# 5: Data Exploration

Environmental Data Analytics | John Fay and Luana Lima | Developed by Kateri Salk

Spring 2022

## Objectives

1. Import and explore datasets in R
2. Graphically 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] "/Users/ataliefischer/Desktop/EDA/Environmental_Data_Analytics_2022/Lessons"
```

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

# 3. Import datasets
USGS.flow.data <- read.csv("../Data/Processed/USGS_Site02085000_Flow_Processed.csv")

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

colnames(USGS.flow.data) #column names
```

```
##  [1] "agency_cd"                "site_no"
##  [3] "datetime"                 "discharge.max"
##  [5] "discharge.max.approval"   "discharge.min"
##  [7] "discharge.min.approval"   "discharge.mean"
##  [9] "discharge.mean.approval"  "gage.height.max"
## [11] "gage.height.max.approval" "gage.height.min"
## [13] "gage.height.min.approval" "gage.height.mean"
## [15] "gage.height.mean.approval"
```

```
str(USGS.flow.data) #structure
```

```
## 'data.frame':    33690 obs. of  15 variables:
##  $ agency_cd                : chr  "USGS" "USGS" "USGS" "USGS" ...
##  $ site_no                  : int  2085000 2085000 2085000 2085000 2085000 2085000 2085000 2085000 20
##  $ datetime                 : chr  "1927-10-01" "1927-10-02" "1927-10-03" "1927-10-04" ...
```

```
##  $ discharge.max           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ discharge.max.approval  : chr  "" "" "" "" ...
##  $ discharge.min           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ discharge.min.approval  : chr  "" "" "" "" ...
##  $ discharge.mean          : num  39 39 39 39 39 39 39 39 39 39 ...
##  $ discharge.mean.approval : chr  "A" "A" "A" "A" ...
##  $ gage.height.max         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gage.height.max.approval : chr  "" "" "" "" ...
##  $ gage.height.min         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gage.height.min.approval : chr  "" "" "" "" ...
##  $ gage.height.mean        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gage.height.mean.approval: chr  "" "" "" "" ...
```

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

```
## [1] 33690    15
```

```
# Check our date column
class(USGS.flow.data$datetime)
```

```
## [1] "character"
```

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

```
## [1] "Date"
```

## Visualization for Data Exploration

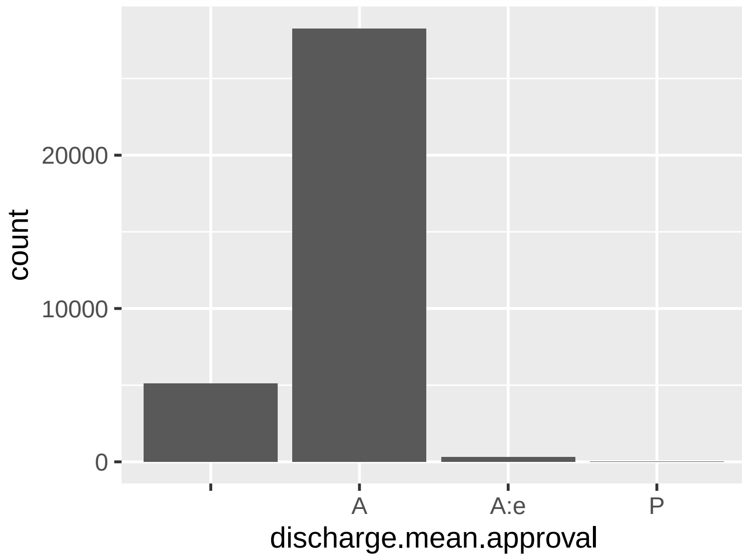
Although the `summary()` function is helpful in getting an idea of the spread of values in a numeric dataset, it can be useful to create visual representations of the data to help form hypotheses and direct downstream data analysis. Below is a summary of the useful types of graphs for data exploration.

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.

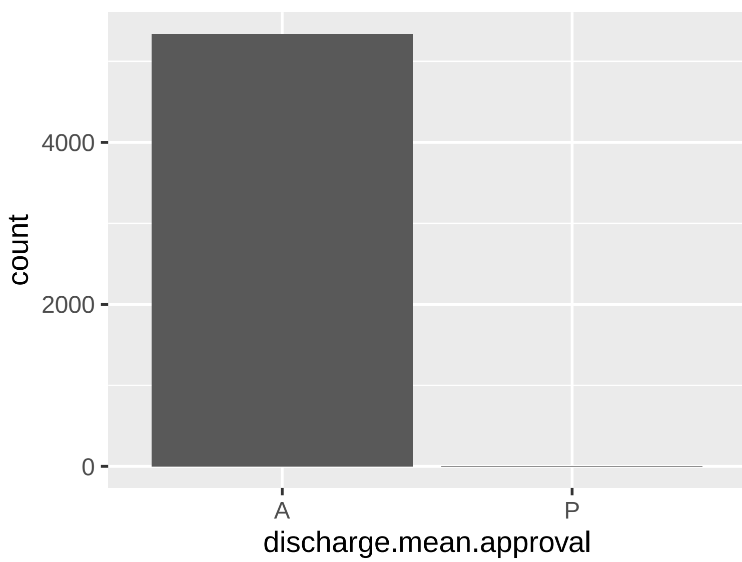
### Bar Chart (function: geom_bar)

Visualize count data for categorical variables.

```
#categorical data representation
ggplot(USGS.flow.data, aes(x = discharge.mean.approval)) +
  geom_bar()
```




```
      #geom_bar() --> specifies bar plot

ggplot(na.omit(USGS.flow.data), aes(x = discharge.mean.approval)) +
  geom_bar()
```



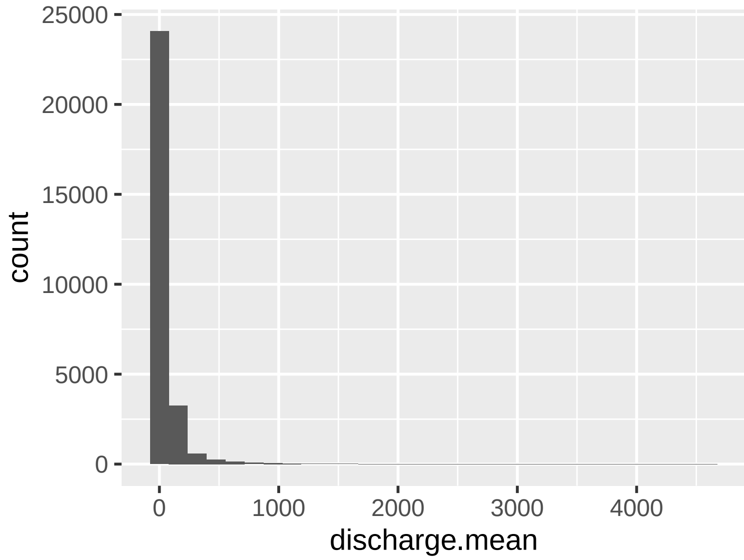

**Histogram (function: geom_histogram)**

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

```
#non-normal distribution, skewed right
ggplot(USGS.flow.data) +
  geom_histogram(aes(x = discharge.mean))
```

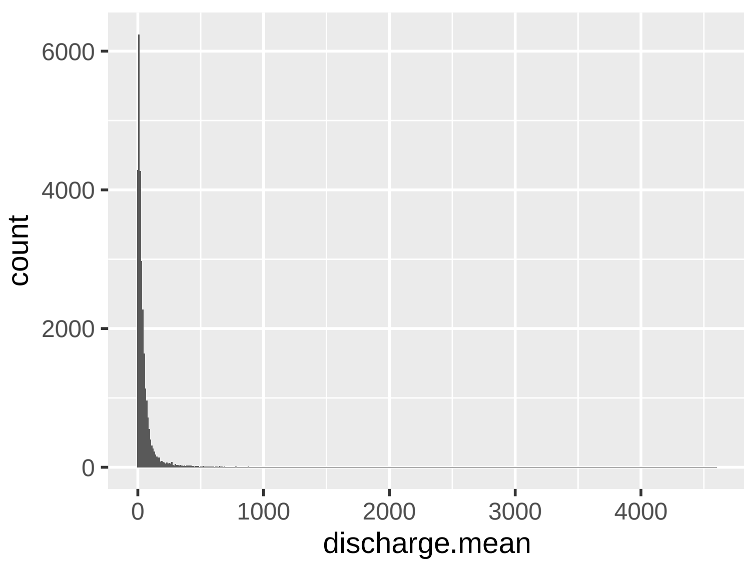
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 5108 rows containing non-finite values (stat_bin).
```



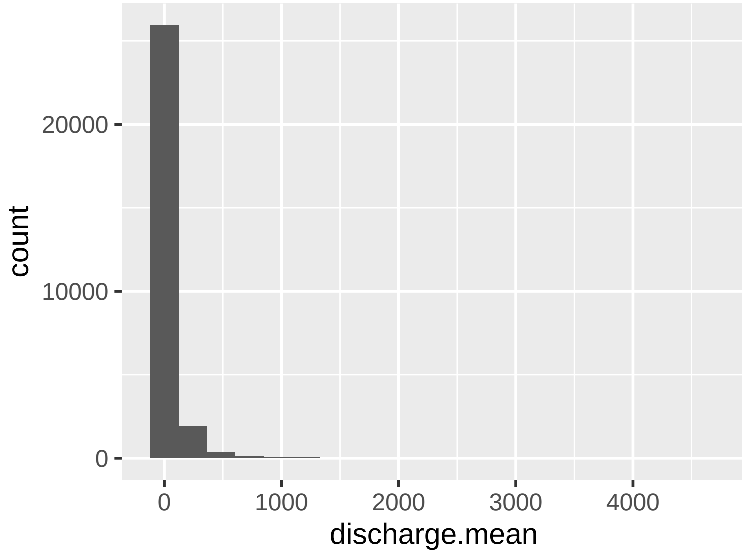

```
#binwidth (default is 30)
ggplot(USGS.flow.data) +
  geom_histogram(aes(x = discharge.mean), binwidth = 10)
```

```
## Warning: Removed 5108 rows containing non-finite values (stat_bin).
```




```
#number of bins (good practice, between 5 and 20 bins)
ggplot(USGS.flow.data) +
  geom_histogram(aes(x = discharge.mean), bins = 20)
```

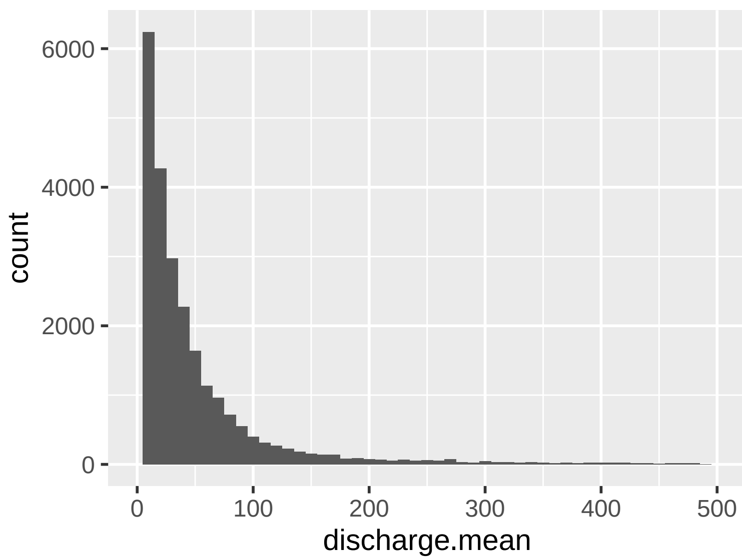
```
## Warning: Removed 5108 rows containing non-finite values (stat_bin).
```




```
#scale_x_continuous --> forces x to go from 0 to 500
ggplot(USGS.flow.data, aes(x = discharge.mean)) +
  geom_histogram(binwidth = 10) +
  scale_x_continuous(limits = c(0, 500))
```

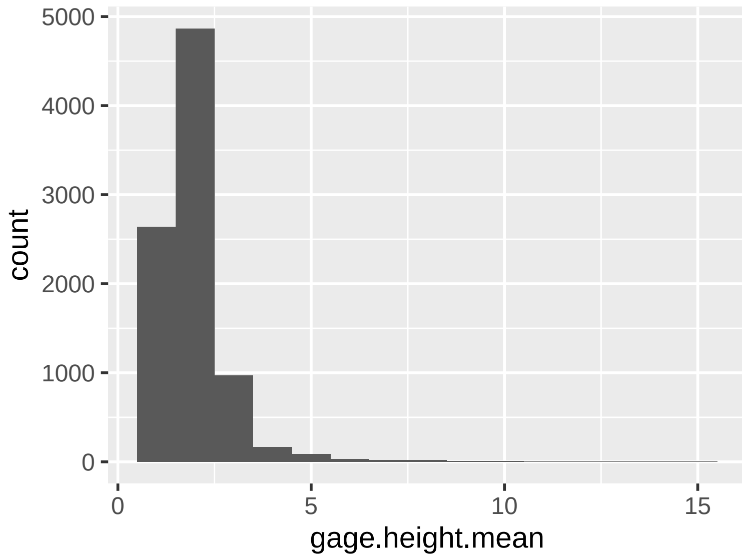
```
## Warning: Removed 5577 rows containing non-finite values (stat_bin).
```

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




```
#
ggplot(USGS.flow.data) +
  geom_histogram(aes(x = gage.height.mean), binwidth = 1)
```

```
## Warning: Removed 24870 rows containing non-finite values (stat_bin).
```

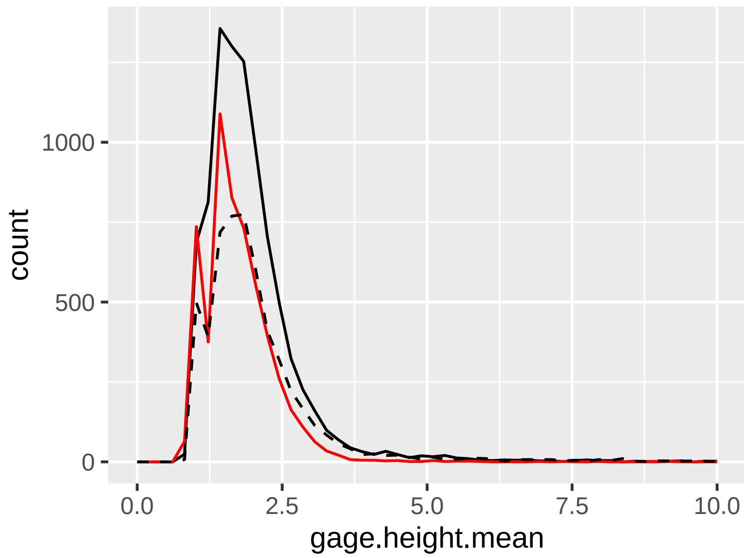



### Frequency line graph (function: geom_freqpoly)

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.

```
#can add multiple lines on top of one another with different colors and line types
ggplot(USGS.flow.data) +
  geom_freqpoly(aes(x = gage.height.mean), bins = 50) +
  geom_freqpoly(aes(x = gage.height.min), bins = 50, color = "red") +
  geom_freqpoly(aes(x = gage.height.max), bins = 50,  lty = 2) +
  scale_x_continuous(limits = c(0, 10))
```

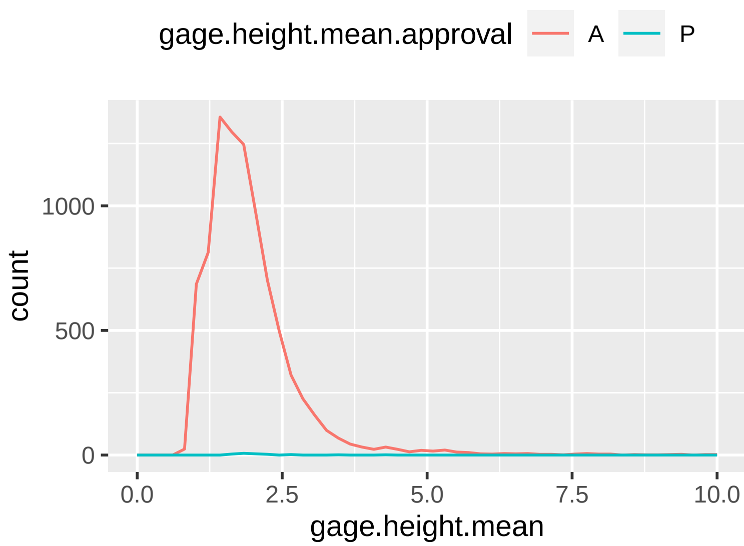
```
## Warning: Removed 24887 rows containing non-finite values (stat_bin).

## Warning: Removed 28229 rows containing non-finite values (stat_bin).

## Warning: Removed 28266 rows containing non-finite values (stat_bin).

## Warning: Removed 2 row(s) containing missing values (geom_path).

## Warning: Removed 2 row(s) containing missing values (geom_path).

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




```
#set color to a categorical value to show two different categories of data on the same plot
ggplot(USGS.flow.data) +
  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 24887 rows containing non-finite values (stat_bin).
```

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




### Box-and-whisker plots (function: geom_boxplot, geom_violin)

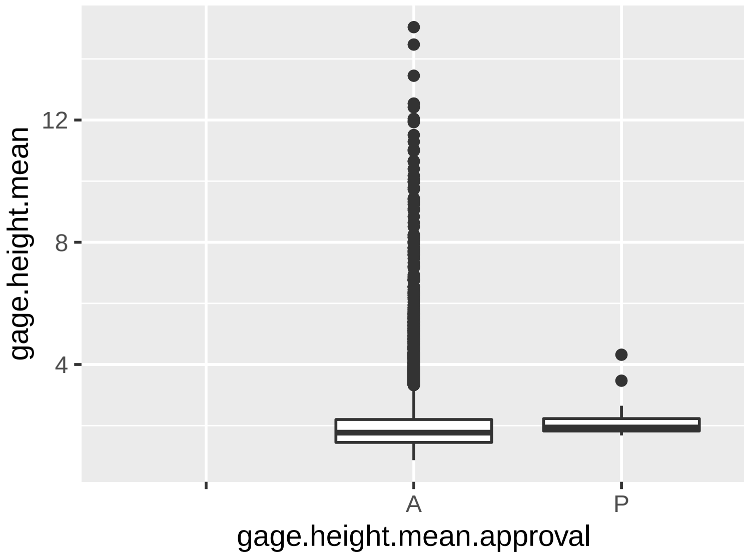
A box-and-whisker plot is yet another alternative to histograms (distributions of values for continuous numerical variables). These plots consist of:

- 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.

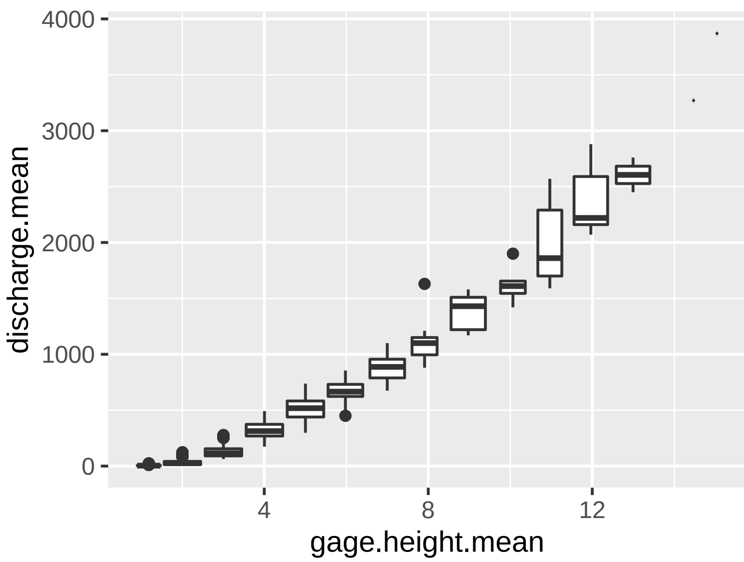
```r
#x --> what data we want to show, y--> multiple datasets
ggplot(USGS.flow.data) +
  geom_boxplot(aes(x = gage.height.mean.approval, y = gage.height.mean))
```

```
## Warning: Removed 24870 rows containing non-finite values (stat_boxplot).
```



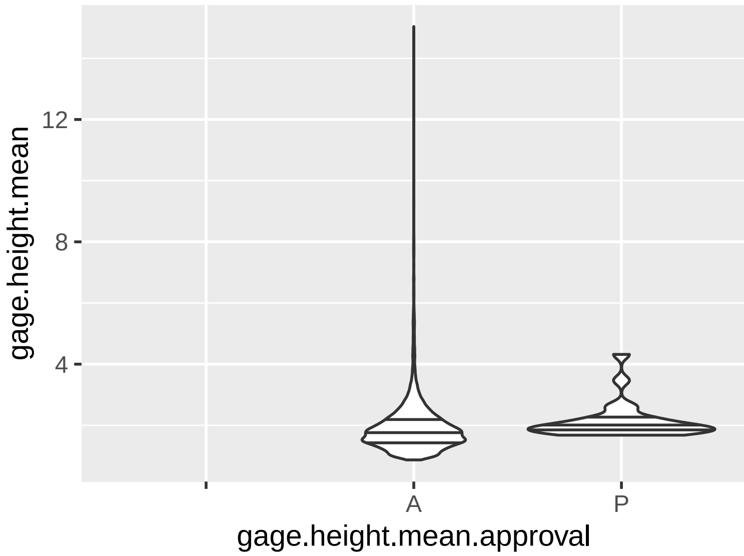

```r
#group=cut_width --> distribution of chunks of data (like a histogram of boxplots)
ggplot(USGS.flow.data) +
  geom_boxplot(aes(x = gage.height.mean, y = discharge.mean, group = cut_width(gage.height.mean, 1)))
```

```
## Warning: Removed 24870 rows containing missing values (stat_boxplot).
```




```r
#geom_violin--> range of values and distribution within that range
ggplot(USGS.flow.data) +
  geom_violin(aes(x = gage.height.mean.approval, y = gage.height.mean),
              draw_quantiles = c(0.25, 0.5, 0.75))
```

```
## Warning: Removed 24870 rows containing non-finite values (stat_ydensity).
```

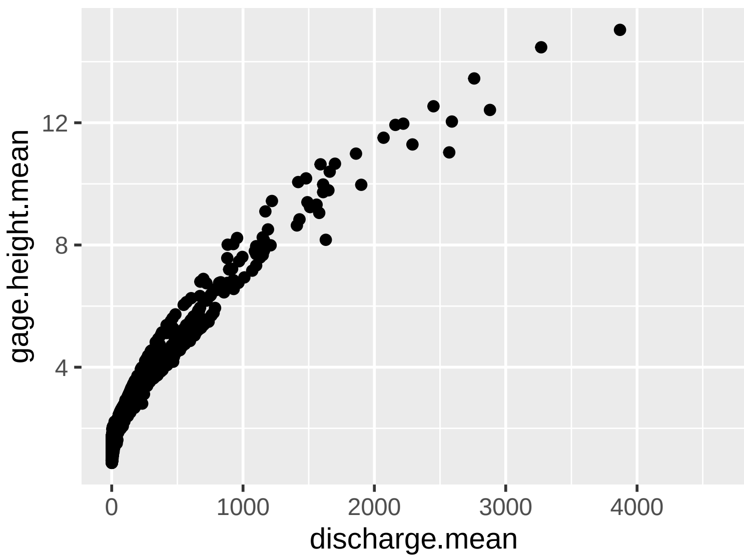



**Scatterplot (function: geom_point)**

Visualize relationships between continuous numerical variables.

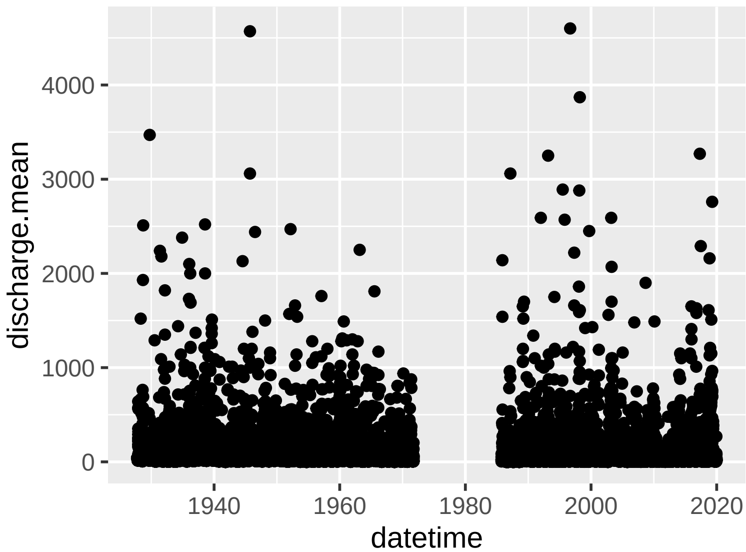
```
ggplot(USGS.flow.data) +
  geom_point(aes(x = discharge.mean, y = gage.height.mean))
```

```
## Warning: Removed 24870 rows containing missing values (geom_point).
```




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

```
## Warning: Removed 5108 rows containing missing values (geom_point).
```




```
#can see big break in data through time
```

Question: under what circumstances would it be beneficial to use each of these graph types (bar plot, histogram, frequency polygon, box-and whisker, violin, scatterplot)?

> Answer: Barplots are useful when comparing categorical data. Histograms are useful for examining the distribution of continuous numerical data. Frequency line graphs are also useful for examining the distribution of continuous numerical data, but can display more than one distribution at a time. Box-and-whisker plots are useful for comparing medians and range of data, as well as showing outliers. Violin plots are useful for examining density distributions (how much of the data lies where). Scatter plots are useful for examining the relationship between continuous numerical data. It can elucidate trends.

## Ending discussion

What did you learn about the USGS discharge dataset today? What separate insights did the different graph types offer?

> Answer:

How can multiple options for data exploration inform our understanding of our data?

> Answer:

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

> Answer: