Using  for statistical analyses

Robert Bauer

Warnemünde, 08/02/2012

Day 5 - Agenda:

- t-test
- One-Way ANOVA
- Exercises
- multiple pairwise comparisons
  - post hoc tests (Scheffé, Tukey)
  - a priori tests (Dunnett, user defined)
- barplots with error bars

One-Sample t-test

```
setwd('~/Dropbox/R_course/day5') # set working directory

file <- "Clone1.csv"
datasheet <- read.table(file, header=T, sep=',', dec=".") #
    load dataframe

# Student's t-test
# H0: mean == 0
t.test(datasheet$growth.rate)
# if p-value < 0.05: reject H0, else: keep H0
```

```
One Sample t-test
data:  datasheet$growth.rate
t = 24.1703, df = 23, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
 95 percent confidence interval:
 2.596819 3.082930
sample estimates:
mean of x
  2.839875
```

One-Sample t-test

```r
# H0: mean == 0
t.test(datasheet$growth.rate)
# if p-value < 0.05: reject H0, else: keep H0

# H0: mean == 2
t.test(datasheet$growth.rate, mu=2)
t.test(datasheet$growth.rate, mu=2, alternative="two.sided")

# H0: mean <= 2
t.test(datasheet$growth.rate, mu=2, alternative="greater")

# H0: mean >= 2
t.test(datasheet$growth.rate, mu=2, alternative="less")
```

One-Sample t-test

```
# H0: mean == 2
t.test(datasheet$growth.rate, mu=2)
```

```
One Sample t-test

data:  datasheet$growth.rate
t = 8.4734, df = 23, p-value = 1.577e-08
alternative hypothesis: true mean is not equal to 2
95 percent confidence interval:
 3.616580 4.660859
```

```
# H0: mean == 3
t.test(datasheet$growth.rate, mu=3)
```

```
One Sample t-test

data:  datasheet$growth.rate
t = 4.5115, df = 23, p-value = 0.0001573
alternative hypothesis: true mean is not equal to 3
95 percent confidence interval:
 3.616580 4.660859
```

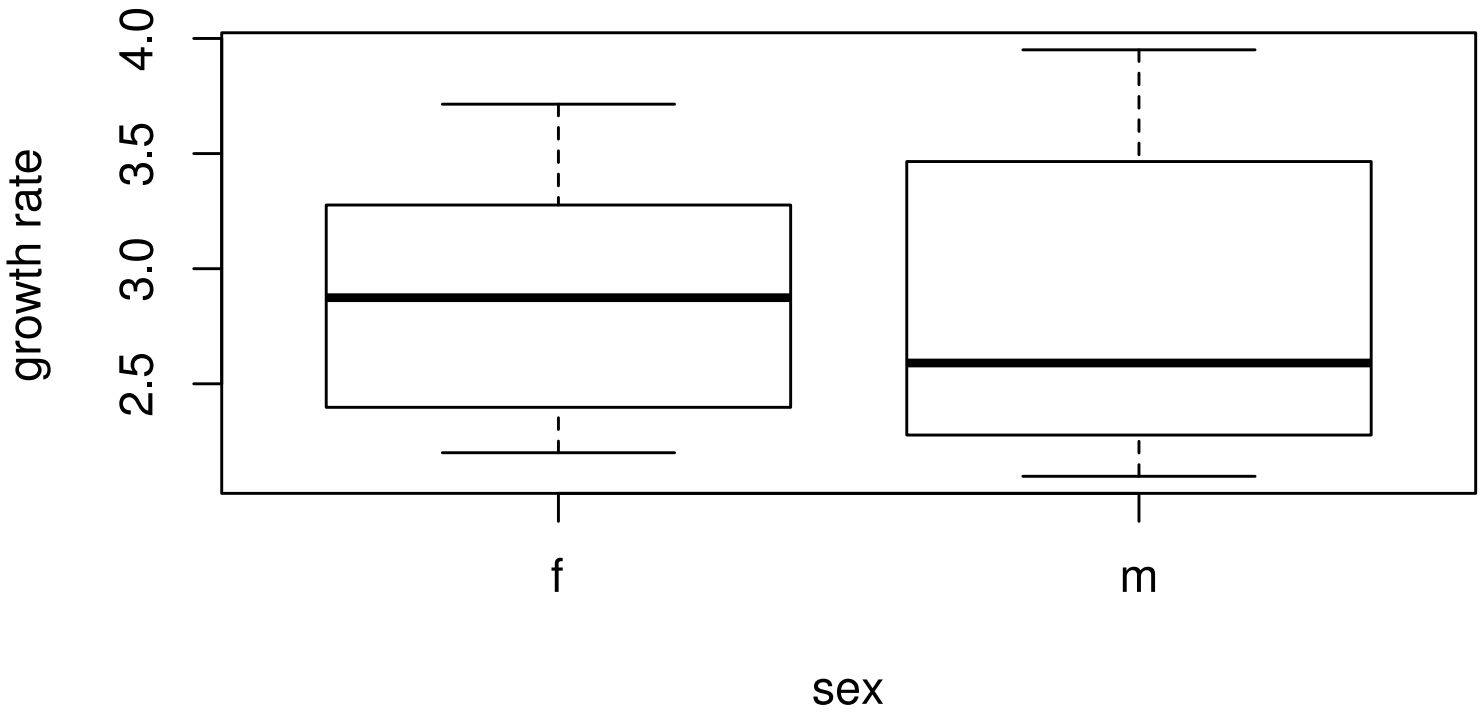
confidence intervals

```
# changing the confidence interval
t.test(datasheet$growth.rate, mu=3, conf.level=0.99)
# more precise information cause larger intervals!

# accessing test results
results <- t.test(datasheet$growth.rate, mu=3,
                  conf.level=0.99)
names(results) # return names of test output
results$conf.int # call confidence intervals
```

Two-Sample t-test

```
# compare means between sexes
Data = datasheet
Data$sex <- factor(Data$sex)
formula <- growth.rate~sex
ylabel <- "growth rate"
xlabel <- "sex"

boxplot(formula, data=Data, ylab=ylabel, xlab=xlabel)
```

Two-Sample t-test



Two-Sample t-test

- Assumptions
  - normal distributed samples
  - variance homogeneity
  - independent data

Testing Assumptions - Normal Distribution

```r
# Shapiro-Wilk test for Normal Distribution
# better for small sampling sizes (<50)
# H0: normal distributed data
shapiro.test(Data$growth.rate[Data$sex == "f"])
shapiro.test(Data$growth.rate[Data$sex == "m"])
# if p-value < 0.05: reject H0, else: keep H0
```

```
Shapiro-Wilk normality test

data:  Data$growth.rate[Data$sex == "f"]
W = 0.9507, p-value = 0.6469

data:  Data$growth.rate[Data$sex == "m"]
W = 0.8856, p-value = 0.1034
```

Testing Assumptions - homogeneity of variance

```r
# Levene's test for homogeneity of variance across groups
# H0: variances are equal (homogeneity of variance)
# H1: variances differ
install.packages('car') # install required package
library(car)            # load package
leveneTest(formula, data=Data)
# if Pr(>F) < 0.05: reject H0, else: keep H0
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  1  0.8712 0.3608
      22
```

Two-Sample t-test

```r
# compare means between sexes
Data = datasheet
Data$sex <- factor(Data$sex)
formula <- growth.rate~sex

# Two-Sample t-test
# H0: average growth rates are equal between sexes
# H1: average growth rates differ between sexes
t.test(formula, alternative="two.sided",
       paired=F, var.equal=T, data=Data)
# if p-value < 0.05: reject H0, else: keep H0
```

```
Two Sample t-test

data:  growth.rate by sex
t = 0.2342, df = 22, p-value = 0.817
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.4414751  0.5538642
sample estimates:
mean in group f mean in group m
       2.867972        2.811777
```

One-Way ANOVA

```r
# compare means between sexes
Data = datasheet
Data$sex <- factor(Data$sex)
formula <- growth.rate~sex

# One-Way ANOVA
# H0: average growth rates are equal between sexes
anova(lm(formula, data=Data))     # version 1
summary(aov(formula, data=Data)) # version 2
# if Pr(>F) < 0.05: reject H0, else: keep H0
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
sex          1  0.019  0.0189   0.055  0.817
Residuals   22  7.601  0.3455
```

One-Way ANOVA

```
# combining results from 3 Clones
Clone.levels <- 1:3
for(i in Clone.levels){
  # load dataframe
  file <- paste("Clone", i, ".csv", sep="")
  datasheet <- read.table(file, header=T, sep=',', dec=".")

  # add column consisting Clone number information
  datasheet <- data.frame(datasheet,
                          Clone=rep(i, dim(datasheet)[1]))

  # add all subtables to new data frame
  if(i == 1){
    daphnia <- datasheet
  }
  else{
    daphnia <- rbind(daphnia, datasheet)
  }
}
```

One-Way ANOVA

```
summary(daphnia)
```

```
 growth.rate    sex        Clone
Min.   :1.762   f:36   Min.   :1
1st Qu.:2.797   m:36   1st Qu.:1
Median :3.788          Median :2
Mean   :3.852          Mean   :2
3rd Qu.:4.807          3rd Qu.:3
Max.   :6.918          Max.   :3
```

```
daphnia$Clone <- factor(daphnia$Clone)
summary(daphnia)
```

```
 growth.rate    sex    Clone
Min.   :1.762   f:36   1:24
1st Qu.:2.797   m:36   2:24
Median :3.788          3:24
Mean   :3.852
3rd Qu.:4.807
Max.   :6.918
```

One-Way ANOVA

```r
# Perform an ANOVA to compare means between sexes
Data <- daphnia
formula <- growth.rate~sex
ylabel <- "growth rate"
xlabel <- "sex"

# boxplot
boxplot(formula, data=Data, ylab=ylabel, xlab=xlabel)

# Shapiro-Wilk test for Normal Distribution
# better for small sampling sizes (<50)
# H0: normal distributed data
shapiro.test(Data$growth.rate[Data$sex == "f"])
shapiro.test(Data$growth.rate[Data$sex == "m"])
# if p-value < 0.05: reject H0, else: keep H0

# test homogeneity of variances
leveneTest(formula, data=Data)
# if Pr(>F) < 0.05: reject H0! (H0: Variances are equal)

# One-way ANOVA
# H0: average growth rates are equal between sexes
summary(aov(formula, data=Data))
# if Pr(>F) < 0.05: reject H0, else: keep H0
```

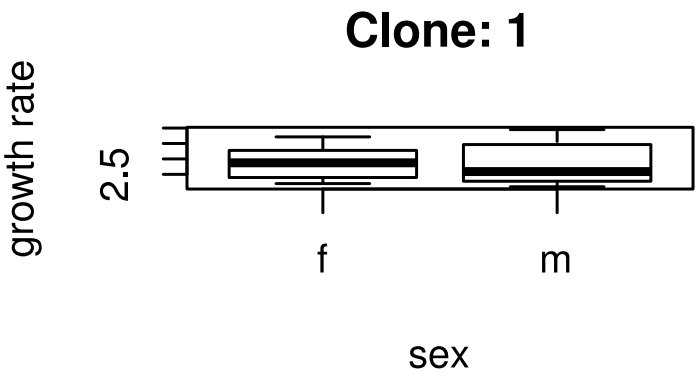
Exercises

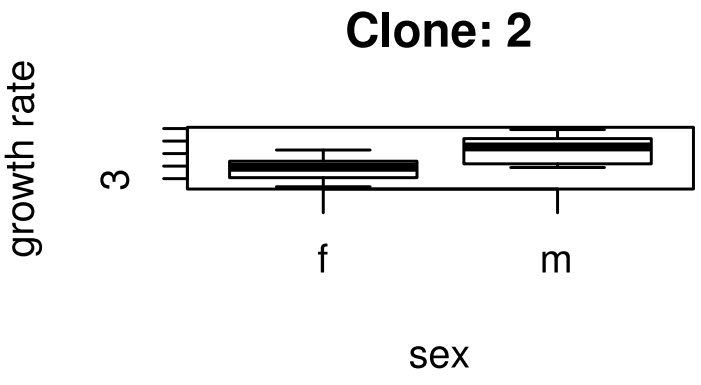
1. Compare growth rates between sexes, but for each Clone separately
    1.1 use a for loop
        1.1.1 to create boxplots
        1.1.2 to return results to the console
2. Compare growth rates between clones, disregarding sexes

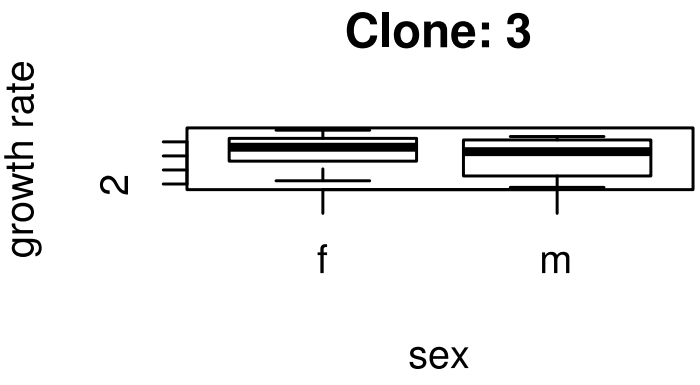
Exercises

```r
# 1. Compare growth rates between sexes, but for each Clone
    separately
formula <- growth.rate~sex
ylabel <- "growth rate"
xlabel <- "sex"

Clone.levels <- 1:3
par(mfrow=c(2,2))
for(i in Clone.levels){
  # option 1: subset data
  Data <- subset(daphnia, daphnia$Clone == i)

  # option 2: reload data
  # file <- paste("Clone", i, ".csv", sep="")
  # Data <- read.table(file, header=T, sep=',', dec=".")

  Title <- paste("Clone: ", i, sep="")
  boxplot(formula, data=Data,
          main= Title, ylab=ylabel, xlab=xlabel)
}
```

Exercises


Clone: 1
growth rate
2.5
f
m
sex


Clone: 2
growth rate
3
f
m
sex


Clone: 3
growth rate
2
f
m
sex

Exercises

```r
Clone.levels <- 1:3
for(i in Clone.levels){
  # option 1: subset data
  Data <- subset(daphnia, daphnia$Clone == i)
  Title <- paste("Clone: ", i, sep="")
  print(Title)

  # Shapiro-Wilk test for Normal Distribution
  # better for small sampling sizes (<50)
  # H0: normal distributed data
  print(shapiro.test(Data$growth.rate[Data$sex == "f"]))
  print(shapiro.test(Data$growth.rate[Data$sex == "m"]))
  # if p-value < 0.05: reject H0, else: keep H0

  # test homogeneity of variances
  print(leveneTest(formula, data=Data))
  # if Pr(>F) < 0.05: reject H0! (H0: Variances are equal)

  # One-way ANOVA
  # H0: average growth rates are equal between sexes
  print(summary(aov(formula, data=Data)))
  # if Pr(>F) < 0.05: reject H0 (growth rates are equal)
}
```

Exercises

```r
# 2. Compare growth rates between Clones, disregarding sexes
Data <- daphnia
formula <- growth.rate~Clone
ylabel <- "growth rate"
xlabel <- "Clones"

# boxplot
par(mfrow=c(1,1))
boxplot(formula, data=Data, ylab=ylabel, xlab=xlabel)

# Shapiro-Wilk test for Normal Distribution
# better for small sampling sizes (<50)
# H0: normal distributed data
shapiro.test(Data$growth.rate[Data$Clone == "1"])
shapiro.test(Data$growth.rate[Data$Clone == "2"])
shapiro.test(Data$growth.rate[Data$Clone == "3"])
# if p-value < 0.05: reject H0, else: keep H0

# test homogeneity of variances
leveneTest(formula, data=Data)
# if Pr(>F) < 0.05: reject H0! (H0: Variances are equal)

# One-way ANOVA
# H0: average growth rates are equal between Clones
summary(aov(formula, data=Data))# if Pr(>F) < 0.05: reject H0
```

multiple pairwise comparisons - post hoc tests - Scheffé

```
ANOVA <- aov(formula, data=Data)

# a) Scheffe Test
install.packages('agricolae')
library(agricolae)
scheffe.test(ANOVA, "Clone")
```

```
Scheffe Test for growth.rate
Mean Square Error  : 1.130137
  growth.rate   std.err replication
1    2.839875 0.1174944          24
2    4.577121 0.2524957          24
3    4.138719 0.2524048          24
alpha: 0.05 ; Df Error: 69
Critical Value of F: 3.129644

Minimum Significant Difference: 0.7677811

Means with the same letter are not significantly different.
Groups, Treatments and means
a   2   4.57712052183333
a   3   4.13871948341667
b   1   2.83987470295833
```

multiple pairwise comparisons - post hoc tests - Tukey

```
# b) Tukey Test
# version 1
TukeyHSD(ANOVA) # p adj:
plot(TukeyHSD(ANOVA))

# version 2 - "General Linear Hypotheses"
library(multcomp)
TUKEY <- glht(ANOVA, linfct=mcp(Clone="Tukey"),
              interaction_average=TRUE)

summary(TUKEY)
```

multiple pairwise comparisons - post hoc tests - Tukey

```
summary(TUKEY)
```

```
   Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts

Fit: aov(formula = formula, data = Data)

Linear Hypotheses:
           Estimate Std. Error t value Pr(>|t|)
2 - 1 == 0   1.7372     0.3069   5.661  < 1e-04 ***
3 - 1 == 0   1.2988     0.3069   4.232 0.000188 ***
3 - 2 == 0  -0.4384     0.3069  -1.429 0.331993
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- none method)
```

```
summary(TUKEY, test=adjusted("none"))
```

```
Linear Hypotheses:
           Estimate Std. Error t value Pr(>|t|)
2 - 1 == 0   1.7372     0.3069   5.661 3.18e-07 ***
3 - 1 == 0   1.2988     0.3069   4.232 6.99e-05 ***
3 - 2 == 0  -0.4384     0.3069  -1.429    0.158
---
```

multiple pairwise comparisons - a priori tests

```
ANOVA <- aov(formula, data=Data)

# a) Dunnett Contrasts, treating group 1 as control group
DUNNET <- glht(ANOVA, linfct=mcp(Clone="Dunnett"))
summary(DUNNET)
```

```
 Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Dunnett Contrasts

Fit: aov(formula = formula, data = Data)

Linear Hypotheses:
           Estimate Std. Error t value Pr(>|t|)
2 - 1 == 0   1.7372     0.3069   5.661 6.34e-07 ***
3 - 1 == 0   1.2988     0.3069   4.232 0.000138 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

multiple pairwise comparisons - a priori tests

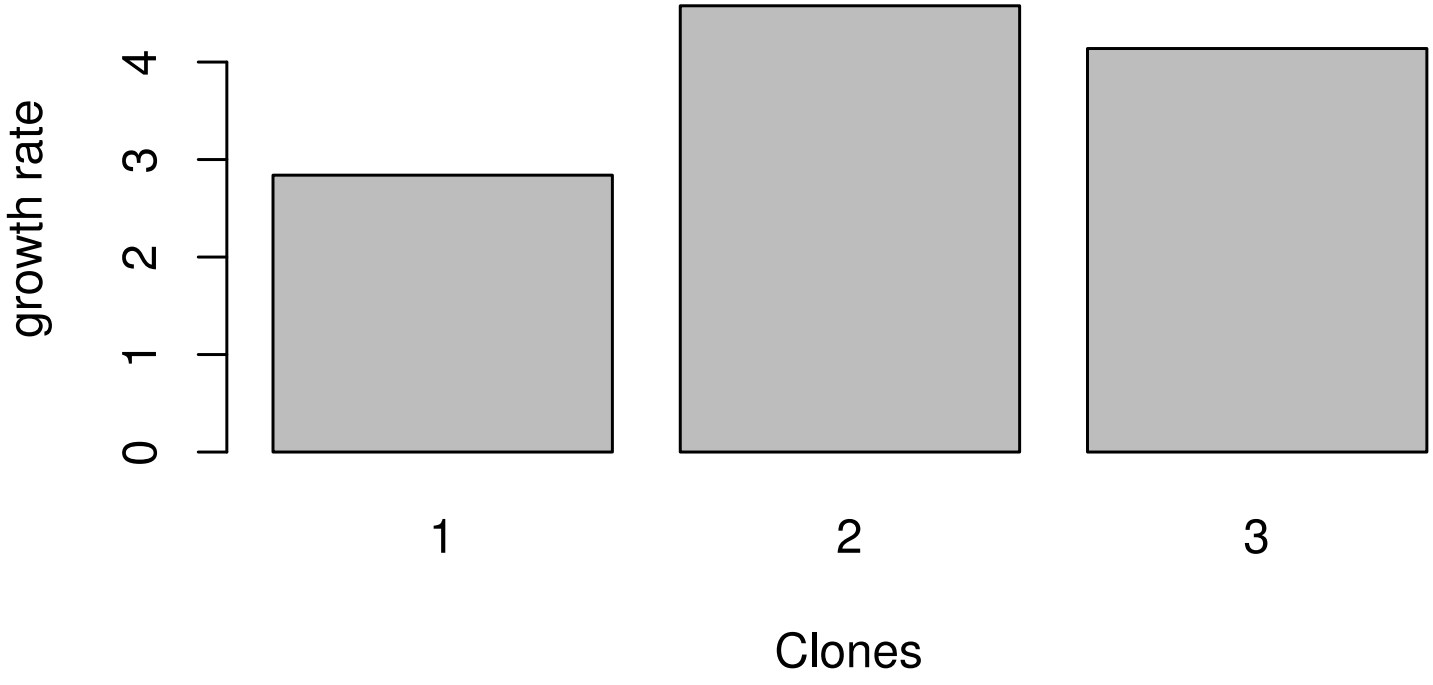
```
ANOVA <- aov(formula, data=Data)

# a) Dunnett Contrasts, treating group 1 as control group
DUNNET <- glht(ANOVA, linfct=mcp(Clone="Dunnett"))
summary(DUNNET)
```

```
# b) User-defined Contrasts
contrast <- rbind(c(-1,1,0),
                  c(-1,0,1))

contrast <- rbind("2 - 1"=c(-1,1,0),
                  "3 - 1"=c(-1,0,1))

USER <- glht(ANOVA, linfct=mcp(Clone=contrast))
summary(USER)
```

barplots with error bars

```
ylabel <- "growth rate"
xlabel <- "Clones"
attach(daphnia)

# calculate average, sd, sampling size and standard error of
    growth rates
Means <- aggregate(growth.rate, list(Clone), mean)$x
sd <- aggregate(growth.rate, list(Clone), sd)$x
sampling_size <- aggregate(growth.rate, list(Clone), length)$x
Error <- sd/sqrt(sampling_size)

# start plotting procedure
BARPLOT <- barplot(Means, ylab=ylabel, xlab=xlabel, names=1:3,
    ylim=c(0,max(Means+Error)))
# assignment returns x-coordinates of barplots
BARPLOT
```

barplots with error bars



barplots with error bars

```
# draw error bars
arrows(BARPLOT, Means+Error, # starting coordinates (x,y) of
    arrows
       BARPLOT, Means,       # end coordinates (x,y) of arrows
       angle=90,             # angle between the arrow shaft
           and the arrow head
       code=1,               # arrow type
       length=0.1)           # length of arrow head
```

```
# add labels from variance analysis
text(BARPLOT, Means+1, # x and y-coordinates
     xpd = TRUE,       # allow text placement outside ylim
     c("a", "b", "b"), # text to plot
     font=2)           # font type
# 1=plain, 2=bold, 3=italic, 4=bold italic
```

barplots with error bars