Social Network Analysis Home Assignment 4

Lev Mazaev

due date - 11.06.2019 23:59

Contents

Network Epidemics

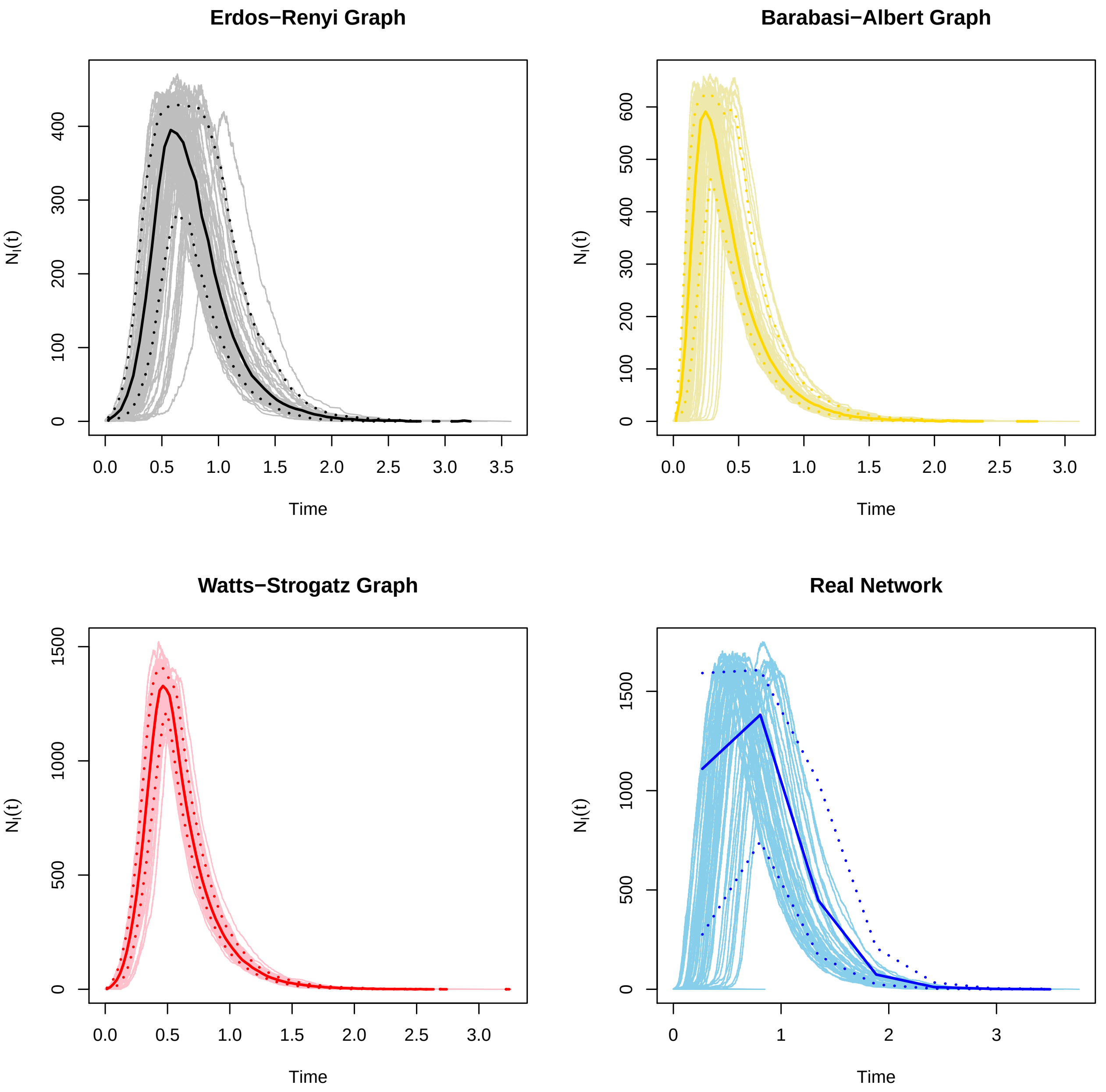
SIR Model

Generation of random graphs and loading real network

```r
graphList <- list()
graphList$`Erdos-Renyi` <- erdos.renyi.game(n = 1000, p.or.m = 0.01, type=c("gnp"))
graphList$`Barabasi-Albert` <- barabasi.game(n = 1000, m = 10, directed=FALSE)
graphList$`Watts-Strogatz` <- watts.strogatz.game(dim = 2, size = 50, nei = 3, p = 0.001)
graphList$`Real Network` <- read_graph(file = 'Net.txt', directed = FALSE)
```

Simulation plots

```r
sim <- lapply(graphList, sir, beta = 2, gamma = 4, no.sim = 100)
par(mfrow = c(2, 2))
plot(sim$`Erdos-Renyi`, main = 'Erdos-Renyi Graph',
     color = 'gray', median_color = 'black', quantile_color = 'black')
plot(sim$`Barabasi-Albert`, main = 'Barabasi-Albert Graph',
     color = "palegoldenrod", median_color = "gold", quantile_color = "gold")
plot(sim$`Watts-Strogatz`, main = 'Watts-Strogatz Graph',
     color = "pink", median_color = "red", quantile_color = "red")
plot(sim$`Real Network`, main = 'Real Network')
```



On these plots we observe simulation lines for the number of infected nodes for each graph model. But it's an inconvenient representation to make some inferences about SIR model in case of different graphs.

Plots of the graphs

Functions

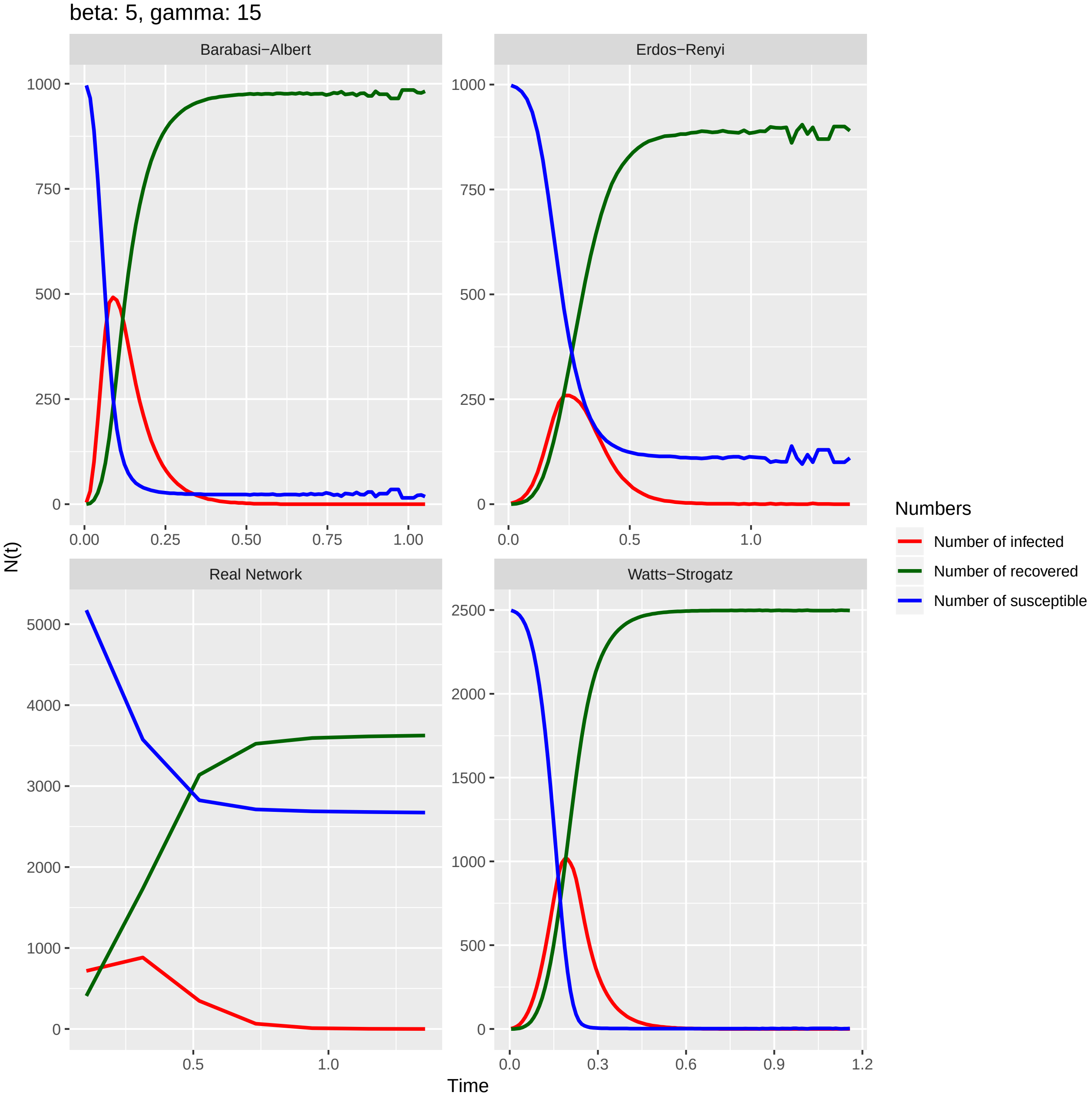
```
naImpute <- function(x) {
  # this function imputes NA values of the vector
  # with its previous values
  # this is to remove line breaks
  for (i in 2:length(x)) {
    if (is.na(x[i])) x[i] <- x[i-1]
  }
  return(x)
}

plotFunc <- function(gl, beta, gamma, niter) {
  # this function plots 4 graphs: one for each graph
  simG <- lapply(X = gl, FUN = sir, beta = beta, gamma = gamma, no.sim = niter)
  binMed <- lapply(X = simG, FUN = time_bins, middle = TRUE)
  medians <- lapply(X = simG, FUN = median, na.rm = TRUE)
  dataFrames <- map2(binMed, medians, function(bin, numList) {
    numList <- lapply(X = numList, FUN = naImpute)
    df <- data.frame(Time = bin, numList)
    return(df)
  })
  dataFrames <- bind_rows(dataFrames, .id = 'Graph')
  dataFrames <- gather(data = dataFrames, key = 'Numbers', value = 'N(t)', 3:5)
  plotTitle <- paste0('beta: ', beta, ', gamma: ', gamma)
  plotObj <- ggplot(data = dataFrames, mapping = aes(x = Time, y = `N(t)`, color = Numbers)) +
    geom_line(size = 1) +
    facet_wrap(facets = ~ Graph, nrow = 2, ncol = 2,
               scales = 'free') +
    scale_colour_manual(values = c('NI' = 'red',
                                   'NR' = 'darkgreen',
                                   'NS' = 'blue'),
                        labels = c('Number of infected',
                                   'Number of recovered',
                                   'Number of susceptible')) +
    ggtitle(plotTitle)
  return(list(values = dataFrames, plot = plotObj))
}
```

Plots

β is infection rate, γ is recovery rate in SIR model.

1st case: $\beta = 5$, $\gamma = 15$

```
sim1 <- plotFunc(gl = graphList, beta = 5, gamma = 15, niter = 2000)
sim1$plot
```



All 3 random graph models (ER, BA and WS) show the same pattern: the number of infected has a substantial peak in the beginning of timeline (for approx. 1 unit of timeline) and then it gradually decreases to zero, so the infection is epidemic. The absolute values differ: the infection is the strongest in WS graph, where the number of susceptible nodes drops to almost zero in the end, and is much weaker in ER graph, where approximately 7/8 of the nodes were infected. The BA graph is somewhere in the middle. Regarding real network there are some problems with modelling, only 7 time points are calculated. It can be observed that the infection is gradually decreasing since the start and that a bit more than half of the nodes were affected.

Also there must be some modelling artifacts causing these oscillations of number lines in the end of timeline.

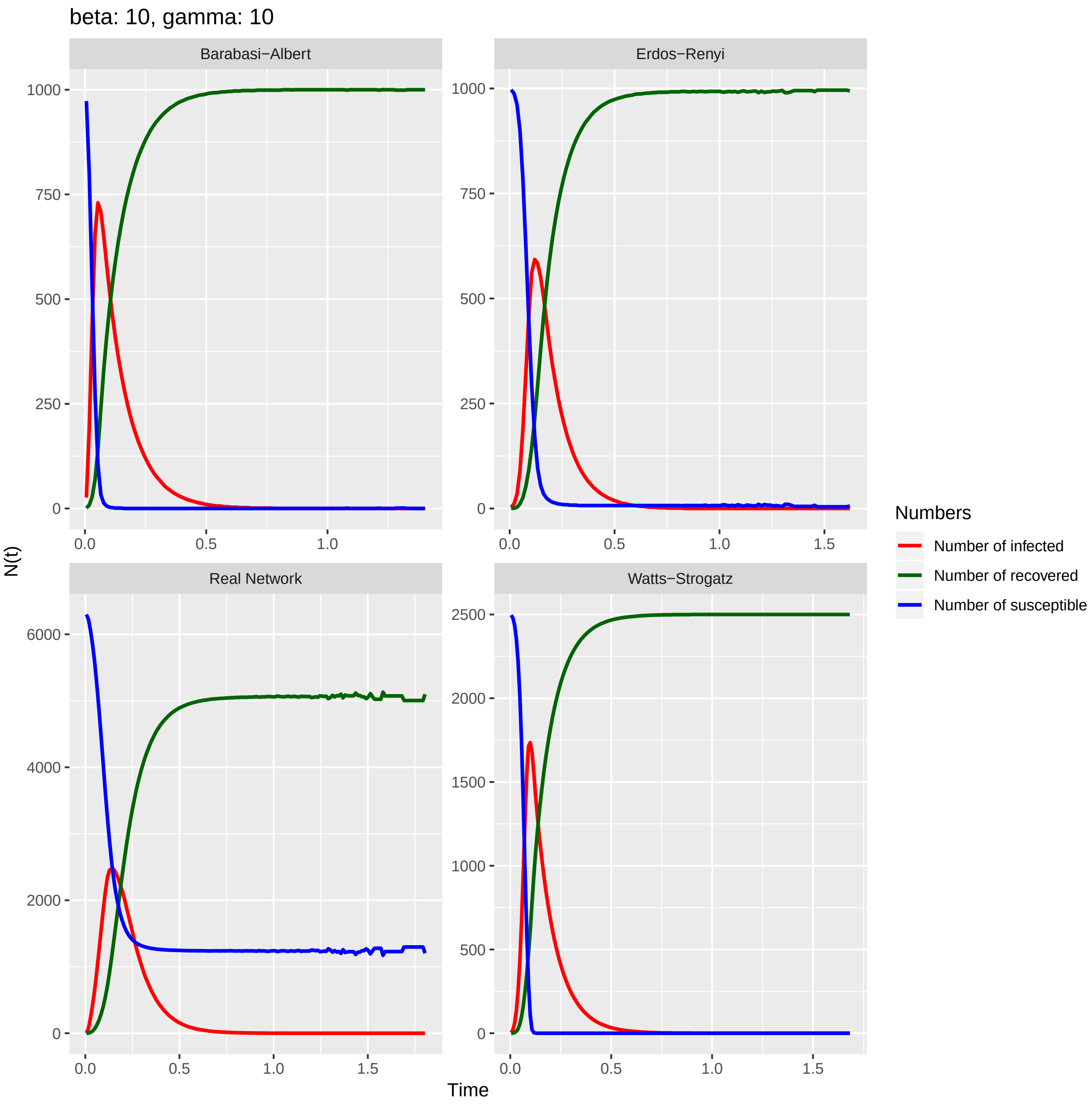
Below are minimum and maximum numbers of susceptible/infected/recovered nodes of all graphs.

```
knitr::kable(sim1$values %>%
               group_by(Graph, Numbers) %>%
               summarise(min = min(`N(t)`), max = max(`N(t)`)))
```

Graph	Numbers	min	max
Barabasi-Albert	NI	0.0	492.0
Barabasi-Albert	NR	0.0	985.0
Barabasi-Albert	NS	15.0	996.0
Erdos-Renyi	NI	0.0	259.0
Erdos-Renyi	NR	0.0	904.5
Erdos-Renyi	NS	95.5	998.0
Real Network	NI	0.0	883.0
Real Network	NR	408.0	3625.0
Real Network	NS	2673.0	5173.0
Watts-Strogatz	NI	0.0	1015.0
Watts-Strogatz	NR	0.0	2499.0
Watts-Strogatz	NS	1.0	2498.0

2nd case: $\beta = 10$, $\gamma = 10$

```r
sim2 <- plotFunc(gl = graphList, beta = 10, gamma = 10, niter = 2000)
sim2$plot
```



The infection has become epidemic in all 4 cases. Zero susceptible nodes left in WA and BA graphs, almost zero in ER graph and 18% in real network. Regarding the timeline the behavior is the same for all graphs - the infection peak is within the 1st timeline unit.

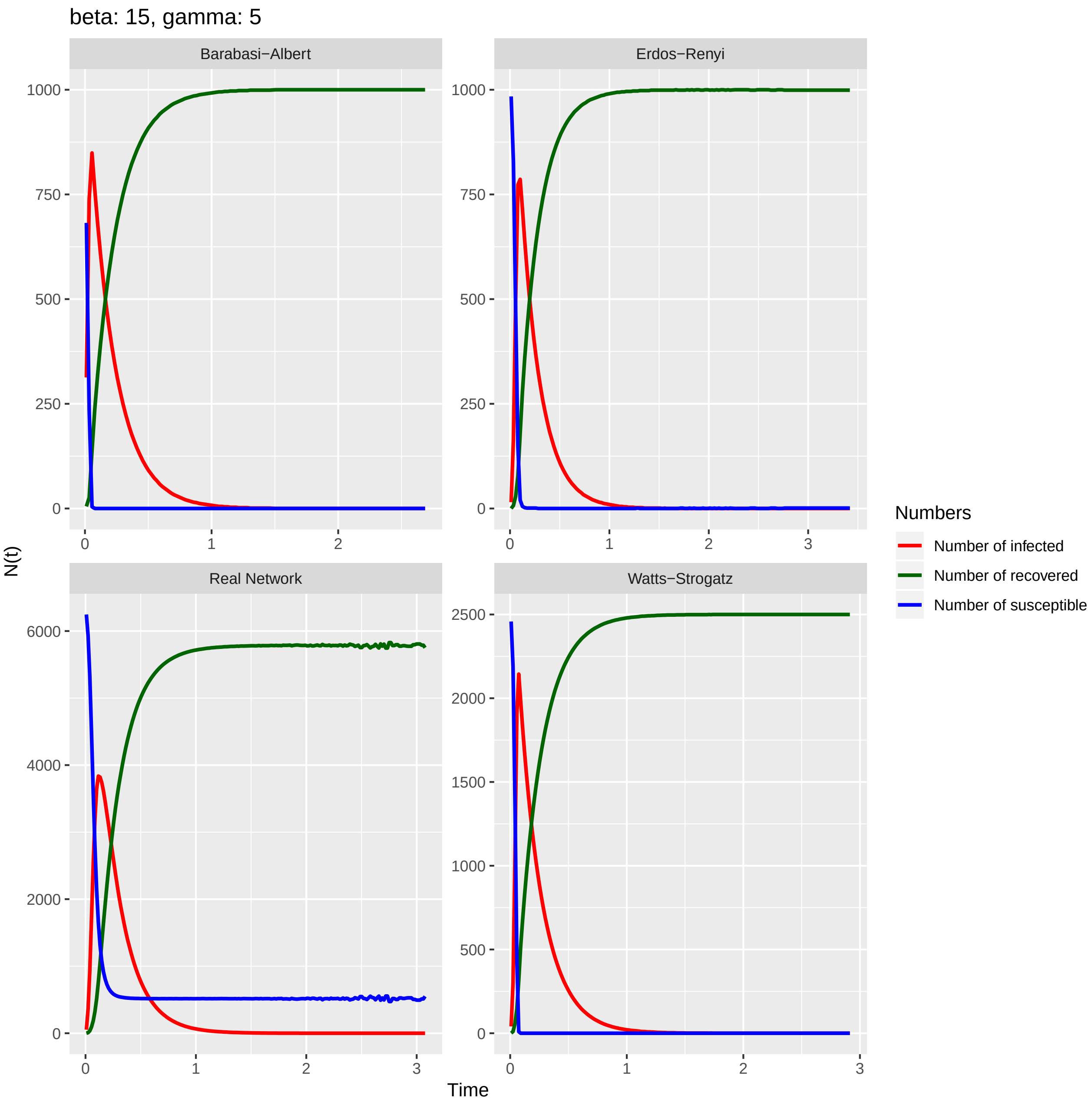
```
knitr::kable(sim2$values %>%
               group_by(Graph, Numbers) %>%
               summarise(min = min(`N(t)`), max = max(`N(t)`)))
```

Graph	Numbers	min	max
Barabasi-Albert	NI	0	730
Barabasi-Albert	NR	1	1000
Barabasi-Albert	NS	0	973
Erdos-Renyi	NI	0	593
Erdos-Renyi	NR	0	996

Graph	Numbers	min	max
Erdos-Renyi	NS	4	997
Real Network	NI	0	2474
Real Network	NR	0	5130
Real Network	NS	1171	6298
Watts-Strogatz	NI	0	1735
Watts-Strogatz	NR	0	2500
Watts-Strogatz	NS	0	2496

3rd case: $\beta = 15$, $\gamma = 5$

```r
sim3 <- plotFunc(gl = graphList, beta = 15, gamma = 5, niter = 2000)
sim3$plot
```



The infection is epidemic again in all 4 cases. Number of susceptible nodes left in random graphs is zero and 7% in real network. The timeline of simulation has become longer, the infection peaks are in the first 2 units.

```
knitr::kable(sim3$values %>%
               group_by(Graph, Numbers) %>%
               summarise(min = min(`N(t)`), max = max(`N(t)`)))
```

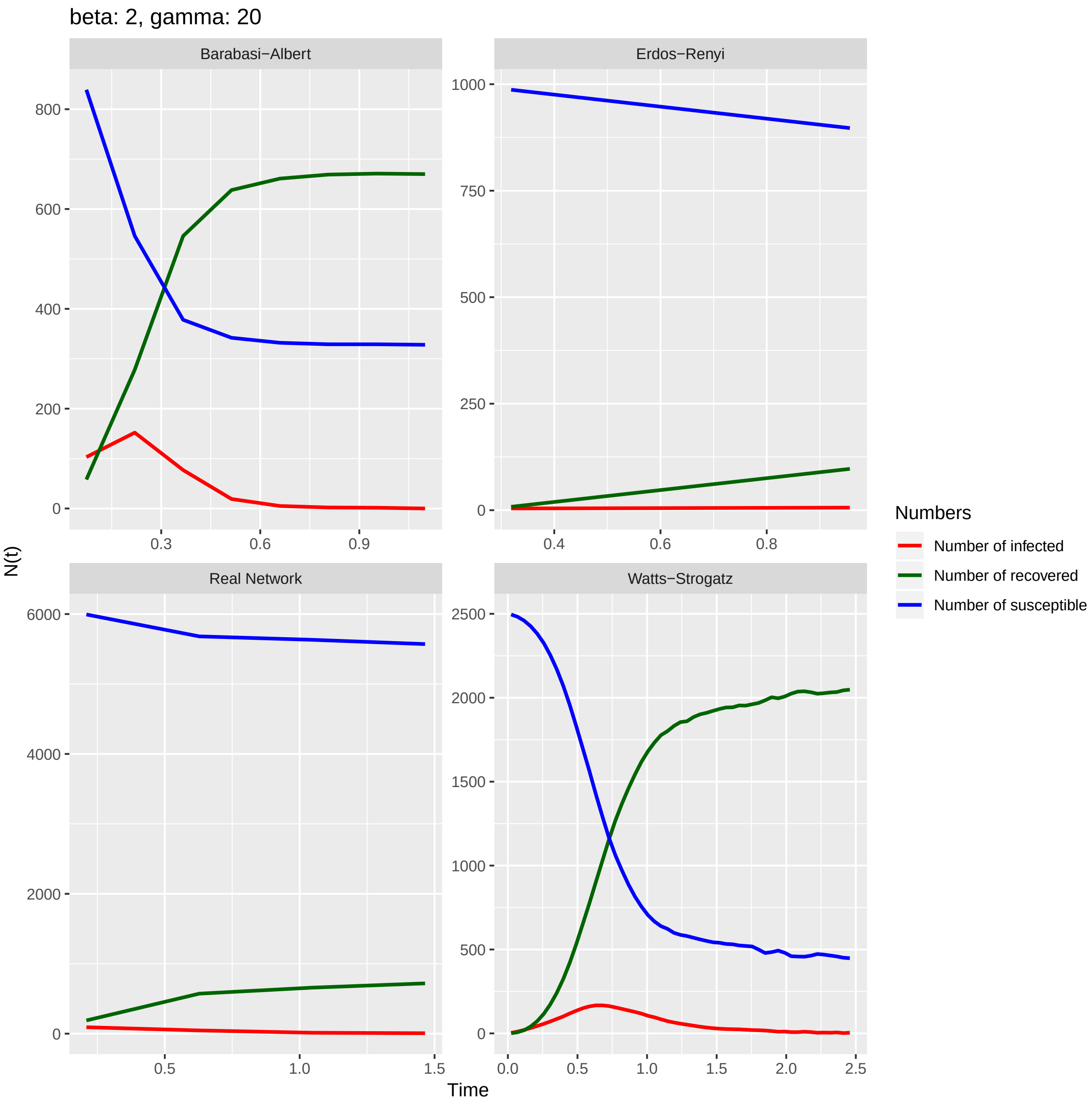
Graph	Numbers	min	max
Barabasi-Albert	NI	0	849
Barabasi-Albert	NR	5	1000
Barabasi-Albert	NS	0	682
Erdos-Renyi	NI	0	786
Erdos-Renyi	NR	0	1000
Erdos-Renyi	NS	0	984

Graph	Numbers	min	max
Real Network	NI	0	3836
Real Network	NR	1	5828
Real Network	NS	473	6247
Watts-Strogatz	NI	0	2144
Watts-Strogatz	NR	1	2500
Watts-Strogatz	NS	0	2458

Since we have not seen the infection not becoming epidemic except possibly real network in the 1st case, let's try one additional set of parameters.

4th case: $\beta = 2$, $\gamma = 20$

```
sim4 <- plotFunc(gl = graphList, beta = 2, gamma = 20, niter = 2000)
sim4$plot
```



The infection has become epidemic in BA and WS graphs with lots of nodes remaining unaffected. In ER and real network graph there is no peak of infection, it just slowly spreads through graph without becoming epidemic but not disappearing. The timeline of modelling is the shortest across all simulations. The WA modelling lines experience strong oscillations.

```r
knitr::kable(sim4$values %>%
               group_by(Graph, Numbers) %>%
               summarise(min = min(`N(t)`), max = max(`N(t)`)))
```

Graph	Numbers	min	max
Barabasi-Albert	NI	0	152
Barabasi-Albert	NR	58	671
Barabasi-Albert	NS	328	839
Erdos-Renyi	NI	4	6

Graph	Numbers	min	max
Erdos-Renyi	NR	8	97
Erdos-Renyi	NS	897	987
Real Network	NI	6	92
Real Network	NR	189	719
Real Network	NS	5571	5994
Watts-Strogatz	NI	2	167
Watts-Strogatz	NR	1	2048
Watts-Strogatz	NS	448	2496

In conclusion, it can be said that the modelling time (length of timeline) and quality (number of points) strongly depends on input parameters sometimes complicating inferences. These particular Barabasi-Albert and Watts-Strogatz graphs are the most vulnerable to the infection, the Erdos-Renyi one is less, and the real network is the least vulnerable one. Two latter stay below epidemic threshold in the 4th case where β is 10 times less than γ.