Leibniz Universität Hannover

Mensch-Computer-Interaktion 1
(Mensch-Maschine-Kommunikation):

Evaluation 2

Human-Computer Interaction Group

Prof. Dr. Michael Rohs
michael.rohs@hci.uni-hannover.de

Vorlesungen

Termin	Datum	Thema
1	18.10.	Introduction
2	25.10.	History and Paradigms of HCI
3	1.11.	Human Information Processing
4	8.11.	Input and Output Technologies
5	15.11.	Models of Interaction
6	22.11.	Interaction Design Process
7	29.11.	Understanding and Envisionment
8	6.12.	Prototyping
9	13.12.	Design Rules and HCI Principles
10	20.12.	Evaluation
11	10.1.	Evaluation
12	17.1.	Dialog Design Notation
13	24.1.	Information Design
14	31.1.	Web Usability

Klausur (90 Minuten):
Mi, 1.3., 9-11 Uhr

Königsworther Platz 1 (Conti-Campus, Hörsaal-Gebäude): Geb. 1507, Räume 201 und 002

Online-Prüfungsanmeldung: 6.1.-20.1.

Review

- Lab vs. field studies?
- Evaluation techniques without users?
  - CW, HE, LR, MbE
- Evaluation techniques with users?
  - Qualitative: CME, SO, TA, CI, RT
  - Quantitative: Experiments
- Experiments
  - Independent vs. dependent variables?
  - Hypothesis
  - Difference between-groups and within-groups study?

Evaluation Techniques

Evaluating Without Users

E1 Literature Review
E2 Cognitive Walkthrough
E3 Heuristic Evaluation
E4 Model-Based Evaluation

Evaluating With Users

Qualitative

E5 Conceptual Model Extraction
E6 Silent Observation
E7 Think Aloud
E8 Constructive Interaction
E9 Retrospective Testing

Quantitative

E10 Controlled Experiments

+ Interviews, questionnaires,...

Preview

- Logging
- Summarizing data
  - Mean, median
  - Standard deviation
  - Range, min, max, quartiles
- Exploring data graphically
  - Box plots
  - Histograms
  - Scatter plots
- Testing whether results are systematic
  - Statistical analysis

DATA LOGGING

Logging

- Poor logging results in loss of valuable information
- Use human-readable format (comma separated values, CSV)
  - Easy to open with spreadsheet program
- Log everything, including meta data
  - Ease of analysis over avoidance of redundancy
  - Avoid need for assemble data from multiple sources
  - Anonymize data: User IDs, not real names
- Self-contained log files
  - Header row with well named variables
  - For each variable: Coding, units, granularity / precision
  - Meta data: Date, time, location, experimenter, project name
  - Criterion: Somebody else should be able to interpret the data a year later

Logging

- Debugging
  - Do pilot tests, check whether the logged data make sense
- Logging must not slow down test program
  - For experiments in which precise timing is important
- Minimize human intervention
  - Experimenter needs to focus on other things during test
  - Minimize possibility for errors

Logging

- Backups
  - As soon as possible
  - One folder for each experiment
  - Name files with date and time, potentially user ID
  - Include all material
    - Source code of test software, raw data, consent form, task descriptions, invitations, scripts for processing data, etc.
- Use version control for data and test software
  - Connects experimental data with exact version of test software

Preparing Data for Statistical Analysis

- Cleaning up data
  - Sanity check for values (ranges, consistency)
    - Important for survey data (e.g. age 233)
    - Important for (buggy) experimental prototypes (e.g. duration -3 ms)
  - If not correctable, remove dubious data
- Coding data
  - Harmonize formatting
  - Replace textual data by numeric data (if reasonable)
- Organizing data
  - Arrange columns such that can be processed by software

Coding Data

- Original data

Participant	Age	Gender	Degree	Knows Software A
Frank	23	male	College	yes
Mary	27	female	Graduate	yes
Marc	31	male	High school	no

- Coded data

Participant	Age	Gender	Degree	Knows Software A
1	23	1	2	1
2	27	0	3	1
3	31	1	1	0

- Remember the mapping!

Organizing Data

- Typically tabular form, redundancy is not an issue
- Each line has full information
  - user, trial, start/end time of trial, state of IVs, measured DVs

independent variables (conditions)

dependent variables (measured)

select	user	bg	bgIdx	count	countIdx	trialIdx	cursorX	cursorY	startTime	endTime	duration	correct
1	1	0	0	2	0	0	291	53	883062	883276	3.34375	1
1	1	0	0	2	0	1	329	319	883396	883980	9.125	1
1	1	0	0	2	0	2	441	78	884075	884409	5.21875	1
1	1	0	0	2	0	3	856	289	884533	884965	6.75	1
1	1	0	0	2	0	4	1230	349	885065	885670	9.453125	1
1	1	0	0	2	0	5	1148	180	885768	886157	6.078125	1
1	1	0	0	2	0	6	717	219	886264	886679	6.484375	1
1	1	0	0	2	0	7	650	544	886779	887271	7.6875	1
[illegible]	[illegible]	0	0	4	1	0	171	307	887415	888489	16.78125	1
[illegible]	[illegible]	0	0	4	1	1	1272	462	888591	889674	16.92188	1
[illegible]	[illegible]	0	0	4	1	2	1091	217	890026	890684	10.28125	1
[illegible]	[illegible]	0	0	4	1	3	663	312	890773	891954	18.45313	1
[illegible]	[illegible]	0	0	4	1	4	800	144	892043	892817	12.09375	1
1	1	0	0	4	1	5	127	233	892896	893765	13.57813	1
1	1	0	0	4	1	6	430	326	893881	894649	12	1
1	1	0	0	4	1	7	172	504	894727	895804	16.82813	1
1	1	0	0	8	2	0	1230	246	895948	896950	15.65625	1

each line provides full information

DESCRIPTIVE STATISTICS

Descriptive Statistics

- Summarizing data in order to understand it
  - Raw data too complex to draw conclusions
  - Characterize the data in a few meaningful numbers
- Central tendency
  - Mean, average
  - Median
- Spread
  - Standard deviation, variance
  - Range
  - Quartiles

Mean, Average

- Mean is also called arithmetic average
- data = { 45, 19, 22, 33, 29, 15, 50 }
- n = count(data) = 7
- mean = sum(data) / count(data) = 213 / 7 ≈ $30.\overbrace{42857143}^{\text{how many decimal digits?}}$

- Symbol: μ (whole population), $\bar{x}$ (sample from population)

- Affected by outliers

Median

- Median is the middle value in the sorted set
  - Divides data into two halves of equal size
- data = { 45, 19, 22, 33, 29, 15, 50 }
- sorted data = $\underbrace{15, 19, 22}_{\text{3 left}}, \underset{\text{median}}{29}, \underbrace{33, 45, 50}_{\text{3 right}}$
- median(data) = 29

- $data_2$ = { 1, 2 }
- median($data_2$) = 1.5

- Immune against outliers

Standard Deviation, Variance

- Standard deviation is a measure of variability about the mean
- Variance is the squared standard deviation
- Example
  - data = { 45, 19, 22, 29, 15 }
  - mean(data) = 26
  - d = data - mean(data) = { 19, -7, -4, 3, -11 }
  - s = sum(d^2) = sum({ 361, 49, 16, 9, 121}) = 556
  - **σ** = sqrt(s / count(data)) = sqrt(556 / 5) ≈ 10.545
- Formula

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(data_i - \overline{data}\right)^2}$$

1/(n–1) if data is a sample from a (larger) population

Range

- Difference between the largest and the smallest values
- data = { 45, 19, 22, 33, 29, 15, 50 }
- range(data) = max(data) - min(data) = 50 - 15 = 35

- Extremely vulnerable to outliers

Quartiles

- Quartiles are 3 points dividing data into 4 equally-sized groups
- data = { 45, 19, 22, 33, 29, 15, 50 }
- $data_{sorted}$ = { 15, 19, 22, 29, 33, 45, 50 }

Q_1 Q_2 Q_3

25% of data 50% of data 25% of data

  - Q_1: lower quartile
  - Q_2: median
  - Q_3: upper quartile
- Interquartile range = Q_3(data) - Q_1(data)
  - Robust in case of outliers
- Generalization: x%-quantile, x% are smaller than that value

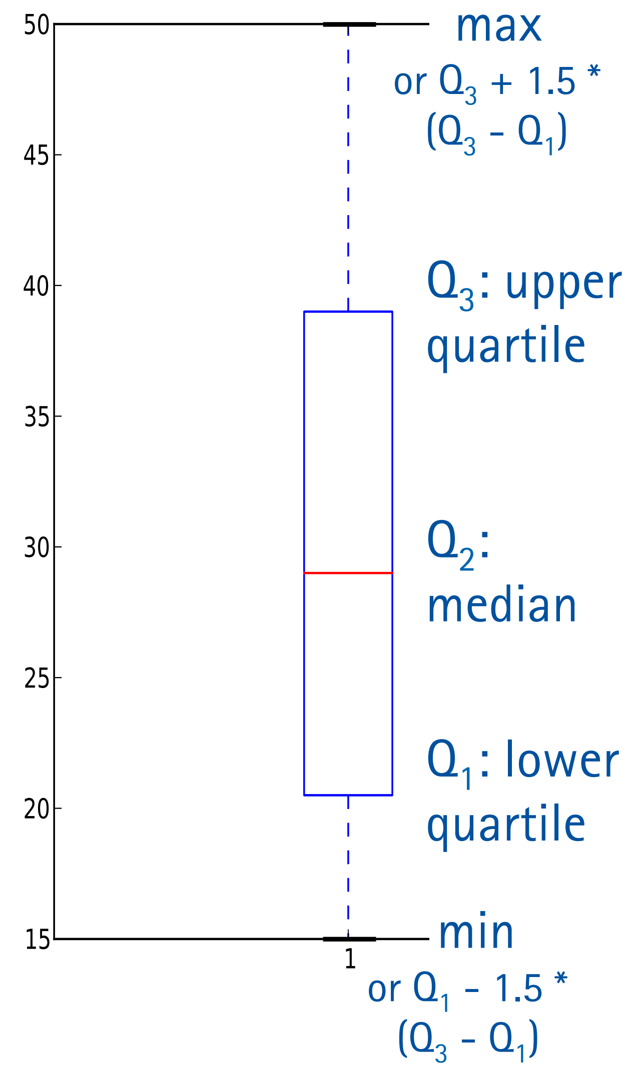
Graphical Analysis

- Box plot
- Histogram
- Scatterplot

Boxplots

- Helpful to get "big picture" of data
  - Median
  - Interquartile range (Q_3 - Q_1)
  - Min, max, outliers
  - Skewness: Median not centered in box
- $data_{sorted}$ = { 15, 19, 22, 29, 33, 45, 50 }

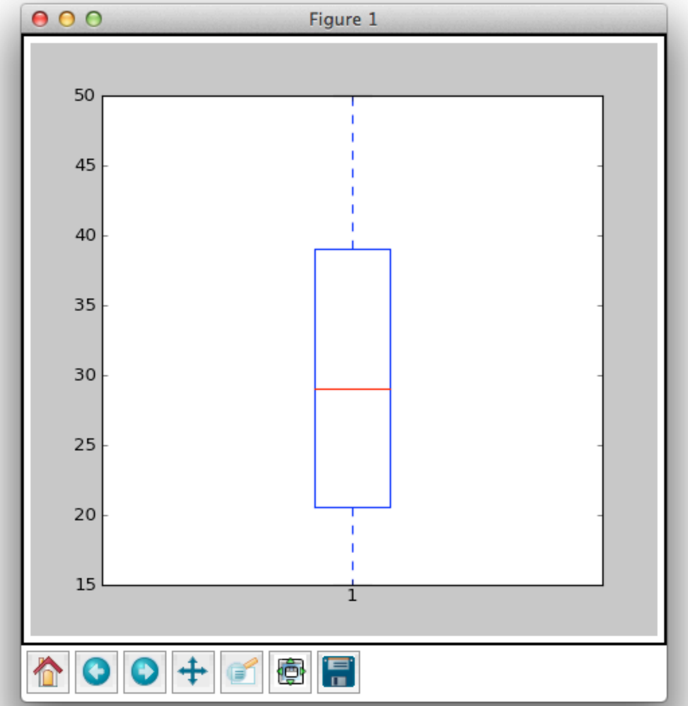
  Q_1 Q_2 Q_3
  - $Q_2 = Q_{50\%} = data_{sorted}[50\%*len] = 29$
  - $Q_1 = Q_{25\%} = data_{sorted}[25\%*len] =$ $data_{sorted}[1.5] = (19+22)/2 = 20.5$
  - $Q_3 = Q_{75\%} = data_{sorted}[75\%*len] =$ $data_{sorted}[4.5] = (33+45)/2 = 39$



Producing Boxplots

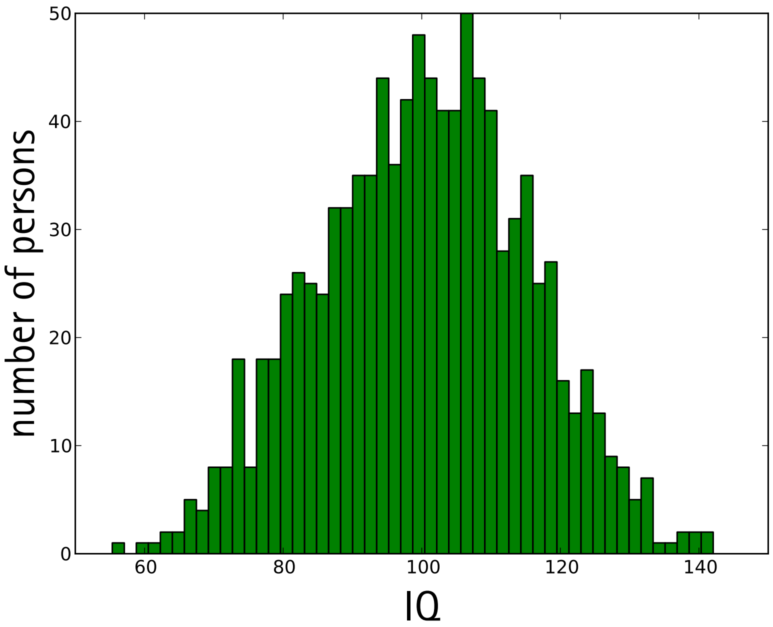
- Numpy and Matplotlib
  - http://matplotlib.org/examples/pylab_examples/boxplot_demo.html
  - Python & Eclipse Plugin
  - or iPython Notebook (browser-based interactive notebook, inline graphics) http://ipython.org/notebook.html
- Example

```
import numpy as np
import matplotlib.pyplot as pp
data = np.array([ 45, 19, 22, 33, 29, 15, 50 ])
pp.figure()
pp.boxplot(data)
pp.show()
```



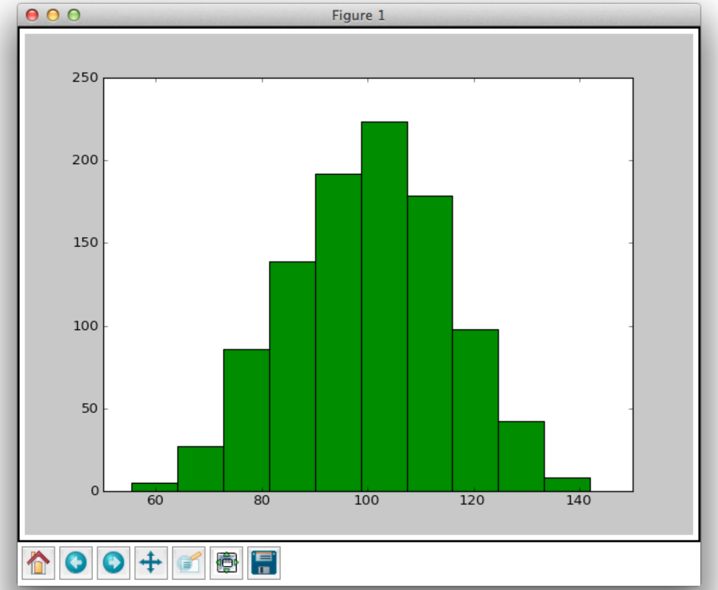
Histogram

- Estimate of probability distribution of a random variable
  - x-axis: adjacent intervals (bins)
  - y-axis: frequencies of occurrence
- Area of each bar: number of observations in this interval
  - Height: divide number of observations by width of interval
- To estimate probability density, normalize total area to 1
  - Then: area of each bin equal to portion that falls into that interval



Producing Histograms

```
import numpy as np
import matplotlib.pyplot as pp

mu, sigma = 100, 15
x = mu + sigma * np.random.randn(1000) # array of 1000 floats, from normal distribution

pp.figure()
pp.hist(x, bins=10, facecolor='green')

pp.figure()
pp.hist(x, bins=50, facecolor='green')

pp.figure()
pp.hist(x, bins=50, facecolor='green', normed=1)

pp.show()
```

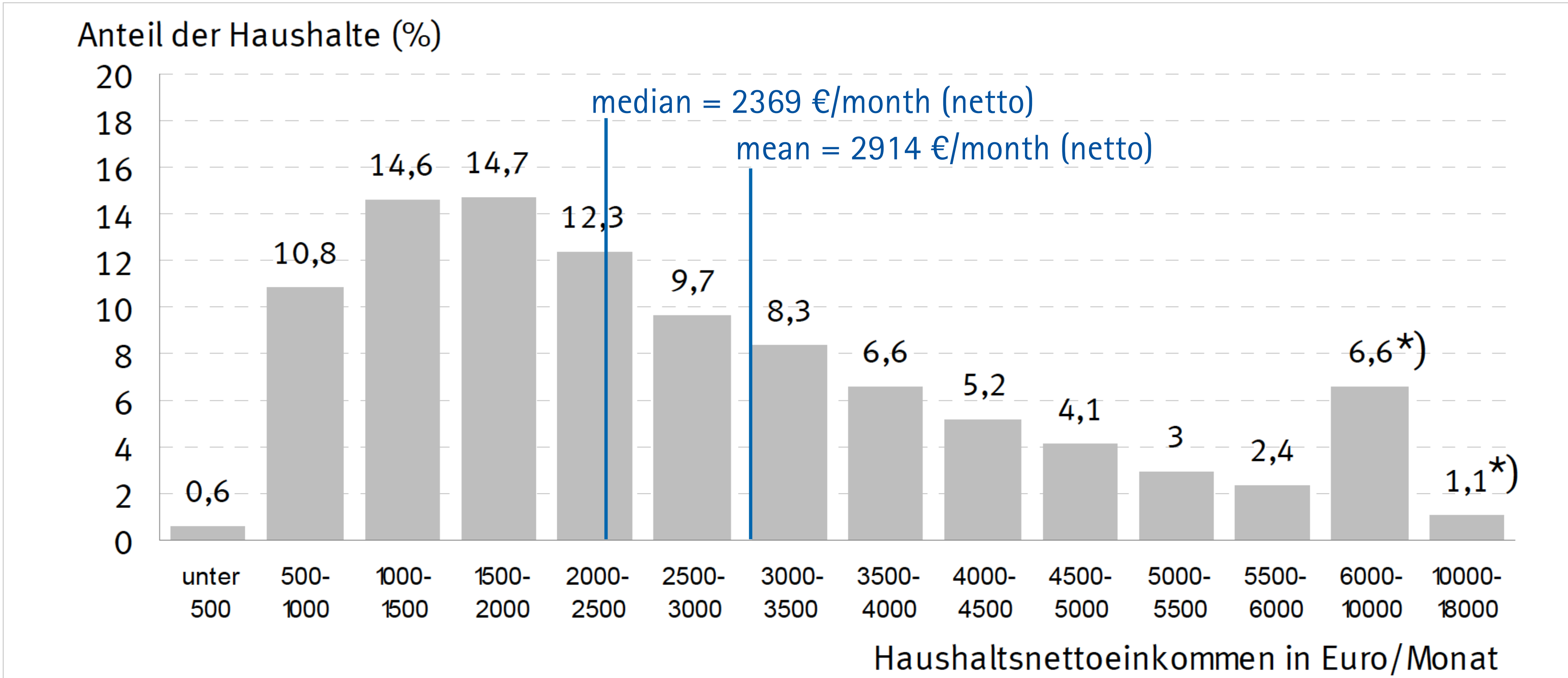


Mean vs. Median

- Median assets of German households (Source: Deutsche Bundesbank)
  - Germany: 51.400 €
  - East: 21.400 € per household
  - West: 78.900 € per household
- Mean
  - Germany: 195.200 Euro
  - West: 230.240 Euro
- Explanation, Spiegel Online:
  - "Die Bundesbank berechnete für die Untersuchung die mittleren Vermögen der Haushalte mit dem sogenannten Median. Dabei handelt es sich zwar um einen Mittelwert, der aber auf andere Weise ermittelt wird als der herkömmliche Durchschnittswert, das arithmetische Mittel. Der Median wird weniger stark durch Ausreißerwerte nach oben und unten verzerrt. Besonders reiche oder arme Haushalte fallen also weniger ins Gewicht."

Mean vs. Median

Abbildung 3: Monatliches Haushaltsnettoeinkommen in Deutschland 2008 (Klassenbreite „500 Euro“)

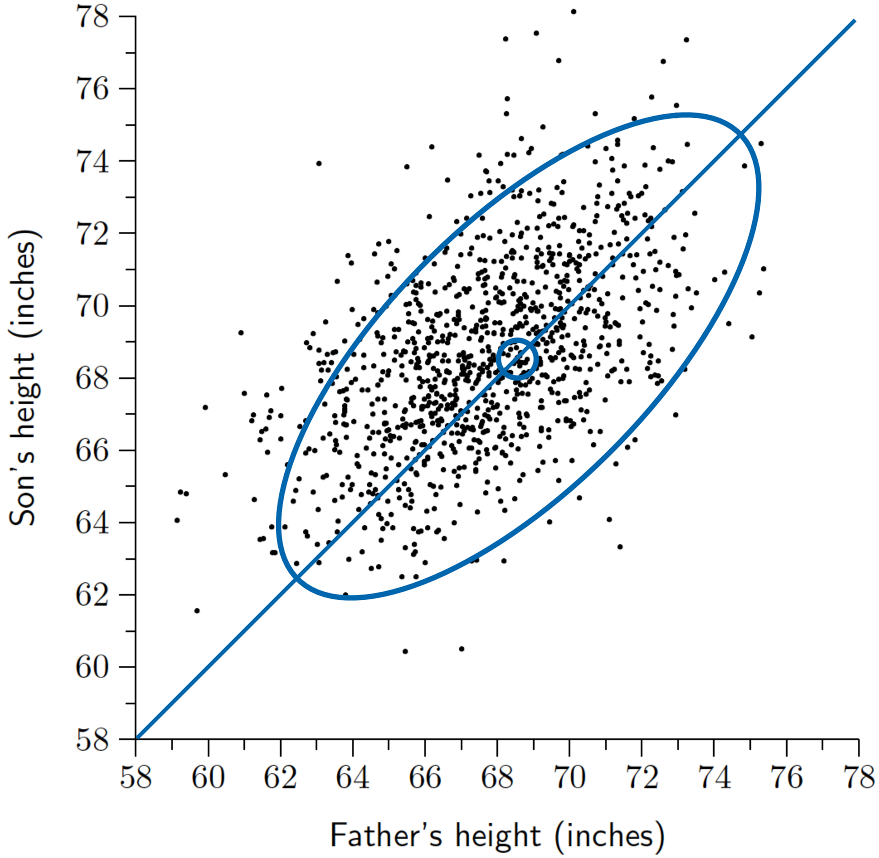


*) *Ab 6 000 Euro wurden aus Darstellungsgründen größere Klassenbreiten gewählt.*

Statistisches Bundesamt, Fachserie 15 Heft 6, EVS 2008

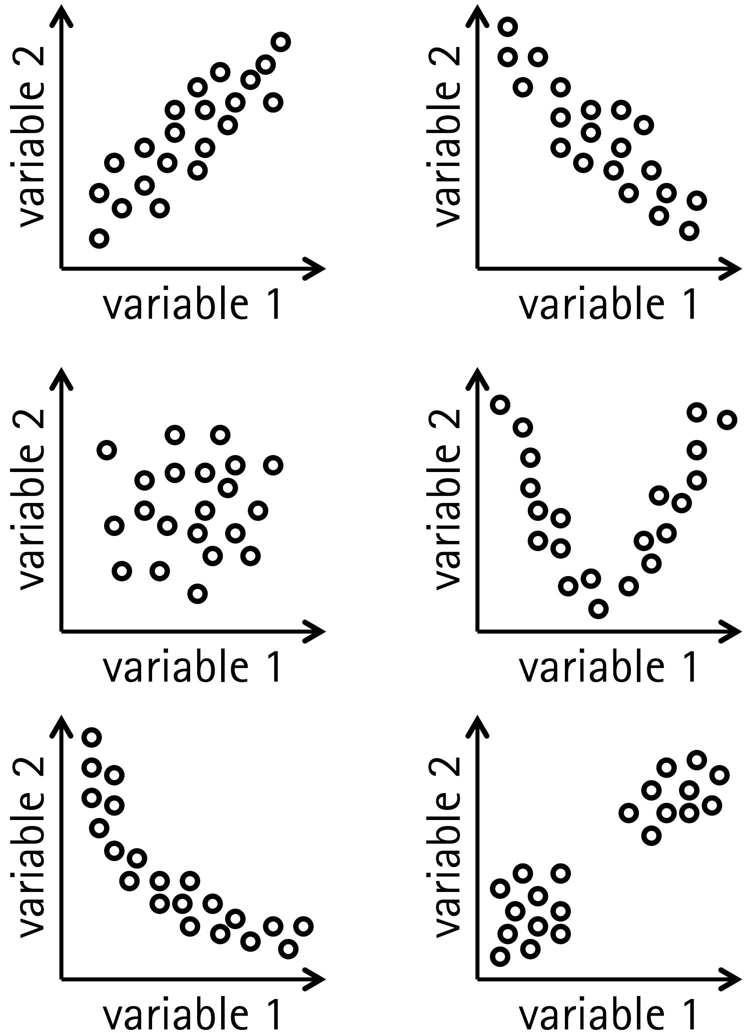
Scatter Plot

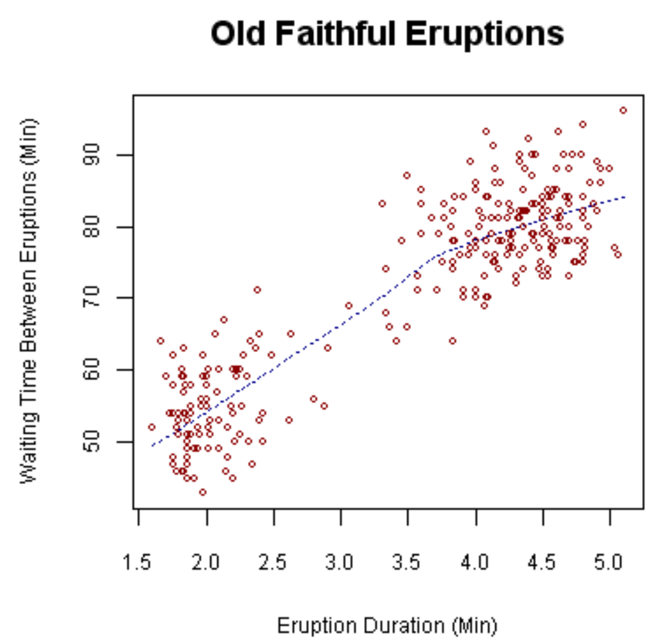
- Shows the relationship between two variables
  - Shape, center, density
  - Correlation
  - Line of best fit
- Example: heights of fathers and their (grown up) sons
  - 1078 data points
  - each point is a (father, son)-pair
- Show value of dependent variable in response to independent variable



Data: Karl Pearson (1857-1936)

Scatter Plot Shows Characteristics of Relationship

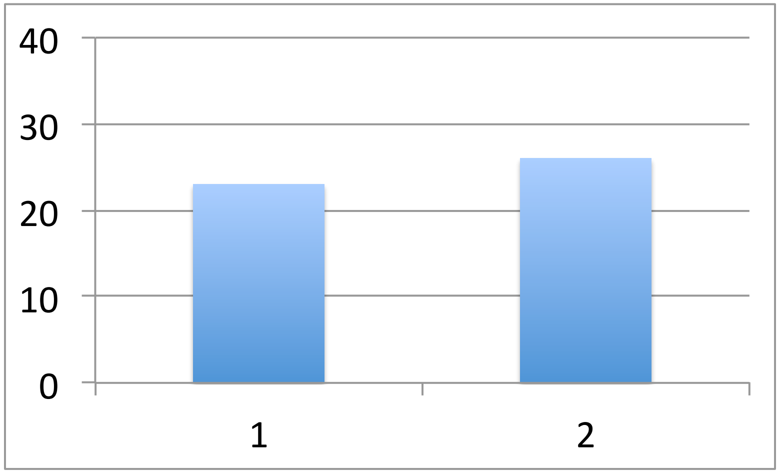


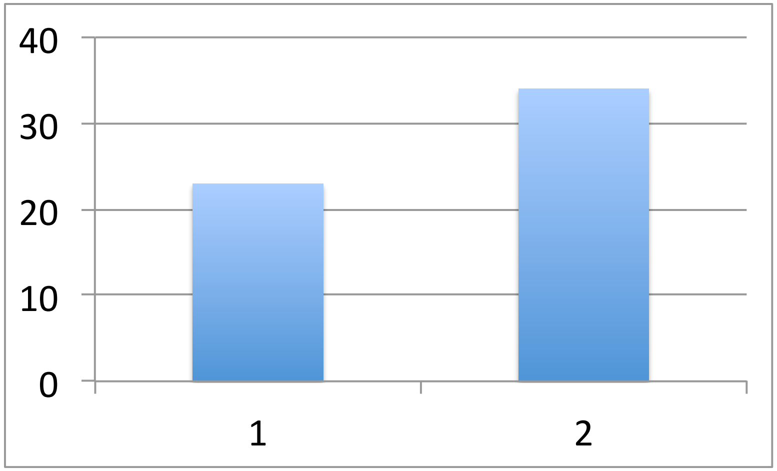


STATISTICAL TESTING

Comparing Means

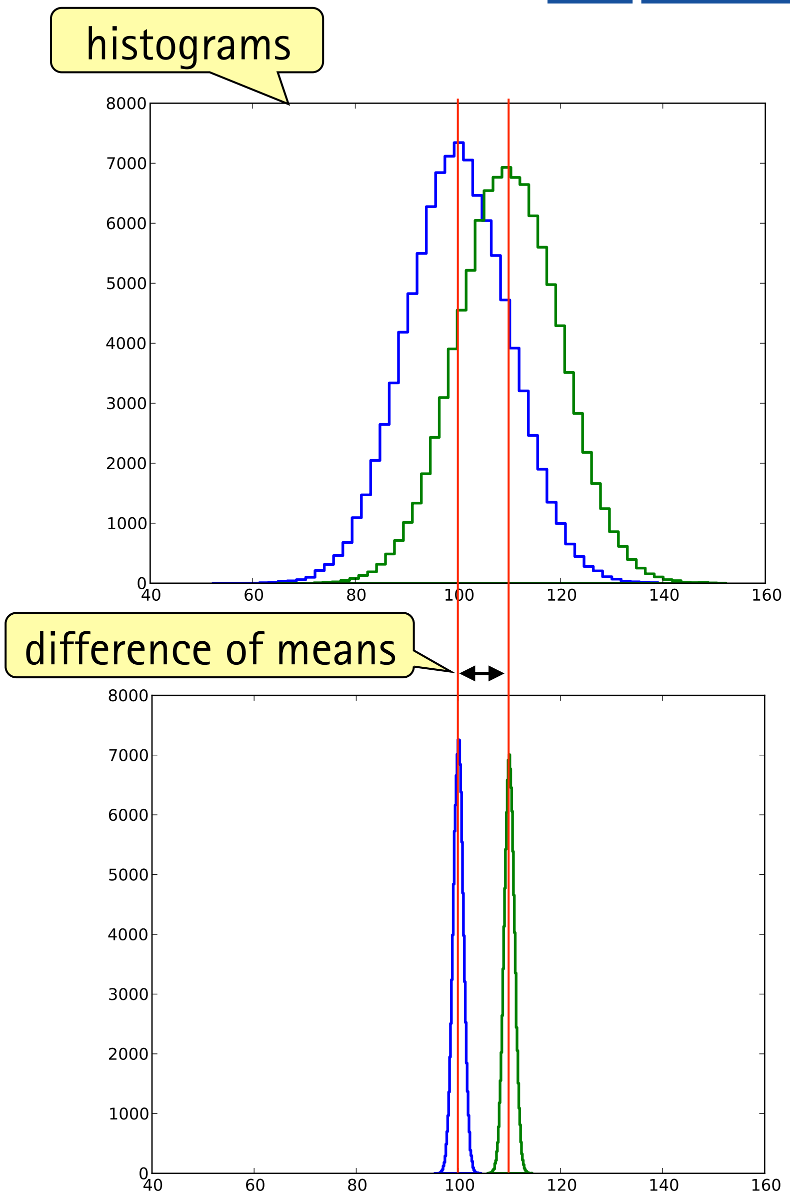
- Assume an experiment has two conditions
  - One factor (independent variable, IV) with two levels
- Repeated measurements are taken of a dependent variable (DV)
  - Multiple users
  - Repetitions of each condition (trials)
- Within one condition, compute averages across trials and users
  - Results in two mean values
- Conclusion?





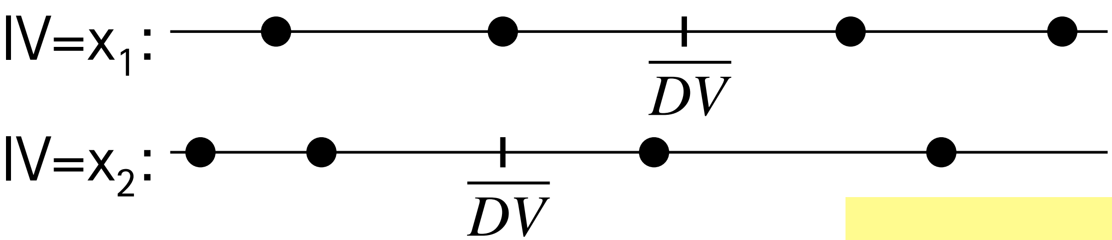
Difference of Means

- Look at distribution of values within a group
- Compare to difference of group means
- More confident that a systematic effect exists if
  - Difference between means is large
  - Deviation within group is small
- Statistical testing formalizes this
  - Provides probabilities of making these observations by chance rather than as a result of the experiment



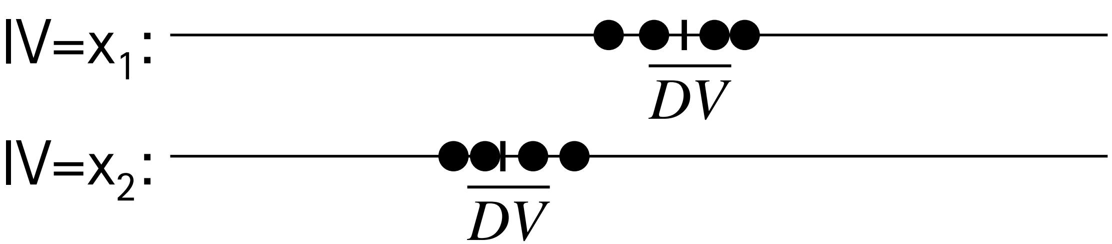
Between-Group and Within-Group Variability

- Single independent variable IV (with levels x_1 and x_2)
- Continuous dependent variable DV is measured
- Idea: Look at variability within groups and across means
  - More confident if small variability within group and large variability across means



situation 1: small confidence in systematic effect

$$\text{test-statistic} = \frac{\text{between-group variability}}{\text{within-group variability}}$$



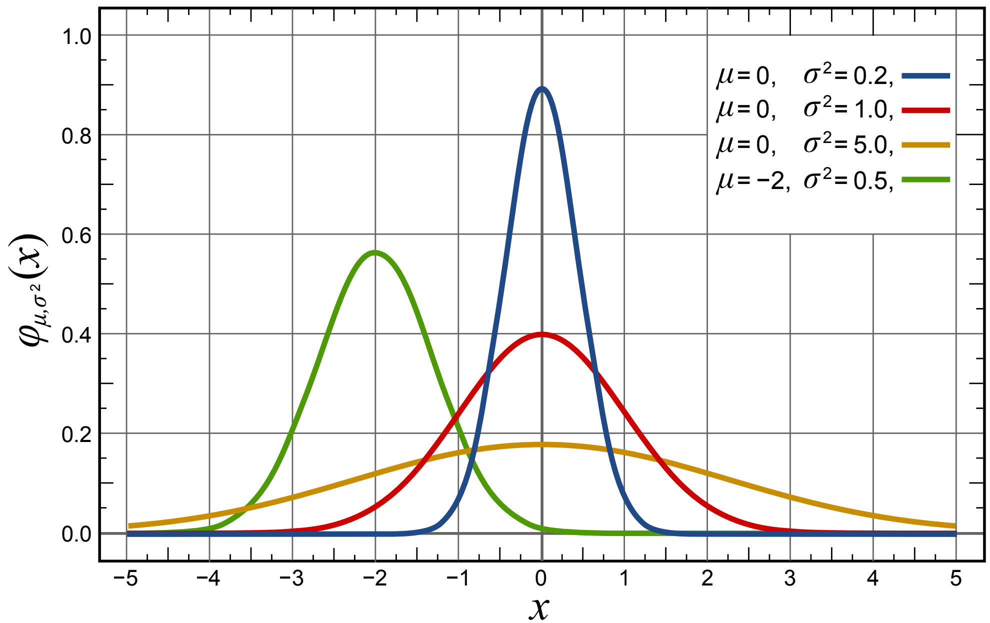
situation 2: large confidence in systematic effect

Normal Distribution

- View DV as a random variable
- Assumes DV is normally distributed

$$DV = N\left(\mu,\sigma^2\right)$$

  - μ = population mean
  - σ = population standard deviation
- An experiment samples this random variable



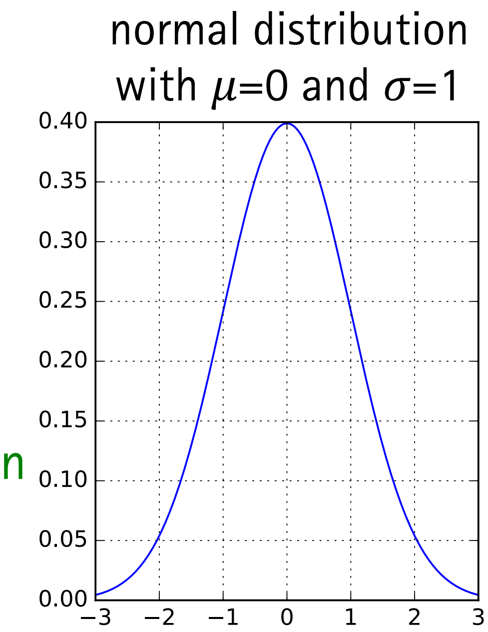
Normal distribution:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

iPython Notebook: Experiment Simulations

```
# NumPy fundamental numerical library, efficient array and matrix operations
import numpy as np

# matplotlib plotting library for creating graphs and figures
import matplotlib.pyplot as pp

# statistics functions
import scipy.stats as st

# Show graphs inline, as part of the iPython notebook
%matplotlib inline
```

iPython Notebook: Experiment Simulations




```
sigma = 1.0               # population standard deviation
mu1 = 0.0                 # population mean for condition 1
mu2 = 0.0                 # population mean for condition 2
sampleSize = 10           # number of samples in each condition
experiments = 100000      # number of simulated experiments

meanDiffs = np.zeros(experiments)   # result array for mean differences
...
```

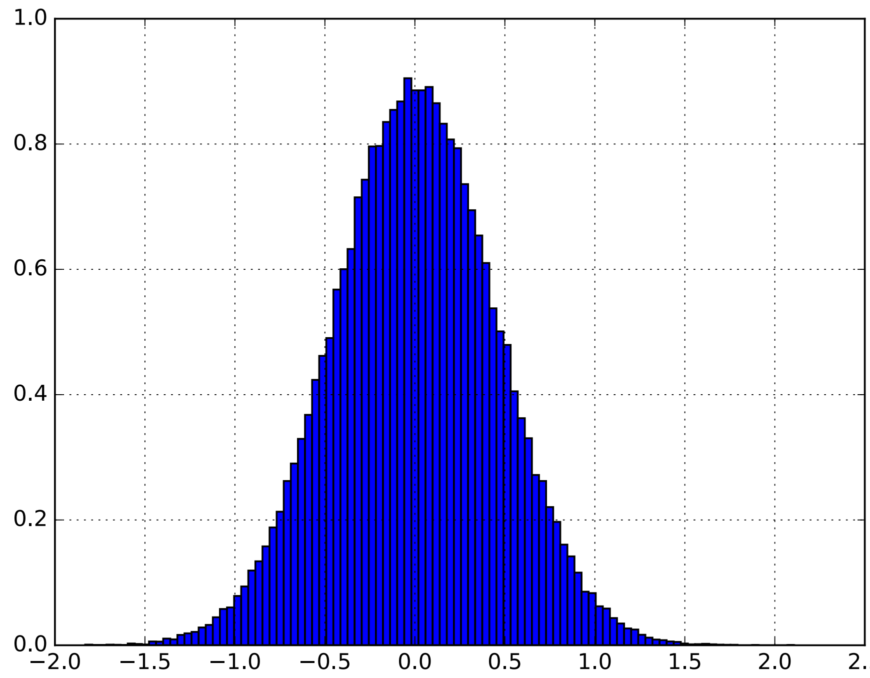
In this experiment: $\mu_1 = \mu_2$, i.e., difference between conditions, no effect of condition on dependent variable

iPython Notebook: Experiment Simulations

```
for i in xrange(experiments):          # loop over 100k experiment simulations
    # condition 1: take a sample of size sampleSize
    data1 = sigma * np.random.randn(sampleSize, 1) + mu1; # sample normal dist.

    # condition 2: take a sample of size sampleSize
    data2 = sigma * np.random.randn(sampleSize, 1) + mu2; # sample normal dist.

    mean1 = np.mean(data1)             # mean for sample 1
    mean2 = np.mean(data2)             # mean for sample 2

    meanDiffs[i] = mean1 - mean2       # difference between means
```

iPython Notebook: Experiment Simulations

```
# show distribution of differences between means
pp.figure() # a new figure window
# a normalized, cumulative histogram with 100 bins
pp.hist(meanDiffs, bins = 100, normed = True, cumulative = False)
pp.grid()
pp.show() # actually show the result
```



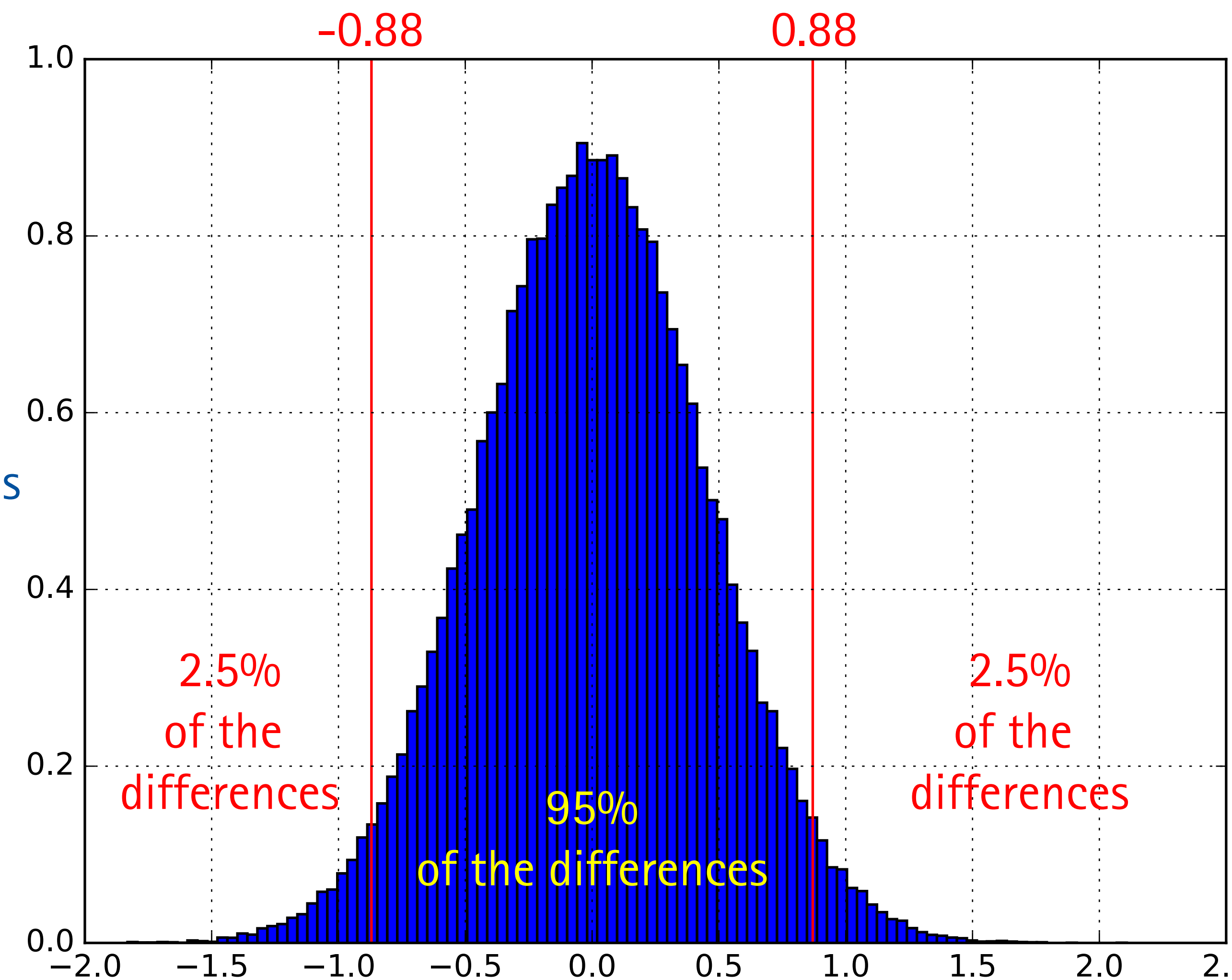
distribution of mean differences

iPython Notebook: Experiment Simulations

Mean differences less than -0.88 and greater than 0.88 are very unlikely

For population:

- Normal distributions
- $\mu_1 = \mu_2 = 0$
- $\sigma_1 = \sigma_1 = 1$



distribution of mean differences

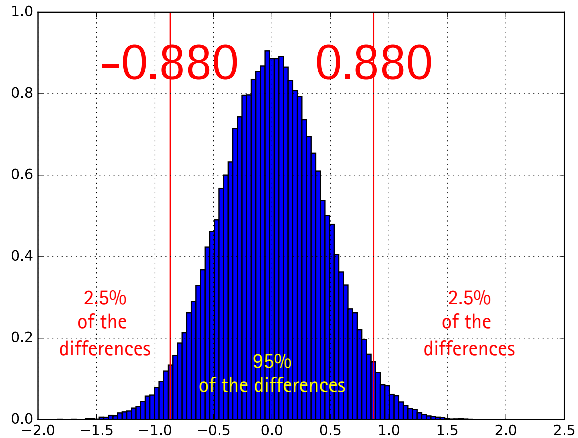
iPython Notebook: Experiment Simulations

```
meanDiffs.sort()                                    # sort the array of differences
i = int(experiments * 0.025)                        # 2.5% index
print meanDiffs[i], meanDiffs[experiments - i]      # 2.5% and 97.5% percentiles
```

- -0.880 0.879

```
# or simply use the percentile function
print np.percentile(meanDiffs, 2.5), np.percentile(meanDiffs, 97.5)
```

- -0.880 0.880



Problem: The percentiles depend on the population parameters, which are no known in a real experiment.

Null Hypothesis Tests

- Measurements of DV are random samples of populations
- Null hypothesis: all measurements are from the same population
  - H_0: $\mu_1 = \mu_2$ (population means are equal)
- Alternative hypothesis: not all means are equal
  - Many possibilities, difficult to analyze → focus on H_0
- Obtained data → test-statistic → probability of obtaining this data (and test-statistic) under the null hypothesis
  - If test-statistic indicates low probability (typically $\alpha = 5\%$) of obtaining this data under the null hypothesis,
  - then reject the null hypothesis,
  - i.e., accept the alternative hypothesis

iPython Notebook: Experiment Simulations

- Result
  - If DV = N(μ=0, σ=1)
  - and for samples of size 10
  - and if there is no difference between conditions (null hypothesis holds)
  - then 95% of the obtained values will be between -0.880 and 0.880
- Result depends on σ (which is not known in a real experiment)
  - need to get rid of (or normalize for) σ
  - → use the t-statistic

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n} + \frac{s_2^2}{n}}}$$

t-Statistic

- t-statistic normalizes for standard deviation of the difference of the sample means

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\underbrace{\sqrt{\frac{s_1^2}{n} + \frac{s_2^2}{n}}}_{\text{std. of mean diff.}}}$$

- X_1 is a sample of n values: sample mean $\bar{X}_1$, sample variance s_1^2
- X_2 is a sample of n values: sample mean $\bar{X}_2$, sample variance s_2^2
- Variance of the means is s_1^2/n and s_2^2/n, respectively
  - Repeatedly take samples of size n and compute means
  - The means of these samples will have variance s_i^2/n
- Variance of the difference of means is $s_1^2/n + s_2^2/n$
- The standard deviation is the square root of the variance

Independent Two-Sample t-Test

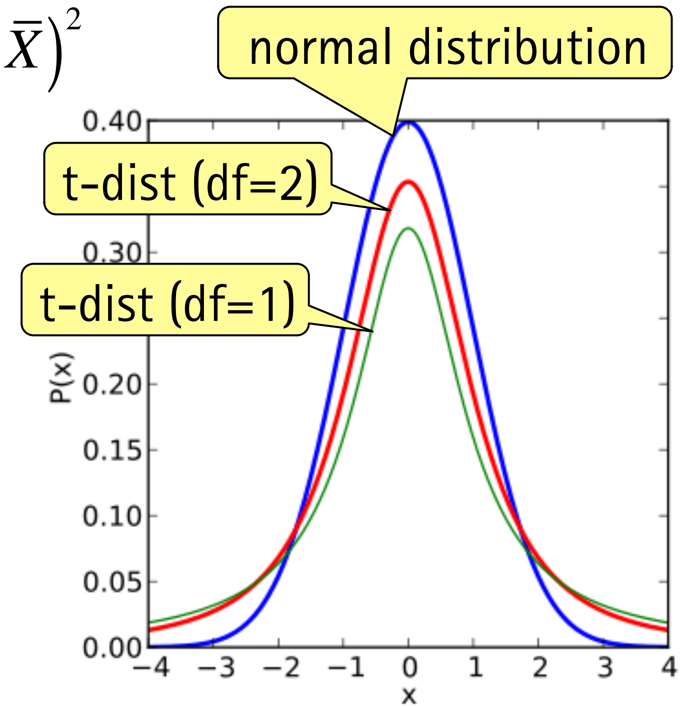
- Assumptions
  - Normally distributed population (samples are taken from this distribution)
  - Independent samples (different users)
  - Equal sample sizes (n users in each group)
- t-statistic

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n} + \frac{s_2^2}{n}}}$$

- Under these assumptions the t-statistic is t(df)-distributed
  - Only depends on n (known sample size)
  - Degrees of freedom: df = 2n - 2
  - Does not depend on µ or σ (unknown population parameters)

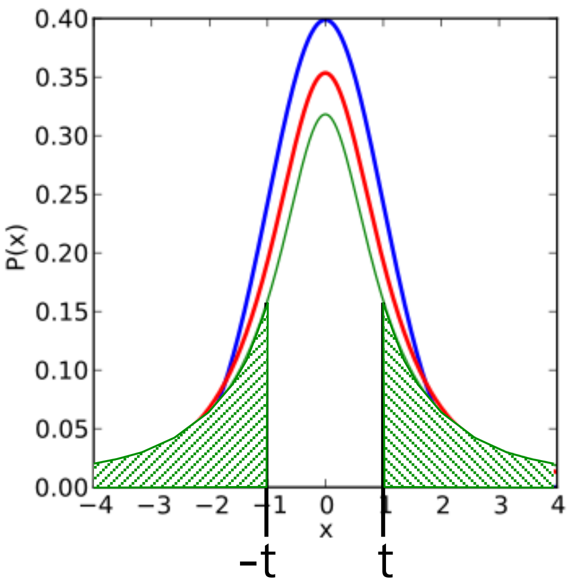
t(df)-Distribution

- $DV = N(\mu,\sigma^2)$ normal distribution
- let X = a sample of size n from DV
- let $\overline{X}$ = sample mean
- let s^2 = sample variance $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{X})^2$
- then $\frac{\bar{X}-\mu}{\sqrt{\frac{\sigma^2}{n}}} = N(0,1)$
- and $\frac{\bar{X}-\mu}{\sqrt{\frac{s^2}{n}}} = t(n-1)$ i.e., t-distributed with n - 1 degrees of freedom
- the larger n, the closer t(n-1) to N(0,1)



Independent Two-Sample t-Test

- Look for 2.5%- and 97.5%-percentiles of the t(df)-distribution
- Is t_{exp}-value from experiment above or below these percentiles?
- If so, then null hypothesis is unlikely



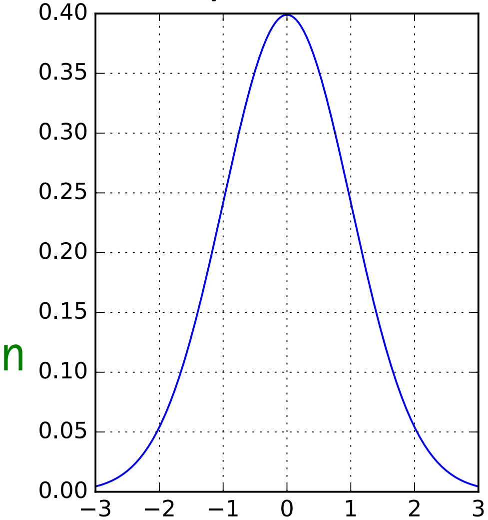
- Probability of obtaining a certain t-value if no effect
  - If no effect, then t-statistic follows t_{df}-distribution
  - Degrees of freedom: df = 2n - 2
  - $p(t_{df}) = p(t_{df} < -t_{exp}) + p(t_{df} > t_{exp})$ (two-tailed test)
  - If $p(t_{df}) < 0.05$, then we reject the null hypothesis

iPython Notebook: t-Test Simulation

```
# NumPy fundamental numerical library, efficient array and matrix operations
import numpy as np

# matplotlib plotting library for creating graphs and figures
import matplotlib.pyplot as pp

# statistics functions
import scipy.stats as st

# Show graphs inline, as part of the iPython notebook
%matplotlib inline
```

iPython Notebook: t-Test Simulation

normal distribution
with μ=1 and σ=1




```
sigma = 1.0               # population standard deviation
mu1 = 0.0                 # population mean for condition 1
mu2 = 0.0                 # population mean for condition 2
sampleSize = 10           # number of samples in each condition
experiments = 100000 # number of simulated experiments

meanDiffs = np.zeros(experiments)   # result array for mean differences
ts = np.zeros(experiments)          # result array for t-values
...
```

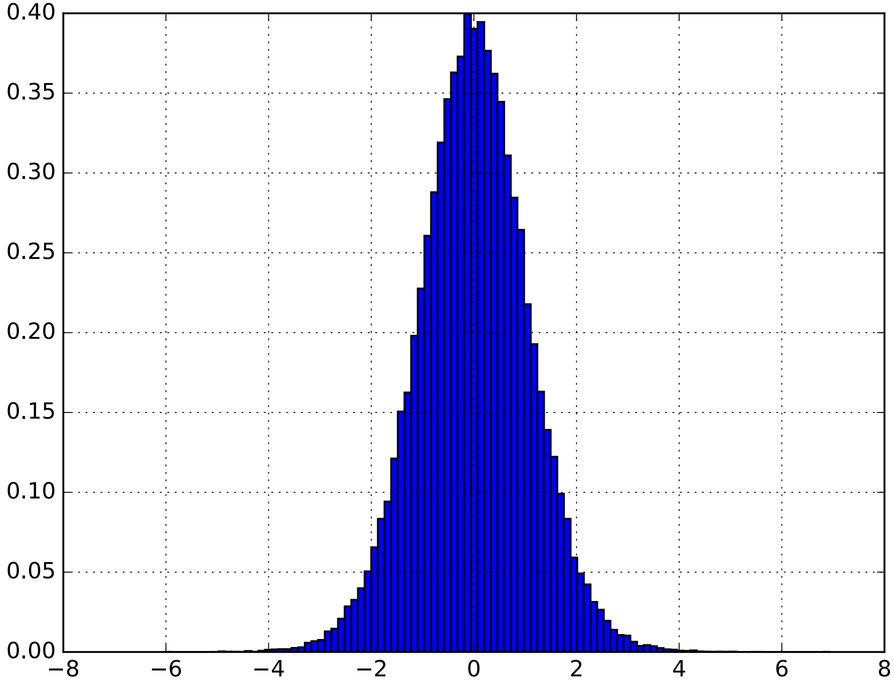
iPython Notebook: t-Test Simulation

```
for i in xrange(experiments):
    # condition 1: take a sample of size sampleSize
    data1 = sigma * np.random.randn(sampleSize, 1) + mu1; # sample normal dist.
    # condition 2: take a sample of size sampleSize
    data2 = sigma * np.random.randn(sampleSize, 1) + mu2; # sample normal dist.

    mean1 = np.mean(data1)          # mean for sample 1
    mean2 = np.mean(data2)          # mean for sample 2
    meanDiffs[i] = mean1 - mean2    # difference between means

    std1 = np.std(data1, ddof=1)    # standard deviation for sample 1
    std2 = np.std(data2, ddof=1)    # standard deviation for sample 2

    ts[i] = meanDiffs[i] / np.sqrt((std1**2 + std2**2) / sampleSize)
```

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n} + \frac{s_2^2}{n}}}$$

iPython Notebook: t-Test Simulation

```
# show distribution of t-values
pp.figure() # a new figure window
# a normalized, cumulative histogram with 100 bins
pp.hist(ts, bins = 100, normed = True, cumulative = False)
pp.grid()
pp.show() # actually show the result
```



distribution of t-values

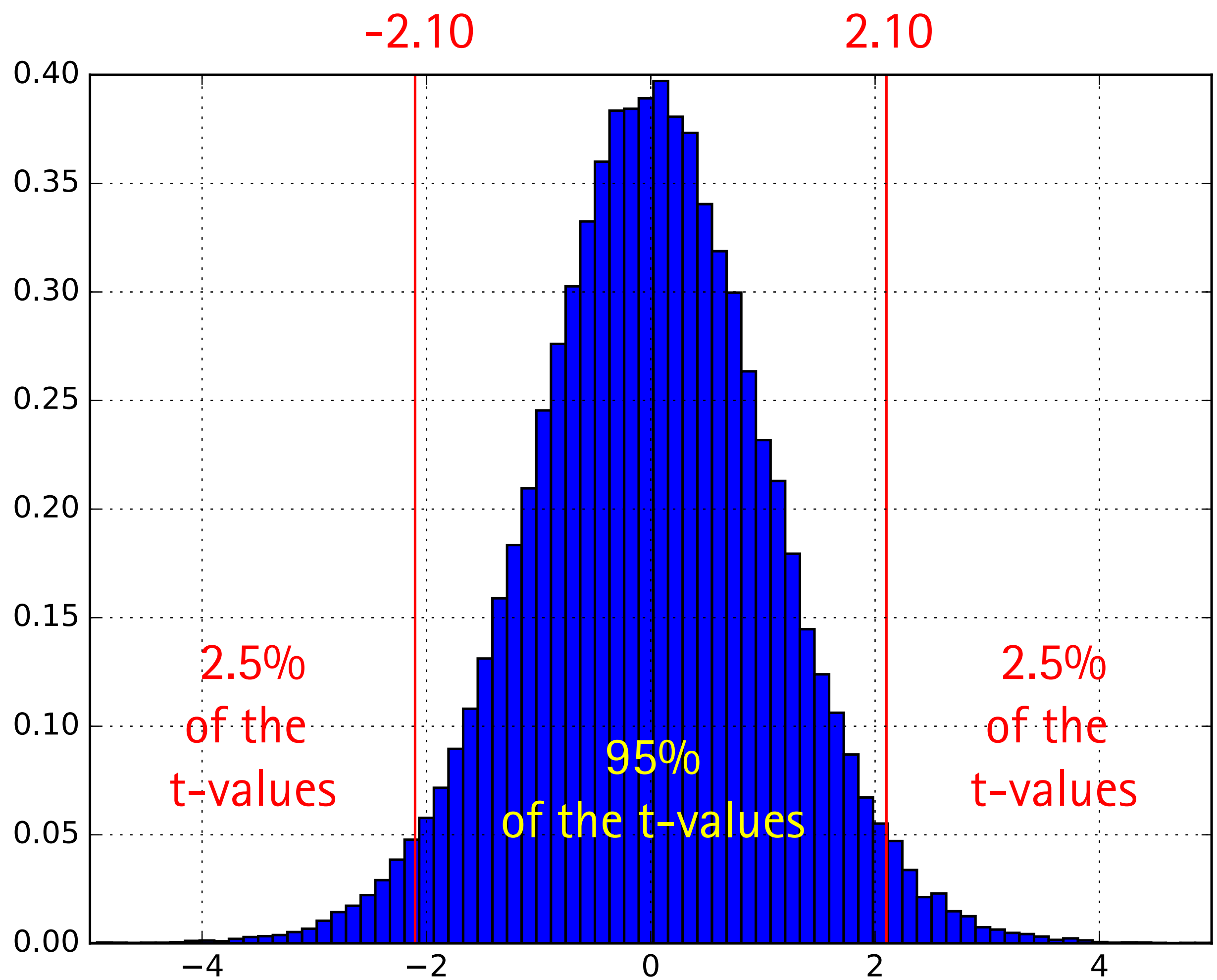
iPython Notebook: t-Test Simulation

t-values less than -2.10 and greater than 2.10 are very unlikely

For population:

- Normal distributions
- $\mu_1 = \mu_2 = 0$
- sample size = 10
- df = 2*10-2=18

distribution of t-values



iPython Notebook: t-Test Simulation

```
ts.sort()                                   # sort the array of t-values
i = int(experiments * 0.025)                # 2.5% index
print ts[i], ts[experiments - i]            # 2.5% and 97.5% percentiles
```

- -2.105 2.115

```
# or simply:
print np.percentile(ts, 2.5), np.percentile(ts, 97.5)
```

- -2.108 2.112

```
# or the theoretical value:
df = 2 * sampleSize - 2                     # degrees of freedom
st.t.ppf([0.025, 0.975], df, loc=0, scale=1)   # percent point function
```

- -2.101, 2.101

Choice of Statistical Test Method

- Statistical analysis
  - Often assumptions about underlying distribution
  - t-test: Compare two groups, normal distribution
  - Analysis of variance (ANOVA): Compare two or more groups, normal distribution
  - Regression analysis: How well does result fit to a model?
  - Rank orders: Wilcoxon- or Mann/Whitney test
  - Category counts: Chi-squared test
- Choice depends on
  - Number, continuity, and assumed distribution of dependent variables
  - Desired form of the result (yes/no, size of difference, confidence of estimate)

Overview of Common Significance Tests

Experiment Design	Independent Variables (IV)	Conditions for each IV	Type of Test
between-group	1	2	independent-samples t-test
	1	≥ 2	one-way ANOVA
	≥ 2	≥ 2	factorial ANOVA
within-group	1	2	paired-samples t-test
	1	≥ 2	repeated measured ANOVA
	≥ 2	≥ 2	repeated measures ANOVA
between-and-within-group	≥ 2	≥ 2	split-plot ANOVA

One-Way ANOVA

Group	Participants	Task completion time	Coding
Standard	Participant 1	245	0
Standard	Participant 2	236	0
Standard	Participant 3	321	0
Standard	Participant 4	212	0
Standard	Participant 5	267	0
Standard	Participant 6	334	0
Standard	Participant 7	287	0
Standard	Participant 8	259	0
Prediction	Participant 1	246	1
Prediction	Participant 2	213	1
Prediction	Participant 3	265	1
Prediction	Participant 4	189	1
Prediction	Participant 5	201	1
Prediction	Participant 6	197	1
Prediction	Participant 7	289	1
Prediction	Participant 8	224	1
Speech-based dictation	Participant 1	178	2
Speech-based dictation	Participant 2	289	2
Speech-based dictation	Participant 3	222	2
Speech-based dictation	Participant 4	189	2
Speech-based dictation	Participant 5	245	2
Speech-based dictation	Participant 6	311	2
Speech-based dictation	Participant 7	267	2
Speech-based dictation	Participant 8	197	2

Table 4.6 Sample data for one-way ANOVA test.

Lazar, Feng, Hochheiser: Research Methods in Human-Computer Interaction. Wiley 2010.

One-Way ANOVA

- For between-group designs: Different users in each group
  - k = 3 groups, n = 24 values, 24 users
  - degrees of freedom between groups: k - 1 = 2
  - degrees of freedom within groups: n-k = 21

Source	Sum of squares	df	Mean square	*F*	Significance
Between-group	7842.250	2	3921.125	2.174	0.139
Within-group	37880.375	21	1803.827		

Table 4.7 Result of the one-way ANOVA test.

- Reported as:

  "A one-way ANOVA test using task completion time as the dependent variable and group as the independent variable suggests that there is no significant difference among the three conditions ($F(2,21) = 2.174$, $p = 0.139$)."

Lazar, Feng, Hochheiser: Research Methods in Human-Computer Interaction. Wiley 2010.

One-Way Repeated Measures ANOVA

- For within-group designs: Same users for each group

	Standard	Prediction	Speech
Participant 1	245	246	178
Participant 2	236	213	289
Participant 3	321	265	222
Participant 4	212	189	189
Participant 5	267	201	245
Participant 6	334	197	311
Participant 7	287	289	267
Participant 8	259	224	197

Table 4.11 Sample data for one-way repeated measures ANOVA.

Lazar, Feng, Hochheiser: Research Methods in Human-Computer Interaction. Wiley 2010.

One-Way Repeated Measures ANOVA

- k = 3 groups, n = 24 values, u = 8 users
  - degrees of freedom between groups: k - 1 = 2
  - degrees of freedom within groups: (n-1) - (k-1) - (u-1) = 14

Source	Sum of square	Df	Mean square	*F*	Significance
Entry method	7842.25	2	3921.125	2.925	0.087
Error	18767.083	14	1340.506		

Table 4.12 Result of the one way repeated measures ANOVA test.

Lazar, Feng, Hochheiser: Research Methods in Human-Computer Interaction. Wiley 2010.

Chi-Square Test (for Nominal Data)

- Dependent variable is nominal / categorical
- Example: Preferred device: mouse or touch screen

Preferred device	
Mouse	Touch screen
14	6

  - For equal preference would expect 50% mouse, 50% touch screen
- Idea: measure deviation of observation from from expectation
- χ^2-statistic: $\chi^2 = \sum_{i=1}^{k} \frac{(observed_i - \mathrm{exp}ected_i)^2}{\mathrm{exp}ected_i}$
- follows χ^2-distribution if there is no systematic difference

Chi-Square Test (for Nominal Data)

		Preferred device	
		Mouse	Touch screen
	<65	14	6
Age	≥65	4	16

Table 4.25 A 2-by-2 frequency count table.

- $\chi^2(1) = 10.1, p < 0.005$

		Preferred device		
		Mouse	Touch screen	Stylus
	<18	4	9	7
	18–65	12	6	2
Age	≥65	4	15	1

Table 4.26 A 3 by 3 frequency count table.

- $\chi^2(4) = 16.8, p < 0.005$

Lazar, Feng, Hochheiser: Research Methods in Human-Computer Interaction. Wiley 2010.

Chi-Square Test (for Nominal Data)

		Preferred device				
		Mouse	Touch	Stylus	row sums	probabilities
Age	< 18	4	9	7	20	20/60=1/3
	18-65	12	6	2	20	20/60=1/3
	≥65	4	15	1	20	20/60=1/3
column sums		20	30	10	N=60	
probabilities		20/60=1/3	30/60=1/2	10/60=1/6		

Independence:
P(<18 ∧ Mouse) = P(<18) * P(Mouse)

- Expected values:

		Preferred device		
		Mouse	Touch	Stylus
Age	< 18	60*1/3*1/3=6.67	60*1/3*1/2=10	60*1/3*1/6=3.33
	18-65	60*1/3*1/3=6.67	60*1/3*1/2=10	60*1/3*1/6=3.33
	≥65	60*1/3*1/3=6.67	60*1/3*1/2=10	60*1/3*1/6=3.33

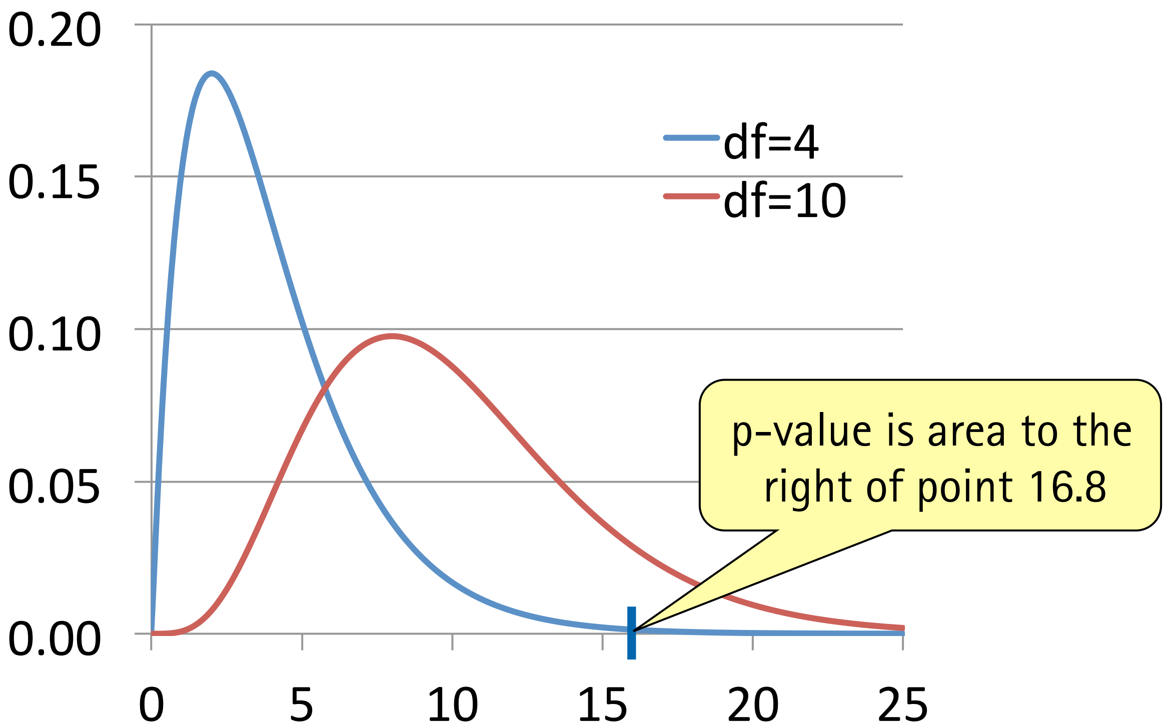
Chi-Square Test (for Nominal Data)

Preferred device

Age		Mouse	Touch	Stylus
	< 18	4	9	7
Age	18-65	12	6	2
	≥65	4	15	1

- χ^2-statistic: $\chi^2 = \sum_{i=1}^{k} \frac{(observed_i - \exp ected_i)^2}{\exp ected_i} = 16.8$
- Degrees of freedom = (rows-1) * (columns-1) = (3-1) * (3-1) = 4

Chi-Square Distribution



$p = P(\chi^2 \geq 16.8)$

- $\chi^2(4) = 16.8$, $p < 0.005$
- Degrees of freedom = (rows-1) * (columns-1) = (3-1) * (3-1) = 4

Summary

- Carefully log data
- Explore data graphically
  - Box plots
  - Histograms
  - Scatter plots
- Summarize data
  - Mean, median
  - Standard deviation
  - Range, min, max, quartiles
- Test whether the results are systematic
  - Statistical analysis (t-test, ANOVA, χ^2-test)