Some Network Basics with R

Nick Switanek

Northwestern University

Chicago R User Group

2012.10.03

Nick Switanek

Visiting Assistant Professor at Northwestern U

Kellogg School of Management

NU Institute for Complex Systems

Evolution & flow of information over networks

1. Science of science
2. Financial market microstructure

Dots and Lines

Vertices

Substrate, receptacle, something w memory

Edges

Conduit, pathway, pipe, tie, relationship

Vertices	Edges
Science	
Scholars	Collaboration
Scholarly teams	Apprenticeship
Departments	Competition
Articles	Similarity
NSF, funding agencies	Financial resources
Finance	
Traders	Communication
Securities, Companies	Proximity
News	Trades (Long/Short)

Raw Data

```
> IM_raw <- read.csv('IMs.txt',
                     header = FALSE,
                     stringsAsFactors = FALSE)
```

465181,77622D63-C579-4685-884F-30AD47404A64,trader,{05b1e80a-c9d5-4053-b073-aae299fbfcc3},trader,2007-01-05 17:04:57,see what i'm dealing with,TRITON,1,0,0,2

465183,F73D0ADD-2E22-40BD-B735-7CE159D22600,trader,,trader,2007-01-05 17:05:04,dont know,TRITON,0,0,0,1

465184,F73D0ADD-2E22-40BD-B735-7CE159D22600,trader,,trader,2007-01-05 17:05:14,but it looks fine,TRITON,0,0,0,1

465185,77622D63-C579-4685-884F-30AD47404A64,trader,,trader,2007-01-05 17:05:21,ugh,TRITON,0,0,0,1

465187,77622D63-C579-4685-884F-30AD47404A64,trader,,trader,2007-01-05 17:05:27,go to japonais,TRITON,0,0,0,1

465225,5694D93B-F0F8-4462-9191-A1A3E5A54C72,trader,,trader,2007-01-05 17:06:13,quite messy,TRITON,0,0,0,1

465226,566F4512-A372-42F7-8BC8-AF63F227A511,trader,,trader,2007-01-05 17:06:13,but,TRITON,0,0,0,1

465227,77622D63-C579-4685-884F-30AD47404A64,trader,,trader,2007-01-05 17:06:14,too packed,TRITON,0,0,0,1

…

“Map” Step

```
> IM <- IM_raw[  , c(7, 8, 4, 6)]
```

2007-01-05 17:11:16,**bio tech**,trader,trader
2007-01-05 17:11:17,my im jsut started giving me an away message ,trader,trader
2007-01-05 17:11:20,**im holding a core,**trader,trader
2007-01-05 17:11:21,**swing**,trader,trader
2007-01-05 17:11:22,how do i shut it off ,trader,trader
2007-01-05 17:11:25,**great**,trader,trader
2007-01-05 17:11:34,**could be some drug or something**,trader,trader
2007-01-05 17:11:38,no clue,trader,trader
2007-01-05 17:11:40,**10 points easy**,trader,trader
2007-01-05 17:11:45,i dont know where mine came from either,trader,trader
2007-01-05 17:11:48,**awesome**,trader,trader
2007-01-05 17:11:00,**somethings going on**,trader,trader
2007-01-05 17:11:02,**with it**,trader,trader
2007-01-05 17:11:13,**double bottom now monthlys**,trader,trader
2007-01-05 17:11:13,**could be**,trader,trader
…

Vertices

Traders

```
> traders <- data.frame(trader =
                        sort(unique(c(IM$sender,
                                      IM$receiver))))
> traders$traderLabels <- as.character(traders$trader)
```

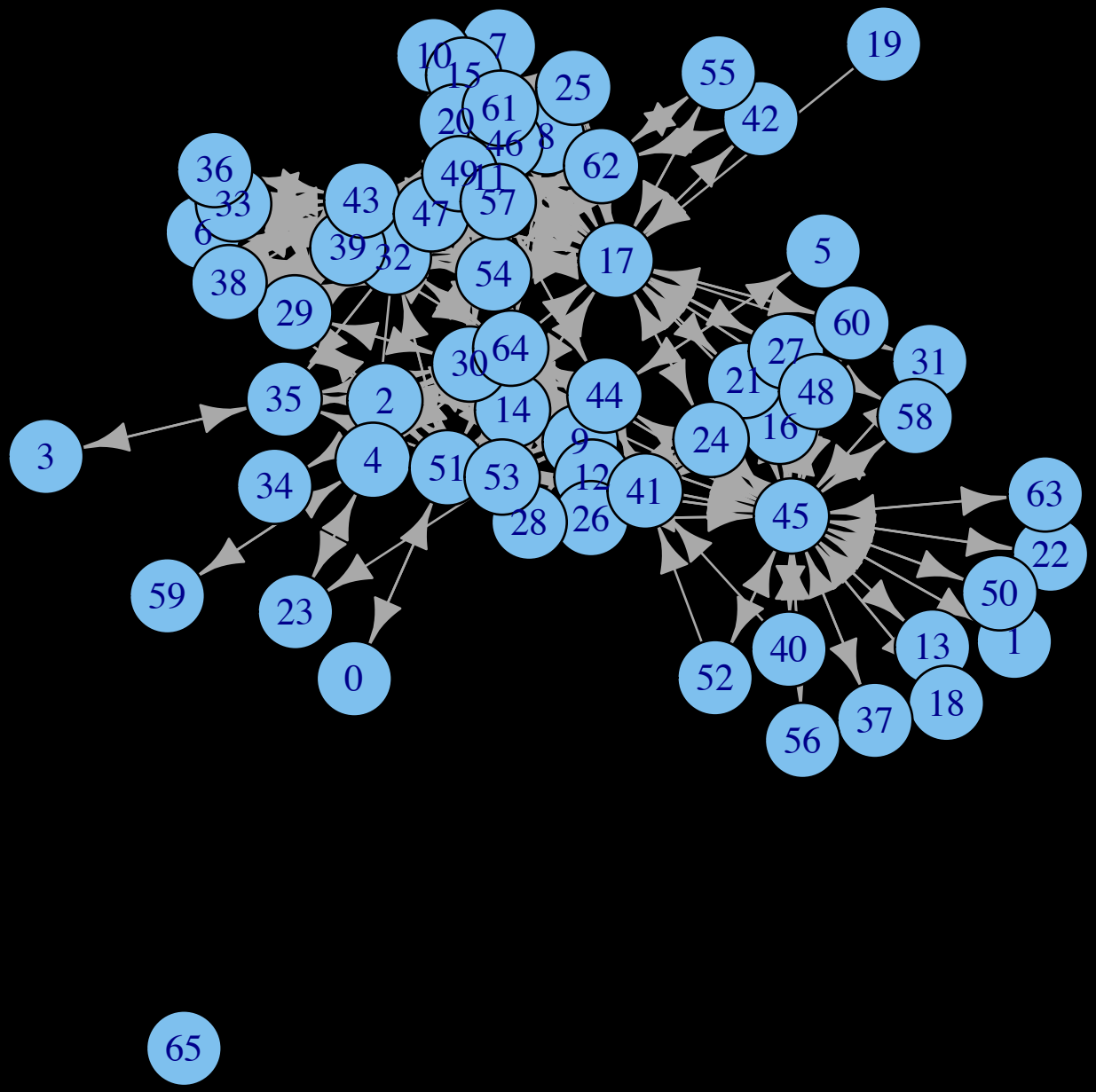
Edges

Sends instant message

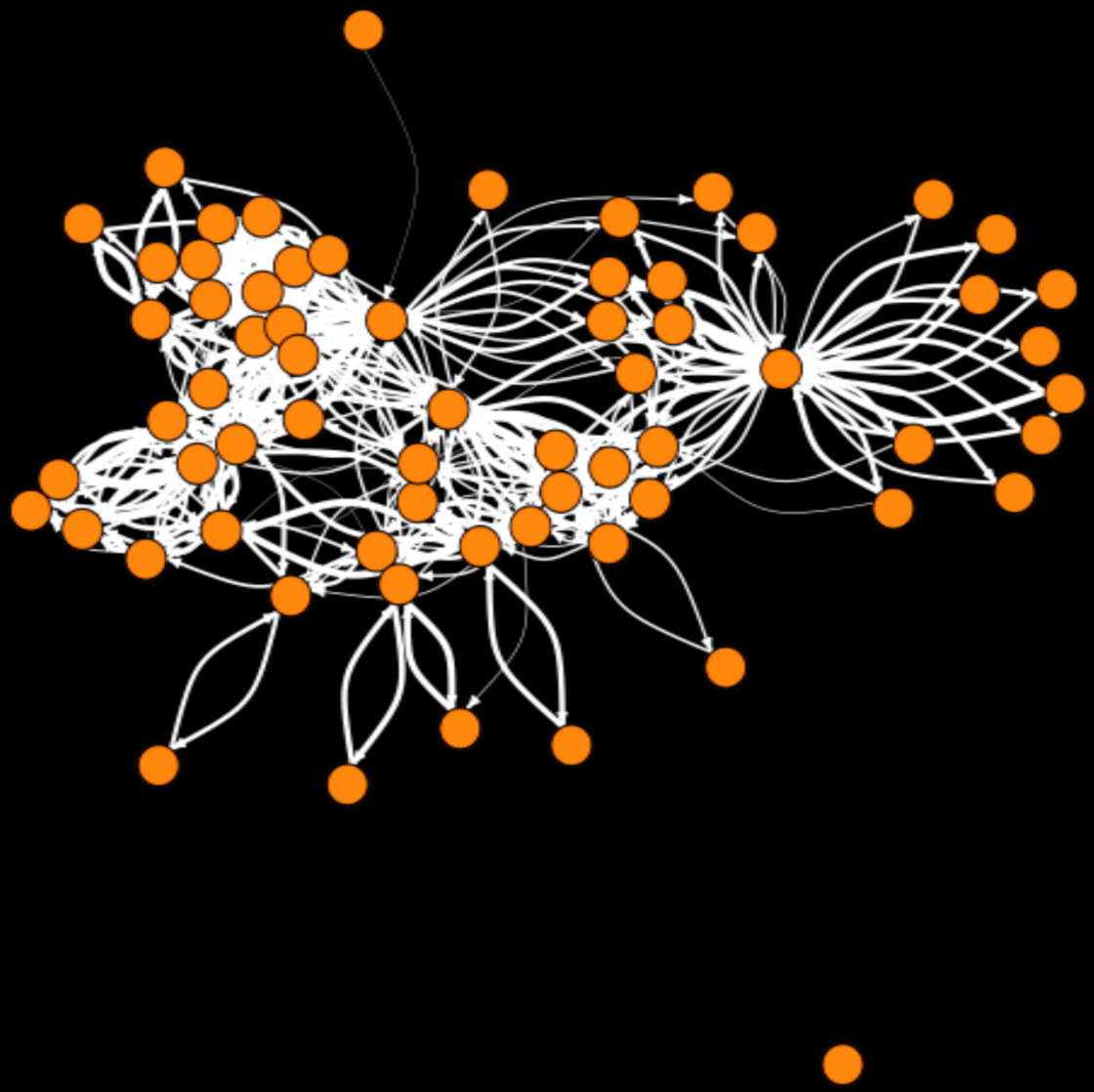
```
> library(plyr)
> relations <- ddply(IM, .(sender, receiver), function(df)
                     summarize(df,
                               nTexts = length(imText),
                               nChar = sum(nchar(imText))
                               ),
                     .progress = 'text’)
```

Data into igraph

```
> library(igraph)
> g <- graph.data.frame(relations,
                        vertices = traders,
                        directed = TRUE)
> # naïve first visualization
> glFR <- layout.fruchterman.reingold(g)
> plot(g, layout = glFR)
```


19
10
7
15
25
55
61
42
20
46
8
62
36
49
11
43
57
33
47
6
39
32
54
17
5
38
29
60
64
27
31
30
21
48
35
2
44
58
14
24
16
3
9
4
51
53
12
41
63
34
28
26
45
22
59
23
50
40
13
1
0
52
18
37
56
65

```
> plot(g, layout = glFR,
      # vertex formatting
      vertex.color = 'chocolate1',
       vertex.size = 8,
       vertex.label = “”,
      #as.character(V(g)$traderLabels),
      # edge formatting
      edge.width = .8 * log10(E(g)$nTexts),
      edge.curved = TRUE,
      # arrow formatting
      edge.arrow.size = .3,
      edge.color = 'white'
  )
```



Vertices

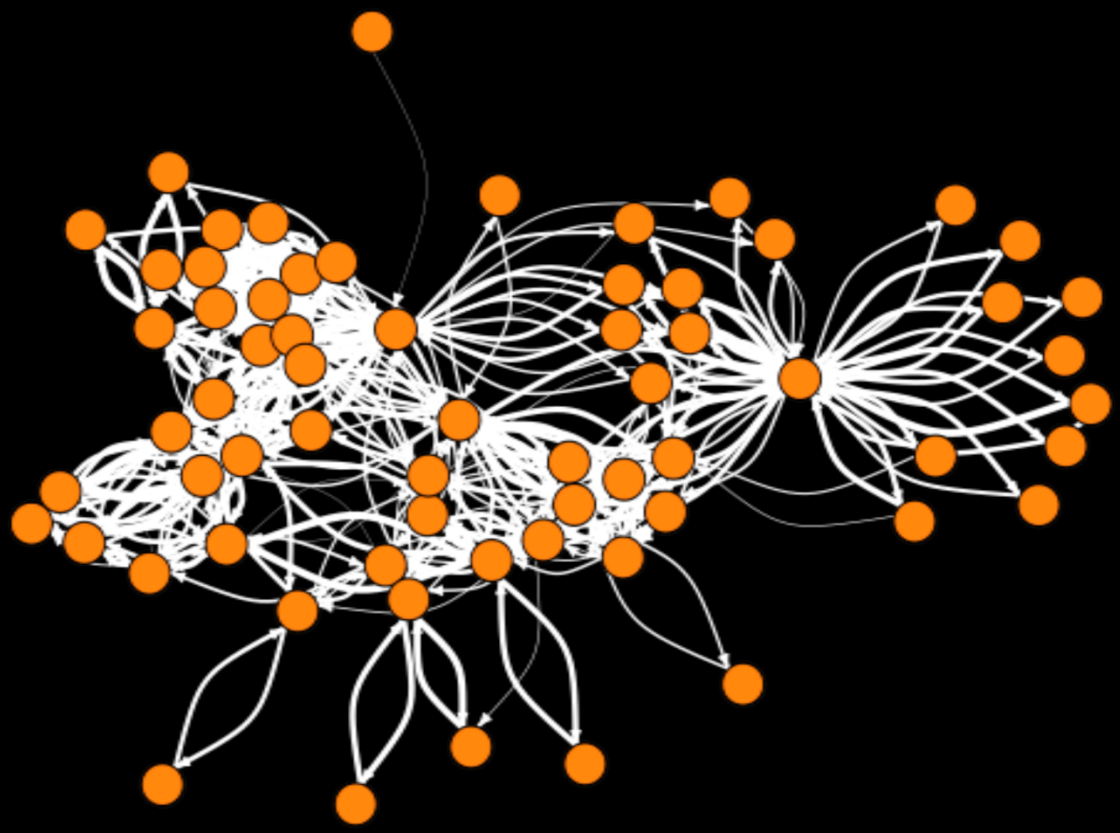
Traders

Edges

<u>Sends instant message</u>

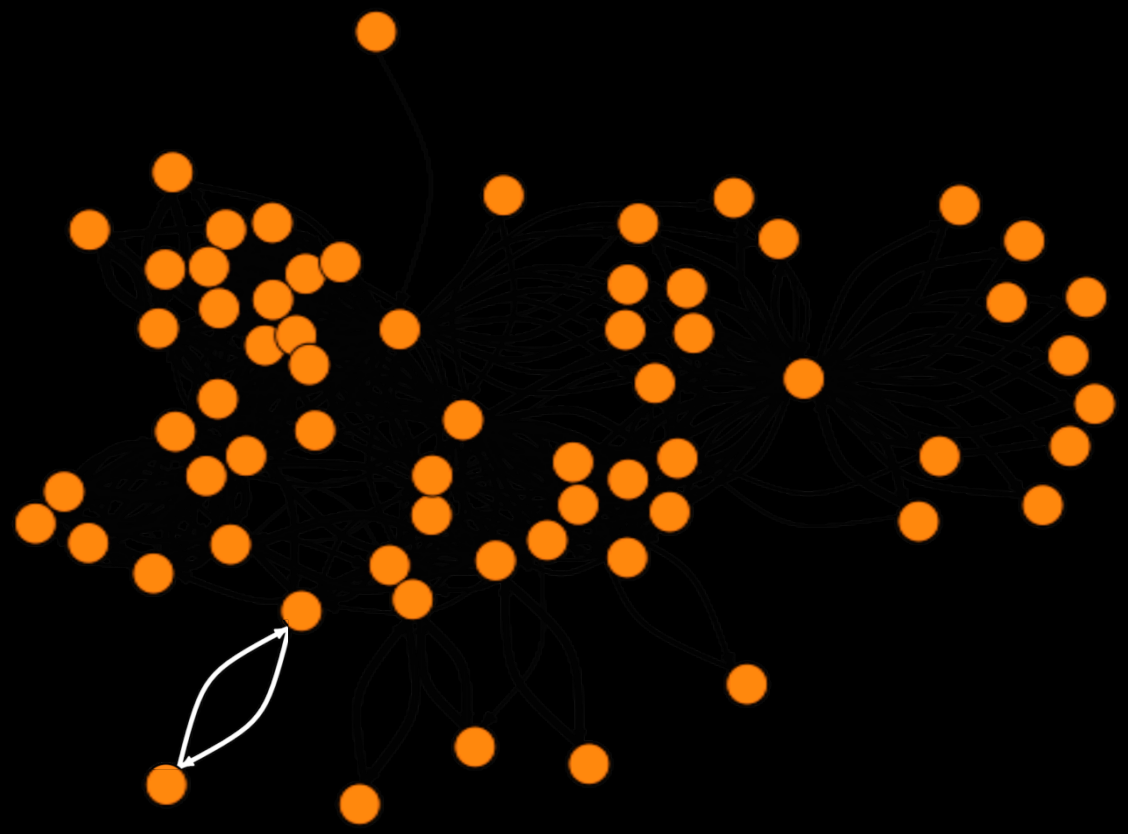
But when?

(Implicitly) **At <u>any</u> time**



Time Span: 2 years

Giant component > 98% of vertices



Time Span: 2 seconds

NO Giant component

Aggregation over Time

Concurrency

2007-01-05 17:11:16 bio tech,trader,trader

2007-01-05 17:11:17 my im jsut started giving me an away message ,trader,trader

2007-01-05 17:11:20 im holding a core,trader,trader

2007-01-05 17:11:21 swing,trader,trader

2007-01-05 17:11:22 how do i shut it off ,trader,trader

Create Time Resolution Variables

```
> str(IM$datetime)
 POSIXlt[1:2311572], format: "2008-12-01 14:40:42" "2008-12-01 14:40:45" ...
> IMin$yday_15min <- factor(paste(IMin$datetime[['yday']],
                                  IMin$datetime[['hour']],
                                  floor(IMin$datetime[['min']]/15), sep='_'))
> IMin$yday_hour <- factor(paste(IMin$datetime[['yday']],
                                  IMin$datetime[['hour']], sep='_'))
> IMin$yday <- factor(IMin$datetime[['yday']])
> IMin$week <- factor(floor(IMin$datetime[['yday']]/7))
> IMin$month <- factor(IMin$datetime[['mon']])
> IMin$quarter <- factor(floor(IMin$datetime[['mon']]/3))
```

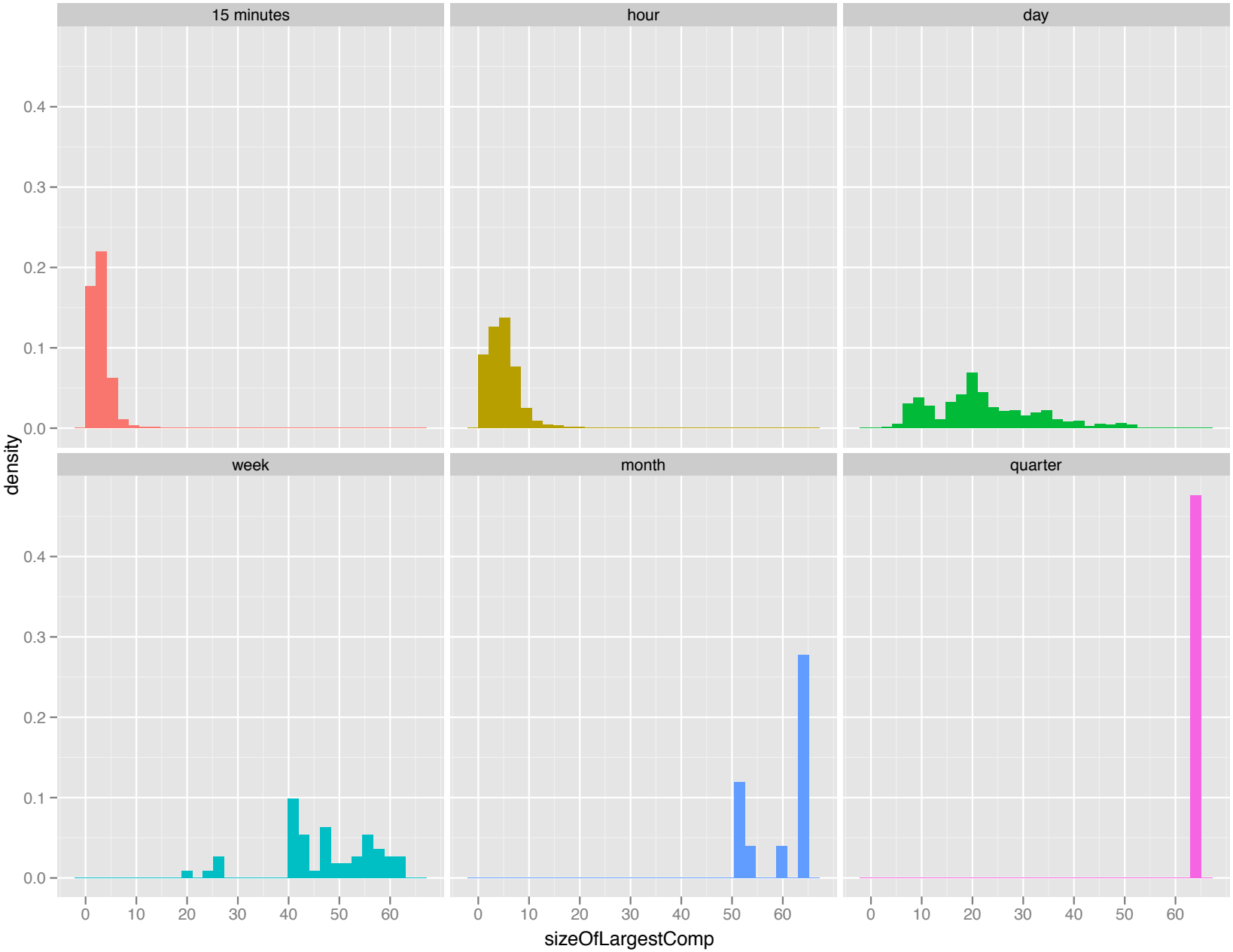
NB: The above procedure introduces selection bias…

Collect Sizes of Largest Components

```
> sizesLargestComponentByTimeWindowSize <- list()
> for (v in c('yday_15min','yday_hour','yday','week','month','quarter')) {
    sizesLargestComponentByTimeWindowSize[[v]] <- list()
    for (period in levels(IMin[[v]])) {
        # subset the IM data, collect mere existence of edges (thus the unique())
        IMsub <- unique(IMin[IMin[[v]] == period, c('sender','receiver')])
        # construct graph
        gPeriod <- graph.data.frame(IMsub)
        # get components
        comps <- decompose.graph(gPeriod, min.vertices=2)
        # get largest component
        compSizes <- unlist(lapply(comps, function(comp) unlist(comp[1])))
        largestCompIndex <- which(compSizes == max(compSizes))[1] # in case of ties
        sizeOfLargestComp <- compSizes[largestCompIndex]
        sizesLargestComponentByTimeWindowSize[[v]][[period]] <- sizeOfLargestComp
    }
}
```

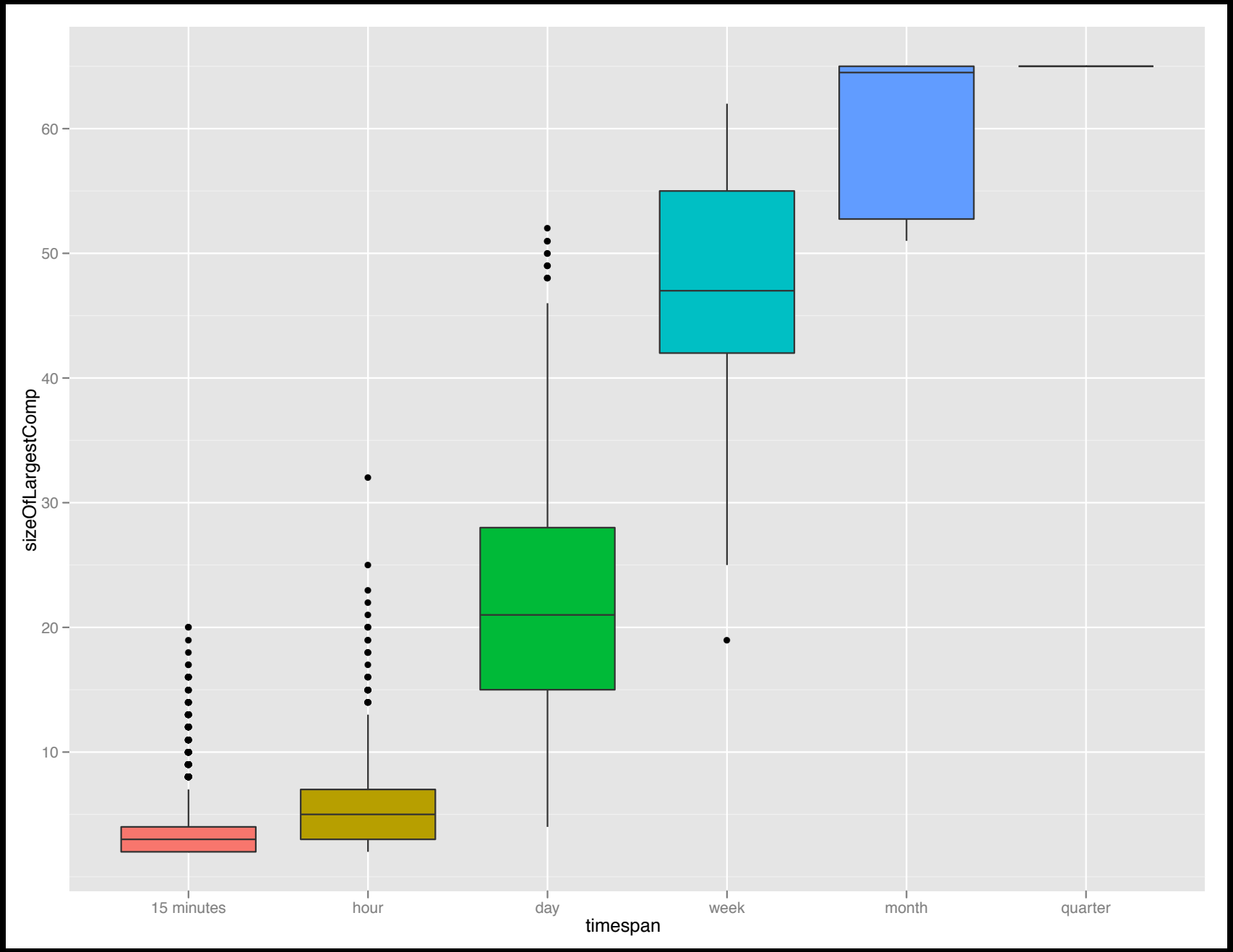
ggplot the Distributions

```
# massage into a data.frame
vars <- names(sizesLargestComponentByTimeWindowSize)
dfList <- list()
for (v in vars) {
   print(v)
   sizes <- unlist(sizesLargestComponentByTimeWindowSize[[v]])
   timeDF <- data.frame(sizeOfLargestComp = sizes,
                        timespan = v)
   dfList[[v]] <- timeDF
}

sizesOfLargestCompsDF <- rbind.fill(dfList)
levels(sizesOfLargestCompsDF$timespan) <-
   c('15 minutes', 'hour', 'day', 'week', 'month', 'quarter')

# now ggplot
gs <- ggplot(data = sizesOfLargestCompsDF,
            aes(x = sizeOfLargestComp, fill = timespan))

gs + geom_histogram(aes(y = ..density..)) + facet_wrap(~ timespan) +
   opts(legend.position = 'none’)
```


15 minutes
hour
day
week
month
quarter
density
0.0
0.1
0.2
0.3
0.4
0
10
20
30
40
50
60
sizeOfLargestComp

ggplot the boxplots

```
gs2 <- ggplot(data = sizesOfLargestCompsDF,
        aes(y = sizeOfLargestComp, x = timespan,
            fill = timespan))

gs2 + geom_boxplot() + opts(legend.position =
'none')
```


sizeOfLargestComp
60
50
40
30
20
10
15 minutes
hour
day
week
month
quarter
timespan

Thank you

nswitanek@gmail.com