Resampling Techniques and their Application

-Class 2-

Frank Konietschke

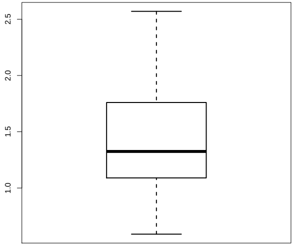
Institut für Biometrie und Klinische Epidemiologie

Charité - Universitätsmedizin Berlin, Berlin

frank.konietschke@charite.de

Motivation and Examples-II

The diameter of cork of a Champagne bottle is supposed to be 1.5 cm. If the cork is either too large or too small, it will not fit in the bottle. The manufacturer measures the diameter in a random sample of **n=36** bottles and obtains:

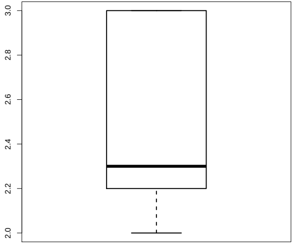



```
x=c(
0.59, 1.23, 1.00, 0.84, 0.88, 1.71,
1.81, 1.84, 2.03, 1.39, 1.30, 1.31,
1.96, 1.33, 2.57, 1.19, 1.01, 2.06,
1.32, 1.55, 1.28, 0.93, 1.63, 1.24,
1.83, 1.81, 0.94, 1.46, 1.25, 1.56,
0.61, 0.83, 1.17, 2.24, 1.68, 1.51)
```

- Data Analysis: Confidence interval and t-test

Motivation and Examples-III

A researcher measures the reaction time of $n = 10$ mice to signal pain when a stitch is applied to their tail.




```
x=c(
2.4, 3.0, 3.0, 2.2, 2.2,
2.2, 2.2, 2.8, 2.0, 3.0)
```

- Data Analysis: Confidence interval and t-test

Example

The diameter of cork of a Champagne bottle is supposed to be 1.5 cm. If the cork is either too large or too small, it will not fit in the bottle. The manufacturer measures the diameter in a random sample of n=36 bottles. Is there evidence at 1% level that the true mean diameter has moved away from the target?

- Hypothesis: $H_0 : \mu = 1.5$ vs. $H_1 : \mu \neq 1.5$

```
mx <- mean(x)
vx <- var(x)
T <- sqrt(36)*(mx-1.5)/sqrt(vx)
p <- 2*min(pt(T,35),1-pt(T,35))
```

Example

The diameter of cork of a Champagne bottle is supposed to be 1.5 cm. If the cork is either too large or too small, it will not fit in the bottle. The manufacturer measures the diameter in a random sample of n=36 bottles. Is there evidence at 1% level that the true mean diameter has moved away from the target?

- Hypothesis: $H_0 : \mu = 1.5$ vs. $H_1 : \mu \neq 1.5$
- Estimator: $\widehat{\mu} = 1.4136$

```
mx <- mean(x)
vx <- var(x)
T <- sqrt(36)*(mx-1.5)/sqrt(vx)
p <- 2*min(pt(T,35),1-pt(T,35))
```

Example

The diameter of cork of a Champagne bottle is supposed to be 1.5 cm. If the cork is either too large or too small, it will not fit in the bottle. The manufacturer measures the diameter in a random sample of n=36 bottles. Is there evidence at 1% level that the true mean diameter has moved away from the target?

```
mx <- mean(x)
vx <- var(x)
T <- sqrt(36)*(mx-1.5)/sqrt(vx)
p <- 2*min(pt(T,35),1-pt(T,35))
```

- Hypothesis: $H_0 : \mu = 1.5$ vs. $H_1 : \mu \neq 1.5$
- Estimator: $\widehat{\mu} = 1.4136$
- Standard deviation: $\widehat{\sigma} = 0.46$

Example

The diameter of cork of a Champagne bottle is supposed to be 1.5 cm. If the cork is either too large or too small, it will not fit in the bottle. The manufacturer measures the diameter in a random sample of n=36 bottles. Is there evidence at 1% level that the true mean diameter has moved away from the target?

```
mx <- mean(x)
vx <- var(x)
T <- sqrt(36)*(mx-1.5)/sqrt(vx)
p <- 2*min(pt(T,35),1-pt(T,35))
```

- Hypothesis: $H_0 : \mu = 1.5$ vs. $H_1 : \mu \neq 1.5$
- Estimator: $\widehat{\mu} = 1.4136$
- Standard deviation: $\widehat{\sigma} = 0.46$
- Test statistic: $T = \sqrt{36} * \frac{1.4136 - 1.5}{0.46} = -1.13$

Example

The diameter of cork of a Champagne bottle is supposed to be 1.5 cm. If the cork is either too large or too small, it will not fit in the bottle. The manufacturer measures the diameter in a random sample of n=36 bottles. Is there evidence at 1% level that the true mean diameter has moved away from the target?

```
mx <- mean(x)
vx <- var(x)
T <- sqrt(36)*(mx-1.5)/sqrt(vx)
p <- 2*min(pt(T,35),1-pt(T,35))
```

- Hypothesis: $H_0 : \mu = 1.5$ vs. $H_1 : \mu \neq 1.5$
- Estimator: $\widehat{\mu} = 1.4136$
- Standard deviation: $\widehat{\sigma} = 0.46$
- Test statistic: $T = \sqrt{36} * \frac{1.4136 - 1.5}{0.46} = -1.13$
- p-value: 2 times the area to the **left** of T under the t curve with 35 df. (Here, p=0.27)

Example

The diameter of cork of a Champagne bottle is supposed to be 1.5 cm. If the cork is either too large or too small, it will not fit in the bottle. The manufacturer measures the diameter in a random sample of n=36 bottles. Is there evidence at 1% level that the true mean diameter has moved away from the target?

```
mx <- mean(x)
vx <- var(x)
T <- sqrt(36)*(mx-1.5)/sqrt(vx)
p <- 2*min(pt(T,35),1-pt(T,35))
```

- Hypothesis: $H_0 : \mu = 1.5$ vs. $H_1 : \mu \neq 1.5$
- Estimator: $\widehat{\mu} = 1.4136$
- Standard deviation: $\widehat{\sigma} = 0.46$
- Test statistic: $T = \sqrt{36} * \frac{1.4136-1.5}{0.46} = -1.13$
- p-value: 2 times the area to the **left** of T under the t curve with 35 df. (Here, p=0.27)
- Quality of the estimator? Is the method valid? Is $p = 0.27$ a good estimate?

Example

A researcher measures the reaction time of $n = 10$ mice to signal pain when a stitch is applied to their tail. Compute a 95% confidence interval.

```
x=c(2.4, 3.0, 3.0, 2.2, 2.2,
2.2, 2.2, 2.8, 2.0, 3.0)
mx <- mean(x)
vx <- var(x)
crit <- qt(0.975,9)
lower <- mx-crit/sqrt(10)*sqrt(vx)
upper <- mx+crit/sqrt(10)*sqrt(vx)
```

- Estimator: $\widehat{\mu} = 2.5$

Example

A researcher measures the reaction time of $n = 10$ mice to signal pain when a stitch is applied to their tail. Compute a 95% confidence interval.

```
x=c(2.4, 3.0, 3.0, 2.2, 2.2,
2.2, 2.2, 2.8, 2.0, 3.0)
mx <- mean(x)
vx <- var(x)
crit <- qt(0.975,9)
lower <- mx-crit/sqrt(10)*sqrt(vx)
upper <- mx+crit/sqrt(10)*sqrt(vx)
```

- Estimator: $\widehat{\mu} = 2.5$
- Standard deviation: $\widehat{\sigma} = 0.40$

Example

A researcher measures the reaction time of $n = 10$ mice to signal pain when a stitch is applied to their tail. Compute a 95% confidence interval.

```
x=c(2.4, 3.0, 3.0, 2.2, 2.2,
2.2, 2.2, 2.8, 2.0, 3.0)
mx <- mean(x)
vx <- var(x)
crit <- qt(0.975,9)
lower <- mx-crit/sqrt(10)*sqrt(vx)
upper <- mx+crit/sqrt(10)*sqrt(vx)
```

- Estimator: $\widehat{\mu} = 2.5$
- Standard deviation: $\widehat{\sigma} = 0.40$
- Quantile: $t_9(0.975) = 2.26$

Example

A researcher measures the reaction time of $n = 10$ mice to signal pain when a stitch is applied to their tail. Compute a 95% confidence interval.

```
x=c(2.4, 3.0, 3.0, 2.2, 2.2,
2.2, 2.2, 2.8, 2.0, 3.0)
mx <- mean(x)
vx <- var(x)
crit <- qt(0.975,9)
lower <- mx-crit/sqrt(10)*sqrt(vx)
upper <- mx+crit/sqrt(10)*sqrt(vx)
```

- Estimator: $\widehat{\mu} = 2.5$
- Standard deviation: $\widehat{\sigma} = 0.40$
- Quantile: $t_9(0.975) = 2.26$
- CI: $[2.21; 2.79]$

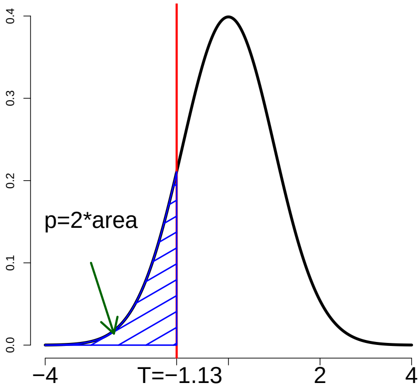
Example

A researcher measures the reaction time of $n = 10$ mice to signal pain when a stitch is applied to their tail. Compute a 95% confidence interval.

```
x=c(2.4, 3.0, 3.0, 2.2, 2.2,
2.2, 2.2, 2.8, 2.0, 3.0)
mx <- mean(x)
vx <- var(x)
crit <- qt(0.975,9)
lower <- mx-crit/sqrt(10)*sqrt(vx)
upper <- mx+crit/sqrt(10)*sqrt(vx)
```

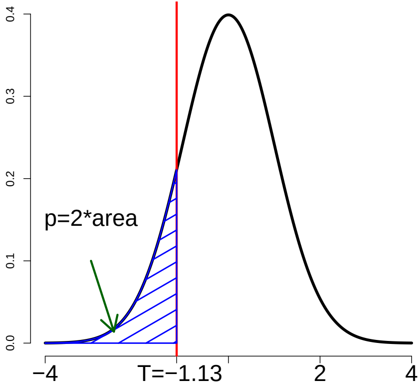
- Estimator: $\widehat{\mu} = 2.5$
- Standard deviation: $\widehat{\sigma} = 0.40$
- Quantile: $t_9(0.975) = 2.26$
- CI: $[2.21; 2.79]$
- Quality of the confidence interval?

Statistical Testing Illustration



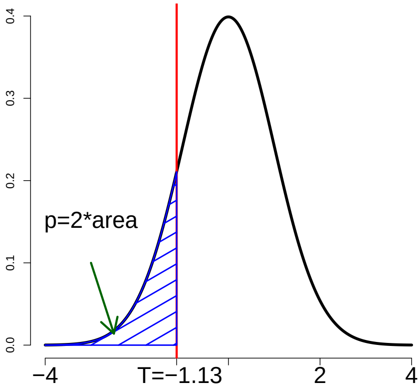
Statistical Testing Illustration

- Compute test statistic and p-value. Sounds good so far, **but....**



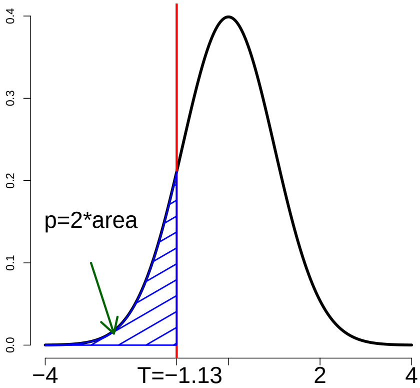
Statistical Testing Illustration

- Compute test statistic and p-value. Sounds good so far, **but....**
- **Might there be a problem?**



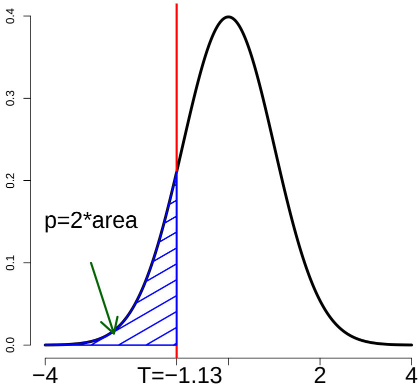
Statistical Testing Illustration

- Compute test statistic and p-value. Sounds good so far, **but....**
- **Might there be a problem?**
- **All that is only valid if distributional assumption is fulfilled** $(X_1, \ldots, X_n \sim N(\mu, \sigma^2))$



Statistical Testing Illustration

- Compute test statistic and p-value. Sounds good so far, **but....**
- **Might there be a problem?**
- **All that is only valid if distributional assumption is fulfilled** $(X_1, \ldots, X_n \sim N(\mu, \sigma^2))$
- **What happens if data come from a different distribution?**



Verifying Assumptions

- Self check: Generate $n = 10$ data from distribution you know and compute a boxplot.

Verifying Assumptions

- Self check: Generate $n = 10$ data from distribution you know and compute a boxplot.
- Can model assumptions be verified?

Verifying Assumptions

- Self check: Generate $n = 10$ data from distribution you know and compute a boxplot.
- Can model assumptions be verified?
- Draw from normal, exponential and χ^2_7

Verifying Assumptions

- Self check: Generate $n = 10$ data from distribution you know and compute a boxplot.
- Can model assumptions be verified?
- Draw from normal, exponential and χ^2_7

- ```
  set.seed(1)
  x<-rnorm(10)
  boxplot(x)
  ```
  ```

Verifying Assumptions

- Self check: Generate $n = 10$ data from distribution you know and compute a boxplot.
- Can model assumptions be verified?
- Draw from normal, exponential and χ^2_7

- ```
  set.seed(1)
  x<-rnorm(10)
  boxplot(x)
  ```
- ```
  set.seed(1)
  x<-rexp(10)
  boxplot(x)
  ```

Verifying Assumptions

- Self check: Generate $n = 10$ data from distribution you know and compute a boxplot.
- Can model assumptions be verified?
- Draw from normal, exponential and χ^2_7

- ```
  set.seed(1)
  x<-rnorm(10)
  boxplot(x)
  ```
- ```
  set.seed(1)
  x<-rexp(10)
  boxplot(x)
  ```
- ```
  set.seed(1)
  x<-rchisq(10,7)
  boxplot(x)
  ```
  ```

Verifying Assumptions

- Self check: Generate $n = 10$ data from distribution you know and compute a boxplot.
- Can model assumptions be verified?
- Draw from normal, exponential and χ^2_7

- ```
  set.seed(1)
  x<-rnorm(10)
  boxplot(x)
  ```
- ```
  set.seed(1)
  x<-rexp(10)
  boxplot(x)
  ```
- ```
  set.seed(1)
  x<-rchisq(10,7)
  boxplot(x)
  ```

- Can we verify/visualize the distribution from the boxplot?
  ```

Type-1 Error Rate Simulations

- **Simulate** the rejection rate of the procedure using statistical software (e.g. R)

Type-1 Error Rate Simulations

- **Simulate** the rejection rate of the procedure using statistical software (e.g. R)
- Principle:

Type-1 Error Rate Simulations

- **Simulate** the rejection rate of the procedure using statistical software (e.g. R)
- Principle:
    1. Generate random data from a population (under the null hypothesis)

Type-1 Error Rate Simulations

- **Simulate** the rejection rate of the procedure using statistical software (e.g. R)
- Principle:
  1. Generate random data from a population (under the null hypothesis)
  2. Compute the test statistic T

Type-1 Error Rate Simulations

- **Simulate** the rejection rate of the procedure using statistical software (e.g. R)
- Principle:
  1. Generate random data from a population (under the null hypothesis)
  2. Compute the test statistic T
  3. Safe whether the hypothesis was rejected or not (an indicator) (1=rejected; 0 = non-rejected)

Type-1 Error Rate Simulations

- **Simulate** the rejection rate of the procedure using statistical software (e.g. R)
- Principle:
  1. Generate random data from a population (under the null hypothesis)
  2. Compute the test statistic T
  3. Safe whether the hypothesis was rejected or not (an indicator) (1=rejected; 0 = non-rejected)
  4. Repeat the above a large number of times

Type-1 Error Rate Simulations

- **Simulate** the rejection rate of the procedure using statistical software (e.g. R)
- Principle:
  1. Generate random data from a population (under the null hypothesis)
  2. Compute the test statistic T
  3. Safe whether the hypothesis was rejected or not (an indicator) (1=rejected; 0 = non-rejected)
  4. Repeat the above a large number of times
  5. Estimate the type-1 error by averaging the indicators

Learning simulations

- Simulate the *t*-Test
- What to investigate? Impact of

Learning simulations

- Simulate the t-Test
- What to investigate? Impact of
    - Sample size n

Learning simulations

- Simulate the t-Test
- What to investigate? Impact of
    - Sample size n
    - Variance σ^2

Learning simulations

- Simulate the t-Test
- What to investigate? Impact of
  - Sample size n
  - Variance σ^2
  - **Distribution**, shape

Learning simulations

- Simulate the t-Test
- What to investigate? Impact of
  - Sample size n
  - Variance σ^2
  - **Distribution**, shape
    - **Symmetric** distributions: Normal, Uniform, Laplace, etc.

Learning simulations

- Simulate the t-Test
- What to investigate? Impact of
  - Sample size n
  - Variance σ^2
  - **Distribution**, shape
    - **Symmetric** distributions: Normal, Uniform, Laplace, etc.
    - **Skewed** distributions: χ^2_f, exponential, Log-Normal, etc.

Learning simulations

- Simulate the t-Test
- What to investigate? Impact of
  - Sample size n
  - Variance σ^2
  - **Distribution**, shape
    - **Symmetric** distributions: Normal, Uniform, Laplace, etc.
    - **Skewed** distributions: χ^2_f, exponential, Log-Normal, etc.
- Make sure to generate the variables by

$$X_k = \frac{Y_k - E(Y_k)}{\sqrt{Var(Y_k)}} \cdot \sigma$$

Learning simulations

- Simulate the t-Test
- What to investigate? Impact of
  - Sample size n
  - Variance σ^2
  - **Distribution**, shape
    - **Symmetric** distributions: Normal, Uniform, Laplace, etc.
    - **Skewed** distributions: χ^2_f, exponential, Log-Normal, etc.
- Make sure to generate the variables by

$$X_k = \frac{Y_k - E(Y_k)}{\sqrt{Var(Y_k)}} \cdot \sigma$$

- $E(X_k) = 0$ and $Var(X_k) = \sigma^2$
  - $Y_k \sim Exp(\lambda)$: $E(Y_k) = \lambda$ and $Var(Y_k) = \lambda^2$

Learning simulations

- Simulate the t-Test
- What to investigate? Impact of
  - Sample size n
  - Variance σ^2
  - **Distribution**, shape
    - **Symmetric** distributions: Normal, Uniform, Laplace, etc.
    - **Skewed** distributions: χ^2_f, exponential, Log-Normal, etc.
- Make sure to generate the variables by

$$X_k = \frac{Y_k - E(Y_k)}{\sqrt{Var(Y_k)}} \cdot \sigma$$

- $E(X_k) = 0$ and $Var(X_k) = \sigma^2$
  - $Y_k \sim Exp(\lambda)$: $E(Y_k) = \lambda$ and $Var(Y_k) = \lambda^2$
  - $Y_k \sim LN(0, 1)$: $E(Y_k) = exp(1/2)$ and $Var(Y_k) = exp(2) - exp(1)$

Learning simulations

- Simulate the t-Test
- What to investigate? Impact of
  - Sample size n
  - Variance σ^2
  - **Distribution**, shape
    - **Symmetric** distributions: Normal, Uniform, Laplace, etc.
    - **Skewed** distributions: χ^2_f, exponential, Log-Normal, etc.
- Make sure to generate the variables by

$$X_k = \frac{Y_k - E(Y_k)}{\sqrt{Var(Y_k)}} \cdot \sigma$$

- $E(X_k) = 0$ and $Var(X_k) = \sigma^2$
  - $Y_k \sim Exp(\lambda)$: $E(Y_k) = \lambda$ and $Var(Y_k) = \lambda^2$
  - $Y_k \sim LN(0,1)$: $E(Y_k) = exp(1/2)$ and $Var(Y_k) = exp(2) - exp(1)$
  - ...

```
mysimulation <-function(Needed Parameters){
```

```
Data Generation
```

```
Compute the Test Statistic
Hypothesis Rejected? (0/1)
```

```
Estimate Type-1 error rate
output table}
```

```
mysimulation(...)
```

```
mysimulation <-function(n,s2,nsim){
crit= qt(0.975, n-1) # critical value at 5\% level
ttest <-c(); set.seed(1)
for(i in 1:nsim){ #Begin Simulation Loop

x <- rnorm(n)*sqrt(s2) #generate from normal dist.
#mu=0 and has variance s2

mx <-mean(x); sdx <- sd(x); T<-sqrt(n)*mx/sdx #compute test statistic
ttest[i] <- (abs(T)>=crit) #Reject yes/no
} #End Simulation Loop

result= data.frame(n=n, alpha=0.05, sigma2=s2, tTest=mean(ttest))
result}
mysimulation(10,1,10000)
```

```r
mysimulation <-function(n,s2,nsim){
crit= qt(0.975, n-1) # critical value at 5\% level
ttest <-c(); set.seed(1)
for(i in 1:nsim){ #Begin Simulation Loop
```

```r
x <- (rexp(n)-1)*sqrt(s2) #generate data from exponential
#(hypothesis is TRUE! has variance=s2)
```

```r
mx <-mean(x); sdx <- sd(x); T<-sqrt(n)*mx/sdx #compute test statistic
ttest[i] <- (abs(T)>=crit) #Reject yes/no
} #End Simulation Loop
```

```r
result= data.frame(n=n, alpha=0.05, sigma2=s2, tTest=mean(ttest))
result}
mysimulation(10,1,10000)
```

```
mysimulation <-function(n,s2,nsim, Distribution){ #use Distribution as argument
```

```
crit= qt(0.975, n-1) # critical value at 5\% level
ttest <-c()
set.seed(1)
for(i in 1:nsim){ #Begin Simulation Loop
```

```
if(Distribution=="Exp"){#generate data from exponential
x <- (rexp(n)-1)*sqrt(s2) }
if(Distribution=="Normal"){#generate data from normal
x <- rnorm(n)*sqrt(s2) }
```

```
mx <-mean(x); sdx <- sd(x); T<-sqrt(n)*mx/sdx #compute test statistic
ttest[i] <- (abs(T)>=crit) #Reject yes/no
} #End Simulation Loop
```

```
result= data.frame(n=n, alpha=0.05, Dist=Distribution,sigma2=s2, tTest=mean(ttest))
result}
mysimulation(10,1,10000,"Exp")
```

Impact of the Small Samples

- Type-I error simulation (10K simulations, $\alpha = 5\%$); Test $H_0 : \mu = 0$

Impact of the Small Samples

- Type-I error simulation (10K simulations, $\alpha = 5\%$); Test $H_0 : \mu = 0$
- Sample sizes: Small (n=10), moderate (n=25) and large (n=50)

Impact of the Small Samples

- Type-I error simulation (10K simulations, $\alpha = 5\%$); Test $H_0 : \mu = 0$
- Sample sizes: Small (n=10), moderate (n=25) and large (n=50)
- Different distributions

Impact of the Small Samples

- Type-I error simulation (10K simulations, $\alpha = 5\%$); Test $H_0 : \mu = 0$
- Sample sizes: Small (n=10), moderate (n=25) and large (n=50)
- Different distributions
- Fill the table:

Sample Size	Distribution	Shape	Emp. Type-I	Accurate (yes/no)
Small	Normal	Symmetric		
	Exponential	Skewed		
Moderate	Normal	Symmetric		
	Exponential	Skewed		
Large	Normal	Symmetric		
	Exponential	Skewed		

Impact of the Small Samples

- Type-I error simulation (10K simulations, $\alpha = 5\%$); Test $H_0 : \mu = 0$
- Sample sizes: Small (n=10), moderate (n=25) and large (n=50)
- Different distributions
- Fill the table:

Sample Size	Distribution	Shape	Emp. Type-I	Accurate (yes/no)
Small	Normal	Symmetric		
	Exponential	Skewed		
Moderate	Normal	Symmetric		
	Exponential	Skewed		
Large	Normal	Symmetric		
	Exponential	Skewed		

Impact of the Small Samples

- Type-I error simulation (10K simulations, $\alpha = 5\%$); Test $H_0 : \mu = 0$
- Sample sizes: Small (n=10), moderate (n=25) and large (n=50)
- Different distributions
- Fill the table:

Sample Size	Distribution	Shape	Emp. Type-I	Accurate (yes/no)
Small	Normal	Symmetric		
	Exponential	Skewed		
Moderate	Normal	Symmetric		
	Exponential	Skewed		
Large	Normal	Symmetric		
	Exponential	Skewed		

- Can we analyze the data sets with the method?

Impact of the Small Samples

- Type-I error simulation (10K simulations, $\alpha = 5\%$); Test $H_0 : \mu = 0$
- Sample sizes: Small (n=10), moderate (n=25) and large (n=50)
- Different distributions
- Fill the table:

Sample Size	Distribution	Shape	Emp. Type-I	Accurate (yes/no)
Small	Normal	Symmetric		
	Exponential	Skewed		
Moderate	Normal	Symmetric		
	Exponential	Skewed		
Large	Normal	Symmetric		
	Exponential	Skewed		

- Can we analyze the data sets with the method?
- Can we take data characteristics into account? ($\Rightarrow$ Resampling)

Efficient Implementation

- Numerical implementation is important

Efficient Implementation

- Numerical implementation is important
- Use **matrix techniques** to generate the data

Efficient Implementation

- Numerical implementation is important
- Use **matrix techniques** to generate the data
- Data matrix with nsim columns

Efficient Implementation

- Numerical implementation is important
- Use **matrix techniques** to generate the data
- Data matrix with nsim columns
    - R functions *colMeans(x)*, *colSums(x)*

Efficient Implementation

- Numerical implementation is important
- Use **matrix techniques** to generate the data
- Data matrix with nsim columns
  - R functions *colMeans(x)*, *colSums(x)*
  - These functions are very fast

Efficient Implementation

- Numerical implementation is important
- Use **matrix techniques** to generate the data
- Data matrix with nsim columns
    - R functions *colMeans(x)*, *colSums(x)*
    - These functions are very fast
    - Variance formula

$$\widehat{\sigma}^2 = \frac{1}{n-1}\sum_{k=1}^{n}(X_k - \overline{X}_.)^2 = \frac{1}{n-1}\left(\sum_{k=1}^{n} X_k^2 - n\overline{X}_.^2\right)$$

Efficient Implementation

- Numerical implementation is important
- Use **matrix techniques** to generate the data
- Data matrix with nsim columns
    - R functions *colMeans(x)*, *colSums(x)*
    - These functions are very fast
    - Variance formula

$$\hat{\sigma}^2 = \frac{1}{n-1}\sum_{k=1}^{n}(X_k - \overline{X}_.)^2 = \frac{1}{n-1}(\sum_{k=1}^{n} X_k^2 - n\overline{X}_.^2)$$

    - Try to avoid ”for” loops

Efficient Implementation

- Numerical implementation is important
- Use **matrix techniques** to generate the data
- Data matrix with nsim columns
    - R functions *colMeans(x)*, *colSums(x)*
    - These functions are very fast
    - Variance formula

$$\widehat{\sigma}^2 = \frac{1}{n-1}\sum_{k=1}^{n}(X_k - \overline{X}_.)^2 = \frac{1}{n-1}\left(\sum_{k=1}^{n} X_k^2 - n\overline{X}_.^2\right)$$

- Try to avoid ”for” loops
- Generate a matrix of variables (example next slide)

```
mysimulation <-function(n,s2,Distribution,nsim){
crit= qt(0.975, n-1) # critical value at 5\% level
set.seed(1)
```

```
if(Distribution=="Normal"){
x <- matrix(rnorm(n=n*nsim)*sqrt(s2),ncol=nsim)}
if(Distribution=="Exp"){#(Hypothesis is TRUE!)
x <- matrix((rexp(n=n*nsim)-1)*sqrt(s2),ncol=nsim)}
```

```
mx <-colMeans(x)
vx <- (colSums(x^2)-n*mx^2)/(n-1)
T <- sqrt(n)*mx/sqrt(vx)
ttest <- (abs(T)>=crit)
```

```
result= data.frame(n=n,Dist=Distribution,sigma2=s2,ttest=mean(ttest))
result}
mysimulation(10,1,"Normal",10000)
```

Simulation of Point Estimators

- Simulations can be used to assess the quality of point estimators

Simulation of Point Estimators

- Simulations can be used to assess the quality of point estimators
- Estimator should be *unbiased* and *consistent* (low variance)

Simulation of Point Estimators

- Simulations can be used to assess the quality of point estimators
- Estimator should be *unbiased* and *consistent* (low variance)
- Task: Estimate the variance σ^2

Simulation of Point Estimators

- Simulations can be used to assess the quality of point estimators
- Estimator should be *unbiased* and *consistent* (low variance)
- Task: Estimate the variance σ^2
- Estimators

$$\widehat{\sigma}_1^2 = \frac{1}{n-1}\sum_{k=1}^{n}(X_k - \overline{X}.)^2 \qquad \widehat{\sigma}_2^2 = \frac{1}{n}\sum_{k=1}^{n}(X_k - \overline{X}.)^2$$

Simulation of Point Estimators

- Simulations can be used to assess the quality of point estimators
- Estimator should be *unbiased* and *consistent* (low variance)
- Task: Estimate the variance σ^2
- Estimators

$$\widehat{\sigma}_1^2 = \frac{1}{n-1}\sum_{k=1}^{n}(X_k - \overline{X}.)^2 \qquad \widehat{\sigma}_2^2 = \frac{1}{n}\sum_{k=1}^{n}(X_k - \overline{X}.)^2$$

- Which one is better?

Simulation of Point Estimators

- Simulations can be used to assess the quality of point estimators
- Estimator should be *unbiased* and *consistent* (low variance)
- Task: Estimate the variance σ^2
- Estimators

$$\widehat{\sigma}_1^2 = \frac{1}{n-1}\sum_{k=1}^{n}(X_k - \overline{X}.)^2 \qquad \widehat{\sigma}_2^2 = \frac{1}{n}\sum_{k=1}^{n}(X_k - \overline{X}.)^2$$

- Which one is better?
- Assess a simulation to compute a) their bias and b) their standard error

Simulation

- Structure of the Program

Simulation

- Structure of the Program
  - Generate data (know parameter θ)

Simulation

- Structure of the Program
  - Generate data (know parameter θ)
  - Begin Simulation Loop

Simulation

- Structure of the Program
  - Generate data (know parameter θ)
  - Begin Simulation Loop
  - Compute estimator and safe the value of $\widehat{\theta}_\ell, \; \ell = 1, \ldots, n_{sim}$

Simulation

- Structure of the Program
  - Generate data (know parameter θ)
  - Begin Simulation Loop
  - Compute estimator and safe the value of $\widehat{\theta}_\ell, \ \ell = 1, \dots, n_{sim}$
  - End of simulation loop

Simulation

- Structure of the Program
  - Generate data (know parameter θ)
  - Begin Simulation Loop
  - Compute estimator and safe the value of $\widehat{\theta}_\ell, \ \ell = 1, \ldots, n_{sim}$
  - End of simulation loop
  - Assess bias and mean square error as

$$bias = \frac{1}{n_{sim}} \sum_{\ell=1}^{n_{nsim}} (\widehat{\theta}_\ell - \theta) \qquad MSE = \frac{1}{n_{sim}} \sum_{\ell=1}^{n_{sim}} (\widehat{\theta}_\ell - \theta)^2$$

```
mysimulation <-function(Needed Parameters){

Data Generation (know true value)

Compute the Point Estimator and safe the value

Compute the bias and MSE
output table}

mysimulation(...)
```

```r
mysimulation <-function(n,s2,Distribution,nsim){

if(Distribution=="Normal"){
x <- matrix(rnorm(n=n*nsim)*sqrt(s2),ncol=nsim)}
if(Distribution=="Exp"){
x <- matrix((rexp(nsim*n)-1)*sqrt(s2),ncol=nsim)} #(Parameter is known)

mx <-colMeans(x)
v1 <- (colSums(x^2)-n*mx^2)/(n-1)
v2<- (colSums(x^2)-n*mx^2)/n

result= data.frame(n=n, sigma2=s2, Distribution= Distribution,
bias.v1=mean(v1-s2), MSE.v1=mean((v1-s2)^2), bias.v2=mean(v2-s2),
MSE.v2=mean((v2-s2)^2))
result}
mysimulation(10,1,"Normal",10000)
```

Project

1. Write a simulation program to assess the quality of 95% confidence interval for mean

2. Let $X_1, \dots, X_n$ have $E(X_k) = \mu$. We want to estimate $\theta = \mu^2$. Which of the estimators

$$\widehat{\theta}_1 = \overline{X}_{\cdot}^2 \qquad \text{or} \qquad \widehat{\theta}_2 = \frac{1}{n(n-1)} \sum_{i \neq j} X_i \cdot X_j$$

   would you recommend?