R Tutorial for STAT 350 Lab 3

Author: Leonore Findsen, Jeremy Troisi

1. Generate random samples from a normal distribution

We are going to generate random samples from a variety of different distributions in this laboratory. The following code is for the normal distribution. I will also be providing similar code for the other distributions that we will be using in part three of this tutorial. The function that is used in R is `rnorm(number of data points, mu =, sigma =)`.

a) Generate 20 random numbers from a normal distribution with $\mu = 572$ and $\sigma = 51$ and calculate the mean and standard deviation of the data set.

Solution:

```
#rnorm(n,mean=x,sd=y) generates n random numbers
#  that belong to the normal distribution with mean of x
#  and standard deviation of y.
RandomData <- rnorm(20, mean = 572 ,sd = 51)
RandomData
mean(RandomData)
sd(RandomData)
title <- "RandomData" # This variable is used below
```

```
> RandomData
 [1] 526.4139 608.2455 555.0126 476.8590 595.5401 603.8924 557.9370
627.6589 531.4398
[10] 510.8164 546.7202 556.9589 573.7344 574.0468 603.6946 476.5665
505.6171 531.2527
[19] 600.8774 659.8651

> mean(RandomData)
[1] 561.1575
> sd(RandomData)
[1] 49.33362
```

Note: Each time that the program is run, you will get different values, means, and standard deviations.

2. Determine if a distribution is normal

b) Make a histogram of the data in part (a) and visually assess if the normal density curve and the histogram density estimate (i.e., the kernel) are similar.
c) Make a normal probability plot of the data in part (a) and visually assess if the plotted points are randomly scattered below and above the line without a discernable pattern.

Solution:

Remember that you may have to run the appropriate install.packages() and will have to run library(ggplot2) before you can run the code. I am doing the problem with the data from part (a), but it doesn't matter what data is used. I am adding titles to the plots, making it easier for you to identify which graph belongs to which part.

R Tutorial for STAT 350 Lab 3

Author: Leonore Findsen, Jeremy Troisi

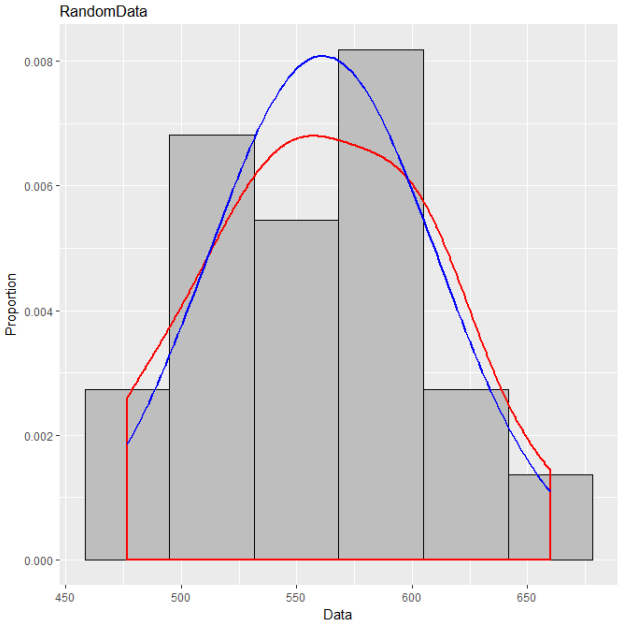
The ggplot function expects a data frame as input. If we wish to plot data that is stored in a vector (such as RandomData above), we may use the data.frame function to turn it into a data frame temporarily for ggplot.

Remember to set the number of histogram bins appropriately.

```
library(ggplot2)
#First, we need the sample mean and standard deviation to help draw
# the theoretical normal
xbar <- mean(RandomData)
s <- sd(RandomData)
#Histogram
windows()
#We have to use length() instead of nrow() because RandomData
# is a vector.
# If the data set is part of a data.frame (table) already, you
# should use nrow().
ggplot(data.frame(RandomData=RandomData), aes(x=RandomData)) +
  geom_histogram(aes(y=..density..),bins = sqrt(length(RandomData))+2,
                 fill = "grey",col = "black") +
  geom_density(col = "red", lwd = 1) +
  stat_function(fun=dnorm,  args=list(mean=xbar, sd=s), col="blue",
                lwd = 1) +
  ggtitle(title) +
  xlab("Data") +
  ylab("Proportion") # Need to use proportion with density curves
#
#Normal Probability Plot (AKA: QQ Plot)
# First, we need the sample mean and standard deviation to help draw
# the comparison line.  We will use the values from above.
windows()
ggplot(data.frame(RandomData=RandomData), aes(sample=RandomData)) +
  stat_qq() +
  geom_abline(slope = s, intercept = xbar) +
  ggtitle(title)
```

b) Make a histogram of the data in part (a) and visually assess if the normal density curve and the histogram density estimate are similar.

Solution:

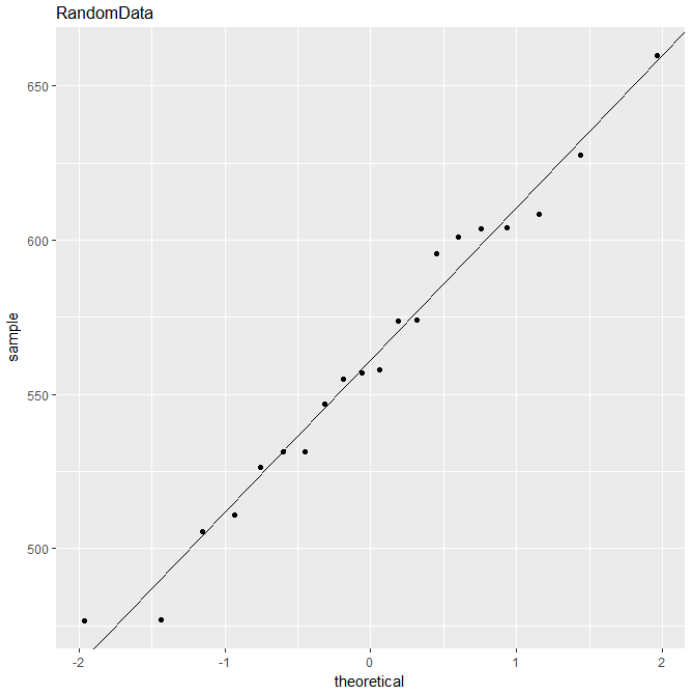


There are two complimentary ways to determine if this distribution is normal. 1) You can assess whether the blue normal curve is 'close' to the red smoothed curve. 2) You can look at the symmetry, modality and tails of the histogram and red smoothed curve to see if it is possible that the distribution is normal. When you are assessing normality from a histogram, always use both methods. In this case, the two curves are similar, the distribution is unimodal, and looks approximately symmetric. Therefore, this distribution resembles a normal distribution. When you are using a histogram to determine normality, please always include the two extra curves.

Author: Leonore Findsen, Jeremy Troisi

c) Make a normal probability plot of the data in part (a) and visually assess if the sample quantiles are randomly scattered below and above the line without a discernable pattern.

Solution:



Since the data points are randomly scattered below and above the line without a pattern, the randomly generated data does not appear to deviate substantially from a normal distribution.

3. Generate random samples for right skewed, left skewed, short tailed, long tailed distributions

The following four distributions illustrate different types of skewness and tails. We will generate random samples from each of them respectively, and compare them with the normal distribution using the visual methods introduced in part (2).

Right skewed: Exponential distribution ($\lambda = 5$)
Left skewed: Beta distribution (on [0,1], $\alpha = 7$, $\beta = 0.8$)
Short tailed: Uniform (on [a = -3, b = 3])
Long tailed: t-distribution (df = 1)

The following code is used for the above distributions. Use each random number generator to define RandomData, then pass RandomData to your code above to make the histograms and the probability plots for each set of random numbers.

R Tutorial for STAT 350 Lab 3

Author: Leonore Findsen, Jeremy Troisi

```
#n is the number of data points, this is constant
n <- 100

#nonnormal distributions
# right skewed: exponential distribution (lambda=5)
# left skewed: Beta distribution (on [0,1], alpha = 7, beta = 0.8)
# long tailed: t-distribution (df = 1)
# short tailed: Uniform (on [-3,3]);
right <- rexp(n,rate=5)
left <- rbeta(n,7,0.8)
short <- runif(n,min=-3,max=3)
long <- rt(n,df=1)

#There are only two things that need to be changed in the code below.
#1) Change which data set that you will be using (in RandomData).
#  I have it set for right, you will need to change this to
#  left, long, short as appropriate.
#2) The first (and second) word in the main title needs to be changed.
#  I have it set to right, while you will need to change this to left,
#  long tailed, or short tailed as appropriate.

RandomData <- right
title <- "Right Skewed Distribution"

# the rest of the code is the same as above and will not be repeated.
```

No output is provided.