

Data visualization with ggplot2

IS381 - Statistics & Probability for Data Analytics

Jason Bryer, Ph.D.

September 15, 2025

One Minute Paper Results

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

NULL

What important question remains unanswered for you?

NULL

Grammar 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

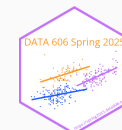
```
facet_wrap(~ cut), facet_grid(~ cut)
```

- Scales

```
scale_y_log10()
```

- Other options

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

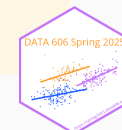


Lots of geoms



```
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.



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>) +  
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>)) +  
  stat = <STAT>, position = <POSITION> +  
  <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.

ggplot(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, unemployment))  
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, yend, 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 = unemployment - 900, ymax = unemployment + 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(fit))
```

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

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_tile()(aes(fill = z)), x, y, alpha, color, fill, linetype, size, width

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, unemployment))
```

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

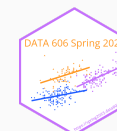
maps

```
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



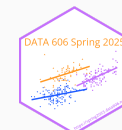
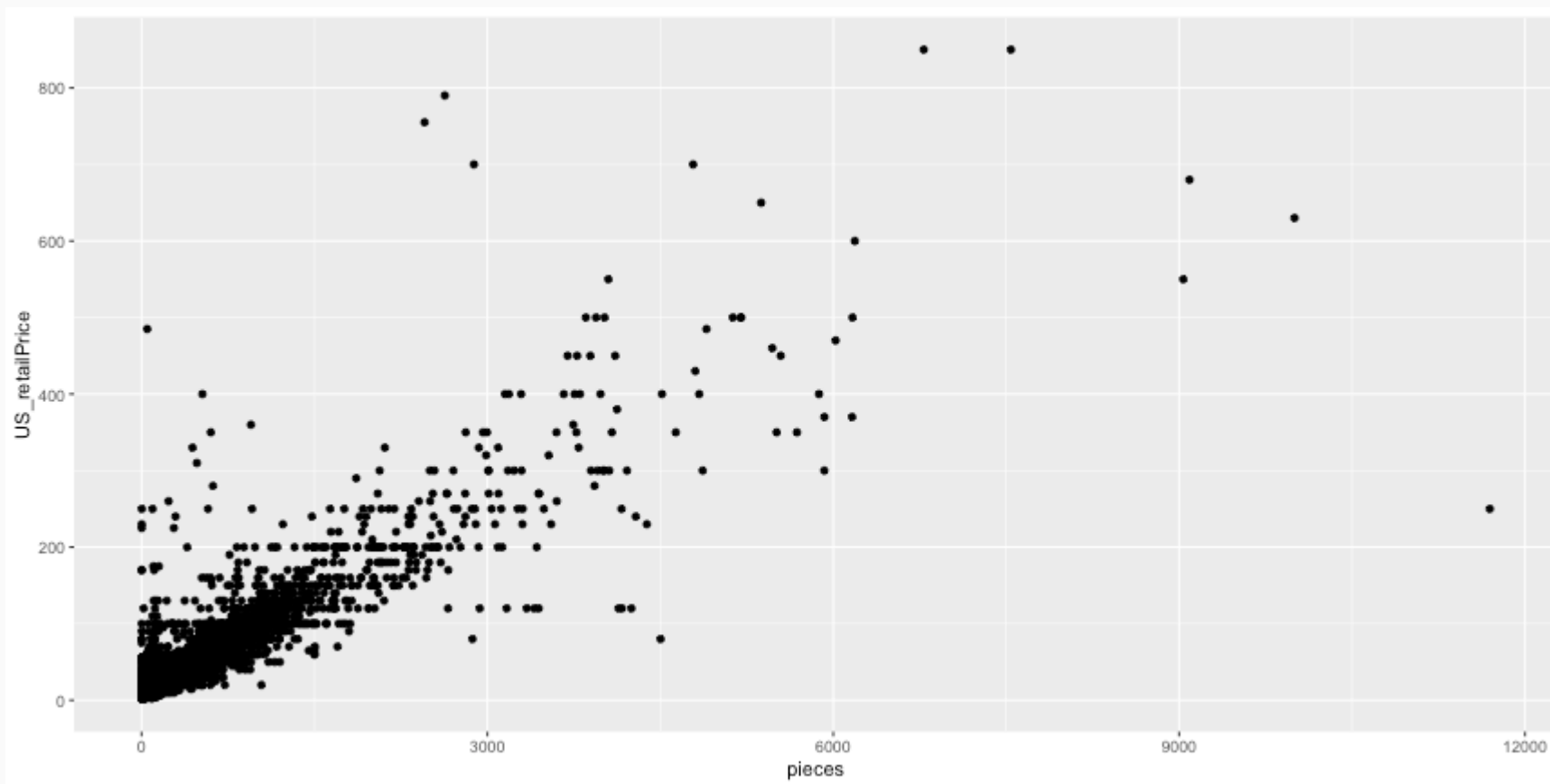
RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at <http://ggplot2.tidyverse.org> • ggplot2 2.1.0 • Updated: 2016-11



Scatterplot



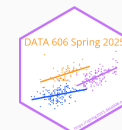
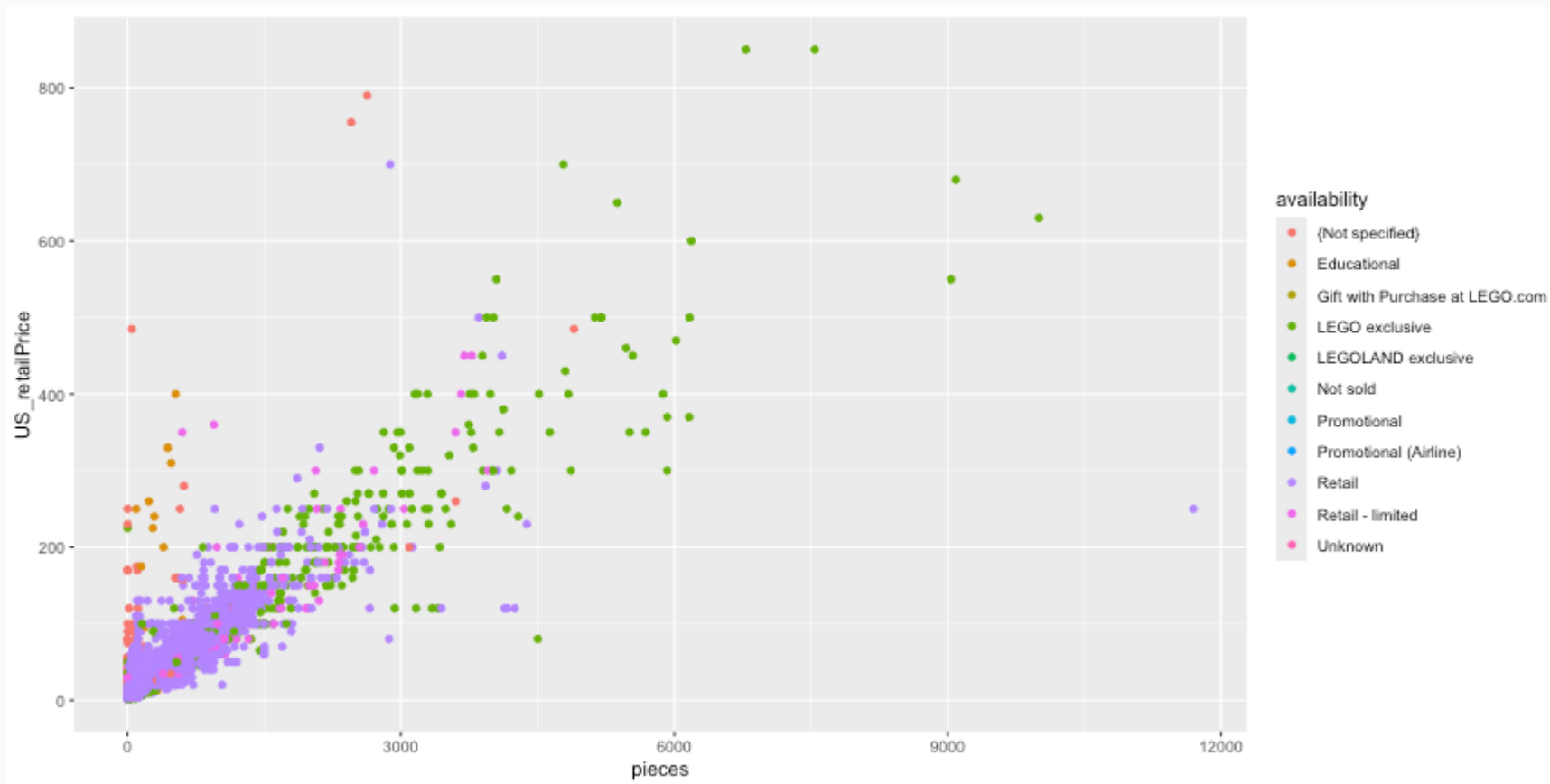
```
ggplot(legosets, aes(x=pieces, y=US_retailPrice)) + geom_point()
```



Scatterplot (cont.)



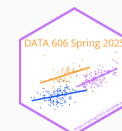
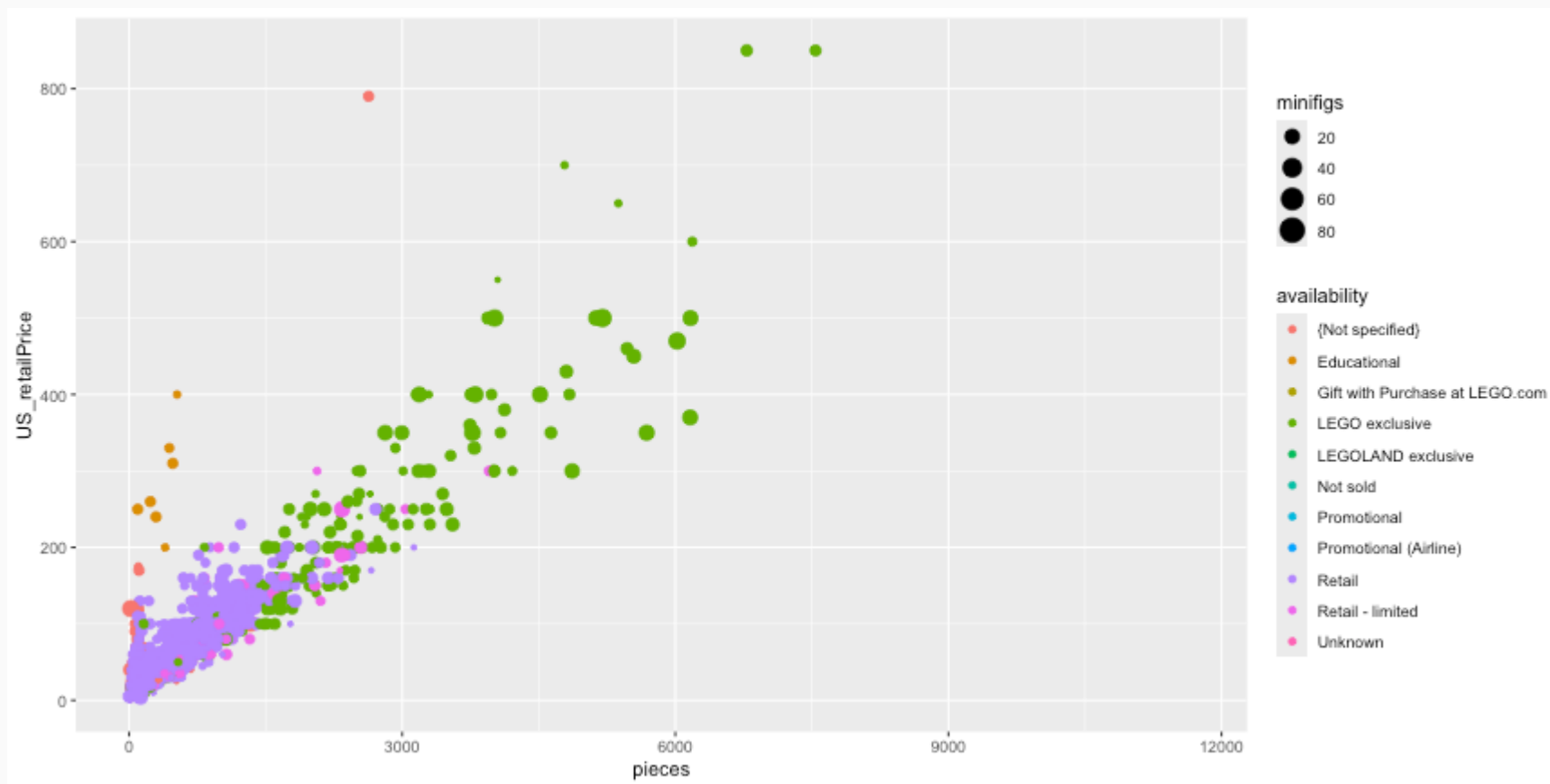
```
ggplot(legosets, aes(x=pieces, y=US_retailPrice, color=availability)) + geom_point()
```



Scatterplot (cont.)



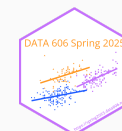
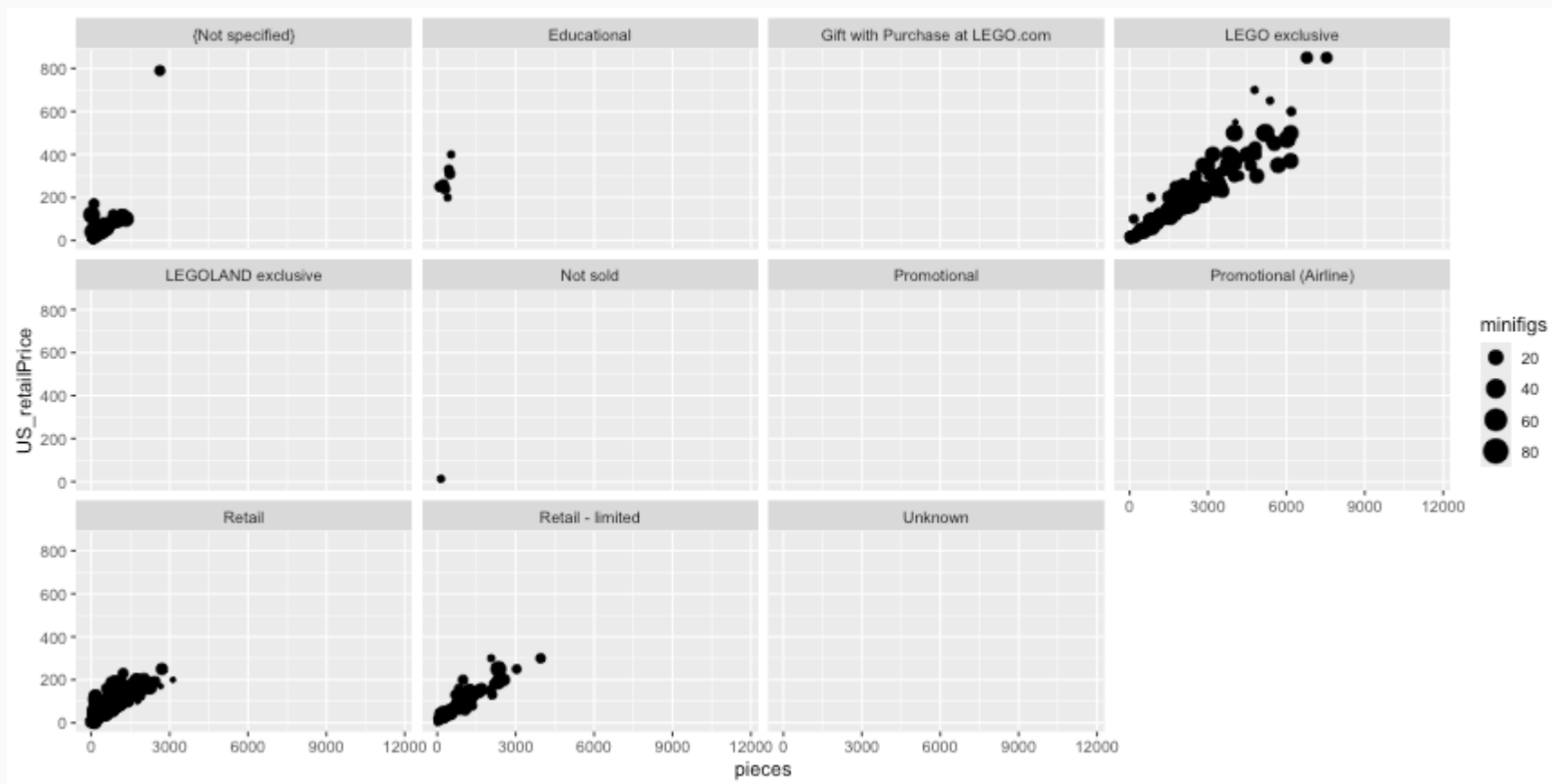
```
ggplot(legosets, aes(x=pieces, y=US_retailPrice, size=minifigs, color=availability)) + geom_point()
```



Scatterplot (cont.)



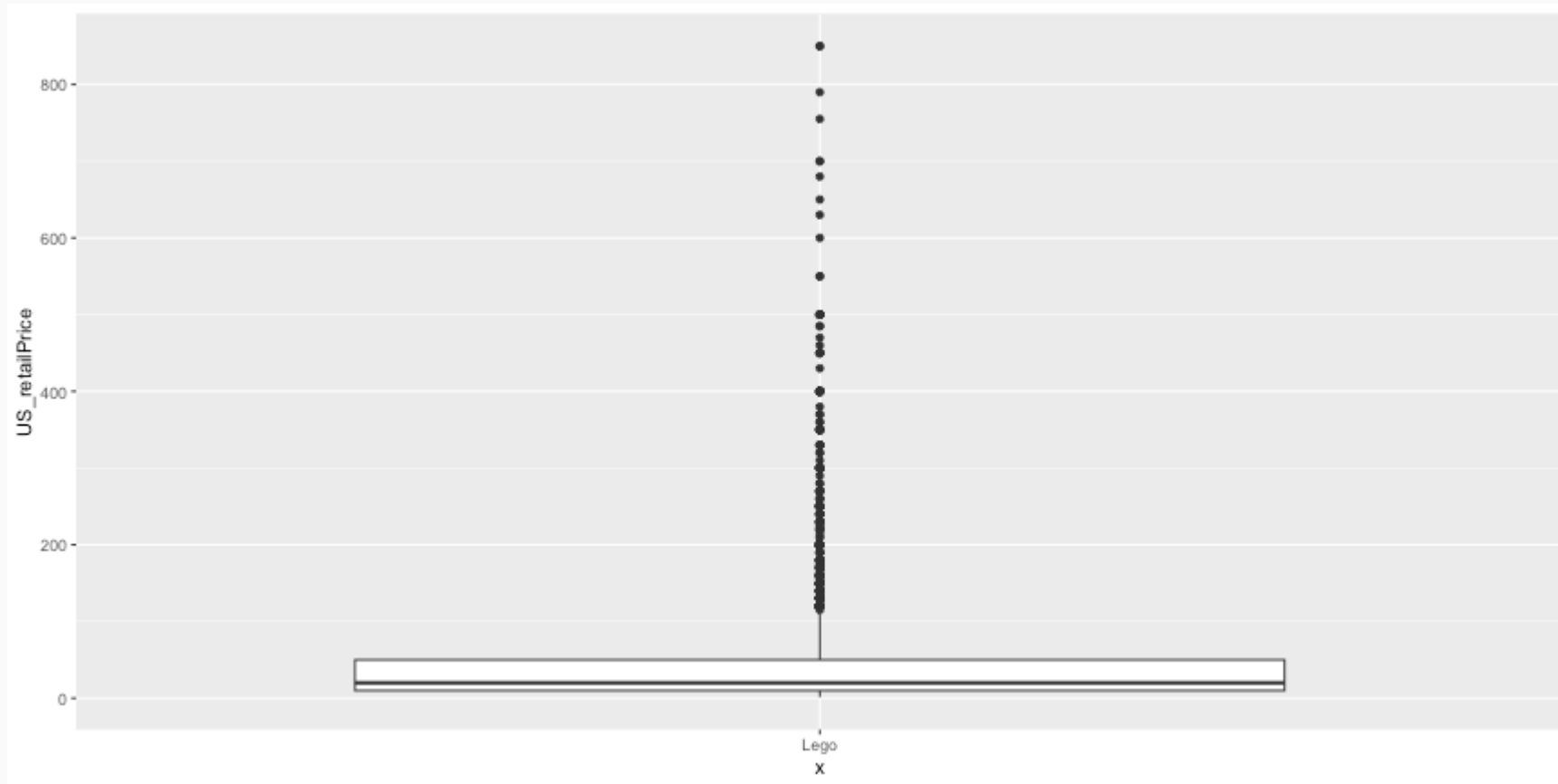
```
ggplot(legosets, aes(x=pieces, y=US_retailPrice, size=minifigs)) + geom_point() + facet_wrap(~ availability)
```



Boxplots



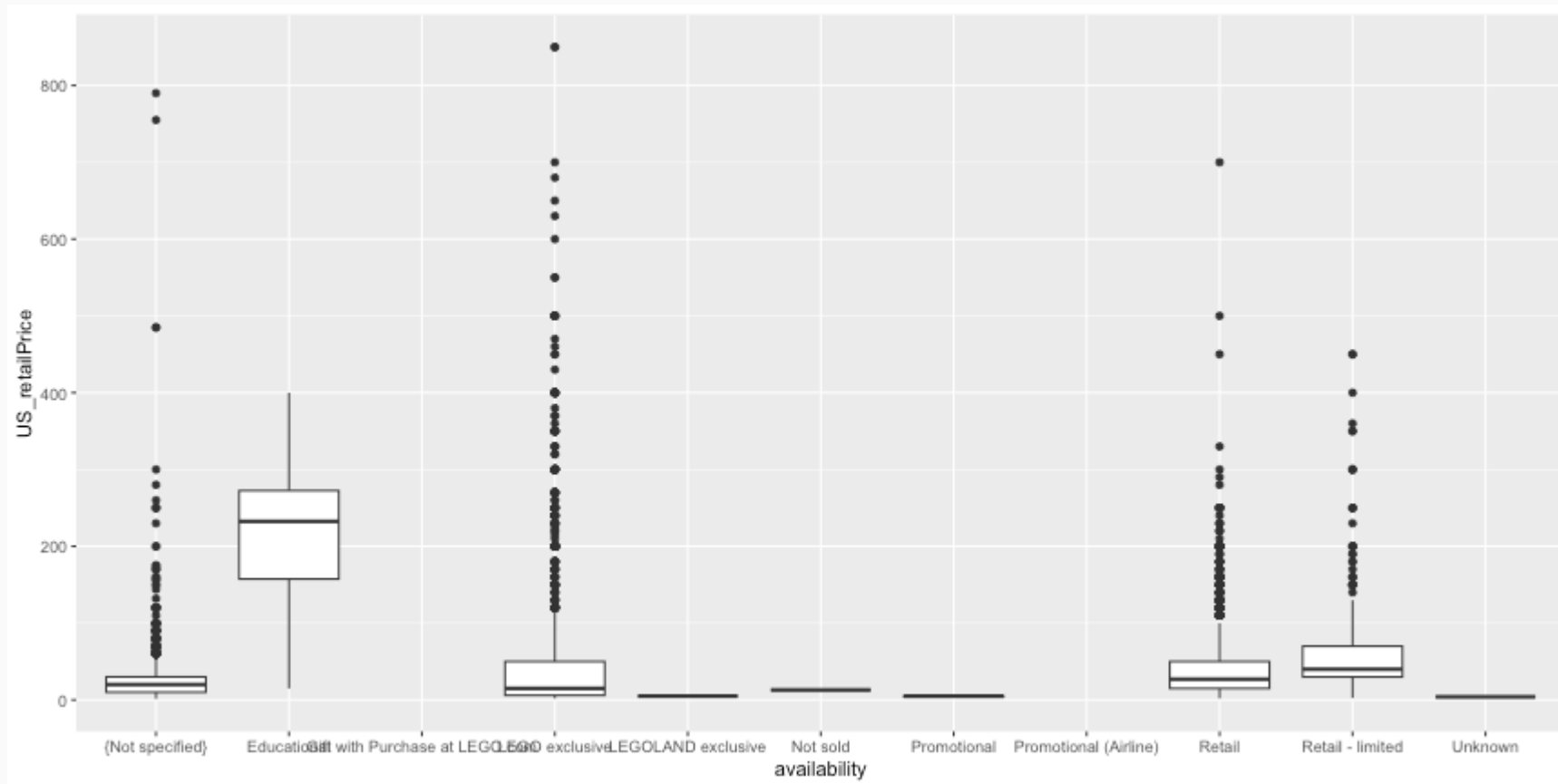
```
ggplot(legosets, aes(x='Lego', y=US_retailPrice)) + geom_boxplot()
```



Boxplots (cont.)



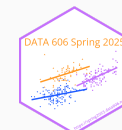
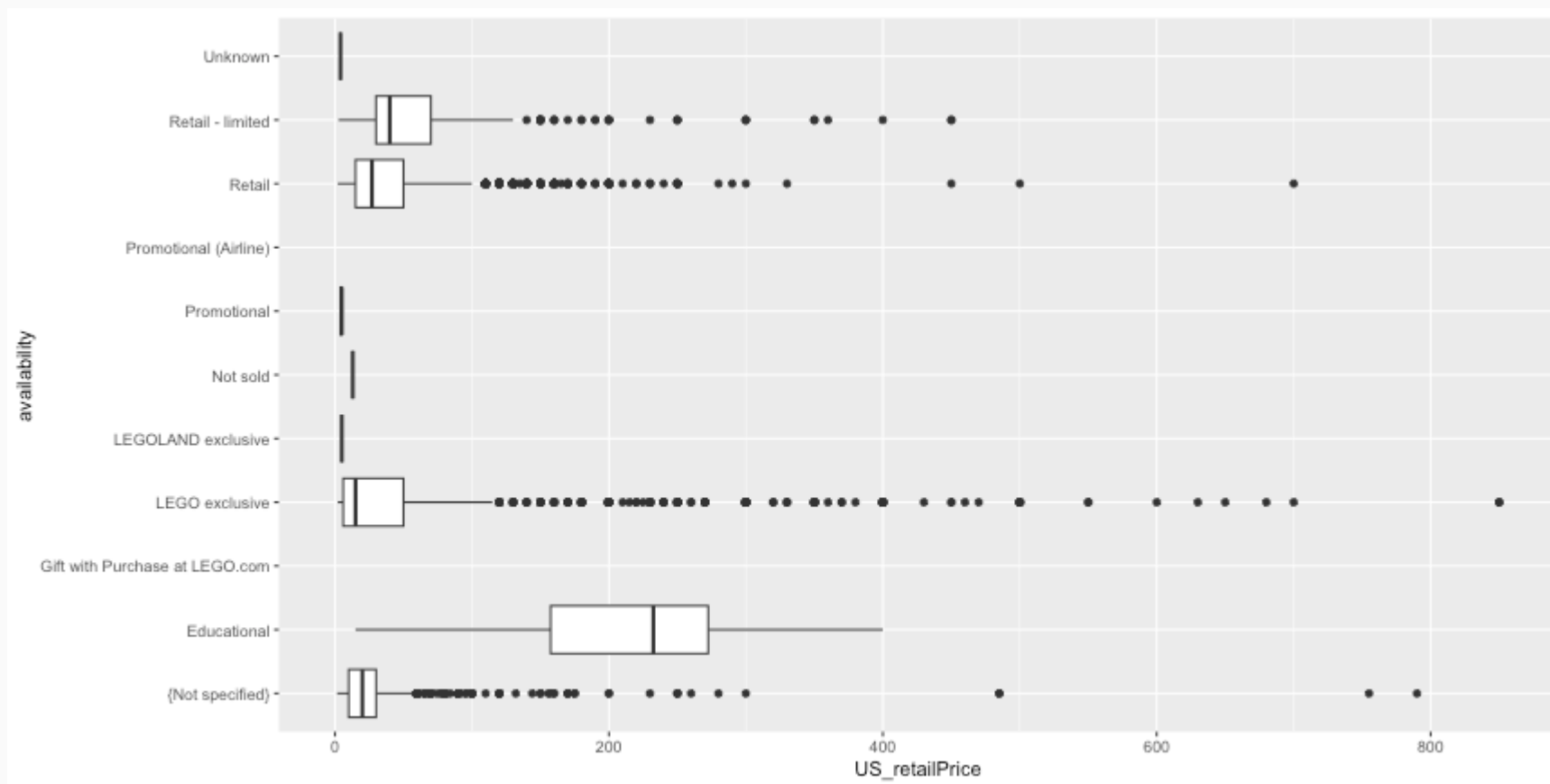
```
ggplot(legosets, aes(x=availability, y=US_retailPrice)) + geom_boxplot()
```



Boxplot (cont.)



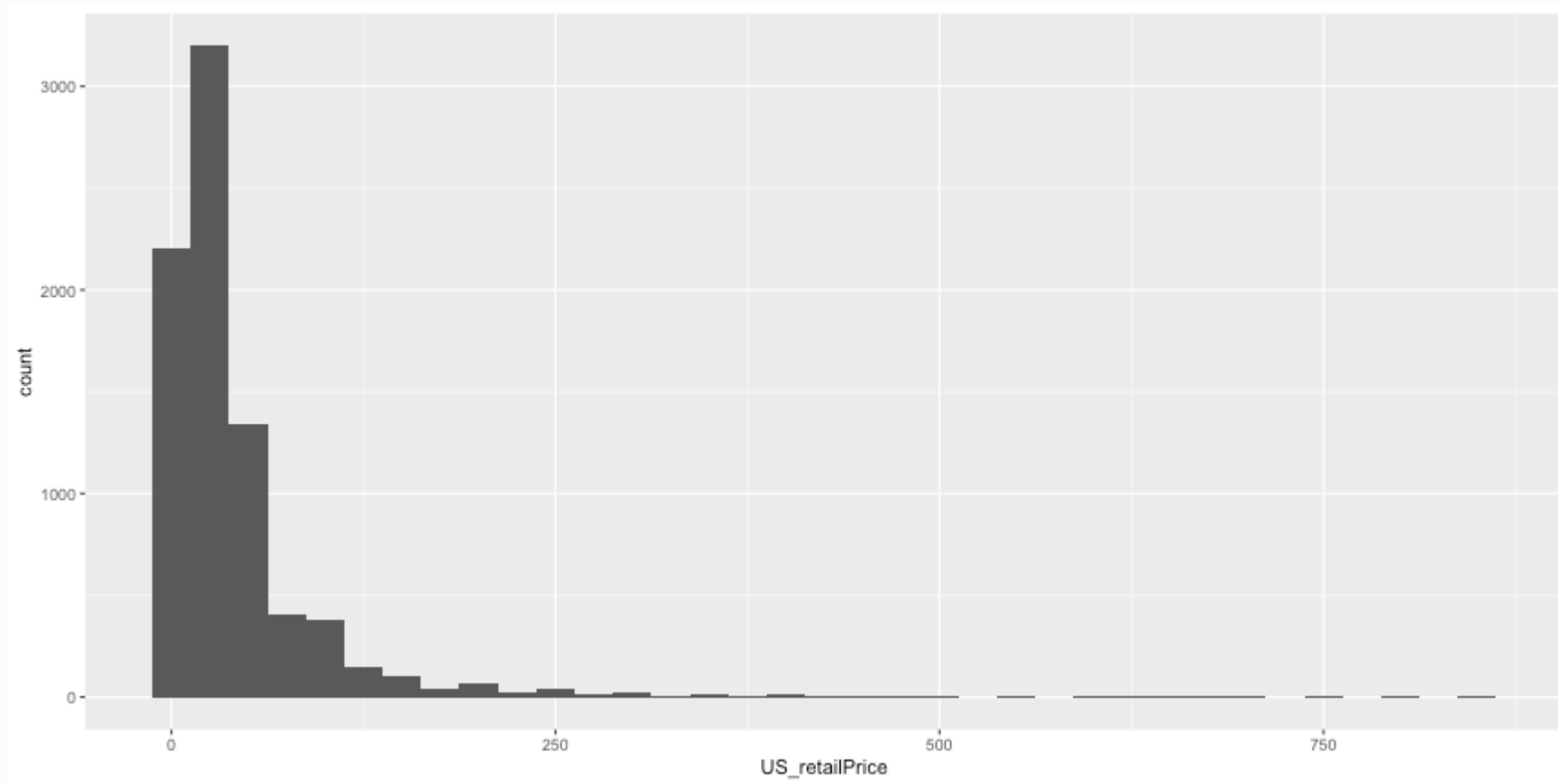
```
ggplot(legosets, aes(x=availability, y=US_retailPrice)) + geom_boxplot() + coord_flip()
```



Histograms



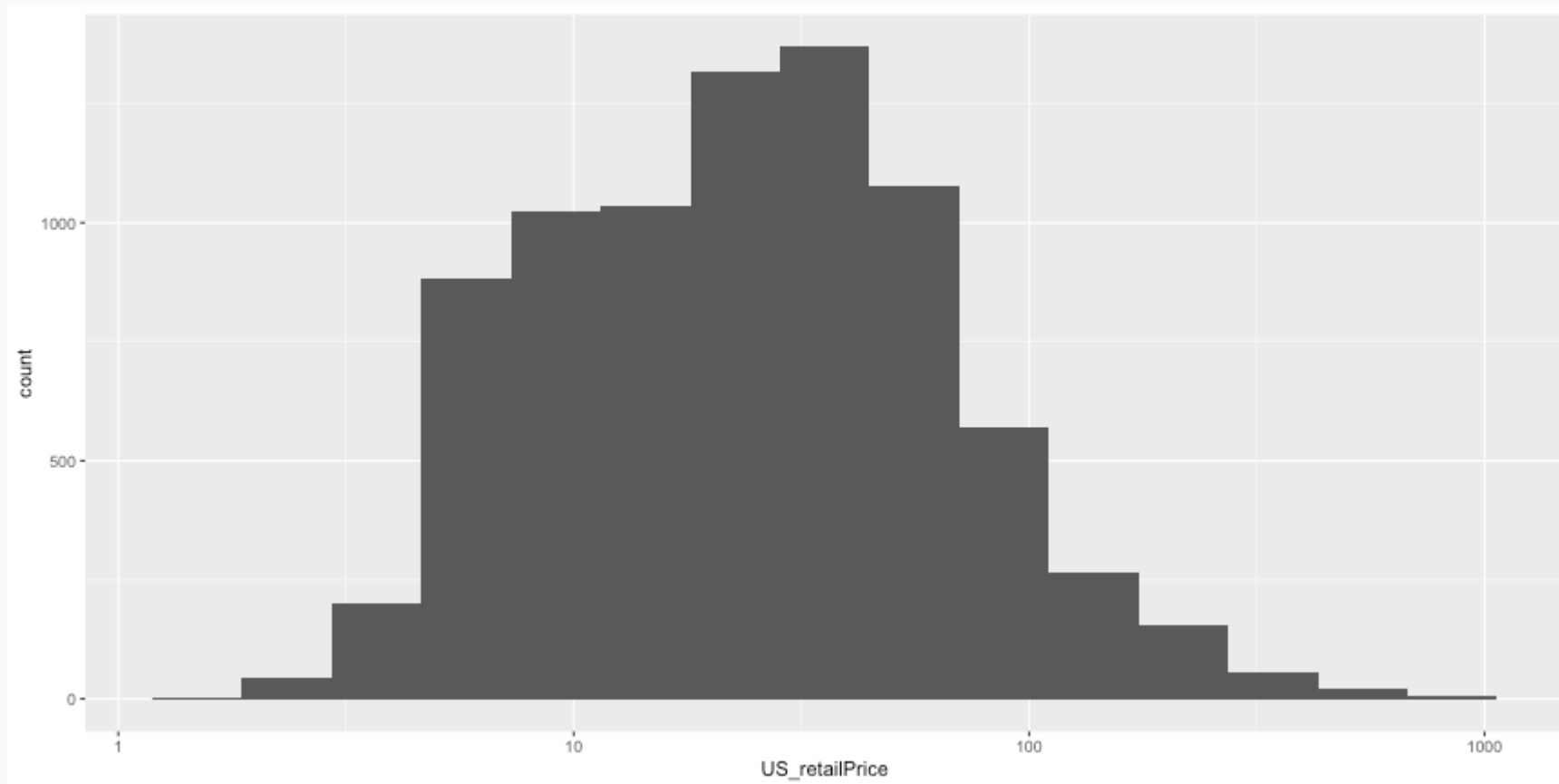
```
ggplot(legosets, aes(x = US_retailPrice)) + geom_histogram(binwidth = 25)
```



Histograms (cont.)



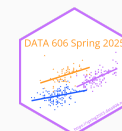
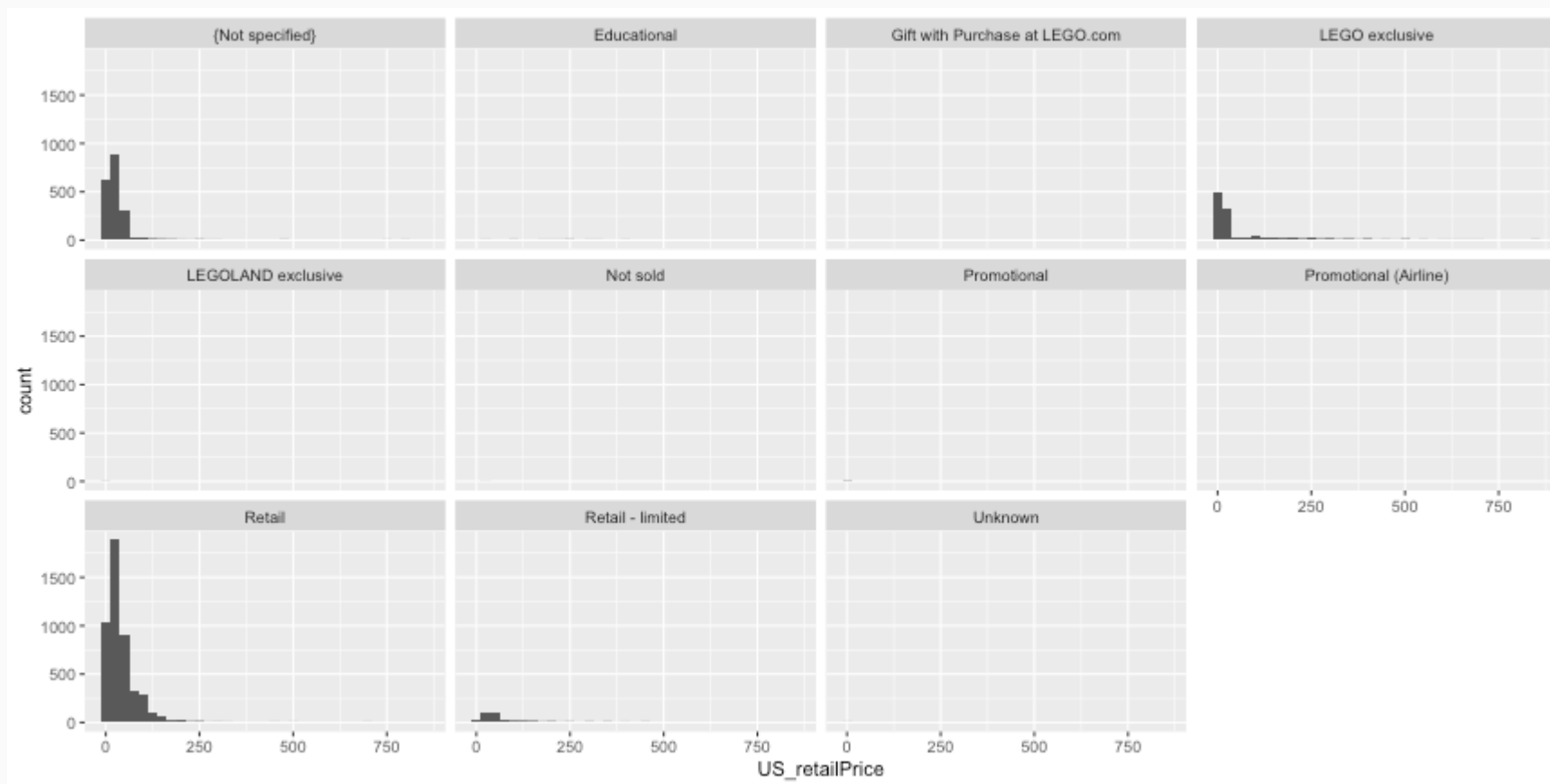
```
ggplot(legosets, aes(x = US_retailPrice)) + geom_histogram(bins = 15) + scale_x_log10()
```



Histograms (cont.)



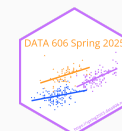
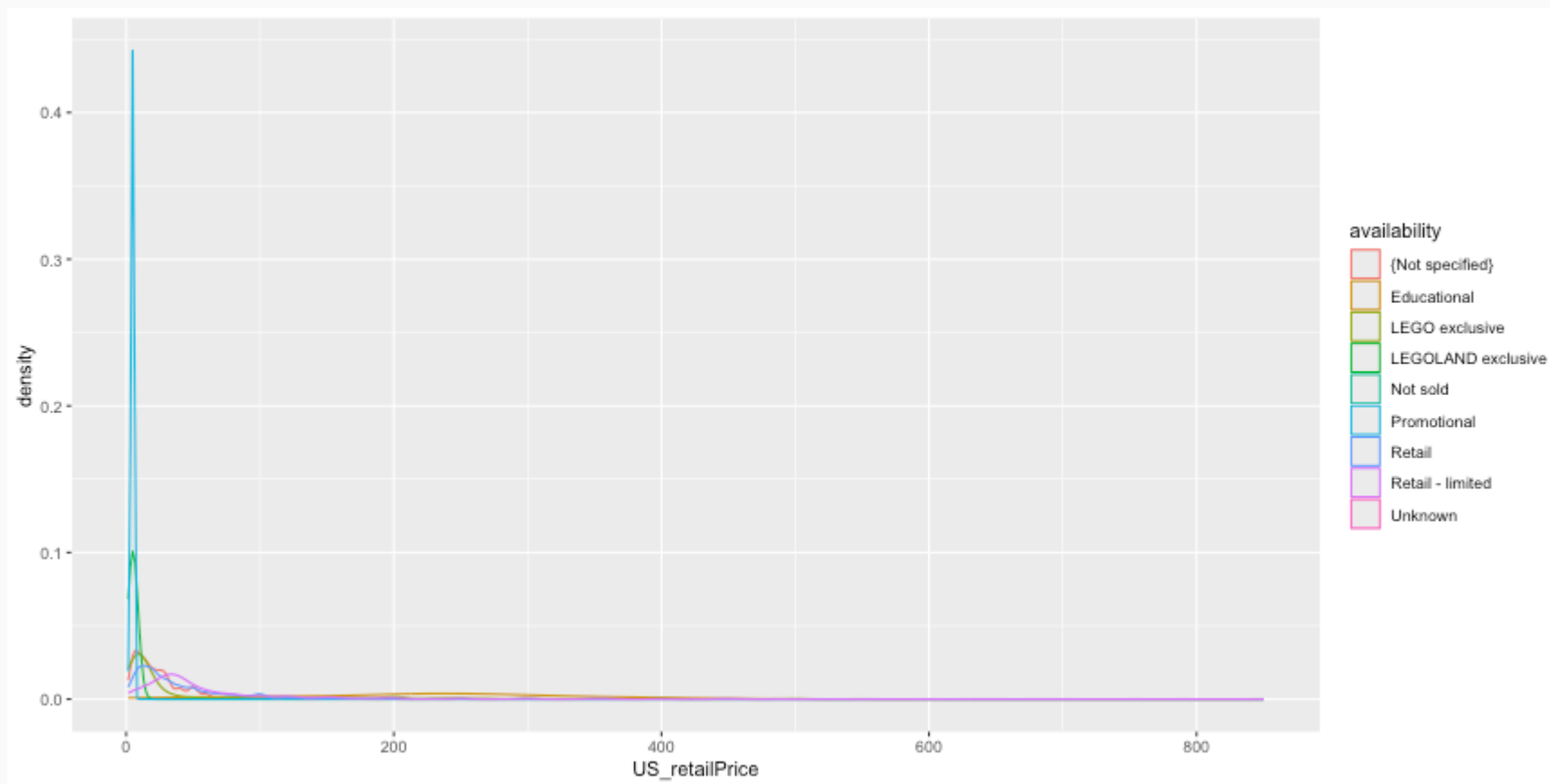
```
ggplot(legosets, aes(x = US_retailPrice)) + geom_histogram(binwidth = 25) + facet_wrap(~ availability)
```



Density Plots



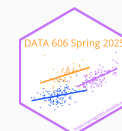
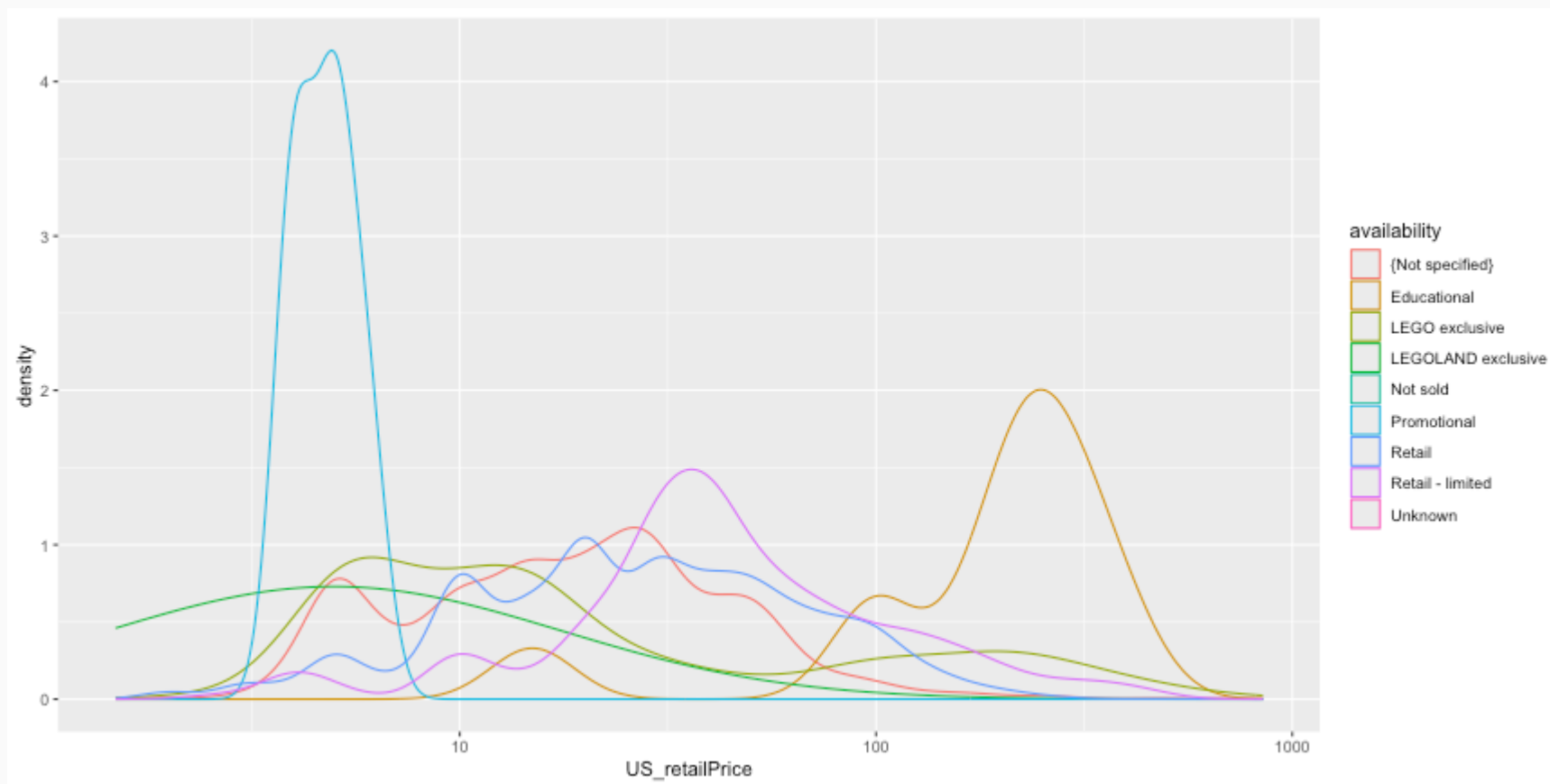
```
ggplot(legosets, aes(x = US_retailPrice, color = availability)) + geom_density()
```



Density Plots (cont.)



```
ggplot(legosets, aes(x = US_retailPrice, color = availability)) + geom_density() + scale_x_log10()
```



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)`

	group	color	linetype	fill	alpha	size	x	y	xmin	ymax	yend	shape	width	angle	hjust	label	fontface	lineheight	
area																			area
bar (vertical)																			bar (vertical)
bar (horizontal)																			bar (horizontal)
bin2d																			bin2d
boxplot																			boxplot
col																			col
contour																			contour
contour_filled																			contour_filled
count																			count
crossbar (vertical)																			crossbar (vertical)
crossbar (horizontal)																			crossbar (horizontal)
curve																			curve
density																			density
density_2d																			density_2d
dotplot																			dotplot
errorbar																			errorbar
errorbarh																			errorbarh
freqpoly																			freqpoly
hex																			hex
histogram (on x-axis)																			histogram (on x-axis)
histogram (on y-axis)																			histogram (on y-axis)
jitter																			jitter
label																			label
line																			line
linrange (vertical)																			linrange (vertical)
linrange (horizontal)																			linrange (horizontal)
map																			map
path																			path
point																			point
pointrange (vertical)																			pointrange (vertical)
pointrange (horizontal)																			pointrange (horizontal)
polygon																			polygon
quantile																			quantile
raster																			raster
rect																			rect
ribbon (variation y-axis)																			ribbon (variation y-axis)
ribbon (variation x-axis)																			ribbon (variation x-axis)
rug																			rug
segment																			segment
smooth																			smooth
spoke																			spoke
step																			step
text																			text
tile																			tile
violin																			violin

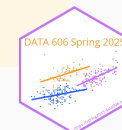
● usually must be inside `aes()` ■ can be inside `aes()` ◆ must be outside `aes()`

idea and design: Christian Burkhardt
design advice: Ida Aarnio



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.))
```





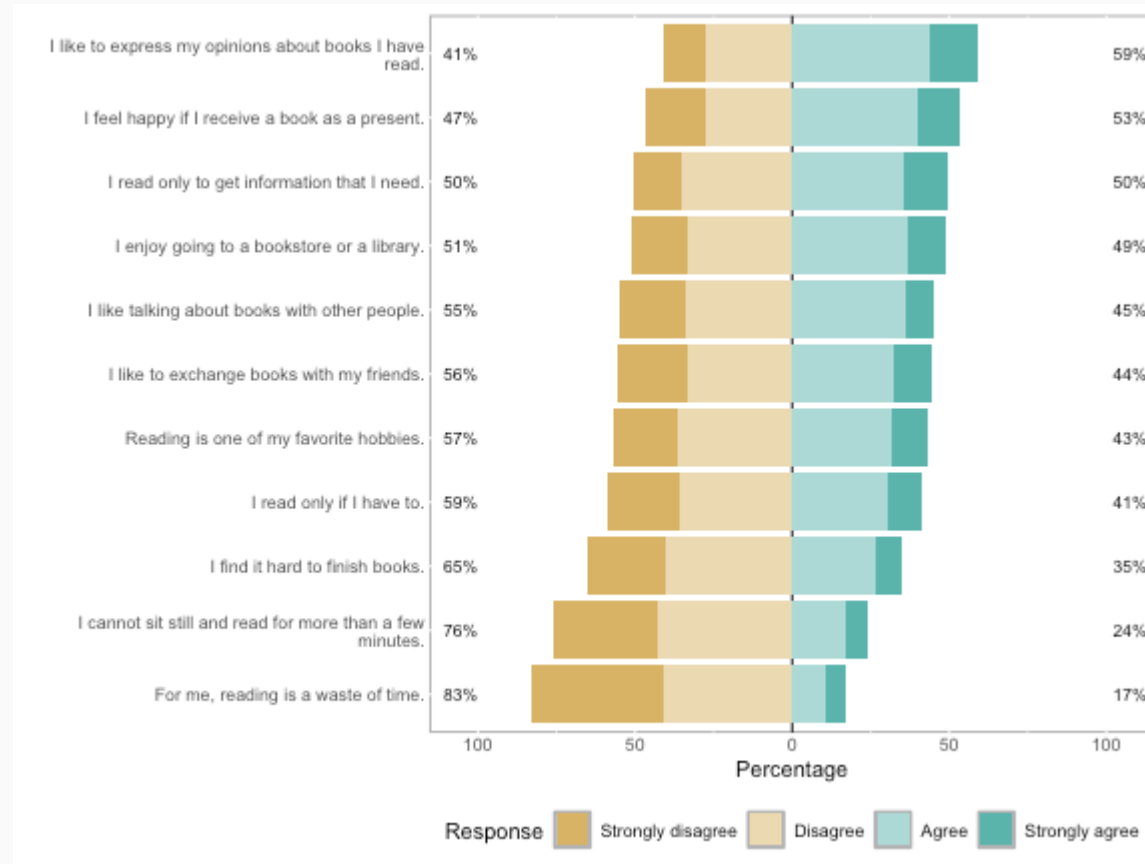
```
l24 <- likert(items24)
summary(l24)
```

```
##                               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

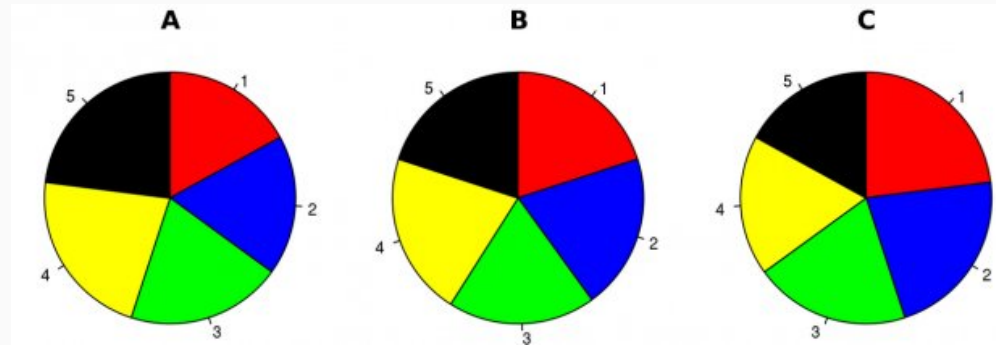


```
plot(l24)
```



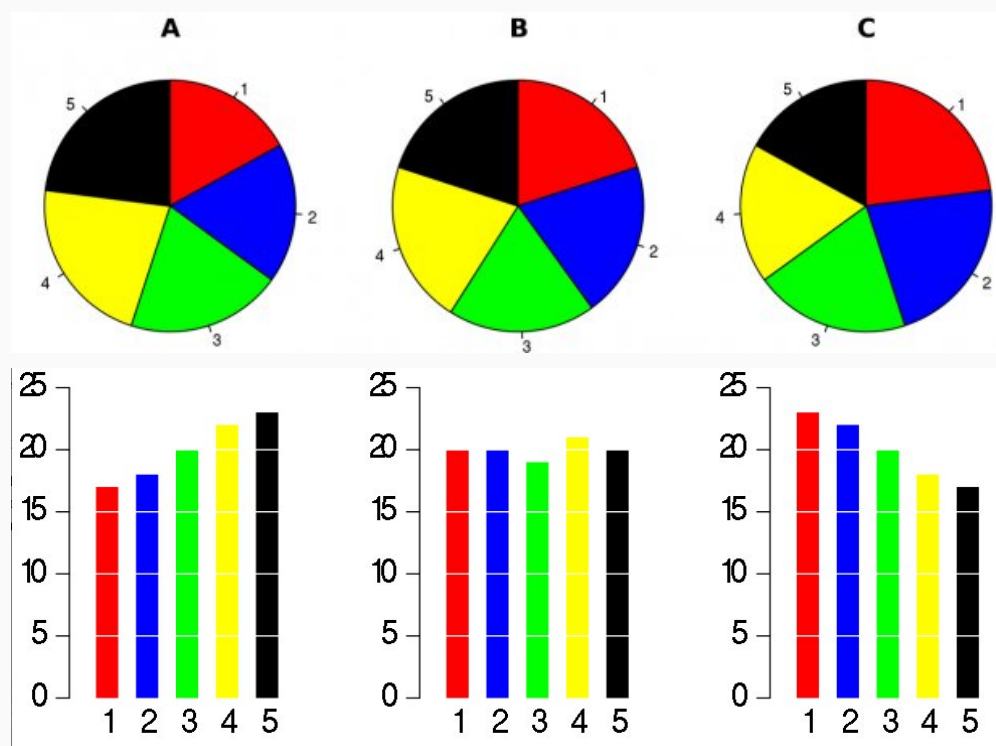
Pie Charts

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

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



<https://forms.gle/N8WjTAysfKbGLptLA>