Intro to Data

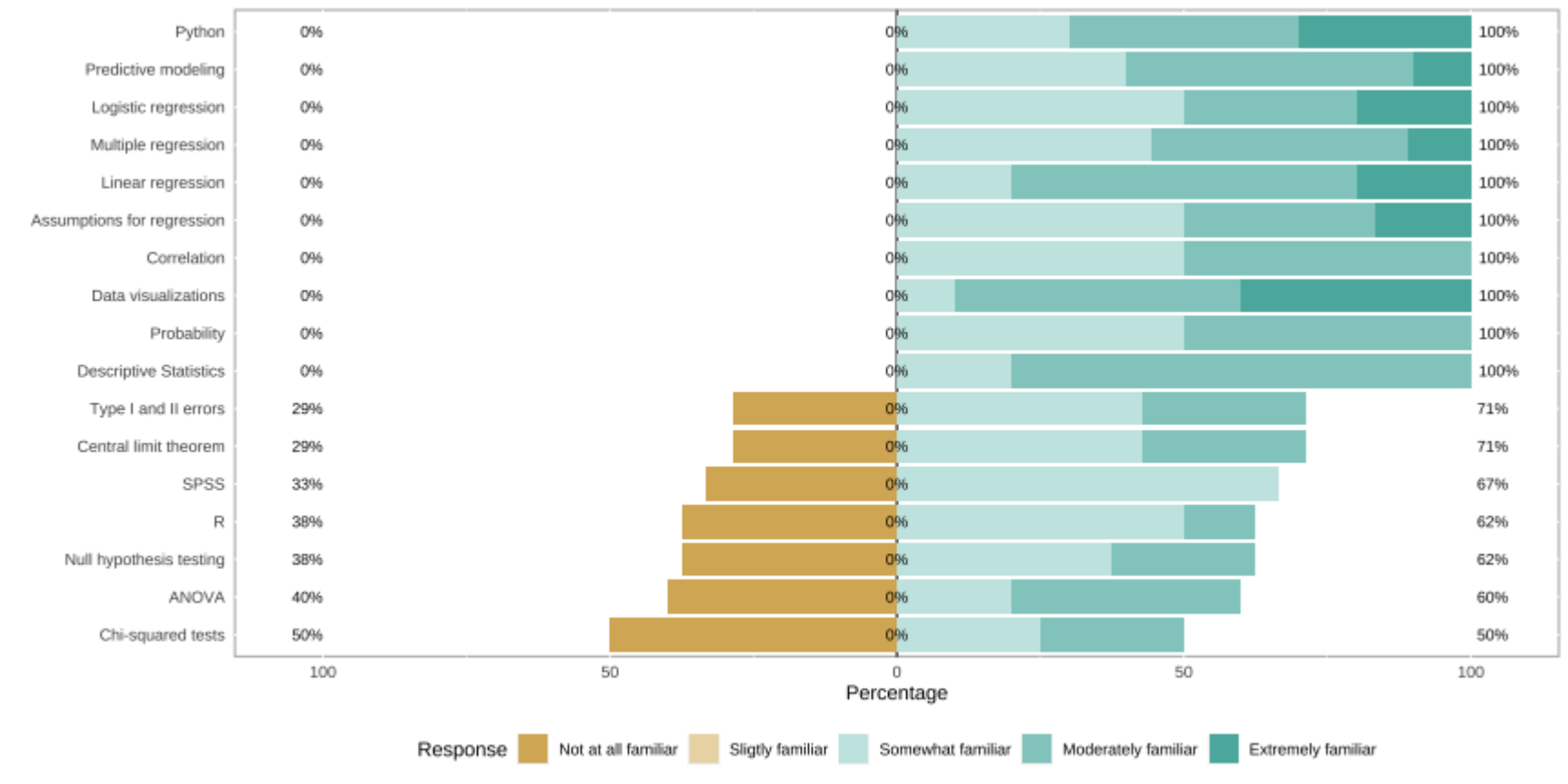
Computational Mathematics and Statistics

Jason Bryer, Ph.D.

January 23, 2024

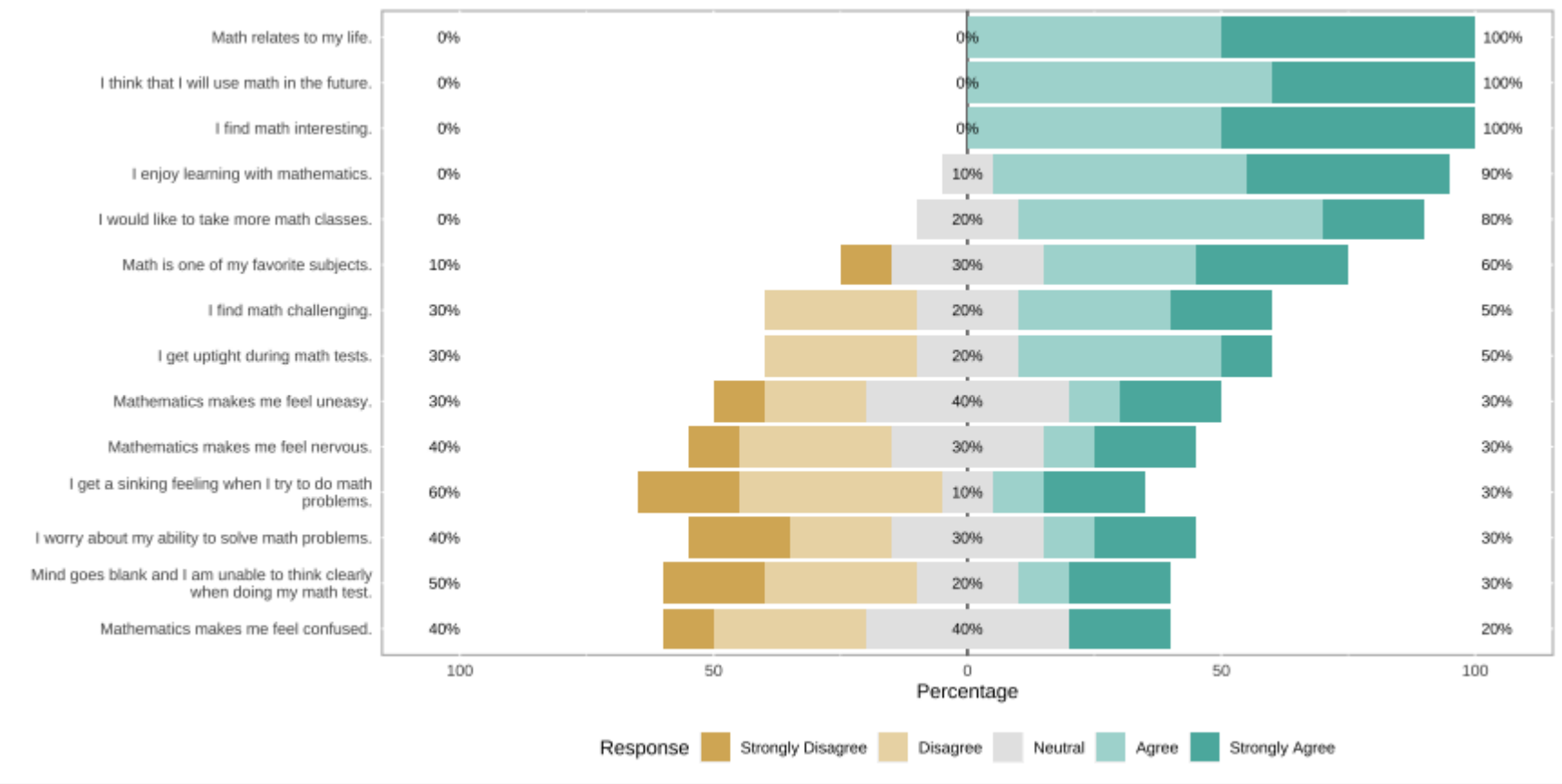
Familiarity with Statistical Topics

```
likert(stats.results) %>% plot(center = 2.5)
```



Math Anxiety Survey Scale

```
likert(mass.results) %>% plot()
```



Validity and Reliability

- An assessment is **valid** if it measures what it is supposed to measure.
- An assessment is **reliable** if it measures the same thing consistently and reproducibly.
- Trusted assessments must be reliable AND valid.

Not reliable

Reliable but not valid

Reliable and valid

More info: Riezler, S., & Hagmann, M. (2021). Validity, Reliability, and Significance Empirical Methods for NLP and Data Science

Case Study

Treating Chronic Fatigue Syndrome

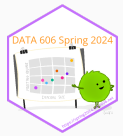
- Objective: Evaluate the effectiveness of cognitive-behavior therapy for chronic fatigue syndrome.
- Participant pool: 142 patients who were recruited from referrals by primary care physicians and consultants to a hospital clinic specializing in chronic fatigue syndrome.
- Actual participants: Only 60 of the 142 referred patients entered the study. Some were excluded because they didn't meet the diagnostic criteria, some had other health issues, and some refused to be a part of the study.

Source: Deale et. al. *Cognitive behavior therapy for chronic fatigue syndrome: A randomized controlled trial*. The American Journal of Psychiatry 154.3 (1997).

Study design

Patients randomly assigned to treatment and control groups, 30 patients in each group:

- **Treatment**: Cognitive behavior therapy -- collaborative, educative, and with a behavioral emphasis. Patients were shown on how activity could be increased steadily and safely without exacerbating symptoms.
- **Control**: Relaxation -- No advice was given about how activity could be increased. Instead progressive muscle relaxation, visualization, and rapid relaxation skills were taught.



Results

The table below shows the distribution of patients with good outcomes at 6-month follow-up. Note that 7 patients dropped out of the study: 3 from the treatment and 4 from the control group.

	Yes	No	Total
Treatment	19	8	27
Control	5	21	26

- Proportion with good outcomes in treatment group:

$$19/27 \approx 0.70 \rightarrow 70\%$$

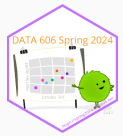
- Proportion with good outcomes in control group:

$$5/26 \approx 0.19 \rightarrow 19\%$$

Understanding the results

Do the data show a "real" difference between the groups?

- Suppose you flip a coin 100 times. While the chance a coin lands heads in any given coin flip is 50%, we probably won't observe exactly 50 heads. This type of fluctuation is part of almost any type of data generating process.
- The observed difference between the two groups (70 - 19 = 51%) may be real, or may be due to natural variation.
- Since the difference is quite large, it is more believable that the difference is real.
- We need statistical tools to determine if the difference is so large that we should reject the notion that it was due to chance.



Generalizing the results

Are the results of this study generalizable to all patients with chronic fatigue syndrome?

These patients had specific characteristics and volunteered to be a part of this study, therefore they may not be representative of all patients with chronic fatigue syndrome. While we cannot immediately generalize the results to all patients, this first study is encouraging. The method works for patients with some narrow set of characteristics, and that gives hope that it will work, at least to some degree, with other patients.

Sampling vs. Census

A census involves collecting data for the entire population of interest. This is problematic for several reasons, including:

- It can be difficult to complete a census: there always seem to be some individuals who are hard to locate or hard to measure. And these difficult-to-find people may have certain characteristics that distinguish them from the rest of the population.
- Populations rarely stand still. Even if you could take a census, the population changes constantly, so it's never possible to get a perfect measure.
- Taking a census may be more complex than sampling.

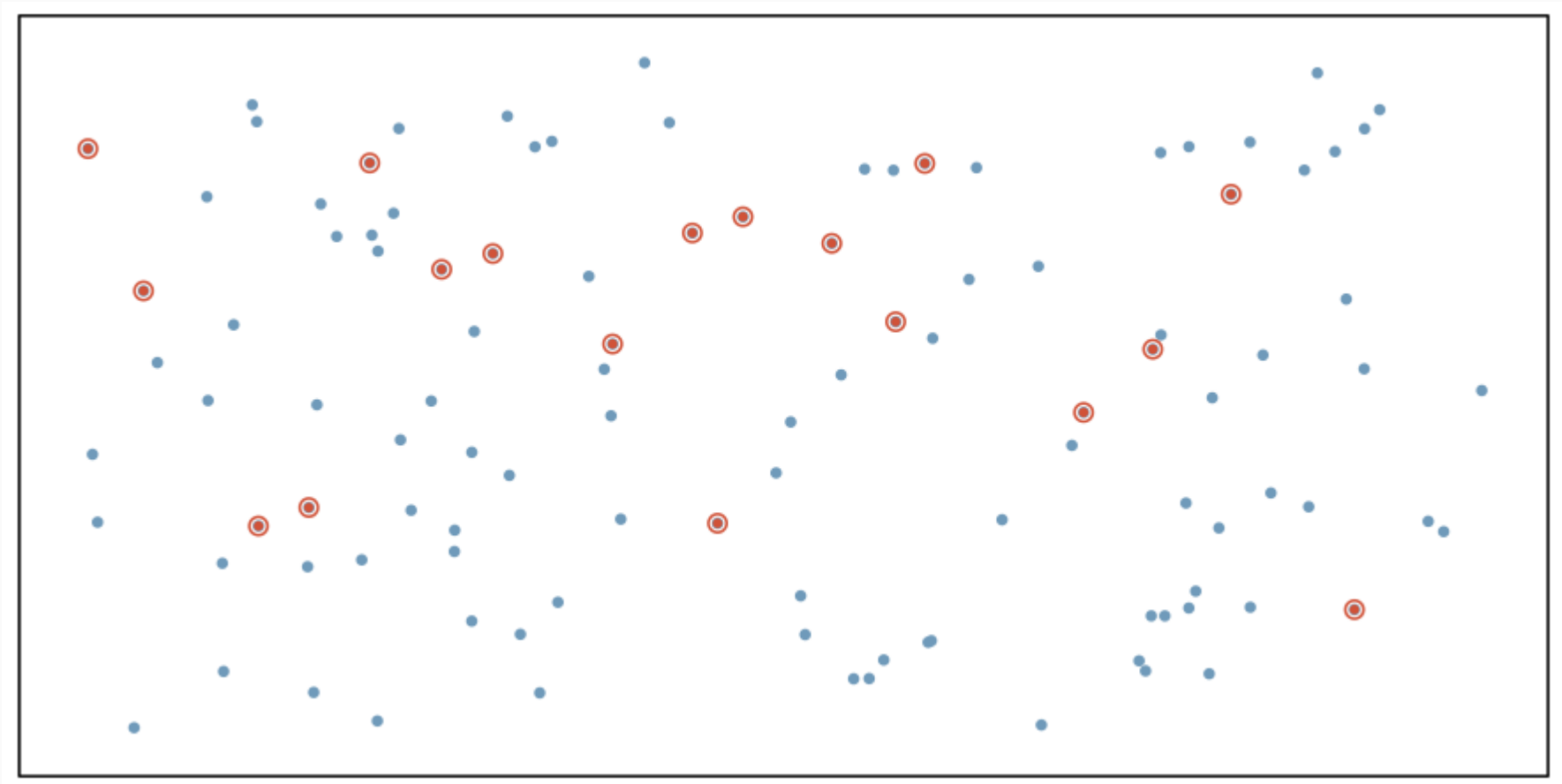
Sampling involves measuring a subset of the population of interest, usually randomly.

Sampling Bias

- **Non-response**: If only a small fraction of the randomly sampled people choose to respond to a survey, the sample may no longer be representative of the population.
- **Voluntary response**: Occurs when the sample consists of people who volunteer to respond because they have strong opinions on the issue. Such a sample will also not be representative of the population.
- **Convenience sample**: Individuals who are easily accessible are more likely to be included in the sample.

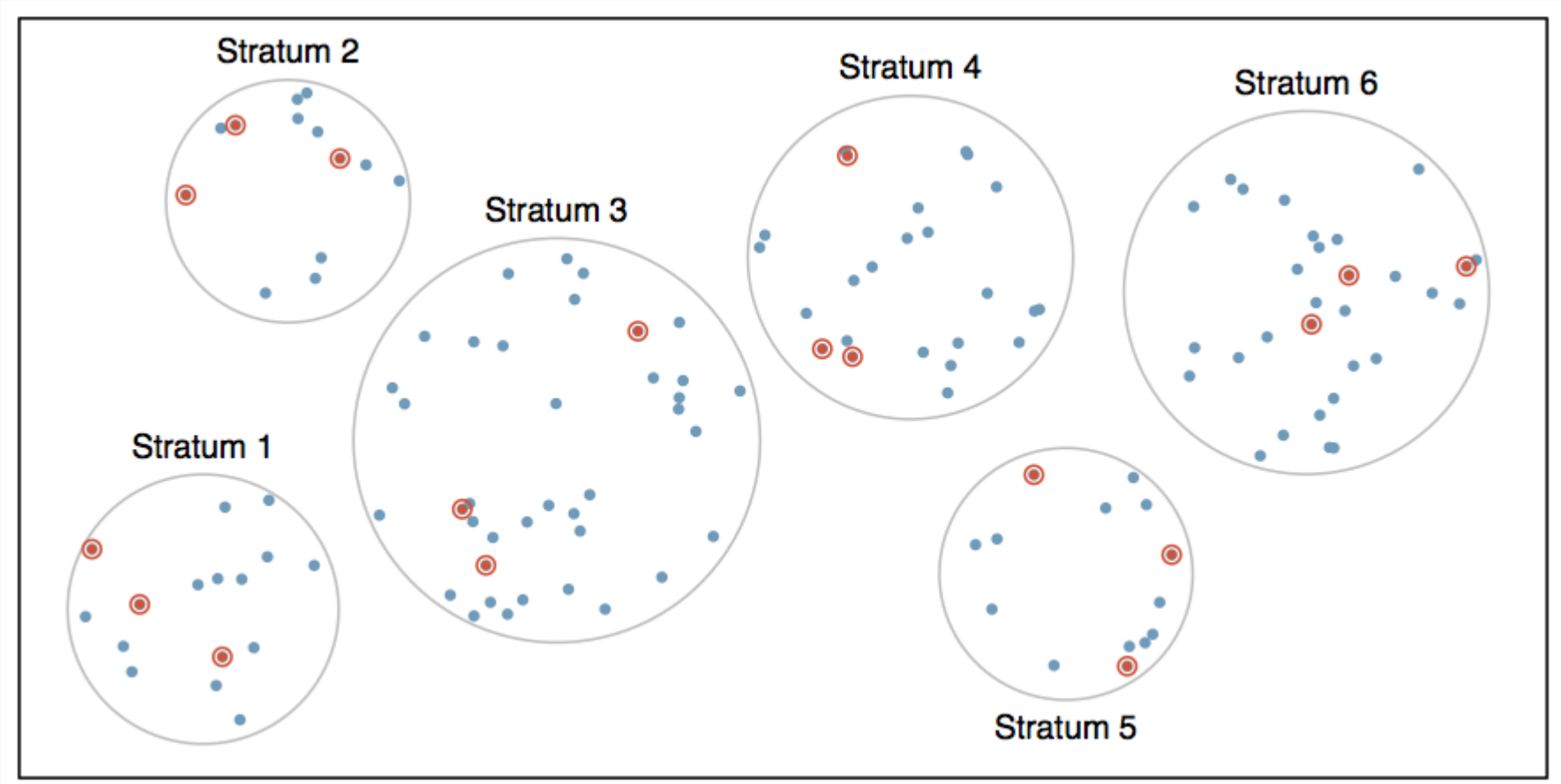
Simple Random Sampling

Randomly select cases from the population, where there is no implied connection between the points that are selected.



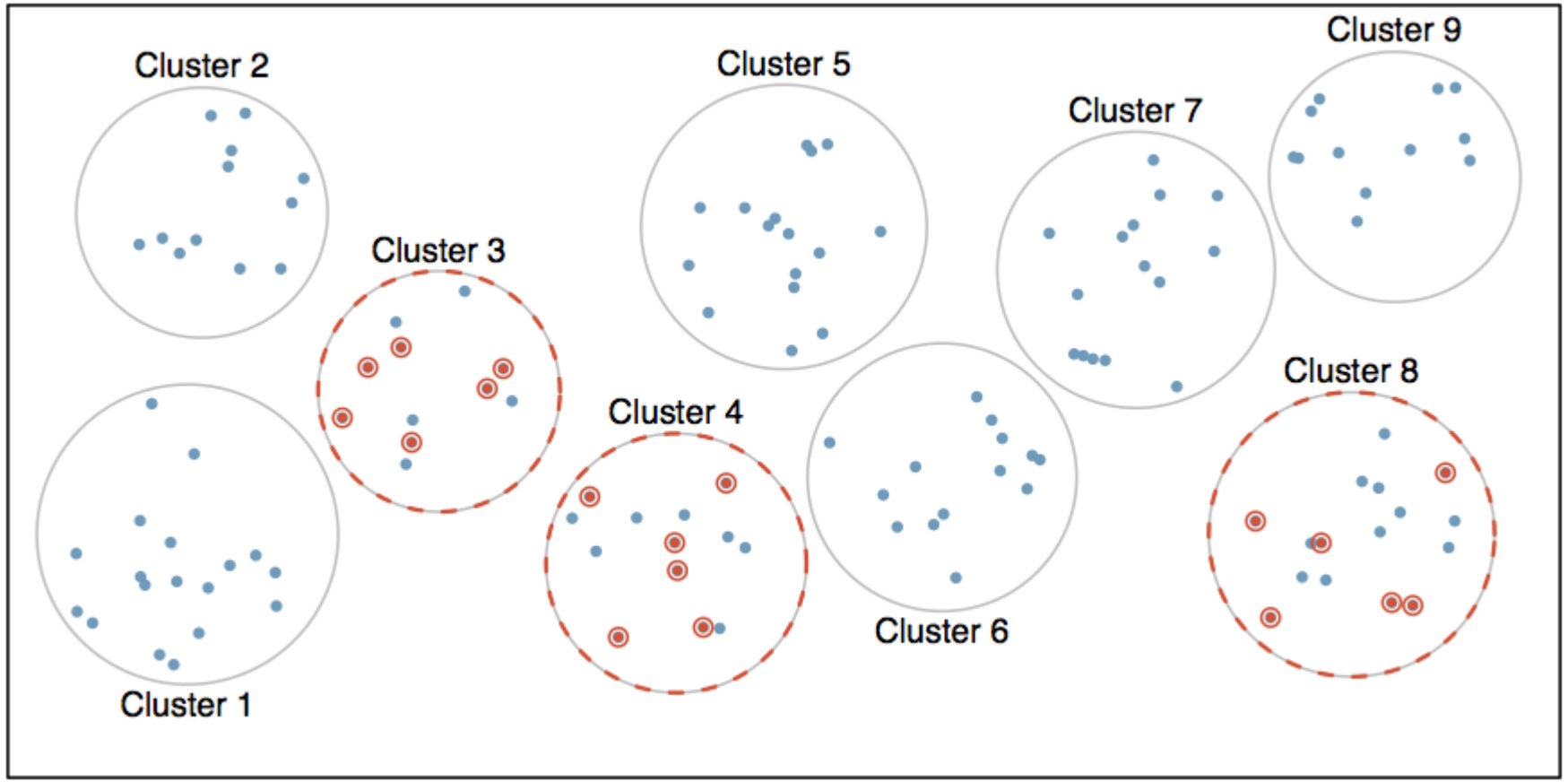
Stratified Sampling

Strata are made up of similar observations. We take a simple random sample from each stratum.



Cluster Sampling

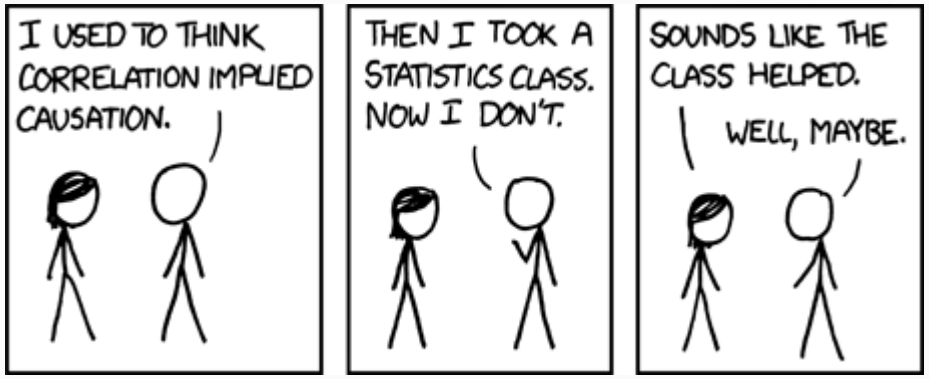
Clusters are usually not made up of homogeneous observations so we take random samples from random samples of clusters.



Observational Studies vs. Experiments

- **Observational study**: Researchers collect data in a way that does not directly interfere with how the data arise, i.e. they merely "observe", and can only establish an association between the explanatory and response variables.

- **Experiment**: Researchers randomly assign subjects to various treatments in order to establish causal connections between the explanatory and response variables.



Source: [XKCD 552 http://xkcd.com/552/](http://xkcd.com/552/)

Correlation does not imply causation!

Principles of experimental design

1. **Control**: Compare treatment of interest to a control group.

2. **Randomize**: Randomly assign subjects to treatments, and randomly sample from the population whenever possible.

3. **Replicate**: Within a study, replicate by collecting a sufficiently large sample. Or replicate the entire study.

4. **Block**: If there are variables that are known or suspected to affect the response variable, first group subjects into blocks based on these variables, and then randomize cases within each block to treatment groups.

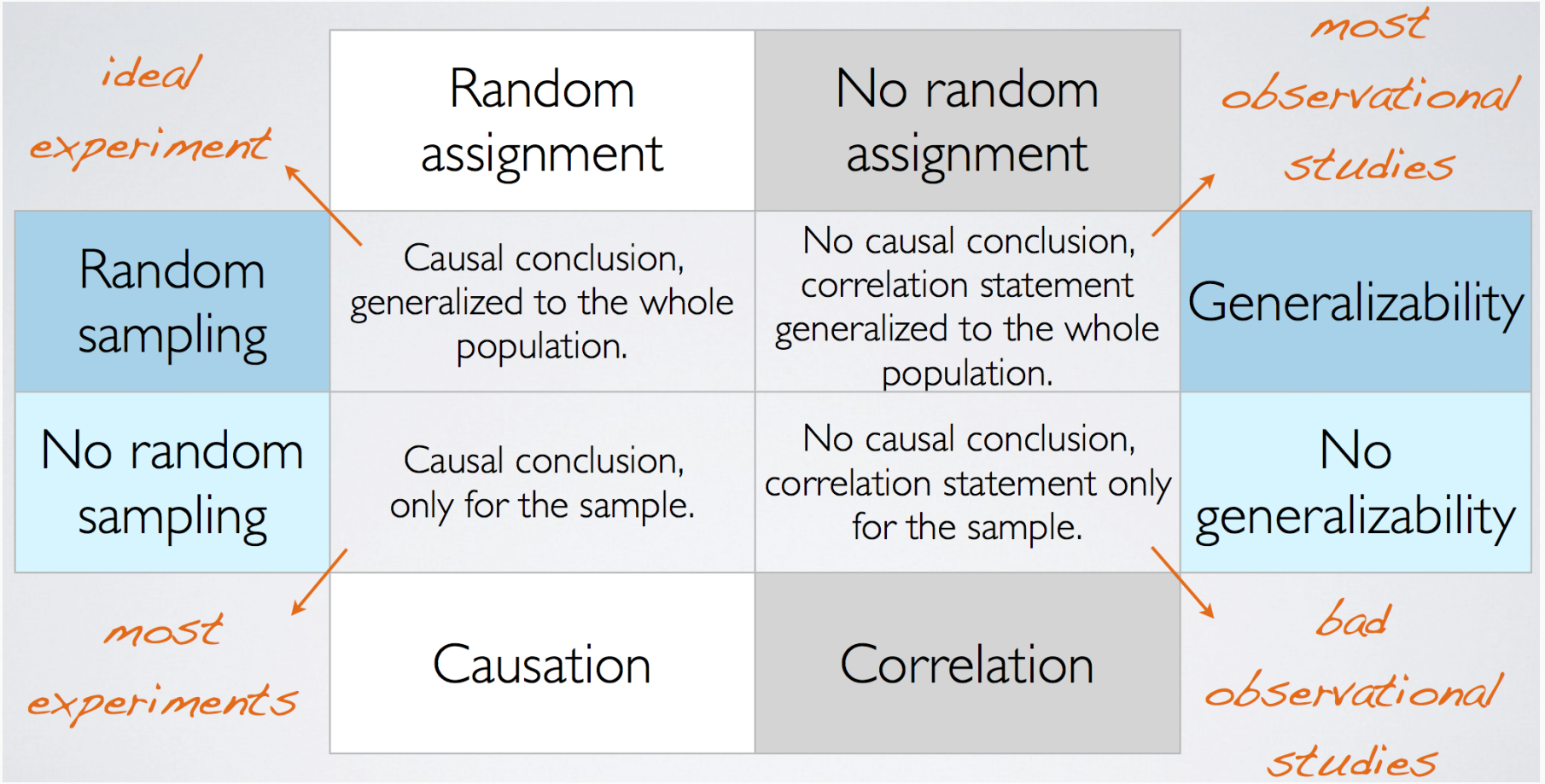
Difference between blocking and explanatory variables

- Factors are conditions we can impose on the experimental units.
- Blocking variables are characteristics that the experimental units come with, that we would like to control for.
- Blocking is like stratifying, except used in experimental settings when randomly assigning, as opposed to when sampling.

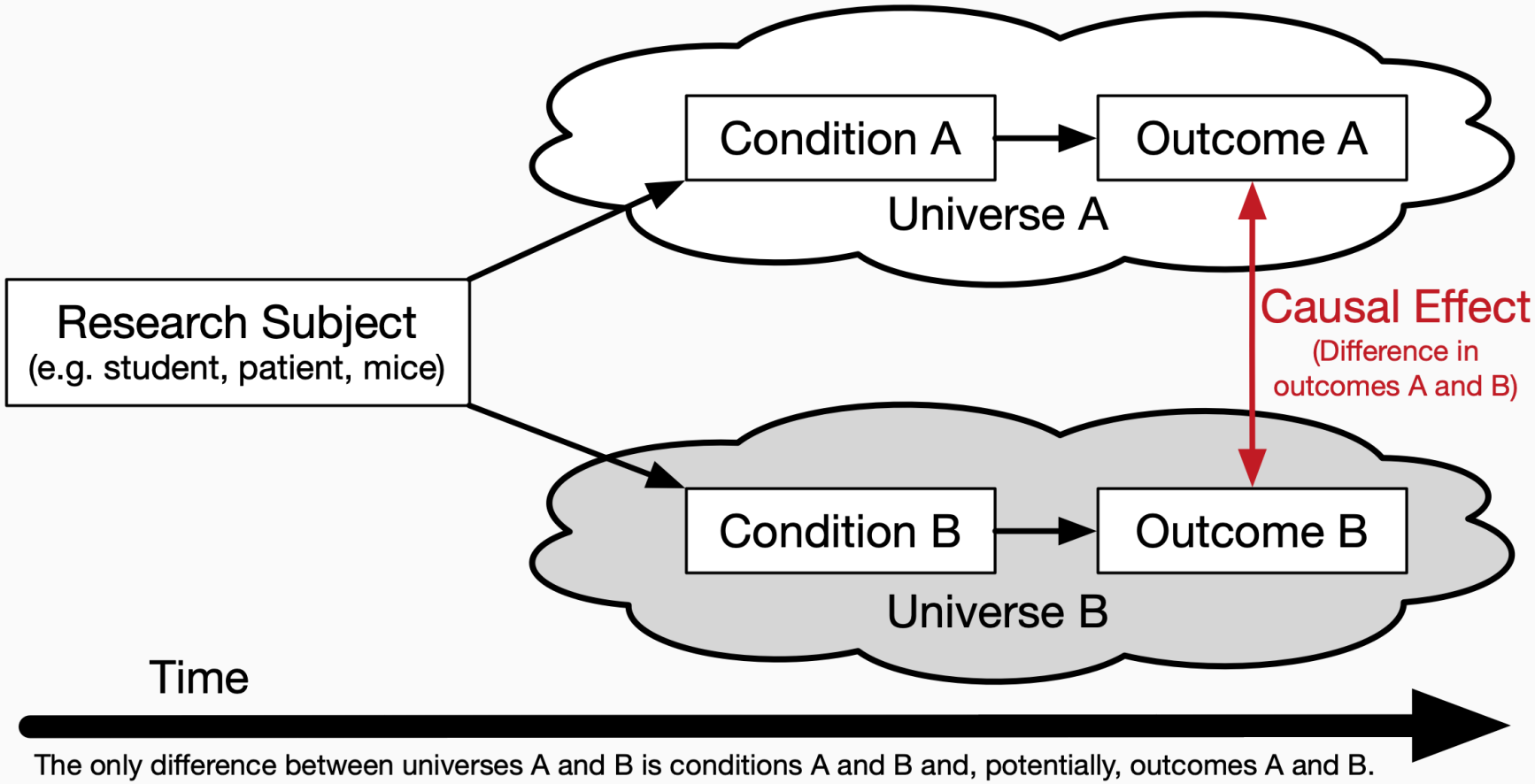
More experimental design terminology...

- **Placebo**: fake treatment, often used as the control group for medical studies
- **Placebo effect**: experimental units showing improvement simply because they believe they are receiving a special treatment
- **Blinding**: when experimental units do not know whether they are in the control or treatment group
- **Double-blind**: when both the experimental units and the researchers who interact with the patients do not know who is in the control and who is in the treatment group

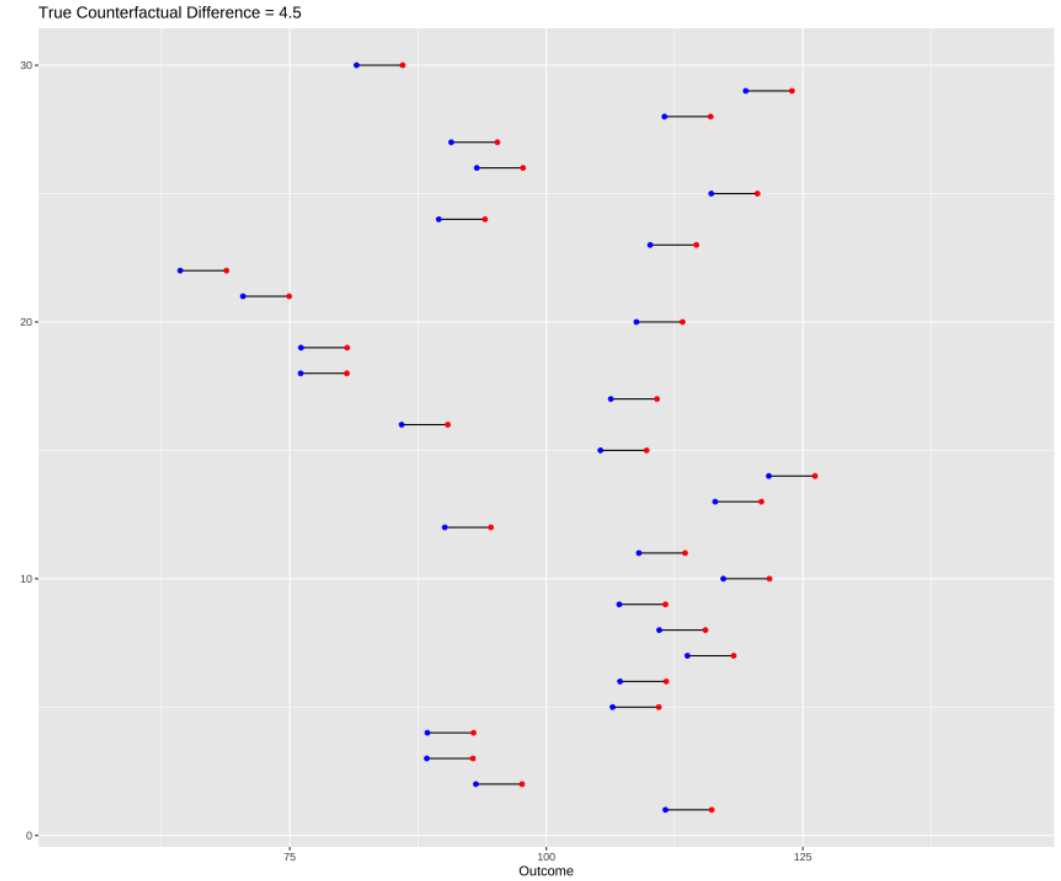
Random assignment vs. random sampling



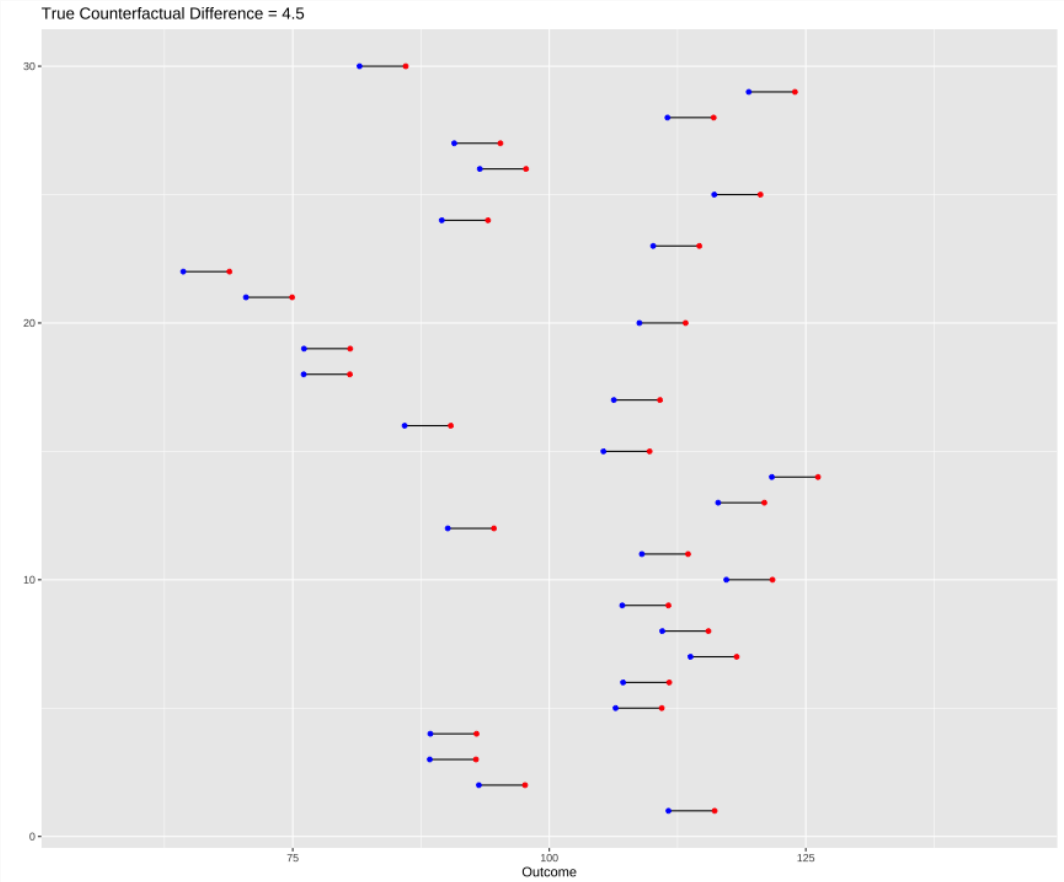
Causality

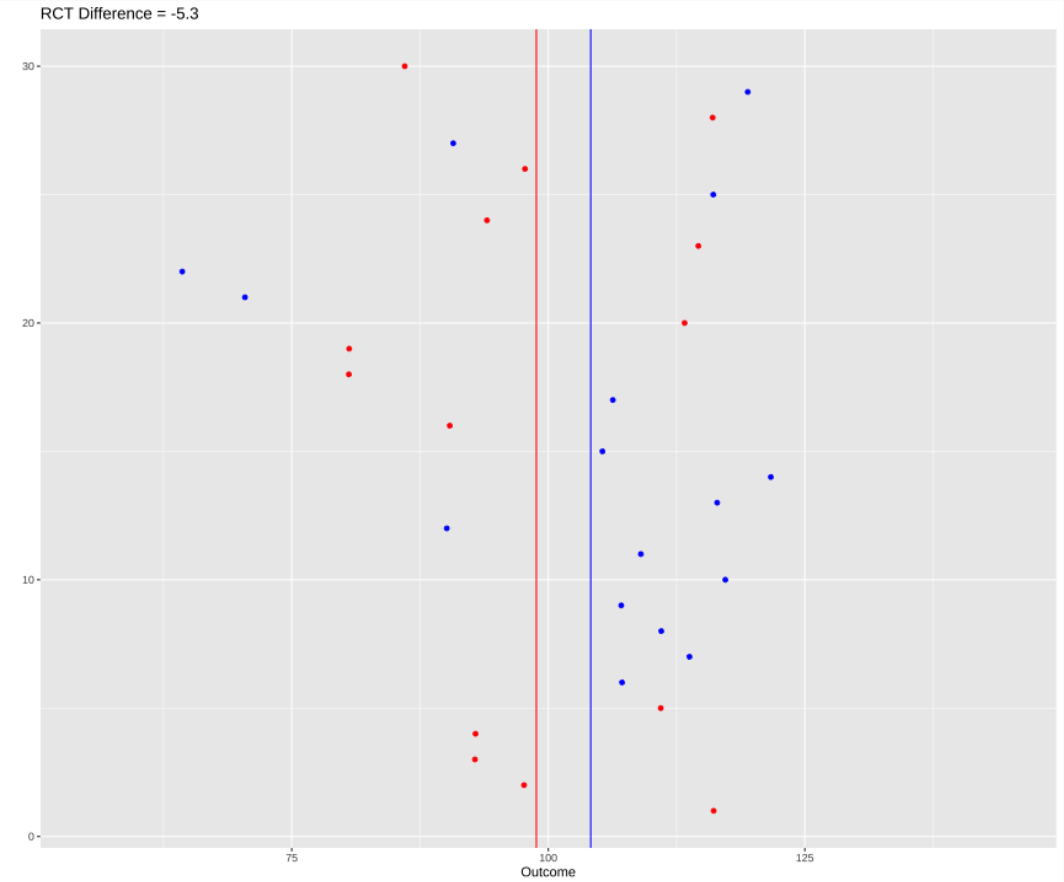

Condition A
Outcome A
Universe A
Causal Effect
(Difference in
outcomes A and B)
Research Subject
(e.g. student, patient, mice)
Condition B
Outcome B
Universe B
Time
The only difference between universes A and B is conditions A and B and, potentially, outcomes A and B.

Randomized Control Trials



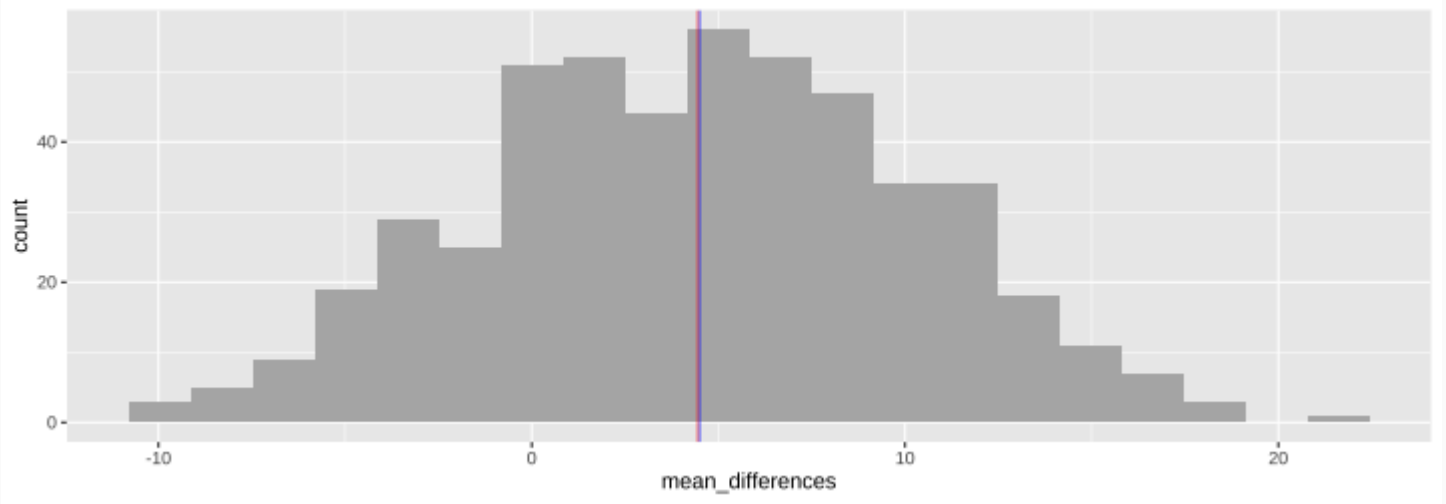
Randomized Control Trials



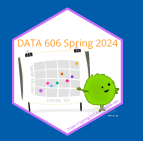


What if we take a lot of random samples?

```
mean_differences <- numeric(500)
for(i in 1:length(mean_differences)) {
    thedata$RCT_Assignment <- sample(c('placebo', 'treatment'), nrow(thedata), replace = TRUE)
    thedata$RCT_Value <- as.numeric(apply(thedata, 1,
                    FUN = function(x) { return(x[x['RCT_Assignment']]) }))
    tab.out <- describeBy(thedata$RCT_Value, group = thedata$RCT_Assignment, mat = TRUE, skew = FALSE)
    mean_differences[i] <- diff(tab.out$mean)
}
ggplot() + geom_histogram(aes(x = mean_differences), bins = 20, fill = 'grey70') +
    geom_vline(xintercept = mean(mean_differences), color = 'red', alpha = 0.5) +
    geom_vline(xintercept = pop.sd * pop.es, color = 'blue', alpha = 0.5)
```



Data Visualization



Grammer of Graphics

Data Visualizations with ggplot2

- `ggplot2` is an R package that provides an alternative framework based upon Wilkinson's (2005) Grammar of Graphics.
- `ggplot2` is, in general, more flexible for creating "prettier" and complex plots.
- Works by creating layers of different types of objects/geometries (i.e. bars, points, lines, polygons, etc.) `ggplot2` has at least three ways of creating plots:
    1. `qplot`
    2. `ggplot(...) + geom_XXX(...) + ...`
    3. `ggplot(...) + layer(...)`
- We will focus only on the second.

Parts of a `ggplot2` Statement

- Data

  ```
  ggplot(myDataFrame, aes(x=x, y=y))
  ```

- Layers

  ```
  geom_point(), geom_histogram()
  ```

- Facets

  ```
  facet_wrap(~ cut), facet_grid(~ cut)
  ```

- Scales

  ```
  scale_y_log10()
  ```

- Other options

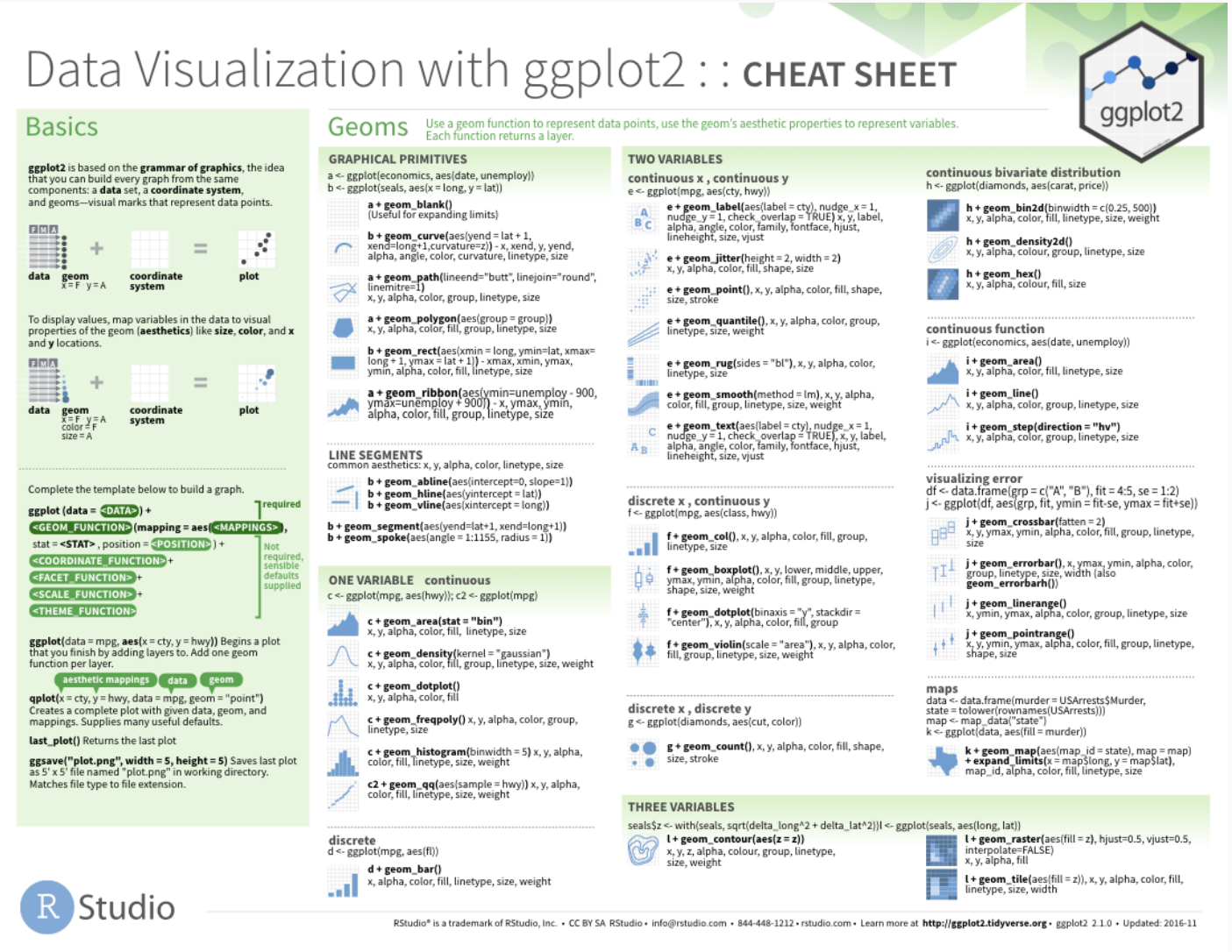
  ```
  ggtitle('my title'), ylim(c(0, 10000)), xlab('x-axis label')
  ```

Lots of geoms

```
ls('package:ggplot2')[grep('^geom_', ls('package:ggplot2'))]
```

```
##  [1] "geom_abline"            "geom_area"              "geom_bar"
##  [4] "geom_bin_2d"            "geom_bin2d"             "geom_blank"
##  [7] "geom_boxplot"           "geom_col"               "geom_contour"
## [10] "geom_contour_filled"    "geom_count"             "geom_crossbar"
## [13] "geom_curve"             "geom_density"           "geom_density_2d"
## [16] "geom_density_2d_filled" "geom_density2d"         "geom_density2d_filled"
## [19] "geom_dotplot"           "geom_errorbar"          "geom_errorbarh"
## [22] "geom_freqpoly"          "geom_function"          "geom_hex"
## [25] "geom_histogram"         "geom_hline"             "geom_jitter"
## [28] "geom_label"             "geom_line"              "geom_linerange"
## [31] "geom_map"               "geom_path"              "geom_point"
## [34] "geom_pointrange"        "geom_polygon"           "geom_qq"
## [37] "geom_qq_line"           "geom_quantile"          "geom_raster"
## [40] "geom_rect"              "geom_ribbon"            "geom_rug"
## [43] "geom_segment"           "geom_sf"                "geom_sf_label"
## [46] "geom_sf_text"           "geom_smooth"            "geom_spoke"
## [49] "geom_step"              "geom_text"              "geom_tile"
## [52] "geom_violin"            "geom_vline"
```

Data Visualization Cheat Sheet



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.

data | geom (x = F, y = A) | coordinate system | plot

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

data | geom (x = F, y = A, color = F, size = A) | coordinate system | plot

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.

aesthetic mappings | data | geom

qplot(x = cty, y = hwy, data = mpg, geom = "point") Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

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.

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() (Useful for expanding limits)

b + geom_curve(aes(yend = lat + 1, xend=long+1,curvature=z)) - 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(group = group)) x, y, alpha, color, fill, group, 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

continuous x , continuous y

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

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

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

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, check_overlap = TRUE), x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

discrete x , continuous y

```
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

discrete x , discrete y

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

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

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_density2d() x, y, alpha, colour, group, linetype, size

h + geom_hex() x, y, alpha, colour, 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, colour, group, linetype, size, weight

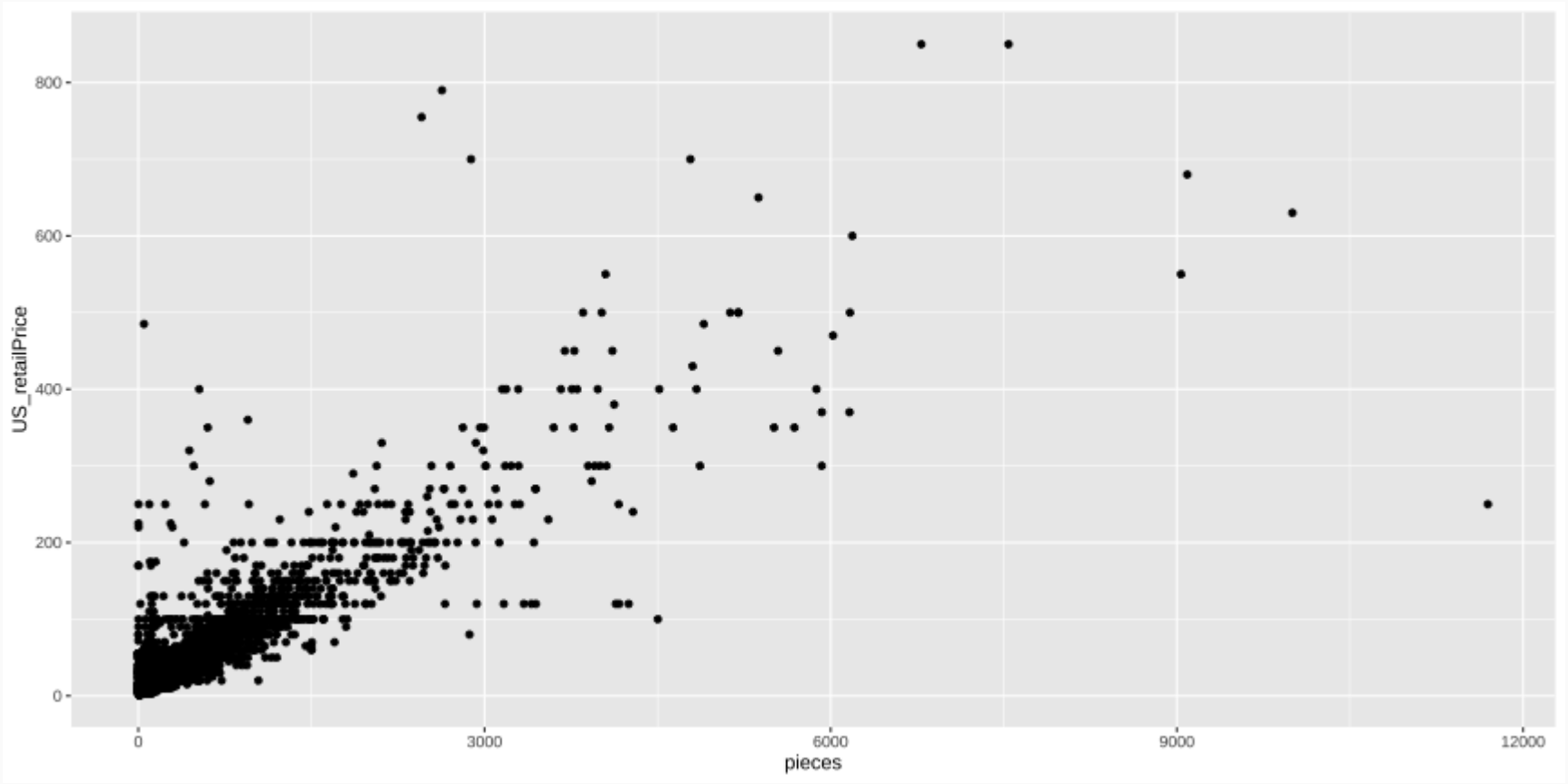
l + geom_raster(aes(fill = z), hjust=0.5, vjust=0.5, interpolate=FALSE) x, y, alpha, fill

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

RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at **http://ggplot2.tidyverse.org** • ggplot2 2.1.0 • Updated: 2016-11

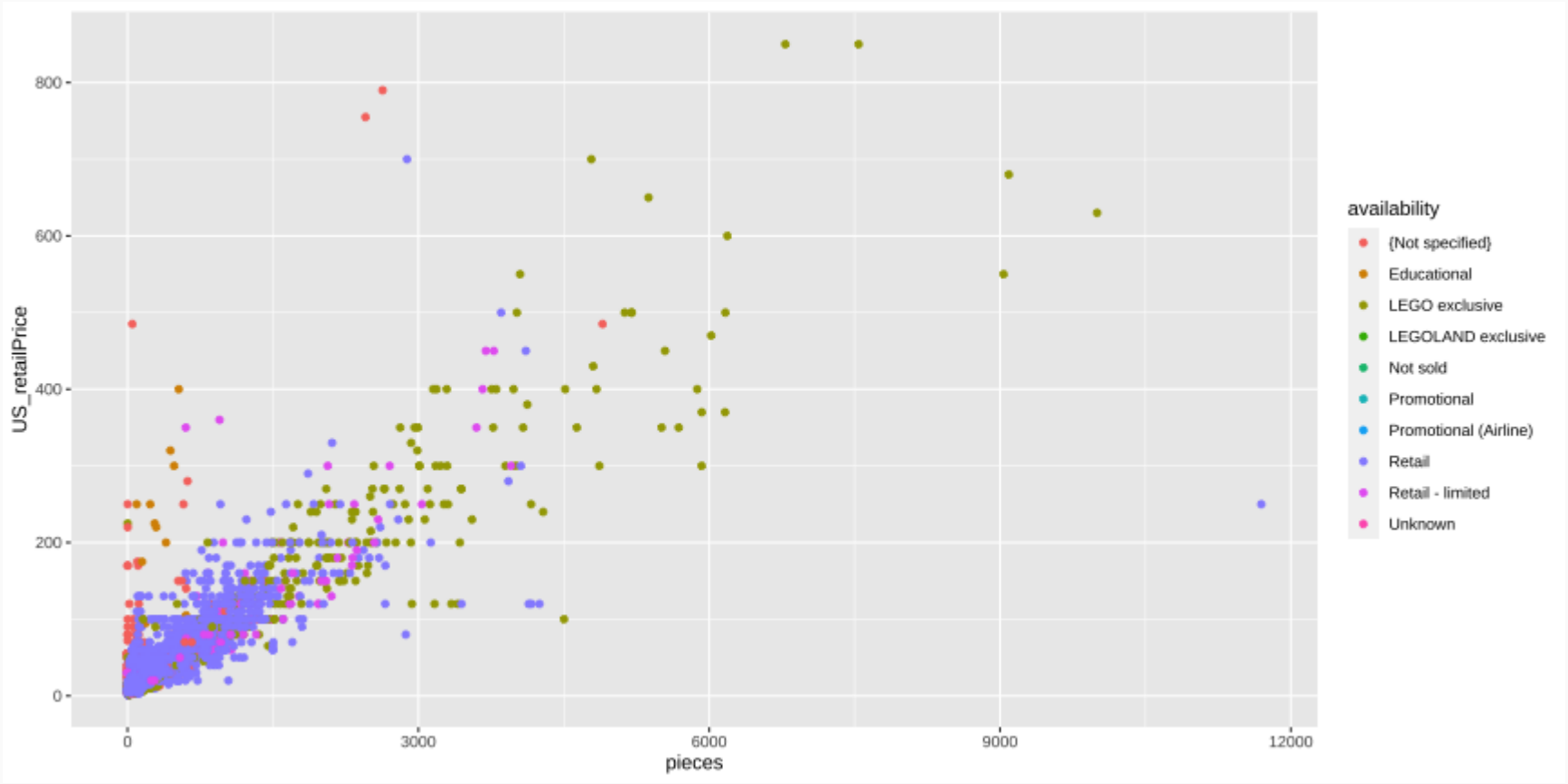
Scatterplot

```
ggplot(legosets, aes(x=pieces, y=US_retailPrice)) + geom_point()
```



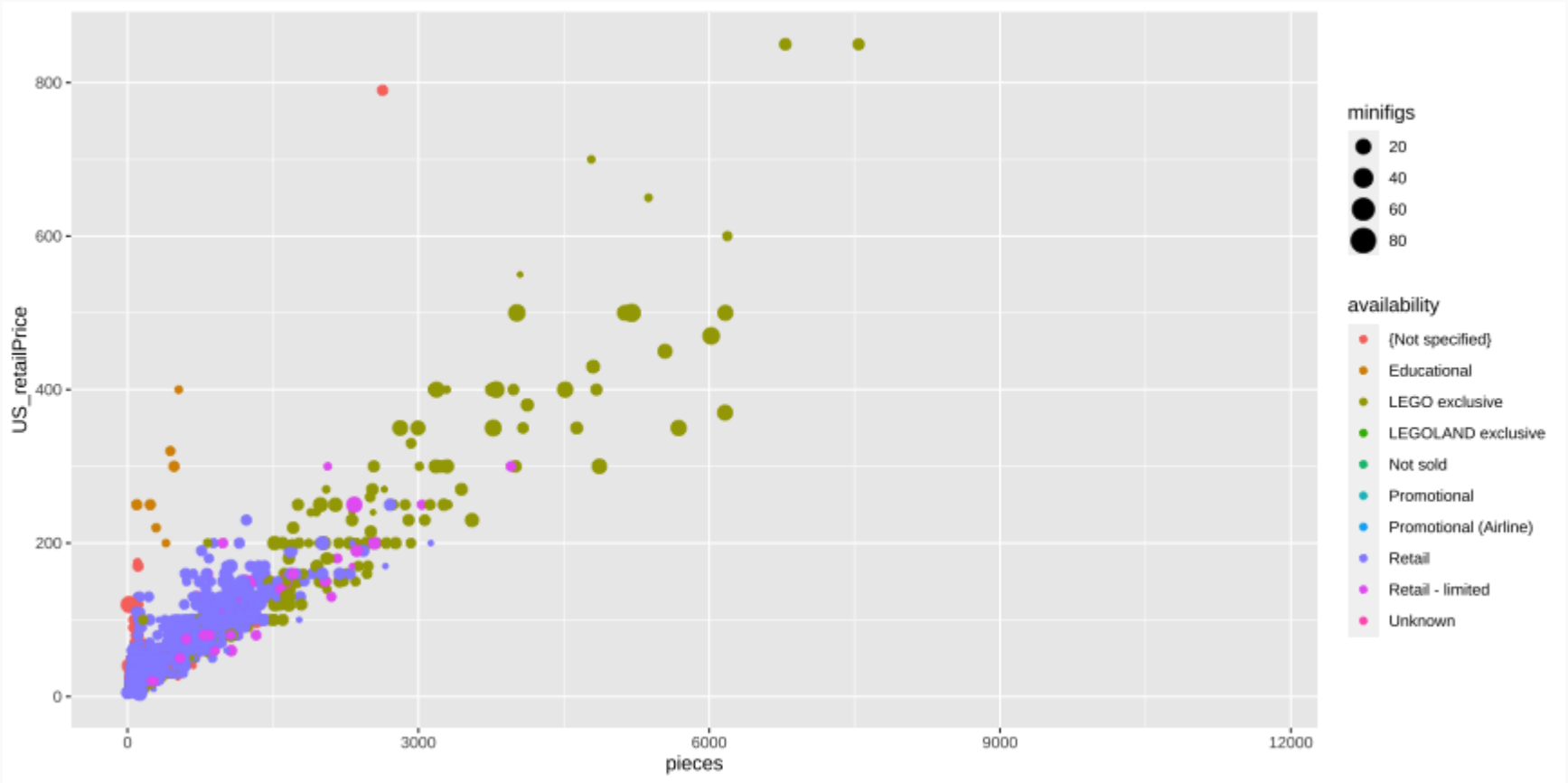
Scatterplot (cont.)

```
ggplot(legosets, aes(x=pieces, y=US_retailPrice, color=availability)) + geom_point()
```



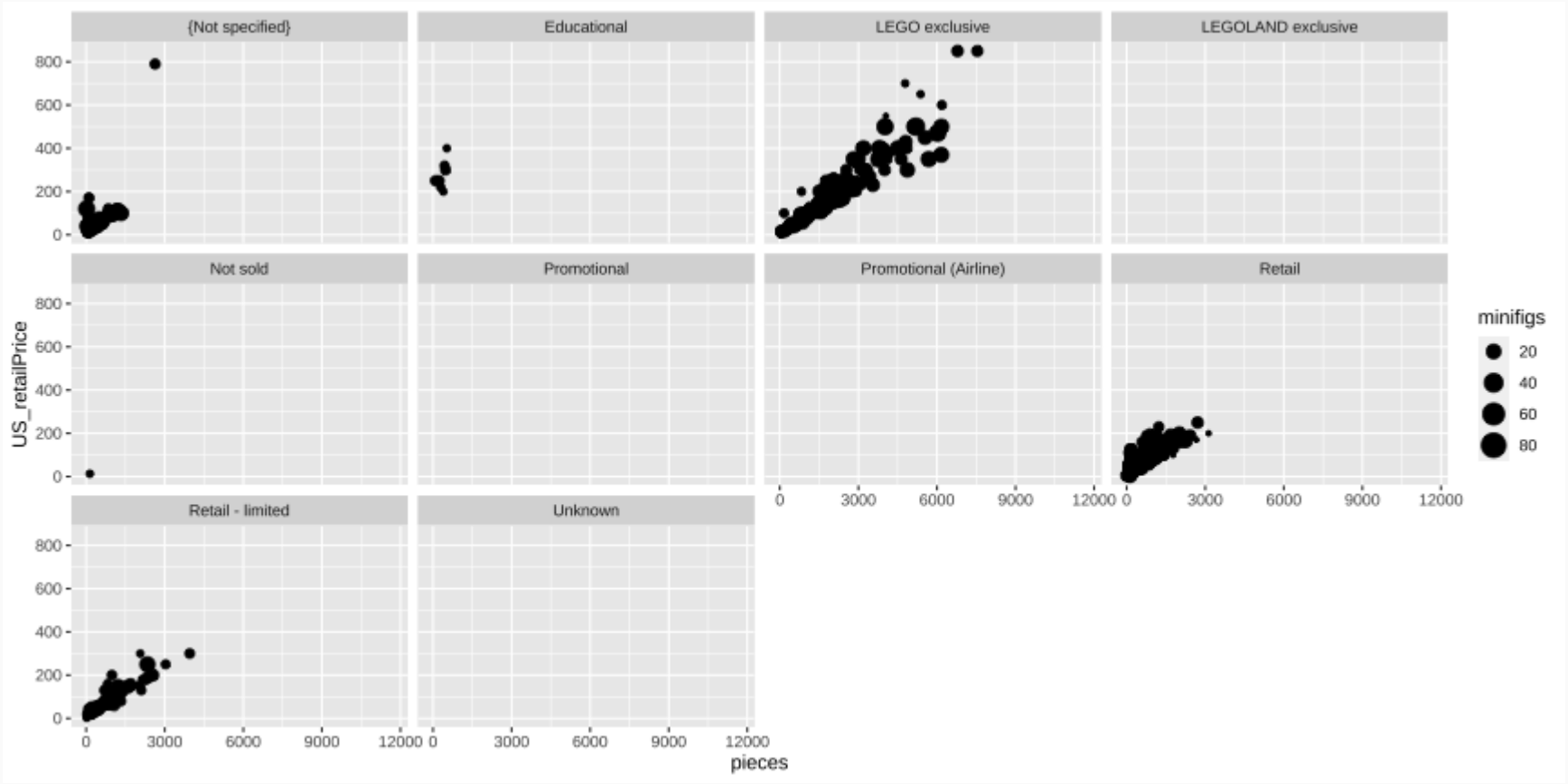
Scatterplot (cont.)

```
ggplot(legosets, aes(x=pieces, y=US_retailPrice, size=minifigs, color=availability)) + geom_point()
```



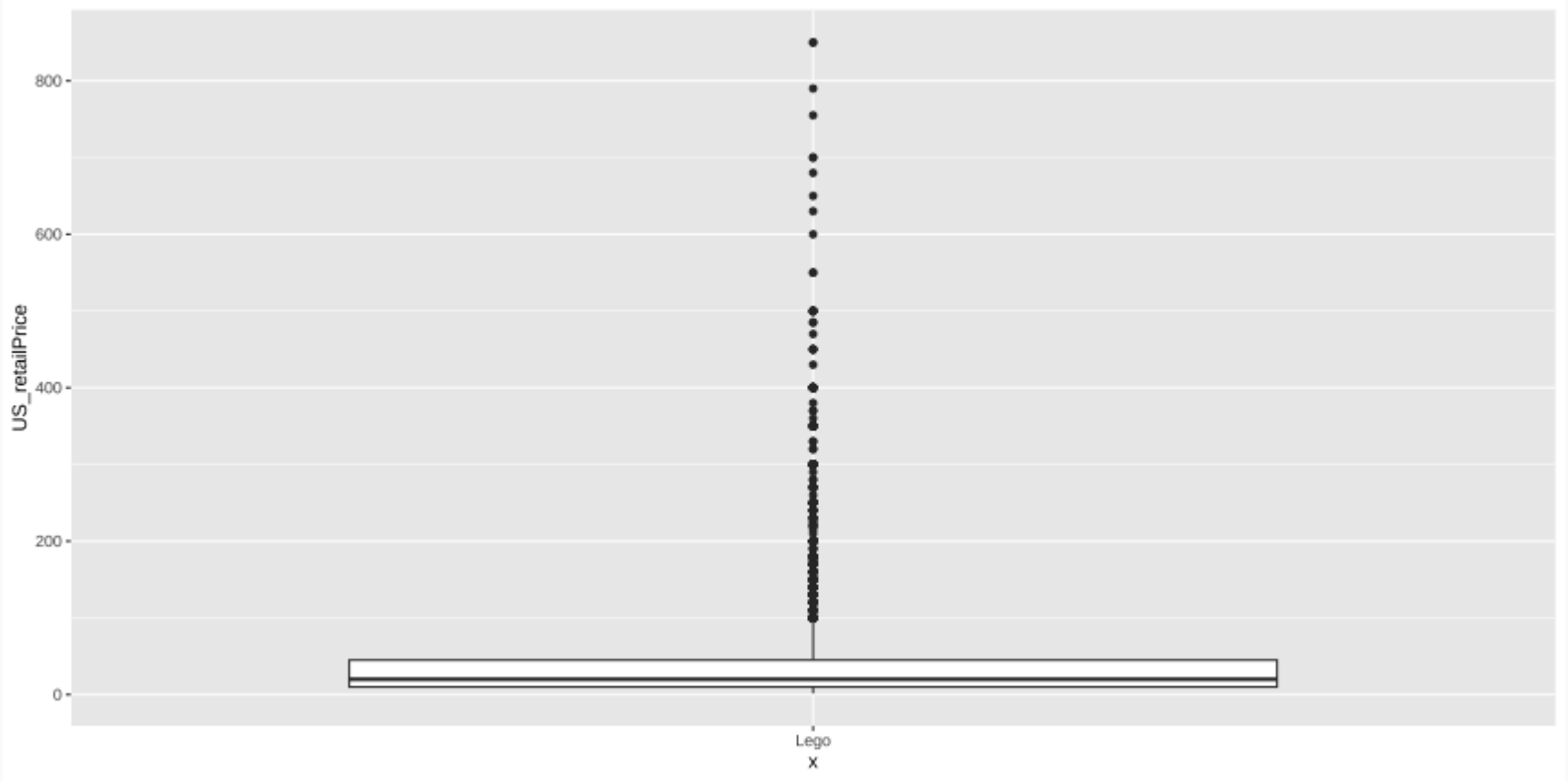
Scatterplot (cont.)

```
ggplot(legosets, aes(x=pieces, y=US_retailPrice, size=minifigs)) + geom_point() + facet_wrap(~ availability)
```



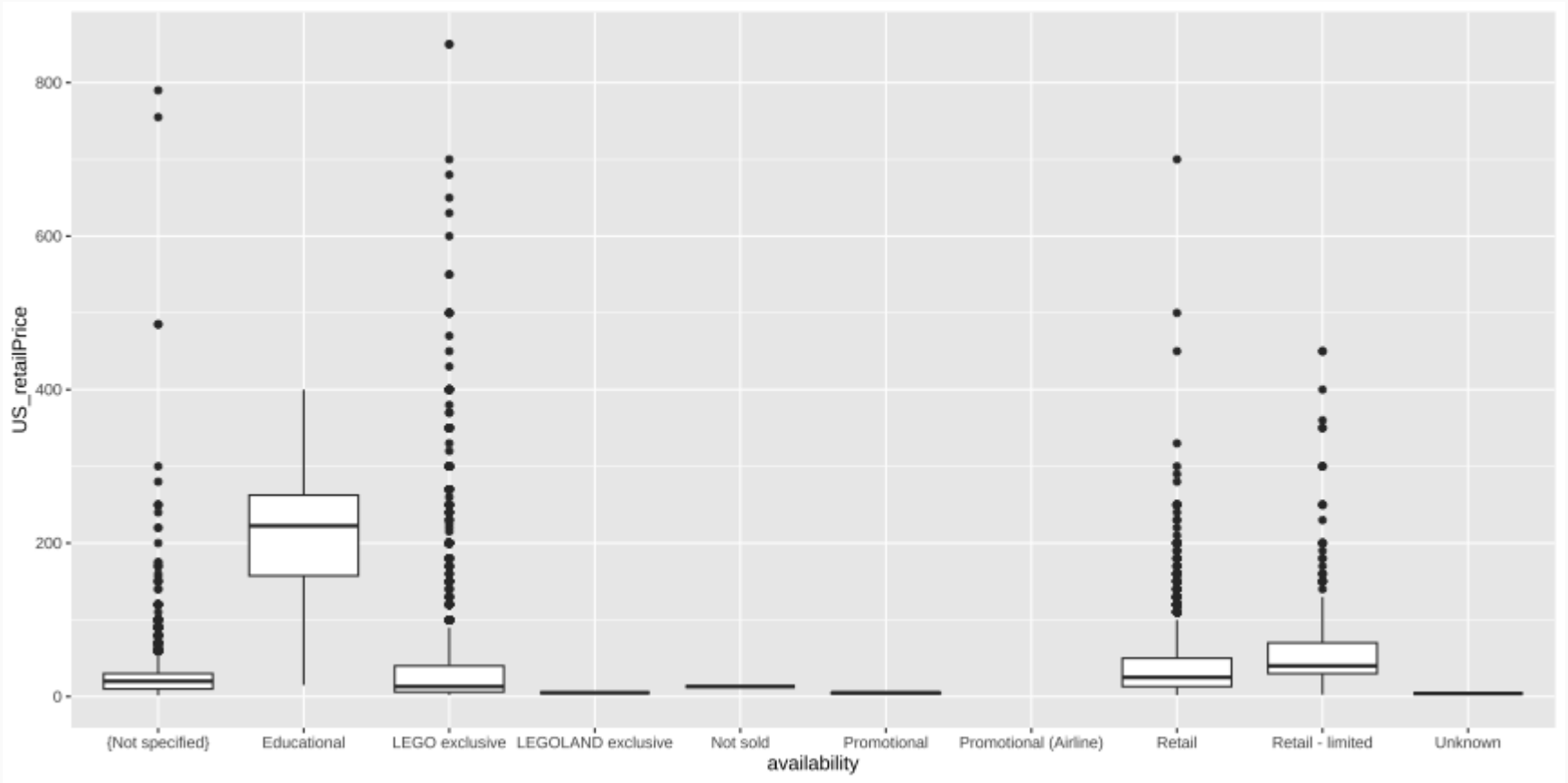
Boxplots

```
ggplot(legosets, aes(x='Lego', y=US_retailPrice)) + geom_boxplot()
```



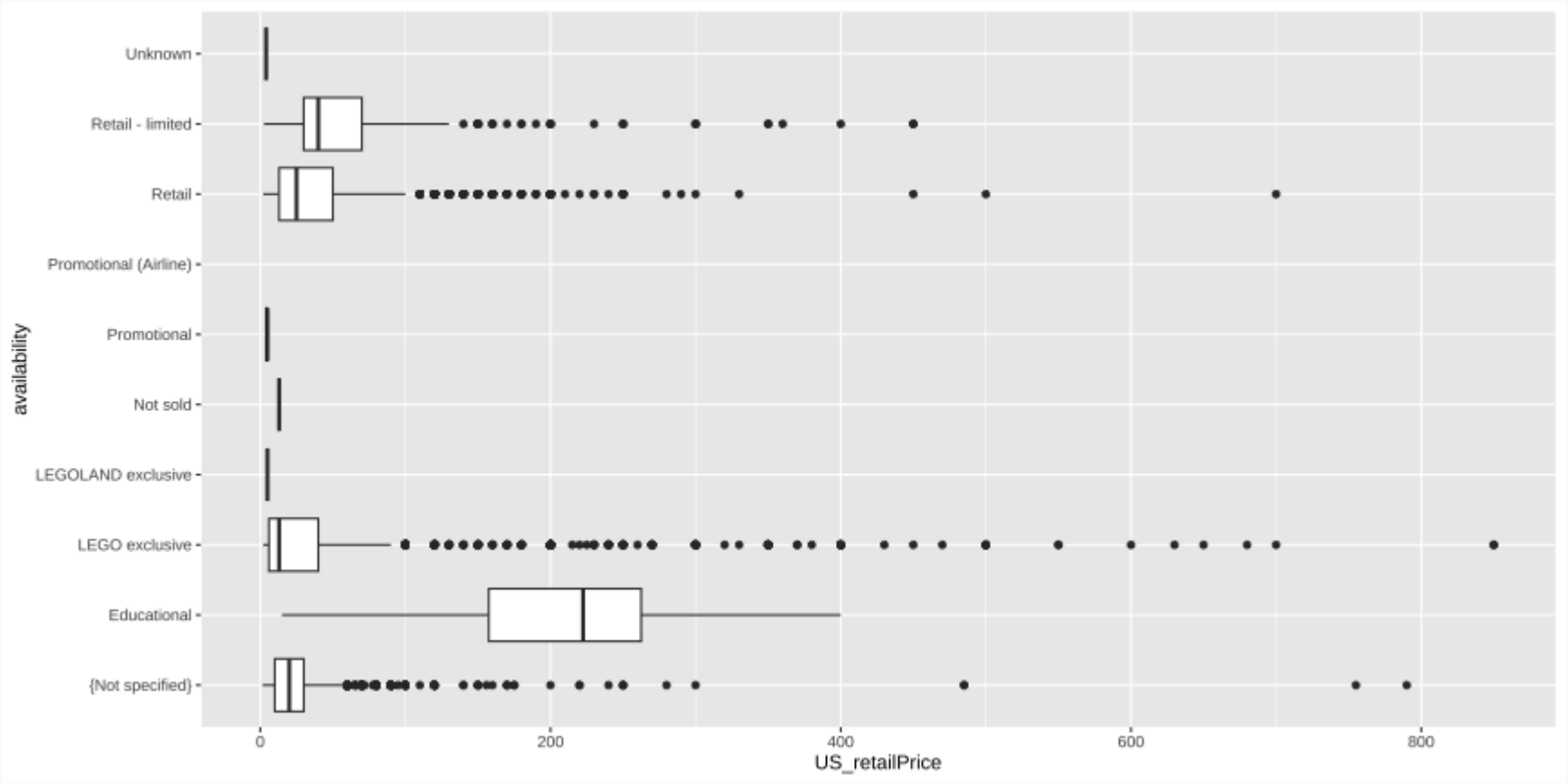
Boxplots (cont.)

```
ggplot(legosets, aes(x=availability, y=US_retailPrice)) + geom_boxplot()
```



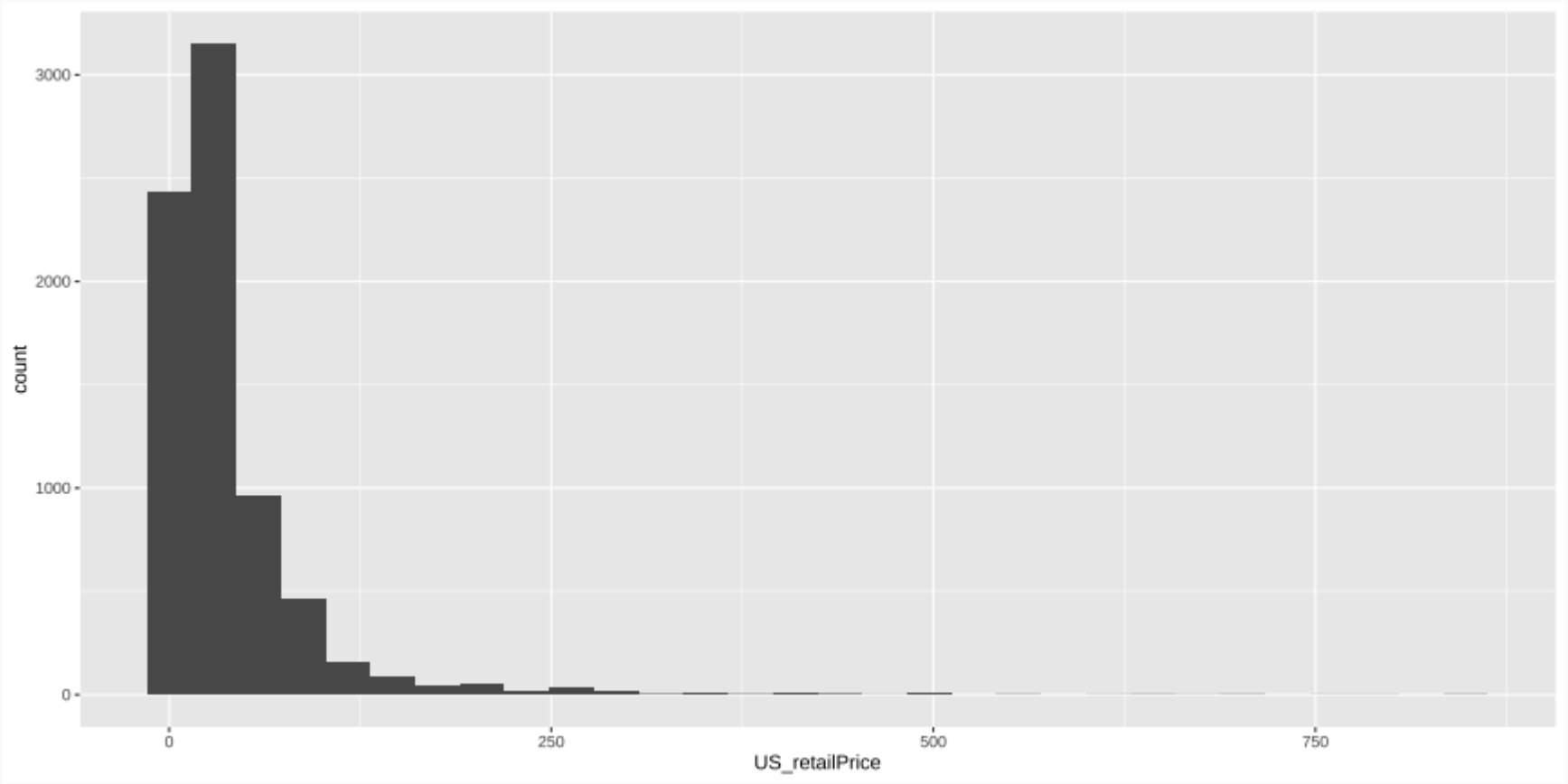
Boxplot (cont.)

```
ggplot(legosets, aes(x=availability, y=US_retailPrice)) + geom_boxplot() + coord_flip()
```



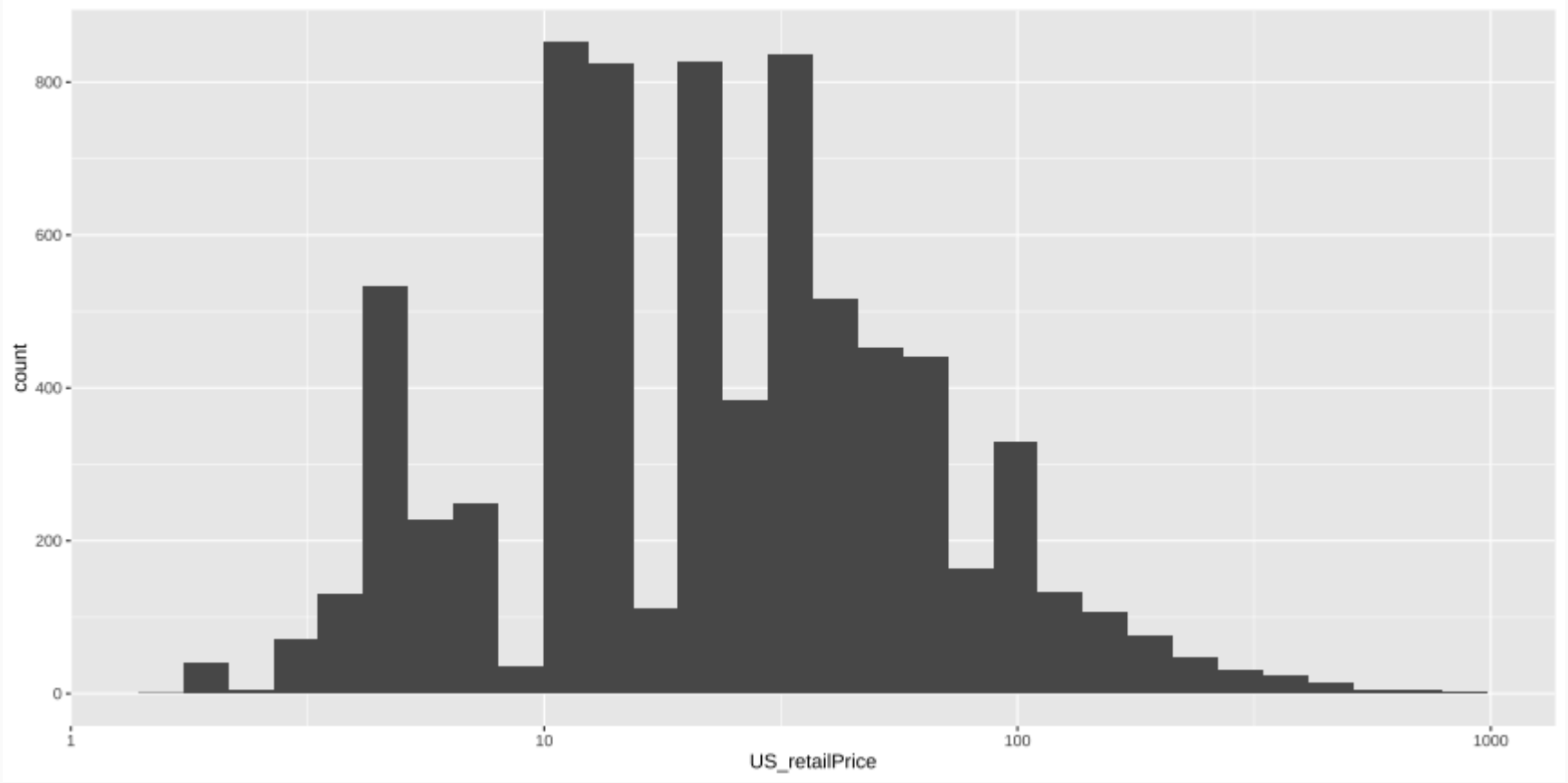
Histograms

```
ggplot(legosets, aes(x = US_retailPrice)) + geom_histogram()
```



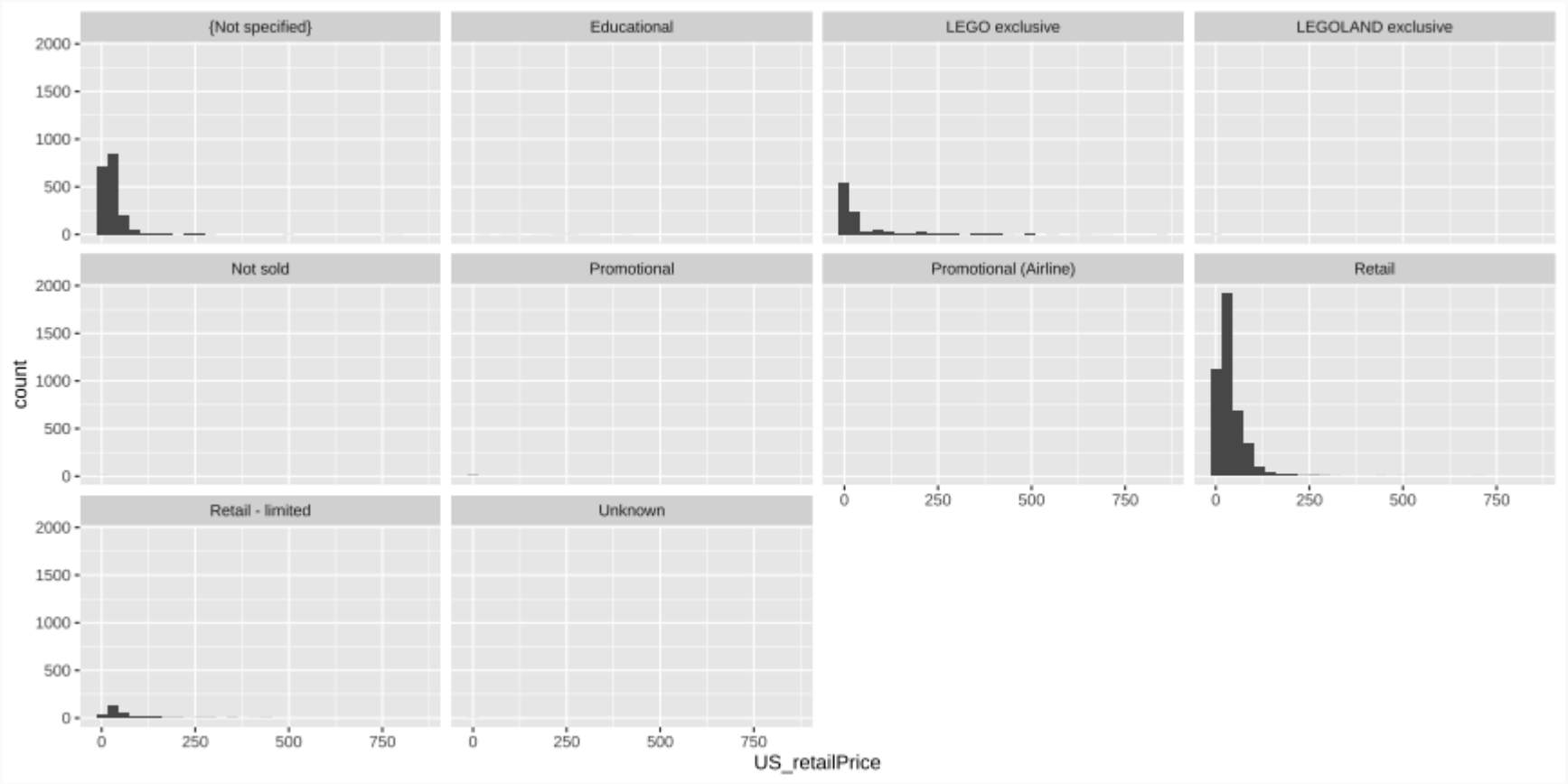
Histograms (cont.)

```
ggplot(legosets, aes(x = US_retailPrice)) + geom_histogram() + scale_x_log10()
```



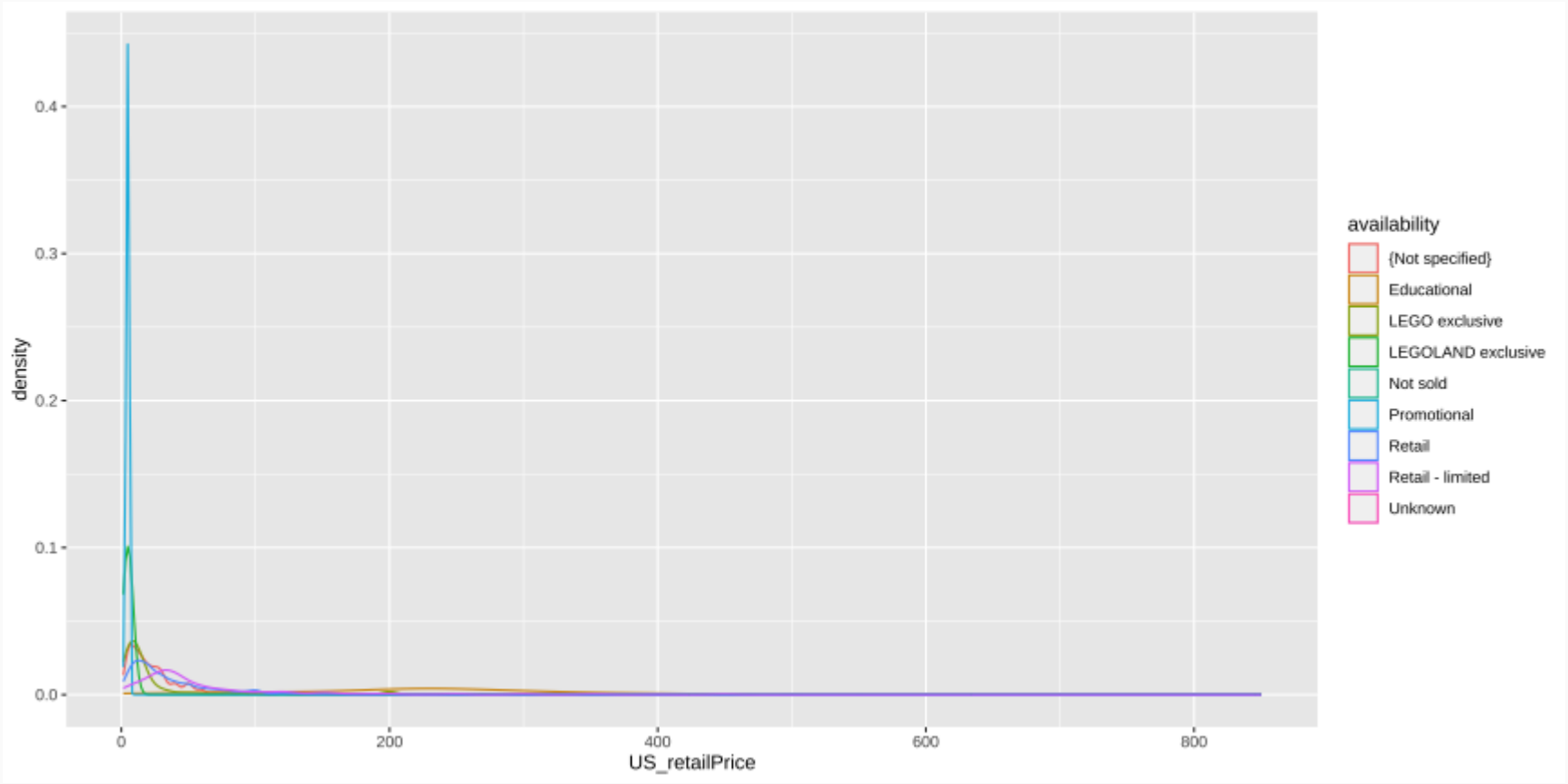
Histograms (cont.)

```
ggplot(legosets, aes(x = US_retailPrice)) + geom_histogram() + facet_wrap(~ availability)
```



Density Plots

```
ggplot(legosets, aes(x = US_retailPrice, color = availability)) + geom_density()
```



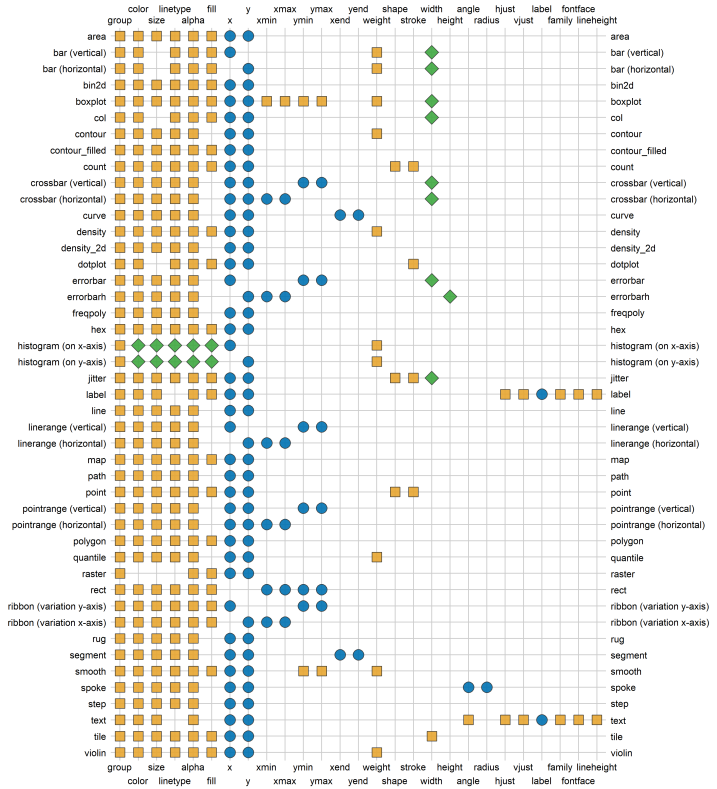
ggplot2 aesthetics

ggplot2 aesthetics cheat sheet

Use this table to find the right aesthetics for your geoms:

Aesthetics that usually must be mapped to the data: use inside aes()
Aesthetics that can be mapped to the data: use in or outside aes()
Aesthetics that cannot be mapped to the data: use outside aes()

e.g., ggplot(mpg, aes(x = class, y = displ)) + geom_col(aes(fill = class), width = .9)

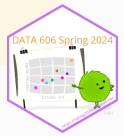


idea and design: Christian Burkhart
design advice: Ida Aarnio

Likert Scales

Likert scales are a type of questionnaire where respondents are asked to rate items on scales usually ranging from four to seven levels (e.g. strongly disagree to strongly agree).

```
library(likert)
library(reshape)
data(pisaitems)
items24 <- pisaitems[,substr(names(pisaitems), 1,5) == 'ST24Q']
items24 <- rename(items24, c(
            ST24Q01="I read only if I have to.",
            ST24Q02="Reading is one of my favorite hobbies.",
            ST24Q03="I like talking about books with other people.",
            ST24Q04="I find it hard to finish books.",
            ST24Q05="I feel happy if I receive a book as a present.",
            ST24Q06="For me, reading is a waste of time.",
            ST24Q07="I enjoy going to a bookstore or a library.",
            ST24Q08="I read only to get information that I need.",
            ST24Q09="I cannot sit still and read for more than a few minutes.",
            ST24Q10="I like to express my opinions about books I have read.",
            ST24Q11="I like to exchange books with my friends."))
```



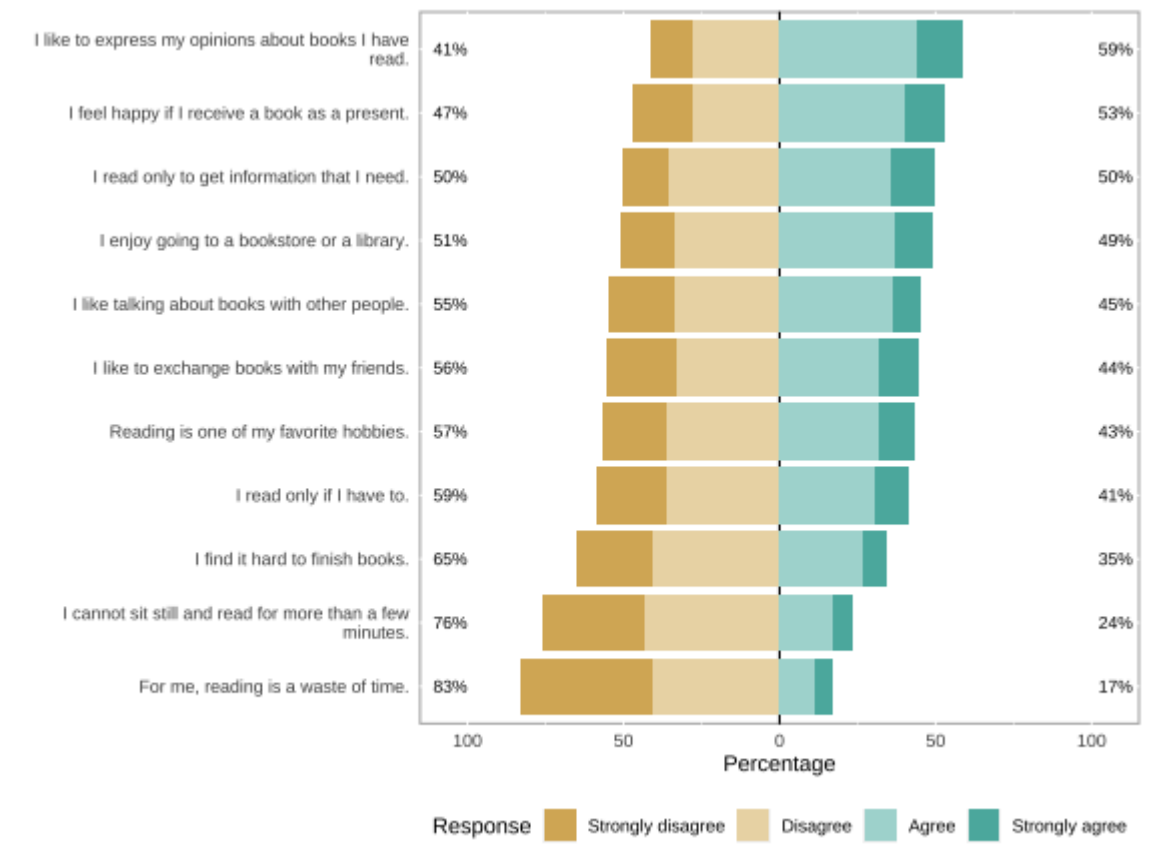
likert R Package

```
l24 <- likert(items24)
summary(l24)
```

```
##                                                        Item      low neutral
## 10   I like to express my opinions about books I have read. 41.07516       0
## 5            I feel happy if I receive a book as a present. 46.93475       0
## 8               I read only to get information that I need. 50.39874       0
## 7                I enjoy going to a bookstore or a library. 51.21231       0
## 3             I like talking about books with other people. 54.99129       0
## 11               I like to exchange books with my friends. 55.54115       0
## 2                   Reading is one of my favorite hobbies. 56.64470       0
## 1                                I read only if I have to. 58.72868       0
## 4                          I find it hard to finish books. 65.35125       0
## 9  I cannot sit still and read for more than a few minutes. 76.24524       0
## 6                       For me, reading is a waste of time. 82.88729       0
##        high     mean        sd
## 10 58.92484 2.604913 0.9009968
## 5  53.06525 2.466751 0.9446590
## 8  49.60126 2.484616 0.9089688
## 7  48.78769 2.428508 0.9164136
## 3  45.00871 2.328049 0.9090326
## 11 44.45885 2.343193 0.9609234
## 2  43.35530 2.344530 0.9277495
```

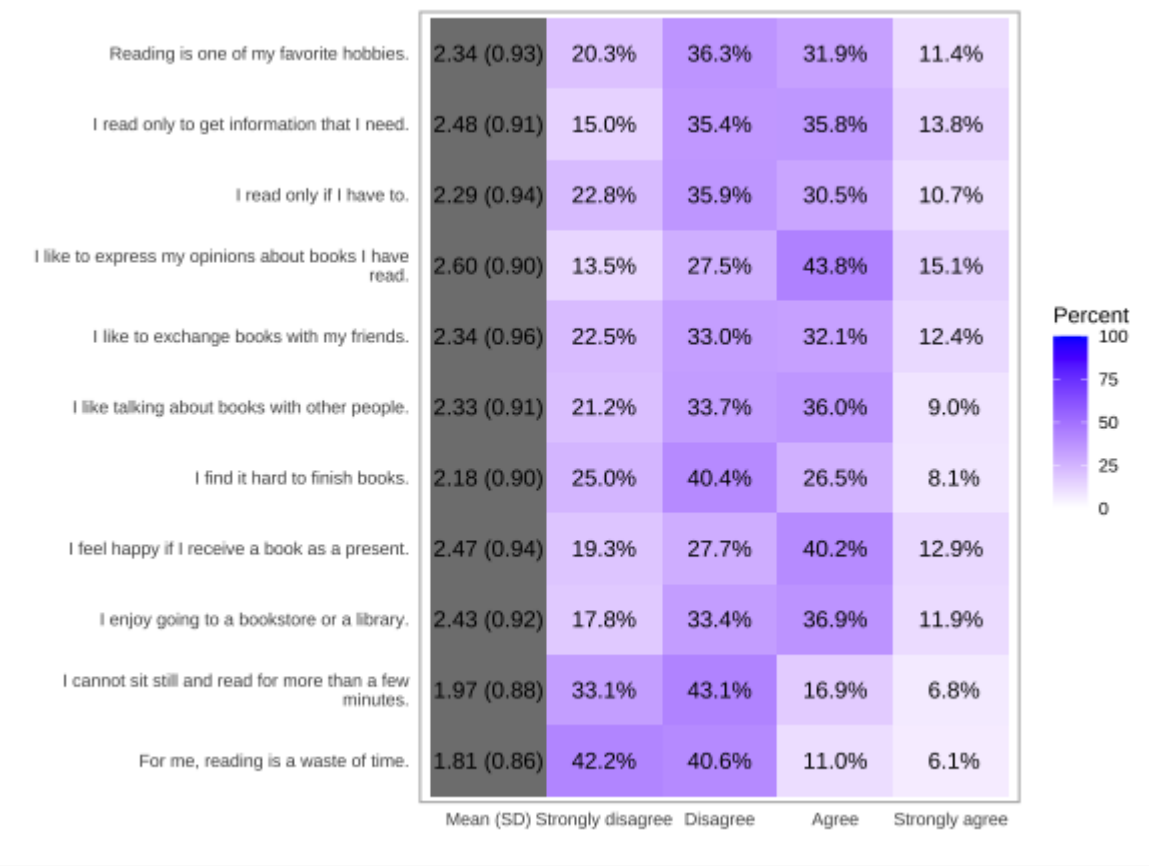
likert Plots

```
plot(l24)
```



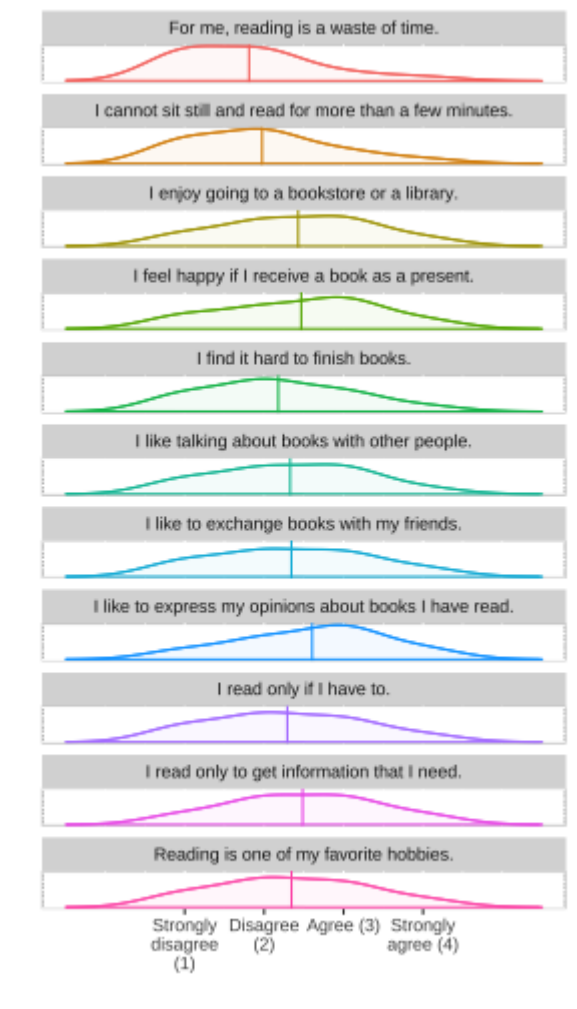
likert Plots

```
plot(l24, type='heat')
```



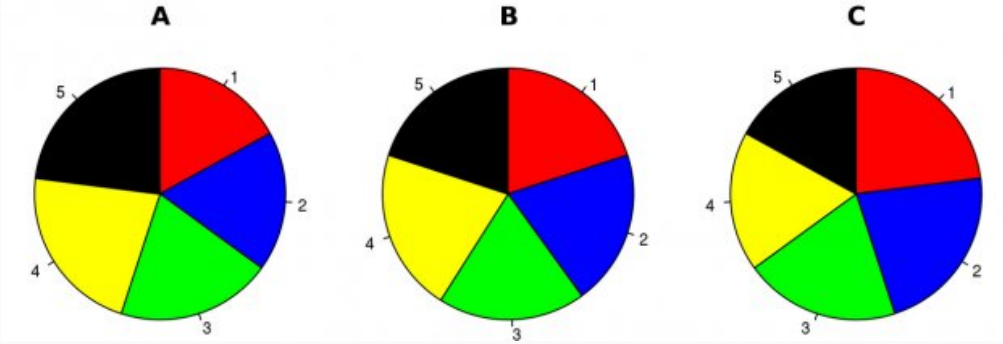
likert Plots

```
plot(l24, type='density')
```



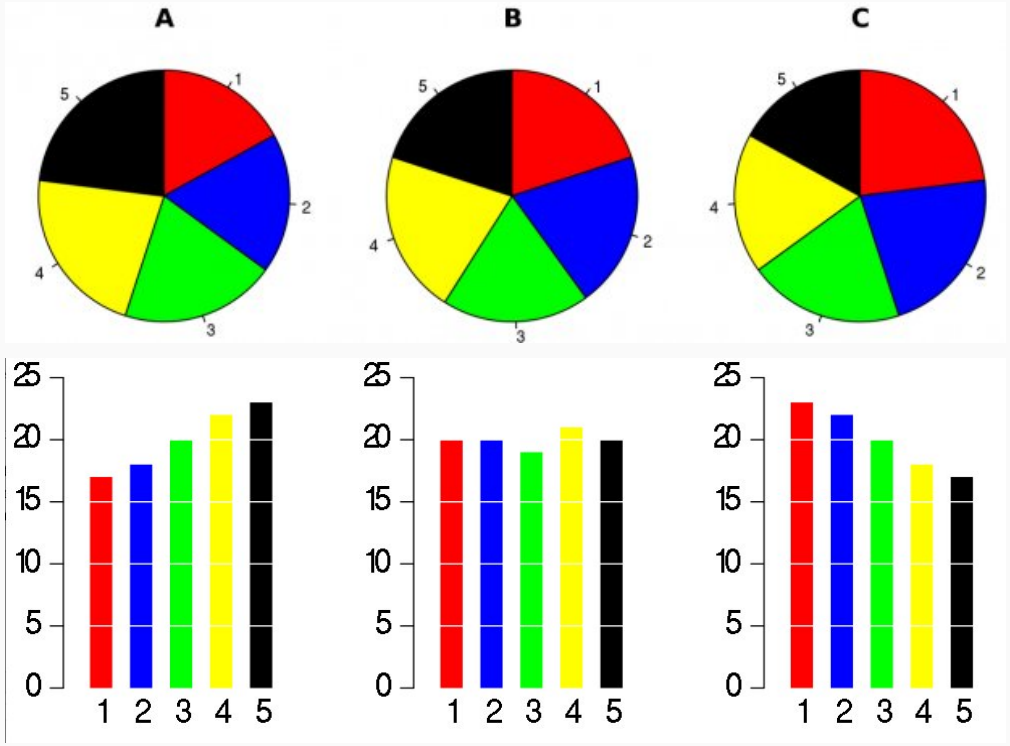
Pie Charts

There is only one pie chart in *OpenIntro Statistics* (Diez, Barr, & Çetinkaya-Rundel, 2015, p. 48). Consider the following three pie charts that represent the preference of five different colors. Is there a difference between the three pie charts? This is probably a difficult to answer.



Pie Charts

There is only one pie chart in *OpenIntro Statistics* (Diez, Barr, & Çetinkaya-Rundel, 2015, p. 48). Consider the following three pie charts that represent the preference of five different colors. Is there a difference between the three pie charts? This is probably a difficult to answer.



Source: https://en.wikipedia.org/wiki/Pie_chart.

"There is no data that can be displayed in a pie chart that cannot better be displayed in some other type of chart"

John Tukey

Additional Resources

For data wrangling:

- `dplyr` website: https://dplyr.tidyverse.org
- R for Data Science book: https://r4ds.had.co.nz/wrangle-intro.html
- Wrangling penguins tutorial: https://allisonhorst.shinyapps.io/dplyr-learnr/#section-welcome
- Data transformation cheat sheet: https://github.com/rstudio/cheatsheets/raw/master/data-transformation.pdf

For data visualization:

- `ggplot2` website: https://ggplot2.tidyverse.org
- R for Data Science book: https://r4ds.had.co.nz/data-visualisation.html
- R Graphics Cookbook: https://r-graphics.org
- Data visualization cheat sheet: https://github.com/rstudio/cheatsheets/raw/master/data-visualization-2.1.pdf

One Minute Paper

Complete the one minute paper:
https://forms.gle/CD5Qxkq3xtdxSheW8

1. What was the most important thing you learned during this class?

2. What important question remains unanswered for you?