Praphull Kumar
UID: 204 271 732

STATS 202A
Fall 2014
Homework 1

<u>Output:</u>

Question 1:

a).

```
func90 = function (v) {
  v_sorted = sort(v)
  v_90 = v_sorted[90]
  return (v_90)
}
```

b).

```
func91 = function (v) {
  v_sorted = sort(v)
  v_91 = v_sorted[91]
  return (v_91)
}
```

c).

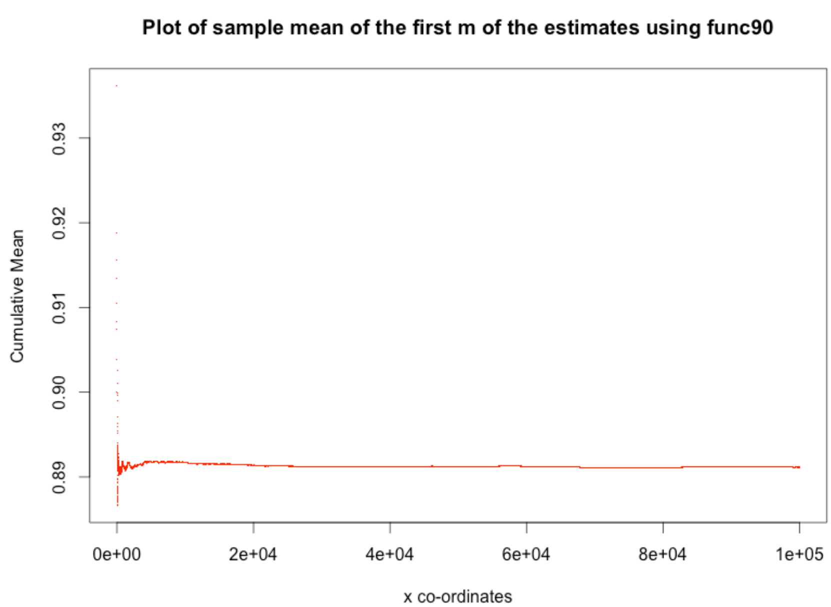
```
funcMeanOf90_91 = function (v) {
  v_sorted = sort(v)
  v_meanOf_90_91 = (v_sorted[90] + v_sorted[91])/2
  return (v_meanOf_90_91)
}
```

d).

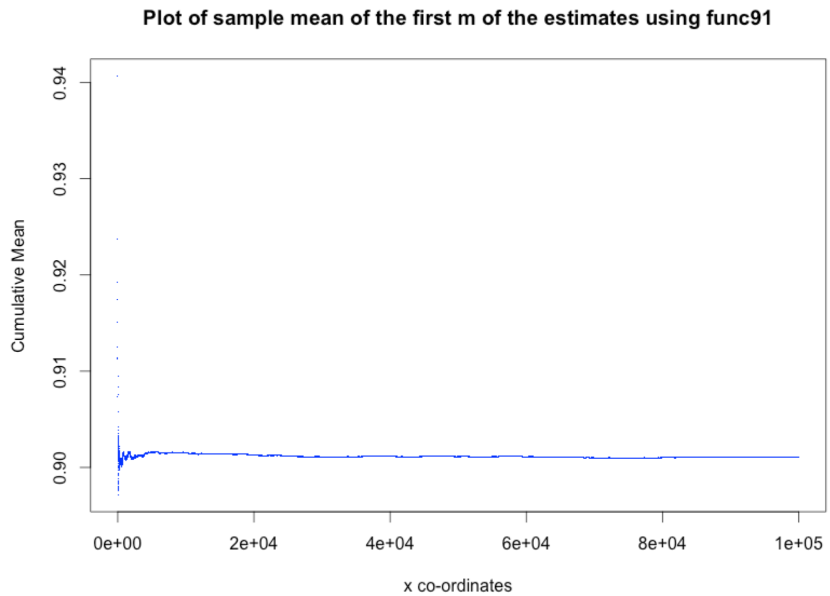
i).

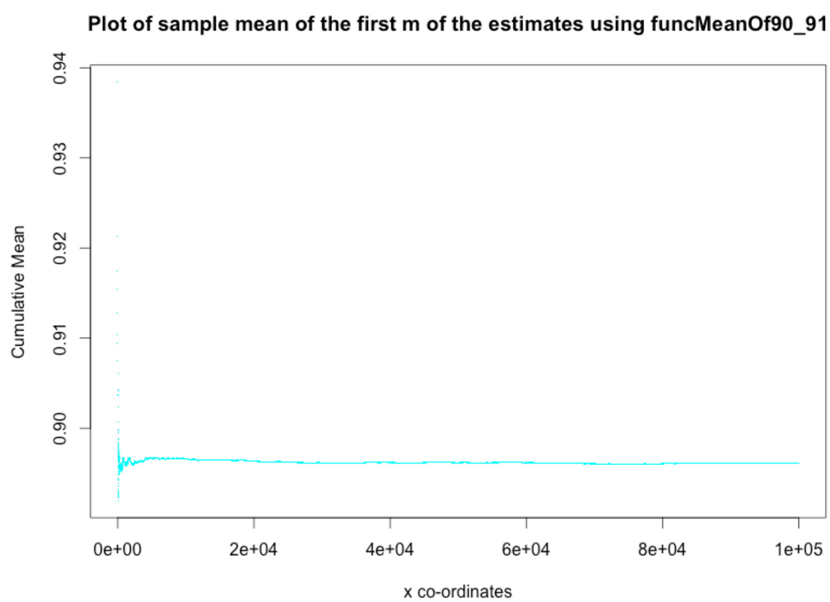
90th percentile using func90 = 0.8391208
91st percentile using func91 = 0.8429538
Mean of 90th and 91st percentile using funcMeanOf90_91 = 0.8410373
90th percentile using quantile function = 0.8395041

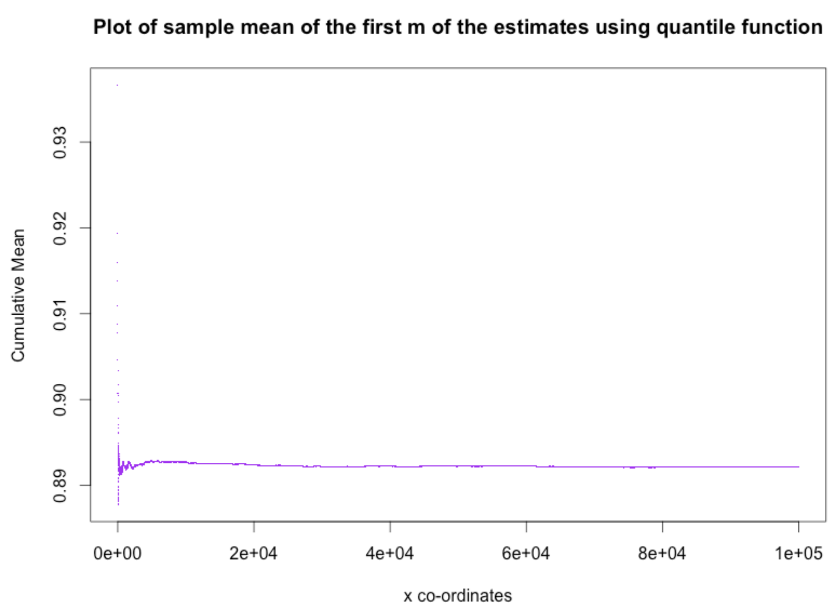
iii). Plots for all four functions are as follows:



Praphull Kumar
UID: 204 271 732





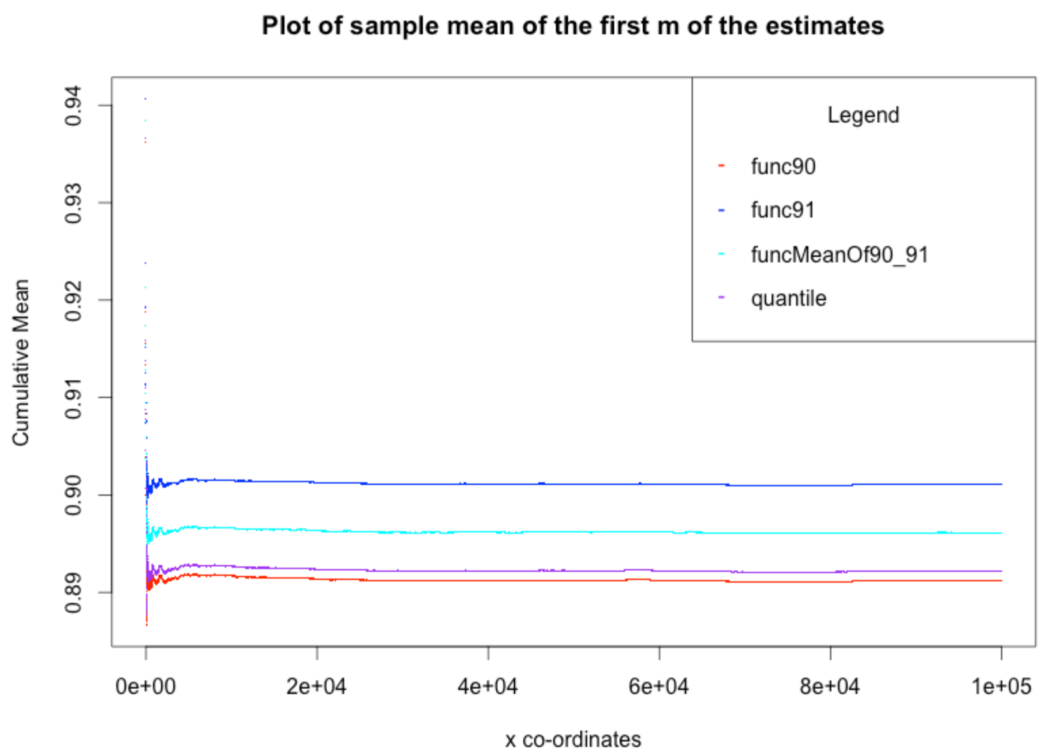


Praphull Kumar
UID: 204 271 732

e).

Mean of 100000 estimates using func90 = 0.8911428
Mean of 100000 estimates using func91 = 0.9010436
Mean of 100000 estimates using funcMeanOf90_91 = 0.8960932
Mean of 100000 estimates using quantile = 0.8921329

Based upon above output, we can infer that the best way to approximate the 90th percentile is to take the 91st element since its value as printed above came out to be 0.9010436 which is closest to 0.9



Question 2:

a).

```
pi2 = function(n) {
  return(sqrt(6*sum(1/(c(1:n)^2))))
}
```

Value of pi2(10^0) = 2.44949
Value of pi2(10^1) = 3.049362
Value of pi2(10^2) = 3.132077
Value of pi2(10^3) = 3.140638
Value of pi2(10^4) = 3.141497
Value of pi2(10^5) = 3.141583
Value of pi2(10^6) = 3.141592

b).

```
pi3 = function(n) {
  x = runif(n, -1, 1)
  y = runif(n, -1, 1)
  d = (x)^2 + (y)^2
  pin = which(d <= 1)
  pout = which(d > 1)
```

Praphull Kumar
UID: 204 271 732

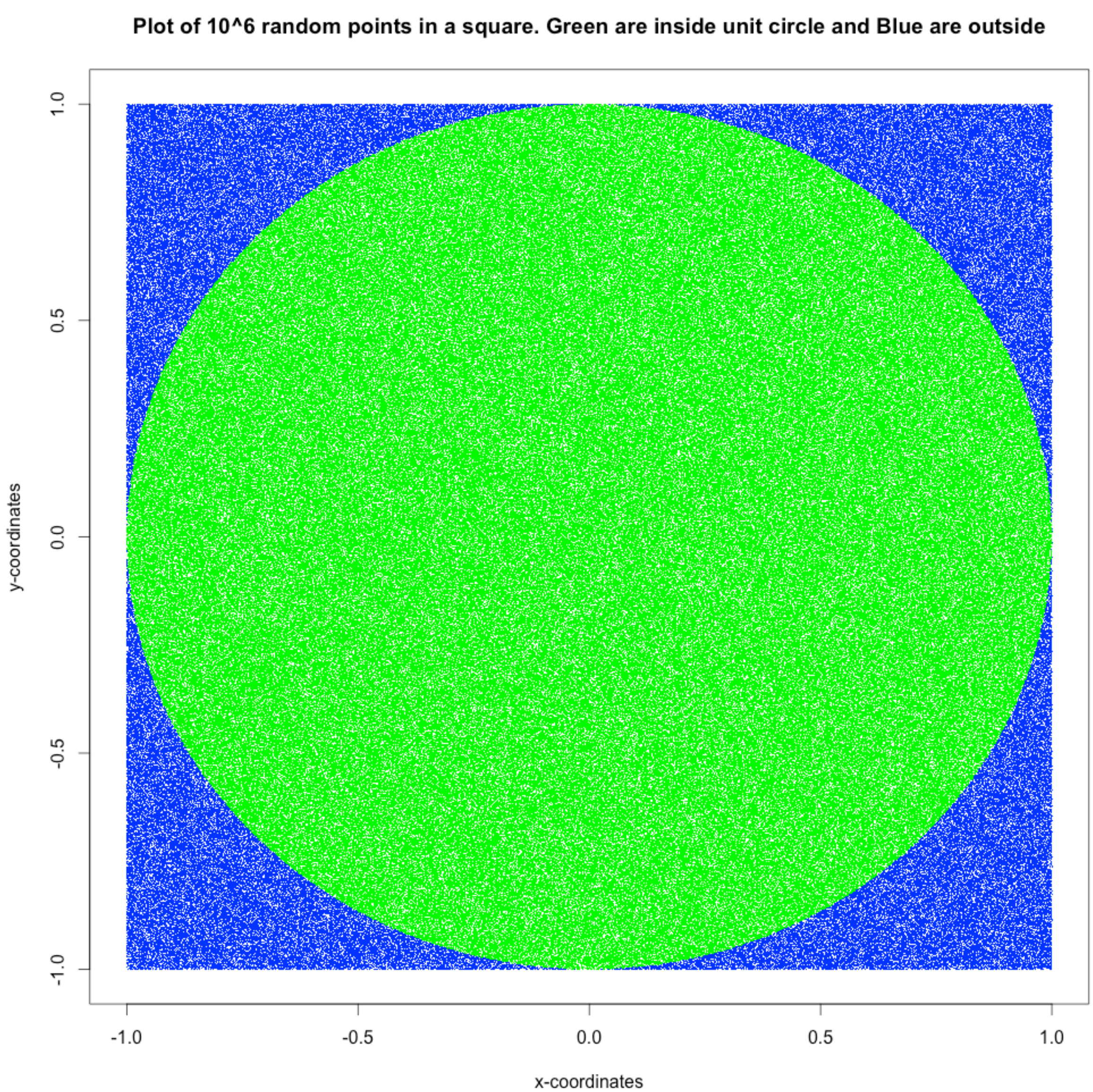
```
  pi3_v = length(pin) * 4 / n
  ret = list(pi3_v, x, y, pin, pout)
  return(ret)
}
```

Value of pi3(10^0) = 4
Value of pi3(10^1) = 3.6
Value of pi3(10^2) = 3.36
Value of pi3(10^3) = 3.048
Value of pi3(10^4) = 3.1356
Value of pi3(10^5) = 3.13216
Value of pi3(10^6) = 3.141264

In this question, I have used the Monte Carlo Approximation of Pi which is based on the formula that

$$\text{Pi}/4 \sim \frac{\text{Number of points inside unit circle}}{\text{Total number of points}}$$

So first I generated n random numbers and then calculated the distance of each of these points with respect to origin. If this distance is less than one, than that point is inside the circle else it's outside.



Praphull Kumar
UID: 204 271 732

Code:

```
## Question 1.a
func90 = function (v) {
  v_sorted = sort(v)
  v_90 = v_sorted[90]
  return (v_90)
}

## Question 1.b
func91 = function (v) {
  v_sorted = sort(v)
  v_91 = v_sorted[91]
  return (v_91)
}

## Question 1.c
funcMeanOf90_91 = function (v) {
  v_sorted = sort(v)
  v_meanOf_90_91 = (v_sorted[90] + v_sorted[91])/2
  return (v_meanOf_90_91)
}

## Question 1.d (i)
ud_100 = runif(100)
v_90 = func90(ud_100)
v_91 = func91(ud_100)
v_mean_90_91 = funcMeanOf90_91(ud_100)
quantile_90 = quantile(ud_100, 0.9)

cat("90th percentile using func90 =", v_90)
cat("91st percentile using func91 =", v_91)
cat("Mean of 90th and 91st percentile using funcMeanOf90_91 =", v_mean_90_91)
cat("90th percentile using quantile function =", quantile_90)

## Question 1.d (ii)
l_90 = c()
l_91 = c()
l_mean_90_91 = c()
l_quantile_90 = c()
for (i in 1:100000) {
  ud_100_t = runif(100)
  l_90[i] = func90(ud_100_t)
  l_91[i] = func91(ud_100_t)
  l_mean_90_91[i] = funcMeanOf90_91(ud_100_t)
  l_quantile_90[i] = quantile(ud_100_t, 0.9)
}

## Question 1.d (iii)
mean_l_90 = cumsum(l_90) / 1:100000
mean_l_91 = cumsum(l_91) / 1:100000
mean_l_mean_90_91 = cumsum(l_mean_90_91) / 1:100000
mean_l_quantile_90 = cumsum(l_quantile_90) / 1:100000

plot(c(0,100000), c(min(mean_l_90),max(mean_l_90)),type="n", xlab="x co-ordinates",
ylab="Cumulative Mean", main="Plot of sample mean of the first m of the estimates
using func90")
points(mean_l_90,pch=".", col="red")

plot(c(0,100000), c(min(mean_l_91),max(mean_l_91)),type="n", xlab="x co-ordinates",
ylab="Cumulative Mean", main="Plot of sample mean of the first m of the estimates
using func91")
points(mean_l_91,pch=".", col="blue")
```

```
plot(c(0,100000), c(min(mean_l_mean_90_91),max(mean_l_mean_90_91)),type="n", xlab="x co-ordinates", ylab="Cumulative Mean", main="Plot of sample mean of the first m of the estimates using funcMeanOf90_91")
points(mean_l_mean_90_91,pch=".", col="cyan")

plot(c(0,100000), c(min(mean_l_quantile_90),max(mean_l_quantile_90)),type="n", xlab="x co-ordinates", ylab="Cumulative Mean", main="Plot of sample mean of the first m of the estimates using quantile function")
points(mean_l_quantile_90,pch=".", col="purple")

## Question 1.e
cat("Mean of 100000 estimates using func90 =", mean_l_90[100000])
cat("Mean of 100000 estimates using func91 =", mean_l_91[100000])
cat("Mean of 100000 estimates using funcMeanOf90_91 =", mean_l_mean_90_91[100000])
cat("Mean of 100000 estimates using quantile =", mean_l_quantile_90[100000])
y_min = min(c(min(mean_l_90), min(mean_l_91), min(mean_l_mean_90_91), min(mean_l_quantile_90)))
y_max = max(c(min(mean_l_90), max(mean_l_91), max(mean_l_mean_90_91), max(mean_l_quantile_90)))

plot(c(0,100000), c(y_min,y_max),type="n", xlab="x co-ordinates", ylab="Cumulative Mean", main="Plot of sample mean of the first m of the estimates")
points(mean_l_90,pch=".", col="red")
points(mean_l_91,pch=".", col="blue")
points(mean_l_mean_90_91,pch=".", col="cyan")
points(mean_l_quantile_90,pch=".", col="purple")
legend("topright", c("func90","func91", "funcMeanOf90_91", "quantile"), pch = '-', title = "Legend", col=c("red", "blue", "cyan", "purple"))


## Question 2.a
pi2 = function(n) {
  return(sqrt(6*sum(1/(c(1:n)^2))))
}

cat ("Value of pi2(10^0) =", pi2(10^0))
cat ("Value of pi2(10^1) =", pi2(10^1))
cat ("Value of pi2(10^2) =", pi2(10^2))
cat ("Value of pi2(10^3) =", pi2(10^3))
cat ("Value of pi2(10^4) =", pi2(10^4))
cat ("Value of pi2(10^5) =", pi2(10^5))
cat ("Value of pi2(10^6) =", pi2(10^6))

## Question 2.b
pi3 = function(n) {
  x = runif(n, -1, 1)
  y = runif(n, -1, 1)
  d = (x)^2 + (y)^2
  pin = which(d <= 1)
  pout = which(d > 1)
  pi3_v = length(pin) * 4 / n
  ret = list(pi3_v, x, y, pin, pout)
  return(ret)
}

cat("Value of pi3(10^0) =", (pi3(10^0))[[1]])
cat("Value of pi3(10^1) =", (pi3(10^1))[[1]])
cat("Value of pi3(10^2) =", (pi3(10^2))[[1]])
cat("Value of pi3(10^3) =", (pi3(10^3))[[1]])
cat("Value of pi3(10^4) =", (pi3(10^4))[[1]])
cat("Value of pi3(10^5) =", (pi3(10^5))[[1]])
res_6 = pi3(10^6)
cat("Value of pi3(10^6) =", res_6[[1]])
```

```
plot(c(-1,1),c(-1,1),type="n",xlab="x-coordinates",ylab="y-coordinates", main="Plot
of 10^6 random points in a square. Green are inside unit circle and Blue are
outside");
points((res_6[[2]])[(res_6[[4]])],(res_6[[3]])[(res_6[[4]])],pch='.',col="green") ;
points((res_6[[2]])[(res_6[[5]])],(res_6[[3]])[(res_6[[5]])],pch='.',col="blue");
```