Visualizing Distributions - Word Doc

Liz Carey

5/14/2019

### Set up

Before we begin, install and load necessary packages to run this notebook.

library(tidyverse) #includes functions to run code in this notebook and  
library(gapminder) #includes dataset we'll be plotting

## What is a histogram?

A histogram is a visualization of the distribution of numerical data. It’s created by “binning” or bucketing the numerical values into intervals.

### When to use a histogram:

* Check the distribution (shape) of your data
* Check for skewness (the tendency of observations to fall more on the low end or the high end of the x-axis)
* Check the data for outliers

### How is a histogram different from a bar chart?

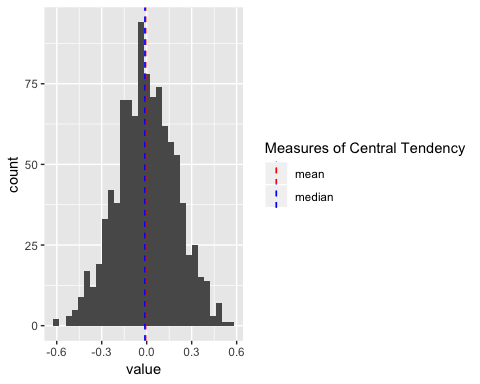
* A bar chart compares two variables while **a histogram only relates one.**
* A bar chart plots categorical data while **a histogram plots numerical binned data**
* A bar chart can be reordered while **a histogram cannot be reordered**
* A bar chart uses the length of a bar to display value while **a histogram uses the area of the bar**
* The width of the bars is the same in a bar chart whereas **the width need not be the same in a histogram**
* Visually, in a bar chart, there are gaps between the bars, whereas **in a histogram there are no gaps between the bars**

## Histogram shapes

There are different shapes that your histogram can take. Each shape says something about your data set.

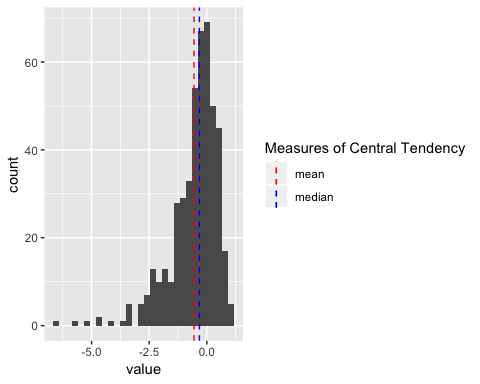
### Normal Distribution

A normal distribution takes the shape of a bell-shaped curve. Most of the data are clustered around the center so that the median and mean will be in the center of the curve and are approximately the same. The distribution is symmetric meaning that about half of the data is to the left of the mean, while the other half is to the right of the mean.



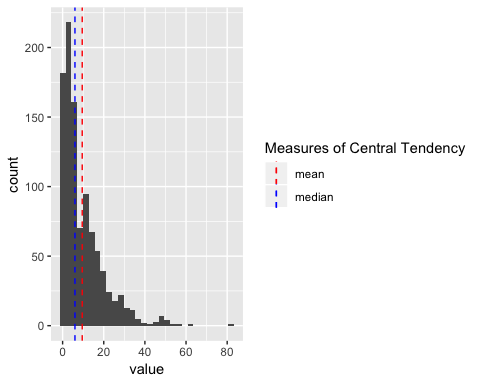
### Left skew

A left skewed shape occurs when data are clustered around a large value and mostly fall to the right of the mean (and typically the median, too). Left skewed data is also often referred to as negative skewed with a long left tail.



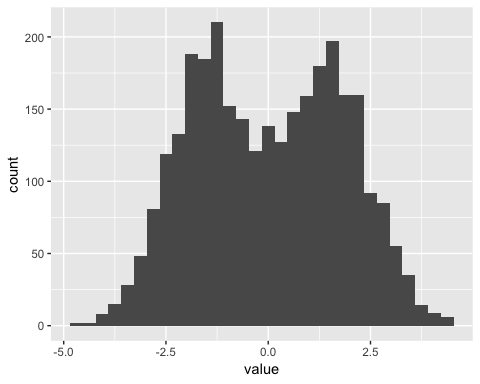
### Right skew

A right skew shape occurs when the data are clustered around a small value and most of the data (including the median) fall to the left of the mean. Right skewed data is also often referred to as positive skewed with a long right tail.



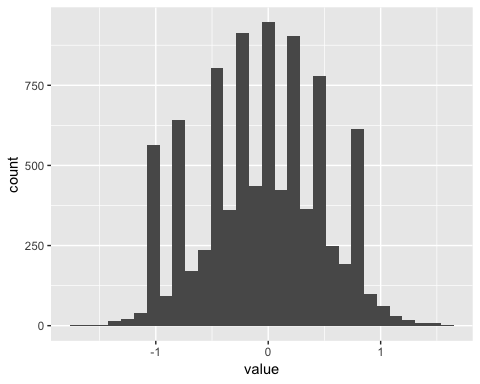
### Bimodal distribution

A Bimodal distribution shows two peaks of data. The peaks shows that there are two modes (two values that occur more frequently than any other) for the value you are plotting. These two peaks typically arise when your data set contains two distinct groups.  
When you see a bimodal distribution, consider breaking your data set into different groups to separate the peaks.



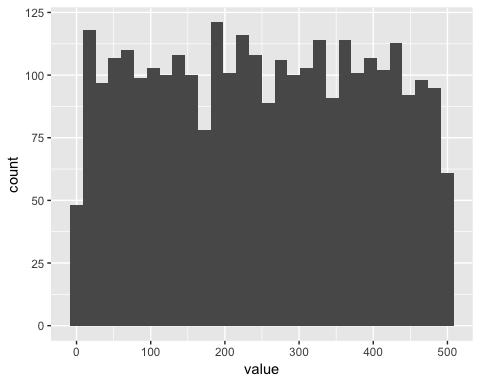
### Comb distribution

The Comb distribution alternates between high and low peaks. A comb shape can be caused by rounding numbers. Common examples of rounding that create comb distributions include: rounding temperatures, height, or age to the nearest whole number.



### Uniform distribution

A uniform distribution shows that the data are equally distributed: there are no clear peaks in the data.



## Plotting a histogram

#### Print the data

By printing the data, we get a feel for what we’ll be analyzing. From the printed table, we see the data set contains the life expectancy, population, and GDP per capita of various countries separated by year.

gapminder #name of data set from the gapminder package

## # A tibble: 1,704 x 6  
## country continent year lifeExp pop gdpPercap  
## <fct> <fct> <int> <dbl> <int> <dbl>  
## 1 Afghanistan Asia 1952 28.8 8425333 779.  
## 2 Afghanistan Asia 1957 30.3 9240934 821.  
## 3 Afghanistan Asia 1962 32.0 10267083 853.  
## 4 Afghanistan Asia 1967 34.0 11537966 836.  
## 5 Afghanistan Asia 1972 36.1 13079460 740.  
## 6 Afghanistan Asia 1977 38.4 14880372 786.  
## 7 Afghanistan Asia 1982 39.9 12881816 978.  
## 8 Afghanistan Asia 1987 40.8 13867957 852.  
## 9 Afghanistan Asia 1992 41.7 16317921 649.  
## 10 Afghanistan Asia 1997 41.8 22227415 635.  
## # ... with 1,694 more rows

#### What’s the average life expectancy?

After getting a brief overview of our data set, now we want to know, what’s the the average life expectancy in our data set?

We can calculate the average life expectancy by using the summarize and mean function:

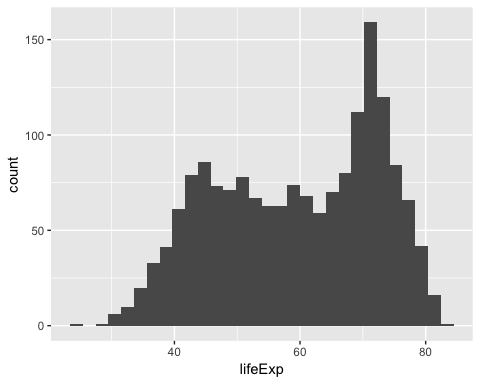
gapminder %>% summarize(mean = mean(lifeExp))

## # A tibble: 1 x 1  
## mean  
## <dbl>  
## 1 59.5

But we know that averages can be susceptible to outliers. So, let’s take a look at our underlying data.

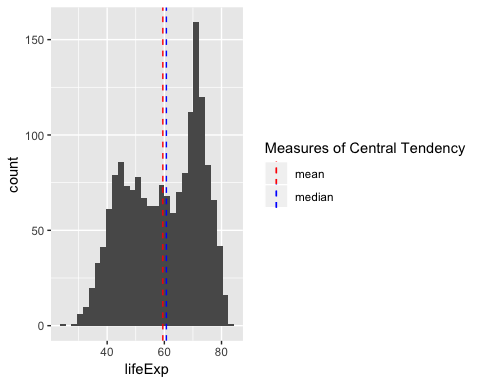
To do this, let’s plot the distribution of life expectancies in the gapminder data set.

ggplot(gapminder, aes(lifeExp)) +   
 geom\_histogram()



Now, let’s plot the histogram with measures of central tendency: median and mean.

#create a seperate data tabe calculating the mean and median of the data set  
measures\_of\_ct <- gapminder %>% summarise(mean = mean(lifeExp),   
 median = median(lifeExp))  
  
  
ggplot(gapminder, aes(lifeExp)) +   
 geom\_histogram() +   
 # add a vertical line of the mean  
 geom\_vline(data = measures\_of\_ct,   
 aes(xintercept = mean, color ="mean"),  
 show.legend = TRUE,  
 linetype = "dashed") +   
 # add a vertical line of the median  
 geom\_vline(data = measures\_of\_ct,  
 aes(xintercept = median, color = "median"),  
 linetype = "dashed",   
 show.legend = TRUE) +  
 scale\_color\_manual(  
 #Title the legend  
 name = "Measures of Central Tendency",   
 #Assign colors to the mean and median lines   
 values = c(median = "blue", mean = "red"))



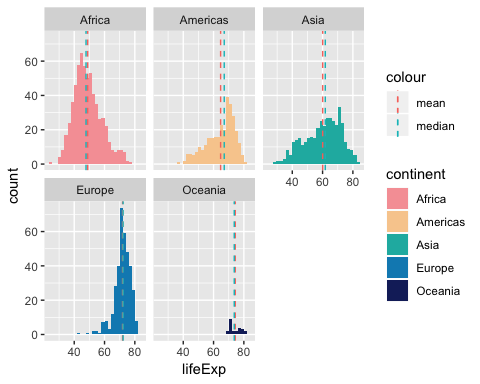
From this plot, we can see the life expectancies center around 60 years of age.

But, we also notice the shape of this histogram is bimodal. This bimodal shape tells us that we’re likely looking at two distinct groups (or more).

#### Divide the groups, separate the data

Since we suspect we’re looking at multiple groups from the bimodal distribution, let’s separate the histograms by one of the variables in our data set: continent.

#create a seperate data tabe calculating the mean and median of each continent  
measures\_of\_ct\_by\_continent <- gapminder %>%   
 group\_by(continent) %>%   
 summarize(mean = mean(lifeExp),   
 median = median(lifeExp))  
  
#plot the distribution of life expectancies separated by continent  
ggplot(gapminder, aes(lifeExp, fill = continent)) +  
 geom\_histogram() +   
 geom\_vline(data = measures\_of\_ct\_by\_continent,   
 aes(xintercept = mean, color = "mean"),   
 linetype = "dashed") +  
 geom\_vline(data = measures\_of\_ct\_by\_continent,   
 aes(xintercept = median, color = "median"),   
 linetype = "dashed") +  
 scale\_fill\_manual(values = set\_continent\_color(continents, Pamplemousse, 1)) +  
 facet\_wrap(~continent)

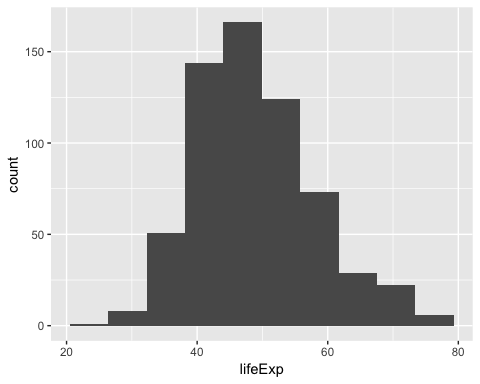


Now we can see that Africa and Europe have two very distinct averages of life expectancy and were likely driving the bimodal distribution.

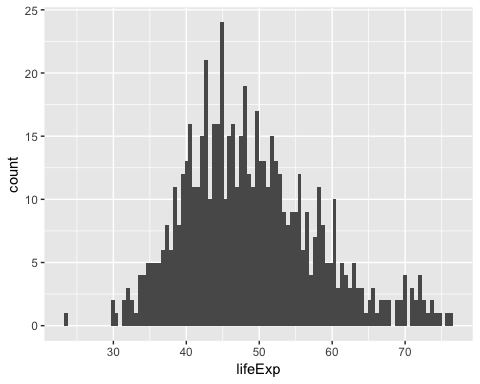
### Adjusting the bin width

Adjusting the binwdith of your data can sometimes change the interpretation of your histogram. Here we look at the difference between the life expectancies in Africa with 10 vs. 100 bins

gapminder %>%   
 filter(continent == "Africa") %>%   
 ggplot(., aes(lifeExp)) +  
 geom\_histogram(bins = 10)



gapminder %>%   
 filter(continent == "Africa") %>%   
 ggplot(., aes(lifeExp)) +  
 geom\_histogram(bins = 100)

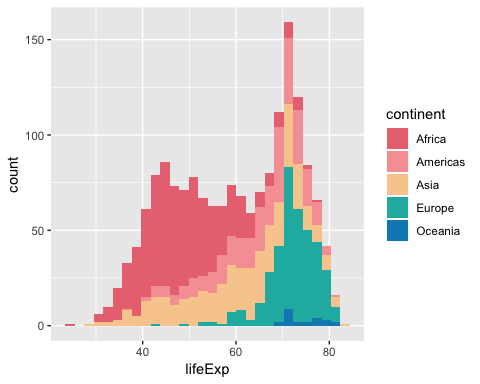


In this case, we see a comb distribution in the histogram with more bins, likely due to rounding of ages to the third decimal.

### Changing the colors of your histogram

Let’s say you want to customize your histogram. One way is to manually hard code each of the variables to the colors you want.

ggplot(gapminder, aes(lifeExp, fill = continent)) +  
 geom\_histogram() +   
 scale\_fill\_manual(values = c("Africa" = "#EA7580",   
 "Americas" = "#F6A1A5",   
 "Asia" = "#F8CD9C",   
 "Europe"= "#1BB6AF",   
 "Oceania" = "#088BBE"))



Alternatively, if there is a color palette you want to use, you can create a function to populate colors for your plot.

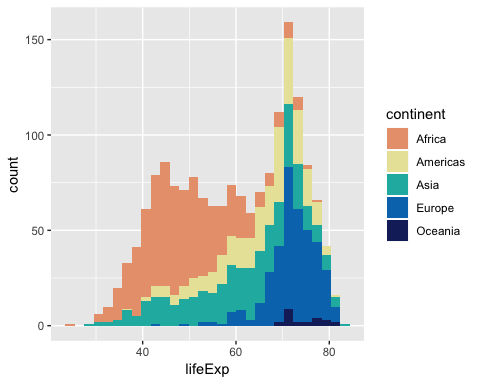
Below is an example using the LaCroixColoR package.

#store continents in a list  
continents <- c("Africa", "Americas", "Asia", "Europe", "Oceania")  
  
#create a function that will assign the continent names to a color. The function takes three inputs: the list of variables to be colored (e.g. continent names), the type of laCroix flavor palette, and the start of the index. change the index\_start to 0 or 1 to get variations of a laCroix flavor palette.  
set\_continent\_color <- function(continent\_names, lacroix\_color, index\_start) {  
   
 #create empty vector to store values  
 output <- vector("character", length(continent\_names))  
   
 #loop over the length of the list of continent names, and assign a color to each continent  
 for (i in seq\_along(continent\_names)) {  
 output[[i]] <- lacroix\_color[[index\_start + i]]  
 }  
 output   
   
}

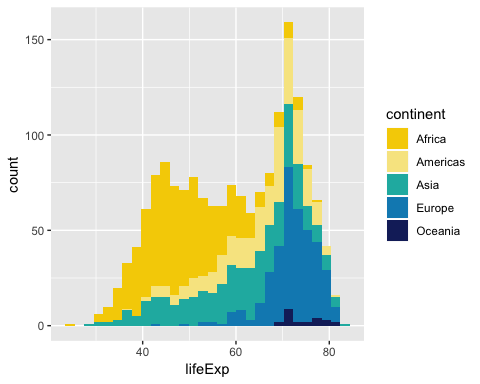
#### Plot the histogram

Set custom colors using the set\_continent\_color function in the scale\_fill\_manual function.

ggplot(gapminder, aes(lifeExp, fill = continent)) +  
 geom\_histogram() +   
 scale\_fill\_manual(values = set\_continent\_color(continents, PeachPear, 1))



ggplot(gapminder, aes(lifeExp, fill = continent)) +  
 geom\_histogram() +   
 scale\_fill\_manual(values = set\_continent\_color(continents, Lemon, 1))



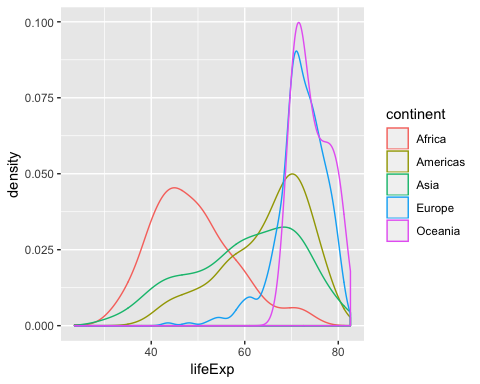
## Density Plots

A density plot is a smoothed version of the histogram. It uses kernel smoothing to plot values so it removed the “noise” from a typical histogram visualization.

### Plot life expectancies

The below is code to produce a density plot. Notice the y-axis is labeled “density”, not “count” like the histogram plots.

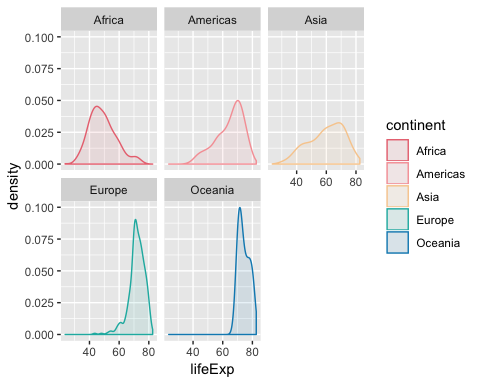
ggplot(gapminder,   
 aes(lifeExp, colour = continent)) +  
 geom\_density()



### Customizing your density plot

The below code shows you how to choose custom colors and split each continent plot.

# Store colors you want for each continent  
density\_plot\_colors <- c( "Africa" = "#EA7580",   
 "Americas" = "#F6A1A5",   
 "Asia" = "#F8CD9C",   
 "Europe"= "#1BB6AF",   
 "Oceania" = "#088BBE")  
  
ggplot(gapminder,   
 aes(lifeExp, fill = continent, colour = continent)) +  
 geom\_density(alpha = 0.1) +   
 scale\_color\_manual(values = density\_plot\_colors) +   
 scale\_fill\_manual(values = density\_plot\_colors) +  
 facet\_wrap(~continent)



## Histogram mistakes to avoid:

* **Minimize clutter and overlap**
* **Only plot numerical data**

## Histograms in Survey Analysis

How does knowing how to make histograms help you?

Let’s say you fielded a survey to 5,000 Uber riders. The survey received a 10% response rate, so a total of 500 respondents.

Is 10% a good response rate?

Well, let’s see if the respondents are *different* from those who didn’t respond to the survey. A quick lightweight way to visualize this is by plotting your data set.

Visualizing your data set should be one of the first steps you complete when conducting exploratory analysis.

Meaningful data we have on Uber riders in the sample include:

* days since they first signed up (tenure) and
* lifetime trips completed (lt\_trips)

set.seed(300)  
  
#Generate a data set with fake sample respondents  
sample <- tibble(  
 respondent\_id = 1:5000,   
 tenure = rnorm(n = 5000, mean = 900, sd=200),   
 lt\_trips = (c(runif(500, min = 1, max = 50), runif(1000, min = 51, max = 100), runif(3200, min = 101, max = 150), runif(300, min = 151, max = 200)))  
)  
  
#randomly sample 500 respondents from the sample data set and label them as "respondents"  
respondents <- sample\_n(sample, 500) %>%   
 mutate(respondent\_type = "respondent")

Let’s label the non-respondents in the sample data set in a column named respondent\_type.

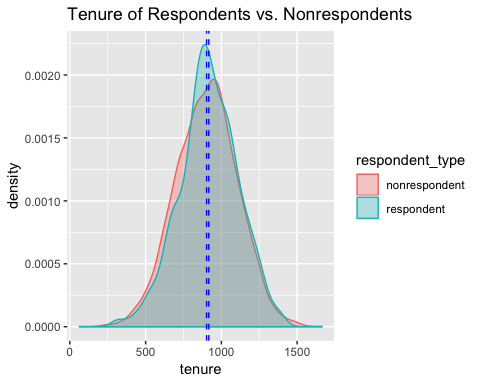
nonrespondents <- sample %>%   
 anti\_join(respondents, by = c("respondent\_id" = "respondent\_id")) %>%   
 mutate(respondent\_type = "nonrespondent")  
  
sample\_recoded <- respondents %>%   
 bind\_rows(nonrespondents)

Let’s calculate the average tenure and lifetime trips of the respondents and non-respondents

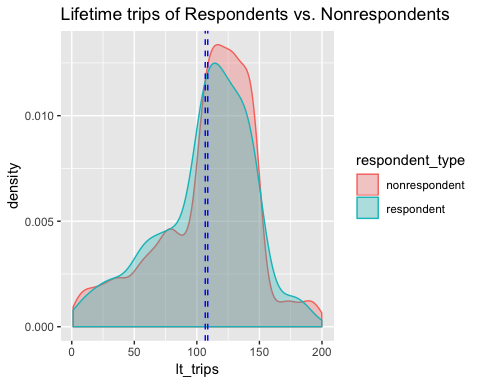
sample\_recoded\_measures\_of\_ct <- sample\_recoded %>%   
 group\_by(respondent\_type) %>%   
 summarize(mean\_tenure = mean(tenure),   
 mean\_lt\_trips = mean(lt\_trips)   
 )

Now, let’s plot the average tenure and lifetime trips.

ggplot(sample\_recoded,   
 aes(tenure,   
 fill = respondent\_type,   
 colour = respondent\_type)) +  
 geom\_density(alpha = 0.3) +   
 geom\_vline(data = sample\_recoded\_measures\_of\_ct,   
 aes(xintercept = mean\_tenure),   
 linetype = "dashed", color = "blue") +   
 labs(title = "Tenure of Respondents vs. Nonrespondents")



ggplot(sample\_recoded,   
 aes(lt\_trips,   
 fill = respondent\_type,   
 colour = respondent\_type)) +  
 geom\_density(alpha = 0.3) +   
 geom\_vline(data = sample\_recoded\_measures\_of\_ct,   
 aes(xintercept = mean\_lt\_trips),   
 linetype = "dashed", color = "blue") +   
 labs(title = "Lifetime trips of Respondents vs. Nonrespondents")



From this visualization, we can see the respondents and non-respondents are likely similar groups.  
To further validate this, we can compare the means of the two groups. Check out Ryan’s workbook on Confidence Intervals to learn how to test means across two groups.

## But I’m still uncomfortable with R…

OK, we heard you. There’s a ramp up to R. If you’re still uncomfortable, you can view a histogram example using the same gapminder data set in Google sheets [here](https://docs.google.com/spreadsheets/d/1KC1Cx1nHfNrVRSom1QXc42q8S_VSo9KmJA9nrZx3Ils/edit?usp=sharing) and view a walk through on histograms on the Google support page [here](https://support.google.com/docs/answer/9146867?hl=en)

## Resources

Read more about different visualization types and how to build them from the following resources:

* [Data to Viz](https://www.data-to-viz.com/)  
  *From Data to Viz leads you to the most appropriate graph for your data. It links to the code to build it and lists common caveats you should avoid.*
* [Smoothed Density Estimates - ggplot2](https://ggplot2.tidyverse.org/reference/geom_density.html)  
  *Documentation on how to use the geom\_density and stat\_density plotting functions for ggplot2*
* [LaCroixColoR - LaCroix Color Palettes for R](https://github.com/johannesbjork/LaCroixColoR)  
  *Choose from 20 different LaCroix flavors to color your plots.*