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).

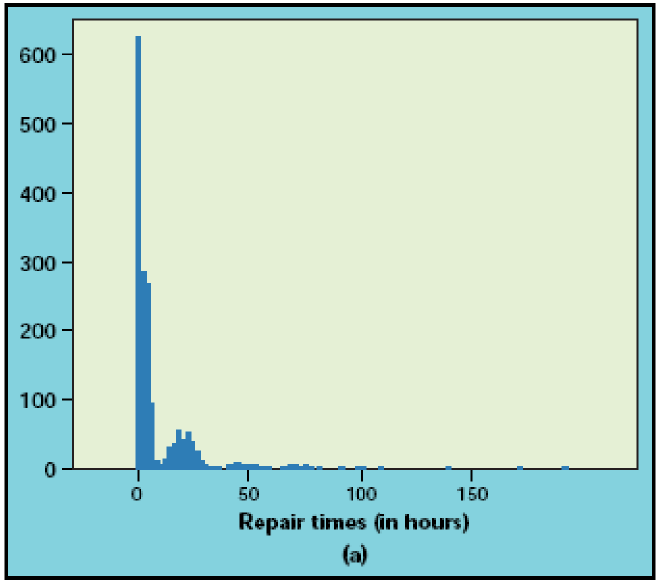
We begin our analysis by focusing on Verizon's own customers. A random sample of 1664 repair times has been observed.

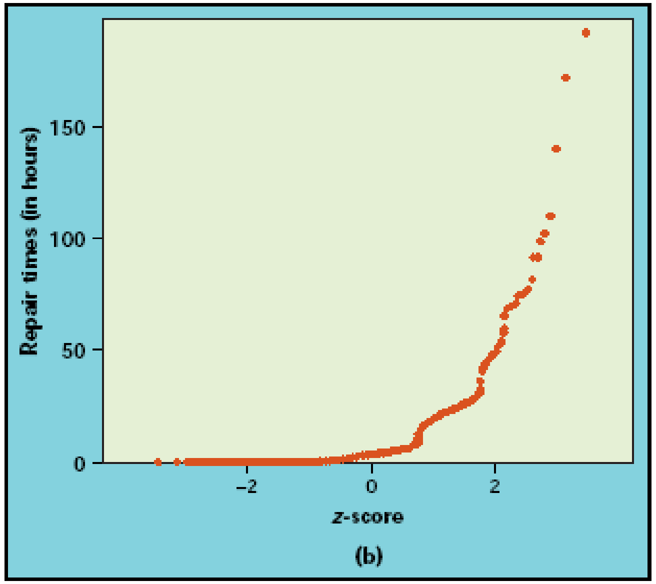
Goal $\rightarrow$ To get some insight about the expected value of a repair time in the population, μ, namely to determine a 95% confidence interval for this parameter. The first step to get an answer to this problem is to look at the sampling distribution of $\overline{X}$.

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

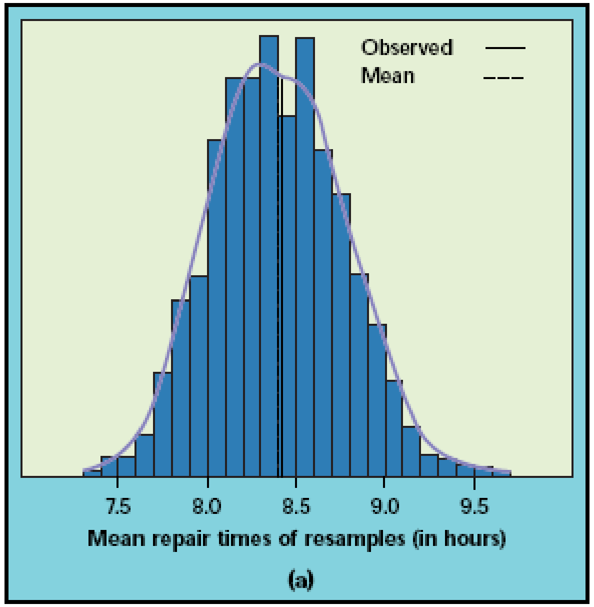
Steps 1 and 2	Step 3
$\begin{array}{c}(x_1,x_2,\cdots,x_n)\\ \text{original sample}\rightarrow\begin{cases}\text{bootstrap sample 1} & \rightarrow\ t_1^* \\ \text{bootstrap sample 2} & \rightarrow\ t_2^* \\ \cdots & \\ \text{bootstrap sample } B & \rightarrow\ t_B^*\end{cases}\\ \downarrow\\ t_{obs}\end{array}$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

Bootstrap applied to repair times for Verizon customers

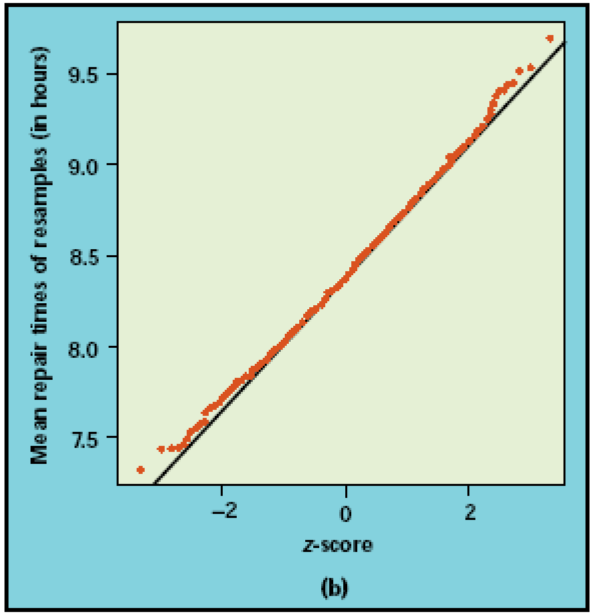
Observed sample and bootstrap distribution

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

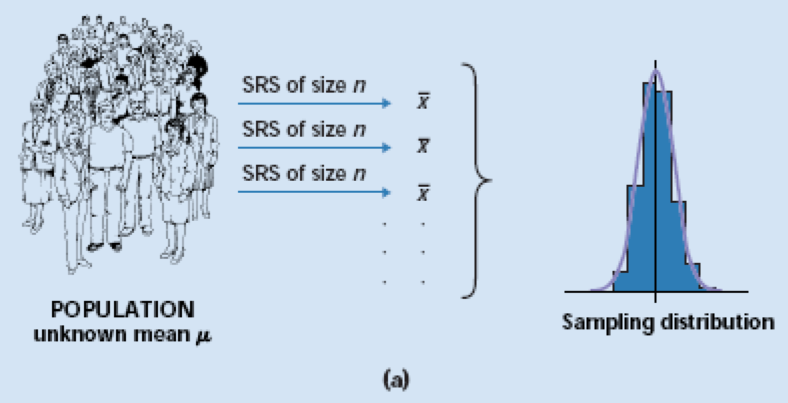
```
time1=scan("C:/Users/joaoas/Documents/My Dropbox/Risk models
slides/Verizon1_ilec.prn")
n1=length(time1)
mean.time1=mean(time1); sd.time1=sd(time1)
sk_nc=(1/(n1-1))*sum((time1-mean.time1)^3)/(sd.time1^3)
kurt_nc=(1/(n1-1))*sum((time1-mean.time1)^4)/(sd.time1^4)-3
mean.time1; sd.time1; sk_nc; kurt_nc; 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

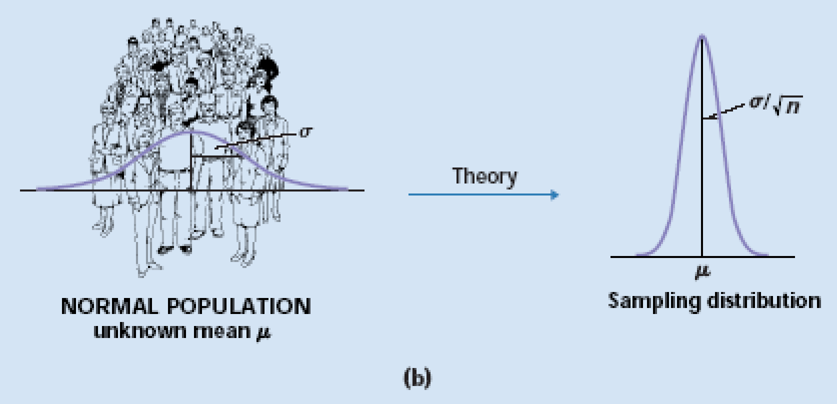
```
library(boot)
b.mean=function(x,i) {
  data=x[i]; return( mean(data))
  }
out1=boot(time1,b.mean,R=1001)
out1
```

2 . Comparing bootstrap with other approaches

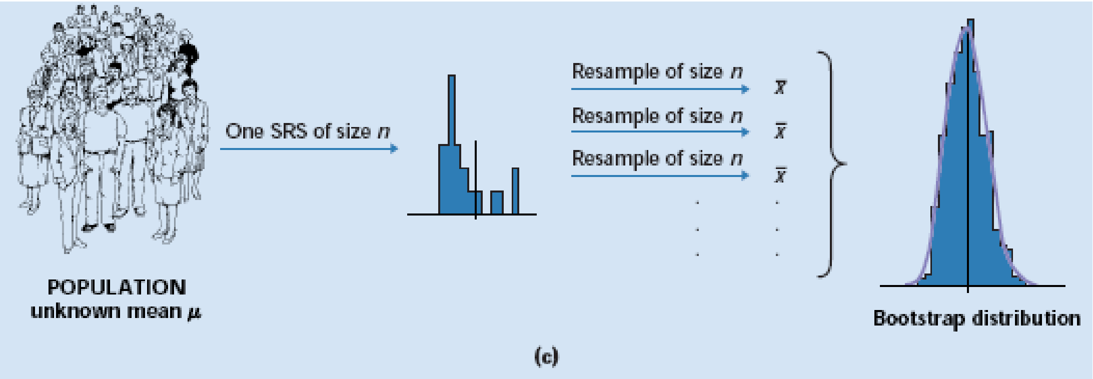
Simulation



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 *Loss Models* - Example 13.4 (4th edition) or 15.4 (3rd edition))

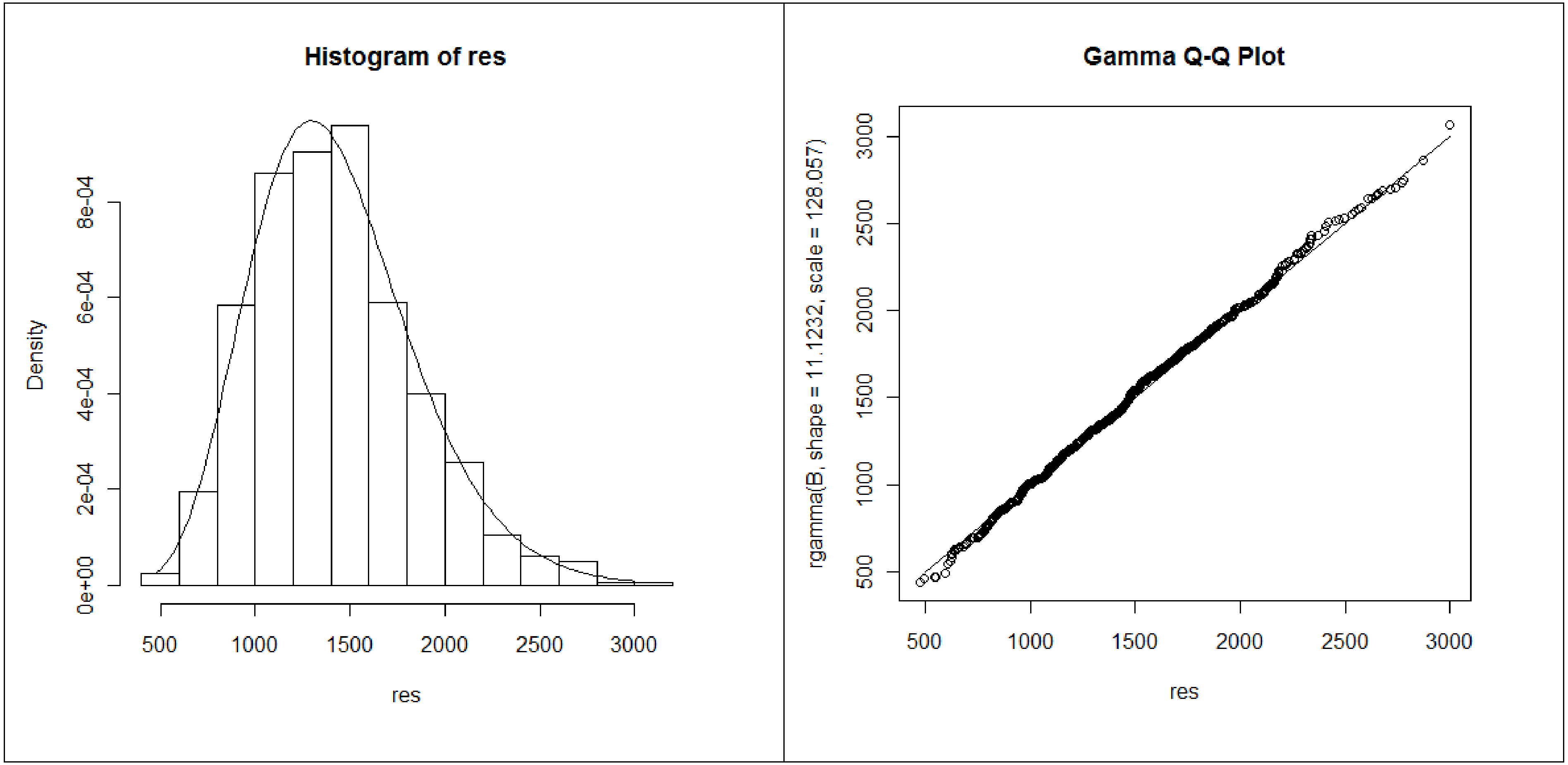
A parametric bootstrap approach can be designed resampling from a Gamma distribution with parameters $\hat{\alpha}=0.55616$ and $\hat{\theta}=2561.14$ instead of resampling from the observed sample (non-parametric bootstrap)

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 $\left(\overline{x}_1,\overline{x}_2,\cdots,\overline{x}_B\right)$ 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
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: $\overset{\wedge}{bias}_T = \overline{t}^* - t_{obs}$ (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")
```

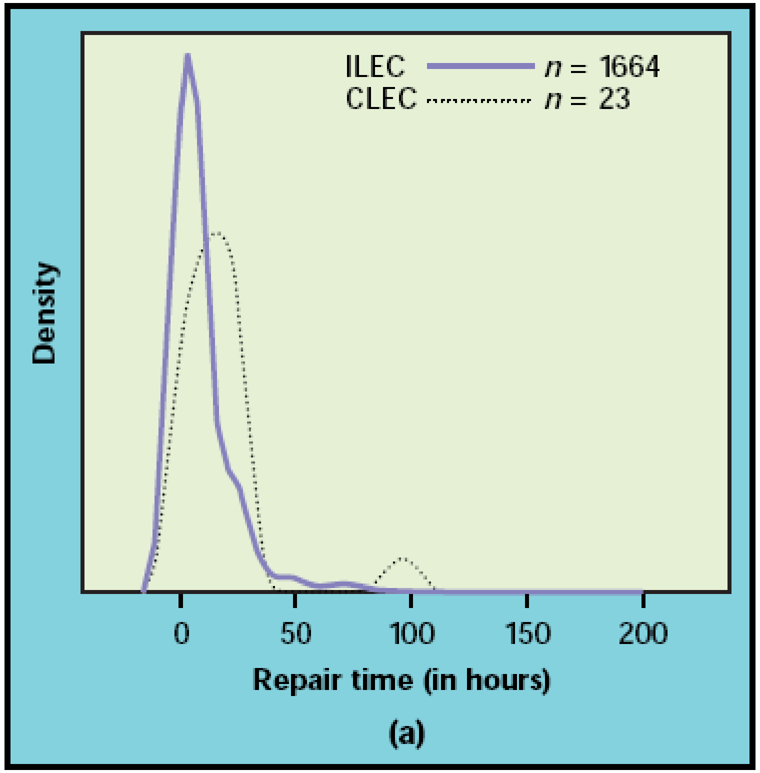
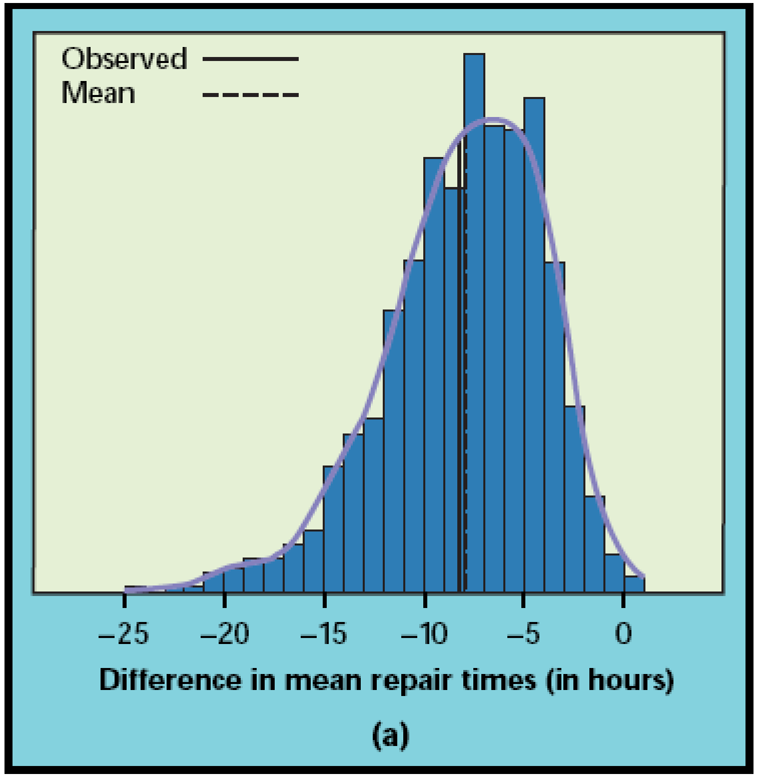
- More efficient confidence intervals can be determined:
  - "bootstrap Bias-Corrected and Accelerated" (BCa): To overcome the effects of populations (and samples) highly asymmetric we can improve the result by choosing non symmetrical percentiles (see Efron e Tibshirami)
  - "bootstrap tilting": The 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.

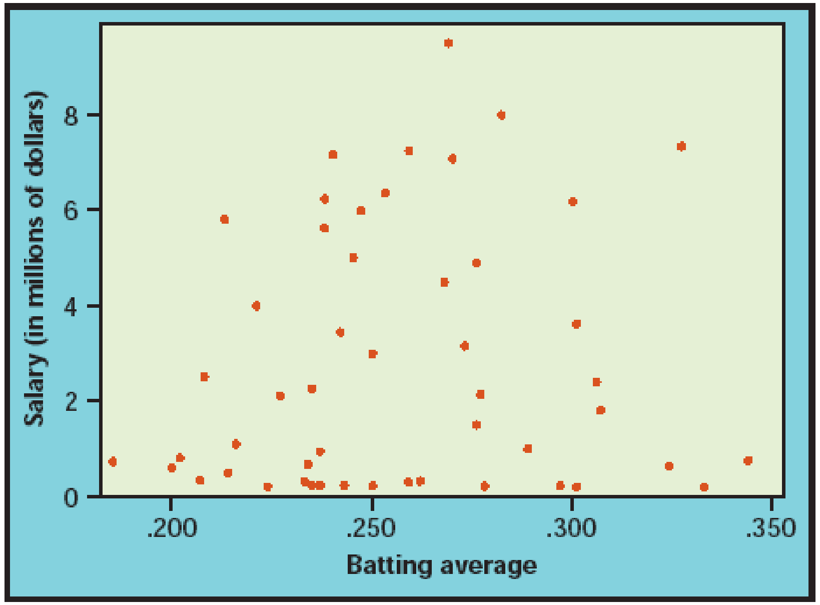
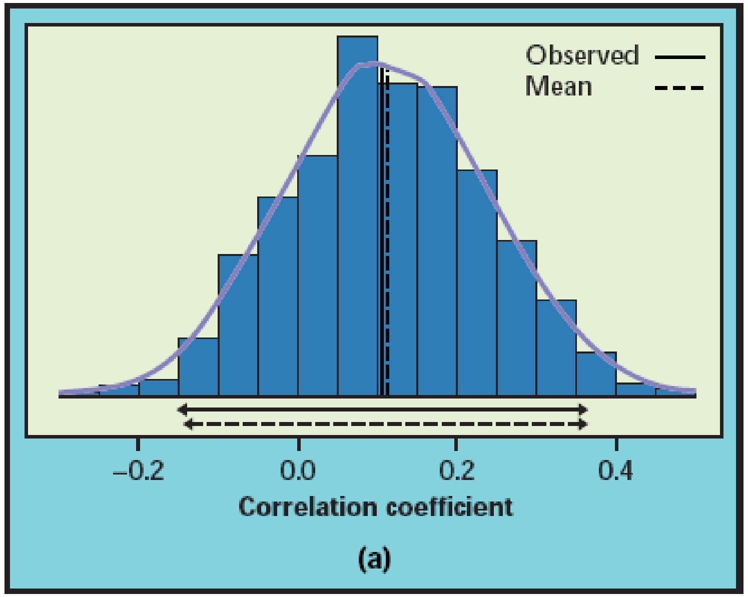
Observed samples	Bootstrap results
	

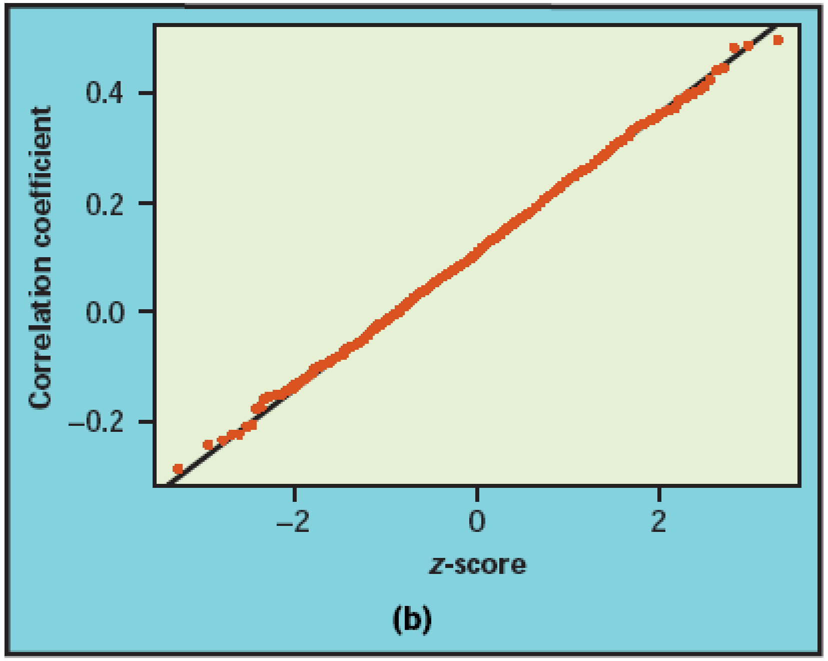
```
time1=scan("Verizon1_ilec.prn")
n1=length(time1); mean.time1=mean(time1); sd.time1=sd(time1)
time2=scan("Verizon1_clec.prn")
n2=length(time2); mean.time2=mean(time2); sd.time2=sd(time2)
d.means= mean.time1- mean.time1;
d.means; mean.time1; sd.time1; mean.time2; sd.time2;

# 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
	



(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
```