CIND 123 Winter 2018 - Assignment #3

Brendan Dagys

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

Use RStudio for this assignment. Edit the file assignment-3.Rmd and insert your R code where wherever you see the string “INSERT YOUR ANSWER HERE”

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.

## Sample Question and Solution

Use seq() to create the vector .

#Insert your code here.  
seq(2,20,by = 2)

## [1] 2 4 6 8 10 12 14 16 18 20

This assignment questions 1 - 4 make use of data that is provided by the mosaic package. (install mosaic package and load KidsFeet using data(KidsFeet) ).

#install.packages('mosaic')  
library(mosaic)

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

## Loading required package: lattice

## Loading required package: ggformula

## Loading required package: ggplot2

##   
## New to ggformula? Try the tutorials:   
## learnr::run\_tutorial("introduction", package = "ggformula")  
## learnr::run\_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData

## Loading required package: Matrix

##   
## The 'mosaic' package masks several functions from core packages in order to add   
## additional features. The original behavior of these functions should not be affected by this.  
##   
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.

##   
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':  
##   
## mean

## The following objects are masked from 'package:dplyr':  
##   
## count, do, tally

## The following objects are masked from 'package:stats':  
##   
## binom.test, cor, cor.test, cov, fivenum, IQR, median,  
## prop.test, quantile, sd, t.test, var

## The following objects are masked from 'package:base':  
##   
## max, mean, min, prod, range, sample, sum

data(KidsFeet)

## Question 1 - 30%

In this question you will explore the KidsFeet dataset available in the datasets library in the mosaic package.

1. Display the first 6 rows of the KidsFeet dataset.

head(KidsFeet)

## name birthmonth birthyear length width sex biggerfoot domhand  
## 1 David 5 88 24.4 8.4 B L R  
## 2 Lars 10 87 25.4 8.8 B L L  
## 3 Zach 12 87 24.5 9.7 B R R  
## 4 Josh 1 88 25.2 9.8 B L R  
## 5 Lang 2 88 25.1 8.9 B L R  
## 6 Scotty 3 88 25.7 9.7 B R R

1. Display the type of each column of the airquality dataset, use only one function in R to do so.

#sapply(KidsFeet, class) <---- ASSUMING YOU ARE ASKING FOR AIRQUALITY DATASET  
sapply(airquality, class)

## Ozone Solar.R Wind Temp Month Day   
## "integer" "integer" "numeric" "integer" "integer" "integer"

1. Use a histogram to assess the normality of the width variable.(In order to get the output diagram inserted in your answer use attach(dataframe name))

attach(KidsFeet)  
hist(width, main = "Histogram of Width", xlab = "Width")

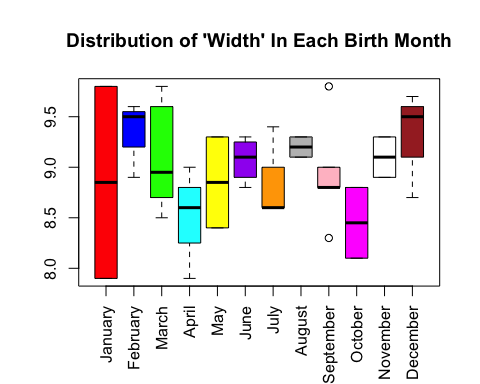


1. Does it appear normally distributed? Why?

# The distribution does not appear to be normally distributed. This is because it does not at all closely resemble the normal distribution. There seem to be two or three peaks in the histogram. Perhaps with a larger sample, the increase in data would create a normal distribution.

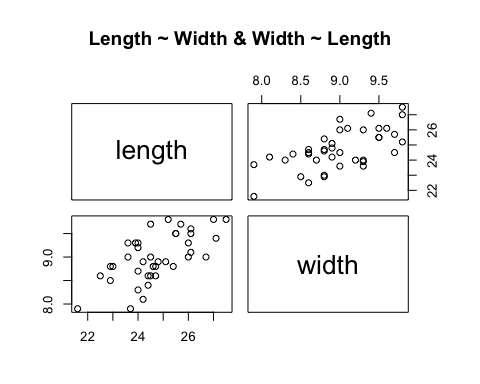
1. Create a boxplot which shows the distribution of width in each birth month. Use different colors for each birth month.

boxplot(width ~ birthmonth, KidsFeet, main = "Distribution of 'Width' In Each Birth Month", col = c("red", "blue", "green", "cyan", "yellow", "purple", "orange", "grey", "pink", "magenta", "white", "brown"), xaxt = "n")  
axis(1, 1:12, month.name, las = 2)



1. Create one scatter plot matrix of the numeric variables (length, width) within the KidsFeet dataset. (Hint investigate pairs())

pairs(KidsFeet[c(4, 5)], main = "Length ~ Width & Width ~ Length")



## Question 2 - 20%

a)Use simulation to estimate the mean and variance of a binomial random variable with n = 45 and p = 0.32. One time use 100 samples and the other time use 10000 samples.

set.seed(1)  
a = rbinom(100, 45, 0.32)  
cat("The mean of 100 samples is:", mean(a), "\n")

## The mean of 100 samples is: 14.55

cat("The variance of 100 samples is:", var(a), "\n\n")

## The variance of 100 samples is: 7.361111

b = rbinom(10000, 45, 0.32)  
cat("The mean of 10,000 samples is:", mean(b), "\n")

## The mean of 10,000 samples is: 14.395

cat("The variance of 10,000 samples is:", var(b))

## The variance of 10,000 samples is: 10.04778

print('Test')

## [1] "Test"

1. Calculate the values using the theory (state the value and the equation in your answer),compare the values you get with the values you got in (a), write one sentence summarizing the comparision. Explain the difference between 100 samples and 10000 samples and which one seems to be more accurate and why?

#Using the theories to calculate the values as below, we get expected values of 14.4 for the mean, and 9.792 for the variance. The calculations using 10,000 samples are more accurate (closer to the theory mean and standard deviation) because we have more information and hence less uncertainty.  
cat("Theoretical values:\n\n")

## Theoretical values:

cat("Mean =", 45 \* 0.32, "\n")

## Mean = 14.4

cat("Variance =", 45 \* 0.32 \* (1 - 0.32))

## Variance = 9.792

## Question 3 - 30%

1. If there are twelve customers entering a mall per minute on average, find the probability of having seventeen or more customers entering the mall in a particular minute.

cat("Probability of having 17 or more customers enter the mall in a particular minute:", ppois(16, 12, lower.tail = 0))

## Probability of having 17 or more customers enter the mall in a particular minute: 0.101291

1. Estimate the mean and variance of a Poisson random variable in the previous question by simulating 100 and 10,000 Poisson random numbers.

set.seed(1)  
c = rpois(100, 12)  
cat("The mean of 100 samples is:", mean(c), "\n")

## The mean of 100 samples is: 12.33

cat("The variance of 100 samples is:", var(c), "\n\n")

## The variance of 100 samples is: 10.72838

d = rpois(10000, 12)  
cat("The mean of 10,000 samples is:", mean(d), "\n")

## The mean of 10,000 samples is: 12.0174

cat("The variance of 10,000 samples is:", var(d), "\n\n")

## The variance of 10,000 samples is: 12.08091

1. Compare the mean value you got in (b),with the one stated in the question. Write one sentence summarizing the comparision. Explain the difference between 100 samples and 10,000 samples and which one seems to be more accurate and why?

# The mean stated in the question is 12 customers per minute. Our results were 12.33 customers per minute (when using 100 samples) and 12.0174 (when using 10,000 samples). Both results are not exactly equal to the true mean. However, the result from 10,000 samples appears to be more accurate. This is because using a larger sample provides us more information. The more variable the population, the greater the uncertainty in our estimate. When we have more information due to a larger sample size, the uncertainty reduces and our answer converges on the true mean. The formula for SE is sigma/sqrt(n). It is easy to see that when n increases, the denominator increases (and hence, SE decreases).

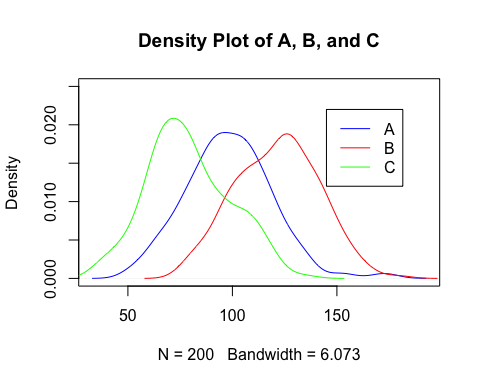
## Question 4 - 20%

1. Generate normally distributed random numbers for three categories: A (n = 200, mean = 100, sd = 20), B (n = 200, mean = 120, sd = 20), and C (n = 200, mean = 80, sd = 20)

A = rnorm(200, 100, 20)  
B = rnorm(200, 120, 20)  
C = rnorm(200, 80, 20)

1. Combine all the three categories in one dataframe then generate a density plot colored by category.

mydataframe = data.frame(A, B, C)  
plot(density(mydataframe$A), col = "blue", main = "Density Plot of A, B, and C", ylim = c(0.000, 0.025))  
legend(145, 0.022, c("A", "B", "C"), col = c("blue", "red", "green"), lty = 1)  
lines(density(mydataframe$B), col = "red")  
lines(density(mydataframe$C), col = "green")



END of Assignment #3.