Statistical Computing with R: Masters in Data Science 503 (S15) Second Batch, SMS, TU, 2023

Shital Bhandary

Associate Professor

Statistics/Bio-statistics, Demography and Public Health Informatics

Patan Academy of Health Sciences, Lalitpur, Nepal

Faculty, Data Analysis and Decision Modeling, MBA, Pokhara University, Nepal

Faculty, FAIMER Fellowship in Health Professions Education, India/USA.

Review Preview

- Grammar of graphics
  - Book by Wilkinson (et. al.) published in 2005
  - Reviewed by N. Cox in Journal of Statistical Software
- Find them and read!
- ggplot2 package
  - How grammar of graphics is implemented by Wickham in this package?
  - How is it different from the GG proposed by Wilkinson et. al.?

Grammar of graphics:

- What is a graphic?
- How can we succinctly describe a graphic?
- And how can we create the graphic that we have described?
- **One way to answer these questions is to develop a grammar: “the fundamental principles or rules of an art or science”**

Grammar:

https://byrneslab.net/classes/biol607/readings/wickham_layered-grammar.pdf

- A grammar provides a strong foundation for understanding a diverse range of graphics.

- A grammar may also help guide us on what a well-formed or correct graphic looks like, but there will still be **many grammatically correct but nonsensical graphics**. (Figure don't lie, liars figure!)

- This is easy to see by analogy to the English language: **good grammar is just the first step in creating a good sentence**

Grammar of graphics:

- A grammar of graphics is a tool that enables us to concisely describe the components of a graphic.
- Such a grammar allows us to move beyond named graphics (e.g., the “scatterplot”) and gain insight into the deep structure that underlies statistical graphics.
- **ggplot2 proposes an alternative parameterization of the grammar, based around the idea of building up a graphic from multiple layers of data.**

Building a plot:

- We need data
- Is this a “tidy” data?
- How many cases are there?
- How many variables are there?
- What are the “types” of variables?

Data:

Sample Dataset

A	B	C	D
2	3	4	a
1	2	1	a
4	5	15	b
9	10	80	b

Sample Dataset with variable named according to **aesthetic**

x	y	Shape
2	4	a
1	1	a
4	15	b
9	80	b

Shape = facet!

Geometric objects:

- Point
- Line
- Bar etc.

Final Graphic: Graphical objects + Scale + Coordinate system

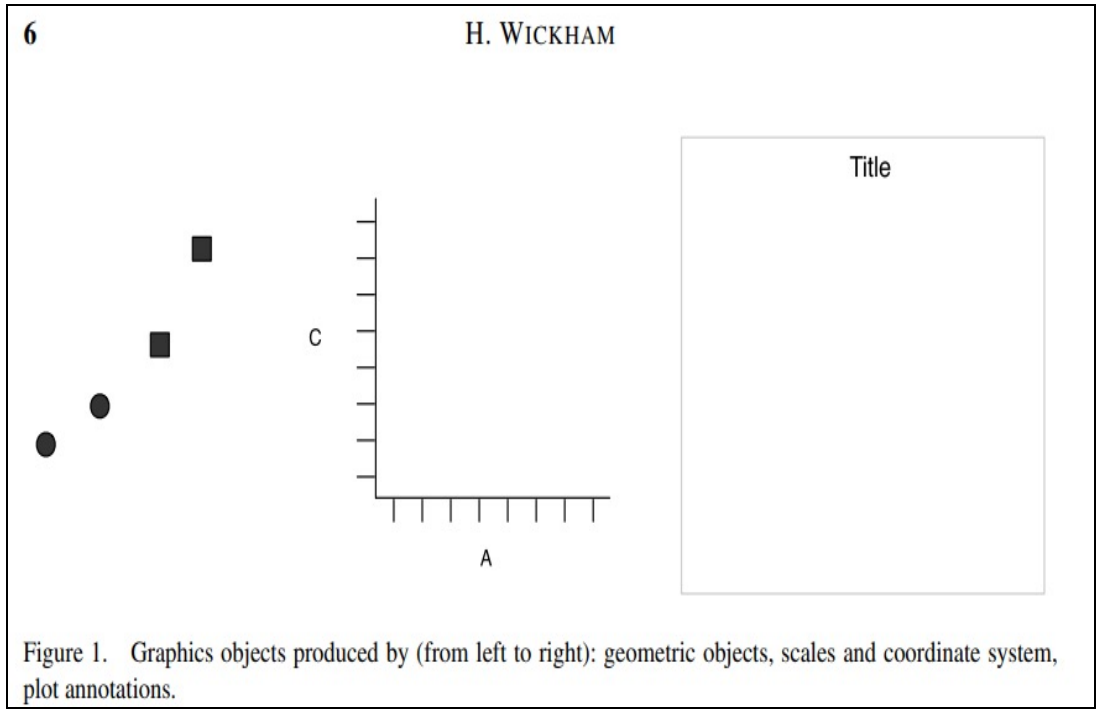

6 H. WICKHAM

Figure 1. Graphics objects produced by (from left to right): geometric objects, scales and coordinate system, plot annotations.

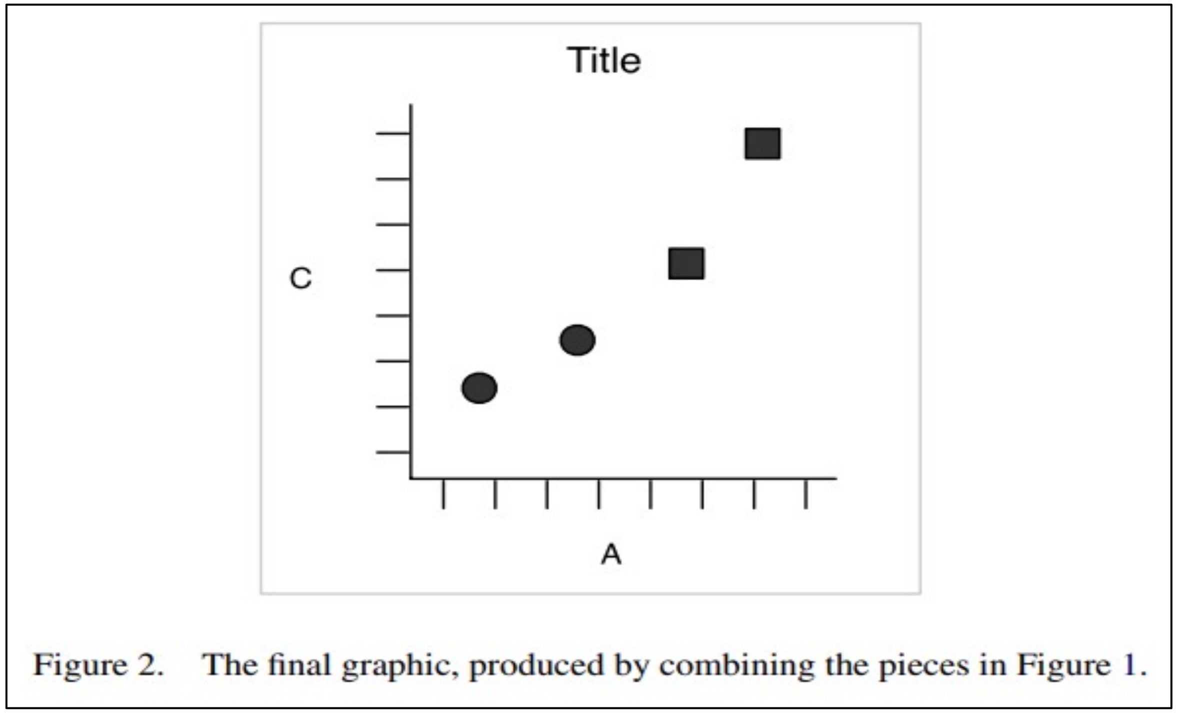


Figure 2. The final graphic, produced by combining the pieces in Figure 1.

Components of “layered” grammar: ggplot2

- Data and aesthetic mapping +
- Geometric objects +
- Scales +
- Facet specification

plus

- Statistical transformation
  - Log, reciprocal, square etc.
- Coordinate system
  - Cartesian space
  - Non-Cartesian space

- Together, the data, mappings, statistical transformation, and geometric object form a layer.

- A plot may have multiple layers, for example, when we overlay a scatterplot with a smoothed line

Layered grammar of graphics:

- To be precise, the layered grammar defines the components of a plot as:
    - a default dataset and set of mappings from variables to aesthetics,
    - one or more layers, with each layer having one geometric object, one statistical transformation, one position adjustment, and optionally, one dataset and set of aesthetic mappings,
    - one scale for each aesthetic mapping used,
    - a coordinate system,
    - the facet specification

GG (GPL) vs LG (ggplot2):

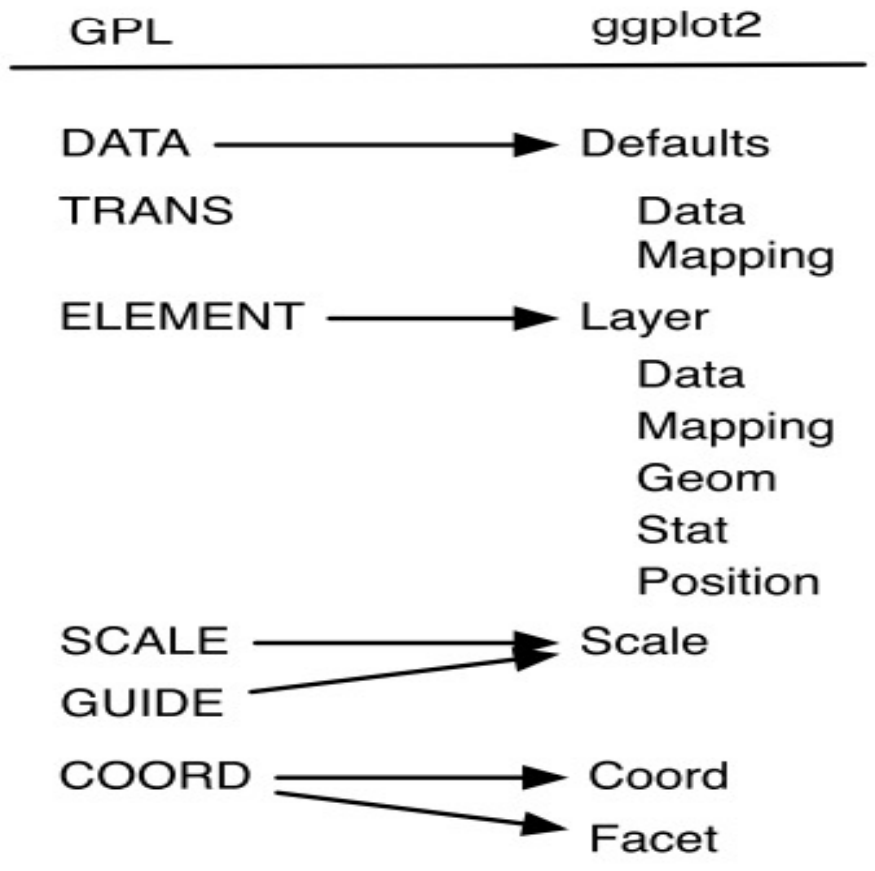


Figure 4. Mapping between components of Wilkinson's grammar (left) and the layered grammar (right). TRANS has no correspondence in `ggplot2`: its role is played by built-in R features.

Layers:

- Layers are responsible for creating the objects that we perceive on the plot.
- A layer is composed of four parts:
  - data and aesthetic mapping
  - a statistical transformation (stat)
  - a geometric object (geom)
  - a position adjustment

GPL (GG) vs ggplot2 (LGG):

- GPL: line(position(smooth.linear(x * y)), color(z))

- ggplot2: layer(aes(x = x, y = y, color = z),

  geom="line",

  stat="smooth")

Where,

data = data in both

aes = aesthetic = x variable in x-position, y variable in y-position, z variable to produce color, geometric object is “line” and a smoothed line to be added using statistical procedure

Geometric objects:

- Point geom = scatterplot
- Line geom = line plot etc.

- Geom Dimensionality as per the “ggplot2” package:
  - 0d: point, text
  - 1d: path, line (ordered path)
  - 2d: polygon, interval (continuous)

Interval geometric objects: Four representations

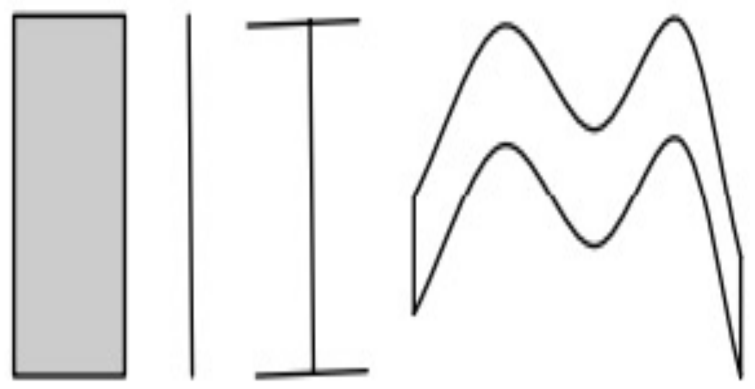


Figure 6. Four representations of an interval geom. From left to right: as a bar, as a line, as an error bar, and (for continuous x) as a ribbon.

Geom aesthetics:

- Point geom = Position, color, shape, size
- Bar geom = position, height, width, fill color
- **Line geom = ???**
- It is always better to ask help about these aesthetics with “?” in R!
- We can also tweak the position e.g. stock (sub-divide) or dodge (side-by-side) bar plots like we did using “beside” in the Base R graphics!

Statistical transformations in ggplot2:

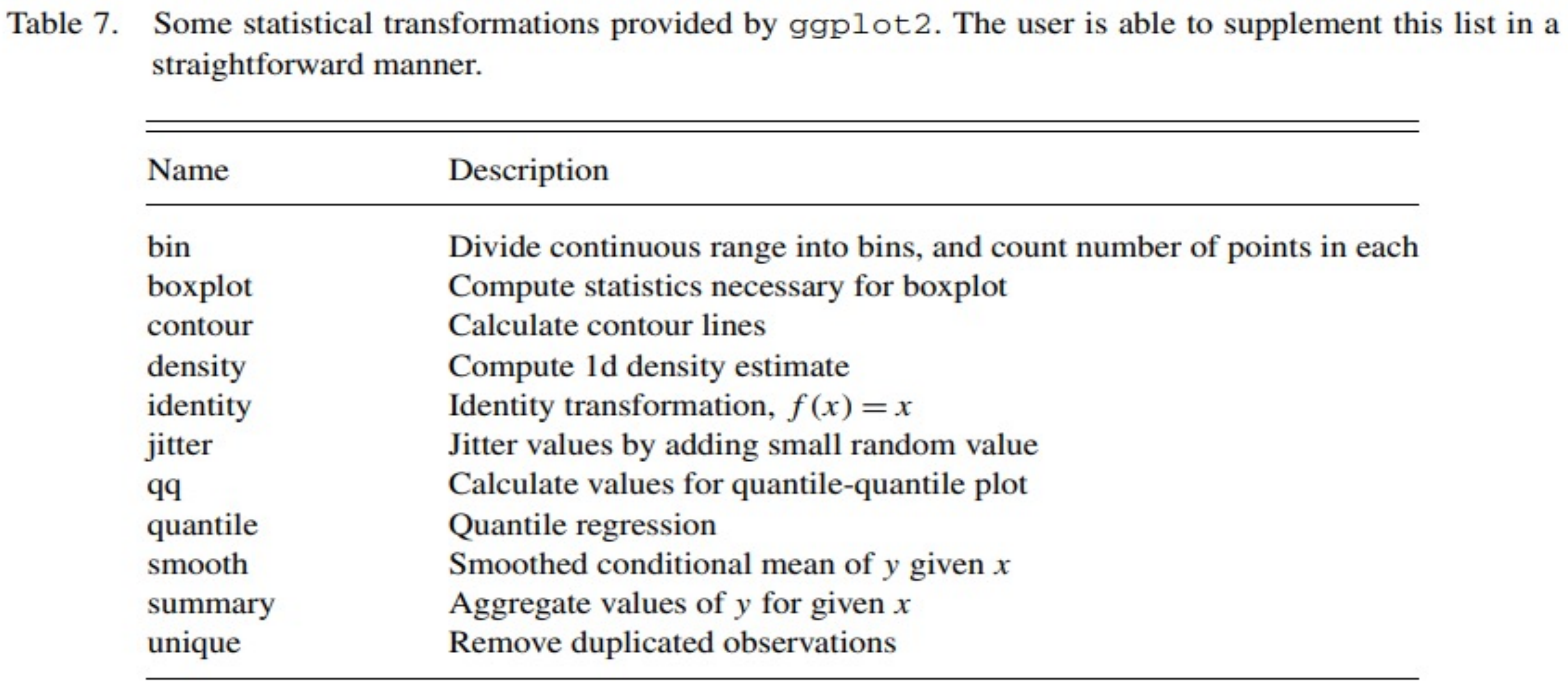


Table 7. Some statistical transformations provided by `ggplot2`. The user is able to supplement this list in a straightforward manner.

Name	Description
bin	Divide continuous range into bins, and count number of points in each
boxplot	Compute statistics necessary for boxplot
contour	Calculate contour lines
density	Compute 1d density estimate
identity	Identity transformation, $f(x) = x$
jitter	Jitter values by adding small random value
qq	Calculate values for quantile-quantile plot
quantile	Quantile regression
smooth	Smoothed conditional mean of y given x
summary	Aggregate values of y for given x
unique	Remove duplicated observations

Scales in ggplot2:

A scale is a function, and its inverse, along with a set of parameters. For example, the color gradient scale maps a segment of the real line to a path through a color space. The

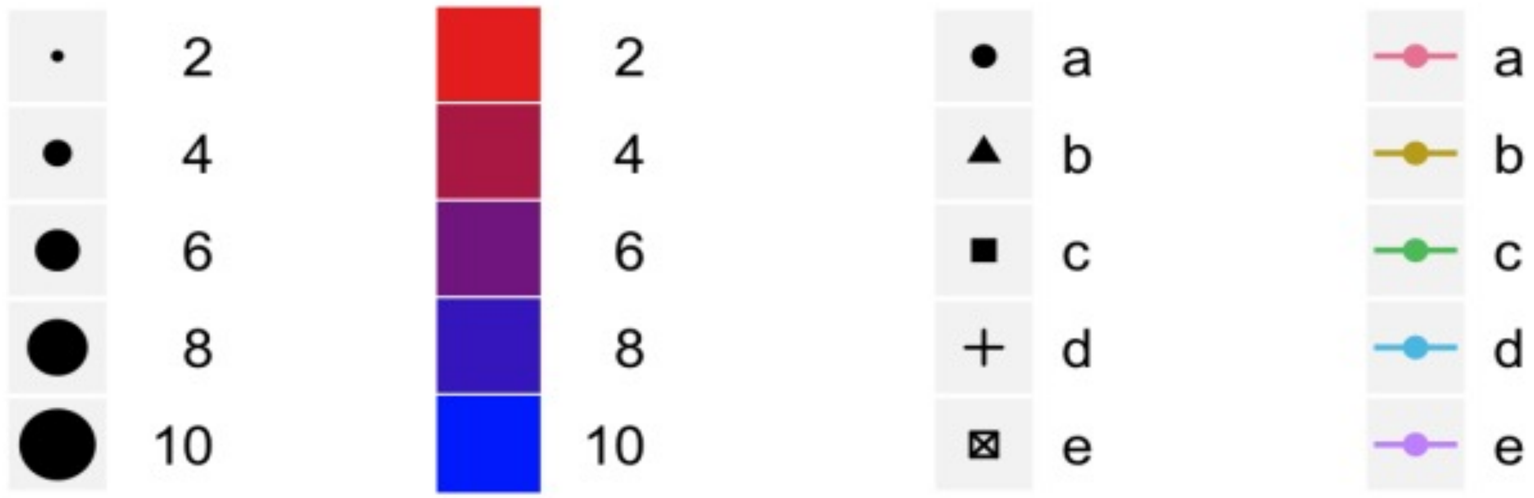
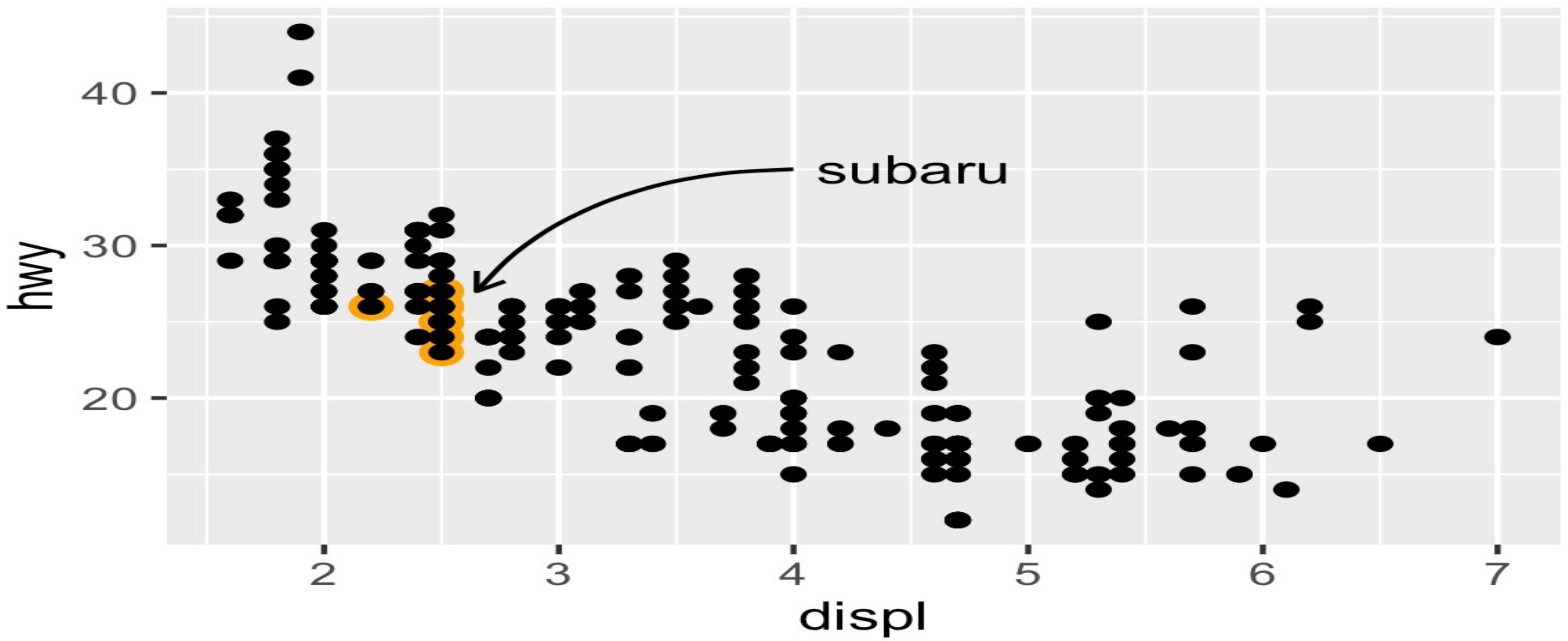


Figure 7. Examples of legends from four different scales. From left to right: continuous variable mapped to size and color, discrete variable mapped to shape and color. The legend automatically responds to the geoms used in the plot, from left to right: points, tiles, points, points and lines.

Scales:

- The scale of the layered grammar is equivalent to the SCALE and GUIDE of Wilkinson’s grammar.
- There are two types of guides: **scale guides and annotation guides**.
- In the layered grammar, the scale guides (axes and legends) are largely drawn automatically based on options supplied to the relative scales.
- Annotation guides, used to highlight important data points, are not needed because they can be constructed with creative use of geoms if data dependent, or if not, the underlying drawing system can be used directly.

ggplot: Annotation (will discuss more later!)



Coordinate system in ggplot2:

- A coordinate system, coord for short, maps the position of objects onto the plane of the plot. Position is often specified by two coordinates (x, y), but could be any number of coordinates.
- The Cartesian coordinate system is the most common coordinate system for two dimensions, whereas polar coordinates and various map projections are used less frequently.
- For **higher dimensions**, we have parallel coordinates (a projective geometry), mosaic plots (a hierarchical coordinate system), and linear projections onto the plane.
- **Coordinate systems control how the axes and grid lines are drawn.**

Example:

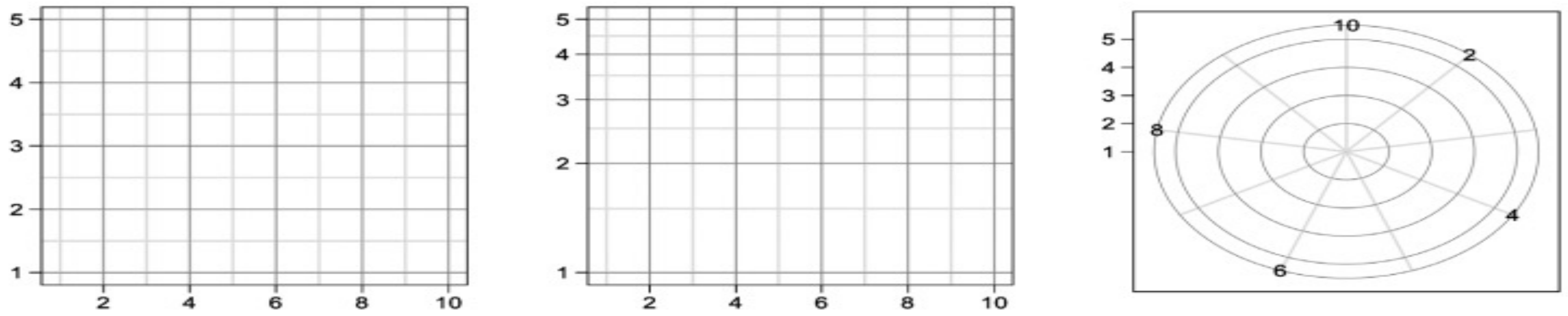


Figure 8. Examples of axes and grid lines for three coordinate systems: Cartesian, semi-log, and polar. The polar coordinate system illustrates the difficulties associated with non-Cartesian coordinates: it is hard to draw the axes well.

```
COORD: rect(dim(3), dim(1,2))
ELEMENT: point(position(x * y * z))

geom_point(aes(x, y)) + facet_grid(. ~ z)
```

Figure 9. Difference between GPL (top) and `ggplot2` (bottom) parameterizations. Note that z is included in the position specification for the GPL element.

ggplot2: Let’s see the codes now!

What is asked in the code/s shown below?

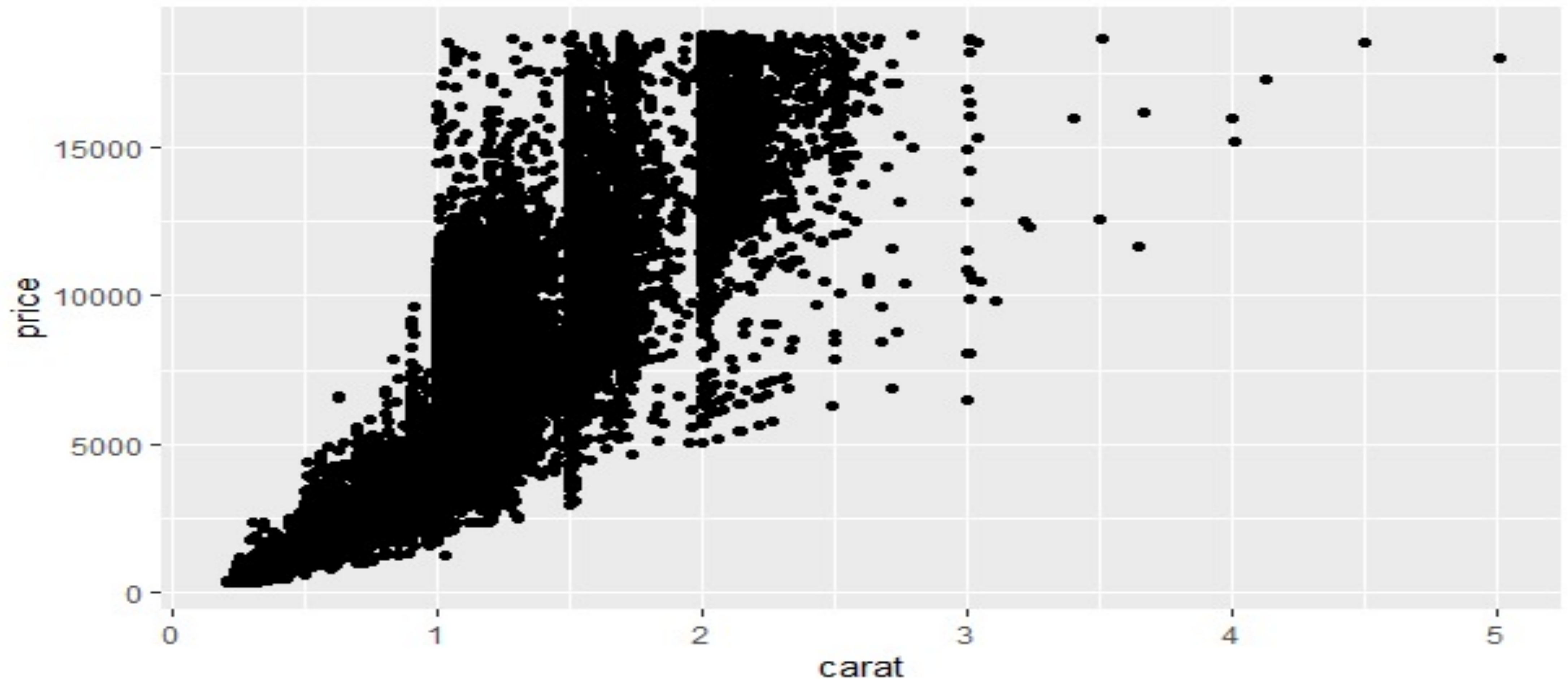
```
ggplot() +
layer(data = diamonds, mapping
= aes(x = carat, y = price),
geom = "point", stat = "identity",
position = "identity" ) +
        scale_y_continuous() +
        scale_x_continuous() +
        coord_cartesian()
```

```
ggplot(diamonds, aes(carat, price)) +
geom_point() +
scale_x_continuous() +
scale_y_continuous()
```

What happens if scales are not defined! Will this code works?

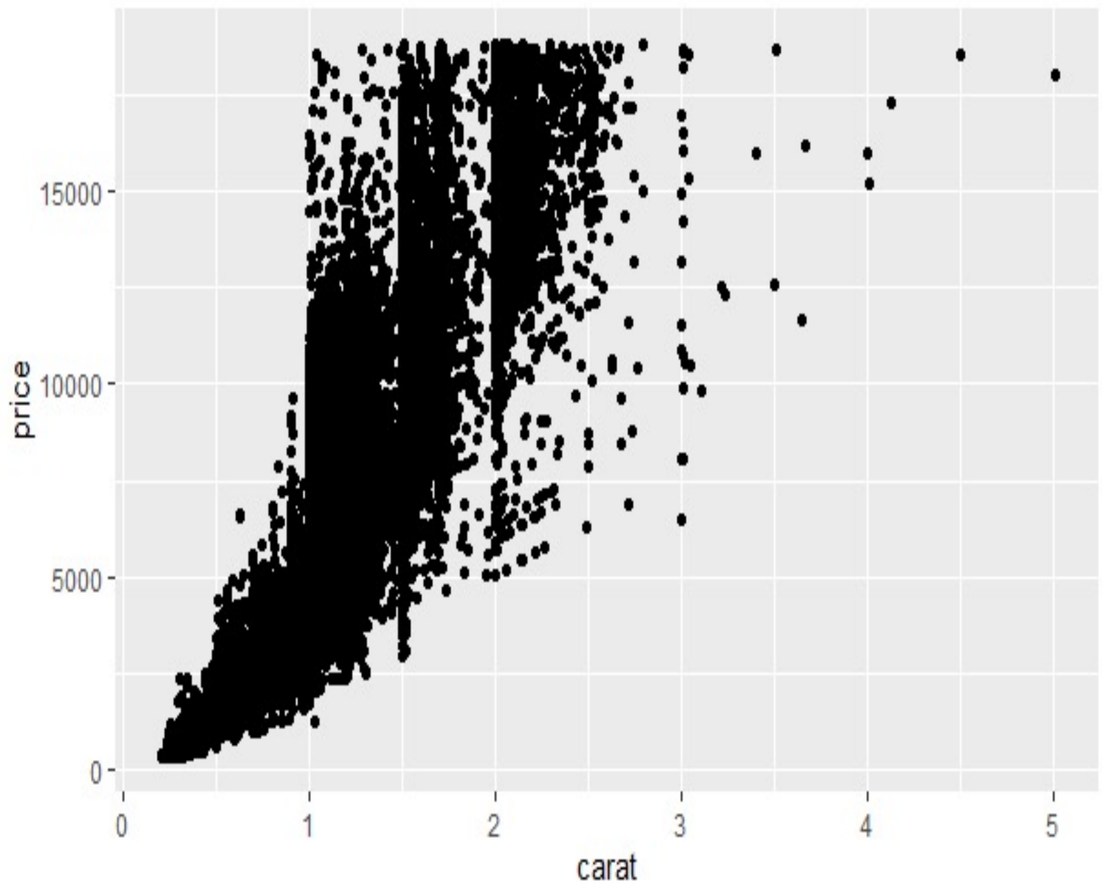
```
ggplot(diamonds, aes(carat, price)) +
geom_point()
```

Output:



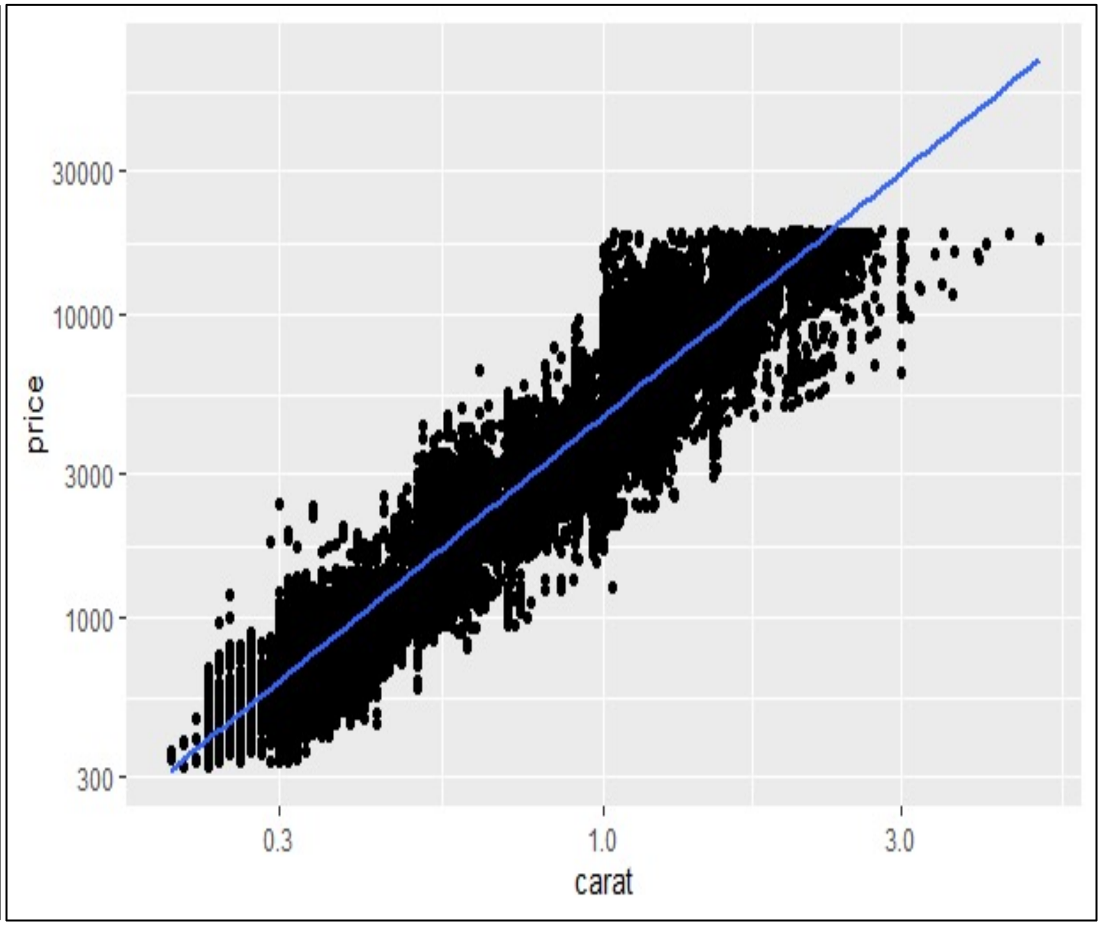
This will also work: why?

```
qplot(carat, price, data =
diamonds)
```



Log-transformed scatterplot of carat and price variables with linear model fitted line:

```
ggplot(diamonds, aes(carat,
price)) +
        geom_point() +
        stat_smooth(method = lm) +
        scale_x_log10() +
        scale_y_log10()
```



This is a “layered” version of the previous code for log-transformed scatterplot:

```
ggplot() + layer( data = diamonds, mapping = aes(x = carat, y = price),
geom = "point", stat = "identity", position = "identity"
) +
layer( data = diamonds, mapping = aes(x = carat, y = price),
geom = "smooth", position = "identity", stat = "smooth”, method = lm
) +
scale_y_log10() + scale_x_log10() +
coord_cartesian()
```

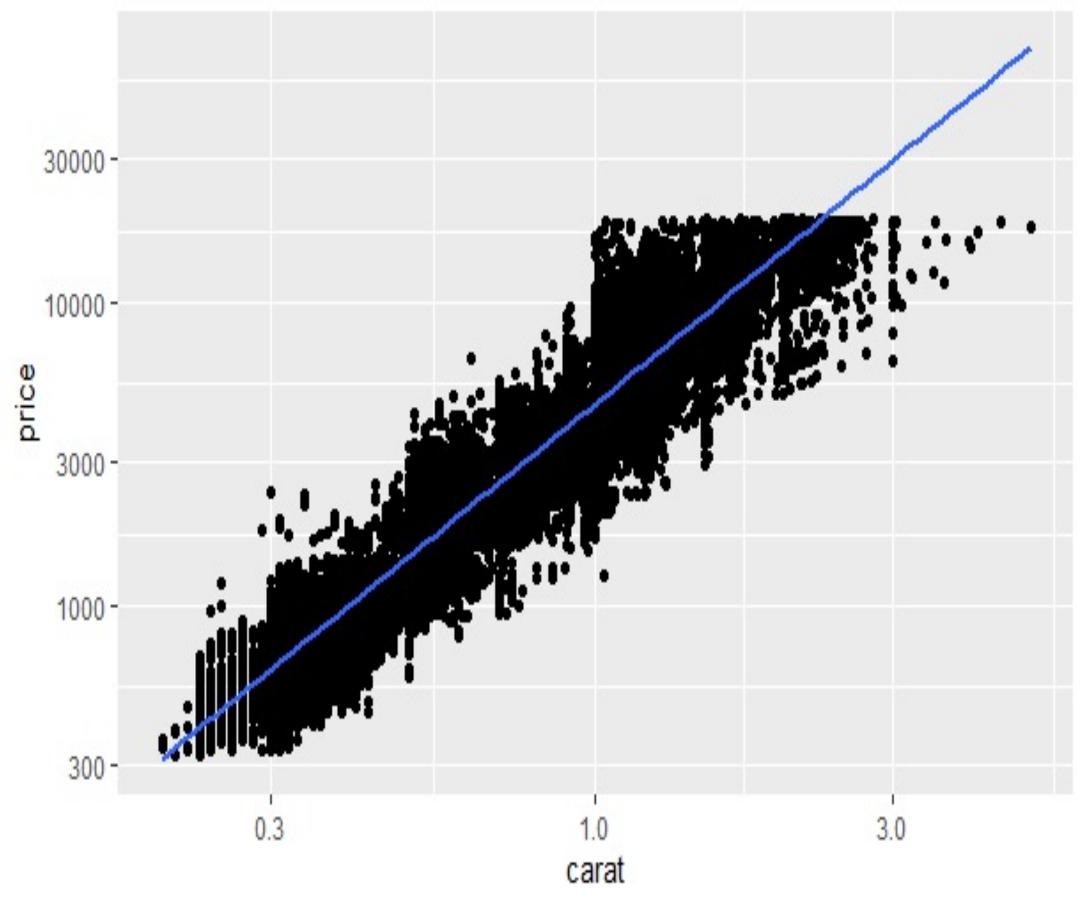
If you encounter the problem in the geom = “smooth” layer with method = lm code then try to run the full code without the “method = lm”

Which smoothing will be “default” now?

Why?

This will also work: why?

```
qplot(carat, price, data =
diamonds, geom = c("point",
"smooth"), method = "lm", log =
"xy" )
```



Implications: Each command works!

mapping = aes(..y..) after position = “identity” is optional for the 1st code!

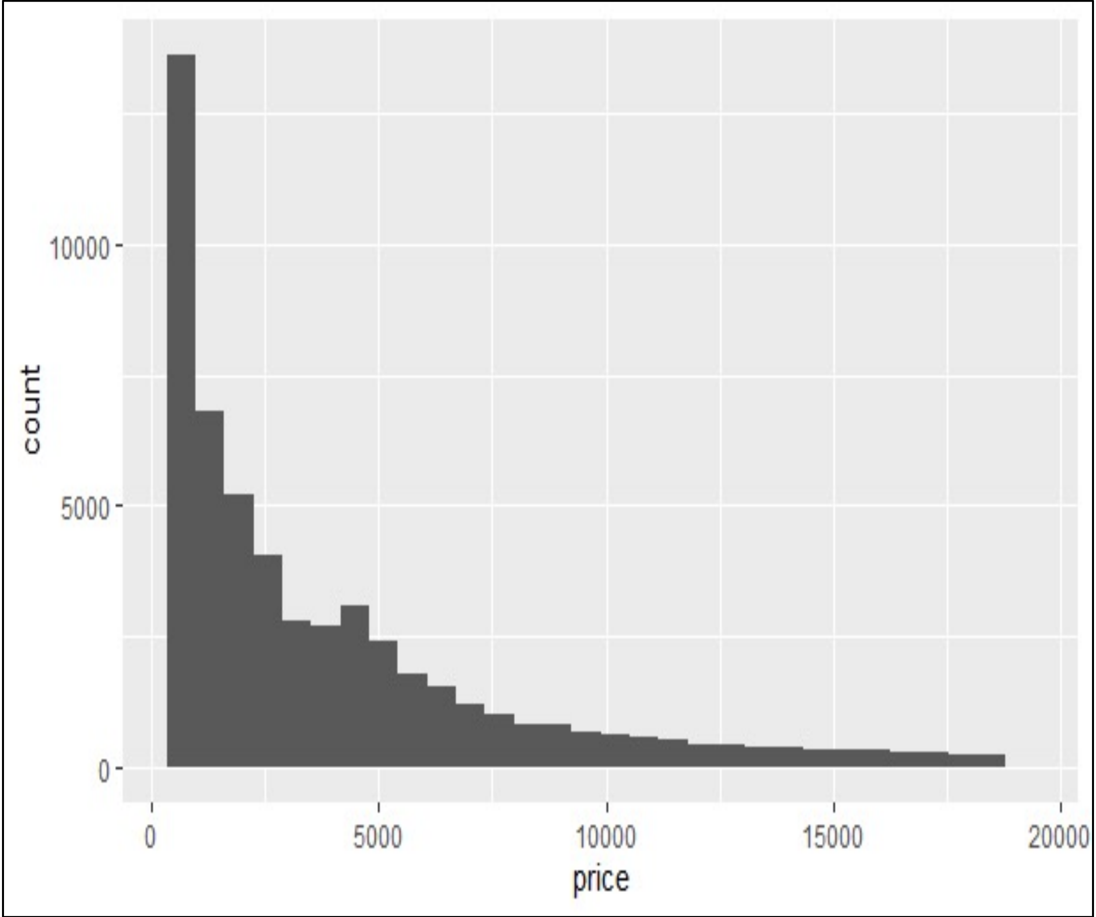
- ggplot(data = diamonds, mapping = aes(price)) + layer(geom = "bar", stat = "bin", position = “identity”)

OR

- ggplot(diamonds, aes(x = price)) + geom_histogram()

OR

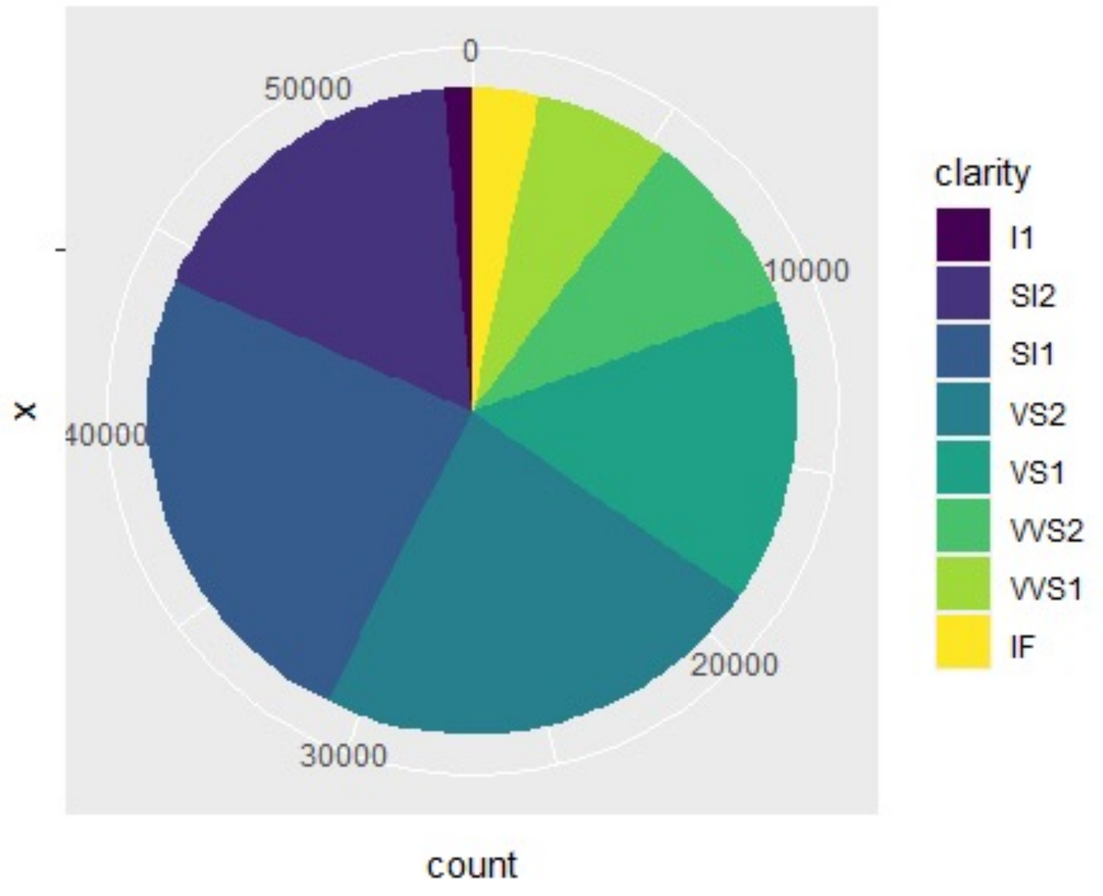
- qplot(price, data = diamonds, geom = "histogram")



Implications: barplot with polar coordinate 1!

```
ggplot(diamonds,aes(x = "",
fill=clarity)) +
geom_bar(width = 1) +
coord_polar (theta="y")
```

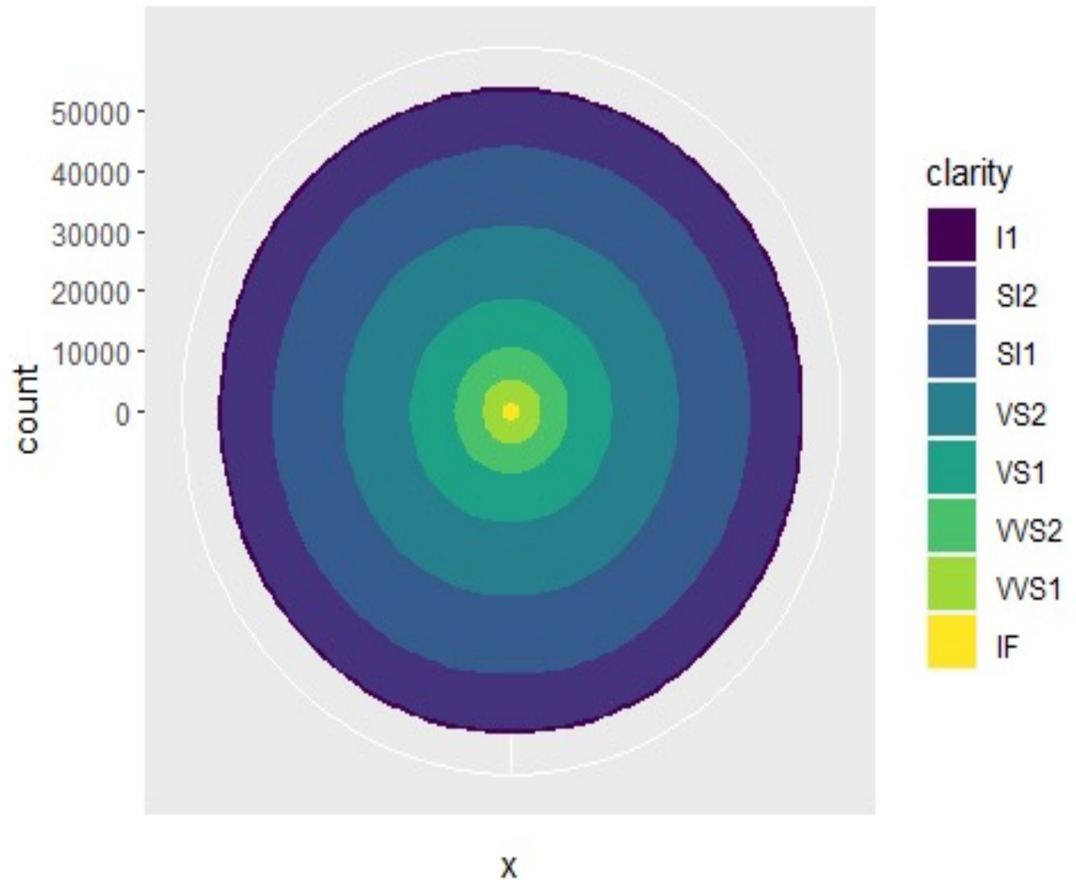
What is the interpretation of this plot?



Implications: barplot with polar coordinate 2!

```
ggplot(diamonds,aes(x = "",
fill=clarity)) +
geom_bar(width = 1) +
coord_polar(theta="x")
```

What is the interpretation of this plot?



Combining “ggplot2” and programming for complex plot:

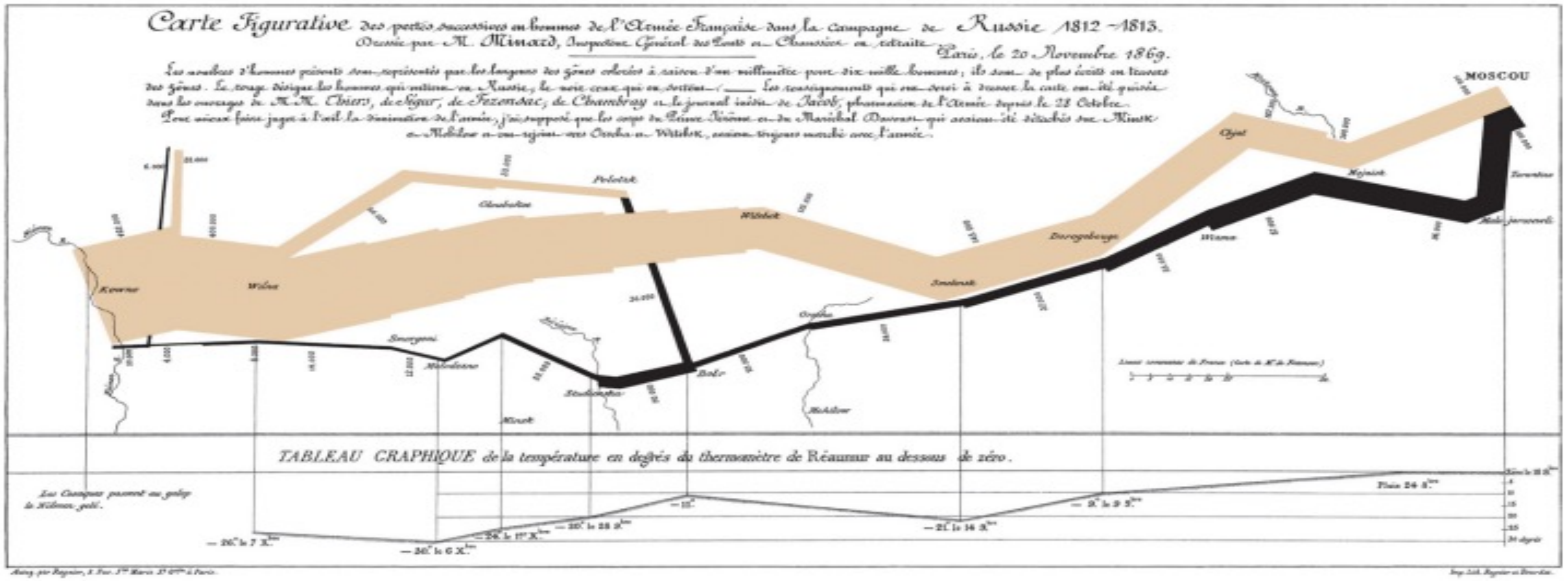


Figure 11. “Carte figurative des pertes successives en hommes de l’Armee Français dans la campagne de Russe 1812–1813” by Charles Joseph Minard. Public domain image from *http://en.wikipedia.org/wiki/File:Minard.png*.

Using “ggplot2” with a programming in R: http://vita.had.co.nz/papers/layered-grammar.pdf

plot_troops <- ggplot(troops, aes(long, lat)) + geom_path(aes(size = survivors, color = direction, group = group))

plot_both <- troops_plot + geom_text(aes(label = city), size = 4, data = cities)

plot_polished <- both + scale_size(to = c(1, 10), breaks = c(1, 2, 3) * 10^5, labels = comma(c(1, 2, 3) * 10^5)) + scale_color_manual(values = c("grey50","red")) + xlab(NULL) + ylab(NULL)

Outputs: You need "troops" data for this!

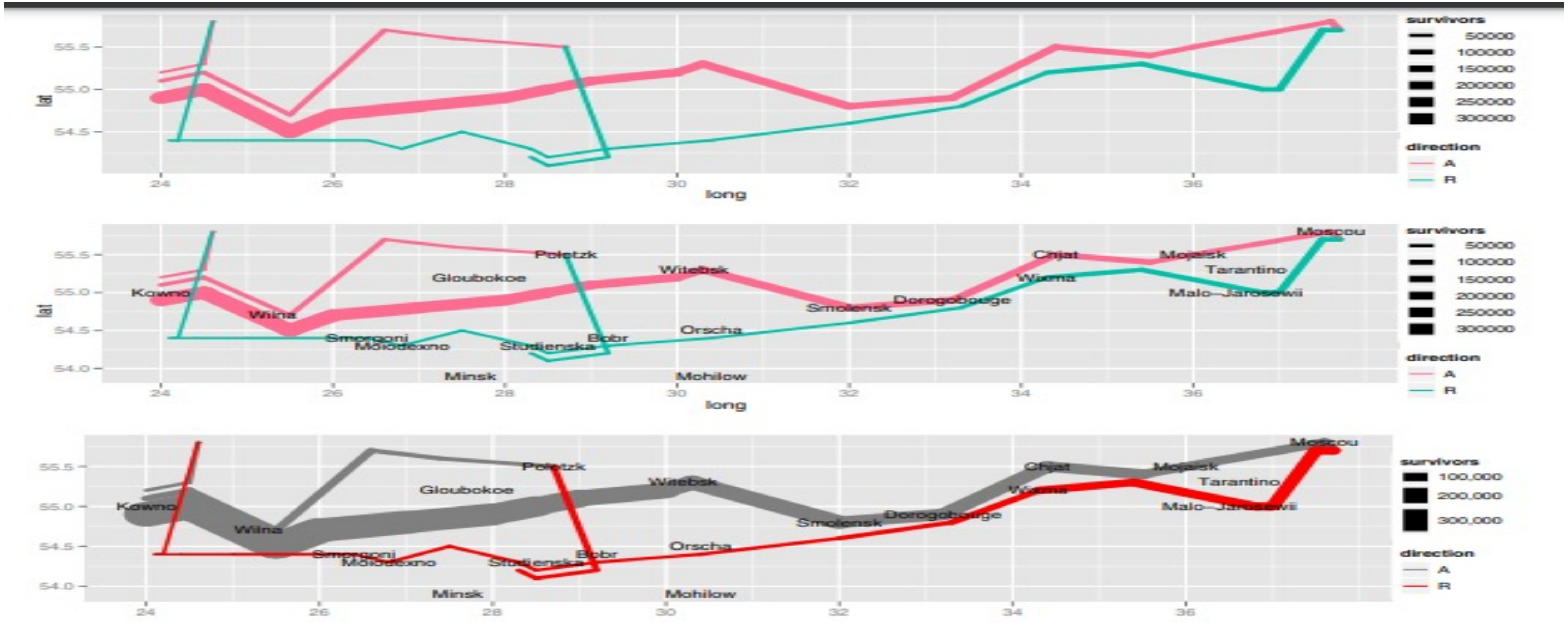


Figure 12. Iteratively reproducing the depiction of Napoleon's March by Minard. (Top) Displaying the key troop movement data. (Center) Adding town locations as reference points. (Bottom) Tweaking scales to produce polished plot.

Question/Queries?

Next class (Saturday):

- **Next class, we will discuss on the applications of the ggplot2 package**
- Read chapter 3 (Data visualization) of the course book for next class
- Read ggplot2 book: https://ggplot2-book.org/index.html
- **We will also start hypothesis testing with an example for testing validity of the correlation coefficient in the next class.**

Thank you!

@shitalbhandary