Stat 470/670 Lecture 2: Univariate Data Visualization

Julia Fukuyama, with thanks to Brad Luen

August 23, 2018

The Basics/Our Goals

Goals:

- Describe a single univariate dataset.
- Compare a univariate dataset to a reference distribution or to another univariate dataset.
- Compare many univariate datasets.

Even if you have multivariate data, you should start out looking at the variables one by one, as if they were univariate.

Quantile Plots and ECDFs

Goal: Describe a single univariate dataset

Empirical CDF: Definition

Let $x_1, \dots, x_n$ be our dataset.

The empirical cumulative distribution function (ecdf) is defined as

$$\text{ecdf}(x) = \frac{\text{\# of elements in the dataset with value } \leq x}{\text{\# of elements in the dataset}}$$

Properties:

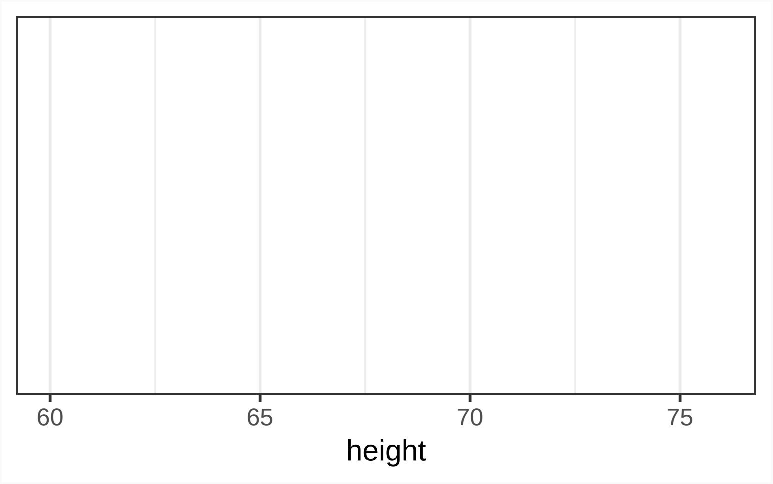
- The function is monotone increasing.
- Regions in which the curve is steep $\leftrightarrow$ regions of high density.
- Regions in which the curve is flat $\leftrightarrow$ regions of low density.
- Can easily read off the fraction of points in an interval from the function.
- Assuming the samples are exchangeable, there is no loss of information going from the original dataset to the ecdf.

Let’s try this out in R

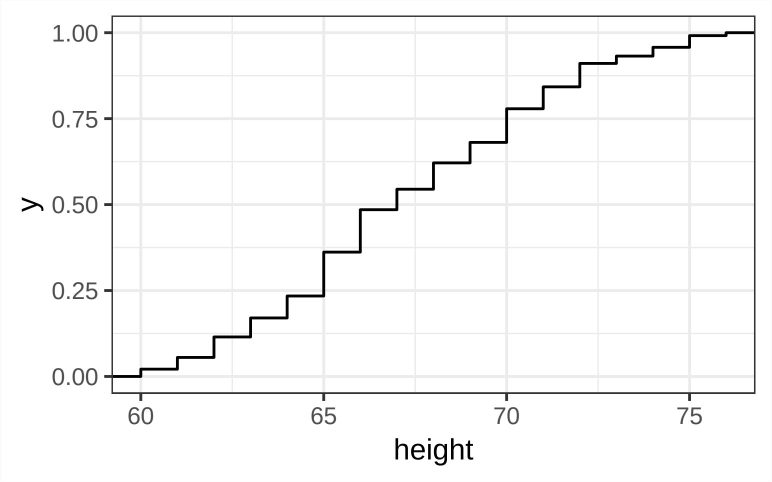
```r
## lattice has the singer data that we're going to use
library(lattice)
library(ggplot2)
library(dplyr)
library(magrittr)
library(stringr)
```

Let’s try out ggplot

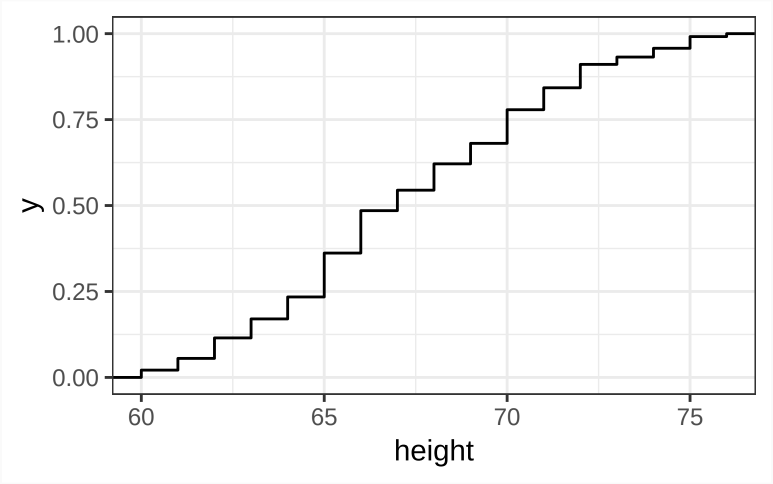
```r
## nothing gets plotted! why not?
ggplot(singer, aes(x = height))
```



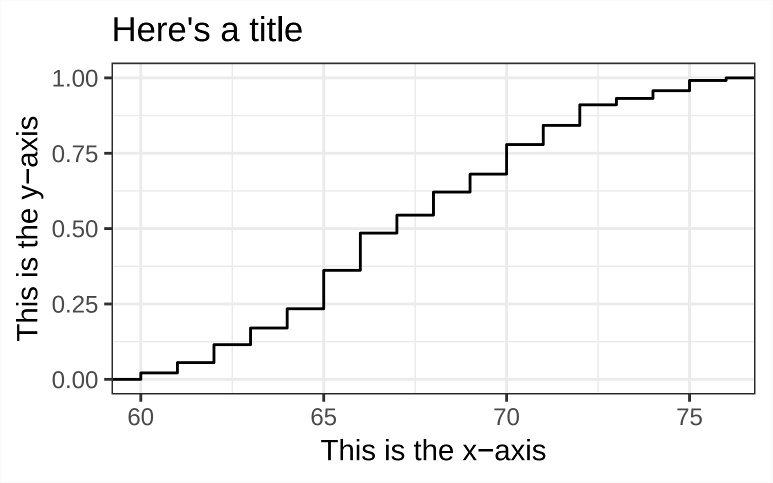
```r
## we need to tell it not just that we want to plot height, but how to
ing to plot height as an ecdf
ggplot(singer, aes(x = height)) + stat_ecdf()
```



```
## another way of doing the same thing, ggplots can come in pieces
singer.gg = ggplot(singer, aes(x = height))
singer.gg + stat_ecdf()
```



```r
## and if we want to label the axes
singer.gg + stat_ecdf() +
    xlab("This is the x-axis") +
    ylab("This is the y-axis") +
    ggtitle("Here's a title")
```



Exercise: Work with your neighbor!

Make a plot with the ecdfs for each of the voice parts. Try one with all of the voice parts on the same plot, and one with the voice parts faceted out by voice part. Is facet_wrap the best way?

Quantile function: Definition

The f quantile, $q(f)$, of a set of data is a value with the property that approximately a fraction f of the data are less than or equal to $q(f)$.

Note:

- This definition doesn't completely specify the quantile function.
- The ecdf is one example of a quantile function.

For concreteness, we use Cleveland's definition of the quantile function:

- Let $x_1, \ldots, x_n$ be the data from the n samples.
- Let $x_{(1)}, \ldots, x_{(n)}$ be the ordered data, so that $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$
- Let $f_i = \frac{i-.5}{n}$
- Let $q(f_i) = x_{(i)}$

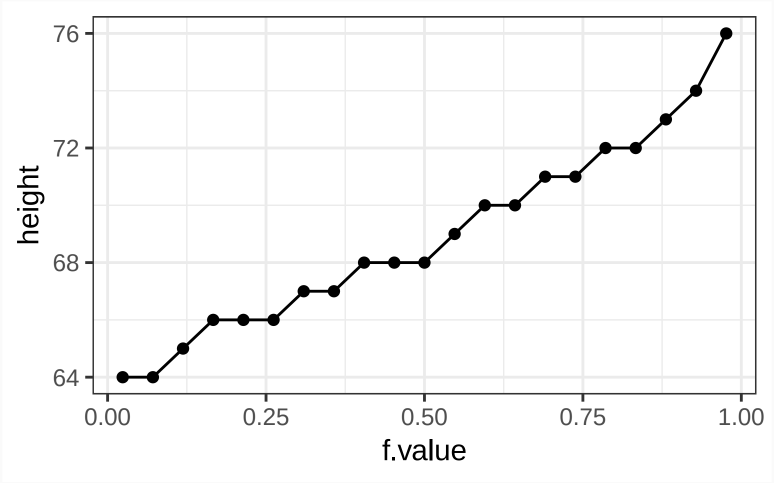
Now we have a partial specification of a quantile function.

Create the remainder by linear interpolation of the points we do have.

There isn't a nice R function for making quantile plots using Cleveland's definition, but we can still work it out by hand.

```
## quantile plots by hand
Tenor1 = singer %>%
    subset(voice.part == "Tenor 1") %>%
    arrange(height)
## exactly the same as
Tenor1 = arrange(
    subset(singer, voice.part == "Tenor 1"), height)
## close to the same as
sort(singer$height[singer$voice.part == "Tenor 1"])


##  [1] 64 64 65 66 66 66 67 67 68 68 68 69 70 70 71 71 72 72 73 74 76


nTenor1 = nrow(Tenor1)
f.value = (0.5:(nTenor1 - 0.5))/nTenor1
Tenor1$f.value = f.value
```

```r
ggplot(Tenor1, aes(x = f.value, y = height)) +
    geom_line() +
    geom_point()
```



Exercise: Do this for all the voice parts (either by hand, one at a time, or preferably programmatically), and plot all of the quantile plots faceted out by voice part or plotted all on the same color scale

Other interpretations of the ECDF

Let X be a random variable obtained by drawing uniformly at random from the dataset $x_1, \dots, x_n$.

$$\text{ecdf}(x) = P(X \leq x)$$

Why is the ECDF a good representation of the data?

Remember from your other statistics courses the definition of a cumulative distribution function:

Let X be a random variable taking values in $\mathbb{R}$, the cumulative distribution function F_X is defined as

$$F_X(x) = P(X \leq x)$$

The empirical CDF is the analogous quantity for our data, and is the nonparametric maximum likelihood estimate of the population CDF.

Histograms and Density Estimates

Histogram: Definition

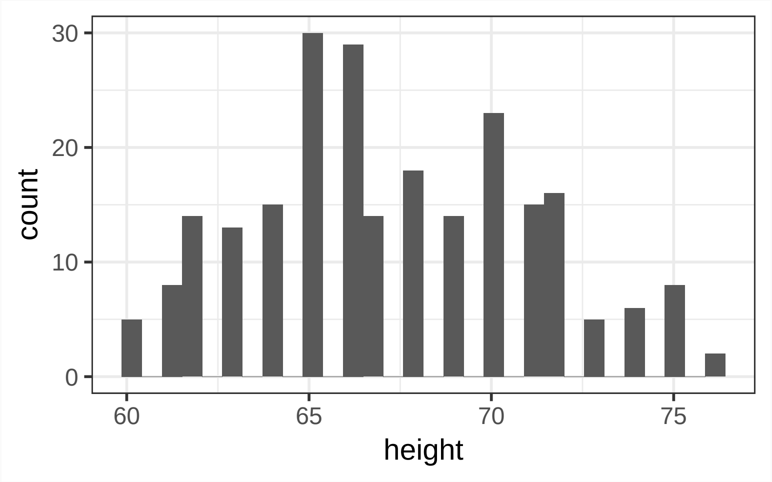
Let $x_{(1)}, \dots, x_{(n)}$ be the ordered data.

- Set a number of bins m.
- Let $r = x_{(n)} - x_{(1)}$.
- Create m equally sized intervals: $I_i = [x_{(1)} + \frac{r(i-1)}{m}, x_{(1)} + \frac{ri}{m})$
- For each $i = 1, \dots, m$, let $c_i = \#\{j : x_j \in I_i\}$.
- For each interval I_i, plot a bar with height c_i, width r/m, centered around $x_{(1)} + \frac{r(i-1)}{m}$.

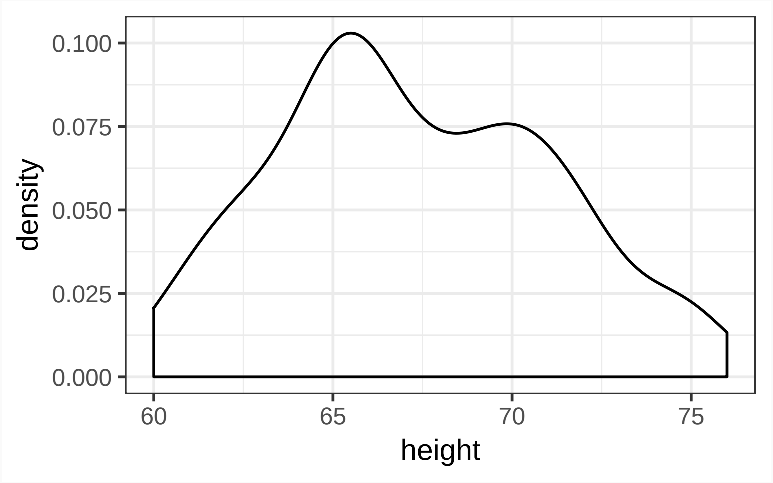
Histogram/Density estimate demonstration

```r
ggplot(singer, aes(x = height)) + geom_histogram()
```

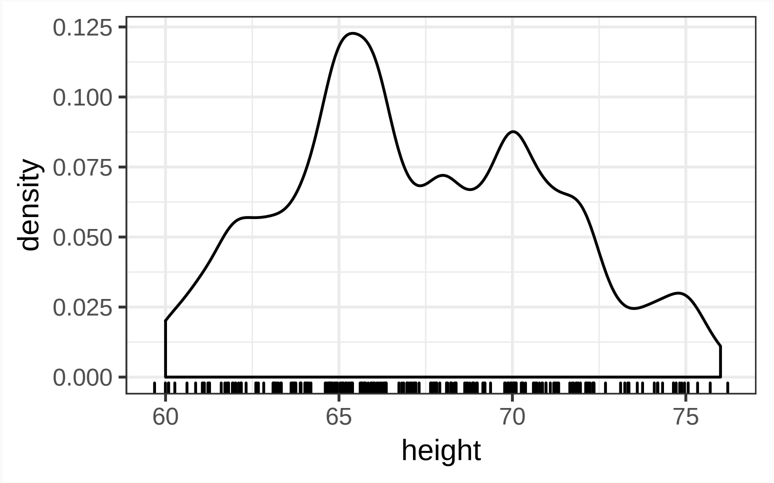
```
## `stat_bin()` using `bins = 30`. Pick better value with
`binwidth`.
```



```r
ggplot(singer, aes(x = height)) + geom_density()
```



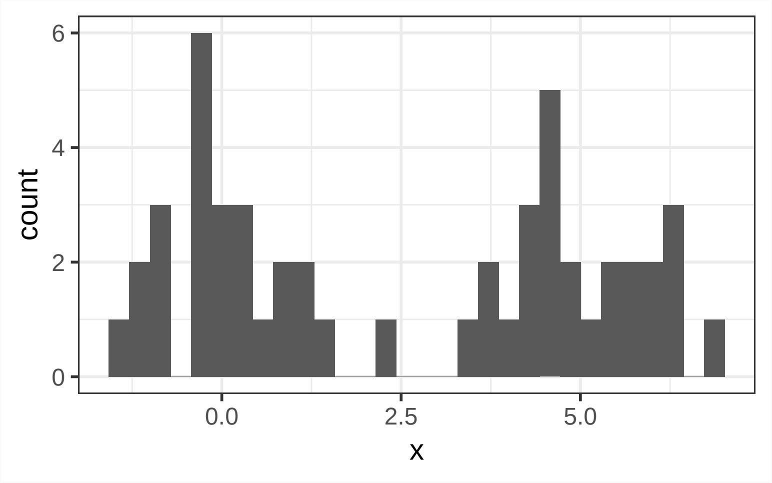
```r
ggplot(singer, aes(x = height)) +
    geom_density(adjust = .5) +
    geom_rug(aes(y = 0), sides = "b",
             position = position_jitter(height = 0))
```



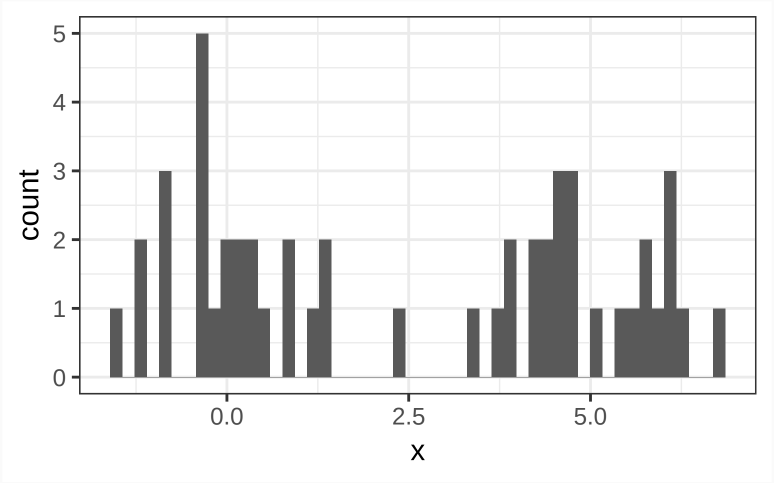
Exercise: play around with the jittering on the rug and the adjust parameter in the density. What do you like best? Again, try faceting out the histograms or plotting them over one another. Do different versions bring out different features of the data? What do you notice in the different plots?

```r
set.seed(0)
df = data.frame(x = c(rnorm(25, 0, 1), rnorm(25, 5, 1)))
ggplot(df) + geom_histogram(aes(x = x))
```

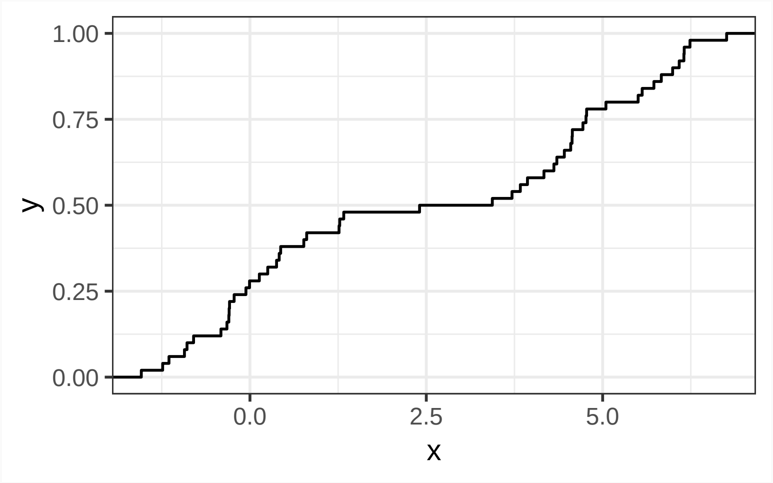
```
## `stat_bin()` using `bins = 30`. Pick better value with
`binwidth`.
```



```r
ggplot(df) + geom_histogram(aes(x = x), bins = 50)
```



```r
ggplot(df) + stat_ecdf(aes(x = x))
```



Drawbacks of histograms:

- Can be sensitive to bin size.
- Difficult to compare to each other.
- You are implicitly imposing a model on the data, and this can be a poor fit.
- Poor visualization/sensitive to noise when you don't have much data.

Q-Q Plots

Goal: Compare two univariate samples to each other

Quantile-Quantile (q-q) plot definition:

Suppose we have two sets of univariate measurements, $x_{(1)}, \dots, x_{(n)}$ and $y_{(1)}, \dots, y_{(m)}$, with $m \leq n$.

For each $i = 1, \dots, m$, plot the $(i - .5)/m$ quantile of the y dataset against the $(i - .5)/m$ quantile of the x dataset.

Note:

If $m = n$, then

- $q_x((i - .5)/m) = x_{(i)}$
- $q_y((i - .5)/m) = y_{(i)}$

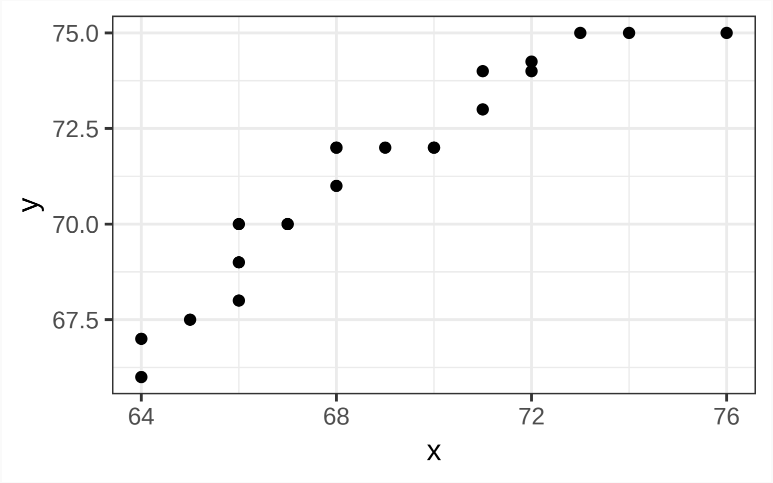
So in this case, we are simply plotting $x_{(i)}$ against $y_{(i)}$

q-q plot demonstration

There isn’t a nice way to do q-q plots in ggplot, but there is a function in base R that will return a data frame with the values we need to plot.

```r
Tenor1 = singer %>% subset(voice.part == "Tenor 1")
Bass2 = singer %>% subset(voice.part == "Bass 2")
qq.df = as.data.frame(qqplot(Tenor1$height, Bass2$height,
    plot.it = FALSE))
```

```r
ggplot(qq.df, aes(x = x, y = y)) + geom_point()
```



Q-Normal plots

Goal: Check how well a normal distribution approximates the data

Definition: Theoretical quantiles

Let

$$q_{\mu,\sigma}(f) = \{x : P(\mathcal{N}(\mu, \sigma^2) \leq x = f)\}$$

In words: the value x such that the probability that a $\mathcal{N}(\mu, \sigma^2)$ random variable takes value at most x is equal to f.

Note the analogy to data quantiles, $q_x(f)$ defined before.

Definition: Q-Normal plot

Let $x_{(1)}, \dots, x_{(n)}$ be our ordered data.

Recall that we defined the sample quantile function at values $f_i = (i - .5)/n$ as $q_x(f_i) = x_{(i)}$.

For each value f_i, $i = 1, \dots, n$, compute

- $q_x(f_i)$, the sample quantile at f_i
- $q_{0,1}(f_i)$, the theoretical quantile for $\mathcal{N}(0, 1)$ at f_i

A Q-normal plot shows sample quantiles on the y-axis and theoretical quantiles on the x-axis.

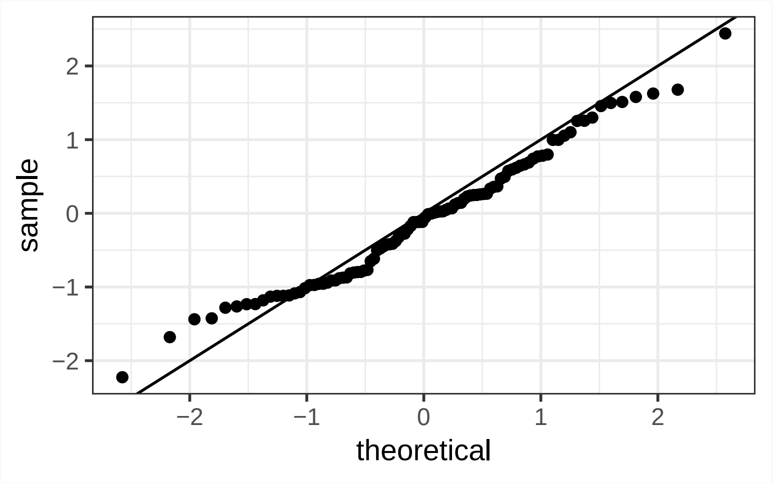
Properties:

- If the points lie along the line $y = x$, the distribution is well approximated by a $\mathcal{N}(0, 1)$ distribution.
- If the points lie along another straight line with intercept μ and slope σ, the distribution is well approximated by a $\mathcal{N}(\mu, \sigma^2)$ distribution.
- We can make the analogous plot to check how well our data is approximated by any distribution.
- Q-Q plot (plotting the quantiles of two datasets) can be thought of as comparing the quantiles of one sample to an estimate of the distribution of the other sample.

Q-Normal plot demonstration

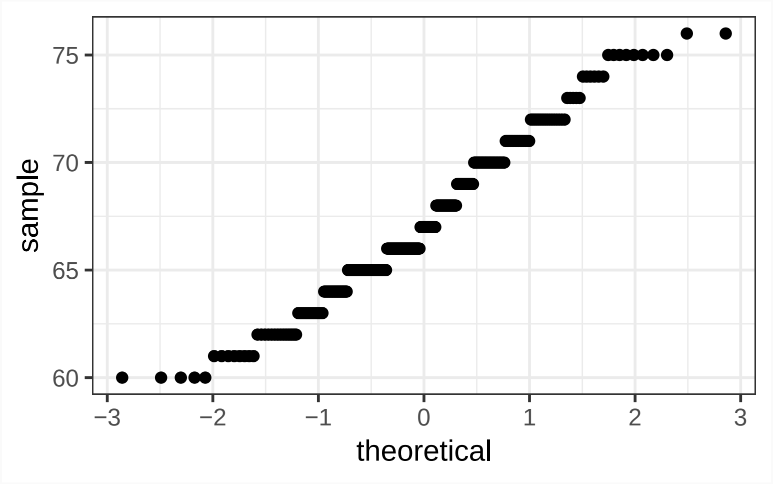
Let’s first see what a Q-normal plot looks like when the data really come from a normal distribution.

```r
ggplot(data.frame(x = rnorm(100))) +
    stat_qq(aes(sample = x)) +
    geom_abline(aes(slope = 1, intercept = 0))
```



Then we have a reference for how closely to the line the points should lie when we're looking at real data.

```
ggplot(singer) + stat_qq(aes(sample = height), distribution = qnorm)
```



Exercise: Make a q-uniform plot for simulated data and for the choral data. (Hint: look up the stat_qq documentation, see what you would have to switch out to change the reference from normal to uniform)

Boxplots

Goal: More parsimonious representation of a distribution.

Why might we want this? Shouldn't we always try to keep all the data?

Boxplot Statistics

Suppose our data is $x_1, \ldots, x_n$. We compute five statistics of the data:

- $q_x(.5)$, the median.
- $q_x(.25)$, the .25 quantile of $x_1, \ldots, x_n$ aka the lower quartile
- $q_x(.75)$, the .75 quantile of $x_1, \ldots, x_n$ aka the upper quartile
- Upper adjacent value (UAV), lower adjacent value (LAV)

$$\begin{aligned} r &= q_x(.75) - q_x(.25) \\ \text{UAV} &= \text{max}\{x_i : x_i \leq q_{.75} + 1.5r\} \\ \text{LAV} &= \text{min}\{x_i : x_i \geq q_{.25} - 1.5r\} \end{aligned}$$

Boxplot:

- Bar in the middle represents the median.
- Edges of box represent $q_x(.25)$ and $q_x(.75)$.
- Whiskers represent the UAV and LAV.
- Any values outside of the range [lAV, UAV] are referred to as *outside values* and are plotted individually.

Example: Theoretical boxplot values for normal data

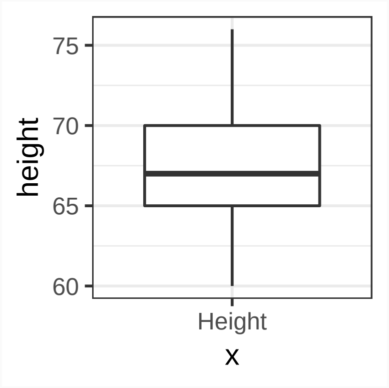
```
(iqr = qnorm(.75) * 2)

## [1] 1.34898

(uav = qnorm(.75) + 1.5 * iqr)

## [1] 2.697959

(prob_outside = pnorm(uav, lower.tail = FALSE) * 2)

## [1] 0.006976603
```

Boxplot demonstration

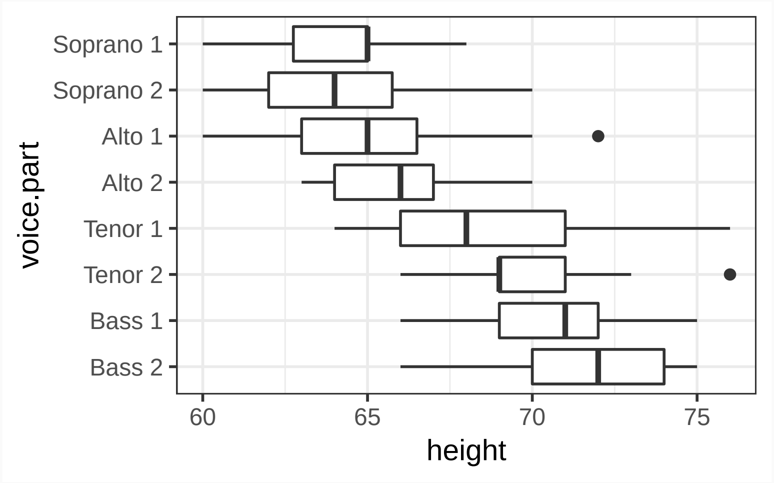
We can make a boxplot of just one variable, but only by hacking the syntax a bit.

The primary purpose of a boxplot is really to compare multiple distributions.

```
ggplot(singer, aes(x = "Height", y = height)) +
    geom_boxplot()
```



```r
ggplot(singer, aes(x = voice.part, y = height)) +
    geom_boxplot() +
    coord_flip()
```



Wrapping up

Which representation do you like the best?

Are some representations better in some situations?

Cleveland really dislikes histograms. Do you agree?

My advice:

- If you have just a couple of numbers, don't use a histogram or a density estimate, use an ecdf, quantile plot, or even stem-and-leaf plot.
- If you have a lot of data and simply want to summarize a single distribution, a histogram or other sort of density estimate is fine.
- Try to learn how to read ecdf or quantile plots.

Homework 0 is due tomorrow.

Homework 1 assigned, due next Friday, also covers some of the material that we'll be talking about on Tuesday.