

Lineare Regression

Jan-Philipp Kolb

07 Mai, 2019

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
```

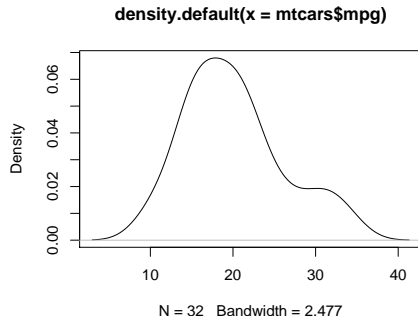
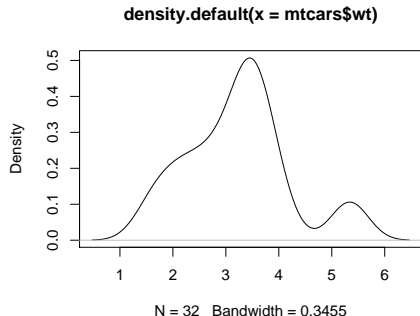
	mpg	cyl	disp	hp	drat	wt	model
21.0	6	160	110	3.90	2.620		Mazda RX4
21.0	6	160	110	3.90	2.875		Mazda RX4 Wag
22.8	4	108	93	3.85	2.320		Datsun 710
21.4	6	258	110	3.08	3.215		Hornet 4 Drive
18.7	8	360	175	3.15	3.440		Hornet Sportabout
18.1	6	225	105	2.76	3.460		Valiant

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 Achsenabschnitt

```
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

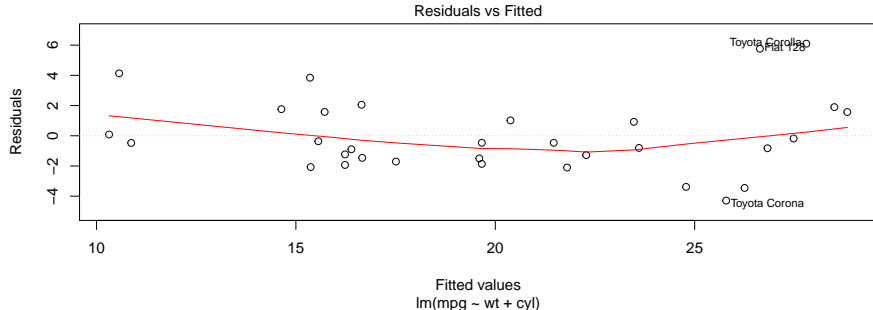
```
predict(m3) # Vorhersage
```

##	Mazda RX4	Mazda RX4 Wag	Datsun 710
##	22.27914	21.46545	26.25203
##	Hornet 4 Drive	Hornet Sportabout	Valiant
##	20.38052	16.64696	19.59873
##	Duster 360	Merc 240D	Merc 230
##	16.23213	23.47588	23.60352
##	Merc 280	Merc 280C	Merc 450SE
##	19.66255	19.66255	14.63665
##	Merc 450SL	Merc 450SLC	Cadillac Fleetwood
##	15.72158	15.56203	10.87130
##	Lincoln Continental	Chrysler Imperial	Fiat 128

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)

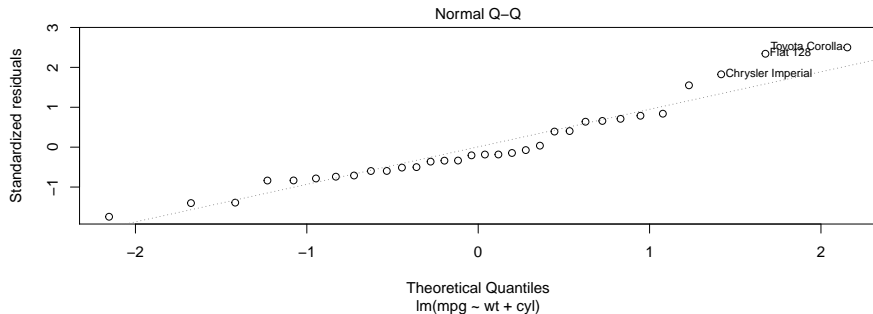
```
plot(m3,1)
```



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-04
##	wt:disp	0.01170542	0.003255102	3.596022	1.226988e-03

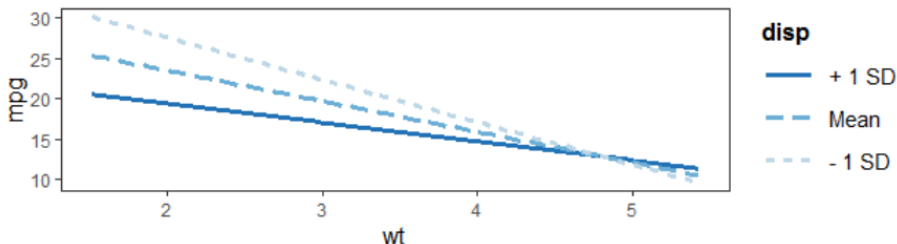
Interaktionen untersuchen

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

```
library(jtools)
```

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

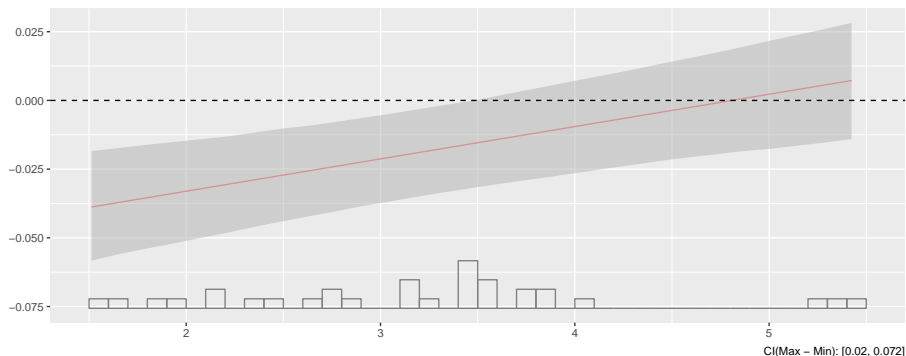
- Mit einem kontinuierlichen Moderator (in unserem Fall Disp) erhält man drei Zeilen - 1 Standardabweichung über und unter dem Mittelwert und der Mittelwert selbst.



Das Paket interplot

```
library(interplot)
```

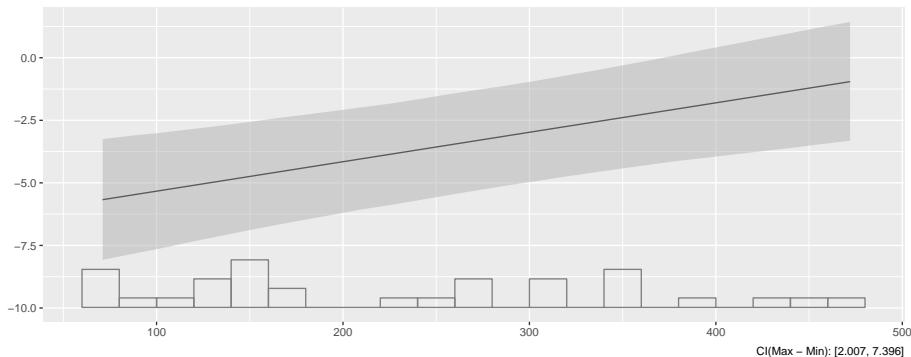
```
interplot(m = m3d, var1 = "disp", var2 = "wt", hist = TRUE) +  
  aes(color = "pink") + theme(legend.position="none") +  
  geom_hline(yintercept = 0, linetype = "dashed")
```



Noch ein interplot

- Effekt wird auf die y-Achse geplottet - wt auf der x-Achse

```
interplot(m = m3d, var1 = "wt", var2 = "disp", hist = TRUE)
```



- 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	Ho
##	22.27914	21.46545	26.25203	
##	Hornet Sportabout	Valiant		
##	16.64696	19.59873		
##	Mazda RX4	Mazda RX4 Wag	Datsun 710	Ho
##	-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
```

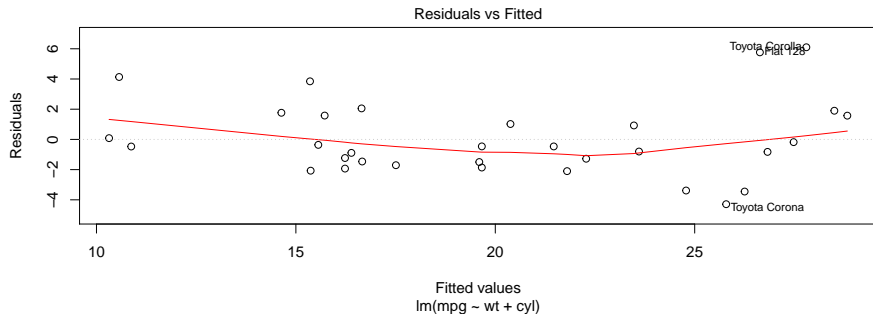
```
head(pre)
```

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

Residuenplot - Modellannahmen verletzt?

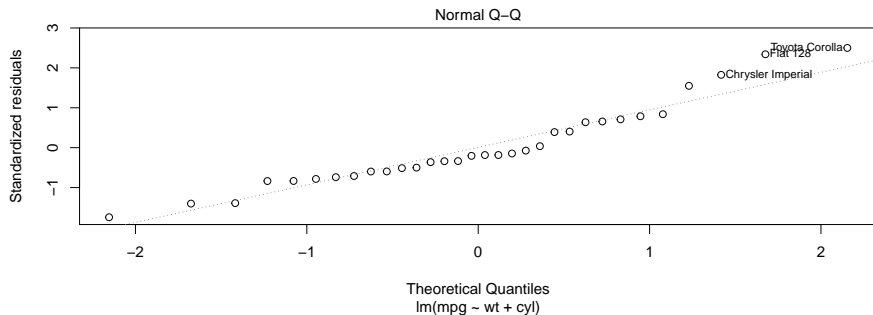
- Gibt es ein Muster in der Abweichung von der Linie

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



Residuenplot

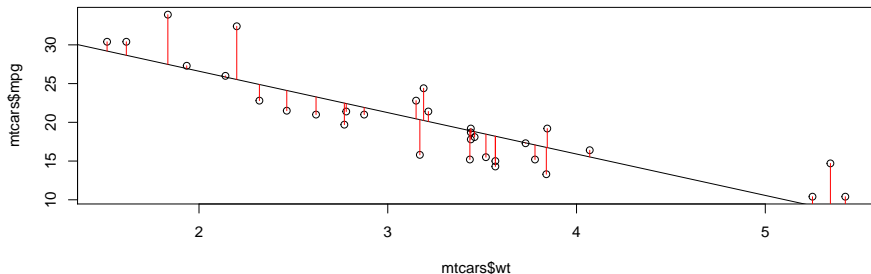
```
plot(m3,2)
```



- Bei Normalverteilung liegen Residuen auf gleicher Linie

Regressionsdiagnostik mit Basis-R

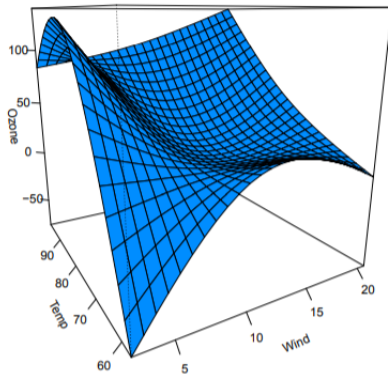
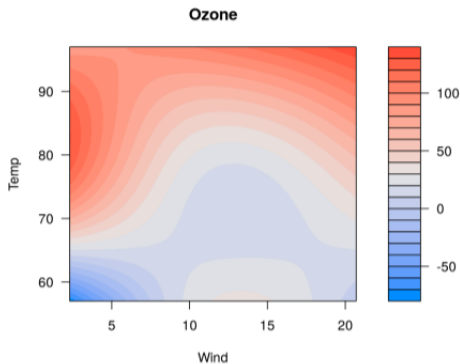
```
plot(mtcars$wt,mtcars$mpg)
abline(m1)
segments(mtcars$wt, mtcars$mpg, mtcars$wt, pre, col="red")
```



Das visreg-Paket

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

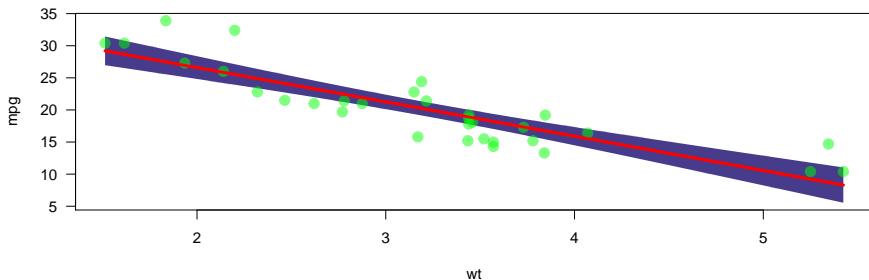
```
library(visreg)
```



Das visreg-Paket

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

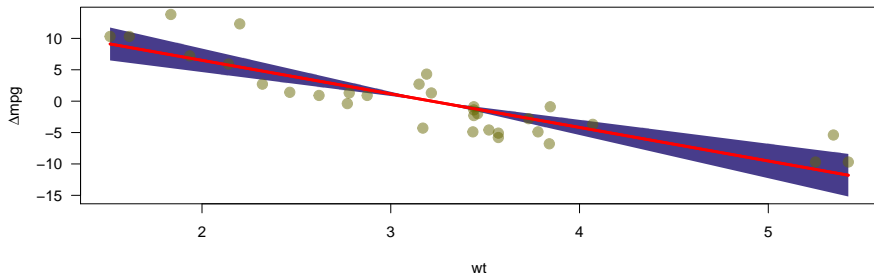
```
visreg(m1, "wt", type = "conditional")
```



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

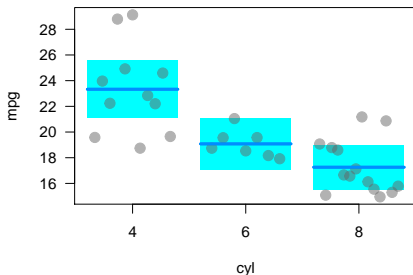
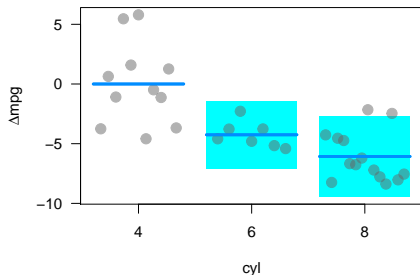
- Die Effekte von Faktoren können auch mit visreg visualisiert werden:

```
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
## cyl16	-4.255582	1.3860728	-3.070244	4.717834e-03
## cyl18	-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")
```



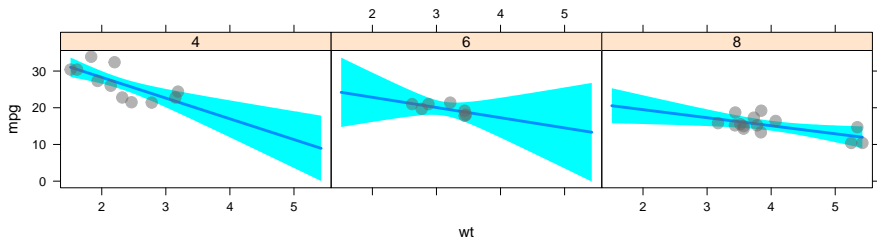
Das Paket visreg - Interaktionen

```
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-12
## cyl6	-11.162351	9.355346	-1.1931522	2.435843e-01
## cyl8	-15.703167	4.839464	-3.2448150	3.223216e-03
## wt	-5.647025	1.359498	-4.1537586	3.127578e-04
## cyl6:wt	2.866919	3.117330	0.9196716	3.661987e-01
## cyl8:wt	3.454587	1.627261	2.1229458	4.344037e-02

Den Graphikoutput mit layout kontrollieren

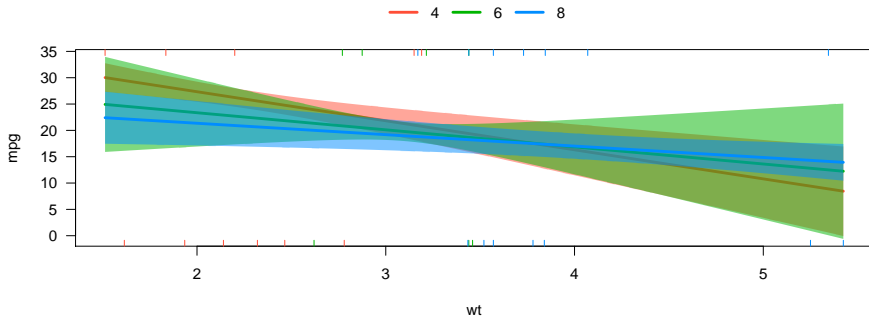
```
visreg(m5, "wt", by = "cyl", layout=c(3,1))
```



Das Paket visreg - Interaktionseffekte übereinander legen

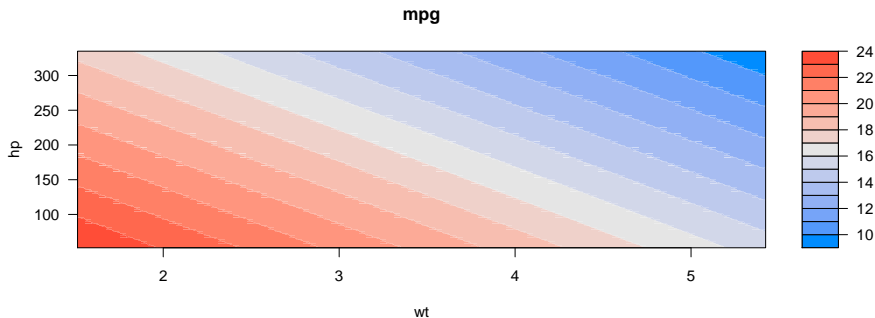
```
m6 <- lm(mpg ~ hp + wt * cyl, data = mtcars)
```

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



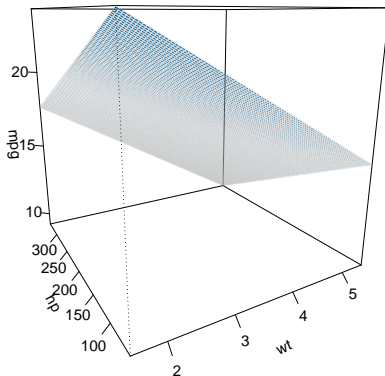
Das Paket visreg - visreg2d

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



Das Paket visreg - surface

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



B3A Aufgabe lineare Regression

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

- `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.

B3A Aufgabe lineare Regression II

- (c) Schätze für jeden Autotyp getrennt 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*** (2.641)

Shiny App - Diagnostiken für die einfache lineare Regression

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

Diagnostics for simple linear regression

Select a trend:

- ☐ Linear up
- ☐ Linear down
- ☐ Curved up
- ☐ Curved down
- ☒ Fan-shaped

☒ Show residuals

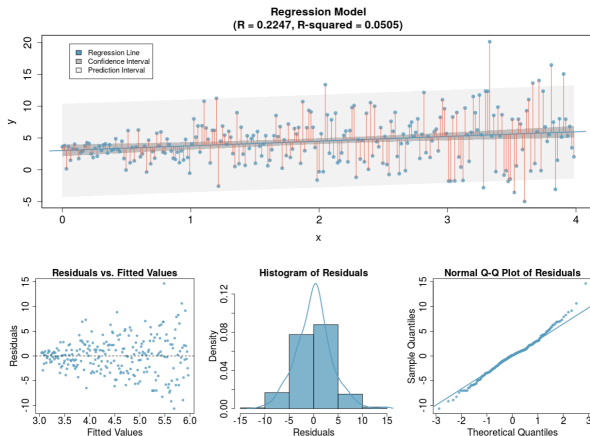
This applet uses ordinary least squares (OLS) to fit a regression line to the data with the selected trend. The applet is designed to help you practice evaluating whether or not the linear model is an appropriate fit to the data. The three diagnostic plots on the lower half of the page are provided to help you identify undesirable patterns in the residuals that may arise from non-linear trends in the data.

[Rate this app!](#)

[View code](#)

[Check out other apps](#)

[Want to learn more for free?](#)



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