



Aalto University
School of Science

Categorical associations

Experimental and Statistical Methods in
Biological Sciences I

Heini Saarimäki, BECS

16/10/2014

Outline for today

- Categorical associations
- Nonparametric tests
- Exercises part I

- Summary of the demo sessions
- Exercises part II

What we know after today

1. Preparing your data

2. Describing your data

3. T-tests and ANOVA

4. Correlations and regression

5. Categorical associations and nonparametric tests

Contingency tables

Contingency tables (from demo2)

- Frequency of observations per category

```
> table(sex, agegroup)
```

```
> round(prop.table(table(sex, agegroup), margin=1)*100,1)
```

Chi-square test

The chi-square test tests the probability that samples of observations come from the same distributions.

We explore two possibilities:

- Goodness-of-fit test: test whether the sample distribution is same as known population distribution
- Test for independence: test whether two independent samples come from the same distributions

Chi-square test

Pearson's chi-squared goodness-of-fit test

```
> chisq.test(table(sex), p=c(.5,.5))
```

```
> chisq.test(table(sex), p=c(.5,.5))
```

```
Chi-squared test for given probabilities
```

```
data: table(sex)
```

```
X-squared = 0, df = 1, p-value = 1
```

Chi-square test

Chi-squared test for independence

```
> chisq.test(sex, agegroup)
```

```
> chisq.test(sex, agegroup)
```

```
Pearson's Chi-squared test
```

```
data: sex and agegroup
```

```
X-squared = 6.0171, df = 2, p-value = 0.04936
```

```
Warning message:
```

```
In chisq.test(sex, agegroup) : Chi-squared approximation may be incorrect
```

Sex distribution is different in different age groups ($\chi^2(2) = 6.02, p < .05$).



Non-parametric tests

Mann-Whitney-Wilcoxon test (U-test)

- Independent samples t-test

```
> wilcox.test(headache_on ~ sex)
```

```
> wilcox.test(headache_on ~ sex)
```

```
Wilcoxon rank sum test with continuity correction
```

```
data: headache_on by sex
```

```
W = 285.5, p-value = 0.963
```

```
alternative hypothesis: true location shift is not equal to 0
```


Non-parametric tests

Wilcoxon's signed ranks test

- Repeated measures t-test

```
> wilcox.test(headache1, headache_on, paired=T)
```

```
> wilcox.test(headache1, headache_on, paired=T)
```

```
Wilcoxon signed rank test with continuity correction
```

```
data: headache1 and headache_on
```

```
V = 36, p-value = 0.2999
```

```
alternative hypothesis: true location shift is not equal to 0
```

Non-parametric tests

Kruskal-Wallis test

- One-way ANOVA

```
> kruskal.test(headache_on ~ agegroup)
```

```
> kruskal.test(headache_on ~ agegroup)
```

```
      Kruskal-Wallis rank sum test
```

```
data:  headache_on by agegroup
```

```
Kruskal-Wallis chi-squared = 1.7418, df = 2, p-value = 0.4186
```



Which test should I use?

Lots of cook book solutions available online – but aim for understanding the differences between each method yourself!

http://www.ats.ucla.edu/STAT/mult_pkg/whatstat/default.htm

Data for part I

Experience of subjective symptoms caused by cellphone

- baseline, cellphone on, cellphone off

Variable	Type	Description
sex	categorical	male or female
age group	categorical	three age groups
rfjarj	categorical	order of the experiment
dizziness	continuous	measured three times
headache	continuous	measured three times
tiredness	continuous	measured three times
itchiness	continuous	measured three times
blushing	continuous	measured three times
warmth	continuous	measured three times



Aalto University
School of Science

Summary of the course

Experimental and Statistical Methods in
Biological Sciences I

Heini Saarimäki, BECS

16/10/2014

Take home

- The most important tool for research is you!
 - Starting from planning the experiment...
 - ... to stating the hypotheses...
 - ... to evaluating and interpreting the results...
 - ... to drawing conclusions.
- Statistics is a tool that helps you to draw conclusions based on your data.
- And R is just a tool that runs the statistics for you.

What you should be able to do based on demos

- Formulate the hypothesis you are testing
- Choose a relevant test to test your hypothesis
- Interpret the R output to get results
- Report the results verbally, numerically (and graphically)
 - "Reading times of regular words are faster than those of exceptional words ($t(997) = -68.8, p < .001$)".

If you need help in how to report the results...

Use APA format:

- Consult lecture slides
- Consult Jeffrey Kahn's summary:
<http://my.ilstu.edu/~jhkahn/apastats.html>

Recap of the course

For each topic:

- Basic concepts
- Useful functions
- Important outputs

Structure

1. Preparing your data
2. Describing your data
3. Basics in statistical testing
4. Correlations
5. Group comparisons using T-tests
6. Group comparisons using ANOVA
7. Regression

1. Preparing your data: Before R

Standard format:

columns = variables

rows = observations (e.g., participants)

- Collect your data into one table using your favorite software (Excel, Matlab, R, ...)
- Save preferably as .csv or .txt

1. Preparing your data:

Load in data

```
# Load in data from .csv file:
```

```
> data <- read.csv('naming.csv')
```

```
# Load in data where first line is header information:
```

```
> data <- read.csv('naming.csv', header=T)
```

```
# Load in data from .txt file:
```

```
> data <- read.table('naming', header=T, sep="")
```

```
# Load in online data - here, .csv file:
```

```
> data <- read.csv('http://becs.aalto.fi/~heikkih3/naming_wide.csv',  
header=T)
```



1. Preparing your data: Checking data

```
# Check your data:  
> summary(data)  
  
# Correct for factors:  
> data$variable1 <- factor(data$variable1)  
  
# Identify funny coding of missing values:  
> summary(data)  
> which(data$variable1 == 999)  
  
# Recode missing values as NAs:  
> data$variable1[which(data$variable1 == 999)] <- NA
```

1. Preparing your data: adding/removing variables

```
# Adding new variables in the data frame  
> data$new_variable <- 0
```

```
# Removing variables from the data frame  
> data$new_variable <- NULL
```

1. Preparing your data: Saving changes into a new data frame

```
# Copy the old data frame to a new data frame:  
> data2 <- data
```

```
# Now make changes to the new data frame data2.  
> data2$variable1 <- NULL
```

```
# E.g., remove a row from data frame and save to a new dataframe at  
the same time:  
> data2 <- data[-10,] # row 10 removed from data2
```

1. Preparing your data: attach/detach data

```
# When the data frame is ready,  
> attach(data)
```

```
# When attaching a new data frame, detach the previous one  
> detach(data)
```


1. Preparing your data: saving your data

```
# Save your data frame in .csv format:  
> write.csv(data, "my_data.csv")
```



1. Preparing your data

Important outputs: `summary()`

output for continuous variables:
range (min-max),
quartiles (1st, 3rd),
median, mean

```
> summary(data)
```

```
      iq  
Min.   : 45.13  
1st Qu.: 90.42  
Median :100.02  
Mean   :100.00  
3rd Qu.:110.23  
Max.   :163.41
```

```
      hrs  
Min.   :2.424  
1st Qu.:3.646  
Median :3.999  
Mean   :4.000  
3rd Qu.:4.353  
Max.   :5.472
```

```
ms.regular  
Min.   : 701.0  
1st Qu.: 951.2  
Median : 998.7  
Mean   :1000.0  
3rd Qu.:1048.6  
Max.   :1263.1
```

```
ms.exception  
Min.   : 942  
1st Qu.:1147  
Median :1199  
Mean   :1200  
3rd Qu.:1252  
Max.   :1452
```

```
      sex  
female:457  
male   :541
```

output for factors:
frequency in each
category

2. Describing your data

- Descriptive statistics
- Frequencies
- Plotting

2. Describing your data: descriptive statistics

```
# Take a basic summary:  
> summary(data)
```

```
# Take all descriptive statistics:  
> library(psych) # load package first  
> describe(data)
```

```
# Get single descriptive statistics:  
> mean(iq)  
> sd(iq)
```

2. Describing your data: frequencies

```
# Take a frequency table:  
> table(sex)
```

```
# Take a contingency table:  
> table(sex, reading_group)
```

2. Describing your data

Important outputs:
`describe()`

`describe()` is included in
package 'psych'

look up ?describe if you
are unsure what the
displayed statistics are

```
> library(psych)
> describe(data)
```

```
          var    n   mean    sd  median trimmed   mad
iq          1 998  100.00 15.00   100.02  100.10  14.50
hrs         2 998    4.00  0.50    4.00    4.00   0.52
sex*        3 998    1.54  0.50    2.00    1.55   0.00
ms.regular  4 998 1000.00 75.00   998.69  999.14  72.56
ms.exception 5 998 1200.00 80.00 1198.70 1199.75  78.61
reading_time* 6 998    2.02  0.85    2.00    2.02   1.48
          min    max  range  skew kurtosis   se
iq          45.13 163.41 118.29 -0.05    0.53 0.47
hrs          2.42  5.47   3.05 -0.01   -0.18 0.02
sex*         1.00  2.00   1.00 -0.17   -1.97 0.02
ms.regular  700.97 1263.12 562.14  0.07    0.30 2.37
ms.exception 941.96 1452.20 510.23  0.06    0.16 2.53
reading_time* 1.00   3.00   2.00 -0.03   -1.60 0.03
```

factors are marked
with asterisks

2. Describing your data

How to save your table?

```
> desc <- round(describe(data[c(1:2,4:5)]),2)
> write.csv(desc, "descriptives.csv")
```

Open in Excel.

Choose first column, then Data → Text to columns.

2. Describing your data: Structure of plotting functions

```
# 1. Define your basic format:  
> plot(variable1, variable2)  
> hist(variable1)
```

```
# 2. Then build it to a nice graph with named arguments:  
> plot(variable1, variable2, col='red', main="My graph",  
        xlab='variable 1', ylab = 'variable 2', lty='dashed')
```



2. Describing your data: Plotting distribution of one variable

```
# Histograms:
```

```
> hist(iq)
```

```
# Adding a normal curve to the histogram:
```

```
> h<- hist(iq, freq=F)
```

```
> x <- min(h$breaks):max(h$breaks)
```

```
> y <- dnorm(x, mean=mean(iq), sd=sd(iq))
```

```
> lines(x, y, col="red", lty="dashed", lwd=2)
```

```
# Boxplots:
```

```
boxplot(iq)
```



2. Describing your data

Plotting associations between variables

```
# Scatterplots:  
> plot(iq ~ hrs)
```

```
# Add regression line to a scatterplot:  
> plot(iq ~ hrs)  
> abline(lm(iq ~ hrs))
```

2. Describing your data: Plotting categories separately

```
# Boxplots for categories separately:  
> boxplot(iq ~ sex)
```

```
# Scatterplots for categories separately:  
> coplot(iq ~ hrs | sex)  
> library(lattice)  
> xyplot(iq ~ hrs | sex)    # from package 'lattice'
```

2. Describing your data: Saving your plots

1. Switch the device to a file (the command you use depends on the graphics format you want: pdf is good, or you can find a whole list with ?Devices);

```
> png(filename="my_plot.png")
```

2. Run the commands which create the graphics;

```
> plot(iq ~ hrs)
```

3. Switch the file device off (which switches back to the screen by default)

```
> dev.off()
```

3. Basics in statistical testing

Idea behind all statistical testing:

You are testing a null hypothesis. If the test statistic is significant (usually $p < .05$) you have enough confidence to *reject* the null hypothesis.

3. Basics in statistical testing

Underlying assumptions

- T test assumes:
 - Normality, sample size > 20
- ANOVA assumes:
 - Normality, sample size > 15 , equality of variances
- Regression assumes:
 - Normality, linearity, sample size > 50 , multicollinearity
- What if I cannot meet the assumptions?
 - consider data transformations or non-parametric tests!

3. Basics in statistical testing

Testing for normality

```
# Descriptive statistics:
```

```
> describe(iq)
```

```
# Histogram
```

```
> hist(iq)
```

```
# Quantile-quantile plot
```

```
> qqnorm(iq)
```

```
> qqplot(iq)
```

```
# Normality tests
```

```
> ks.test(iq, "pnorm", mean=mean(iq), sd=sd(iq))
```

```
> shapiro.test(iq)
```



Testing for assumptions

Important outputs:
`describe()`

mean close to median
suggests normality

```
> describe(data[c(1:2,4:5)])
```

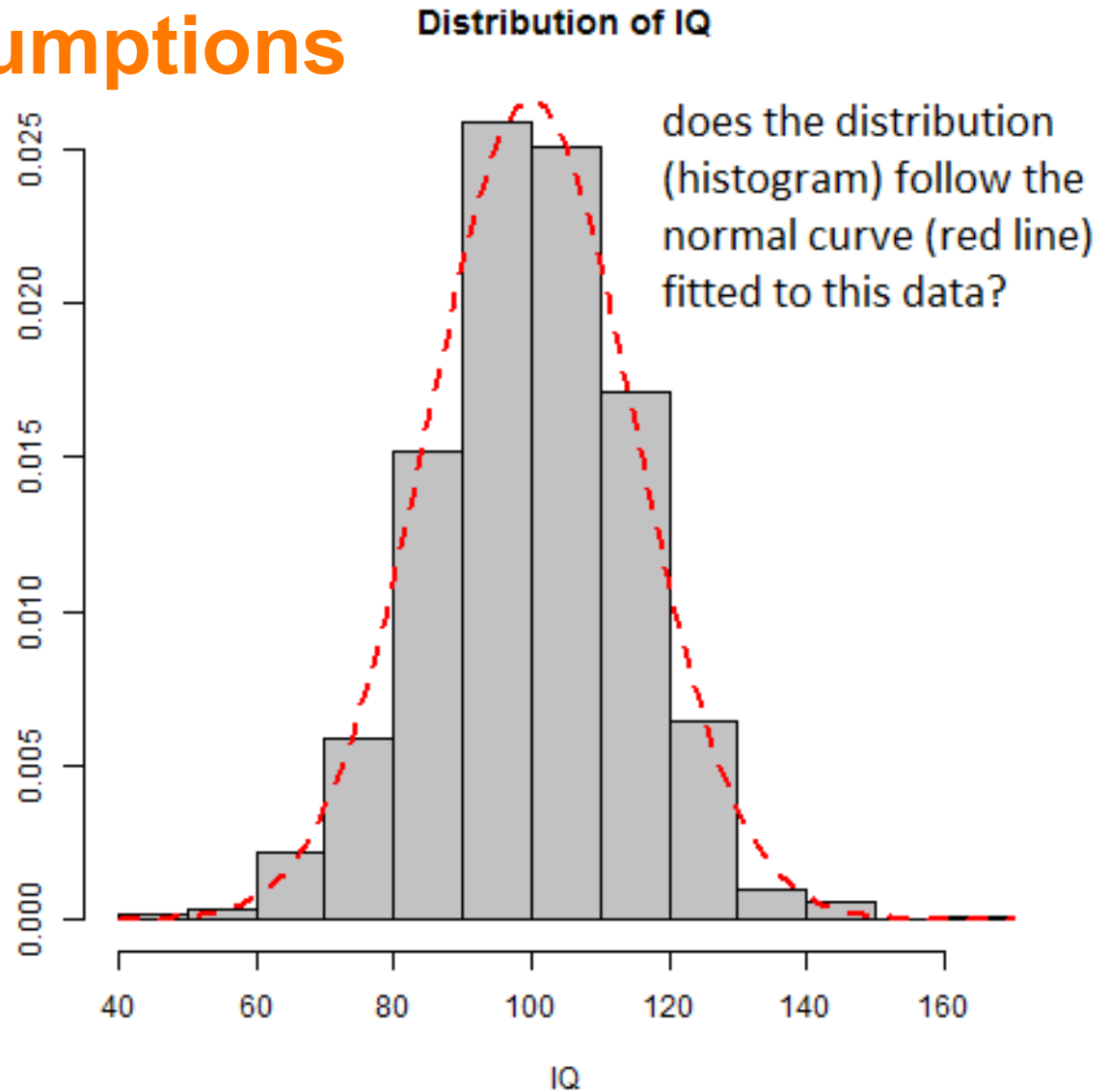
	var	n	mean	sd	median	trimmed	mad			
iq	1	998	100	15.0	100.02	100.10	14.50			
hrs	2	998	4	0.5	4.00	4.00	0.52			
ms.regular	3	998	1000	75.0	998.69	999.14	72.56			
ms.exception	4	998	1200	80.0	1198.70	1199.75	78.61			
	min	max	range	skew	kurtosis	se				
iq	45.13	163.41	118.29	-0.05	0.53	0.47				
hrs	2.42	5.47	3.05	-0.01	-0.18	0.02				
ms.regular	700.97	1263.12	562.14	0.07	0.30	2.37				
ms.exception	941.96	1452.20	510.23	0.06	0.16	2.53				

skewness and kurtosis
close to 0 suggest
normality

Testing for assumptions

Important outputs:

```
hist()  
lines()
```

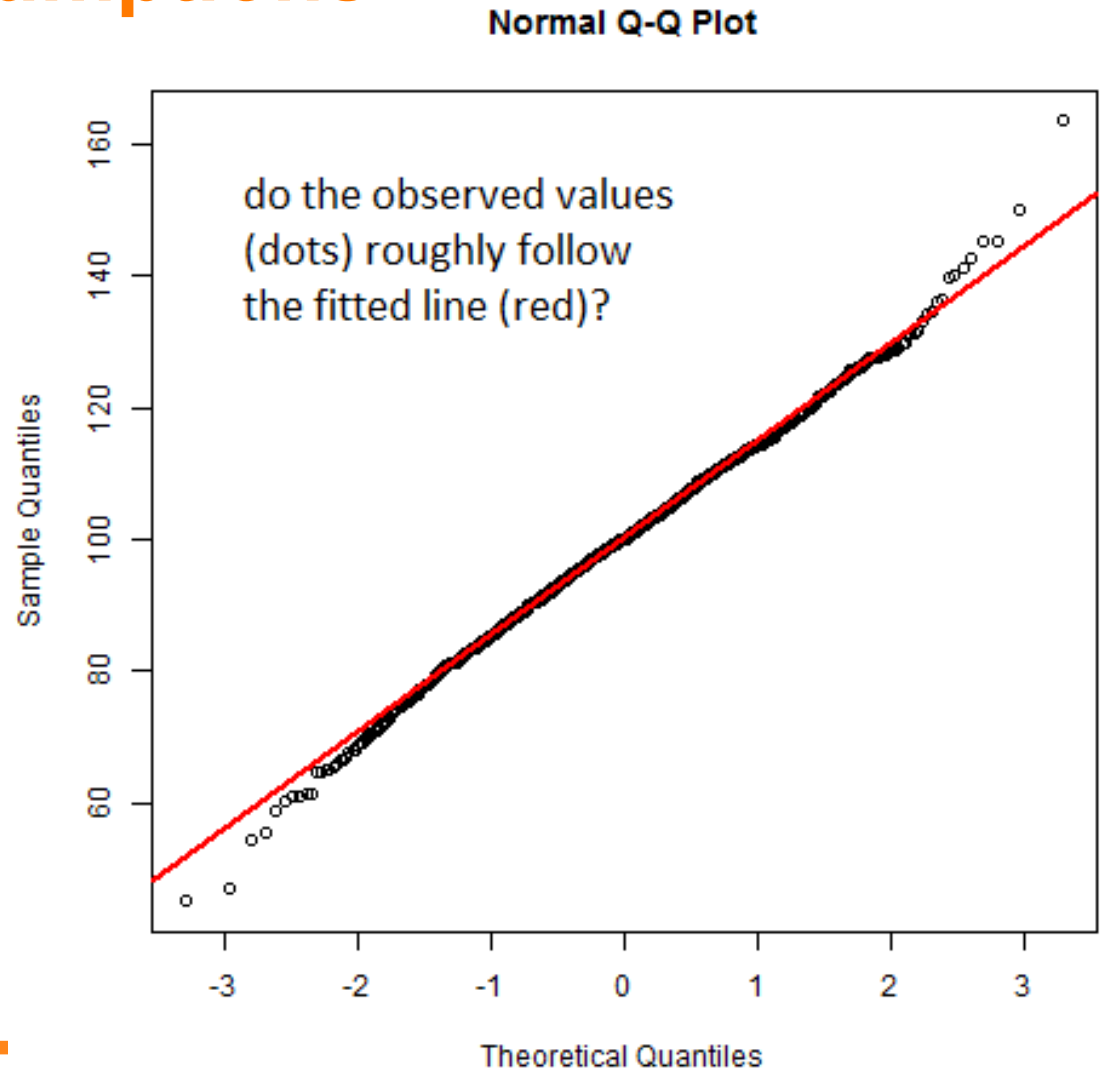


Testing for assumptions

Important outputs:

`qqnorm()`

`qqline()`



Testing for assumptions

Important outputs: `ks.test()` or `shapiro.test()`

tests the null hypothesis that the distributions for the first variable (here, "iq") and for the second variable (here, "pnorm", i.e. normal distribution with mean and sd based on "iq") come from the same distribution

```
> ks.test(iq, "pnorm", mean=mean(iq), sd=sd(iq))
```

One-sample Kolmogorov-Smirnov test

```
data: iq  
D = 0.0196, p-value = 0.8386  
alternative hypothesis: two-sided
```

p values > .05 suggest that the null hypothesis cannot be rejected, i.e., in both tests that the test suggest normality

tests the null hypothesis that the sample (here, "iq") comes from a normally distributed population

```
> shapiro.test(iq)
```

Shapiro-Wilk normality test

```
data: iq  
W = 0.997, p-value = 0.05761
```

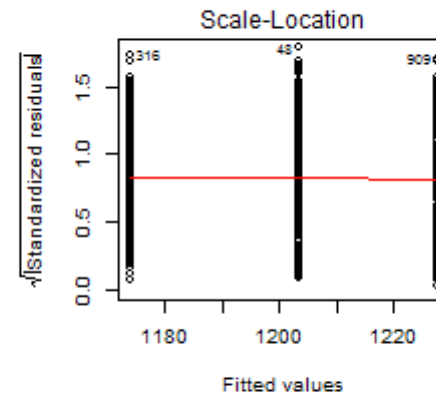
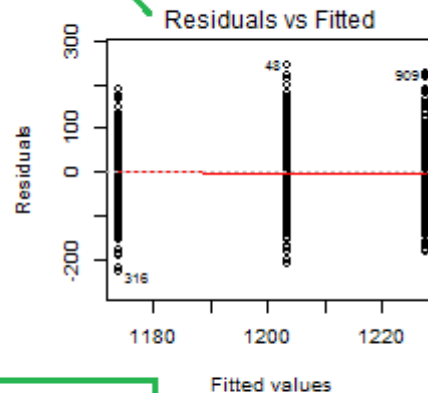
Testing for assumptions

- Diagnostic plots (regression, ANOVA):
 - Equality of variances
 - Influential cases
 - Normality (in ANOVA)
 - Linearity (in regression)
- Other assumptions:
 - Multicollinearity (in regression)
 - correlation matrix (no really high correlations)

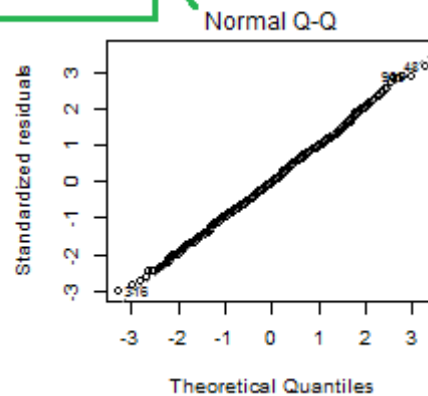
Testing for assumptions

Important outputs:
diagnostic plots
(here, ANOVA)

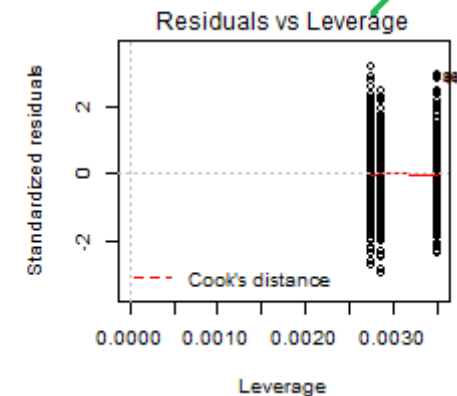
are the variances equal?



is the distribution normal?



are there highly influential cases?



4. Correlations

Tests for associations between numeric variables: how much do the variables covary.

Null hypothesis:

The true correlation between the variables is equal to 0.

4. Correlations: Pearson product-moment correlation coefficient

```
# Correlation between two variables:  
> cor.test(iq, hrs)
```

```
# Correlation matrix for all numeric variables:  
> corr.test(data[c(1:2,4:5)])
```

4. Correlations:

Spearman rank-correlation coefficient

```
# Correlation between two variables:  
> cor.test(iq, hrs, method="spearman")  
  
# Correlation matrix for all numeric variables:  
> corr.test(data[c(1:2,4:5)], method="spearman")
```


4. Correlations

Important outputs:

`cor.test()`

```
> cor.test(iq, hrs)
```

```
        Pearson's product-moment correlation
```

```
data:  iq and hrs
```

```
t = 0, df = 996, p-value = 1
```

```
alternative hypothesis: true correlation is not equal to 0
```

```
95 percent confidence interval:
```

```
-0.0620552  0.0620552
```

```
sample estimates:
```

```
      cor  
1.379778e-15
```

p value tells you whether you can safely reject the null hypothesis

— correlation coefficient



4. Correlations

Important outputs:
`corr.test()`

```
> corr.test(data[c(1:2,4:5)])  
Call:corr.test(x = data[c(1:2, 4:5)])
```

Correlation matrix

	iq	hrs	ms.regular	ms.exception
iq	1.0	0.0	-0.6	-0.4
hrs	0.0	1.0	-0.1	-0.3
ms.regular	-0.6	-0.1	1.0	0.3
ms.exception	-0.4	-0.3	0.3	1.0

correlation coefficient for each pairwise comparison (ranges between -1 and 1) similar below and above the diagonal

Sample Size

	iq	hrs	ms.regular	ms.exception
iq	998	998	998	998
hrs	998	998	998	998
ms.regular	998	998	998	998
ms.exception	998	998	998	998

sample size for each pairwise comparison

Probability values (Entries above the diagonal are adjusted for multiple tests.)

	iq	hrs	ms.regular	ms.exception
iq	0	1	0	0
hrs	1	0	0	0
ms.regular	0	0	0	0
ms.exception	0	0	0	0

above diagonal = adjusted for multiple comparisons

p-value for each pairwise comparison -

below diagonal = unadjusted for multiple comparisons,

5. Comparing means with t tests

Null hypothesis:

The true differences in means is equal to 0.

Compare the means between

- Sample and population
→ one-sample t-test, Mann-Whitney-Wilcoxon test
- Independent samples
→ independent samples t-test, Mann-Whitney-Wilcoxon test
- Repeated measures
→ repeated measures t-test, Wilcoxon signed-rank test

5. Comparing means with t tests: parametric tests

```
# One-sample t-test
```

```
> t.test(iq, mu=100)
```

```
# Independent samples t-test
```

```
> t.test(iq ~ sex)
```

```
# Repeated-measures t-test
```

```
> t.test(ms.regular, ms.exception, paired=T)
```

5. Comparing means with t tests: parametric tests

Important outputs: one-sample t-test

values you need to report:
t value, degrees of freedom, p

```
> t.test(iq, mu=100)
```

```
One Sample t-test
```

```
data: iq
```

```
t = 0, df = 997, p-value = 1
```

```
alternative hypothesis: true mean is not equal to 100
```

```
95 percent confidence interval:
```

```
99.06825 100.93175
```

```
sample estimates:
```

```
mean of x
```

```
100
```

p value tells you
whether you can reject
the null hypothesis

sample mean

5. Comparing means with t tests: parametric tests

Important outputs: independent samples t-test

values you need to report:

t value, degrees of freedom, p

```
> t.test(iq ~ sex)
```

```
Welch Two Sample t-test
```

```
data: iq by sex
```

```
t = 1.0684, df = 960.965, p-value = 0.2856
```

```
alternative hypothesis: true difference in means is not equal to 0
```

```
95 percent confidence interval:
```

```
-0.853577  2.893535
```

```
sample estimates:
```

mean in group female	mean in group male
100.55291	99.53294

p value tells you whether
you can reject the null
hypothesis

group means

5. Comparing means with t tests: parametric tests

Important outputs: repeated measures t-test

values you need to report:

t value, degrees of freedom, p

p value tells you whether you can reject the null hypothesis

```
> t.test(ms.regular, ms.exception, paired=T)

Paired t-test

data: ms.regular and ms.exception
t = -68.8352, df = 997, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-205.7016 -194.2984
sample estimates:
mean of the differences
-200
```

95% confidence interval for the actual mean of differences

mean of differences

5. Comparing means with t tests: non-parametric tests

```
# One sample: Mann-Whitney-Wilcoxon Test
```

```
> wilcox.test(iq, mu=100)
```

```
# Independent samples: Mann-Whitney-Wilcoxon Test
```

```
> wilcox.test(iq ~ sex)
```

```
# Repeated-measures: Wilcoxon Signed-Rank Test
```

```
> wilcox.test(ms.regular, ms.exception, paired=T)
```



5. Comparing means with t tests: non-parametric tests

Important outputs: Mann-Whitney-Wilcoxon test

(similar principle for Wilcoxon signed-rank test)

test value

p value tells you whether you
can safely reject the null
hypothesis

```
> wilcox.test(iq ~ sex)
```

```
Wilcoxon rank sum test with continuity correction
```

```
data: iq by sex
```

```
W = 127645, p-value = 0.3749
```

```
alternative hypothesis: true location shift is not equal to 0
```

6. Comparing groups with ANOVA

Null hypothesis:

The means in all groups are equal.

One group-level variable

→ one-way between-subjects ANOVA

Two or more group-level variables

→ two-way between-subjects ANOVA



6. Comparing groups with ANOVA

- Which pairwise differences between groups cause the significant group differences?
 - *[Planned comparisons: contrasts (preferred)]*
 - Post hoc comparisons

6. Comparing groups with ANOVA: post-hoc comparisons

- If ANOVA shows a surprising significant group difference, and you want to know where the pairwise differences are
- Several options for post-hoc comparisons
 - Pairwise t-test with correction for multiple comparisons

6. Comparing groups with ANOVA: one-way ANOVA

```
# Save and run anova model  
> model.anova <- aov(ms.exception ~ reading_time)  
  
# Get sums of squares  
> summary(model.anova)  
  
# Get coefficients if needed  
> summary.lm(model.anova)  
  
# Take the means for each group  
> tapply(ms.exception, reading_time, mean)
```

6. Comparing group with ANOVA: post-hoc comparisons

```
# Bonferroni-corrected:  
> pairwise.t.test(ms.exception, reading_time,  
p.adj="bonf")  
# Look for other ways to correct for multiple comparisons  
# using the same function!
```

```
# Tukey's test:  
> TukeyHSD(model.anova)
```

6. Comparing groups with ANOVA: two-way ANOVA

```
# Save and run anova model
> model.anova.3 <- aov(ms.exception ~ reading_time * sex)

# Get sums of squares
> summary(model.anova.3)

# Get coefficients if needed
> summary.lm(model.anova.3)

# Plot the results
interaction.plot(model.anova.3)
```

6. Comparing groups with ANOVA

Important outputs: summary for one-way ANOVA

values you need to report:
F value, degrees of freedom, p

p value tells you whether
you can reject the null
hypothesis

```
> summary(model.anova)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
reading time	2	463754	231877	38.99	<2e-16 ***
Residuals	995	5917046	5947		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

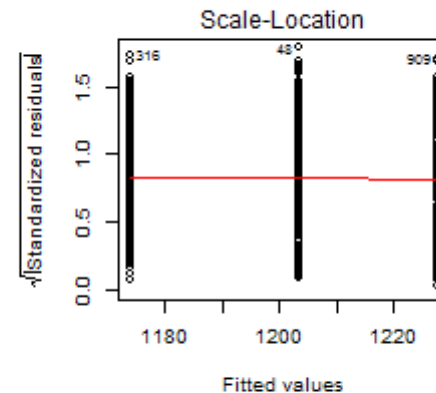
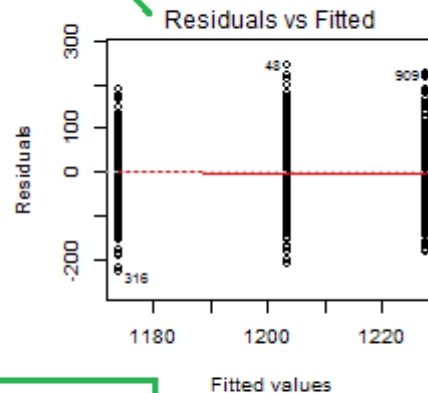
categorical variable

stars tell you about the
significance level of p

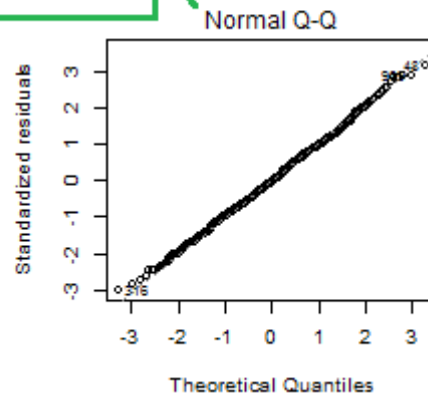
6. Comparing groups with ANOVA

Important outputs:
diagnostic plots

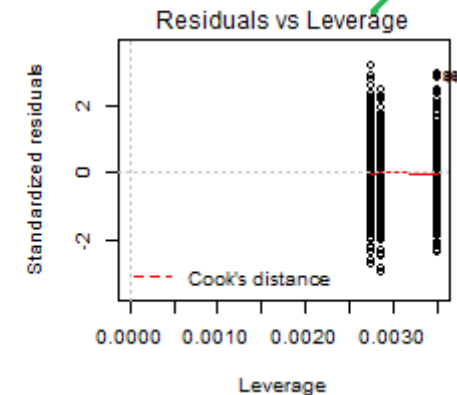
are the variances equal?



is the distribution normal?



are there highly influential cases?



6. Comparing groups with ANOVA

Important outputs: results for contrast comparison using `summary.lm()`

contrasts specified using `contrasts()`

```
> summary.lm(anova.model.2)
```

```
Call:  
aov(formula = ms.exception ~ reading_time)
```

```
Residuals:  
      Min       1Q   Median       3Q      Max  
-231.781  -52.773   -2.025   54.068  246.614
```

p value < .05 tells you the contrast is significant

```
Coefficients:  
              Estimate Std. Error t value Pr(>|t|)  
(Intercept)  1201.621     2.455 489.442 < 2e-16 ***  
reading_time1 -27.879     3.424  -8.142 1.15e-15 ***  
reading_time2  12.205     3.048   4.004 6.68e-05 ***
```

```
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 77.12 on 995 degrees of freedom  
Multiple R-squared: 0.07268,    Adjusted R-squared: 0.07082  
F-statistic: 38.99 on 2 and 995 DF,  p-value: < 2.2e-16
```

6. Comparing groups with ANOVA

Important outputs: post-hoc comparison using `pairwise.t.test()`

```
> pairwise.t.test(ms.exception, reading_time, p.adj="bonf")
```

```
Pairwise comparisons using t tests with pooled SD
```

```
data: ms.exception and reading_time
```

	high	low
low	< 2e-16	-
medium	1.1e-06	2e-04

p values: p value < .05
tells you the pairwise
difference is significant

```
P value adjustment method: bonferroni
```

6. Comparing groups with ANOVA

Important outputs: post-hoc comparison using `TukeyHSD()`

```
> TukeyHSD(model.anova)
Tukey multiple comparisons of means
 95% family-wise confidence level

Fit: aov(formula = ms.exception ~ reading_time)

$reading_time
      diff      lwr      upr      p adj
low-high  54.02305  39.56251  68.48359 0.0000000
medium-high 29.61292  16.05155  43.17428 0.0000011
medium-low -24.41013 -38.71825 -10.10202 0.0001971
```

p value < .05 tells you that the pairwise group difference is significant

pairwise comparisons

6. Comparing groups with ANOVA

Important outputs: summary for two-way ANOVA

values you need to report:
F value, degrees of freedom, p

p value tells you whether
the main / interaction
effects are significant

```
> model.anova.3 <- aov(ms.exception ~ reading_time * sex)
> summary(model.anova.3)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
reading_time	2	463754	231877	39.031	<2e-16 ***
sex	1	7884	7884	1.327	0.250
reading_time:sex	2	15793	7897	1.329	0.265
Residuals	992	5893369	5941		

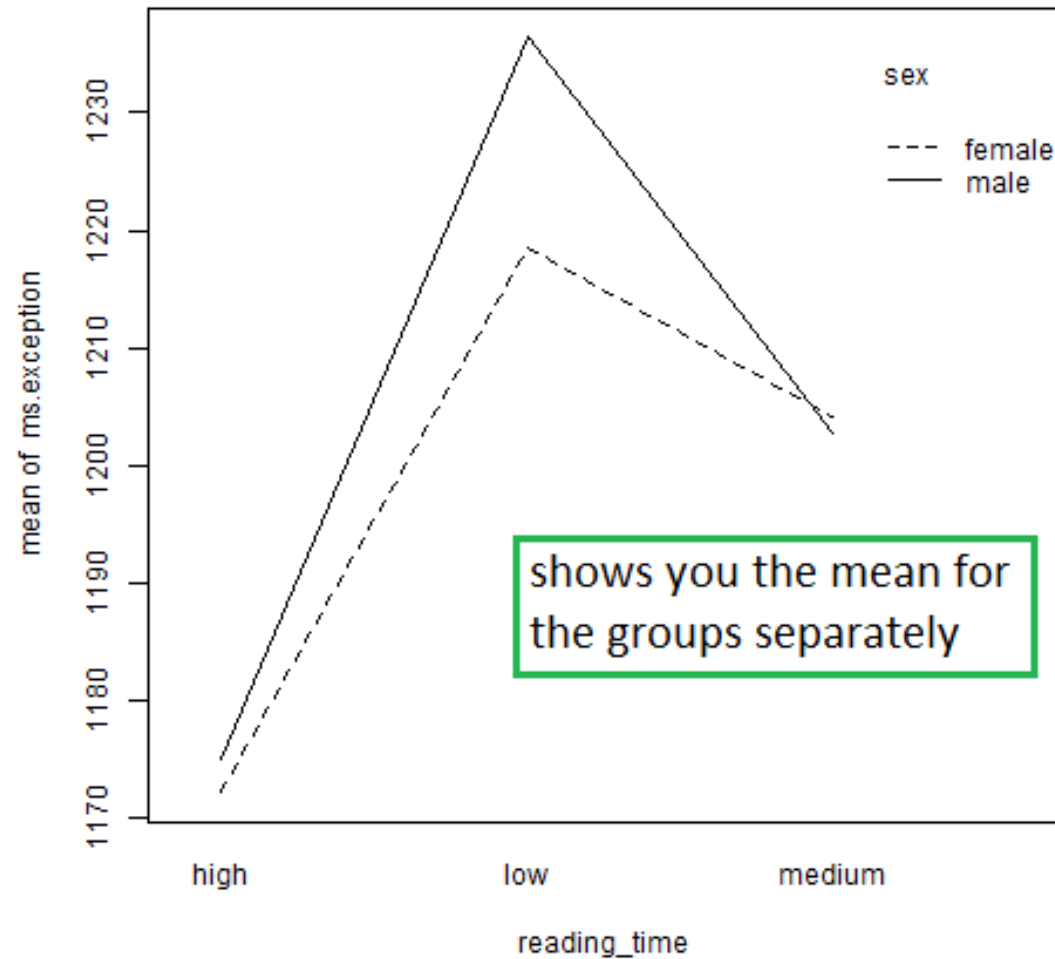
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

main effects

interaction
effect

6. Comparing groups with ANOVA

Important outputs:
`interaction.plot()`



7. Regression

Does variance in one or more predictors account for variance in another variable?

7. Regression

```
# Model with one predictor:  
> model1 <- lm(ms.regular ~ iq)  
  
# Summary of the regression results:  
> summary(model1)  
  
# Model with two predictors:  
> model2 <- lm(ms.regular ~ iq * hrs)  
  
# Compare models:  
> anova(model1, model2)
```


7. Regression

Important outputs: `summary(model)`

```
> model1 <- lm(ms.regular ~ iq)
> summary(model1)
```

Call:
lm(formula = ms.regular ~ iq)

Residuals:

Min	1Q	Median	3Q	Max
-185.08	-41.56	-1.15	40.58	180.77

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1300.0000	12.8161	101.44	<2e-16 ***
iq	-3.0000	0.1267	-23.67	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 60.03 on 996 degrees of freedom
Multiple R-squared: 0.36, Adjusted R-squared: 0.3594
F-statistic: 560.3 on 1 and 996 DF, p-value: < 2.2e-16

coefficients

b0

predictor b1

p value < .05 suggests that the predictor is significant

amount of variance explained by the model

does the model fit the data?
F value, degrees of freedom, p

7. Regression

Important outputs: `anova(model1,model2)`

```
> anova(model1, model2)
Analysis of Variance Table
```

```
Model 1: ms.regular ~ iq
Model 2: ms.regular ~ iq * hrs
```

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	996	3589200				
2	994	3524996	2	64204	9.0524	0.000127 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

p value < .05 suggest that the second model has better model fit than the first model

F	Pr(>F)
9.0524	0.000127 ***

8. Extra topics: multiple comparisons

Goal: reduce the number of false positives given by the statistical test

E.g.

- Structural MRI: Testing for group differences in 10 different regions-of-interest
- Functional MRI: testing for differences between conditions in the whole brain (i.e., running a separate linear model for each voxel)
- Genomics: testing for multiple genes, environmental factors, their interactions...

<http://www.biostathandbook.com/multiplecomparisons.html>

Finally

- Good luck with the exam
- Remember to give feedback (Oodi)
- See you in spring 2015 at Experimental and Statistical Methods in Biological Sciences II

Our final dataset: NHEFS

- Personality variables (continuous):
 - Neuroticism
 - Extraversion
 - Openness
- Depression variable (continuous):
 - CES-D
- Health variables (categorical):
 - Heart disease
 - Cancer
 - Stroke
 - Diabetes
 - Hypertension

