

Summarizing Data Part 2

DATA 606 - Statistics & Probability for Data Analytics

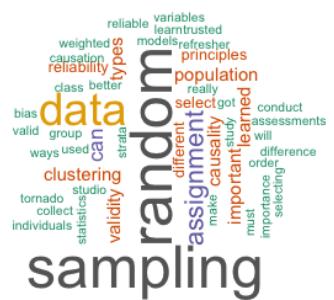
Jason Bryer, Ph.D. and Angela Lui, Ph.D.

September 14, 2022

One Minute Paper Results

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

What important question remains unanswered for you?



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"            "geom_blank"          "geom_contour"        "geom_crossbar"       "geom_density_2d"      "geom_hex"            "geom_jitter"         "geom_linerange"      "geom_point"          "geom_pointrange"     "geom_raster"         "geom_rect"           "geom_sf_label"       "geom_spoke"          "geom_tile"           "geom_vline"          "geom_violin"
```

```
## [4] "geom_bin_2d"          "geom_bin2d"          "geom_col"            "geom_count"          "geom_curve"          "geom_density"        "geom_errorbar"       "geom_function"       "geom_hline"          "geom_line"           "geom_map"            "geom_path"           "geom_polygon"        "geom_quantile"       "geom_rect"           "geom_sf"             "geom_smooth"         "geom_text"           "geom_vline"
```

```
## [7] "geom_boxplot"         "geom_col"            "geom_count"          "geom_crossbar"       "geom_density2d"     "geom_errorbarh"      "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_raster"         "geom_rect"           "geom_sf_label"       "geom_spoke"          "geom_tile"           "geom_vline"
```

```
## [10] "geom_contour_filled" "geom_density2d_filled" "geom_errorbarh"      "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_raster"         "geom_rect"           "geom_sf_label"       "geom_smooth"         "geom_text"           "geom_vline"
```

```
## [13] "geom_dotplot"         "geom_errorbar"       "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_rect"           "geom_sf_label"       "geom_smooth"         "geom_text"           "geom_vline"
```

```
## [16] "geom_density_2d_filled" "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_rect"           "geom_sf_label"       "geom_smooth"         "geom_text"           "geom_vline"
```

```
## [19] "geom_freqpoly"        "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_rect"           "geom_sf_label"       "geom_smooth"         "geom_text"           "geom_vline"
```

```
## [22] "geom_histogram"        "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_rect"           "geom_sf_label"       "geom_smooth"         "geom_text"           "geom_vline"
```

```
## [25] "geom_label"           "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_rect"           "geom_sf_label"       "geom_smooth"         "geom_text"           "geom_vline"
```

```
## [28] "geom_label"           "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_rect"           "geom_sf_label"       "geom_smooth"         "geom_text"           "geom_vline"
```

```
## [31] "geom_map"              "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_rect"           "geom_sf_label"       "geom_smooth"         "geom_text"           "geom_vline"
```

```
## [34] "geom_pointrange"       "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_rect"           "geom_sf_label"       "geom_smooth"         "geom_text"           "geom_vline"
```

```
## [37] "geom_qq_line"          "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_rect"           "geom_sf_label"       "geom_smooth"         "geom_text"           "geom_vline"
```

```
## [40] "geom_rect"              "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_rect"           "geom_sf_label"       "geom_smooth"         "geom_text"           "geom_vline"
```

```
## [43] "geom_segment"          "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_rect"           "geom_sf_label"       "geom_smooth"         "geom_text"           "geom_vline"
```

```
## [46] "geom_sf_text"          "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_rect"           "geom_sf_label"       "geom_smooth"         "geom_text"           "geom_vline"
```

```
## [49] "geom_step"              "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_rect"           "geom_sf_label"       "geom_smooth"         "geom_text"           "geom_vline"
```

```
## [52] "geom_violin"           "geom_hex"            "geom_jitter"         "geom_linerange"     "geom_point"          "geom_pointrange"     "geom_rect"           "geom_sf_label"       "geom_smooth"         "geom_text"           "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 **marks**—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>) +
  <GEO FUNCTION> (mapping = aes(<POSITION>),
  stat = <STAT>, position = <POSITION>) +
  <COORDINATE FUNCTION> +
  <FACET FUNCTION> +
  <SCALE FUNCTION> +
  <THEME FUNCTION>
```

required
Not required, supplied by defaults or supplied

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

aesthetic mappings **data** **geom**

plot(x = cyl, 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, y))

a + geom_blank()
a + geom_curve(aes(yend = lat + 1,
xend = long + 1, curvature = z), x, yend, y, end,
alpha, angle, color, curvature, linetype, hjust,
lineheight, size, vjust)
a + geom_path(linewidth = "butt", linejoin = "round",
x, y, alpha, color, group, linetype, size)
a + geom_polygon(aes(group = group))
a + geom_rect(aes(xmin = long, ymin = lat, xmax =
long + 1, ymax = lat + 1), x, 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_segment(aes(yend = lat + 1, xend = long + 1))
b + geom_spoke(aes(angle = 1:115, radius = 1))
```

ONE VARIABLE continuous

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

c + geom_area(stat = "bin")
c + geom_density(kernel = "gaussian")
c + geom_dotplot()
c + geom_freqpoly()
c + geom_histogram(binwidth = 5)
c2 + geom_qq(aes(sample = hwy))
```

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 = cyl), nudge_x = 1,
nudge_y = 1, check_overlap = TRUE)
e + geom_text(aes(label = cyl), nudge_x = 1,
nudge_y = 1, check_overlap = TRUE)
```

discrete x , continuous y

```
f <- ggplot(mpg, aes(class, hwy))
f + geom_col()
f + geom_boxplot()
f + geom_dotplot(binaxis = "y", stackdir =
"center")
f + geom_violin(scale = "area")
```

discrete x , discrete y

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

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))
l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5,
interpolate = FALSE)
l + geom_tile(aes(fill = z))
```

continuous bivariate distribution

```
h <- ggplot(diamonds, aes(carat, price))
h + geom_hex()
h + geom_bin2d(binwidth = c(0.25, 500))
h + geom_density2d()
h + geom_rug()
```

continuous function

```
i <- ggplot(economics, aes(date, unemploy))
i + geom_area()
i + geom_line()
i + geom_step(direction = "hv")
```

visualizing error

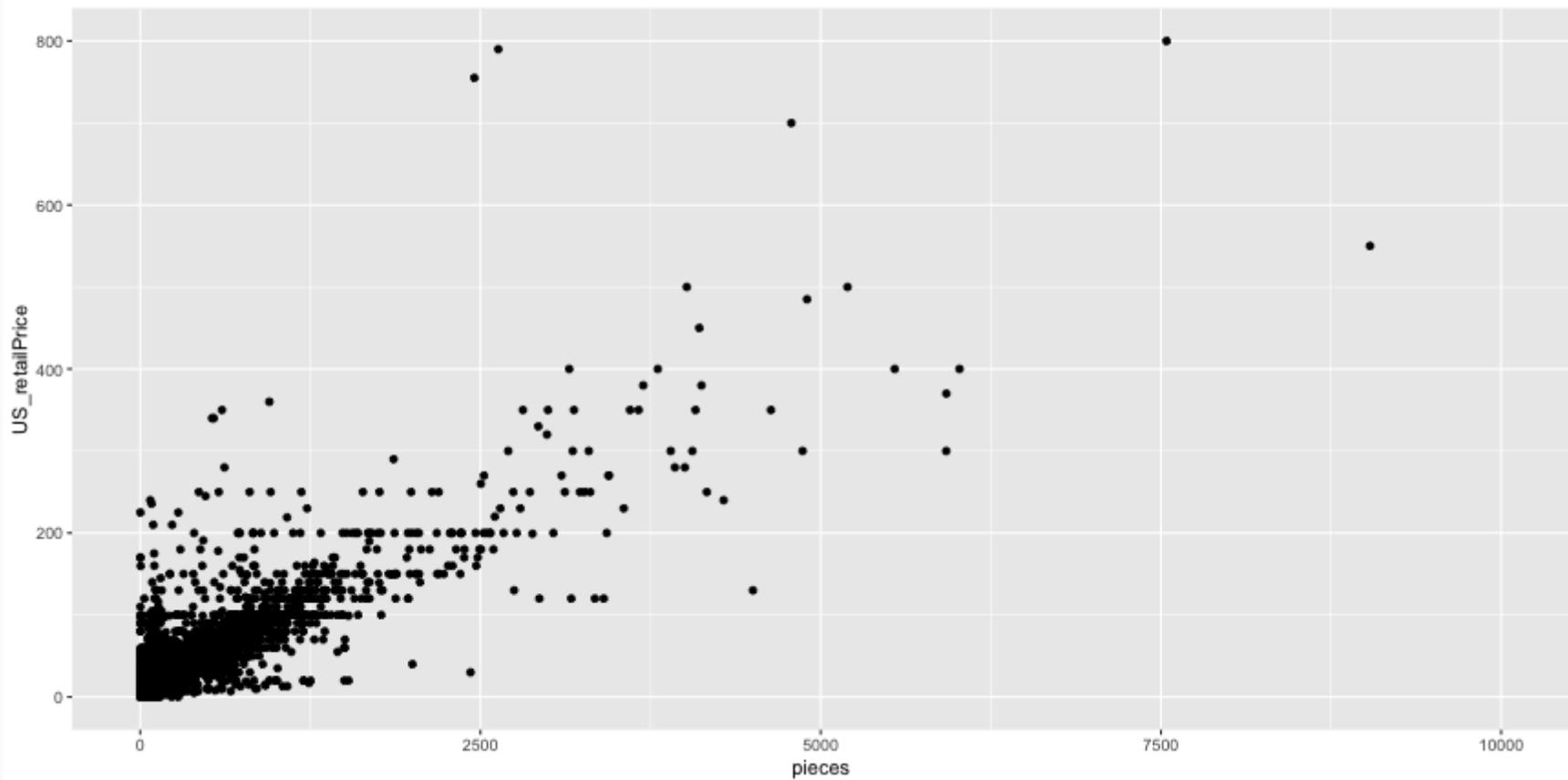
```
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))
j + geom_crossbar(fatten = 2)
j + geom_errorbar()
```

map <- data.frame(murder = USArrests\$Murder,
state = tolowerrownames(USArrests))
map <- map_data("state")

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

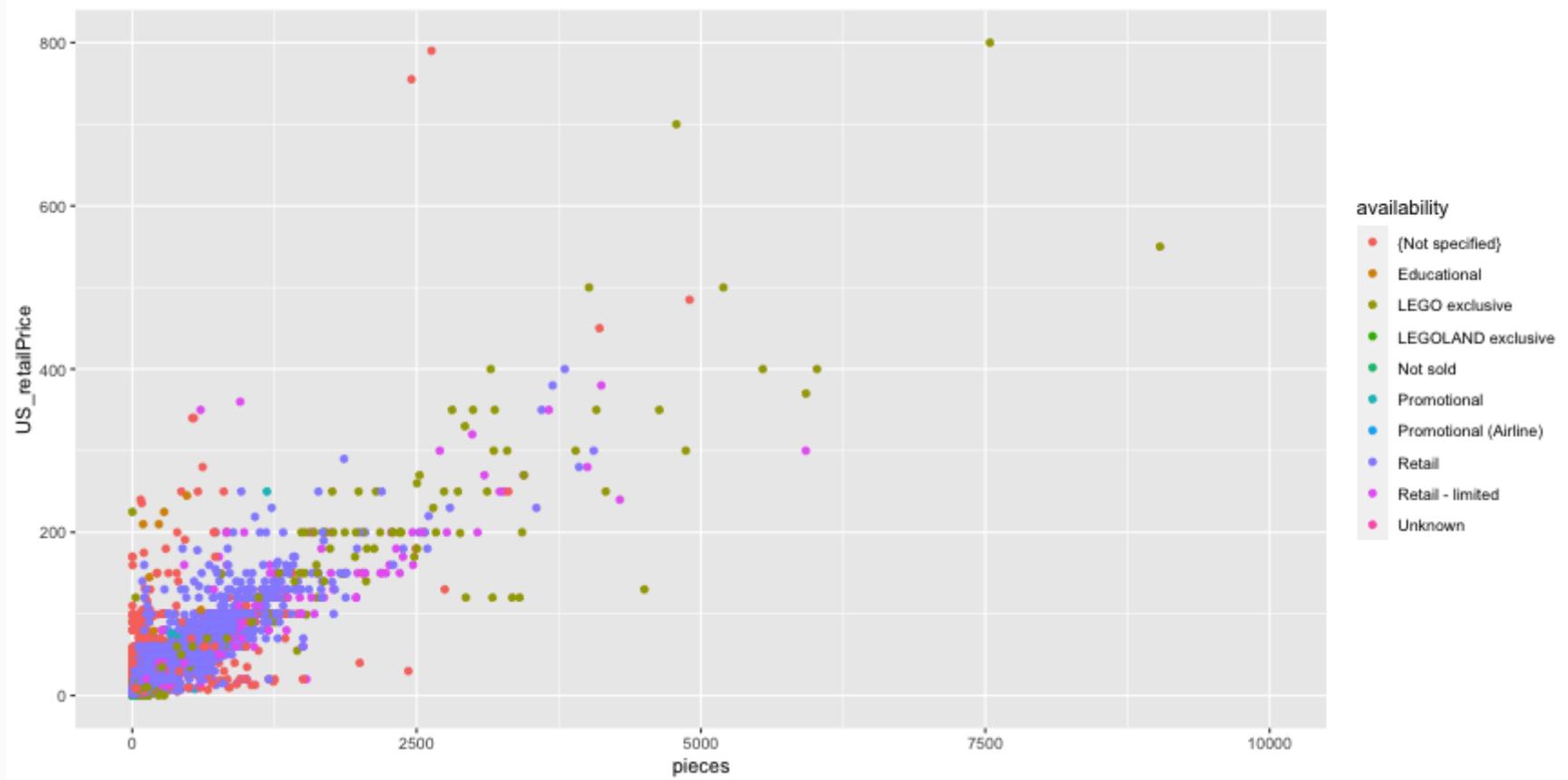
Scatterplot

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



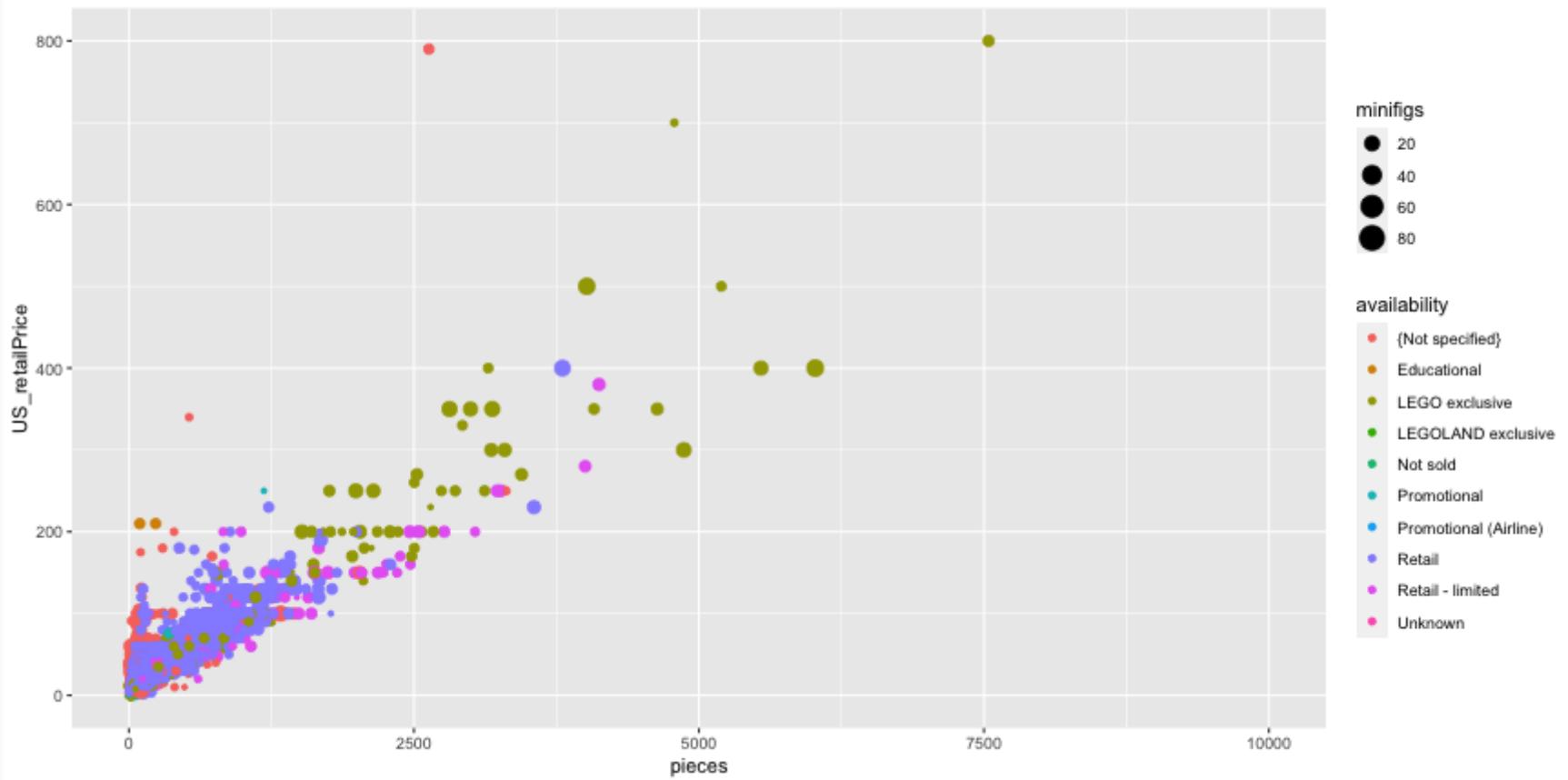
Scatterplot (cont.)

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



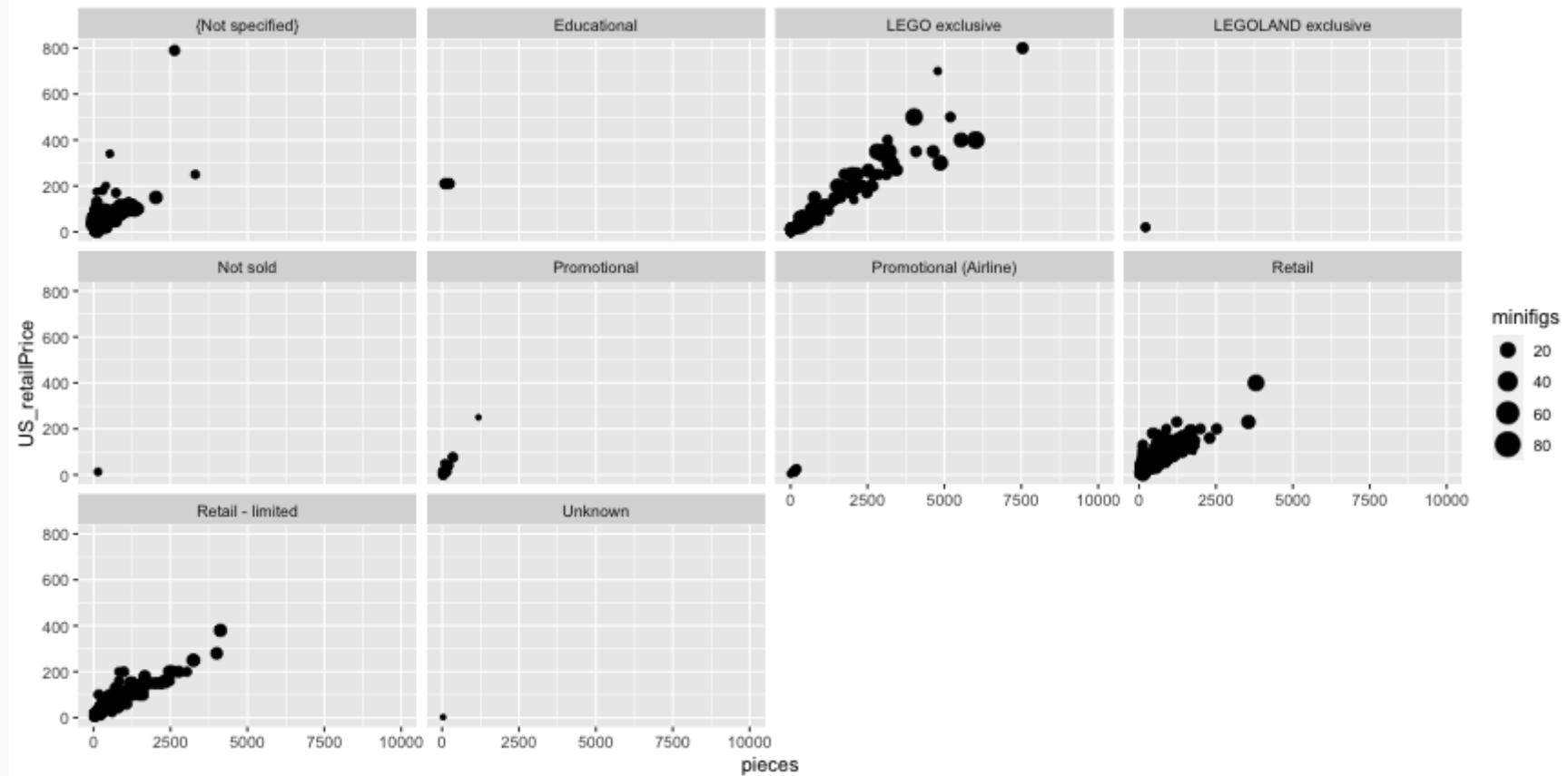
Scatterplot (cont.)

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



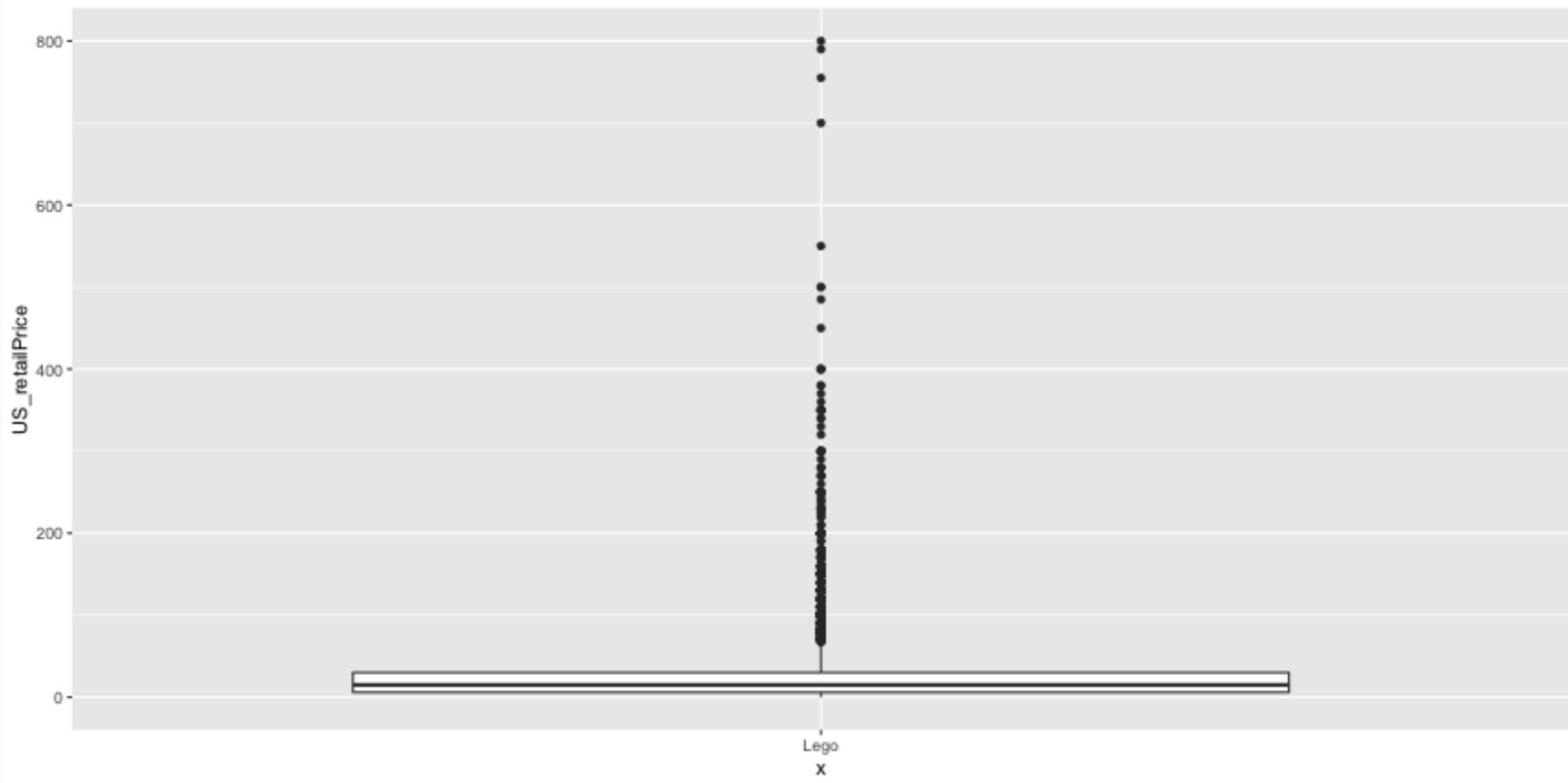
Scatterplot (cont.)

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



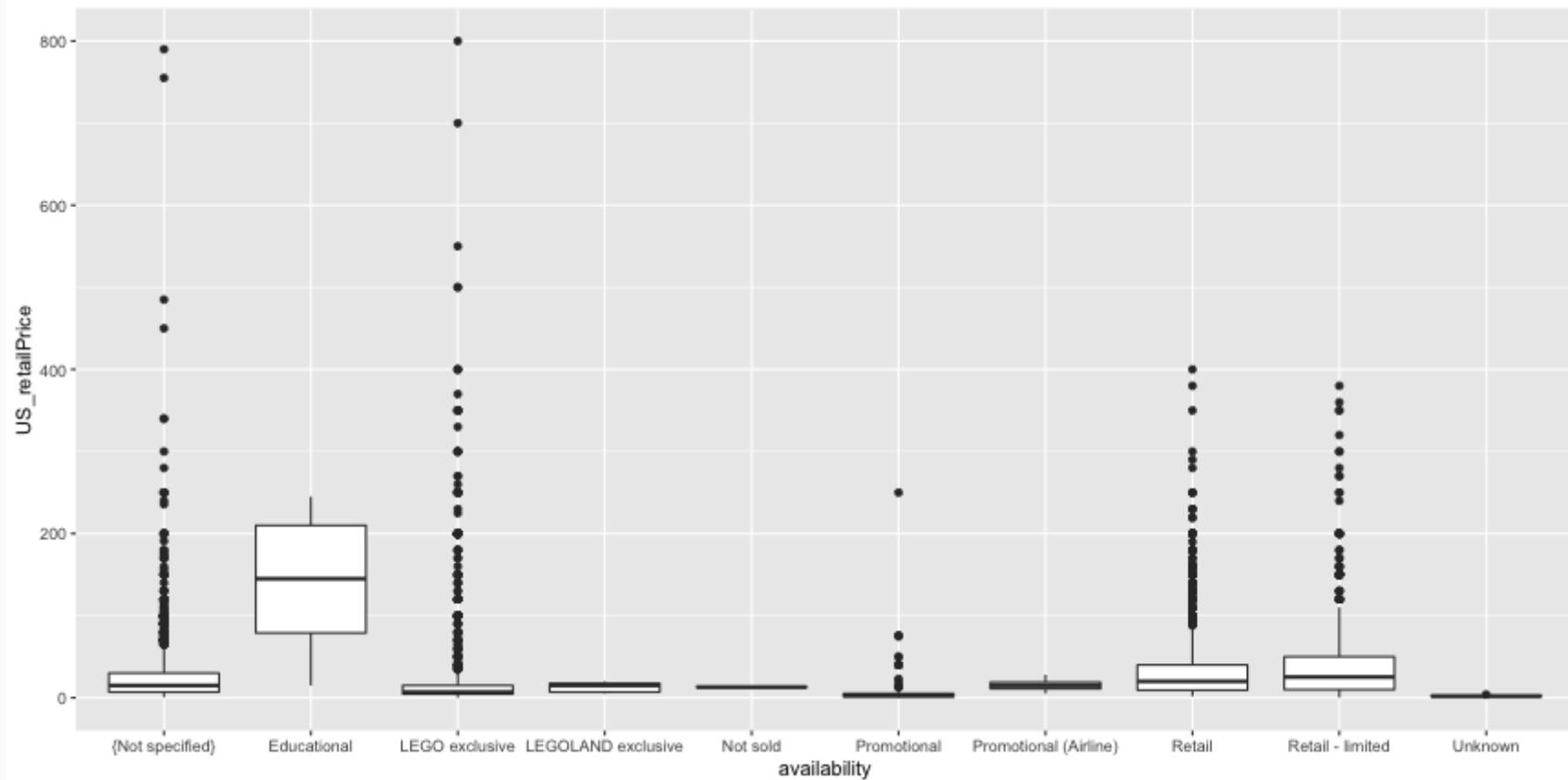
Boxplots

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



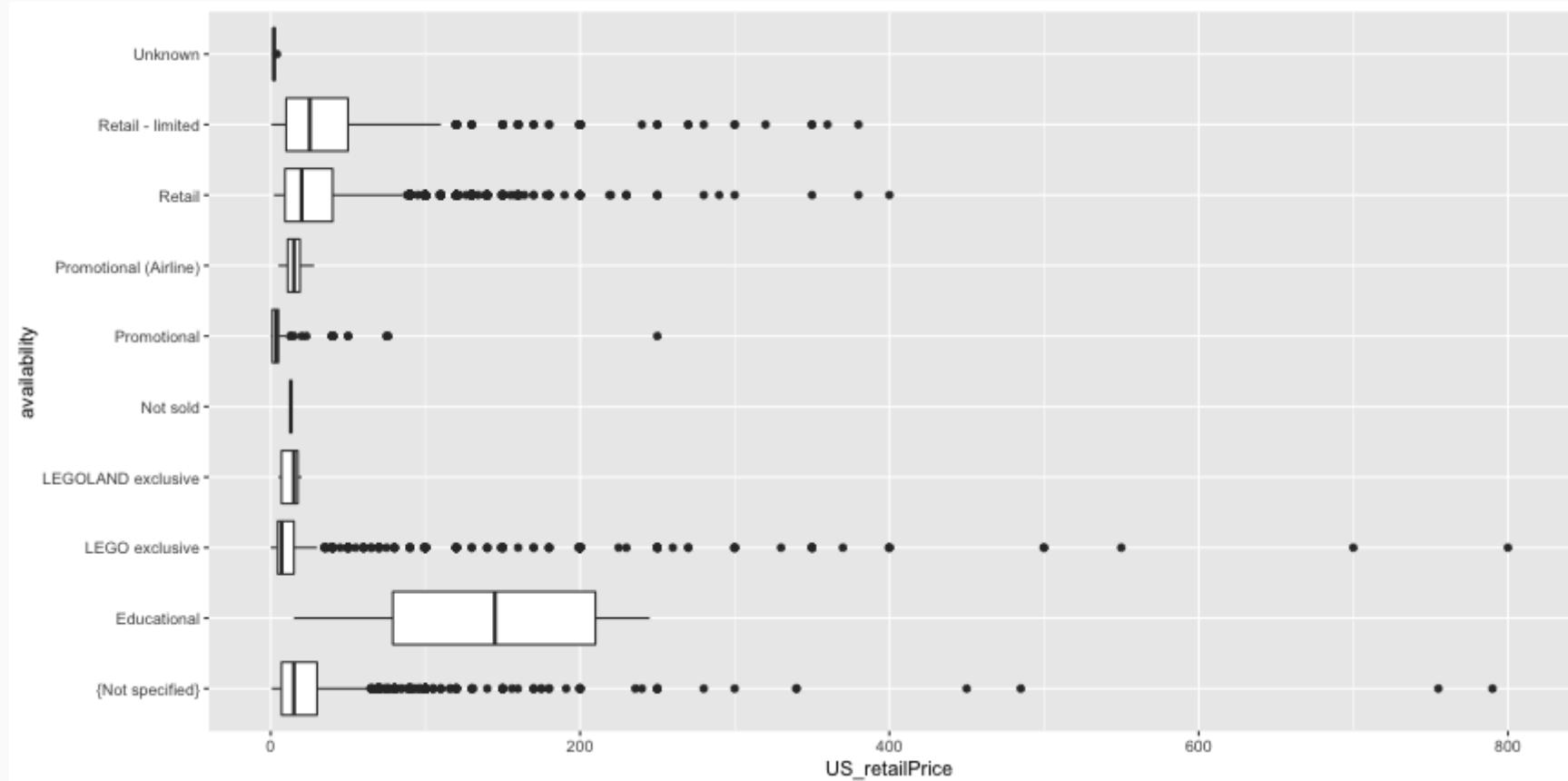
Boxplots (cont.)

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



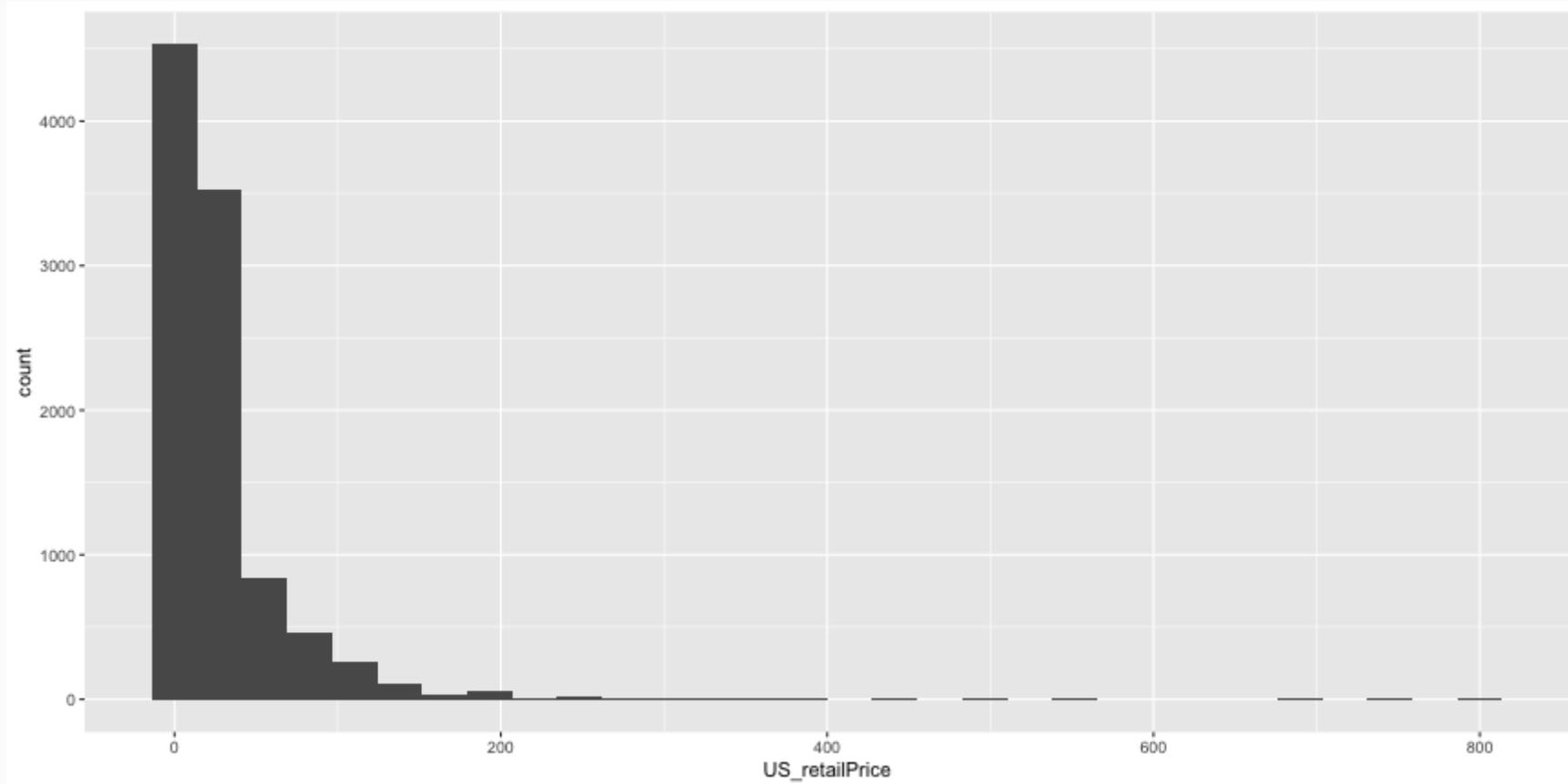
Boxplot (cont.)

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



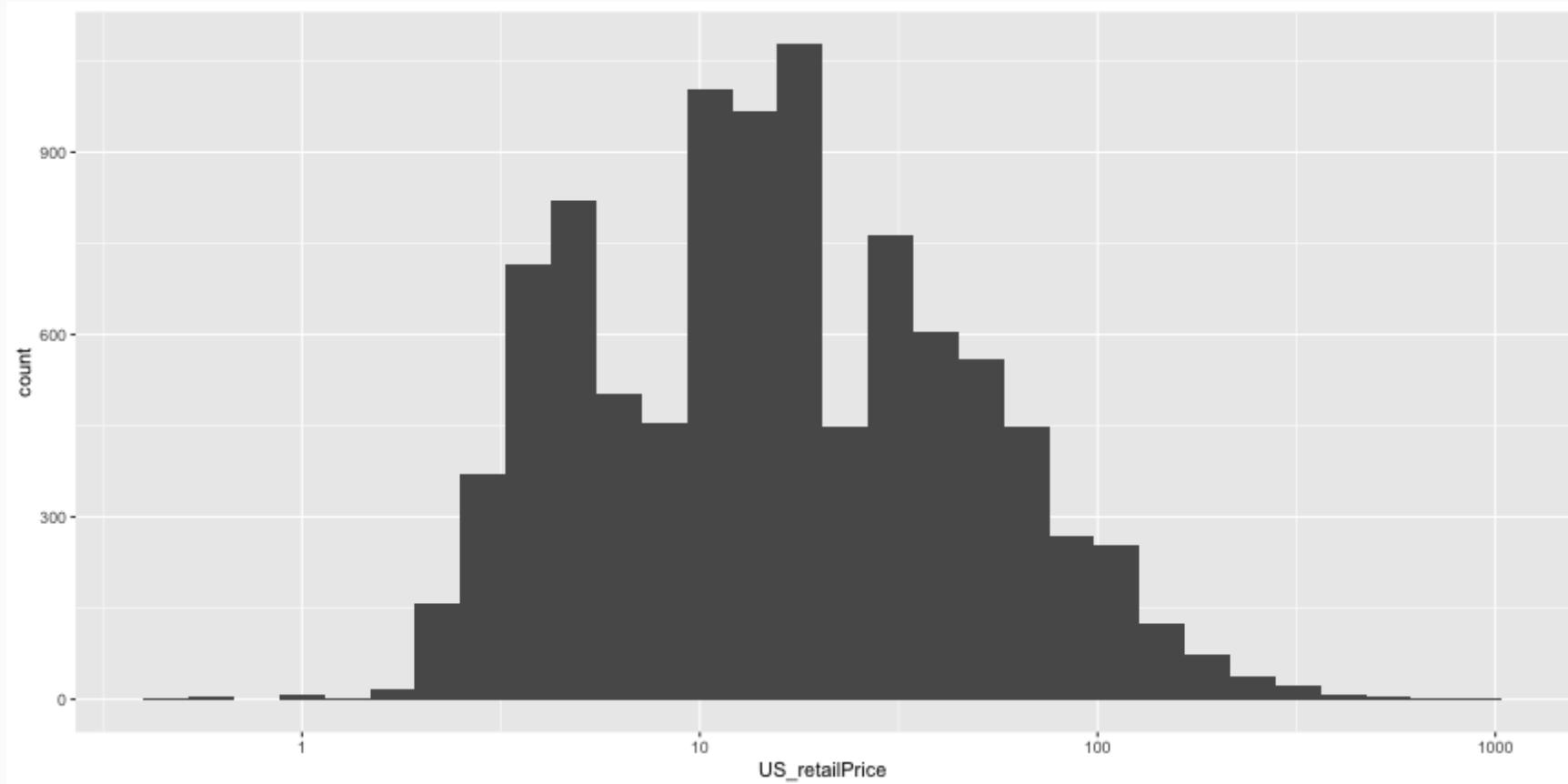
Histograms

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



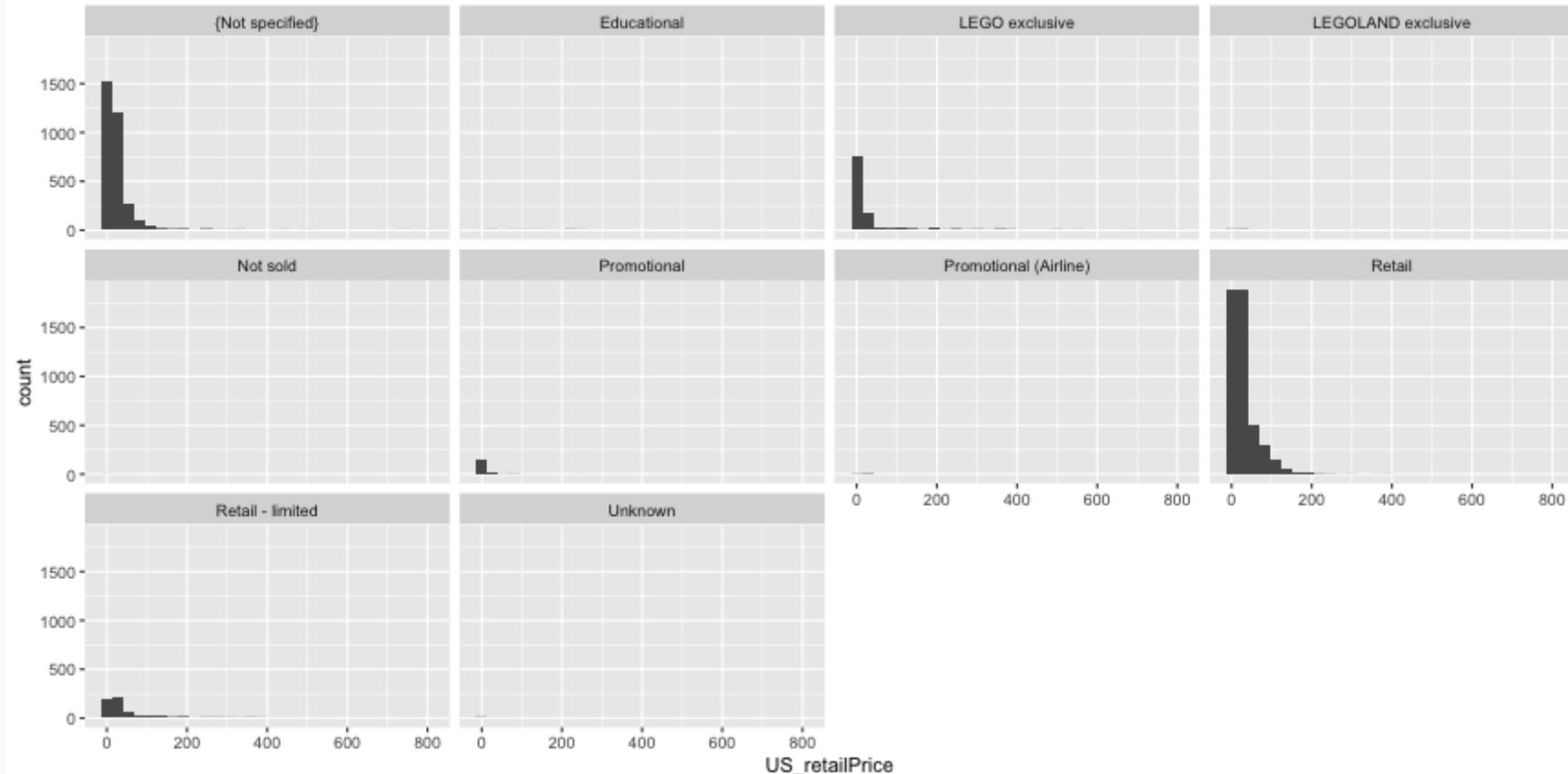
Histograms (cont.)

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



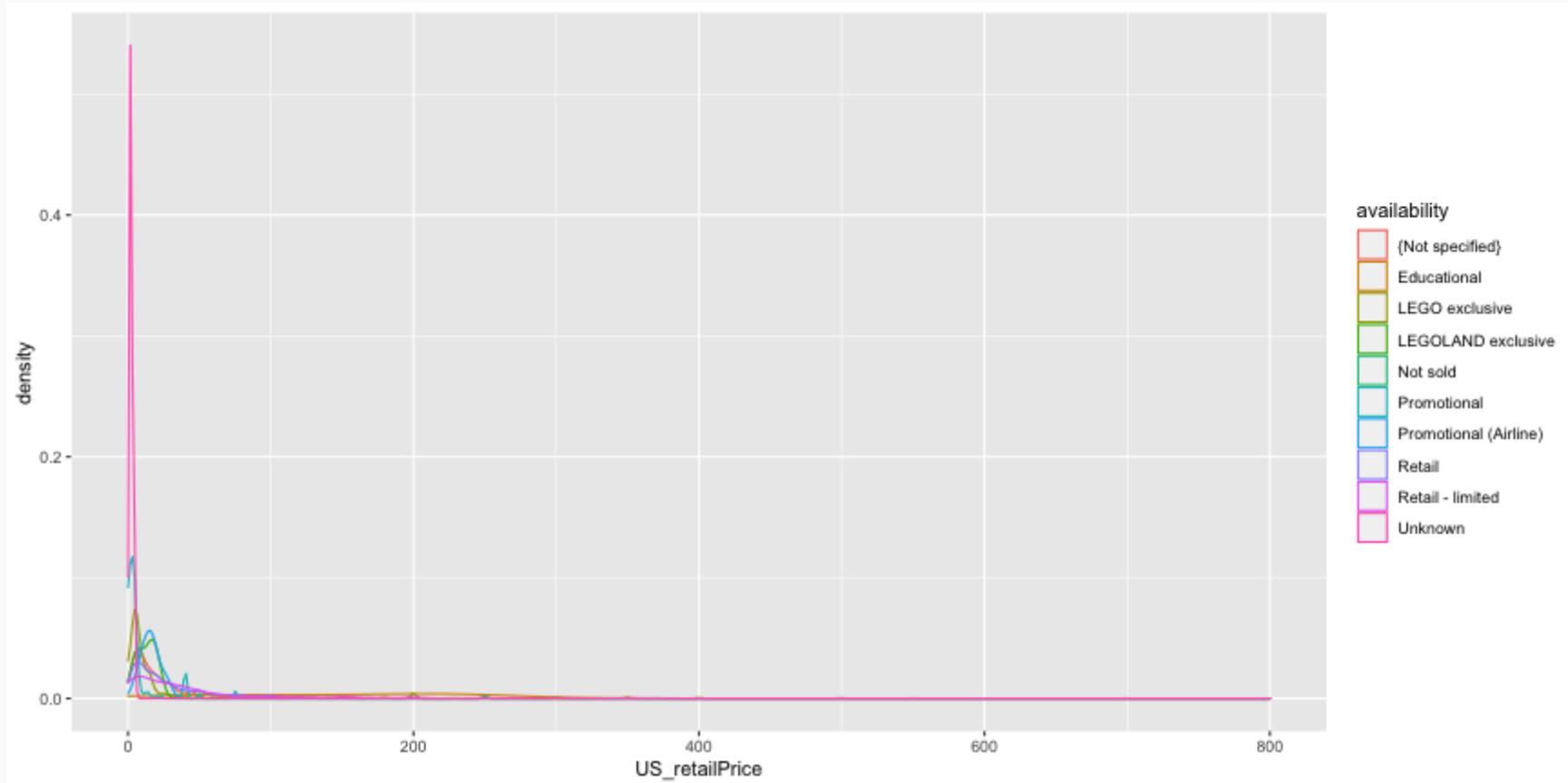
Histograms (cont.)

```
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



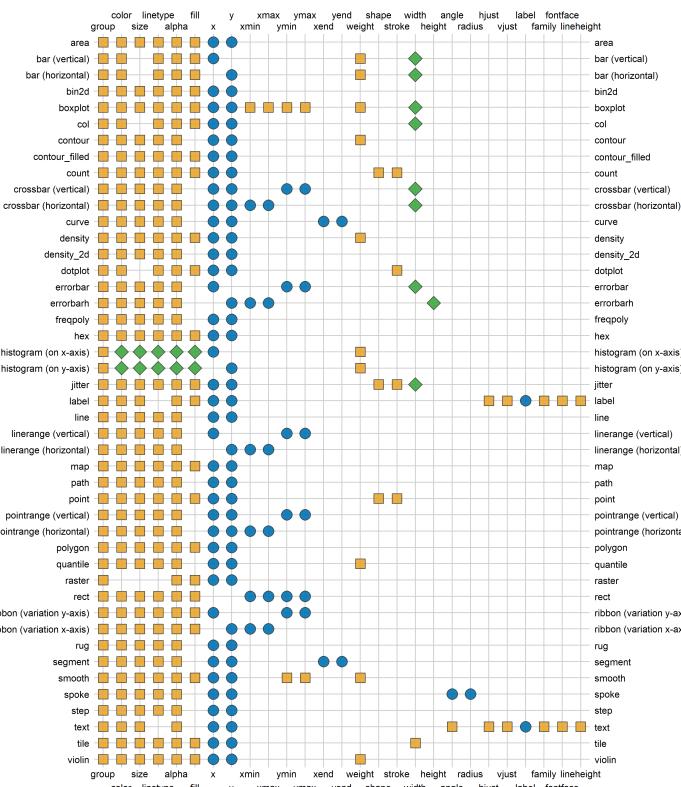
ggplot2 aesthetics cheat sheet

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()`



● usually must be inside aes() ■ can be inside aes() ♦ must be outside aes()

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

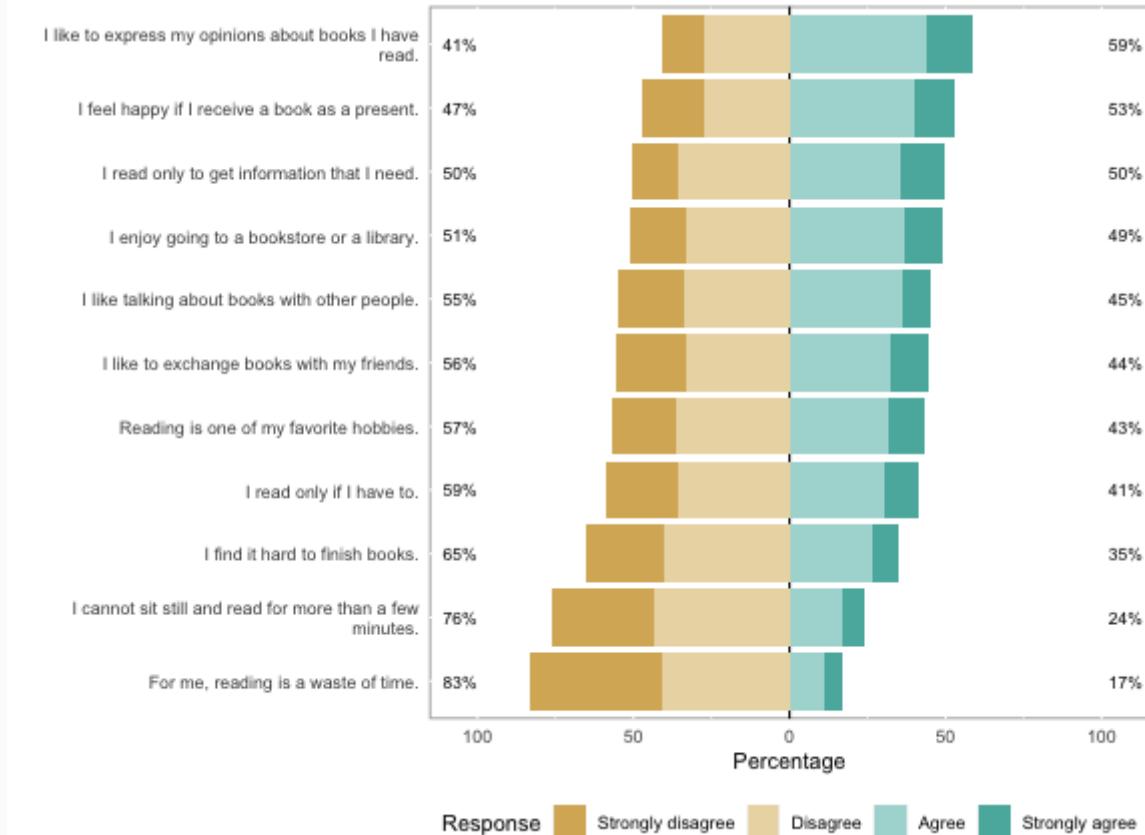
```
l24 <- likert(items24)
summary(l24)
```

```
##                                     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

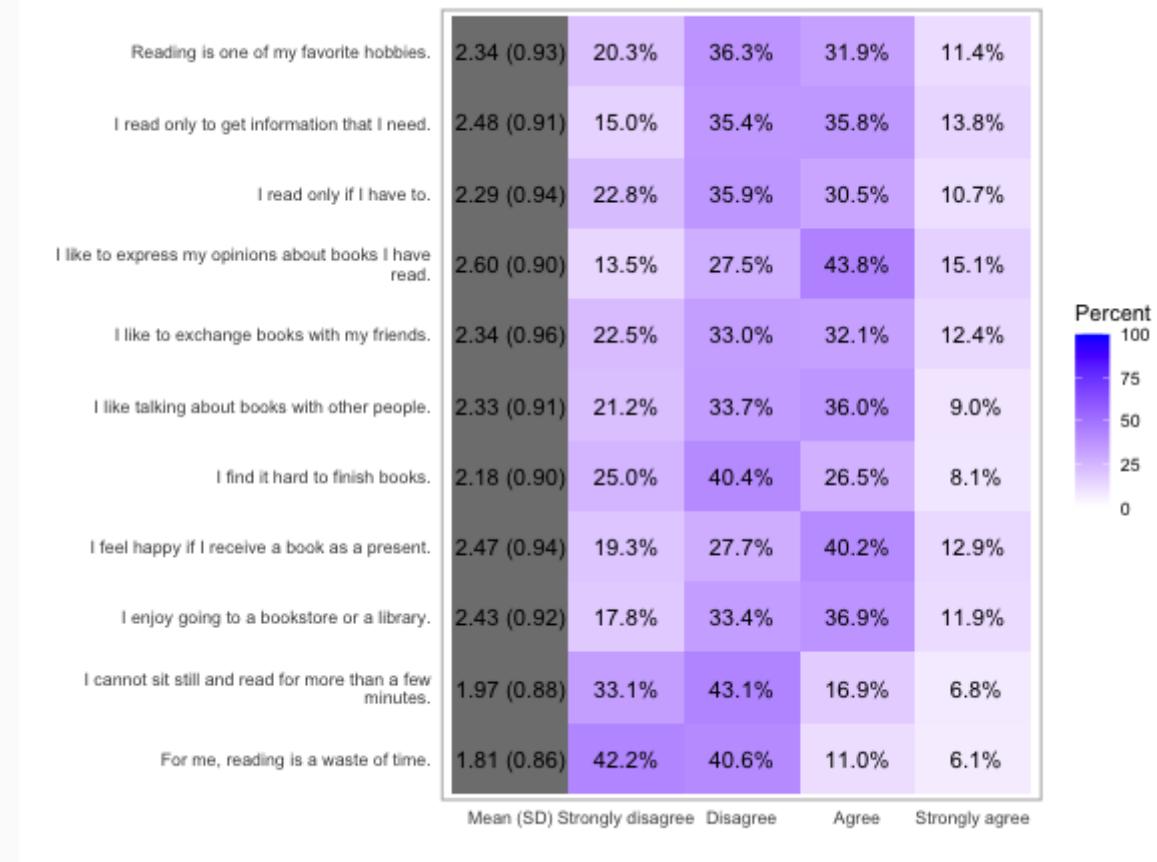
```
plot(l24)
```





likert Plots

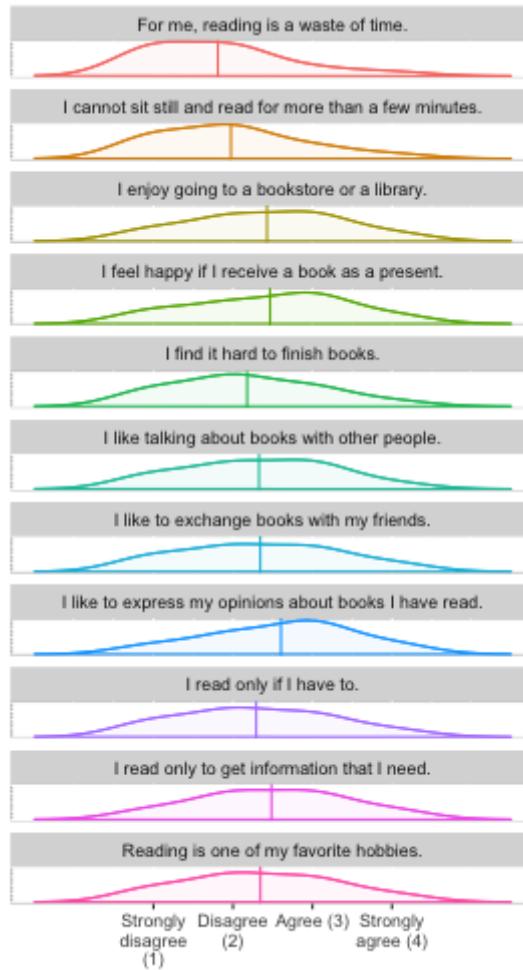
```
plot(l24, type='heat')
```





likert Plots

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



Dual Scales

Some problems¹:

- The designer has to make choices about scales and this can have a big impact on the viewer
- "Cross-over points" where one series cross another are results of the design choices, not intrinsic to the data, and viewers (particularly unsophisticated viewers)
- They make it easier to lazily associate correlation with causation, not taking into account autocorrelation and other time-series issues
- Because of the issues above, in malicious hands they make it possible to deliberately mislead

This example looks at the relationship between NZ dollar exchange rate and trade weighted index.

```
DATA606::shiny_demo('DualScales', package='DATA606')
```

My advise:

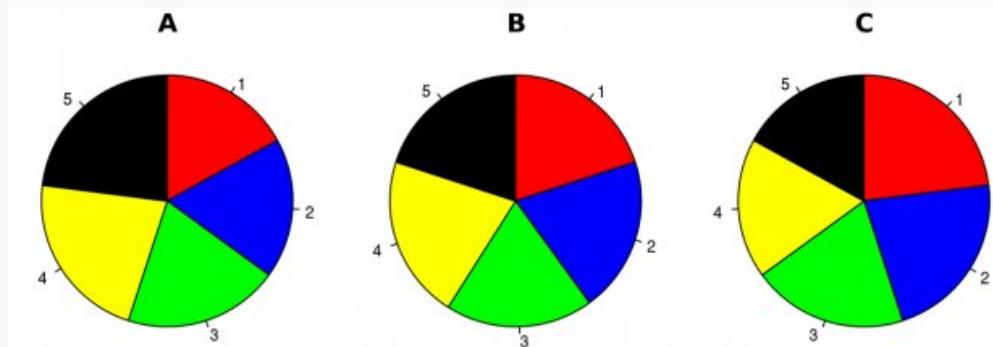
- Avoid using them. You can usually do better with other plot types.
- When necessary (or compelled) to use them, rescale (using z-scores, we'll discuss this in a few weeks)

¹ <http://blog.revolutionanalytics.com/2016/08/dual-axis-time-series.html>

² <http://ellisp.github.io/blog/2016/08/18/dualaxes>

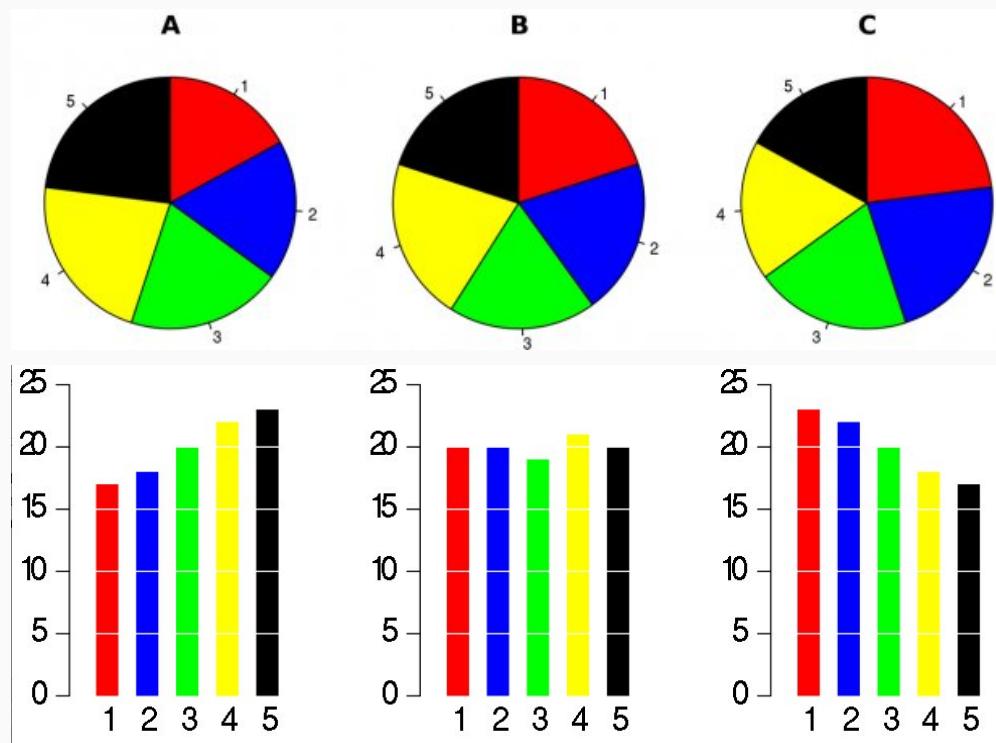
Pie Charts

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.



"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/CA1dbnMtqQ7Zyj5Y8>

1. What was the most important thing you learned during this class?
2. What important question remains unanswered for you?

