

A2 - Das ggmap Paket

Jan-Philipp Kolb

22 Oktober 2018

Inhalt dieses Abschnitts

Arten von räumlichen Daten:

- **Straßenkarten**
- **Satelliten Bilder**
- **Physische Daten und Karten**
- **Abstrakte Karten**
- ...

Das R-paket `ggmap` wird im folgenden genutzt um verschiedene Kartentypen darzustellen.

Mit `qmap` kann man eine schnelle Karte erzeugen.

Installieren des Paketes

- Zur Erstellung der Karten brauchen wir das Paket ggmap:

```
devtools::install_github("dkahle/ggmap")
devtools::install_github("hadley/ggplot2")
install.packages("ggmap")
```

Paket ggmap - Hallo Welt

- Um das Paket zu laden verwenden wir den Befehl library
- ```
library(ggmap)
```

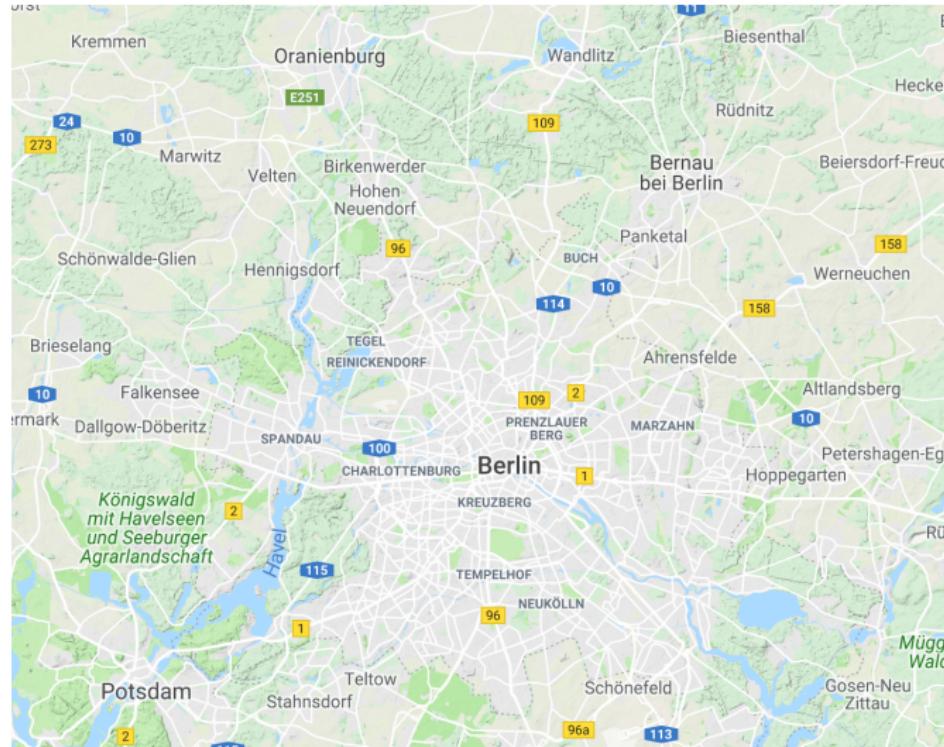
Und schon kann die erste Karte erstellt werden:

```
qmap("Mannheim")
```



# Karte für eine Sehenswürdigkeit

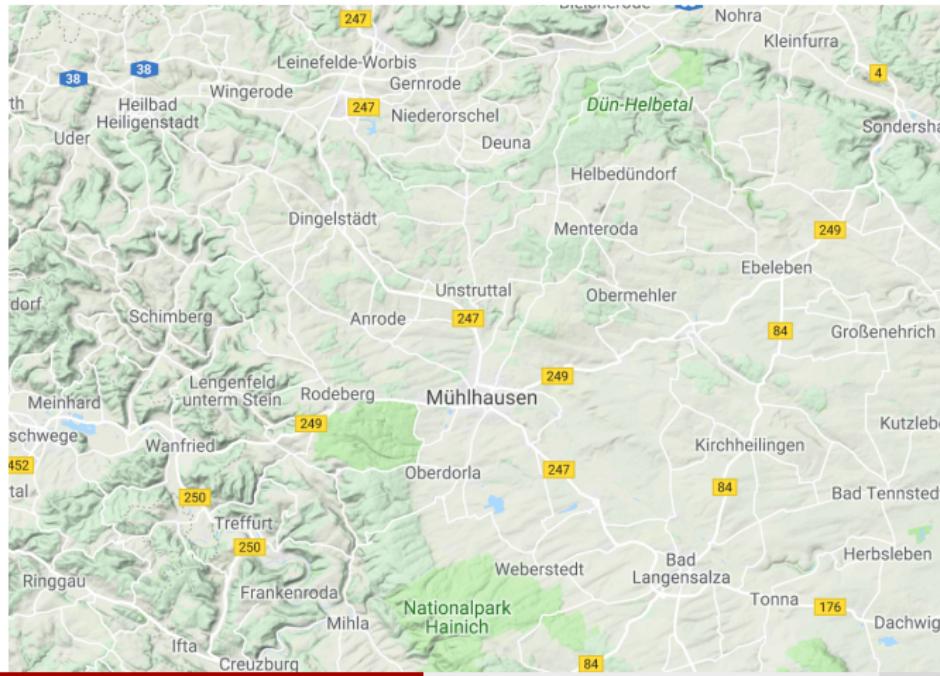
`qmap("Berlin Brandenburger Tor")`



# Karte für einen ganzen Staat

`qmap("Germany")`

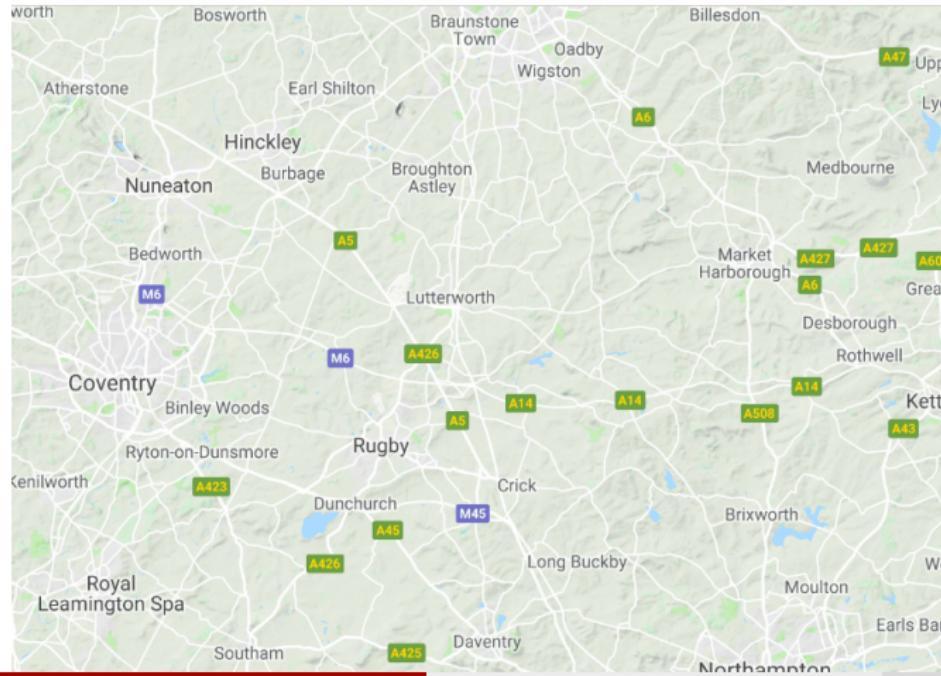
- Wir brauchen ein anderes *zoom level*



# Ein anderes *zoom level*

- level 3 - Kontinent / level 10 - Stadt / level 21 - Gebäude

```
qmap("England", zoom = 6)
```



# Hilfe bekommen wir mit dem Fragezeichen

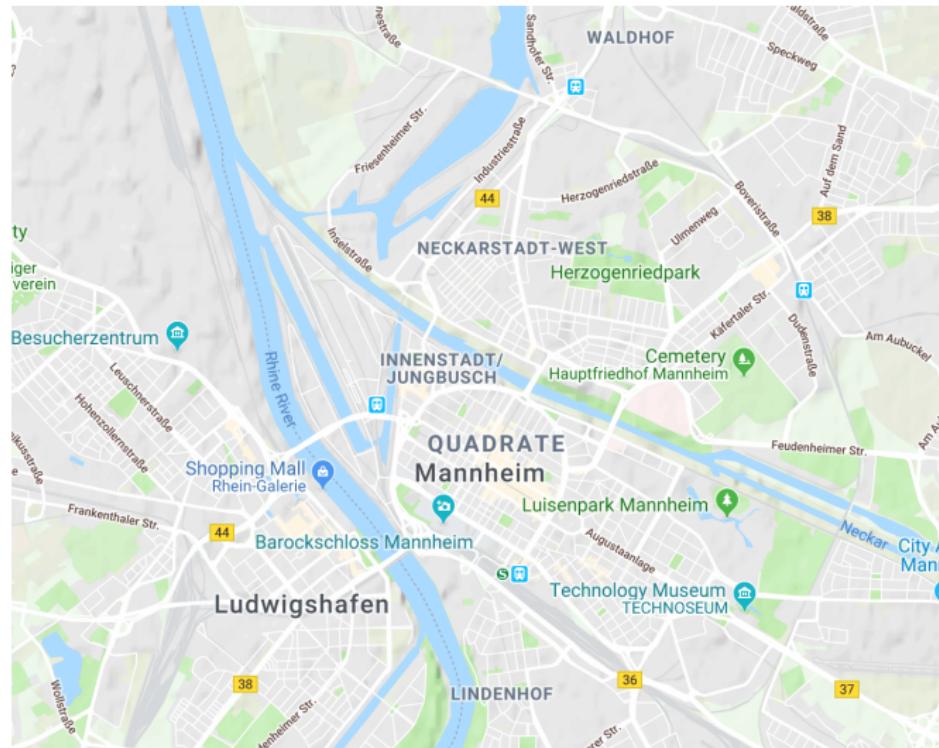
```
?qmap
```

Verschiedene Abschnitte in der Hilfe:

- Description
- Usage
- Arguments
- Value
- Author(s)
- See Also
- Examples

# Ganz nah dran

```
qmap('Mannheim', zoom = 20)
```



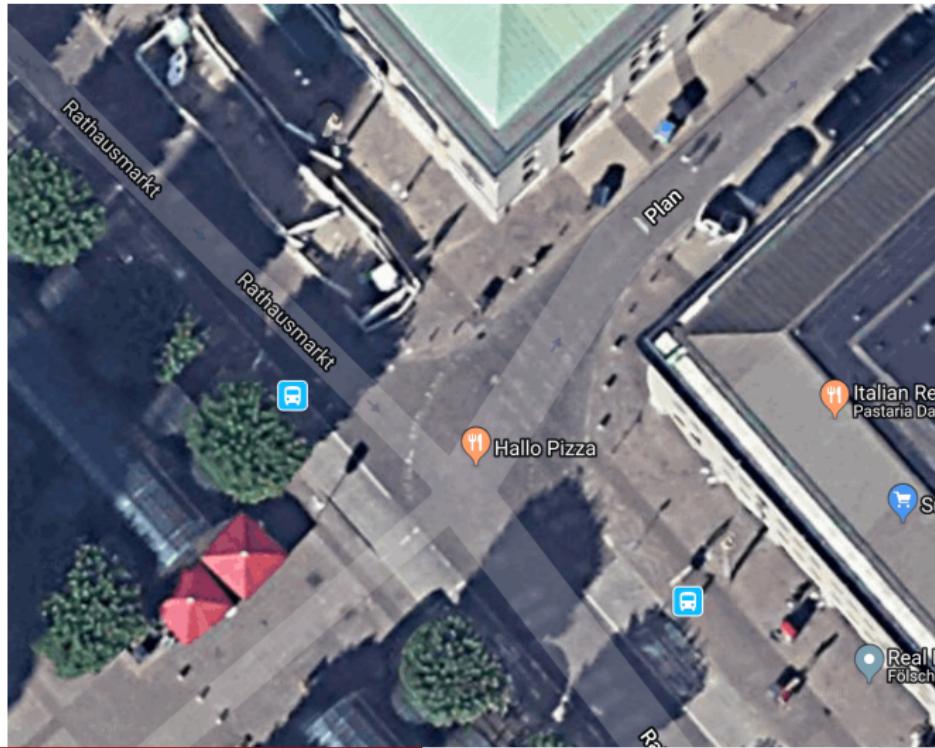
## ggmap - maptype satellite

```
qmap('Hamburg', zoom = 14, maptype="satellite")
```



# ggmap - maptype satellite zoom 20

```
qmap('Hamburg', zoom = 20, maptype="hybrid")
```



# Terrain/physical maps

- Aus Physischen Karten kann man Informationen über Berge, Flüsse und Seen ablesen.
- Farben werden oft genutzt um Höhenunterschiede zu visualisieren

```
qmap('Arequipa', maptype="terrain")
```

# Eine physische Karte von Arequipa



# Abstrahierte Karten (<http://www.designfaves.com>)

- Abstraktion wird genutzt um nur essentielle Informationen darzustellen.
- Bsp. U-Bahn Karten - wichtig sind Richtungen und Orientierung
- Im folgenden werden Karten vorgestellt, die sich gut als Hintergrundkarten eignen.

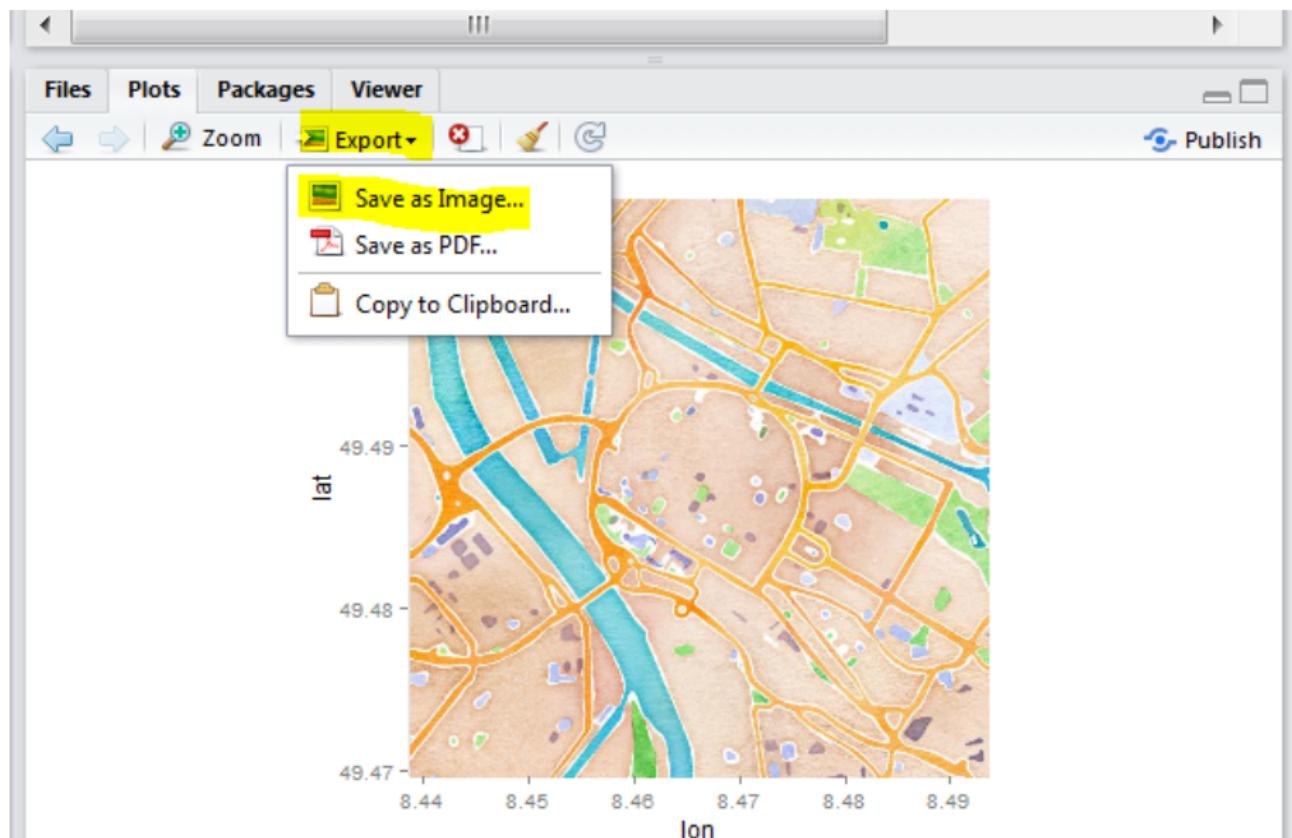


# ggmap - maptype watercolor

```
qmap('Los Angeles', zoom = 14,
 maptype="watercolor",source="stamen")
```



# Graphiken speichern



# ggmap - ein Objekt erzeugen

- <- ist der Zuweisungspfeil um ein Objekt zu erzeugen
- Dieses Vorgehen macht bspw. Sinn, wenn mehrere Karten nebeneinander gebraucht werden.

```
MA_map <- qmap('Mannheim',
 zoom = 14,
 maptype="toner",
 source="stamen")
```

# Eine Karte für Trier

- Mit dem Befehl `OSM_scale_lookup` bekommt man heraus, welchen Wert man für `scale` angeben muss.

```
OSM_scale_lookup(zoom = 10)
qmap(location = "Trier", zoom = 10, source = "osm",
 scale=575000)
```

## A2A Aufgabe - Nutzung von ggmap

- Erzeuge mit dem Paket ggmap eine Karte von einem Ort, der Ihnen besonders gut gefällt und speichern Sie diesen.

# Cheatsheet

## • Cheatsheet zu data visualisation

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

### Data Visualization with ggplot2

Cheat Sheet

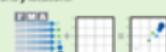


#### Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same few components: a **data set**, a set of **geoms**—visual marks that represent data points, and a **coordinate system**.



To display data values, map variables in the data set to aesthetic properties of the geom like **size**, **color**, and **x** and **y** locations.



Build a graph with **epitools** or **ggplot2**

**epitools**: `epitools::epitools::epitools`  
`epitools::epitools::epitools::epitools`(x = cyl, y = hmpg, color = cyl, data = mpg, geom = "point")  
Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

**ggplot2**: `ggplot(mpg, aes(x = cyl, y = hwy))`

Begins a plot that you finish by adding layers to. No defaults, but provides more control than **epitools**.

`ggplot(mpg, aes(x = cyl, y = hwy)) +  
 geom_point(mapping = c(cyl ~ color, hwy ~ size))`

Creates a scatter plot with colored points and varying sizes.

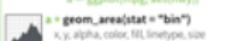
Creates a scatter plot with colored points and varying sizes.

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

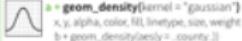
#### One Variable

##### Continuous

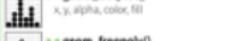
`a + geom_area(stat = "bin")`



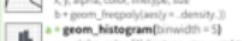
`a + geom_density(kernel = "gaussian")`



`a + geom_dotplot()`



`a + geom_freqpoly()`



`a + geom_histogram(binwidth = 5)`



`b + geom_hexagon(hexdensity = density)`



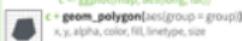
##### Discrete

`b + geom_bar()`



#### Graphical Primitives

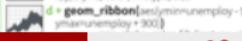
`c + ggplot(mpg, aes(long, lat))`



`d + ggplot(economics, aes(date, unemploy))`



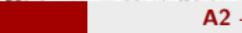
`d + geom_point(pathend = "but", linejoin = "mitre", size = 1)`



`d + geom_dotplot(binaxis = "y", stackdir = "center")`



`e + geom_violin(scale = "area")`



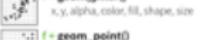
#### Two Variables

##### Continuous X, Continuous Y

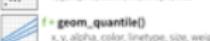
`f + ggplot(mpg, aes(x, hwy))`



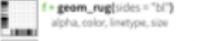
`f + geom_jitter()`



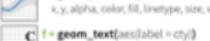
`f + geom_point()`



`f + geom_quantile()`



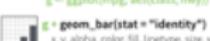
`f + geom_rug(sides = "tl")`



`f + geom_smooth(model = lm)`



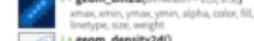
`f + geom_text(label = city)`



##### Continuous Bivariate Distribution

`i +`

`geom_bin2d(binwidth = c(0.5, 0.5))`



`i +`

`geom_hex2d()`



`i +`

`geom_hex()`

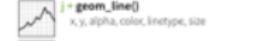


##### Continuous Function

`j + ggplot(economics, aes(date, unemploy))`



`j + geom_area()`



`j + geom_line()`



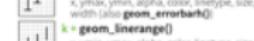
`j + geom_step(direction = "hv")`



##### Visualizing error

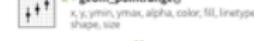
`k + data.frame(g = c("A", "B"), fit = 4.5, se = 1.2)`

`k + ggplot(df, aes(g, fit, ymin = fit - se, ymax = fit + se))`



`k +`

`geom_errorbar()`



`k +`

`geom_linerange()`



`k +`

`geom_pointrange()`



##### Maps

`data <- data.frame(murder = USArrests$Murder,`

Jan-Philipp Kolb

A2 - Das ggmap Paket

22 Oktober 2018

20 / 21

# Resourcen und Literatur

- Artikel von **David Kahle und Hadley Wickham** zur Nutzung von `ggmap`.
- **Schnell eine Karte bekommen**
- **Karten machen mit R**