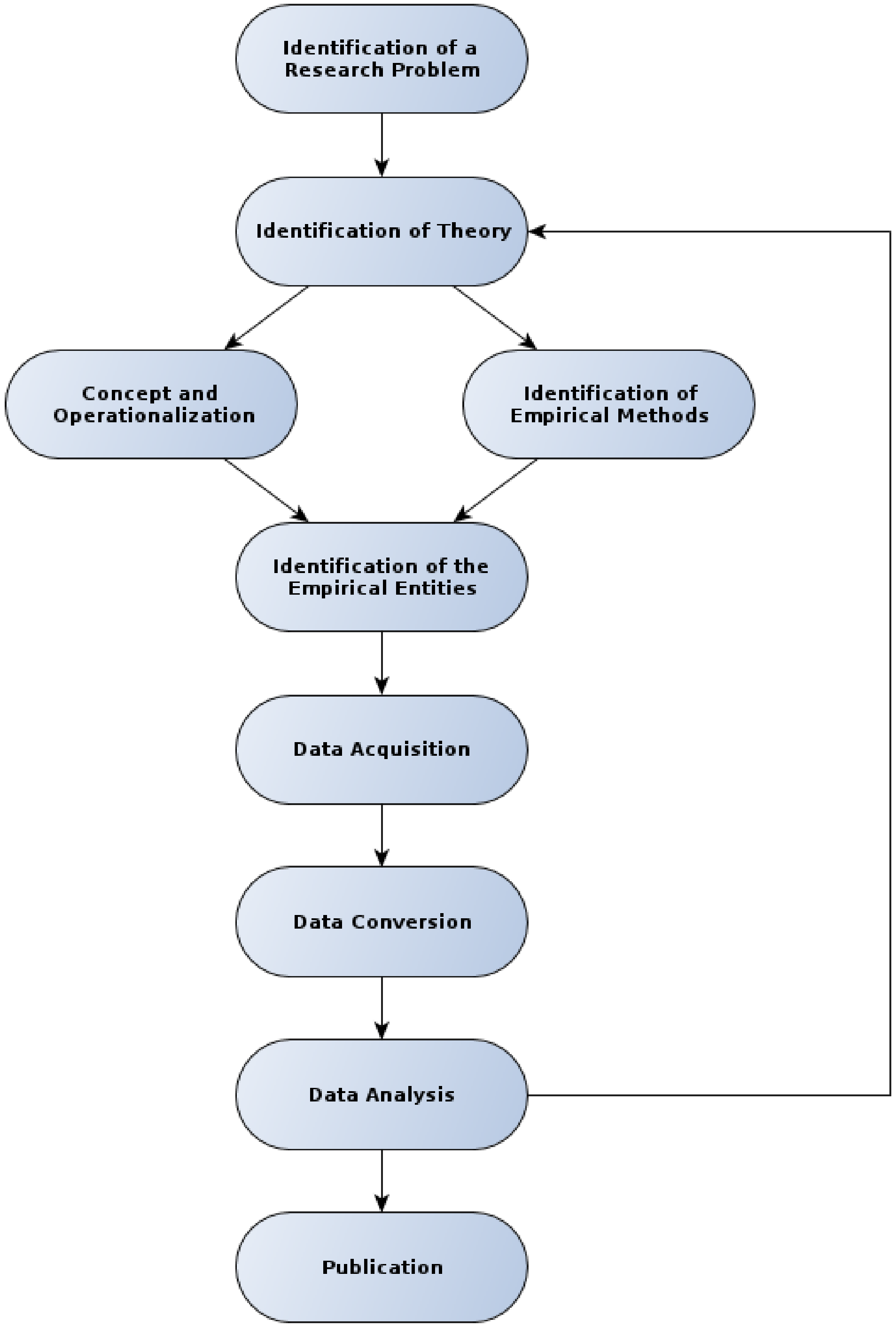
VERGLEICHENDE SOZIALFORSCHUNG MIT MEHREBENENMODELLEN IN R

Forschungspraktikum I und II
Dr. Christian Czymara
Research process

AGENDA

- How to do quantitative research
- The role of theory
- Tutorial: Introduction of European Social Survey
- Descriptive results and data vizualization

DOING EMPIRICAL-QUANTITATIVE SOCIAL SCIENCES

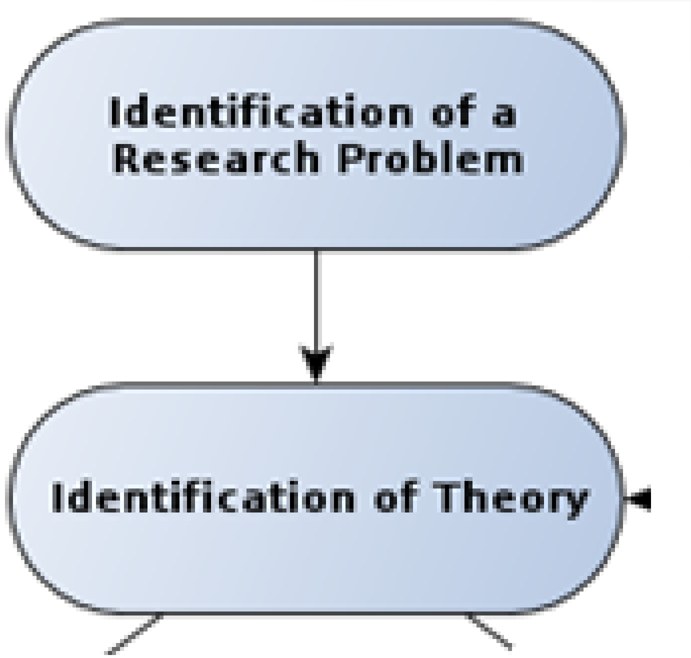


http://www.sosciso.de/en/forschungsprozess/

Adapted from Schnell, Hill & Esser: Methoden der empirischen Sozialforschung

RESEARCH PROCESS

- (Partly) covered
- Example studies from four topics
- From one of these – or any other – you develop your own research question

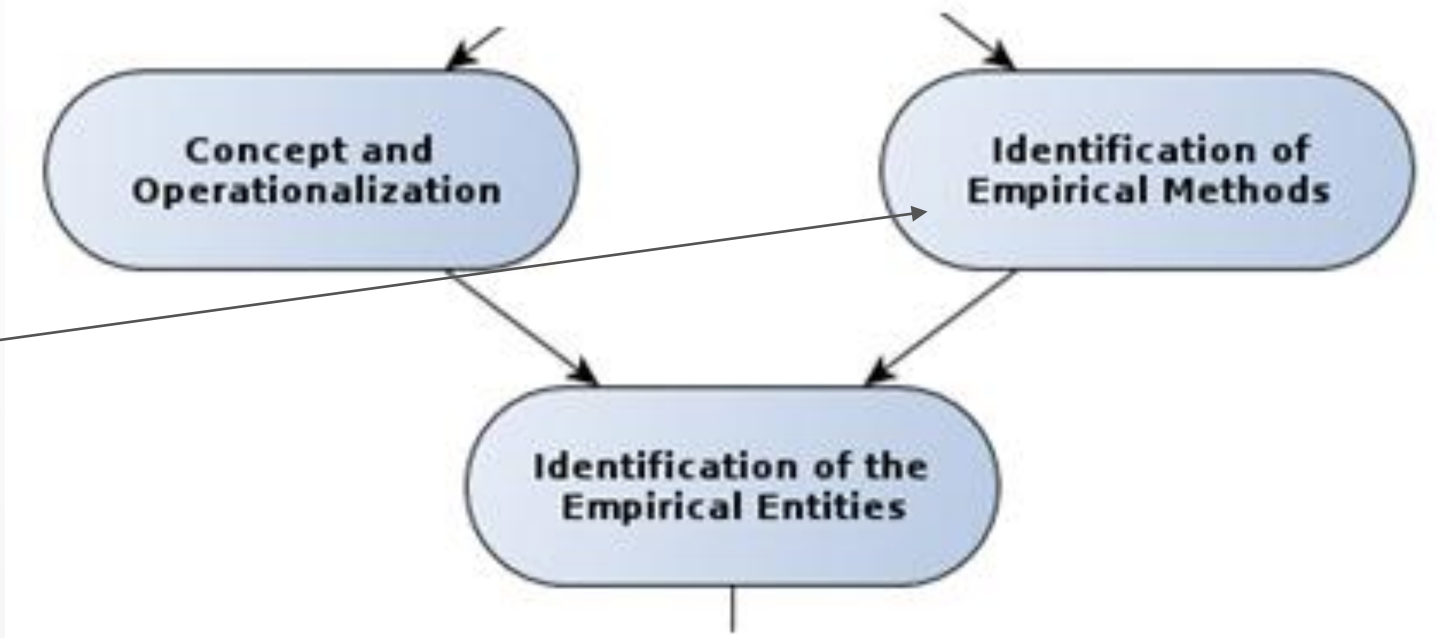


http://www.sosciso.de/en/forschungsprozess/

Adapted from Schnell, Hill & Esser: Methoden der empirischen Sozialforschung

RESEARCH PROCESS

- Translate theory into something testable
- (Partly) covered
- *Hierarchal Linear Models*

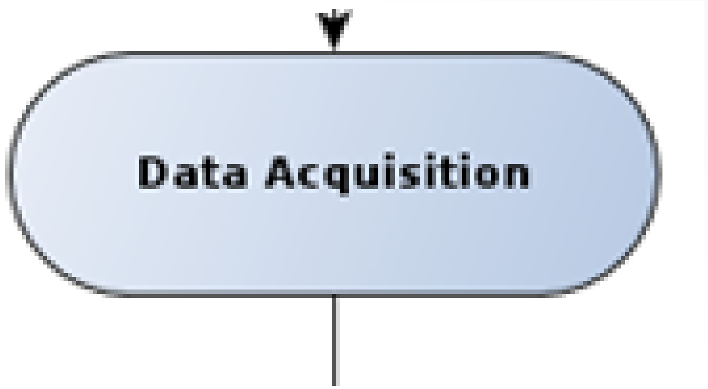


http://www.sosciso.de/en/forschungsprozess/

Adapted from Schnell, Hill & Esser: Methoden der empirischen Sozialforschung

RESEARCH PROCESS

- Collecting data can be the most time consuming part of the whole research project
- Not covered
- Secondary data (e. g., *European Social Survey*)
- Other sources of comparative data are fine as well

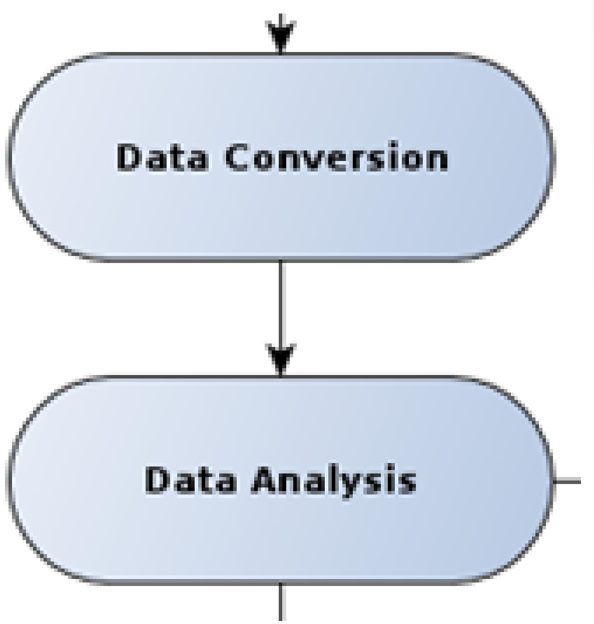


http://www.sosciso.de/en/forschungsprozess/

Adapted from Schnell, Hill & Esser: Methoden der empirischen Sozialforschung

RESEARCH PROCESS

- Covered
- We will discuss the code together
- Adapt this code for your research question
- And correctly interpret the output



http://www.sosciso.de/en/forschungsprozess/

Adapted from Schnell, Hill & Esser: Methoden der empirischen Sozialforschung

THEORY

WHAT IS THEORY (HERE)?

- Reality is infinitely complex
- We are interested in a certain part of reality
- Reduce complexity to those parts which are relevant for our research question
- Dilemma: Realistic but complicated vs. simplified but artificial

→Middle-range theories (Robert K. Merton)

WHAT IS THEORY?

- "A social theory is a set of two or more *propositions* in which *concepts* referring to certain social phenomena are assumed to be causally related" (Bohrnstedt & Knoke, 1982: 3)

"A connection between two *concepts or variables*, of either a covariational or causal nature" (ibid.)

"A statement about the *relationship* between abstract *concepts*" (ibid.)

"Person, object, relationship or event which is the referent of a social theory" (ibid.)

CONCEPTS

- Basic elements of theory
- Have to be defined clearly and unambiguously
- The definition of a concepts also defines how to measure it (*operationalization*)
- Can be located at different levels
  - Macro-level: For example: countries
  - Meso-level: For example: states, counties, neighborhoods
  - Micro-level: individual characteristics

theory

set of

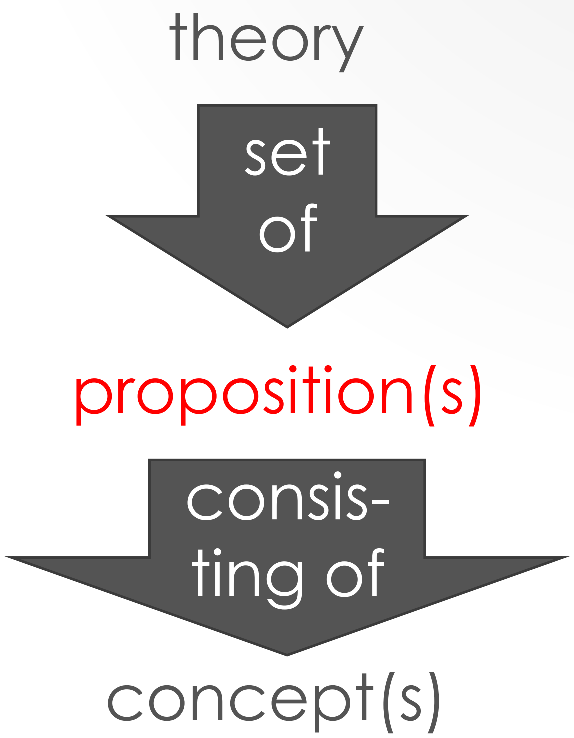
proposition(s)

consis-
ting of

concept(s)

PROPOSITION

- Propositions are statements about relationships between the concepts of the theory
- Usually, we want *causal* relationships



EXAMPLE

- Following examples from Czymara & Dochow (2018): *Mass Media and Concerns about Immigration in Germany in the 21st Century: Individual-Level Evidence over 15 Years*. European Sociological Review 34 (4): pages 381-401
- Available at: https://doi.org/10.1093/esr/jcy019

EXAMPLE: PROPOSITION AND CONCEPTS

- *“We expect that higher levels of media attention on immigration issues (media salience) increase the accessibility of related information in people’s minds and consequently raises individual concerns about these issues”* (384)
- Propositions are statements about relationships between the concepts of the theory
- Has to be empirically *testable*

EXAMPLE: PROPOSITION AND CONCEPTS

- *"We expect that higher levels of media attention on immigration issues (media salience) increase the accessibility of related information in people's minds and consequently raises individual concerns about these issues"* (384)
- Propositions are statements about relationships between the concepts of the theory

media attention

X

concerns

Y

EXAMPLE: CONCEPTS

- "Media attention on immigration raises concerns about this issue"
- But how do you measure media attention?
  - What kind of media? TV, print, online, …?
  - Mainstream media only?
  - Moreover, is tone important?
  - Or a focus on certain subissues only?
  - Or mere quantity sufficient?
  - etc…
- And what are concerns?
  - Worries?
  - Individual attention on the issue?
  - Xenophobia?
  - …

EXAMPLE: RELATIONSHIPS

- Propositions are statements about relationships between the concepts of the theory
- "Media attention on immigration raises concerns about this issue"

media attention

X

concerns

Y

EXAMPLE: RELATIONSHIPS

- Propositions are statements about relationships between the concepts of the theory
- "Media attention on immigration raises concerns about this issue"

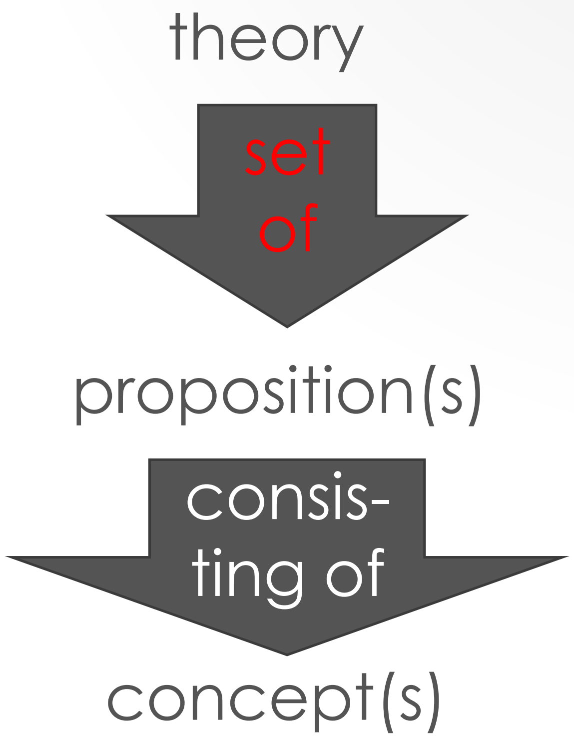
media attention　　　　　　　　　concerns

$$X \xrightarrow{\quad + \quad} Y$$

- Probabilistic (more → more) not deterministic (if → then)

RELATING PROPOSITIONS

- Theories consist of various propositions
- Different propositions are also related
- In that sense, theory is a complex of propositions allowing logical conclusions or predictions

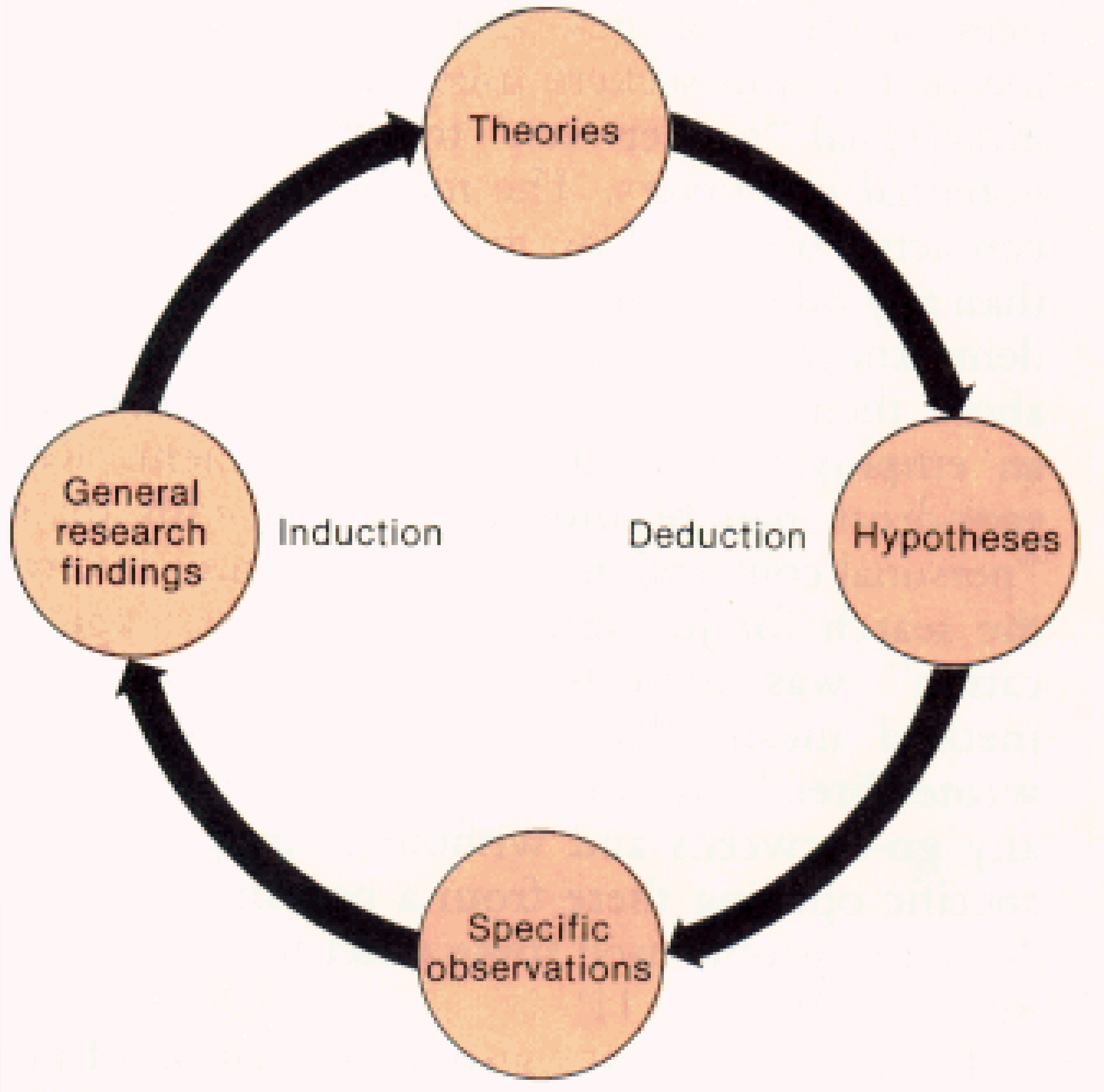


EXAMPLE

- Proposition 1: "Media attention on immigration raises concerns about this issue"
- Proposition 2: "*Voters are more open for information that is in line with their existing beliefs because they aim to uphold their long-term values*" (384)
- Deduction (=proposition 3): "*The negative effect of media salience […] is weaker (stronger) for natives who identify with more liberal (conservative) parties*" (ibid.)
- Not a factual statement, but a deduction from propositions 1 and 2

HYPOTHESES

RESEARCH CYCLE



Persell, Caroline Hodges. 1990. "Doing Social Research." Pp. 26-36 in Understanding Society: An Introduction to Sociology. 3rd ed. New York , NY : Harper & Row, Publishers, Inc.

http://www.asanet.org/sites/default/files/savvy/introtosociology/Documents/Persell%20Methods%20Reading11.htm

EXAMPLE: HYPOTHESES

- A hypothesis logically follows from the propositions of a theory
- Hypotheses are about precise and measurable constructs
- Example:
  - “Media attention on immigration raises concerns about this issue”

  →Hypothesis 1: High visibility of immigration issues in the media triggers individual concerns. (Salience-Hypothesis)” (384)

OPERATIONALIZATION

- Hypotheses tell us what we want to test
- But how do we actually test that?
- Operationalization is the way you measure your constructs
- Constructs → variables (varying characteristics)

EXAMPLE: OPERATIONALIZATION OF MEDIA ATTENTION

- “We combine the GSOEP with data from a **quantitative content analysis of German newspapers and news magazines** […] We scanned the content of all newspaper articles in our period of investigation with a **search string based on a keyword list of immigration-related terms** […] For our final media salience measure, we ran an **exploratory factor analysis** with four count variables indicating the **number of articles in each of the four outlets in the past 21 days** with the single days as units of analysis and extracted the factor values” (385, emphasize added)

EXAMPLE: OPERATIONALIZATION OF CONCERNS

- “Respondents are asked to rate how **much they are concerned about certain topics** in each year, including immigration to Germany on a **three-point scale**. […] This item is likely to capture a combination of two things: a negative evaluation of immigration and individual salience of immigration issue […] we understand threat perceptions to be the theoretical mechanism relating media salience and individual concerns” (384 f.)

TESTING HYPOTHESES

EMPIRICAL TESTING

- Research can not confirm hypotheses (only refute or fail to refute)
- Propositions are never definitely true (critical rationalism)
- Accumulated evidence that fails to refute hypotheses indicates that the proposition is correct
- Science as a evolutionary process

EXAMPLE: ACCUMULATED EVIDENCE

- Prior research: "*In sum, prior research suggests that the role of mass media remains rather ambivalent and context-dependent. […] We thus aim to advance the state of research on mass media effects on individual perceptions and attitudes by employing a more nuanced design than previous studies with similar scope*" (383)
- Future research: "*As manual coding with such a large number of articles is impossible, the rapidly growing field of text as data in the information sciences should be of great help […] adding such information to our approach could lead to further insights and, thus, deepen the understanding of the relationship between mass media and public opinion formation*" (395)

(SOME) PROBLEMS OF QUANTITATIVE SOCIAL RESEARCH

PROBLEMS OF QUANTITATIVE SOCIAL RESEARCH

- Most, if not all, social phenomena are highly abstract
- Sometimes several ways to derive hypotheses from the same proposition
- Most variables are not directly observable but latent (attitudes, ideology, social positions, political systems …)
- Issues of measurement and validity (especially when measured in different social contexts and languages)
- In the example: Does the count variable of articles really measure the media environment people are exposed to?

PROBLEMS OF QUANTITATIVE SOCIAL RESEARCH

- Social scientists often can not (and should not) manipulate the explanatory variable of interest
- E. g., can not randomly assign individuals to different levels of media salience (or income, or education, …)
- Complicates identification of causal effects
- A good design helps (e. g., natural experiments)
- Example:
  - Media salience for each person depends on the interview day
  - Is timing of interviews random?

ON REFUTING HYPOTHESES

- "Media attention on immigration raises concerns about this issue."
- Operationalization:
  - Media salience: Count of newspaper articles in a certain time
  - Being concerned (yes / no)
- What if we would not find the hypothesized relationship?
  - Either it is not statistically significant
  - Or it is statistically significant but small

NULL RESULTS

- Proposition / theory not “true” → *improved knowledge*
- Variable is not a good measure → technical problem, hypothesis not really refuted → maybe improved methodology
- Hypotheses not properly derived from theory → hypothesis refuted but proposition / theory not → not very helpful
- Proposition only valid under certain circumstances → modify theory → test again → *improved knowledge*

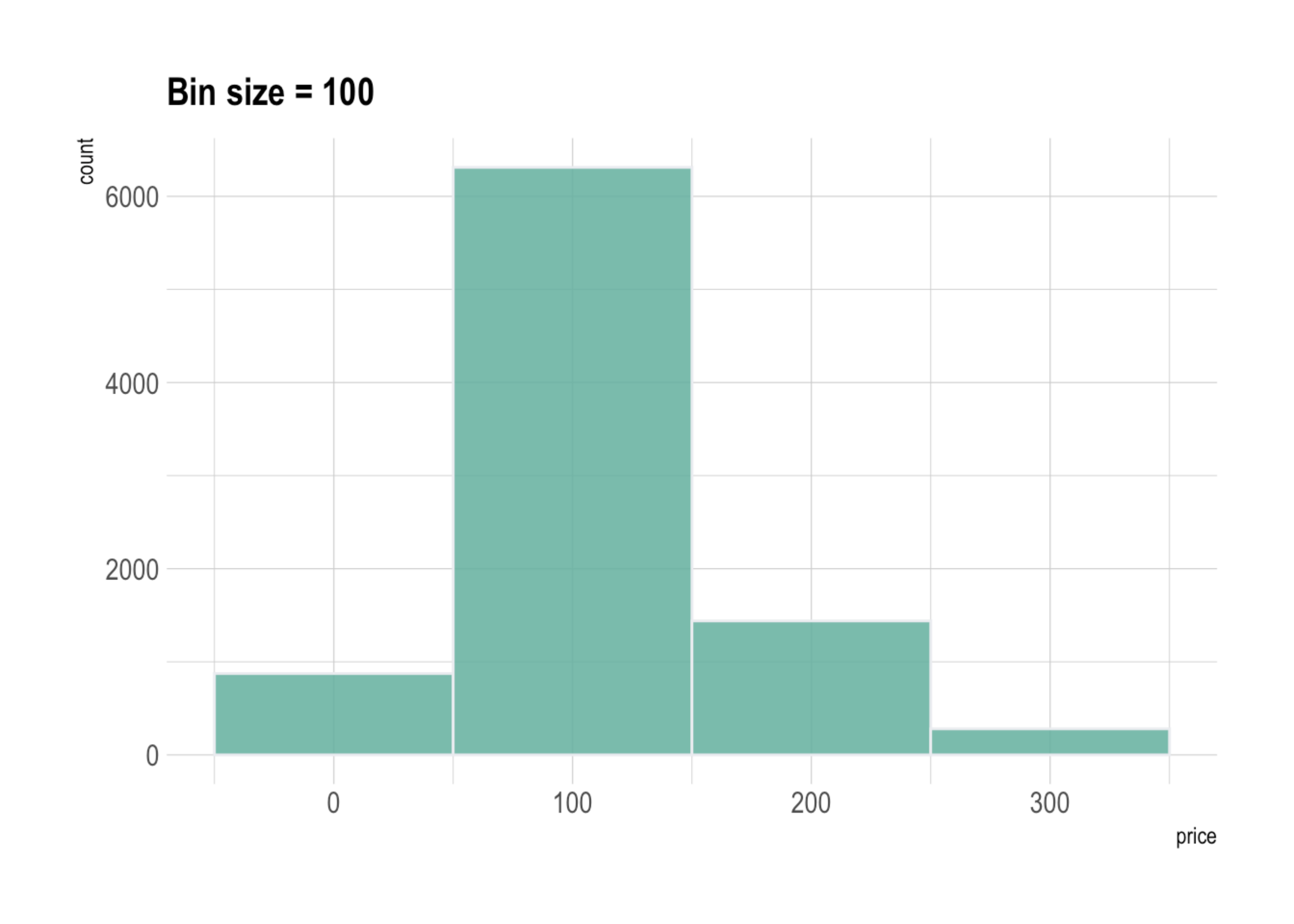
DATA VIZUALIZATION

DATA VIZUALIZATION

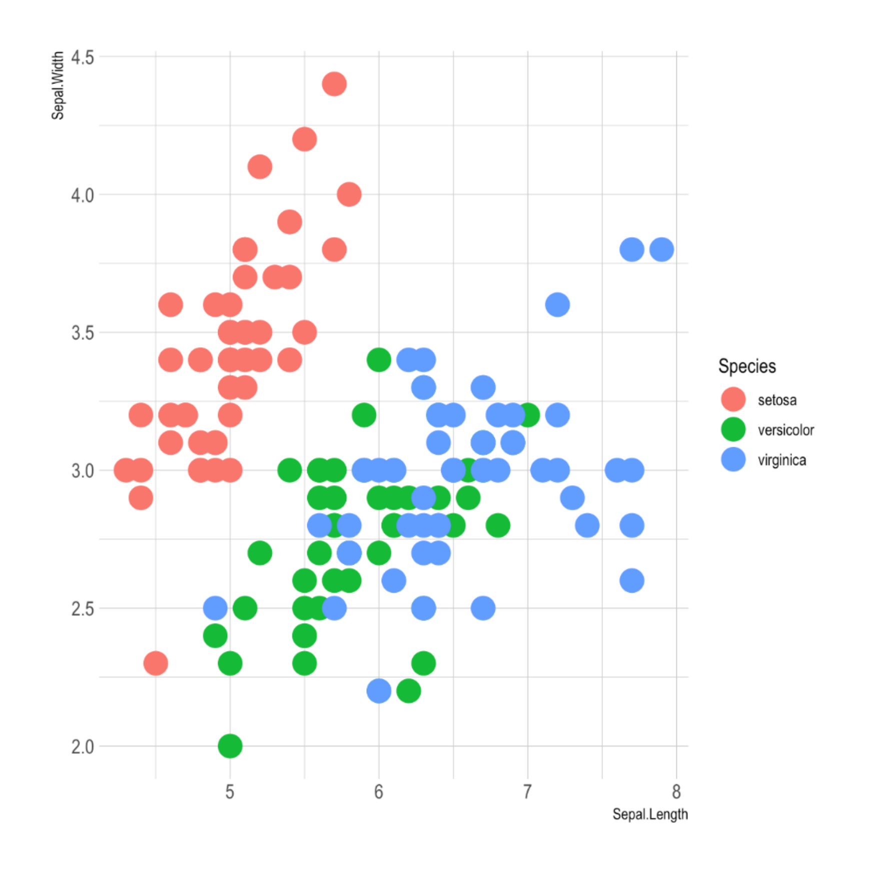
- Categorical variables → analyze frequency of single categories
  - Univariate: Tables no problem, graphs (e. g., bar chart)
  - Bivariate: Cross table, conditional bar chart
- Continuous variables → analyze distribution
  - Univariate: Tables not suited in most cases, graphs (histogram, boxplot)
  - Bivariate: scatter diagram
- Bivariate graphs with one continuous and one categorical variable: conditional box plot, conditional bar chart

DATA VIZUALIZATION

- Categorical:



- Continuous:



See https://www.r-graph-gallery.com/

GGPLOT2

- Powerful tool for data vizualisation (can plot basically everything)
- Basic logic: `ggplot(data, aes(x, y))`
- Adding, for example, a scatter plot:
  - `ggplot(data, aes(x, y)) + geom_point()`
- GGPart of the Tidyverse, so can be combined with `dplyr` etc.
- https://ggplot2.tidyverse.org/

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 components: a **data** set, a **coordinate system**, and **geoms**—visual marks that represent data points.

To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.

Complete the template below to build a graph.

```
ggplot (data = <DATA> ) +                                   required
  <GEOM_FUNCTION>(mapping = aes( <MAPPINGS> ),
  stat = <STAT> , position = <POSITION> ) +                  Not required, sensible defaults supplied
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION> +
  <SCALE_FUNCTION> +
  <THEME_FUNCTION>
```

ggplot(data = mpg, **aes**(x = cty, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

last_plot() Returns the last plot.

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Aes

Common aesthetic values.

color and **fill** - string ("red", "#RRGGBB")

linetype - integer or string (0 = "blank", 1 = "solid", 2 = "dashed", 3 = "dotted", 4 = "dotdash", 5 = "longdash", 6 = "twodash")

lineend - string ("round", "butt", or "square")

linejoin - string ("round", "mitre", or "bevel")

size - integer (line width in mm)

shape - integer/shape name or a single character ("a")

RStudio

Geoms

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

GRAPHICAL PRIMITIVES

a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + geom_blank() and **a + expand_limits()**
Ensure limits include values across all plots.

b + geom_curve(aes(yend = lat + 1, xend = long + 1), curvature = 1) - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

a + geom_path(lineend = "butt", linejoin = "round", linemitre = 1)
x, y, alpha, color, group, linetype, size

a + geom_polygon(aes(alpha = 50)) - x, y, alpha, color, fill, group, subgroup, linetype, size

b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1)) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

a + geom_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900)) - x, ymax, ymin, alpha, color, fill, group, linetype, size

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

b + geom_abline(aes(intercept = 0, slope = 1))
b + geom_hline(aes(yintercept = lat))
b + geom_vline(aes(xintercept = long))

b + geom_segment(aes(yend = lat + 1, xend = long + 1))
b + geom_spoke(aes(angle = 1:1155, radius = 1))

ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian")
x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot()
x, y, alpha, color, fill

c + geom_freqpoly()
x, y, alpha, color, group, linetype, size

c + geom_histogram(binwidth = 5)
x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy))
x, y, alpha, color, fill, linetype, size, weight

discrete

d <- ggplot(mpg, aes(fl))

d + geom_bar()
x, alpha, color, fill, linetype, size, weight

TWO VARIABLES

both continuous

e <- ggplot(mpg, aes(cty, hwy))

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_point()
x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile()
x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl")
x, y, alpha, color, linetype, size

e + geom_smooth(method = lm)
x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

one discrete, one continuous

f <- ggplot(mpg, aes(class, hwy))

f + geom_col()
x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot()
x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + geom_dotplot(binaxis = "y", stackdir = "center")
x, y, alpha, color, fill, group

f + geom_violin(scale = "area")
x, y, alpha, color, fill, group, linetype, size, weight

both discrete

g <- ggplot(diamonds, aes(cut, color))

g + geom_count()
x, y, alpha, color, fill, shape, size, stroke

e + geom_jitter(height = 2, width = 2)
x, y, alpha, color, fill, shape, size

continuous bivariate distribution

h <- ggplot(diamonds, aes(carat, price))

h + geom_bin2d(binwidth = c(0.25, 500))
x, y, alpha, color, fill, linetype, size, weight

h + geom_density_2d()
x, y, alpha, color, group, linetype, size

h + geom_hex()
x, y, alpha, color, fill, size

continuous function

i <- ggplot(economics, aes(date, unemploy))

i + geom_area()
x, y, alpha, color, fill, linetype, size

i + geom_line()
x, y, alpha, color, group, linetype, size

i + geom_step(direction = "hv")
x, y, alpha, color, group, linetype, size

visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))

j + geom_crossbar(fatten = 2) - x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + geom_errorbar() - x, ymax, ymin, alpha, color, group, linetype, size, width
Also **geom_errorbarh()**.

j + geom_linerange()
x, ymin, ymax, alpha, color, group, linetype, size

j + geom_pointrange() - x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

maps

data <- data.frame(murder = USArrests$Murder,
state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))

k + geom_map(aes(map_id = state), map = map)
+ expand_limits(x = map$long, y = map$lat)
map_id, alpha, color, fill, linetype, size

THREE VARIABLES

seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)); l <- ggplot(seals, aes(long, lat))

l + geom_contour(aes(z = z))
x, y, z, alpha, color, group, linetype, size, weight

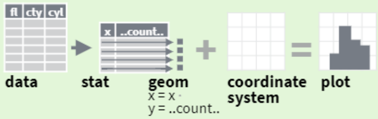
l + geom_contour_filled(aes(fill = z))
x, y, alpha, color, fill, group, linetype, size, subgroup

l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE)
x, y, alpha, fill

l + geom_tile(aes(fill = z))
x, y, alpha, color, fill, linetype, size, width

Stats An alternative way to build a layer.

A stat builds new variables to plot (e.g., count, prop).



Visualize a stat by changing the default stat of a geom function, **geom_bar(stat="count")** or by using a stat function, **stat_count(geom="bar")**, which calls a default geom to make a layer (equivalent to a geom function). Use **..name..** syntax to map stat variables to aesthetics.

geom to use · stat function · geommappings

i + stat_density_2d(aes(fill = ..level..), geom = "polygon")

variable created by stat

c + stat_bin(binwidth = 1, boundary = 10)
x, y | ..count.., ..ncount.., ..density.., ..ndensity..

c + stat_count(width = 1) **x, y** | ..count.., ..prop..

c + stat_density(adjust = 1, kernel = "gaussian")
x, y | ..count.., ..density.., ..scaled..

e + stat_bin_2d(bins = 30, drop = T)
x, y, fill | ..count.., ..density..

e + stat_bin_hex(bins = 30) **x, y, fill** | ..count.., ..density..

e + stat_density_2d(contour = TRUE, n = 100)
x, y, color, size | ..level..

e + stat_ellipse(level = 0.95, segments = 51, type = "t")

l + stat_contour(aes(z = z)) **x, y, z, order** | ..level..

l + stat_summary_hex(aes(z = z), bins = 30, fun = max)
x, y, z, fill | ..value..

l + stat_summary_2d(aes(z = z), bins = 30, fun = mean)
x, y, z, fill | ..value..

f + stat_boxplot(coef = 1.5)
x, y | ..lower.., ..middle.., ..upper.., ..width.. , ..ymin.., ..ymax..

f + stat_ydensity(kernel = "gaussian", scale = "area") **x, y** | ..density.., ..scaled.., ..count.., ..n.., ..violinwidth.., ..width..

e + stat_ecdf(n = 40) **x, y** | ..x.., ..y..

e + stat_quantile(quantiles = c(0.1, 0.9), formula = y ~ log(x), method = "rq") **x, y** | ..quantile..

e + stat_smooth(method = "lm", formula = y ~ x, se = T, level = 0.95) **x, y** | ..se.., ..x.., ..y.., ..ymin.., ..ymax..

ggplot() + xlim(-5, 5) **+ stat_function**(fun = dnorm, n = 20, geom = "point") **x** | ..x.., ..y..

ggplot() + stat_qq(aes(sample = 1:100))
x, y, sample | ..sample.., ..theoretical..

e + stat_sum() x, y, size | ..n.., ..prop..

e + stat_summary(fun.data = "mean_cl_boot")

h + stat_summary_bin(fun = "mean", geom = "bar")

e + stat_identity()

e + stat_unique()

Scales Override defaults with **scales** package.

Scales map data values to the visual values of an aesthetic. To change a mapping, add a new scale.

n <- d + geom_bar(aes(fill = fl))

scale_ · aesthetic to adjust · prepackaged scale to use · scale-specific arguments

n + scale_fill_manual(
values = c("skyblue", "royalblue", "blue", "navy"),
limits = c("d", "e", "p", "r"), **breaks** =c("d", "e", "p", "r"),
name = "fuel", **labels** = c("D", "E", "P", "R"))

range of values to include in mapping · title to use in legend/axis · labels to use in legend/axis · breaks to use in legend/axis

GENERAL PURPOSE SCALES

Use with most aesthetics

scale_*_continuous() - Map cont' values to visual ones.
scale_*_discrete() - Map discrete values to visual ones.
scale_*_binned() - Map continuous values to discrete bins.
scale_*_identity() - Use data values as visual ones.
scale_*_manual(values = c()) - Map discrete values to manually chosen visual ones.
scale_*_date(date_labels = "%m/%d"), date_breaks = "2 weeks") - Treat data values as dates.
scale_*_datetime() - Treat data values as date times. Same as scale_*_date(). See ?strptime for label formats.

X & Y LOCATION SCALES

Use with x or y aesthetics (x shown here)

scale_x_log10() - Plot x on log10 scale.
scale_x_reverse() - Reverse the direction of the x axis.
scale_x_sqrt() - Plot x on square root scale.

COLOR AND FILL SCALES (DISCRETE)

n + scale_fill_brewer(palette = "Blues")
For palette choices:
RColorBrewer::display.brewer.all()

n + scale_fill_grey(start = 0.2, end = 0.8, na.value = "red")

COLOR AND FILL SCALES (CONTINUOUS)

o <- c + geom_dotplot(aes(fill = ..x..))

o + scale_fill_distiller(palette = "Blues")

o + scale_fill_gradient(low="red", high="yellow")

o + scale_fill_gradient2(low = "red", high = "blue", mid = "white", midpoint = 25)

o + scale_fill_gradientn(colors = topo.colors(6))
Also: rainbow(), heat.colors(), terrain.colors(), cm.colors(), RColorBrewer::brewer.pal()

SHAPE AND SIZE SCALES

p <- e + geom_point(aes(shape = fl, size = cyl))

p + scale_shape() + scale_size()
p + scale_shape_manual(values = c(3:7))

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

p + scale_radius(range = c(1,6))
p + scale_size_area(max_size = 6)

Coordinate Systems

r <- d + geom_bar()

r + coord_cartesian(xlim = c(0, 5)) - xlim, ylim
The default cartesian coordinate system.

r + coord_fixed(ratio = 1/2)
ratio, xlim, ylim - Cartesian coordinates with fixed aspect ratio between x and y units.

ggplot(mpg, aes(y = fl)) **+ geom_bar()**
Flip cartesian coordinates by switching x and y aesthetic mappings.

r + coord_polar(theta = "x", direction=1)
theta, start, direction - Polar coordinates.

r + coord_trans(y = “sqrt”) - x, y, xlim, ylim
Transformed cartesian coordinates. Set xtrans and ytrans to the name of a window function.

π + coord_quickmap()
π + coord_map(projection = "ortho", orientation = c(41, -74, 0)) - projection, xlim, ylim
Map projections from the mapproj package (mercator (default), azequalarea, lagrange, etc.).

Position Adjustments

Position adjustments determine how to arrange geoms that would otherwise occupy the same space.

s <- ggplot(mpg, aes(fl, fill = drv))

s + geom_bar(position = "dodge")
Arrange elements side by side.

s + geom_bar(position = "fill")
Stack elements on top of one another, normalize height.

e + geom_point(position = "jitter")
Add random noise to X and Y position of each element to avoid overplotting.

e + geom_label(position = "nudge")
Nudge labels away from points.

s + geom_bar(position = "stack")
Stack elements on top of one another.

Each position adjustment can be recast as a function with manual **width** and **height** arguments:
s + geom_bar(position = position_dodge(width = 1))

Themes

r + theme_bw()
White background with grid lines.

r + theme_gray()
Grey background (default theme).

r + theme_dark()
Dark for contrast.

r + theme_classic()

r + theme_light()

r + theme_linedraw()

r + theme_minimal()
Minimal theme.

r + theme_void()
Empty theme.

r + theme() Customize aspects of the theme such as axis, legend, panel, and facet properties.
r + ggtitle("Title") + theme(plot.title.postion = "plot")
r + theme(panel.background = element_rect(fill = "blue"))

Faceting

Facets divide a plot into subplots based on the values of one or more discrete variables.

t <- ggplot(mpg, aes(cty, hwy)) + geom_point()

t + facet_grid(cols = vars(fl))
Facet into columns based on fl.

t + facet_grid(rows = vars(year))
Facet into rows based on year.

t + facet_grid(rows = vars(year), cols = vars(fl))
Facet into both rows and columns.

t + facet_wrap(vars(fl))
Wrap facets into a rectangular layout.

Set **scales** to let axis limits vary across facets.

t + facet_grid(rows = vars(drv), cols = vars(fl), scales = "free")
x and y axis limits adjust to individual facets:
"free_x" - x axis limits adjust
"free_y" - y axis limits adjust

Set **labeller** to adjust facet label:

t + facet_grid(cols = vars(fl), labeller = label_both)

fl: c	fl: d	fl: e	fl: p	fl: r

t + facet_grid(rows = vars(fl), labeller = label_bquote(alpha ^ .(fl)**))**

α^c	α^d	α^e	α^p	α^r

Labels and Legends

Use **labs()** to label the elements of your plot.

t + labs(x = "New x axis label", **y** = "New y axis label",
title ="Add a title above the plot",
subtitle = "Add a subtitle below title",
caption = "Add a caption below plot",
alt = "Add alt text to the plot",
<AES> = "New <AES> legend title")

t + annotate(geom = "text", x = 8, y = 9, label = “A”)
Places a geom with manually selected aesthetics.

p + guides(x = guide_axis(n.dodge = 2)) Avoid crowded or overlapping labels with guide_axis(n.dodge or angle).

n + guides(fill = “none”) Set legend type for each aesthetic: colorbar, legend, or none (no legend).

n + theme(legend.position = "bottom")
Place legend at "bottom", "top", "left", or “right”.

n + scale_fill_discrete(name = "Title", labels = c("A", "B", "C", "D", "E"))
Set legend title and labels with a scale function.

Zooming

Without clipping (preferred):
t + coord_cartesian(xlim = c(0, 100), ylim = c(10, 20))

With clipping (removes unseen data points):
t + xlim(0, 100) **+ ylim**(10, 20)

t + scale_x_continuous(limits = c(0, 100)) **+ scale_y_continuous**(limits = c(0, 100))

RStudio

LITERATURE

- Bohrnstedt & Knoke (1982): Kapitel 1 in “Statistics for Social Data Analysis”. Peacock Publishers