Class test II (1 December 2016)

Please use the exam account that has already been logged in. Do not log into Windows using your own credentials. During the test you are not allowed to talk to or otherwise communicate with other students (email, instant messaging, etc.) or access the internet / course material.

At the end of the test, please log on to Moodle and upload your script by clicking on the link *Upload class test 2* on Moodle.

Please make sure you regularly save your R script, just in case RStudio crashes.

You have to attempt all questions.

1. Your Desktop contains two data files which you will use in this question.

   The file `stations.csv` contains the list of bike stations of the Bay Area Bike Share system in the San Francisco Bay Area. It has the following columns.

`id`	Numeric identifer of the station
`name`	Name of the station
`lat`	Latitude of the station
`long`	Longitude of the station
`docks`	Number of docks at the station
`city`	City in which the station is located

   The file `trips.csv` contains all trips made at the end of August 2013. It has the following columns.

`start_id`	Numeric identifer of the station where the trip started
`end_id`	Numeric identifier of the station where the trip ended
`start_date`	Date and time the trip started
`end_date`	Date and time the trip ended
`subscription_type`	User type ("Subscriber" or "Customer")

   Give the R code required to answer the questions below.

   (a) Read both data files into R and store them in a data frame each. [4 marks]

   (b) How many docks are there in the entire system? How many docks are there om the city of Mountain View? [2 marks]

   (c) What proportion of trips were made by subscribers and what proportion of trips were made by customers? [2 marks]

   (d) Find the station name of the station with the fewest docks. [2 marks]

   (e) Create a plot of the GPS coordinates of the stations in San Francisco. The label of the x-axis should be "Longitude" and the label of the y-axis should be "Latitude". The title of the plot should be "Bicycle trips in San Francisco". [2 marks]

   (f) Create a so-called origin-destination matrix. The matrix should be 70×70 and the (i, j)-th entry should contain the number of trips made from station i to station j. [4 marks]

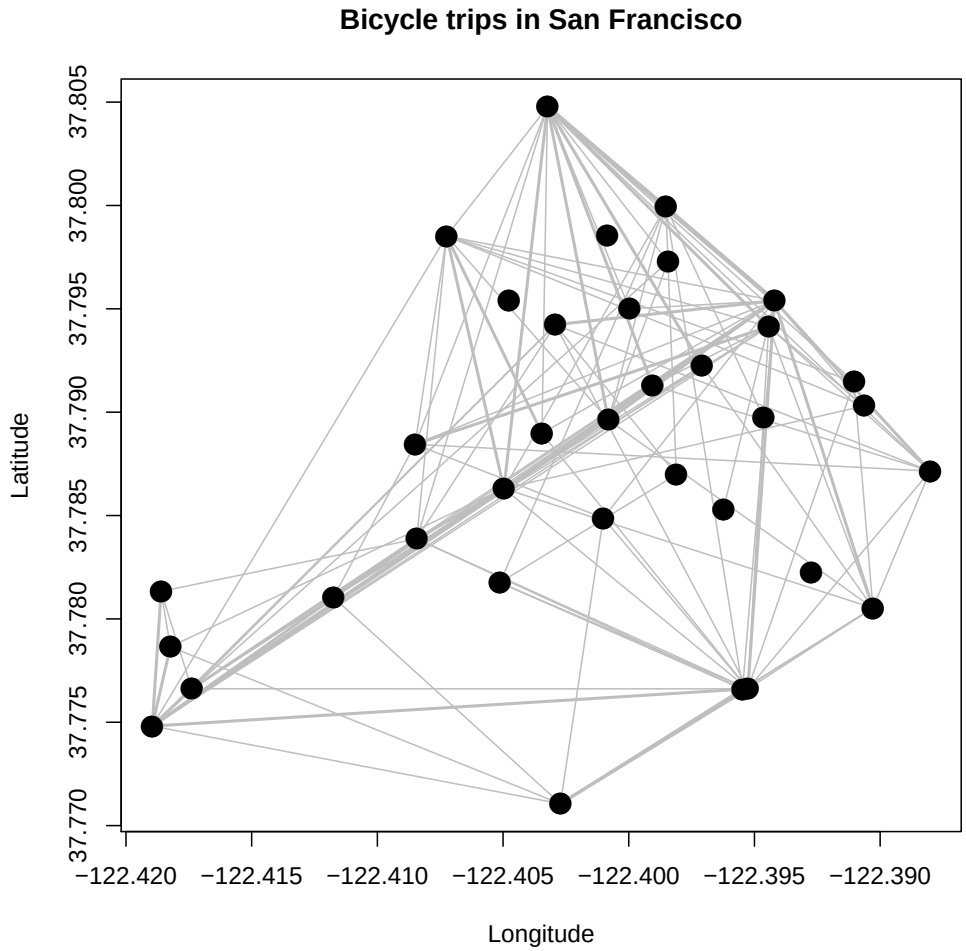
   (g) Add lines to your plot representing the number of trips between the stations in San Francisco. Do not show lines corresponding to trips involving stations outside San Francisco.
   If there are less than 5 trips between i and j (either way), do not show a line on your plot.
   If there are between 5 and 9 trips between i and j (either way), draw a line of width 1.
   If there are between 10 and 19 trips between i and j (either way), draw a line of width 2.
   If there are more than 20 trips between i and j (either way), draw a line of width 3.

   Your plot should look similar to the plot shown overleaf.

   If your code from part (f) is not working you can create a fake origin-destination matrix using the code below.

   ```
   od <- matrix(rpois(4900, 5), ncol=70)
   ```

   [4 marks]



2. (a) The number of ways in which we can draw n objects from a population of N objects is given by $\frac{N!}{(N-n)!}$, assuming that we are retaining the order. For large N this formula suffers from numerical overflow.

   Write a function `approx.permut`, which takes `N` and `n` as arguments and which returns the approximation

$$\sqrt{\frac{N}{N-n}} \cdot \left(\frac{N}{N-n}\right)^N \cdot (N-n)^n \cdot \exp(-n).$$

   [5 marks]

   (b) Alternatively we can calculate $\frac{N!}{(N-n)!}$ by rewriting it as

$$N \cdot (N-1) \cdots (N-2) \cdot (N-n+1) = \prod_{i=0}^{n-1} (N-i) \tag{1}$$

   This formula is exact and numerically stable. Write a function `exact.permut` which takes N and n as arguments and which returns the value of $\frac{N!}{(N-n)!}$ calculated as set out in equation (1).

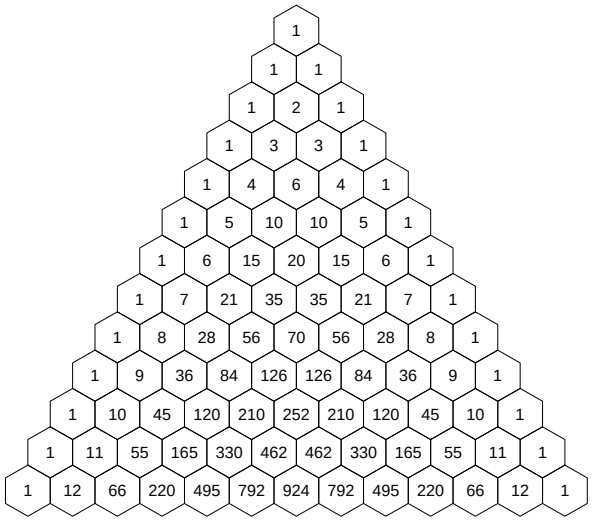
   Do not calculate the result as $N!$ divided by $(N-n)!$, as this would be numerically unstable. [5 marks]

   (c) Calculate $\frac{200!}{(200-10)!}$ using your functions `approx.permut` from part (a) and `exact.permut` from part (b).
   You can compare your results to `exp(lgamma(200+1)-lgamma(200-10+1))`. [2 marks]

3. Pascal's triangle is an illustration of Pascal's rule for binomial coefficients:

$$\binom{k}{i} = \binom{k-1}{i-1} + \binom{k-1}{i}.$$

Consider a triangle such that the rop row consists of $\binom{0}{0} = 1$ and the second row consists of $\binom{1}{0} = 1$ and $\binom{1}{1} = 1$. The k-th row consists of $\binom{k}{0}, \binom{k}{1}, \ldots, \binom{k}{k}$. The recursive formula from above just implies that each entry in the triangle is the sum of the two entries immediately above it (see below).



More formally, if $\mathbf{r}_{k-1}$ is the vector of numbers in the $(k-1)$-st row of Pascal's triangle, then the k-th row is given by

$$\mathbf{r}_k = [0, \mathbf{r}_{k-1}] + [\mathbf{r}_{k-1}, 0].$$

Write a function `pascal` which takes the desired number of rows n as argument and which returns the first n rows of Pascal's triangle computed using the recursive formula set out above. You should return a list, in which each entry corresponds to a row of the triangle. For $n = 3$ your function should return

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

[[2]]
[1] 1 1

[[3]]
[1] 1 2 1
```

Use your function to compute the first 10 rows of Pascal's triangle. [8 marks]

4. In this question you will simulate from a discrete stochastic SIR model. An SIR (susceptible – infected – recovered) model is a simple compartmental model for the spread of a non-life-threatening infection in a population of size N.

   Suppose that, at time t ($t = 0, 1, 2, \ldots, T$), the number of individuals susceptible to the infection is S_t, the number of individuals currently infected is I_t, and the number of individuals who have recovered and are now immune to the infection is R_t, with $S_t + I_t + R_t = N$.

   We now assume that the number of new infections has a binomial distribution with the probability of a susceptible individual being infected being dependent on the proportion of infected individuals, i.e.

$$\Delta S_t | S_{t-1}, I_{t-1} \sim \mathsf{Bi}\left(S_{t-1}, 1 - \exp(-\beta I_{t-1}/N)\right) \tag{2}$$

   Infected individuals are assumed to recover at any given time with a probability of α, independently of how long they have already been infected, i.e.

$$\Delta I_t | I_{t-1} \sim \mathsf{Bi}(I_{t-1}, \alpha) \tag{3}$$

   Then the updated number of susceptible, infected and recovered individuals is

$$S_t = S_{t-1} - \Delta S_t \qquad I_t = I_{t-1} + \Delta S_t - \Delta I_t \qquad R_t = R_{t-1} + \Delta I_t \tag{4}$$

   We can simulate from this model as follows. We assume that the quantities N, I_0, α, β and T are given.

   1. Set $S_0 = N - I_0$ and $R_0 = 0$.

2. For $t = 1, 2, \ldots, T$

   i. Draw ΔS_t from the binomial distribution given in (2).

   ii. Draw ΔI_t from the binomial distribution given in (3).

   iii. Update S_t, I_t and R_t as set out in (4).

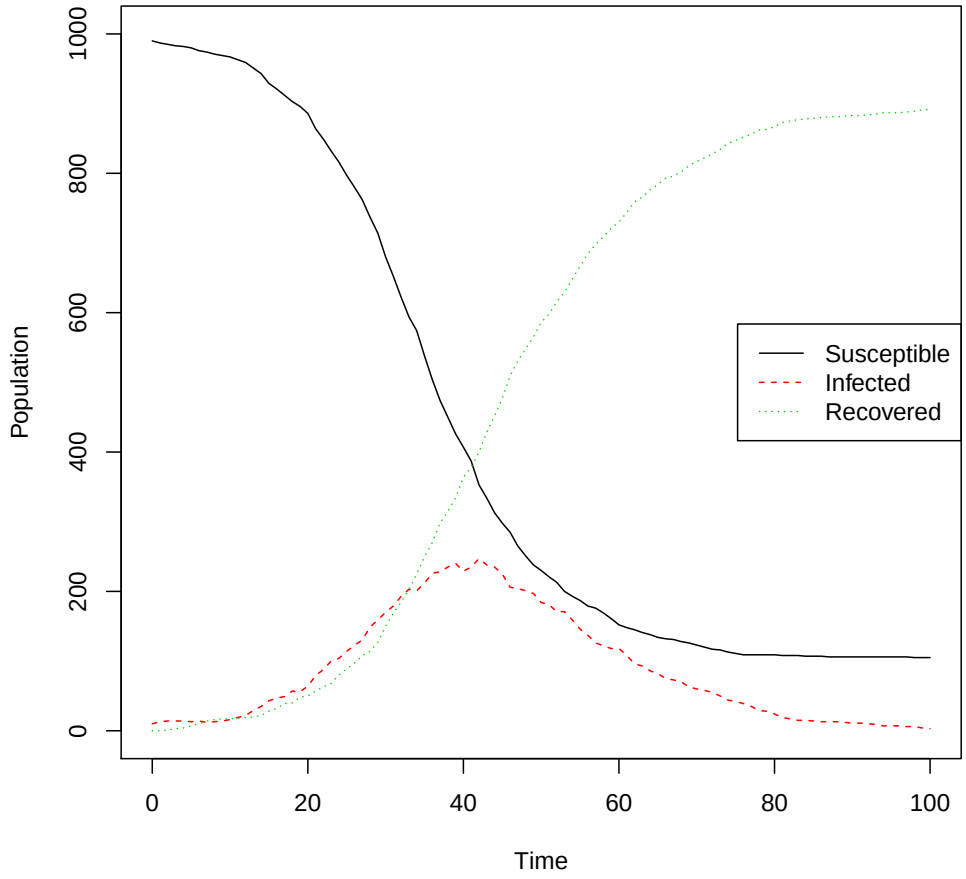
(a) Write a function `simulate.sir` which simulates from the SIR model as set out above. The function should take the parameters `N`, `initial.infected` (I_0), `alpha` (α), `beta` (β) and `T` as arguments. It should returns a data frame with $T + 1$ rows, in which the $(t + 1)$-st row contains the values of S_t, I_t and R_t.

Your function should check whether $0 \leq I_0 \leq N$, $0 < \alpha < 1$ and $\beta > 0$ and produce an error message if this is not the case.

The default values should be $N = 1000$, $I_0 = 0.01N$ (rounded to the nearest integer), $\alpha = \frac{1}{10}$, $\beta = \frac{1}{4}$ and $T = 100$.

Hint: You can draw one realisation from the Bi(n, θ) distribution using the R function `rbinom(1, n, theta)`. [10 marks]

(b) Use your function `simulate.sir` to create one realisation from the SIR model (using the above default values for the parameters) and plot the three columns as three curves in one plot as shown below. The curves should use a different colour and/or line type. Add a legend to your plot.



Please note that the SIR model is a stochastic model, so every realisation will look slightly different. [5 marks]

(c) At the end of the epidemic (which we assume is at time $T = 100$) not every individual will have had the infection. It is of epidemiological interest to determine the proportion of individuals who have had the infection, R_{100}/N.

Create 50 realisations from the above model, each created by calling the function `simulate.sir` from part (a), again using the above default values for the parameters. Calculate the mean value of R_{100}/N from the simulations. [5 marks]

If you do not get your code to work for part (a) you can use the following function instead of `simulate.sir` *to answer parts (b) and (c).*

```
fake.sir <- function(N=1000, T=100) {
  a <- N * runif(1)
  infected <- 25 * dnorm(0:T, 45, 25) * a
```

```
  susceptible <- N - pnorm(0:T, 45, 25) * a
  data.frame(susceptible=susceptible, infected=infected, recovered=N-susceptible-infected)
}
```

Total (Honours):	60
Total (MSc):	60