6: Data Exploration

Environmental Data Analytics | Kateri Salk

Spring 2019

LESSON OBJECTIVES

1. Set up a data analysis session in RStudio
2. Import and explore datasets in R
3. Apply data exploration skills to a real-world example dataset

OPENING DISCUSSION: WHY DO WE EXPLORE OUR DATA?

Why is data exploration our first step in analyzing a dataset? What information do we gain? How does data exploration aid in our decision-making for data analysis steps further down the pipeline?

IMPORT DATA AND VIEW SUMMARIES

```
# 1. Set up your working directory
getwd()
```

```
## [1] "C:/Users/Felipe/OneDrive - Duke University/1. DUKE/1. Ramos 2 Semestre/EOS-872 Env. Data Analyti
```

```
# 2. Load packges
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.0.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.6
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts -------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# 3. Import datasets
USGS.flow.data <- read.csv("./Data/Raw/USGS_Site02085000_Flow_Raw.csv")

View(USGS.flow.data)
# Alternate option: click on data frame in Environment tab

class(USGS.flow.data)
```

```
## [1] "data.frame"
```

```
colnames(USGS.flow.data)
```

```
##  [1] "agency_cd"              "site_no"
##  [3] "datetime"               "X165986_00060_00001"
##  [5] "X165986_00060_00001_cd" "X165987_00060_00002"
##  [7] "X165987_00060_00002_cd" "X84936_00060_00003"
```

```
##  [9] "X84936_00060_00003_cd"  "X84937_00065_00001"
## [11] "X84937_00065_00001_cd"  "X84938_00065_00002"
## [13] "X84938_00065_00002_cd"  "X84939_00065_00003"
## [15] "X84939_00065_00003_cd"
```

```
# Rename columns
colnames(USGS.flow.data) <- c("agency_cd", "site_no", "datetime",
                              "discharge.max", "discharge.max.approval",
                              "discharge.min", "discharge.min.approval",
                              "discharge.mean", "discharge.mean.approval",
                              "gage.height.max", "gage.height.max.approval",
                              "gage.height.min", "gage.height.min.approval",
                              "gage.height.mean", "gage.height.mean.approval")
str(USGS.flow.data) #structure
```

```
## 'data.frame':    33216 obs. of  15 variables:
##  $ agency_cd                : Factor w/ 1 level "USGS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ site_no                  : int  2085000 2085000 2085000 2085000 2085000 2085000 2085000 2085000 2
##  $ datetime                 : Factor w/ 33216 levels "1/1/00","1/1/01",..: 20 1021 2022 2295 2386 24
##  $ discharge.max            : num  74 61 56 54 48 47 44 41 44 57 ...
##  $ discharge.max.approval   : Factor w/ 4 levels "","A","A:e","P": 2 2 2 2 2 2 2 2 2 2 ...
##  $ discharge.min            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ discharge.min.approval   : Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
##  $ discharge.mean           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ discharge.mean.approval  : Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
##  $ gage.height.max          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gage.height.max.approval : Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
##  $ gage.height.min          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gage.height.min.approval : Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
##  $ gage.height.mean         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gage.height.mean.approval: Factor w/ 3 levels "","A","P": 1 1 1 1 1 1 1 1 1 1 ...
```

```
dim(USGS.flow.data) #dimension
```

```
## [1] 33216    15
```

```
#head(USGS.flow.data) # not useful for kniting it (looks at six rows)
class(USGS.flow.data$datetime) # it is a factor. Problem. Formating dates
```

```
## [1] "factor"
```

ADJUSTING DATASETS

Formatting dates

R will often import dates as factors or characters rather than dates. To fix, this we need to tell R that it is looking at dates. We also need to specify the format the dates are in. By default, if you don't provide a format, R will attempt to use %Y-%m-%d or %Y/%m/%d as a default. Note: if you are working collaboratively in an international setting, using a year-month-day format in spreadsheets is the least ambiguous of date formats. Make sure to check whether month-day-year or day-month-year is used in an ambiguously formatted spreadsheet.

Formatting of dates in R:

%d day as number (0-31) %m month (00-12, can be e.g., 01 or 1) %y 2-digit year %Y 4-digit year %a abbreviated weekday %A unabbreviated weekday %b abbreviated month %B unabbreviated month

In some cases when dates are provided as integers, you may need to provide an origin for your dates. Beware: the "origin" date for Excel (Windows), Excel (Mac), R, and MATLAB all have different origin dates. Google this if it comes up.

```
help(as.Date)
```

```
## starting httpd help server ... done
```

```
# Adjust date formatting for today
# Write code for three different date formats.
# An example is provided to get you started.
# (code must be uncommented)
today <- Sys.Date()
format(today, format = "%B")
```

```
## [1] "febrero"
```

```
format(today, format = "%d-%m-%Y")
```

```
## [1] "22-02-2019"
```

```
format(today, format = "%a")
```

```
## [1] "vi."
```

```
format(today-1, format = "%A %d de %B de %Y")
```

```
## [1] "jueves 21 de febrero de 2019"
```

```
USGS.flow.data$datetime <- as.Date(USGS.flow.data$datetime, format = "%m/%d/%y")
```

Note that for every date prior to 1969, R has assigned the date in the 2000s rather than the 1900s. This can be fixed with an `ifelse` statement inside a function. Run through the code below and write what is happening in the comment above each line.

```
# changes the format to remove 19 and 20
USGS.flow.data$datetime <- format(USGS.flow.data$datetime, "%y%m%d")

# Changes 20 with 19
create.early.dates <- (function(d) {
       paste0(ifelse(d > 181231,"19","20"),d)
       })
# use the function
USGS.flow.data$datetime <- create.early.dates(USGS.flow.data$datetime)

# fix the format
USGS.flow.data$datetime <- as.Date(USGS.flow.data$datetime, format = "%Y%m%d")
```

Removing NAs

Notice in our dataset that our discharge and gage height observations have many NAs, meaning no measurement was recorded for a specific day. In some cases, it might be in our best interest to remove NAs from a dataset. Removing NAs or not will depend on your research question.

```
summary(USGS.flow.data$discharge.mean)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##    0.220    5.005   15.200   44.598   40.600 3270.000    28049
```

```
summary(USGS.flow.data$gage.height.mean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.870   1.430   1.720   1.856   2.120  14.470   28171
```

Question: What types of research questions might make it favorable to remove NAs from a dataset, and what types of research questions might make it favorable to retain NAs in the dataset?

> Answer: NA can mean something (survey, valid response). In other cases (quant) . careful with blank spaces. Important to explain in readme file what NA is. Is the next given there where years with no data we eliminate them.

```
USGS.flow.data.complete <- na.omit(USGS.flow.data) #remove the entire row  (look for how to removes NAs
dim(USGS.flow.data)
```

```
## [1] 33216    15
```

```
dim(USGS.flow.data.complete)
```

```
## [1] 5045   15
```

```
#complete.cases. tells you what rows (or column) is complete.

mean(USGS.flow.data.complete$discharge.mean)
```

```
## [1] 45.63464
```

```
sd(USGS.flow.data.complete$discharge.mean)
```

```
## [1] 124.4523
```

```
summary(USGS.flow.data.complete$discharge.mean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.22    5.43   15.90   45.63   41.80 3270.00
```

VISUALIZATION FOR DATA EXPLORATION

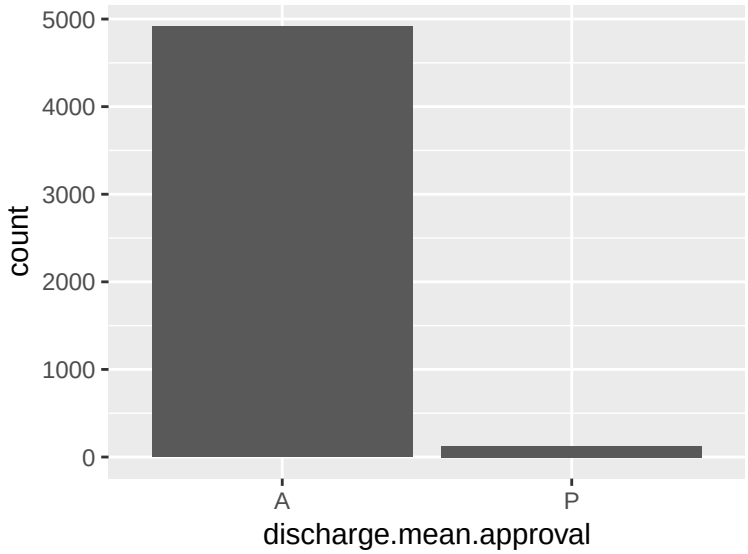
Although the `summary()` function is helpful in getting an idea of the spread of values in a numeric dataset, it can be useful to create visual representations of the data to help form hypotheses and direct downstream data analysis. Below is a summary of the useful types of graphs that can be

Note: each of these approaches utilize the package "ggplot2". We will be covering the syntax of ggplot in a later lesson, but for now you should familiarize yourself with the functionality of what each command is doing.

Bar Chart (function: geom_bar)

Visualize count data for categorical variables.

```
ggplot(USGS.flow.data.complete, aes(x = discharge.mean.approval)) +
  geom_bar() #works great for categorical. also for quan but you can use histogram.
```

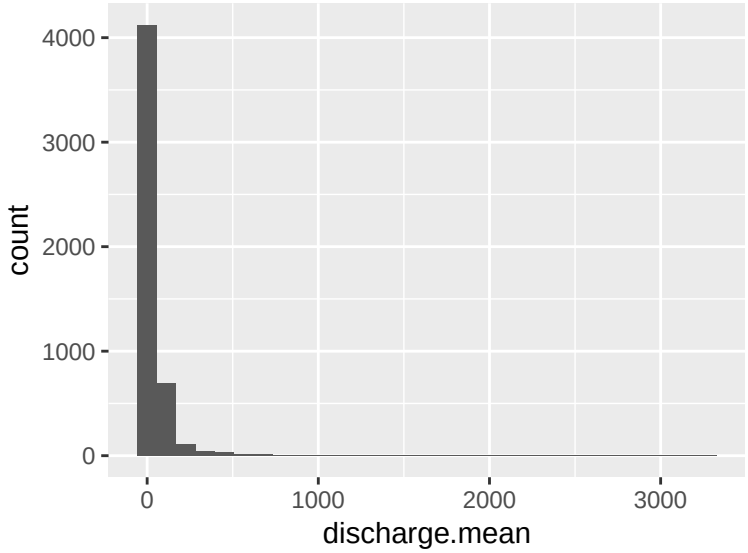


Histogram (function: geom_histogram)

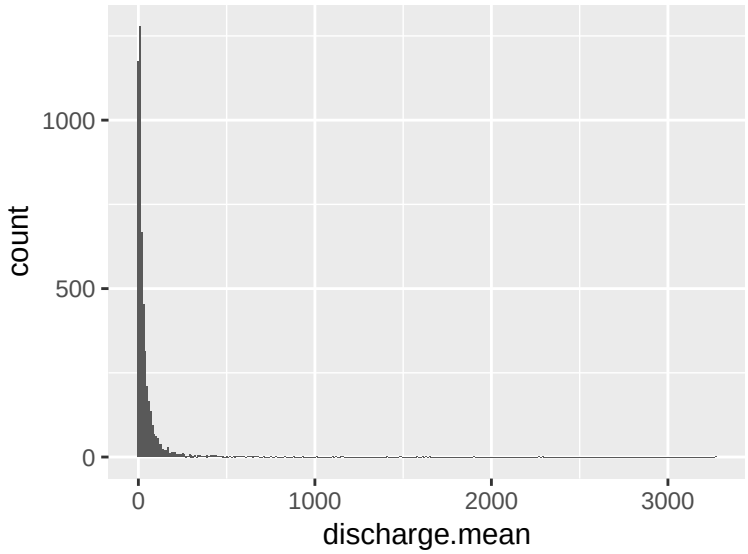
Visualize distributions of values for continuous numerical variables. What is happening in each line of code? Insert a comment above each line.

```
# better to use the aes there. Is more specific for bigger plots.
ggplot(USGS.flow.data.complete) +
  geom_histogram(aes(x = discharge.mean))
```

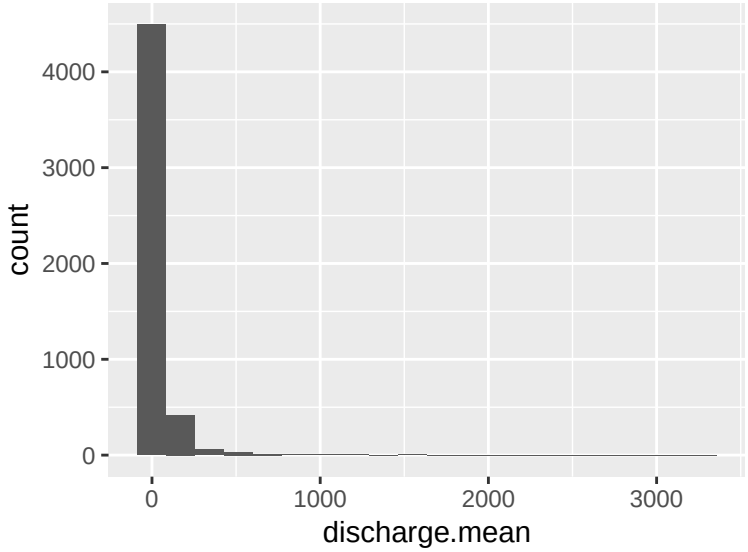
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# narrower the bins (units of the variable)
ggplot(USGS.flow.data.complete) +
  geom_histogram(aes(x = discharge.mean), binwidth = 10)
```

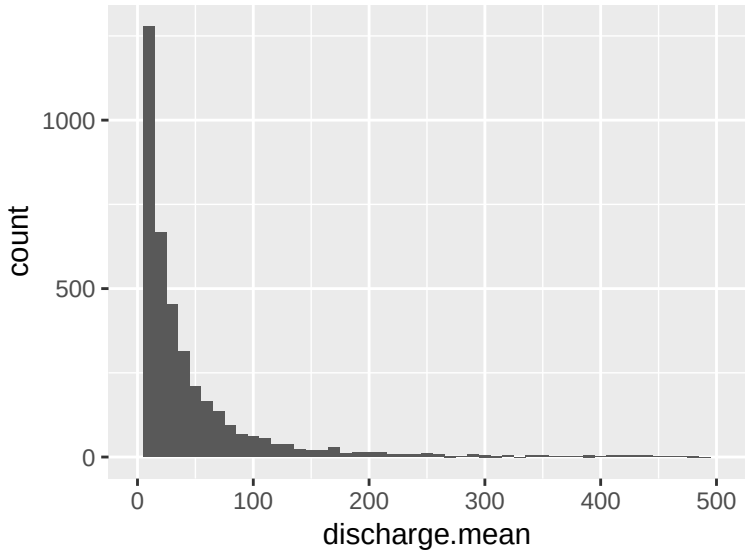



```
# or the number of bins
ggplot(USGS.flow.data.complete) +
  geom_histogram(aes(x = discharge.mean), bins = 20)
```



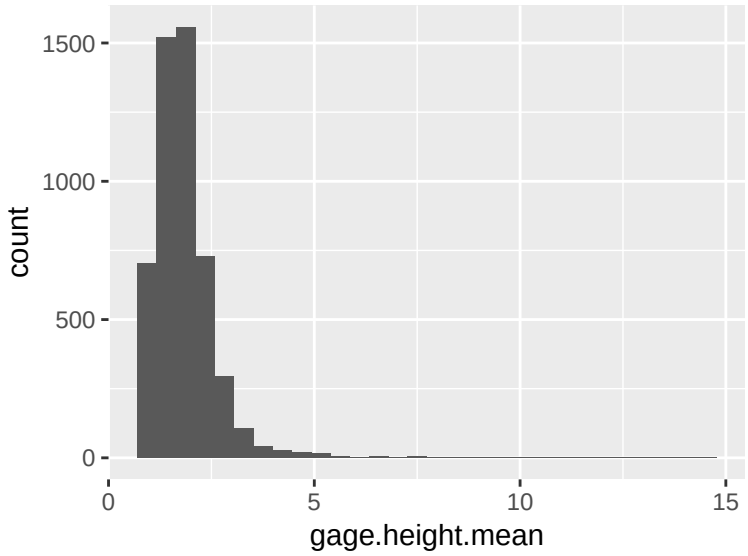

```
# we remove some data from the graph. is like doing a some. explain that you are doing that and why.
ggplot(USGS.flow.data.complete, aes(x = discharge.mean)) +
  geom_histogram(binwidth = 10) +
  scale_x_continuous(limits = c(0, 500))
```

```
## Warning: Removed 53 rows containing non-finite values (stat_bin).
```




```
# another variable
ggplot(USGS.flow.data.complete) +
  geom_histogram(aes(x = gage.height.mean))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# you can name your graph to call it later
```

Frequency line graph (function: geom_freqpoly)

An alternate to a histogram is a frequency polygon graph (distributions of values for continuous numerical variables). Instead of displaying bars, counts of continuous variables are displayed as lines. This is advantageous if you want to display multiple variables or categories of variables at once.

```
#three frecuency poligons for diff data. we dont want to mix variables with diff units.
ggplot(USGS.flow.data.complete) +
  geom_freqpoly(aes(x = gage.height.mean), bins = 50) +
```

```
  geom_freqpoly(aes(x = gage.height.min), bins = 50, color = "blue") +
  geom_freqpoly(aes(x = gage.height.max), bins = 50, color = "red") +
  scale_x_continuous(limits = c(0, 10))
```

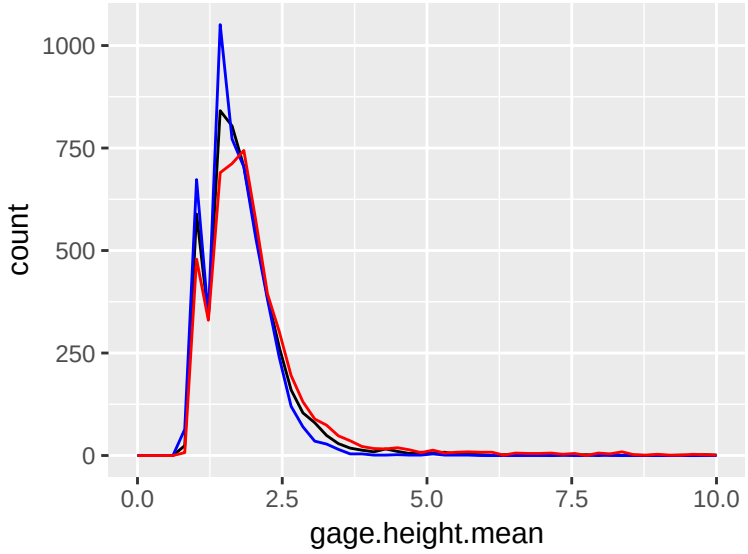
```
## Warning: Removed 4 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 28 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_path).
```

```
## Warning: Removed 2 rows containing missing values (geom_path).
```

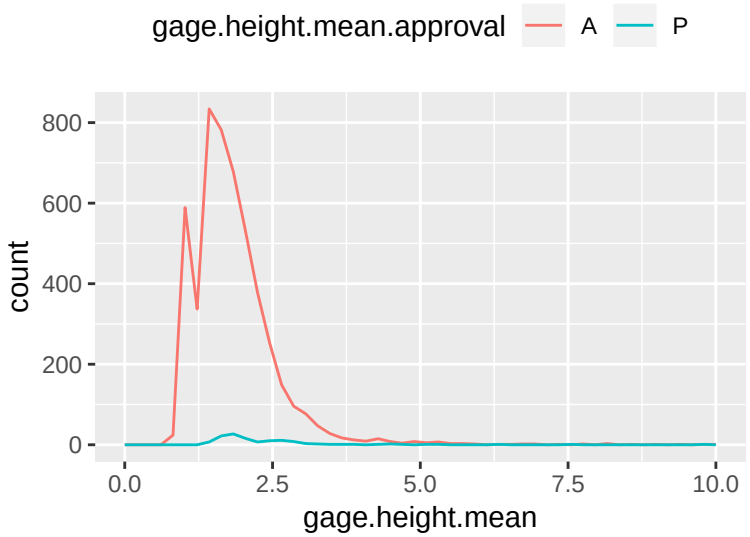
```
## Warning: Removed 2 rows containing missing values (geom_path).
```




```
# color it depending of the status of aproval
ggplot(USGS.flow.data.complete) +
  geom_freqpoly(aes(x = gage.height.mean, color = gage.height.mean.approval), bins = 50) +
  scale_x_continuous(limits = c(0, 10)) +
  theme(legend.position = "top")
```

```
## Warning: Removed 4 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 4 rows containing missing values (geom_path).
```



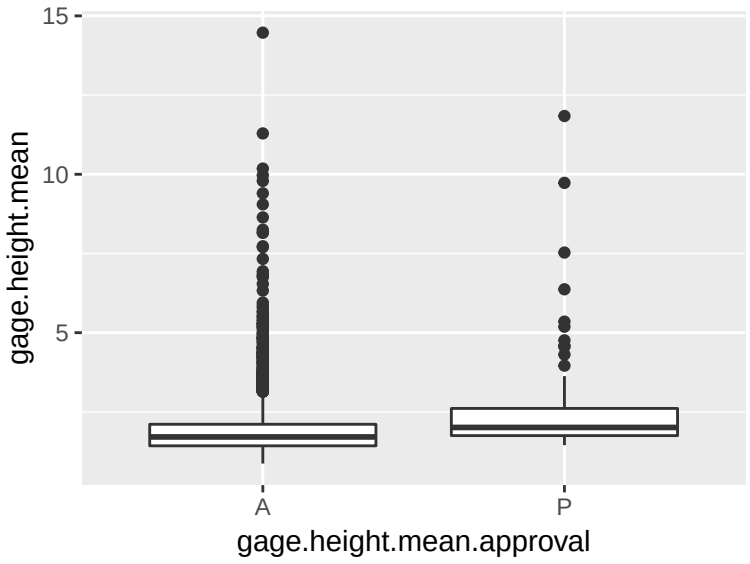
Box-and-whisker plots (function: geom_boxplot)

A box-and-whisker plot is yet another alternative to histograms (distributions of values for continuous numerical variables). These plots consist of:

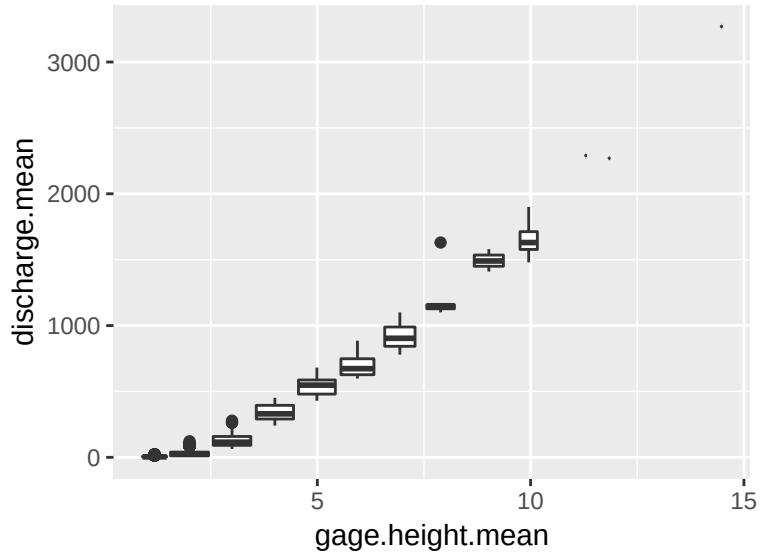
- A box from the 25th to the 75th percentile of the data, called the interquartile range (IQR).
- A bold line inside the box representing the median value of the data. Whether the median is in the center or off to one side of the IQR will give you an idea about the skewness of your data.
- A line outside of the box representing values falling within 1.5 times the IQR.
- Points representing outliers, values that fall outside 1.5 times the IQR.

An alternate option is a violin plot, which displays density distributions, somewhat like a hybrid of the box-and-whiskers and the frequency polygon plot.

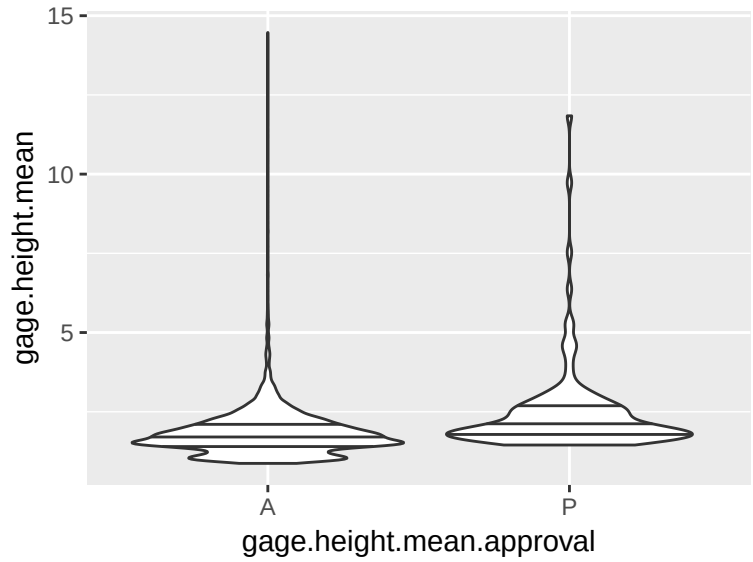
```
#
ggplot(USGS.flow.data.complete) +
  geom_boxplot(aes(x = gage.height.mean.approval, y = gage.height.mean))
```



```
# we divided the data in groups (of 1?)
ggplot(USGS.flow.data.complete) +
  geom_boxplot(aes(x = gage.height.mean, y = discharge.mean, group = cut_width(gage.height.mean, 1)))
```




```
# another type of boxplot.You have to specify the quantiles
ggplot(USGS.flow.data.complete) +
  geom_violin(aes(x = gage.height.mean.approval, y = gage.height.mean), draw_quantiles = c(0.25, 0.5, 0
```



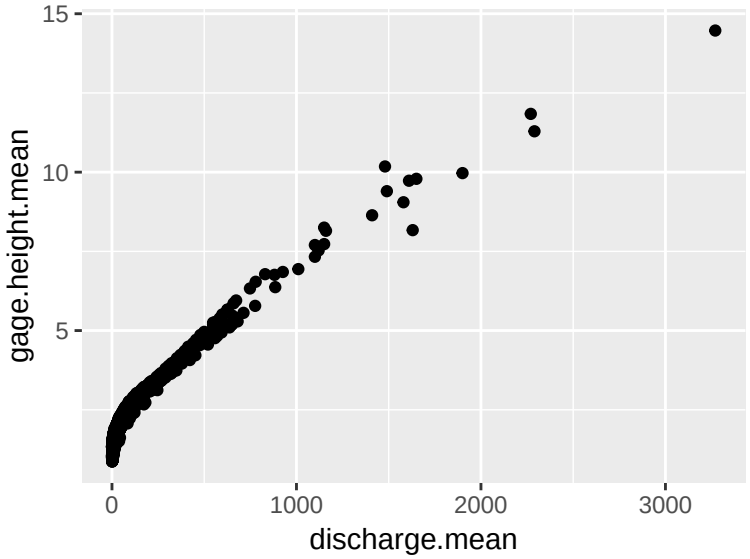
Question: what are the pros and cons of each type of frequency graph (histogram, frequency polygon, box-and whisker, violin)?

Answer:

Scatterplot (function: geom_point)

Visualize relationships between continuous numerical variables.

```
ggplot(USGS.flow.data.complete) +
  geom_point(aes(x = discharge.mean, y = gage.height.mean))
```



ENDING DISCUSSION

How can multiple options for data exploration inform our understanding of our data? What did you learn about the USGS discharge dataset today?

ANSWER:

This passage from R for Data Science sums up some of the questions we should ask ourselves when initially exploring a dataset. "Patterns in your data provide clues about relationships. If a systematic relationship exists between two variables it will appear as a pattern in the data. If you spot a pattern, ask yourself:

"Could this pattern be due to coincidence (i.e. random chance)?

"How can you describe the relationship implied by the pattern?

"How strong is the relationship implied by the pattern?

"What other variables might affect the relationship?

"Does the relationship change if you look at individual subgroups of the data?"

Do you see any patterns in the USGS data for the Eno River? What might be responsible for those patterns and/or relationships?

ANSWER: