Summarizing Data using visualization

3

M3 ICA2

Contents

Introduction

Summarizing data graphically and numerically usually requires substantial thought about the data, its structure, what might be obscured by certain representations, and so on. This assignment presents data where different representations yield dramatically different results.

Univariate data

We have seen that a box plot, which graphically represents the median, the first and third quartiles, and the overall range of the data, can be an effective graphical summary of a single variable.

First read in a data frame that contains five different univariate data sets. The first line, `options(digits = 3)`, tells R to print three significant digits for numeric values. (The usual default in R is seven.)

```
options(digits = 3)
fiveUnivariate <- readRDS(url("https://raw.githubusercontent.com/vfmelfi/STT180SS20/master/fiveUnivaria
str(fiveUnivariate)
```

```
Classes 'tbl_df', 'tbl' and 'data.frame':   12420 obs. of  2 variables:
 $ dataSet    : chr  "left" "left" "left" "left" ...
 $ observation: num  -9.77 -9.76 -9.75 -9.77 -9.76 ...
```

```
names(fiveUnivariate)
```

```
[1] "dataSet"     "observation"
```

```
unique(fiveUnivariate$dataSet)
```

```
[1] "left"   "right"  "normal" "split"  "lines"
```

```r
head(fiveUnivariate)
```

```
  dataSet observation
1    left       -9.77
2    left       -9.76
3    left       -9.75
4    left       -9.77
5    left       -9.76
6    left       -9.77
```

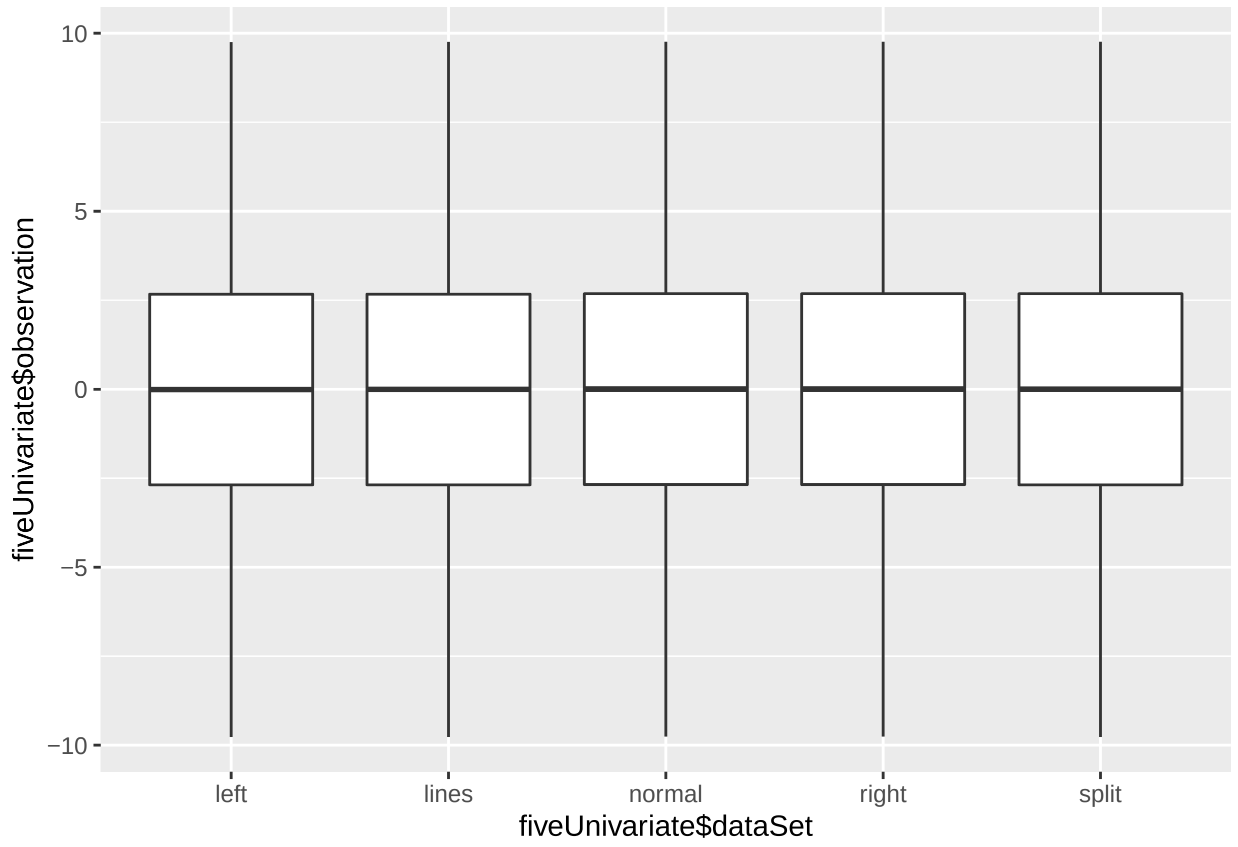
The data frame `fiveUnivariate` has two columns, called `dataSet` and `observation`. To, for example, extract the observations from the data set called `normal` and look at the first few entries we would use

```r
head(fiveUnivariate$observation[fiveUnivariate$dataSet == "normal"])
```

```
[1] -9.76 -9.72 -9.68 -9.64 -9.60 -9.56
```

(1) Using `ggplot` draw side-by-side boxplots of the five data sets in `fiveUnivariate`.

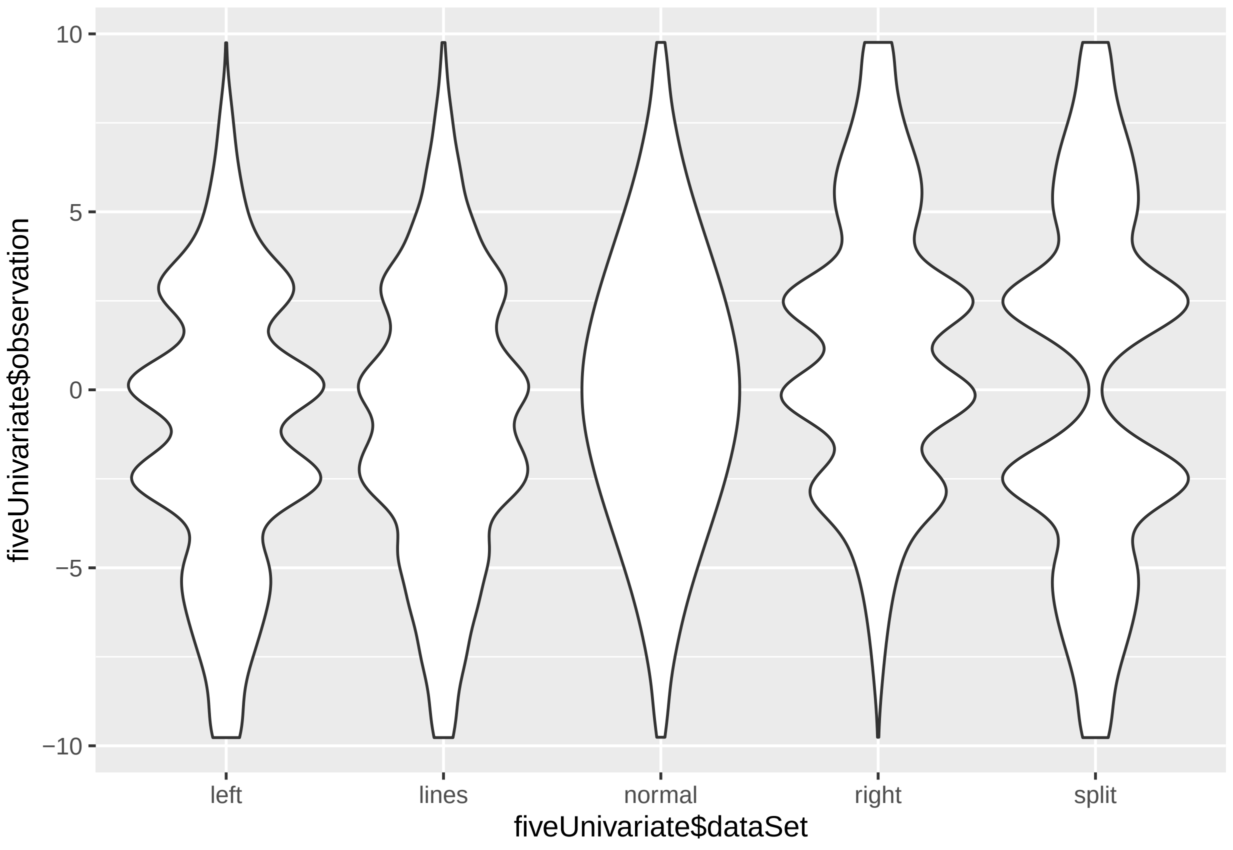
```r
library(tidyverse)
ggplot(data=fiveUnivariate, mapping = aes(x=fiveUnivariate$dataSet, y=fiveUnivariate$observation)) + geo
```



(@) Based on the boxplots, what would you conclude about the five data sets? Do they seem similar? Different? Explain. ## I would say looking at the boxplots, the data sets are all similar. However, I don't think it is that way intentionally.

(2) Hopefully you answered that the five datasets look similar based on the boxplots. Since any particular graphical or numerical representation of data can hide features, choose a different graphical representation of the data (again producing a separate graphical representation for each of the five data sets). Based on this representation, do your answers to the above question change? Why or why not? ## The representations look different now because they actually don't represent the same thing. It only looked that way with a boxplot

```
ggplot(data=fiveUnivariate, mapping = aes(x=fiveUnivariate$dataSet, y=fiveUnivariate$observation)) + geo
```



```
#Hint: can use geom_histogram or geom_violin
```

The function `tapply()` provides a way to compute summary statistics (such as the mean, median, standard deviation, etc.) separately for each data set. In its simplest use, the function has the form

```
tapply(X, INDEX, FUN)
```

where `X` is a vector of observations, `INDEX` is a list of one or more factors, each having the same length as `X`, and `FUN` is a function to be applied to the values in `X`, separately for each of the values of `INDEX`. It's easiest to understand via an example. Here the standard deviation of each of the five data sets is computed.

```
tapply(fiveUnivariate$observation, fiveUnivariate$dataSet, sd)
```

```
  left  lines normal  right  split
  3.90   3.88   3.84   3.90   4.98
```

```
with(fiveUnivariate, tapply(observation, dataSet, sd))
```

```
  left  lines normal  right  split 
  3.90   3.88   3.84   3.90   4.98 
```

In the second line of code, the `with()` function was used. Sometimes the "dollar-sign" notation for extracting a column of a data frame makes code messy, and the `with()` function can clean up the code. Either of the two lines of code seems fine in this case.

(3) Use `tapply` with the `summary()` function in place of `FUN` to compute the summary of each of the five data sets. How does the summary add to the information contained in the boxplots?

```
tapply(fiveUnivariate$observation, fiveUnivariate$dataSet, summary)
```

```
$left
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  -9.77   -2.69   -0.01   -1.18    2.67    9.75 

$lines
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  -9.77   -2.69   -0.01   -0.83    2.67    9.76 

$normal
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  -9.76   -2.68    0.00    0.00    2.68    9.76 

$right
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  -9.76   -2.68    0.00    1.17    2.68    9.76 

$split
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  -9.77   -2.69    0.00    0.00    2.68    9.76 
```

```
with(fiveUnivariate, tapply(observation, dataSet, summary))
```

```
$left
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  -9.77   -2.69   -0.01   -1.18    2.67    9.75 

$lines
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  -9.77   -2.69   -0.01   -0.83    2.67    9.76 

$normal
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  -9.76   -2.68    0.00    0.00    2.68    9.76 

$right
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  -9.76   -2.68    0.00    1.17    2.68    9.76 
```

```
$split
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -9.77   -2.69    0.00    0.00    2.68    9.76
```

```
## The summary shows that a lot of the data can look similar to a boxplot, which is why when represente
```

Bivariate data

Next we consider data sets with two variables. For such data sets it is common to calculate the mean and standard deviation of each varaible, and also to compute the correlation coefficient between the two variables. (The correlation coefficient measures the strength and direction of the linear relationship between two variables, with values between -1 and 1. Points with a correlation coefficient of 1 all lie on a line with positive slope. Points with a correlation coefficient of -1 all lie on a line with negative slope.)

First, read in the data.

```
fourBivariate <- readRDS(url("https://raw.githubusercontent.com/vfmelfi/STT180SS20/master/fourBivariate
str(fourBivariate)
```

```
'data.frame':   568 obs. of  3 variables:
 $ dataSet: chr  "dino" "dino" "dino" "dino" ...
 $ x      : num  55.4 51.5 46.2 42.8 40.8 ...
 $ y      : num  97.2 96 94.5 91.4 88.3 ...
```

```
unique(fourBivariate$dataSet)
```

```
[1] "dino"   "away"   "star"   "circle"
```

(4) Use `tapply()` to calculate the mean of `x` for each of the four data sets. Are the means similar?

```
tapply(fourBivariate$x, fourBivariate$dataSet, mean)
```

```
  away circle   dino   star 
  54.3   54.3   54.3   54.3 
```

```
## They are all the same
```

(5) Use `tapply()` to calculate the mean of `y` for each of the four data sets. Are the means similar?

```
tapply(fourBivariate$y, fourBivariate$dataSet, mean)
```

```
  away circle   dino   star 
  47.8   47.8   47.8   47.8 
```

```
## They are all the same
```

(6) Use `tapply()` to calculate the standard deviation of `x` for each of the four data sets. Are the standard deviations similar?

```
tapply(fourBivariate$x, fourBivariate$dataSet, sd)
```

```
  away circle   dino   star 
  16.8   16.8   16.8   16.8 
```

```
##They are all the same
```

(7) Use `tapply()` to calculate the standard deviation of `y` for each of the four data sets. Are the standard deviations similar?

```
tapply(fourBivariate$y, fourBivariate$dataSet, sd)
```

```
  away circle   dino   star 
  26.9   26.9   26.9   26.9 
```

```
## They are all the same
```

(8) Use the `cor()` function to calculate the correlation coefficient between `x` and `y` for each of the four data sets. In a later class we will learn how to do this in one line of code, but for now, use a separate line of code for each data set. Are the correlation coefficients similar?

```
cor(fourBivariate$x[fourBivariate$dataSet == "dino"],(fourBivariate$y[fourBivariate$dataSet == "dino"]))
```

```
[1] -0.0645
```

```
cor(fourBivariate$x[fourBivariate$dataSet == "away"],(fourBivariate$y[fourBivariate$dataSet == "away"]))
```

```
[1] -0.0641
```

```
cor(fourBivariate$x[fourBivariate$dataSet == "star"],(fourBivariate$y[fourBivariate$dataSet == "star"]))
```

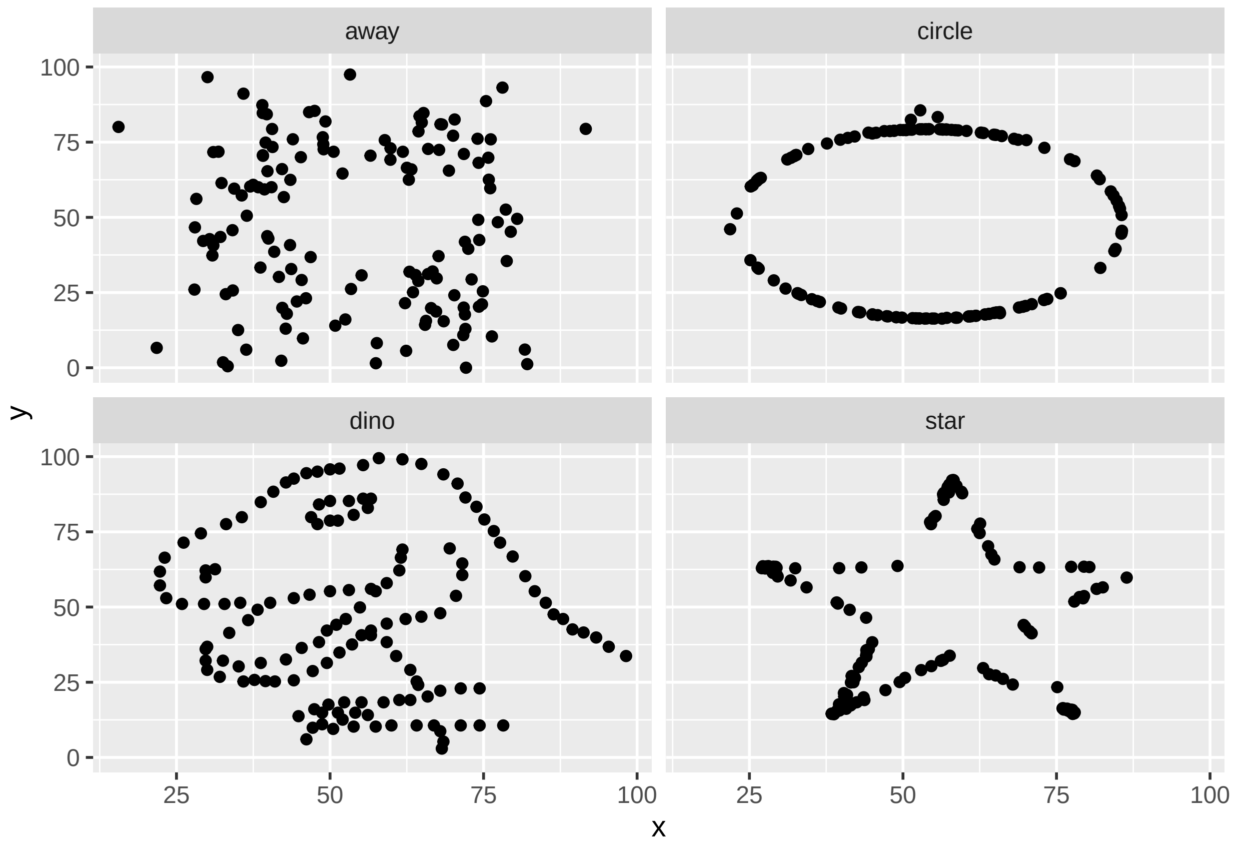
```
[1] -0.063
```

```
cor(fourBivariate$x[fourBivariate$dataSet == "circle"],(fourBivariate$y[fourBivariate$dataSet == "circl
```

```
[1] -0.0683
```

Based on the previous questions, you hopefully noticed that, based on the means, standard deviations, and correlation coefficients, the four data sets look quite similar. ## They are similar, but they have their differences (@) Using `ggplot()` draw scatter plots of each of the four data sets, with `y` on the vertical axis and `x` on the horizontal axis. Use faceting to put the four scatter plots in two rows and two columns.

```
ggplot(data=fourBivariate, aes(x, y)) + geom_point() +
  facet_wrap(~ dataSet)
```



(9) Do the scatter plots change the suggestion from the means, standard deviations, and correlation coefficients that the four data sets are similar? Why or why not? ##The scatterplots did change rhw suggestion from the means, sd, and correlation coefficients. This shows that the numbers did not represent the graphical representation, and that it is important to understand the data. The data isn't always represented / visualized how the numbers show it should be. ## References

1. A description of the data used in today's ICA

2. An R package containing the data used in today's ICA