STAT 341: Assignment 1

DUE: Tuesday September 26, 2022 by 11:59pm EST

NOTES

Your assignment must be submitted by the due date listed at the top of this document, and it must be submitted electronically in .pdf format via Crowdmark. This means that your responses for different questions should begin on separate pages of your .pdf file. Note that your .pdf solution file must have been generated by R Markdown. Additionally:

- For mathematical questions: your solutions must be produced by LaTeX (from within R Markdown). Neither screenshots nor scanned/photographed handwritten solutions will be accepted – these will receive zero points.
- For computational questions: R code should always be included in your solution (via code chunks in R Markdown). If code is required and you provide none, you will receive zero points.
- For interpretation questions: plain text (within R Markdown) is required. Text responses embedded as comments within code chunks will not be accepted.

Organization and comprehensibility is part of a full solution. Consequently, points will be deducted for solutions that are not organized and incomprehensible. Furthermore, if you submit your assignment to Crowdmark, but you do so incorrectly in any way (e.g., you upload your Question 2 solution in the Question 1 box), you will receive a 5% deduction (i.e., 5% of the assignment's point total will be deducted from your point total).

QUESTION 1: Basic R Calculations [10 points]

R is an excellent replacement for a graphing calculator, or even Wolfram Alpha. Solve the following expressions using R, and provide the code you used to do it.

(a) [1 point]

$$4^3$$

```
4^3
```

```
## [1] 64
```

(b) [1 point]

$$\log_9(87)$$

```
log(87, base = 9)
```

```
## [1] 2.032522
```

(c) [1 point]

$$\sum_{x=1}^{100} \frac{(-1)^{x+1}}{x}$$

```
x <- 0

for(i in 1:100){
  x <- x + ((-1)^(i+1)) / i
}

x
```

```
## [1] 0.6881722
```

(d) [1 point]

$$87 \mod 9$$

```
87 %% 9
```

```
## [1] 6
```

For the next three parts, estimate the following definite integrals with the sum of areas of very thin rectangles using `sum(width * height)` where the width is a very small `dx`. Get the `x` values for each rectangle using the `seq(..., by=dx)` function. Notes:

- Using an extremely small `dx` will use a lot of computational resources, so press escape if a calculation is taking too long and try something more moderate.
- You can check your work with the `integrate(f=..., lower=..., upper=...)` function. Your answers should be within 0.001 of the correct answers.

(e) [2 points] $sin(x)$ over $[0, \pi]$

```
dx <- 0.01
rect <- seq(from = 0, to = pi, by = dx)
height <- sin(rect)
sum <- sum(height * dx)

sum
```

```
## [1] 1.99999
```

(f) [2 points] The exponential probability density function with mean 3 (i.e., rate 1/3) over $[0, 2]$. (Hint: Consider the `dexp` function.)

```
dx <- 0.01
exp_rect <- seq(from = 0, to = 2, by = dx)
exp_height <- dexp(exp_rect, rate = 1/3)
exp_width <- rep(x = dx, times = length(exp_rect))

exp_sum <- sum(exp_width * exp_height)

exp_sum
```

```
## [1] 0.4891057
```

(g) [2 points] $x^2 + 3$, over $[-2, 2]$. (Hint: Define a `x^2 + 3` function first.)

```
dx <- 0.01
parabola <- function(x) {
  x^2 + 3
}

parabola_rect <- seq(from = -2, to = 2, by = dx)
parabola_height <- parabola(parabola_rect)

parabola_sum <- sum(parabola_height * dx)

parabola_sum
```

```
## [1] 17.4034
```

QUESTION 2: Investigating Network Analytics Attributes [12 points]

The statistical study of networks, often referred to as *network analytics*, has made significant contributions to the modeling and understanding of complex systems. A network model contains a collection of nodes and edges, where nodes represent units of interest and edges between nodes represent relationships between the units. Simply put, networks are used to model relational, or interconnected data.

We live in a highly connected world. For instance, friends may be connected on a social network; students may be connected by taking courses together; and colleagues may be connected via the emails they exchange. Thus, understanding how to handle network data is becoming an increasingly important skill. In this question you will be gently introduced to the topic of network data, and you will study two particular attributes of networks: the *average degree* and the *density*.

Consider a population $\mathcal{P}$ of N potentially connected units. Each of these units can be considered a node on a graph, and two nodes are connected by an edge if the units they represent are connected in the real world. The graph below is an example network with $N = 5$ units, some of whom are connected and some of whom are not.

Suppose that this network represents an email exchange network; nodes represent colleagues and an edge between nodes exists when those colleagues have exchanged one or more emails. Suppose further that we have the following information on the *number* of emails exchanged between each colleague.

Node Pair (u, v)	Emails Exchanged $(y_{u,v})$
(1,2)	5
(1,4)	3
(1,5)	2
(2,4)	3
(3.4)	1
(3,5)	1

This network can be described statistically with a *weight matrix* $\mathbf{W}$. Rows and columns in this matrix represent nodes, and the elements of the matrix quantify the strength of the relationship between nodes. In particular, the (u, v) element denoted by $y_{u,v} \geq 0$ quantifies the strength of the relationship between nodes u and v. For simplicity we will assume the network is *undirected*, meaning the direction of the connection between units is not recorded in which case $y_{u,v} = y_{v,u}$ and the weight matrix is symmetric. We also assume that there are no self-loops, meaning a node **cannot** have an edge to itself and so the weight matrix $\mathbf{W}$ must have zeros along the diagonal (i.e., $y_{u,u} = 0 \quad \forall u \in \mathcal{P}$). In the email exchange example above, if we define $y_{u,v}$ as the number of emails exchanged between units u and v, the weight matrix is given by:

$$\mathbf{W} = \begin{bmatrix} 0 & 5 & 0 & 3 & 2 \\ 5 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 3 & 3 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

Here we will consider the collection of node pairs to be the population of interest. In particular, the N^2 ordered pairs (u, v), $u, v \in \mathcal{P} = \{1, 2, \ldots, N\}$ will compose the population $\mathcal{P}_{u,v}$. One way to describe this population (and hence the network) is with the *average degree* which quantifies the typical number of connections in the network. The total weight for a given node is called its *degree*: $d_u = \sum_{v \in \mathcal{P}} y_{u,v}$. This can be calculated as a row sum (or, equivalently a column sum) of the weight matrix $\mathbf{W}$. Thus the *average degree* is defined as:

$$a_1(\mathcal{P}_{u,v}) = \frac{1}{N} \sum_{u \in \mathcal{P}} d_u = \frac{1}{N} \sum_{u \in \mathcal{P}} \sum_{v \in \mathcal{P}} y_{u,v} = \frac{1}{N} \sum_{(u,v) \in \mathcal{P}_{u,v}} y_{u,v}$$

which is the sum of all weight matrix elements divided by N. Below you will investigate the location, scale, and replication properties of this attribute.

(a) [2 points] Determine whether the average degree attribute is location invariant, location equivariant, or neither.

$$a_1(\mathcal{P}_{u,v}) = \frac{1}{N}\sum_{u\in\mathcal{P}} d_u = \frac{1}{N}\sum_{u\in\mathcal{P}}\sum_{v\in\mathcal{P}} y_{u,v} = \frac{1}{N}\sum_{(u,v)\in\mathcal{P}_{u,v}} y_{u,v}$$

Suppose for some $b \in \mathbb{R}$, we have

$$a_1(y_{0,0}+b, y_{0,1}+b, \ldots, y_{N,N}+b)$$

$$= \frac{1}{N}\sum_{(u,v)\in\mathcal{P}_{u,v}} y_{u,v} + b$$

$$= \frac{1}{N}(N^2 b + \sum_{(u,v)\in\mathcal{P}_{u,v}} y_{u,v})$$

$$= Nb + \frac{1}{N}(\sum_{(u,v)\in\mathcal{P}_{u,v}} y_{u,v})$$

$$= Nb + a_1(\mathcal{P}_{u,v})$$
$$\neq a_1(\mathcal{P}_{u,v})$$
$$\neq a_1(\mathcal{P}_{u,v}) + b$$

Is neither.

(b) [2 points] Determine whether the average degree attribute is scale invariant, scale equivariant, or neither.

Suppose for some $m \in \mathbb{R}$, we have

$$a_1(my_{0,0}, my_{0,1}, \ldots, my_{N,N})$$

$$= \frac{1}{N}\sum_{(u,v)\in\mathcal{P}_{u,v}} my_{u,v}$$

$$= \frac{m}{N}\sum_{(u,v)\in\mathcal{P}_{u,v}} y_{u,v}$$

$$= ma_1(\mathcal{P}_{u,v})$$

It is scale invariant

(c) [2 points] Determine whether the average degree attribute is replication invariant, replication equivariant, or neither. Note that the replicated population is defined as k independent copies of the network. This yields a weight matrix $\mathbf{W}^k$ that is an $Nk \times Nk$ block diagonal matrix with k copies of $\mathbf{W}$ along the diagonal.

Suppose for some $k \in \mathbb{R}$, we have a weight matrix $\mathbf{W}^k$

$$a_1(W^k) = a_1(y_{0,0}, y_{0,1}, \ldots, y_{N,N}, y_{0,0}, y_{0,1}, \ldots, y_{N,N}, \ldots, y_{0,0}, y_{0,1}, \ldots, y_{N,N})$$

$$= \frac{1}{kN} \sum_{(u,v) \in \mathcal{P}^{\|}{}_{u,v}} y_{u,v}$$

$$= \frac{1}{kN} \sum_{(u,v) \in \mathcal{P}_{u,v}} k y_{u,v}$$

$$= \frac{k}{kN} \sum_{(u,v) \in \mathcal{P}_{u,v}} y_{u,v}$$

$$= a_1(\mathcal{P}_{u,v})$$

It is replication invariant

Above we encoded the edges between nodes based on the strength of their connection. However, it is also common to simply consider the presence or absence of a connection. In this case the edges are binary and the network can be described statistically with an *adjacency matrix* $\mathbf{A}$. Like the weight matrix, rows and columns in the adjacency matrix represent nodes, but here the elements of the matrix are 1's and 0's respectively representing the presence and absences of edges. In particular, the elements of the adjacency matrix can be obtained by binarizing the elements of the weight matrix; the (u, v) element of $\mathbf{A}$ is $\mathbb{I}(y_{u,v} \neq 0)$. The adjacency matrix for the example email exchange network is

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

One way to summarize such a network is via its *density*, which describes the connectivity of the network. Formally, density is defined as the total number of edges divided by the total possible number of edges:

$$a_2(\mathcal{P}_{u,v}) = \frac{1}{N(N-1)} \sum_{(u,v) \in \mathcal{P}_{u,v}} \mathbb{I}(y_{u,v} \neq 0).$$

In other words, it is the sum of all adjacency matrix elements divided by $N(N-1)$. Below you will investigate the location, scale, and replication properties of this attribute.

(d) [2 points] Determine whether the density attribute is location invariant, location equivariant, or neither.

Suppose for some $b \in \mathbb{R}$, we have

$$a_2(\mathcal{P}_{0,0} + b, P_{0,1} + b, \ldots, P_{N,N} + b)$$

$$= \frac{1}{N(N-1)} \sum_{(u,v) \in \mathcal{P}_{u,v}} b + \mathbb{I}(y_{u,v} \neq 0)$$

$$= \frac{1}{N(N-1)} (N^2 b + \sum_{(u,v) \in \mathcal{P}_{u,v}} \mathbb{I}(y_{u,v} \neq 0))$$

$$= \frac{Nb}{(N-1)} + a_2(\mathcal{P}_{u,v})$$

This is neither.

(e) [2 points] Determine whether the density attribute is scale invariant, scale equivariant, or neither.

Suppose for some $m \in \mathbb{R}$, we have

$$\begin{aligned} & a_2(m\mathcal{P}_{0,0}, mP_{0,1}, \ldots, mP_{N,N}) \\ &= \frac{1}{N(N-1)} \sum_{(u,v)\in\mathcal{P}_{u,v}} m\mathbb{I}(y_{u,v} \neq 0) \\ &= \frac{1}{N(N-1)}(N(N-1)m \sum_{(u,v)\in\mathcal{P}_{u,v}} \mathbb{I}(y_{u,v} \neq 0)) \end{aligned}$$

since only $N(N-1)$ edges exist

$$= ma_2(\mathcal{P}_{u,v})$$

This is scale equivariant

(f) [2 points] Determine whether the density attribute is replication invariant, replication equivariant, or neither. Note that the replicated population is defined as k independent copies of the network. This yields an adjacency matrix $\mathbf{A}^k$ that is an $Nk \times Nk$ block diagonal matrix with k copies of $\mathbf{A}$ along the diagonal.

Suppose for some $k \in \mathbb{R}$, we have

$$\begin{aligned} a_2(W^k) &= a_1(y_{0,0}, y_{0,1}, \ldots, y_{N,N}, y_{0,0}, y_{0,1}, \ldots, y_{N,N}, \ldots, y_{0,0}, y_{0,1}, \ldots, y_{N,N}) \\ &= \frac{1}{kN(kN-1)} \sum_{(u,v)\in\mathcal{P}\|_{u,v}} \mathbb{I}(y_{u,v} \neq 0) \\ &= \frac{1}{kN(kN-1)}(\sum_{(u,v)\in\mathcal{P}_{u,v}} k\mathbb{I}(y_{u,v} \neq 0)) \\ &= \frac{1}{N(kN-1)}(\sum_{(u,v)\in\mathcal{P}_{u,v}} \mathbb{I}(y_{u,v} \neq 0)) \end{aligned}$$

This is neither

QUESTION 3: Write a Raincloud Plot function [15 points]

Visualizing distributions of numeric data is commonplace in real-world data analysis. Many options exist for this (e.g., histograms, density plots, boxplots, individual values plots), each emphasizing slightly different aspects of a distribution. In 2019, so called raincloud plots were proposed as a unified visualization that combines several of these other graphics into a single, aesthetically pleasing visualization that conveys "maximal statistical information". Specifically, this plot overlays the density plot, the boxplot, and a plot of the individual data values in the same figure. The resulting effect resembles a cloud that is raining. An example raincloud is shown below.

(a) [13 points] In this question you will make a function called `raincloud_plot()` that can take in any vector `y` of numeric data and make a raincloud plot out of it. This function should also take the following three inputs `xlabel`, `title`, and `colour` to allow the user to customize the plot. The resulting raincloud plot should resemble the example above. In particular, your plot should have:

- A density curve that is shaded underneath. You will find the function `density()` useful for the density curve and the function `polygon()` useful for shading beneath it. [4 points]
- A horizontal boxplot that lies at the base of the density curve and whose whiskers extend to the smallest and largest data values. You will find the function `boxplot()` and the input `add = TRUE` useful here. [3 points]
- A plot of the individual values scattered beneath the density curve and boxplot. The opacity of these points' colour should be such that overlapping points are coloured darker. You will find the function `jitter()` useful for scattering the points and the function `adjustcolour()` useful for adjusting opacity. [3 points]
- All of these elements must be the same colour, which must be customizable through the input `colour`. [1 point]
- Customizable x-axis label and plot title through the inputs `xlabel` and `title`. [2 points]

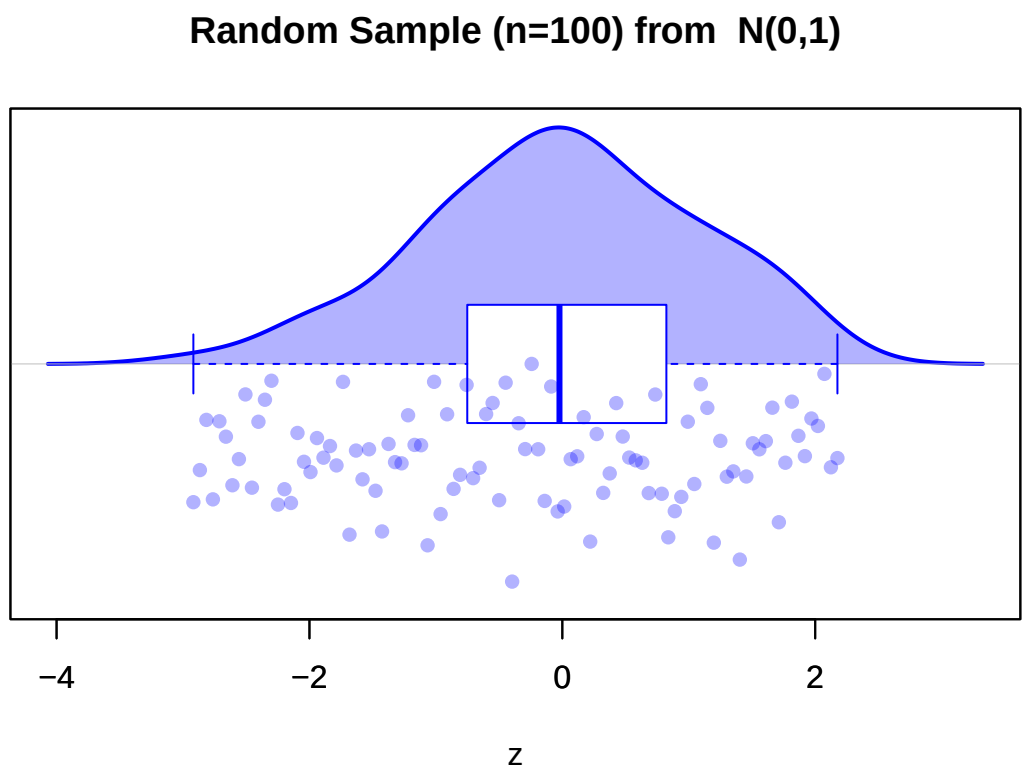
Note: You must create the graph using functions available in `base` R (all that you need has been laid out above). You may not, for example, use existing raincloud functions (or any functions) available in external packages or that exist on the web.

```
raincloud_plot <- function(data, xlabel, title, colour) {
    val <- density(data)

    plot(val, yaxt="n", ylab="", col=colour, main=title,
         xlab=xlabel,
         ylim=c(min(val$y) - max(val$y), max(val$y)), lwd=2)

    polygon(val, density=NA, col=adjustcolor(colour, alpha.f = 0.3), border=FALSE)

    bp <- boxplot(x = data, add=TRUE, horizontal=TRUE, at=0.0,
                  boxwex=max(val$y), col="white", border=colour)
    bp

    xvals <- seq(min(bp$stats), max(bp$stats), length.out = length(data))

    # need to push these values downwards
    jit <- jitter(data)
    yvals <- (jit-max(jit)) / (3*(max(jit) - min(jit)))
```

```
    points(x = xvals, y= yvals, col=adjustcolor(colour, alpha.f=0.3), pch=16)
}
```

(b) [2 points] In this question you will test your `raincloud_plot()` function on the following data. Note that it should produce a plot that looks similar to the one above.

```
set.seed(341)
z <- rnorm(n = 100)

raincloud_plot(z, "z", "Random Sample (n=100) from  N(0,1)", colour="blue")
```



QUESTION 4: R Analysis Question [20 points]

Enron was an American energy company headquartered in Houston, Texas that grew to prominence in the late 1990s. The company (and its value) grew exponentially in this time; Fortune magazine named it the most innovative company in the US between 1996 and 2000 and among the country's top 100 employers in 2000. However, much of its purported growth was an illusion propped up by fraud and financial scandal. When this became apparent to shareholders, the Enron stock price plummeted, ultimately bankrupting the company in December 2001.

After the Enron Corporation filed for bankruptcy, the Federal Energy Regulatory Commission investigated the company and made public a corpus of emails sent to and from employees within the organization. Since then this data has been extensively studied by researchers in many fields for many purposes. You will work with this data in this question. The version available to you contains information on emails exchanged between $N = 184$ employees between November 1998 and June 2002. This data is available in the `enron_employees.csv` and `enron_emails.csv` files which are described below.

The `enron_employees.csv` file contains the job titles for each of the $N = 184$ Enron employees. Each row corresponds to a specific employee and the columns and their contents are described below.

Column	Description
`Node`	An integer between 1 and 184 signifying an employee. Each employee is uniquely identified by these numbers.
`Job Title`	The employees' job titles. Note many titles are missing in which case an `NA` is recorded.

The `enron_emails.csv` file contains information about the emails sent between November 1998 and June 2002. Each row corresponds to a specific email and the columns and their contents are described below.

Column	Description
`Month`	The month the email was sent.
`Day`	The day the email was sent.
`Year`	The year the email was sent.
`Sender`	The employee who sent the email. Note that these numbers correspond to the `Node` numbers in `enron_employees.csv`.
`Recipient`	The employee who received the email. Note that these numbers correspond to `Node` numbers in `enron_employees.csv`.

(a) [4 points] Using R, read in the data found in `enron_emails.csv` and create two subsets called `emailsSep2001` and `emailsOct2001` containing only the rows corresponding to September 2001 and October 2001 respectively. How many emails were exchanged in each of these months?

```
emails <- read.csv(file = "enron_emails.csv", header = TRUE)
emailsSep2001 <- emails[which(emails$Month == 9 & emails$Year == 2001), ]
emailsOct2001 <- emails[which(emails$Month == 10 & emails$Year == 2001), ]

nrow(emailsSep2001)
```

```
## [1] 3762
```

```
nrow(emailsOct2001)
```

```
## [1] 10796
```

There are 3762 emails sent during September 2001 and 10796 during October 2001

(b) [2 points] Using R, obtain the weight matrix and calculate the average degree for the Enron email exchange network in September 2001 and separately in October 2001. Note that the data in `enron_emails.csv` is not in weight matrix format, it's in so called *edge list* format. To construct a weight matrix from an edge list, you may use the `getWeightMatrix()` function. To get the weight matrix for September and October 2001 you may run the lines of code below. Note that `getWeightMatrix()` is available in the appendix of this assignment.

```
getWeightMatrix <- function(edgeList, N){
  W <- matrix(data = 0, nrow = N, ncol = N)
  for(i in 1:(N-1)){
    for(j in (i+1):N){
      W[i,j] <- nrow(edgeList[(edgeList$Sender==i & edgeList$Recipient == j),]) +
                nrow(edgeList[(edgeList$Sender==j & edgeList$Recipient == i),])
    }
  }
  W <- W + t(W)
  return(W)
}

septWeightMatrix <- getWeightMatrix(edgeList = emailsSep2001, N = 184)
octWeightMatrix <- getWeightMatrix(edgeList = emailsOct2001, N = 184)

septAvg <- sum(septWeightMatrix) / 184
octAvg <- sum(octWeightMatrix) / 184

septAvg
```

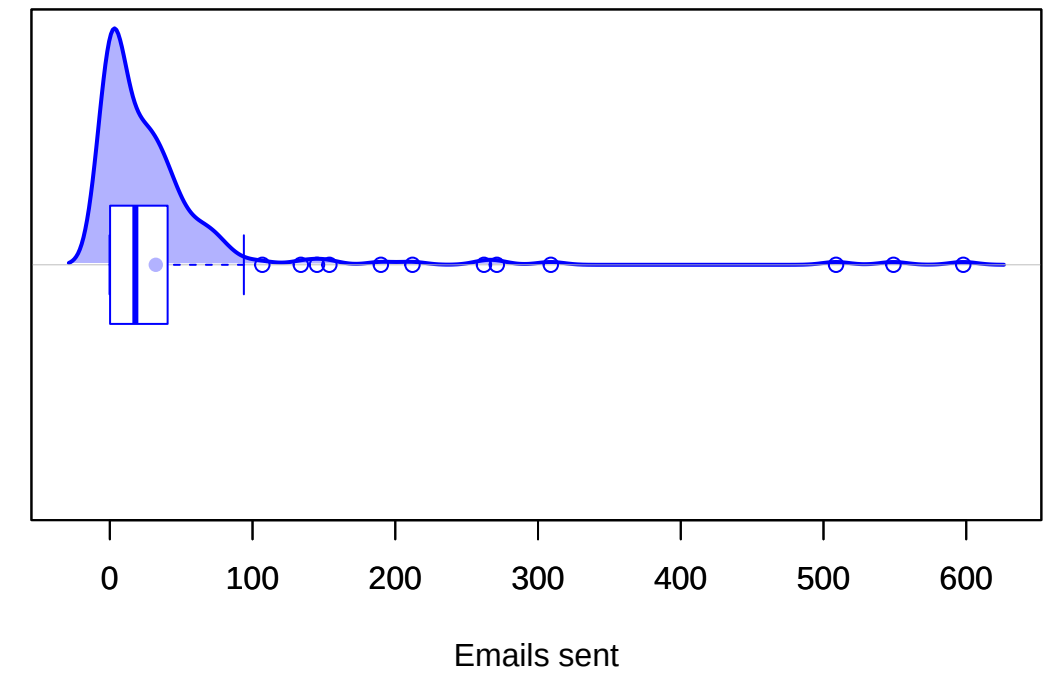
```
## [1] 38.20652
```

```
octAvg
```

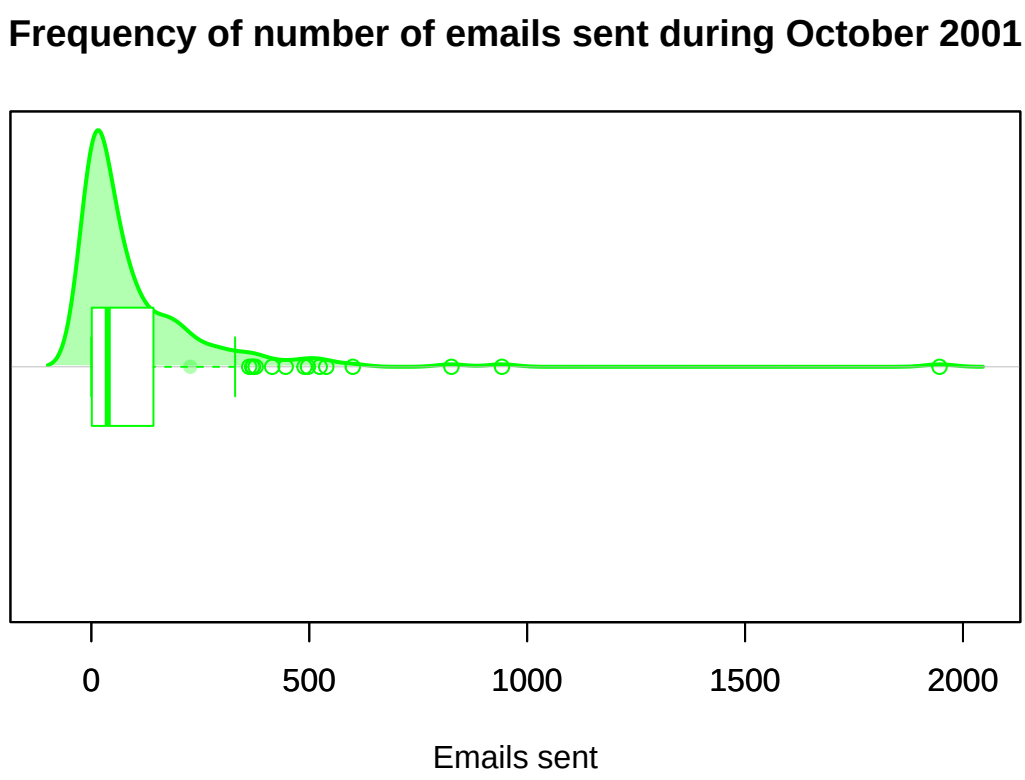
```
## [1] 109.2826
```

(c) [4 points] Using the `raincloud_plot()` function you created in Question 3, construct two raincloud plots that visualize the degree distribution for September 2001 and October 2001. Use different colours for each plot and be sure to label the axes and titles informatively.

```
raincloud_plot(jitter(rowSums(septWeightMatrix)), "Emails sent",
               "Frequency of number of emails sent during September 2001",
               colour="blue")
```




```
raincloud_plot(jitter(rowSums(octWeightMatrix)), "Emails sent",
               "Frequency of number of emails sent during October 2001",
               colour="green")
```



(d) [4 points] Using R, obtain the adjacency matrix and calculate the density of the Enron email exchange network in September 2001 and separately in October 2001.

```
adjMatrix <- function(matrix, N) {
  copy <- matrix
  copy[copy != 0] <- 1
  copy
}

sept_adj <- adjMatrix(septWeightMatrix, 184)

sum(sept_adj) / (184 * 183)
```

```
## [1] 0.02280827
```

```
oct_adj <- adjMatrix(octWeightMatrix, 184)

sum(oct_adj) / (184 * 183)
```

```
## [1] 0.03444999
```

(e) [2 points] Based on your results from parts (a)-(d), comment on the change in email behaviour between September and October 2001. Do some light reading on the Enron scandal (Wikipedia is a good place to start) and identify an event in the downfall of the company that explains these changes.

The change in the number of emails sent between the month of September to October can be seen by the increased frequency of emails sent, where it was rare to have more than 100 emails sent in September to the same number being a low number of emails sent in October. Moreover, the average emails jumped from 39 to 110. This is due to the fact that Enron was being audited at the time and they switched CEO, CFO and COO in the same amount of time frame. Because Enron was being audited, the emails were likely from executives about how they can cover up this case from auditors.

(f) [4 points] Let d_u denote the degree of node u in the network (i.e., the number of emails exchanged by employee u), and let $a(\mathcal{P}) = \overline{d}$ (average degree) be the attribute of interest. Define the influence of employee u on $a(\mathcal{P})$ to be:

$$\Delta(a, u) = |a(d_1, \ldots, d_{u-1}, d_u, d_{u+1}, \ldots, d_N) - a(d_1, \ldots, d_{u-1}, d_{u+1}, \ldots, d_N)| .$$

Construct an influence plot of Δ vs. employee number and identify the employee with the largest influence on the average degree attribute in October 2001. Using the information in `enron_employees.csv`, determine the job title of this employee and conjecture why they have such a large influence.

```
employees <- read.csv(file = "enron_employees.csv", header = TRUE)

octAvg
```

```
## [1] 109.2826
```

```
influence <- rep(0, nrow(emailsOct2001))
for(i in 1: nrow(emailsOct2001)){
  exceptCurr <- sum(octWeightMatrix[-i, -i]) / 183
  influence[i] <- abs(octAvg - exceptCurr)
}

plot(influence, main="Influence of employee",
     xlab="Employee Number", ylab="Influence",
     col= adjustcolor("blue", alpha.f =0.3), xlim=c(0, 184))
```


```
mostInfluence <- which(influence == max(influence))

employees$Job.Title[which(employees$Node == mostInfluence)]
```

```
## [1] "In House Lawyer"
```

The person with the most email influence is the in house lawyer, likely because they are the one covering up any incriminated evidence and they represent Enron at court hearings.

Appendix

```
getWeightMatrix <- function(edgeList, N){
  W <- matrix(data = 0, nrow = N, ncol = N)
  for(i in 1:(N-1)){
    for(j in (i+1):N){
      W[i,j] <- nrow(edgeList[(edgeList$Sender==i & edgeList$Recipient == j),]) +
                nrow(edgeList[(edgeList$Sender==j & edgeList$Recipient == i),])
    }
```

```
  }
  W <- W + t(W)
  return(W)
}
```