
# Linear Regression

- to learn more about the relationship between 2 numeric variables

- it is assumed that the Y values are independent
- and can be seen as a linear function to x
- homoscedasticity is met

head(cars)

attach(cars)

- visual impression of our data

plot(x=speed, y=dist)

- the correlation

cor(x=speed, y=dist)  # **positive correlation**

- our linear model

lm(dist~speed)  # **note that first variable here is the Y variable**

- to get a summary

```
summary(lm(dist~speed))
```

- in the field coefficients we can see that the slope for speed is 3.9
- the according p value shows us the slope is not 0 ($p<0.05$)

- to extract the coefficients

```
coef(lm(dist~speed))
```

- to get the conf intervals

```
confint(lm(dist~speed))
```

- to get the anova table

```
anova(lm(dist~speed))
```

- to check for the regression assumptions - preset plots

```
plot(lm(dist~speed))
```

- extrapolation or prediction of the model

```
plot(speed, dist)
```

- we want to know what dist would be at speed 45

- at first we add the speed value

addon = data.frame(speed=45)

- this gives the prediction

predict(lm(dist~speed),addon)

## Multiple Linear Regression

- one numeric Y variable but several x (explanatory) variables

head(mtcars)

attach(mtcars)

- lets create a model for mpg (explained with drat and wt)

mymodel = lm(mpg ~ drat + wt) # **more x variables can be attached by using +**

summary(mymodel)

- R sq tells us that we can explain approx. 76% of the outcome with our model
- the overal p value shows significance of our model

- intercept tells the mean y value when all x are 0

- drat can not be assumed as influencing in this model (p value)
- wt is a significant part of the model with a negative slope of -4.8
- this means: if wt increases, mpg decreases

- shorter output

mymodel

- lets get the pearson correlation

cor(drat, wt) # **we see a negative correlation**

**Confidence interval**

confint(mymodel)

- variables can also be manipulated before feeding into the model
- using the Interpret function : I

lm(mpg ~ drat + I(wt^2)+wt)

**ANOVA table**

anova(lm(mpg ~ drat + I(wt^2)+wt))

- the sum square shows us which variable brings the biggest variance in the model


## Exercise multiple linear Regression

- Dataset = diamonds , library = ggplot2

- fit a model for price explained with depth, x,y,z and check if the variables contribute significantly

- get summary information of the model. How much of the price can you explain?

- get confintervals, correlations and anova tables

**Solution**

```
library(ggplot2)

head(diamonds)

attach(diamonds)


mymodel = lm(price ~ depth + x + y + z)

summary(mymodel) # we can explain approx 78% of the price with our model


confint(mymodel) # to get the confidence intervals


cor(diamonds[,c(5,8:10)]) # to check the cor matrix of the x variables


anova(mymodel) # to check for the variance of the x variables
```

# Logistic Regression

- can be used to predict a binary (2 possible values) outcome variable (probability)
- the explanatory or independent variables can be continuous (numeric)

- in this exercise we want to predict the probability of the outcome am=1

- am = Transmissions is binary

- we are trying to explain am with the variables mpg, drat and wt

- at first we are going to check if all three x variables contribute to our model

```
attach(mtcars)
```

```
glm(data=mtcars, am ~  wt + mpg + drat, family ="binomial")
```

```
summary(glm(data=mtcars, am ~  wt + mpg + drat, family ="binomial"))
```

- mpg and drat do not show significance - therefore we can delete them from our model

```
mylog = glm(data=mtcars, am ~  wt, family ="binomial")
```

```
summary(mylog)
```

- now we see what the model would predict for wt of 4.500

```
addon = data.frame(wt=4.500)
```

```
predict(mylog, addon, type="response")
```

- type response for probabilities

The model would predict that a car of 4500 lb has 0 % probability of having a manual transm

## Exercise - logistic Regression

- get a visual impression of the PlantGrowth df. How does the group influence weight?

- extract Treatment gr 1 and 2 from the PlantGrowth dataframe (do not include the control)

- fit a logistic regression model and check for significance of the variable weight

- add a weight of 7.5 to the dataframe and predict the group of this weight value

**Solution**

```
attach(PlantGrowth)
plot(group,weight)  # gets us a boxplot

mysubset = subset(PlantGrowth, group !="ctrl") # ctrl is omitted

model = glm(data=mysubset, group ~ weight, family="binomial")
summary(model) # fitting the model and checking for significance

addon = data.frame(weight=7.5) # creating the extrapolation to 7.5

predict(model, addon, type="response") # getting the probability
```

We can be 99% sure that if the plant has 7.5 weight units it will be in group 2 = trt2