````
---
title: STAT 457 Homework 02
author: Martha Eichlersmith
date: 2019-10-07
output:
  pdf_document:
    fig_caption: yes
header-includes:
  - \usepackage{color}
  - \usepackage{mathtools}
---
```{r, echo=FALSE, results="hide", warning=FALSE, message=FALSE}
library(ggplot2)
library(readr)
library(gridExtra)
library(grid)
library(png)
library(downloader)
library(grDevices)
library(latex2exp)
library(knitr)
library(leaps)
library(directlabels)
library(diffusr)
library(MASS)
library(invgamma)
library(condMVNorm)
```

```{r, echo=FALSE}
#FUNCTION FOR PROBLEMS 0 & 1
#Sinlge Random Walk - Discrete (output is each step)
Single_Disc_Walk_2d <-function(n_steps)
{ rw <- matrix(0, ncol = 2, nrow = n_steps)
  indx <- cbind(seq(n_steps), sample(c(1, 2), n_steps, TRUE))
  rw[indx] <- sample(c(-1, 1), n_steps, TRUE)
  rw[1,1] <- 0
  rw[1,2] <- 0
  rw[,1] <- cumsum(rw[, 1]) # cumsum the columns
  rw[,2] <- cumsum(rw[, 2]) # cumsum the columns
  return(as.data.frame(rw)) } # return values of each step in
random walk
#Multiple Discrete Random Walks, output is coordinates to end
points
Disc_Walk_2d <- function(n_steps, n_walks){
````
````

````
for(i in 1:n.walks){
  theta <- runif(n_steps, 0, 2*pi)
  dx <- jump*cos(theta)
  dy <- jump*sin(theta)
  x[i] <- sum(dx)
  y[i] <- sum(dy) }
coord <- data.frame(x,y)
return(coord) }
#calculate the percetage of end points in each quadrant, output
is vector of percentages
Quad_pct <- function(coord){
Q1.points <- subset(coord, coord[1] >= 0 & coord[2] >=0)
Q2.points <- subset(coord, coord[1] <  0 & coord[2] > 0)
Q3.points <- subset(coord, coord[1] <= 0 & coord[2] <=0)
Q4.points <- subset(coord, coord[1] >  0 & coord[2] < 0)
Q1.pct <- nrow(Q1.points) / n.walks
Q2.pct <- nrow(Q2.points) / n.walks
Q3.pct <- nrow(Q3.points) / n.walks
Q4.pct <- nrow(Q4.points) / n.walks
prop<-c(Q1.pct, Q2.pct, Q3.pct, Q4.pct)
return(prop)}
```

## Problem 0.
```{r, fig.width=10, fig.height=6, echo=FALSE}
set.seed(010000) #Homework 01 | Problem 00 | Part 00
walk <- Single_Disc_Walk_2d(50)
ggplot(aes(x=walk[,1], y=walk[,2]),data=walk) + ggtitle("Graph a
random walk")+
  geom_path(color="blue", size=1) +
  geom_hline(yintercept=0, linetype="dashed") +
  geom_vline(xintercept=0, linetype="dashed") +
  geom_point(x=0, y=0, color="darkgreen", size=5) +
  geom_point(x=walk[nrow(walk),1], y=walk[nrow(walk),2],
color="red", size=5) +
  theme(axis.title.y=element_blank(),
axis.title.x=element_blank()    )
```

## Problem 1a.
Simulate 10,000 random walks of length 50 in $\mathbb{R}^2$
starting at (0, 0). Compute the proportion of walks which end up
in each of the four quadrants.  What values would you expect for
these proportions? Do the observed proportion vary significantly
from the expected values?
````
````

Similar to 1a, the percetnages are very close to 0.25.

Problem 1c.
Repeat 1a and 1b using random walks on a lattice in $\mathbb{R}^2$.
```{r, echo=FALSE}
set.seed(010103) #Homework 01 | Problem 01 | Part 03.
n.steps <- 50
n.walks <- 10000
coord <- Disc_Walk_2d(n.steps, n.walks)
quad.pct <- Quad_pct(coord)
quad.pct
chisq.test(quad.pct)

set.seed(010104) #Homework 01 | Problem 01 | Part 04 .
n.steps <- 500
n.walks <- 10000
coord <- Disc_Walk_2d(n.steps, n.walks)
quad.pct <- Quad_pct(coord)
quad.pct
chisq.test(quad.pct)
```
Both of these situations have percentages that are close to 0.25. It is interesting to point out that the χ^2 values are larger for the lattice random walks rather than the continous random walks.

\newpage
Problem 2.
From each of the following distributions: (i) draw 10,000 deviates; (ii) compute the sample average and sample SD and compare these numbers to the true values; (iii) draw a histogram of the deviates along with the plot of the true density. Binomial(14, .90); Beta(.5, .5); Gamma(12, 2); Inverse.Gamma(12, 2); χ^2 on 2 df; χ^{-2} on 2 df.

Problem 2a: Binomial(14, .9)
```{r, fig.width=12, fig.height=6, message=FALSE, echo=FALSE}
set.seed(010201) #Homework 01 | Problem 02 | Part 01
r <- 5 #rounding decimals
binom.fun <- function(d, n, p){
dist <- paste("Binomal - ", d, "devaites")
X <- rbinom(d, n, p)
X.df <- data.frame(X=X)
Name <- c("Mean", "SD")
```

````
```{r, fig.width=12, fig.height=6, message=FALSE, warnings=FALSE,
error=FALSE, echo=FALSE}
set.seed(010202) #Homework 01 | Problem 02 | Part 02
r <- 5 #rounding decimals

beta.fun <- function(d, a, b){
dist <- paste("Beta - ", d, "devaites")
X <- rbeta(d, a, b)
X.df <- data.frame(X=X)
true <- c(round(a / (a + b), r), round(sqrt(a*b / ((a+b)^2 * (a +
b +1))), r))
Name <- c("Mean", "SD")
sample <- c(round(mean(X), r), round(sd(X), r))
compare <- data.frame("Name"=Name, "Sample"=sample, "True"=true)
plot <- ggplot(X.df, aes(X)) + geom_histogram(aes(y=..density..))
+
  ggtitle(dist)+
  xlab("") +
  coord_cartesian(ylim=c(0, 2.5))+
  annotation_custom(tableGrob(compare), ymin=1) +
  stat_function(
  fun = dbeta,
  args = list(shape1=a, shape2=0.5),
  lwd = 1,
  linetype="dashed",
  col = 'blue'
  )
return(plot)}

beta1 <- beta.fun(10000, .5, .5)
beta2 <- beta.fun(100000, .5, .5)
beta3 <- beta.fun(1000000, .5, .5)
grid.arrange(beta1, beta2, beta3, nrow = 1)
```

### Problem 2c: Gamma(12, 2)
```{r, fig.width=12, fig.height=6, message=FALSE, echo=FALSE}
set.seed(010203) #Homework 01 | Problem 02 | Part 03
r <- 5 #rounding decimals

gamma.fun <- function(d, a, b){
dist <- paste("Gamma - ", d, " devaites")
X <- rgamma(d, a, b)
X.df <- data.frame(X=X)
true <- c(round(a/b, r), round(sqrt(a/(b^2)), r))
````
````

````
gamma3 <- gamma.fun(1000000, 12, 2)
grid.arrange(gamma1, gamma2, gamma3, nrow = 1)
```

### Problem 2d: Inverse.Gamma(12, 2)
```{r, fig.width=12, fig.height=6, message=FALSE, echo=FALSE}
set.seed(010204) #Homework 01 | Problem 02 | Part 04
r <- 5 #rounding decimals

invgamma.fun <- function(d, a, b){
dist <- paste("Inverse Gamma - ", d, " devaites")
X <- rinvgamma(d, a, b)
X.df <- data.frame(X=X)
true <- c(round(b/(a -1), r), round(sqrt( b^2 / ((a - 1)^2 * (a
-2))), r))
Name <- c("Mean", "SD")
sample <- c(round(mean(X), r), round(sd(X), r))
compare <- data.frame("Name"=Name, "Sample"=sample, "True"=true)
plot <- ggplot(X.df, aes(X)) + geom_histogram(aes(y=..density..))
+
  ggtitle(dist)+
  xlab("") +
  #ylim(0, 1.5)+
  annotation_custom(tableGrob(compare), ymin=5, xmin=.2) +
  stat_function(
  fun = dinvgamma,
  args = list(shape=a, rate=b, log=FALSE),
  lwd = 1,
  linetype="dashed",
  col = 'blue'
  )
return(plot)}

invgam1 <- invgamma.fun(10000, 12, 2)
invgam2 <- invgamma.fun(100000, 12, 2)
invgam3 <- invgamma.fun(100000, 12, 2)
grid.arrange(invgam1, invgam2, invgam3, nrow = 1)
```


### Problem 2e: $\chi^2$ on 2 df
```{r, fig.width=12, fig.height=6, message=FALSE, echo=FALSE}
set.seed(010205) #Homework 01 | Problem 02 | Part 05
r <- 5 #rounding decimals
````

````
  linetype="dashed",
  col = 'blue'
  )
return(plot)}

chisq1 <- chisq.fun(10000, 2)
chisq2 <- chisq.fun(100000, 2)
chisq3 <- chisq.fun(1000000, 2)
grid.arrange(chisq1, chisq2, chisq3, nrow = 1)
```

### Problem 2f: $\chi^{-2}$ on 2 df
```{r, fig.width=12, fig.height=6, message=FALSE, echo=FALSE}
set.seed(010206) #Homework 01 | Problem 02 | Part 06
r <- 5 #rounding decimals

invchisq.fun <- function(d, df){
dist <- paste("Inverse Chi-Squared - ", d, " devaites")
X <- rinvchisq(d, df)
X.df <- data.frame(X=X)
true <- c("DNE when df <=2", "DNE when df <=4")
Name <- c("Mean", "SD")
sample <- c(round(mean(X), r), round(sd(X), r))
compare <- data.frame("Name"=Name, "Sample"=sample, "True"=true)
plot <- ggplot(X.df, aes(X)) + geom_histogram(aes(y=..density..))
+
  ggtitle(dist)+
  xlab("") +
  #ylim(0, 1.5)+
  annotation_custom(tableGrob(compare)) +
  stat_function(
  fun = dinvchisq,
  args = list(df=df),
  lwd = 1,
  linetype="dashed",
  col = 'blue'
  )
return(plot)}

invchi1 <- invchisq.fun(10000, 2)
invchi2 <- invchisq.fun(100000, 2)
invchi3 <- invchisq.fun(1000000, 2)
grid.arrange(invchi1, invchi2, invchi3, nrow = 1)
```
````
````

```
mu <- c(0,0,0)
sig <- matrix(c(1,4.5,9,4.5,25,49,9,49,100),nrow=3, ncol=3)
mvn.fun <- function(d, mu, sig){
dist <- paste("Multivariate Normal - ", d, "devaites")
MVN.data <-mvrnorm(d, mu, sig, tol = 1e-6, empirical = FALSE,
EISPACK = FALSE)
MVN.data<-data.frame(MVN.data, nrow = d, ncol = 3)
colnames(MVN.data)<-c("X1","X2","X3")
X1 <- MVN.data$X1
true <- c(0, 1)
Name <- c("Mean", "SD")
sample <- c(round(mean(X1), r), round(sd(X1), r))
compare <- data.frame("Name"=Name, "Sample"=sample, "True"=true)
plot <- ggplot(MVN.data, aes(X1)) +
geom_histogram(aes(y=..density..)) +
  ggtitle(dist)+xlab("") +
  annotation_custom(tableGrob(compare),xmax=0,ymin =0.05) +
  stat_function(fun = dnorm,args = list(mean=0, sd=1),lwd = 1,
                linetype="dashed", col = 'blue')
return(plot)}

mvn1 <- mvn.fun(10000, mu, sig)
mvn2 <- mvn.fun(100000, mu, sig)
mvn3 <- mvn.fun(1000000, mu, sig)
grid.arrange(mvn1, mvn2, mvn3, nrow = 1)
```

\newpage
Problem 3c.
Draw a sample size of 10,000 from the conditional distribution $p(X_1 \mid X_2, X_3)$, take $X_2 = X_3 = 1$. Compute the sample mean and sample SD and compare these numbers to the true values. Draw a histogram of the simulate values, along with the true conditional distribution.

```{r, fig.width=12, fig.height=6, message=FALSE, echo=FALSE}
set.seed(010303)  #Homework 01 | Problem 03 | Part 03
mu <- c(0, 0, 0)
sig <- matrix(c(1,4.5,9,4.5,25,49,9,49,100),nrow=3, ncol=3)
cmvn.fun <- function(d, mu, sig){
X<-rcmvnorm(d, mu, sig, dependent.ind = c(1), given.ind = c(2,3),
X.given = c(1,1) )
X<-matrix(X, nrow = d, ncol = 1)
X.df <- data.frame(X=X)
x.g<- matrix(c(1,1), ncol=1)
```

````
cmvn3 <- cmvn.fun(1000000, mu, sig)
grid.arrange(cmvn1, cmvn2, cmvn3, nrow = 1)
```

We want to find  expected value and sd of $X_1 | X_2 = X_3 = 1$.
Let $\pmb{X}_g = (X_2, X_3)$

 $$
 \begin{aligned}
 X_1 |\pmb{X}_g &\sim N( \mu_1 + \Sigma_{1g}\Sigma_{gg}^{-1}
(\pmb{x}_g -\pmb{\mu}_g), \Sigma_{11} - \Sigma_{22}^{-1}
\Sigma_{12}^T)
 \\ \quad \\
\mathbb{E} \left[ X_1 \mid \pmb{X}_g \right] & = \mu_1 +
\Sigma_{1g}\Sigma_{gg}^{-1}(\pmb{x}_g -\pmb{\mu}_g)
\\
& = 0 +
\begin{bmatrix} 4.5 & 9 \end{bmatrix}
\cdot \begin{bmatrix} 25 & 49 \\ 49 & 100 \end{bmatrix}^{-1}
\left(
\begin{bmatrix} 1 \\ 1 \end{bmatrix}
-
\begin{bmatrix} 0 \\ 0 \end{bmatrix}
\right)
\\
& = 0.1363636
\\
\quad \\
\text{s.d.} ( X_1 \mid \pmb{X}_g ) & = 0.4264014
 \end{aligned}
 $$



## Problem 3d.
Use the sample size of 10,000 from the joint trivariate
distribution to obtain he conditional mean and SD of $X_1$ given
$X_2 = X_3 = 1$, i.e. look at the triples whose second and third
components are within $\epsilon$ of 1.  How do these values
compare to the true mean and SD for various values of $\epsilon$?
```{r, echo=FALSE}
set.seed(010304)  #Homework 01 | Problem 03 | Part 04
d <-10000
mu <- c(0, 0, 0)
sig <- matrix(c(1,4.5,9,4.5,25,49,9,49,100),nrow=3, ncol=3)
````