



Einführung in R

Prof. Dr. Stephan Trahasch
Hochschule Offenburg

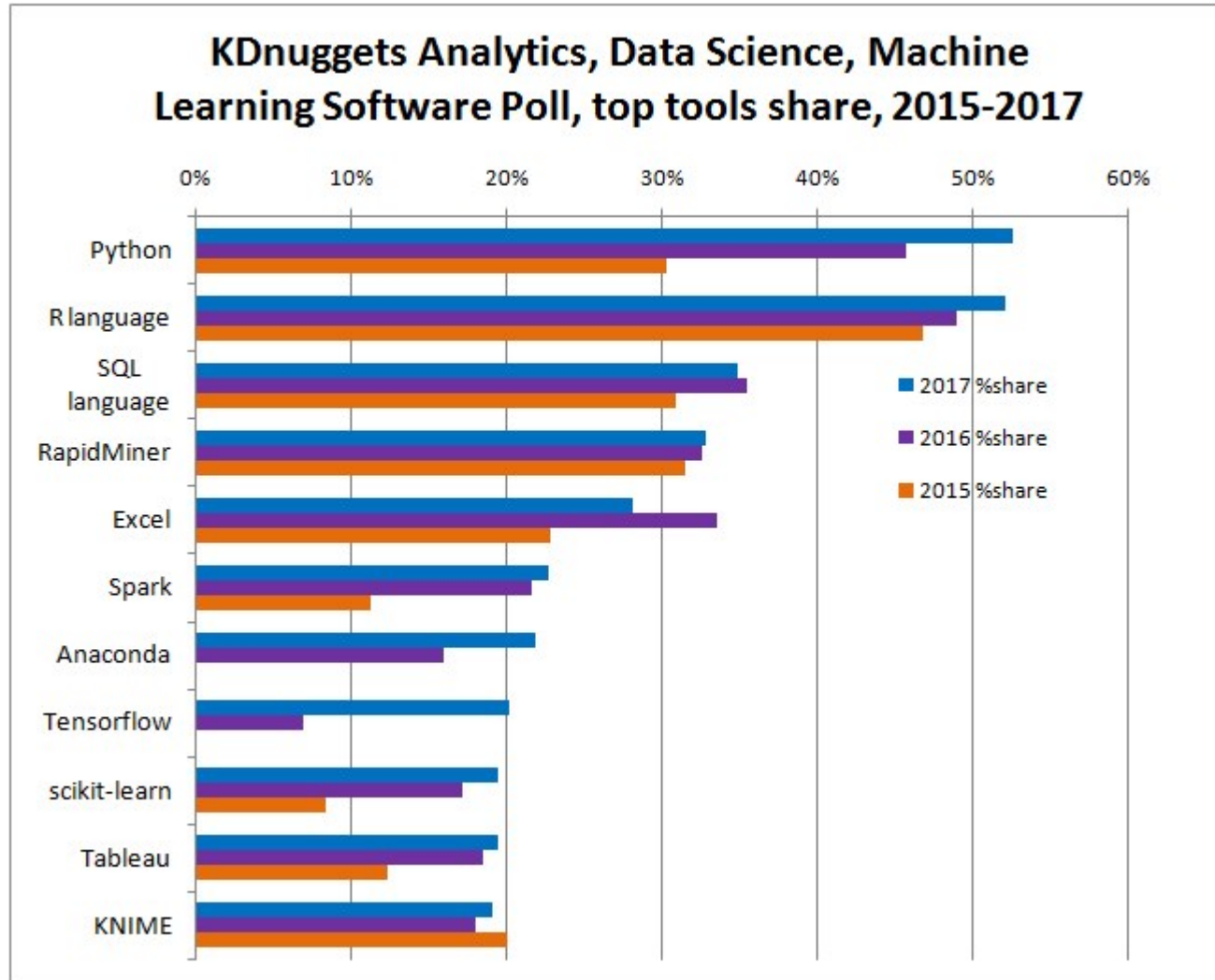
Links

- <https://www.rstudio.com/resources/cheatsheets/>
- **R for Data Science**
<http://r4ds.had.co.nz/>
- <https://www.tidyverse.org/>
- Free course **Introduction to R**
<https://www.datacamp.com/courses/free-introduction-to-r>
- <https://www.r-bloggers.com/>

Übersicht

- Was ist R?
- Ecosystem
- Shiny
- R in Produktion
- Syntax
- ggplot2

Survey 2015-2017 von kdnuggets.com



<https://www.kdnuggets.com/2017/05/poll-analytics-data-science-machine-learning-software-leaders.html>

Geschichte: S, S-Plus und R

- 1977 wurde am Bell Labs Sprache S zur Organisation, Analyse und Visualisierung von Daten entwickelt. <http://ect.bell-labs.com/sl/S/>
“In 1998, S became the first statistical system to receive the Software System Award, the top software award from the ACM.”
- S-PLUS kommerzielle Implementierung von S
- R Open Source, die auf S aufbaut, unabhängig von S weiter entwickelt wird. Ross Ihaka und Robert Gentleman implementierten erste R-Version an University of Auckland.
- R Foundation fördert „das ‘R Project for Statistical Computing’ um eine freie Open Source Softwareumgebung für Datenanalyse und Graphik zur Verfügung zu stellen.“
<http://www.R-project.org>
- R Consortium fördert Entwicklung von R, um R im Unternehmensumfeld komfortabler einzusetzen.
<https://www.r-consortium.org/>



R Ecosystem



R Core Group

 #rstats

CRAN

R-bloggers



R User Groups



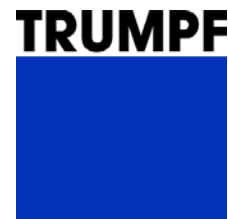
consortium



Vorteile von R

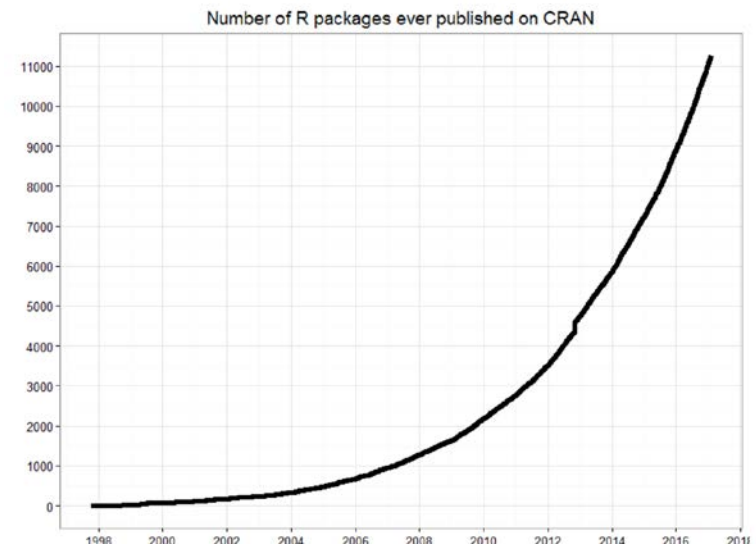
- Domain Specific Language zur Datenanalyse und Visualisierung
→ R for Reproducible Scientific Analysis
- Open Source (GNU GPL)
- Verfügbar für (fast) alle Plattformen: Windows, Mac OS, Linux, Solaris, ...
- Sehr große User Community
- Große Anzahl an Packages (> 10.000) in einer sehr guten Qualität!!!
- Integration in viele Tools und Sprachen
wie Microsoft SQL, Power BI, KINIME, RapidMiner, SAS etc. und Bindings/Interfaces für Programmiersprachen

R wird in vielen Unternehmen produktiv eingesetzt



CRAN - Comprehensive R Archive Network

- Funktionsumfang kann durch eine Vielzahl von zusätzlichen Paketen erweitert werden.
- Zentrales Archiv ist Comprehensive R Archive Network (CRAN) mit zahlreichen Spiegelservers wie <https://stat.ethz.ch/CRAN/>
- <https://mran.microsoft.com/> Daily Snapshots
- 12.323 Packages sind verfügbar (Stand 18.3.2018)
- Weitere Pakete bei Bioconductor (ca. 3.000) und GitHub
- Pakete bei CRAN und Bioconductor haben eine hohe Qualität und sind dokumentiert.
- R Manuals auch auf CRAN



Entwicklungstools

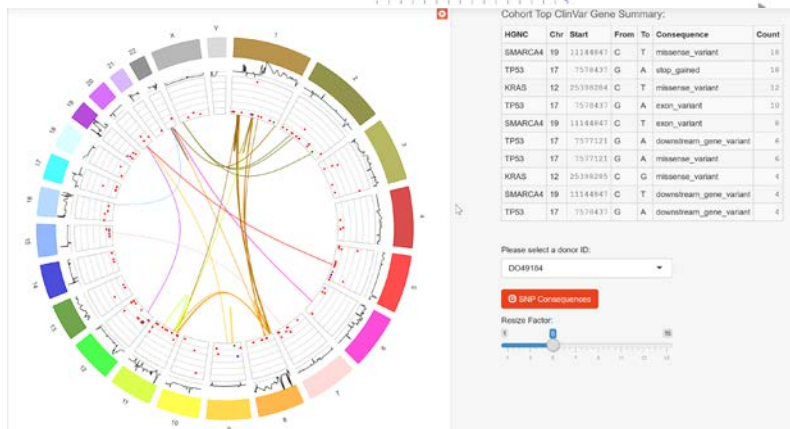
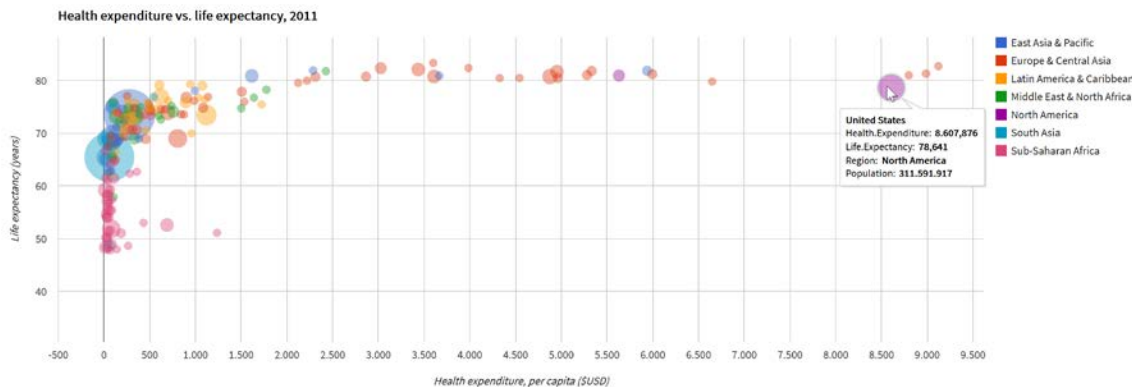


git



R im Web mit Shiny

<https://shiny.rstudio.com/>



Spalten verbinden

Zeitstempel definieren

Operation wählen

Umwandeln von Unixtime

Spalte wählen:

time2

Zeitzone wählen:

Etc/GMT-0

Ändern

Einfache Bereinigung

Show 10 entries

ZeilenNr.	time1	time2	strdate
1	4.9.2017, 03:00:00	1504486800	2017/04/07
2	4.9.2017, 04:00:00	1504490400	2017/04/07
3	4.9.2017, 05:00:00	1504494000	2017/04/07
4	4.9.2017, 06:00:00	1504497600	2017/04/07
5	4.9.2017, 07:00:00	1504501200	2017/04/07
6	4.9.2017, 08:00:00	1504504800	2017/04/07
7	4.9.2017, 09:00:00	1504508400	2017/04/07
8	4.9.2017, 10:00:00	1504512000	2017/04/07
9	4.9.2017, 11:00:00	1504515600	2017/04/07
10	4.9.2017, 12:00:00	1504519200	2017/04/07

Datentyp: Datum/Zeit

Ganzzahl

Zeichenket

Showing 1 to 10 of 18 entries

Previous 1 2 Next

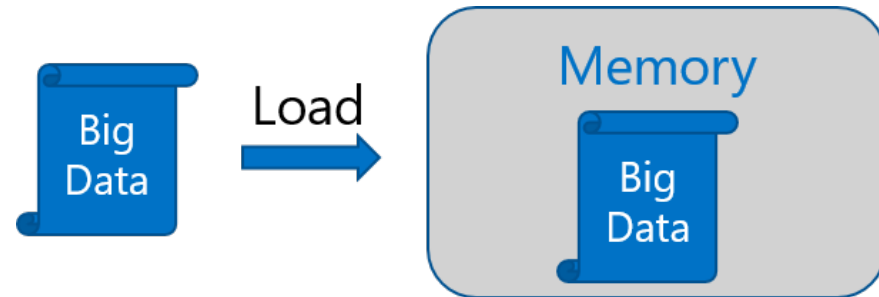
ShinyProxy

- Deploying Shiny Apps
- LDAP Authentication und Authorization
- Skalierbar
- <https://www.shinyproxy.io>

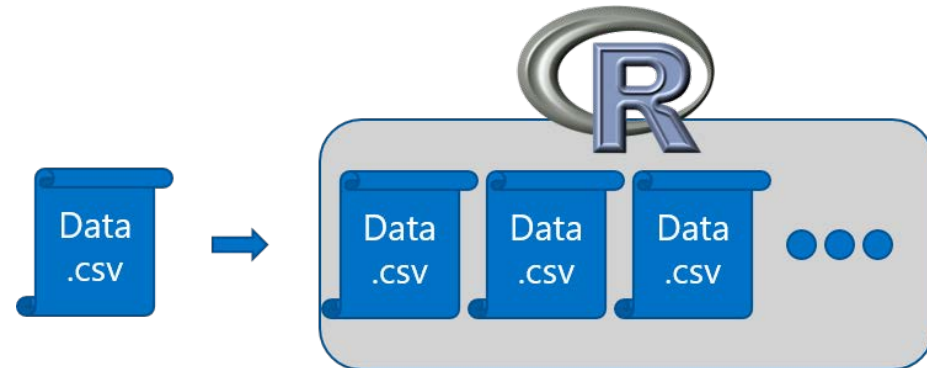


Verarbeitung großer Datenmengen mit R

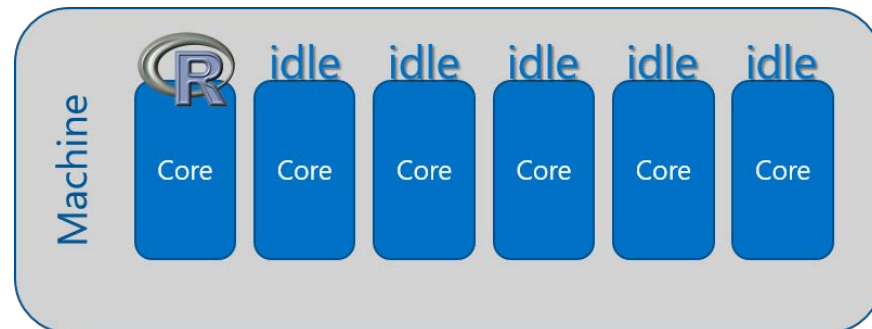
In-Memory Operation



Expensive Data Movement
& Duplication



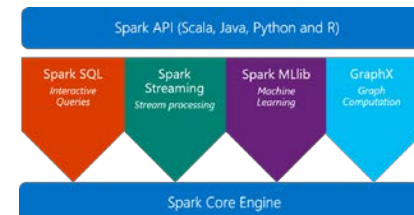
Lack of Parallelism



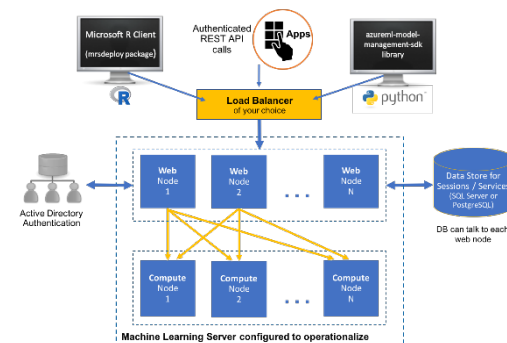
Verschiedene Skalierungsmöglichkeiten

Scale-Out: Verwende mehrere Server (Distributed Computing)

- Apache Spark mit SparkR oder sparklyr



- Microsoft R Server jetzt unter dem Name Microsoft Machine Learning Server



- R on Docker → Rocker

R Syntax

```

1  ##### Decision Trees
2
3  # Libraries -----
4  library(rpart)          # Popular decision tree algorithm
5  library(rattle)         # Fancy tree plot
6  library(RColorBrewer)   # Color selection for fancy tree plot
7  library(party)          # Alternative decision tree algorithm
8  library(caret)          # Just a data source for this script but also a very import
9
10 # help rpart
11 ?rpart
12
13 # Example Iris data -----
14
15 # The famous Fisher Iris dataset is included in R
16 # but you should import it from UCI
17 # http://archive.ics.uci.edu/ml/machine-learning-databases/iris/
18 # t.url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.dat
19 # iris <- read.csv(t.url, header = FALSE, sep = ",", quote = "\"", dec = ".")
20 # colnames(iris) <- c("Sepal.L", "Sepal.W", "Petal.L", "Petal.W", "Class")
21
22 ?iris
23 data(iris)
24 iris
25
26 formula <- Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
27 # alternative way for formula. Dot . means all features
28 # formula <- Species ~ .
29
30 tree <- rpart(formula, data=iris, method="class", control=rpart.control(cp=0.0, mir
31
32

```

Libraries

- Example Iris data
- Example breast cancer data
- Big tree

R als Taschenrechner

$+$, $-$, $*$, $/$, $\sin(x)$, \sqrt{x} , $\exp(x)$, ...

```
3.5 + 1.5  
[1] 5
```

```
x <- 6 * 1/3      # Zuweisung  
X                # empfohlen  
[1] 2
```

```
x = 2^2  
print(x)  
[1] 4
```



Einfache Datentypen

Data type	Example
Integer	1
Logical	TRUE
Numeric	1.1
String / character	"Red"
Factor (enumerated string)	"Amber" or 2 in c ("Red", "Amber", "Green")
Complex	i
Date	2015-04-24
NA	NA

Vergleiche

Comparison of numerics

`-6 * 5 + 2 >= 10 + 1`

[1] FALSE

Comparison of numerics

`6 * 15 != 17 - 101`

[1] TRUE

Comparison of character strings

`"useR" == "user"`

[1] FALSE

Comparison of character strings

`"raining" <= "raining dogs"`

[1] TRUE

Compare a logical with a numeric

`TRUE == 1`

[1] TRUE

Comparison of logicals

`TRUE > FALSE`

[1] TRUE

Kontrollstrukturen

```
# Variables
medium <- "LinkedIn"
num_views <- 14

# Control structure for medium
if (medium == "LinkedIn") {
  print("Showing LinkedIn information")
} else if (medium == "Twitter") {
  # do something
} else {
  print("Unknown medium")
}

# Control structure for num_views
if (num_views > 15) {
  print("More than 15 views")
} else if (num_views <= 15 & num_views > 10) {
  # do something
} else {
  print("Nothing do show.")
}
```

Vektoren

Vektorisiert (empfohlen)

```
a <- seq(from = 1, to = 3, by = 1) # entspricht c(1,2,3)
b <- 9:7 # entspricht c(9, 8, 7)

a
[1] 1 2 3
b
[1] 9 8 7
```

Manuell

```
c <- c(0, 0, 0)
for (i in 1:length(a)) {
  c[i] <- a[i] + b[i]
}

c
[1] 10 10 10
```

```
c <- a + b

c
[1] 10 10 10
```

Vektoren und Funktionen

Funktionen werden auf jedes Element eines Vektors angewandt.

```
a <- 1:4  
sqrt(a) # square root
```

```
[1] 1.000000 1.414214 1.732051 2.000000
```

```
max(a^2) # biggest element
```

```
[1] 16
```

```
sum(a^2) # sum of all elements
```

```
[1] 30
```

Data Frame

- Liste aus Vektoren gleicher Länge (=Spalten), die Namen haben
- Wichtigste Datenstruktur

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	carname
1	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
2	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
3	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
4	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
5	17.0	8	302.0	140	3449	10.5	70	1	ford torino
6	15.0	8	429.0	198	4341	10.0	70	1	ford galaxie 500
7	14.0	8	454.0	220	4354	9.0	70	1	chevrolet impala
8	14.0	8	440.0	215	4312	8.5	70	1	plymouth fury iii
9	14.0	8	455.0	225	4425	10.0	70	1	pontiac catalina
10	15.0	8	390.0	190	3850	8.5	70	1	amc ambassador dpl

- Zwei Indices: `df[Zeile(n) , Spalte(n)]`

Data Frame: Zugriff auf Inhalte

Zugriff auch über Spaltenname möglich

```
> df[1, ]      # erste Zeile, alle Spalten
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	carname
1	18	8	307	130	3504	12	70	1	chevrolet chevelle malibu

```
> df[,2]      # alle Zeile, zweite Spalte
```

```
[1] 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 4 6 6 6 4 4 4 4 4 4 6 8 8 8 8 4 4 4 6 6 6 6 6 8 8 8 8 8 8 8 8 6
[55] 4 4 4 4 4 4 4 8 8 8 8 8 8 8 8 8 3 8 8 8 8 4 4 4 4 4 4 4 4 4 8 8 8 8 8 8 8 8 8 8 8 8 8 8 6 6 6
[109] 4 4 3 4 6 4 8 8 4 4 4 4 8 4 6 8 6 6 6 6 4 4 4 4 6 6 6 8 8 8 8 8 4 4 4 4 4 4 4 4 4 4 4 4 6 6 6
```

```
> df$cylinder  # alle Zeile, Merkmal cylinder
```

```
> df[1,3]      # erste Zeile, dritte Spalten
```

```
[1] 307
```

Zugriff auf Elemente

```
x <- 1:5
```

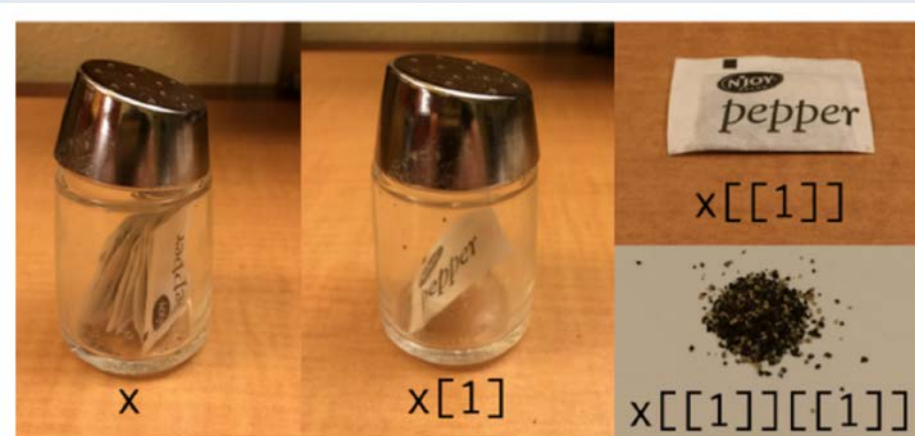
```
x[3]
```

```
## [1] 3
```

```
x[3] <- 42
```

```
x
```

```
## [1] 1 2 42 4 5
```



From [@hadleywickham](#)

Packages

Vor der ersten Nutzung und nach R-Updates müssen Packages einmalig installiert werden.

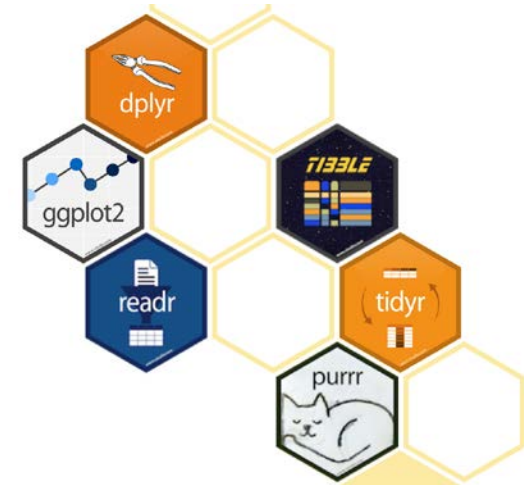
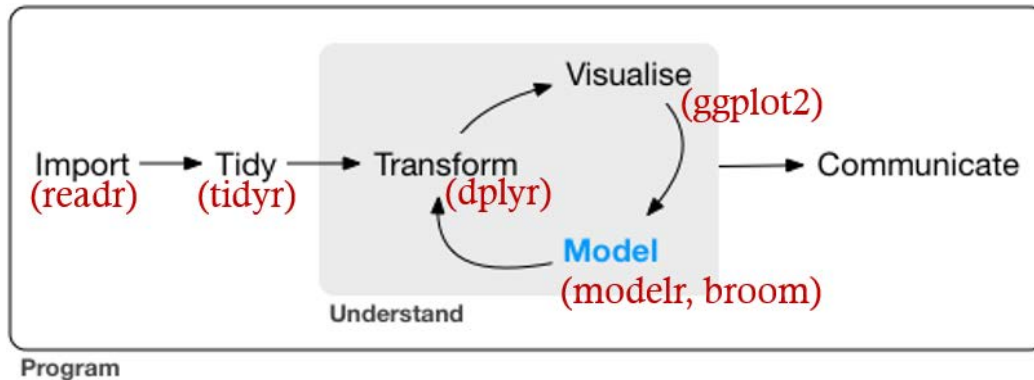
Im Code müssen die Packages in die Session geladen werden.

```
# Get a package  
install.packages("caret")  
  
# Activate a package  
library(caret)
```

Learning R the Tidyverse

Tidyverse ist eine Sammlung von R Packages, die alle nach gleichen Grundprinzipien arbeiten.

<https://www.tidyverse.org/>



Um alle Packages zu nutzen, genügt:
`install.packages("tidyverse")`
`library(tidyverse)`

Pipe-Operator

%>%

- Weitergabe des Ergebnisses der ersten Funktion als Input für die nächste Funktion “... und dann ...”
- Statt `f2(f1(x))` schreibt man `x %>% f1 %>% f2`

```
finally_last_step(  
  and_then_third(  
    then_second(  
      do_first(data)  
    )  
  )  
)
```

```
data %>%  
  do_first() %>%  
  then_second() %>%  
  and_then_third() %>%  
  finally_last_step()
```

Beispiel

```
data(iris)

iris[, 1:4] %>%      # select first 4 columns
  head() %>%         # show 1st 6 rows of 4 columns
  rowSums()          # row sums of 6 rows of 4 cols

#>      1      2      3      4      5      6
#> 10.2   9.5   9.4   9.4 10.2 11.4
```

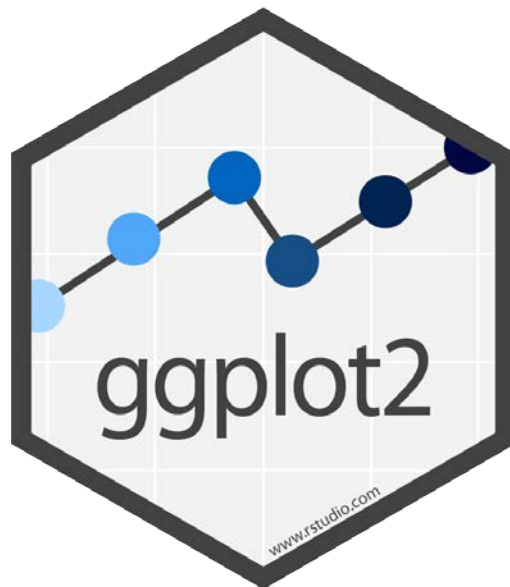
RStudio

The screenshot shows the RStudio IDE interface. The main window is divided into several panes:

- Editor:** The central pane for writing R code. It contains a script titled "110-Solution-Basic-Data-Analysis.R" with the following code:


```
1 # Exercise: Introduction to Data Analysis
2
3 # Importing Data -----
4 # Import data from https://archive.ics.uci.edu/ml/datasets/Auto+MPG
5 auto <- read.table("https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data",
6                   header=FALSE, dec=".", na.strings = "?")
7 colNames(auto) <- c("mpg", "cylinders", "weight", "acceleration", "name")
8 str(auto)
9
10
11
12 # Basic Analysis -----
13
14 # remove NA
15 auto <- na.omit(auto)
16 auto$horsepower
17
18 # numerical data
19 mean(auto$horsepower)
20 median(auto$horsepower)
21 sd(auto$horsepower)
22 var(auto$horsepower)
23 min(auto$horsepower)
24 max(auto$horsepower)
25 range(auto$horsepower)
26 quantile(auto$horsepower)
27 # all together
28 summary(auto$horsepower)
29 summary(auto)
30
31 # correlation
32 cor(auto[,c(1,4,5)])
```
- Environment:** The top-right pane showing the current environment. It displays "Global Environment" and "Data" with a table named "auto" containing 392 observations and 9 variables. A yellow callout box labeled "Umgebung, Git" is overlaid on this pane.
- R Console:** The bottom-left pane showing the output of R commands. It displays the R version (3.4.4), copyright information, and a list of contributors. A yellow callout box labeled "R Console" is overlaid on this pane.
- Files:** The bottom-right pane showing a file explorer view of the project directory. It lists files such as "Tutorial.Rproj", "101-Basic-Introduction.R", ".gitignore", ".Rhistory", "Dataset description.txt", "README.md", ".RData", "110-Exercise-Basic-Data-Analysis.R", "110-Basic-Data-Analysis.R", "110-Solution-Basic-Data-Analysis.R", "Apriori", "Cheatsheets", "Classification", "Clustering", "Cross-Valid", "data", "Deep-Learn", "Generalized Linear Models GLM", and "Introduction". A yellow callout box labeled "Files, Plots, Help ..." is overlaid on this pane.

Visualize Data with

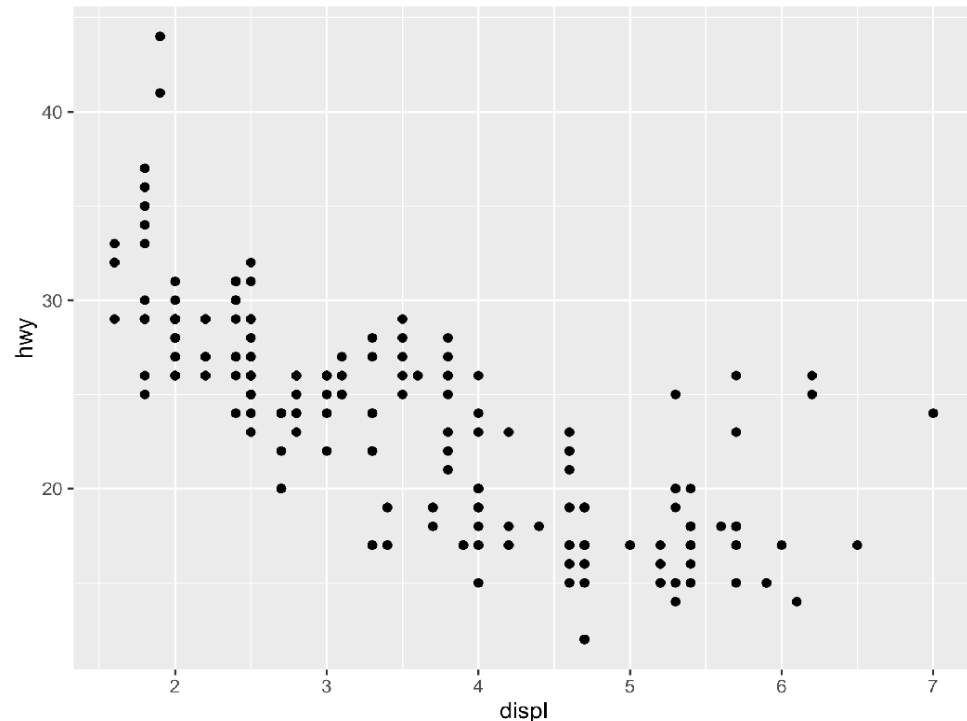


"The simple graph has brought more information to the data analyst's mind than any other device. "

- John Tukey

Vorgehen

1. Initialisiere einen Plot mit `ggplot()`
2. Füge Layers mit `geom_X` Funktionen hinzu.



```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```

Syntax

data

+ before new line

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```

type of layer

aes()

x variable

y variable

```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

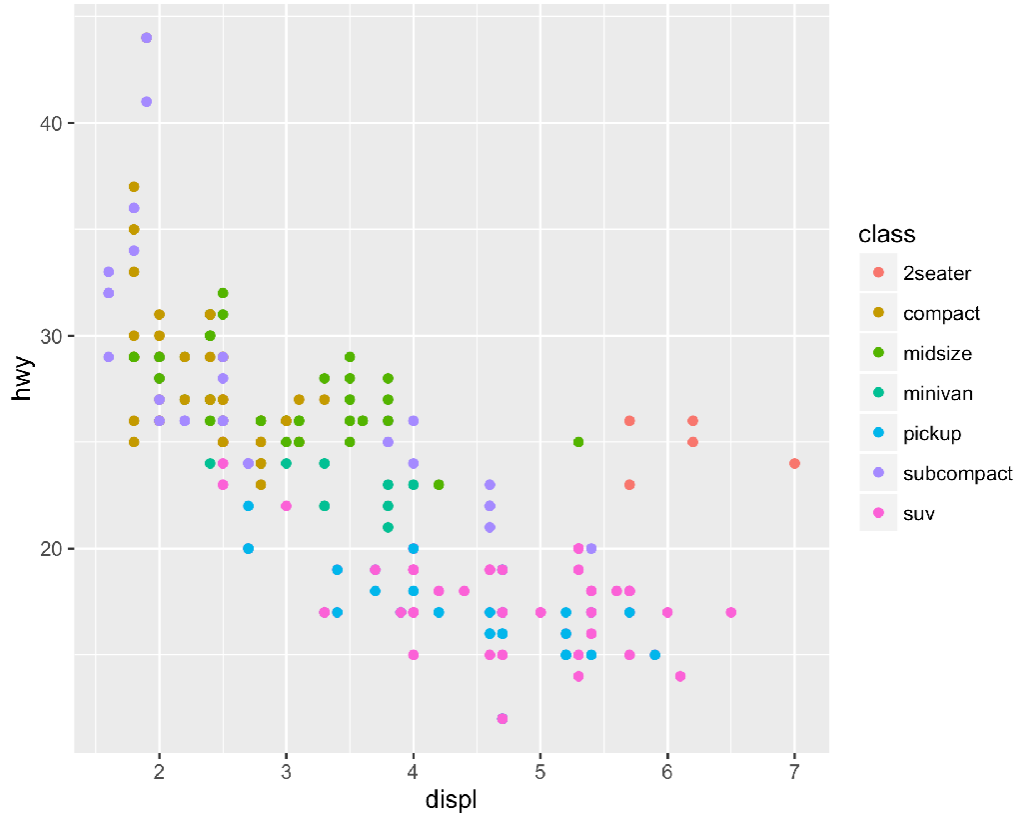

Aesthetics

aesthetic
property

Variable to
map it to

```
ggplot(mpg) +geom_point(aes(x = displ, y = hwy, color = class))
ggplot(mpg) +geom_point(aes(x = displ, y = hwy, size = class))
ggplot(mpg) +geom_point(aes(x = displ, y = hwy, shape = class))
ggplot(mpg) +geom_point(aes(x = displ, y = hwy, alpha = class))
```

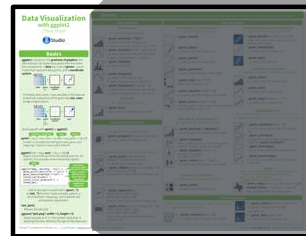
Legend added automatically



Inside of aes():
maps an aesthetic
to a variable

```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, color = class))
```

geom_ functions

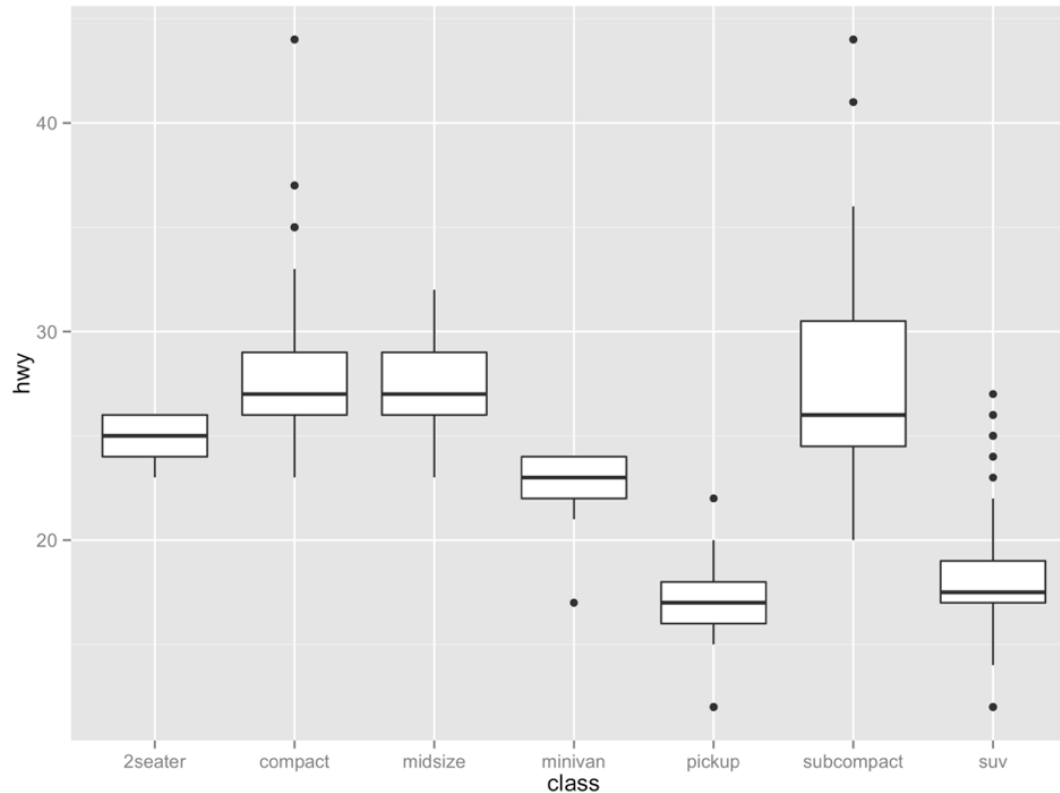


Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

One Variable	Two Variables
Continuous a <- ggplot(mpg, aes(hwy)) a + geom_area(stat = "bin") x, y, alpha, color, fill, linetype, size b + geom_area(aes(y = ..density..), stat = "bin") a + geom_density(kernel = "gaussian") x, y, alpha, color, fill, linetype, size, weight b + geom_density(aes(y = ..county..)) a + geom_dotplot() x, y, alpha, color, fill a + geom_freqpoly() x, y, alpha, color, linetype, size b + geom_freqpoly(aes(y = ..density..)) a + geom_histogram(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight b + geom_histogram(aes(y = ..density..)) Discrete b <- ggplot(mpg, aes(fll)) b + geom_bar() x, alpha, color, fill, linetype, size, weight	Continuous X, Continuous Y f <- ggplot(mpg, aes(cty, hwy)) f + geom_blank() (Useful for expanding limits) f + geom_jitter() x, y, alpha, color, fill, shape, size f + geom_point() x, y, alpha, color, fill, shape, size f + geom_quantile() x, y, alpha, color, linetype, size, weight f + geom_rug(sides = "bl") alpha, color, linetype, size f + geom_smooth(method = lm) x, y, alpha, color, fill, linetype, size, weight f + geom_text(aes(label = cty)) x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust Continuous Bivariate Distribution i <- ggplot(movies, aes(year, rating)) i + geom_bin2d(binwidth = c(5, 0.5)) xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight i + geom_density2d() x, y, alpha, colour, linetype, size i + geom_hex() x, y, alpha, colour, fill size Continuous Function j <- ggplot(economics, aes(date, unemployment)) j + geom_area() x, y, alpha, color, fill, linetype, size j + geom_line() x, y, alpha, color, linetype, size j + geom_step(direction = "hv") x, y, alpha, color, linetype, size Visualizing error df <- data.frame(grp = c("A", "B"), fit = 4.5, se = 1.2) k <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se)) k + geom_crossbar(fatten = 2) x, y, ymax, ymin, alpha, color, fill, linetype, size k + geom_errorbar() x, ymax, ymin, alpha, color, linetype, size, width (also geom_errorbarh) k + geom_linerange() x, ymin, ymax, alpha, color, linetype, size k + geom_pointrange() x, y, ymin, ymax, alpha, color, fill, linetype, shape, size Maps data <- data.frame(murder = USArrests\$Murder, state = tolower(row.names(USArrests))) map <- map_data("state") ggplot(data, aes(fill = murder)) i + geom_map(aes(map_id = state), map = map) + expand_limits(x = map\$long, y = map\$lat) map_id, alpha, color, fill, linetype, size Discrete X, Continuous Y g <- ggplot(mpg, aes(class, hwy)) g + geom_bar(stat = "identity") x, y, alpha, color, fill, linetype, size, weight g + geom_boxplot() lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, shape, size, weight g + geom_dotplot(binaxis = "y", stackdir = "center") x, y, alpha, color, fill g + geom_violin(scale = "area") x, y, alpha, color, fill, linetype, size, weight Discrete X, Discrete Y h <- ggplot(diamonds, aes(cut, color)) h + geom_jitter() x, y, alpha, color, fill, shape, size Three Variables m <- geom_raster(aes(fill = z), hjust=0.5, vjust=0.5, interpolate=FALSE) x, y, alpha, fill (fast) m + geom_tile(aes(fill = z)) x, y, alpha, color, fill, linetype, size (slow) m + geom_contour(aes(z = z)) x, y, z, alpha, colour, linetype, size, weight

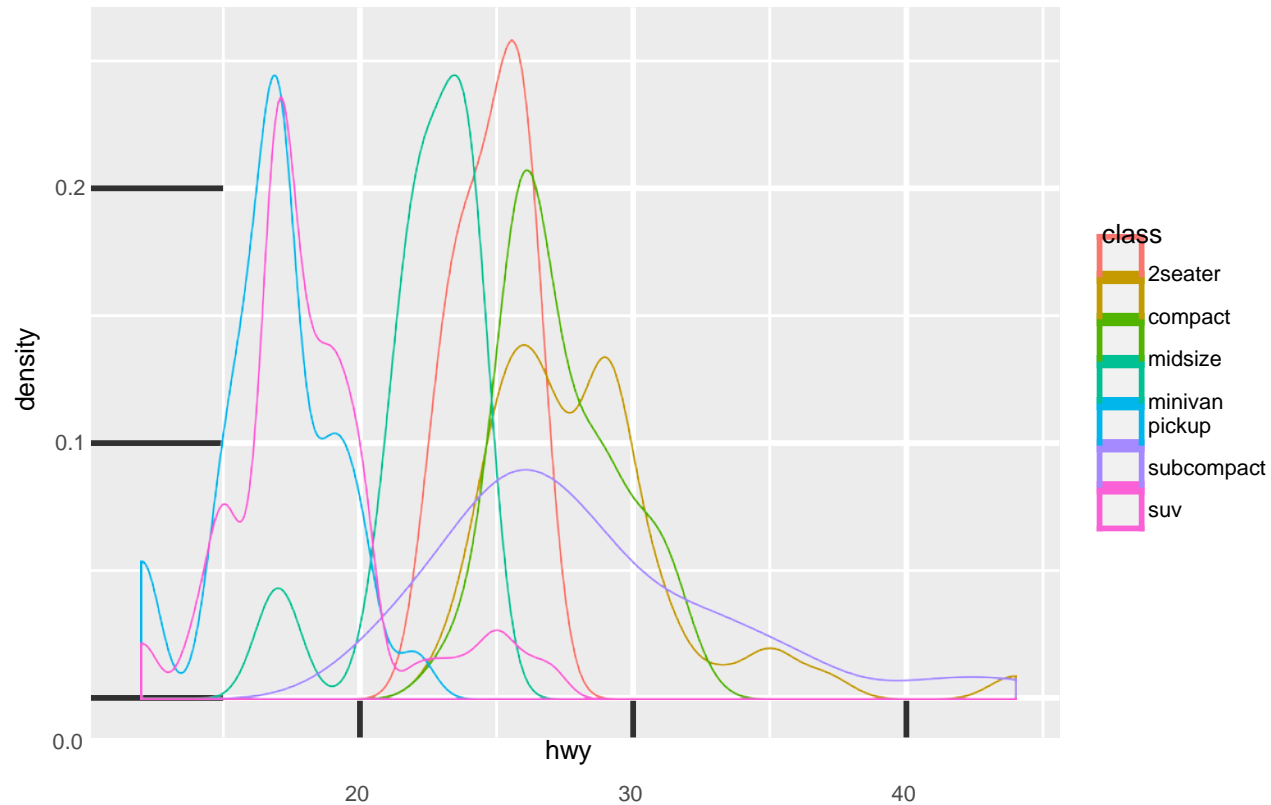
Graphical Primitives
 map <- map_data("state")
 c <- ggplot(map, aes(long, lat))
 c + geom_polygon(aes(group = group))
 x, y, alpha, color, fill, linetype, size
 d <- ggplot(economics, aes(date, unemployment))
 d + geom_path(linetype = "butt", linejoin = "round", linemitre = 1)
 x, y, alpha, color, linetype, size
 d + geom_ribbon(aes(ymin = unemployment - 900, ymax = unemployment + 900))
 x, y, alpha, color, fill, linetype, size
 e <- ggplot(seals, aes(x = long, y = lat))
 e + geom_segment(aes(xend = long + delta_long, yend = lat + delta_lat))
 x, xend, y, yend, alpha, color, linetype, size
 e + geom_rect(aes(xmin = long, ymin = lat, xmax = long + delta_long, ymax = lat + delta_lat))
 xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

Beispiel für geom_function



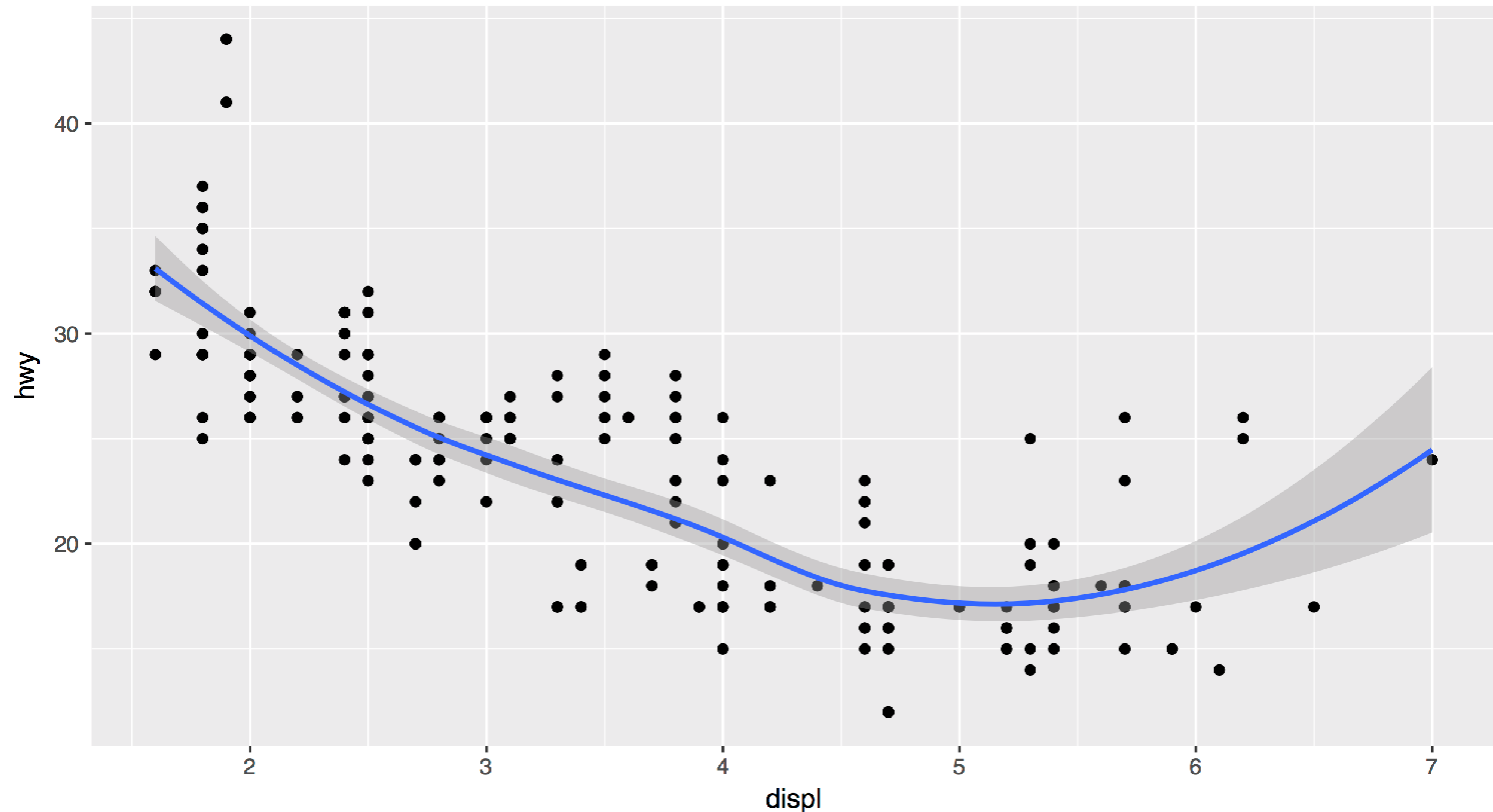
```
ggplot(data = mpg) +  
  geom_boxplot(mapping = aes(x = class, y = hwy))
```

Beispiel für geom_function



```
ggplot(data = mpg)+  
  geom_density(mapping = aes(x = hwy, color = class))
```

Jedes geom_X fügt einen neuen Layer hinzu



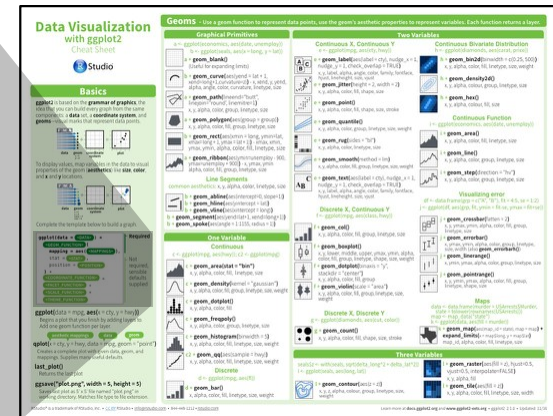
```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  geom_smooth(mapping = aes(x = displ, y = hwy))
```

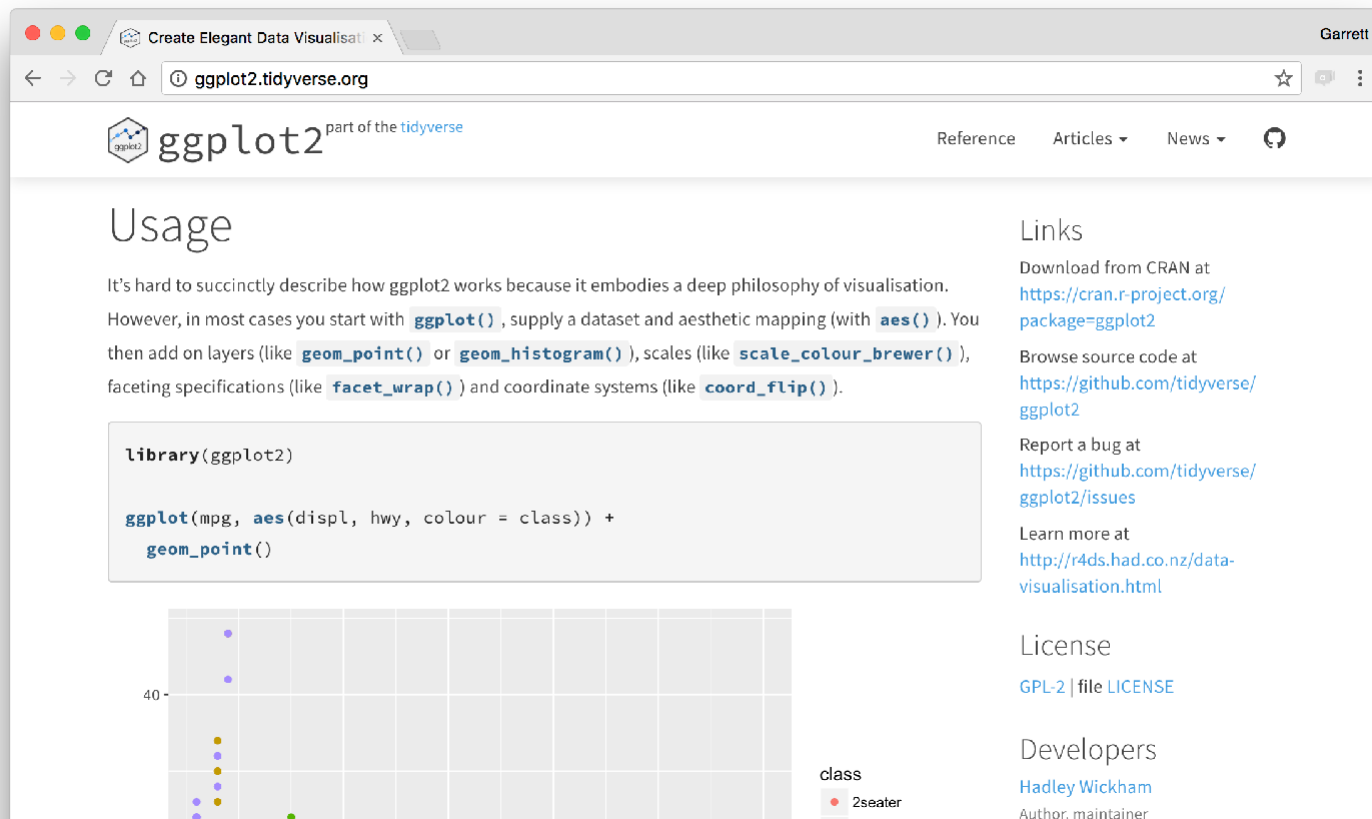
ggplot2 Template

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION> (
    mapping = aes(<MAPPINGS>),
    stat = <STAT>,
    position = <POSITION>
  ) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION> +
  <SCALE_FUNCTION> +
  <THEME_FUNCTION>
```

Required

Not
required,
sensible
defaults
supplied





The screenshot shows the ggplot2 website. The browser tab is titled 'Create Elegant Data Visualisations'. The address bar shows 'ggplot2.tidyverse.org'. The website header includes the ggplot2 logo, the text 'part of the tidyverse', and navigation links for 'Reference', 'Articles', and 'News'. The main content area is titled 'Usage' and contains a paragraph explaining the philosophy of ggplot2. Below the text is a code block with the following R code:

```
library(ggplot2)

ggplot(mpg, aes(displ, hwy, colour = class)) +
  geom_point()
```

Below the code block is a scatter plot showing highway mileage (hwy) on the y-axis versus engine displacement (displ) on the x-axis. The points are colored by car class. A legend on the right indicates that red points represent '2seater' cars. The plot shows a clear negative correlation between engine displacement and highway mileage, with 2-seater cars generally having higher mileage for a given displacement compared to other classes.

Links

- Download from CRAN at <https://cran.r-project.org/package=ggplot2>
- Browse source code at <https://github.com/tidyverse/ggplot2>
- Report a bug at <https://github.com/tidyverse/ggplot2/issues>
- Learn more at <http://r4ds.had.co.nz/data-visualisation.html>

License

GPL-2 | [file LICENSE](#)

Developers

[Hadley Wickham](#)
Author, maintainer