CHARITÉ
UNIVERSITÄTSMEDIZIN BERLIN

Tag 3 – Statistische Tests & Regression

Lukas Mödl, Matthias Becher, Erin Sprünken
biometrie-rkurs@charite.de

R-Kurs

Aktualisiert: 24. Juli 2023

Institut für Biometrie
und klinische Epidemiologie

STATISTISCHE TESTS IN R

- t-Test = `t.test()`
- Chi-Quadrat Test = `chisq.test()`
- Wilcoxon-Mann-Whitney-Test = `wilcox.test()`
- Fisher Test = `fisher.test()`
- McNemar's Test = `mcnemar.test()`
- Binomial Test = `binom.test()`
- ...

T-TEST

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

Parameter:

- x = Ein Vektor mit Daten
- y = Ein optionaler Vektor mit Daten, falls man zwei Gruppen vergleichen möchte
- alternative = c(two.sided", less", "greater")
- mu = Der angenommene Mittelwert unter der Nullhypothese
- paired = c(TRUE, FALSE)

BEISPIEL T-TEST:

```
> t.test(data$Age)

        One Sample t-test

data:  data$Age
t = 57.5, df = 130, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 51.13222 54.77617
sample estimates:
mean of x
  52.9542
```

BEISPIEL T-TEST:

```
> t.test(data[data$Klinik == 1, "Age"], data[data$Klinik == 2, "Age"])

        Welch Two Sample t-test

data:  data[data$Klinik == 1, "Age"] and data[data$Klinik == 2, "Age"]
t = 0.10025, df = 119.44, p-value = 0.9203
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.506035  3.879984
sample estimates:
mean of x mean of y
 53.04412  52.85714
```

Anmerkung: Per default nimmt R beim Zwei-Stichproben-t-Test ungleiche Varianz an

CHI-QUADRAT TEST:

`chisq.test()`

Beispiel:

```
> table(data[,c("Augenfarbe", "Haarfarbe")])
           Haarfarbe
Augenfarbe blond braun schwarz
     blau     15    15      24
     braun    13    13      11
     grün     11    16      14
```

```
> chisq.test(data$Augenfarbe, data$Haarfarbe)

        Pearson's Chi-squared test

data:  data$Augenfarbe and data$Haarfarbe
X-squared = 2.9076, df = 4, p-value = 0.5734
```

FORMELN IN R

Um eine Regression durchzuführen müssen wir der Funktion sagen, welche Spalten in unseren Daten die unabhängigen Variablen sind und welche Spalte die abhängige Variable ist. Dafür gibt es in R die Formelschreibweise:

- Nur bestimmte Variablen sollen in der Regression verwendet werden:

$$Y \sim X_1 + X_2 + X_3 + \ldots$$

- Alle Variablen im Datensatz sollen in der Regression verwendet werden:

$$Y \sim .$$

LINEARE REGRESSION

- `model <- lm(Weight~Age+Sex+Height+Klinik, data =data)`
- `summary(model)`
- Anmerkung: "0 +"am Anfang der Formel führt zu einer Regression ohne Intercept

```
Call:
lm(formula = Weight ~ Age + Sex + Height + Klinik, data = data)

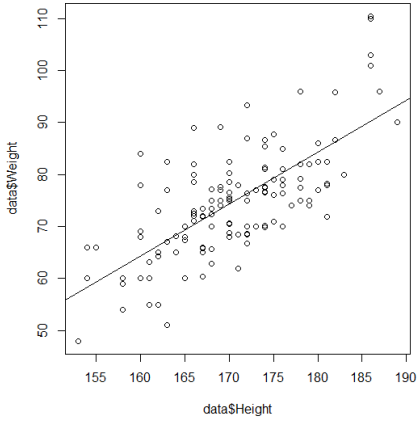
Residuals:
     Min       1Q   Median       3Q      Max
-16.2218  -5.6996  -0.2926   3.7819  20.1909

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -94.769974  19.631621  -4.827 3.93e-06 ***
Age           0.000201   0.065644   0.003    0.998
Sex           0.439927   1.744252   0.252    0.801
Height        1.001254   0.110381   9.071 1.98e-15 ***
Klinik       -0.832225   1.327066  -0.627    0.532
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.486 on 126 degrees of freedom
  (1 Beobachtung als fehlend gelöscht)
Multiple R-squared:  0.4994,    Adjusted R-squared:  0.4835
F-statistic: 31.42 on 4 and 126 DF,  p-value: < 2.2e-16
```

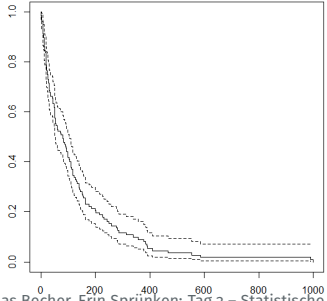
LINEARE REGRESSION PLOT

- `plot(data$Height,data$Weight)`
- `abline(model)`



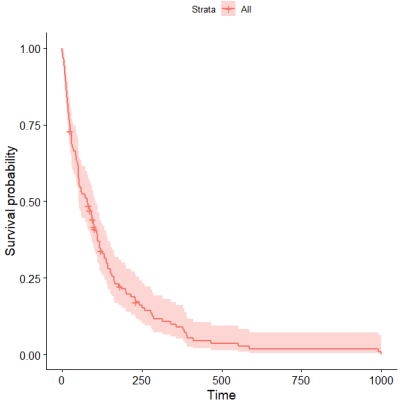
KAPLAN-MEIER PLOT

- ```
  library(survival)
  data_vet <- veteran
  km_fit <- survfit(Surv(time, status) ~ 1, data=data_vet)
  plot(km_fit)
  ```


  ```

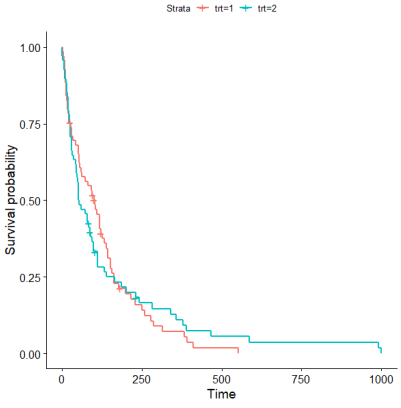
KAPLAN-MEIER PLOT

- `library(survminer)`
  `ggsurvplot(km_fit)`



KAPLAN-MEIER PLOT

- `km_fit <- survfit(Surv(time, status) ~ trt, data=data_vet)`
  `ggsurvplot(km_fit)`



LOGISTISCHE REGRESSION

- `model <- glm(y~., data = logistic_data, family = binomial)`
- `summary(model)`

```
Call:
lm(formula = Sex ~ Weight + Height + Augenfarbe, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.77828 -0.29252 -0.04797  0.28105  1.11699

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.9629560  0.9017111   7.722 2.98e-12 ***
Weight        -0.0001223  0.0046242  -0.026    0.979
Height        -0.0387331  0.0065085  -5.951 2.43e-08 ***
Augenfarbebraun  0.1370216  0.0838478   1.634    0.105
Augenfarbegrün  -0.0021609  0.0810893  -0.027    0.979
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.391 on 127 degrees of freedom
Multiple R-squared:  0.3796,    Adjusted R-squared:   0.36
F-statistic: 19.43 on 4 and 127 DF,  p-value: 1.719e-12
```

ONE-WAY ANOVA

- `model <- aov(formula, data)`

```
> one_way <- aov(Height~Augenfarbe, data = data)
> summary(one_way)
             Df Sum Sq Mean Sq F value Pr(>F)
Augenfarbe    2     28   14.14   0.249   0.78
Residuals   129   7317   56.72
```

TWO-WAY ANOVA

```
> two_way <- aov(Height~Augenfarbe + Haarfarbe, data = data)
> summary(two_way)
             Df Sum Sq Mean Sq F value Pr(>F)
Augenfarbe    2     28   14.14   0.246  0.782
Haarfarbe     2     10    5.00   0.087  0.917
Residuals   127   7307   57.53
```

INTERACTION ANOVA

```
> interaction_model <- aov(Height~Augenfarbe*Haarfarbe, data = data)
> summary(interaction_model)
                    Df Sum Sq Mean Sq F value Pr(>F)
Augenfarbe           2     28   14.14   0.239  0.788
Haarfarbe            2     10    5.00   0.085  0.919
Augenfarbe:Haarfarbe 4     40   10.06   0.170  0.953
Residuals          123   7266   59.08
```

INSTALLATION WEITERER R PAKETE

Jede R Umgebung installiert und lädt standardmäßig die Pakete `base`, `stats`, `datasets`, `methods` und `graphics`.

- Installation weiterer Pakete mit:

```
install.packages("name-des-pakets", dependencies = TRUE)
```

- Bei jedem Start von R muss das Paket, wenn es verwendet werden soll, geladen werden:

```
library("name-des-pakets")
```

- Aktualisieren der Pakete mit:

```
update.packages()
```

BEISPIEL: INSTALLATION UND LADEN DES R PAKETS `MASS`

```
> install.packages("MASS")
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.1/MASS_7.3-55.zip'
Content type 'application/zip' length 1192198 bytes (1.1 MB)
downloaded 1.1 MB

package 'MASS' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\[redacted]\AppData\Local\Temp\RtmpSMaYtV\downloaded_packages
> library("MASS")
```

EMPFEHLENSWERTE PAKETE

- `MatchIt` für Propensity Score Matching
- `MASS` für Negativ-binomiale Regression
- `lmer` bzw. `lme4` für Mixed-Models
- `pwr` für Power-Analyse und insbesondere zur Fallzahlplanung
- `ggplot2` für schöne Plots
- `haven` für das Einlesen von `.sav`-Dateien (SPSS)
- ...