

INTRO DATENANALYSE MIT R - DRITTER TEIL

Jan-Philipp Kolb

14 Mai, 2019

LATTICE GRAPHIKEN

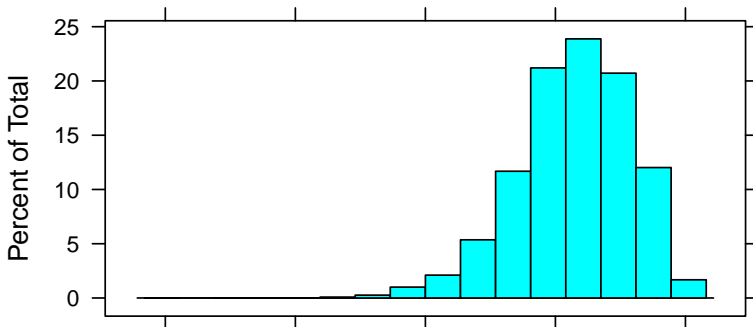
DAS LATTICE-PAKET

It is designed to meet most typical graphics needs with minimal tuning, but can also be easily extended to handle most nonstandard requirements.

<http://stat.ethz.ch/R-manual/R-devel/library/lattice/html/Lattice.html>

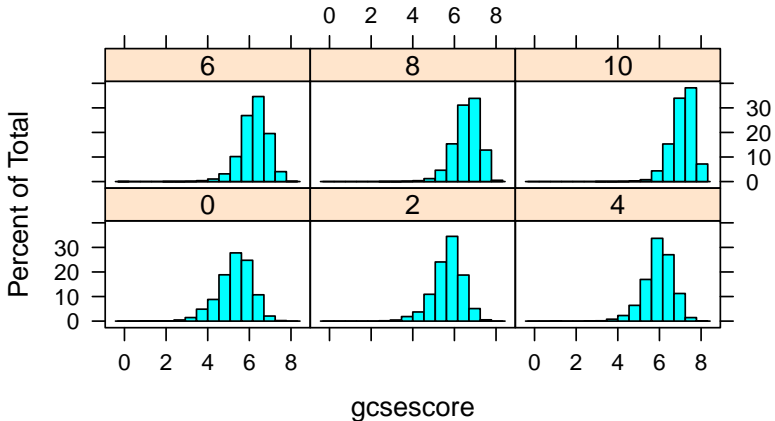
HISTOGRAMM MIT LATTICE

```
library("lattice")  
library("mlmRev")  
data(Chem97)  
histogram(~ gcsescore, data = Chem97)
```



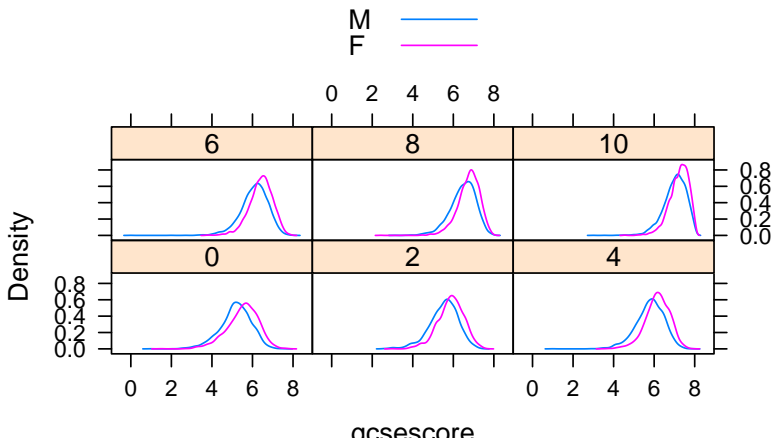
HISTOGRAMM MIT LATTICE

```
histogram(~ gcsescore | factor(score), data = Chem97)
```



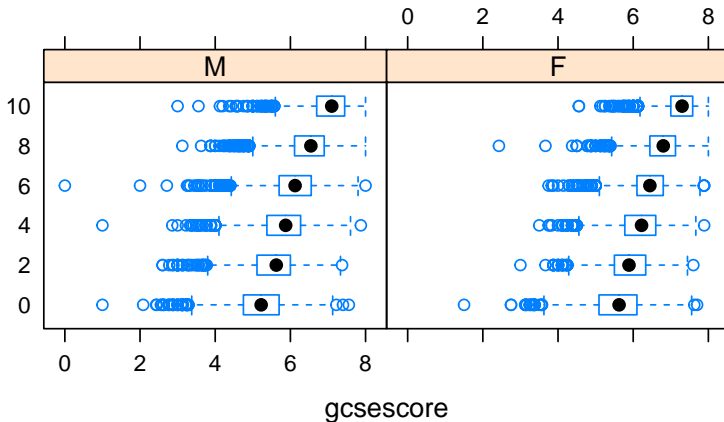
DIE DICHTE MIT LATTICE ZEICHNEN

```
densityplot(~ gcsescore | factor(score), Chem97,  
            groups=gender, plot.points=FALSE, auto.key=TRUE)
```



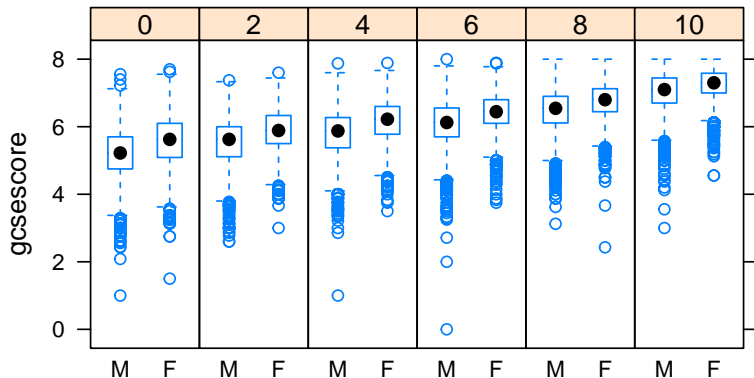
BOXPLOT MIT LATTICE ZEICHNEN

```
bwplot(factor(score) ~ gcsescore | gender, Chem97)
```



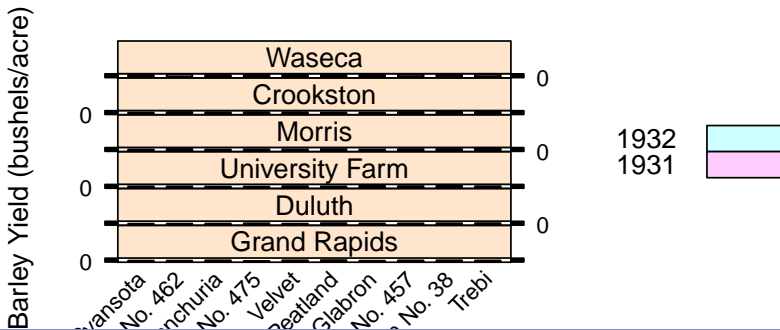
BOXPLOT MIT LATTICE ZEICHNEN

```
bwplot(gcsescore ~ gender | factor(score), Chem97,  
       layout = c(6, 1))
```



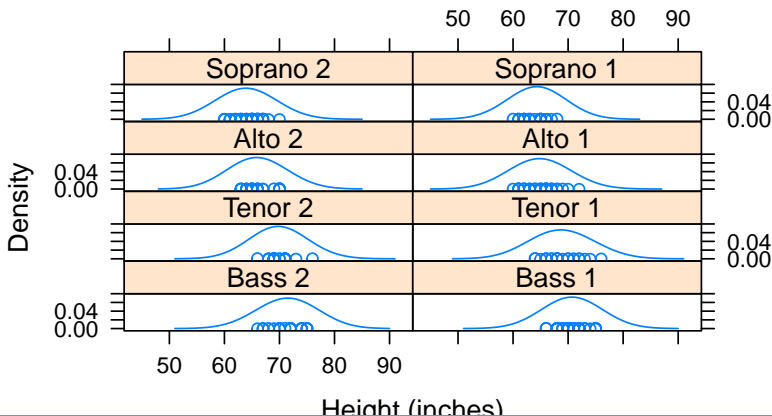
UNIVARIATE PLOTS

```
barchart(yield ~ variety | site, data = barley,  
  groups = year, layout = c(1,6), stack = TRUE,  
  auto.key = list(space = "right"),  
  ylab = "Barley Yield (bushels/acre)",  
  scales = list(x = list(rot = 45)))
```



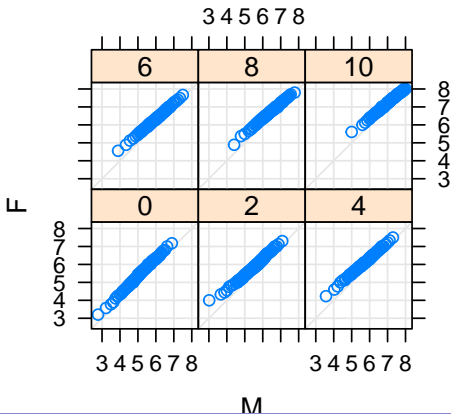
DENSITYPLOT

```
densityplot( ~ height | voice.part, data = singer, layout  
            xlab = "Height (inches)", bw = 5)
```



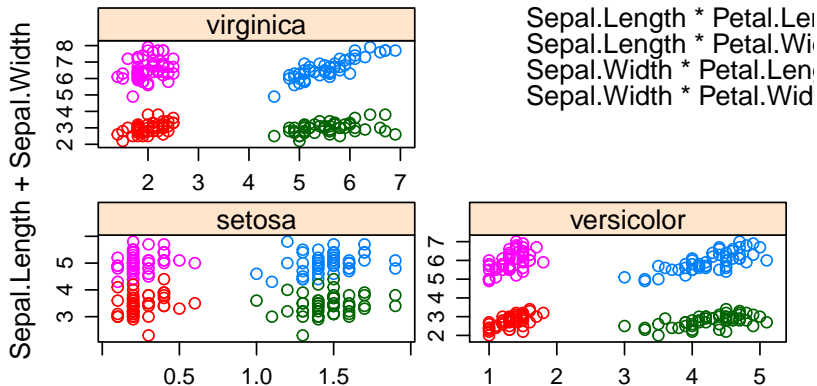
BIVARIATE PLOTS

```
qq(gender ~ gcsescore | factor(score), Chem97,  
  f.value = ppoints(100), type = c("p", "g"), aspect = 1)
```



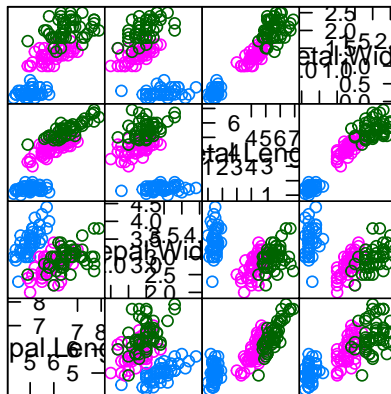
XYPLOT

```
xyplot(Sepal.Length + Sepal.Width ~ Petal.Length + Petal.Width,  
       data = iris, scales = "free", layout = c(2, 2),  
       auto.key = list(x = .6, y = .7, corner = c(0, 0)))
```



MULTIVARIATE PLOTS

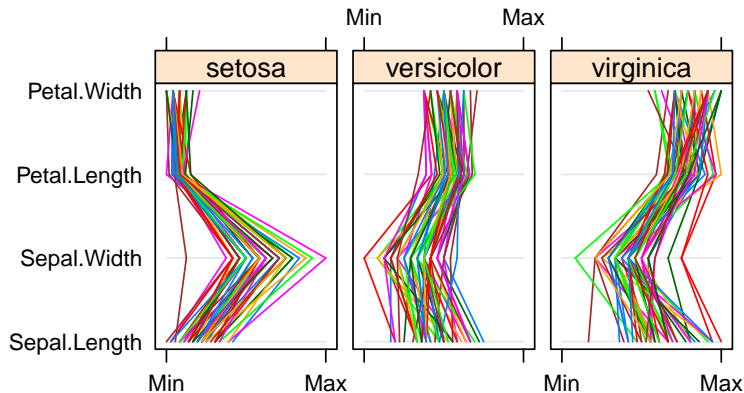
```
splom(~iris[1:4], groups = Species, data = iris)
```



Scatter Plot Matrix

PARALLEL PLOT

```
parallelplot(~iris[1:4] | Species, iris)
```



LATTICE BEFEHLE

- ▶ Übersicht aller Lattice Befehle

DIE LINEARE REGRESSION

DIE LINEARE REGRESSION

JOHN H. MAINDONALD AND W. JOHN BRAUN - **Data Analysis and Graphics Data and Functions**

- ▶ Einführung in R
- ▶ Datenanalyse
- ▶ Statistische Modelle
- ▶ Inferenzkonzepte
- ▶ Regression mit einem Prädiktor
- ▶ Multiple lineare Regression
- ▶ Ausweitung des linearen Modells
- ▶ ...

LINEARE REGRESSION IN R - BEISPIELDATENSATZ

```
data(mtcars)
```

HILFE FÜR DEN MTCARS DATENSATZ:

```
?mtcars
```

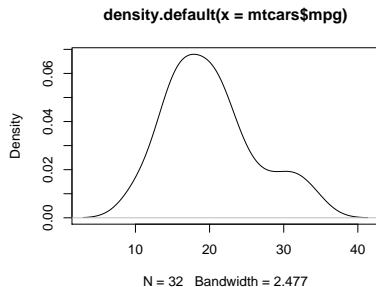
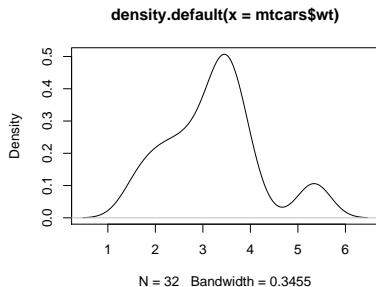
	mpg	cyl	disp	hp	drat	wt	model
21.0	6	160	110	3.90	2.620	Mazda RX4	
21.0	6	160	110	3.90	2.875	Mazda RX4 Wag	
22.8	4	108	93	3.85	2.320	Datsun 710	
21.4	6	258	110	3.08	3.215	Hornet 4 Drive	
18.7	8	360	175	3.15	3.440	Hornet Sportabout	
18.1	6	225	105	2.76	3.460	Valiant	

VARIABLEN DES MTCARS DATENSATZES

- ▶ mpg - Miles/(US) gallon
- ▶ cyl - Number of cylinders
- ▶ disp - Displacement (cu.in.)
- ▶ hp - Gross horsepower
- ▶ drat - Rear axle ratio
- ▶ wt - Weight (1000 lbs)
- ▶ qsec - 1/4 mile time
- ▶ vs - Engine (0 = V-shaped, 1 = straight)
- ▶ am - Transmission (0 = automatic, 1 = manual)
- ▶ gear - Number of forward gears
- ▶ carb - Number of carburetors

VERTEILUNGEN FÜR ZWEI VARIABLEN VON MTCARS

```
par(mfrow=c(1,2))  
plot(density(mtcars$wt)); plot(density(mtcars$mpg))
```



EIN EINFACHES REGRESSIONSMODELL

ABHÄNGIGE VARIABLE - MEILEN PRO GALLONE (MPG)

UNABHÄNGIGE VARIABLE - GEWICHT (WT)

```
m1 <- lm(mpg ~ wt,data=mtcars)
```

```
m1
```

```
##
```

```
## Call:
```

```
## lm(formula = mpg ~ wt, data = mtcars)
```

```
##
```

```
## Coefficients:
```

```
## (Intercept)          wt
```

```
##      37.285      -5.344
```

DIE MODELLFORMEL

MODELL OHNE ABSCHNITT

```
m2 <- lm(mpg ~ - 1 + wt,data=mtcars)
summary(m2)$coefficients
```

```
##      Estimate Std. Error  t value    Pr(>|t|)
## wt  5.291624   0.5931801  8.920771 4.55314e-10
```

WEITERE VARIABLEN HINZUFÜGEN

```
m3 <- lm(mpg ~ wt + cyl,data=mtcars)
summary(m3)$coefficients
```

```
##              Estimate Std. Error  t value    Pr(>|t|)
## (Intercept) 39.686261   1.7149840 23.140893 3.043182e-20
## wt          -3.190972   0.7569065 -4.215808 2.220200e-04
## cyl          -1.507795   0.4146883 -3.635972 1.064282e-03
```

SUMMARY DES MODELLS

```
summary(m3)
```

```
##
## Call:
## lm(formula = mpg ~ wt + cyl, data = mtcars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.2893 -1.5512 -0.4684  1.5743  6.1004
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   39.6863     1.7150  23.141  < 2e-16 ***
## wt           -3.1910     0.7569  -4.216  0.000222 ***
## cyl          -1.5078     0.4147  -3.636  0.001064 **
## ---
```

R ARBEITET MIT OBJEKTEN

- ▶ m3 ist nun ein spezielles Regressions-Objekt
- ▶ Auf dieses Objekt können nun verschiedene Funktionen angewendet werden

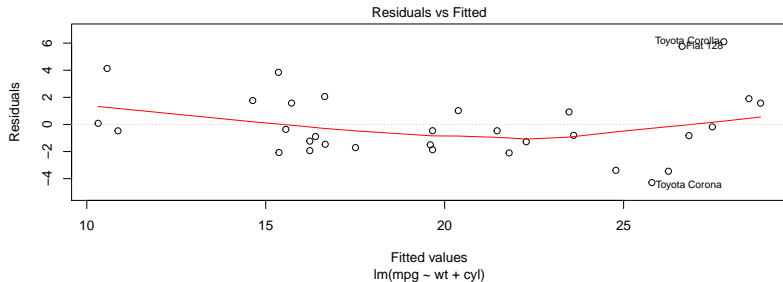
```
predict(m3) # Vorhersage
```

##	Mazda RX4	Mazda RX4 Wag	Datsun
##	22.27914	21.46545	26.25
##	Hornet 4 Drive	Hornet Sportabout	Vali
##	20.38052	16.64696	19.59
##	Duster 360	Merc 240D	Merc
##	16.23213	23.47588	23.60
##	Merc 280	Merc 280C	Merc 45
##	19.66255	19.66255	14.63
##	Merc 450SL	Merc 450SLC	Cadillac Fleetw
##	15.72158	15.56203	10.87

RESIDUENPLOT

- ▶ Sind Annahmen des linearen Regressionsmodells verletzt?
- ▶ Dies ist der Fall, wenn ein Muster abweichend von einer Linie zu erkennen ist. (Hier ist der Datensatz sehr klein)

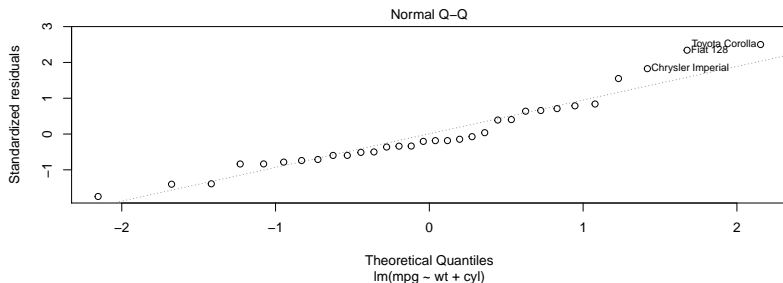
```
plot(m3,1)
```



RESIDUENPLOT

- ▶ Wenn Residuen normalverteilt sind sollten sie auf Linie sein.

```
plot(m3,2)
```



WEITERE MÖGLICHKEITEN DIE FORMEL ZU SPEZIFIZIEREN

INTERAKTIONSEFFEKT

```
# effect of cyl and interaction effect:  
m3a<-lm(mpg~wt*cyl,data=mtcars)  
  
# only interaction effect:  
m3b<-lm(mpg~wt:cyl,data=mtcars)
```

DEN LOGARITHMUS NEHMEN

```
m3d<-lm(mpg~log(wt),data=mtcars)
```

EIN MODELL MIT INTERAKTIONSEFFEKT

DISP - HUBRAUM

```
m3d<-lm(mpg~wt*disp,data=mtcars)
m3dsum <- summary(m3d)
m3dsum$coefficients
```

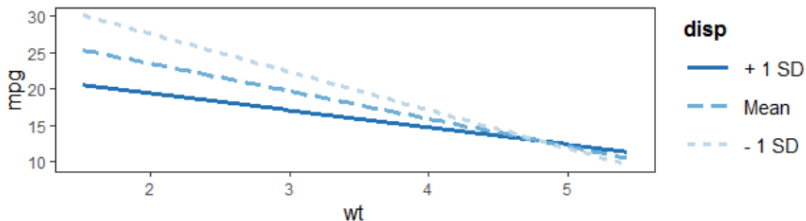
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	44.08199770	3.123062627	14.114990	2.955567e-14
wt	-6.49567966	1.313382622	-4.945763	3.216705e-05
disp	-0.05635816	0.013238696	-4.257078	2.101721e-05
wt:disp	0.01170542	0.003255102	3.596022	1.226988e-03

INTERAKTIONEN UNTERSUCHEN

```
install.packages("jtools")
```

```
library(jtools)  
interact_plot(m3d, pred = "wt", modx = "disp")
```

- ▶ Mit einem kontinuierlichen Moderator (in unserem Fall Disp) erhält man drei Zeilen - 1 Standardabweichung über und unter dem Mittelwert und der Mittelwert selbst.



EIN GENAUERER BLICK AUF INTERAKTIONSEFFEKTE

```
m_cyl <- lm(mpg ~ wt * cyl, data = mtcars)
```

DAS PAKET INTERPLOT

```
library(interplot)
```

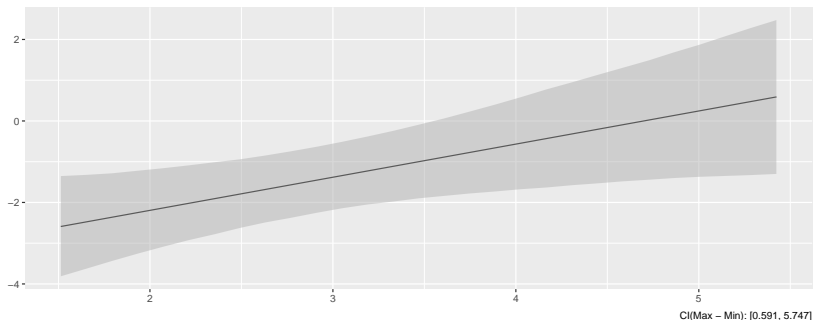
```
interplot(m = m_cyl, var1 = "cyl", var2 = "wt")
```

- ▶ var1 - die Variable für die der Koeffizient geplottet werden soll
- ▶ var2 - Variable auf die der Koeffizient konditional ist

FRAGESTELLUNG FÜR BEISPIEL

Wir wollen wissen, wie sich das Gewicht eines Autos auf den Koeffizienten für die Anzahl der Zylinder auswirkt. Zu erklärende Variable ist die Laufleistung.

INTERAKTIONSEFFEKT VISUALISIEREN



Die Darstellung zeigt, dass mit zunehmendem Fahrzeuggewicht (x-Achse) auch die Größe des Koeffizienten der Anzahl der Zylinder zunimmt (y-Achse).

- Eine detailliertere Beschreibung ist in der **interplot Vignette** zu bekommen.

BEISPIEL: OBJEKTORIENTIERUNG

- ▶ m3 ist nun ein spezielles Regressionsobjekt
- ▶ Verschiedene Funktionen können auf dieses Objekt angewendet werden

```
predict(m3) # Prediction  
resid(m3)  # Residuals
```

##	Mazda RX4	Mazda RX4 Wag	Datsun 710
##	22.27914	21.46545	26.25203
##	Hornet Sportabout	Valiant	
##	16.64696	19.59873	
##	Mazda RX4	Mazda RX4 Wag	Datsun 710
##	-1.2791447	-0.4654468	-3.4520262
##	Hornet Sportabout	Valiant	
##	2.0530424	-1.4987281	

EINE MODELLVORHERSAGE MACHEN

```
pre <- predict(m1)
head(mtcars$mpg)
```

```
## [1] 21.0 21.0 22.8 21.4 18.7 18.1
```

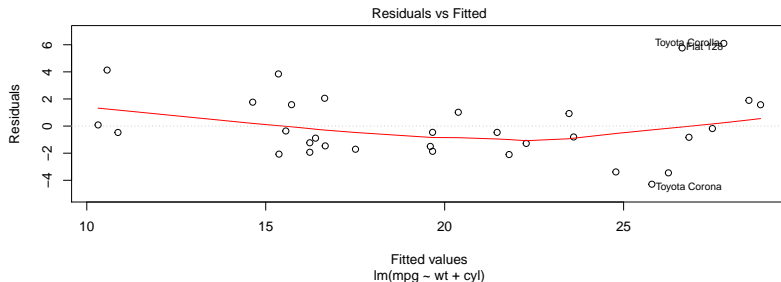
```
head(pre)
```

##	Mazda RX4	Mazda RX4 Wag	Datsun 710
##	23.28261	21.91977	24.88595
##	Hornet Sportabout	Valiant	
##	18.90014	18.79325	

RESIDUENPLOT - MODELLANNAHMEN VERLETZT?

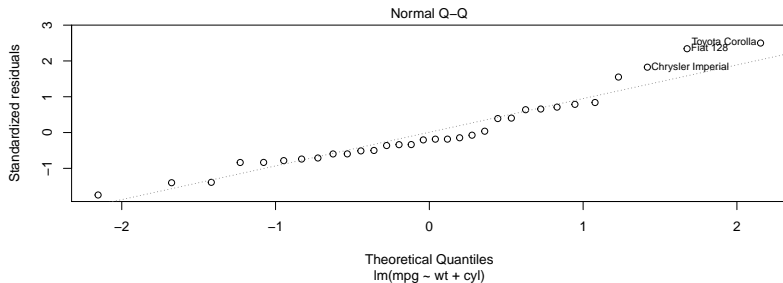
- Gibt es ein Muster in der Abweichung von der Linie

```
plot(m3,1)
```



RESIDUENPLOT

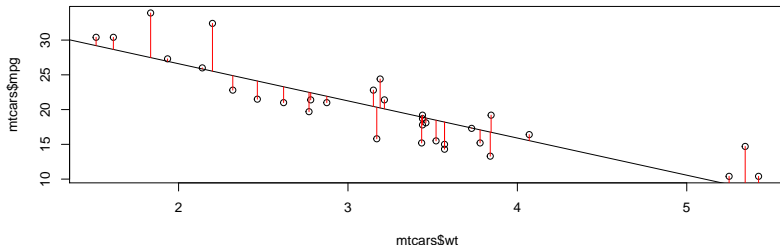
```
plot(m3,2)
```



- Bei Normalverteilung liegen Residuen auf gleicher Linie

REGRESSIONSDIAGNOSTIK MIT BASIS-R

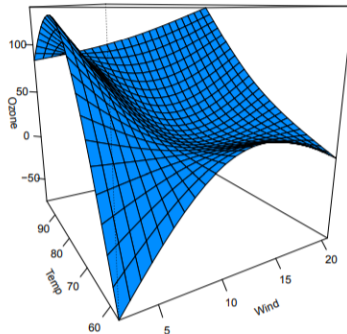
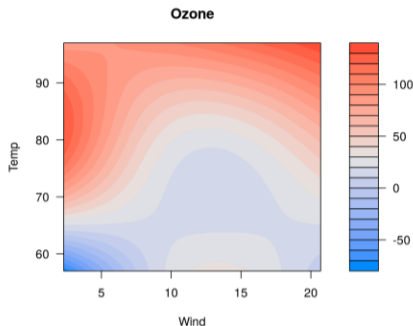
```
plot(mtcars$wt,mtcars$mpg)
abline(m1)
segments(mtcars$wt, mtcars$mpg, mtcars$wt, pre, col="red")
```



DAS VISREG-PAKET

```
install.packages("visreg")
```

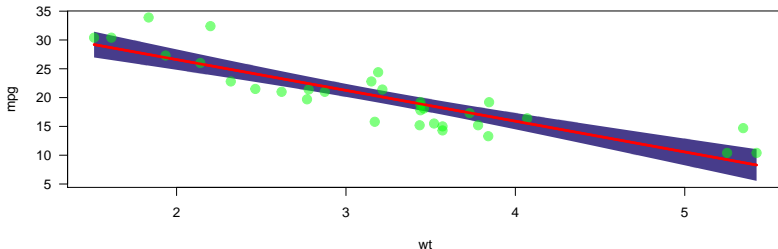
```
library(visreg)
```



DAS VISREG-PAKET

- ▶ Das Default-Argument für `type` ist `conditional`.
- ▶ Scatterplot von `mpg` und `wt` mit Regressionslinie und Konfidenzbändern

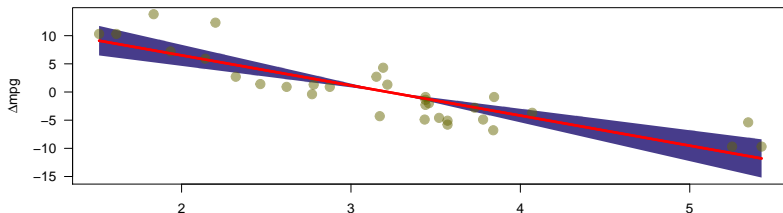
```
visreg(m1, "wt", type = "conditional")
```



Visualisierung mit visreg

- ▶ Zweites Argument - Spezifikation der Kovariaten in der Graphik
- ▶ Das Diagramm zeigt die Auswirkung auf den erwarteten Wert des Regressors, wenn die Variable x von einem Referenzpunkt auf der x -Achse wegbewegt wird (bei numerischen Variablen der Mittelwert).

```
visreg(m1, "wt", type = "contrast")
```



REGRESSION MIT FAKTOREN

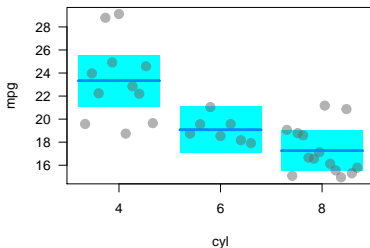
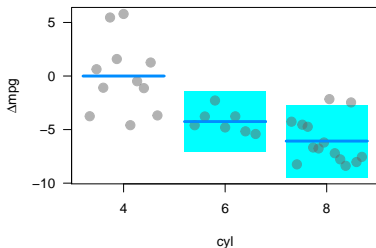
- ▶ Die Effekte von Faktoren können auch mit visreg visualisiert werden:

```
mtcars$cyl <- as.factor(mtcars$cyl)
m4 <- lm(mpg ~ cyl + wt, data = mtcars)
# summary(m4)
```

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	33.990794	1.8877934	18.005569	6.257246e-17
## cyl6	-4.255582	1.3860728	-3.070244	4.717834e-03
## cyl8	-6.070860	1.6522878	-3.674214	9.991893e-04
## wt	-3.205613	0.7538957	-4.252065	2.130435e-04

EFFEKTE VON FAKTOREN

```
par(mfrow=c(1,2))  
visreg(m4, "cyl", type = "contrast")  
visreg(m4, "cyl", type = "conditional")
```



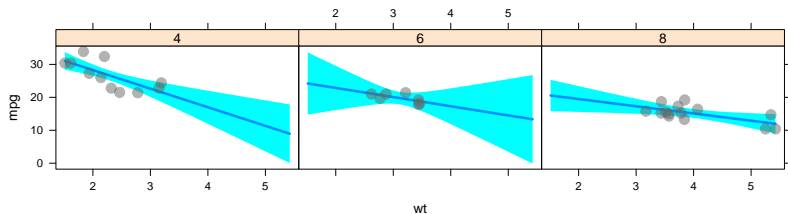
DAS PAKET VISREG - INTERAKTIONEN

```
m5 <- lm(mpg ~ cyl*wt, data = mtcars)
# summary(m5)
```

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	39.571196	3.193940	12.3894599	2.058359e-1
## cyl6	-11.162351	9.355346	-1.1931522	2.435843e-0
## cyl8	-15.703167	4.839464	-3.2448150	3.223216e-0
## wt	-5.647025	1.359498	-4.1537586	3.127578e-0
## cyl6:wt	2.866919	3.117330	0.9196716	3.661987e-0
## cyl8:wt	3.454587	1.627261	2.1229458	4.344037e-0

DEN GRAPHIKOUTPUT MIT LAYOUT KONTROLLIEREN

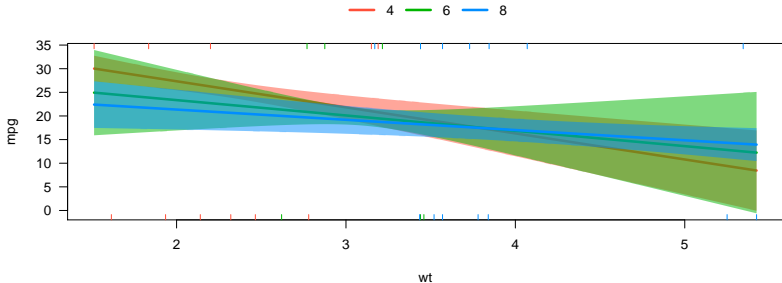
```
visreg(m5, "wt", by = "cyl", layout=c(3,1))
```



DAS PAKET VISREG - INTERAKTIONSEFFEKTE ÜBEREINANDER LEGEN

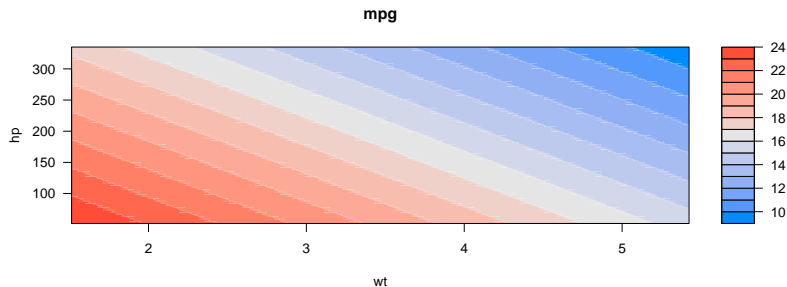
```
m6 <- lm(mpg ~ hp + wt * cyl, data = mtcars)
```

```
visreg(m6, "wt", by="cyl", overlay=TRUE, partial=FALSE)
```



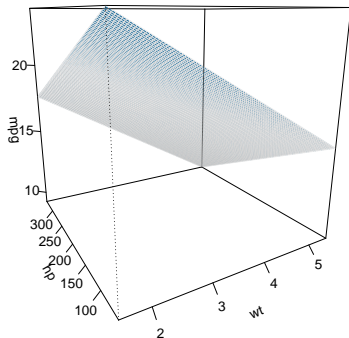
DAS PAKET VISREG - VISREG2D

```
visreg2d(m6, "wt", "hp", plot.type = "image")
```



DAS PAKET VISREG - SURFACE

```
visreg2d(m6, "wt", "hp", plot.type = "persp")
```



AUFGABE LINEARE REGRESSION

Der Datensatz `toycars` beschreibt die Route von drei Spielzeugautos, die Rampen in verschiedenen Winkeln runterfahren.

- ▶ `angle`: Rampenwinkel
 - ▶ `distance`: Entfernung die von dem Spielzeugauto zurück gelegt wird.
 - ▶ `car`: Autotyp (1, 2 or 3)
- (A) Lese den Datensatz `toycars` ein und konvertiere die Variable `car` des Datensatzes in einen Faktor (`as.factor`).
- (B) Erstelle drei Box-Plots, in denen die von den Autotypen zurückgelegte Strecke visualisiert wird.

AUFGABE LINEARE REGRESSION II

- (c) Schätze für jeden Autotyp die Parameter des folgenden linearen Modell; nutze dafür die Funktion `lm()`

$$distance_i = \beta_0 + \beta_1 \cdot angle_i + \epsilon_i$$

- (d) Überprüfe die Anpassung des Modells indem Du die drei Regressionslinien in den Scatterplot einzeichnest (`distance` gegen `angle`). Spricht das R^2 für eine gute Modellanpassung?

EINEN SCHÖNEN OUTPUT MIT DEM PAKET **stargazer** erzeugen

```
library(stargazer)
stargazer(m3, type="html")
```

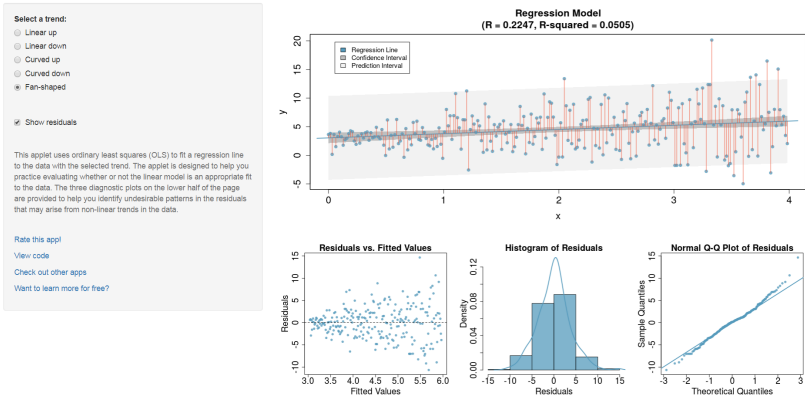
BEISPIEL HTML OUTPUTS:

	<i>Dependent variable:</i>
	mpg
wt	-3.125*** (0.911)
cyl	-1.510*** (0.422)
am	0.176 (1.304)
Constant	39.418***

SHINY APP - DIAGNOSTIKEN FÜR DIE EINFACHE LINEARE REGRESSION

https://gallery.shinyapps.io/slr_diag/

Diagnostics for simple linear regression



LINKS - LINEARE REGRESSION

- ▶ Regression - **r-bloggers**
- ▶ Das komplette Buch von **Faraway**- sehr intuitiv geschriebenes Buch
- ▶ Gute Einführung auf **Quick-R**
- ▶ **Multiple Regression**
- ▶ **15 Arten von Regressionen die man kennen sollte**
- ▶ **ggeffects** - Erzeuge saubere Datensätze mit marginellen Effekten für 'ggplot' aus Modell Outputs

DIE LOGISTISCHE REGRESSION

AGRESTI - Categorical Data Analysis (2002)



- ▶ Sehr intuitiv geschriebenes Buch
- ▶ Sehr detailliertes Skript von **Laura A. Thompson**
- ▶ Das Buch behandelt die kategoriale Datenanalyse ganz grundsätzlich.

Extending the Linear Model with R

- ▶ Logistische Regression eingängig erklärt
- ▶ Beispiel mit R-Code
 - ▶ Faraway - **Extending the linear model with R**
 - ▶ Faraway - **Practical Regression and Anova using R**

IMPORTIEREN DES GESIS PANELS DATENSATZES

```
library(readstata13)
path <- "D:/Daten/GitLab/IntroDataAnalysis/data/"
datf <- read.dta13(paste0(path,"ZA5666_v1-0-0_Stata14.dta"),
                   convert.factors = F)
```

Das Argument `convert.factors`:

- ▶ `logical`. Wenn `TRUE`, werden Faktoren aus dem Stata Werte Labeln erzeugt.

EINE FUNKTION UM FEHLENDE WERTE ZU REKODIEREN

```
code_miss <- function(var){  
  misvals <- c(-11,-22,-33,-44,-55,-66,-77,-88,-99,-111)  
  var[var %in% misvals] <- NA  
  return(var)  
}
```


VARIABLEN FÜR DAS GLM

- ▶ a11d056z: Altersgruppe

```
table(datf$a11d056z)
```

```
##
```

```
## -99    1    2    3    4    5    6    7    8    9   10   11   12   13  
##    5   31   87  101   91   83  100  163  159  133   64   56  105  44
```

```
age <- code_miss(datf$a11d056z)
```

```
table(age)
```

```
## age
```

```
##    1    2    3    4    5    6    7    8    9   10   11   12   13  
##   31   87  101   91   83  100  163  159  133   64   56  105  44
```

GP VARIABLE A11D094A: KINDER UNTER 16 JAHRE

Leben in Ihrem Haushalt Kinder unter 16 Jahren?

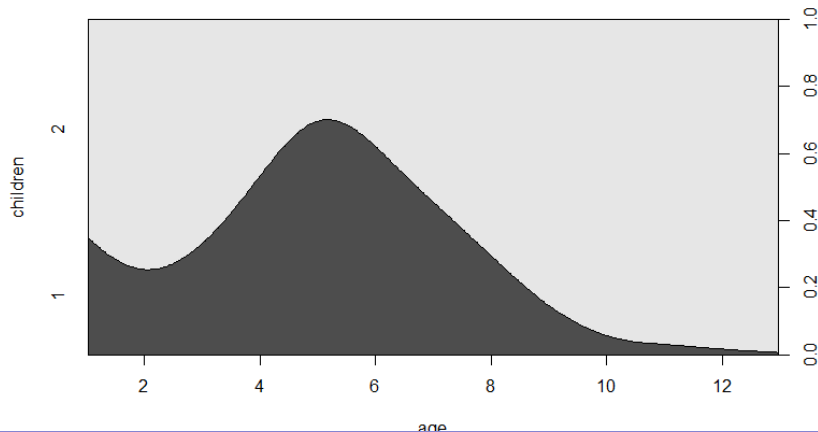
- ▶ 1 Ja
- ▶ 2 Nein

```
children <- as.factor(code_miss(datf$a11d094a))  
table(children)
```

```
## children  
##      1      2  
## 325 681
```

CONDITIONAL DENSITY PLOT (GESIS PANEL)

```
cdplot(children ~ age)
```



BINÄRE ABHÄNGIGE VARIABLEN IM GLM

- ▶ Die **logistische Regression** gehört zur Klasse der generalisierten linearen Modellen (GLM)
- ▶ Die Funktion zur Schätzung eines Modells dieser Klasse heißt `glm()`

EIN GLM SPEZIFIZIEREN

- ▶ Formel-Objekt
- ▶ die Klasse (binomial, gaussian, gamma)
- ▶ mit einer Link Funktion (logit, probit, cauchit, log, cloglog)

muss spezifiziert

UMGANG MIT FEHLENDEN WERTEN

```
dat1 <- data.frame(children,age)
dat1 <- na.omit(dat1)
```

LOGISTISCHE REGRESSION MIT R

```
glm_1 <- glm(children ~ age, data=dat1,  
             family = binomial())
```

```
sum_glm1 <- summary(glm_1)  
sum_glm1$coefficients
```

	Estimate	Std. Error	z value	Pr(> z)
## (Intercept)	-0.7194058	0.16384386	-4.390801	1.129338e-05
## age	0.2225862	0.02376266	9.367056	7.458415e-21

Die Koeffizienten interpretieren

Wir betrachten das logistische Modell der Kinder im Haushalt als eine Funktion des Alters.

```
sum_glm1$coefficients
```

	Estimate	Std. Error	z value	Pr(> z)
## (Intercept)	-0.7194058	0.16384386	-4.390801	1.129338e-05
## age	0.2225862	0.02376266	9.367056	7.458415e-21

- ▶ Die Schätzungen und Standardfehler werden mit Log Odds angegeben, nicht mit der Wahrscheinlichkeit.
- ▶ Die p-Werte bedeuten das Gleiche, wie bei der linearen Regression.

DER INVERSE LOGIT

```
sum_glm1$coefficients
```

```
##              Estimate Std. Error   z value    Pr(>|z|)
## (Intercept) -0.7194058 0.16384386 -4.390801 1.129338e-05
## age          0.2225862 0.02376266  9.367056 7.458415e-21
```

- ▶ Die Koeffizienten können nicht so einfach interpretiert werden
- ▶ Wir müssen den inversen Logit verwenden, um etwas auszusagen.

Werte für die Log-odds von 0.2225862 sind das Gleiche, wie die Wahrscheinlichkeit: 0.5554179.

```
faraway::ilogit(sum_glm1$coefficients[1,1])
```

```
## [1] 0.3275238
```


ZUM AchsenABSCHNITT IN EINEM LOGISTISCHEN MODELL

- ▶ Es ist möglich, dass der Schätzwert für den Achsenabschnitt kleiner als null ist.
- ▶ Das bedeutet, dass die log-odds negativ sind und NICHT die Wahrscheinlichkeit.
- ▶ Ein Log-Odd Wert von 0 bedeutet eine Wahrscheinlichkeit von 0.5.

LOG-ODDS UND DIE WAHRSCHEINLICHKEIT

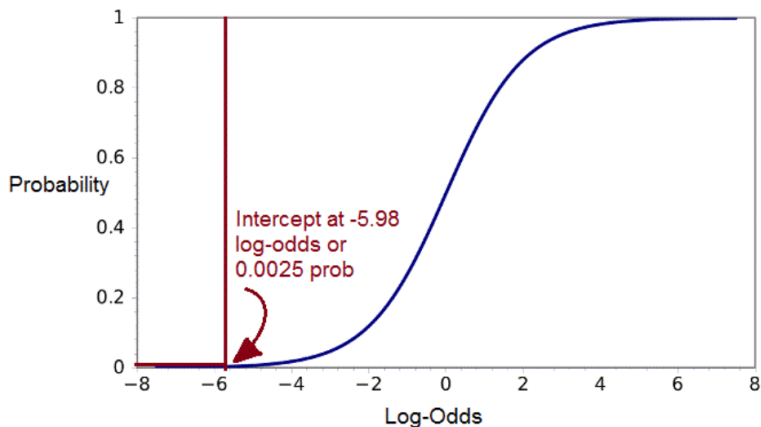
- ▶ Die Log-odds steigen an, wenn die Wahrscheinlichkeit auch ansteigt.

Daraus folgt. . .

- ▶ Ein positiver Steigungskoeffizient bedeutet, dass der Response-Wert mit zunehmendem Wert für die erklärende Variable auch zunimmt.
- ▶ In unserem Fall heißt das: Die Wahrscheinlichkeit, dass sich im Haushalt Kinder befinden steigt mit dem Alter des Befragten.

DAS ERGEBNIS GRAPHISCH DARSTELLEN

Es resultiert eine Sigmoid-Kurve, anstatt einer Geraden mit konstanter Steigerungsrate wie bei der linearen Regression.



LOGISTISCHE REGRESSIONSFORMEL

DAS MODELL ALS FORMEL:

$$\text{Log-Odds(Children)} = -0.7194058 + 0.2225862(\text{Age}) + \text{Fehler}$$

- Wir können Werte in die Formel einsetzen um die vorhergesagten Log-Odds für unterschiedliche Altersklassen zu bekommen.

BEISPIEL: LOG-ODDS FÜR DIE ALTERSGRUPPE 5

$$-0.7194058 + 0.2225862 \cdot (5) = 0.3935251$$

Wahrscheinlichkeit für Kinder in der Altersgruppe 5

```
ilogit(0.3935251)
```

```
## [1] 0.597131
```

Die Ergebnisse interpretieren

```
anova(glm_1, test="Chisq")
```

```
## Analysis of Deviance Table
```

```
##
```

```
## Model: binomial, link: logit
```

```
##
```

```
## Response: children
```

```
##
```

```
## Terms added sequentially (first to last)
```

```
##
```

```
##
```

```
##      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
```

```
## NULL                1000        1259
```

```
## age      1      98.956         999      1160 < 2.2e-16 ***
```

```
## ---
```

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

DIE DEVIANZ

ABWEICHUNG VOM IDEALWERT

- ▶ Zweimal die Differenz zwischen der maximalen Log-Likelihood $\ell^{(M)}$ und dem Wert für das angepasste Modell
- ▶ Je niedriger die Devianz, desto besser.

```
sum_glm1 <- summary(glm_1)
sum_glm1$deviance
```

```
## [1] 1160.044
```

MC FADDEN'S R^2

```
library(psc1)  
pR2(glm_1)
```

llh	The log-likelihood from the fitted model
llhNull	The log-likelihood from the intercept-only restricted model
G2	Minus two times the difference in the log-likelihoods
McFadden	McFadden's pseudo r-squared
r2ML	Maximum likelihood pseudo r-squared
r2CU	Cragg and Uhler's pseudo r-squared

GROSSSTADTNÄHE WOHNGEEND

WIE WEIT IST ES VON IHRER WOHNUNG BIS INS ZENTRUM DER NÄCHSTEN GROSSSTADT?

- ▶ 1 - Im Großstadtzentrum
- ▶ 6 - 60 km und mehr

```
region <- code_miss(datf$bczd001a)
table(region)
```

```
## region
##      1      2      3      4      5      6
##  87 191 279 157 126 165
```


ZUFRIEDENHEIT LEBEN IN WOHNORT

ALLES IN ALLEM, WIE ZUFRIEDEN SIND SIE MIT DEM LEBEN IN [WOHNORT]?

- ▶ 1 - Sehr zufrieden
- ▶ 5 - Sehr unzufrieden

```
satisfactionplace <- datf$a11c019a  
table(satisfactionplace)
```

```
## satisfactionplace  
##    1    2    3    4    5  
## 553 534  99  30    6
```

EIN ANDERES MODELL

```
glm_2 <- glm(children ~ age + satisfactionplace*region,  
             family = binomial())
```

```
pseudor2 <- pR2(glm_2)  
pseudor2["McFadden"]
```

EINE WEITERE VARIABLE AUS DEM GESIS PANEL DATENSATZ

► Anzahl Tattoos:

```
Tatoos <- code_miss(datf$bdao067a)
Tatoos[Tatoos==97]<-0
```

```
table(Tatoos)
```

```
## Tatoos
```

```
##      0      1      2      3      4      5      6
```

```
## 871  56  28  13   7   4   8
```

GENERALISIERTE REGRESSION MIT R - MEHR FUNKTIONEN

- ▶ Logistisches Modell mit einem Probit Link:

```
probitmod <- glm(children ~ age,  
  family=binomial(link=probit))
```

- ▶ Regression mit Zähldaten:


```
modp <- glm(Tatoos ~ age,family=poisson)
```

- ▶ Proportional Odds logistic Regression aus dem Paket MASS:

```
library("MASS")  
mod_plr<-polr(a11c020a ~ a11d096b ,data=dat)
```

LINKLISTE - LOGISTISCHE REGRESSION

- Einführung in die **logistische Regression**



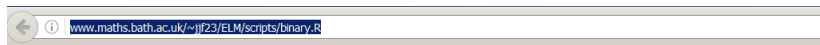
Tutorials

by William B. King, Ph.D.
Coastal Carolina University

Copyright © William B. King, Ph.D.

*I think,
therefore I
R.*

- **Code zum Buch von Faraway**



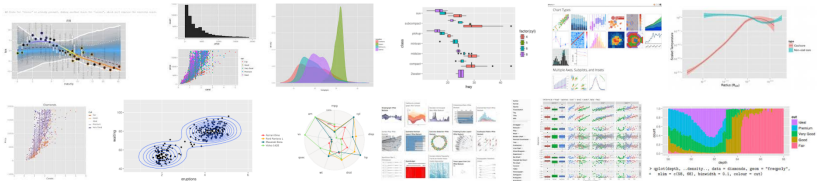
```
library(faraway)
data(oringes)
plot(damage/6 ~ temp, orings, xlim=c(25,85), ylim = c(0,1), xlab="Temperature", ylab="Prob of damage")
lmod <- lm(damage/6 ~ temp, orings)
abline(lmod)
logitmod <- glm(cbind(damage,6-damage) ~ temp, family=binomial, orings)
summary(logitmod)
plot(damage/6 ~ temp, orings, xlim=c(25,85), ylim = c(0,1), xlab="Temperature", ylab="Prob of damage")
```

- **Kategoriale Daten: - Durchführung logistische Regression in R**

GRAFIKEN MIT GGLOT

DAS PAKET GGLOT2

- ▶ Entwickelt von Hadley Wickham
- ▶ Viele Informationen unter:
 - ▶ <http://ggplot2.org/>
 - ▶ Den Graphiken liegt eine eigene Grammatik zu Grunde



EINFÜHRUNG IN GGLOT2

<http://www.r-bloggers.com/basic-introduction-to-ggplot2/>

```
install.packages("ggplot2")
```

```
library(ggplot2)
```

```
?ggplot2
```

ggplot2-package [ggplot2]

R Documentation

ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics

Description

A system for 'declaratively' creating graphics, based on "The Grammar of Graphics". You provide the data, tell 'ggplot2' how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

Author(s)

Maintainer: Hadley Wickham hadley@rstudio.com

Authors:

- Winston Chang winston@rstudio.com

DER DIAMONDS DATENSATZ

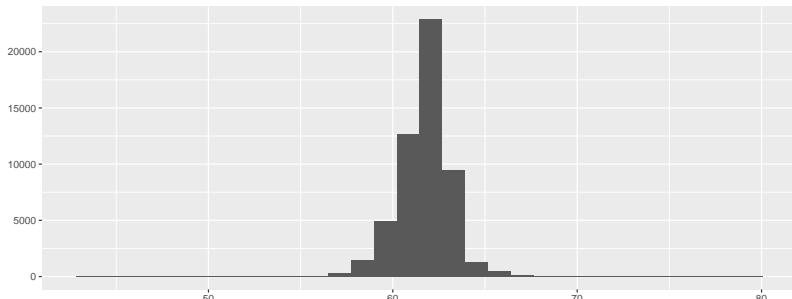
```
head(diamonds)
```

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	
0.24	Very Good	J	VVS2	62.8	57	336	3.94	3.96	

WIE NUTZT MAN QPLOT

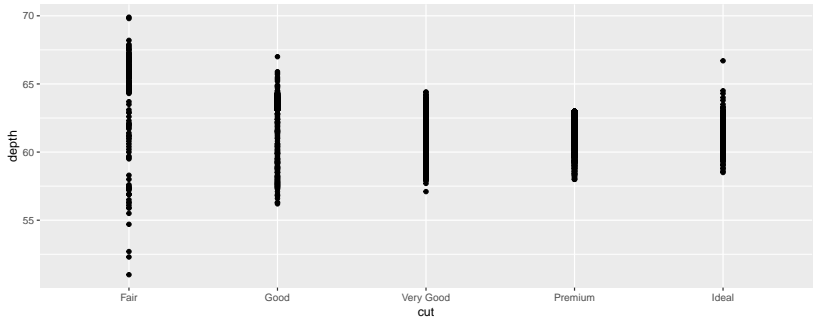
- ▶ Die Funktion `qplot` wird für schnelle Graphiken verwendet (quick plots)
- ▶ bei der Funktion `ggplot` kann man alles bis ins Detail kontrollieren

```
# histogram  
qplot(depth, data=diamonds)
```



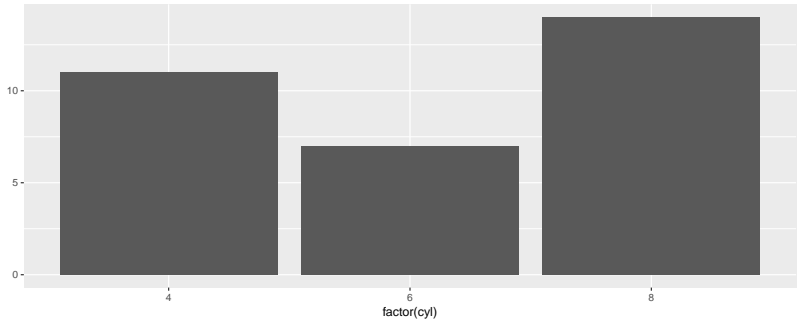
EIN BALKENDIAGRAMM

```
qplot(cut, depth, data=diamonds)
```



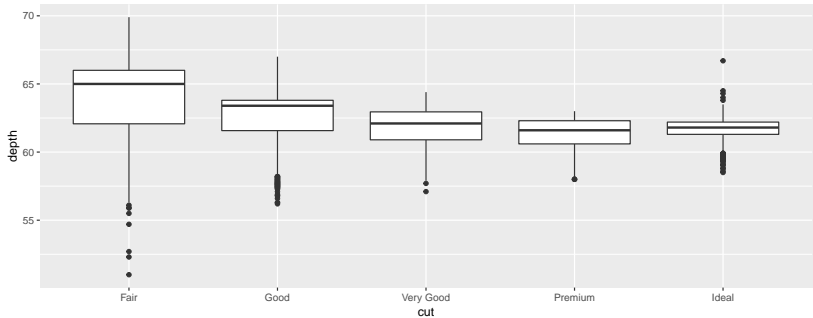
EIN WEITERES BALKENDIAGRAMM

```
qplot(factor(cyl), data=mtcars, geom="bar")
```



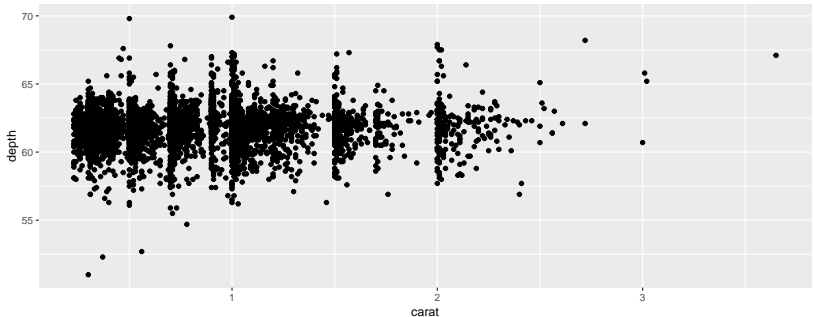
BOXPLOT

```
qplot(data=diamonds,x=cut,y=depth,geom="boxplot")
```



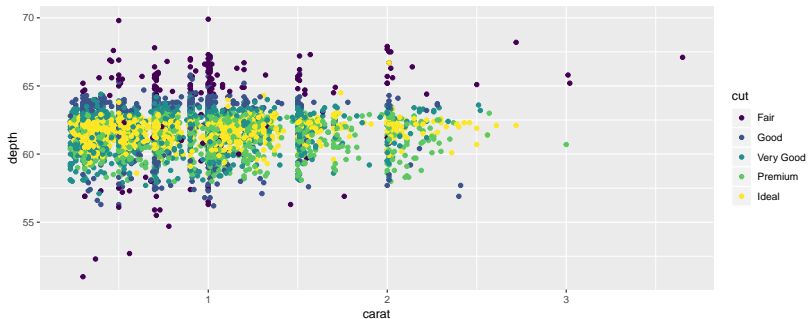
SCATTERPLOT

```
# scatterplot  
qplot(carat, depth, data=diamonds)
```



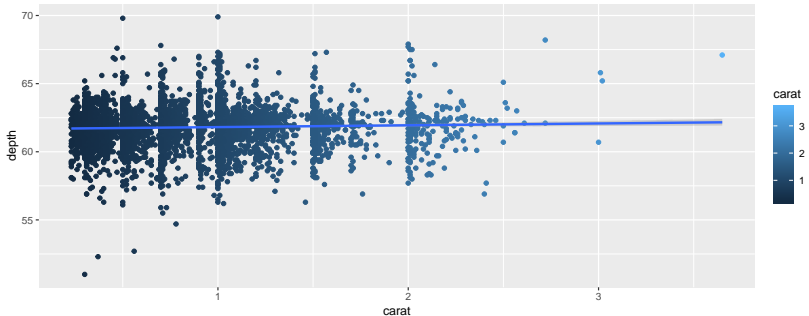
FARBE HINZU:

```
qplot(carat, depth, data=diamonds, color=cut)
```



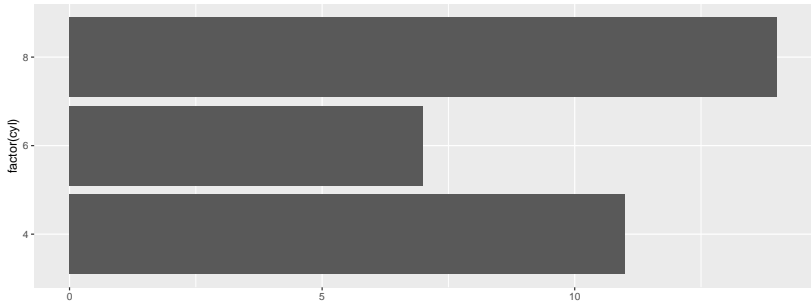
TRENDLINIE HINZUFÜGEN

```
myGG<-qplot(data=diamonds,x=carat,y=depth,color=carat)  
myGG + stat_smooth(method="lm")
```



GRAPHIK DREHEN

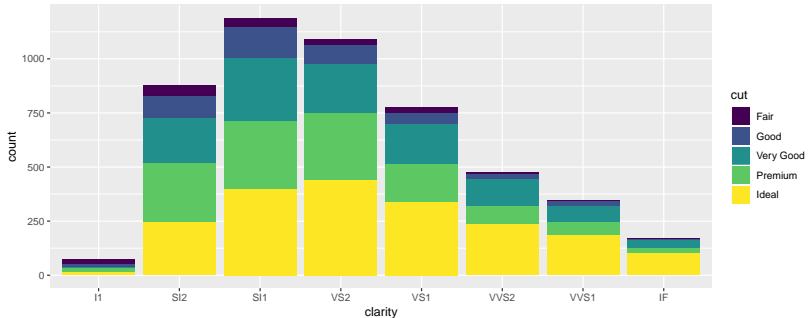
```
qplot(factor(cyl), data=mtcars, geom="bar") +  
coord_flip()
```



WIE NUTZT MAN GGPLOT

- die aesthetics:

```
ggplot(diamonds, aes(clarity, fill=cut)) + geom_bar()
```



FARBEN SELBER WÄHLEN

Es wird das Paket RColorBrewer verwendet um die Farbpalette zu ändern

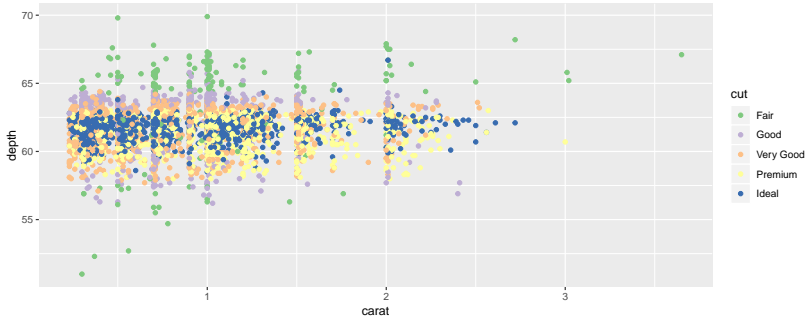
```
install.packages("RColorBrewer")
```

```
library(RColorBrewer)
myColors <- brewer.pal(5,"Accent")
names(myColors) <- levels(diamonds$cut)
colScale <- scale_colour_manual(name = "cut",
                                values = myColors)
```

<http://stackoverflow.com/questions/6919025/>

EINE GRAPHIK MIT DEN GEWÄHLTEN FARBEN

```
p <- ggplot(diamonds,aes(carat, depth,colour = cut)) +  
  geom_point()  
p + colScale
```



SPEICHERN MIT GGSAVE

```
ggsave("Graphik.jpg")
```

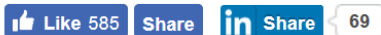
LINKS

- ▶ Warum man ggplot2 für einfache Grafiken nutzen sollte

Why I use ggplot2

February 12, 2016

By David Robinson



(This article was first published on [Variance Explained](#), and kindly contributed to R-bloggers)

590
SHARES

f Share

🐦 Tweet

- ▶ Einführung in ggplot2