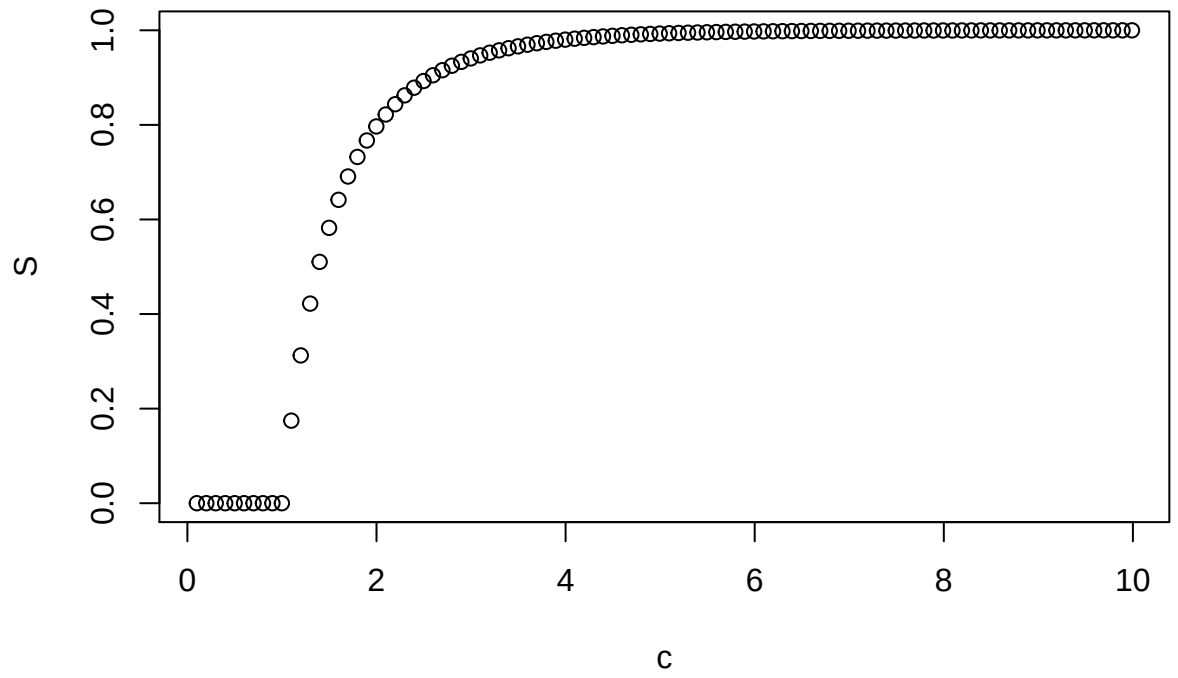
Homework 5

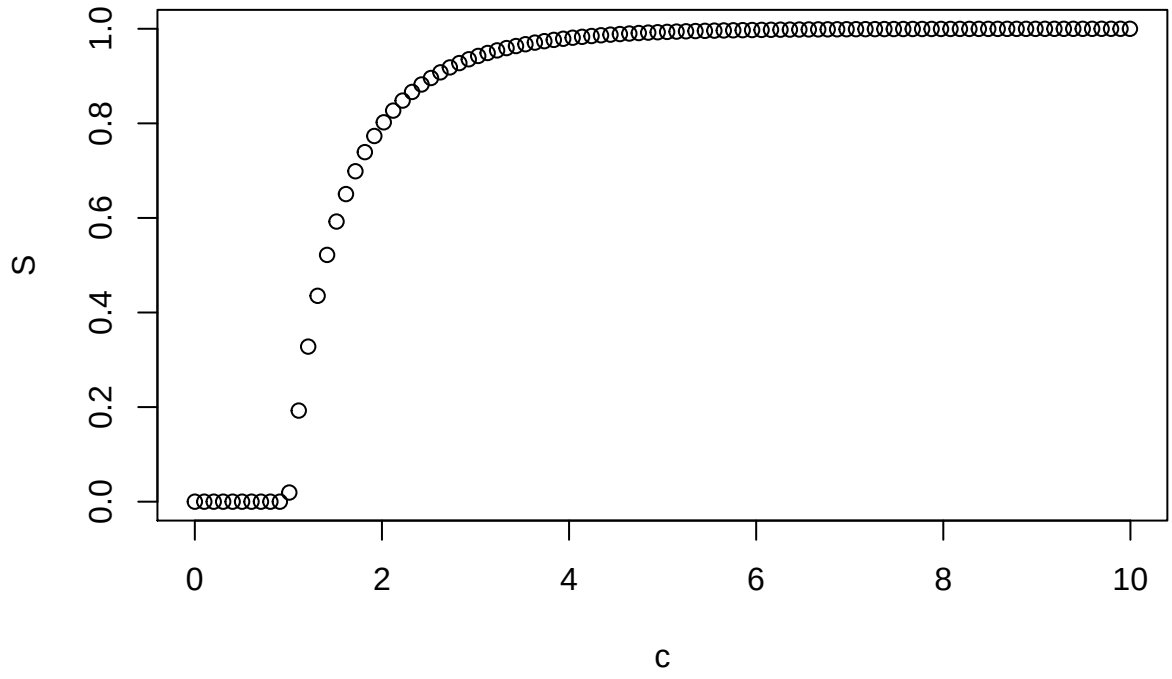
Bright

2/21/2021

```r
n=1000 #number of nodes
p=seq(from = 0.0001, to = 0.01, length.out = 100)
u_theoretical=rep(0,100) # theoretical solution of u
library("nleqslv")
for (j in 1:100){
  fn <- function(u) {
    u-(1-p[j]+p[j]*u)^(n-1)
  }
  sol=nleqslv(c(0,1), fn) #extract the $x, that is the solution
  u_theoretical[j]=sol$x[1]#the solutions are ordered increasingly, and 1 is always a solution, so take
}
S=1-u_theoretical
c=p*(n-1)
plot(c,S,ylim=c(0,1))
```



```r
n=2000 #number of nodes
p=seq(from = 0.0000001, to = 0.005, length.out = 100)
u_theoretical=rep(0,100) # theoretical solution of u
library("nleqslv")
for (j in 1:100){
  fn <- function(u) {
    u-(1-p[j]+p[j]*u)^(n-1)
  }
  sol=nleqslv(c(0,1), fn) #extract the $x, that is the solution
  u_theoretical[j]=sol$x[1]#the solutions are ordered increasingly, and 1 is always a solution, so take
}
S=1-u_theoretical
c=p*(n-1)
plot(c,S,ylim=c(0,1))
```




```r
library(igraph)
```

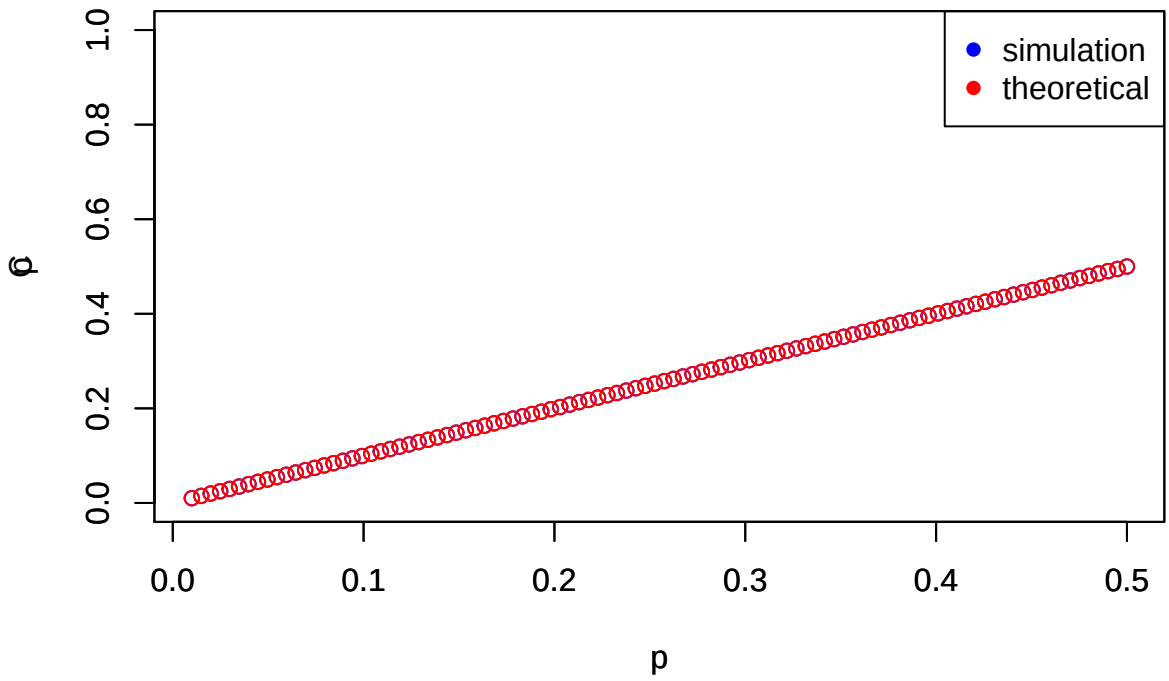
```
##
## Attaching package: 'igraph'
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
## The following object is masked from 'package:base':
##
##     union
```

```r
n=500
p=seq(from = 0.01, to = 0.5, length.out = 100) #Generate 100 different p in a reasonable range
C=rep(0,100) #save the result in the vector u
simulation_number=50
for (j in 1:100){
  simulation_result=rep(0,simulation_number)
  for (i in 1:simulation_number){
    g <- erdos.renyi.game(n, p[j], type = "gnp")
 simulation_result[i]=transitivity(g,type = "undirected", isolates = "zero")
  }
  C[j]=mean(simulation_result)
}

plot(p,C,xlim=c(min(p),max(p)),ylim=c(0,1),col="blue")
par(new=TRUE)



plot(p,p,xlim=c(min(p),max(p)),ylim=c(0,1),col="red")
legend("topright",legend=c("simulation","theoretical"),col = c("blue","red"),pch=16)
```




```r
library(igraph)
n=500
p=seq(from = 0.01, to = 0.5, length.out = 100) #Generate 100 different p in a reasonable range
lvals=rep(0,100) #save the result in the vector u
l=rep(0,100)
```

```r
simulation_number=50
for (j in 1:100){
  simulation_result=rep(0,simulation_number)
  for (i in 1:simulation_number){
    g <- erdos.renyi.game(n, p[j], type = "gnp")
 simulation_result[i]=mean_distance(g, directed = F, unconnected = T)
  }
  lvals[j]=mean(simulation_result)
l[j]=log(n)/log((n-1)*p[j])
}


plot(p,lvals,xlim=c(min(p),max(p)),ylim=c(0,5),col="blue")
par(new=TRUE)

plot(p,l,xlim=c(min(p),max(p)),ylim=c(0,5),col="red")
legend("topright",legend=c("simulation","theoretical"),col = c("blue","red"),pch=16)
```