Homework 01 - Statistical Methods for Data Science

Andrea Gasparin, Rossella Marvulli, Victor Plesco

LAB Exercises

Exercise 1

1. Write a function Binomial(x, n, p) for the binomial distribution above, depending on parameters x, n, p and test it with some prespecified values. Use the function **choose()** for the binomial coefficient.
2. Plot two binomials with $n = 20$, and $p = \{0.3, 0.6\}$ respectively.

Solution 1.1

```
options(scipen = 999)

Binomial <- function(x, n, p)
{
  for(i in x)
  {
    x = choose(n, i) * p^i * (1 - p)^(n - i);
    cat(c(x, " "));
  };
}

Binomial(2:4, 4, 0.2);
```

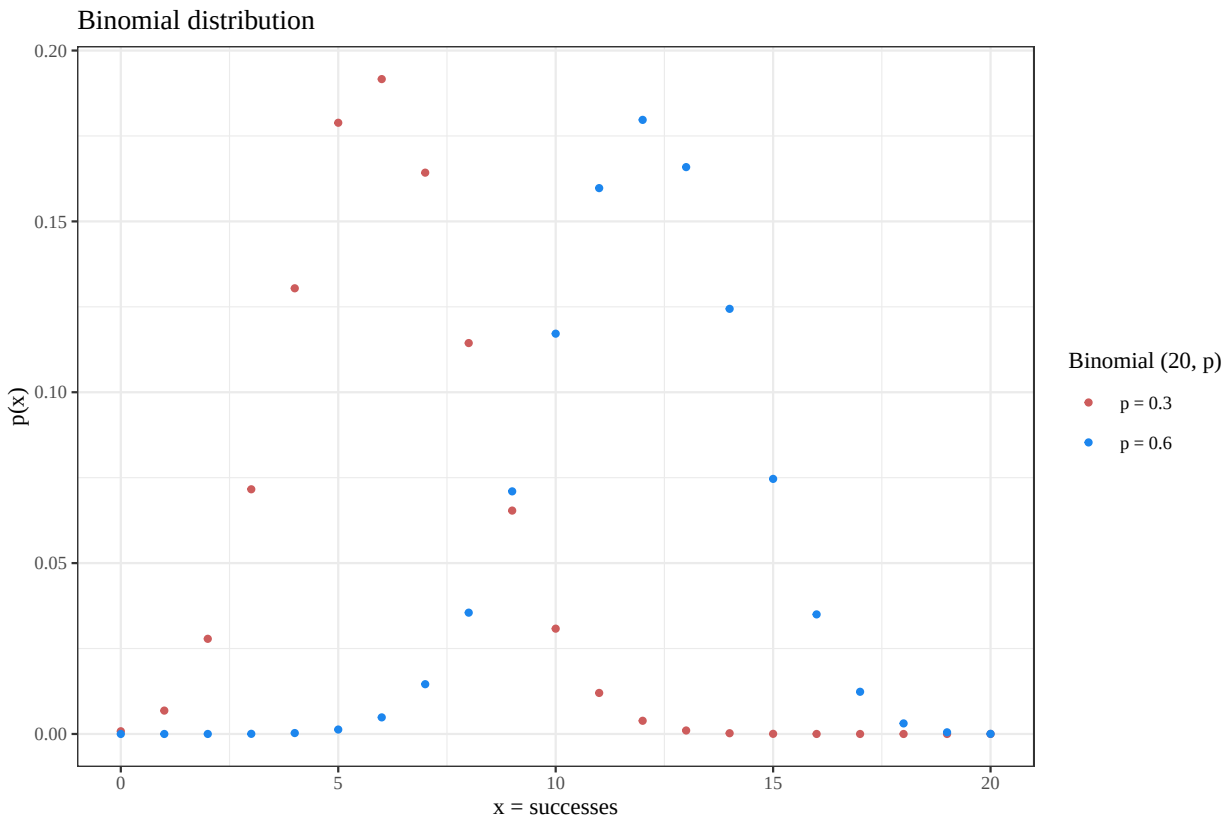
```
> 0.1536  0.0256  0.0016
```

```
Binomial(1, 10, 0.8);
```

```
> 0.00000409599999999999
```

Solution 1.2

```
require(ggplot2)
require(extrafont)

dtf_binomials <- data.frame(successes = rep(c(0:20), 2),
                            probability =
                              c(dbinom(0:20, 20, 0.3),
                                dbinom(0:20, 20, 0.6)),
                            label =
                              c(rep("p = 0.3", 21),
                                rep("p = 0.6", 21)));

ggplot(data = dtf_binomials, aes(x = successes,
                                 y = probability)) +
  geom_point(aes(color = label), size = 1) +
  labs(title = "Binomial distribution",
       x     = "x = successes",
       y     = "p(x)",
       col   = "Binomial (20, p)") +
  scale_color_manual(values = c("indianred", "dodgerblue2")) +
  theme_bw(base_size = 10, base_family = "Times")
```

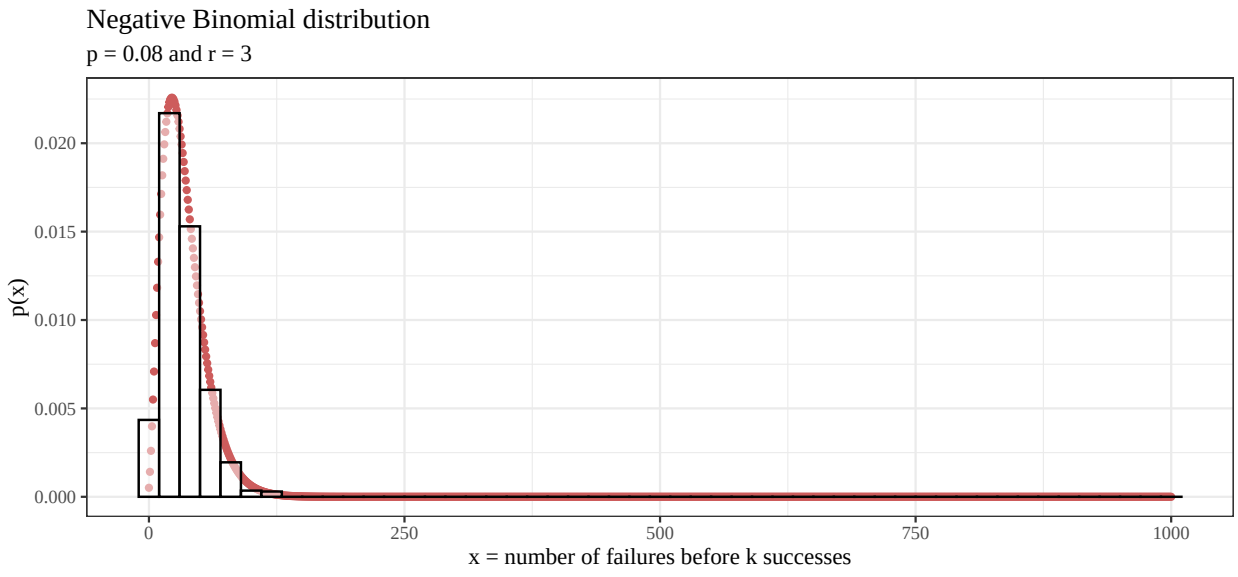


Exercise 2

1. Generate in **R** the same output, but using **rgeom()** for generating the random variables. *Hint*: generate n times three geometric distribution $X_1, \ldots, X_3$ with $p = 0.08$, store them in a matrix and compute then the sum Y.

Solution 2.1

```
require(ggplot2)
require(extrafont)

matrix_geom <- matrix(data = NA, nrow = 1000, ncol = 4)
for(i in 1:3)
{
  matrix_geom[, i] = rgeom(nrow(matrix_geom), 0.08);
};
matrix_geom[, 4] <- rowSums(matrix_geom[, 1:3])

ggplot() +
  geom_point(aes(x = c(0:1000), y = dnbinom(0:1000, 3, 0.08)),
             size = 1,
             colour = "indianred") +
  geom_histogram(aes(x = matrix_geom[, 4], y = ..density..),
                 binwidth = 20,
                 colour   = "black",
                 fill     = "white",
                 alpha    = 0.5) +
  labs(title    = "Negative Binomial distribution",
       subtitle = "p = 0.08 and r = 3",
       x        = "x = number of failures before k successes",
       y        = "p(x)") +
  theme_bw(base_size = 10, base_family = "Times")
```



Exercise 3

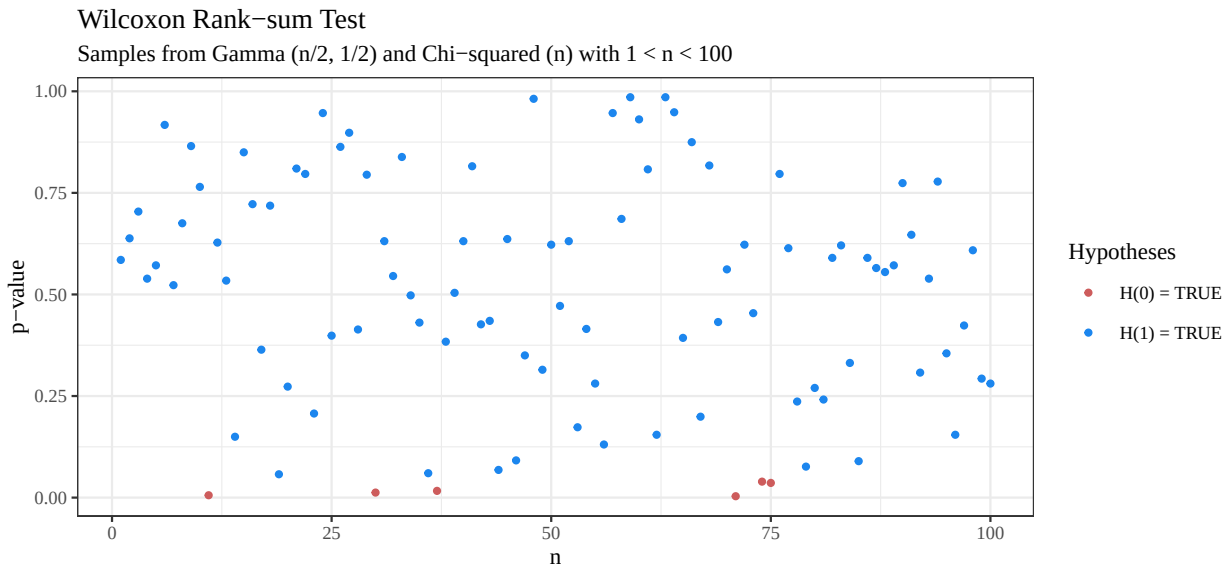
1. Show in **R**, also graphically, that Gamma$(\frac{n}{2}, \frac{1}{2})$ coincides with a χ^2_n.
2. Find the 5% and the 95% quantiles of a Gamma$(3, 3)$.

Solution 3.1

```
matrix_gamma <- matrix(data = NA, nrow = 100, ncol = 100)
for(i in 1:nrow(matrix_gamma))
{
  matrix_gamma[, i] = rgamma(ncol(matrix_gamma), i/2, 1/2);
};

matrix_chisq <- matrix(data = NA, nrow = 100, ncol = 100)
for(i in 1:ncol(matrix_gamma))
{
  matrix_chisq[, i] = rchisq(ncol(matrix_chisq), i);
};
```

One way to prove the coincidence of the two distributions is by the means of the **Wilcoxon rank-sum test**. The latter is a non-parametric alternative to the independent two sample t-test for comparing two independent groups of samples, in the situation where the data are not normally distributed.

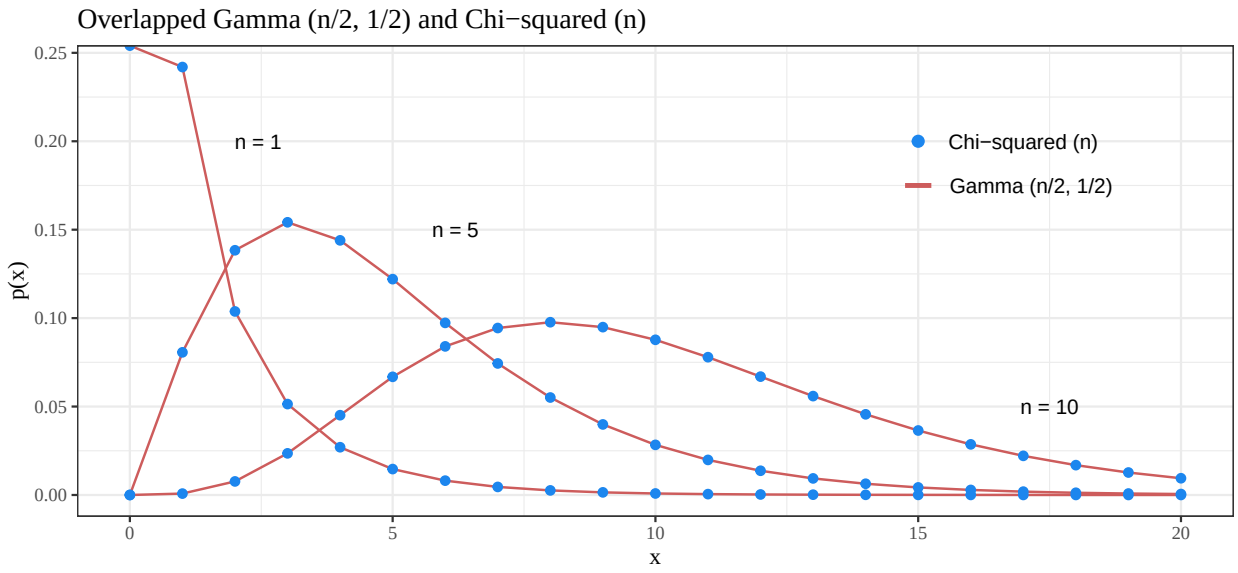
```
require(ggplot2)
require(extrafont)

test_wilcoxon <- data.frame(p_value  = rep(NA, 100),
                            accepted = rep(NA, 100))
for(i in 1:100)
{
  test_wilcoxon$p_value[i]  <- wilcox.test(matrix_gamma[, i], matrix_chisq[, i])$p.value;
  test_wilcoxon$accepted[i] <- as.character(ifelse(test_wilcoxon$p_value[i] <= 0.05,
                                                   "H(0) = TRUE",
                                                   "H(1) = TRUE"));
}

ggplot(data = test_wilcoxon, aes(x = c(1:100), y = p_value)) +
  geom_point(aes(color = accepted), size = 1) +
  labs(title    = "Wilcoxon Rank-sum Test",
       subtitle = "Samples from Gamma (n/2, 1/2) and Chi-squared (n) with 1 < n < 100",
       x        = "n",
       y        = "p-value",
       col      = "Hypotheses") +
  scale_color_manual(values = c("indianred",
                                "dodgerblue2")) +
  theme_bw(base_size = 10, base_family = "Times")
```



Our assumption can be further enhanced by representing graphically the two distributions

```
require(ggplot2)
require(extrafont)

ggplot() +

## Custom Legend
  geom_point(aes(x = 15, y = 0.2),
             colour = "dodgerblue2", size = 2) +
   geom_text(aes(x = 17, y = 0.2),
             label = "Chi-squared (n)", size = 3) +

  geom_line(aes(x = seq(14.75, 15.25, 0.25), y = rep(0.175, 3)),
            colour = "indianred", size = 1) +
   geom_text(aes(x = 17.15, y = 0.175),
            label = "Gamma (n/2, 1/2)", size = 3) +

## Custom Label
  geom_text(aes(x = 2.50, y = 0.20), label = "n = 1 ", size = 3) +
  geom_text(aes(x = 6.25, y = 0.15), label = "n = 5 ", size = 3) +
  geom_text(aes(x = 17.5, y = 0.05), label = "n = 10", size = 3) +

## n = 1
  geom_line(aes(x = c(0:20), y = dgamma(0:20, 1/2, 1/2)),
             colour = "indianred") +
  geom_point(aes(x = c(0:20), y = dchisq(0:20, 1)),
             colour = "dodgerblue2") +

## n = 5
  geom_line(aes(x = c(0:20), y = dgamma(0:20, 5/2, 1/2)),
            colour = "indianred") +
  geom_point(aes(x = c(0:20), y = dchisq(0:20, 5)),
             colour = "dodgerblue2") +
```

```
## n = 10
  geom_line(aes(x = c(0:20), y = dgamma(0:20, 10/2, 1/2)),
            colour = "indianred") +
  geom_point(aes(x = c(0:20), y = dchisq(0:20, 10)),
             colour = "dodgerblue2") +

## Labels
  labs(title = "Overlapped Gamma (n/2, 1/2) and Chi-squared (n)",
       x     = "x",
       y     = "p(x)") +
  theme_bw(base_size = 10, base_family = "Times")
```



Solution 3.2

```
cat(qgamma(c(0.05, 0.95), 3, 3))
```

```
> 0.2725638 2.098598
```

Exercise 4

1. Generate $n = 1000$ values from a Beta$(5, 2)$ and compute the sample mean and the sample variance.

Solution 4.1

```
cat(mean(rbeta(1000, 5, 2)))
```

```
> 0.7130336
```

```
cat(var(rbeta(1000, 5, 2)))
```

```
> 0.02647337
```

Exercise 5

1. Analogously, show with a simple **R** function that a negative binomial distribution may be seen as a mixture between a Poisson and a Gamma. In symbols: $X|Y \sim \mathcal{P}(Y), Y \sim \text{Gamma}(\alpha, \beta),$ then$X \sim ....$

Solution 5.1

Formally we can show that given:

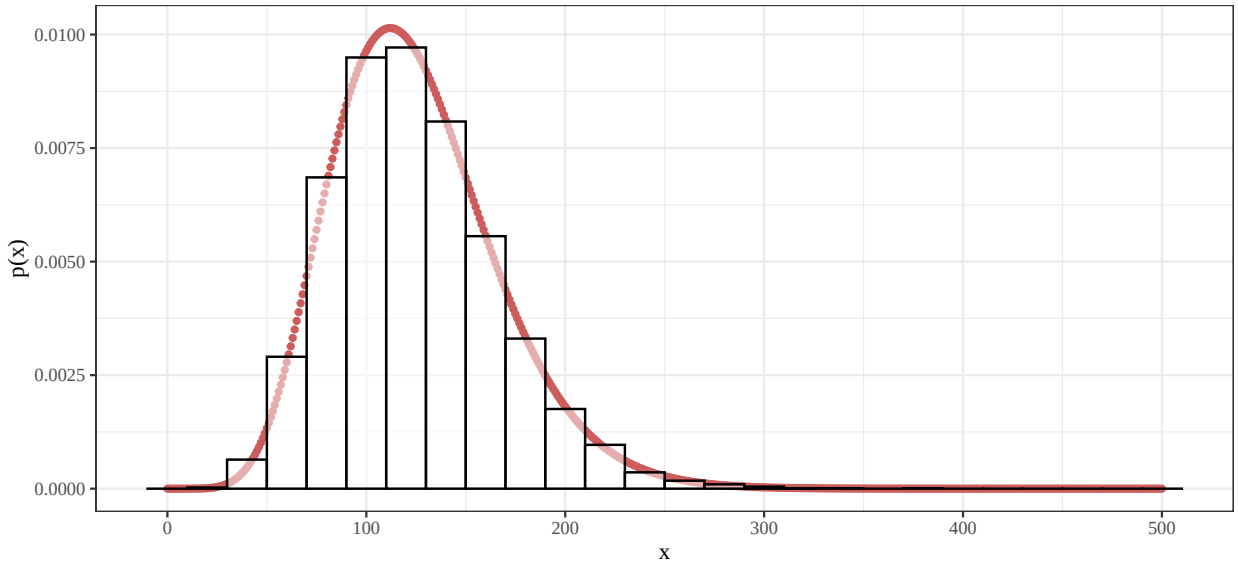
- $X|Y \sim P(Y) \rightarrow P(X = x|Y = y) = \frac{e^{-y}y^x}{x!}$, for $x = 0, 1, 2, \ldots$
- $Y \sim \text{Gamma}(\alpha, \beta) \rightarrow h_y = \frac{\alpha^\beta}{\Gamma(\beta)y^{\beta-1}e^{-\alpha y}}$, for $y > 0$

the marginal distribution of X is:

$$
\begin{aligned}
P(X = x) &= \int_0^\infty P(X = x|Y = y) \cdot h_Y(y)dy \\
&= \int_0^\infty \frac{e^{-y}y^x}{x!} \cdot \frac{\beta^\alpha}{\Gamma(\alpha)} \cdot y^{\alpha-1}e^{-\beta y}dy \\
&= \frac{\beta^\alpha}{x! \cdot \Gamma(\alpha)} \cdot \frac{\Gamma(x+\alpha)}{(\beta+1)^{x+\alpha}} \underbrace{\int_0^\infty \frac{(\beta+1)^{x+\alpha}}{\Gamma(x+\alpha)} \cdot y^{x+\alpha-1} \cdot e^{-(\beta+1)y}dy}_{=1} \\
&= \frac{\Gamma(x+\alpha)}{\Gamma(x+1)\Gamma(\alpha)} \cdot \left(\frac{\beta}{\beta+1}\right)^\alpha \left(\frac{1}{\beta+1}\right)^x \\
&= \binom{x+\alpha-1}{x} \left(\frac{\beta}{\beta+1}\right)^\alpha \left(\frac{1}{\beta+1}\right)^x,
\end{aligned}
$$

which is a NB$(x|\alpha, \frac{1}{1+\beta})$.

```
nb_mixture<-function(n, r, p)
{
  lambda = rgamma(n, r, p)
  X = rpois(n, lambda)
  return(X)
}

ggplot() +
  geom_point(aes(x = c(0:500), y = dnbinom(0:500, 10, 0.07407407)),
             size = 1,
             colour = "indianred") +
  geom_histogram(aes(x = nb_mixture(10000, 10, 0.08), y = ..density..),
                 binwidth = 20,
                 colour   = "black",
                 fill     = "white",
                 alpha    = 0.5) +
  labs(x        = "x",
       y        = "p(x)") +
  theme_bw(base_size = 10, base_family = "Times")
```



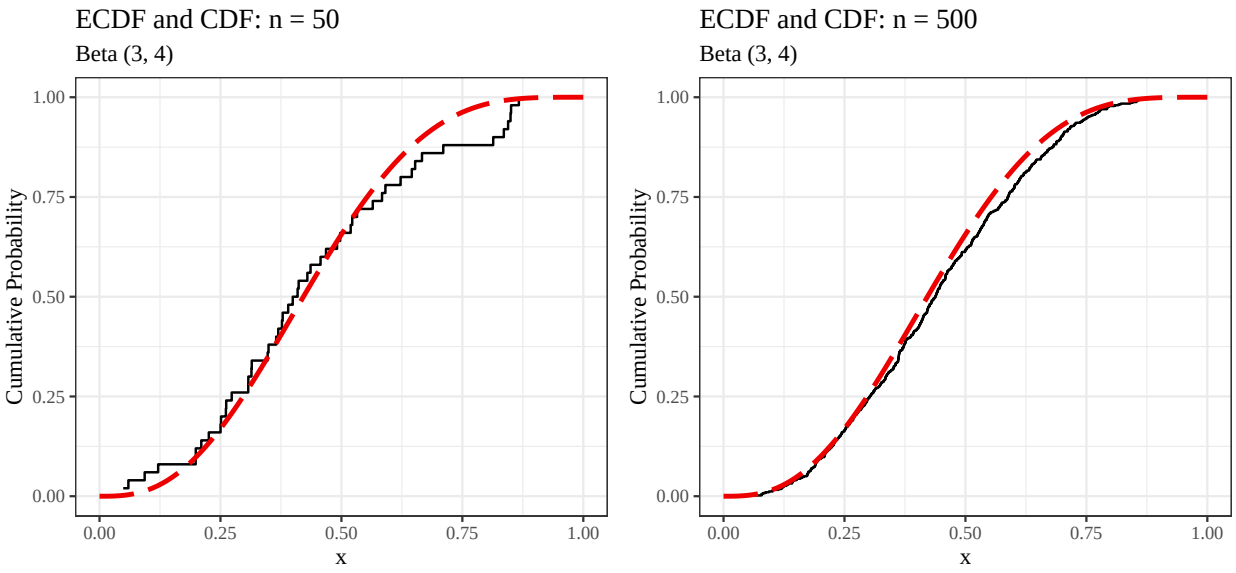
Exercise 6

1. Instead of using the built-in function **ecdf()**, write your own **R** function for the empirical cumulative distribution function and reproduce the two plots above.

Solution 6.1

```
require(ggplot2)
require(extrafont)
require(gridExtra)
```

```r
set.seed(2)

empirical_cdf <- function(x)
{
  ecdf <- data.frame(x    = sort(unique(x)),
                     F_x  = cumsum(data.frame(table(sort(x)))$Freq)/
                            sum(data.frame(table(sort(x)))$Freq));
}

beta_one <- ggplot() +
              geom_step(data = empirical_cdf(rbeta(50, 3, 4)),
                        aes(x = x, y = F_x)) +
              geom_line(aes(x = seq(0, 1, 0.01), y = pbeta(seq(0, 1, 0.01), 3, 4)),
                        colour   = "red2",
                        size     = 1,
                        linetype = "longdash") +
              labs(title    = "ECDF and CDF: n = 50",
                   subtitle = "Beta (3, 4)",
                   x = "x",
                   y = "Cumulative Probability") +
              theme_bw(base_size = 10, base_family = "Times")

beta_two <- ggplot() +
              geom_step(data = empirical_cdf(rbeta(500, 3, 4)),
                        aes(x = x, y = F_x)) +
              geom_line(aes(x = seq(0, 1, 0.01), y = pbeta(seq(0, 1, 0.01), 3, 4)),
                        colour   = "red2",
                        size     = 1,
                        linetype = "longdash") +
              labs(title    = "ECDF and CDF: n = 500",
                   subtitle = "Beta (3, 4)",
                   x = "x",
                   y = "Cumulative Probability") +
              theme_bw(base_size = 10, base_family = "Times")

grid.arrange(beta_one, beta_two, nrow = 1)
```


ECDF and CDF: n = 50
Beta (3, 4)
Cumulative Probability
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
x
ECDF and CDF: n = 500
Beta (3, 4)
Cumulative Probability
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
x

Exercise 7

Compare in **R** the assumption of normality for these samples:

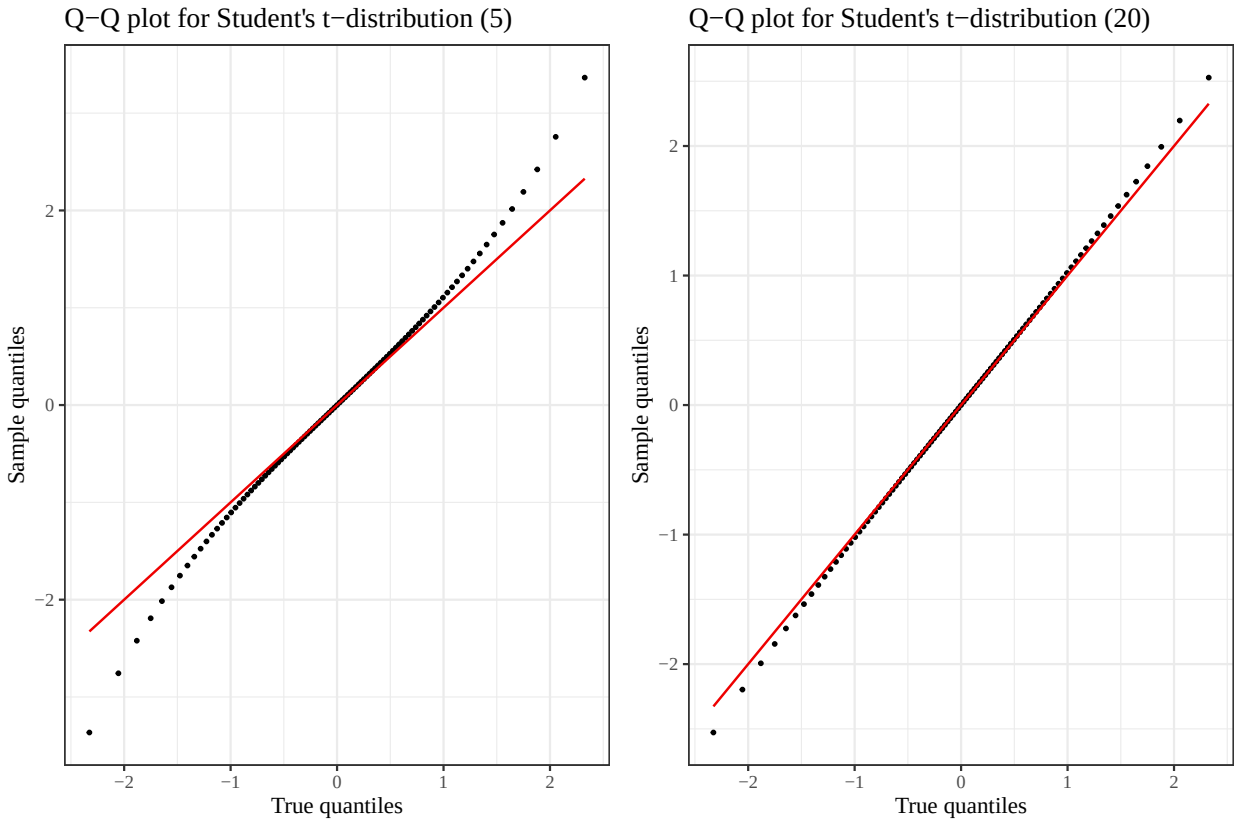
1. $y_1, \ldots, y_{100} \sim t_v$, with $v = 5, 20, 100$. What happens when the number of degrees of freedom v increases?
2. $y_1, \ldots, y_{100} \sim \text{Cauchy}(0, 1)$. Do you note something weird for the extremes quantiles?

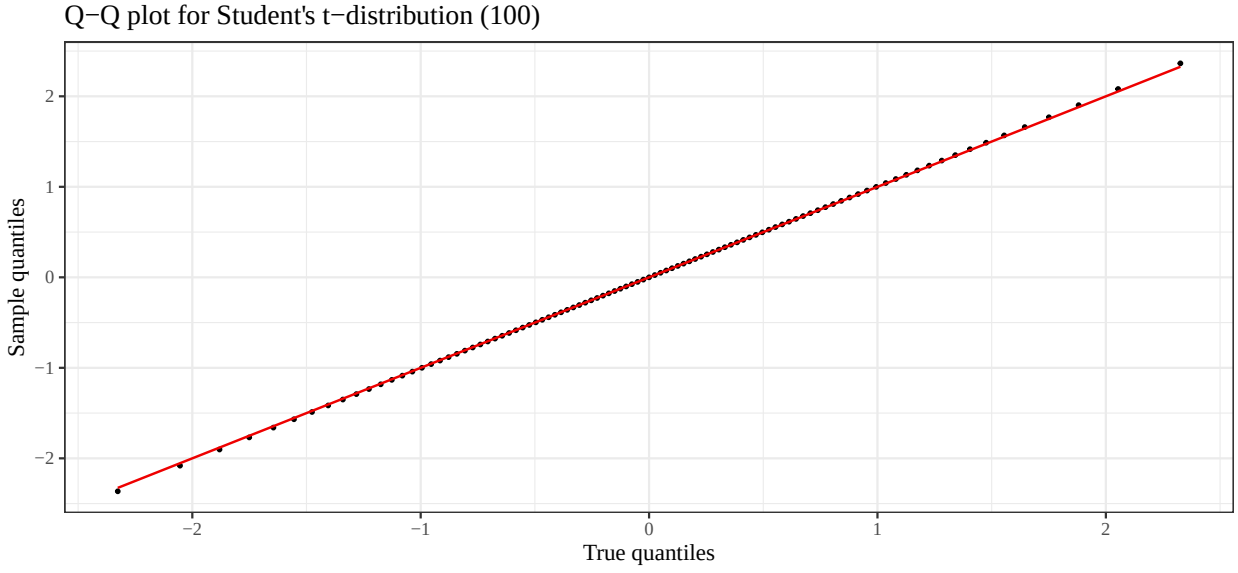
Solution 7.1

The t-distribution is symmetric and bell-shaped, like the normal distribution, but has heavier tails, meaning that it is more prone to producing values that fall far from its mean. However, as we increase the degrees of freedom, a t-distribution approaches the shape of a normal distribution. This can be seen by comparing the **Q-Q** plots of a t-distribution with $1, 20$ and 100 degrees of freedom.

```
require(ggplot2)
require(extrafont)
require(gridExtra)

t_one <- ggplot() +
          geom_point(aes(x = qnorm(seq(0.01, 0.99, 0.01), 0, 1),
                         y = qt(seq(0.01, 0.99, 0.01), 5)),
                     size = 0.5) +
          geom_line(aes(x = qnorm(seq(0.01, 0.99, 0.01), 0, 1),
                        y = qnorm(seq(0.01, 0.99, 0.01), 0, 1)),
                    color = "red2") +
          labs(title = "Q-Q plot for Student's t-distribution (5)",
               x     = "True quantiles",
               y     = "Sample quantiles") +
          theme_bw(base_size = 10, base_family = "Times")

t_two <- ggplot() +
          geom_point(aes(x = qnorm(seq(0.01, 0.99, 0.01), 0, 1),
                         y = qt(seq(0.01, 0.99, 0.01), 20)),
                     size = 0.5) +
          geom_line(aes(x = qnorm(seq(0.01, 0.99, 0.01), 0, 1),
                        y = qnorm(seq(0.01, 0.99, 0.01), 0, 1)),
                    color = "red2") +
          labs(title = "Q-Q plot for Student's t-distribution (20)",
               x     = "True quantiles",
               y     = "Sample quantiles") +
          theme_bw(base_size = 10, base_family = "Times")

t_three <- ggplot() +
            geom_point(aes(x = qnorm(seq(0.01, 0.99, 0.01), 0, 1),
                           y = qt(seq(0.01, 0.99, 0.01), 100)),
                       size = 0.5) +
            geom_line(aes(x = qnorm(seq(0.01, 0.99, 0.01), 0, 1),
                          y = qnorm(seq(0.01, 0.99, 0.01), 0, 1)),
                      color = "red2") +
            labs(title = "Q-Q plot for Student's t-distribution (100)",
                 x     = "True quantiles",
```

```
                y     = "Sample quantiles") +
           theme_bw(base_size = 10, base_family = "Times")

grid.arrange(t_one, t_two, nrow = 1, ncol = 2)
```

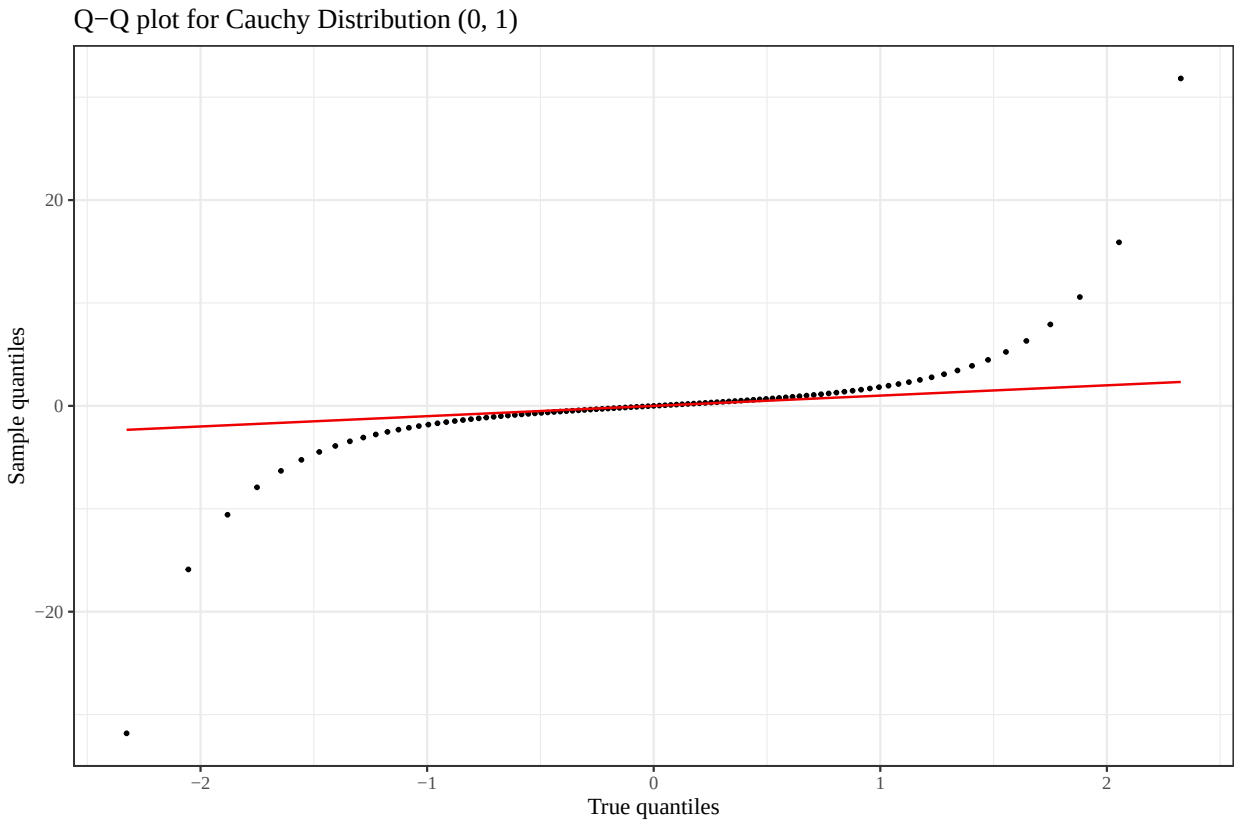




Solution 7.2

The Cauchy distribution is a heavy tailed distribution. Its probability density function $f(x)$ decreases at a *polynomial rate* as $x \to \infty$ and $x \to -\infty$.

```
require(ggplot2)
require(extrafont)
require(gridExtra)

ggplot() +
  geom_point(aes(x = qnorm(seq(0.01, 0.99, 0.01), 0, 1),
                 y = qcauchy(seq(0.01, 0.99, 0.01), 0, 1)),
             size = 0.5) +
  geom_line(aes(x = qnorm(seq(0.01, 0.99, 0.01), 0, 1),
                y = qnorm(seq(0.01, 0.99, 0.01), 0, 1)),
            color = "red2") +
  labs(title = "Q-Q plot for Cauchy Distribution (0, 1)",
       x     = "True quantiles",
       y     = "Sample quantiles") +
  theme_bw(base_size = 10, base_family = "Times")
```



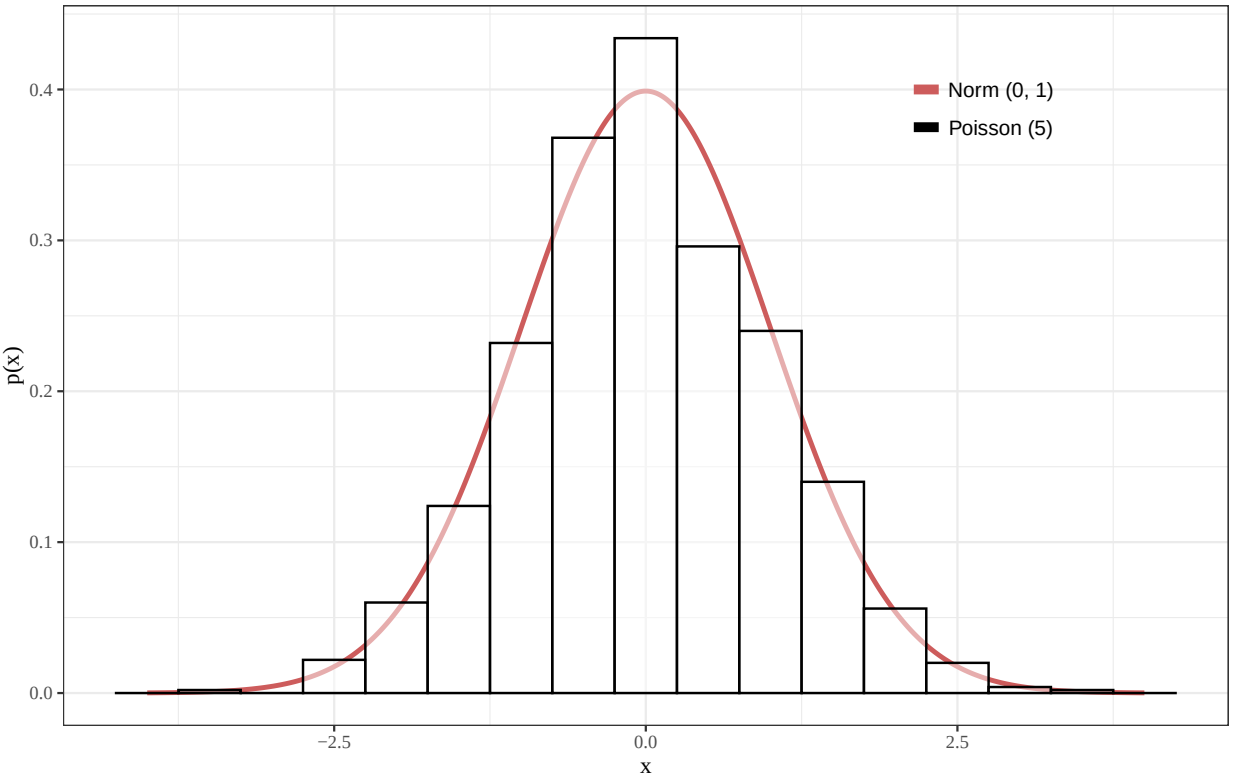
Exercise 8

1. Write a general **R** function for checking the validity of the central limit theorem. *Hint*: The function will consist of two parameters: clt_function <- function(n, distr), where the first one is the sample size and the second one is the kind of distribution from which you generate. Use plots for visualizing the results.

Solution 8.1

```
require(ggplot2)
require(extrafont)

clt_function <- function(n, distr)
{
  sample_means = scale(rowMeans(matrix(distr, n)));
};

ggplot() +

# Custom Legend
  geom_line(aes(x = seq(2.15, 2.35, 0.10), y = rep(0.4, 3)),
            colour = "indianred", size = 2) +
  geom_text(aes(x = 2.85, y = 0.4),
            label = "Norm (0, 1)", size = 3) +

  geom_line(aes(x = seq(2.15, 2.35, 0.10), y = rep(0.375, 3)),
            colour = "black", size = 2) +
  geom_text(aes(x = 2.85, y = 0.375),
            label = "Poisson (5)", size = 3) +

# Norm (0, 1)
  geom_line(aes(x = seq(-4, 4, 0.01),
                y = dnorm(seq(-4, 4, 0.01), 0, 1)),
            colour = "indianred",
            size   = 1) +

# Poisson (5)
  geom_histogram(aes(x = clt_function(1000, rpois(1000*1000, 5)),
                     y = ..density..),
                 binwidth = 0.5,
                 colour   = "black",
                 fill     = "white",
                 alpha    = 0.5) +

# Labels
  labs(title    = "Central Limit Theorem",
       x        = "x",
       y        = "p(x)") +
  theme_bw(base_size = 10, base_family = "Times")
```

Central Limit Theorem



DAAG Exercises

Solution 4.1

```
require(DAAG)

for (i in 1:length(ais))
{cat(str(ais[, i]), paste0("NA: ", sum(is.na(ais[, i]))), "\n")}
```

```
>  num [1:202] 3.96 4.41 4.14 4.11 4.45 4.1 4.31 4.42 4.3 4.51 ...
>  NA: 0
>  num [1:202] 7.5 8.3 5 5.3 6.8 4.4 5.3 5.7 8.9 4.4 ...
>  NA: 0
>  num [1:202] 37.5 38.2 36.4 37.3 41.5 37.4 39.6 39.9 41.1 41.6 ...
>  NA: 0
>  num [1:202] 12.3 12.7 11.6 12.6 14 12.5 12.8 13.2 13.5 12.7 ...
>  NA: 0
>  num [1:202] 60 68 21 69 29 42 73 44 41 44 ...
>  NA: 0
>  num [1:202] 20.6 20.7 21.9 21.9 19 ...
>  NA: 0
>  num [1:202] 109.1 102.8 104.6 126.4 80.3 ...
>  NA: 0
>  num [1:202] 19.8 21.3 19.9 23.7 17.6 ...
>  NA: 0
>  num [1:202] 63.3 58.5 55.4 57.2 53.2 ...
>  NA: 0
>  num [1:202] 196 190 178 185 185 ...
>  NA: 0
>  num [1:202] 78.9 74.4 69.1 74.9 64.6 63.7 75.2 62.3 66.5 62.9 ...
>  NA: 0
>  Factor w/ 2 levels "f","m": 1 1 1 1 1 1 1 1 1 1 ...
>  NA: 0
>  Factor w/ 10 levels "B_Ball","Field",..: 1 1 1 1 1 1 1 1 1 1 ...
>  NA: 0
```

Solution 4.2

```
require(DAAG)
require(knitr)

tab <- table(ais$sex, ais$sport)
tab <- rbind(tab, tab[1,]/tab[2,])
rownames(tab) <- c(rownames(tab[1:2,]), "ratio")
knitr::kable(round(tab, 2))
```

	B_Ball	Field	Gym	Netball	Row	Swim	T_400m	T_Sprnt	Tennis	W_Polo
f	13.00	7.00	4	23	22.00	9.00	11.00	4.00	7.00	0
m	12.00	12.00	0	0	15.00	13.00	18.00	11.00	4.00	17
ratio	1.08	0.58	Inf	Inf	1.47	0.69	0.61	0.36	1.75	0

So the sports with a ratio greater than 2:1 are Gym, Netball, T_Sprint and Polo.

Solution 6.1

```
dtf_Manitoba <- data.frame(elevation = c(217, 254, 248,
                                         254, 253, 227,
                                         178, 207, 217),
                           area = c(24387, 5374, 4624,
                                    2247, 1353, 1223,
                                    1151, 755, 657))

row.names(dtf_Manitoba) = c("Winnipeg", "Winnipegosis", "Manitoba",
                            "SouthernIndian", "Cedar", "Island",
                            "Gods", "Cross", "Playgreen")

plot(log2(dtf_Manitoba$area) ~ dtf_Manitoba$elevation, pch = 16, xlim = c(170, 280),
     xlab = "elevation", ylab = "log2(area)")
text(log2(dtf_Manitoba$area) ~ dtf_Manitoba$elevation, labels = row.names(dtf_Manitoba),
     pos = 4)
text(log2(dtf_Manitoba$area) ~ dtf_Manitoba$elevation, labels = dtf_Manitoba$area,
     pos = 2)
title("Manitoba's Largest Lakes")
```

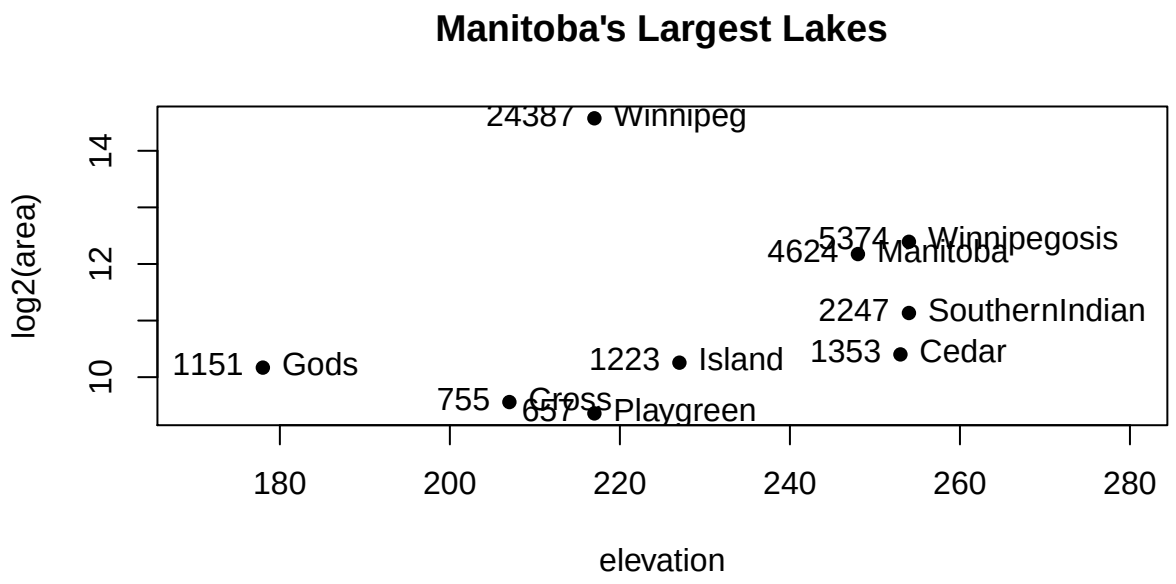
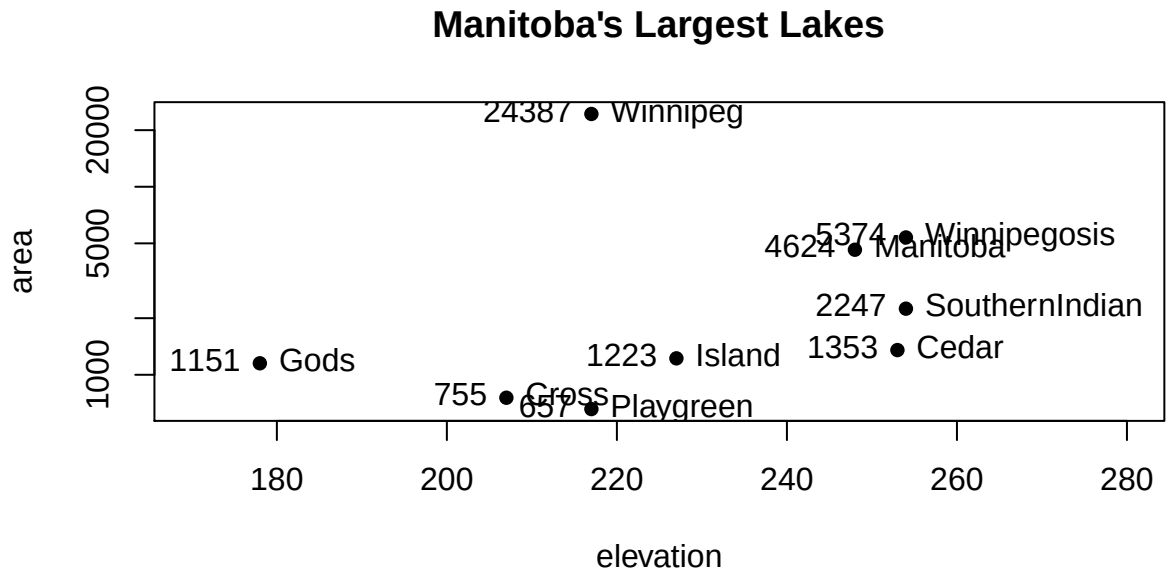


Figure 1: Each point is labeled with the name of the lake and its area. The Y-axis uses a base-2 logarithm scale which correspond to a doubling factor increase in the area on each step.

```
plot(dtf_Manitoba$area ~ dtf_Manitoba$elevation  , pch = 16, log = "y", xlim = c(170, 280),
     xlab = "elevation", ylab = "area")
text(dtf_Manitoba$area ~ dtf_Manitoba$elevation, labels=row.names(dtf_Manitoba),
```

```
     pos = 4)
text(dtf_Manitoba$area ~ dtf_Manitoba$elevation, labels=dtf_Manitoba$area,
     pos = 2)
title("Manitoba's Largest Lakes")
```



Solution 11

```
require(knitr)

gender <- factor(c(rep("female", 91), rep("male", 92)))
knitr::kable(table(gender))
```

gender	Freq
female	91
male	92

Here, a factor is assigned to gender with repetition of two categories. The table counts the occourences of the values encountered in gender and assigns levels just based on their representation. The table summary is shown in alphabetical order.

```
require(knitr)

gender <- factor(gender, levels = c("male", "female"))
knitr::kable(table(gender))
```

gender	Freq
male	92
female	91

The levels specification just imposes the order of the levels, so table function shows them respecting it.

```
require(knitr)

gender <- factor(gender, levels = c("Male", "female"))
# Note the mistake: "Male" should be "male"
knitr::kable(table(gender))
```

gender	Freq
Male	0
female	91

Factors values ar case sensitive. To gender is reassigned leves with “Male” uppercase. No occurrency of such value are present in “gender” and old “male” occurrences are interpreted as a missing value.

```
require(knitr)

knitr::kable(table(gender, exclude = NULL))
```

gender	Freq
Male	0
female	91
NA	92

This last table shows the transormation of “male” into a missing value.

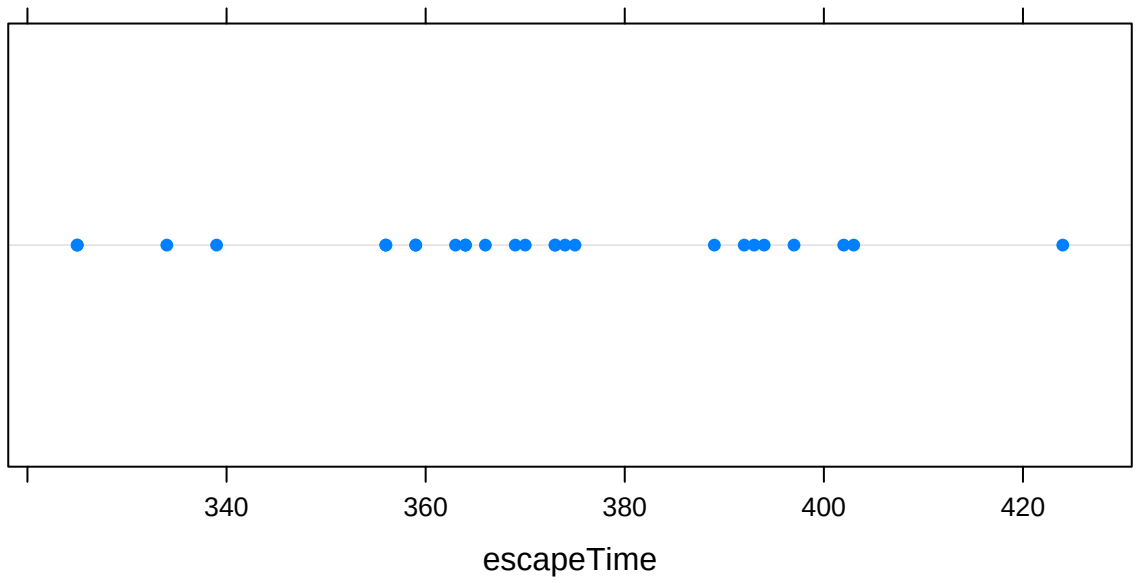
Solution 12.1

```
cutoffFun <- function(value, vect)
{return(length(vect[vect > value])/(length(vect)))}
# If i insert 40 i would expect 0.6 as output:
cat(cutoffFun(40, c(1:100)))
```

```
> 0.6
```

Solution 12.2

```
require(Devore7)

escapeTime = ex01.36[, 1]
dotplot(escapeTime)
```

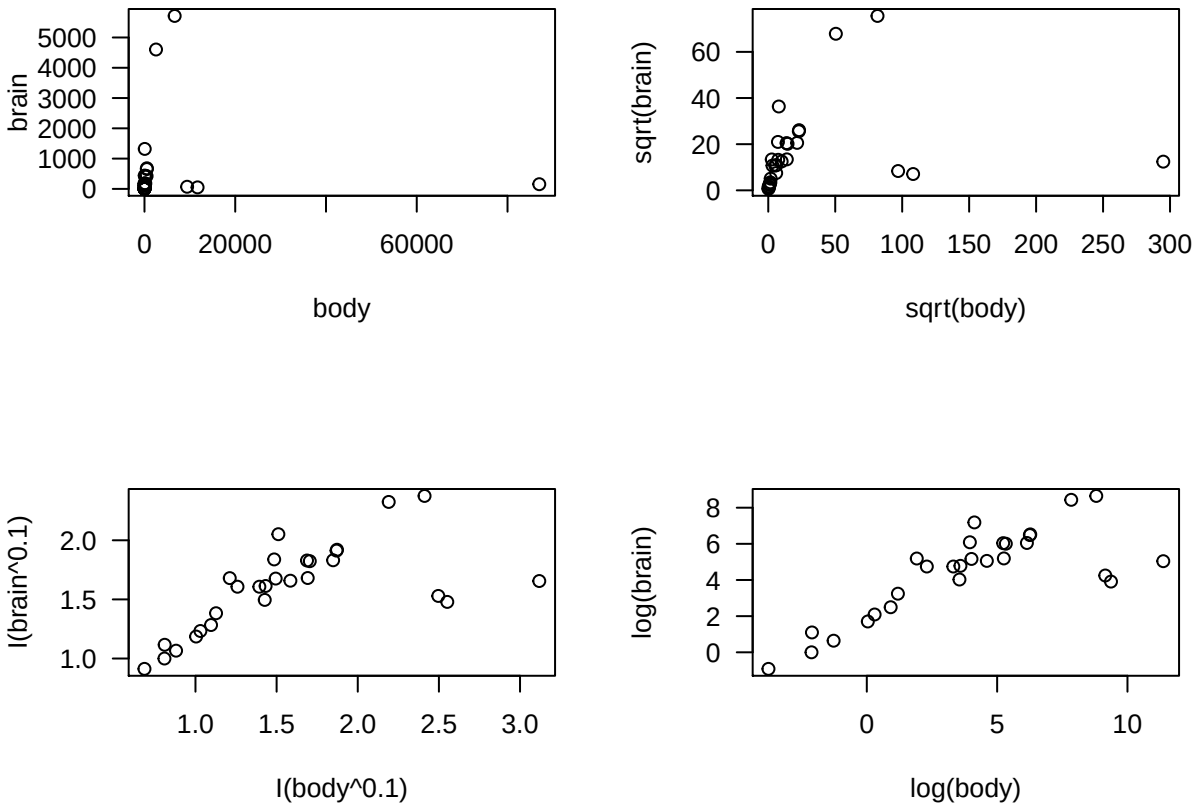



```
cat("The proportion of escaping time exceeding 7 minutes is: ",
        cutoffFun(7 * 60, escapeTime))
```

```
> The proportion of escaping time exceeding 7 minutes is:  0.03846154
```

Solution 13.1

```
require(MASS)

par(mfrow = c(2, 2),las = 1) # 2 by 2 layout on the page;

plot(brain        ~ body, data = Animals)
plot(sqrt(brain)  ~ sqrt(body), data = Animals)
plot(I(brain^0.1) ~ I(body^0.1), data = Animals)
plot(log(brain)   ~ log(body), data = Animals)
```

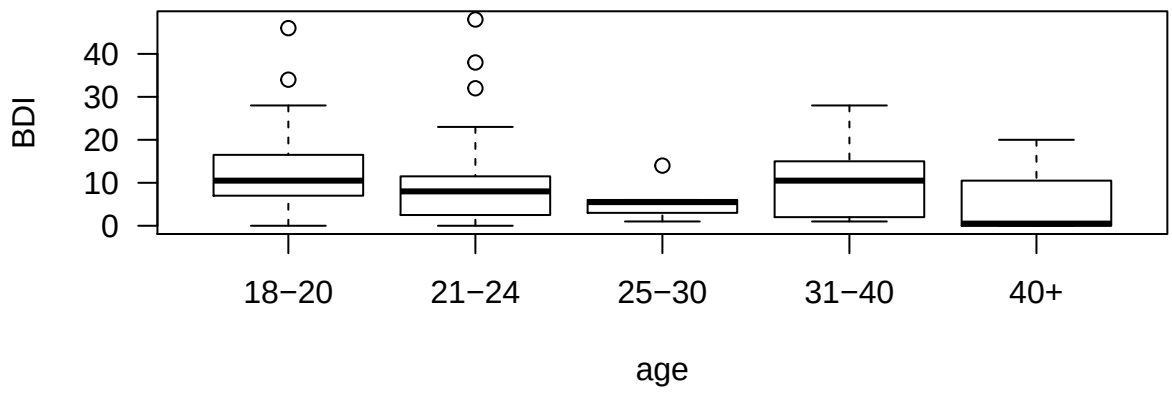


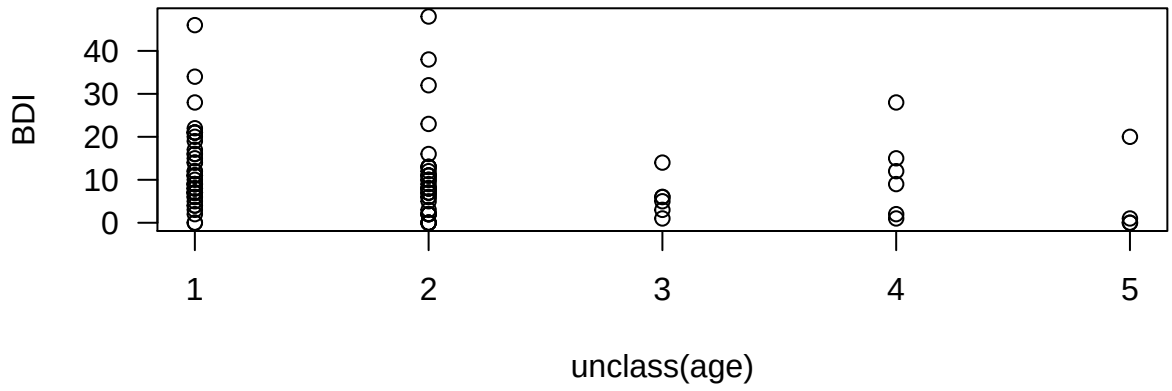
The four plots are progressively increasing the scale and therefore reducing the visual distance between points with low components and the onces with high components. The log transformations are particularly useful whenever there are outliers negating the possibility to correctly visualize the data.

Solution 15

```
require(DAAG)
```

```r
par(mfrow = c(2, 1), las = 1)
plot(BDI ~ age, data = socsupport)
plot(BDI ~ unclass(age), data = socsupport)
```





In order to identify high score cases we need (also to define what high score means) to capture whether an individual has a score much higher that what we would expect them to have, if compared with similar cases. Here the criteria of similarity is given by age group. The first plot results much more indicative toward this purpose as it shows how an outlier looks when compared to the mean and the standard deviation interval. The issue of using these plots for this analysis (in this particular case), is that for the groups with very little data it can be misleading to draw any conclusion as obviously a very sample can't provide any reliable statistics.

Solution 17

```
x <- NULL
print(seq(1, length(x)))
```

```
> [1] 1 0
```

```
print(seq(along = x))
```

```
> integer(0)
```

The second formulation is the correct one to achieve the task.

Solution 20

```
dni3 <- dimnames(iris3)
ii <- data.frame(matrix(aperm(iris3, c(1,3,2)), ncol = 4,
                        dimnames = list(NULL, sub(" L.",".Length",
                                          sub(" W.",".Width", dni3[[2]])))),
    Species = gl(3, 50, labels = sub("S", "s", sub("V", "v", dni3[[3]]))))
cat(all.equal(ii, iris))
```

```
> TRUE
```

- dni3 <- dimnames(iris3): gets names of iris3 df dimensions. At this stage iris is a 3 dimensional array whith x = observation index, y = class, and z = feature. The goal is to convert it into a two dimensional array of dimension (150X4) indicating for all observations the characteristics (features), and finally append a column indicating as an additional feature the class of the observations.
- matrix(aperm(iris3, c (1,3,2)), ncol=4,...): creates the 2 dim matrix after *aperm* swaps y and z in order to guarantee that the new column is made with the class values.
- dimnames = list(NULL, sub(" L.",".Length",sub(" W.",".Width", dni3[[2]]))): changes string values. Important to notice that *sub* operates recursively to obtain in one single line the name update of both suffixes L and W.
- Species = gl(3, 50, labels = sub("S", "s", sub("V", "v", dni3[[3]]))): appends to the dataframe the additional column. *gl* creates a factor of 50 repetitions of the 3 values containd in *dni3[[3]]*. This order matches the one which has been produced by the matrix command.

Core Statistics Exercises

Exercise 1.1

Exponential random variable, $X \geq 0$, has p.d.f. $f(x) = \lambda \exp(-\lambda x)$.

1. Find the c.d.f. and the quantile function for X.
2. Find $\Pr(X < \lambda)$ and the median of X.
3. Find the median and variance of X.

Solution:

1. The cumulative distribution function of an exponential distribution $X \geq 0$ is by definition

$$\begin{aligned} F(x) = P(X \leq x) &= \int_0^x f(s)\, ds \\ &= \int_0^x \lambda e^{-\lambda s} ds \\ &= 1 - e^{-\lambda x} \end{aligned}$$

In the case of an exponential distribution (and of any c.d.f. which is continuous), we compute the quantile function for X as the value q_α s.t.

$$F(q_\alpha) = \alpha, \quad \alpha \in [a, b]$$

Hence we have to solve $1 - e^{-\lambda q_\alpha} = \alpha$ and we get

$$q_\alpha = -\frac{\ln(1-\alpha)}{\lambda} = \frac{1}{\lambda} \ln\left(\frac{1}{1-\alpha}\right)$$

2. By definition

$$\begin{aligned} Pr(X < \lambda) &= \int_0^\lambda f(s) ds \\ &= \int_0^\lambda \lambda e^{-\lambda s} ds \\ &= 1 - e^{-\lambda^2} \end{aligned}$$

The median of X is the quantile function of order $\frac{1}{2}$, i.e. the value $q_{\frac{1}{2}}$ s.t. $F(q_{\frac{1}{2}}) = \frac{1}{2}$ with F the c.d.f. By solving $1 - e^{-\lambda q_{\frac{1}{2}}} = \frac{1}{2}$ we get

$$q_{\frac{1}{2}} = \frac{1}{\lambda} ln(2)$$

3. By definition,

$$\begin{aligned} \mathbb{E}[X] &= \int_0^\infty s f(s) ds \\ &= \int_0^\infty \lambda s e^{\lambda s} ds = \frac{1}{\lambda} \end{aligned}$$

By definition,

$$\begin{aligned} Var(X) &= \int_0^\infty (s - \mathbb{E}[X])^2 f(s) ds \\ &= \int_0^\infty (s-\lambda)^2 \lambda e^{-\lambda s} ds \\ &= \frac{1}{\lambda^2} \end{aligned}$$

Exercise 1.2

Evaluate $\Pr(X < 0.5, Y < 0.5)$ if X and Y have joint p.d.f. (1.2).

Solution:

X and Y have the following joint p.d.f.

$$f(x,y) = \begin{cases} x + \frac{3}{2}y^2 & \text{if} \quad 0 < x < 1 \quad \text{and} \quad 0 < y < 1 \\ 0 & \text{otherwise} \end{cases}$$

Hence

$$\begin{aligned} P(X < \frac{1}{2}, Y < \frac{1}{2}) &= \int_{-\infty}^{\frac{1}{2}} \int_{-\infty}^{\frac{1}{2}} f(x,y)dxdy \\ &= \int_{0}^{\frac{1}{2}} \int_{0}^{\frac{1}{2}} (x + \frac{3}{2}y^2)dxdy \\ &= \frac{3}{32} \end{aligned}$$

Exercise 1.6

Let X and Y be non-independent random variables, such that $Var(X) = \sigma_x^2$, $Var(Y) = \sigma_y^2$ and $Cov(X,Y) = \sigma_{x,y}^2$. Using the result of section 1.6.2, find $Var(X+Y)$ and $Var(X-Y)$.

Solution:

1. By definition

$$Var(X+Y) = \mathbb{E}[((X+Y) - \mathbb{E}[\mathbb{X} + \mathbb{Y}])^2]$$

and we know that for any couple of random variables X, Y

$$\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$$

Hence

$$\begin{aligned} Var(X+Y) =& \mathbb{E}[X^2 + Y^2 + 2XY + \mathbb{E}[X]^2 + \mathbb{E}[Y]^2 + 2\mathbb{E}[X]\mathbb{E}[Y] - 2X\mathbb{E}[X] - 2X\mathbb{E}[Y] - 2Y\mathbb{E}[X] - 2Y\mathbb{E}[Y]] \\ =& Var(X) + Var(Y) + 2(\mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]) \end{aligned}$$

since

$$\begin{aligned} \mathbb{E}[(X - \mathbb{E}[X])^2] &= Var(X) \\ \mathbb{E}[(Y - \mathbb{E}[Y])^2] &= Var(Y) \\ \mathbb{E}[X\mathbb{E}[Y]] = \mathbb{E}[Y\mathbb{E}[X]] &= \mathbb{E}[X]\mathbb{E}[Y] \end{aligned}$$

Now, since

$$\mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = Cov(X,Y)$$

we get

$$Var(X+Y) = Var(X) + Var(Y) + 2Cov(X,Y)$$

Since $Var(X) = \sigma_x^2$, $Var(Y) = \sigma_y^2$, $Cov(X,Y) = \sigma_{x,y}^2$, we get

$$Var(X+Y) = \sigma_x^2 + \sigma_y^2 + 2\sigma_{x,y}^2$$

2. We have that

$$\begin{aligned} Var(X-Y) &= \mathbb{E}[((X-Y) - \mathbb{E}[X-Y])^2] \\ &= Var(X) + Var(Y) - 2(\mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]) \end{aligned}$$

using the same observations of the previous computations. Hence

$$\begin{aligned} Var(X-Y) &= Var(X) + Var(Y) - 2Cov(X,Y) \\ &= \sigma_x^2 + \sigma y^2 - 2\sigma_{x,y}^2 \end{aligned}$$

Exercise 1.8

If $X \sim N(\mu, \sigma^2)$, find the p.d.f. of X.

Solution:

La funzione $t \to \log(t)$ is a diffeomorphism and the composition $T := \log \circ X$ is a continuous random variable. Moreover the p.d.f. of T is the one of $N(\mu, \sigma^2)$, i.e.

$$f_T(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

On the other hand, the function $x \to \exp(x)$ is also a diffeomorphism and we can compose this one with the random variable T getting $X := \exp(T) = \exp(\log(X))$. X is continuous and for any $t \in \Re$

$$\begin{aligned} f_X(t) &= f_T(\exp^{-1}(t)) \cdot |(\exp^{-1})'(t)| \\ &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\log(t)-\mu)^2}{2\sigma^2}\right) \cdot \frac{1}{|t|} \end{aligned}$$

Exercise 1.9

Discrete random variables Y has a Poisson distribution with parameter λ if its p.d.f. is $f(y) = \lambda^y e^{-\lambda}/y!$, for $y = 0, 1, 2....$

a. Find the moment generating function for Y (hint: the power series representation of the exponential function is useful).

b. If $Y_1 \sim \text{Poi}(\lambda_1)$ and independently $Y_2 \sim \text{Poi}(\lambda_2)$, deduce the distribution of $Y_1 + Y_2$, by employing a general property of the m.g.f.s.

c. Making use of the previous result and the central limit theorem, deduce the normal approximation to the Poisson distribution.

d. Confirm the previous result graphically, using R functions dpois, dnorm, plot or barplot and lines. Confirm that the approximation improves with increasing λ.

Solution:

a. The moment generating function $g(t)$ of X is:

$$\begin{aligned} g(t) = \mathbb{E}[e^{tx}] &= \sum_{k=1}^{+\infty} e^{kt} \frac{e^{-\lambda}\lambda^k}{k!} \\ &= \sum_{k=1}^{+\infty} e^{\lambda} \frac{(e^t\lambda)^k}{k!} \\ &= e^{\lambda(e^t-1)}. \end{aligned}$$

b. Let's compute the moment-generating function of $Y_1 + Y_2$:

$$\begin{aligned} g(t) &= \mathbb{E}[e^{t(y_1+y_2)}] \\ &= \mathbb{E}[e^{ty_1}e^{ty_2}] \\ &= \mathbb{E}[e^{ty_1}]\mathbb{E}[e^{ty_2}] \end{aligned}$$

the last operation being allowed since Y_1, Y_2 are independent. We get

$$\begin{aligned} g(t) &= \mathbb{E}[e^{ty_1}]\mathbb{E}[e^{ty_2}] \\ &= \sum_{k=1}^{+\infty} e^{-\lambda_1} \frac{(e^t\lambda_1)^k}{k!} \cdot \sum_{k=1}^{+\infty} e^{-\lambda_2} \frac{(e^t\lambda_2)^k}{k!} \\ &= e^{\lambda_1(e^t-1)}e^{\lambda_2(e^t-1)} \\ &= e^{(e^t-1)(\lambda_1+\lambda_2)} \end{aligned}$$

which is the m.g.f. of a Poisson with parameter $\lambda_1 + \lambda_2$.

c. The central limit theorem states that if we have i.i.d. random variables $X_1, ..., X_n$ with mean μ and variance σ^2 and $\bar{X}_n := \sum_{i=1}^n \frac{X_i}{n}$, then $\bar{X}_n \sim N(\mu, \frac{\sigma^2}{n})$. So let $X_\lambda = Poi(\lambda)$, $\lambda = 1, 2, ...$, with p.d.f. $f_{X_\lambda}(x) = \frac{\lambda^x e^{-\lambda}}{x!}$, $x = 0, 1, 2, ...$, with mean $\mu = \lambda$ and variance $\sigma^2 = \lambda$. Let's consider the "standardized" Poisson r.v.

$$\frac{X_\lambda - \mu}{\sigma} = \frac{X_\lambda - \lambda}{\sqrt{\lambda}}$$

and let's apply the central limit theorem to it. Since any finite sum of Poisson random variables with parameters $\lambda_1, ...\lambda_N$ is a Poisson with parameter $\lambda_1 + ... + \lambda_N$ (previous point of this exercise), whenever we add a new Poisson r.v. to our sum we get a new Poisson with a larger parameter, hence we can express this infinite summing by sending λ to infinity. So if we compute the m.g.f. of the "limiting" Poisson we get

$$\begin{aligned} \lim_{\lambda\to+\infty} M_{\frac{X_\lambda-\lambda}{\sqrt{\lambda}}}(t) &= \lim_{\lambda\to+\infty} e^{-t\sqrt{\lambda}} \cdot \mathbb{E}\left[\exp\left(\frac{tX_\lambda}{\sqrt{\lambda}}\right)\right] \\ &= \lim_{\lambda\to+\infty} e^{-t\sqrt{\lambda}} \cdot e^{\lambda(e^{\frac{t}{\sqrt{\lambda}}}-1)} \\ &= \lim_{\lambda\to+\infty} e^{-t\sqrt{\lambda}} \cdot e^{\lambda\left(\frac{t}{\sqrt{\lambda}}+\frac{t^2}{2\lambda}+\frac{t^3}{6\lambda^{3/2}}+\cdots\right)} \\ &= \lim_{\lambda\to+\infty} e^{\frac{t^2}{2}+\frac{t^3}{6\lambda^{3/2}}+\cdots} \\ &= e^{\frac{t^2}{2}} \end{aligned}$$

which is the m.g.f. of a standard normal r.v. $N(0, 1)$. So for λ large $\frac{X_\lambda-\lambda}{\sqrt{\lambda}} \sim N(0, 1)$, that is $X_\lambda \sim N(\lambda, \lambda)$.

d. Let's implement on R the previous result:

```
par(mfrow= c(1,3), mar=c(2,2,2,1), oma=c(2,2,2,2))
n <- 5; N<- 50; M <- 500;
sqn <- sqrt(n); sqN <- sqrt(N); sqM <- sqrt(M)

plot(dpois(0:(2*n), n))
lines(dnorm(0:(2*n), n, sqn), col = "green", lwd = 1)
plot(dpois(0:(2*N), N))
lines(dnorm(0:(2*N), N, sqN), col = "green", lwd = 1)
plot(dpois(0:(2*M), M))
lines(dnorm(0:(2*M), M, sqM), col= "red", lwd=1)
```