Appendix:

The complete code (including embedded latex) can be found at the following address:
https://raw.githubusercontent.com/malooney/Decision_Theory/master/HW/HW2.Rmd

R Code used in analysis:

```
# Housekeeping -------------------------------------------------------------
rm(list= ls())
cat("\014")

set.seed(1)

library(summarytools)

# Problem 2.b --------------------------------------------------------------

mu1= runif(1, -5, 5)
mu2= runif(1, -5, 5)
sig2.1=runif(1, 0, 10)
sig2.2= runif(1, 0, 10)

n1= n2= 10000
X1= rnorm(n1, mu1, sqrt(sig2.1))
X2= rnorm(n2, mu2, sqrt(sig2.2))
Y= X1+ X2

mu= mu1+ mu2
sig.sq= sig2.1+ sig2.2

Y.grid= seq(-20, 20, by= 0.01)
true.density= 1/ sqrt(2* pi* sig.sq)* exp(-1/ (2* sig.sq)* (Y.grid- mu)^2)

Y.mean <- mean(Y)
Y.var <- var(Y)

# Problem 2.b Plots---------------------------------------------------------

par(mfrow= c(1, 2), ps= 10, cex= 0.55, cex.main= 1.5)

hist(Y, prob= TRUE, xlab= "Y", ylab= "p(Y)", xlim= c(-30, 30))

lines(density(Y), lwd= 2, col= "green")

lines(Y.grid, true.density, col= "red", lty= 1, lwd= 1)

legend(4, 0.1,  c("True Density of Y", "MC Simulation of Y"),
       lty= c(2, 1), col= c("red", "black"))


hist(Y, prob= TRUE, xlab= "Y", ylab= "p(Y)", xlim= c(-10, 5),
     ylim= c(0.06, 0.11), main= "Zoomed Histogram of Y")
```

```
lines(density(Y), lwd= 2, col= "green")

lines(Y.grid, true.density, col= "red", lty= 1, lwd= 1)

legend(-3, 0.15,  c("True Density of Y", "MC Simulation of Y"),
       lty= c(2, 1), col= c("red", "green"))

# Problem 2.c ------------------------------------------------------------------
# Initial paramater values taken from Problem 1.b

# MC convolution ---------------------------------------------------------------
N= length(Y.grid)
dY= numeric(N)
for(i in seq_len(N)){
  X2.x2 = rnorm(N, mu2, sqrt(sig2.2))
  dY[i] = (1/ N) *sum((dnorm(Y.grid[i]- X2.x2, mu1, sqrt(sig2.1))))
}

# Numerical Convolution using R "integrate" funtion (quadrature) --------------
f.X1 <- function(x1){
  dnorm(x1, mu1, sqrt(sig2.1))
}

f.X2 <- function(x2){
  dnorm(x2, mu2, sqrt(sig2.2))
}

f.Y <- function(Y.grid){
  integrate(function(x1, Y.grid)
    f.X2(Y.grid- x1)* f.X1(x1), -Inf, Inf, Y.grid)$value
}

f.Y <- Vectorize(f.Y)

# Y.grid is assumed to be the MC integration support region
convolution.Y <- f.Y(Y.grid)

# Problem 2.c Plots ------------------------------------------------------------

par(mfrow= c(1, 2), ps= 10, cex= 0.55, cex.main= 1.5)

hist(Y, prob= TRUE, xlab= "Y", ylab= "p(Y)", xlim=c(-30, 30))

lines(Y.grid, true.density, col= "red", lty= 2, lwd= 1)

legend(4, 0.1,  c("True Density of Y"), lty= c(2), col= c("red"))

hist(Y, prob= TRUE, xlab= "Y", ylab= "p(Y)", xlim= c(-30, 30),
     main= "Convolution Method - Histogram of Y")

lines(Y.grid, true.density, col = "red", lty= 2, lwd= 2)

lines(Y.grid, convolution.Y, col = "green", lty= 1)
```

```
legend(4, 0.1,  c("True Density of Y", "Convolution of Y"),
       lty=c(2, 1), col=c("red","green"))

# Problem 2.d ------------------------------------------------------------------

a1 = runif(1, 0, 5)
b1 = runif(1, 0, 5)
a2 = runif(1, 0, 5)
b2 = runif(1, 0, 5)

n1 = n2 = 10000
X1 = rgamma(n1, a1, b1)
X2 = rgamma(n2, a2, b2)
Y = X1 + X2

Y.grid= seq(0, max(Y), by= 0.01)

# MC convolution ---------------------------------------------------------------
N= length(Y.grid)
dY= numeric(N)

for(i in seq_len(N)){
    X2.x2= rgamma(N, a2, b2)
    dY[i]= (1/N)* sum(dgamma(Y.grid[i]- X2.x2, a1, b1))
}

# Numerical Convolution using R "integrate" funtion (quadrature) ---------------
f.X1 <- function(x1){
  dgamma(x1, a1, b1)
}

f.X2 <- function(x2){
  dgamma(x2, a2, b2)
}

f.Y <- function(Y.grid){ integrate(function(x1, Y.grid)
  f.X2(Y.grid- log(x1))* f.X1(x1), 0, Inf, Y.grid)$value
}

f.Y <- Vectorize(f.Y)

# Y.grid is assumed to be the MC integration support region
convolution.Y <- f.Y(exp(Y.grid))

Y.mean <- mean(Y)
#Y.sig2 <- var(Y)

dY.mean <- Y.grid[which.max(dY)]
#dY.var <- var(dY)

# Problem 2.d Plots ------------------------------------------------------------

par(mfrow= c(1, 2), ps= 10, cex= 0.55, cex.main= 1.5)
```

```
hist(Y, prob = TRUE, xlab = "Y", ylab = "p(Y)", ylim= c(0, 0.03))

lines(density(Y), lwd = 1, col="red", lty= 2)

hist(Y, prob = TRUE, xlab = "Y", ylab = "p(Y)", ylim= c(0, 0.03))

lines(density(Y), lwd = 1, col="red", lty= 2)

lines(Y.grid, dY, col = rgb(0, 1, 0.8, alpha=0.4), lty = 1)

# Problem 3.c ------------------------------------------------------------------
# game of Craps

Craps <- function(){
  prob <- c(1:6, 5:1)/ 36
  first.roll <- sample(2:12, 1, prob= prob)
  if(first.roll %in% c(7, 11) || first.roll %in% c(2, 3, 12)){
    return(1* (first.roll %in% c(7, 11)))
  } else{
    point= first.roll
    new.roll <- sample(2:12, 1, prob= prob)
    while(point != new.roll && new.roll != 7){
      new.roll <- sample(2:12, 1, prob= prob)
    }
    return(1* (point == new.roll))
  }
}

N = 10000
n = 10
initialBet.Matrix <- matrix(10, ncol= 1, nrow= N)

  play <- function(bet= initialBet.Matrix){
    iBet= bet
  revenue= 0
  for(i in seq_len(n)){
    if(Craps() == 0){
      revenue= revenue- bet
      bet= 2* bet
    } else{
      revenue= revenue+ bet
      bet= iBet
    }
  }
  return( matrix(revenue, nrow= 1, ncol= 1))
  }

#start_time <- Sys.time()
earnings <-   matrix(apply(initialBet.Matrix, 1, play), nrow=N, ncol=1)
#end_time <- Sys.time()
#end_time - start_time

earn.mean <- mean(earnings)
```

```
earn.max <-  max(earnings)
earn.min <-  min(earnings)

# Problem 3.c Plots ------------------------------------------------------------

par(mfrow= c(1, 2), ps= 10, cex= 0.55, cex.main= 1.5)

hist(earnings, main= "Histogram of Earnings", xlab= "Earnings",
xlim= c(earn.min+100, earn.max+10000))

plot(earnings[1:10000], type="l", main= "plot of 10,000 Experiments",
     ylab= "Earnings", ylim= c(earn.min+100, earn.max+10000))

# Problem 3.c freq table -------------------------------------------------------

freq(as.vector(earnings), display.nas= FALSE, style= "rmarkdown", method = "pander",
     omit.headings =T)
```