Networks

Jake Hofman

April 7, 2017

Contents


```
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(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## as_data_frame(): dplyr, tibble, igraph
## compose():       purrr, igraph
## crossing():      tidyr, igraph
## filter():        dplyr, stats
## groups():        dplyr, igraph
## lag():           dplyr, stats
## simplify():      purrr, igraph
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
```

```
##     discard
```

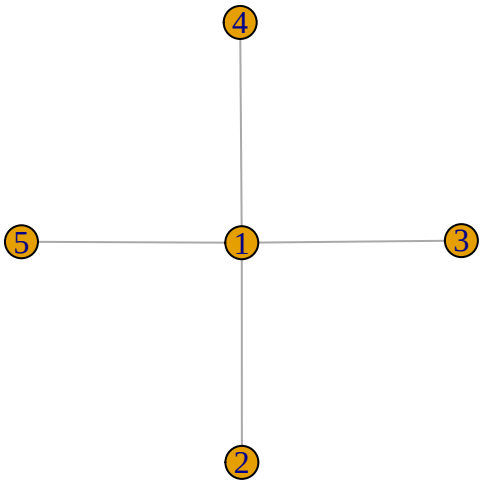
```
## The following objects are masked from 'package:readr':
##
##     col_factor, col_numeric
```

```
theme_set(theme_bw())
```

Toy networks

Star network

```
# look at edge list and adjacency matrix
star <- graph.star(5, mode="undirected", center=1)
plot(star)
```




```
get.edgelist(star)
```

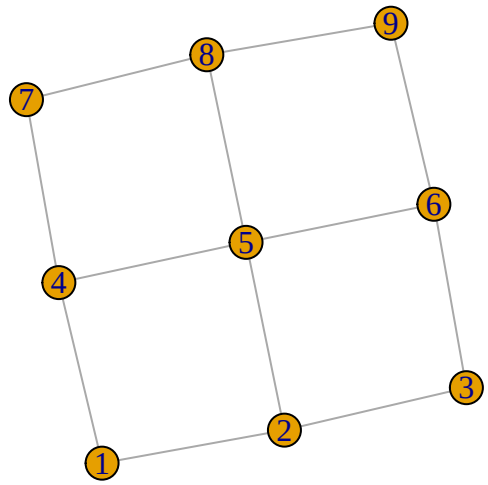
```
##      [,1] [,2]
## [1,]    1    2
## [2,]    1    3
## [3,]    1    4
## [4,]    1    5
```

```
get.adjacency(star)
```

```
## 5 x 5 sparse Matrix of class "dgCMatrix"
##
## [1,] . 1 1 1 1
## [2,] 1 . . . .
## [3,] 1 . . . .
## [4,] 1 . . . .
## [5,] 1 . . . .
```

Lattice network

```
# look at edge list and adjacency matrix
grid <- graph.lattice(length=3, dim=2)
plot(grid)
```




```
get.edgelist(grid)
```

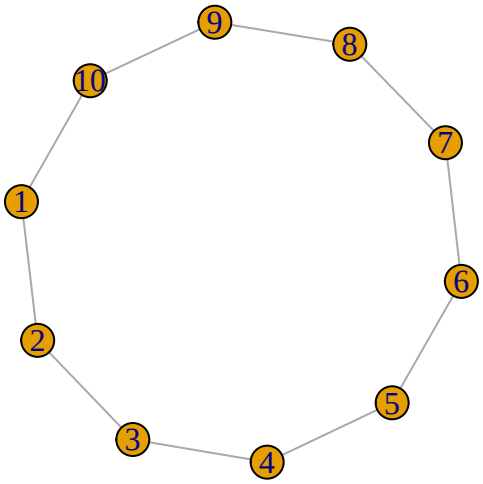
```
##       [,1] [,2]
##  [1,]    1    2
##  [2,]    1    4
##  [3,]    2    3
##  [4,]    2    5
##  [5,]    3    6
##  [6,]    4    5
##  [7,]    4    7
##  [8,]    5    6
##  [9,]    5    8
## [10,]    6    9
## [11,]    7    8
## [12,]    8    9
```

```
get.adjacency(grid)
```

```
## 9 x 9 sparse Matrix of class "dgCMatrix"
##
##  [1,] . 1 . 1 . . . . .
##  [2,] 1 . 1 . 1 . . . .
##  [3,] . 1 . . . 1 . . .
##  [4,] 1 . . . 1 . 1 . .
##  [5,] . 1 . 1 . 1 . 1 .
##  [6,] . . 1 . 1 . . . 1
##  [7,] . . . 1 . . . 1 .
##  [8,] . . . . 1 . 1 . 1
##  [9,] . . . . . 1 . 1 .
```

Ring network

```
# look at edge list and adjacency matrix
grid <- graph.ring(10)
plot(grid)
```




```
get.edgelist(grid)
```

```
##       [,1] [,2]
##  [1,]    1    2
##  [2,]    2    3
##  [3,]    3    4
##  [4,]    4    5
##  [5,]    5    6
##  [6,]    6    7
##  [7,]    7    8
##  [8,]    8    9
##  [9,]    9   10
## [10,]    1   10
```

```
get.adjacency(grid)
```

```
## 10 x 10 sparse Matrix of class "dgCMatrix"
##
##  [1,] . 1 . . . . . . . 1
##  [2,] 1 . 1 . . . . . . .
##  [3,] . 1 . 1 . . . . . .
##  [4,] . . 1 . 1 . . . . .
##  [5,] . . . 1 . 1 . . . .
##  [6,] . . . . 1 . 1 . . .
##  [7,] . . . . . 1 . 1 . .
##  [8,] . . . . . . 1 . 1 .
##  [9,] . . . . . . . 1 . 1
## [10,] 1 . . . . . . . 1 .
```

Path length

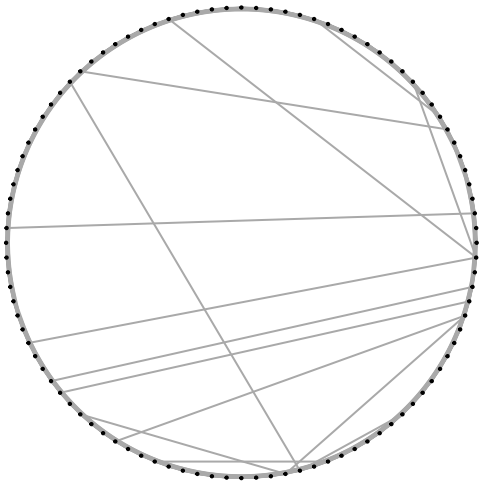
```
# look at all-pairs shortest path distances
shortest.paths(grid)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    0    1    2    3    4    5    4    3    2     1
##  [2,]    1    0    1    2    3    4    5    4    3     2
##  [3,]    2    1    0    1    2    3    4    5    4     3
##  [4,]    3    2    1    0    1    2    3    4    5     4
##  [5,]    4    3    2    1    0    1    2    3    4     5
##  [6,]    5    4    3    2    1    0    1    2    3     4
##  [7,]    4    5    4    3    2    1    0    1    2     3
##  [8,]    3    4    5    4    3    2    1    0    1     2
##  [9,]    2    3    4    5    4    3    2    1    0     1
## [10,]    1    2    3    4    5    4    3    2    1     0
```

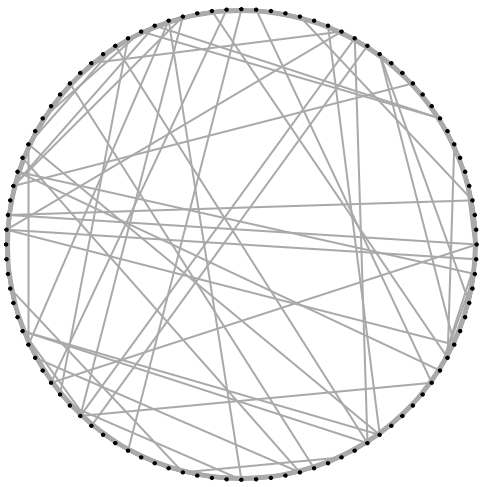
Watts-Strogatz

Plot a few watts-strogatz small world networks

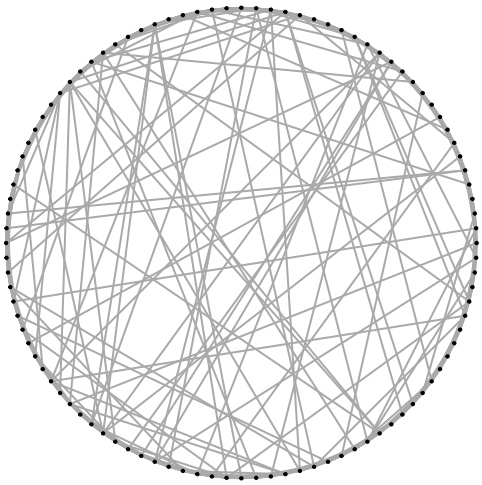
```
# mostly a ring
plot(watts.strogatz.game(1, 100, 5, 0.01), layout=layout.circle, vertex.size=1, vertex.label=NA)
```



```
# some rewiring
plot(watts.strogatz.game(1, 100, 5, 0.05), layout=layout.circle, vertex.size=1, vertex.label=NA)
```



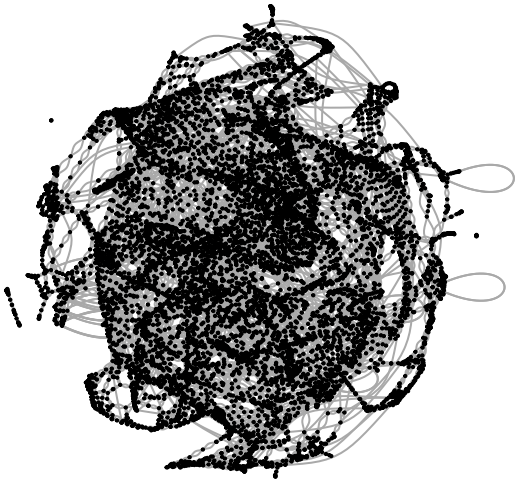
```
# lots of rewiring
plot(watts.strogatz.game(1, 100, 5, 0.10), layout=layout.circle, vertex.size=1, vertex.label=NA)
```



Real networks

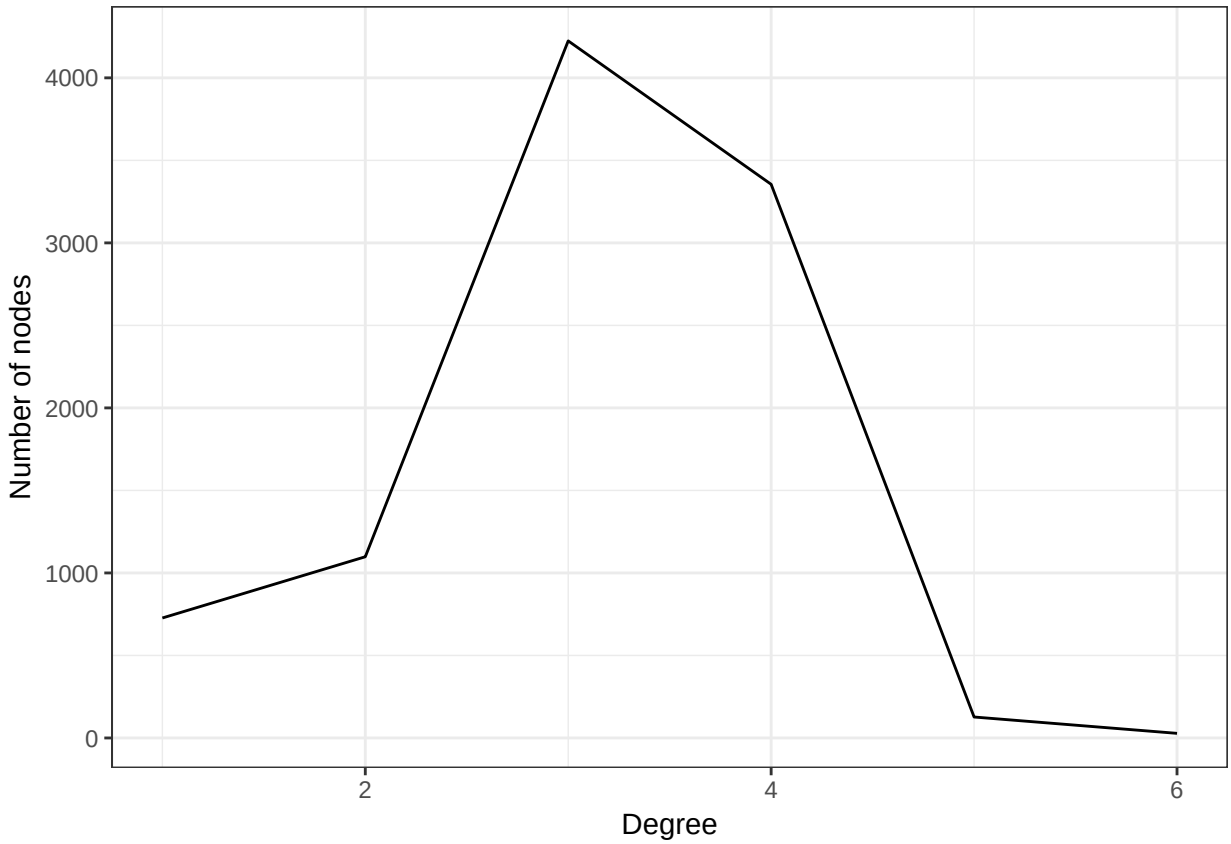
Washington DC road network

```
# read in edge list
dc_edges <- read.table('dc_road_network.tsv', sep="\t", header=F, col.names=c('src','dst'))

# convert to igraph object
dc_graph <- graph.data.frame(dc_edges, directed=F)

# plot hairball
plot(dc_graph, vertex.size=1, vertex.label=NA)
```



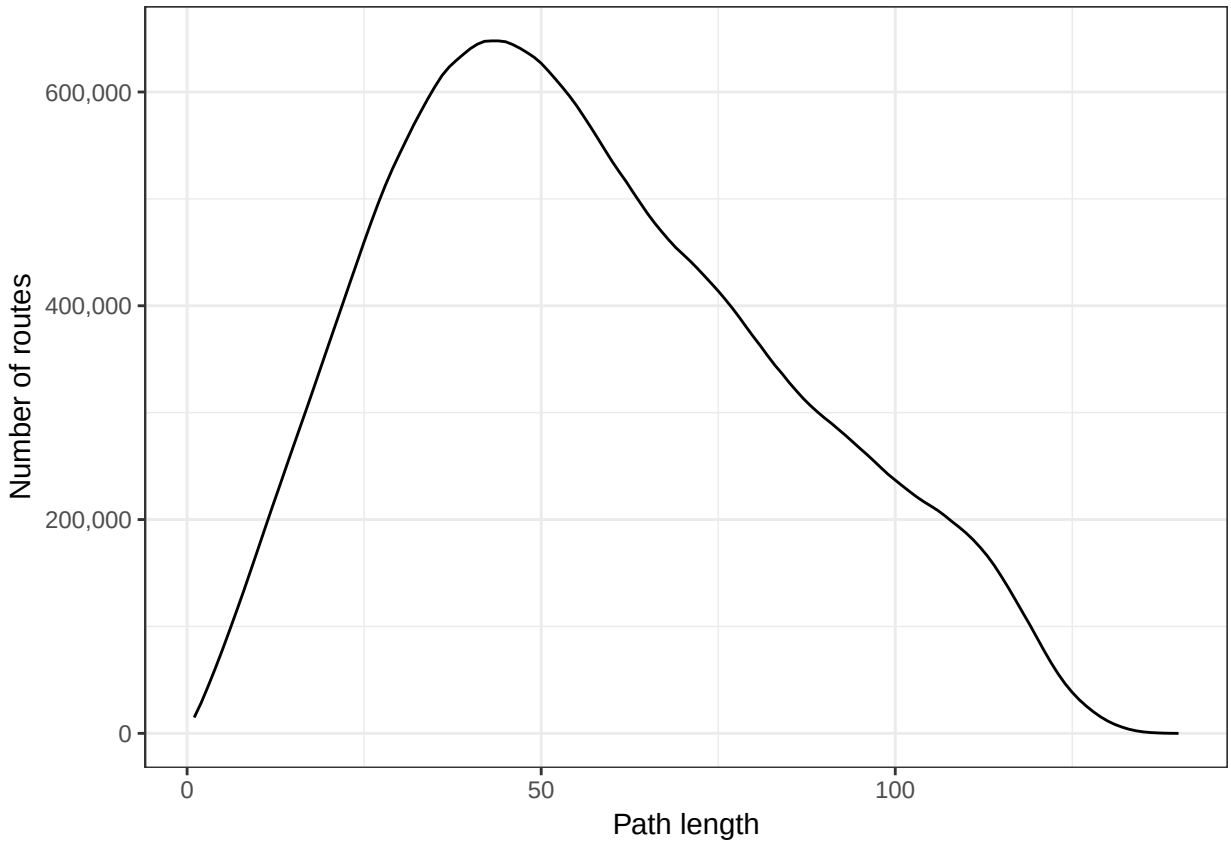
Degree distribution

```
# compute degree distribution
dc_degree_dist <- dc_edges %>%
  group_by(src) %>%
  summarize(degree=n()) %>%
  group_by(degree) %>%
  summarize(num_nodes=n())
ggplot(dc_degree_dist, aes(x = degree, y = num_nodes)) +
  geom_line() +
  xlab('Degree') +
  ylab('Number of nodes')
```



Path length

```
# plot distribution of path lengths
count <- path.length.hist(dc_graph)$res
plot_data <- data.frame(path_length = 1:length(count), count)
ggplot(plot_data, aes(x = path_length, y = count)) +
  geom_line() +
  xlab('Path length') +
  ylab('Number of routes') +
  scale_y_continuous(label = comma)
```




```
# compute mean path length
sum(1:length(count)*count)/sum(count)
```

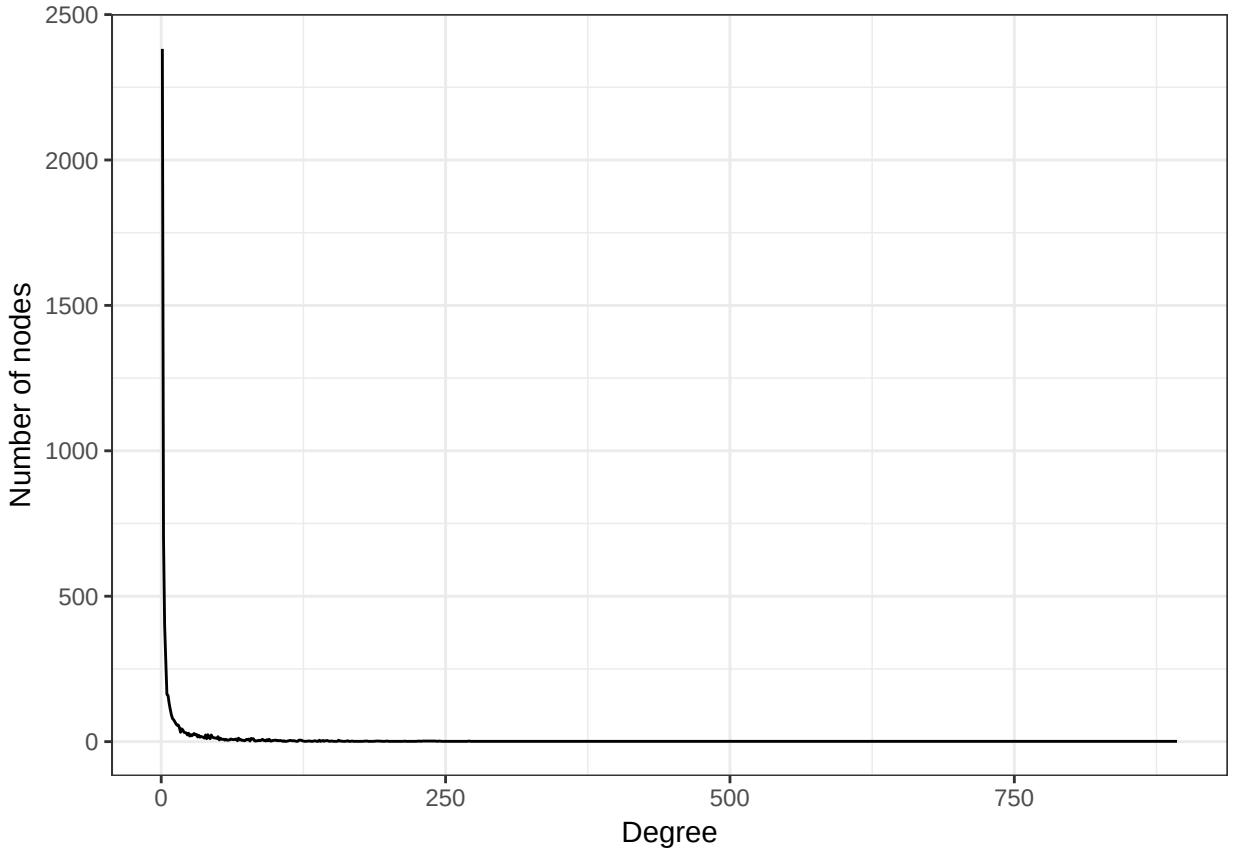
```
## [1] 57.03254
```

Wikipedia voting network

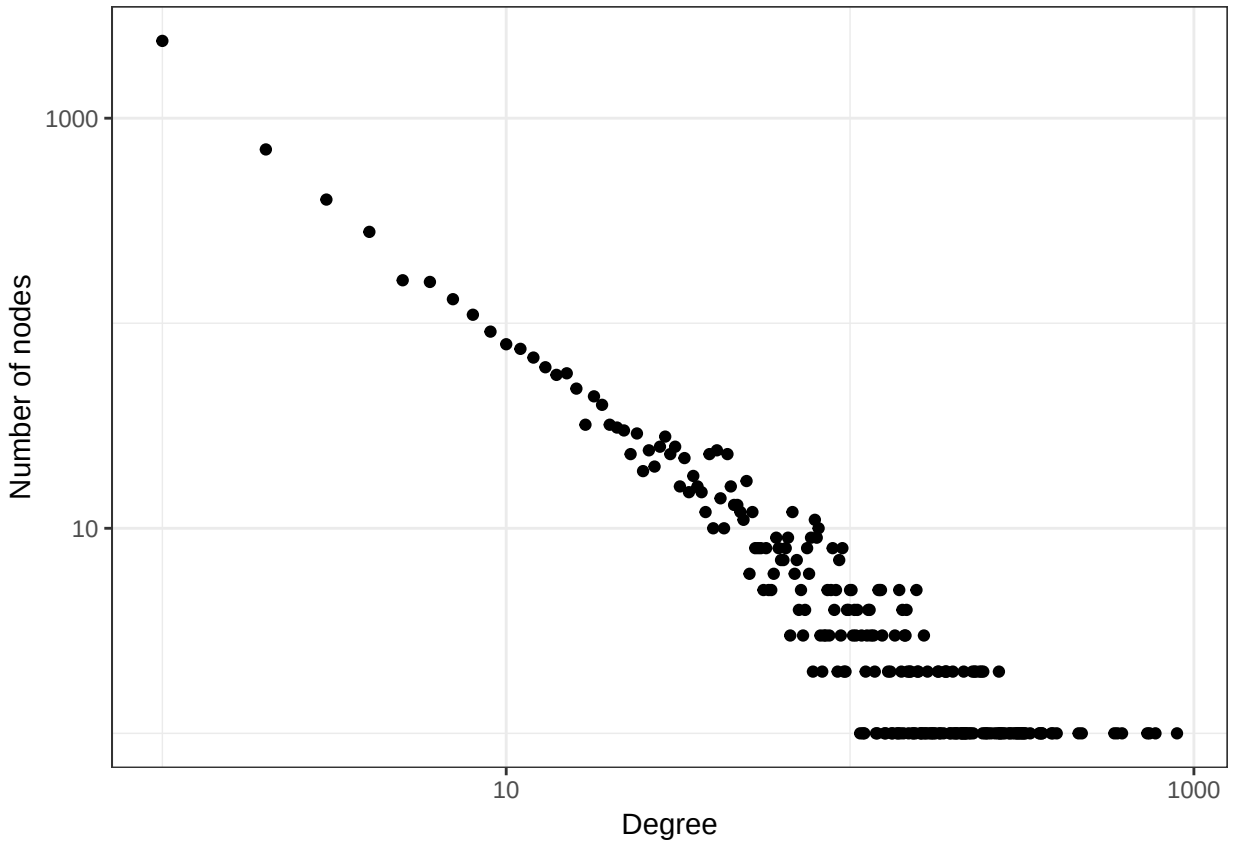
```
# read in edge list
wiki_edges <- read.table('wiki-Vote.txt', sep="\t", header=F, col.names=c('src','dst'))

wiki_graph <- graph.data.frame(wiki_edges, directed=T)
```

Degree distribution

```
wiki_degree_dist <- wiki_edges %>%
  group_by(src) %>%
  summarize(degree=n()) %>%
  group_by(degree) %>%
  summarize(num_nodes=n())
ggplot(wiki_degree_dist, aes(x = degree, y = num_nodes)) +
  geom_line() +
  xlab('Degree') +
  ylab('Number of nodes')
```

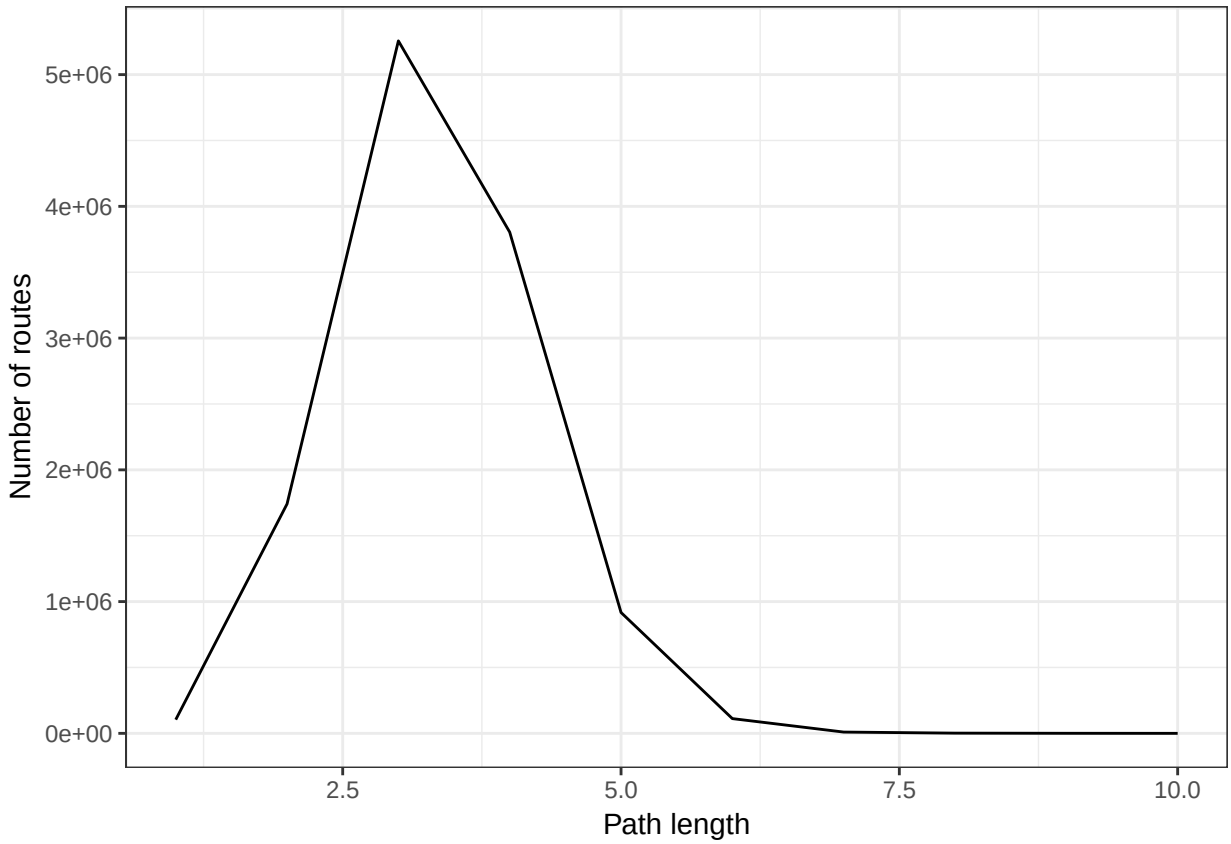



```
ggplot(wiki_degree_dist, aes(x = degree, y = num_nodes)) +
  geom_point() +
  xlab('Degree') +
  ylab('Number of nodes') +
  scale_y_log10() +
  scale_x_log10()
```



Path length

```
# plot distribution of path lengths
count <- path.length.hist(wiki_graph)$res
plot_data <- data.frame(path_length = 1:length(count), count)
ggplot(plot_data, aes(x = path_length, y = count)) +
  geom_line() +
  xlab('Path length') +
  ylab('Number of routes')
```




```
# compute mean path length
sum(1:length(count)*count)/sum(count)
```

```
## [1] 3.341011
```