Summary Statistics and Data Display.*

Greg Kochanski
http://kochanski.org/gpk

2006/02/18 23:54:50 UTC

1 Introduction

In this handout, we will discuss ways of looking and describing one-dimensional data. One-dimensional means that each data point is a single number, a single measurement.

Some of these techniques can also apply to more complex data sets where it takes several numbers to describe one datum.

1.1 Basic R

The examples here are shown for the R statistical package. R works on Windows, Linux and Macs. R has some design flaws that can present traps for the unwary, but it is extremely powerful and has any statistical analysis you can imagine, and it produces useful plots.

2 Histograms and Related Plots

Dot plots, stem plots and histograms are really much the same idea. If we plot the duration of eruptions of Old Faithful Geyser, we get the plots in Figures1, 2, 3.

The histogram plot (Figure 1) is produced by the following instructions[1]

```
R --gui=Tk (This starts the program on Linux.
On Windows, you'd click on an icon.)
> attach(faithful)      (This makes the "faithful" data set into the default.)
> hist(eruptions, xlab="Eruption Duration")
```

The "attach" command takes a data set[2] and makes it easily accessible. After the `attach` command, you can access the `eruptions` column of the `faithful` data set by typing "eruptions". (Before the `attach` command, you would need to type `faithful$eruptions` to access the same data.)

In a histogram, the horizontal axis is divided into "bins", and the vertical axis counts how many data are within each bin. For example, there were about 45 eruptions that lasted more than 4 minutes and less than 4.5 minutes. The width and location of the bin boundaries is your choice and it affects the plotted histogram. Figure 2 shows a histogram created with narrower (perhaps too narrow) bins. This was produced by:

```
> hist(eruptions, xlab="Eruption Duration", nclass=100)
```


[1] Here, R produces the ">" symbol as a prompt, to tell you that it is ready for more input. You type the function like "attach(faithful)", and the parenthetical expression to the left would not be there in a real session – I add it to explain what is going on.

[2] A data set is called a "data frame" in R's terminology. It contains several different columns of data, all related to each other.

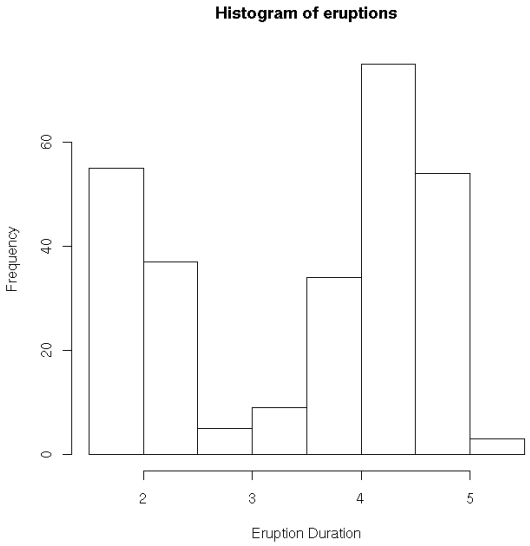

Figure 1: A histogram of the duration of eruptions of Old Faithful. The vertical axis shows how many eruptions have the same duration, and the vertical axis lays out the duration of the eruptions (in minutes).

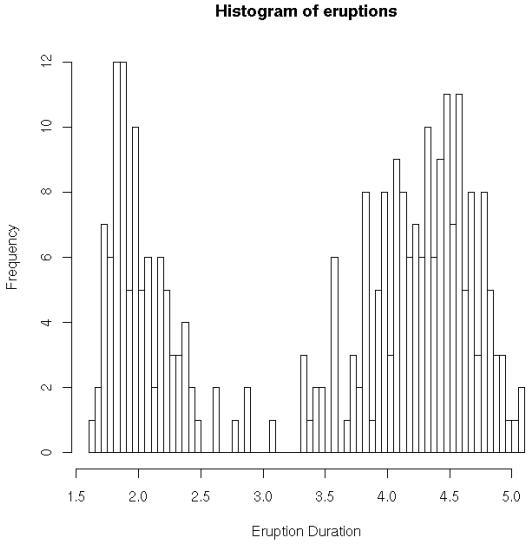

Figure 2: A histogram of the duration of eruptions of Old Faithful; same data as Figure 1, but narrower bins. With the narrower bins, we are making finer distinctions but are running short of data: note that the peaks are no longer smooth, but ragged. The up-and-down jittering is caused by statistical counting errors in each bin.

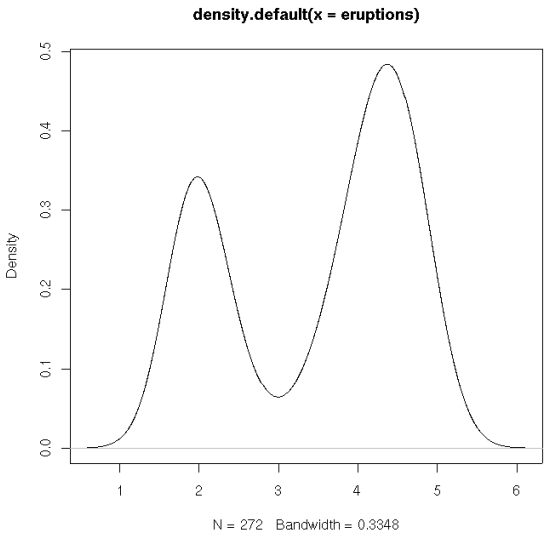

Figure 3: An estimate of the probability density of eruption durations from the R function `density()`. This uses the same data as Figures 1 and 2. The difference is that we are stepping away from the data, going from an observable frequency to an estimate of an underlying probability that may have generated the observations.

Note that one can add extra information into the `hist()` function if you want to customise it. All the possible arguments to the function, along with brief descriptions can be seen by running

```
> help(hist)
```

If you can't quite remember what function you want,

```
> help("histogram")
```

will show you some possibilities that mention the word "histogram."

These histograms, you will notice, have two main bumps. In other words, Old Faithful has two types of eruptions, long and short.

We show a related plot in Figure 3. This is an estimate of the probability density of the duration of Old Faithful's eruptions. In a sense, it is an extrapolation to what the histogram might look like if we had an infinite amount of data, under certain theoretical assumptions, but (of course) it doesn't contain any more information than the histogram.

Without going into detail, we note that this (like all probabilities) is a theoretical construct, a model of the data, and not the data itself (although it is based on the data). The particular theoretical assumption underlying this density estimate (and others are possible) is that the density is as smooth as possible while still remaining consistent with the data. It is assuming something like Ockham's Razor (Carroll [2003]; [Dowe 2000]

Finally, we can show the individual data underneath the histogram in Figure 4. The code for this plot is

```
# Bring in a pre-existing data set:
> attach(faithful)
# Plot the histogram:
> hist(eruptions, xlab=‘‘Duration (min)’’)
# Plot the bottom rug plot:
> rug(eruptions)
# Plot the upper rug plot, using jitter() to break ties:
> rug(jitter(eruptions, amount=0.1), side=3)
```

The calls to `rug()` generate the little plots of grass above and below the histogram; it puts a tick mark at each data point. A trap or trick of `rug()` is that you only get one tick if several data points have exactly the same

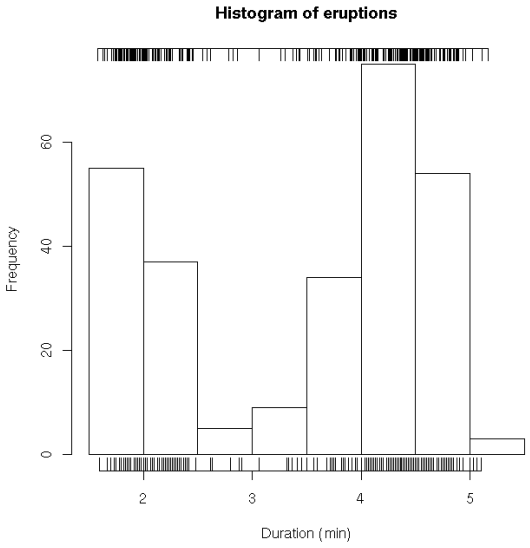

Figure 4: A histogram with added rug plots. The upper rug plot is jittered so that you can see all of the data individually. The lower rug plot is not jittered; one sees the exact data values but some data land on top of other data.

value. This data, for instance, is rounded to 5-second intervals (1/12 of a minute), and a few values appear up to eight times. Consequently, these values are visually underrepresented in the rug plot.

A way to avoid this at little cost in fidelity is to jitter the data left and right by just a little bit, preferably just a few times the width of each tick mark on the rug plot. That's done in the second call to `rug()` which appears above the histogram. Here (for the purposes of the rug plot only) we've pushed the data around randomly by about 0.1 minutes. If the jitter is turned up too far, you risk spreading out narrow clumps in the data.

3 Summary Statistics

The idea of a summary statistic is to compactly say something important about the data. The two most important things people want to know about a data set are "what is a typical value?" and "how repeatable is it?" In other words, the questions are "where is the bump in the histogram?" and "how wide is the bump in the histogram?"

3.1 Location Estimators.

The most common location estimator is the average or mean:

3.1.1 The Mean

The mean is also commonly known as the average[3]. You get it by adding up all the data, then dividing by how many data you have. For instance, the mean of 1, 2, 3, and 4 is $(1+2+3+4)/4$ or 2.5. In R, we'd do it this way:

```
> data = c(1, 2, 3, 4)              (This creates the data set.)
> mean(data)                        (We take the mean.)
[1] 2.5                             (R prints it out.)
```

[3] ...but don't get them confused in R. `Mean` is what one expects: it gives a single value for a location estimate by adding up all the data and dividing. In R, the `average` function, on the other hand, is a fancy thing for doing means of groups. Call it on your data vector, and you'll get multiple copies of the mean, one for each datum.

The `c` concatenates its arguments into a vector[4]. The mean is the best possible estimate of position (i.e. a typical value, or the centre of the probability distribution) for data that comes from a Gaussian distribution. While this is a common distribution, it is not universal, and if you have data that has occasional wild points, the mean is actually not a very good distribution because a few wild points can push the value of the mean around quite a lot. See Huff [1954] for good examples of problems with the mean.

Question: Did you know that most linguists have more than the average number of legs?
Very few people have grown an extra leg, yet a reasonable number have lost one. The average number of legs is therefore slightly less than two. Two-legged members of the audience therefore really do have more than the average number of legs. Try this as an ethnic joke, and you'll find out who understands their math: "Did you know that most X have more than than the average number of legs?" Of course, if they don't understand the math, they'll write you off as an insensitive clod who drops gratuitious insults.

3.1.2 The Median

Instead of the mean, one can use the median. This estimate of location is almost completely un-fazed by wild data: it is "robust"[5]. So robust, in fact, that you can move almost half the data anywhere, including out to infinity, and change the median by only a modest amount. (The mean becomes infinitely bad if you push some of your data out to infinity.)

The median is simply the middle data, with an equal number above and below. If there are an even number of data, the median is taken to be half-way in between the middle pair. In R, one computes like this:

```
> data = c(1, 2, 3, 4)              (This creates the data set.)
> median(data)                      (We take the median.)
[1] 2.5                             (R prints it out.)
```

Note what happens as we change the last data point:

```
> data = c(1, 2, 3, 4)
> mean(data)
[1] 2.5
> median(data)
[1] 2.5
> data = c(1, 2, 3, 5)
> mean(data)
[1] 3.0
> median(data)
[1] 2.5
> data = c(1, 2, 3, 50)
> mean(data)
[1] 14
> median(data)
[1] 2.5
> data = c(1, 2, 3, 50000)
> mean(data)
[1] 12501.5
> median(data)
[1] 2.5
```

[4] You might ask if we can use the `list` function that you see in the Box plot example (§4)? Ha. You will find that `mean` complains that its argument is not numeric.

[5] Note that it is almost always much better to use a robust estimator to drop wildly bad outlying data than to eliminate them by hand. Eliminating them by hand opens up the possibility of biassing your results and makes it impossible to do good statistics on your data. You should only eliminate data when there is a strong, easily explainable reason for it, such as "That's the sentence right after his chair collapsed. Maybe he was still a bit upset." Any data you eliminate should be noted in your report or published paper.

3.1.3 The Mode

The mode is the individually most likely value in a data set or a probability distribution. It is simply the most common value in the data set.

It is most often useful for data sets that have only a few likely values, that is for discrete data. So, saying that the mode of the distribution of houses is three-bedroom works well, since houses only come in 1-bedroom, 2-, 3-, 4-, 5-bedroom sizes[6]. If you have a data set with many possible distinct values, such has the heights of trees measured in millimetres, you need a rather large data set. After all, if you have only 30 trees, they will all probably have different heights, so no one value will be most common. This problem only gets worse if there are an infinite number of possible data values.

Even if there is enough data so that some values have multiple repetitions, the mode turns out to be poorly-behaved that it isn't very stable. If you measure many different samples of data from an experiment, the resulting modes will typically be more variable than a mean or median.

R does not have built-in function for the mode.

Exercise: Medians and Modes: Do you have more or less legs than the median? The mode?

3.1.4 Fancier Estimators

Many fancier estimators for location exist, including robust estimators that are better than the mode for Gaussian data, but still very robust against occasional wild points. The simplest example is the trimmed mean, which simply throws away the highest and lowest few data before averaging the rest. In R, you can do this:

```
mean(data, trim=0.05)
```

and you will trim off the top and bottom 5% of the data before taking the mean. This will protect you from wild answers, so long as less than 5% of your data is wildly wrong.

3.2 Width Estimators.

Three of the standard ways of estimating the width of a distribution of data are the standard deviation, the inter-quartile range, and the median absolute deviation. Detailed definitions of these measures can be found in Gonick and Smith [1994].

The standard deviation is like the mean, in that it provides the best possible estimate for Gaussian data, but fails badly if there are even a few wildly bad data. It is the most commonly reported value, though, so you should generally use it unless you have reason to suspect that there are some wildly wrong data.

The inter-quartile range and median absolute deviation are more robust, so the results will ignore a few wild data. You use them like this:

```
> x = c(1, 2, 3, 4, 5, 6, 7, 8)
> sd(x)
[1] 2.449490
> mad(x)
[1] 2.9652
> IQR(x)
[1] 3.5
```

Those values show that if you look at the numbers between 1 and 8, most of them are within about 3 of the centre of the distribution (i.e. 4.5)[7].

If we apply these measures of location and size to the Old Faithful data, we get a reasonable view of where most of the data is, and how far it spreads (you can compare with Figure 1).

[6] Aside from the occasional stately home.

[7] Note that the `mad` function in R reports the median absolute deviation divided by 1.4826, so that for Gaussian distributed data it agrees with the standard deviation. If you want the actual median of the absolute value of the deviations around the mean, you need to divide by that factor or run `mad(data, constant=1)`.

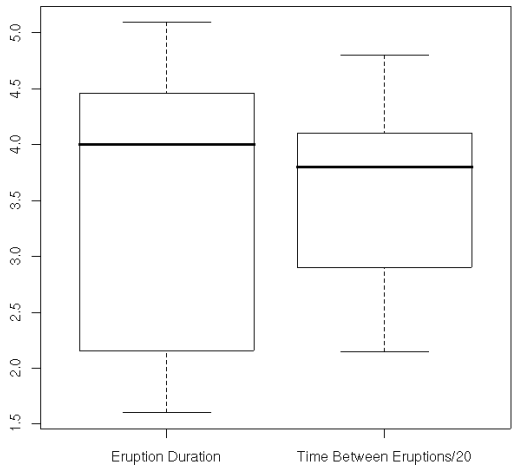

Figure 5: This is a box plot of two sets of data. The left box shows the duration of Old Faithful's eruptions in minutes. The right box shows the interval between eruptions, divided by 20. The Box Plot shows the entire range of observed data, along with the range of "typical variation" and the median.

```
> mean(eruptions)
[1] 3.487783
> median(eruptions)
[1] 4
> sd(eruptions)
[1] 1.141371
> mad(eruptions)
[1] 0.9510879
> IQR(eruptions)
[1] 2.2915
```

This tells you that the typical length of an eruption is about 4 minutes (or a bit less), and most eruptions are within a minute or two of the "typical eruption".

4 Box Plots

Figure 5 is different way of showing the data. In this plot, the data axis is vertical, rather than horizontal as in the histogram. The rectangular box shows the range of "typical" data, the horizontal line inside the box shows the median data ("most typical"), and the dashed lines extend up and down to show the range of variation.

More precisely, the bar in the box shows the median; half of the data falls above the median bar, half below. The ends of the box show the first and third quartiles[8]; half of the data falls within the box, half of it outside. The dashed whiskers extend to the last data within a range of 1.5 times the width of the box (the inter-quartile range) beyond the end of the box. They can be quite short if the data all cluster closely together; they can be quite long if the data have a "halo" of wild measurements around the central part of the distribution. Finally, any further points are called "outliers," and are plotted individually.

You can compare it to a histogram (Figure 1) by simply turning it on its side. For this particular data set, which has two bumps, the Box Plot indeed shows up poorly. It does not even hint at the important fact that

[8] They actually show the medians of the upper and lower halves of the data, where the central point is counted in both halves if there are an odd number of data. This definition gives the same answers as the first and third quartiles for an even number of data, and is trivially different if you have an odd number of measurements.

the distribution is bimodal. However, it will at least tell you about how long an eruption lasts (about 4 minutes) and that the durations of eruptions are quite variable: that you shouldn't be surprised at either a 2 or 5 minute eruption.

One major advantage of Box plots is that they can provide a clear and easy graphical comparison of a group of different data sets. In Figure 5, we generate a second data set by dividing the interval between eruptions by 20. That is then plotted on the same axis as the duration data. You can see immediately that the interval between eruptions is (relatively speaking) more reproducible than the duration. The box is narrower and the whiskers are shorter. Again, there are no outliers: no eruptions that are surprisingly close to each other or at surprisingly long intervals.

The code to produce the pair of box plots is

```
> boxplot( list(eruptions, waiting/20),
+         names=list("Eruption Duration", "Time Between Eruptions/20"))
```

this introduces a new function, `list`, which ties together several things into one list[9]. In this case, we place two data sets into one list (`eruptions` and `waiting/20`, the latter of which is just the time between eruptions divided by 20). We also place two names into one list ("Eruption Duration" and "Time Between Eruptions/20"), and pass it into the `boxplot` function as the `names` of the two boxes, which appear below. Note that on the second line, the computer gives a "+" prompt instead of the normal ">" prompt. This is just a hint that you have not closed the parentheses.

The above is a fairly complicated little expression. We can split it up into several pieces that may make it a bit clearer.

```
> wtdiv = waiting/20
> data = list(eruptions, wtdiv),
> namelist = list("Eruption Duration", "Time Between Eruptions/20")
> boxplot( data, names=namelist)
```

The result should be the same. One can print `wtdiv`, `data`, and `namelist`, if you want to make sure what you are doing, but unfortunately the results are ugly. To add more boxes, you'd create a longer list to assign to `data`, and you would want to also add more names to `namelist` so you'd continue to have one label for each box.

References

Robert Todd Carroll. *The Skeptic's Dictionary*. John Wiley and Sons, New Jersey, 2003. URL `http://skepdic.com/occam.html`.

Larry Gonick and Woollcott Smith. *The Cartoon Guide to Statistics*. HarperCollins, New York, 1994. ISBN 0062731025.

Darrell Huff. *How to Lie with Statistics*. W. W. Norton & Co., New York, 1954. ISBN 0-393-09426-X (republished in 1984).

[9] You'll perhaps wonder why we didn't use that `c` function here that we earlier used to create a vector. Don't ask. It's an awful wart on the language. `c` concatenates things together, so if we did `c(eruptions, waiting/20)`, we would not have a list of two data sets; rather we would have one double-length data set. The `boxplot` function would then complain that it has one data set with two labels.