R Tutorial for STAT 350 Lab 3

Author: Leonore Findsen, Jeremy Troisi

1) Generate random samples using a normal distributions

We are going to generate random samples from a number of different distributions in this laboratory. The following code is for the normal distribution which is the only one that we have discussed so far in class. I will also be providing a similar code for the other distributions that we will be using in part C. 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)
mean(RandomData)
sd(RandomData)
```

```
> mean(RandomData)
[1] 587.91
> sd(RandomData)
[1] 46.96685
```

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

2) Determine if a distribution is normal

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

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:

I am doing the problem with the data from part (a), but it doesn't matter what data is used.

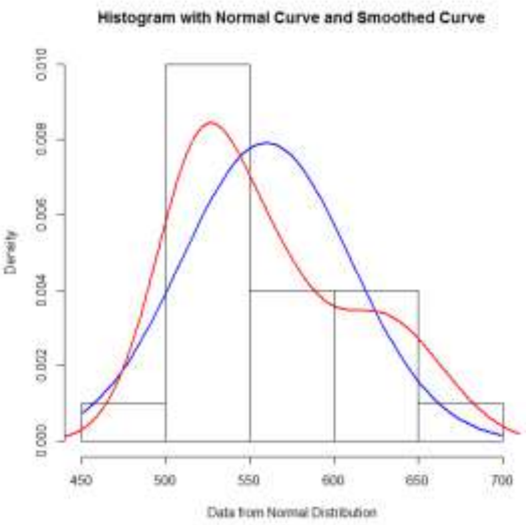
When you are using a histogram to determine normality, there are two extra lines that should be used. The blue line is the normal distribution with the estimated μ and σ; the red line is the density curve (smoothed curve of the histogram itself). Note that code for the histogram has changed.

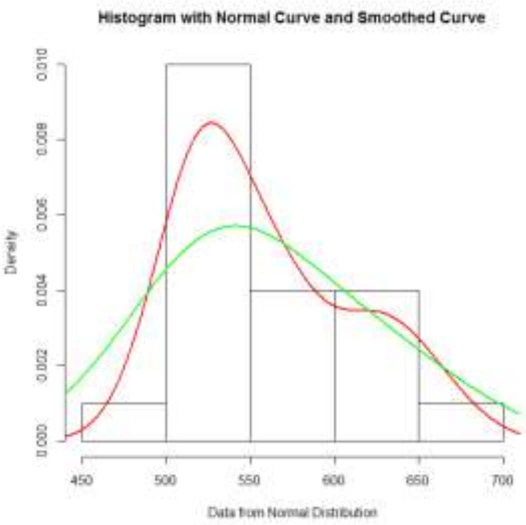
R Tutorial for STAT 350 Lab 3

Author: Leonore Findsen, Jeremy Troisi

```
#generating the histogram with blue line being the normal distribution
#  and red line the smoothed curve.
# freq = FALSE means that we are plotting relative frequencies or
# densities
std<-sd(RandomData)
m <- mean(RandomData)

windows()

#Histogram
#  You can change the titles by using main, xlab, and ylab.
# freq = FALSE is required if you will be adding the extra two lines.
hist(RandomData, xlab="Data from Normal Distribution", freq = FALSE,
main="Histogram with Normal Curve and Smoothed Curve")

# this command plots the normal curve
#  dnorm() plots the density curve, x is the density of quantiles
#  add=TRUE: adds on top of the previous graph
curve(dnorm(x, mean=m, sd=std), col="blue", lwd=2, add=TRUE)

#this command plots the smooth curve (density)
#  If there are too many peaks on the curve, increase the value of
#    adjust
lines(density(RandomData, adjust=1),col = "red", lwd=2)

#If the graphics area in RStudio is too small, you can use the
#    following command to open a window outside of RStudio. You can
#    have multiple windows open.
windows()

#Normal Probability Plot
#qqnorm plots the points for the normal probability plot and qqline
#  includes a line from Q1 to Q3 to help you determine if the data set
#  is normal or not.
qqnorm(RandomData,main="Normal Quantile Plot for normal distribution")
qqline(RandomData)
```

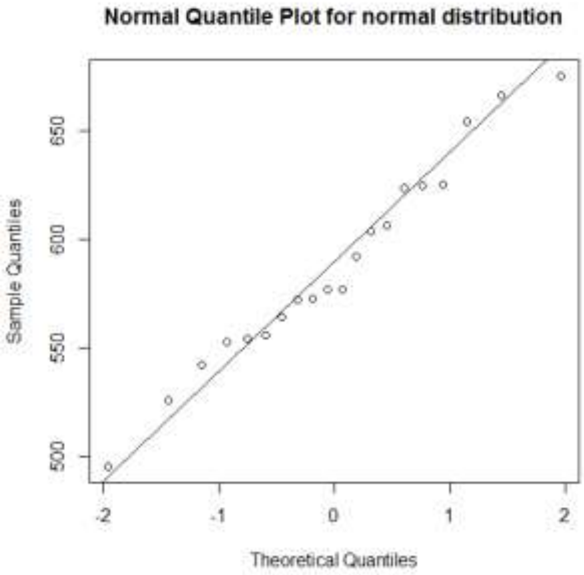
b) Make an appropriate histogram of the data in part (a) and visually assess if the normal density curve and the histogram density estimate are similar.


Histogram with Normal Curve and Smoothed Curve


Histogram with Normal Curve and Smoothed Curve

Since the blue normal curve is close to the red smoothed curve, the randomly generated normal data appears to be a normal distribution.	This curve shows the difference between adjust = 1 (red) and adjust = 2 (green). This is used if the normal red line has too many peaks to be able to compare with the blue normal curve.

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.


Normal Quantile Plot for normal distribution

Since the sample quantiles are randomly scattered below and above the line without a pattern, the randomly generated normal data appears to be a normal distribution.

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

The specific distributions used are:
right skewed: exponential distribution ($\lambda = 2$) with $\mu = 0.5$ and $\sigma = 0.5$
left skewed: Beta distribution (on [0,1], $\alpha = 2$, $\beta = 0.5$) with $\mu = 0.8$ and $\sigma = 0.0457$
short tailed: Uniform (on [a = 0, b = 2]) with $\mu = 1$ and $\sigma = 0.3333$
long tailed: Standard Cauchy with median = 0 and σ is not defined.

Note: The Cauchy distribution has extremely straggly long tails, so much so that the mean is undefined clearly making the median a better descriptor of the center.

The following code is used for the above distributions.

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

#nonnormal distributions
# right skewed: exponential distribution (lambda=2) with mu=0.5 and
#      sigma=0.5
# left skewed: Beta distribution (on [0,1],  alpha = 2, beta = 0.5)
#        with mu = 0.8 and sigma = 0.0457
# short tailed: Uniform (on [0,2]) with mu = 1 and sigma = 0.3333;
# long tailed: Standard Cauchy with median = 0 and sigma = ?
right <- rexp(n,rate=2)
left <- rbeta(n,2,0.5,ncp=2)
short <- runif(n,min=0,max=2)
long <- rcauchy(n,location=0,scale=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 skewed, you will need to change this to
#  left, short, long as appropriate.
#2) The first word in the main title needs to be changed. I have it set
#  to right skewed, you will need to change this to left, long, or
#  short as appropriate.

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

windows()

#generating the histogram with blue line being the normal distribution
#  and red line the smoothed curve.
std<-sd(RandomData)
m <- mean(RandomData)
hist(RandomData, xlab="Data", freq = FALSE, main=title)
```

```
curve(dnorm(x, mean=m, sd=std), col="blue", lwd=2, add=TRUE)

#Notice that we recommend that you use adjust = 3 here. However, if
#  this is too smooth, feel free to reduce that number
lines(density(RandomData,adjust=3),col = "red", lwd=2)

windows()

#plots the qqplot with line on a separate plot
qqnorm(RandomData,main=title)
qqline(RandomData)
```

No output is provided. Every time the code is run, different output will be produced.