Session 4

author: Mandy / Thomas date: `r format(Sys.time(), '%d %B, %Y')` auto-size: true css: talk.css transition: rotate navigation: section font-family: times, serif

Recap

incremental: true ## You should know: - **statistics** is all about **simplifying** - we try to **summarize** and **describe** data through **parameters** of: - *LOCATION* - e.g. *mean, median, mode* - *SCALE* - e.g. *variance, standard deviation* - *SPREAD* - e.g. *minimum / maximum / range / quantile / IQR* - the meaning of these parameters - necessary R commands

Recap

incremental: true ## We have seen how - parameters of
- location of two groups (means) - spread (standard deviation) - uncertainty (sample size) incremental: true - to measure a difference of the location in a standardized manner - this measure is compared to a t-distribution relating to a so called **Null-Hypotheses** which one "hopefully" will be rejected to show an effect could exist

Recap

incremental: true ## We have seen how - this comparision is transformed to a propabiltiy **(p-value)** to get this result by random - comparing this propability with a defined maximum propability for a random result (normally 5%) gives the opportunity to decide whether the effect could exist or not - important is also the effect size itself
(e.g. is an effect of $\mu_2 - \mu_1 = 101 - 100 = 1$ relevant?)

Test Statistic

incremental: true - some property of two groups (Men and Women) are measured. - to compare their means, we apply the so called **T-Test** - to do this we compute the so called **T-Statistic**:

$$t = \frac{\bar{X}_m - \bar{X}_w}{s_{overall}\sqrt{\frac{1}{n_m} + \frac{1}{n_w}}}$$

Decisions can be right or wrong

	H_0 is true	H_0 is false
H_0 is not rejected	Correct decision	Type II error
H_0 is rejected	Type I error	Correct decision

Alternative of alternatives

incremental: true alternative | options
———— | ————- one sided test | less | greater two sided test | equal
————- | ————- one sample test |
two sample test | equal variances | not necessary equal variances
———— | ————- unpaired | paired |

Alternative of alternatives

- two sided - equal

“‘{r,echo=FALSE,fig.height=8,fig.width = 12,fig.align=‘center’} curve(dt(x,df=1000),from = -4,to = 4) text(-4,0.35,“alternative: two sided”,pos=4) abline(h=0) x <- seq(-4,qt(0.025,df=1000),by=0.01) y <- dnorm(x) x <- c(x,qt(0.025,df=1000)) y <- c(y,dt(-4,df=1000)) polygon(x,y,col=“red”)

```
x <- seq(qt(0.975,df=1000),4,by=0.01)
y <- dt(x,df=1000)
x <- c(x,qt(0.975,df=1000))
y <- c(y,dt(4,df=1000))
polygon(x,y,col="red")

text(-2.75,0.02,expression(paste("reject ",H[0])),pos=4)
text(1.9,0.02,expression(paste("reject ",H[0])),pos=4)
```

“‘

Alternative of alternatives

- one sided - less

“‘{r, echo=FALSE,fig.height=8,fig.width=12,fig.align=‘center’} curve(dt(x,df=1000),-4,4) abline(h=0) x <- seq(-4,qt(0.05,df=1000),by=0.01) y <- dt(x,df=1000) x <- c(x,qt(0.05,df=1000)) y <- c(y,dt(-4,df=1000)) polygon(x,y,col=“red”)

text(-4,0.35,“alternative: less”,pos=4) text(-2.45,0.02,expression(paste(“reject”,H[0])),pos=4)

“‘

Alternative of alternatives

- one sided - greater

“‘{r, echo=FALSE,fig.height=8,fig.width=12,fig.align=‘center’} curve(dt(x,df=1000),-4,4) abline(h=0)

x <- seq(qt(0.95,df=1000),4,by=0.01) y <- dt(x,df=1000) x <- c(x,qt(0.95,df=1000)) y <- c(y,dt(4,df=1000)) polygon(x,y,col=“red”)

text(-4,0.35,“alternative: greater”,pos=4) text(1.6,0.02,expression(paste(“reject”,H[0])),pos=4)

“‘

T-Tests in R

incremental: true **There are many options more but only one command in R:**

```
t.test( )
```

T-Tests in R

class: small-code incremental:true **One Sample T-Test**

```
set.seed( 1 )
x <- rnorm( 12 ) ## create random numbers
t.test( x, mu = 0 ) ## population mean 0
```

T-Tests in R

class: small-code incremental:true **One Sample T-Test**

```
t.test( x, mu = 1 ) ## population mean 1
```

T-Tests in R

class: small-code incremental:true **Two Samples T-Test** - we have given a two numeric vectors - we do not assume equal variances for the underlying distributions

```
set.seed( 1 )
x <- rnorm( 12 )
y <- rnorm( 12 )
head( data.frame( x, y ) )
```

T-Tests in R

class: small-code incremental:true **Two Samples T-Test**

```
t.test( x, y )
```

T-Tests in R

class: small-code incremental:true **Two Samples T-Test** - we have one numeric vector and one vector containing the group information - we do not assume equal variances for the underlying distribution

```
## create random group vector
g <- sample( c( "A", "B" ), 12, replace = T )
head( data.frame( x, g ) )
```

T-Tests in R

class: small-code incremental:true **Two Samples T-Test**

```
t.test( x ~ g )
```

T-Tests in R

class: small-code incremental:true **Two Samples T-Test** - equal variances now

```
t.test( x ~ g, var.equal = T )
```

When should one use the t-test?

incremental:true - comparision of mean values against a population value or against each other - the t-test, especially the Welch test is appropriate whenever the underlying distributions are normal - it is also recommended for a group size equal or larger than 30 (robust against deviation from normality)

Exercise

incremental:true Use the ALLBUS data set: - do a test of income (V417) for the groups male and female (V81)! - compare the bmi (V279) of smokers and non-smokers (V272) - compare the bmi (V279) of people with high and normal blood pressure (V242) - how would you interprete the results? - visualize!

Simulations with R

class: small-code incremental:true **Rolling the dice**

Suppose you are rolling a fair dice 600 times!
- How many sixes would you expect? - How many sixes do we need to reject the H_0-Hypotheses using a **two-sided test**?
- test for **EQUALITY**

```
qbinom( p = c( .025, .975 ), size = 600, prob = 1 / 6 )
```

Simulations with R

class: small-code incremental:true What do we have to change for a **one-sided test**? - test for **LESS**

```
qbinom( p = .05, size = 600, prob = 1 / 6 )
```

- test for **GREATER**

  ```
  qbinom( p = .95, size = 600, prob = 1 / 6 )
  ```

Simulations with R

class: tiny-code incremental:true

Now let's R roll the dice.

```
## paranthesis are for executing this row instantly
( dice.trials <- sample( 1 : 6, 600, replace = T ) )
```

Simulations with R

class: tiny-code incremental:true

Find the sixes!

```
dice.trials == 6
```

Simulations with R

class: tiny-code incremental:true

Count them!

```
## length( dice.trials[ dice.trials == 6 ] ) ## one way
sum( dice.trials == 6 ) ## another way
```

Simulations with R

class: tiny-code incremental:true

Now let's R roll the dice very very often!

Now use the following code to replicate the experiment (rolling one fair dice 600 times) 1000 times!
The number of sixes are stored in the vector **dice.trials.1000**.

- How many statistically significant results do you expect for a one-sided alternative?
- How many for a two-sided alternative?
- How many statistically significant results did you get? (You can use **table( )** in combination with a logical function.)
- Visualize the result using **ggplot2** and **geom_histogram( )**! Look at the help of geom_histogram! Alternatively you can use **hist( )**.

Simulations with R

class: tiny-code incremental:true

Now let's R roll the dice very very often!

```
dice.trials.1000 <- replicate( 1000, sum( sample( 1 : 6, 600, replace = T ) == 6 ) )
df <- data.frame( repl.count = 1 : 1000, sixes.count = dice.trials.1000 )
head( df )
```

Simulations with R

class: tiny-code incremental:true

Now let's R roll the dice very very often!

```
table( df$sixes.count )
quantile( df$sixes.count, probs = c( 0, .025, .05, .5, .95, .975, 1 ) )
```

Simulations with R

class: tiny-code incremental:true

Now let's R roll the dice very very often!

```
hist( df$sixes.count, breaks = 50 )
```

Simulations with R

class: tiny-code incremental:true

Now let's R roll the dice very very often!

```
library( ggplot2 )
## ??geom_histogram
ggplot( df, aes( sixes.count ) ) + geom_histogram( binwidth = 1, col = '#1A1A18', fill = '#9
```