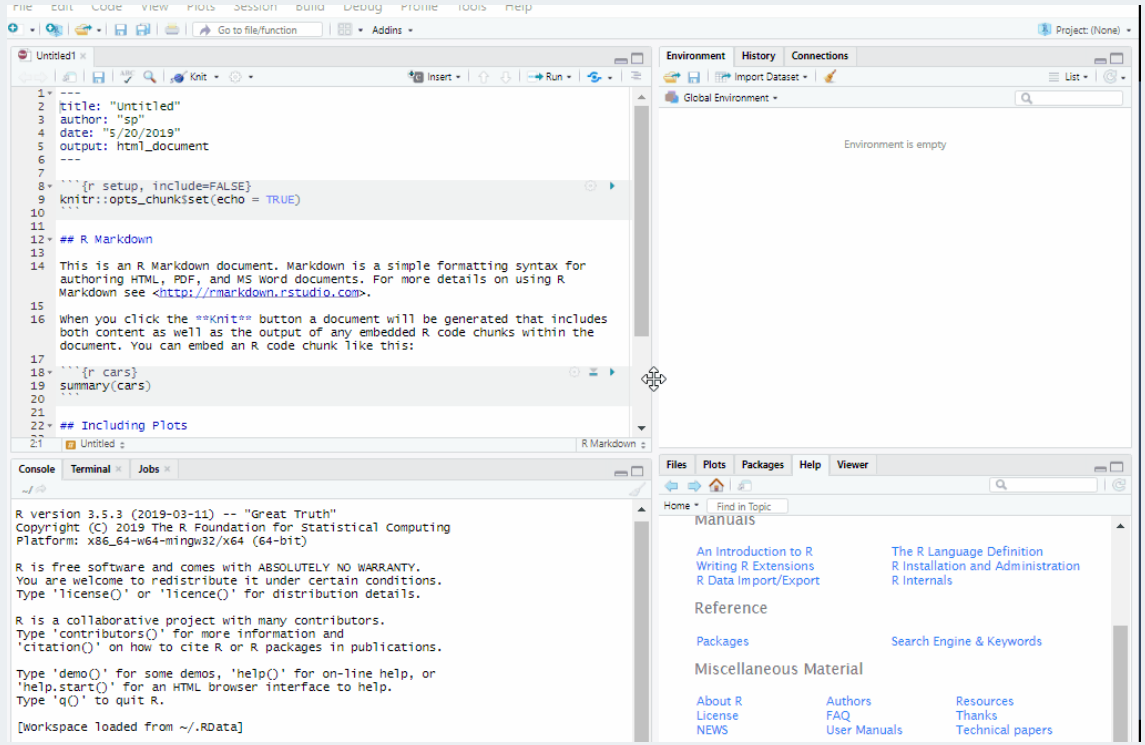
Day 2 CONJ 620

Chapter 2: Data Visualization

Meike Niederhausen, PhD

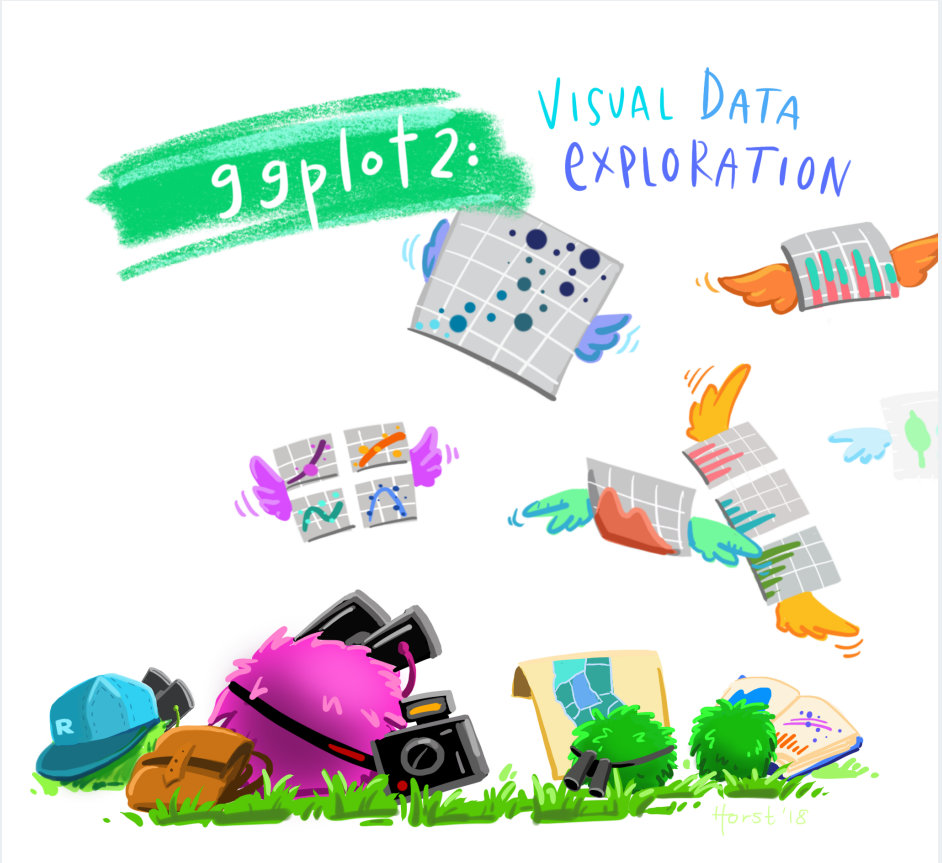
MoRitz's tip of the day

Customize your RStudio interface!
https://www.pipinghotdata.com/posts/2020-09-07-introducing-the-rstudio-ide-and-r-markdown/#background



Goals for today

- R Packages
- **Exploratory Data Analysis (EDA)** (Sections 1.4, 1.5, 1.6, 1.7.1)
    - Data visualization with ggplot
    - numerical & categorical variables, and relationships between variables
    - Summarizing numerical data
    - Frequency (two-way) tables
- Some **data wrangling** techniques along the way



Allison Horst

ggplot "theory"

- The "The Grammar of Graphics," is the theoretical basis for the `ggplot2` package.
  - Much like how we construct sentences in any language by using a linguistic grammar (nouns, verbs, etc.), the grammar of graphics allows us to specify the components of a statistical graphic.

In short, the grammar tells us that:

> A statistical graphic is a `mapping` of `data` variables to `aes`thetic attributes of `geom`etric objects.

3 **essential** components to a graphic:

- `data`: the data-set comprised of variables that we plot
- `geom`: this refers to our type of `geom`etric objects we see in our plot (points, lines, bars, etc.)
- `aes`: aesthetic attributes of the geometric object that we can perceive on a graphic. For example, x/y position, color, shape, and size. Each assigned aesthetic attribute can be mapped to a variable in our data-set.

Basics of a ggplot

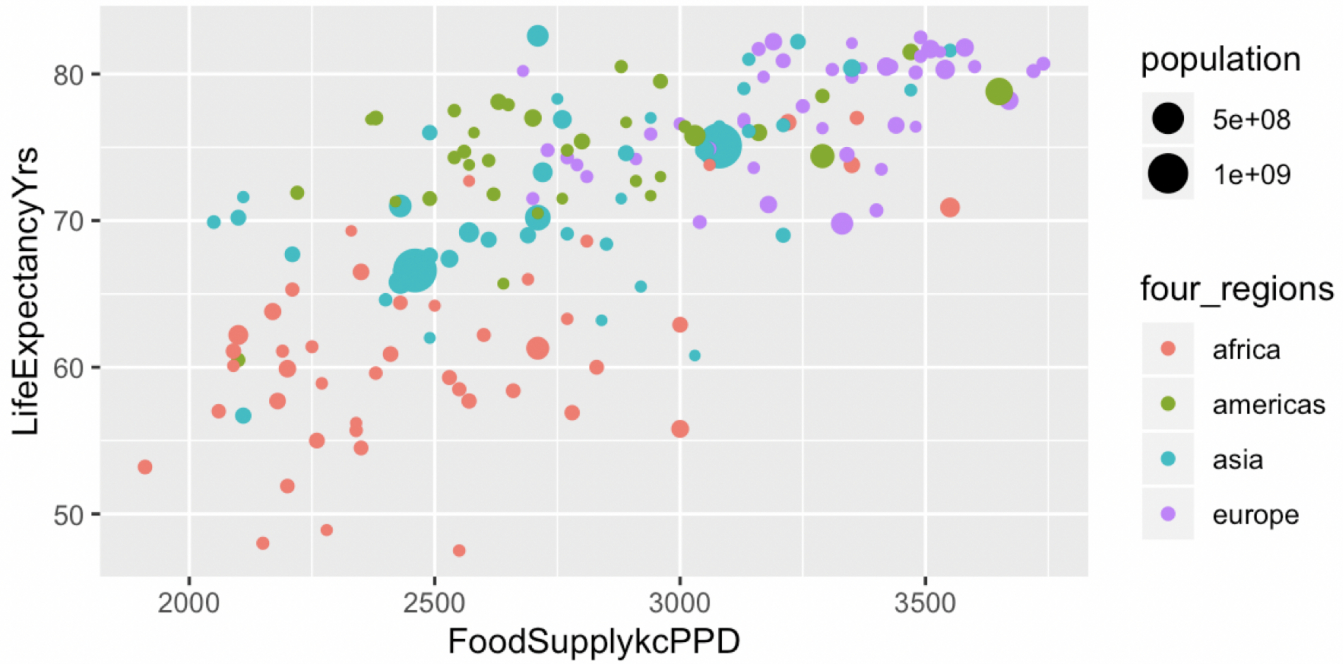
Function

Dataset

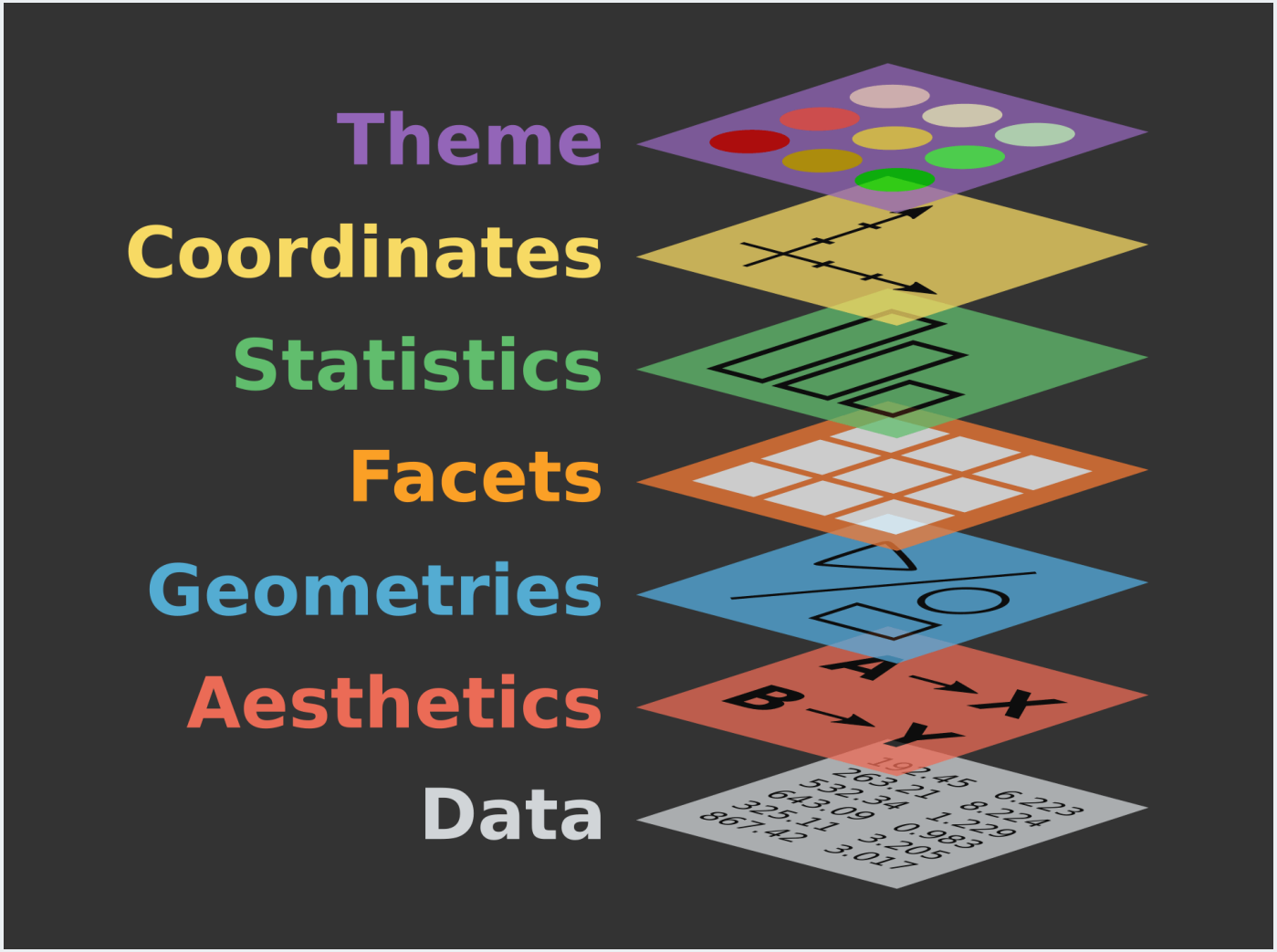
```
ggplot(data = gapminder2011,
       aes(x = FoodSupplykcPPD, y = LifeExpectancyYrs,
           color = four_regions, size = population)) +
       geom_point()
```

Which variables to plot

What kind of plot to make



Layering components in a ggplot



Plot types (geometries) covered in ModernDive Chapter 2

TABLE 2.4: Summary of Five Named Graphs

	Named graph	Shows	Geometric object	Notes
1	Scatterplot	Relationship between 2 numerical variables	`geom_point()`	
2	Linegraph	Relationship between 2 numerical variables	`geom_line()`	Used when there is a sequential order to x-variable, e.g., time
3	Histogram	Distribution of 1 numerical variable	`geom_histogram()`	Facetted histograms show the distribution of 1 numerical variable split by the values of another variable
4	Boxplot	Distribution of 1 numerical variable split by the values of another variable	`geom_boxplot()`	
5	Barplot	Distribution of 1 categorical variable	`geom_bar()` when counts are not pre-counted, `geom_col()` when counts are pre-counted	Stacked, side-by-side, and faceted barplots show the joint distribution of 2 categorical variables

https://moderndive.netlify.app/2-viz.html#summary-table

Case study: discrimination in developmental disability support (V&H 1.7.1)

- In the US, individuals with developmental disabilities typically receive services and support from state governments.
  - California allocates funds to developmentally disabled residents through the Department of Developmental Services (DDS)
  - Recipients of DDS funds are referred to as "consumers."
- Dataset `dds.discr`
  - sample of 1,000 DDS consumers (out of a total of ~ 250,000)
  - age, gender, race/ethnicity, and DDS annual financial support per consumer
- **Previous research**
  - Researchers examined expenditures on consumers by ethnicity
  - Found that the mean annual expenditures on Hispanics was less than that on White non-Hispanics.
- Result: an allegation of ethnic discrimination was brought against the California DDS.

- **Question: Are the data sufficient evidence of ethnic discrimination?**

- See V&H Section 1.7.1 for more details

Load dataset `dds.discr` from package `oibiostat`

- The textbook's datasets are in the R package `oibiostat`
- If you haven't already installed the package `oibiostat`, then first do so using directions in previous slide.
- Load the `oibiostat` package and the dataset `dds.discr`
    - the code below needs to be run *every time* you restart R or knit an Rmd file

```
library(oibiostat)
data("dds.discr")
```

- After loading the dataset `dds.discr` using `data("dds.discr")`, you will see `dds.discr` in the Data list of the Environment window.

Getting to know the dataset

```
dim(dds.discr)
```

```
## [1] 1000    6
```

```
names(dds.discr)
```

```
## [1] "id"           "age.cohort"   "age"          "gender"       "expenditures"
## [6] "ethnicity"
```

```
length(unique(dds.discr$id)) # How many unique id's are there?
```

```
## [1] 1000
```

Getting to know the dataset: `str()`

- We previously used the base R structure command `str()` to get information about variable types in a dataset.
- Note this dataset is a `tibble` instead of a `data.frame`

```
str(dds.discr)      # base R
```

```
tibble [1,000 × 6] (S3: tbl_df/tbl/data.frame)
 $ id          : int [1:1000] 10210 10409 10486 10538 10568 10690 10711 10778 10820 10823 ...
 $ age.cohort  : Factor w/ 6 levels "0-5","6-12","13-17",..: 3 5 1 4 3 3 3 3 3 3 ...
 $ age         : int [1:1000] 17 37 3 19 13 15 13 17 14 13 ...
 $ gender      : Factor w/ 2 levels "Female","Male": 1 2 2 1 2 1 1 2 1 2 ...
 $ expenditures: int [1:1000] 2113 41924 1454 6400 4412 4566 3915 3873 5021 2887 ...
 $ ethnicity   : Factor w/ 8 levels "American Indian",..: 8 8 4 4 8 4 8 3 8 4 ...
 - attr(*, "spec")=
  .. cols(
  ..   ID = col_integer(),
  ..   `Age Cohort` = col_character(),
  ..   Age = col_integer(),
  ..   Gender = col_character(),
  ..   Expenditures = col_integer(),
  ..   Ethnicity = col_character()
  .. )
```

Getting to know the dataset: `summary()`

- We previously used the base R structure command `summary()` to get summary information about variables

```
summary(dds.discr)      # base R
```

```
##        id         age.cohort       age            gender     expenditures
##  Min.   :10210   0-5  : 82   Min.   : 0.0   Female:503   Min.   :  222
##  1st Qu.:31809   6-12 :175   1st Qu.:12.0   Male  :497   1st Qu.: 2899
##  Median :55384   13-17:212   Median :18.0                Median : 7026
##  Mean   :54663   18-21:199   Mean   :22.8                Mean   :18066
##  3rd Qu.:76135   22-50:226   3rd Qu.:26.0                3rd Qu.:37713
##  Max.   :99898   51+  :106   Max.   :95.0                Max.   :75098
##
##                ethnicity
##  White not Hispanic:401
##  Hispanic           :376
##  Asian              :129
##  Black              : 59
##  Multi Race         : 26
##  American Indian    :  4
##  (Other)            :  5
```

Visualize numerical variables with ggplot

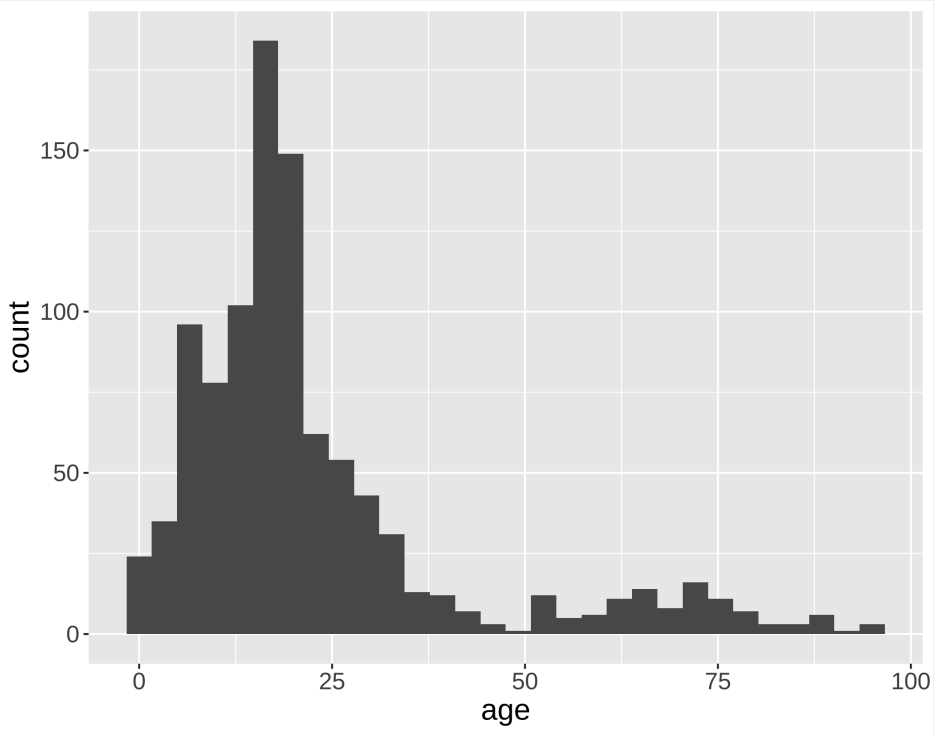
(V&H 1.4.4)

Allison Horst

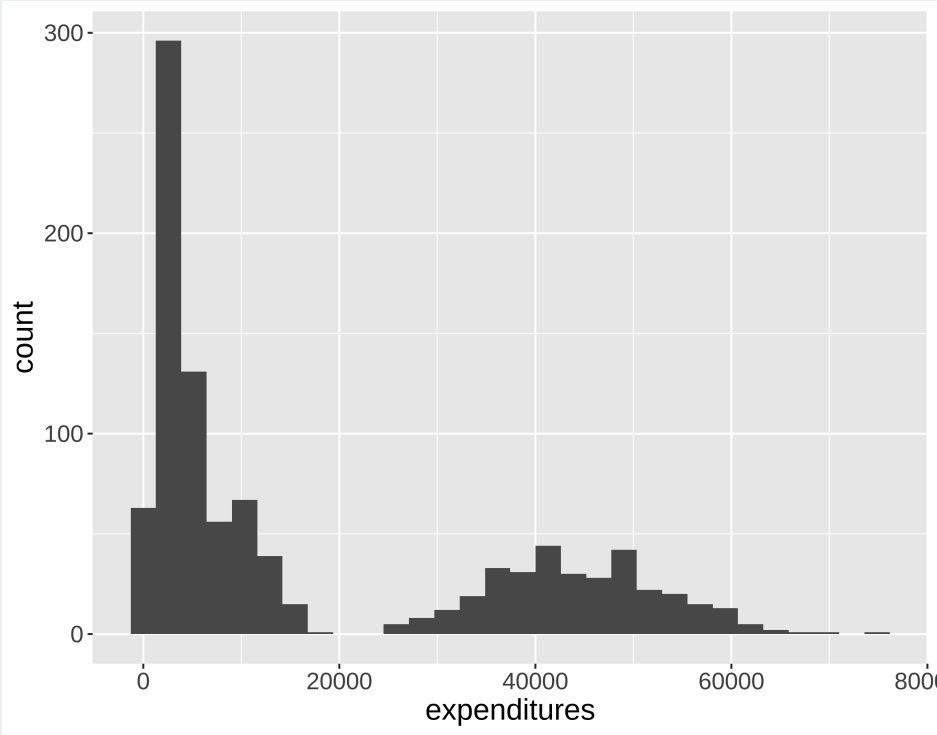
Histograms

What is being measured on the vertical axes?

```
ggplot(data = dds.discr,
       aes(x = age)) +
  geom_histogram()
```



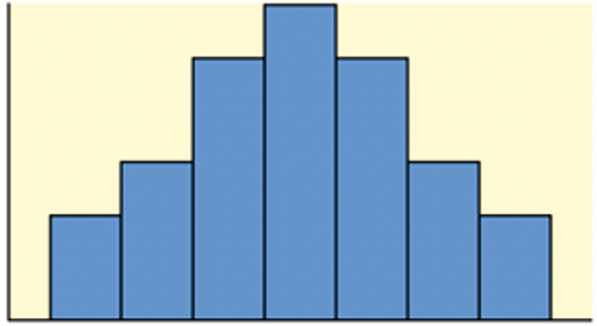

```
ggplot(data = dds.discr,
       aes(x = expenditures)) +
  geom_histogram()
```



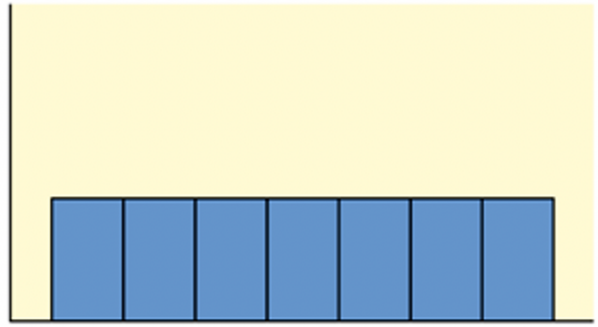
Distribution shapes

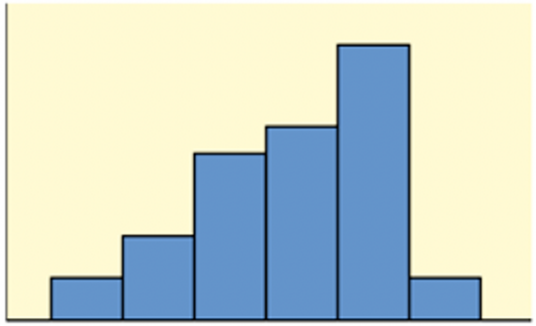
Common distribution shapes

symmetric

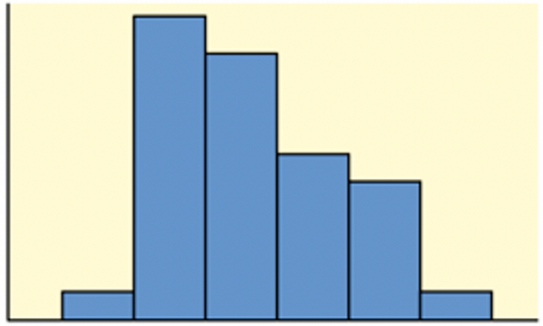


uniform

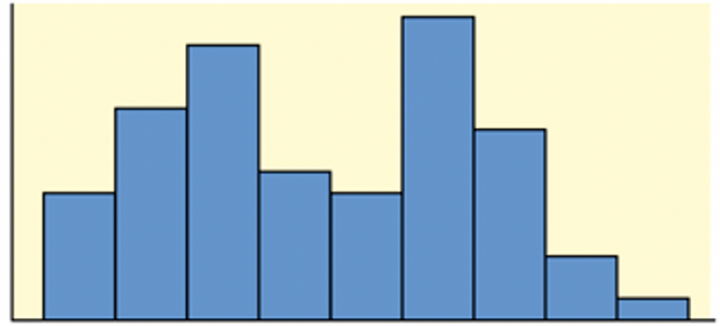




skewed left (negative)



skewed right (positive)

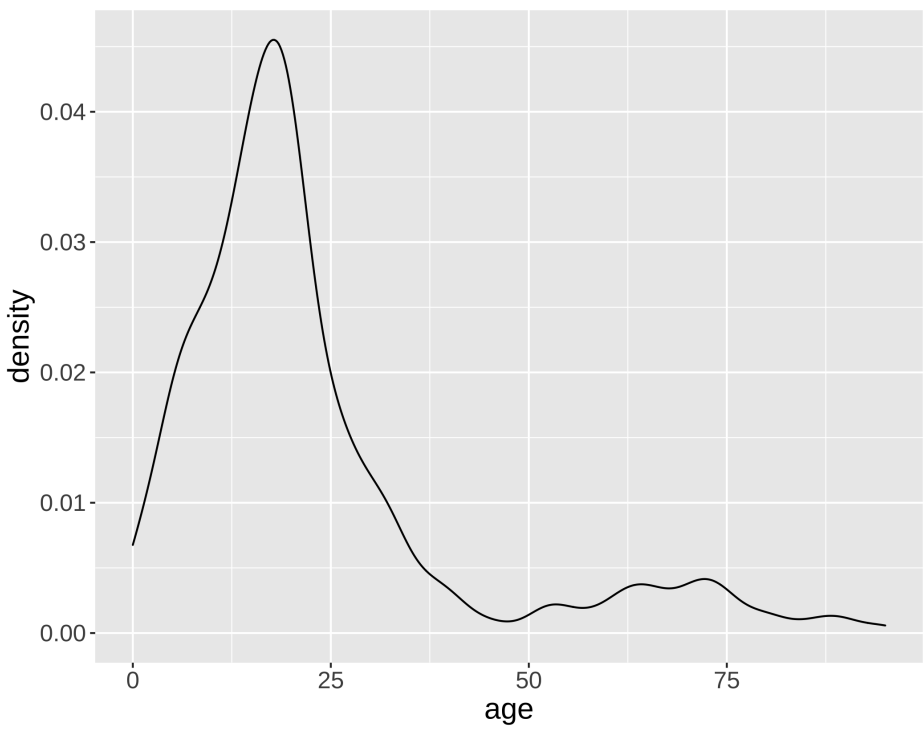


bimodal

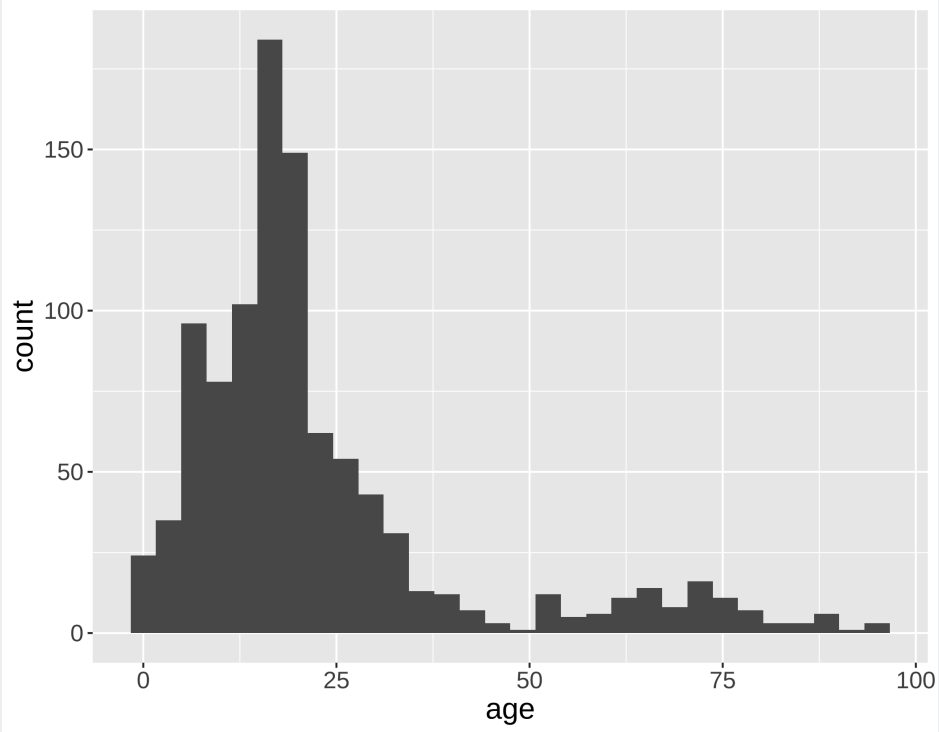
Density plots

What is being measured on the vertical axes?

```
ggplot(data = dds.discr,
       aes(x = age)) +
  geom_density()
```

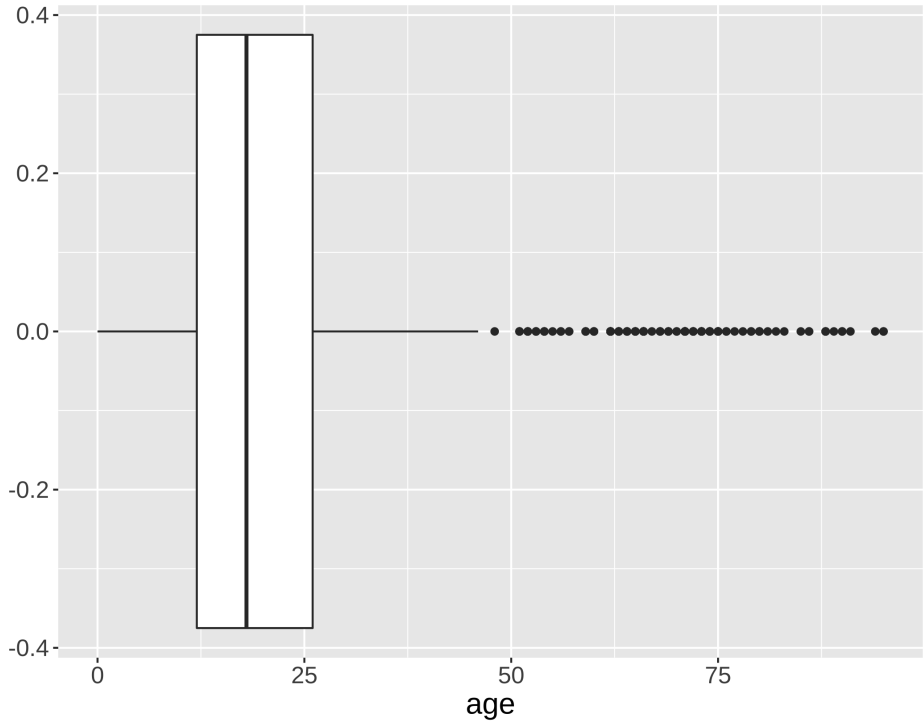



```
ggplot(data = dds.discr,
       aes(x = age)) +
  geom_histogram()
```

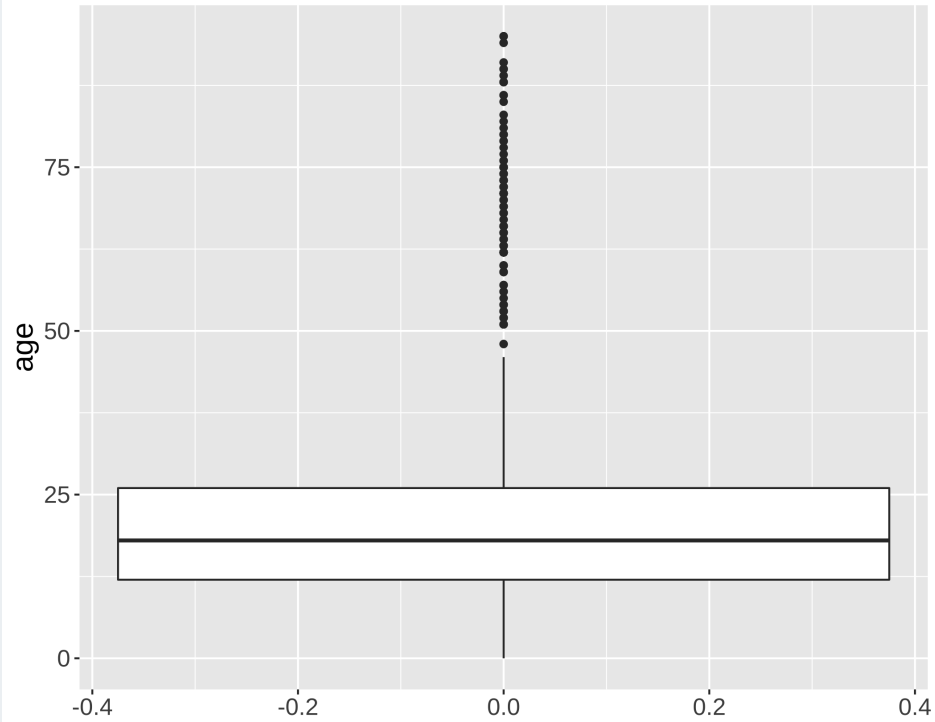


Boxplots

```
ggplot(data = dds.discr,
       aes(x = age)) +
  geom_boxplot()
```

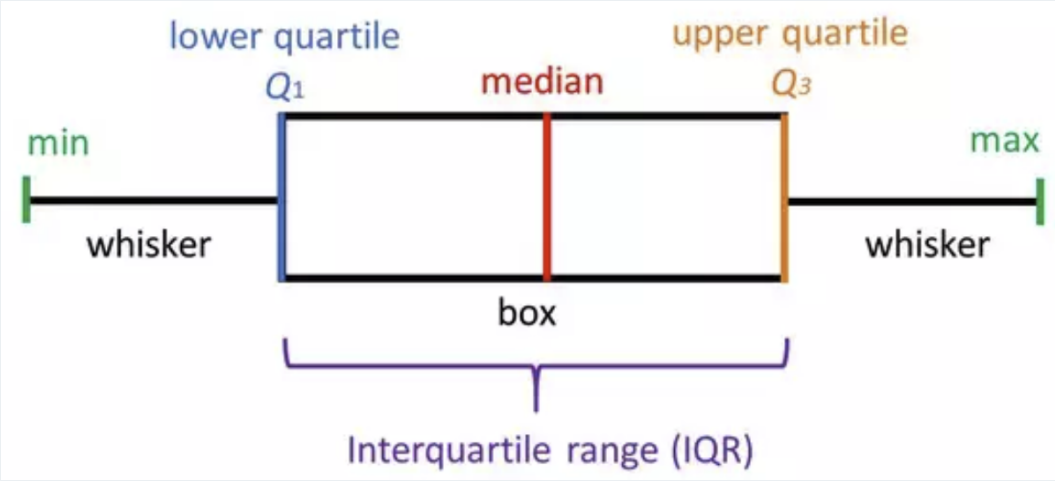



```
ggplot(data = dds.discr,
       aes(y = age)) +
  geom_boxplot()
```



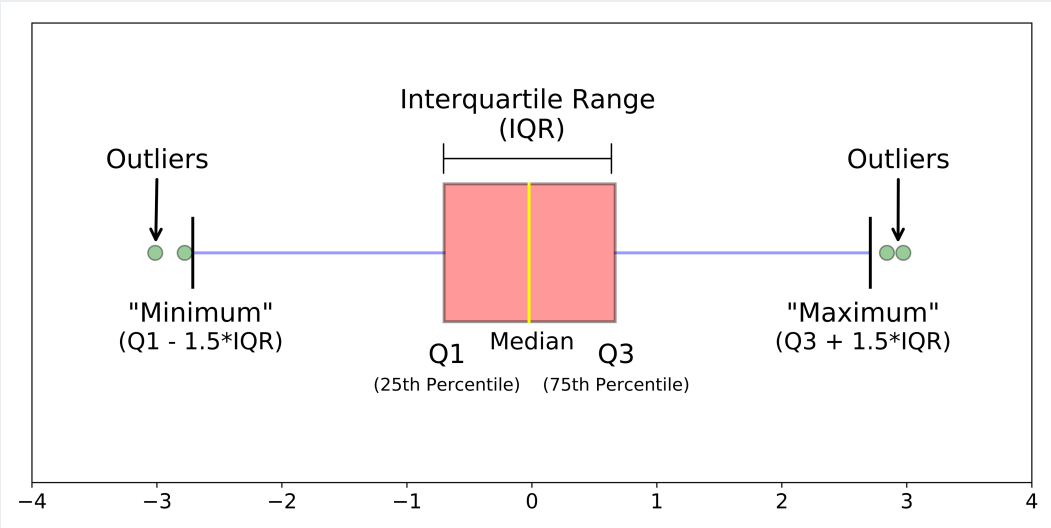
Boxplots: 5 number summary visualization

No outliers:



https://www.simplypsychology.org/boxplots.html

With outliers:



https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51

Visualizing relationships between numerical and categorical variables

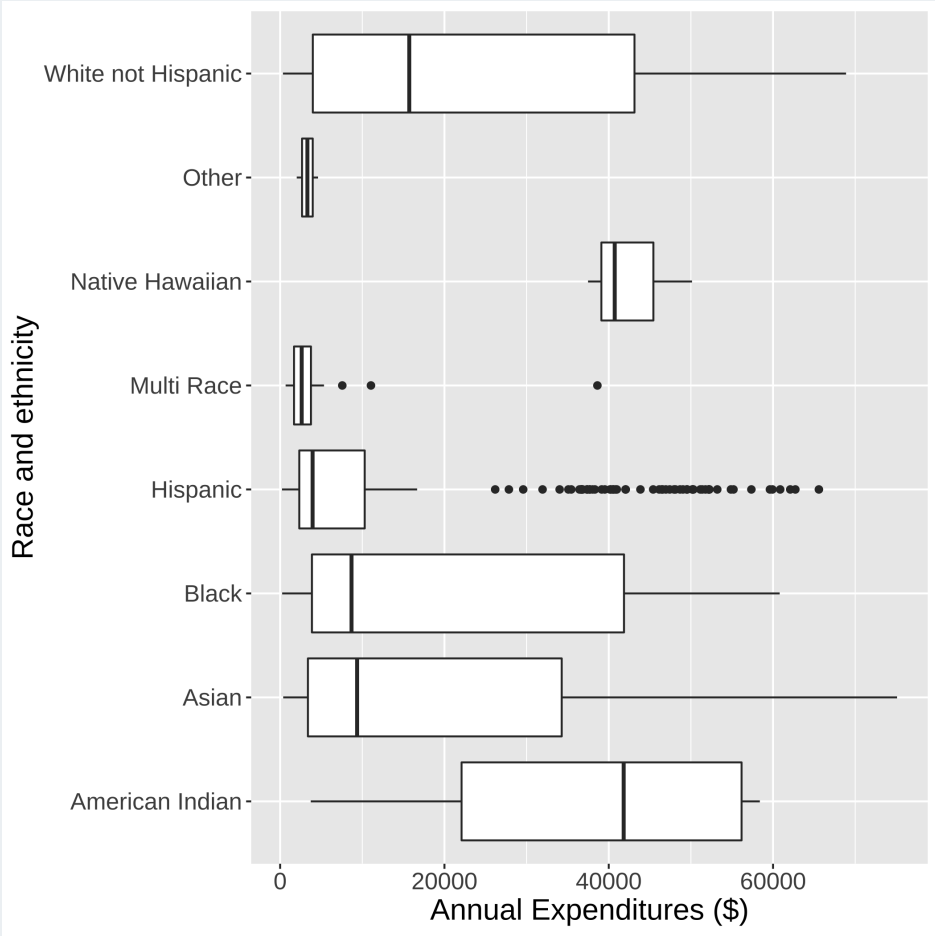
(V&H 1.6.3)

Side-by-side boxplots

```
ggplot(data = dds.discr,
       aes(x = expenditures,
           y = ethnicity)) +
  geom_boxplot() +
  labs(x = "Annual Expenditures ($)",
       y = "Race and ethnicity")
```

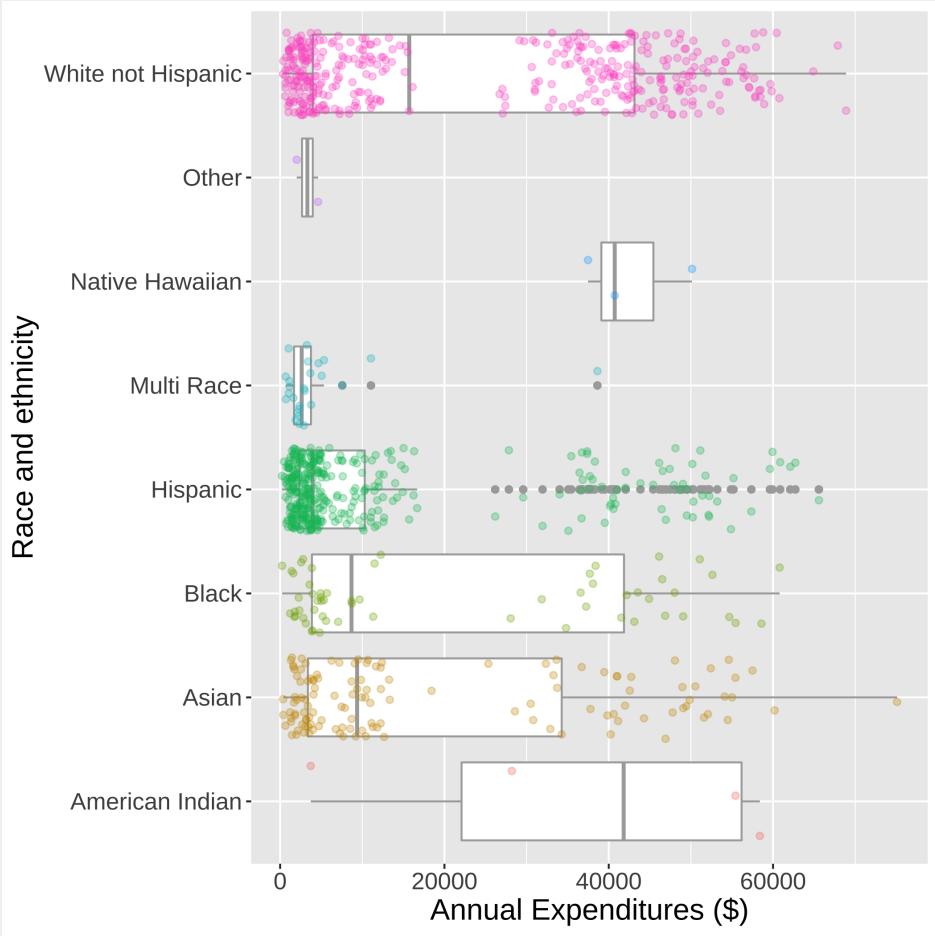
Can you determine the following using boxplots?

- distribution shape
- sample size



Side-by-side boxplots with data points

```
ggplot(data = dds.discr,
       aes(x = expenditures,
           y = ethnicity)) +
  geom_boxplot(color="darkgrey") +
  labs(x = "Annual Expenditures ($)",
       y = "Race and ethnicity") +
  geom_jitter(
    aes(color = ethnicity),
    alpha = 0.3,
    show.legend = FALSE,
    position = position_jitter(
      height = 0.4))
```

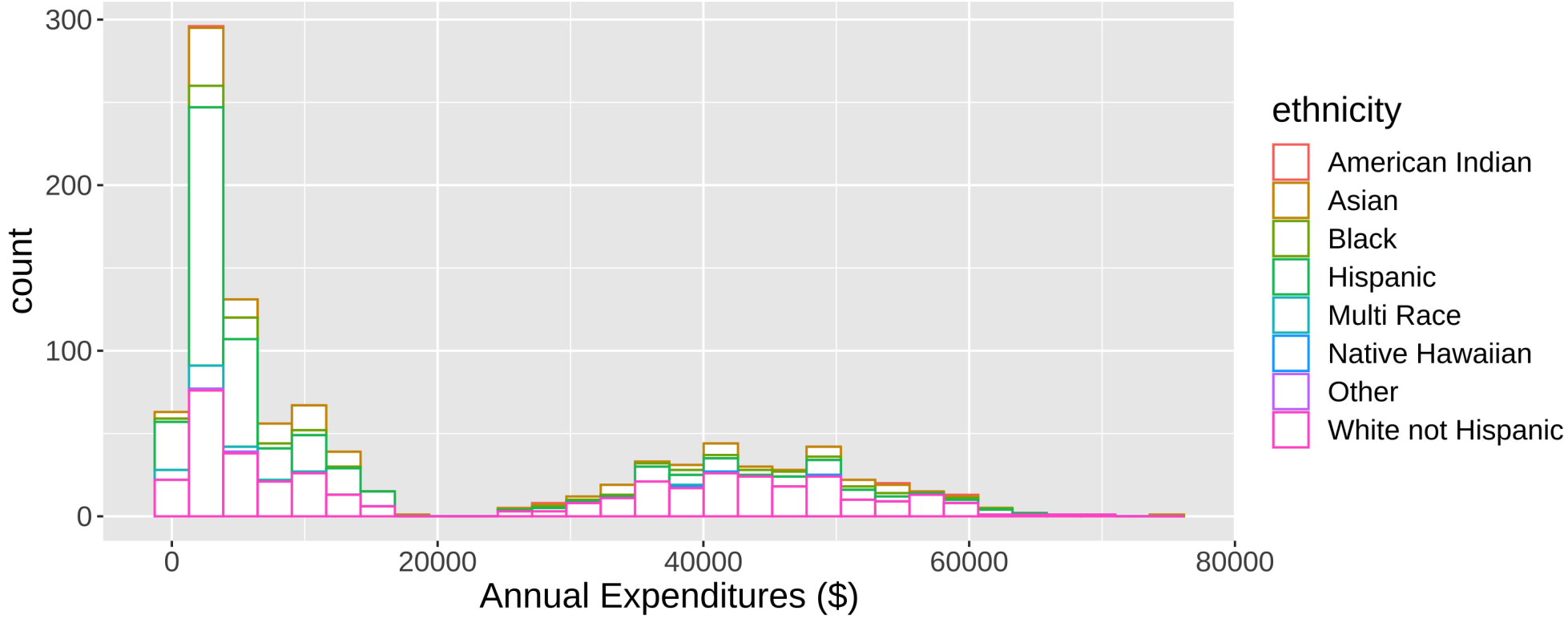


Can you determine the following using boxplots?

- distribution shape
- sample size

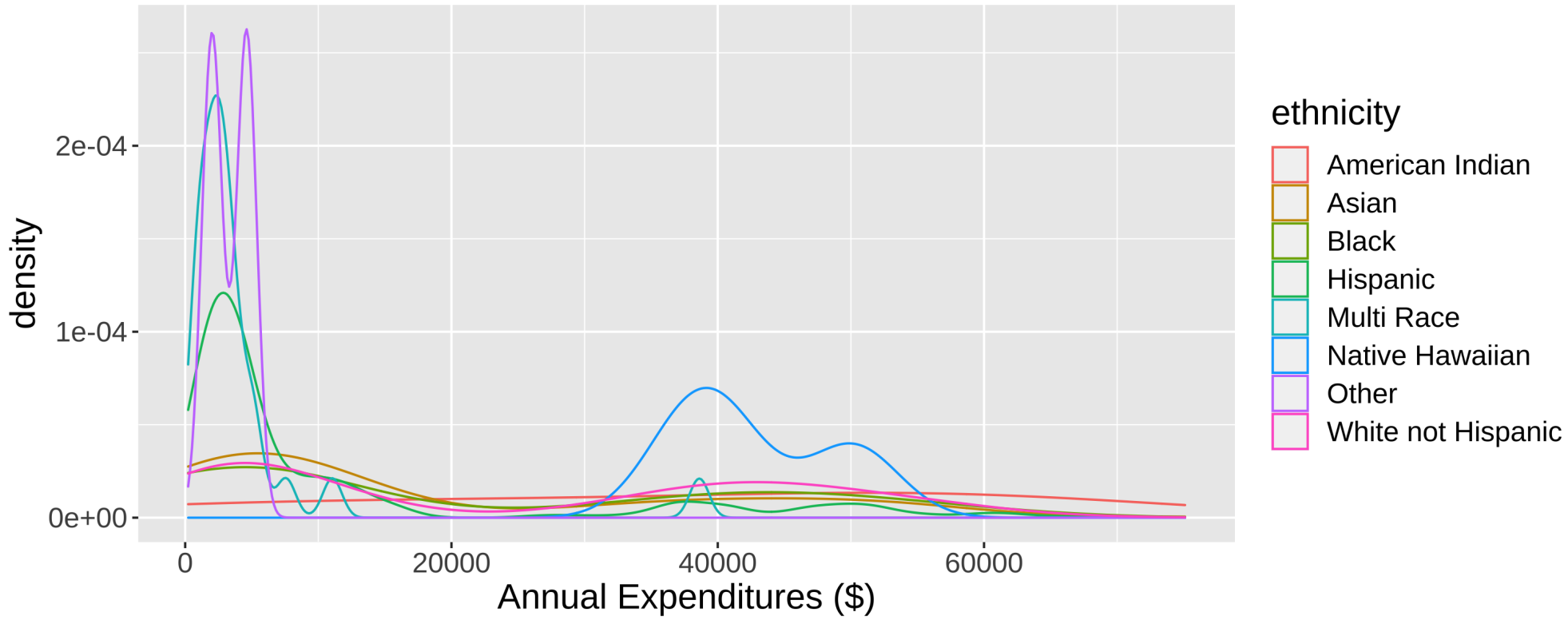
Histograms by group - not pretty

```
ggplot(data = dds.discr,
       aes(x = expenditures,
           color = ethnicity)) +
  geom_histogram(fill = "white") +
  labs(x = "Annual Expenditures ($)")
```



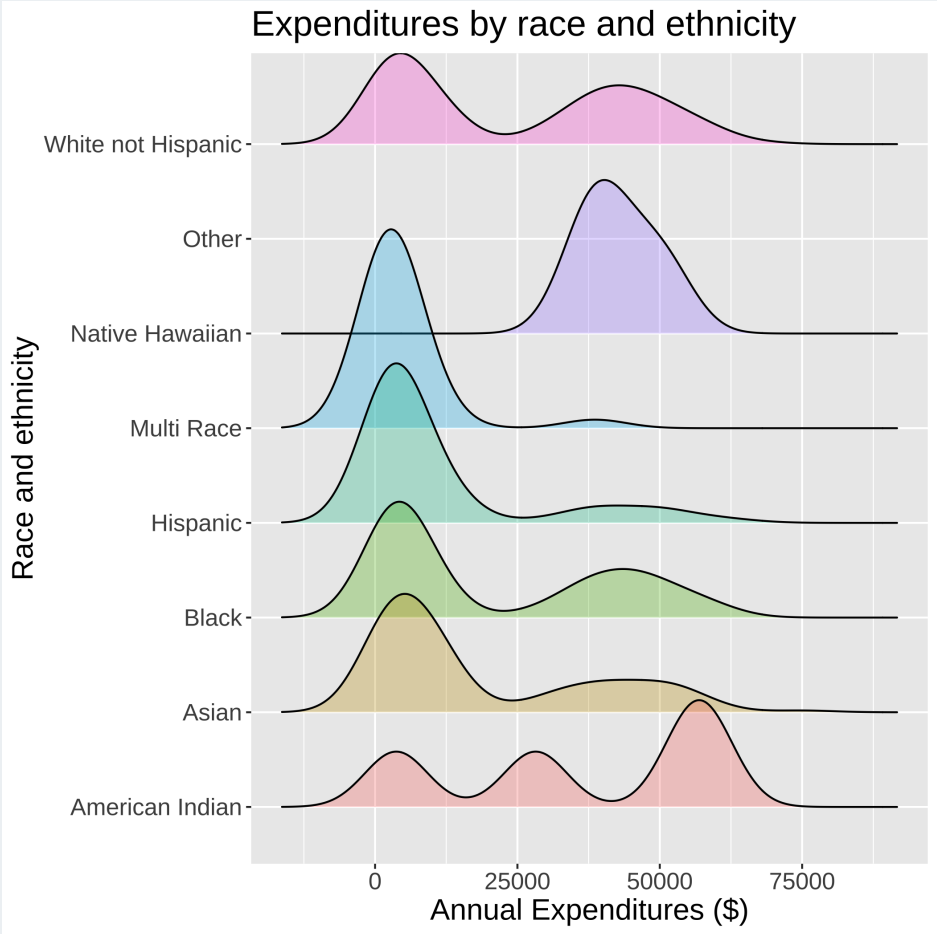
Density plots by group

```
ggplot(data = dds.discr,
       aes(x = expenditures,
           color = ethnicity)) +
  geom_density() +
  labs(x = "Annual Expenditures ($)")
```



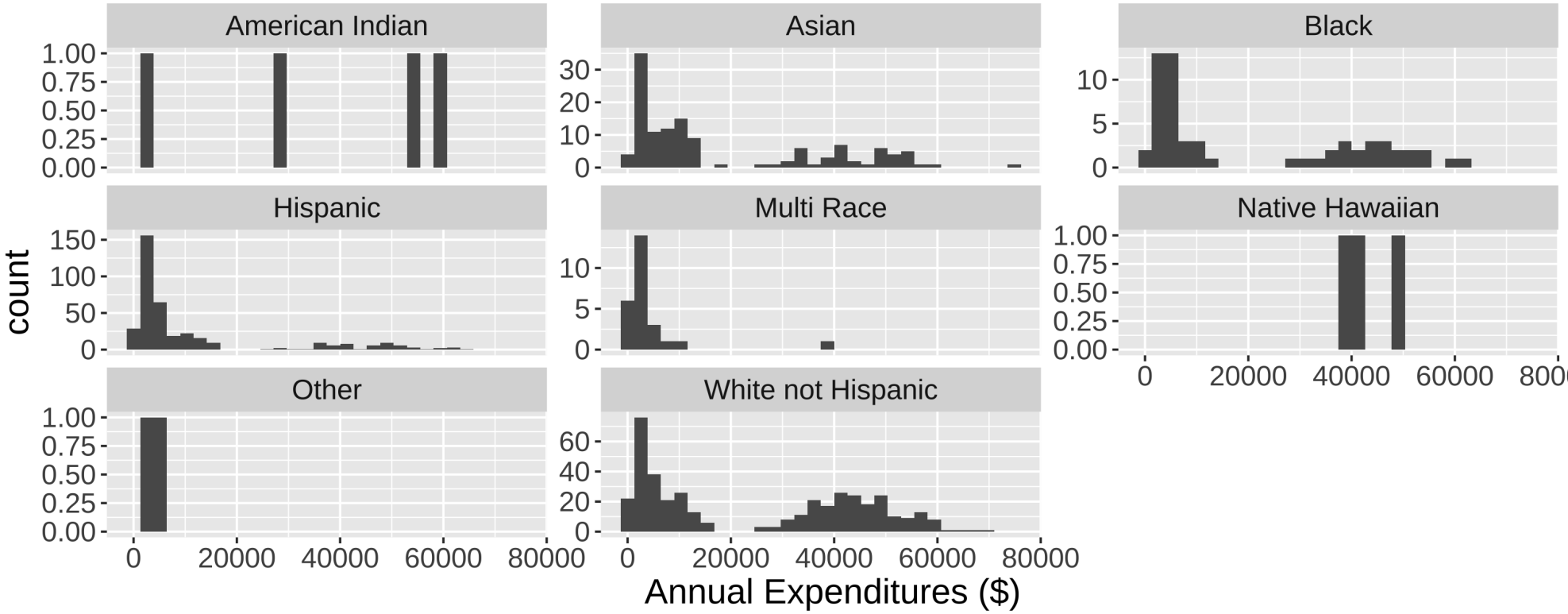
Ridgeline plot

```
library(ggridges)
ggplot(data = dds.discr,
       aes(x = expenditures,
           y = ethnicity,
           fill = ethnicity)
       ) +
  geom_density_ridges(
    alpha = 0.3,
    show.legend = FALSE) +
  labs(x = "Annual Expenditures ($)",
       y = "Race and ethnicity",
       title =
"Expenditures by race and ethnicity")
```



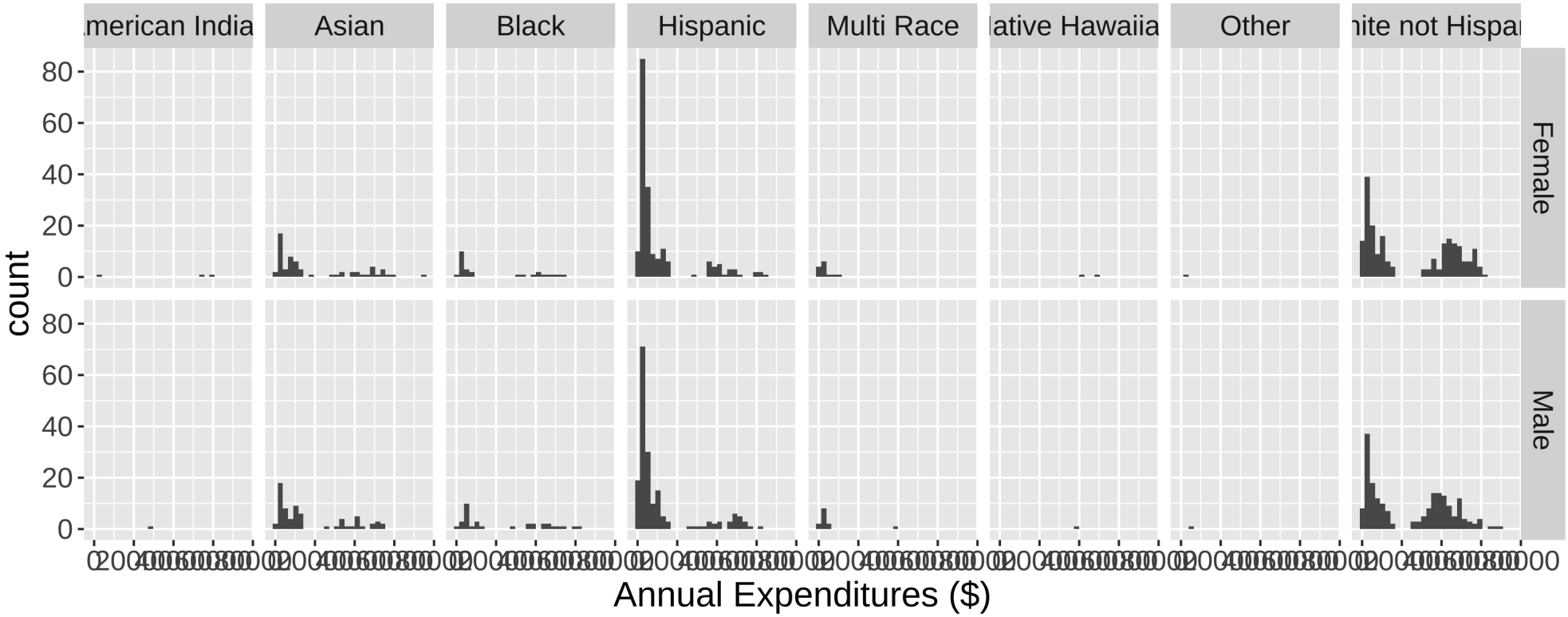
Histograms by group - using facets (facet_wrap)

```
ggplot(data = dds.discr,
       aes(x = expenditures)) +
  geom_histogram() +
  facet_wrap(vars(ethnicity), ncol = 3, scales = "free_y") +
  labs(x = "Annual Expenditures ($)")
```



Histograms by 2 groups - using facets (facet_grid)

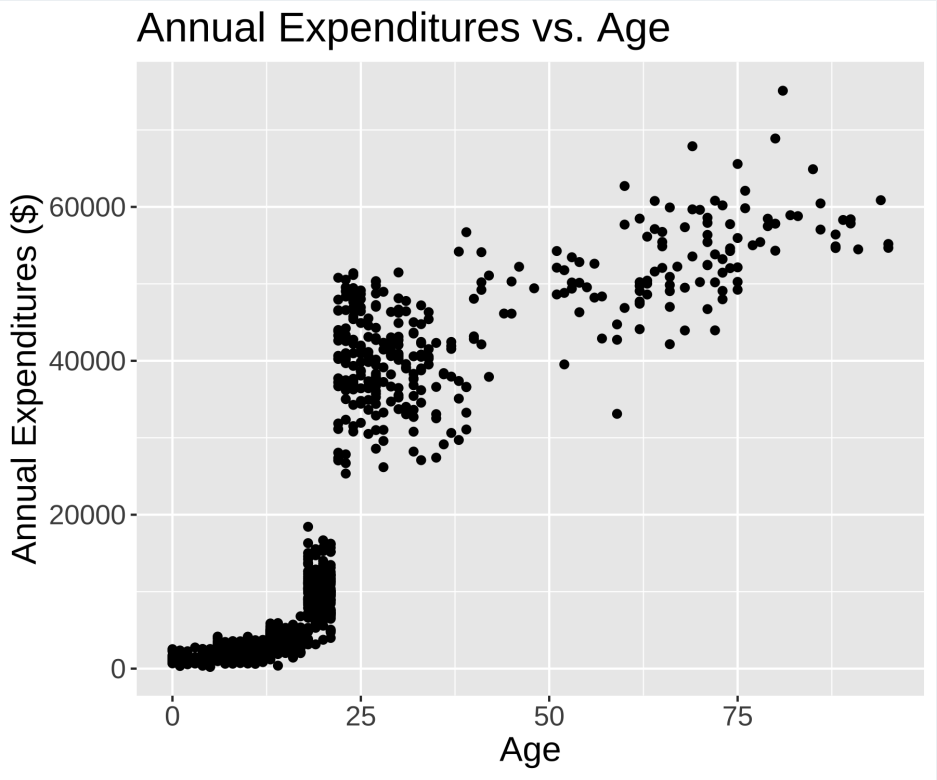
```
ggplot(data = dds.discr,
       aes(x = expenditures)) +
  geom_histogram() +
  facet_grid(rows = vars(gender), cols = vars(ethnicity)) +
  labs(x = "Annual Expenditures ($)")
```



Relationships between two numerical variables (V&H 1.6.1)

Scatterplots

```
ggplot(data = dds.discr,
       aes(x = age,
           y = expenditures)) +
  geom_point() +
  labs(x = "Age",
       y = "Annual Expenditures ($)",
       title = "Annual Expenditures vs
```



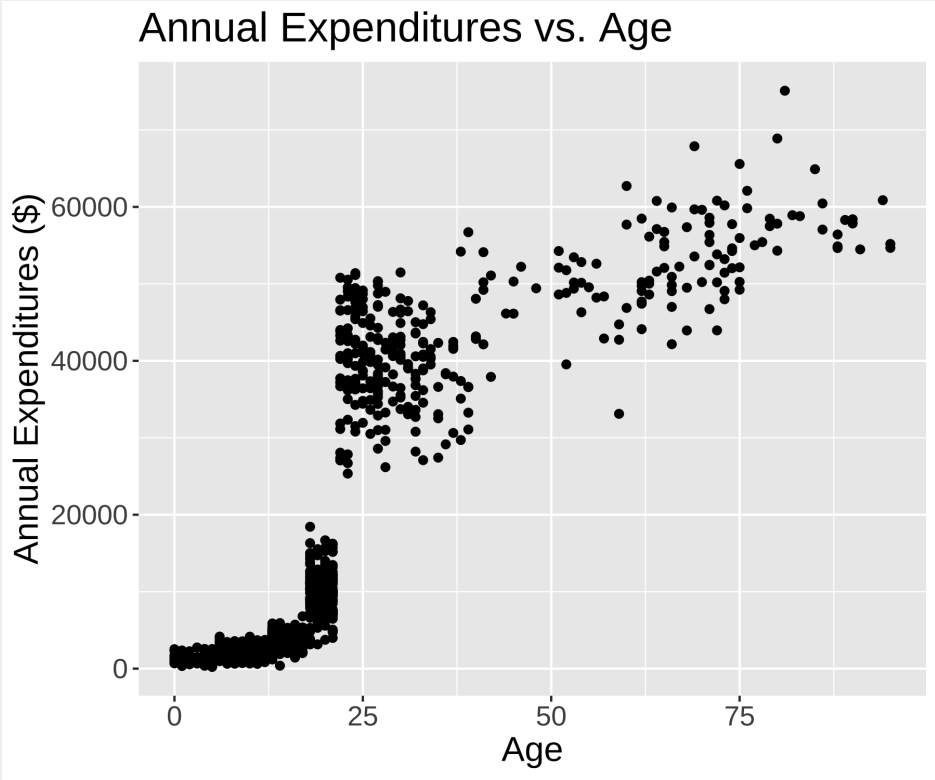
Response vs. explanatory variables (V&H Section 1.2.3)

- A **response variable** measures the outcome of interest in a study
- A study will typically examine whether the values of a response variable differ as values of an **explanatory variable** change

Describing associations between 2 numerical variables

Two variables x and y are

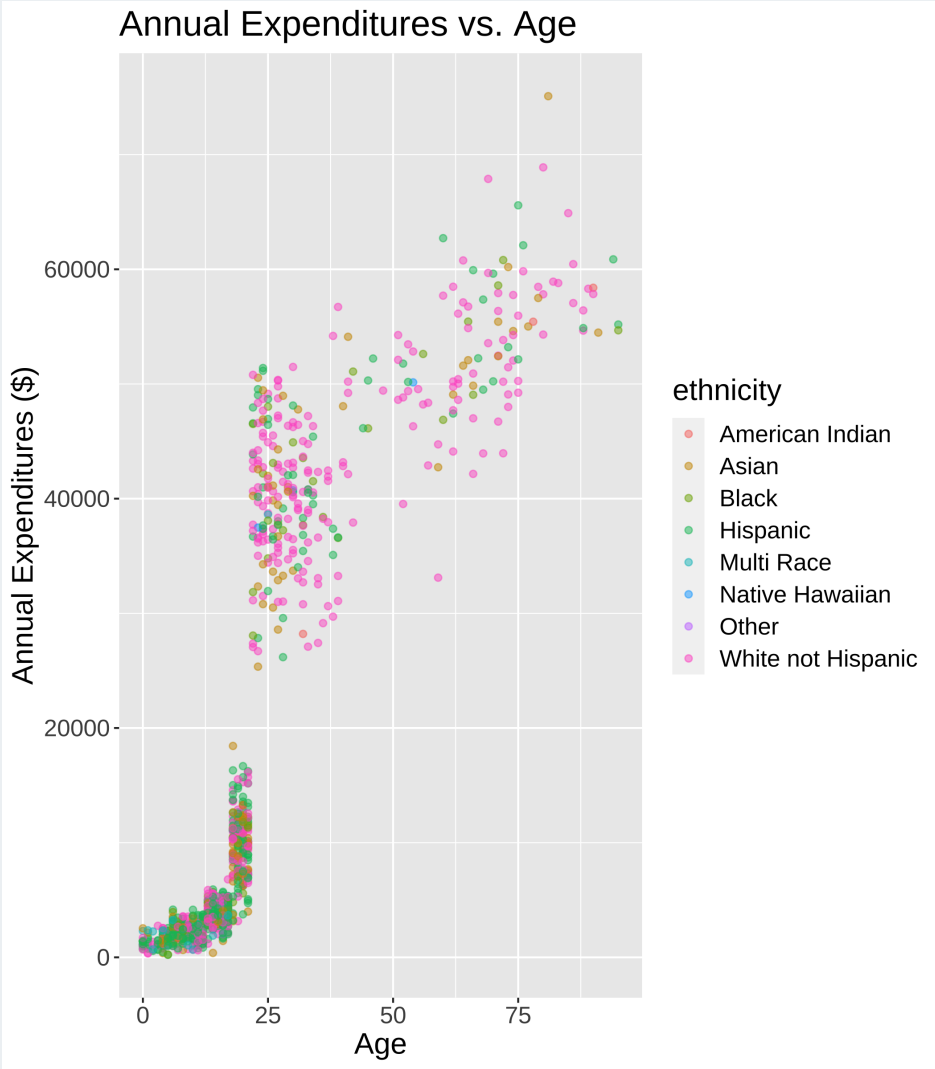
- **positively associated** if y increases as x increases.
- **negatively associated** if y decreases as x increases.
- If there is no association between the variables, then we say they are **uncorrelated** or **independent**.
- The term "association" is a very general term.
  - Can be used for numerical or categorical variables
  - Not specifically referring to linear associations



Sactterplots with color-coded dots

```
ggplot(data = dds.discr,
       aes(x = age,
           y = expenditures,
           color = ethnicity)) +
  geom_point(alpha = .5) +
  labs(x = "Age",
       y = "Annual Expenditures ($)",
       title = "Annual Expenditures vs
```

Describe the association between the variables



Categorical data (V&H 1.5)

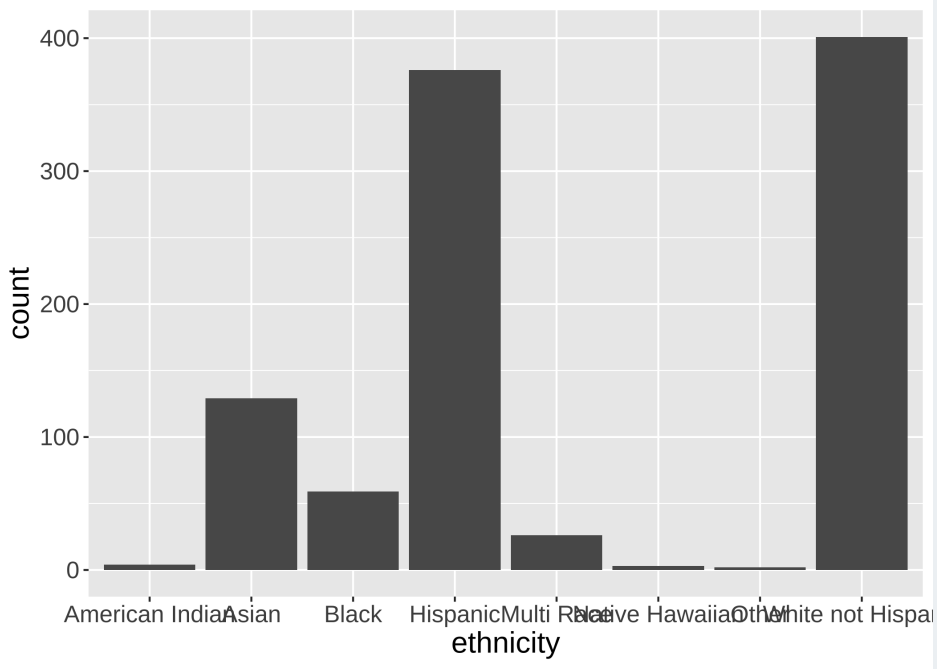
and

Relationships between two categorical variables (V&H 1.6.2)

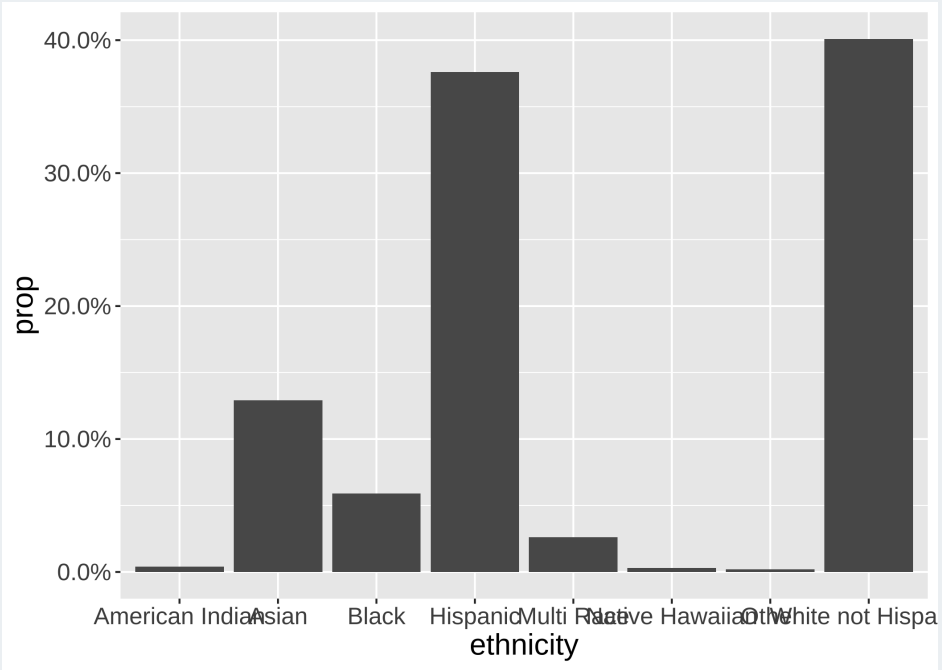
Barplots

Counts (below) vs. percentages (right)

```
ggplot(data = dds.discr,
       aes(x = ethnicity)) +
  geom_bar()
```

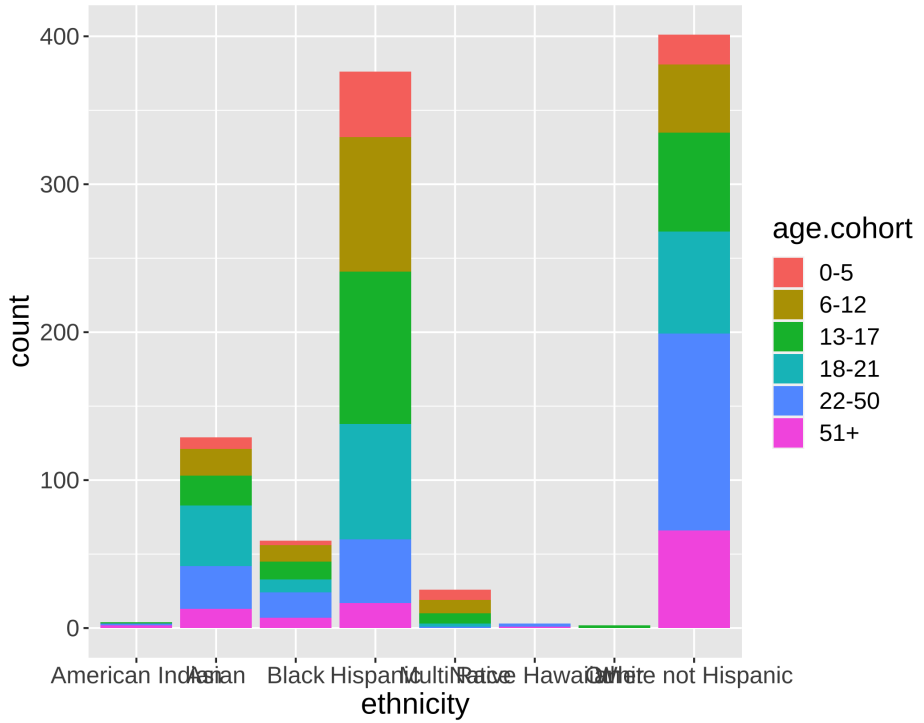



```
ggplot(data = dds.discr,
       aes(x = ethnicity)) +
  geom_bar(aes(y = stat(prop),
               group = 1)) +
  scale_y_continuous(labels =
      scales::percent_format())
```

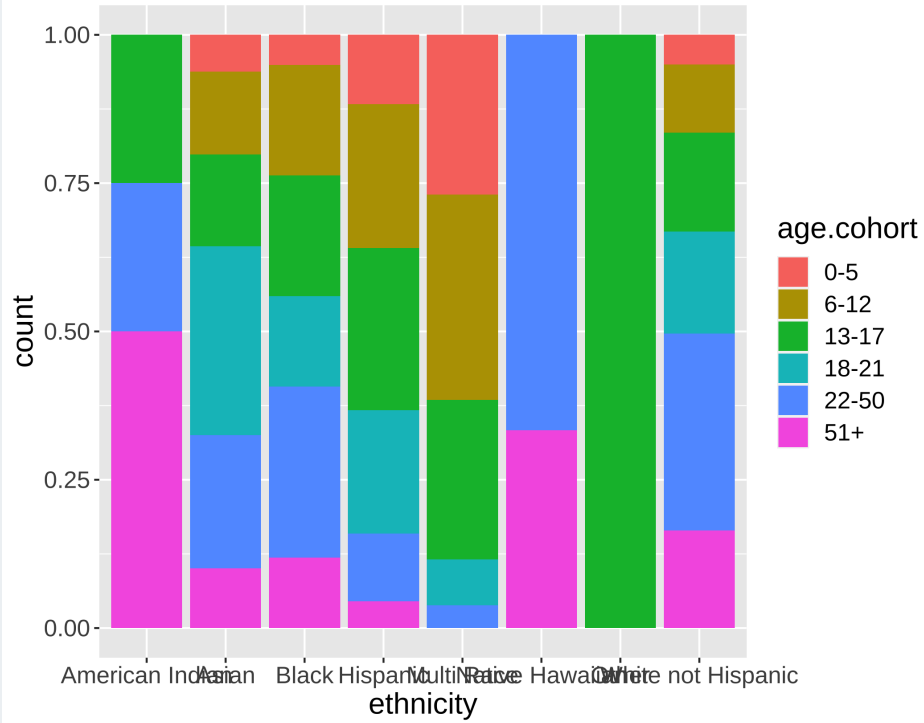


Barplots with 2 variables: segmented bar plots

```
ggplot(data = dds.discr,
       aes(x = ethnicity,
           fill = age.cohort)) +
  geom_bar()
```



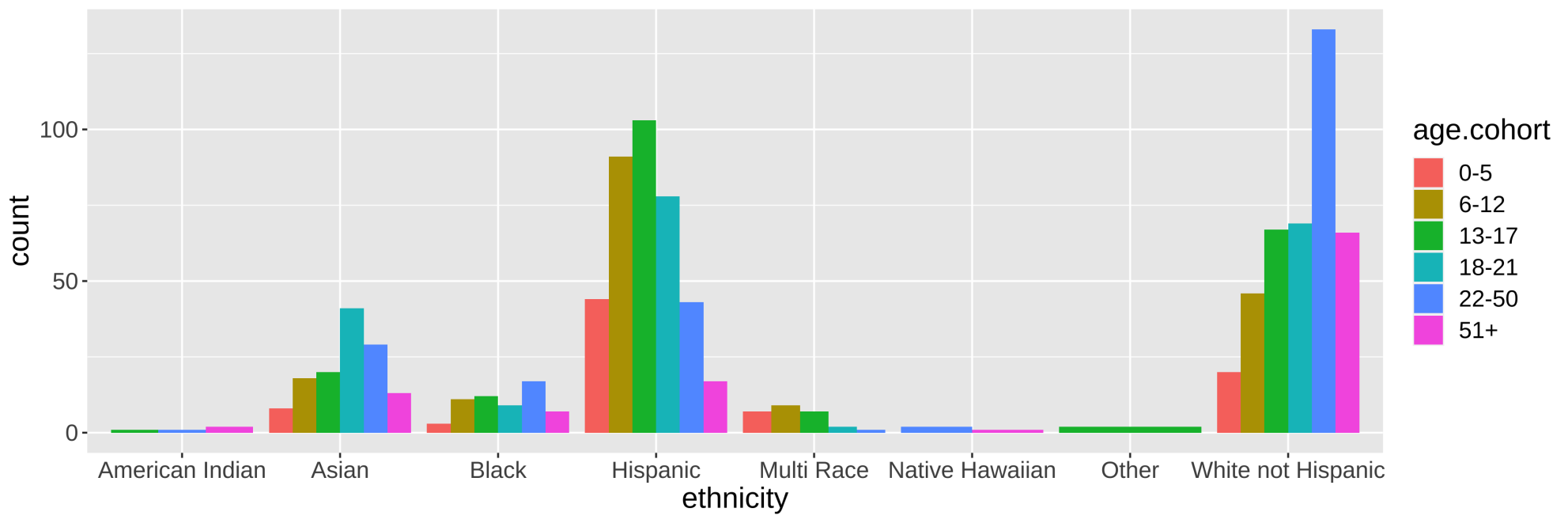

```
ggplot(data = dds.discr,
       aes(x = ethnicity,
           fill = age.cohort)) +
  geom_bar(position = "fill")
```



Barplots with 2 variables: side-by-side bar plots

Side-by-side bar plot

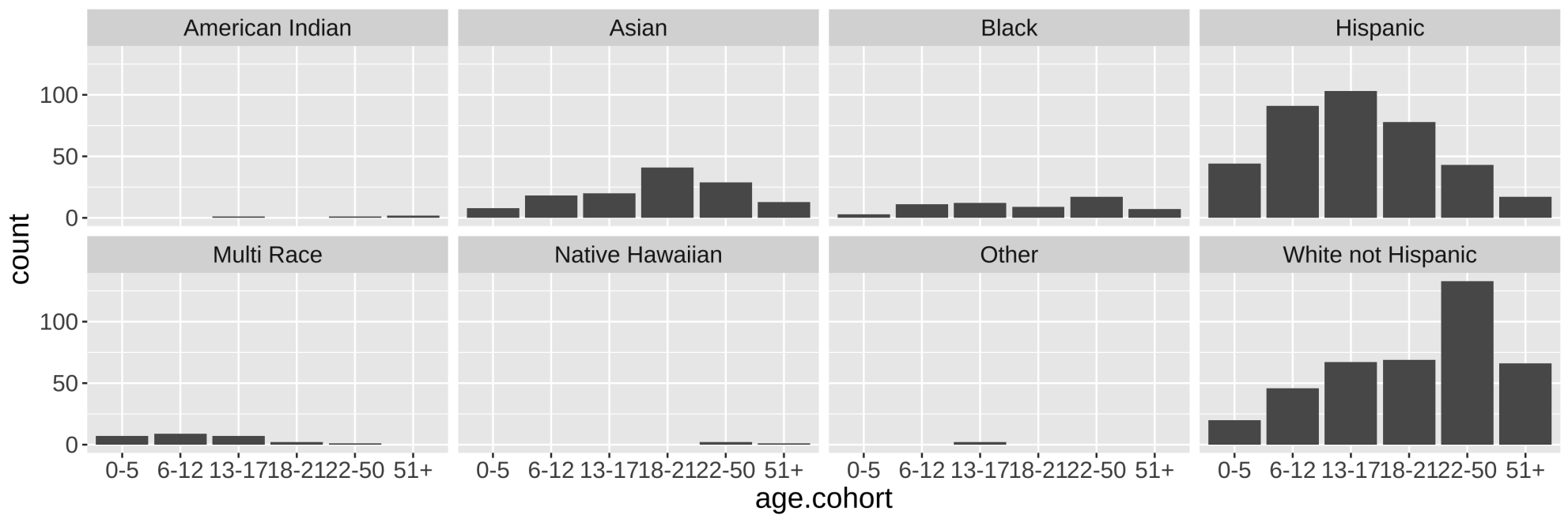
```
ggplot(data = dds.discr,
       aes(x = ethnicity,
           fill = age.cohort)) +
  geom_bar(position = "dodge")
```



Barplots with 2 variables: using facets

Bar plots of race/ethnicity faceted by age groups

```
ggplot(data = dds.discr,
       aes(x = age.cohort)) +
  geom_bar(position = "dodge") +
  facet_wrap(vars(ethnicity), ncol = 4)
```

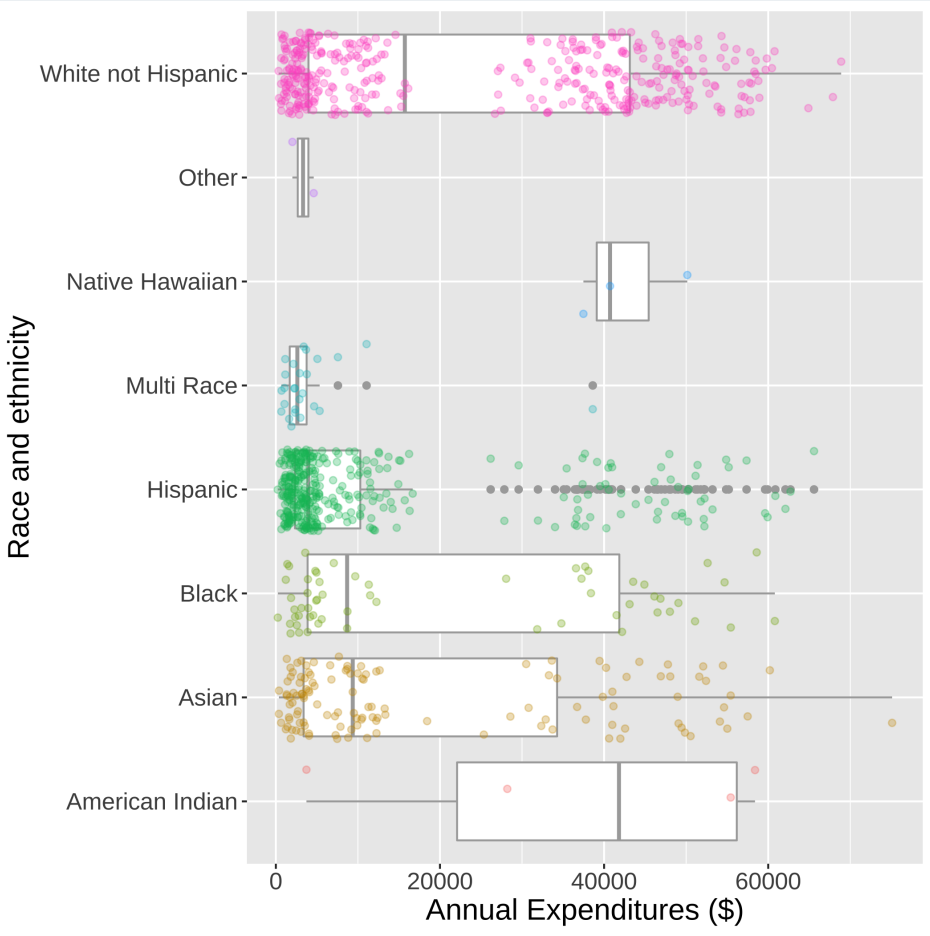


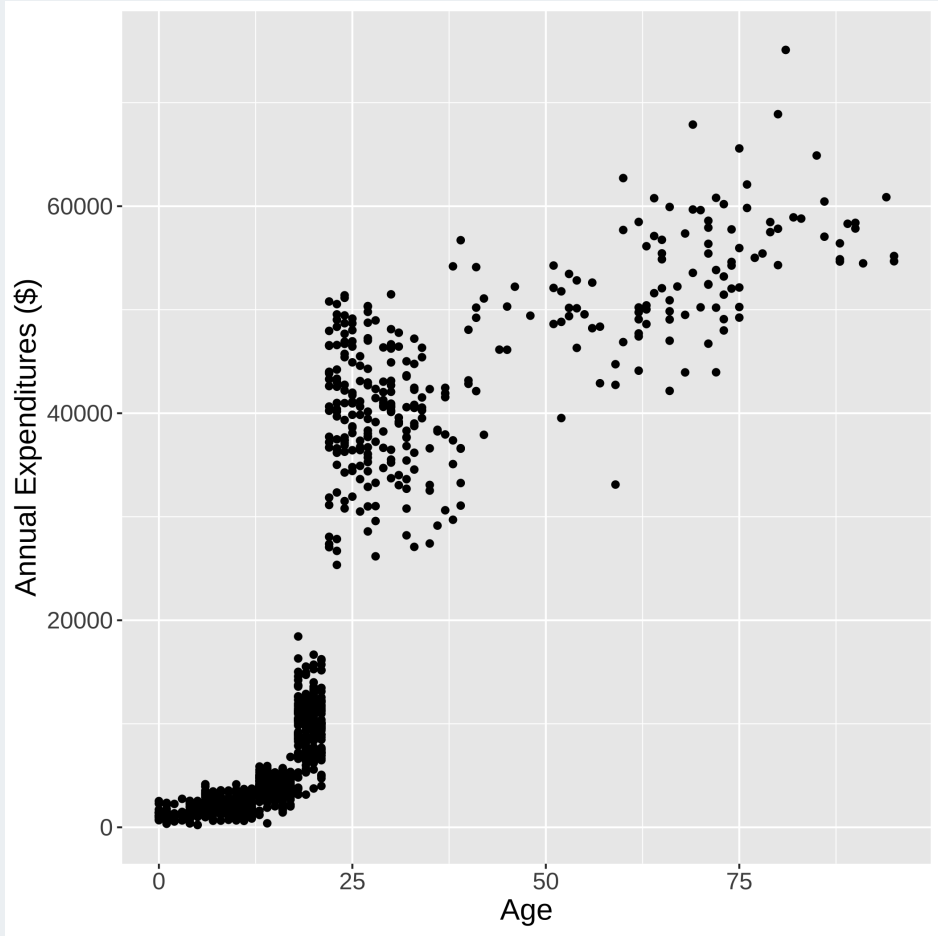
Back to research question

Case study: discrimination in developmental disability support (V&H 1.7.1)

- **Previous research**
  - Researchers examined DDS expenditures for developmentally disabled residents by ethnicity
  - Found that the mean annual expenditures on Hispanics was less than that on White non-Hispanics.
- **Result**: an allegation of ethnic discrimination was brought against the California DDS.
- **Question: Are the data sufficient evidence of ethnic discrimination?**

Recall previous data viz





Visualize in more detail: ethnicity, age, and expenditures (code on next slide)

Code for Visualize in more detail: ethnicity, age, and expenditures

Plot on previous slide

```
dds.discr_Hips_WhnH <- dds.discr %>%
  filter(ethnicity == "White not Hispanic" | ethnicity == "Hispanic" ) %>%
  droplevels()   # remove empty factor levels


ggplot(data = dds.discr_Hips_WhnH,
       aes(x = expenditures,
           y = age.cohort)) +
  geom_boxplot(color="darkgrey") +
  facet_grid(rows = "ethnicity") +
  labs(x = "Annual Expenditures ($)",
       y = "Race and ethnicity") +
  geom_jitter(
    aes(color = ethnicity),
    alpha = 0.3,
    show.legend = FALSE,
    position = position_jitter(
      height = 0.4))
```

ggplot examples

- Examples are from the OCTRI Biostatistics, Epidemiology, Research & Design (BERD) Workshop on **Data Visualization with R and ggplot2** I co-presented with Jessica Minnier on 2020/05/20.

- Download files from Sakai, or use these links:

  - html version: http://bit.ly/berd_ggplot (*takes a while to load, but has animation features*)
  - pdf version: http://bit.ly/berd_ggplot_pdf (*quick to load, but no animations*)

- Jump to slide 12 for the **Visual Table of Contents**

  - Gapminder data description on slide 10
  - We will go through slides 10-24

- *See https://github.com/jminnier/berd_r_courses for more workshop links, including recording.*

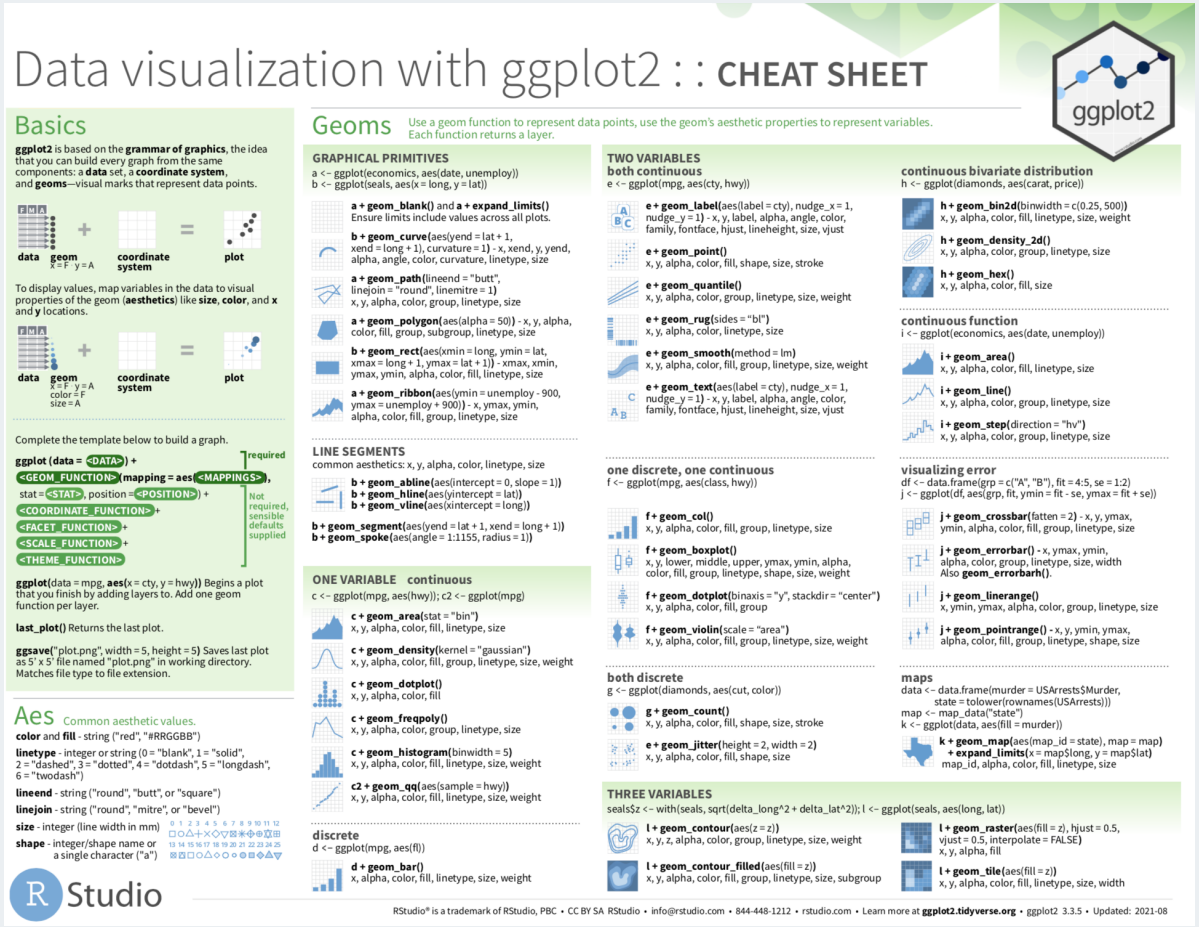
Practice ggplot with RStudio Primers

The Basics -> Visualization Basics:
https://rstudio.cloud/learn/primers/1.1

Visualize Data:
https://rstudio.cloud/learn/primers/3

(see all primers at
https://rstudio.cloud/learn/primers)

ggplot2 cheatsheet - download this!!



Data visualization with ggplot2 : : CHEAT SHEET

ggplot2

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.

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")

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

RStudio

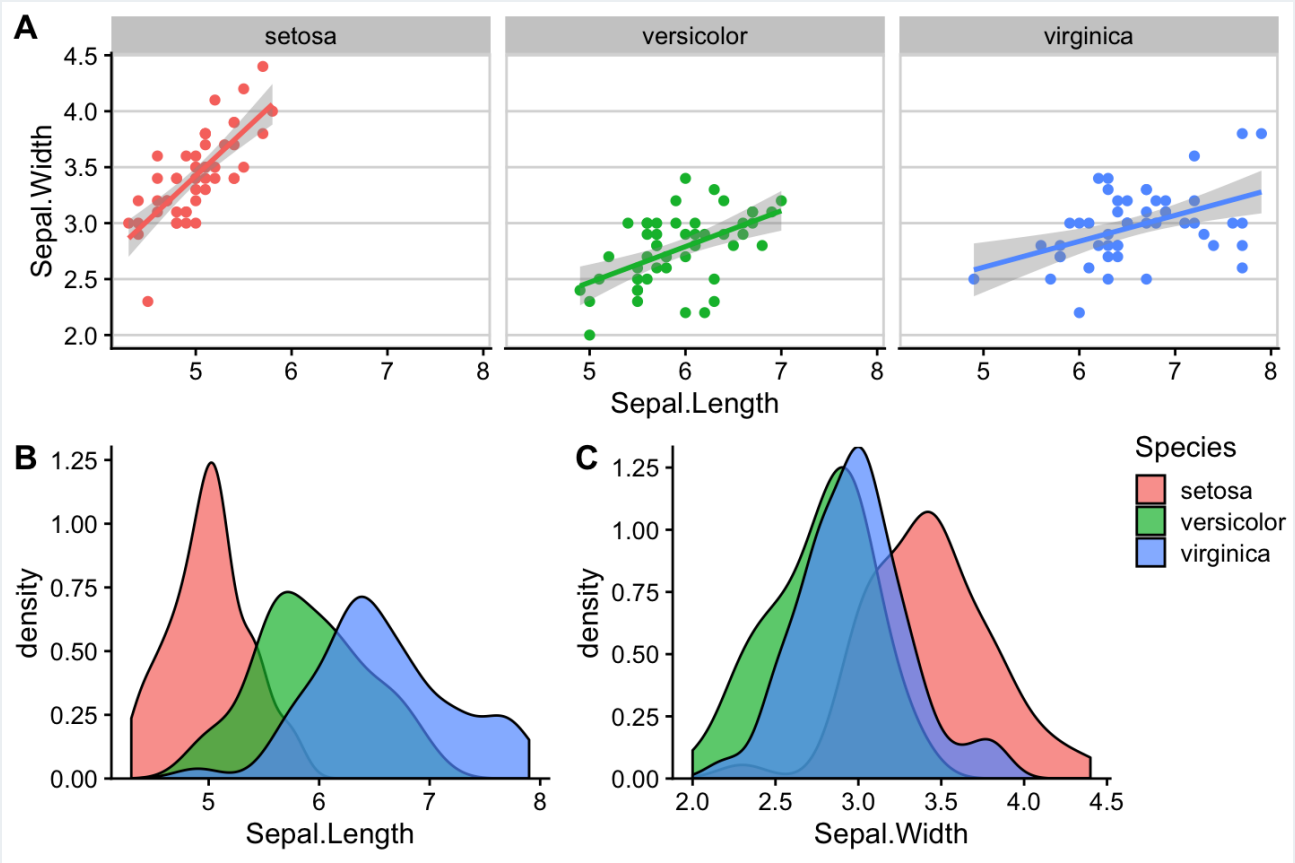
RStudio® is a trademark of RStudio, PBC • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at **ggplot2.tidyverse.org** • ggplot2 3.3.5 • Updated: 2021-08

https://raw.githubusercontent.com/rstudio/cheatsheets/main/data-visualization.pdf

See https://rstudio.com/resources/cheatsheets/ for many more cheatsheets!

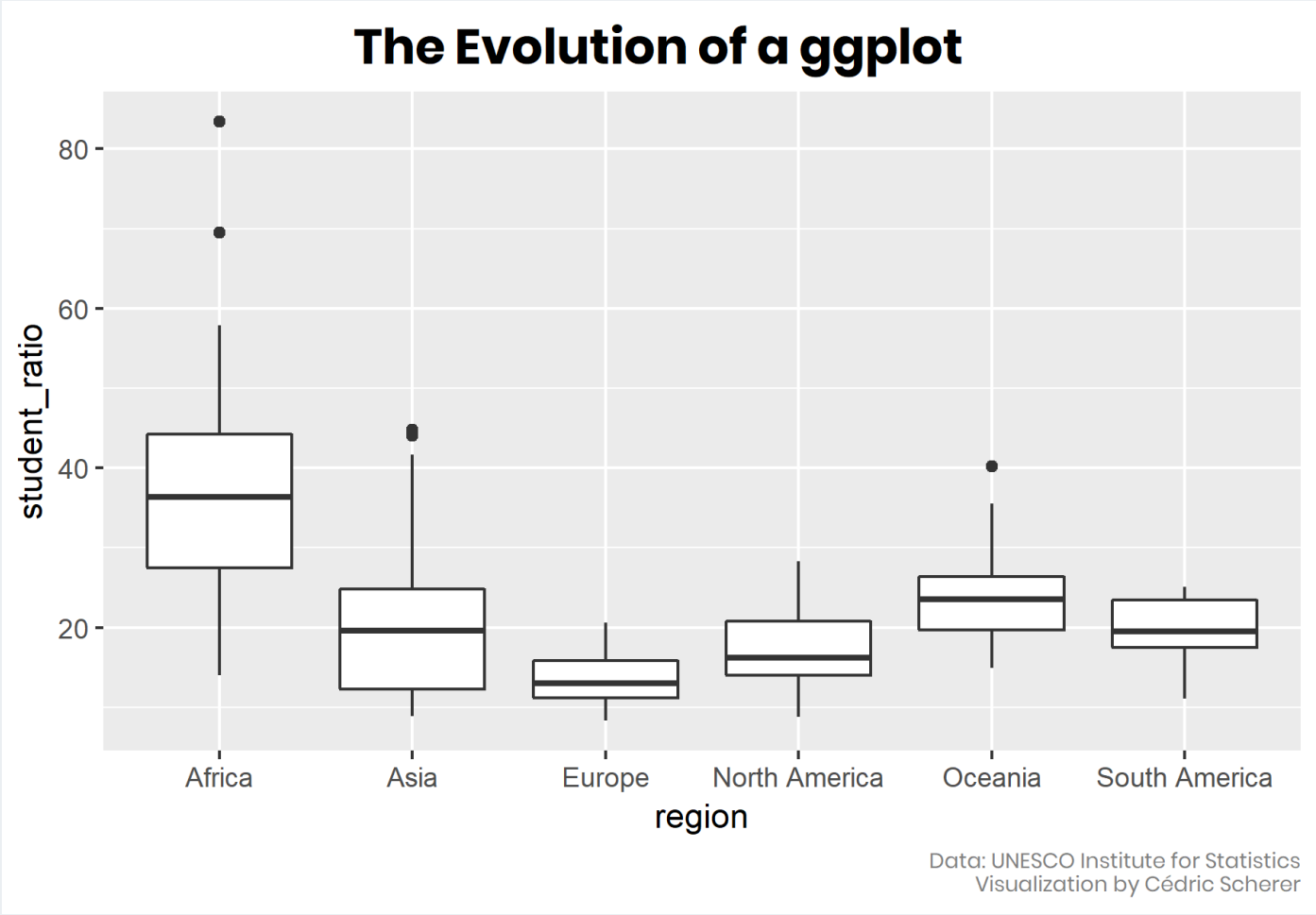
cowplot package for arranging plots

- Claus O. Wilke's cowplot package



Shared legend for all plots!

Evolution of a ggplot



Cédric Scherer's *The Evolution of a ggplot*

There's much, much more out there!

- ggplot cheatsheet
- ggplot2 package reference
- ggplot2: Elegant Graphics for Data Analysis by Hadley Wickham
- *Data Visualizaton* online textbook by Kieran Healy
- R Graphics Cookbook by Winston Chang
- *R for Data Science* online textbook by Hadley Wickham
- *Introduction to Data Science* online textbook by Rafael A. Irizarry

Example plots and extensions:

- R Graph Gallery
- ggplot2 extension gallery
- All Your Figure Are Belong To Us
- from Data to Viz - beautiful flowcharts to help you decide on a plot based on the variable type(s); check out their poster
- Top 50 ggplot2 Visualizations - The Master List (With Full R Code)

OHSU class:

- CS 631 Data Visualization

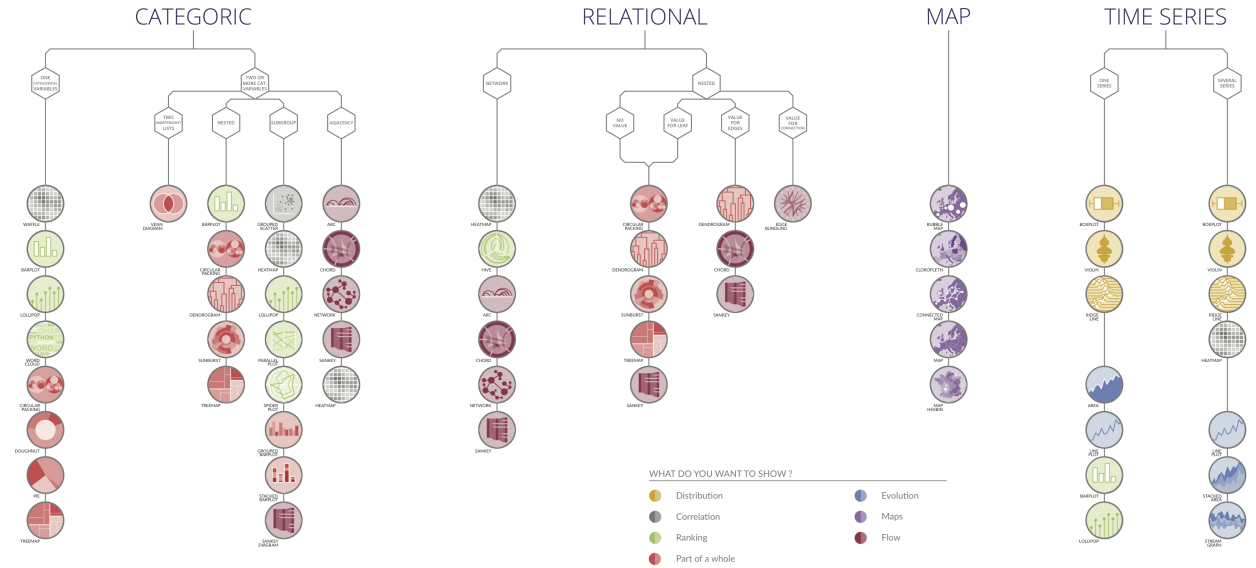
from Data to Viz

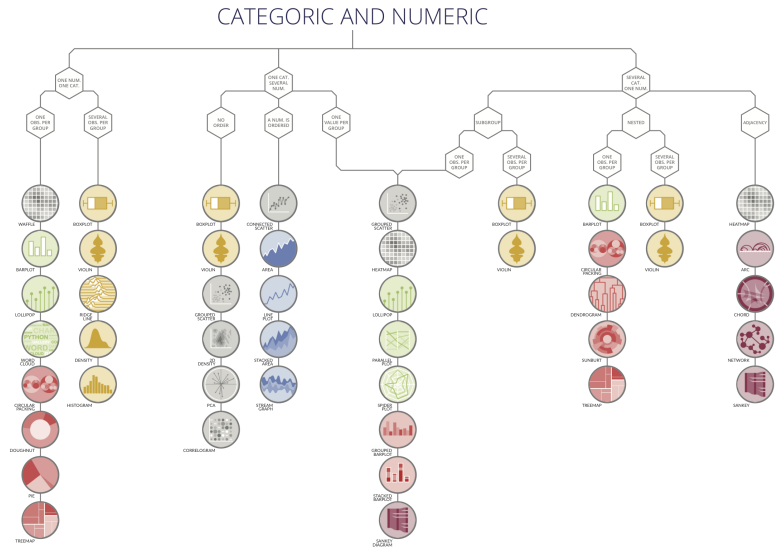
'From Data to Viz' is a classification of chart types based on input data format. It will help you find the perfect chart in three simple steps :

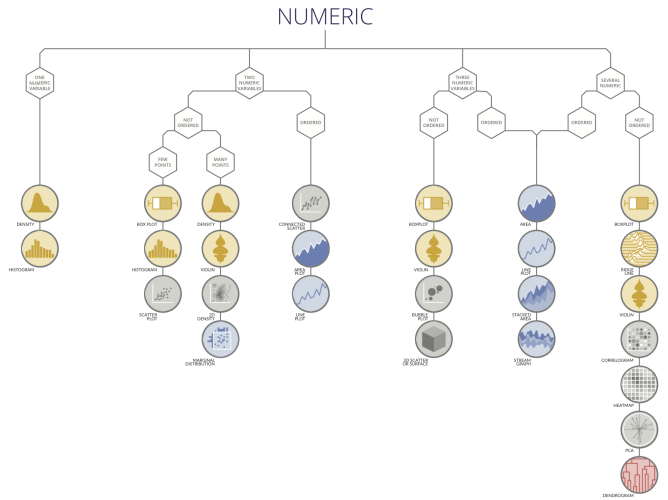
1. Identify what type of data you have.
2. Go to the corresponding decision tree and follow it down to a set of possible charts.
3. Choose the chart from the set that will suit your data and your needs best.

Dataviz is a world with endless possibilities and this project does not claim to be exhaustive. However it should provide you with a good starting point. For an interactive version and much more, visit:

data-to-viz.com







https://www.data-to-viz.com/poster.html