Data Basics

Lt Col Ken Horton Professor Bradley Warner

11 May, 2020

Objectives

1) Define and use properly in context all new terminology to include but not limited to case, observational unit, variables, data frame, associated variables, independent, and discrete and continuous variables.
2) Identify and define the different types of variables.
3) From reading a study, explain the research question.
4) Create a scatterplot in `R` and determine the association of two numerical variables from the plot.

Data basics

Effective presentation and description of data is a first step in most analyses. This lesson introduces one structure for organizing data as well as some terminology that will be used throughout this book.

Observations, variables, and data matrices

For reference we will be using a data set concerning 50 emails received in 2012. These observations will be referred to as the `email50` data set, and they are a random sample from a larger data set. This data is in the `openintro` package so let's load our packages.

```
library(tidyverse)
library(mosaic)
library(openintro)
library(knitr)
```

The table below shows 5 rows of the `email50` data set concerning.

Each row in the table represents a single email or **case**.[1] The columns represent characteristics, called **variables**, for each of the emails. For example, the first row represents email 1, which is not spam, contains 21,705 characters, 551 line breaks, is written in HTML format, and contains only small numbers.

	spam	num_char	line_breaks	format	number
1	0	21.705	551	1	small
2	0	7.011	183	1	big
3	1	0.631	28	0	none
50	0	15.829	242	1	small

[1] A case is also sometimes called a **unit of observation** or an **observational unit**

Let's look at the first 10 rows of data from `email50`. Remember to ask the two questions:
What do we want `R` *to do?* and
What must we give `R` *for it to do this?*

We want the first 10 rows so we use `head` and `R` needs the data object and the number of rows. The data object is called `email50` and is accessible once the `openintro` package is loaded.

```
head(email50,n=10)
```

```
##    spam to_multiple from cc sent_email                time image attach dollar
## 1     0           0    1  0          1 2012-01-04 06:19:16     0      0      0
## 2     0           0    1  0          0 2012-02-16 13:10:06     0      0      0
## 3     1           0    1  4          0 2012-01-04 08:36:23     0      2      0
## 4     0           0    1  0          0 2012-01-04 10:49:52     0      0      0
## 5     0           0    1  0          0 2012-01-27 02:34:45     0      0      9
## 6     0           0    1  0          0 2012-01-17 10:31:57     0      0      0
## 7     0           0    1  0          0 2012-03-17 22:18:55     0      0      0
## 8     0           0    1  0          1 2012-03-31 07:58:56     0      0      0
## 9     0           0    1  1          1 2012-01-10 18:57:54     0      0      0
## 10    0           0    1  0          0 2012-01-07 12:29:16     0      0     23
##    winner inherit viagra password num_char line_breaks format re_subj
## 1      no       0      0        0   21.705         551      1       1
## 2      no       0      0        0    7.011         183      1       0
## 3      no       0      0        0    0.631          28      0       0
## 4      no       0      0        0    2.454          61      0       0
## 5      no       0      0        1   41.623        1088      1       0
## 6      no       0      0        0    0.057           5      0       0
## 7      no       0      0        0    0.809          17      0       0
## 8      no       0      0        0    5.229          88      1       1
## 9      no       0      0        0    9.277         242      1       1
## 10     no       0      0        0   17.170         578      1       0
##    exclaim_subj urgent_subj exclaim_mess number
## 1             0           0            8  small
## 2             0           0            1    big
## 3             0           0            2   none
## 4             0           0            1  small
## 5             0           0           43  small
## 6             0           0            0  small
## 7             0           0            0  small
## 8             0           0            2  small
## 9             1           0           22  small
## 10            0           0            3  small
```

In practice, it is especially important to ask clarifying questions to ensure important aspects of the data are understood. For instance, it is always important to be sure we know what each variable means and the units of measurement. Descriptions of all email variables are given in the reference and can be accessed in `R` using the command

```
?email50
```

The data in `email50` represent a **data matrix** or in `R` terminology **data frame**, which is a common way to organize data. Each row of a data matrix corresponds to a unique case, and each column corresponds to a variable. This is called **tidy data**.[2] The data frame for the stroke study introduced in the previous lesson had patients as the cases and there were three variables recorded for each patient. If we are thinking of patients as the unit of observation, then this data is tidy.

```
## # A tibble: 10 x 3
##    group   outcome30 outcome365
##    <chr>   <chr>     <chr>
##  1 control no_event  no_event
##  2 trmt    no_event  no_event
##  3 control no_event  no_event
##  4 trmt    no_event  no_event
##  5 trmt    no_event  no_event
##  6 control no_event  no_event
##  7 trmt    no_event  no_event
##  8 control no_event  no_event
##  9 control no_event  no_event
## 10 control no_event  no_event
```

If we think of an outcome as a unit of observation then it is not tidy since the two outcome columns are variable values, time. The tidy data for this case would be:

```
## # A tibble: 10 x 4
##    patient_id group   time  result
##         <int> <chr>   <chr> <chr>
##  1          1 control month no_event
##  2          1 control year  no_event
##  3          2 trmt    month no_event
##  4          2 trmt    year  no_event
##  5          3 control month no_event
##  6          3 control year  no_event
##  7          4 trmt    month no_event
##  8          4 trmt    year  no_event
##  9          5 trmt    month no_event
## 10          5 trmt    year  no_event
```

There are three interrelated rules which make a dataset tidy:

1. Each variable must have its own column.

2. Each observation must have its own row.

3. Each value must have its own cell.

Why ensure that your data is tidy? There are two main advantages:

1. There's a general advantage to picking one consistent way of storing data. If you have a consistent data structure, it's easier to learn the tools that work with it because they have an underlying uniformity.

2. There's a specific advantage to placing variables in columns because it allows R's vectorised nature to shine. This will be more clear as the semester progresses. Since most built-in R functions work with vectors of values, it makes transforming tidy data feel particularly natural.

[2] For more information on tidy data see the blog and the book.

Data frames are a convenient way to record and store data. If another individual or case is added to the data set, an additional row can be easily added. Similarly, another column can be added for a new variable.

> Exercise
> We consider a publicly available data set that summarizes information about the 3,143 counties in the United States, and we call this the `county` data set. This data set includes information about each county: its name, the state where it resides, its population in 2000 and 2010, per capita federal spending, poverty rate, and four additional characteristics. This data set is part of `openintro` and is called `county`. How might these data be organized in a data matrix? [3]

Using `R`, we will display seven rows of the `county` data frame. The variables are summarized in help menu built into the `openintro` package.[4]

```
head(county,n=7)
```

```
##              name   state pop2000 pop2010 fed_spend poverty homeownership
## 1 Autauga County Alabama   43671   54571  6.068095    10.6          77.5
## 2 Baldwin County Alabama  140415  182265  6.139862    12.2          76.7
## 3 Barbour County Alabama   29038   27457  8.752158    25.0          68.0
## 4    Bibb County Alabama   20826   22915  7.122016    12.6          82.9
## 5  Blount County Alabama   51024   57322  5.130910    13.4          82.0
## 6 Bullock County Alabama   11714   10914  9.973062    25.3          76.9
## 7  Butler County Alabama   21399   20947  9.311835    25.0          69.0
##   multiunit income med_income
## 1       7.2  24568      53255
## 2      22.6  26469      50147
## 3      11.1  15875      33219
## 4       6.6  19918      41770
## 5       3.7  21070      45549
## 6       9.9  20289      31602
## 7      13.7  16916      30659
```

Types of variables

Examine the `fed_spend`, `pop2010`, and `state`, and variables in the `county` data set. Each of these variables is inherently different from the other three yet many of them share certain characteristics.

First consider `fed_spend`, which is said to be a **numerical variable** since it can take a wide range of numerical values, and it is sensible to add, subtract, or take averages with those values. On the other hand, we would not classify a variable reporting telephone area codes as numerical since their average, sum, and difference have no clear meaning.

The `pop2010` variable is also numerical, it is sensible to add, subtract, or take averages with those values, although it seems to be a little different than `fed_spend`. This variable of the population count can only be a whole non-negative number (0, 1, 2, ...). For this reason, the population variable is said to be **discrete** since it can only take numerical values with jumps. On the other hand, the federal spending variable is said to be **continuous**. Now in practice there are no truly continuous numerical variables since all measurements have finite accuracy. In this course we will treat both variables types of numerical variables the same, that is as continuous variables. The only place this will be different is in probability models which we see in the next block.

[3] Each county may be viewed as a case, and there are ten pieces of information recorded for each case. A table with 3,143 rows and 10 columns could hold these data, where each row represents a county and each column represents a particular piece of information.

[4] These data were collected from the US Census website.

The variable **state** can take up to 51 values after accounting for Washington, DC: *AL*, ..., and *WY*. Because the responses themselves are categories, `state` is called a **categorical** variable,[5] and the possible values are called the variable's **levels**.

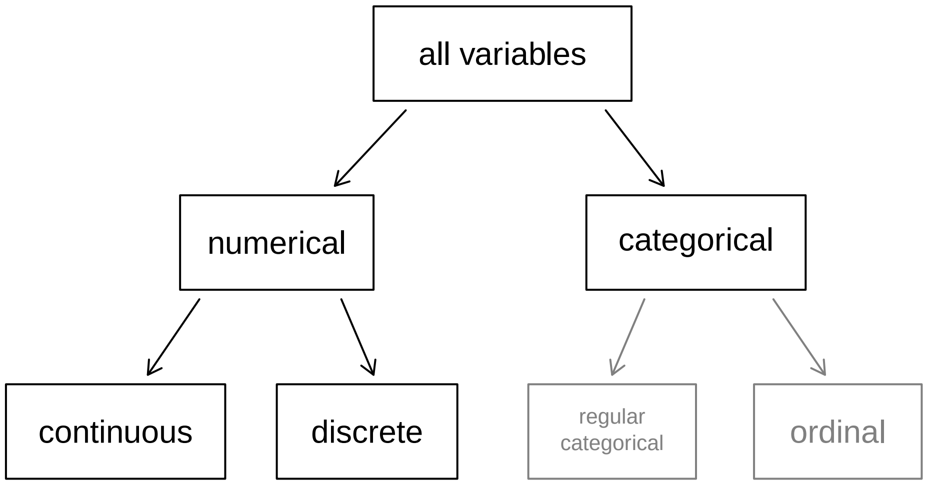


Figure 1: Taxonomy of Variables

Finally, consider a hypothetical variable on education, which describes the highest level of education completed and takes on one of the values *noHS*, *HS*, *College* or *Graduate_school*. This variable seems to be a hybrid: it is a categorical variable but the levels have a natural ordering. A variable with these properties is called an **ordinal** variable. To simplify analyses, any ordinal variables in this course will be treated as categorical variables. In `R` categorical variables can be treated in different ways; one of the key differences is that we can leave them as character values or as factors. When `R` handles factors, it is only concerned about the *levels* of values of the factors. We will learn more about this as the semester progresses.

Figure 1 captures this classification of variables.

> Exercise Data were collected about students in a statistics course. Three variables were recorded for each student: number of siblings, student height, and whether the student had previously taken a statistics course. Classify each of the variables as continuous numerical, discrete numerical, or categorical.}

The number of siblings and student height represent numerical variables. Because the number of siblings is a count, it is discrete. Height varies continuously, so it is a continuous numerical variable. The last variable classifies students into two categories – those who have and those who have not taken a statistics course – which makes this variable categorical.

[5] Sometimes also called a **nominal** variable.

Exercise Consider the variables `group` and `outcome30` from the stent study in the case study lesson. Are these numerical or categorical variables? [6]

Relationships between variables

Many analyses are motivated by a researcher looking for a relationship between two or more variables, this is the heart of statistical modeling. A social scientist may like to answer some of the following questions:

1. Is federal spending, on average, higher or lower in counties with high rates of poverty?

2. If homeownership is lower than the national average in one county, will the percent of multi-unit structures in that county likely be above or below the national average?

To answer these questions, data must be collected, such as the `county` data set. Examining summary statistics could provide insights for each of the three questions about counties. Additionally, graphs can be used to visually summarize data and are useful for answering such questions as well.

Scatterplots are one type of graph used to study the relationship between two numerical variables. Figure 2 compares the variables `fed_spend` and `poverty`. Each point on the plot represents a single county. For instance, the highlighted dot corresponds to County 1088 in the `county` data set: Owsley County, Kentucky, which had a poverty rate of 41.5% and federal spending of $21.50 per capita. The dense cloud in the scatterplot suggests a relationship between the two variables: counties with a high poverty rate also tend to have slightly more federal spending. We might brainstorm as to why this relationship exists and investigate each idea to determine which is the most reasonable explanation.

Exercise Examine the variables in the `email50` data set. Create two questions about the relationships between these variables that are of interest to you.[7]

The `fed_spend` and `poverty` variables are said to be associated because the plot shows a discernible pattern. When two variables show some connection with one another, they are called **associated variables**. Associated variables can also be called **dependent** variables and vice-versa.

Example The relationship between the homeownership rate and the percent of units in multi-unit structures (e.g. apartments, condos) is visualized using a scatterplot in Figure 3. Are these variables associated?

It appears that the larger the fraction of units in multi-unit structures, the lower the homeownership rate. Since there is some relationship between the variables, they are associated.

Because there is a downward trend in Figure 3 – counties with more units in multi-unit structures are associated with lower homeownership – these variables are said to be **negatively associated**. A **positive association** is shown in the relationship between the `poverty` and `fed_spend` variables represented in Figure 2, where counties with higher poverty rates tend to receive more federal spending per capita.

If two variables are not associated, then they are said to be **independent**. That is, two variables are independent if there is no evident relationship between the two.

A pair of variables are either related in some way (associated) or not (independent). No pair of variables is both associated and independent.

[6] There are only two possible values for each variable, and in both cases they describe categories. Thus, each is a categorical variable.

[7] Two sample questions: (1) Intuition suggests that if there are many line breaks in an email then there would also tend to be many characters: does this hold true? (2) Is there a connection between whether an email format is plain text (versus HTML) and whether it is a spam message?

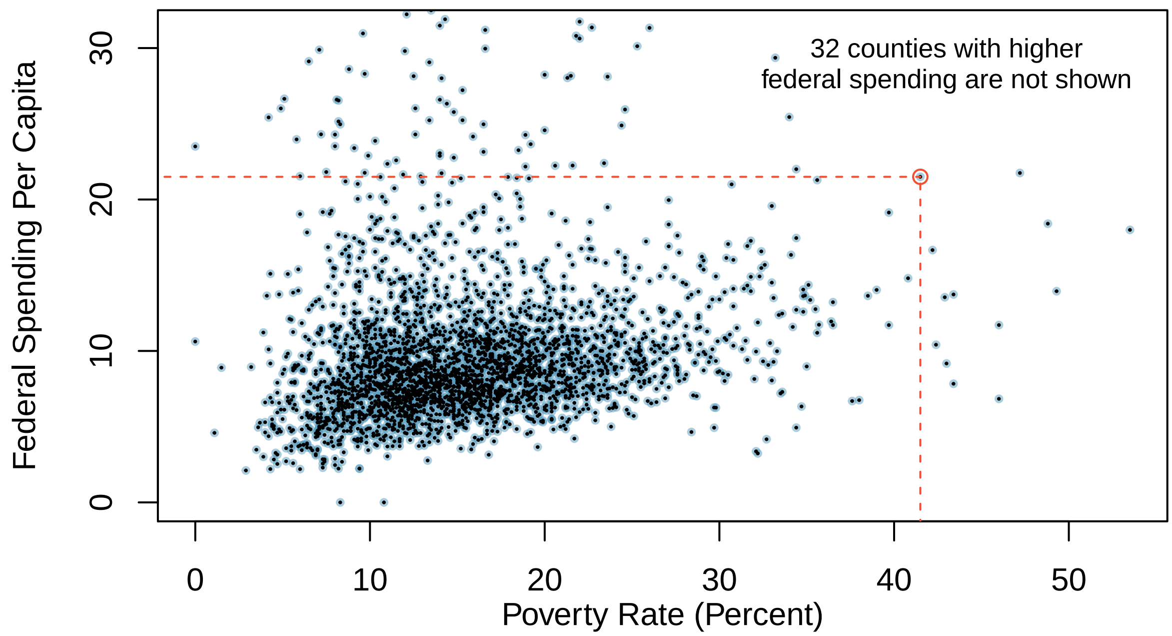


Figure 2: A scatterplot showing fed_spend against poverty. Owsley County of Kentucky, with a poverty rate of 41.5% and federal spending of $21.50 per capita, is highlighted.

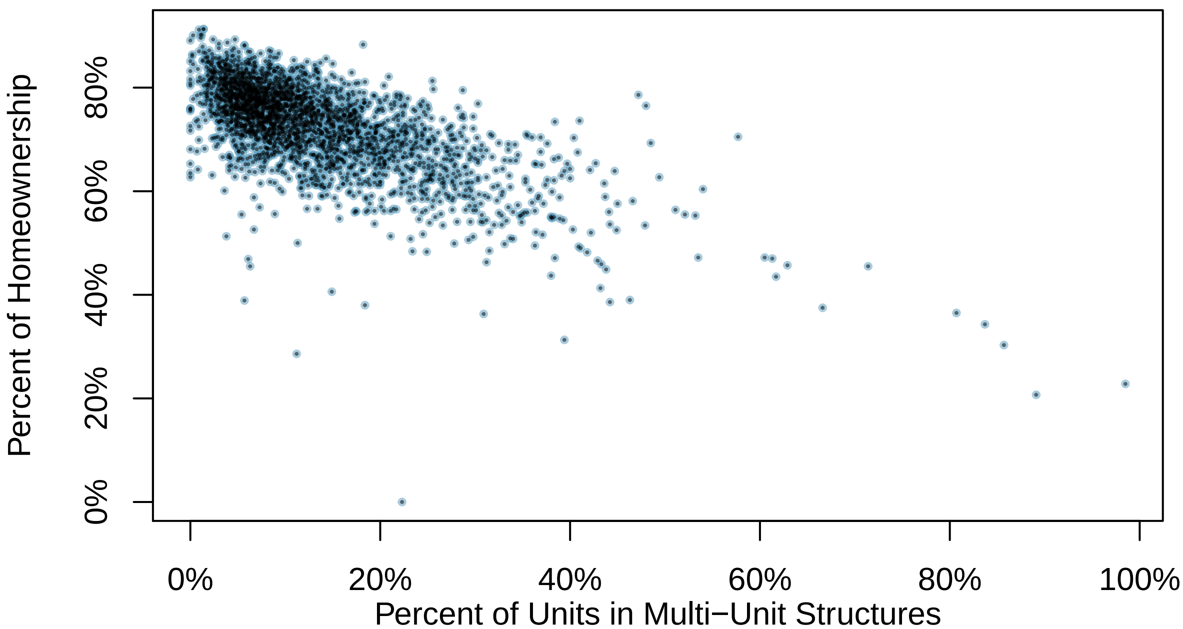


Figure 3: A scatterplot of the homeownership rate versus the percent of units that are in multi-unit structures for all 3,143 counties.

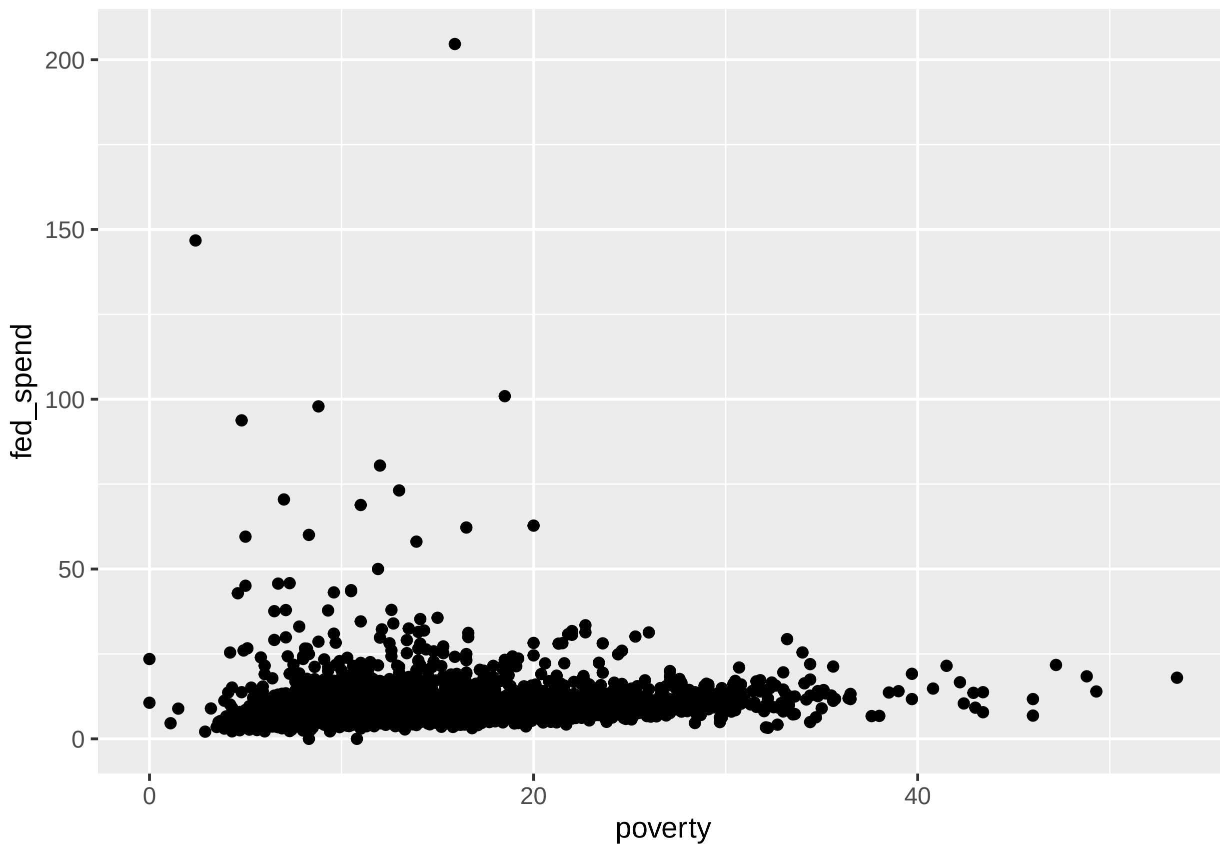
Creating a scatterplot

In this section we will create a simple scatterplot and then ask you to create one on your own. First we will Figure 2. This figure uses the `county` data set.

Here are two questions: *What do we want* `R` *to do?* and
What must we give `R` *for it to do this?*

We want `R` to create a scatterplot and to do this it needs, at a minimum, the data object, what we want on the x-axis, and what we want on the y-axis. More information on `ggformula` can be found by clinking on the link.[8]

```
county %>%
  gf_point(fed_spend~poverty)
```

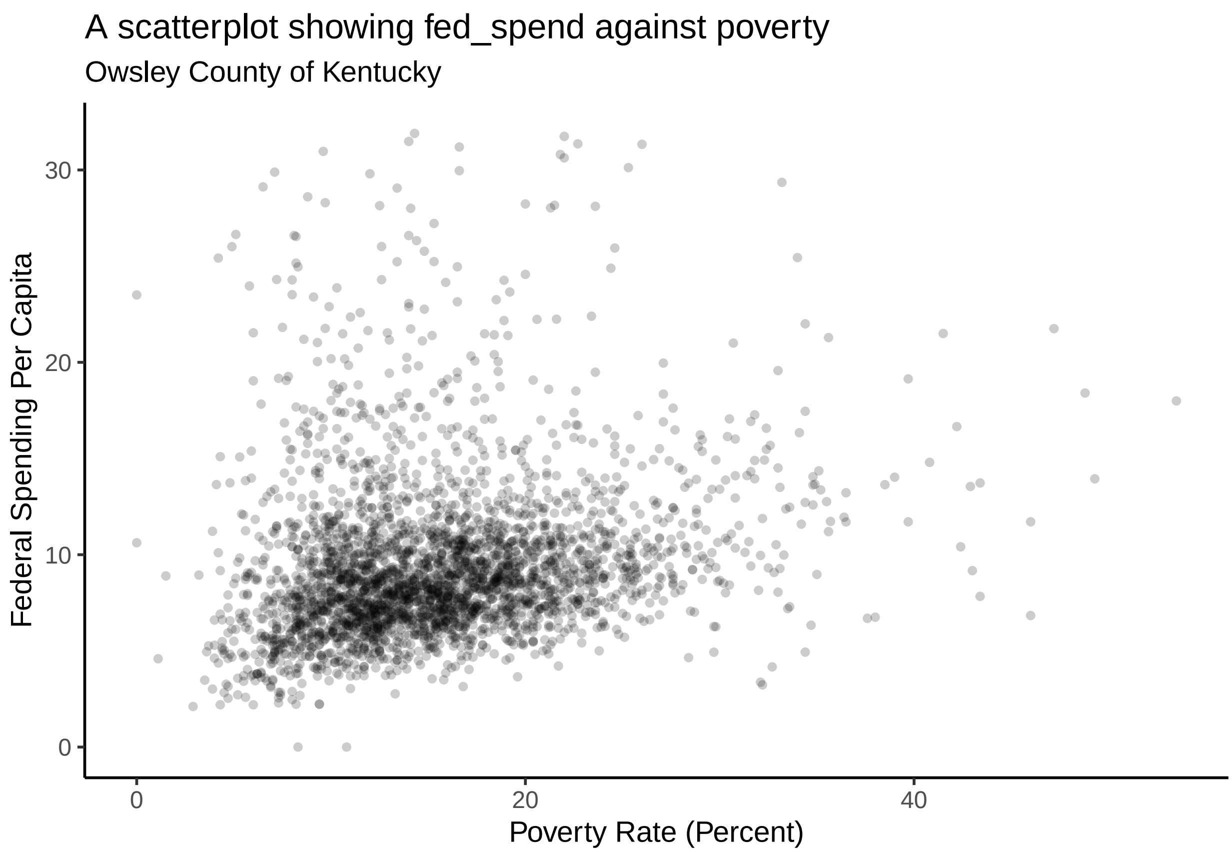


This plot is bad, there are poor axis labels, no title, dense clustering of points, the y-axis is being driven by a couple of extreme points. We will need to clear this up. Again, try to read the code and use help to determine the purpose of each command.

```
county %>%
  filter(fed_spend<32) %>%
  gf_point(fed_spend~poverty,
           xlab="Poverty Rate (Percent)",
           ylab="Federal Spending Per Capita",
           title="A scatterplot showing fed_spend against poverty",
```

[8] https://cran.r-project.org/web/packages/ggformula/vignettes/ggformula-blog.html

```
          subtitle =  "Owsley County of Kentucky",
          cex=1,alpha=0.2) %>%
  gf_theme(theme_classic())
```



Exercise Create the scatterplot in Figure 3.

File creation information

- File creation date: 2020-05-11
- Windows version: Windows 10 x64 (build 17763)
- R version 3.6.3 (2020-02-29)
- `mosaic` package version: 1.6.0
- `tidyverse` package version: 1.3.0
- `openintro` package version": 1.7.1