Computational Statistics and Data Analysis
Sheet No. 4

David Bubeck, Patrick Nisblè

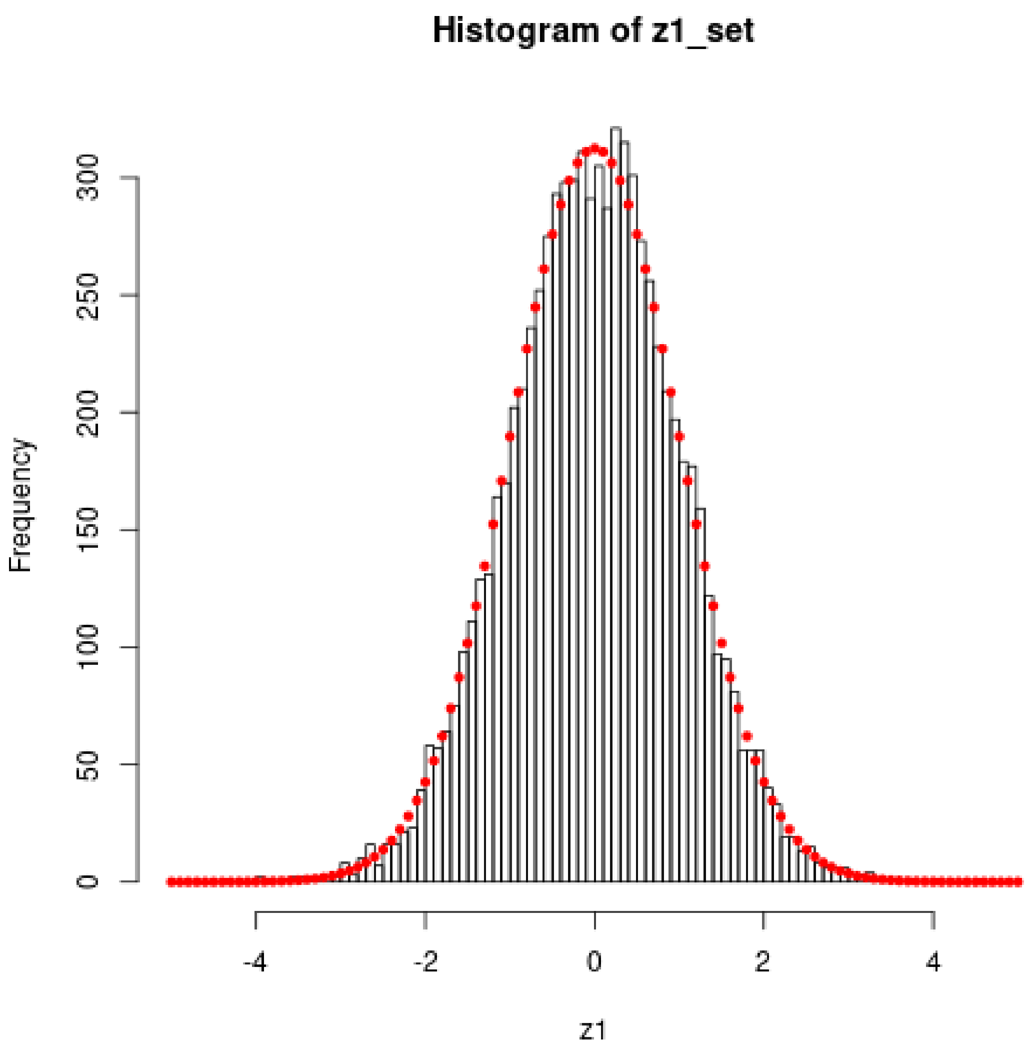
May 17, 2017

1 Generating Gaussian distributed random variables

We are to create uniformly distributed points in a unit circle and show the values for z_1 and z_2 are gaussian distributed

```r
num_randoms = 10000

orig_set = runif(num_randoms*2, min = -1, max = 1)
dim(orig_set) = c(2,num_randoms)

calc_set = orig_set[1,]**2 + orig_set[2,]**2
used_set = orig_set[,calc_set <= 1]

z1 <- function(vec) {
    return(vec[1]*sqrt(-2*(log(vec[1]**2+vec[2]**2))/(vec[1]**2+vec[2]**2)))
}

z2 <- function(vec) {
    return(vec[2]*sqrt(-2*(log(vec[1]**2+vec[2]**2))/(vec[1]**2+vec[2]**2)))
}

z1_set = 1:length(used_set[1,])
z2_set = 1:length(used_set[1,])
for (i in 1:length(used_set[1,])) {
    z1_set[i] = z1(used_set[,i])
    z2_set[i] = z2(used_set[,i])
}

pos_set = seq(-5,5,.1)

png('ex11.png')
h1 = hist(z1_set, breaks=pos_set, xlab='z1')
yfit1 = dnorm(pos_set, mean=0, sd=sd(z1_set))
```

```
yfit1 = yfit1 * diff(h1$mids[1:2])*length(z1_set)
points(pos_set, yfit1, cex=1, pch=20, col='red')


png('ex12.png')
h2 = hist(z2_set, breaks=pos_set, xlab='z2')
yfit2 = dnorm(pos_set, mean=0, sd=sd(z2_set))
yfit2 = yfit2 * diff(h2$mids[1:2])*length(z2_set)
points(pos_set, yfit2, cex=1, pch=20, col='red')
```



66

Figure 1: histogram of z1 and gauss-fit

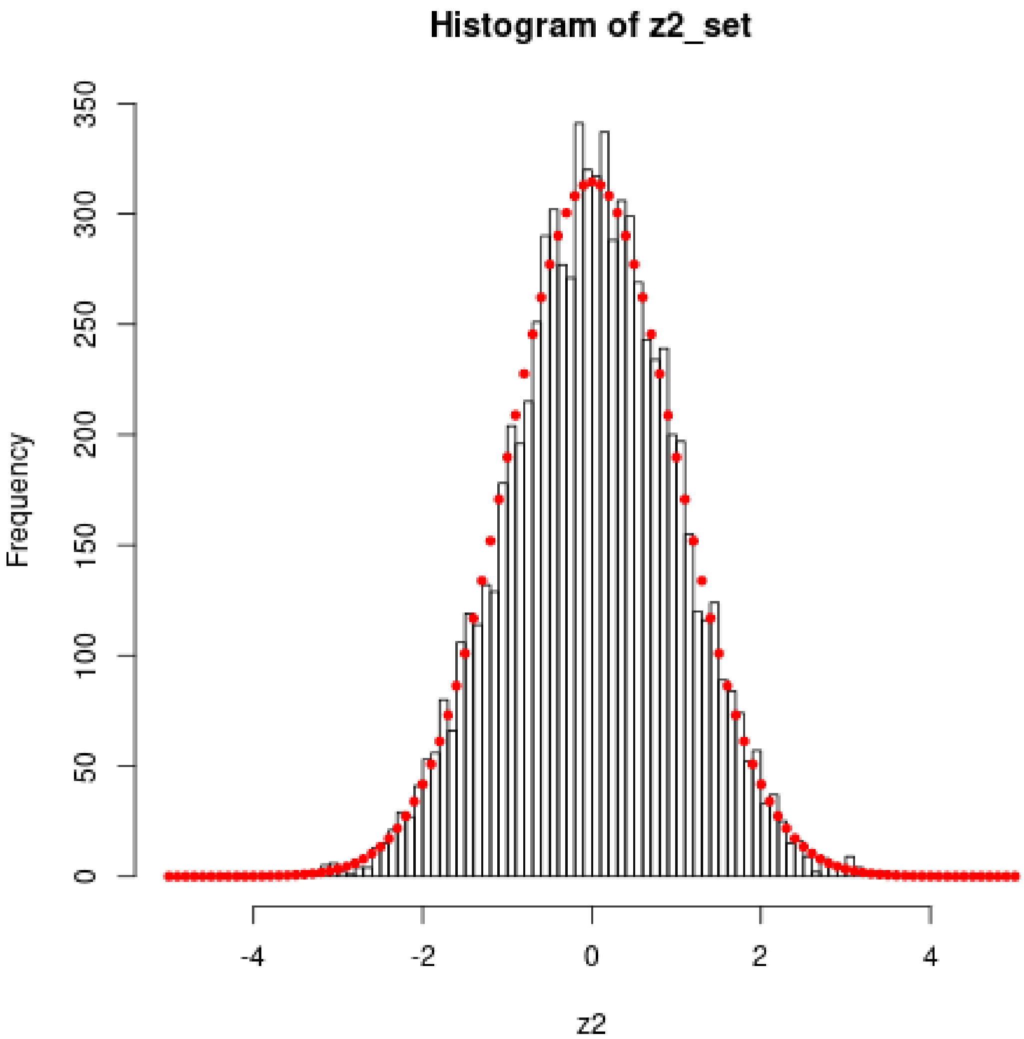


Figure 2: histogram of z2 and gauss-fit

2 Random Walk

We generate the mean number of steps needed for a 1D random walk as a function of the distance n

```r
rand_walk <- function(nmax) {
    pos = 0
    steps = 0
    while (pos < nmax && pos > -nmax){
        step = sample(0:1,1) *2 -1
        pos = pos + step
        steps = steps +1
    }

    return(steps)
}

dist_rand_walk <- function(n){

    num_tests = 100
    res = 1:num_tests
    for (i in 1:num_tests) {
        res[i] = rand_walk(n)
    }

    return(res)
}

nset = 1:50
means = 1:max(nset)
for (i in nset){
    means[i] = mean(dist_rand_walk(i))
    print(i)
}

png('ex2.png')
plot(
    nset, means, pch=20, cex=1,
    xlab='n: width of exprimental area', ylab='average number of steps needed'
)
```

Figure 3: ex2.R

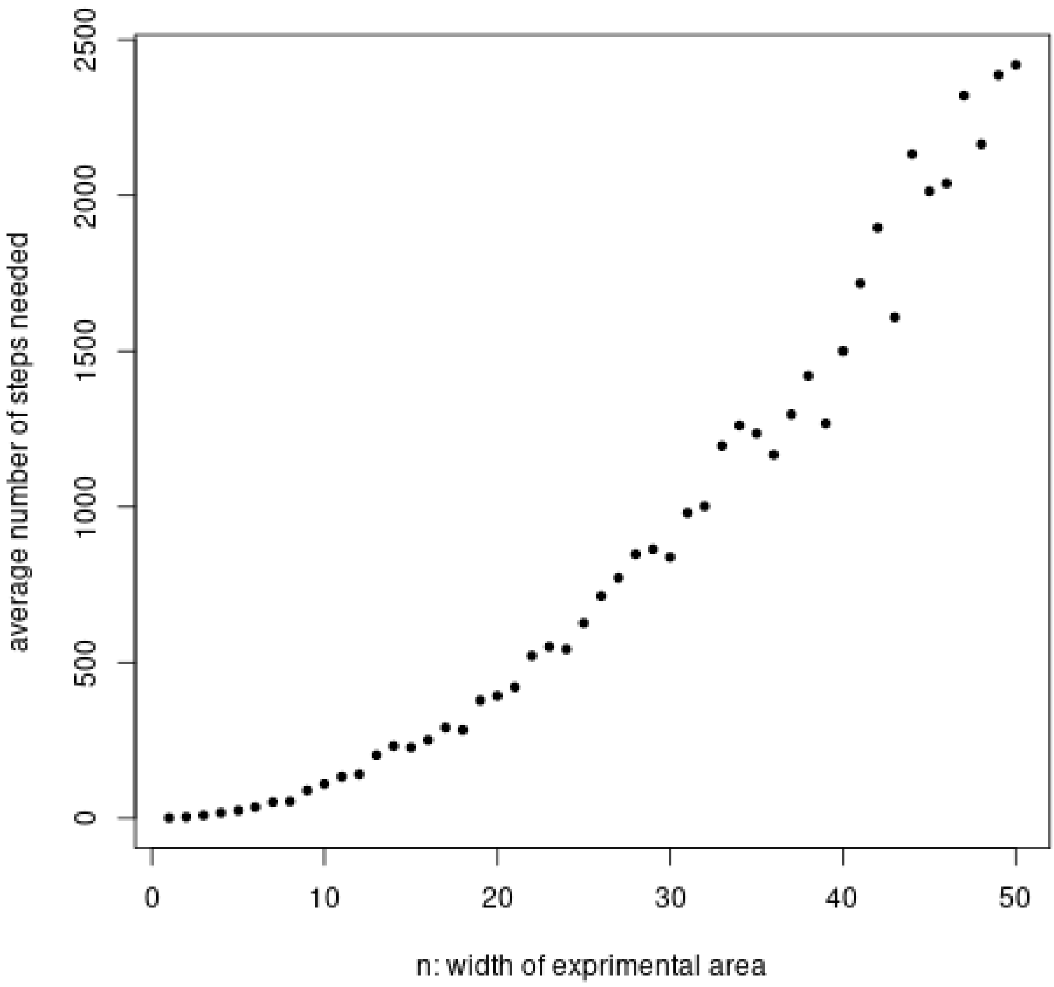


Figure 4: average number of steps taken as a function of n, in 50 tests per n

we can assume the number of steps required rises exponentially with the distance n