

R-code.R

Prasanth Sambaraju

2022-05-06

```
trials=10000
# number of simulations
avg=80
# mean of distribution
stdev=c(1:10)
# range of standard deviation values
cnt=0
m=0
ct=0
cnt1=0
m1=0
n=0
Q=75
# specify Q value
Q_1=(Q+5)
Q_2=(Q-15)
Q_3=(Q-25)
d<-c()
# create empty vector
fn<-c()
count=j=0
for (x in min(stdev):max(stdev)){
  d<-c()
  set.seed(123)
  # to reproduce particular sequence of random numbers
  for (j1 in 1:trials){
    k<-round(c(rnorm(24,avg,x)),digits = 2)
    # simulate data using mean and standard deviation and
    # round it to 2 digits
    d<-cbind(d,t(t(k)))
  }
  write.table(rbind(paste0("Trial# ",c(1:trials)),d),
              paste0("stdev_", x, ".csv"),
              sep = ",",row.names=F,col.names=F)
  # write a .csv file with name stdev_ with x value in the end
  d<-list(d)
  names(d)=paste("d",x,sep = "_")
  # create d_ with x value in the end
  list2env(d,envir = .GlobalEnv)
}
gh<-data.frame()
for (x in min(stdev):max(stdev)){
```

```

gh_s1<-replicate(trials,0)
# create a column vector with number of rows equal to the variable trails
# and fill it with zero's
gh_s2<-replicate(trials,0)
gh_s3<-replicate(trials,0)
Cum_prob1<-replicate(trials,0)
Cum_prob<-replicate(trials,0)
for (j in 1:trials){
  count=0
  d<-get(paste0("d_",x,sep=""))
  # if j=1, x=1 then get d_1
  for (i in 1:6){
    if (d[i,j]>Q_1){
      # checking first six values in d greater than (Q+5)
      count=count+1
    }
  }
  if (count==6){
    gh_s1[j]=1
    # checking S1 criteria
  }
  if(gh_s1[j]==0){
    # s1 criteria fail
    (cnt=m=0)
    for (i in 1:12){
      if (d[i,j]>Q_2){
        # checking if first 12 values in d are greater than (Q-15)
        cnt=cnt+1
      }
      if (mean(d[1:12,j])>=Q){
        # checking if average of first 12 in d are greater than or equal
        # to Q
        m=1
        cnt=cnt*m
        if (cnt==12){
          gh_s2[j]=1
          # checking for S2 criteria
        }
      }
    }
  }
}
if (gh_s2[j]==0 && gh_s1[j]==0) {
  # if failing in S1 and S2 stages
  ct=cnt1=n=m1=0
  for (b in 1:24){
    if (d[b,j]>Q_3){
      # checking if 24 in d are greater than (Q-25)
      ct=ct+1
    }
    if (d[b,j]>Q_2){

```

```

        cnt1=cnt1+1
    }
}
if (mean(d[1:24,j])>=Q){
    m1=1
}
if (cnt1>=22){
    # check atleast 22 are greater than (Q-15)
    n=1
    # checking if average of 24 are greater than Q
}
ct=ct*m1*n
if (ct==24){
    gh_s3[j]=1
    # checking S3 criteria
}
if(gh_s1[j]==0 && gh_s2[j]==0 && gh_s3[j]==0){
    gh<-rbind(gh,c(x,j))
    # lists all trials numbers which fail to meet acceptance criteria
}
}
Cum_prob1[j] = (sum(gh_s1[j],gh_s2[j],gh_s3[j]))
# to calculate cumulative probability as function of trial number
if (j>1){Cum_prob[j]=sum(Cum_prob1)/j}else{Cum_prob[j]=Cum_prob1[j]}
}
pdf(file=paste0("stdev_",x,".pdf"))
plot(Cum_prob,ylim = c(0,1),xlab = "Trials", ylab = "Cumulative
Probability",
     main = paste0("Std dev ", x))
dev.off()
write.table(cbind(c(1:trials),gh_s1,gh_s2,gh_s3,Cum_prob),
            paste0("All stages stdev_",x,".csv"),
            sep =
              ",",row.names=F,col.names=c('Trial#','S1','S2','S3','Cum Prob
'))
# write .csv file
options(scipen = 999)
# disable scientific notation
fn<-rbind(fn,c(avg, x, sum(gh_s1)/trials, sum(gh_s2)/trials,
              sum(gh_s3)/trials, sum(gh_s1,gh_s2,gh_s3)/trials))
# calculate probability of meeting acceptance criteria at every stage and
# final passing probability
}
hder<-c("Avg","Stddev","S1","S2","S3","Final")
result<-as.data.frame(fn)
names(result)<-hder
print(result)

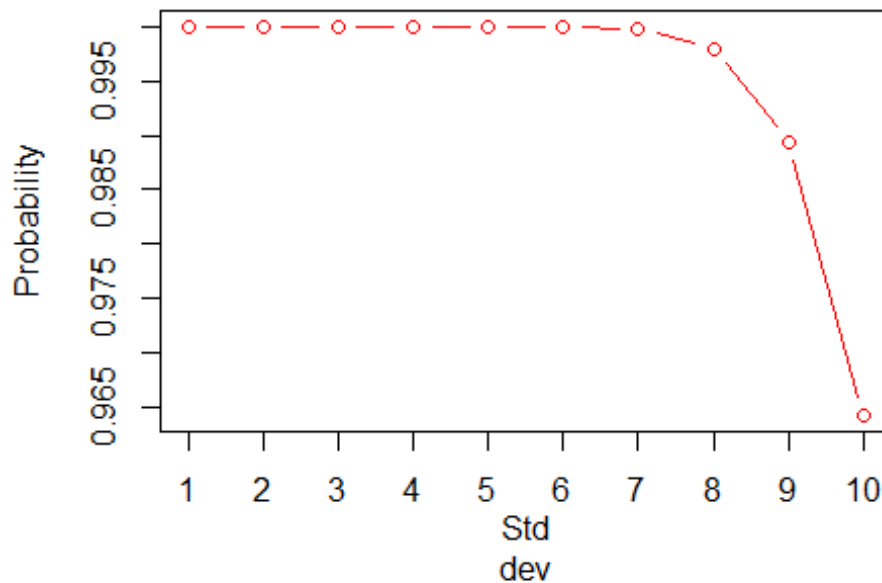
##      Avg Stddev      S1      S2      S3 Final
## 1      80      1 0.0153 0.9847 0.0000 1.0000

```

```
## 2 80 2 0.0154 0.9846 0.0000 1.0000
## 3 80 3 0.0154 0.9846 0.0000 1.0000
## 4 80 4 0.0154 0.9846 0.0000 1.0000
## 5 80 5 0.0154 0.9838 0.0008 1.0000
## 6 80 6 0.0154 0.9778 0.0068 1.0000
## 7 80 7 0.0154 0.9524 0.0320 0.9998
## 8 80 8 0.0154 0.9010 0.0816 0.9980
## 9 80 9 0.0154 0.8289 0.1451 0.9894
## 10 80 10 0.0154 0.7300 0.2188 0.9642
```

```
names(gh)<-c("Stddev", "Trial#")
plot(result$Stddev, result$Final, type="b", cex=1, col="red", pch=1, xlab = "Std
dev", ylab="Probability",
      main = paste("Dissolution passing probability when mean is",
                    avg), xaxt="none")
axis(1, seq(1,10,1))
```

Dissolution passing probability when mean is 80



```
write.csv(result, paste0("Final.csv"))
write.csv(gh, paste0("Fail trails", ".csv"))
```