Einführung in R - Lineare Regression

Jan-Philipp Kolb

12 Juni, 2019

Gute Literatur für lineare Regression in R

J H Maindonald - Using R for Data Analysis and Graphics Introduction, Code and Commentary

- Introduction to R
- Data analysis
- Statistical models
- Inference concepts
- Regression with one predictor
- Multiple linear regression
- Extending the linear model
- ...

Variablen im mtcars Datensatz

Hilfe File für den `roller` Datensatz:

```
?mtcars
```

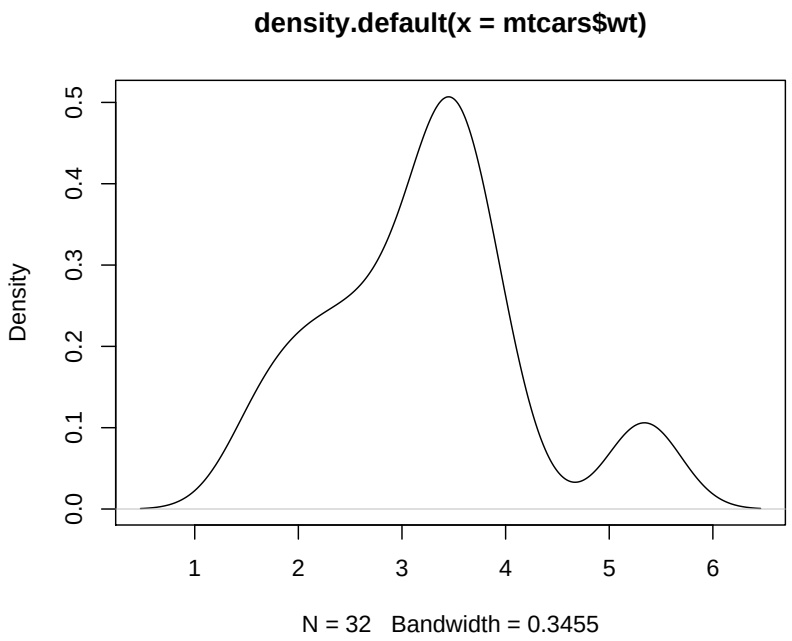
- mpg - Meilen/(US) Gallone
- cyl - Anzahl der Zylinder

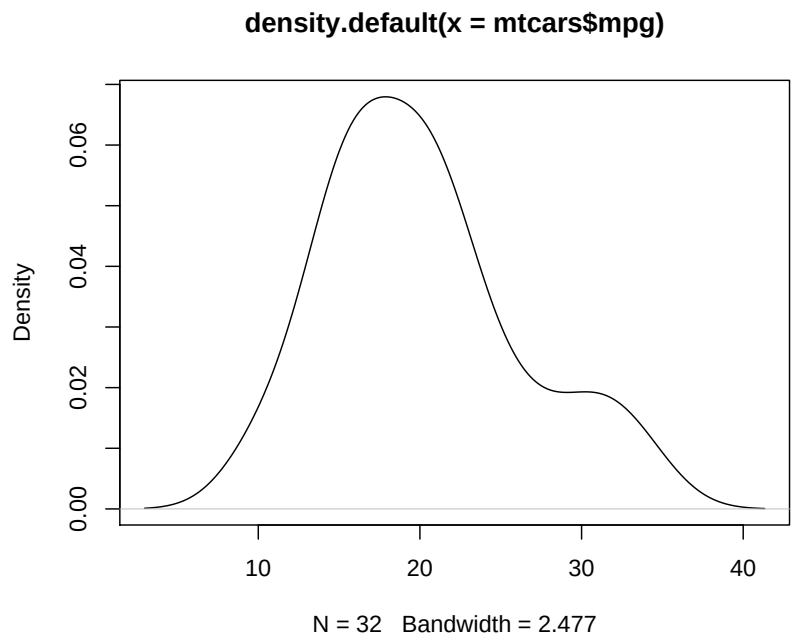
Datensatz mtcars

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0			
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0			
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1			
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1			
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0			
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1			
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0			
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1			
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1			
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1			
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1			
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0			
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0			
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0			
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0			
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0			
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0			
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1			

Verteilungen von zwei Variablen aus dem Datensatz 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 Modell Zusammenfassung:

```
summary(m1)
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5432 -2.3647 -0.1252  1.4096  6.8727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
## wt           -5.3445     0.5591  -9.559 1.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
##
## Residual standard error: 3.046 on 30 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7446
## F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

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
```

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

Variable 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
```

Plot the Effects of Variables in Interaction Terms

```
library(interplot)
```

- Eine detailliertere Erklärung findet man in der **`Interplot`** Vignette

interplot: Plot the Effects of Variables in Interaction Terms

Frederick Solt and Yue Hu

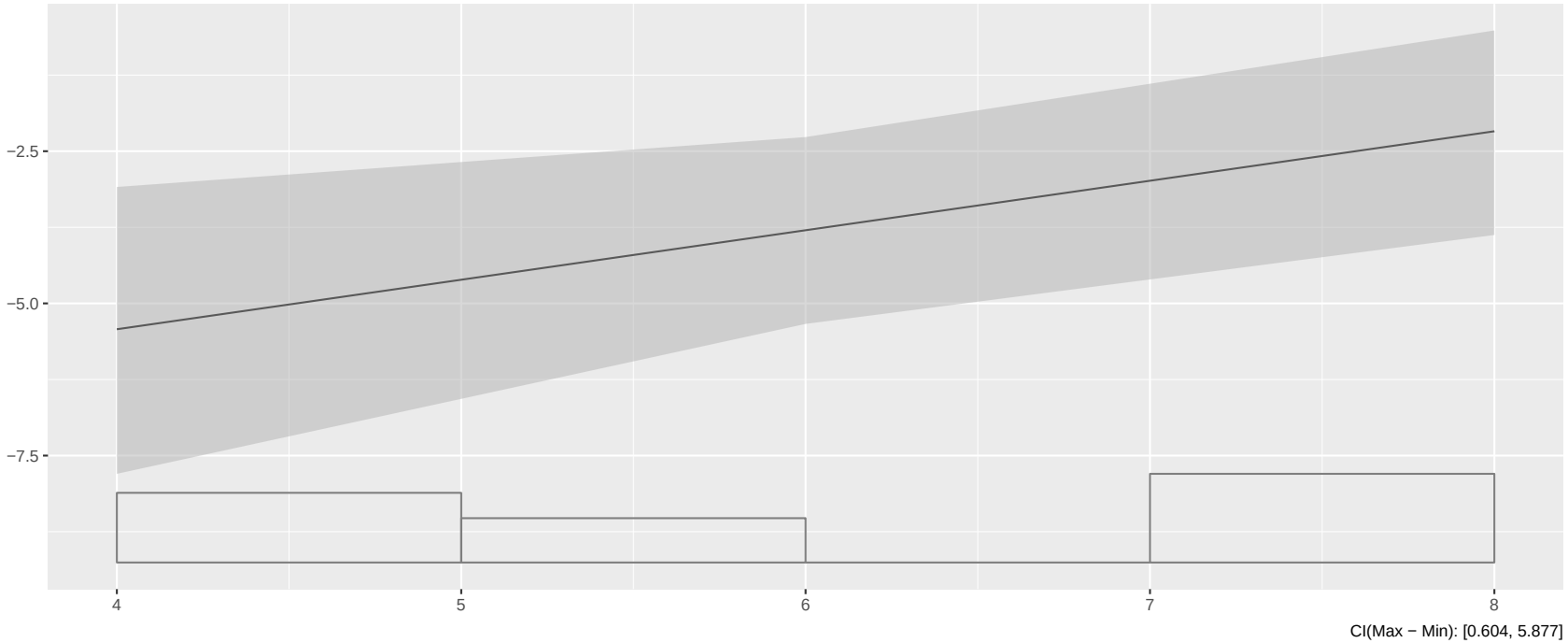
2018-06-30

Interaction is a powerful tool to test conditional effects of one variable on the contribution of another variable to the dependent variable and has been extensively applied in the empirical research of social science since the 1970s (Wright Jr 1976). Unfortunately, the nonlinear nature determines that the statistical estimate of an interactive effect cannot be interpreted as straightforward as the coefficient of a regular regression parameter. Let's use a simple example to illustrate this point: The following model use an interaction term to test the conditional effect of Z on X's contribution (or the conditional effect of X on Z's contribution) to the variance of Y.

$$Y = \beta_0 + \beta_1 X + \beta_2 Z + \beta_3 X \times Z + \varepsilon.$$

- Der Effekt wird auf der y Achse abgetragen - `wt` auf der x-Achse

```
interplot(m = m3a, var1 = "wt", var2 = "cyl", hist = TRUE)
```



Beispiel: Objekt Orientierung

- 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     Horn
##           22.27914           21.46545           26.25203
## Hornet Sportabout            Valiant
##           16.64696           19.59873

##          Mazda RX4      Mazda RX4 Wag          Datsun 710     Horn
##         -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
```

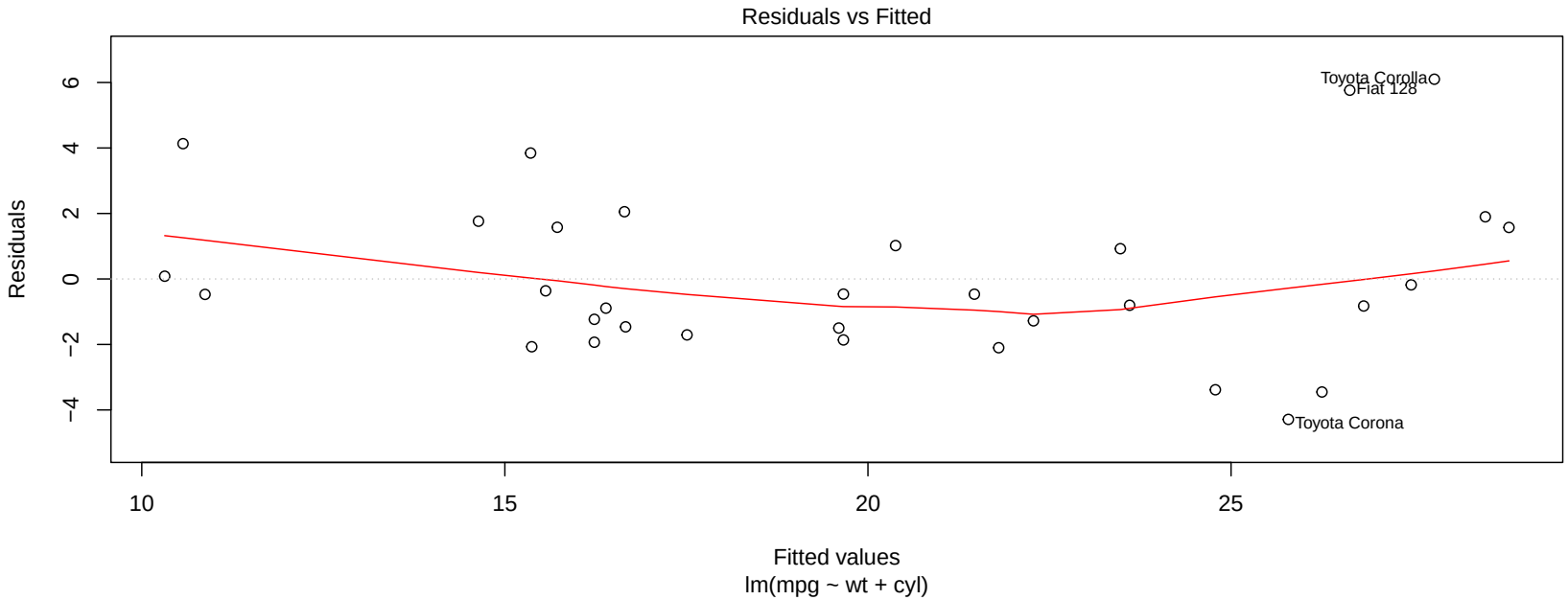
```
head(pre)
```

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

Residuenplot - Modellannahmen verletzt?

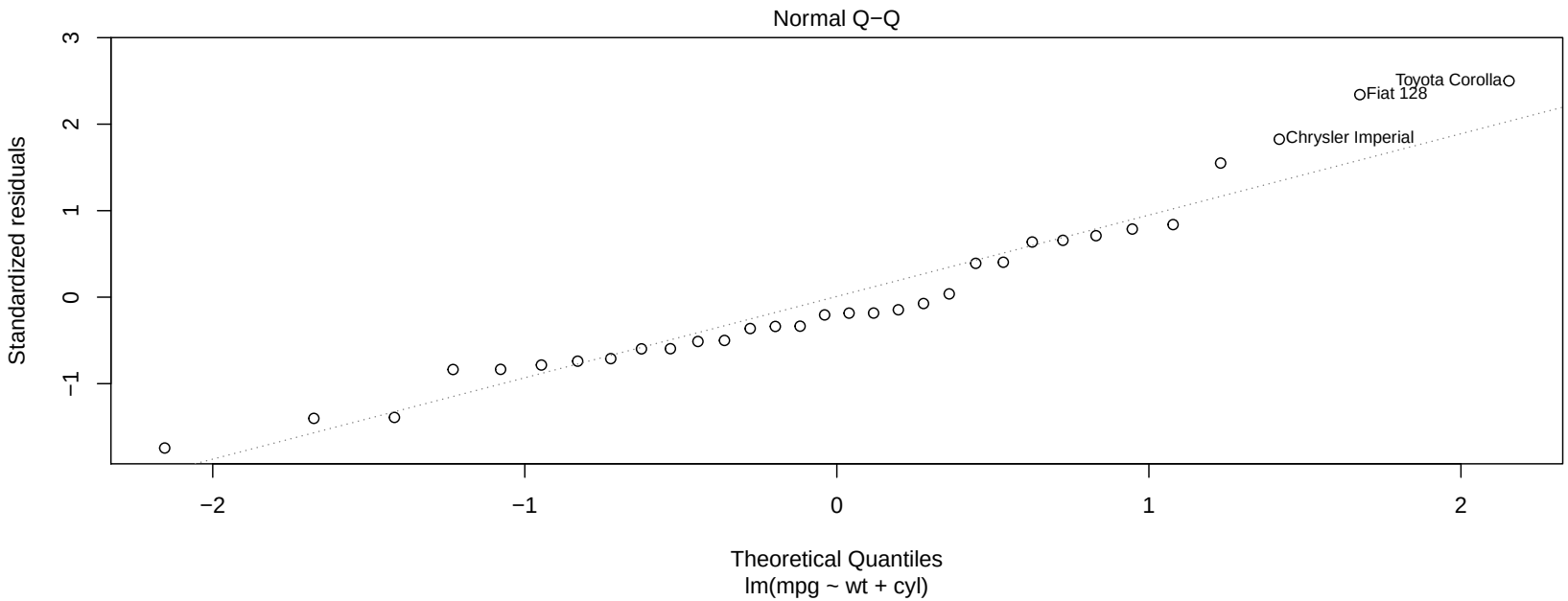
- Gibt es ein Muster in der Abweichung von der Linie

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



Residuenplot

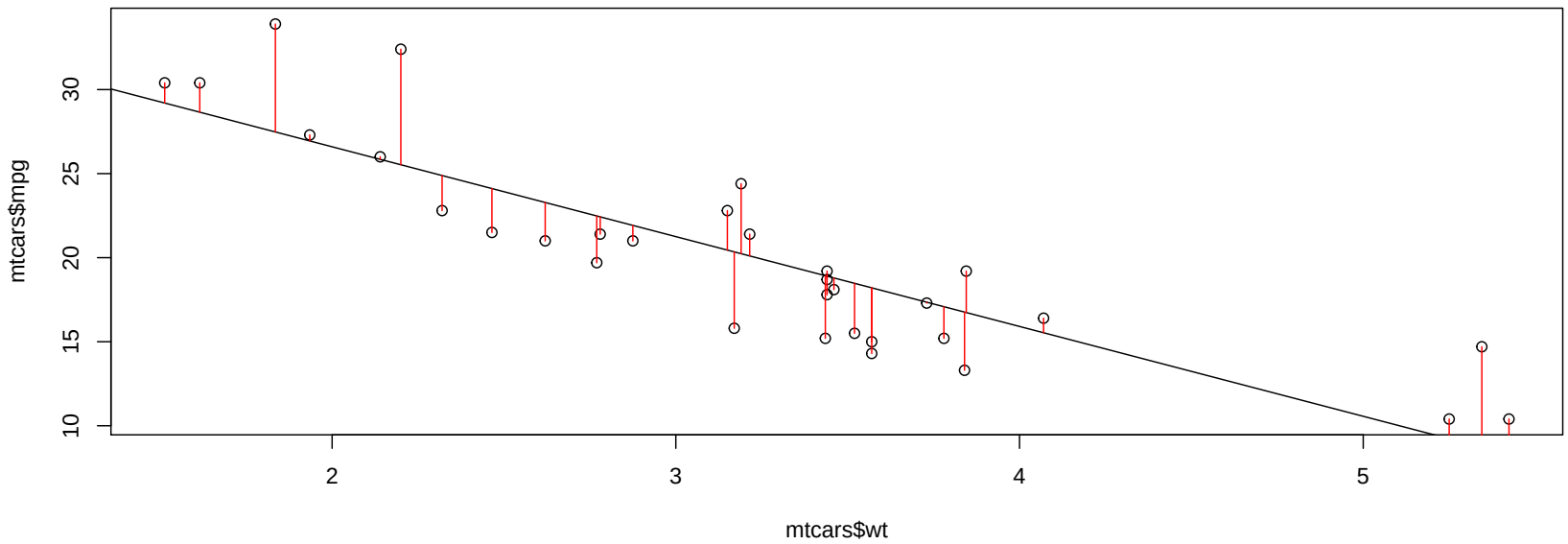
```
plot(m3,2)
```



- Wenn die Residuen normalverteilt sind, dann sollten sie auf der gleichen Linie liegen.

Regressionsdiagnostik mit Basis-R

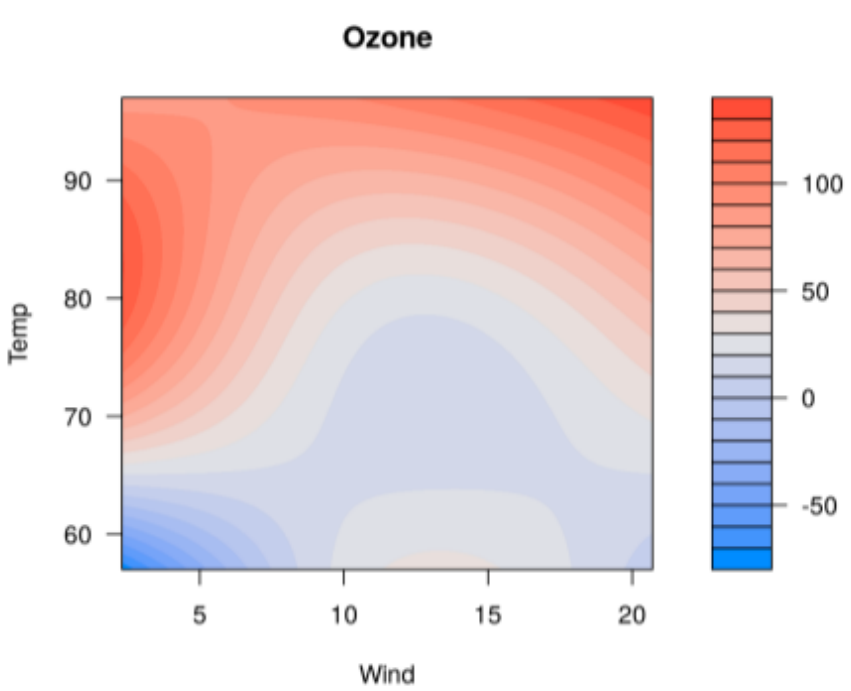
```
plot(mtcars$wt,mtcars$mpg)
abline(m1)
segments(mtcars$wt, mtcars$mpg, mtcars$wt, pre, col="red")
```

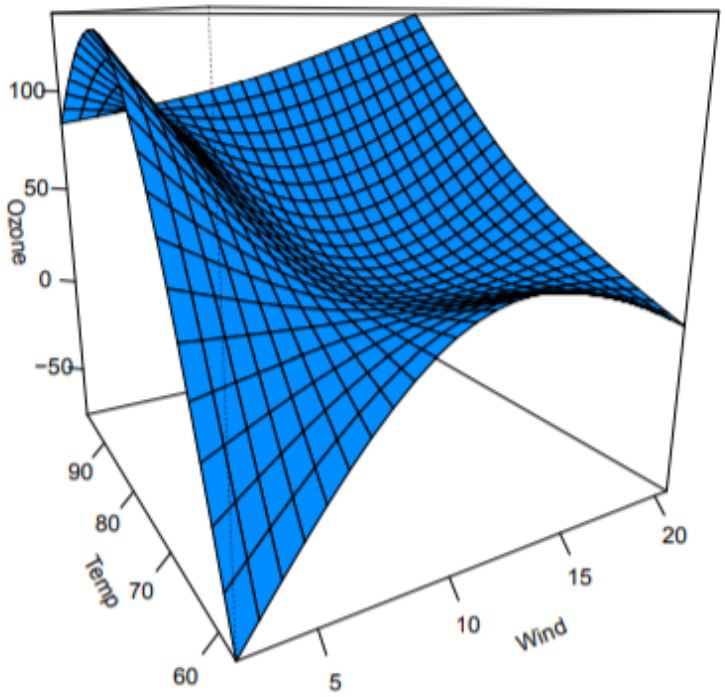


Das visreg-Paket

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

```
library(visreg)
```

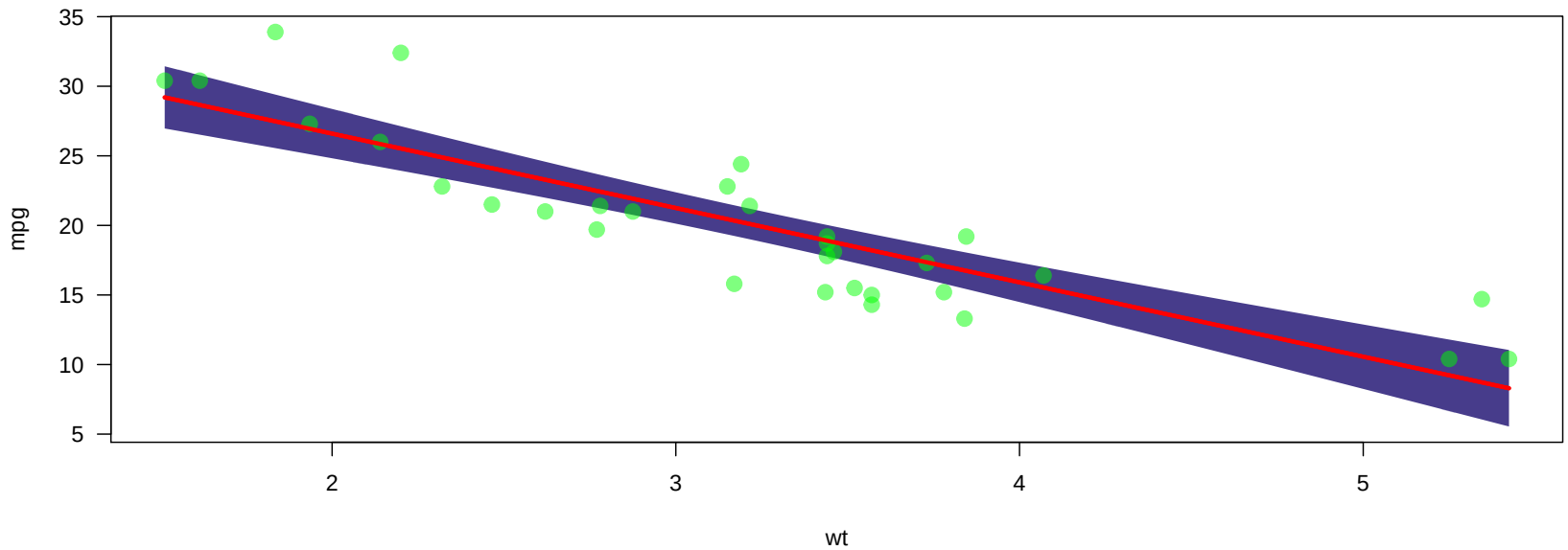




Das visreg-Paket

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

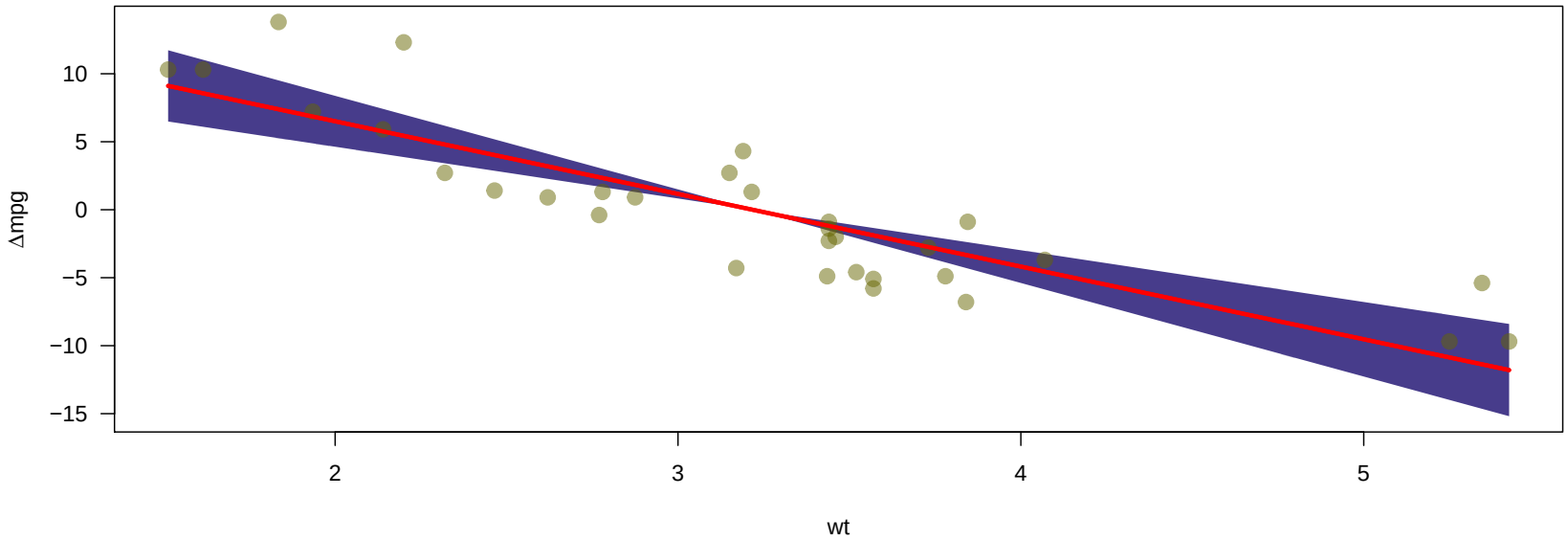
```
visreg(m1, "wt", type = "conditional")
```



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

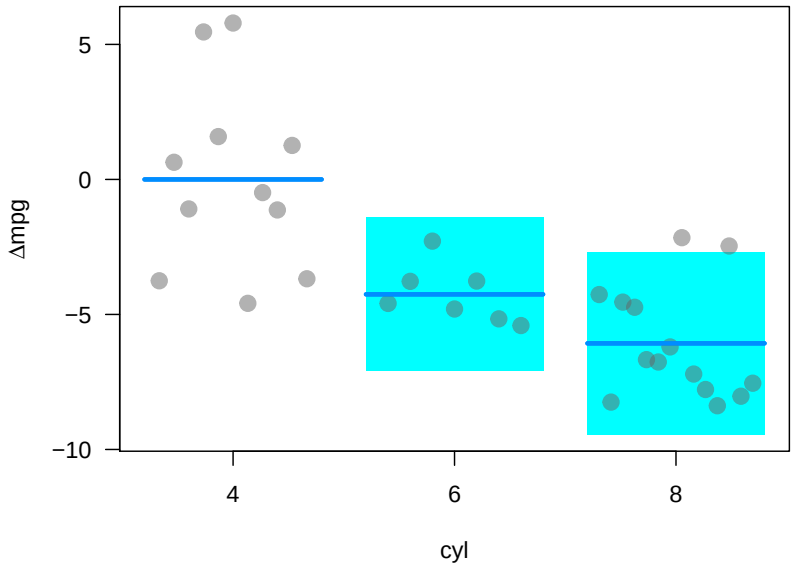
- Die Effekte von Faktoren können auch mit `visreg` visualisiert werden:

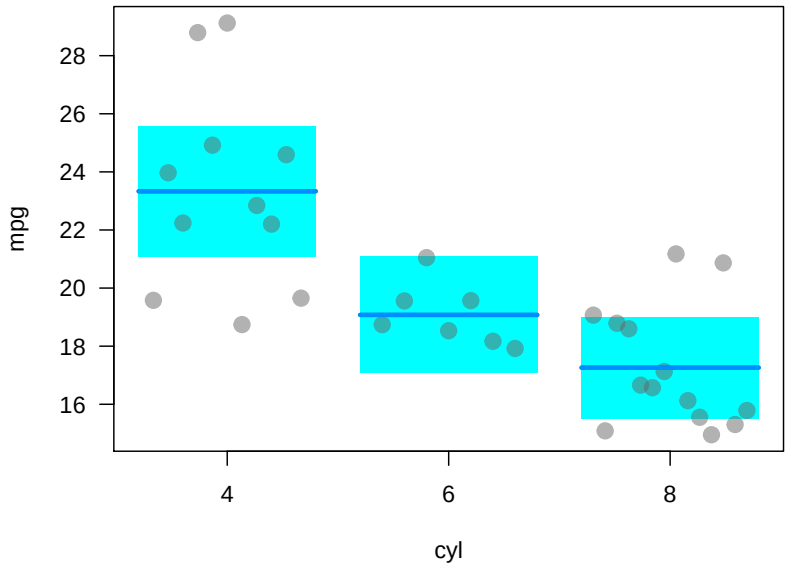
```
mtcars$cyl <- as.factor(mtcars$cyl)
m4 <- lm(mpg ~ cyl + wt, data = mtcars)
# summary(m4)
```

```
##                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")
```





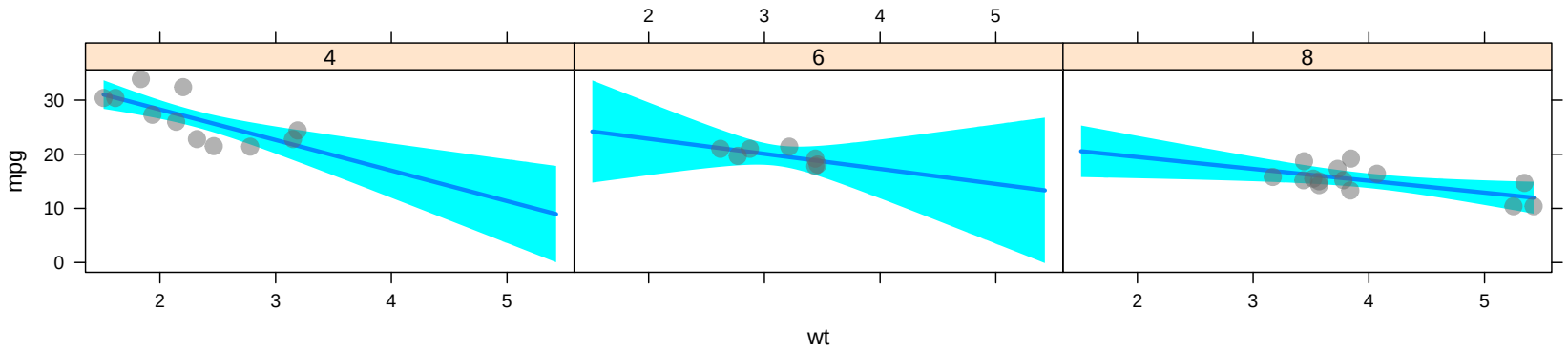
Das Paket visreg - Interaktionen

```
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-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

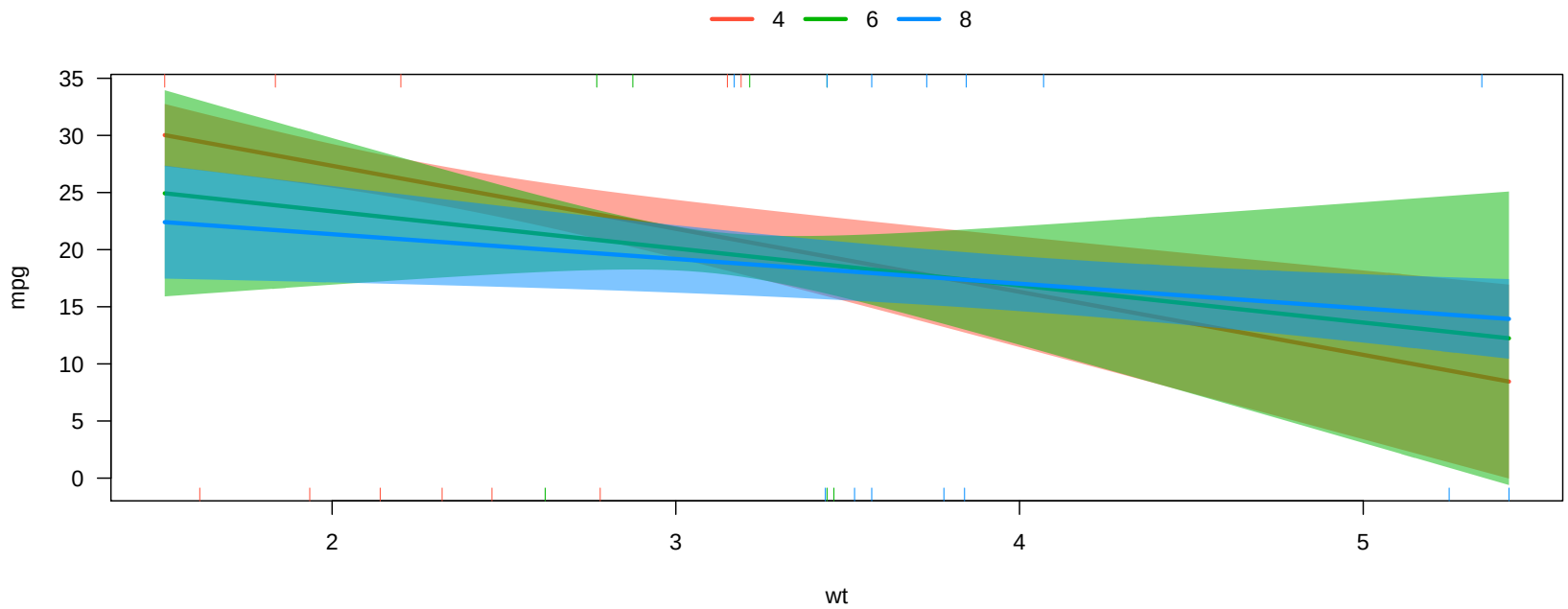
```
visreg(m5, "wt", by = "cyl",layout=c(3,1))
```



Das Paket visreg - Interaktionseffekte übereinander legen

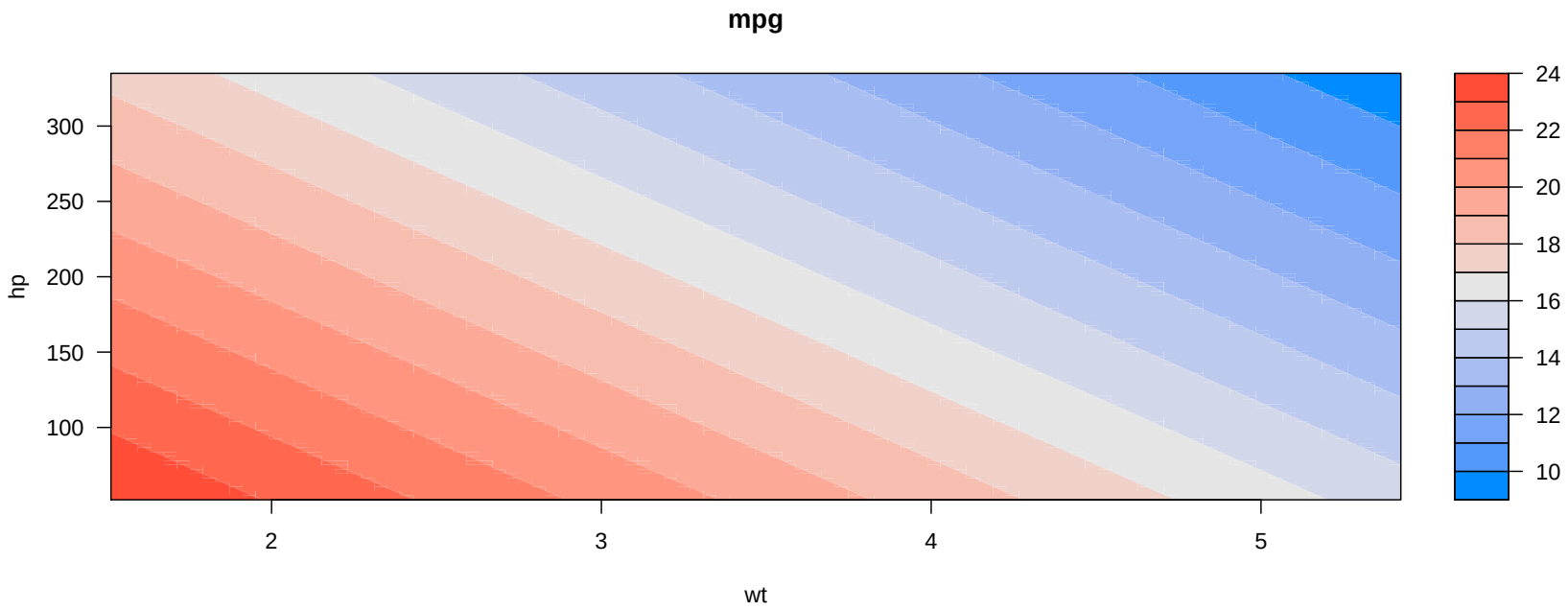
```
m6 <- lm(mpg ~ hp + wt * cyl, data = mtcars)
```

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



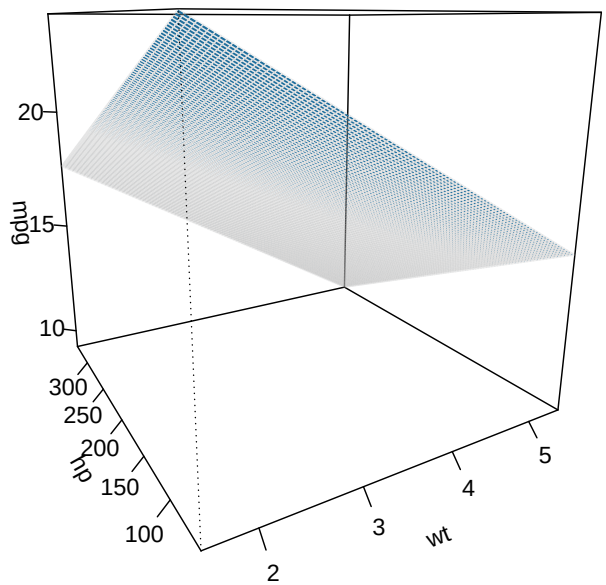
Das Paket visreg - visreg2d

```
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

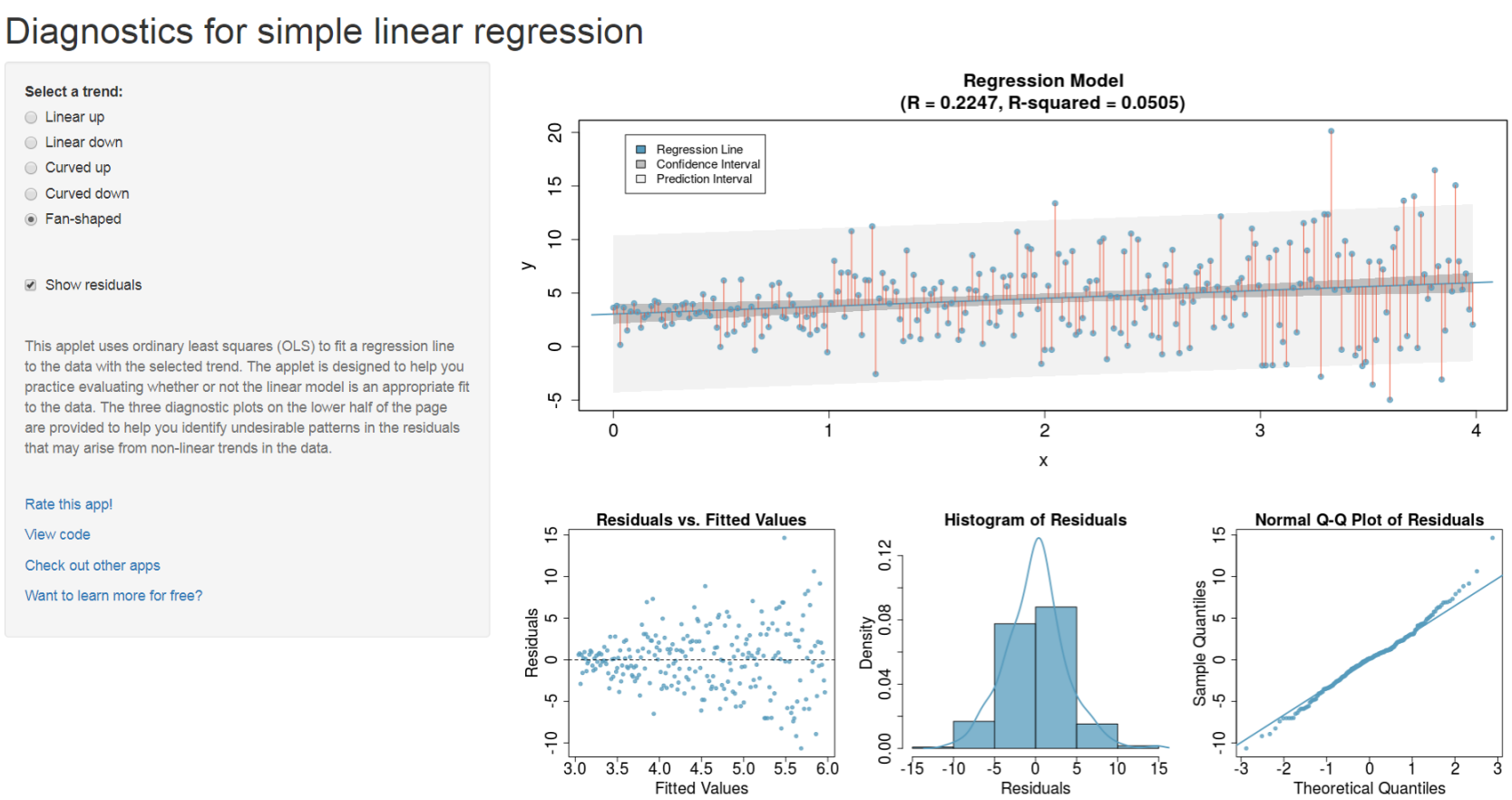
```
library(stargazer)
stargazer(m3, type="html")
```

Beispiel HTML Outputs:

	Dependent variable:
	mpg
wt	-3.125***
	(0.911)
cyl	-1.510***
	(0.422)
am	0.176
	(1.304)
Constant	39.418***
	(2.641)
Observations	32

Shiny App - Diagnostiken für die einfache lineare Regression

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



- Shiny App - **Eine einfache lineare Regression**
- Shiny App - **Multikollinearität in multiplen Regressionen testen**

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**