Problem Set - Probability Distributions & Bootstrapping

Sam Mason

02/03/2020

Part I: The Comparative Anatomy of Distributions

QUESTION 1
For each of the distributions listed below, define the following attributes using words (as opposed to mathematical expressions): **(a)** *support*, **(b)** *parameters and their possible values*, **(c)** *data type (discrete vs. continuous)*, and **(d)** *example data*.

- Poisson
- Negative binomial
- t

EXAMPLE

```
# Normal
# All real numbers from negative infinity to positive infinity
# mu (all real numbers), sigma^2 (all real numbers greater than 0)
# Continuous
# Any morphometric data randomly sampled from a population not having recently
# experienced a bottleneck event
```

HINT: Wikipedia is a wonderful source for this type of inquiry.
HINT: You may need search online to find example data. See if you can find papers that describe their data using these distribution. Look for phrases like *our data was Poisson distributed*.

QUESTION 2
We say that sample data is **overdispersed** when its observed variance is greater than its expected variance given the parameters of the population distribution from which it is assumed to have been drawn.

(a) Under what parametric conditions would a sample from a Poisson distribution be considered overdispersed?

(b) What are the dangers of undetected overdispersion in a dataset? **HINT:** Think about the accuracy of the PMF/PDF.

(c) Why do statisticians often advise modeling overdispersed Poisson data using the negative binomial distribution? What added flexibility does this distribution offer?

QUESTION 3
Let's examine the concept of kurtosis to understand how knowledge of this parameter can inform the way in which we handle outliers. The kurtosis of any dataset can be computed using the following formula:

$$Kurtosis = \frac{1}{N} * \sum_{i=1}^{N} \frac{(x_i - \mu)^4}{(\sigma^2)^2}$$

where μ is the sample mean, σ^2 is the sample variance, N is the sample size, and x_i is some value from the sample.

(a) *Create a vector containing 10,000 draws from a normal distribution with $\mu = 3$ and $\sigma^2 = 1.5$, and calculate its kurtosis. I've reproduced the kurtosis function in R for you below.*

```
kurtosis <- function(x){ # x is the vector of sample data
  mu <- mean(x)
  var <- var(x)
  N <- length(x)
  numer <- (x-mu)^4 # x-mu subtracts mu from each element of x
  denom <- var^2
  kurt <- (1/N)*sum(numer/denom)
  return(kurt)
}
```

(b) *Calculate the kurtosis of a 10,000-draw sample from a normal distribution with $\mu = 12$ and $\sigma^2 = 50$.*

The kurtosis of the normal distribution, irrespective of parameters, is equal to 3. Because the normal distribution is so familiar, the kurtosis metrics of all other distributions are typically reported relative to the normal distribution, that is, *kurtosis* $- 3$ (called the *excess* kurtosis).

Here I'll briefly introduce the concept of **moment matching**, an idea that we'll develop further at the end of the semester. For most distributions, the parameters do not give the mean and variance as is true for the normal distribution. There are, however, instances where we'd like to produce different distributions with the same first and second moment. We can match moments using distribution-specific formulae. For reasons that are beyond the scope of this course, the mean, variance, and kurtosis of a t distribution are only defined for $\nu > 4$, so let's match to a normal distribution when $\nu = 10$.

(c) *When $\nu = 10$, what is the mean (expected value) of a t distribution?*

(d) *Look up the formula that gives the variance of a t distribution, and create a function in R called* `tvar()` *that takes ν as an argument and returns variance.*

(e) *Calculate the variance of a t distribution with $\nu = 10$.*

(f) *Using the mean from (c) and the variance from (e), create a 100,000-draw sample from a normal distribution.*

(g) *Create 100,000-draw sample from a t distribution using* `rt()`*. What is the excess kurtosis of this sample?*

Let's visualize this difference in kurtosis. We've produced large, representative samples from a normal distribution and a t distribution characterized by the same mean and variance. Using the built in probability density functions for normal and t distributions, we can plot both datasets together (where density is on the y-axis) to observe how a difference in kurtosis can change the shape of a distribution. Here's how we can plot the t-distributed data:

```
sorted <- sort(t) # where t is a 100,000 element vector from a t distribution with
                  # nu = 10. The sort() function organizes the data into ascending
                  # order, which is required when plotting a line graph in R
plot(sorted, dt(sorted, 10), type = 'l') # here the first argument of plot() is our
                                         # sample data, and the second argument gives
                                         # the density of each data point based on a
                                         # t distribution with nu = 10
```

(h) *Using the* `lines()` *function, plot the normal sample data on top of the t sample data. Color the normal line red.*

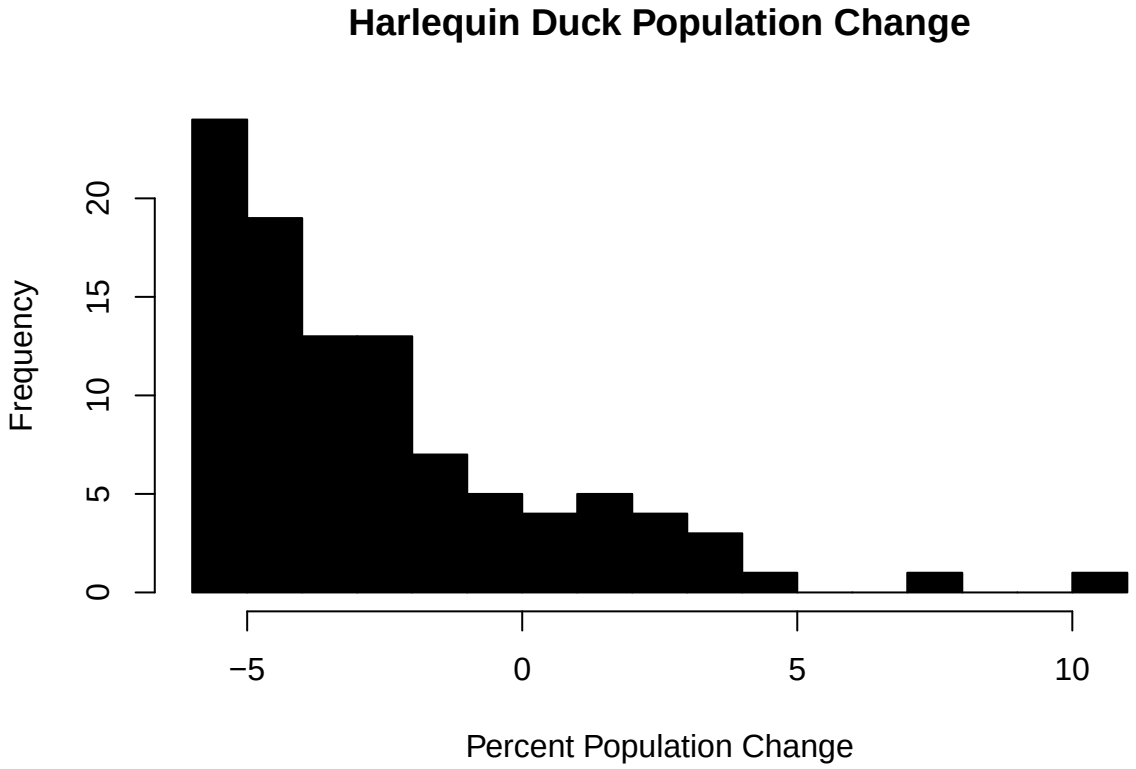
As we'll see in our upcoming unit on linear modeling, simple linear regression assumes that the sample data is normally distributed. This assumption is not well met when a dataset includes outliers because the normal

distribution does not assign much density to values far from the mean. To visualize this better, add the following argument to your plot function and run it again: `ylim=c(0, 0.001)`.

(i) *Look at difference in tail length between the two distributions. Why might it be true to say that the t-distribution is more robust to outliers than the normal distribution?*

QUESTION 4

For the past two years I've estimated the size (number of individuals) of 100 randomly chosen Harlequin Duck populations on the western coast of Greenland, and computed the year-over-year change in population size as a percentage. A histogram of the data is presented below.



(a) *Load in the duck dataset ("ducks.csv") using* `read.csv()`.

(b) *Why can this data not possibly have been drawn from any of the probability distributions that we've discussed so far this semester? Provide a brief explanation for each distribution.*

- Normal
- Gamma
- Beta
- Poisson
- Negative binomial
- t

(c) *Based on this sample data, what is the probability density that a population would undergo a 7% increase in size over the 2-year study period?* **HINT:** Use the `density()` function with the argument `n = 10000` to produce a high-resolution density vector.

In general, we can approximate the *probability* of any value in the support of a parametric distribution by taking the integral of the PDF over a very small interval containing the value of interest. The smaller the interval, the more accurate the approximation. Our data does not appear to be goverened by the parameters of any probability distribution that we are familiar with, and so we have no explicit PDF to integrate over.

We can, however still compute an approximation for the probability of a given value by calculating the area of a rectangle under the density curve containing the value of interest.

EXAMPLE

```
# We would like to know the approximate probability of 1.25 drawn from a normal
# distribution with mean = 0 and variance = 1.

# 1) Select a small interval containing 1.25 and calculate the density of the upper
#    and lower bounds. The midpoint of these densities will give the height of our
#    rectangle.

upper <- dnorm(1.251, 0, 1)
lower <- dnorm(1.249, 0, 1)
height <- (upper + lower)/2

# 2) The width of the rectangle is given by the difference between the upper and
#    lower bounds of the interval.

width <- 1.251 - 1.249

# 3) The approximate probability of 1.25 is then given by the area of the resulting
#    rectangle.

height * width
```

```
## [1] 0.0003652983
```

```
# We can check to see how close our approximation is by using R's integrate() function
# with the upper and lower bounds that we've chosen

integrate(dnorm, mean = 0, sd = 1, lower = 1.249, upper = 1.251)$value
```

```
## [1] 0.0003652982
```

(d) *Approximate the probability of a duck population experiencing a 7% population increase across the 2-year study.* **HINT:** Use the code `which(dens$x >= 7)[1]` to return the vector index of the value closest to, but just larger than 7 where "dens" is the density object, and "dens$x" returns the 10000-element vector of x values at which the density was computed (based on the sample data).

(e) *Use a bootstrapping algorithm to estimate a 95% CI around the probability density of observing a 7% Harlequin Duck population increase on the western coast of Greenland between the years of 2019 and 2020. +5 bonus points if you correcly estimate the probability instead of the probability density*