SPEC Lab R Resources: Basic Data Visualization with ggplot2

Alix Ziff, Gaea Morales, and Zachary Johnson based on earlier materials by Therese Anders

Summer 2020, Version 3: June 9

ggplot2

This workshop provides an introduction to data visualization in `R` using the `ggplot2` package. We will first introduce univariate graphical data summaries, parameters that control the appearance of the plot, the visualization of data across multiple groups, and how to save plots.

We will use data from the World Development Indicators. Specifically, we look at different indicators for the energy consumption of all countries in the WDI dataset for the past 25 years.

Review: Reading the data into `R`

We start by a) setting the working directory, b) installing the `ggplot2` package, and c) loading the `ggplot2` package into the environment.

```
setwd("~/Google Drive File Stream/My Drive/SPEC Summer 2020/Training Data Science/0 Training Data")
getwd()
```

```
## [1] "/Volumes/GoogleDrive/.shortcut-targets-by-id/1TXk1G5ASoQ-vD7rs4E5dMtNk1W-d1k6m/SPEC Summer 2020/
```

```
#install.packages("ggplot2")
library(ggplot2)
```

Now, we can read the file `wdi_cleaned.csv`. Remember, that in order to read `.csv` files with the `read.csv()` function, you need to first load the `foreign` package.

```
library(foreign)
dat <- read.csv("wdi_cleaned_part1.csv", #Also temporarily commented out so I can compile
                stringsAsFactors = F)
```

Let's take a look...

The dataset contains 5425 observations on 5 variables. We can tell `R` to give us an overview of the data using the `str()` function. We can also take a look at the dataset in a spreadsheet format with the `View()` function.

```
str(dat)
```

```
## 'data.frame':    5425 obs. of  5 variables:
##  $ country           : chr  "Afghanistan" "Albania" "Algeria" "American Samoa" ...
##  $ year              : int  1992 1992 1992 1992 1992 1992 1992 1992 1992 1992 ...
##  $ electricity_pop   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ energyuse_pop     : num  NA 418 884 NA NA ...
##  $ renewable_energyuse: num  62.24 46.03 0.29 NA NA ...
```

The dataset contains the following variables:

- `year`: Variable coding the year of observation.
- `country`: Variable coding the country of observation.
- `electricity_pop`: Access to electricity (% of population).
- `energyuse_pop`: Energy use (kg of oil equivalent per capita).
- `renewable_energyuse`: Renewable energy consumption (% of total final energy consumption).

ggplot2 package

The `ggplot2` package was developed by Hadley Wickham based on Leland Wilkinson's "grammar of graphics" principles. According to the "grammar of graphics," you can create each graph from the following components: "a data set, a set of geoms–visual marks that represent data points, and a coordinate system" (Data Visualization with ggplot2 Cheat Sheet.

For most applications, the code to produce a graph in `ggplot2` is roughly structured as follows:

```
ggplot(data = , aes(x = , y = , color = , linetype = )) +

geom() +

[other graphical parameters, e.g. title, color schemes, background]
```

- `ggplot()`: Function to initiate a graph in `ggplot2`.
- `data`: Specifies the data frame from which the plot is produced.
- `aes()`: Specifies aesthetic mappings that describe how variables are mapped to the visual properties of the graph. The minimum value that needs to be specified (for univariate data visualization) is the `x` parameter, where `x` specifies the variable to be plotted on the x-axis. Analogously, the `y` parameter specifies the variable to be plotted on the y-axis. Other examples include the `color` parameter, which specifies the variable to be onto different colors, or the `linetype` parameter, which specifies the variable to be mapped onto different line types in case of line graphs.
- `geom()`: Specifies the type of plot to use. There are many different geoms ("geometric objects") to be specified with the `geom()` layer. Some of the most common ones include `geom_point()` for scatterplots, `geom_line()` for line graphs, `geom_boxplot()` for Boxplots, `geom_bar()` for bar plots for discrete data, and `geom_histogram()` for continuous data.

For an overview of the most important functions and geoms available through `ggplot2`, see the `ggplot2` cheat sheet.

Univariate visualizations (one variable visuals)

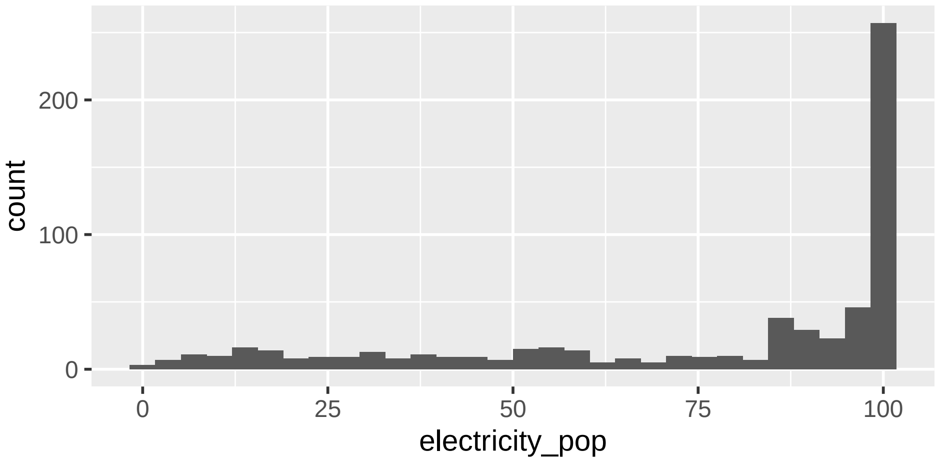
Histograms

Histograms graph the distribution of continuous variables. In this first example, we graph the distribution of the variable `electricity_pop`. Note that because `electricity_pop` specifies a percentage, its value is bound between 0 and 100.

```
summary(dat$electricity_pop) #looking at the summary statistics (mean, minimum, NAs, etc.) for the vari
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   53.44   92.68   75.74  100.00  100.00    4789
```

```
ggplot(dat, aes(electricity_pop)) +
  geom_histogram() #making a histogram of the electricity_pop variable from the dataset dat
```



Question 1 Can you make sense of this graph? What is plotted on the x-axis? What is plotted on the y-axis? What specifies the width of each bar? What specifies the height of each bar?

A histogram plots the distribution of a variable. The x-axis specifies the values of the variable. The y-axis specifies the number of observations for each value (or group of values) of the variable. The width of the bar specifies which values of the variable are grouped into one bin. The height of the bar specifies the number of observations in each bin.

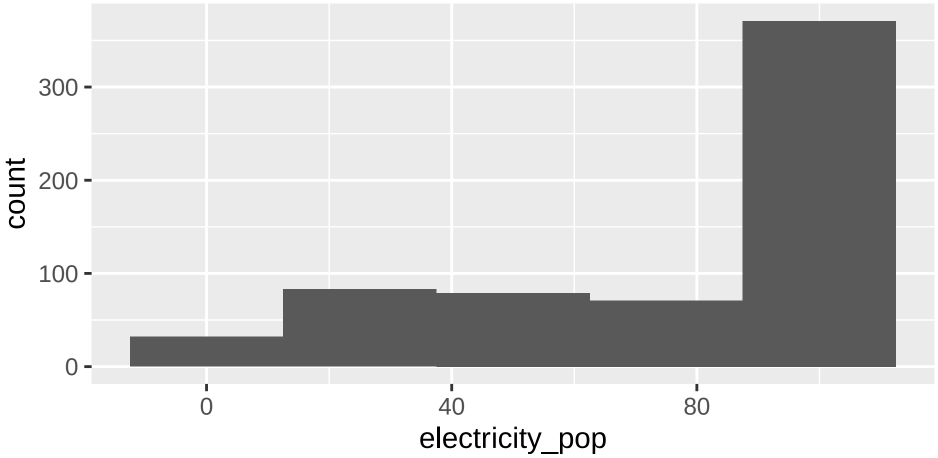
Question 2 Which conclusions do you draw from the histogram above about the distribution of the availability of electricity in the world?

The distribution is not normal (i.e. not a bell curve). It is skewed to the left. There are a lot more observations at the upper than the lower end of the scale, i.e. more country-years have high levels of availability of electricity than low levels.

Adjusting the number of bins

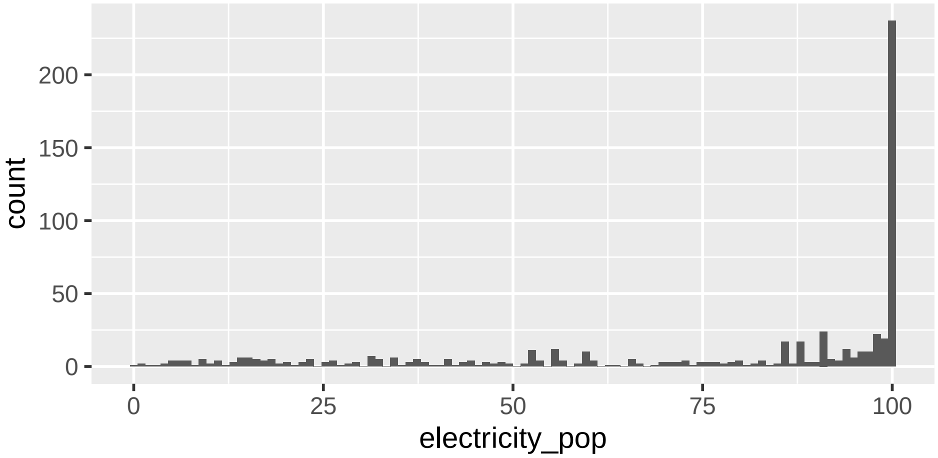
The default number of bins is 30, which means that the entire range of the variable (here 0 to 100) is split into 30 equally spaced bins. We can change the number of bins manually. In this example, since the variable is bound between 0 and 100, specifying `bins = 5` means that approximately values between 0-19 are grouped into one bin, values between 20-39 and so on.

```
ggplot(dat, aes(electricity_pop)) +
  geom_histogram(bins = 5) #taking our same histogram from above but using geom_histogram we adjust the
```



Question 3 How does the graph change if we specify `bins = 100`?

```
ggplot(dat, aes(electricity_pop)) +
  geom_histogram(bins = 100)
```



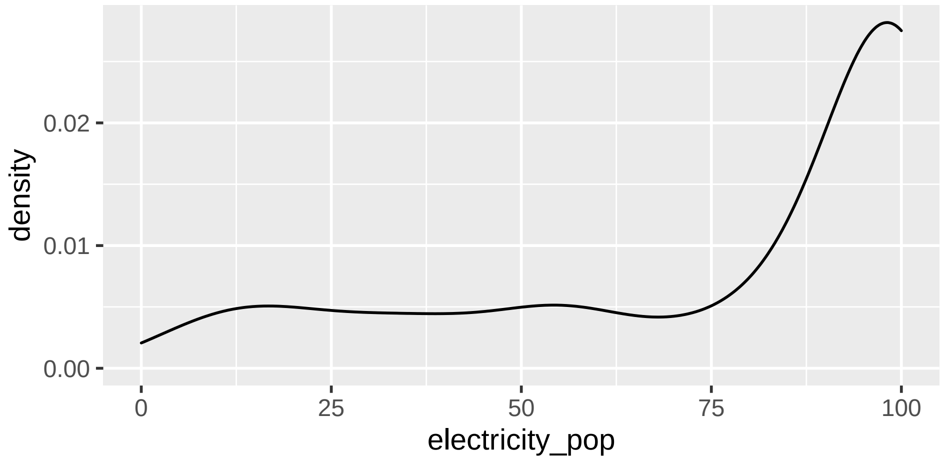
If we change the number of bins to 100 then, the histogram has 100 separate segments.

Density plots

We saw that the shape of the distribution is highly influenced by how many bins we specify. If we specify too few bins, we run the risk of masking a lot of variation within the bins. If we specify too many bins, we trade parsimony for detail–which might make it harder to draw conclusions about the overall distribution of the variable of interest from the graph.

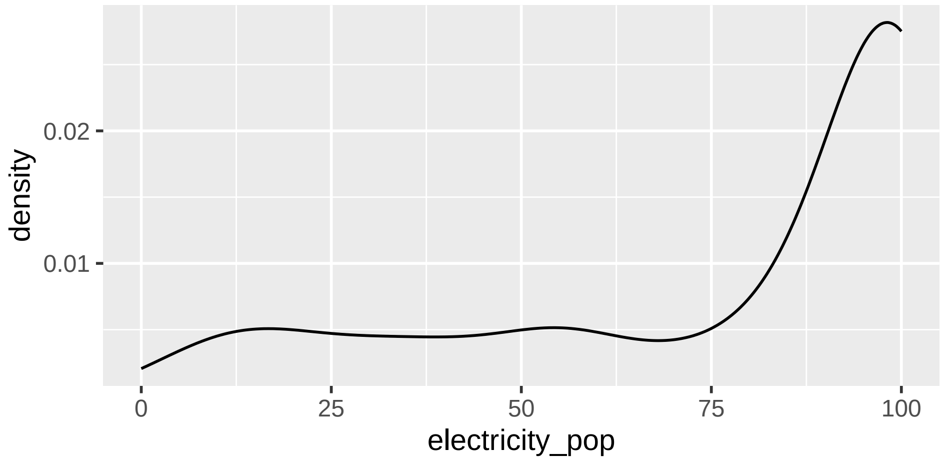
Density plots are continuous alternatives to histograms that do not rely on bins. We will cover details about the mechanics behind density plots and their estimation here. Just know that we can interpret the height of the density curve in a similar way that we interpreted the height of the bars in a histogram: The higher the curve, the more observations we have at that specific value of the variable of interest. In this first example, we use the `geom_density()` function to create the density plot.

```r
ggplot(dat, aes(electricity_pop)) +
  geom_density() #making a density plot of the electricity_pop variable from the dataset dat
```



If you do not want the density graph to be plotted as a closed polygon, you can instead use the `geom_line()` geometric object function with the `stat = "density"` parameter.

```r
ggplot(dat, aes(x = electricity_pop)) +
  geom_line(stat = "density")
```


density
electricity_pop