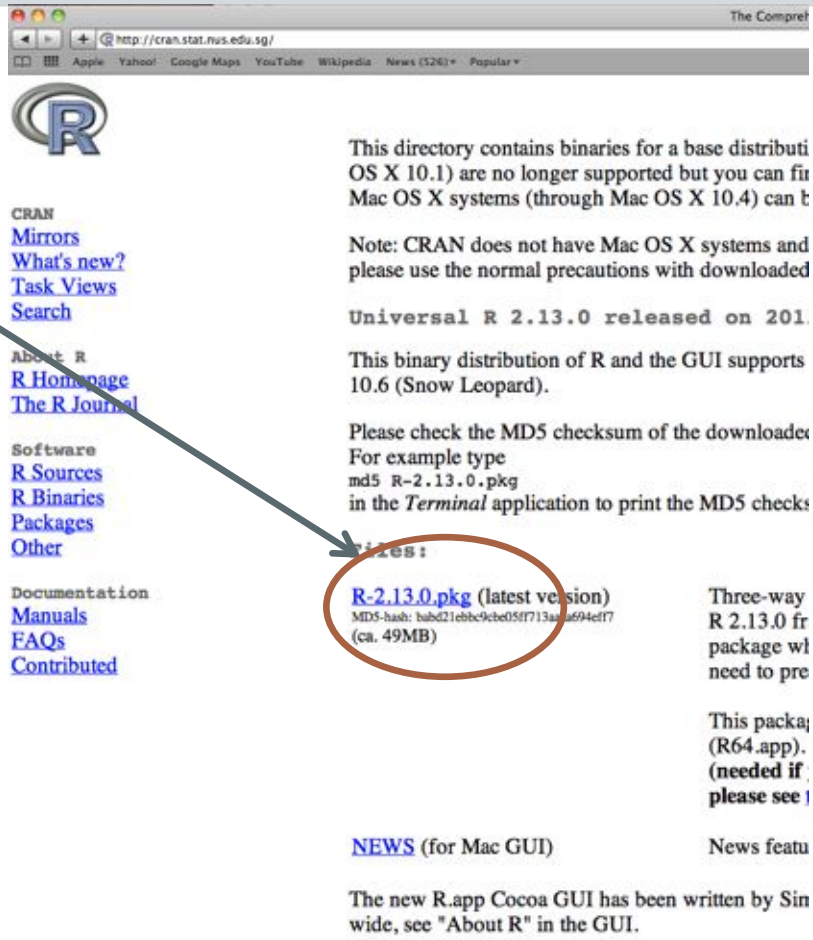
R installation

Select your platform



R installation

Install the latest version
(here the example is for mac)



R installation

Installation time should take less than one minute

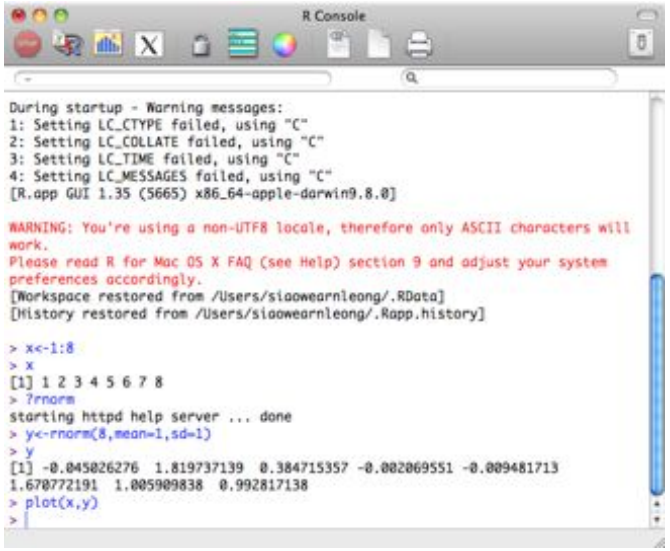
Open R

When you start R, the window which first appears is the R console:

- It shows commands and results

Commands in R can be typed either

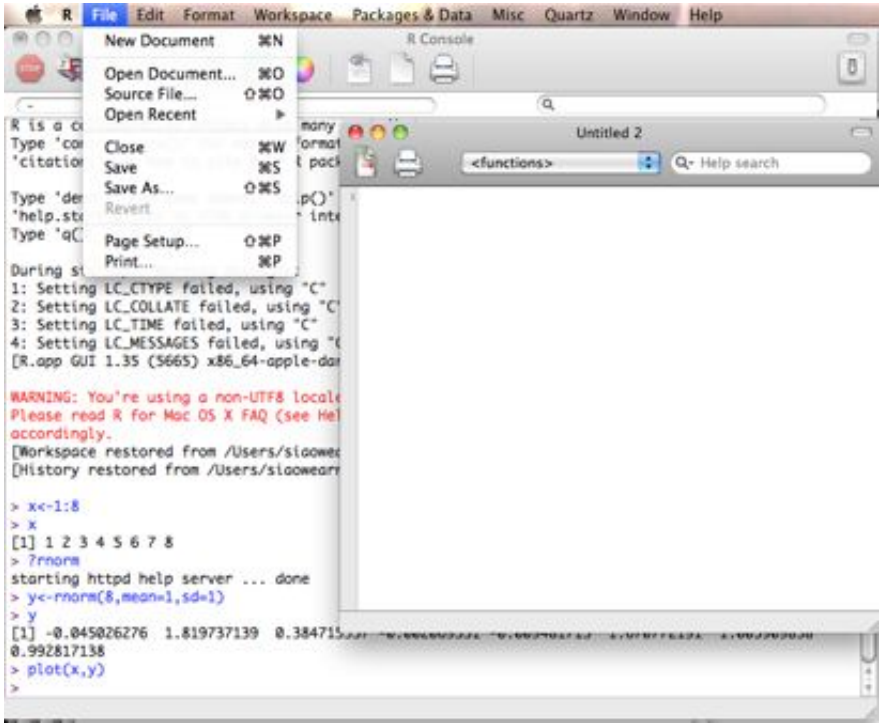
- directly in the R console,
- or in the R editor

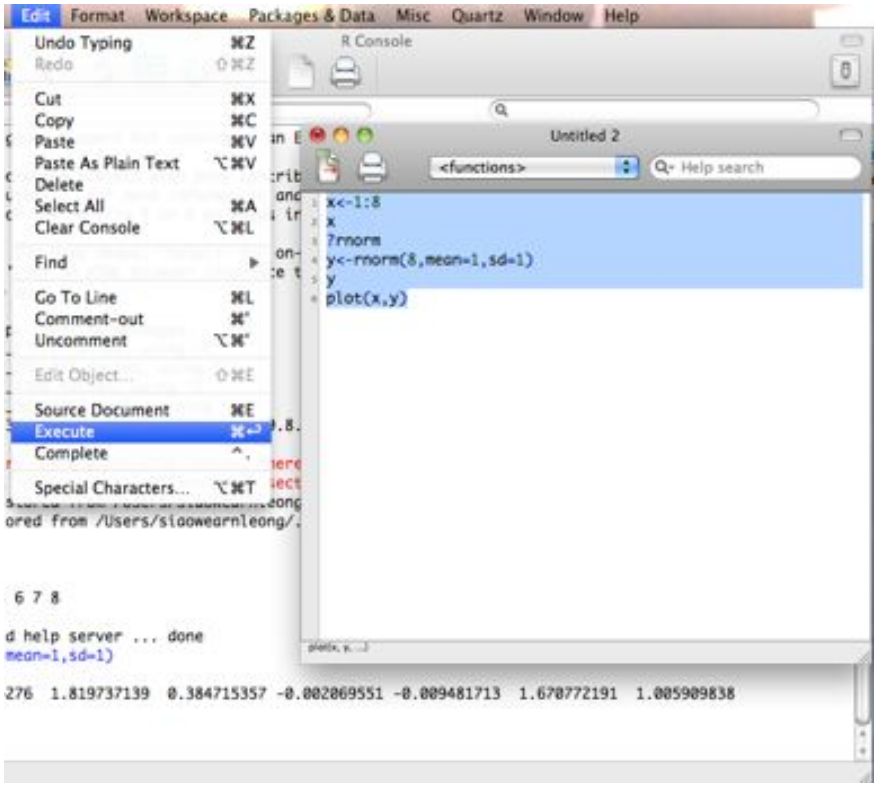


- The > symbol is an invitation to start typing and indicates that R is ready for another command
- The + symbol signals that the command is incomplete
- Text following the # symbol will be ignored
- The ↑ and the ↓ keys are used to replay previously entered commands. They can be modifed and then submitted.

R editor

Usually, work is done in the R Editor as script files can be saved and reused for similar tasks.

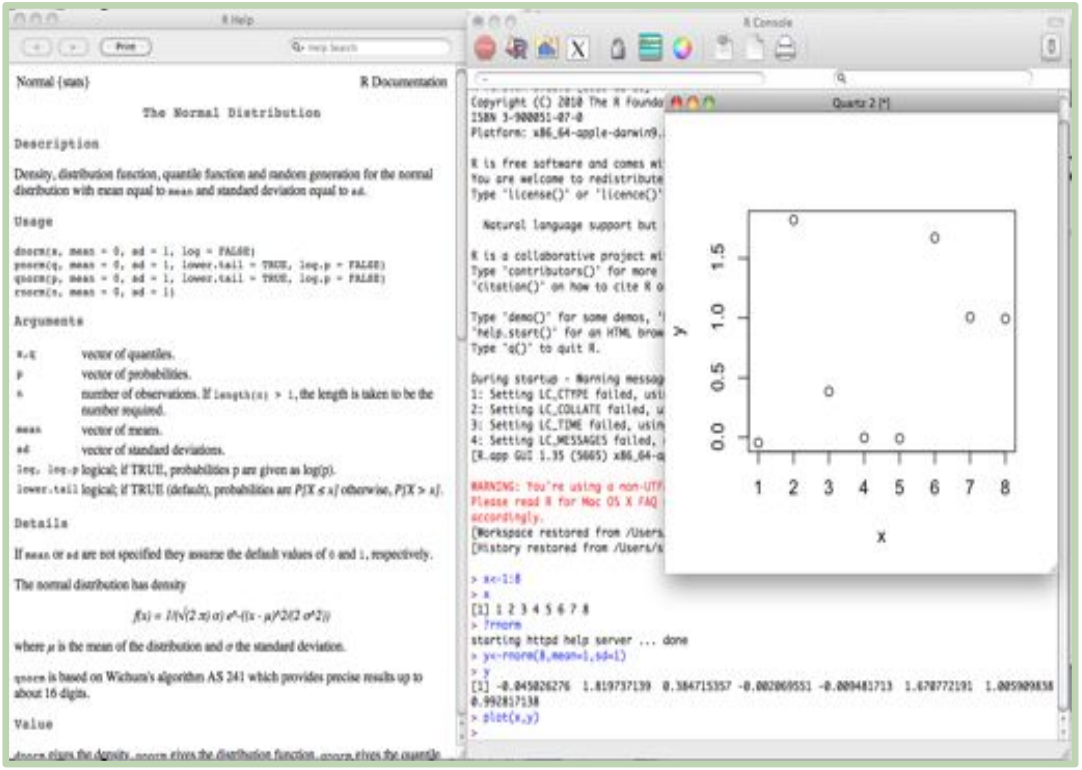




2 additional windows:
The graphical and the R Help windows

The graphical window: shows graphic results. Once the window open, you can reshape it manually to better fit your graphics.

The R Help window : can be called typing "?" following the function name for which you are looking for info



R packages

- Thousands of packages are available:
  - **"official" R packages** that are created by the R Core Team
  - packages that have been contributed by many people. Some of these packages represent **cutting-edge statistical research** as a lot of statistical research is first implemented in R.
- Some have been **developed by actuaries for actuaries**
  - Markus Gesmann and Wayne Zhang's **Chainladder** package which implements Chain Ladder based stochastic reserving methods
  - The **actuar** Package developed and maintained by Vincent Goulet which includes several functions of interest to actuaries
  - **lossDev**, by Christopher Laws and Frank Schmid which uses a Bayesian method of stochastic reserving

Install Packages

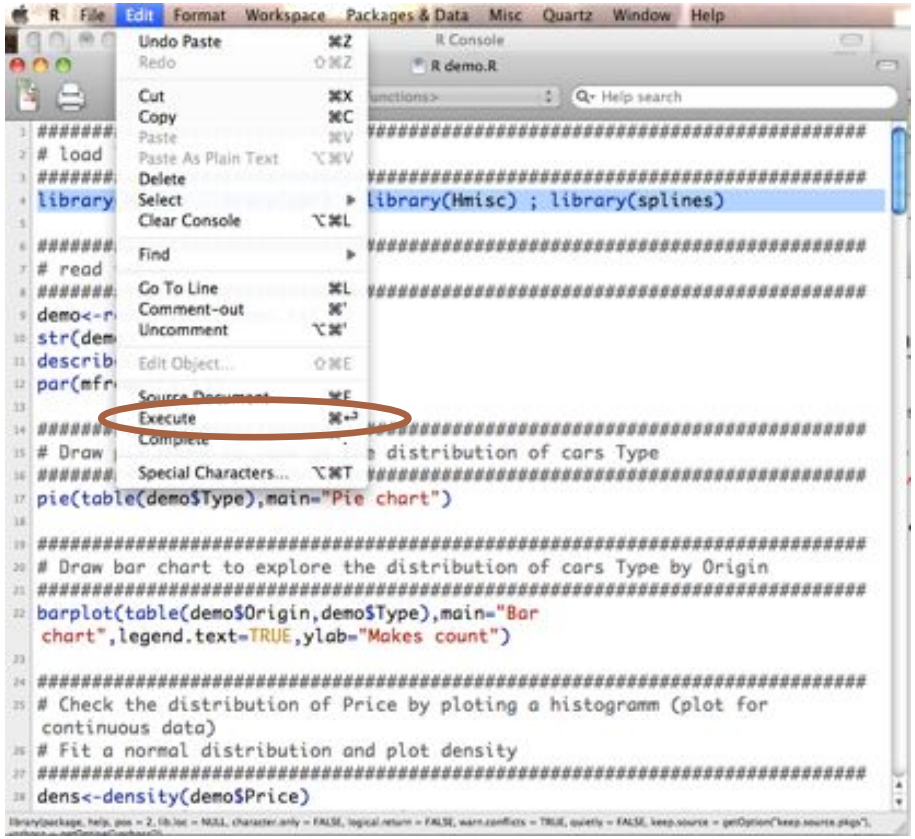
- There are **two main options for installing packages in R**.
  - First, you can download and install a package using the install.packages command:

    > install.packages(*"package name"*) Make sure to include the quotation marks around the package name (either single or double quotes will work).
  - Alternatively, you can choose the "Packages" drop-down menu, and the "Install Package(s)..." option
- Once a package has been installed, you do not need to reinstall it. However, you will need to load it into a library in each session when you wish to use it. You can load a package using:

  > library(*package name*)

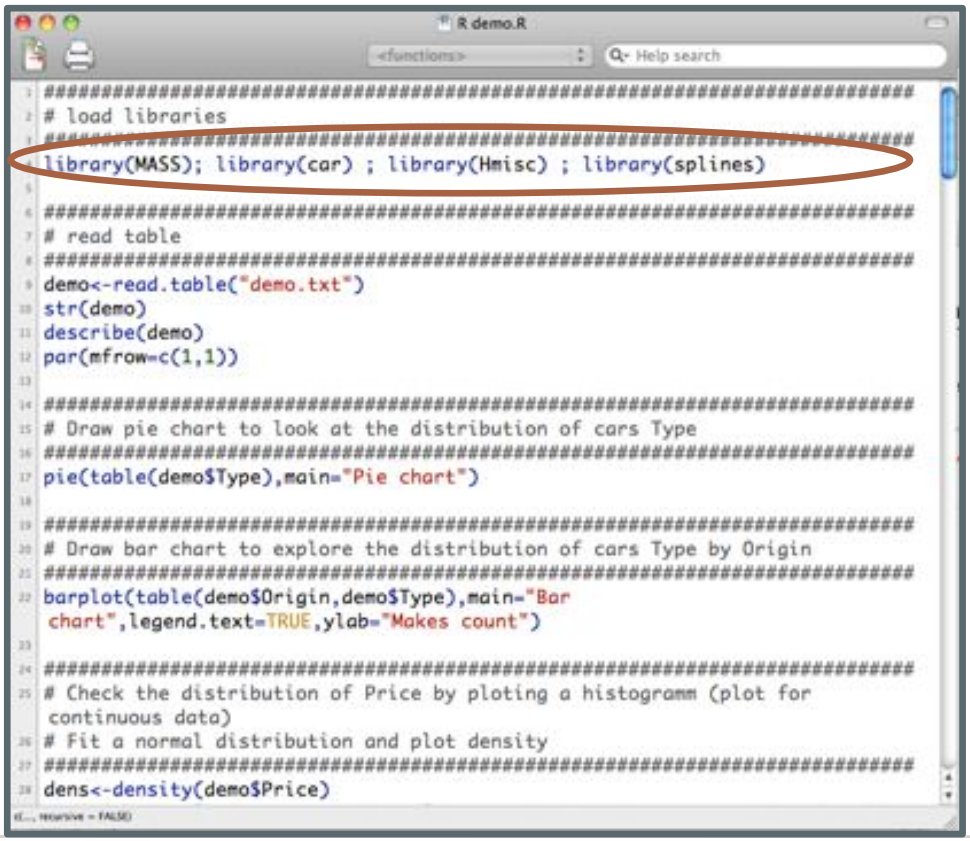
Execute a command line from the editor

- Select a command line or a group of command lines in the editor.
- Execute it by selecting the option “Execute” in the drop-down menu “Edit”
- Next slide will show you an example



Load packages

Select the command line from the editor and execute it.



You shall find the following result in the R console