Class demo of histograms and barplots

Dan Yurovsky

10/5/2016

Load data managemenet and plotting libraries. And suppress package loading messages to have a cleaner doc.

```
library(dplyr)
library(ggplot2)
library(tidyr)
```

First, let's load in the data from our in-class survey

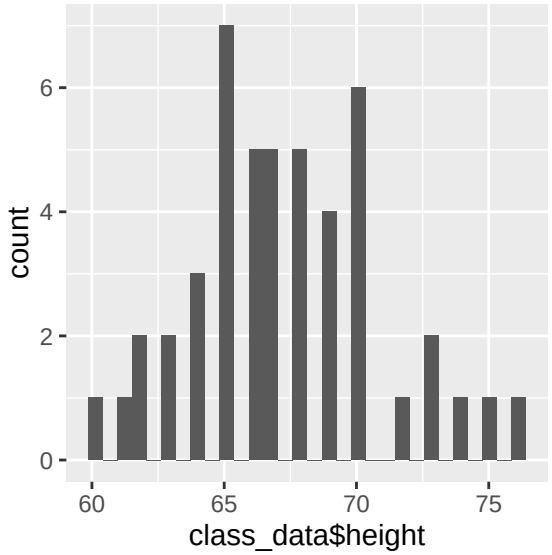
```
class_data <- read.csv('class_data.csv') #read the data from disk

head(class_data) #show the first few rows so we can see how the data are sturctured
```

```
##   height birth siblings
## 1     73     2        2
## 2     70     5        1
## 3     67    10        1
## 4     65     8        1
## 5     68     7        0
## 6     68    10        1
```

Use `qplot` to make a separate histogram for each of the three things we measured. It will cleverly firgure out that I want a histogram without me even telling it!. I also specify how large I want the figures to be (`fig.width` and `fig.height`).

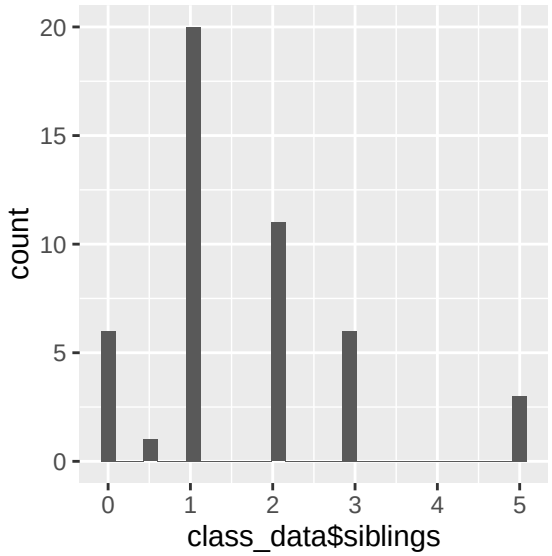
```
qplot(class_data$height)
```

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



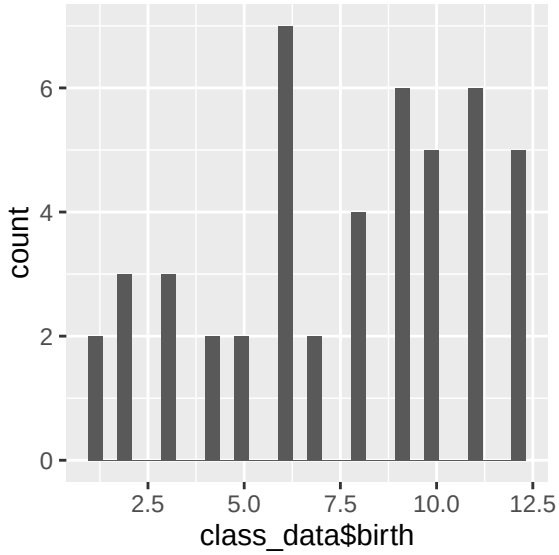
```
qplot(class_data$siblings)
```

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




```
qplot(class_data$birth)
```

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



The current `class_data` data frame is in what's called *wide format* – each row has multiple observations for a single person. To analyze it, I want it in *long format*, where each row has only a single data point.

Munge the data (process it to get it into a tidy format). I'm going to use the `gather` function from the `tidyr` package. Right now, a single row has a value for `siblings`, `height`, and `birth`. I want it to only have a single `value` column and another column `measurement` that indicates which of three variables the value corresponds to.

So what I do is I make a new column called `person` which just gives a unique number to each row of the dataset (from 1 to the number of rows in the dataframe). Then I `gather` together all of the other measures.

```
class_data_long <- class_data %>%
  mutate(person = 1:nrow(class_data)) %>% #make a unique identifier for each row
  gather(measure, value, -person) #gather all of the columns except for person

# now it's in long form!
head(class_data_long)
```

```
##   person measure value
## 1      1  height    73
## 2      2  height    70
## 3      3  height    67
## 4      4  height    65
## 5      5  height    68
## 6      6  height    68
```

Now I want to compute some descriptive statistics–the mean and median. I'll use the `group_by` function from the `dplyr` package to let `R` know that I want to call these functions separately for each `measure` in my dataset. Then I'll use `summarize`, which takes in all of the data for each group, and reduces it down to a single number using the functions I ask for. Here, `mean` and `median`.

```
descriptives <- class_data_long %>%
  group_by(measure) %>%
  summarize(mean = mean(value),
            median = median(value))

#let's print out these descriptives
descriptives
```

```
## # A tibble: 3 × 3
##    measure      mean median
##      <chr>     <dbl>  <dbl>
## 1    birth  7.510638      8
## 2   height 67.276596     67
## 3 siblings  1.606383      1
```

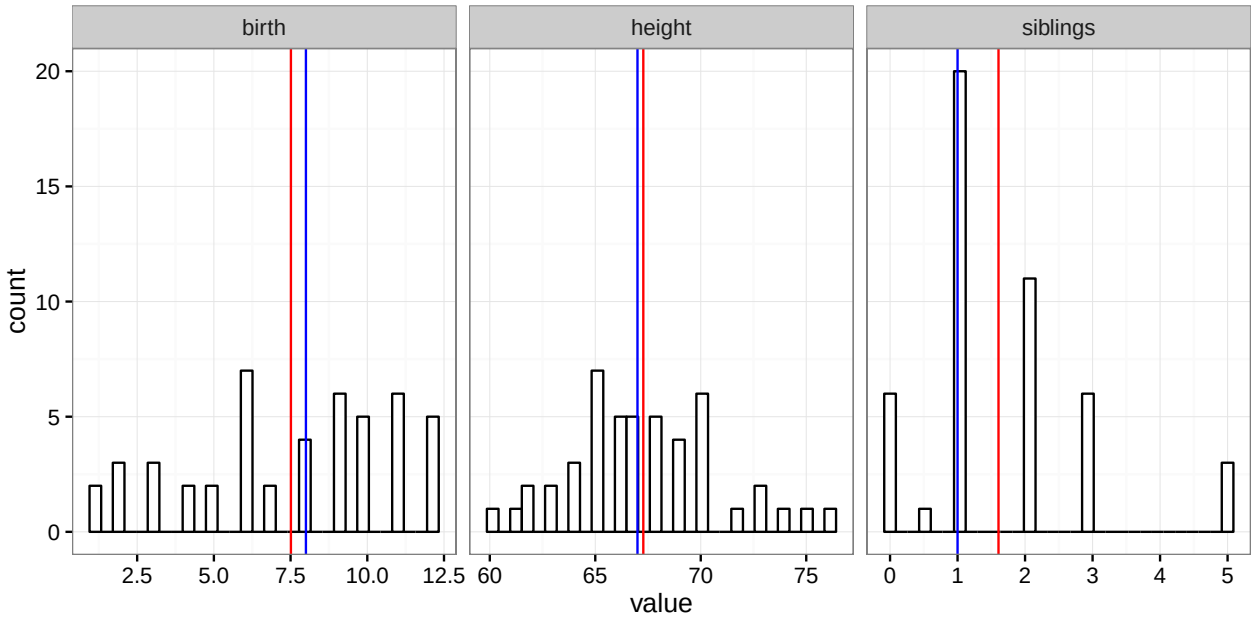
Ok, let's plot all of the histograms together! And let's also put a line on each indicating the position of the mean (in red), and the median (in blue). That way, we can see how skew in our distributions changes the relationship between them.

Also, I'm going to use `theme_bw` to change how the plots look–e.g., removing that horrible gray background that makes them hard to read.

I'm going to the `ggplot` function, which is like `qplot` without all of the smart default assumptions. I want to have more control over my plot, adding things like multiple `facets` and two different geoms: a *histogram* and a *vertical line*

```
ggplot(class_data_long, aes(x = value)) +
  facet_wrap(~ measure, scales = "free_x") +
  geom_histogram(fill = 'white', color = 'black') +
  geom_vline(aes(xintercept = mean), data = descriptives, color = "red") +
  geom_vline(aes(xintercept = median), data = descriptives, color = "blue") +
  theme_bw()
```

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



Now lets make boxplots so we can see how they compare to the histograms. They still show a lot of the same information–skew, variability, center–but they also compress a lot of data about exactly how many people had exactly what height.

```
ggplot(class_data_long, aes(x = measure, y = value)) +
  geom_boxplot() +
  theme_bw()
```