Introduction to the Bootstrap

1. An introductory example

Verizon is the primary local telephone company (the legal term is Incumbent Local Exchange Carrier, ILEC) for a large area in the eastern United States. As such, it is responsible for providing repair service for the customers of other telephone companies (known as Competing Local Exchange Carriers, CLECs) in this region. Verizon is subject to fines if the repair times (the time it takes to fix a problem) for CLEC customers are substantially worse than those for Verizon's own customers. This is determined using hypothesis tests, negotiated with the local Public Utilities Commission (PUC).

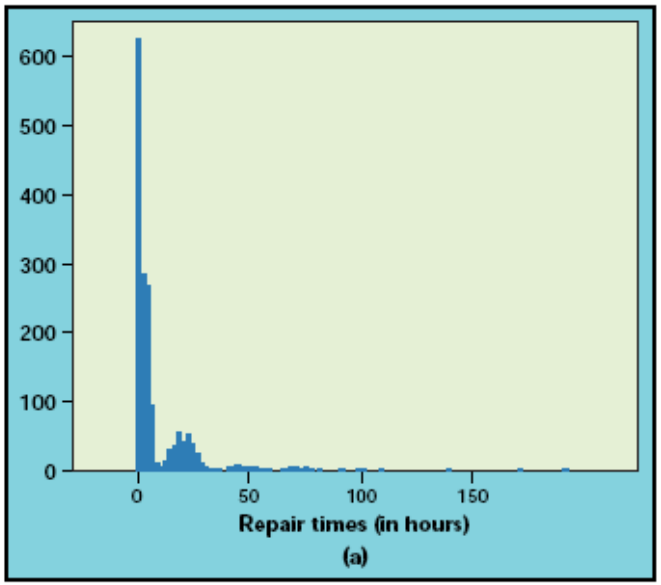
We begin our analysis by focusing on Verizon's own customers. A random sample of 1664 repair times has been observed.

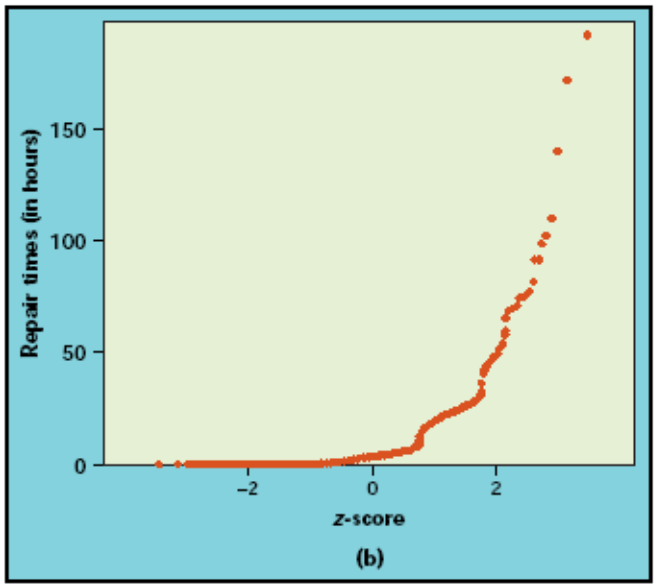
Goal $\rightarrow$ To get some insight about the expected value of a repair time in the population, μ, namely to determine a 95% confidence interval for this parameter. The first step to get an answer to this problem is to look at the sampling distribution of $\overline{X}$.

Observed data

A quick glance at the empirical distribution (see histogram and qq-plot) reveals that the data are far from Normal. The distribution has a long right tail. However, as the sample size is large ($n = 1664$) the Central Limit Theorem is a possible solution, but how can we check if the sampling distribution of $\overline{X}$ can be approximated by a normal distribution?

As the distribution of X is unknown we can't use simulation or take advantage of theoretical results





How to overcome this situation? Use non-parametric bootstrap

The idea is to recognize the sample as the best possible approximation to the distribution of X and to resample from the original sample.

- Step 1 – Create many (hundreds or thousands) bootstrap samples (B) with the same sample size of the original sample using **random selection (resampling) with replacement** from the original sample.
- Step 2 – For each bootstrap sample, calculate the value of the statistic T (here $T = \bar{X}$). This step is similar to simulation except that we are dealing with a bootstrap sample instead of a pseudo random sample from the population. At the end of step 2 we get the bootstrap distribution of T
- Step 3 – Now use the bootstrap distribution to make statistical inference about *T*. For instance, the standard deviation of T is estimated using bootstrap standard error

$$SE_{boot,T} = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}(t_i^* - \bar{t}^*)^2}$$

where t_i^* - observed value of *T* for the *i*-th bootstrap sample and $\bar{t}^* = \frac{1}{B}\sum_{i=1}^{B} t_i^*$

When $T = \bar{X}$ we get $SE_{boot,\bar{X}} = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}(\bar{x}_i^* - \frac{1}{B}\sum_{j=1}^{B}\bar{x}_j^*)^2}$

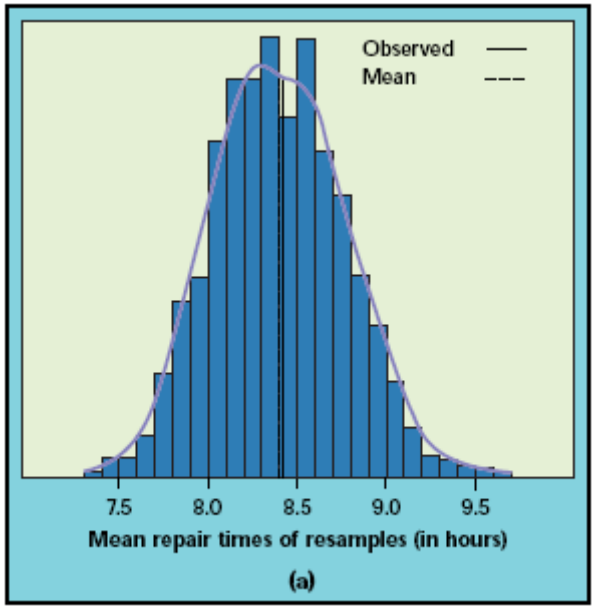
Steps 1 and 2	**Step 3**
$(x_1, x_2, \cdots, x_n)$ original sample $\rightarrow$ $\begin{cases} \text{bootstrap sample 1} \rightarrow t_1^* \\ \text{bootstrap sample 2} \rightarrow t_2^* \\ \cdots \\ \text{bootstrap sample } B \rightarrow t_B^* \end{cases}$ $\downarrow$ t_{obs}	$(t_1^*, t_2^*, \cdots, t_B^*) \rightarrow$ bootstrap distribution of T

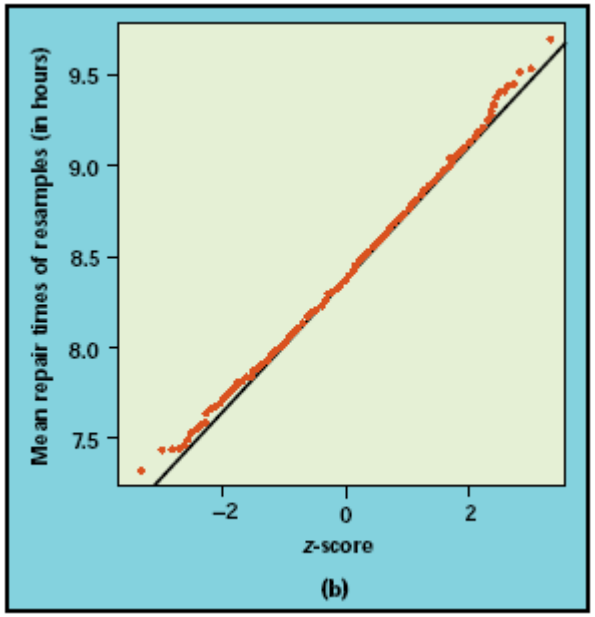
Bootstrap applied to repair times for Verizon customers

Observed sample and bootstrap distribution

Observed sample	Bootstrap dist. (*B*=1000)
$\bar{x} = 8.412$	$\bar{x} = 8.406$
$s = 14.686$ $\quad s/\sqrt{n} = 0.3600$	$SE_{boot,\bar{X}} = 0.3617$
$s' = 14.690$ $\quad s'/\sqrt{n} = 0.3601$	Skew=0.1093
Skew=4.576 $\quad$ Kurt=34.43	Kurt=0.0462

We can then conclude that it seems adequate to use the CLT approximation (see histogram and qq-plot)


Observed
Mean
7.5
8.0
8.5
9.0
9.5
Mean repair times of resamples (in hours)
(a)


Mean repair times of resamples (in hours)
9.5
9.0
8.5
8.0
7.5
−2
0
2
z-score
(b)

A step by step program using R

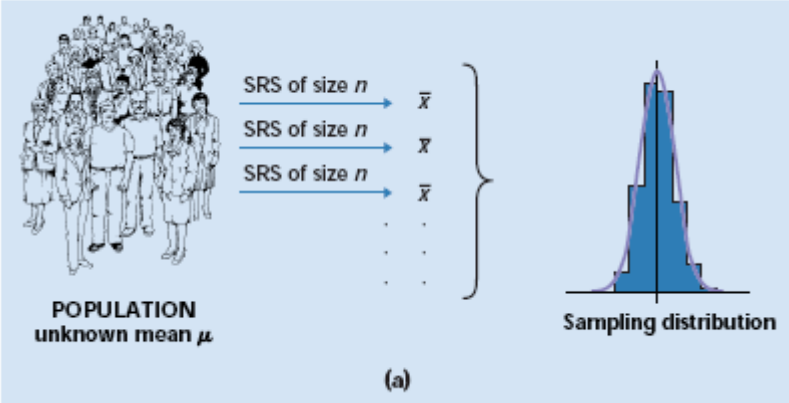
```
time1=scan("C:/Users/joaoas/Documents/My Dropbox/Risk models
slides/Verizon1_ilec.prn")
n1=length(time1)
mean.time1=mean(time1); sd.time1=sd(time1)
sk_nc=(1/(n1-1))*sum((time1-mean.time1)^3)/(sd.time1^3)
kurt_nc=(1/(n1-1))*sum((time1-mean.time1)^4)/(sd.time1^4)-3
mean.time1; sd.time1; sk_nc; kurt_nc; summary(time1); hist(time1)
# Bootstrap replicas
B=1001; y=rep(NA,B);
for(i in 1:B){
  xb=sample(time1,replace=TRUE)
  y[i]=mean(xb)
  }
# Bootstrap results
hist(y);  mean.y=mean(y); sd.y=sd(y);
sky=(1/(B-1))*sum((y-mean.y)^3)/(sd.y^3)
kurty=(1/(B-1))*sum((y-mean.y)^4)/(sd.y^4)-3
mean.y; sd.y; sky; kurty;
```

Using library boot

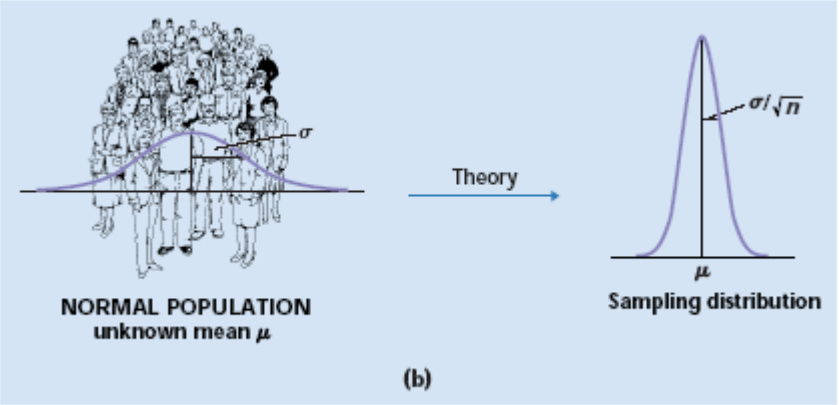
```
library(boot)
b.mean=function(x,i) {
  data=x[i]; return( mean(data))
  }
out1=boot(time1,b.mean,R=1001)
out1
```

2 . Comparing bootstrap with other approaches

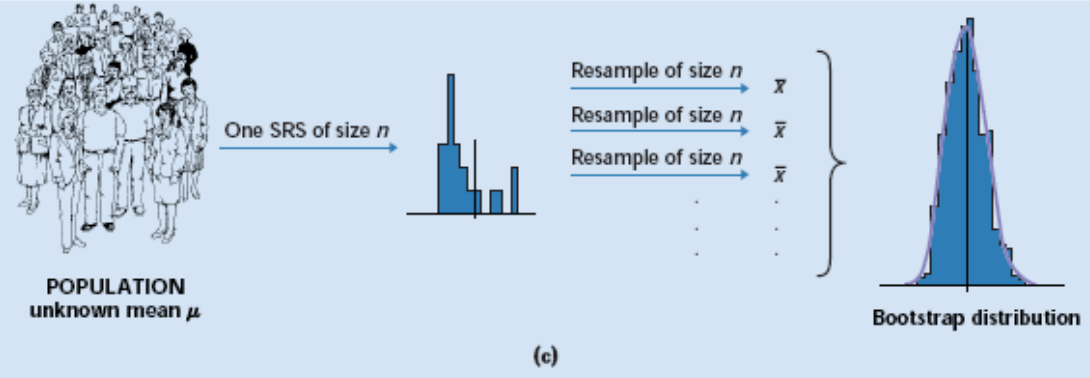
Simulation



Approach based on the normal distribution



Bootstrap approach



3 . How does bootstrap works?

- The observed sample plays the role of the population and the bootstrap samples mimic the simulated observations;
- The Plug-in principle: to estimate a population parameter use the statistic that is the corresponding quantity for the sample;
- For a large number of populations and many statistics, the bootstrap distribution is adequate to approximate the sampling distribution of the statistic; we must exclude (or use appropriate methods to deal with) the treatment of extreme values statistics, namely the sample maximum or minimum. Generally speaking avoid using bootstrap to approximate sampling distribution of order statistic when the sample size is small (the more extreme the order statistic the worse).
- Location is given by the original (observed) sample.
- Bootstrap distribution is useful to
  - Estimate the sampling distribution of the statistic
  - Estimate a dispersion (variance, standard deviation, …)
  - Estimate the bias (and sometimes to correct it)
  - Determine confidence intervals (bootstrap t method and percentile method)
  - Etc...

4. Parametric and non-parametric bootstrap

- Until now we have only considered non-parametric bootstrap, i.e. we assumed that the population distribution is unknown. Let us use the same idea when the population distribution is known up to a set of unknown parameters.

- The first step is obviously to estimate the unknown parameters using maximum likelihood and the observed sample. Now we can improve our resampling procedure using our knowledge about the population distribution. Instead of using only the observed sample we resample from the estimated population.

- **Example:** Let us assume that we observed Data Set B (20 observations) and that we know that our population follows a gamma distribution with unknown parameters. What is the sampling distribution of $\bar{X}$?

  The correct answer is based on theoretical results. As $X \sim G(\alpha,\theta)$, we get (i.i.d. observations) $\sum_{i=1}^{n} X_i = n\bar{X} \sim G(n\alpha,\theta) \Leftrightarrow \bar{X} \sim G(n\alpha,\theta/n)$. Then, we estimate α and β (ML) and use $\bar{X} \sim G(n\hat{\alpha},\hat{\theta}/n)$, i.e. $\bar{X} \sim G(11.1232,128.057)$ as $\hat{\alpha} = 0.55616$ and $\hat{\theta} = 2561.14$ (see Example 15.4 of *Loss Models*)

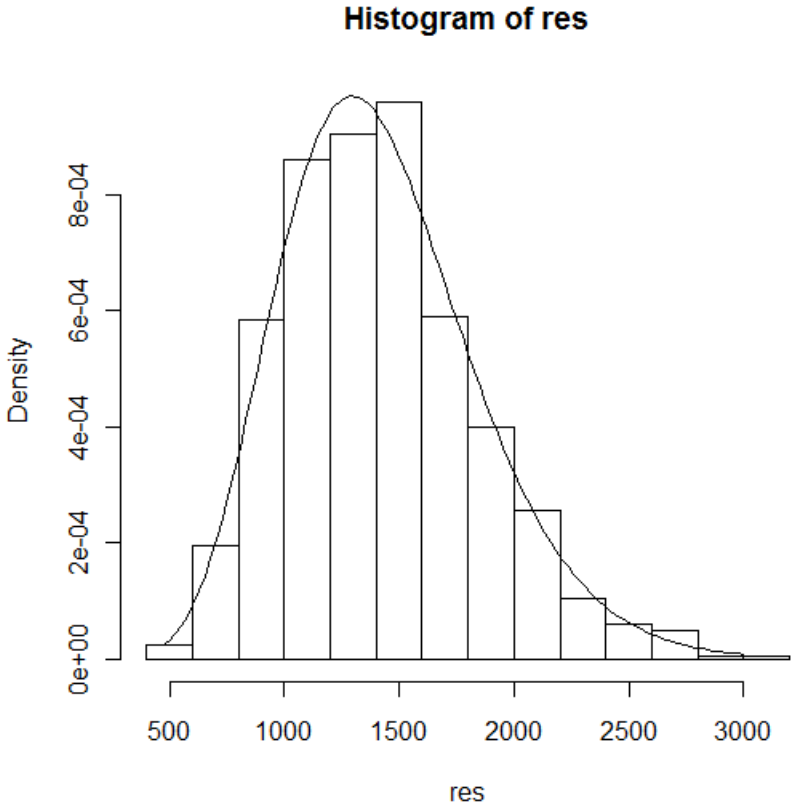
A parametric bootstrap approach can be designed resampling from a Gamma distribution with parameters $\hat{\alpha} = 0.55616$ and $\hat{\theta} = 2561.14$ instead of resampling from the observed sample (non-parametric bootstrap)

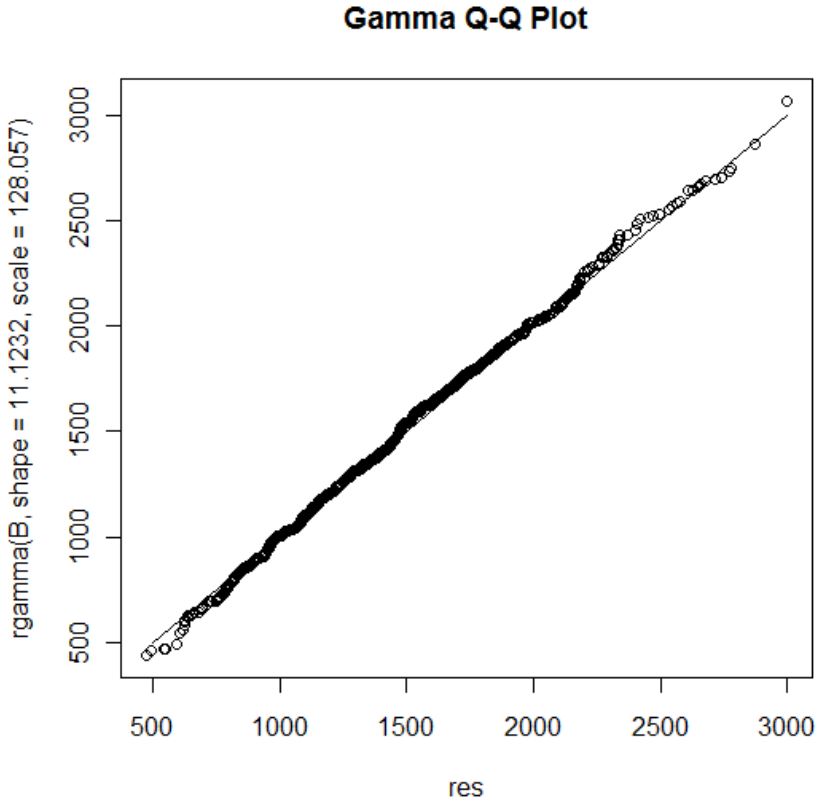
The procedure:

- Estimate the unknown parameters using maximum likelihood
- Determine B, the number of bootstrap replicas to be used
- For each bootstrap replica ($i = 1, 2, \cdots, B$)
    - Generate 20 pseudo gamma random variables (using the estimates obtained in the first stage). We can use an adequate random variables generator or the inverse transform method.
    - Compute the sample mean $\overline{x}_i$
- Using $(\overline{x}_1, \overline{x}_2, \cdots, \overline{x}_B)$ approximate the sampling distribution of $\overline{X}$ (ecdf, empirical statistics, …) and compare with the theoretical results.

R program

```
> theta_hat=2561.14; alpha_hat=0.55616; n=20; B=1000;
> res=rep(NA,B)
> for(i in 1:B){
+   x=rgamma(n,shape=alpha_hat,scale=theta_hat)
+   res[i]=mean(x)
+   }
>
> ks.test(res,"pgamma",shape=11.1232,scale=128.057)

        One-sample Kolmogorov-Smirnov test
data:  res
D = 0.0265, p-value = 0.4815
alternative hypothesis: two-sided

> hist(res,prob=TRUE)
> x=seq(min(res),max(res),length=100)
> dens=dgamma(x,shape=11.1232,scale=128.057)
> lines(x,dens,type="l")
> qqplot(res,rgamma(B,shape=11.1232,scale=128.057),main="Gamma Q-Q
Plot")
> lines(c(500,3000),c(500,3000))
```


Histogram of res
Density
0e+00
2e-04
4e-04
6e-04
8e-04
500
1000
1500
2000
2500
3000
res


Gamma Q-Q Plot
rgamma(B, shape = 11.1232, scale = 128.057)
500
1000
1500
2000
2500
3000
res

5. Bootstrap foundations

- Bootstrap foundations are quite complex and we do not go further in that direction. Two references can be useful both from a theoretical and a **practical** point of view:
  - Efron e Tibshirami, 1993, *An introduction to the bootstrap*, Chapman and Hall;
  - Davison e Hinkley, 1997, *Bootstrap Methods and their Application*, Cambridge University Press.

  For a deeper theoretical approach, see Hall, P., 1992, *The bootstrap and Edgeworth Expansion*, Springer Verlag.
- Bootstrap distribution suffers from 2 sources of randomness: The randomness due to initial sampling process and the randomness due to the random selection of bootstrap samples. In almost all cases the first source of randomness is significantly greater than the second.
  - Increasing sample size can minimize the first point
  - Increasing the number of bootstrap replicas reduces the impact of the second point.

6. Some examples of bootstrap applications

- **One sample problems**:
  - Bias;

    Definition: $bias_T = E(T) - \theta$

    Estimation: $\hat{bias}_T = \overline{t}^* - t_{obs}$ (the sample plays the population role)
  - Confidence intervals;
    - Bootstrap *t*-confidence intervals – If the bootstrap distribution of T is approximately normal (mainly symmetric) and the estimated bias is near 0 we can mimic the usual confidence intervals for normal populations. We get $\left(t_{obs} - z_{\alpha/2}\, SE_{boot,T}; t_{obs} + z_{\alpha/2}\, SE_{boot,T}\right)$
    - Bootstrap percentile confidence intervals – Define the confidence interval using the adequate percentiles of the sampling distribution of T. We get $\left(\pi^*_{\alpha/2}; \pi^*_{1-\alpha/2}\right)$.
  - **Example** – Return to Verizon's time of repairs
    - Analyze the bias of $\overline{X}$ as an estimator of μ
    - Determine two 95% confidence intervals using both approaches;

Step by step program

```
> biasboot=mean.y-mean.time1
> biasboot
[1] 0.02330247
>
> conf=0.95
> # t confidence intervals
> t=-qt((1-conf)/2,B-1)
> cbind(conf,mean(time1)-t*sd.y,mean(time1)+t*sd.y)
     conf
[1,] 0.95 7.716407 9.106814
> # percentile confidence intervals
> cbind(conf,as.numeric(quantile(y,(1-
conf)/2,type=6)),as.numeric(quantile(y,(1+conf)/2,type=6)))
     conf
[1,] 0.95 7.766785 9.146335
>
```

Using boot library

```
boot.ci(out1,conf=0.95,type="norm")
boot.ci(out1,conf=0.95,type="perc")
```

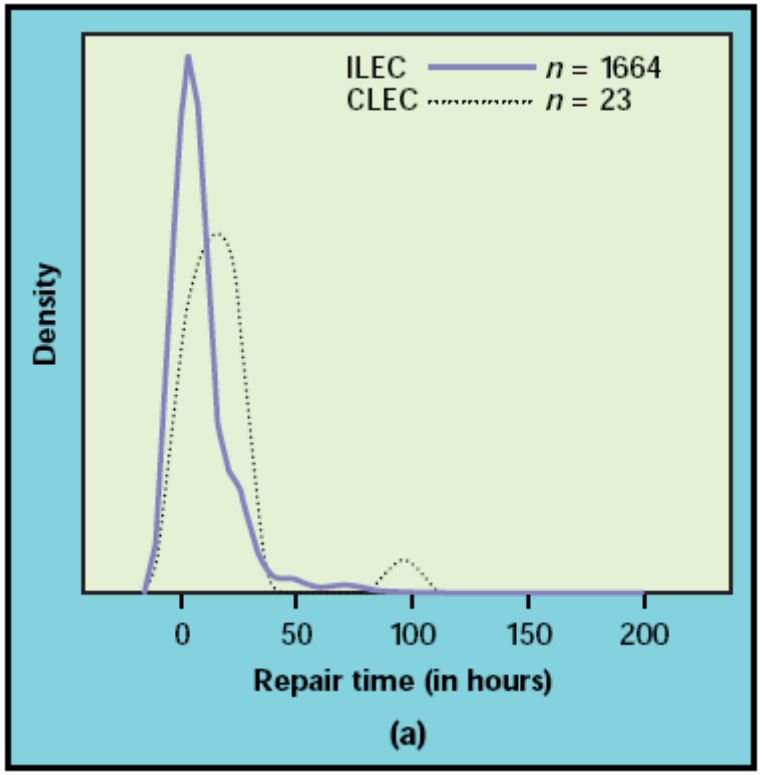
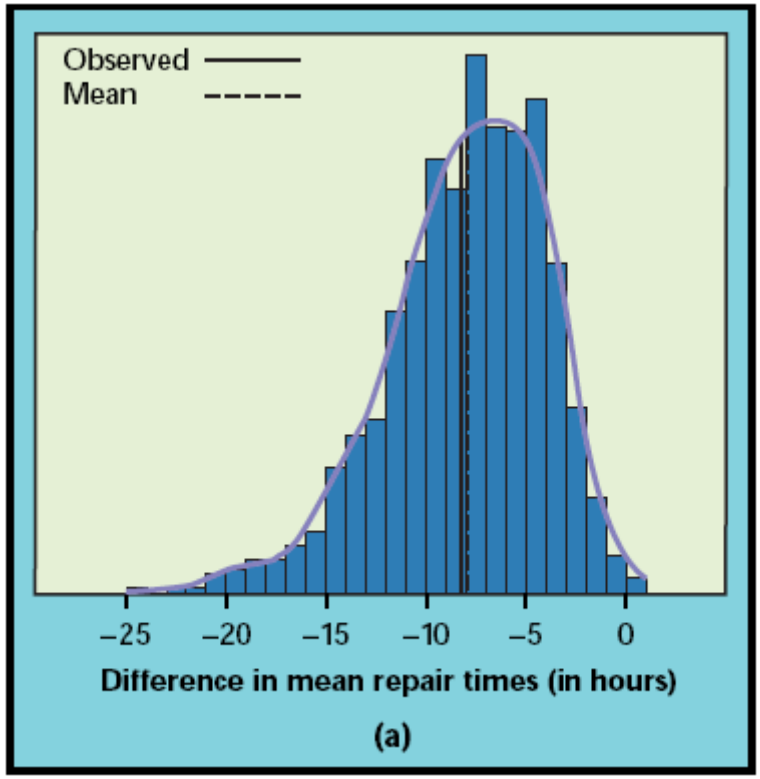
- More efficient confidence intervals can be determined:
  - "bootstrap Bias-Corrected and Accelerated" (BCa): To overcome the effects of populations (and samples) highly asymmetric we can improve the result by choosing non symmetrical percentiles (see Efron e Tibshirami)
  - "bootstrap tilting": The correction is done giving a different probability to each observation (see Davison e Hinkley).

- **Two samples problems**:

  The resampling has to be done according to the problem we are dealing with and to the way information has been collected. Two situations are helpful to understand the point.

  - If we are dealing with a means difference for independent samples, each bootstrap replica is based on independent resampling for each sample. After this independent resampling we use the plug-in principle;

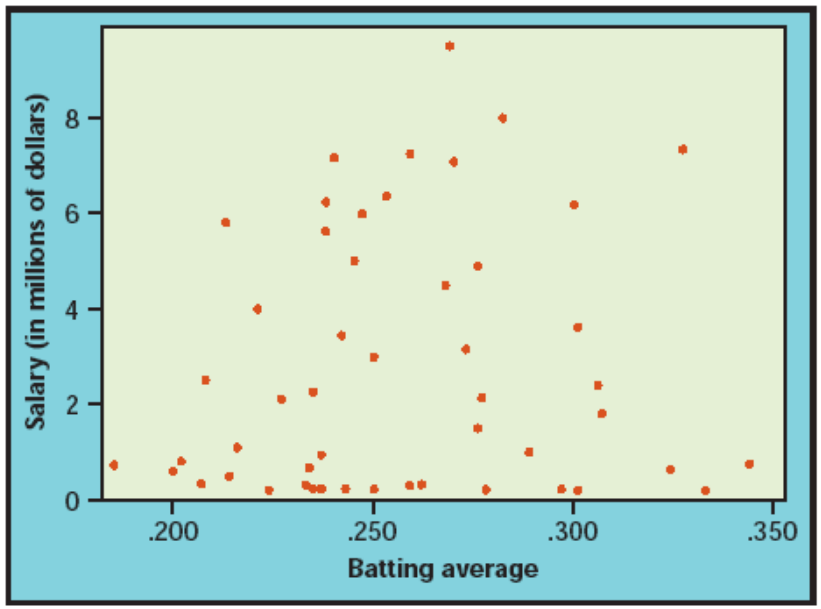
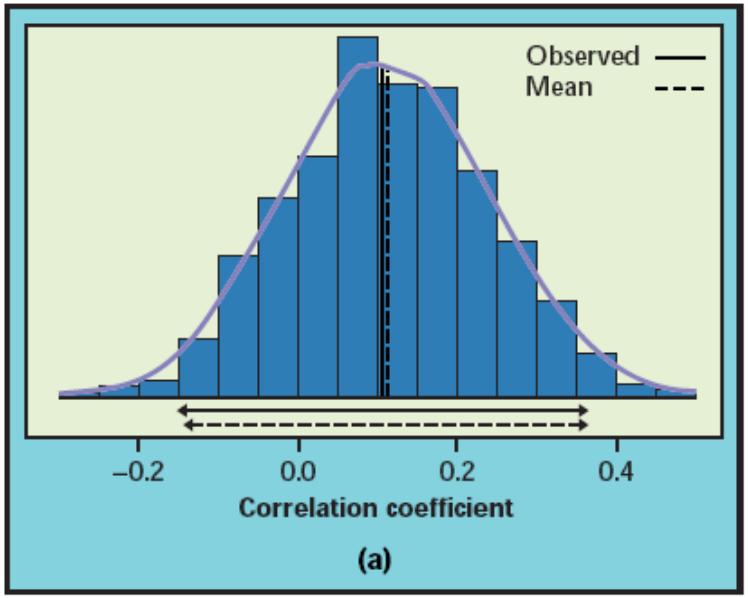
    Example (see 18.7) – Compare the expected repair times for Verizon's customer and for other operators customers.

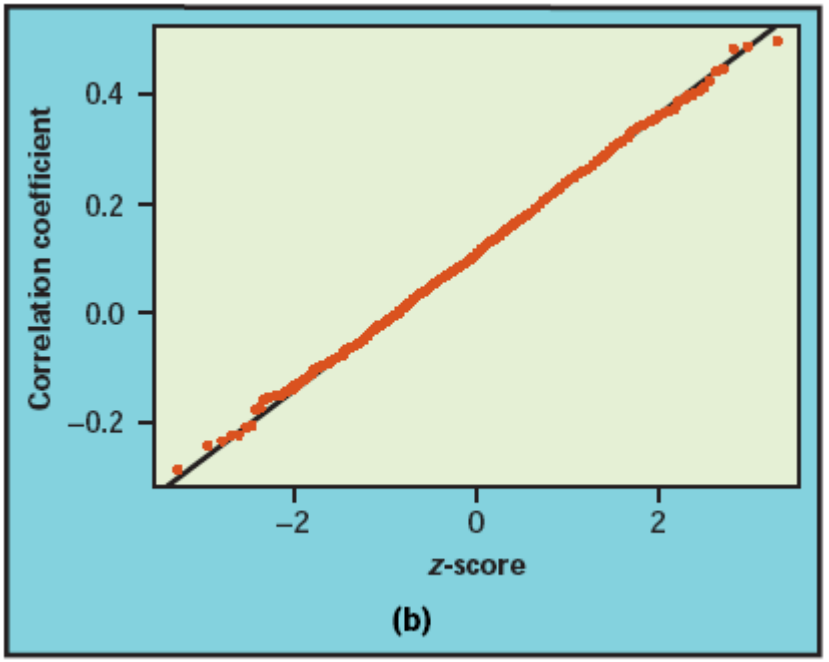
Observed samples	Bootstrap results
	

```
time1=scan("Verizon1_ilec.prn")
n1=length(time1); mean.time1=mean(time1); sd.time1=sd(time1)
time2=scan("Verizon1_clec.prn")
n2=length(time2); mean.time2=mean(time2); sd.time2=sd(time2)
d.means= mean.time1- mean.time1;
d.means; mean.time1; sd.time1; mean.time2; sd.time2;

# bootstrap
NB=1000; y=rep(NA,NB);
for(i in 1:NB){
  x1b=sample(time1,replace=TRUE)
  x2b=sample(time2,replace=TRUE)
  y[i]=mean(x1b)-mean(x2b)
  }
# bootstrap results
hist(y);
mean.y=mean(y); sd.y=sd(y)

   …
```

- If we are dealing with a correlation coefficient we must consider that we have only one sample where each observation is given by a pair of values (one for each variable). The resampling has to be done accordingly.

  Example (see case 18.3) – Correlation between batting average and salary for baseball players

The sample	Bootstrap results
	


Correlation coefficient
0.4
0.2
0.0
–0.2
–2
0
2
z-score
(b)

```
> rm(list=ls(all=TRUE))
> rm(list=ls(all=TRUE))
> baseball=read.table("C:/…/baseball.prn",header=TRUE)
>
> x1=baseball$Salary; x2=baseball$Average;
> mean1=mean(x1); sd1=sd(x1); mean2=mean(x2); sd2=sd(x2);
correla=cor(x1,x2); n=length(x1);
>
> mean1; mean2; correla; sd1; sd2;
[1] 2796046
[1] 0.25704
[1] 0.1067575
[1] 2705920
[1] 0.037434
>
> # bootstrap
> NB=10000; y=rep(NA,NB);
> for(i in 1:NB){
+   j=sample(1:n,replace=T);
+   x1b=x1[j]; x2b=x2[j];
+   y[i]=cor(x1b,x2b)
+   }
```

```
> # bootstrap results
> mean.y=mean(y); sd.y=sd(y)
> sk.y=(1/(NB-1))*sum((y-mean(y))^3)/(sd.y^3)
> kurt.y=(1/(NB-1))*sum((y-mean(y))^4)/(sd.y^4)-3
> mean.y; sd.y; sk.y; kurt.y; mean.y-correla
[1] 0.1065278
[1] 0.1315791
[1] 0.01257684
[1] -0.06775939
[1] -0.0002297379
>
> ks.test(y,"pnorm",mean=correla,sd=sd.y)

        One-sample Kolmogorov-Smirnov test

data:  y
D = 0.0052, p-value = 0.9493
alternative hypothesis: two-sided

>
```

```
> #Using boot library
> x=cbind(x1,x2)
> library(boot)
> corr(x)# function defined in boot library
[1] 0.1067575
> b.correlation=function(x,i) {
+   data=x[i,]; return(corr(data))
+   }
> out2=boot(x,b.correlation,R=NB)
> out2

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = x, statistic = b.correlation, R = NB)


Bootstrap Statistics :
     original     bias    std. error
t1* 0.1067575 -0.0013724   0.1311099
>
```

6. Permutations tests revisited

Introduction

- The idea of permutation tests is due to Fisher in the 30th of the last century, a long time before bootstrap techniques appear. However the use of computer capacities reinforce the applicability of the test and the similarities with bootstrap are obvious;
- Permutations tests are used to compare 2 groups where the null hypothesis is $H_0 : F = G$, i.e assuming H_0 the two populations are undistinguishable. An example of such situation if when we want to test a learning process, a new medical treatment, …
- The idea is to define a test statistic whose value is A filosofia de teste insere-se nos testes de significância (que também se devem a Fisher). A ideia é definir uma estatística de teste que se afaste de determinado valor à medida que H_0 perde credibilidade. Por exemplo suponha-se que se recolheu uma amostra casual de cada população e que a estatística de teste é dada pela diferença das médias amostrais. Trata-se de testar se a diferença entre estas duas estatísticas se pode dever apenas à aleatoriedade inerente ao processo de amostragem ou se é mais razoável assumir que se deve à não verificação de H_0.

- Para um teste de significância, o raciocínio descreve-se da seguinte forma:
  - Estabelecer a hipótese nula.
  - Encontrar uma estatística adequada para efectuar o teste.
  - Obter a distribuição por amostragem da referida estatística, considerando a hipótese H_0 como certa;
  - Localizar o valor observado da estatística em termos da distribuição por amostragem. Caso a observação de um valor tão ou mais extremo do que aquele que se observou (o valor-p ou "p-value") tenha uma probabilidade pequena, rejeita-se que tal se deva apenas aos efeitos aleatórios.

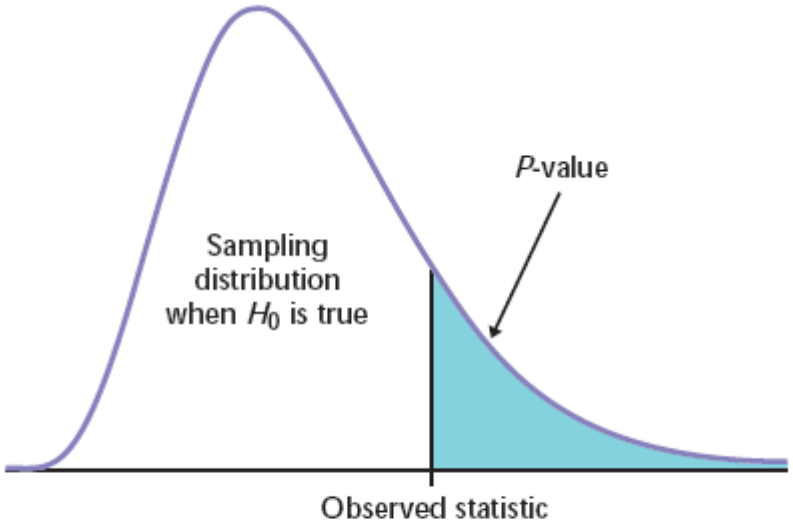


FIGURE 18.19 The *P*-value of a statistical test is found from the sampling distribution the statistic would have if the null hypothesis were true. It is the probability of a result at least as extreme as the value we actually observed.

- Para os testes com $H_0 : F = G$ apenas se consegue obter a distribuição por amostragem da estatística de teste num número restrito de situações, habitualmente ligadas à distribuição normal dos universos ou quando se pode recorrer ao Teorema do Limite Central. A técnica proposta por Fisher permite ultrapassar este problema;

Testes baseados em permutações

- O raciocínio pode então descrever-se da seguinte forma, assumindo que se observaram n_1 elementos do primeiro grupo e n_2 do segundo ($n = n_1 + n_2$):
  - Se a hipótese nula é verdadeira, as populações confundem-se e pode-se atribuir indiferentemente um grupo a qualquer elemento observado;
  - Existem $m = \binom{n}{n_1}$ maneiras de formar os grupos (assumindo que a ordem das observações não é relevante para o cálculo da estatística de teste) tendo cada ordenação particular uma probabilidade de $1/m$. Note-se que cada ordenação particular é obtida permutando os n elementos observados.
  - O valor-p será dado como a probabilidade de uma permutação escolhida aleatoriamente originar um valor tão ou mais anómalo para a estatística de teste (dado H_0) do que o valor observado.
  - Em termos práticos, estima-se o valor-p com base num número elevado de permutações, não sendo necessário percorrer as m situações possíveis.
- Procedimento prático para testar $H_0 : F = G$
  - Escolher a estatística de teste;
  - Gerar *B* permutações ($B \leq m$) e para cada permutação calcular o valor da estatística de teste;

- Estimar o valor-p pela proporção de permutações que originam um valor da estatística de teste tão ou mais "anómalo" do que o valor observado.

Exemplo 18.12 - Observaram-se 44 estudantes com base nos quais se construíram aleatoriamente dois grupos. O primeiro com 21 elementos seguiu um método inovador de ensino enquanto o segundo (23 elementos) seguiu o método convencional. Observaram-se os resultados ao fim de determinado período.

1º ponto → definir o teste $H_0 : \mu_1 = \mu_2$ contra $H_1 : \mu_1 > \mu_2$

se se rejeitar a hipótese nula assume-se que o novo método é melhor.

Também se pode formalizar como $H_0 : \mu_1 \leq \mu_2$ contra $H_1 : \mu_1 > \mu_2$

2º ponto → encontrar uma estatística de teste adequada: $\overline{X}_1 - \overline{X}_2$

3º ponto → sem hipóteses adicionais sobre o universo não se conhece a distribuição por amostragem da estatística e a dimensão das amostras não é de molde a garantir que a aproximação à normal se mostre adequada. No entanto, caso H_0 seja verdadeira, assume-se que as 2 populações se confundem ($H_0 : F = G$). O processo será então:

- Escolher aleatoriamente 21 dos 44 elementos para formarem o grupo 1 e os restantes 23 constituirão o grupo 2.

Esta re-amostragem é feita com reposição, isto é, lida-se sempre com os 44 elementos, apenas se procedendo a arrumações (permutações) diferentes. Para cada permutação, calcular o valor da estatística de teste.

- Repetir o ponto anterior um grande número de vezes (idealmente percorrer-se-iam todas as permutações possíveis). O conjunto dos valores assim obtidos para a estatística de teste designa-se como a distribuição das permutações.
- Obter o valor-p associado a este teste como sendo a proporção de casos na distribuição das permutações em que se obtém um valor menos credível do que aquele que se observou.
- Ilustrar o processo.

Distribuição por amostragem supondo H_0 verdadeira

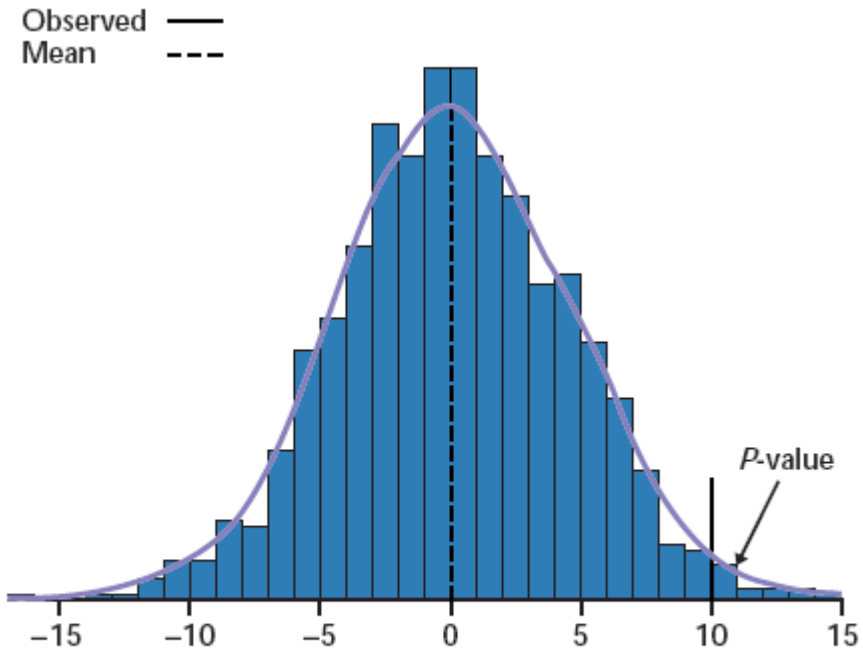


FIGURE 18.21 The permutation distribution of the statistic $\overline{x}_{\text{treatment}} - \overline{x}_{\text{control}}$ based on the DRP scores of 44 students. The observed difference in means, 9.954, is in the right tail.

Vantagens dos testes baseados em permutações

- Validade fora do quadro da distribuição normal ou de um enquadramento onde existam resultados teóricos.
- Poder validar indirectamente a hipótese de normalidade por comparação dos valores obtidos.

Testes de permutações e bootstrap

- Os testes baseados em permutações também recorrem a um processo de re-amostragem, tal como o bootstrap. Duas diferenças importantes:
  - As permutações correspondem a um processo **sem** reposição;
  - A reamostragem é feita de acordo com a hipótese nula;
- Tal como no bootstrap têm-se 2 fontes de variabilidade nos testes de permutações:
  - Processo de amostragem inicial
  - Processo de reamostragem (poder-se-ia eliminar fazendo $B=m$)

Outros testes que podem basear-se no mesmo princípio:

- Amostras emparelhadas
  - Assume-se que se está a testar a ausência de efeito de um "tratamento" em amostras emparelhadas (o mesmo elemento é observado antes e depois do tratamento);

- A re-amostragem consiste na troca (ou não) da ordem de cada par. No quadro da hipótese nula (não produziu efeito), as situações podem ser permutadas;
- Exemplo (18.15)

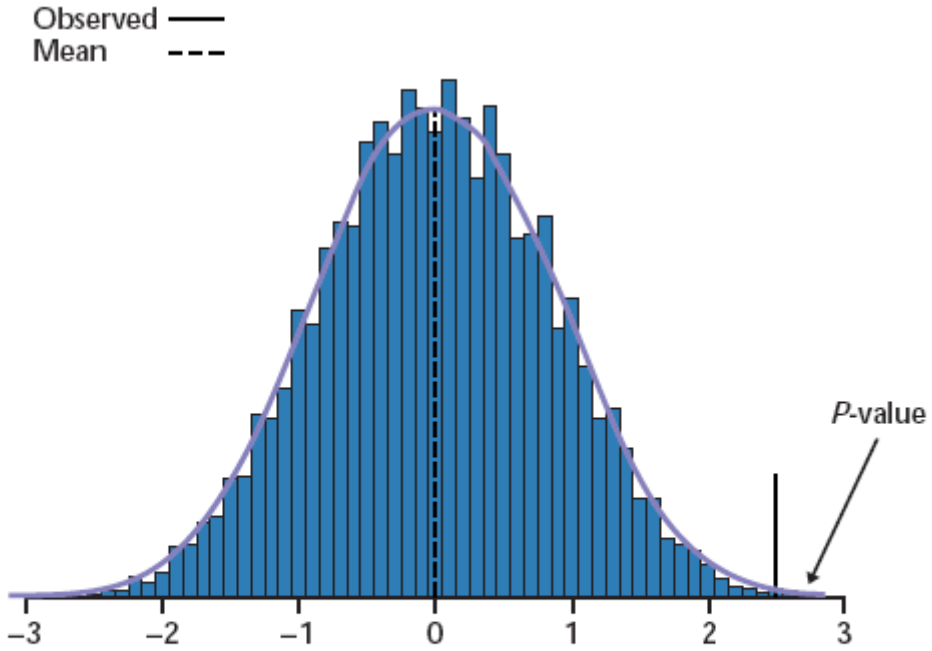


FIGURE 18.23 The permutation distribution for the mean difference (score after instruction minus score before instruction) from 9999 paired resamples from the data in Table 7.2. The observed difference in means, 2.5, is in the right tail.

- Correlação
  - Assume-se que se está a testar a ausência de correlação (hipótese nula) entre duas variáveis;
  - A re-amostragem consiste em manter a ordenação de uma das variáveis e permutar a outra.

7. Comentários finais

- O bootstrap, bem como os testes baseados em permutações, são metodologias que se aplicam em muitos campos da inferência estatística.
- O processo de re-amostragem tem de ser adaptado a cada situação específica e pode ser bastante mais complicado do que os exemplos que se viram: Por exemplo, para amostras cronológicas (que habitualmente exibem um padrão de autocorrelação - não se verifica independência entre as observações) a re-amostragem proposta levaria a destruir este padrão. Recorre-se então ao "block-bootstrap".

- Existe muito software com módulos previstos para efectuar bootstrap. Refira-se o *S-Plus* ou o *R* onde, para além das funções de origem, existem imensas funções disponíveis na Internet.
- Bootstrap não paramétrico *versus* bootstrap paramétrico.

8. Bootstrap e regressão linear

- Tema muito vasto e de actualidade do qual apenas se vai ver uma breve introdução
- Modelo de dados seccionais onde se verificam as hipóteses "classícas"

$$E(Y \mid X) = X\beta$$

$$Y = X\beta + U \text{ com } E(U \mid X) = \vec{0} \text{ e } \operatorname{cov}(U \mid X) = \sigma^2 I$$

$$\hat{\beta} = (X'X)^{-1} X'Y \quad E(\hat{\beta} \mid X) = \beta \quad \operatorname{cov}(\hat{\beta} \mid X) = \sigma^2 (X'X)^{-1}$$

$$\hat{Y} = X\beta = X(X'X)^{-1}X'Y = HY$$

com $H = X(X'X)^{-1}X'$, H é simétrica e idempotente.

$$\hat{U} = Y - \hat{Y} = (I - H)Y \quad E(\hat{U} \mid X) = E\big((I - H)Y \mid X\big) = (I - H)X\beta = \vec{0}$$
$$\operatorname{cov}(\hat{U} \mid X) = \operatorname{cov}\big((I - H)Y \mid X\big) = (I - H)\operatorname{cov}(Y \mid X)(I - H)$$
$$= \sigma^2 (I - H)$$

- logo é fácil verificar que $r_j = \dfrac{\hat{U}_j}{\sqrt{1 - h_{jj}}}$ é uma variável aleatória de média e variância (condicionadas por *X*) iguais à média e variância de U_j (h_{jj} é elemento da diagonal principal de H)
- Vamo-nos restringir à regressão linear simples para maior simplicidade de exposição, isto é, $E(Y_j \mid X_j) = \beta_1 + \beta_2 X_j,\ j = 1,2,\cdots,n$

Alternativa 1 – Re-amostragem pelos resíduos

- Assumindo que o modelo se encontra bem especificado, isto é, que $Y_j = \beta_1 + \beta_2 X_j + U_j$, e considerando os X_j como fixos, a re-amostragem vai incidir apenas nos Y_j. Idealmente pretenderíamos gerar novas amostras de U_j o que não é possível dado que esta variável não é observável. Recorre-se então a r_j.
- O algoritmo é então:

- Estimar o modelo inicial e obter, para além das estimativas dos parâmetros de interesse, $\hat{\beta}_1$ e $\hat{\beta}_2$ nomeadamente, os valores de r_j, $j = 1,2,\cdots,n$.
- Para cada réplica bootstrap, $b = 1,2,\cdots,B$
  - Escolher os valores r_j^* por tiragem aleatória com reposição nos r_j.
  - Construir $Y_j^* = \hat{\beta}_1 + \hat{\beta}_2 X_j + r_j^*$
  - Estimar o modelo $E(Y_j^* \mid X_j) = \beta_1 + \beta_2 X_j$ obtendo-se as estatísticas de interesse, como por exemplo β_1^b ou β_2^b
- Geradas as B amostras obtém a aproximação desejada à distribuição por amostragem.

Alternativa 2 – Re-amostragem pelos pares (Y_j, X_j)

- Considera-se que a amostra observada resulta de uma escolha (de forma independente) de pares aleatórios (Y_j, X_j) no universo;

- Assim sendo, o bootstrap vai assentar na escolha, com reposição de *n* pares aleatórios dentro da amostra observada, o que origina o seguinte algoritmo:
  - Para cada réplica bootstrap, $b = 1,2,\cdots,B$
    - Escolher n pares (Y_j^*, X_j^*) por tiragem aleatória com reposição nos pares observados (Y_j, X_j).
    - Estimar o modelo $E(Y_j^* \mid X_j) = \beta_1 + \beta_2 X_j$ obtendo-se as estatísticas de interesse, como por exemplo β_1^b ou β_2^b
  - Geradas as B amostras obtém a aproximação desejada à distribuição por amostragem.

Comentários

- A segunda alternativa é mais robusta do que a primeira a violação das hipóteses no modelo. Em contrapartida a primeira será mais eficiente caso estas hipóteses se encontrem verificadas. Nomeadamente a primeira alternativa necessita da homogeneidade na variância da variável residual.
- Note-se também que a segunda alternativa origina amostra com processos de amostragem diferentes em ordem aos X_j com consequências nomeadamente no enviesamento dos estimadores bootstrap dos β_j. Habitualmente este enviesamento não tem significado prático.

Exemplo (Davison and Hinkley) Considerem-se 62 observações referentes ao peso do cérebro e ao peso corporal de outras tantas espécies de mamíferos. O nosso propósito é analisar uma teoria que defende que existe uma relação linear entre o valor esperado do logaritmo do peso cerebral e o logaritmo do peso corporal (elasticidades constantes) nas espécies de mamíferos.

ANEXOS – Programas TSP

```
? Programa Bootstrap 1 – Clientes da Verizon
read(file=verizon1.xls);  ? variaveis Time e Group (ilec ou clec) com
                          ? 1664+23 obs ordenadas(ILEC e depois CLEC)
set n=1664;       ? Nº e observações referentes a ILEC
set nt=1687;      ? Nº total de observações

? Apenas se vao estudar os clientes da Verizon
smpl 1 n;
msd time;
hist time;
set media=@mean; ? estatistica de interesse
set s=@STDDEV;   ? desvio padrão corrigido para o IC "normal"

? Prepara o bootstrap
set nr=1000;
smpl 1 nr; Y=0; smpl 1 n;
? Ciclo bootstrap
do i=1 nr;
  random (draw=time) x;
  msd(silent) x;
  set Y(i)=@mean;
enddo;
? Conclui o bootstrap
smpl 1 nr;
hist y;     ? Distribuição por amostragem
msd (all) y;
set SEboot=@stddev;             ? Estimativa bootstrap do erro padrão
set Biasboot=@mean-media; ? Estimativa bootstrap da média
print seboot biasboot;
? intervalos de confiança pelo método dos percentis
sort y;
```

```
set p1=y(25); set p2=y(50); set p3=y(950); set p4=y(975);
print p1 p2 p3 p4;
? intervalos de confiança pelo método t
cdf(t,df=1663,inverse) 0.05 t1; ? com estas opções dá o valor positiva TWOTAILS
cdf(t,df=1663,inverse) 0.10 t2;
print t1 t2;
set p1=media-t1*SEboot;
set p2=media-t2*SEboot;
set p3=media+t2*SEboot;
set p4=media+t1*SEboot;
print p1 p2 p3 p4;
? intervalos de confiança assumindo a normalidade
set p1=media-t1*S/sqrt(n);
set p2=media-t2*S/sqrt(n);
set p3=media+t2*S/sqrt(n);
set p4=media+t1*S/sqrt(n);
print p1 p2 p3 p4;
```

? Programa bootstrap 2 – Diferença de médias em amostras independentes

```
read(file=verizon1.xls);  ? variaveis Time e Group (ilec ou clec) com
                          ? 1664+23 obs ordenadas(ILEC e depois CLEC)
set n=1664;       ? Nº e observações referentes a ILEC
set nt=1687;      ? Nº total de observações
set na=n+1;     ? Primeira observação referente a CLEC

smpl 1 n; msd time; set media1=@mean;
smpl na nt; msd time; set media2=@mean;

? Prepara bootstrap
set nr=1000;
smpl 1 nr; Y=0;
? Ciclo bootstrap
supres smpl;
do i=1 nr;
```

```
  smpl 1 n;
  random (draw=time) x; ? Tiragem no primeiro grupo
  msd(silent) x;
  set x1=@mean;
  smpl na nt;
  random (draw=time) x; ? Tiragem no segundo grupo
  msd(silent) x;
  set x2=@mean;
  set y(i)=x1-x2;       ? Guarda a diferença de médias
enddo;
nosupres smpl;
? Conclui bootstrap
smpl 1 nr;
msd (all) y;
set SEboot=@stddev;
set Biasboot=@mean-(media1-media2);
hist y;
? método dos percentis
sort y;
set o1=round(nr*0.025); set o2=round(nr*0.050);
set o3=round(nr*0.950); set o4=round(nr*0.975);
set p1=y(o1); set p2=y(o2); set p3=y(o3); set p4=y(o4);
print seboot biasboot;
print p1 p2 p3 p4;
? método standard
set n1=n-1;
cdf(inv,t,df=n1,lowtail) 0.025 o1;
cdf(inv,t,df=n1,lowtail) 0.050 o2;
cdf(inv,t,df=n1,lowtail) 0.950 o3;
cdf(inv,t,df=n1,lowtail) 0.975 o4;
set p1=(media1-media2)+o1*SEboot;
set p2=(media1-media2)+o2*SEboot;
set p3=(media1-media2)+o3*SEboot;
set p4=(media1-media2)+o4*SEboot;
print p1 p2 p3 p4;
```

```
? Programa bootstrap 3 – Coeficiente de correlação
? Leitura dos dados e análise preliminar
read(file=baseball.xls);  ?variaveis Name Salary Average com  50 obs
msd (corr) salary average;
set correla=@corr(2,1);
print correla;
set n=50;

? Bootstrap
set nr=1000;
smpl 1 nr; Y=0; smpl 1 n;
mmake Z salary average;   ? constroi matriz para tiragem do par
do i=1 nr;
  random (draw=Z) X;     ? tira o par de variáveis
  unmake X X1 X2;        ? Desfaz a matriz
  msd(silent,corr) X1 X2;
  set Y(i)=@corr(2,1);
enddo;
smpl 1 nr;
msd (all) y;
set SEboot=@stddev;
set Biasboot=@mean-correla;
hist y;
? método dos percentis
sort y;
set o1=round(nr*0.025); set o2=round(nr*0.050);
set o3=round(nr*0.950); set o4=round(nr*0.975);
set p1=y(o1); set p2=y(o2); set p3=y(o3); set p4=y(o4);
print seboot biasboot;
print p1 p2 p3 p4;
? método standard
set n1=n-1;
cdf(inv,t,df=n1,lowtail) 0.025 o1;
cdf(inv,t,df=n1,lowtail) 0.050 o2;
cdf(inv,t,df=n1,lowtail) 0.950 o3;
```

```
cdf(inv,t,df=n1,lowtail) 0.975 o4;
set p1=correla+o1*SEboot;
set p2=correla+o2*SEboot;
set p3=correla+o3*SEboot;
set p4=correla+o4*SEboot;
print p1 p2 p3 p4;
```

```
? Programa bootstrap 4 - Amostras emparelhadas
? Leitura dos dados e tratamento preliminar
read(file=perm2.xls);  ? variaveis Pretest Posttest Gain 20 obs emparelhadas
set n=20;
X=posttest-pretest;      ? corresponde à variável Gain
print pretest posttest X;
msd X; set media=@mean;

? bootstrap
set nr=1000;
smpl 1 nr; Y=0; smpl 1 n;
mmake W pretest posttest;  ? A tiragem vai ser por pares
supres smpl;
do i=1 nr;
  random(draw=W)Z;            ? tiragem do par
  unmake Z Z1 Z2;       ? desfaz o par
  ZZ=Z2-Z1;
  msd(silent) ZZ;
  set Y(i)=@mean;
enddo;
nosupres smpl;
smpl 1 nr;
msd (all) y;
set SEboot=@stddev;
set biasboot=@mean-media;
? método dos percentis
sort y;
```

```
set o1=round(nr*0.025); set o2=round(nr*0.050);
set o3=round(nr*0.950); set o4=round(nr*0.975);
set p1=y(o1); set p2=y(o2); set p3=y(o3); set p4=y(o4);
print seboot biasboot;
print p1 p2 p3 p4;
? método standard
set n1=n-1;
cdf(inv,t,df=n1,lowtail) 0.025 o1;
cdf(inv,t,df=n1,lowtail) 0.050 o2;
cdf(inv,t,df=n1,lowtail) 0.950 o3;
cdf(inv,t,df=n1,lowtail) 0.975 o4;
set p1=media+o1*SEboot;
set p2=media+o2*SEboot;
set p3=media+o3*SEboot;
set p4=media+o4*SEboot;
print p1 p2 p3 p4;


? Programa bootstrap 5 - Bootstrap e regressão (ver bootstrap 3)
? leitura dos dados e primeira análise
read(file=baseball.xls);  ? variaveis Name Salary Average com  50 obs
set n=50;
olsq(hi) salary c average;
set beta1chap=@coef(1);   ? para a re-amostragem pelos resíduos
set beta2chap=@coef(2);
r=@res/@hi;

? Bootstrap
set nr=1000;
smpl 1 nr; beta2a=0; beta2b=0;
? re-amostragem pelas observações (o mais adequado aqui)
smpl 1 n;
mmake Z salary average; ? Para tirar pares de valores
do i=1 nr;
  random (draw=Z) X;
  unmake X Y X2;  ? Desfaz a estrutura matricial
```

```
  olsq(silent) Y C X2;
  set beta2b(i)=@coef(2);
enddo;
smpl 1 nr;
msd (all) beta2b;
set SEboot=@stddev;
set Biasboot=@mean-beta2chap;
hist beta2b;
? IC pelo método dos percentis
sort beta2b;
set o1=round(nr*0.025); set o2=round(nr*0.050);
set o3=round(nr*0.950); set o4=round(nr*0.975);
set p1=beta2b(o1); set p2=beta2b(o2); set p3=beta2b(o3); set p4=beta2b(o4);
print seboot biasboot;
print p1 p2 p3 p4;

? re-amostragem pelos resíduos
smpl 1 n;
do i=1 nr;
  random (draw=r) rb;
  Yb=beta1chap+beta2chap*average+rb;      ? Gera Yb
  olsq(silent) Yb C average;
  set beta2a(i)=@coef(2);
enddo;
smpl 1 nr;
msd (all) beta2a;
hist beta2a;
? IC pelo método dos percentis
sort beta2a;
set p1=beta2a(o1); set p2=beta2a(o2); set p3=beta2a(o3); set p4=beta2a(o4);
print p1 p2 p3 p4;


? Programa perm1 - Tratamento e controle
? Leitura dos dados e tratamento inicial
read(file=perm1.xls);  ? variaveis Score Group 21 tratamento + 23 controlo
```

```
smpl  1 21; msd score; set m1=@mean;
smpl 22 44; msd score; set m2=@mean;
set teste = m1-m2;        ? valor observado da estatística de teste
print teste;

? Permutações
set nr=10000;
smpl 1 nr; Y=0;
supres smpl;
smpl 1 44;
x=score;
do i=1 nr;
  smpl 1 44;
  random Z;        ? A distribuição é indiferente desde que seja contínua
  sort Z X;
  smpl  1 21; msd(silent) X; set m1=@mean;
  smpl 22 44; msd(silent) X; set m2=@mean;
  set Y(i)=m1-m2;
enddo;
nosupres smpl;
smpl 1 nr;
msd (all) y;
Z=(y>=teste);
msd(silent) Z;
set valorp=@mean;
print teste valorp;
```

? programa perm2 - Amostras emparelhadas

```
read(file=perm2.xls);  ? variaveis Pretest Posttest Gain 20 obs
                       ? emparelhadas
smpl 1 20;
X=posttest-pretest;
print pretest posttest;
```

```
msd X;
set teste = @mean;
print teste;

set nr=9999;
smpl 1 nr; Y=0;
smpl 1 20;
supres smpl;
do i=1 nr;
  random Z;
  X1=Pretest*(Z>=0)+Posttest*(Z<0);
  X2=Pretest*(Z<0)+Posttest*(Z>=0);
  W=X2-X1;
  msd(silent) W;
  set Y(i)=@mean;
enddo;
nosupres smpl;
smpl 1 nr;
msd (all) y;
Z=(y>=teste);
msd(silent) Z;
set valorp=@mean;
print teste valorp;
```

? **Programa permutações 3 – Testar a corr. nula nos jogadores baseball**

```
read(file=baseball.xls);  ? variaveis Name Salary Average com  50 obs
msd (corr) salary average;
set correla=@corr(2,1);
print correla;
set n=50;

? bootstrap
set nr=1000;
smpl 1 nr; Y=0; smpl 1 n;
```

```
do i=1 nr;
  random (draw=average) X;
  msd(silent,corr) Salary X;
  set Y(i)=@corr(2,1);
enddo;
smpl 1 nr;
msd (all) y;
hist y;
Z=(y>=correla);
msd(silent) Z;
set valorp=@mean;
print correla valorp;
```