

# RMarkdown Teaser

Workshop: Einführung in R

28.06.2022

## Markdown

### R Markdown Dokument

Das hier ist ein *R Markdown* Dokument. Es kann R Code ausführen und die Ergebnisse darstellen. Damit lassen sich z.B. automatisierte Berichte erzeugen, die mit neuen Daten vorgefertigte Analysen ausführen.

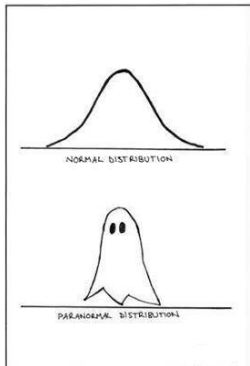
### Formatierung

Mit ein paar einfachen Tricks kann man ein schönes und ev. interaktives **HTML- oder pdf-Dokument** erstellen.

Z.B. mit `# Überschrift` bzw. `## Überschrift` kann man Überschriften in unterschiedlicher Größe erzeugen.

### Bilder

Es lassen sich auch Bilder integrieren.



### Codeblöcke

Man kann Codeblöcke anzeigen:

```
# Das ist ein R Codeblock  
# Der Code hier funktioniert und liefert einen Output  
  
radius <- 3  
flaeche <- radius ^ 2 * pi
```

Ein Kreis mit Radius 3cm hat die Fläche 28.2743339cm<sup>2</sup>.

## Schöne Tabellen

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

## Beispieldatensatz

### Iris Beispieldaten

Der 'iris' Datensatz beinhaltet 150 Messungen der Blattlängen und -breiten von 3 verschiedenen Lilienspezies. Er ist in RStudio integriert und kann direkt verwendet werden.

Sepal und Petal sind verschiedene Blattarten:

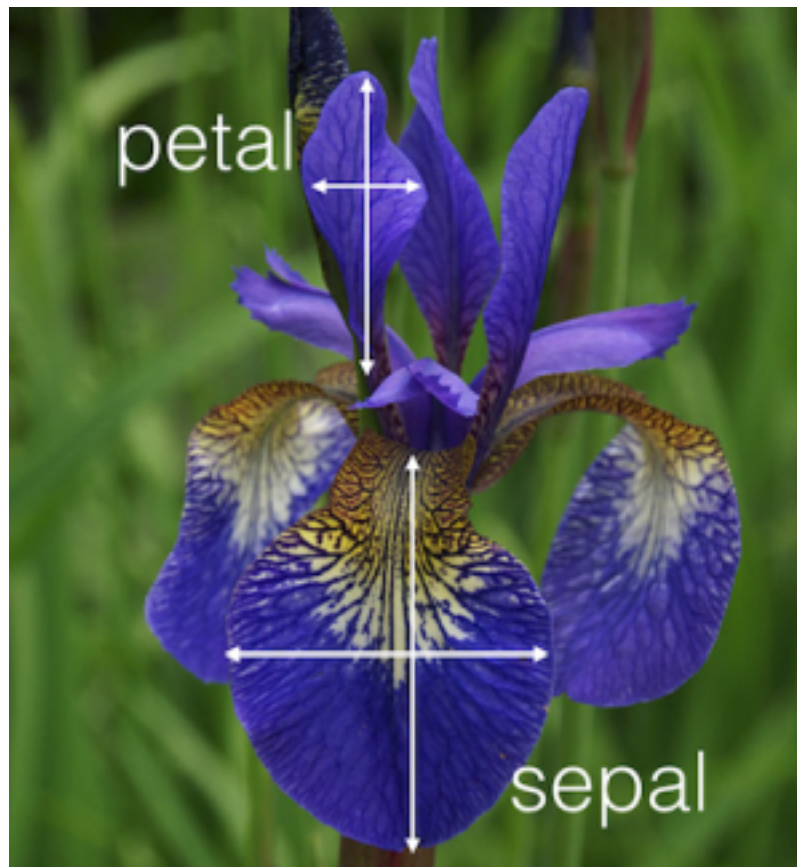


Figure 1: Iris

Ein erster Überblick über die Daten:

```
# head() zeigt die ersten 6 Reihen oder Einzelwerte an:  
head(iris)
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1          3.5          1.4          0.2 setosa
## 2          4.9          3.0          1.4          0.2 setosa
## 3          4.7          3.2          1.3          0.2 setosa
## 4          4.6          3.1          1.5          0.2 setosa
## 5          5.0          3.6          1.4          0.2 setosa
## 6          5.4          3.9          1.7          0.4 setosa
```

```
# summary() berechnet für jede Spalte Minimum, Maximum, Quartile, Mean, Median:
summary(iris)
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width
## Min. :4.300 Min. :2.000 Min. :1.000 Min. :0.100
## 1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300
## Median :5.800 Median :3.000 Median :4.350 Median :1.300
## Mean :5.843 Mean :3.057 Mean :3.758 Mean :1.199
## 3rd Qu.:6.400 3rd Qu.:3.300 3rd Qu.:5.100 3rd Qu.:1.800
## Max. :7.900 Max. :4.400 Max. :6.900 Max. :2.500
## Species
## setosa :50
## versicolor:50
## virginica :50
##
##
##
```

```
# str() zeigt die Datenstruktur der Variablen an
str(iris)
```

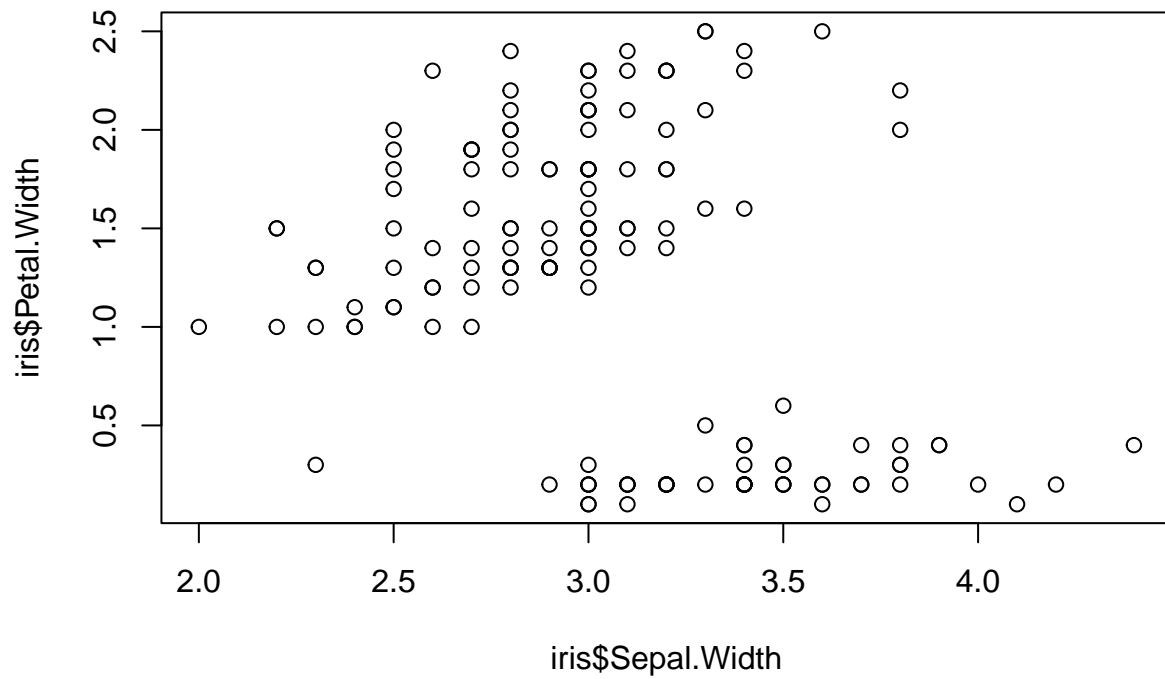
```
## 'data.frame': 150 obs. of 5 variables:
## $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
## $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
## $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
## $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
## $ Species : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
```

## Grafiken

### Grafiken

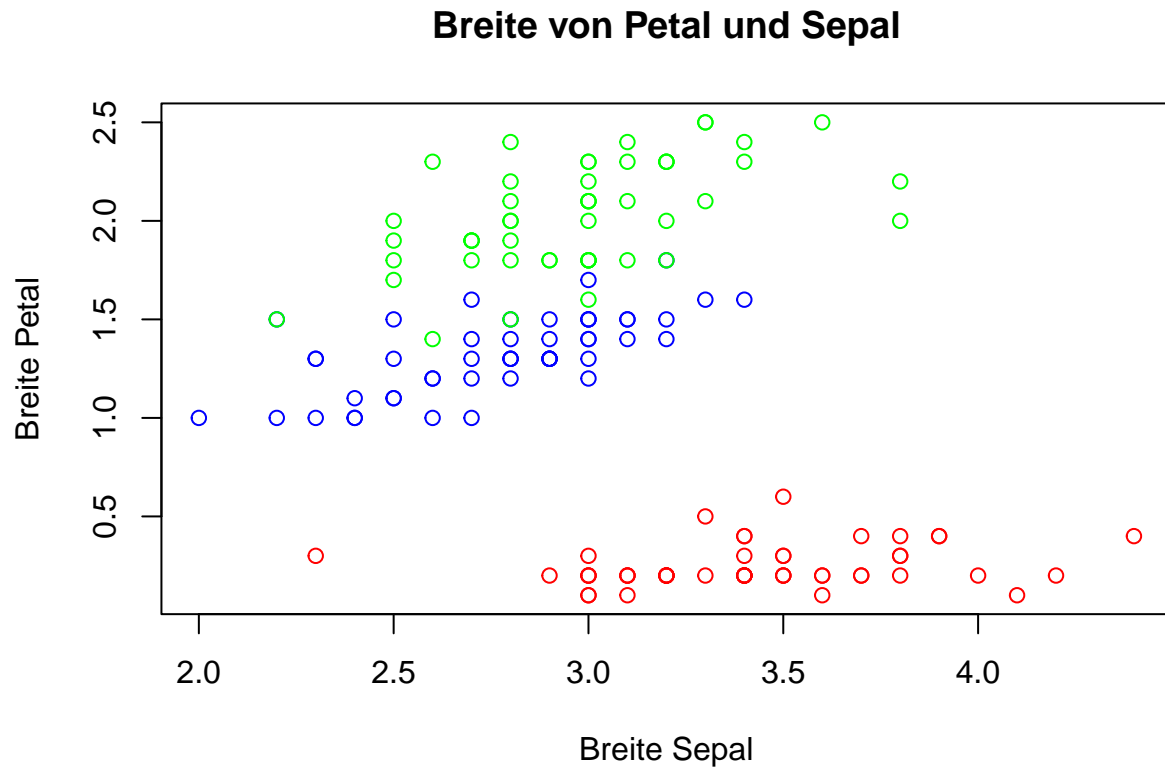
In R kann man schnell und einfach Grafiken erstellen:

```
plot(x=iris$Sepal.Width, y=iris$Petal.Width)
```



Mit zusätzlichen Argumenten in der Funktion kann man die Graphik verschönern und erweitern:

```
plot(x=iris$Sepal.Width, y=iris$Petal.Width,
     col=c("red", "blue", "green")[iris$Species], # nach Spezies einfärben
     xlab="Breite Sepal", ylab="Breite Petal", main="Breite von Petal und Sepal") # Beschriftung
```



## Shiny Apps

### Shiny Apps

Shiny Apps sind browserbasierte Programme mit graphischer Benutzeroberfläche, die sich relativ leicht und schnell erstellen lassen.

Ein Beispiel: <https://shiny.rstudio.com/gallery/kmeans-example.html>.

## Machine Learning

### Machine Learning

Viele Algorithmen für maschinelles Lernen sind in R verfügbar. Z.B. random forest, eine Erweiterung von Decision Trees für Klassifizierung oder Regression.

```
library(randomForest)

rf <- randomForest(Species ~ ., data=iris)
# Das Modell sagt anhand der Maße der Blätter die Spezies voraus
rf

##
## Call:
## randomForest(formula = Species ~ ., data = iris)
##           Type of random forest: classification
##           Number of trees: 500
## No. of variables tried at each split: 2
##
```

```

##          OOB estimate of  error rate: 4%
## Confusion matrix:
##          setosa versicolor virginica class.error
## setosa      50          0          0          0.00
## versicolor   0          47          3          0.06
## virginica    0          3          47          0.06
predict(rf, iris[75, 1:4]) # Vorhersage für Zeile 75 anhand des Modells

##          75
## versicolor
## Levels: setosa versicolor virginica
iris[75, 5] # Wahre Klasse in Zeile 75

## [1] versicolor
## Levels: setosa versicolor virginica

```