

Random Network Models - Chapter 10

Nick Lauerma

Started 12-Mar-2020

Contents

Libraries and Data	1
Libraries used	1
data sets used	1
Models of Network Structures and Formation	2
Erdos-Renyi Random Graph Model	2
Small-World Model	6
Scale-Free Models	10
Comparing Random Models to Empirical models	16

Libraries and Data

Libraries used

```
library(igraph)

##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union

library(lattice)
library(UserNetR)
#library(ergmharris)
```

data sets used

```
data("lhds")
```

Models of Network Structures and Formation

Erdos-Renyi Random Graph Model

```
g <- erdos.renyi.game(n = 12,
                     10,
                     type = "gnm")

g

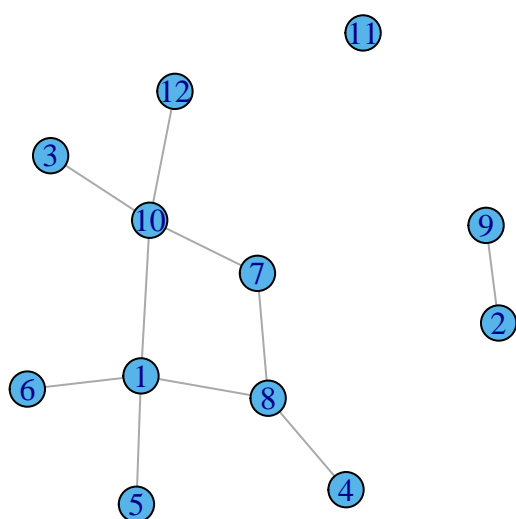
## IGRAPH e8992f5 U--- 12 10 -- Erdos renyi (gnm) graph
## + attr: name (g/c), type (g/c), loops (g/l), m (g/n)
## + edges from e8992f5:
## [1] 2-- 4 3-- 7 1-- 8 4-- 8 3--10 5--10 3--11 4--11 2--12 9--12

graph.density(g)

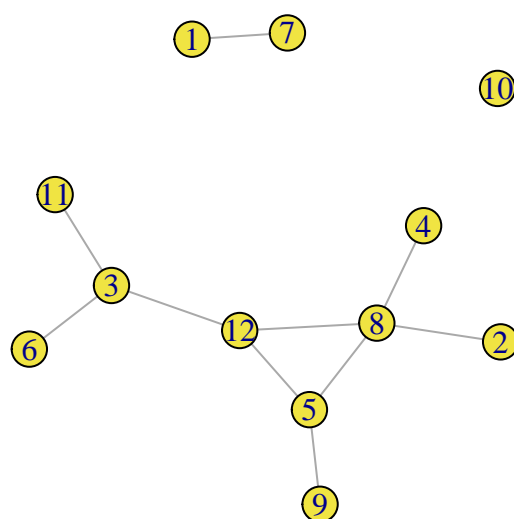
## [1] 0.1515152

op <- par(mar = c(0,1,3,1),
          mfrow = c(1,2))
plot(erdos.renyi.game(n = 12,
                     10,
                     type = "gnm"),
     vertex.color = 2,
     main = "first random graph")
plot(erdos.renyi.game(n = 12,
                     10,
                     type = "gnm"),
     vertex.color = 4,
     main = "second random graph")
```

first random graph



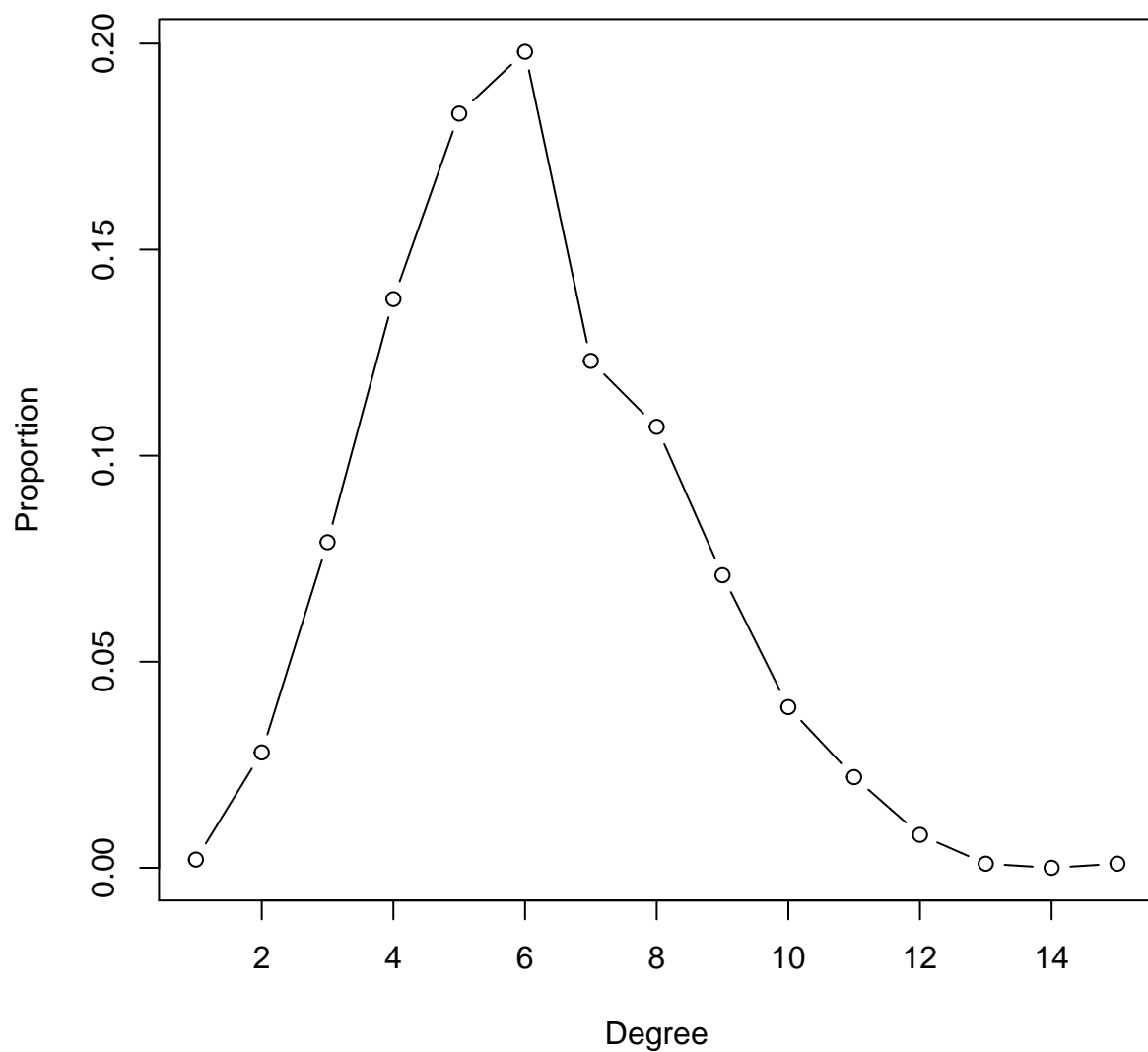
second random graph



```
par(op)
```

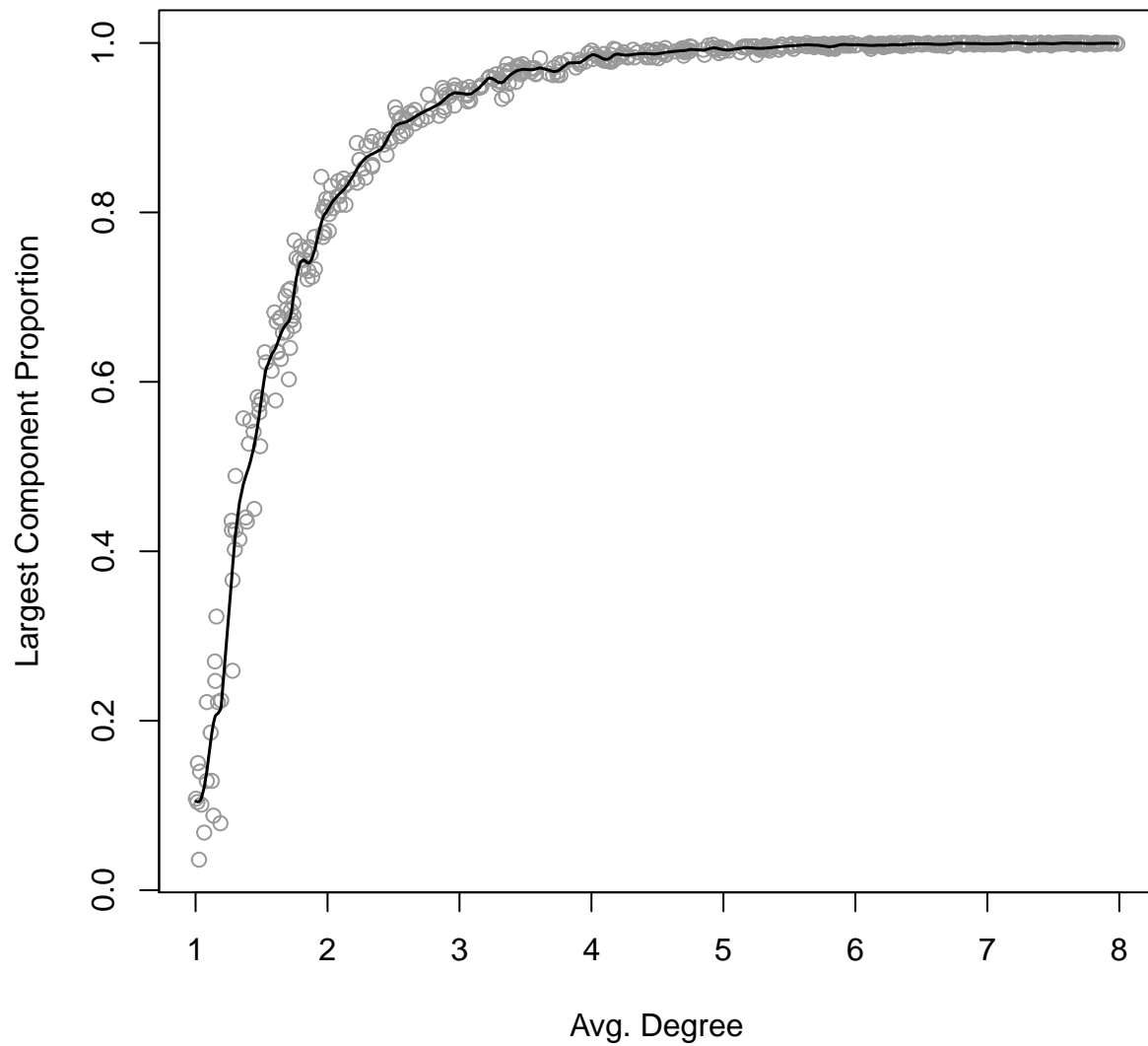
```
g <- erdos.renyi.game(n = 1000,  
                      0.005,  
                      type = "gnp")
```

```
plot(degree.distribution(g),  
     type = "b",  
     xlab = "Degree",  
     ylab = "Proportion")
```



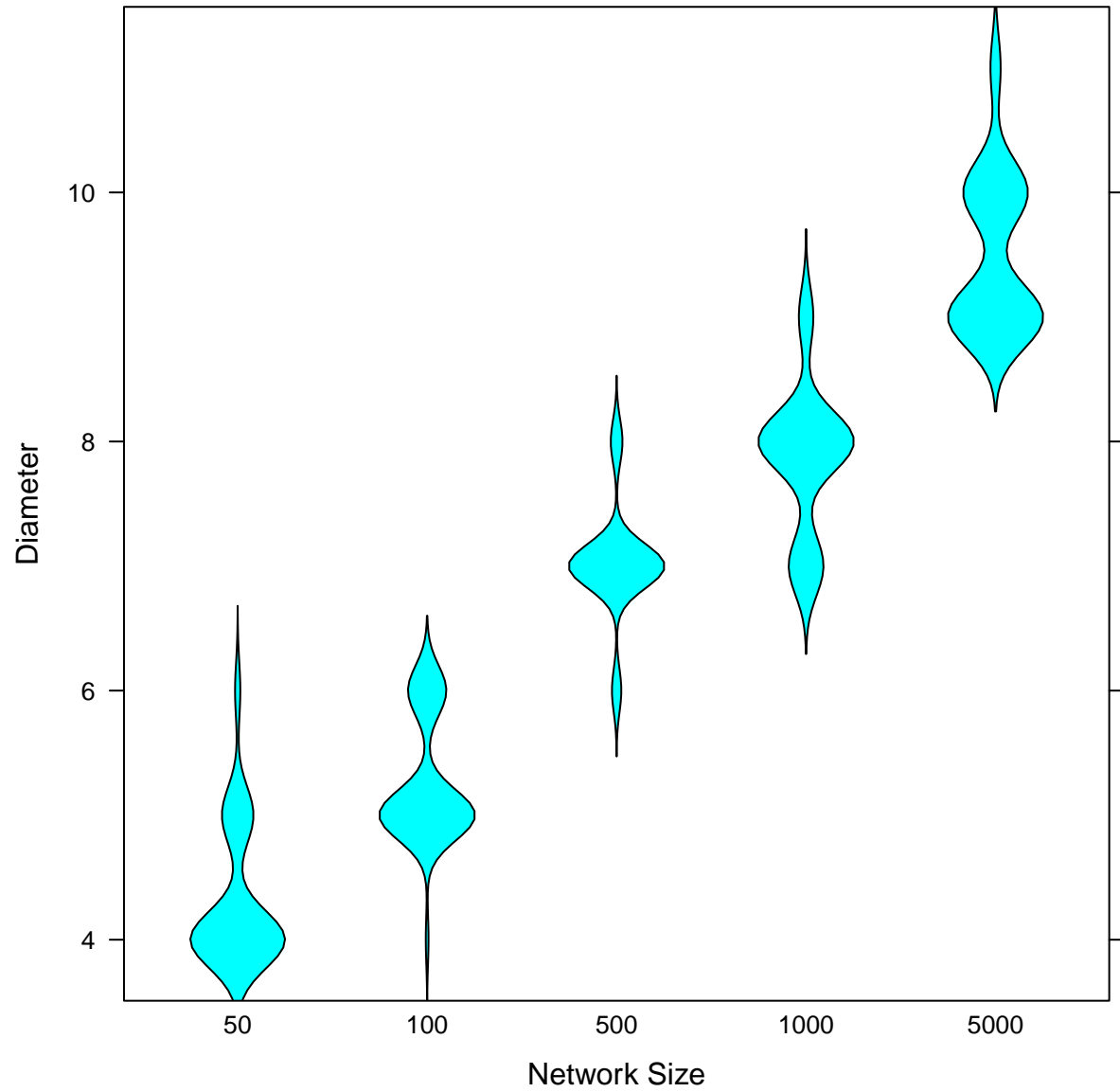
```
crnd <- runif(500, 1, 8)
cmp_prp <- sapply(crnd, function(x)
  max(clusters(erdos.renyi.game(n = 1000,
    p = x/999))$csize)/1000)
smoothingSpline <- smooth.spline(crnd,
  cmp_prp,
  spar = 0.25)
```

```
plot(crnd, cmp_prp,
  col = "grey60",
  xlab = "Avg. Degree",
  ylab = "Largest Component Proportion")
lines(smoothingSpline,
  lwd = 1.5)
```



```
n_vect <- rep(c(50, 100, 500, 1000, 5000),
              each = 50)
g_diam <- sapply(n_vect, function(x)
  diameter(erdos.renyi.game(n = x,
                           p = 6/(x - 1))))
library(lattice)
```

```
bwplot(g_diam ~ factor(n_vect),
       panel = panel.violin,
       xlab = "Network Size",
       ylab = "Diameter")
```



Small-World Model

```
g1 <- watts.strogatz.game(dim = 1,  
                           size = 30,  
                           nei = 2,  
                           p = 0)  
g2 <- watts.strogatz.game(dim = 1,  
                           size = 30,  
                           nei = 2,  
                           p = 0.05)  
g3 <- watts.strogatz.game(dim = 1,  
                           size = 30,  
                           nei = 2,
```

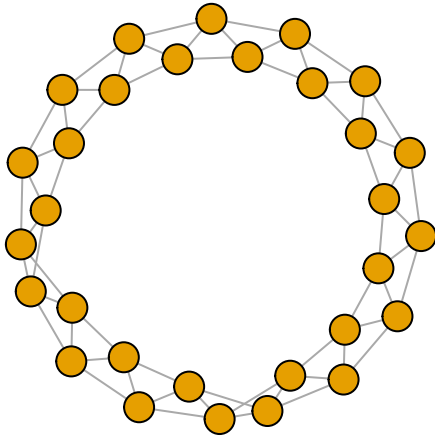
```

                                p = 0.20)
g4 <- watts.strogatz.game(dim = 1,
                          size = 30,
                          nei = 2,
                          p = 1)

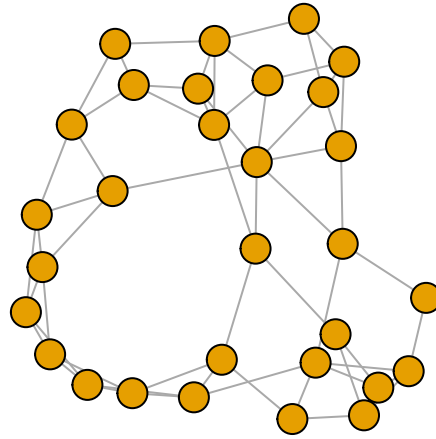
op <- par(mar = c(2,1,3,1),
          mfrow = c(2,2))
plot(g1,
      vertex.label = NA,
      main = expression(paste(italic(p), "= 0")),
      layout = layout_with_kk)
plot(g2,
      vertex.label = NA,
      main = expression(paste(italic(p), "= 0.05")))
plot(g3,
      vertex.label = NA,
      main = expression(paste(italic(p), "= 0.20")))
plot(g4,
      vertex.label = NA,
      main = expression(paste(italic(p), "= 1")))

```

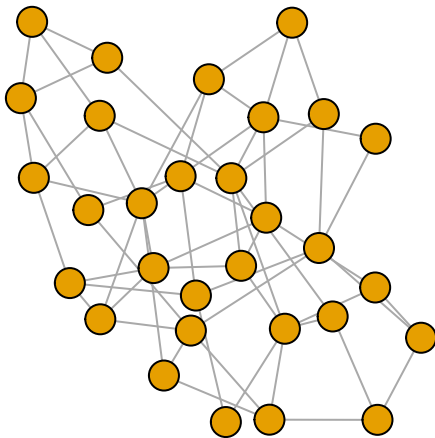
$p=0$



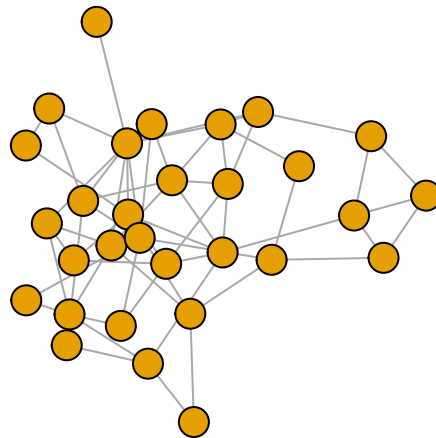
$p=0.05$



$p=0.20$



$p=1$



```
par(op)
```

```
g100 <- watts.strogatz.game(dim = 1,  
                             size = 100,  
                             nei = 2,  
                             p = 0)
```

```
g100
```

```
## IGRAPH 126943f U--- 100 200 -- Watts-Strogatz random graph  
## + attr: name (g/c), dim (g/n), size (g/n), nei (g/n), p (g/n), loops  
## | (g/l), multiple (g/l)  
## + edges from 126943f:  
## [1] 1-- 2 2-- 3 3-- 4 4-- 5 5-- 6 6-- 7 7-- 8 8-- 9 9--10 10--11  
## [11] 11--12 12--13 13--14 14--15 15--16 16--17 17--18 18--19 19--20 20--21  
## [21] 21--22 22--23 23--24 24--25 25--26 26--27 27--28 28--29 29--30 30--31
```



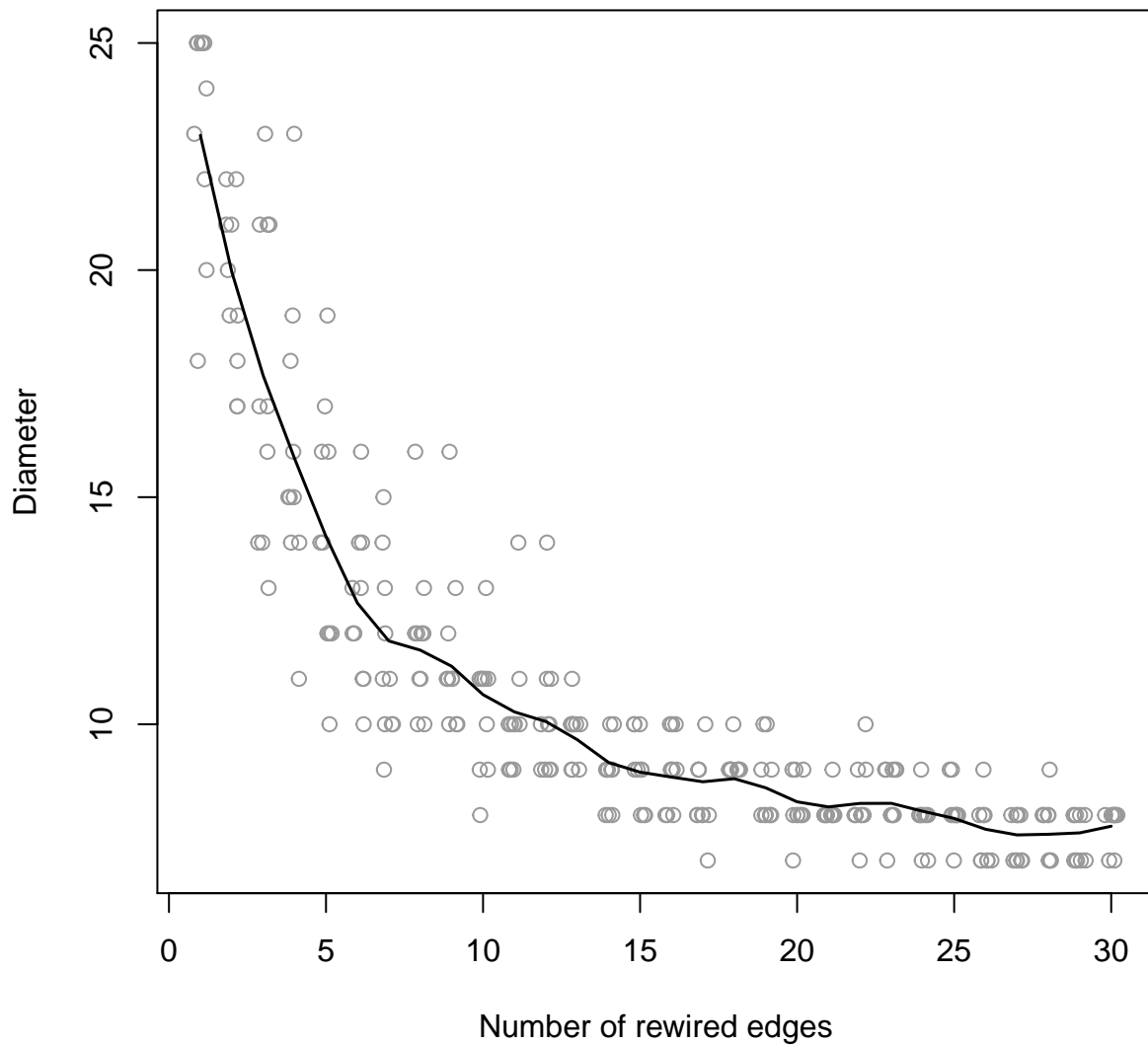
```
## [31] 31--32 32--33 33--34 34--35 35--36 36--37 37--38 38--39 39--40 40--41
## [41] 41--42 42--43 43--44 44--45 45--46 46--47 47--48 48--49 49--50 50--51
## [51] 51--52 52--53 53--54 54--55 55--56 56--57 57--58 58--59 59--60 60--61
## [61] 61--62 62--63 63--64 64--65 65--66 66--67 67--68 68--69 69--70 70--71
## + ... omitted several edges
```

```
diameter(g100)
```

```
## [1] 25
```

```
p_vect <- rep(1:30,
              each = 10)
g_diam <- sapply(p_vect,
                 function(x)
                   diameter(watts.strogatz.game(dim = 1,
                                                  size = 100,
                                                  nei = 2,
                                                  p=x/200)))
smoothingSpline <- smooth.spline(p_vect, g_diam,
                                  spar = 0.35)
```

```
plot(jitter(p_vect, 1),
     g_diam,
     col = "grey60",
     xlab = "Number of rewired edges",
     ylab = "Diameter")
lines(smoothingSpline,
      lwd = 1.5)
```



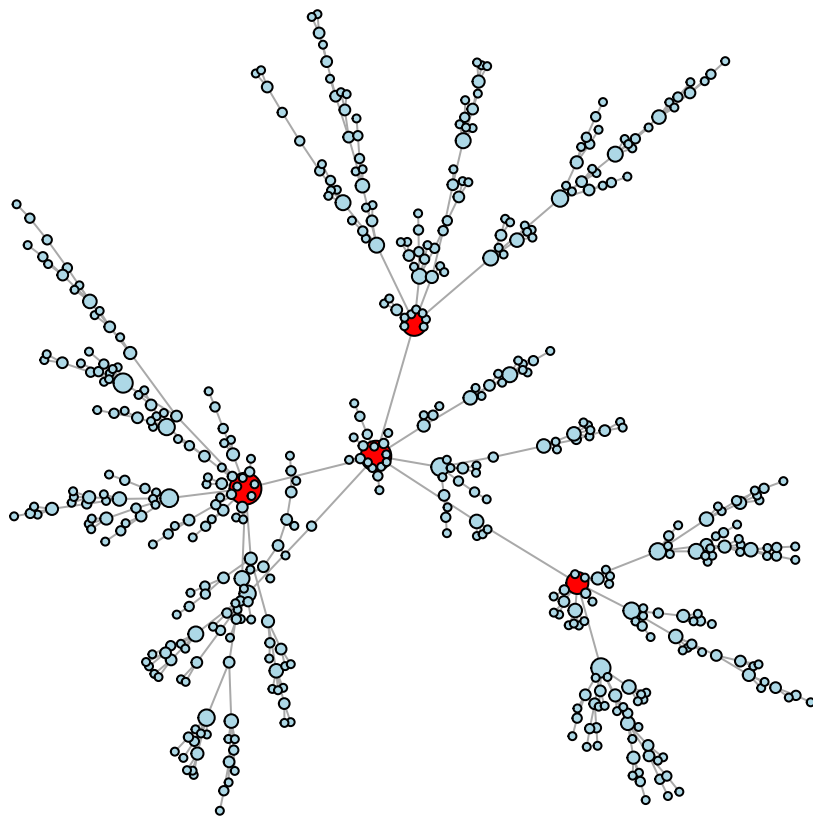
Scale-Free Models

```
# from chapter 5
rescale <- function(nchar, low, high){
  min_d <- min(nchar)
  max_d <- max(nchar)
  rscl <- ((high - low) * (nchar - min_d)) /
    (max_d - min_d) + low
  return(rscl)
}
```

```
g <- barabasi.game(500,
  directed = FALSE)
V(g)$color <- "lightblue"
```

```
V(g)[degree(g) > 9]$color <- "red"  
node_size <- rescale(nchar = degree(g),  
                     low = 2,  
                     high = 8)
```

```
plot(g,  
     vertex.label = NA,  
     vertex.size = node_size)
```



```
median(degree(g))
```

```
## [1] 1
```

```
mean(degree(g))
```

```
## [1] 1.996
```

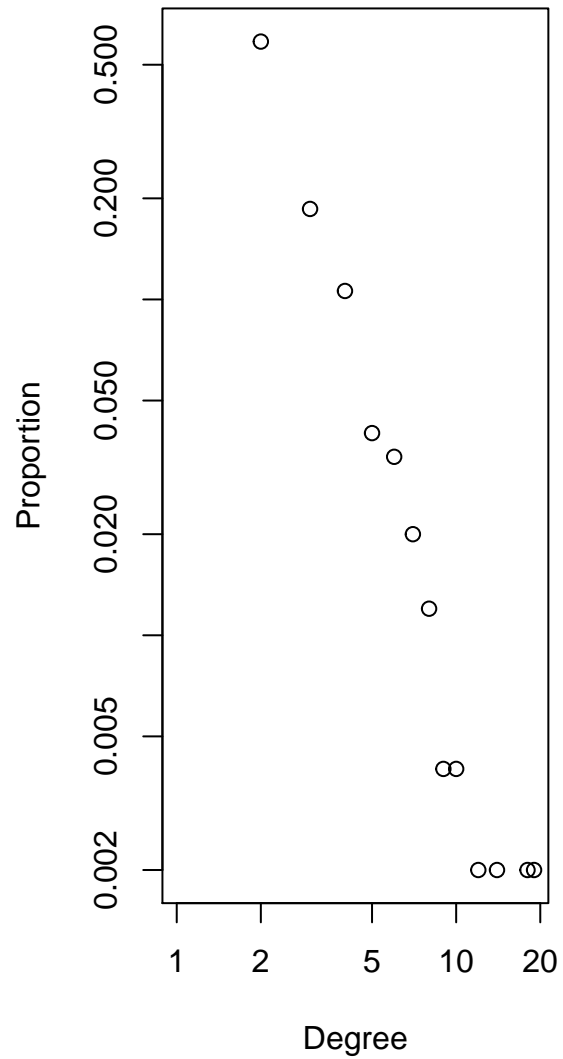
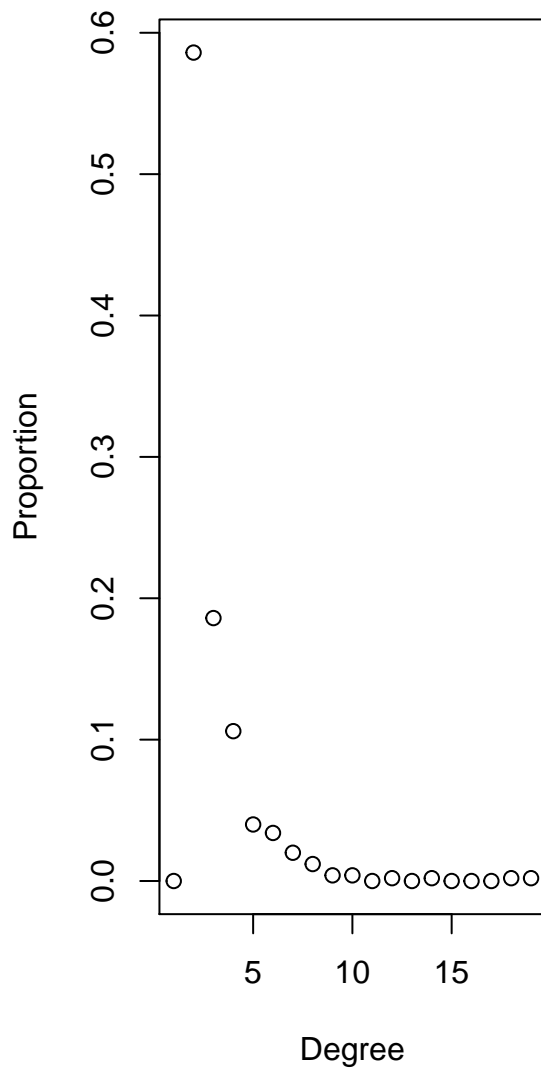
```
table(degree(g))
```

```
##
```

```
##  1  2  3  4  5  6  7  8  9 11 13 17 18  
## 293 93 53 20 17 10  6  2  2  1  1  1  1
```

```
op <- par(mfrow = c(1,2))  
plot(degree.distribution(g),  
      xlab = "Degree",  
      ylab = "Proportion")  
plot(degree.distribution(g),  
      log = "xy",  
      xlab = "Degree",  
      ylab = "Proportion")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 6 y values <= 0 omitted from  
## logarithmic plot
```

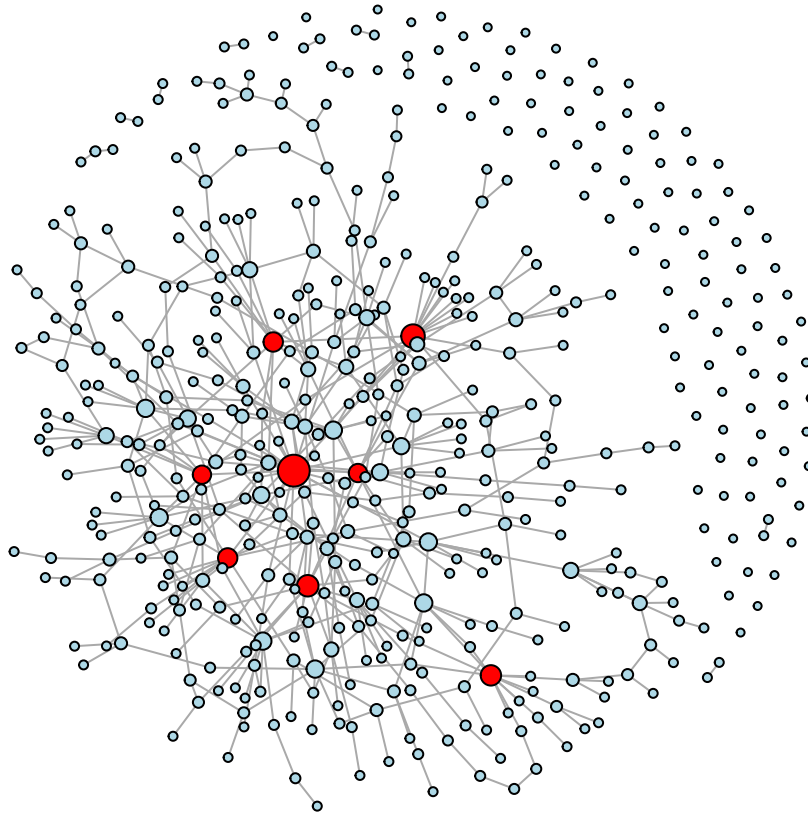


```
par(op)

g <- barabasi.game(500,
  out.dist = c(0.25, 0.5, 0.25),
  directed = FALSE,
  zero.appeal = 1)
V(g)$color <- "lightblue"
V(g)[degree(g) > 9]$color <- "red"
node_size <- rescale(nchar = degree(g),
  low = 2,
  high = 8)

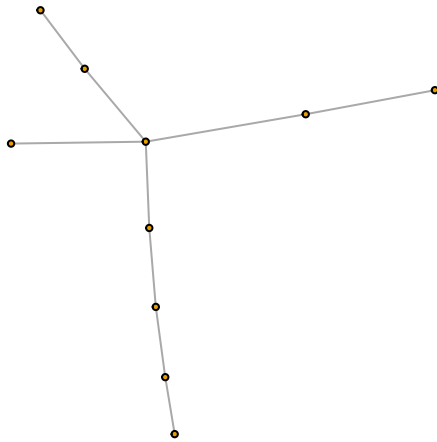
plot(g,
  vertex.label = NA,
```

```
vertex.size = node_size)
```

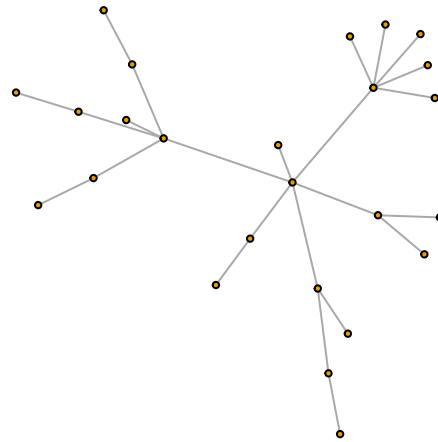


```
g1 <- barabasi.game(10,  
  m = 1,  
  directed = FALSE)  
g2 <- barabasi.game(25,  
  m = 1,  
  directed = FALSE)  
g3 <- barabasi.game(50,  
  m = 1,  
  directed = FALSE)  
g4 <- barabasi.game(100,  
  m = 1,  
  directed = FALSE)
```

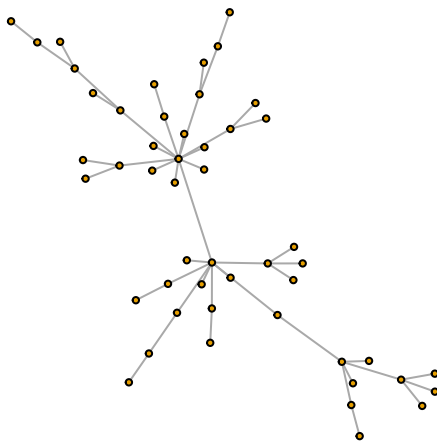
```
op <- par(mfrow = c(2,2),
          mar = c(4, 0, 1, 0))
plot(g1,
      vertex.label = NA,
      vertex.size = 3,
      xlab = " n = 10")
plot(g2,
      vertex.label = NA,
      vertex.size = 3,
      xlab = " n = 25")
plot(g3,
      vertex.label = NA,
      vertex.size = 3,
      xlab = " n = 50")
plot(g4,
      vertex.label = NA,
      vertex.size = 3,
      xlab = " n = 100")
```



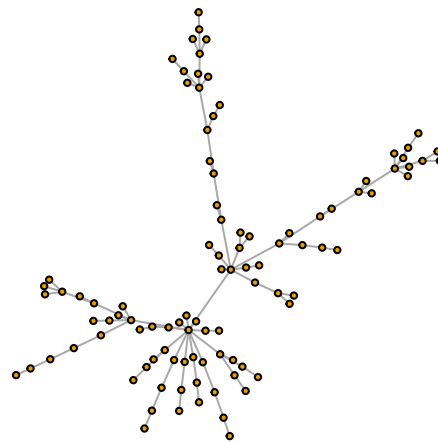
n = 10



n = 25



n = 50



n = 100

```
par(op)
```

Comparing Random Models to Empirical models

```
data(lhds)

ilhds <- as.igraph(lhds) # asIgraph in book is unknown as.igraph is in the igraph package but returns
ilhds
graph.density(ilhds)
mean(degree(ilhds))
g_rnd <- erdos.renyi.game(1283,
                          0.0033,
                          type = "gnp")
```



```
g_smwrld <- watts.strogatz.game(dim = 1,  
                                size = 1283,  
                                nei = 2,  
                                p = 0.25)  
g_prfatt <- barabasi.game(1283,  
                          out.dist = c(0.15,  
                                       0.6,  
                                       0.25),  
                          directed = FALSE,  
                          zero.appeal = 2)
```