Universität Augsburg
Wirtschaftswissenschaftliche
Fakultät

Seminar: Data Science and Decision Science

Data Visualisation & Storytelling

Hanan Loulou & Julian Zacharias

Sommersemester 2024

Gliederung

1. Tidy Data
2. The Grammar of Graphics
3. Das Paket ggplot2
4. Erklärung der von uns erstellten Plots

Data Semantics

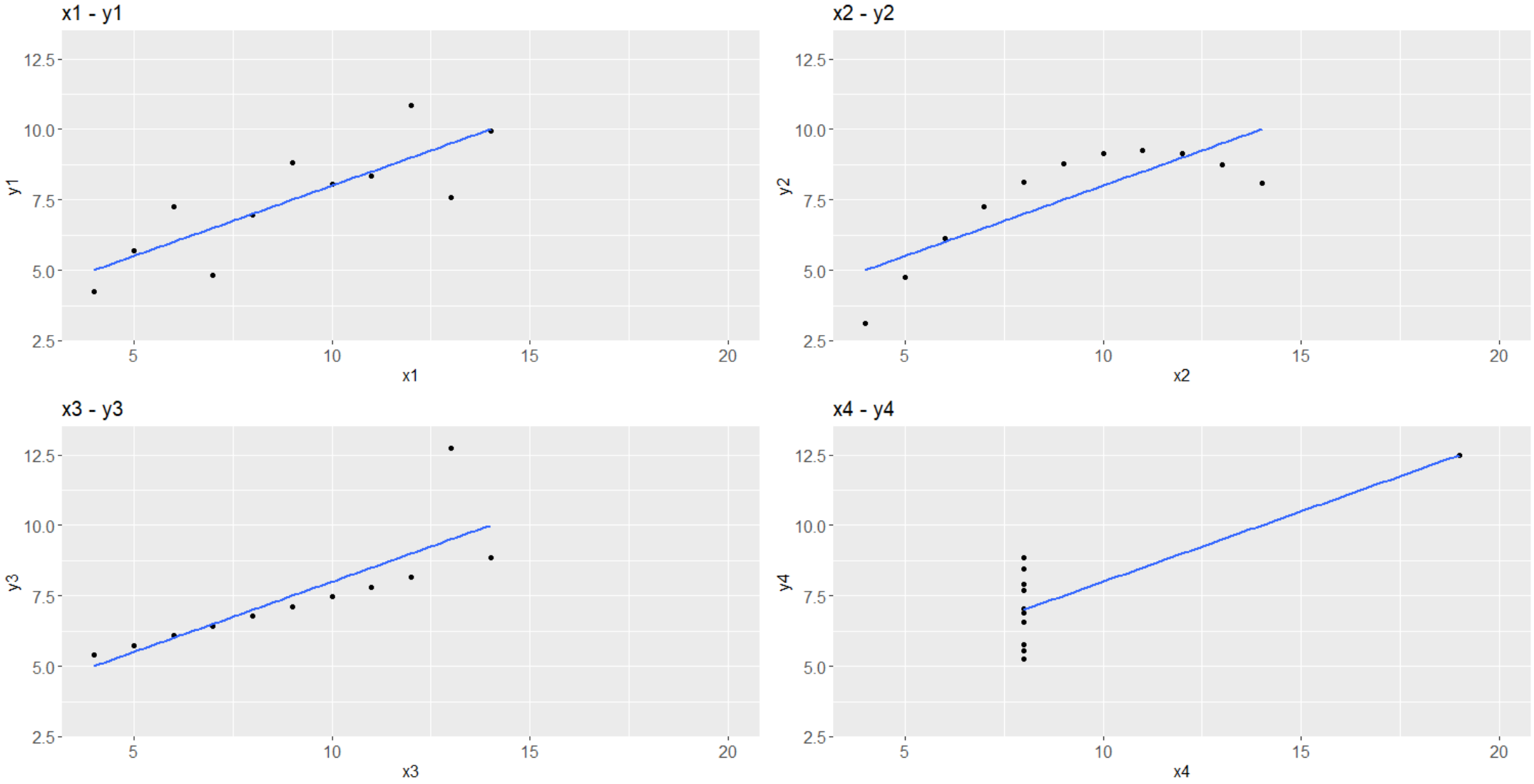
[1]

	Variable	Variable
Beobachtung	Wert	Wert
Beobachtung	Wert	Wert

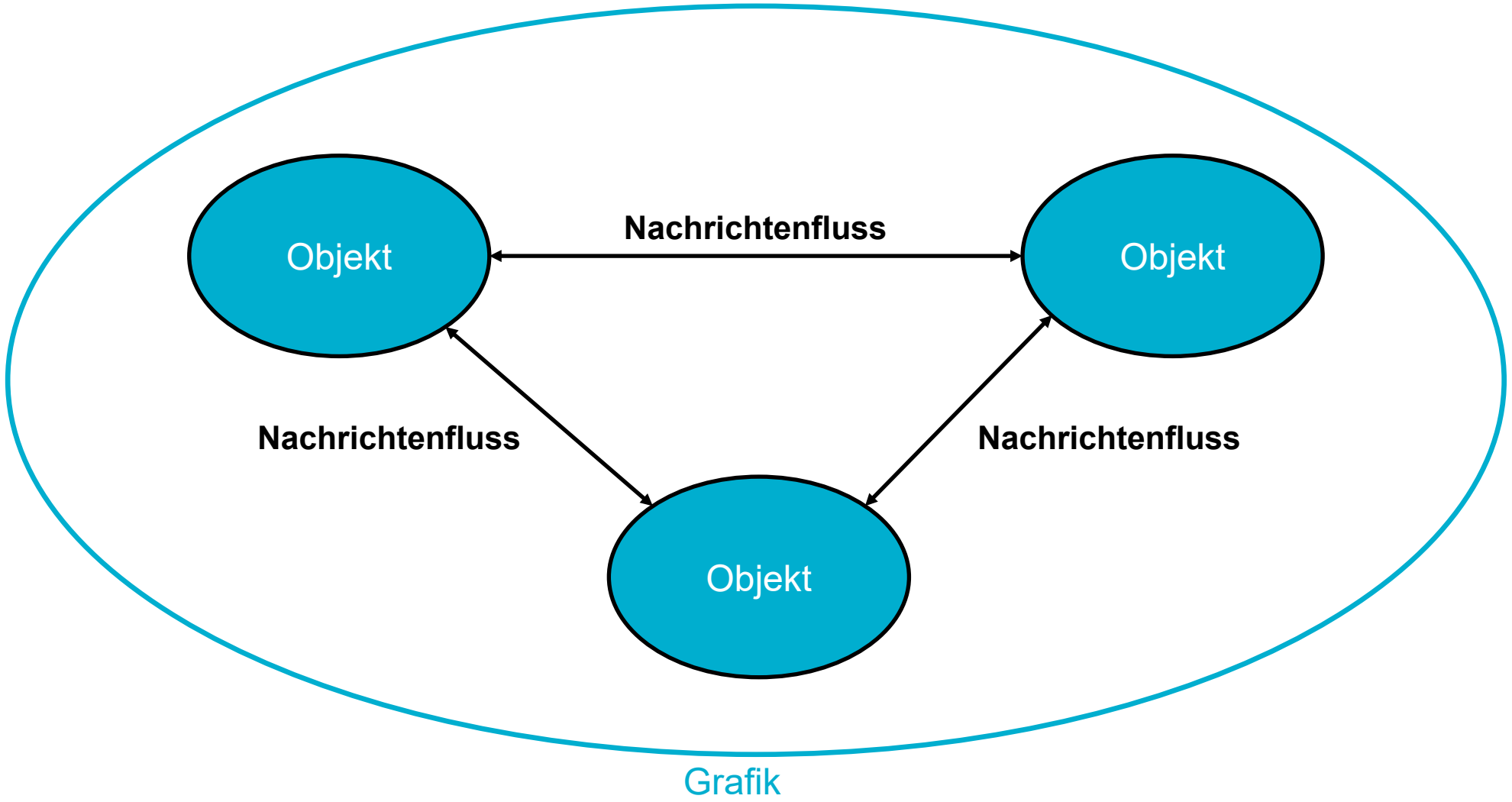
Die 3 Prinzipen von „tidy Data“ [1]

1. Jede Variable bildet eine Spalte
2. Jede Beobachtung bildet eine Zeile
3. Jede Beobachtungseinheit bildet eine Tabelle

Bedeutung von Data Visualization: Das Quartett von Anscombe [7]



Aufbau einer Grafik in einem objektorientierten Grafiksystem [5]



Eigenschaften von Objekten (1) [5]

Polymorphismus

Objekte können auf die selbe Nachricht unterschiedlich reagieren/antworten

Modularität

Komponenten des Systems sind relativ modular
→ Falls Teile des Systems versagen funktioniert das restliche System weiter

Eigenschaften von Objekten (2)

[5]

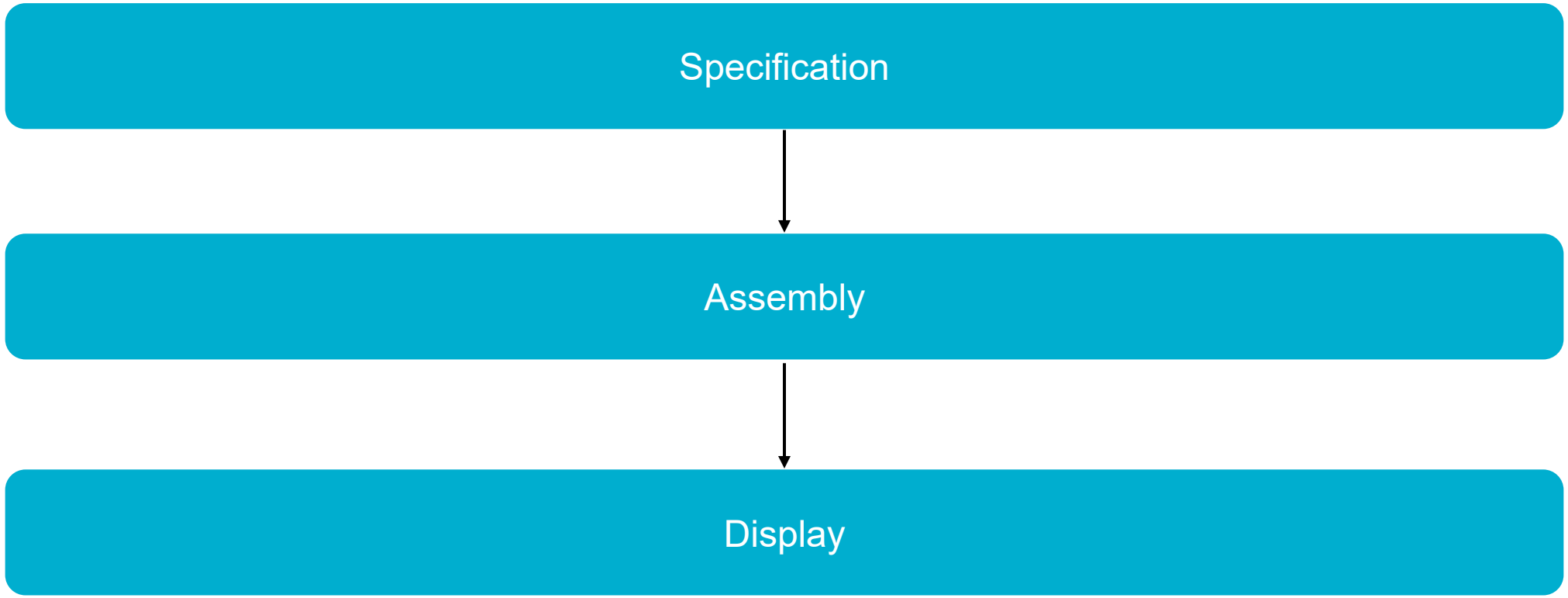
Vererbung

Objekte können Eigenschaften und Methoden von anderen Objekten erben

Wiederverwendung

Objekte können wiederverwendet werden

Stufen der Grafikerstellung [5]



Specification: Die tiefe Grammatik einer Grafik [5]

DATA
(Datenoperationen)

ELEMENT
(Graphen & Ästhetik)

TRANS
(Variablenumwandlungen)

SCALE
(Skalentransformationen)

COORD
(Koordinatensystem)

GUIDE
(Achsen und Legende)

Assembly: Die „Montage“ der Graphik

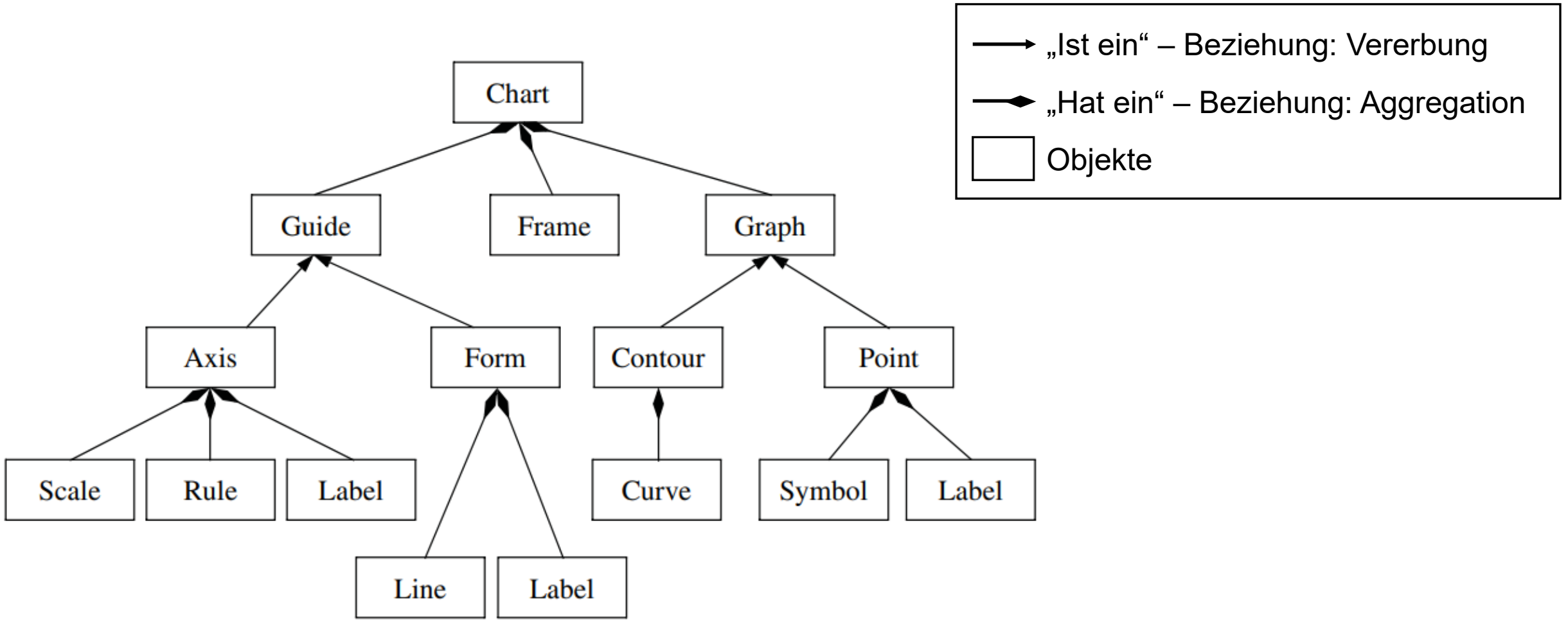
[5]

Grafik ≠ Menge ihrer Objekte

Einzelne Objekte der Specification der Grafik müssen in Beziehung gesetzt werden

Assembly: Design Tree

[5]



Display: Die Anzeige der Grafik

[5]

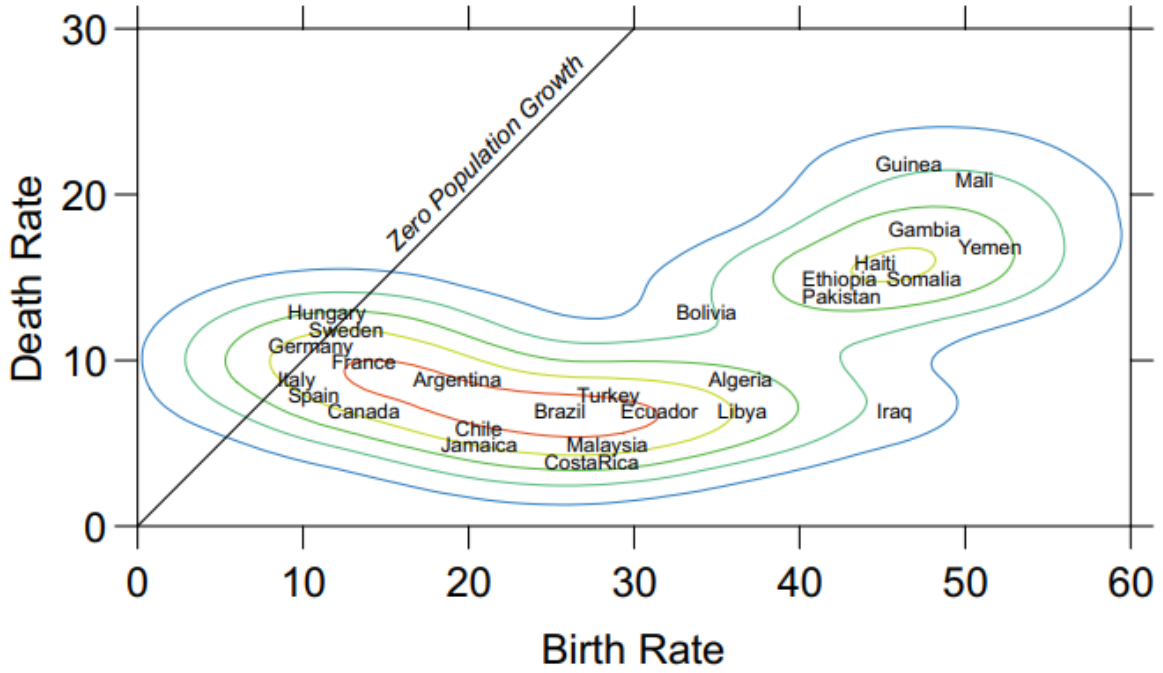
Design Tree

Layout Designer

Rendering Tools
+ Display System

```
ELEMENT: point(position(birth*death), size(0), label(country))
ELEMENT: contour(position(
            smooth.density.kernel.epanechnikov.joint(birth*death)),
            color.hue())
GUIDE: form.line(position((0,0),(30,30)), label("Zero Population Growth"))
GUIDE: axis(dim(1), label("Birth Rate"))
GUIDE: axis(dim(2), label("Death Rate"))
```

Ziel: Gegenüberstellung von Sterbe- und Geburtenraten für verschiedene Länder



Revision: „to see again“ [5]

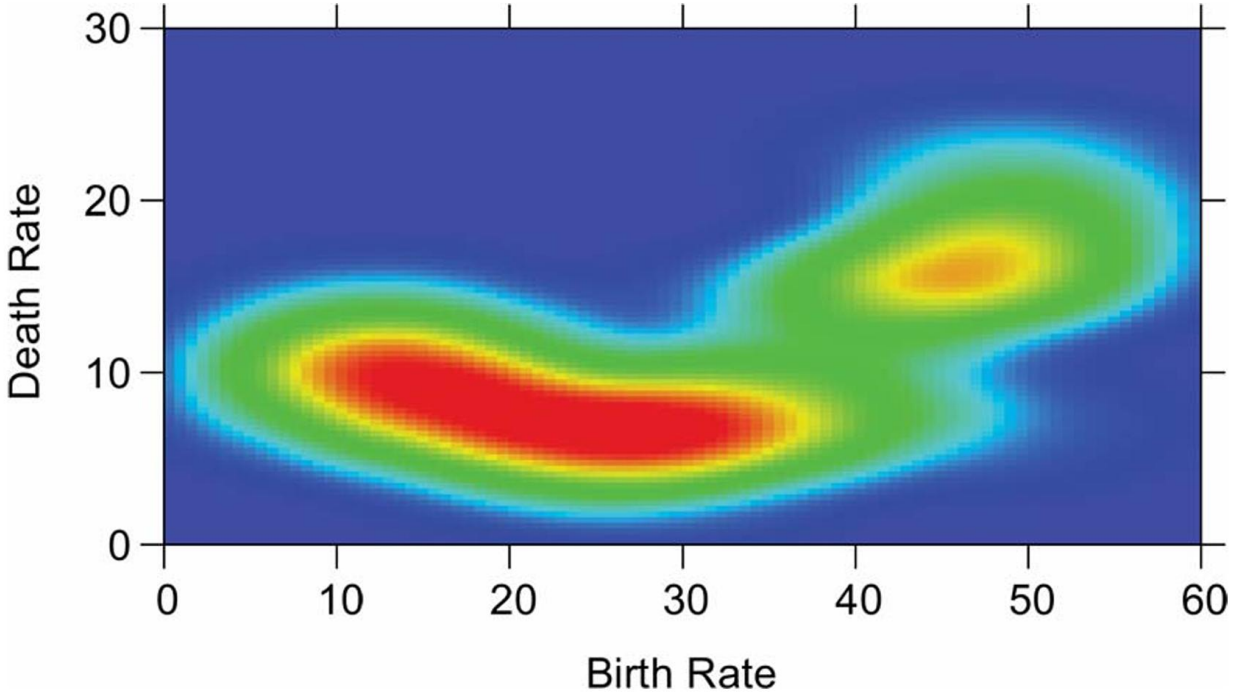
Forderung der Grammar of Graphics:
Zerlegung einer Grafik in ihre einzelnen unabhängigen Komponenten

Flexible Umgebung

Änderung der Grafik durch Steuerung/Veränderung der einzelnen Komponenten möglich, OHNE dass der gesamte Prozess der Erstellung der Graphik erneut durchlaufen werden muss

Revision: Beispiel

[5]



ELEMENT: *polygon*(*position*(
smooth.density.kernel.epanechnikov.joint(birth*death)), *color.hue*())

Fokus auf Dichte:
Anzahl Länder mit ähnlichem
Geburten-/Sterbeverhältnis

Das Paket ggplot2: Die praktische Umsetzung der GoG [3]

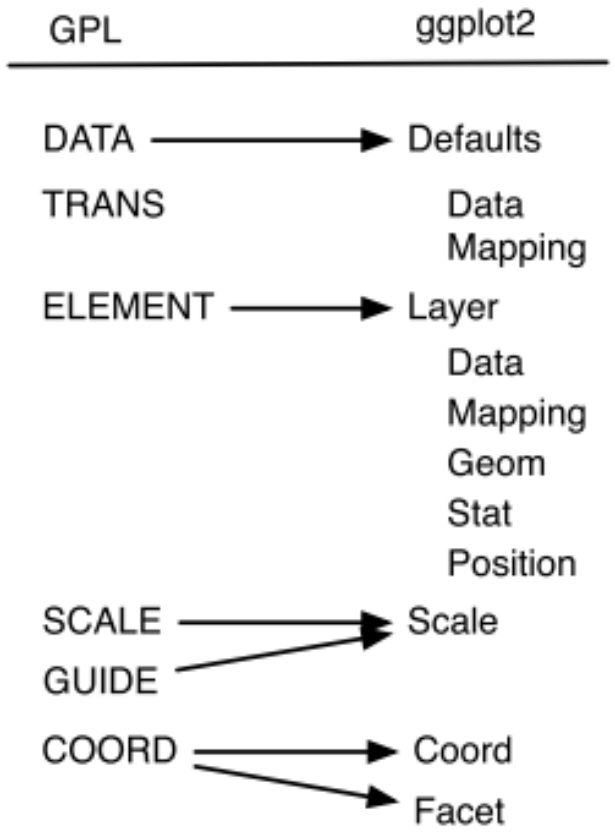
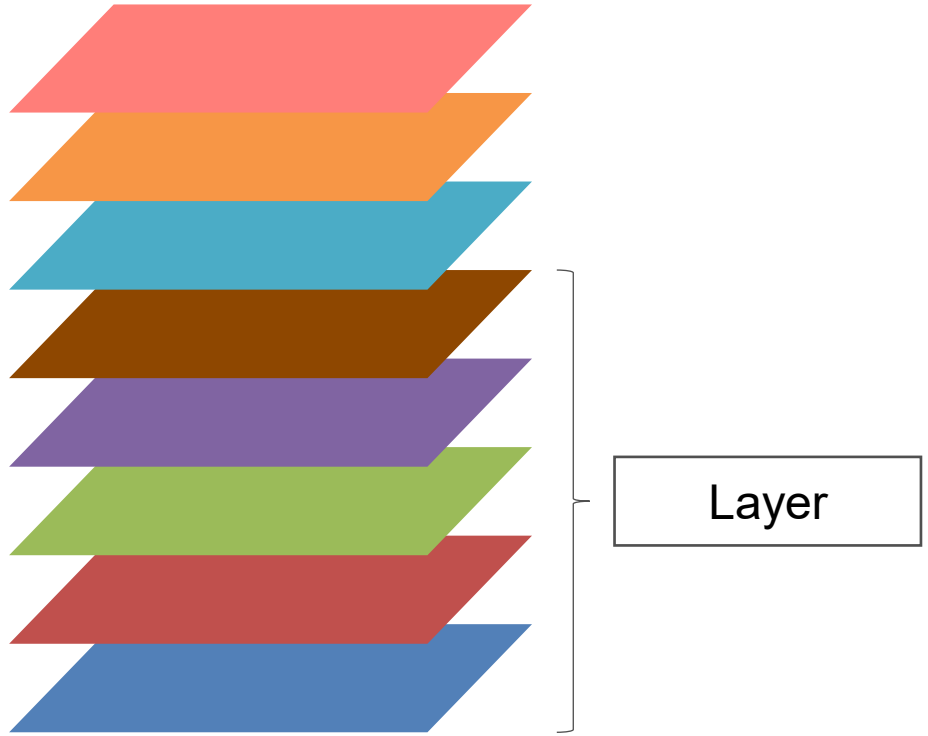


Figure 4. Mapping between components of Wilkinson's grammar (left) and the layered grammar (right). TRANS has no correspondence in `ggplot2`: its role is played by built-in R features.

Hierarchischer Aufbau einer Grafik in ggplot

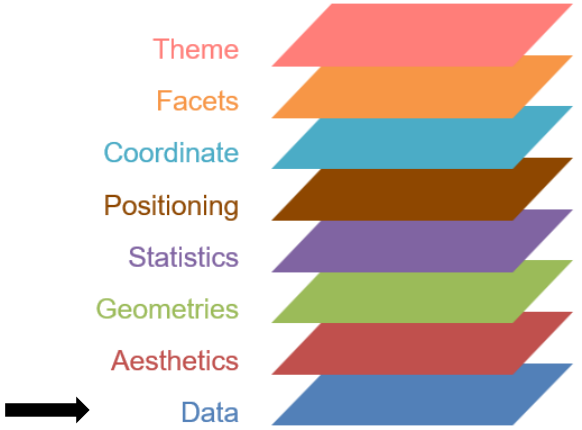
[2], [3], [4]

Anpassung nicht-datenbezogener Elemente	Theme
Matrix für Subplots	Facets
Positionierung von Elementen im Plot	Coordinate
Positionsanpassung von geometrischen Elementen	Positioning
Statistische Modelle und Zusammenfassungen	Statistics
Darstellungsart der Daten	Geometries
Verknüpfung von Daten und ästhetischen Attributen	Aesthetics
Daten, die geplottet werden	Data



1. Daten

[2], [3], [4]



- Müssen in Form eines tidy data frame vorliegen
- Unabhängig von anderen Komponenten
- Abstrakte Grafik —Daten→ Konkrete Grafik

```
ggplot(data = mpg)
```

2. Aesthetics: Mapping

[2], [3], [4]

Mapping: Zuweisung von ästhetischen Attributen zu Variablen

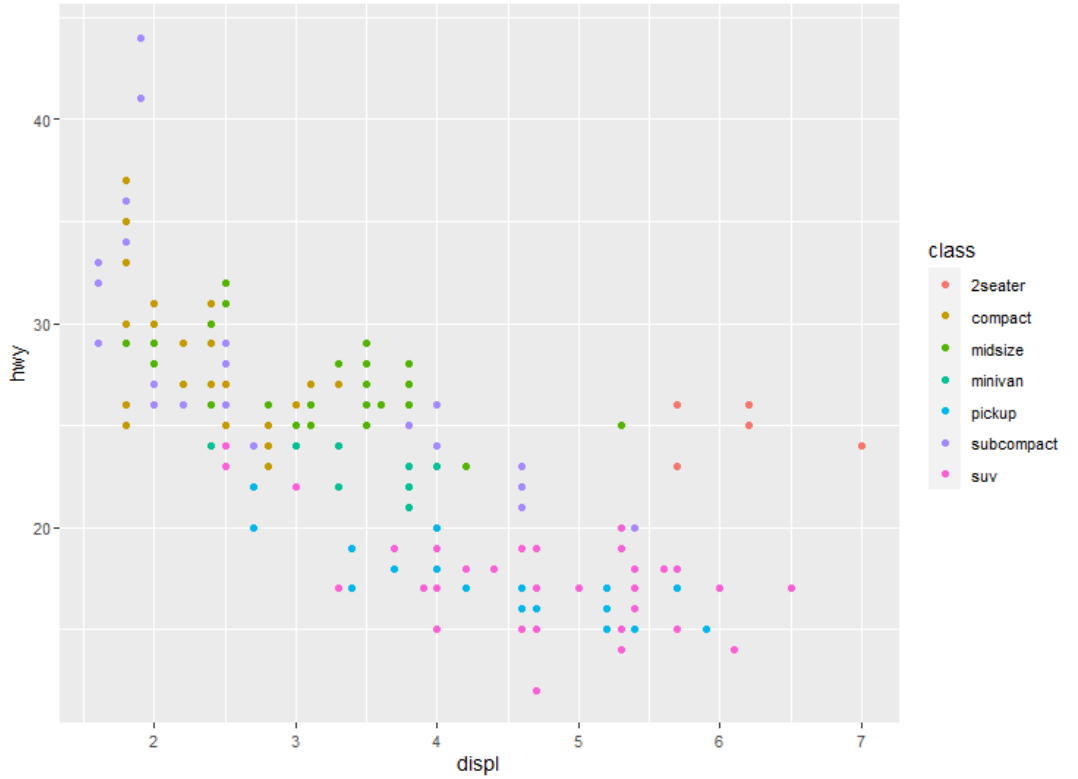
- Beispiele für ästhetische Attribute: `x`, `y`, `fill`, `size`, `shape`, `alpha`
- Mapping erfolgt innerhalb der Funktion `aes()`:

(a) Global

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

(b) Lokal

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



2. Aesthetics: Abgrenzung zu Setting

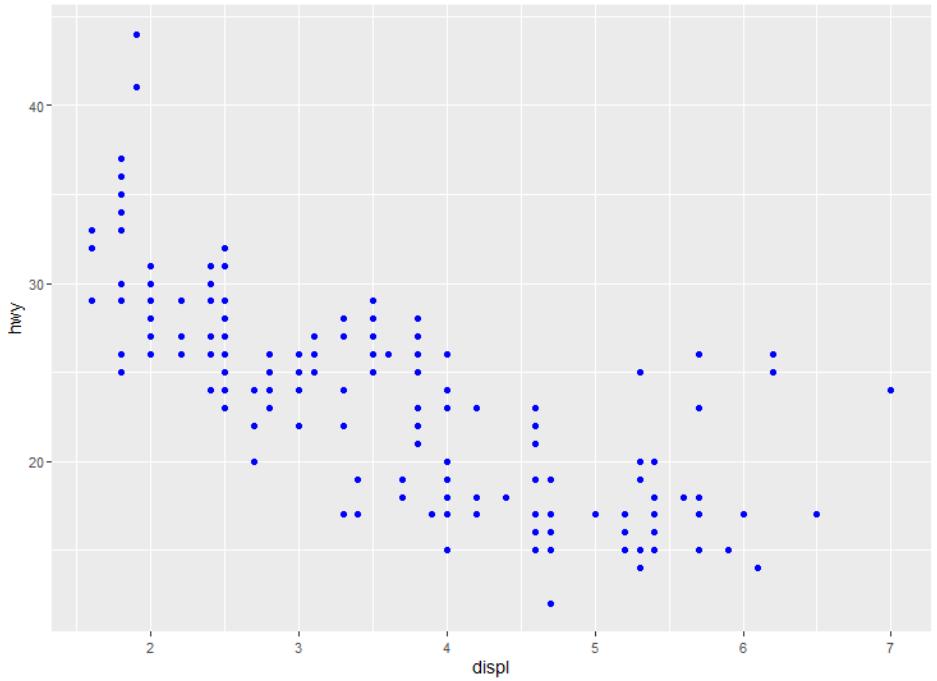
[2], [3], [4]

Setting: manuelle Festlegung einer visuellen Eigenschaft

➡ Kein Informationsgehalt !

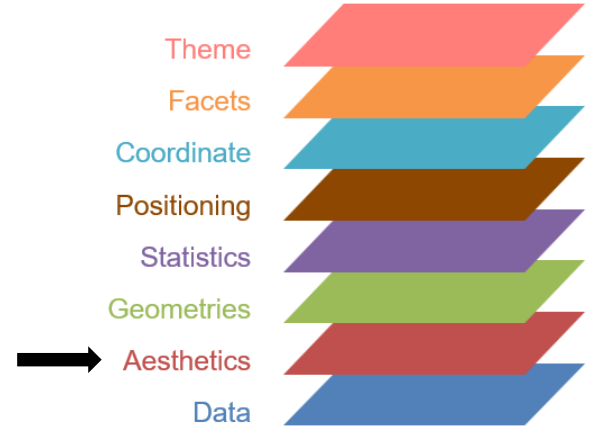
Beispiel: Einfärben aller Punkte in blau

```
ggplot(mpg, mapping = aes(x = displ, y = hwy)) +
geom_point(color = "blue")
```



2. Aesthetics: Scales

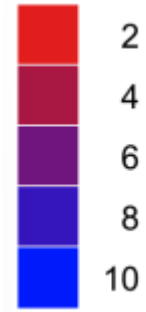
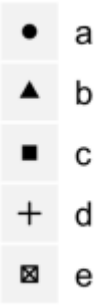
[2], [3], [4]



Skala: „a function from a region in data space (the domain of the scale) to a region in aesthetic space (the range of the scale)"

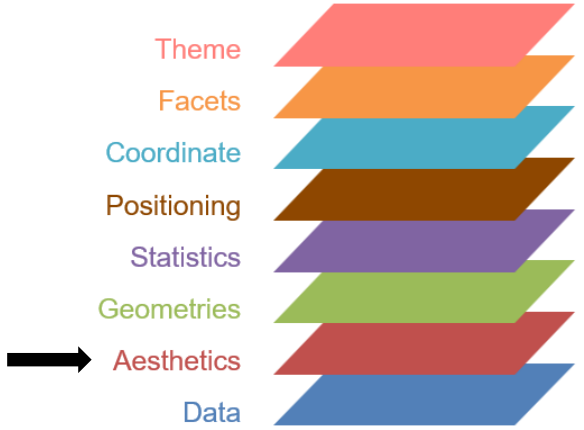
https://ggplot2-book.org/scales-guides#sec-scales-guides-theory

➡ Skalen kontrollieren das Mapping von Daten zu ästhetischen Attributen !

	Stetige Variable	Diskrete Variable
Funktion	Ordnet Daten eine Farbschattierung zu	Ordnet Daten eine Form zu
Inverse (Legende /Achsenbeschriftung)		

2. Aesthetics: Scales

[2], [3], [4]

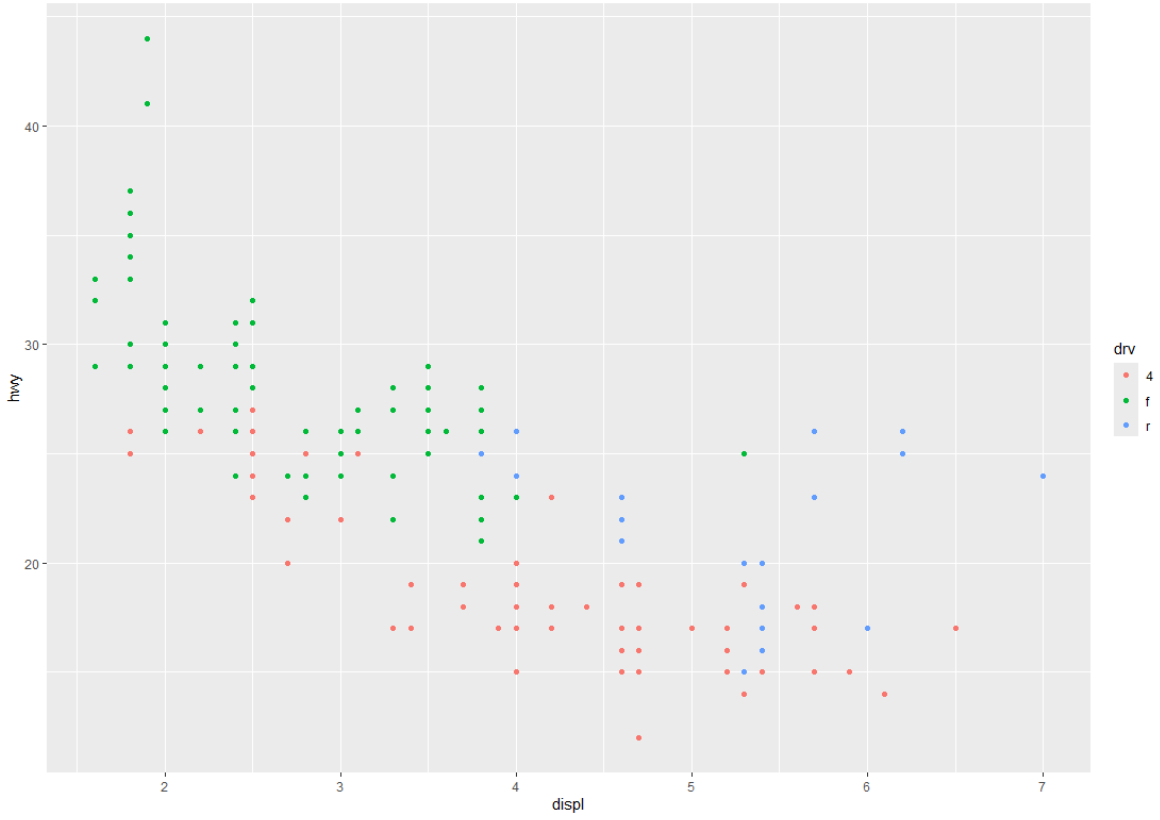


Code: `scale_{aesthetics}_{type}()`

Anpassungsmöglichkeiten:

- `limits`
- `breaks`
- `label`
- transformation

```
base = ggplot(mpg, aes(x= displ, y = hwy)) +
        geom_point(aes(color = drv)) +
        scale_x_continuous() +
        scale_y_continuous() +
        scale_color_discrete()
```



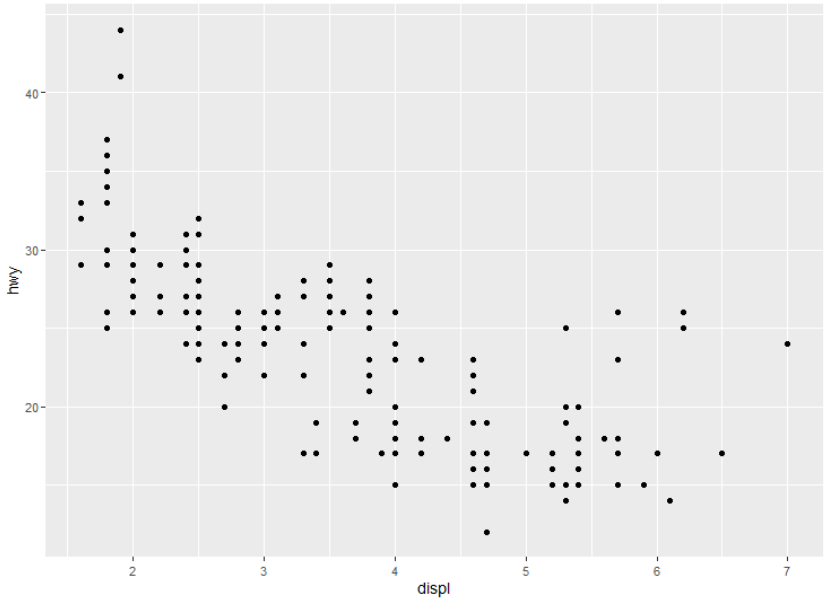
3. Geometries

[2], [3], [4]

Geom: Darstellungsart der Daten

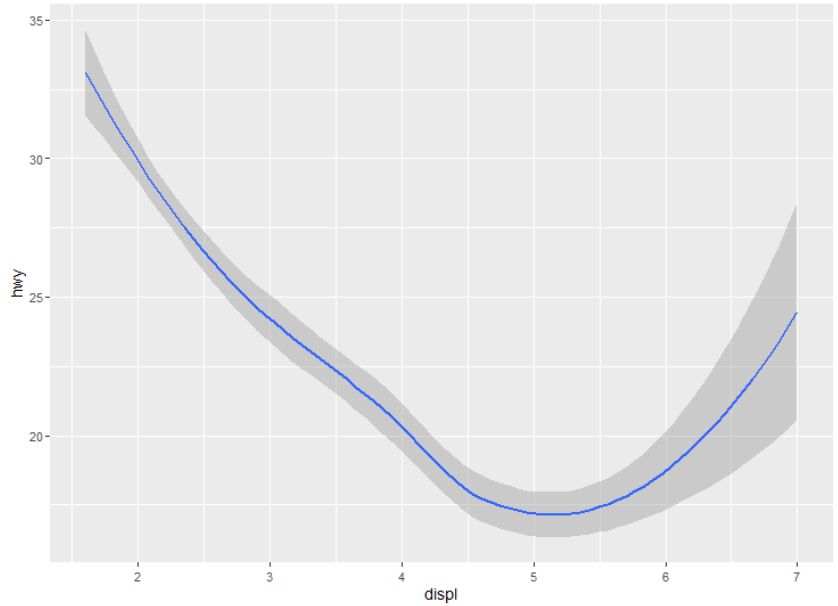
Point-Geom

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



Smooth-Geom

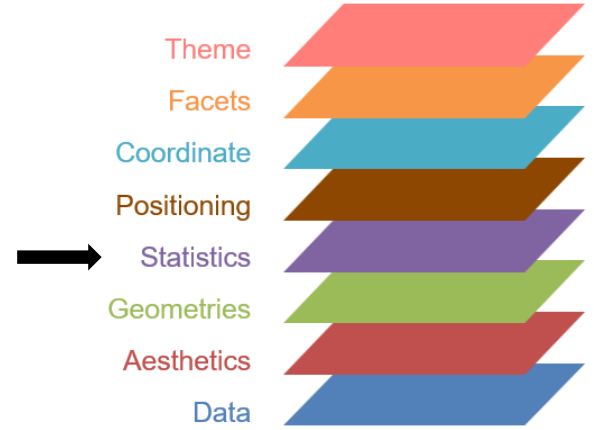
```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_smooth()
```



Klassifikation der Geoms nach Dimensionalität: (0d: Punkte, 1d: Linie, 2d: Intervall)

4. Statistics

[2], [3], [4]



Statistische Transformation:

- Berechnung von neuen Variablen
- Zusammenfassung der Daten

```
ggplot(diamonds, aes(x = cut)) + geom_bar()
```

1. **`geom_bar()`** begins with the **`diamonds`** data set

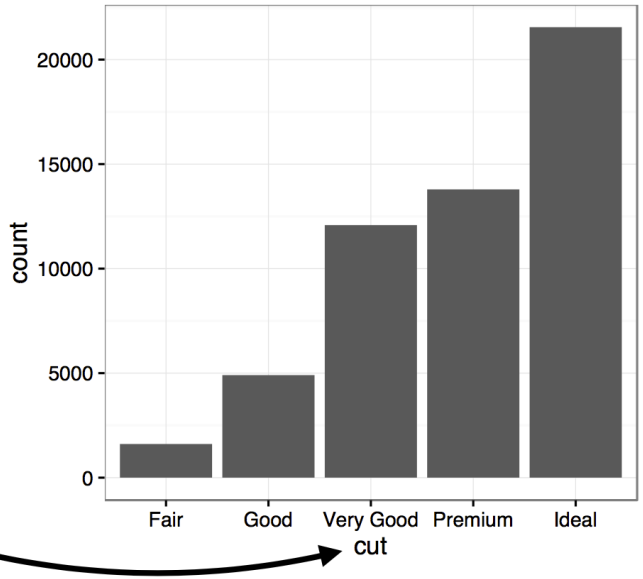
2. **`geom_bar()`** transforms the data with the "count" stat, which returns a data set of cut values and counts.

3. **`geom_bar()`** uses the transformed data to build the plot. cut is mapped to the x axis, count is mapped to the y axis.

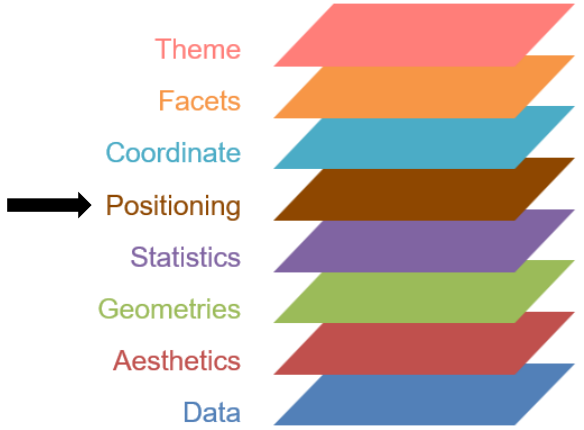
carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
...	...	...	...	...	...	...	...	...	...

stat_count()

cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1



[2], [3], [4]

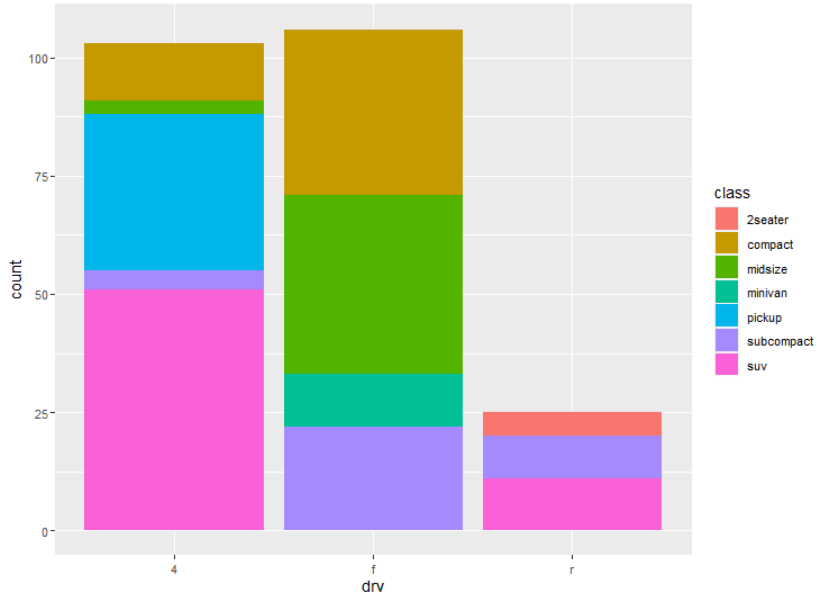


5. Positionsanpassung von geometrischen Elementen

Beispiele für Anpassungen: `position_stack()`, `position_fill()`, `position_dodge()` & `position_jitter()`

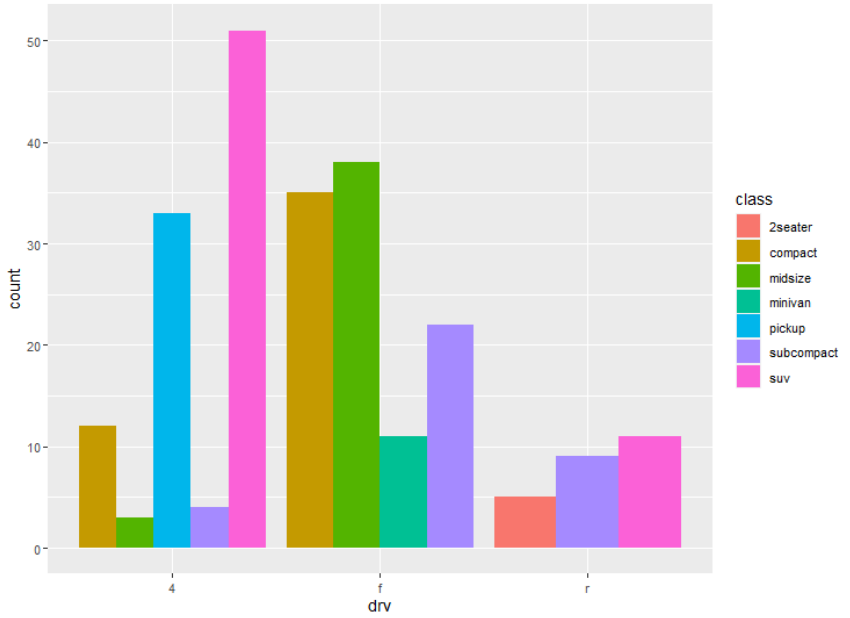
„Stacked“

```
ggplot(mpg, aes(x = drv, fill = class)) +
  geom_bar()
```



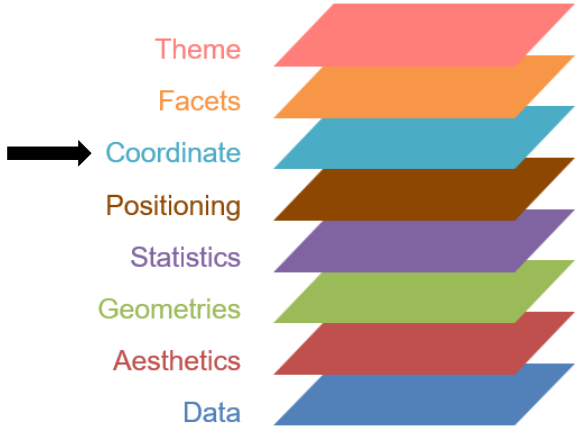
„Dodged“

```
ggplot(mpg, aes(x = drv, fill = class)) +
  geom_bar(position = "dodge")
```



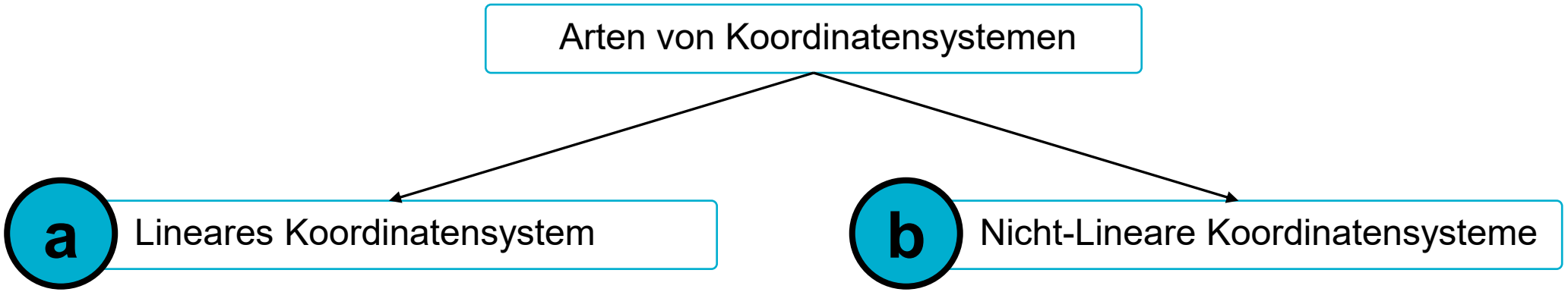
6. Coordinate

[2], [3], [4]



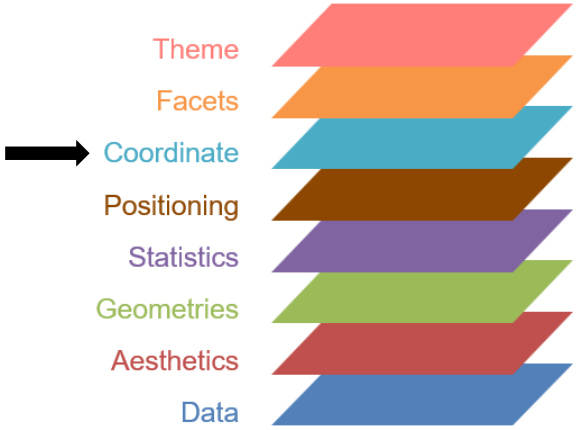
> “A coordinate system maps the position of objects onto the plane of the plot”
>
> [3] Kap. 3.3 S. 13

- Code: `coord_*()`



6. Coordinate

[2], [3], [4]



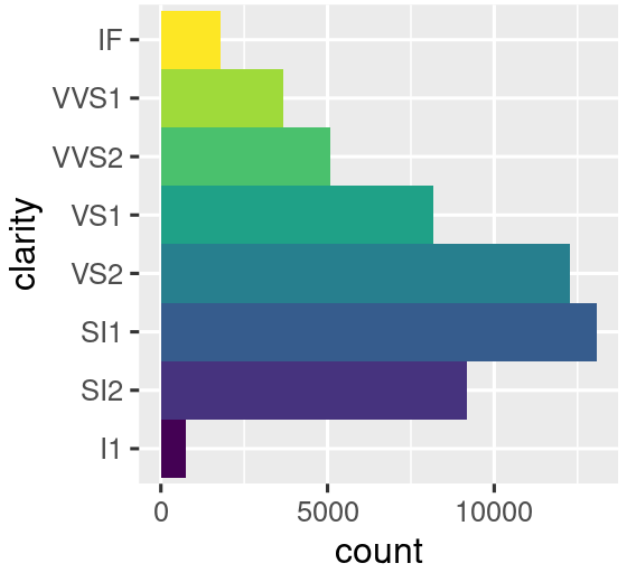
a Lineares Koordinatensystem

- `coord_cartesian()`: kartesisches Koordinatensystem

  ⟶ `coord_flip()`: kartesisches Koordinatensystem mit gespiegelten Achsen

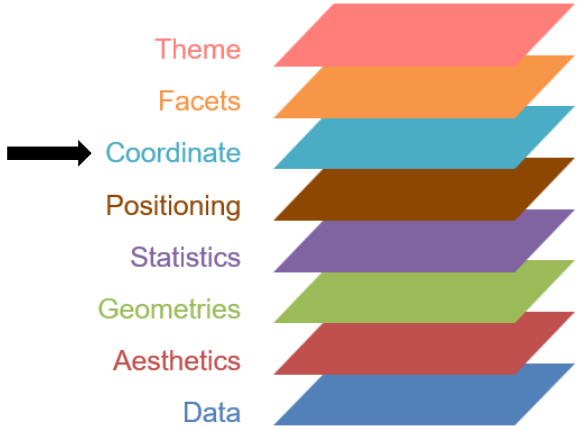
  ⟶ `coord_fixed()`: kartesisches Koordinatensystem mit festem Seitenverhältnis

```
ggplot(diamonds) +
  geom_bar(aes(x = clarity, fill = clarity),
    show.legend = FALSE,
    width = 1) +
  theme(aspect.ratio = 1) +
  coord_flip()
```



6. Coordinate

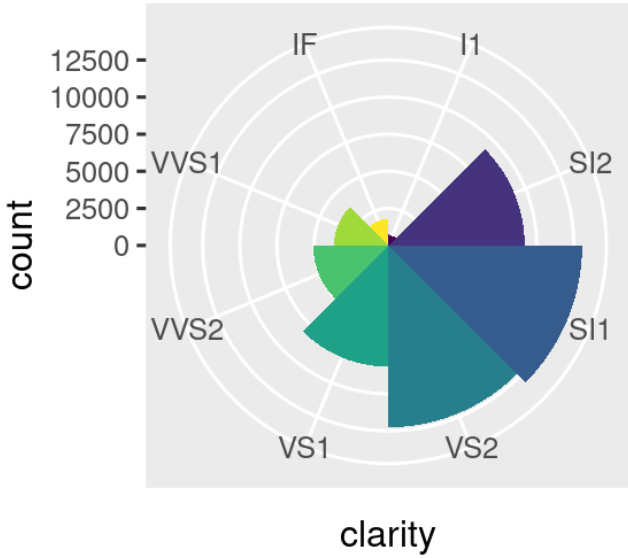
[2], [3], [4]



b Nicht-Lineare Koordinatensysteme

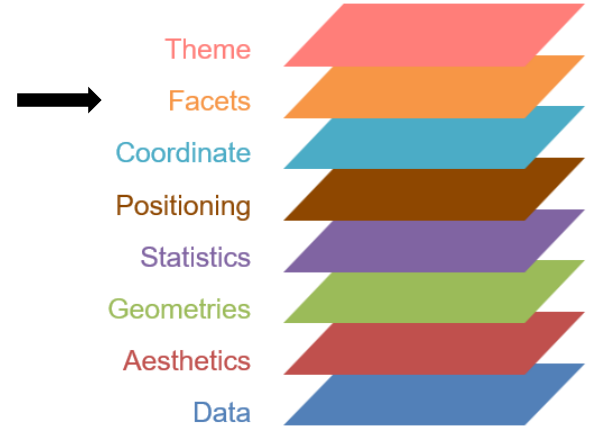
- `coord_polar()`: Polarkoordinaten
- `coord_map()`: Kartenprojektion
- `coord_trans()`: Transformation der x- und y-Positionen nach Datenverarbeitung mit `stat`

```
ggplot(diamonds) +
  geom_bar(aes(x = clarity, fill = clarity),
    show.legend = FALSE,
    width = 1) +
  theme(aspect.ratio = 1) +
  coord_polar()
```



7. Facets

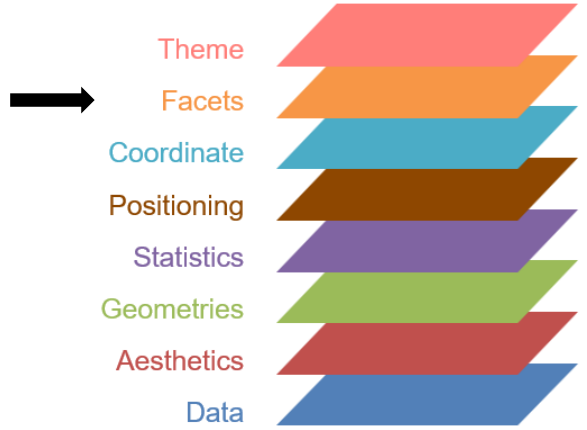
[2], [3], [4]



Aufteilung der Daten anhand einer/mehrerer Variablen in Subplots

➡ Ziel: Identifikation von Mustern/Trends innerhalb der Subplots

- Eigenes Mapping, welches als Formel angegeben wird
- Code: `facet_*()`



7. Facets

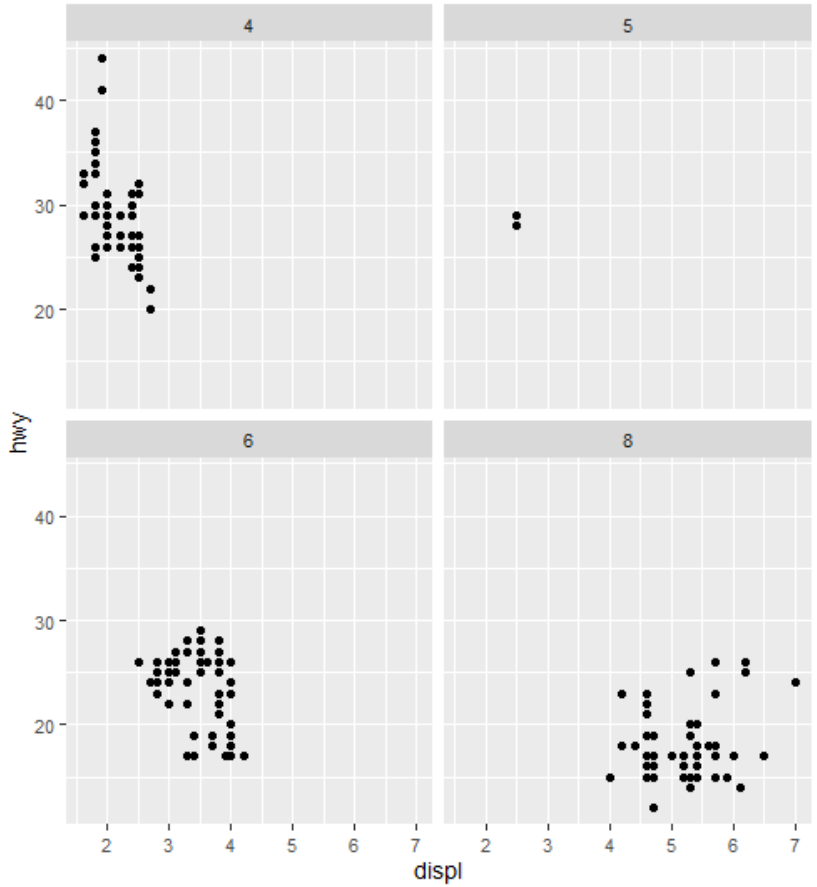
[2], [3], [4]

Arten von `facet`-Funktionen:

(a) `facet_null()`: Einziger Plot

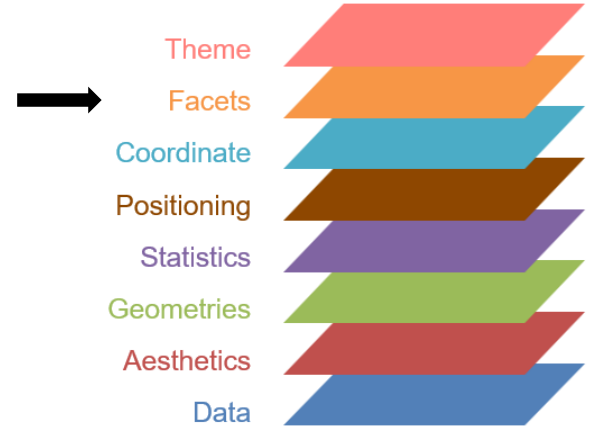
(b) `facet_wrap()`: Unterteilung in Subplots anhand einer kategorialen Variablen

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  facet_wrap(~cyl)
```



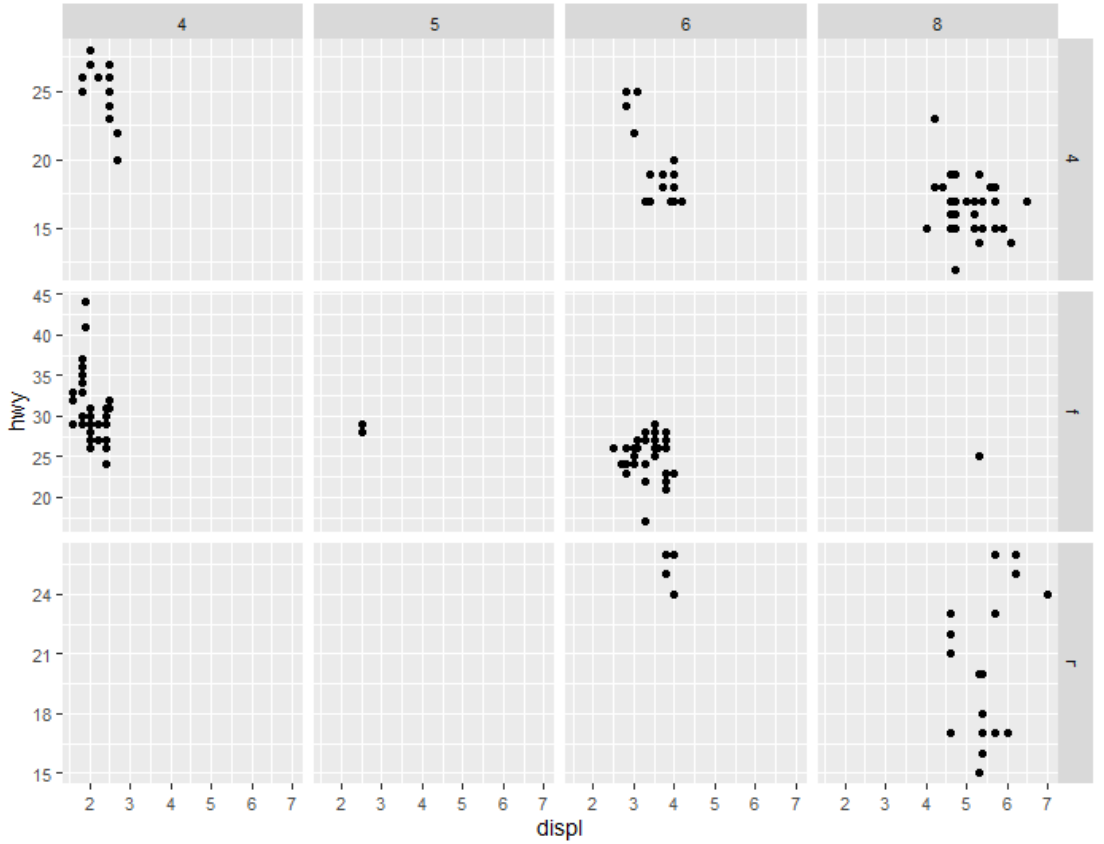
7. Facets

[2], [3], [4]



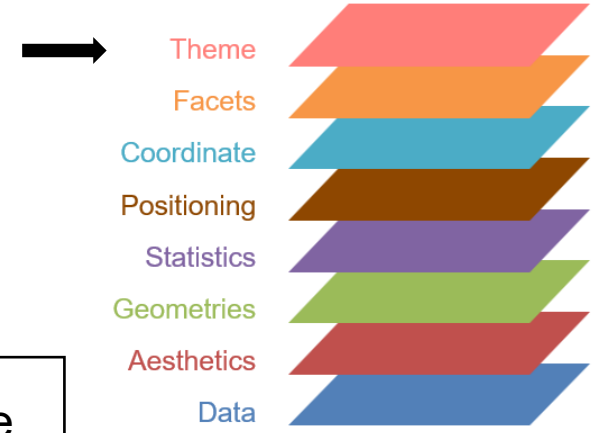
(c) `facet_grid()`: Unterteilung in Subplots anhand von zwei Variablen

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  facet_grid(drv ~ cyl, scales = "free_y")
```



8. Theme

[2], [4]



Theme: System zur Kontrolle der Ästhetik nicht datenbezogener Elemente

Manueller Aufbau des Themes über `theme()`

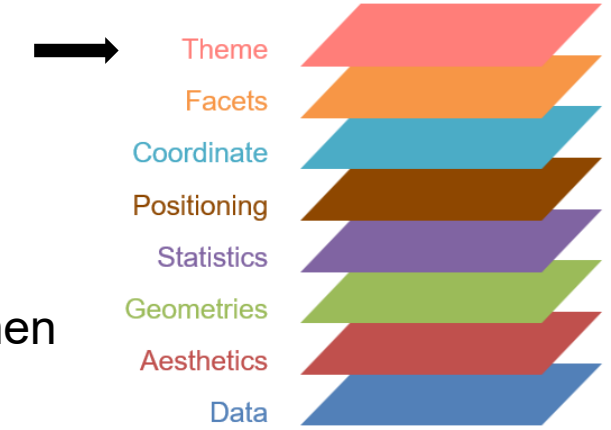
(1) Themenelemente `element.*`

(2) Elementfunktion `element_*()`

(3) Vorgefertigte Themes über `theme_*()`

8. Theme

[2], [4]

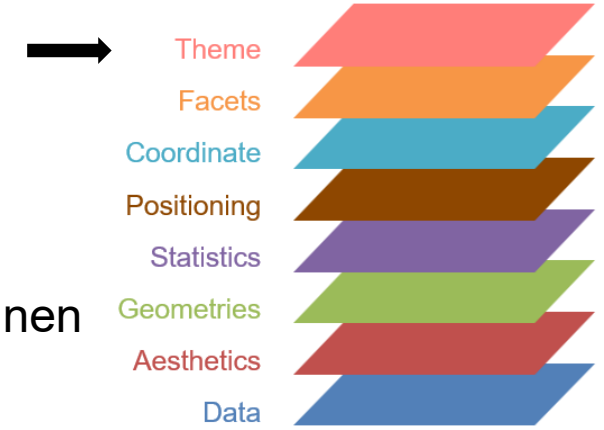


(1) Themenelemente: Elemente, die durch Elementfunktionen geändert werden können

Themenelement	Veränderung…	Veränderungsmöglichkeiten
plot	des ganzen Plots	Hintergrund, Titel, Margen
axis	der Achsen	Linien, Text, Titel, Ticks
legend	der Legenden	Hintergrund, Symbole, Text, Titel & Margen
panel	der Plotfläche	Hintergrund, Rahmen, Gitter & Seitenverhältnis
facet	der Plots bei Facets	Streifenhintergrund, Streifentext, Abstände zwischen Plotflächen

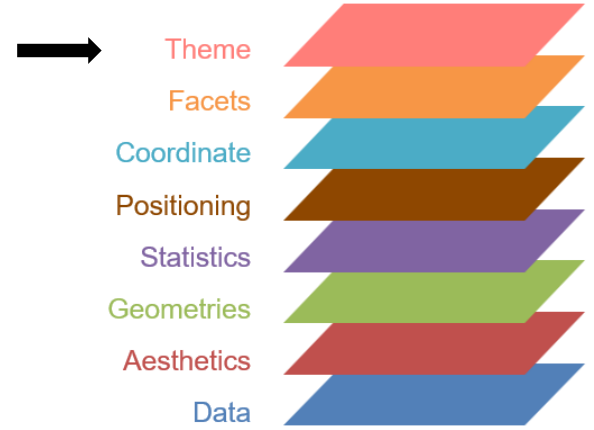
8. Theme

[2], [4]



(2) Elementfunktionen: Veränderungen, die an Elementen vorgenommen werden können

Elementfunktion	Ergebnis	Mögliche Funktionsargumente
`element_rect()`	Rahmen und Hintergründe	`fill, colour, linewidth, linetype, …`
`element_text()`	Text	`family, face, colour, size, angle, …`
`element_line()`	Linien	`colour, linewidth, linetype, …`
`element_blank()`	Entfernen von Themenelementen	keine

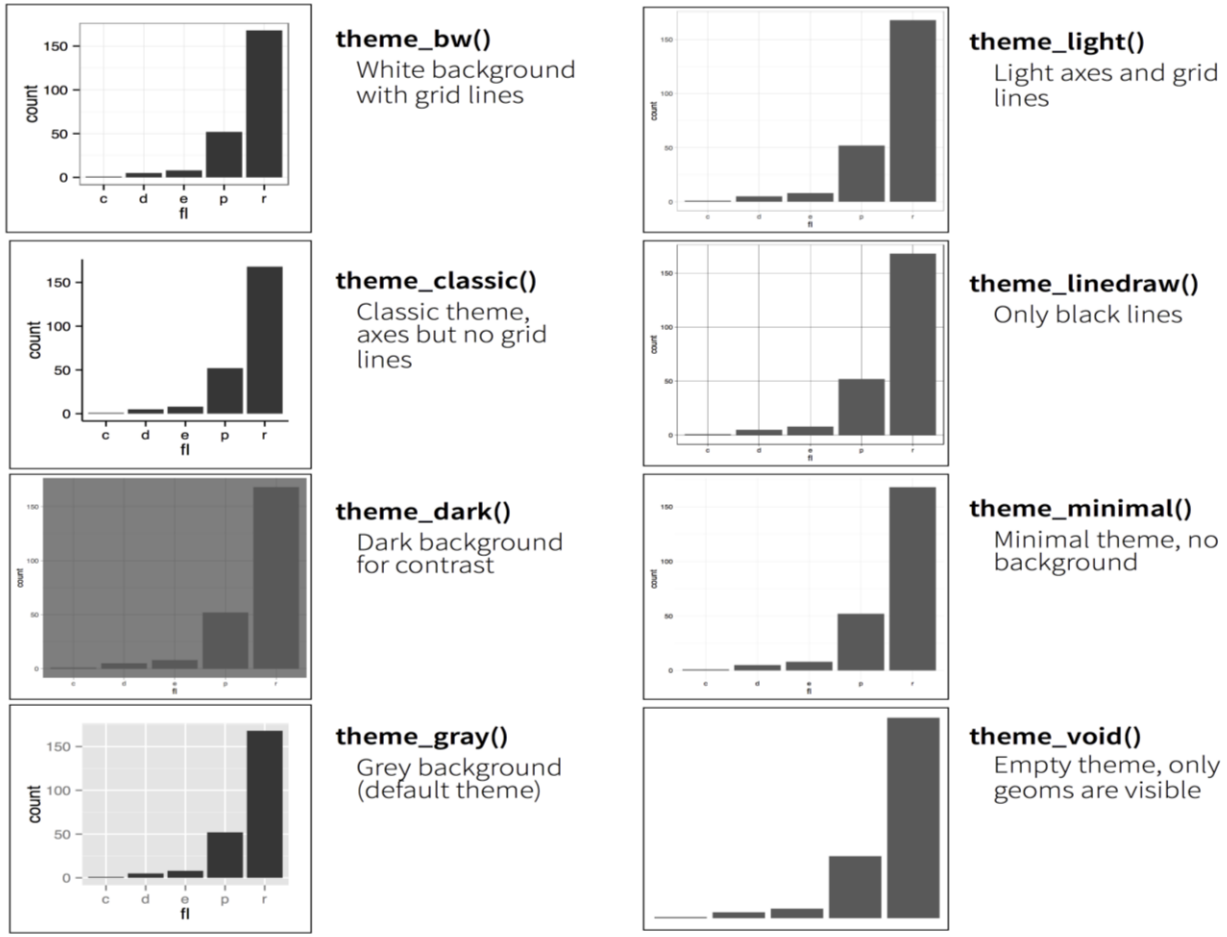


8. Theme

[2], [4]

(3) Vollständige in ggplot2 eingebaute Themen

Unterschiedliche Hintergründe, Gitternetzlinien und Achsen



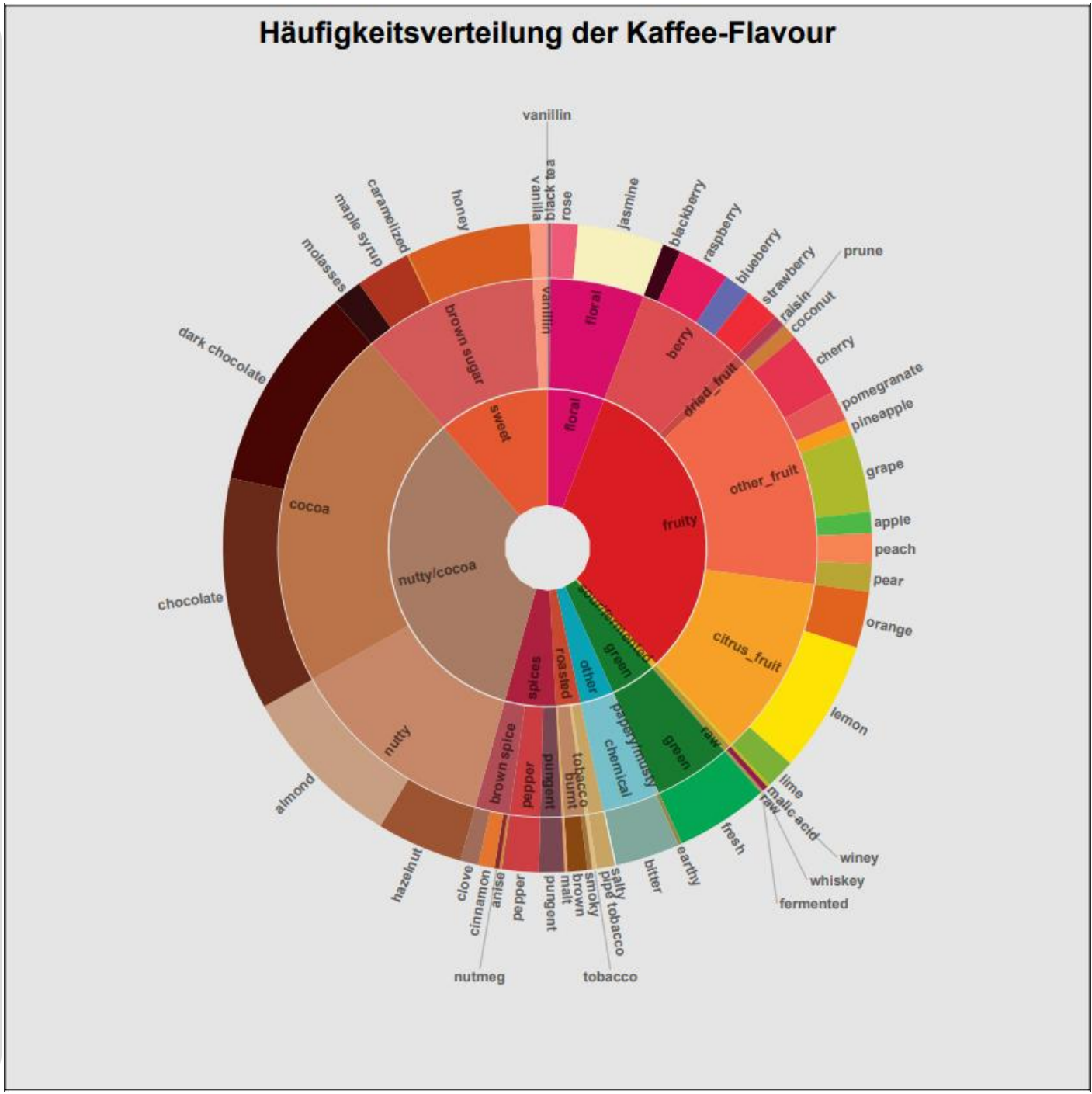
Die Daten

name	roaster	roast	loc_country	origin	100g_USD	rating	review_date	review
Ethiopia Shakiso Mormora	Revel Coffee	Medium-Light	United States	Ethiopia	4.70	92	November 2017	Crisply sweet, cocoa-toned. Lemon blossom, roasted cacao (…)
Kenya Kirinyaga Mukangu AB	Kakalove Cafe	Medium-Light	Taiwan	Kenya	4.60	95	November 2017	Crisply sweet-savory. Lavender, roasted cacao nib, strawberry (…)

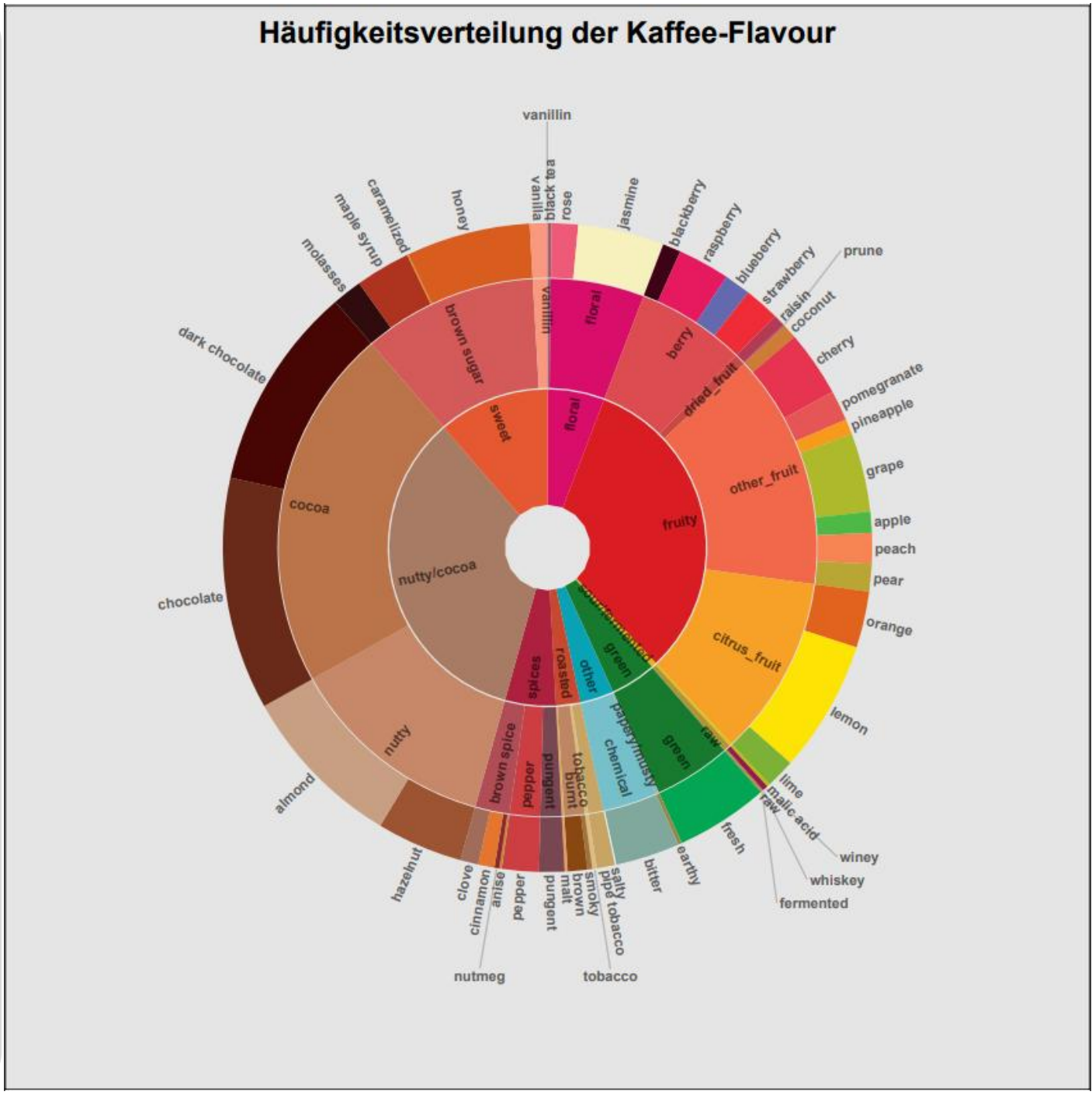
[6]

Das Flavour-Wheel der SCA

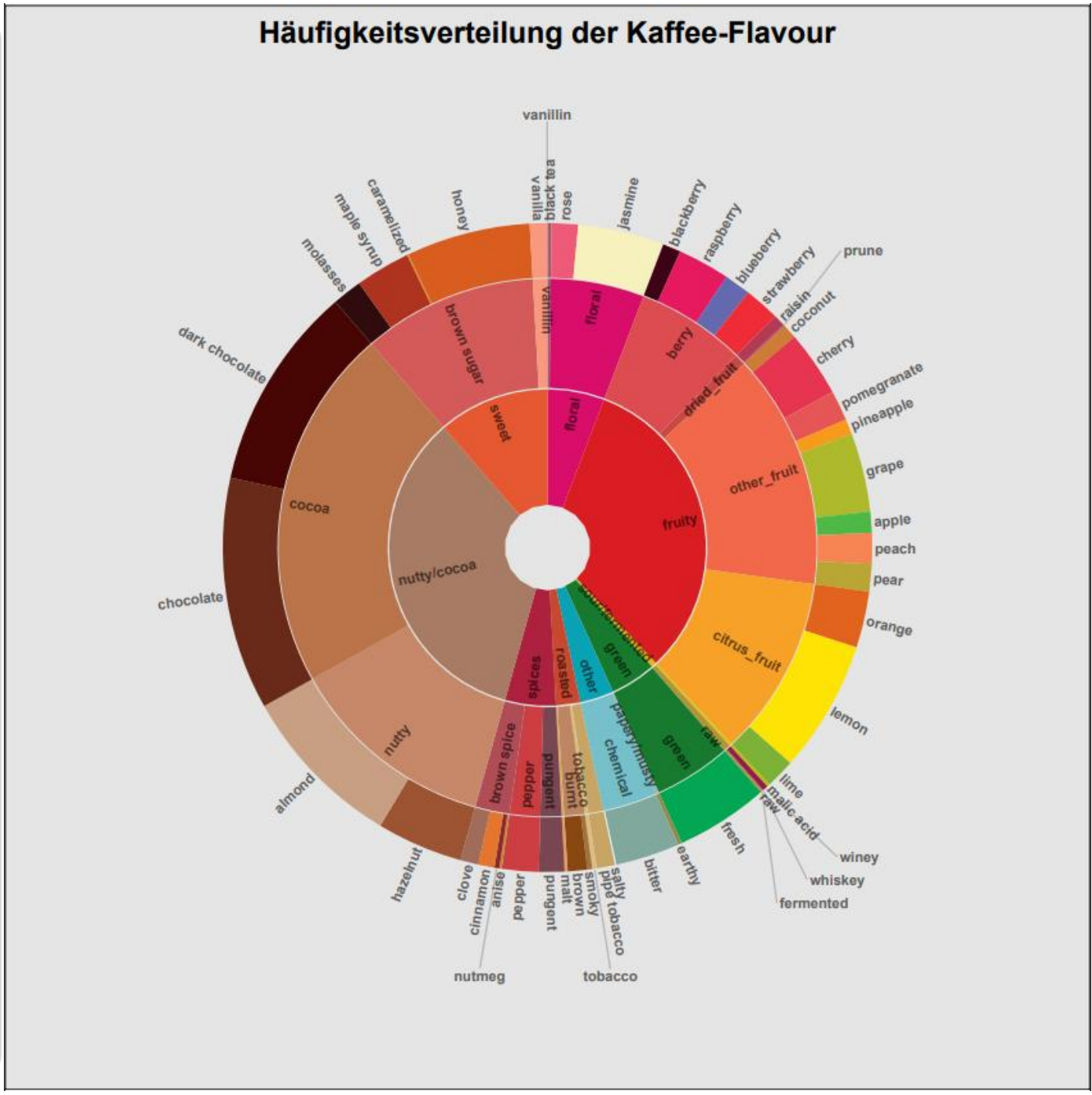
```
ggplot() +
  geom_bar(
    data = flavours_H, mapping = aes(x = 5,
    y = Freq, fill = Var1),
    position = "stack", stat = "identity",
    width = 1
    ) +
  geom_bar(
    data = mid_group_H,
    mapping = aes(x = 3.48, y = Freq, fill =
        mid_group),
    position = "stack", stat = "identity",
    width = 2
    ) +
  geom_bar(
    data = inner_group_H,
    mapping = aes(x = 1, y = Freq,
    fill = inner_group),
    position = "stack", stat = "identity",
    width = 2.9
    ) +
  geom_col(
    data = white_circle,
    mapping = aes(x = x, y = y),
    fill = "gray90", width = 0.67
    ) +
  scale_fill_manual(values = scale_color_SCA)+
```

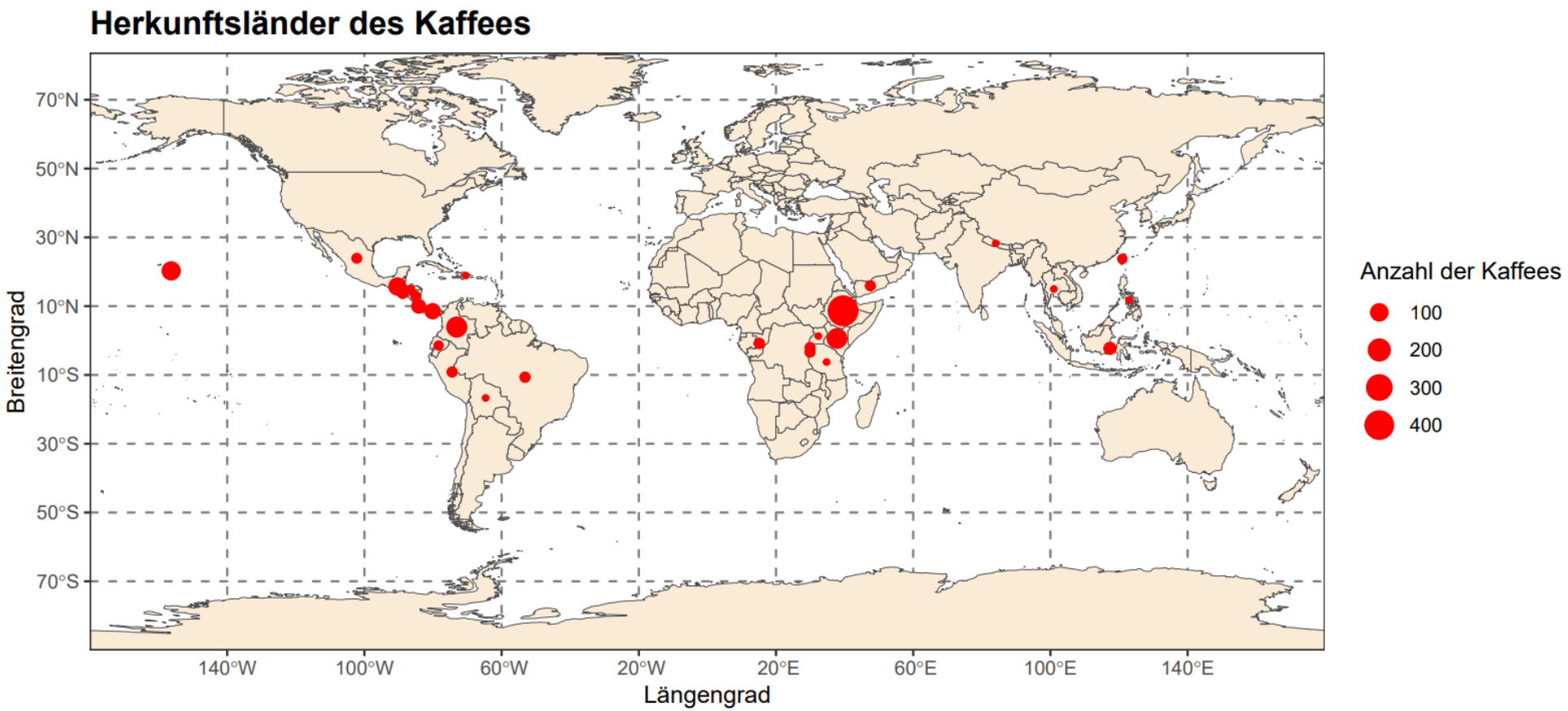


```
geom_text(
    data=label_data_outer,
    aes(x = 5.55, y = Freq_position,
        label= Var1, angle = angle),
    color = "black", fontface = "bold",
    alpha = 0.6, size = 3,
    position = position_identity(),
    hjust = "outward"
    ) +
geom_text(
    data = label_data_mid,
    mapping = aes(x = 4.35, y = Freq_position,
    label = mid_group, angle = angle),
    color = "black", fontface = "bold",
    alpha = 0.6, size = 3,
    position = position_identity(),
    hjust = "inward",check_overlap = T
    ) +
geom_text(
    data = label_data_inner,
    mapping = aes(x=2.35, y = Freq_position,
    label = inner_group, angle = angle),
    color = "black", fontface = "bold",
    alpha = 0.6, size = 3,
    position = position_identity(),
    hjust = "inward"
    ) +
```



```
geom_text_repel(
    data=label_data_removed,
    mapping = aes(x=5.5, y= Freq_position,
    label= Var1),
    color="black",
    fontface="bold",alpha=0.6, size=3,
    min.segment.length = 0,
    nudge_x = 2, segment.alpha = 0.2
    ) +
coord_polar(theta = "y", direction = -1) +
theme_void() +
theme(
    legend.position = "none",
    panel.background =
       element_rect(fill = "gray90",
       color = "black"),
    plot.title =
       element_text(hjust = 0.5,
       vjust = -6, size = 18,
       color = "black", face = "bold")
  ) +
ggtitle("Häufigkeitsverteilung der Kaffee-
Flavour")
```

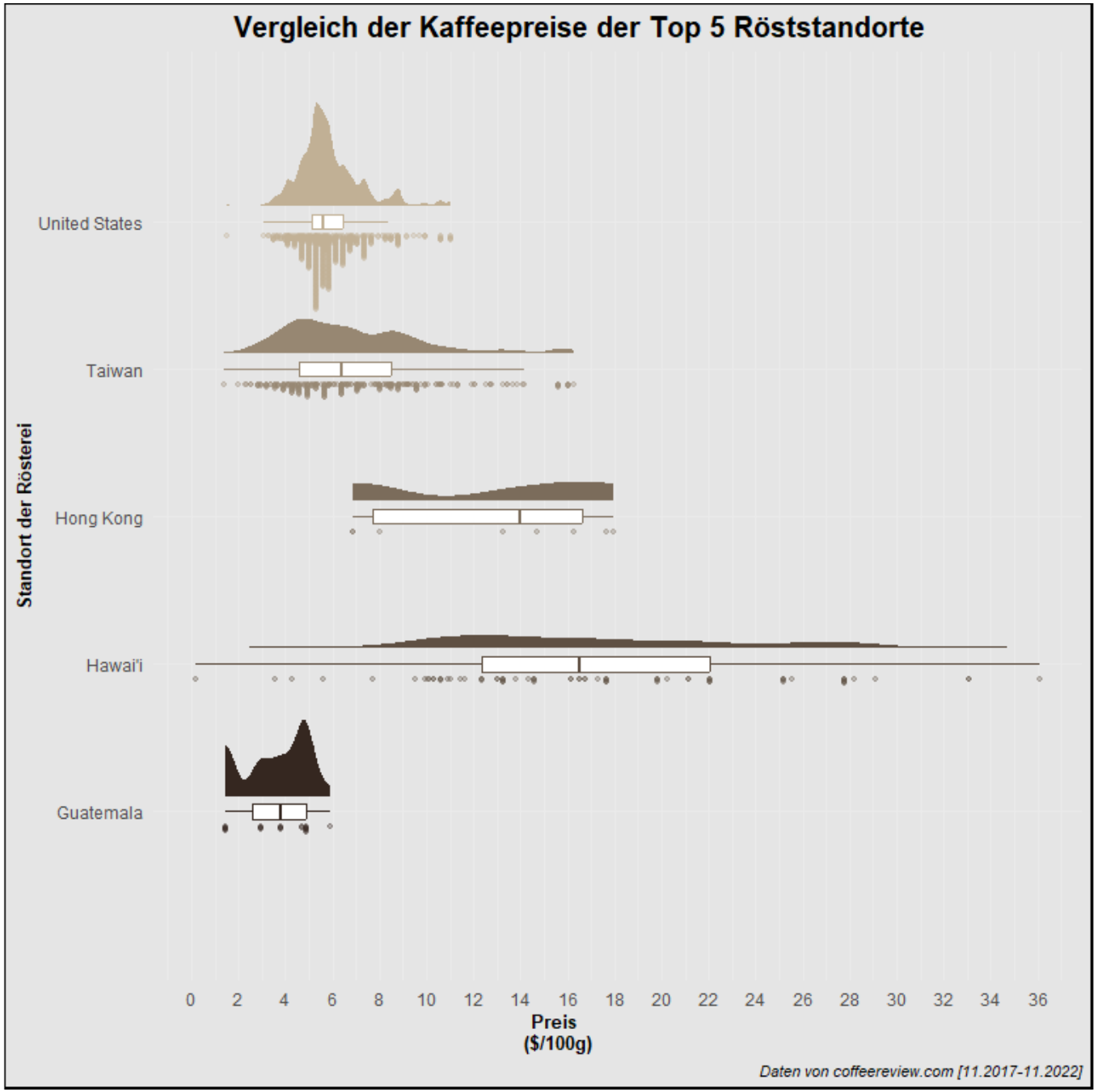




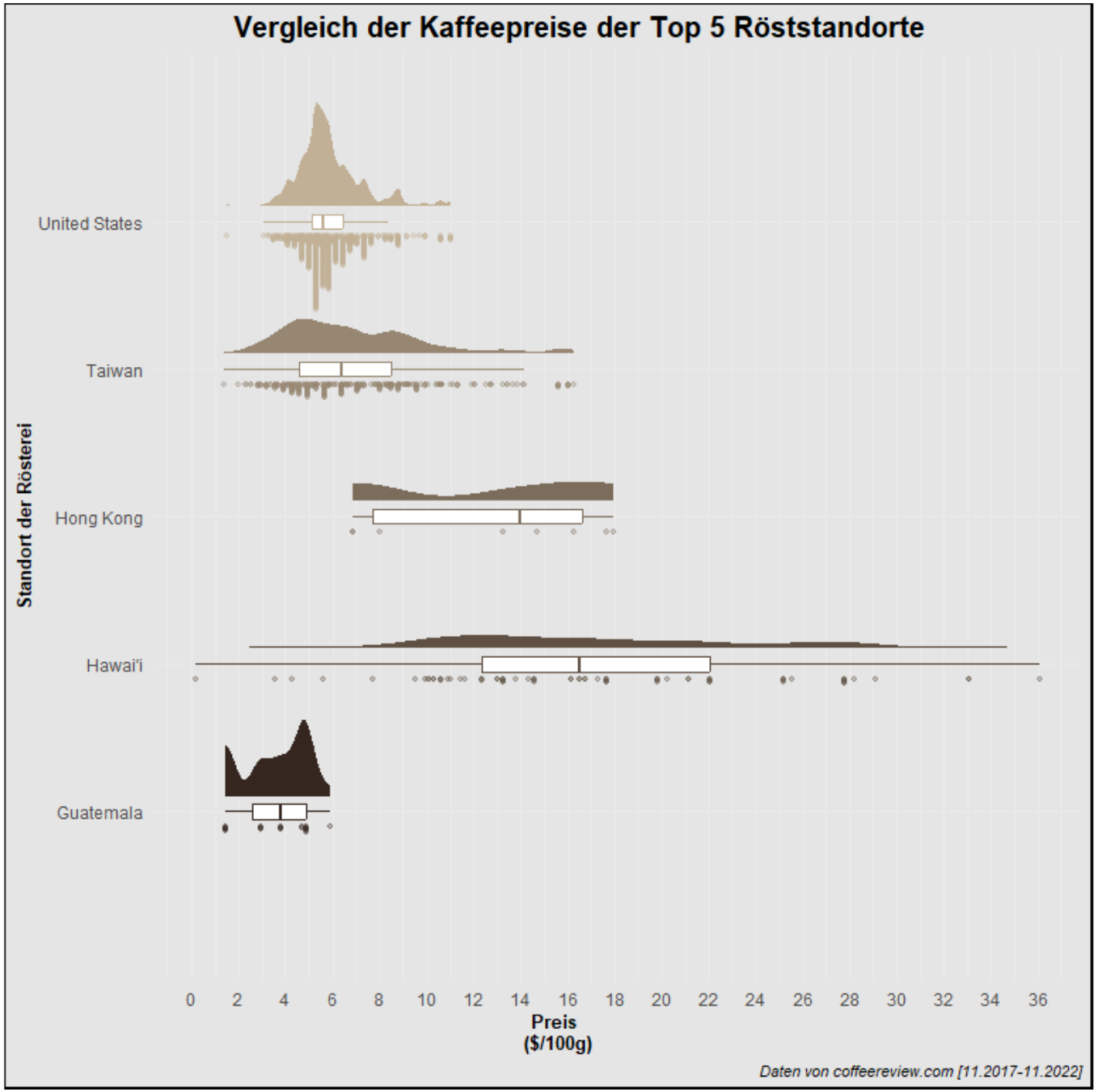

```
ggplot() +
geom_sf(data = world, colour = "grey30",
        fill = "antiquewhite") +
geom_point(data = origin_H_sf,
           mapping = aes(x = lon, y = lat, size =
Freq),
           colour = "red") +
coord_sf() +
labs(x = "Längengrad", y = "Breitengrad",
     size = "Anzahl der Kaffees",
     title = "Herkunftsländer des Kaffees") +
```

```
scale_x_continuous(expand = c(0, 0),
                   breaks = seq(-180,180, 40)) +
scale_y_continuous(expand = c(0, 0),
                   breaks = seq(-90, 90, 20)) +
theme_bw() +
theme(panel.grid.major = element_line(color =
gray(.5), linetype = "dashed", linewidth = 0.5),
plot.title = element_text(face = "bold", size =
15))
```

```
coffee_outlier_rm |>
ggplot(aes(x = loc_country , y = price, color
= loc_country)) +

ggdist::stat_halfeye(aes(fill = loc_country),
adjust = 0.6,
justification = -0.15, scale = 0.7,.width = 0,
point_colour = NA) +

geom_boxplot(width = 0.1,
outlier.color = NA) +

ggdist::stat_dots(aes(fill = loc_country),
alpha = 0.3, binwidth = 0.3, scale = 0.5,
side = "left", justification = 1.15,
position = position_dodge(),
overflow = "compress") +

scale_fill_manual(values = c_scale) +

scale_colour_manual(values = c_scale) +
```



```
labs(title = "Vergleich der Kaffeepreise der
Top 5 Röststandorte", x = "Standort der
Rösterei",    y = "Preis \n($/100g)",
caption = "Daten von coffeereview.com
[11.2017-11.2022]") +

coord_flip() +

scale_y_continuous(breaks = seq(0, 40, 2)) +

theme_minimal() +

theme(legend.position = "none",
aspect.ratio = 1,plot.background =
element_rect(colour = "black",
linewidth = 1, fill = "grey90"),
plot.title = element_text(face = "bold", size
= 15, hjust = 0.34), axis.title =
element_text(face = "bold", size =10),
axis.title.x = element_text(hjust = 0.43),
plot.caption = element_text(face = "italic",
size = 8))
```



Literaturquellen

[1] Wickham, H. (2014). Tidy data. Journal of statistical software, 59, 1-23.

[2] Wickham, H., Çetinkaya-Rundel, M., & Grolemund, G. (2023). R for data science. " O'Reilly Media, Inc.".

[3] Wickham, H. (2010). A layered grammar of graphics. Journal of computational and graphical statistics, 19(1), 3-28.

[4] Wickham H (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York.

[5] Wilkinson, L. (2012). The grammar of graphics (pp. 1-11). Springer Berlin Heidelberg.

[6] Spencer, M., Sage, E., Velez, M., & Guinard, J. X. (2016). Using single free sorting and multivariate exploratory methods to design a new coffee taster's flavor wheel. *Journal of food science*, *81*(12), S2997-S3005.

[7] Anscombe, F. J. (1973). Graphs in Statistical Analysis. The American Statistician, 27(1), 17-21.

Vielen Dank für Ihre Aufmerksamkeit

Universität Augsburg
Wirtschaftswissenschaftliche
Fakultät

Betreuer: Linus Lach

Lehrstuhl für Statistik, Prof. Dr. Yarema Okhrin

Universität Augsburg

Anhang

Data tidying – Probleme bei messy datasets [1]

a Spaltenüberschriften als Werte (keine Variablennamen)

b Speicherung von mehreren Variablen in einer Spalte

c Speicherung von Variablen sowohl in Zeilen als auch in Spalten

d Speicherung von mehreren Beobachtungseinheiten in derselben Tabelle

e Speicherung von einer Beobachtungseinheit in mehreren Tabellen

Data tidying

[1]

a Spaltenüberschriften als Werte (keine Variablennamen)

Messy Dataset

year	artist	track	time	date.entered	wk1	wk2	wk3
2000	2 Pac	Baby Don't Cry	4:22	2000-02-26	87	82	72
2000	2Ge+her	The Hardest Part Of ...	3:15	2000-09-02	91	87	92
2000	3 Doors Down	Kryptonite	3:53	2000-04-08	81	70	68
2000	98^0	Give Me Just One Nig...	3:24	2000-08-19	51	39	34
2000	A*Teens	Dancing Queen	3:44	2000-07-08	97	97	96
2000	Aaliyah	I Don't Wanna	4:15	2000-01-29	84	62	51
2000	Aaliyah	Try Again	4:03	2000-03-18	59	53	38
2000	Adams, Yolanda	Open My Heart	5:30	2000-08-26	76	76	74

Table 7: The first eight Billboard top hits for 2000. Other columns not shown are `wk4`, `wk5`, ..., `wk75`.

Molten Dataset

year	artist	time	track	date	week	rank
2000	2 Pac	4:22	Baby Don't Cry	2000-02-26	1	87
2000	2 Pac	4:22	Baby Don't Cry	2000-03-04	2	82
2000	2 Pac	4:22	Baby Don't Cry	2000-03-11	3	72
2000	2 Pac	4:22	Baby Don't Cry	2000-03-18	4	77
2000	2 Pac	4:22	Baby Don't Cry	2000-03-25	5	87
2000	2 Pac	4:22	Baby Don't Cry	2000-04-01	6	94
2000	2 Pac	4:22	Baby Don't Cry	2000-04-08	7	99
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-02	1	91
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-09	2	87
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-16	3	92
2000	3 Doors Down	3:53	Kryptonite	2000-04-08	1	81
2000	3 Doors Down	3:53	Kryptonite	2000-04-15	2	70
2000	3 Doors Down	3:53	Kryptonite	2000-04-22	3	68
2000	3 Doors Down	3:53	Kryptonite	2000-04-29	4	67
2000	3 Doors Down	3:53	Kryptonite	2000-05-06	5	66

Table 8: First fifteen rows of the tidied billboard dataset. The `date` column does not appear in the original table, but can be computed from `date.entered` and `week`.

Data tidying

[1]

b Speicherung von mehreren Variablen in einer Spalte

Messy Dataset

country	year	m014	m1524	m2534	m3544	m4554	m5564	m65	mu	f014
AD	2000	0	0	1	0	0	0	0	—	—
AE	2000	2	4	4	6	5	12	10	—	3
AF	2000	52	228	183	149	129	94	80	—	93
AG	2000	0	0	0	0	0	0	1	—	1
AL	2000	2	19	21	14	24	19	16	—	3
AM	2000	2	152	130	131	63	26	21	—	1
AN	2000	0	0	1	2	0	0	0	—	0
AO	2000	186	999	1003	912	482	312	194	—	247
AR	2000	97	278	594	402	419	368	330	—	121
AS	2000	—	—	—	—	1	1	—	—	—

Table 9: Original TB dataset. Corresponding to each 'm' column for males, there is also an 'f' column for females, `f1524`, `f2534` and so on. These are not shown to conserve space. Note the mixture of 0s and missing values (—). This is due to the data collection process and the distinction is important for this dataset.

Molten Dataset

country	year	column	cases
AD	2000	m014	0
AD	2000	m1524	0
AD	2000	m2534	1
AD	2000	m3544	0
AD	2000	m4554	0
AD	2000	m5564	0
AD	2000	m65	0
AE	2000	m014	2
AE	2000	m1524	4
AE	2000	m2534	4
AE	2000	m3544	6
AE	2000	m4554	5
AE	2000	m5564	12
AE	2000	m65	10
AE	2000	f014	3

(a) Molten data

Tidy Dataset

country	year	sex	age	cases
AD	2000	m	0-14	0
AD	2000	m	15-24	0
AD	2000	m	25-34	1
AD	2000	m	35-44	0
AD	2000	m	45-54	0
AD	2000	m	55-64	0
AD	2000	m	65+	0
AE	2000	m	0-14	2
AE	2000	m	15-24	4
AE	2000	m	25-34	4
AE	2000	m	35-44	6
AE	2000	m	45-54	5
AE	2000	m	55-64	12
AE	2000	m	65+	10
AE	2000	f	0-14	3

(b) Tidy data

Table 10: Tidying the TB dataset requires first melting, and then splitting the `column` column into two variables: `sex` and `age`.

Data tidying

[1]

c Speicherung von Variablen sowohl in Zeilen als auch in Spalten

Messy Dataset

id	year	month	element	d1	d2	d3	d4	d5	d6	d7	d8
MX17004	2010	1	tmax	—	—	—	—	—	—	—	—
MX17004	2010	1	tmin	—	—	—	—	—	—	—	—
MX17004	2010	2	tmax	—	27.3	24.1	—	—	—	—	—
MX17004	2010	2	tmin	—	14.4	14.4	—	—	—	—	—
MX17004	2010	3	tmax	—	—	—	—	32.1	—	—	—
MX17004	2010	3	tmin	—	—	—	—	14.2	—	—	—
MX17004	2010	4	tmax	—	—	—	—	—	—	—	—
MX17004	2010	4	tmin	—	—	—	—	—	—	—	—
MX17004	2010	5	tmax	—	—	—	—	—	—	—	—
MX17004	2010	5	tmin	—	—	—	—	—	—	—	—

Table 11: Original weather dataset. There is a column for each possible day in the month. Columns `d9` to `d31` have been omitted to conserve space.

Molten Dataset

id	date	element	value
MX17004	2010-01-30	tmax	27.8
MX17004	2010-01-30	tmin	14.5
MX17004	2010-02-02	tmax	27.3
MX17004	2010-02-02	tmin	14.4
MX17004	2010-02-03	tmax	24.1
MX17004	2010-02-03	tmin	14.4
MX17004	2010-02-11	tmax	29.7
MX17004	2010-02-11	tmin	13.4
MX17004	2010-02-23	tmax	29.9
MX17004	2010-02-23	tmin	10.7

(a) Molten data

Tidy Dataset

id	date	tmax	tmin
MX17004	2010-01-30	27.8	14.5
MX17004	2010-02-02	27.3	14.4
MX17004	2010-02-03	24.1	14.4
MX17004	2010-02-11	29.7	13.4
MX17004	2010-02-23	29.9	10.7
MX17004	2010-03-05	32.1	14.2
MX17004	2010-03-10	34.5	16.8
MX17004	2010-03-16	31.1	17.6
MX17004	2010-04-27	36.3	16.7
MX17004	2010-05-27	33.2	18.2

(b) Tidy data

Table 12: (a) Molten weather dataset. This is almost tidy, but instead of values, the `element` column contains names of variables. Missing values are dropped to conserve space. (b) Tidy weather dataset. Each row represents the meteorological measurements for a single day. There are two measured variables, minimum (`tmin`) and maximum (`tmax`) temperature; all other variables are fixed.

Data tidying

[1]

d Speicherung von mehreren Beobachtungseinheiten in derselben Tabelle

Molten Dataset

year	artist	time	track	date	week	rank
2000	2 Pac	4:22	Baby Don't Cry	2000-02-26	1	87
2000	2 Pac	4:22	Baby Don't Cry	2000-03-04	2	82
2000	2 Pac	4:22	Baby Don't Cry	2000-03-11	3	72
2000	2 Pac	4:22	Baby Don't Cry	2000-03-18	4	77
2000	2 Pac	4:22	Baby Don't Cry	2000-03-25	5	87
2000	2 Pac	4:22	Baby Don't Cry	2000-04-01	6	94
2000	2 Pac	4:22	Baby Don't Cry	2000-04-08	7	99
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-02	1	91
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-09	2	87
2000	2Ge+her	3:15	The Hardest Part Of ...	2000-09-16	3	92
2000	3 Doors Down	3:53	Kryptonite	2000-04-08	1	81
2000	3 Doors Down	3:53	Kryptonite	2000-04-15	2	70
2000	3 Doors Down	3:53	Kryptonite	2000-04-22	3	68
2000	3 Doors Down	3:53	Kryptonite	2000-04-29	4	67
2000	3 Doors Down	3:53	Kryptonite	2000-05-06	5	66

Table 8

Tidy Dataset

id	artist	track	time
1	2 Pac	Baby Don't Cry	4:22
2	2Ge+her	The Hardest Part Of ...	3:15
3	3 Doors Down	Kryptonite	3:53
4	3 Doors Down	Loser	4:24
5	504 Boyz	Wobble Wobble	3:35
6	98^0	Give Me Just One Nig...	3:24
7	A*Teens	Dancing Queen	3:44
8	Aaliyah	I Don't Wanna	4:15
9	Aaliyah	Try Again	4:03
10	Adams, Yolanda	Open My Heart	5:30
11	Adkins, Trace	More	3:05
12	Aguilera, Christina	Come On Over Baby	3:38
13	Aguilera, Christina	I Turn To You	4:00
14	Aguilera, Christina	What A Girl Wants	3:18
15	Alice Deejay	Better Off Alone	6:50

Tidy Dataset

id	date	rank
1	2000-02-26	87
1	2000-03-04	82
1	2000-03-11	72
1	2000-03-18	77
1	2000-03-25	87
1	2000-04-01	94
1	2000-04-08	99
2	2000-09-02	91
2	2000-09-09	87
2	2000-09-16	92
3	2000-04-08	81
3	2000-04-15	70
3	2000-04-22	68
3	2000-04-29	67
3	2000-05-06	66

Table 13: Normalised billboard dataset split up into song dataset (left) and rank dataset (right). First 15 rows of each dataset shown; `genre` omitted from song dataset, `week` omitted from rank dataset.

Data tidying

[1]

e Speicherung von einer Beobachtungseinheit in mehreren Tabellen

- Dateien werden durch eine Variable aufgeteilt
- Lösung mit dem Paket **plyr** möglich, falls Format der einzelnen Datensätze konsistent ist:
  i. Dateien in einer Liste von Tabellen einlesen
  ii. Für jede Tabelle eine neue Spalte, die den ursprünglichen Dateiname aufzeichnet, hinzufügen
  iii. Alle Tabellen in einer einzigen Tabelle kombinieren

Tidy tools – drei wichtige Komponenten der Analyse [1]

1. Manipulation
2. Visualisation
3. Modelling

Tidy tools

[1]

1 Manipulation

- Arten: Aufteilung, Transformation, Aggregation und Umsortierung
- Ausführung durch Funktionen aus **base R** oder Funktionen aus dem Paket **plyr**

	Input	Output
base R	tidy	funktionsabhängig tidy
plyr	tidy	tidy

Die wichtigsten Funktionen des plyr Pakets [2]

Spalten

- `mutate()`
- `select()`
- `rename()`

Allgemein

- Pipe Operator `|>`
- `group()`

Zeilen

- `filter()`
- `arrange()`
- `distinct()`

Tidy tools

[1]

2 Visualisation

- „Mapping“ von Variablen auf ästhetischen Eigenschaften des Graphen
- Ausführung durch Funktionen aus **base R** oder Funktionen aus dem Paket **ggplot2**

	Input	Output
base R	tidy / messy	visual
ggplot2	tidy	visual

Tidy tools

[1]

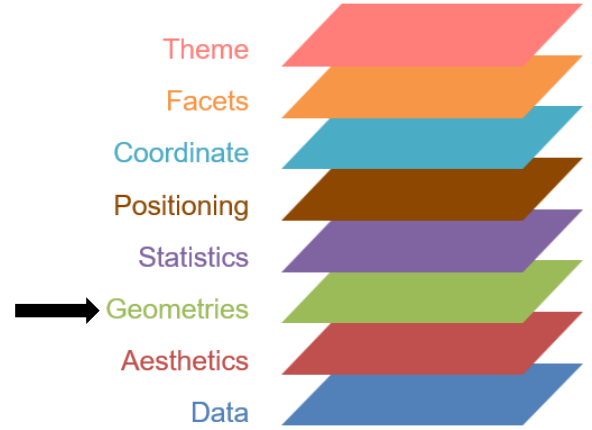
3 Modelling

- Verbindung zwischen verschiedenen Variablen
- Beispiel: `R(lm()): y ~ a + b + c * d`

	Input	Output
base R	tidy	funktionsabhängig tidy

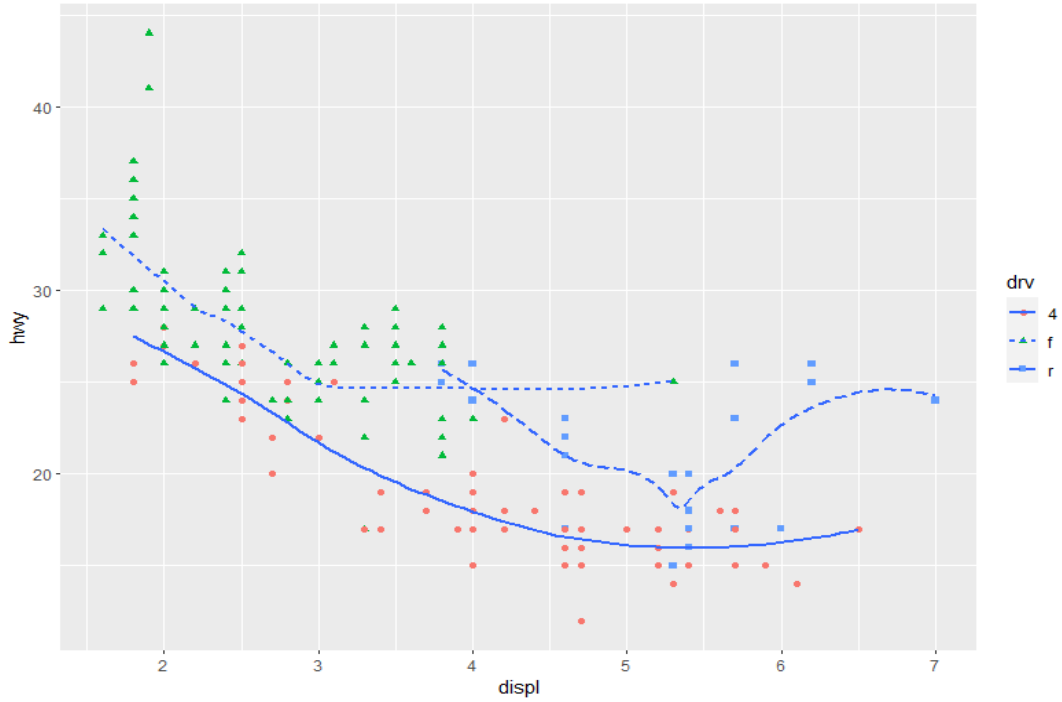
3. Geometries

[2], [3], [4]



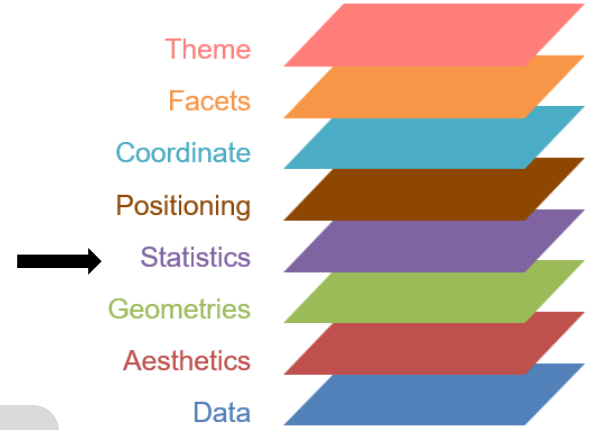
- Jede geom-Funktion benötigt ein Mapping-Argument (lokal/ global)
- Jede geom-Funktion kann nur bestimmte Ästhetiken darstellen

```
ggplot(mpg, aes(x = displ, y = hwy,
shape = drv, linetype = drv)) +
  geom_point(aes(color = drv)) +
  geom_smooth(se = FALSE)
```



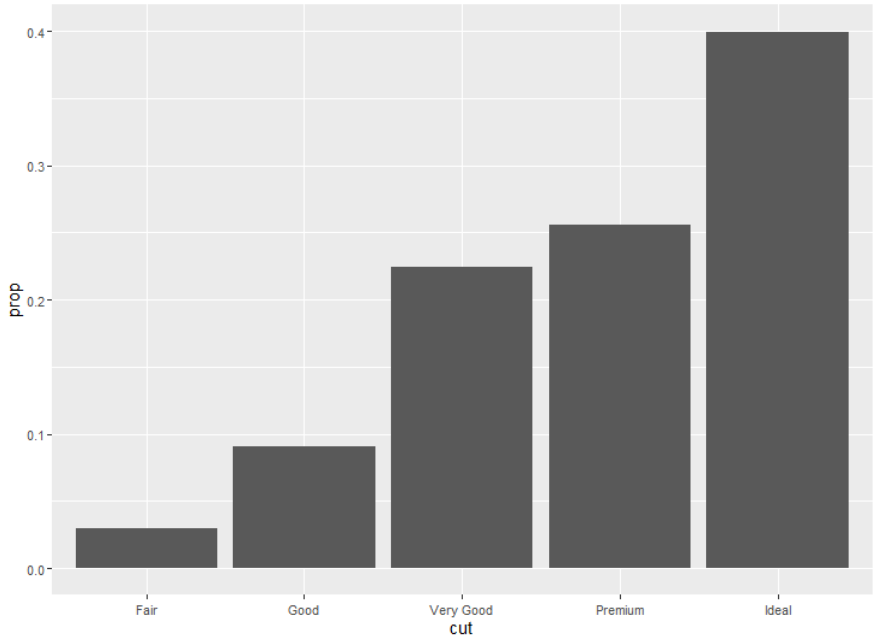
4. Statistics

[2], [3], [4]



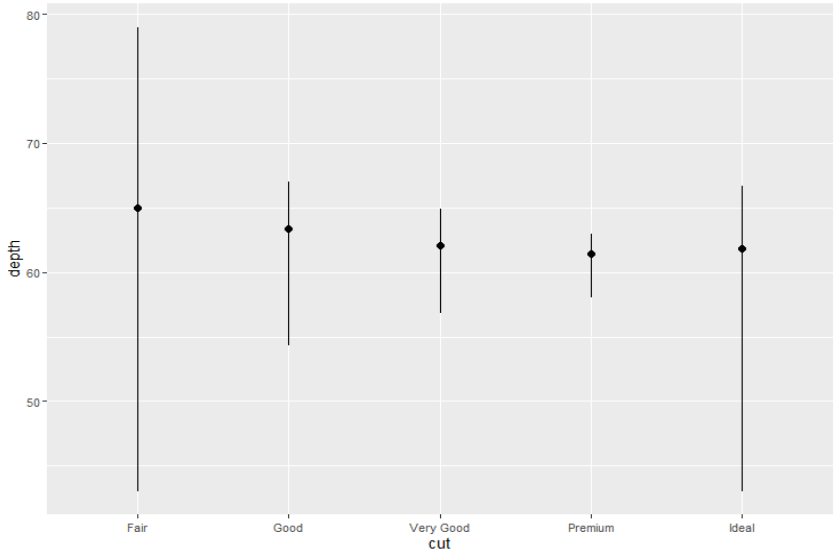
Überschreibung von Standards

```
ggplot(diamonds, aes(x = cut,
y =after_stat(prop), group = 1)) +
  geom_bar()
```



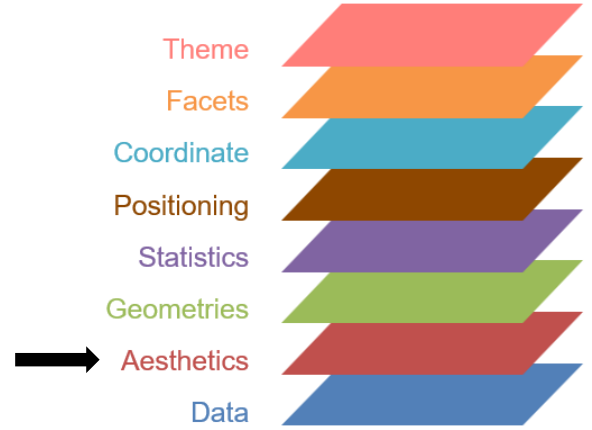
Darstellung statistischer Kennzahlen

```
ggplot(diamonds) +
  stat_summary(
    aes(x = cut, y = depth),
    fun.min = min,
    fun.max = max,
    fun = median)
```



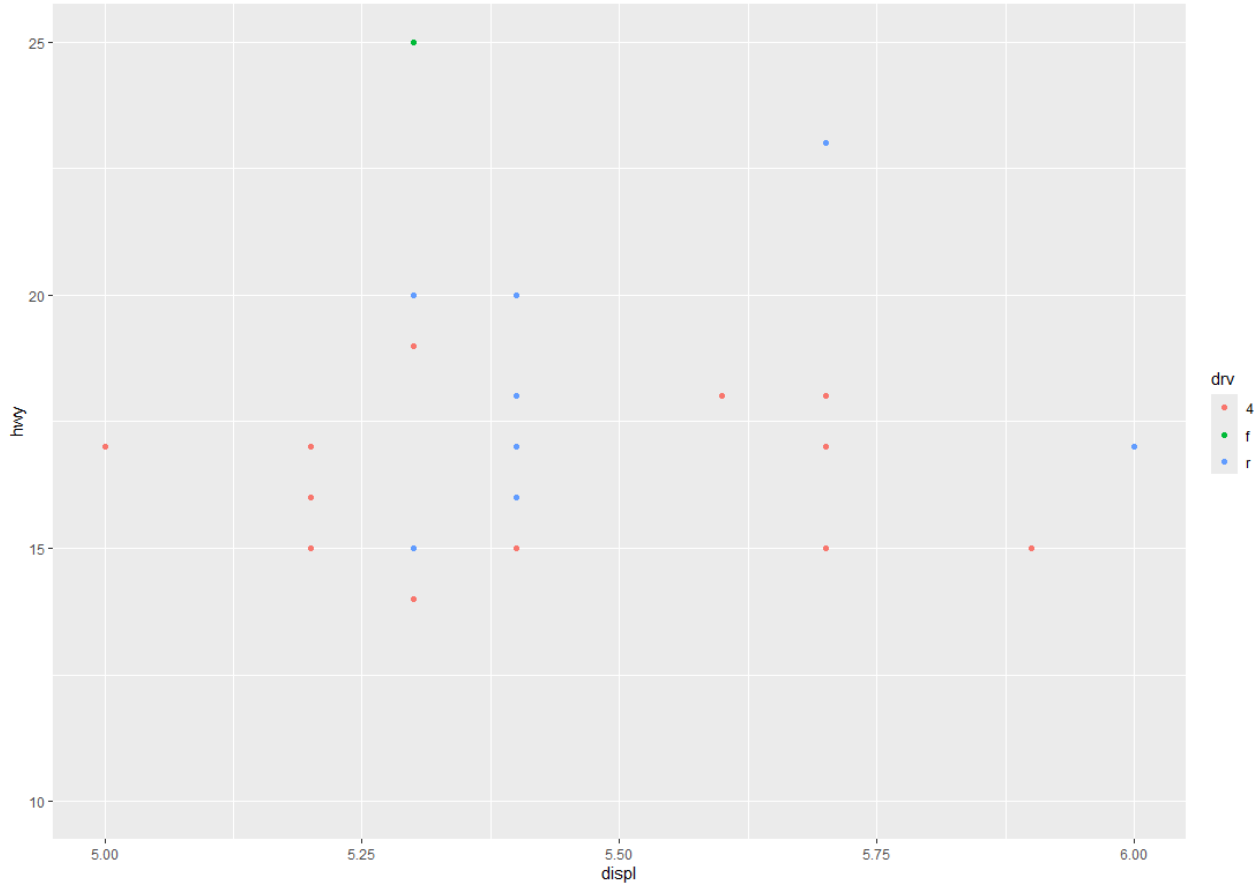
2. Aesthetics: Scales

[2], [3], [4]



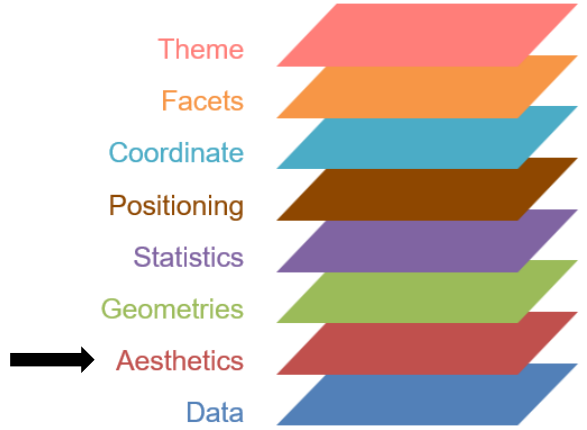
(1) Aktualisierung der Grenzen (`limits`)

```
base +
  scale_x_continuous(limits = c(5, 6)) +
  scale_y_continuous(limits = c(10, 25))
```



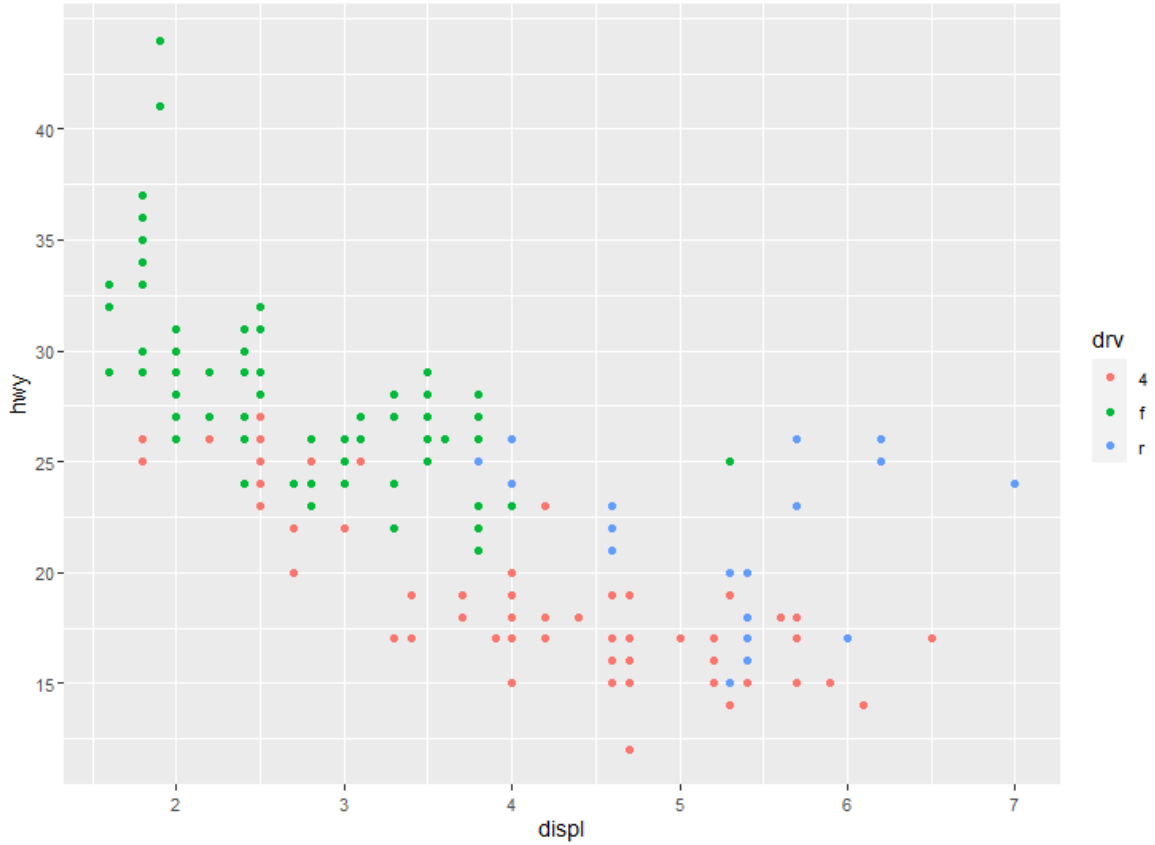
2. Aesthetics: Scales

[2], [3], [4]



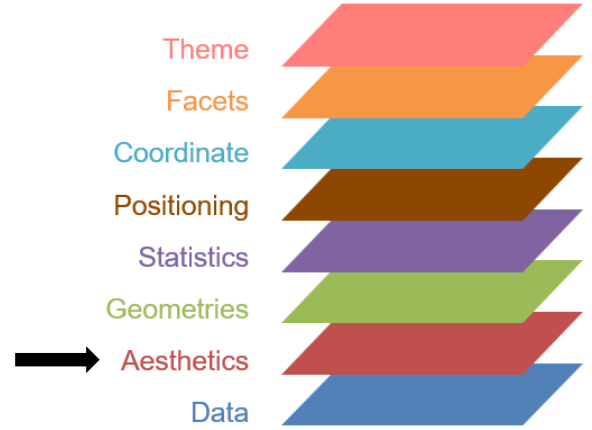
(2) Festlegung der Unterbrechung (`breaks`)

```
base +
  scale_y_continuous(breaks =seq(15, 40, by = 5))
```



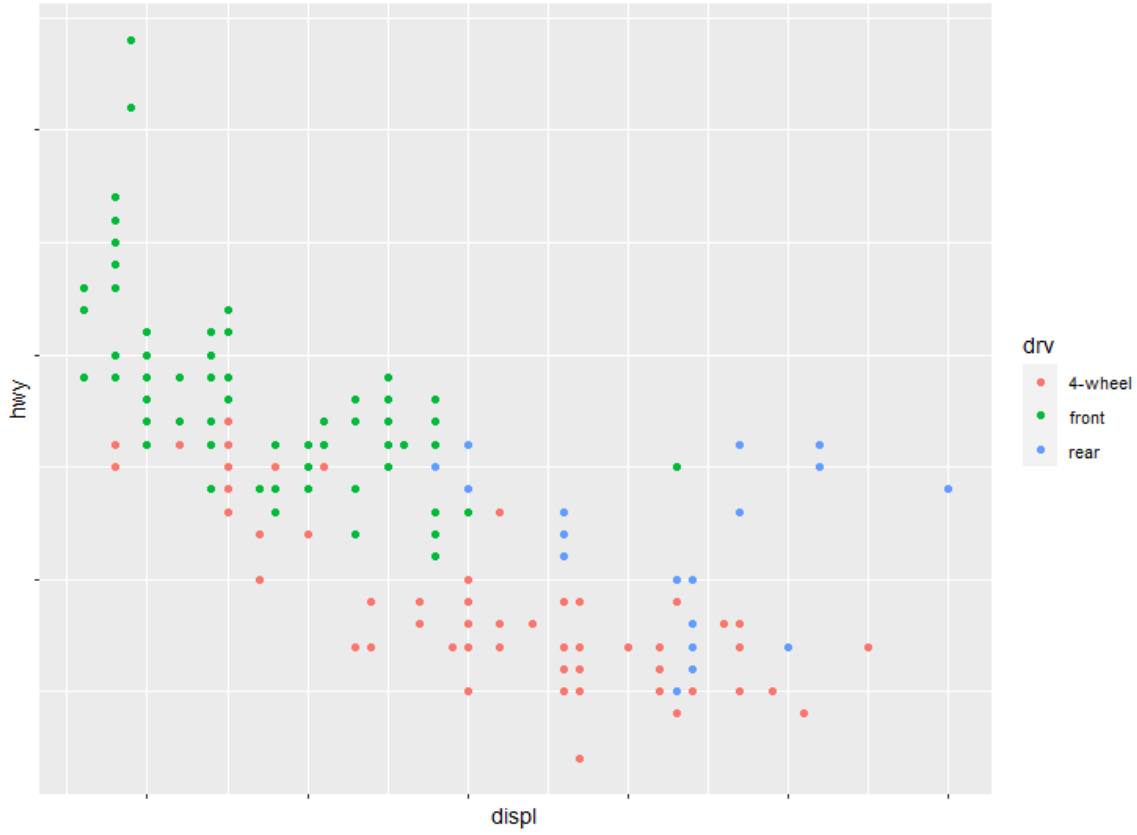
2. Aesthetics: Scales

[2], [3], [4]



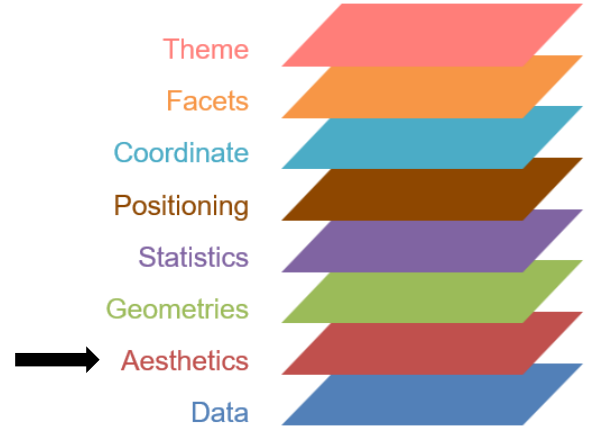
(3) Formatierung der Beschriftungen (`labels`)

```
base +
  scale_x_continuous(labels = NULL) +
  scale_y_continuous(labels = NULL) +
  scale_color_discrete(labels = c("4" =
"4-wheel","f" = "front", "r" = "rear"))
```



2. Aesthetics: Scales

[2], [3], [4]



(4) Durchführung von Transformationen

```
base +
  scale_x_log10() +
  scale_y_log10()
```

Exkurs: Scale vs Coordinate

[2], [3], [4]

	Scale	Coordinate
Zooming In	Bei Zooming mit Scale wird ein Teil der Daten „abgeschnitten“ d.h. ggplot wandelt sie zu NA Werten um	Bei Zooming mit ylim und xlim in coord_cartesian() erfolgt ein tatsächliches Zooming in eine Grafik
Transformation	Transformation erfolgt vor den Statistics ➡ KEINE Veränderung der Geoms	Transformation erfolgt nach den Statistics ➡ Veränderung der Geoms

Exkurs: Scale vs Coordinate

[2], [3], [4]

```
base <- ggplot(diamonds,
   aes(carat, price)) +
   stat_bin2d() +
   geom_smooth(method = "lm")
   + xlab(NULL) + ylab(NULL)
   + theme(legend.position =
   "none")
```

```
base +
   scale_x_log10() +
   scale_y_log10()
```

```
pow10 <-
   scales::exp_trans(10) base
   + scale_x_log10() +
   scale_y_log10() +
   coord_trans(x = pow10, y =
   pow10)
```