FUNKTIONEN UND FORMELN IN 

Referent: Daniel Laskow

Betreuer: Eugen Betke

Programmierung in R

Arbeitsbereich Wissenschaftliches Rechnen

Universität Hamburg

01.06.2016

Gliederung

- Arithmetische Operatoren
- Funktionen aus der Statistik
- Eigene Funktionen
- Plotten von Ergebnissen
- Statistische Modelle
- Zusammenfassung

Arithmetische Operatoren

Operator	Beschreibung	Beispiel	Ausgabe
+ - * /	Addition, Subtraktion usw.	10/3	[1] 3.333333
** oder ^	Potenzieren	2^3	[1] 8
%/%	Ganzzahlige Division	10%/%4	[1] 2
%%	Modulo	10%%3	[1] 1

- Die Operatoren funktionieren auch mit Vektoren, Matrizen, Arrays und Data Frames

```
>vektor=c(1,2,3,4)
>vektor^2
[1]  1  4  9 16
>m=matrix(vektor, nrow=2, byrow=TRUE)
>m*5
     [,1] [,2]
[1,]    5   10
[2,]   15   20
```

```
> y = matrix(c(1,2,3,4), nrow=2)
> y*y
     [,1] [,2]
[1,]    1    9
[2,]    4   16
> y^y
     [,1] [,2]
[1,]    1   27
[2,]    4  256
```

Funktionen aus der Statistik

`mean(x)`	Arithmetisches Mittel	$\frac{1}{n}\sum_{i=1}^{n} x_i = \frac{x_1+\ldots+x_n}{n}$
`median(x)`	Median	$x_1, x_2, \underline{x_3}, x_4, x_5$
`var(x)`	Varianz	$\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$
`sd(x)`	Standardabweichung	$\sqrt{var(x)}$
`cumsum(x)`	Kumulierte Summe	$\sum_{x \in A} x$
`cumprod(x)`	Kumuliertes Produkt	$\prod_{x \in A} x$

Grundlegendes

- Funktionen sind vom Datentyp „function“, Syntax ist wie folgt

```
myfunction <- function(arg1, arg2, ... ){      [1]

   statements

   return(object)

}
```

- Parameter müssen nicht übergeben werden bspw. ist der folgende Aufruf gültig

```
>myfunction2 <- function(a, b){
   return(1)
 }
>myfunction2()
[0] 1
```

- Parameter mit Anfangsbelegungen sind auch möglich

```
myfunction3 <- function(arg1, arg2=5, arg3=2 ){...}
```

Beispiel 1

```
fakultaet <- function(n) {
   if( n == 0 )
      1
   else
       n * fakultaet( n - 1 )
}
```

```
> fakultaet(5)
[1] 120
>y <- fakultaet(5)
>y
[1] 120
```

Beispiel 2.1

```
wahrscheinlichkeit <- function(k,n,p) {
    (fakultaet(n)/(fakultaet(k)*fakultaet(n-k))) * p^(k) * (1-p)^(n-k)
}
```

- Bekannt als Wahrscheinlichkeitsfunktion einer Binomialverteilung

$$B(n,p,k) = \binom{n}{k} p^k (1-p)^{n-k}$$

- Ähnliche Funktion in R: `dbinom(k,n,p)`
- Unterschied: k kann auch ein Vektor, Matrix usw. sein

```
> dbinom(seq(1:3),13,0.4)
[1] 0.01131927   0.04527707   0.11067729
```

Beispiel 2.2

```
wahrscheinlichkeit <- function(k,n,p) {
        x =c()
        for(i in k)
        {
            x = append(x,(fakultaet(n)/(fakultaet(i)*fakultaet(n-i))) * p^(i)
            * (1-p)^(n-i))
         }
        return(x)
}
```

```
>u = c(1,2,5)
>wahrscheinlichkeit(u,10,0.3)
[1] 0.1210608   0.2334744   0.1029193
> wahrscheinlichkeit(5,10,0.3)
[1] 0.1029193
```

Beispiel 3

```
cumWahrscheinlichkeit <- function(k,n,p){
    a=0
    for (i in 0:k){
       a = a + wahrscheinlichkeit(i,n,p)
    }
     return(a)
}
```

- Bekannt als kumulierte Wahrscheinlichkeitsfunktion einer Binomialverteilung

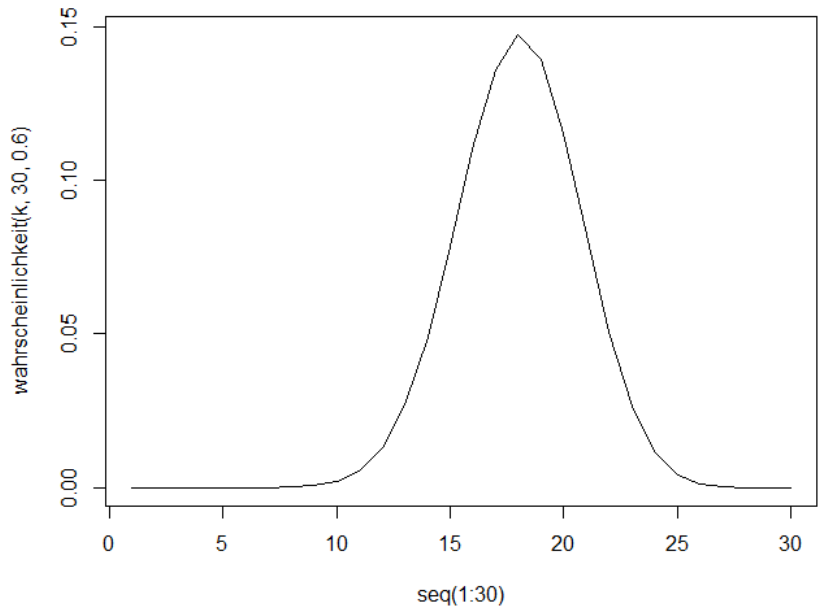
$$F(n,p,k) = \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i}$$

- Ähnliche Funktion in R: `pbinom(k,n,p)`

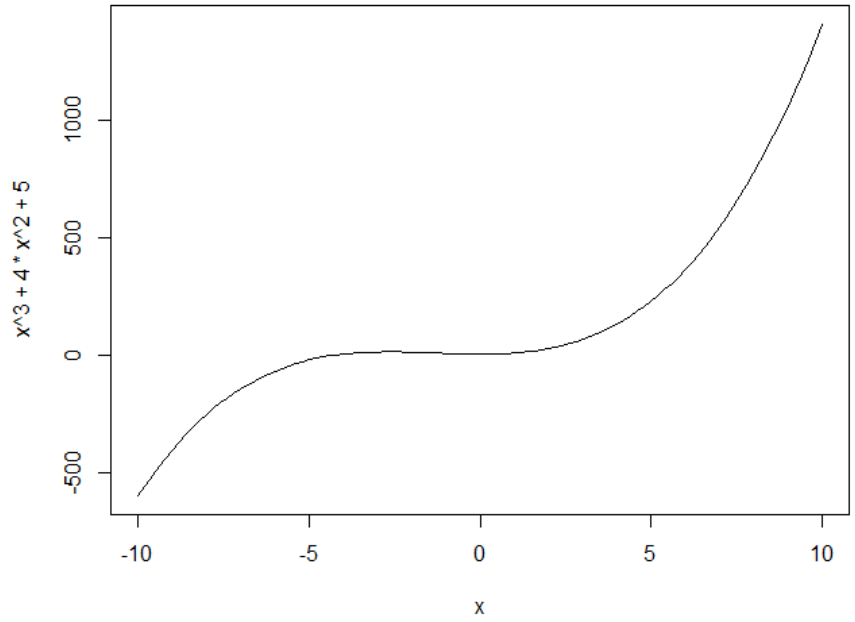
Plotten von Ergebnissen

```
>k= seq(1:30)

>plot(k, wahrscheinlichkeit(k, 30, 0.3), type = "l")
```




```
>curve(x^3+4*x^2+5, from=-10, to=10)
```



Formeln

- Verwendung in statistischen Modellen
- Formelnotation

Operatoren	Beispiel	Beschreibung
`+`	`y ~ x + z`	Einfluss einer Variable
`-`	`y ~ . -x, data=d`	Variable aus der Formel löschen
`:`	`y ~ x:z`	Interaktion zweier Variablen hinzufügen
`*`	`y ~ x*z`	Direkter Einfluss und Interaktion zweier Variablen
`^`	`y ~ (x+z+w)^3`	Direkter Einfluss und alle möglichen Interaktionen
`-1` oder `+0`	`y ~ x + z - 1`	Kein Interzept

  - Erlaubt sind auch mathematische Operationen z.B. `y ~ log(x) + I(z+5)`

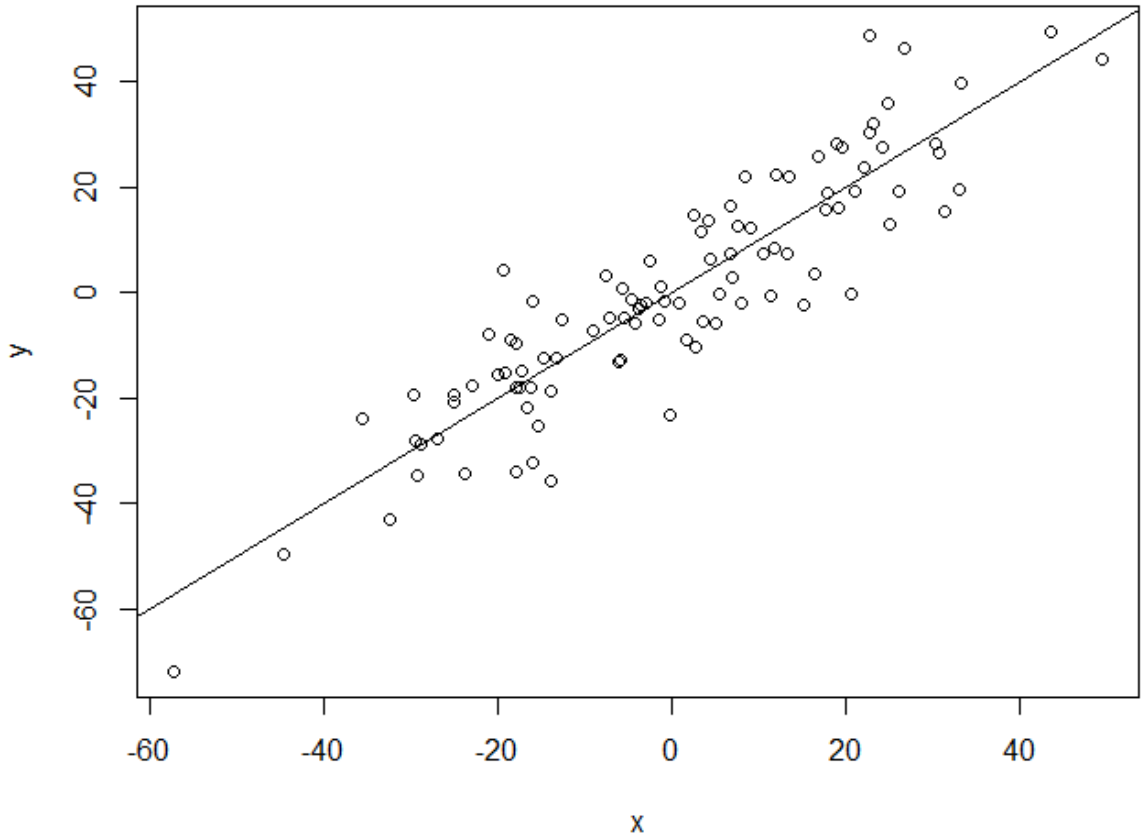
Nutzen davon ist

- Erkennung von Variableneinfluss
- Vorhersagen zu machen

Lineares Modell Beispiel 1.1

```
>x <- rnorm(100, sd = 20)
>y <- x + rnorm(100, sd = 9)
>plot(y ~ x)
>model <- lm(y ~ x)
>abline(model)
```

- x und y sind hierbei jeweils 100 zufällige Werte



Lineares Modell Beispiel 1.2

Zusammenfassung eines linearen Modells

```
>summary(lm(y ~ x))

Residuals:
    Min      1Q  Median      3Q     Max
-24.3014  -6.5649   0.8644   6.0194  27.8797

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.05374    0.97350   0.055    0.956
x            0.96420    0.05117  18.843   <2e-16 ***
...
Residual standard error: 9.718 on 98 degrees of freedom
Multiple R-squared:  0.7837,      Adjusted R-squared:  0.7815
F-statistic: 355.1 on 1 and 98 DF,  p-value: < 2.2e-16
```

Vorhersage eines Werts

```
>wert = data.frame(x=120)
>predict(model, wert)
       1
115.7574
```

Trendlinie folgt also der Funktion $y = 0,9642x + 0,05374$

Lineares Modell Beispiel 2

```
>a = read.csv(file.choose(), header=TRUE, sep=",")
>model = lm(a$cons ~ a$temp)
>plot(a$cons ~ a$temp)
>abline(model)
>summary(model)
```

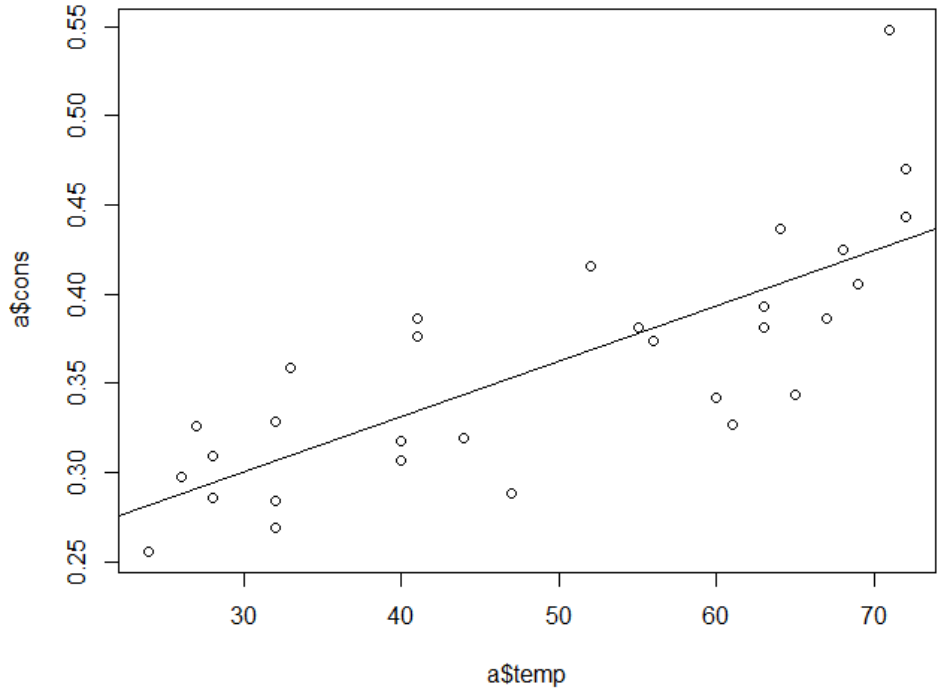
…

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	0.2068621	0.0247002	8.375	4.13e-09	***
a$temp	0.0031074	0.0004779	6.502	4.79e-07	***

…

Residual standard error: 0.04226 on 28 degrees of freedom
Multiple R-squared: 0.6016, Adjusted R-squared: 0.5874
F-statistic: 42.28 on 1 and 28 DF, p-value: 4.789e-07



Trendlinie folgt also der Funktion $a\$cons = 0.0031074 \cdot a\$temp + 0,2068621$

Lineares Modell Beispiel 3

```
>a = read.csv(file.choose(), header=TRUE, sep=",")
>model = lm(a$cons ~ a$temp + a$price + a$income)
>#plot(a$cons~a$temp+a$price+a$income)
>summary(model)
```

…

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	0.1973151	0.2702162	0.730	0.47179	
a$temp	0.0034584	0.0004455	7.762	3.1e-08	***
a$price	-1.0444140	0.8343573	-1.252	0.22180	
a$income	0.0033078	0.0011714	2.824	0.00899	**

…

Residual standard error: 0.03683 on 26 degrees of freedom

Multiple R-squared: 0.719, Adjusted R-squared: 0.6866

F-statistic: 22.17 on 3 and 26 DF, p-value: 2.451e-07

Vergleich von statistischen Modellen

```
> model1<-lm(a$cons ~ a$temp + a$price)
> model2<-lm(a$cons ~ a$temp)
> anova(model1, model2)

Analysis of Variance Table

Model 1: a$cons ~ a$temp + a$price
Model 2: a$cons ~ a$temp
  Res.Df     RSS  Df   Sum of Sq      F  Pr(>F)
1     27 0.046090
2     28 0.050009 -1 -0.0039194 2.296 0.1413
```

```
> model3<-lm(a$cons ~ a$temp)
> model4<-lm(a$cons ~ a$temp + a$income)
> anova(model4, model3)

Analysis of Variance Table

Model 1: a$cons ~ a$temp + a$income
Model 2: a$cons ~ a$temp
 Res.Df    RSS  Df  Sum of Sq      F    Pr(>F)
1   27 0.037399
2   28 0.050009 -1 -0.012611 9.1044 0.005506
```

- Der P-Wert gibt die Signifikanz des Unterschiedes an, je kleiner er ist, desto höher ist die Signifikanz des Unterschiedes
- Im ersten Beispiel ist der Unterschied nicht signifikant, da der P-Wert mit 0.1413 > 0.05 ist
- Im zweiten Beispiel ist der Unterschied signifikant, da der P-Wert mit 0.005506 < 0.05 ist

Zusammenfassung

- Mathematische Operationen sind flexibel
- Viele wichtige Funktionen aus der Statistik sind bereits in R implementiert
- Funktionen sind vom Datentyp „function“
- Der zuletzt berechnete Ausdruck einer Funktion wird zurückgegeben oder explizit mit `return(x)`
- Die Parameter einer Funktion haben keinen vorbestimmten Typ
- Mit Formeln kann man statistische Modelle beschreiben
- Mit `lm(`*`Formel`*`)` macht man Lineare Modelle, mit `summary(`*`model`*`)` macht man Zusammenfassungen, mit `abline(`*`model`*`)` macht man Trendlinien
- Vergleiche von Modellen macht man mit `anova(`*`model1`*`,`*`model2`*`)`, der P-Wert gibt die Signifikanz des Unterschiedes an

Literatur

- [1] Robert I. Kabacoff, "User-written Functions", http://www.statmethods.net/management/userfunctions.html, 23.05.2016
- Richard Hahn, „Statistical Formula Notation in R“ , http://faculty.chicagobooth.edu/richard.hahn/teaching/formulanotation.pdf, 18.05.2016
- „Using R for Linear Regression“, http://www.montefiore.ulg.ac.be/~kvansteen/GBIO0009-1/ac20092010/Class8/Using%20R%20for%20linear%20regression.pdf, 18.05.2016
- Jim Frost, „Regression Analysis“, http://blog.minitab.com/blog/adventures-in-statistics/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit, 18.05.2016
- Vincent Arel-Bundock, „Datasets“, https://vincentarelbundock.github.io/Rdatasets/datasets.html, 18.05.2016
- „Formula“, https://stat.ethz.ch/R-manual/R-devel/library/stats/html/formula.html, 18.05.2016
- „Modeling in R“, http://www.springer.com/cda/content/document/cda_downloaddocument/9781461444749-c1.pdf?SGWID=0-0-45-1344211-p174513603, 18.05.2016
- Steven Buechler, „Statistical Models in R“, http://www3.nd.edu/~steve/Rcourse/Lecture8v1.pdf, 18.05.2016
- „Lineares Modell“, https://de.wikipedia.org/wiki/Lineares_Modell, 18.05.2016