Assignment 2: Data Visualization

Azoacha Forcheh, 20558994

1. Visual fractions can be used to provide a sense of the size of that fraction (provided it is not too small). In the document `VisualFractions.pdf`, you will find an introduction to some graphical primitives that will allow you to draw some visual fractions using circles (the `VisualFractions.Rmd` contains the code in an Rmarkdown file which you might find helpful).

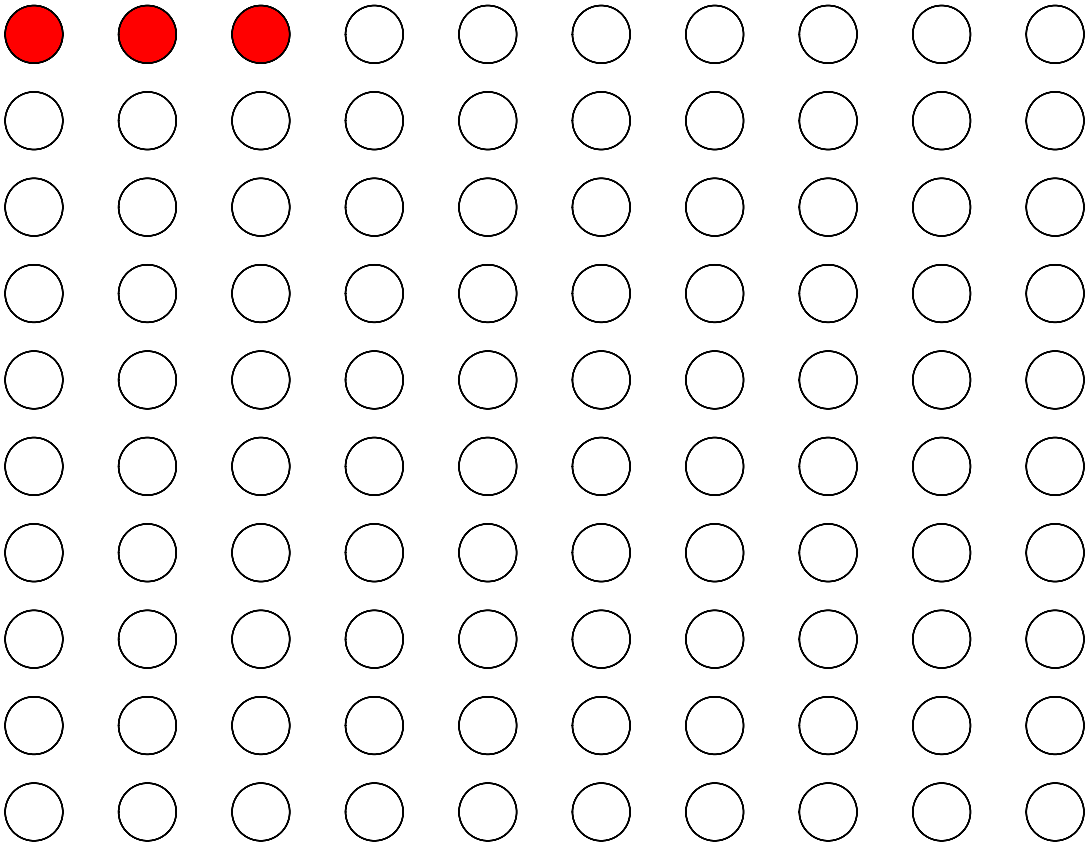
(a) (10 marks) Read that document and complete the definition of the function `visualFraction`(· · ·).

```r
visualFraction <- function(num, # the numerator
                           den, # the denominator
                           numCol="red",
                           # numerator colour
                           denCol="white",
                           # denominator colour
                           random=FALSE,
                           # a logical indicating
                           # whether the numerator values
                           # are to appear at random
                           # locations (if TRUE) or not.
                           ncols = NULL
                           # number of columns to be
                           # used in the array
) {
  # begin with some error checking
  #
  #  Check the logical
  if (!is.logical(random))
    stop(paste("random must be TRUE or FALSE, not:",
               random))
  #
  #  Check the numerator
  if (!is.numeric(num))
    stop(paste("num must be a number, not", num))
  if (length(num) != 1)
    stop(paste("num must be a single number, not of length",
               length(num)))
  if (floor(num) != num | num < 0 )
    stop(paste("num must be a non-negative integer, not",
               num))
  #
  #  Check the denominator
  if (!is.numeric(den))
    stop(paste("den must be a number, not", den))
  if (length(den) != 1)
    stop(paste("den must be a single number, not of length",
               length(den)))
  if (floor(den) != den | den < 0 )
    stop(paste("den must be a non-negative integer, not",
               den))
  #
  #  Check both
  if (num > den)
    stop(paste("num =", num, "> den =", den))
  #
```

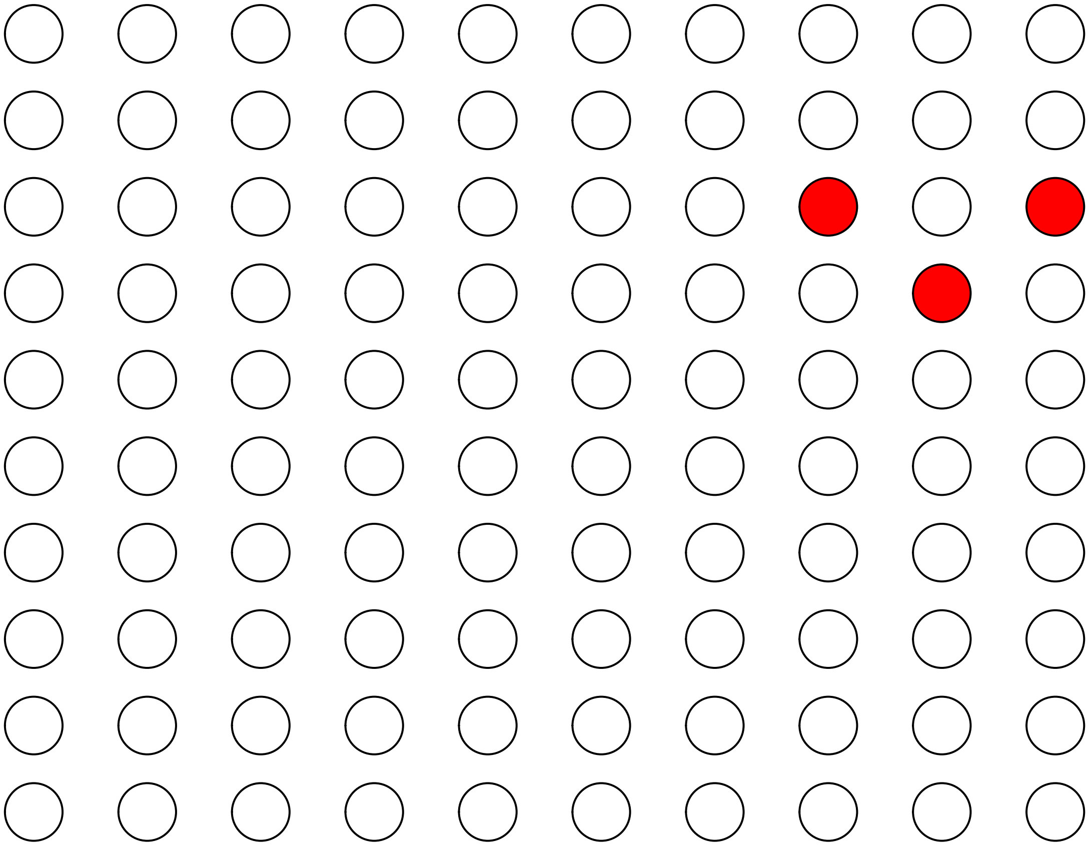
```
#  Check ncols
#
#  Default is NULL, so if user doesn't supply one let's
#  try to make it close to square (default more cols than rows)
if (is.null(ncols)) ncols <- ceiling(sqrt(den))

#  Now check any user supplied value for ncols
if (!is.numeric(ncols))
  stop(paste("ncols must be a number, not", ncols))
if (length(ncols) != 1)
  stop(paste("ncols must be a single number, not of length",
             length(ncols)))
if (floor(ncols) != ncols | ncols < 0 )
  stop(paste("ncols must be a non-negative integer, not",
             ncols))
if (ncols > den )
  stop(paste("ncols =", ncols,"> den =", den))

## If we have ncols columns, we will need
## nrows rows where
nrows <- ceiling(den/ncols)

## We'll also need a radius
## This is size provides spacing for most
radius <- 1/(2*(max(nrows,ncols)+5))

##
## Now it's your turn
## The display should be an nrows x ncols array of den circles
##
## If random=FALSE, the first num circles (from the top left of the
## array and proceeding left to right, then top to bottom)
## should be coloured numCol, the remainder coloured denCol.
##
## If random=TRUE, num circles selected at random in the array
## should be coloured numCol, the remainder denCol.
##
## That is, if we index the array 1 to den from top left by row to bottom
## right, the indices we would need to colour numCol would be
if (random) {indices <- sample(1:den, num)} else {indices <- 1:num}
##
## INSERT YOUR CODE BELOW:

## calculating the coordinates of the centers of the circles
centers = xy2grid(1:ncols, nrows:1)
# dividing to make get points within the unit square
centers[,1] = centers[,1]/(ncols+1)
centers[,2] = (2*centers[,2] - 1)/(2*nrows)

## generating the display
grid.newpage()
for (i in 1:den) {
  if(i %in% indices) {col = numCol} else {col = denCol}
  xcoord = centers[i,1]
  ycoord = centers[i,2]
  grid.circle(x=xcoord,
                y=ycoord,
                r=radius,
                gp=gpar(fill=col))
  }
}
```

(b) Show the results of your program on $\frac{3}{100}$ and also on $\frac{37}{1000}$. In each case, show the results **both when** random = FALSE **and when** random = TRUE.

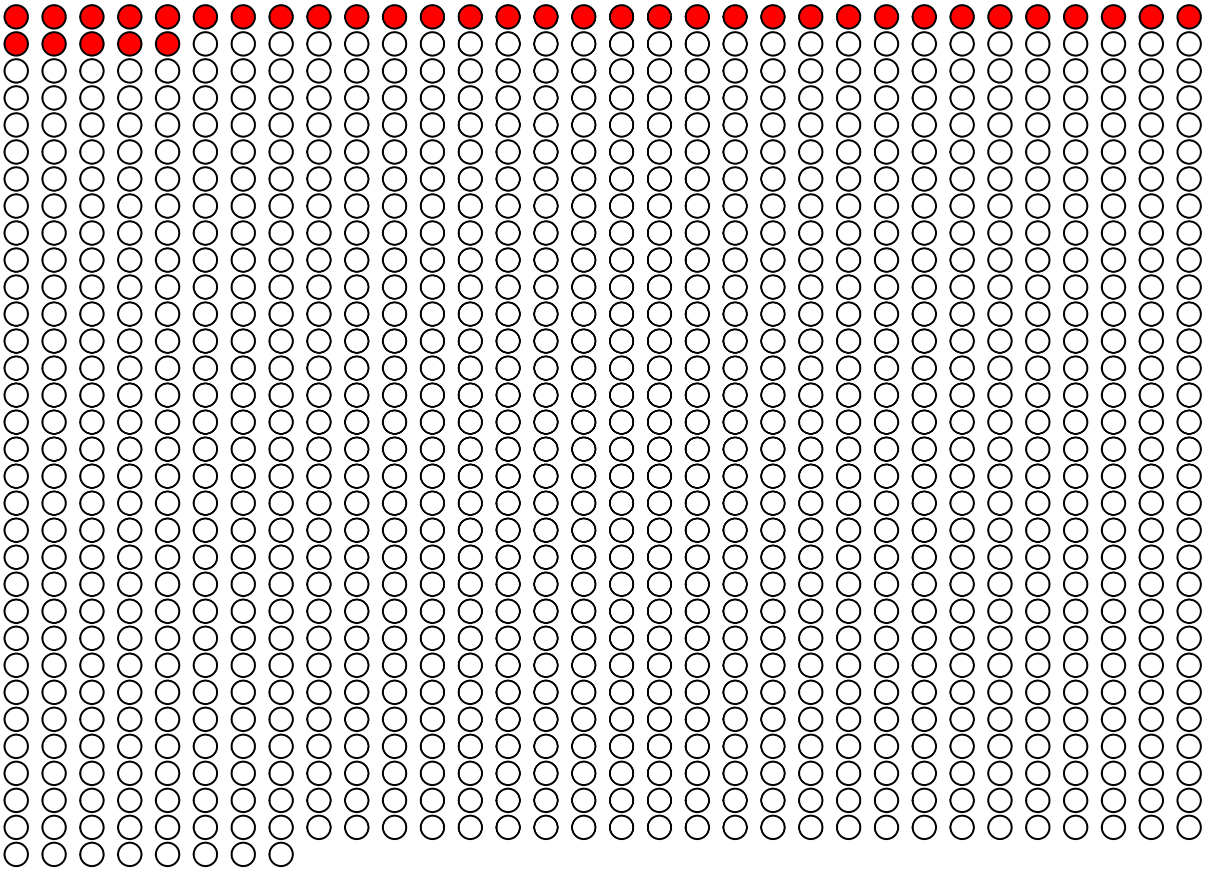
```
# results for 3/100
visualFraction(3,100)
```



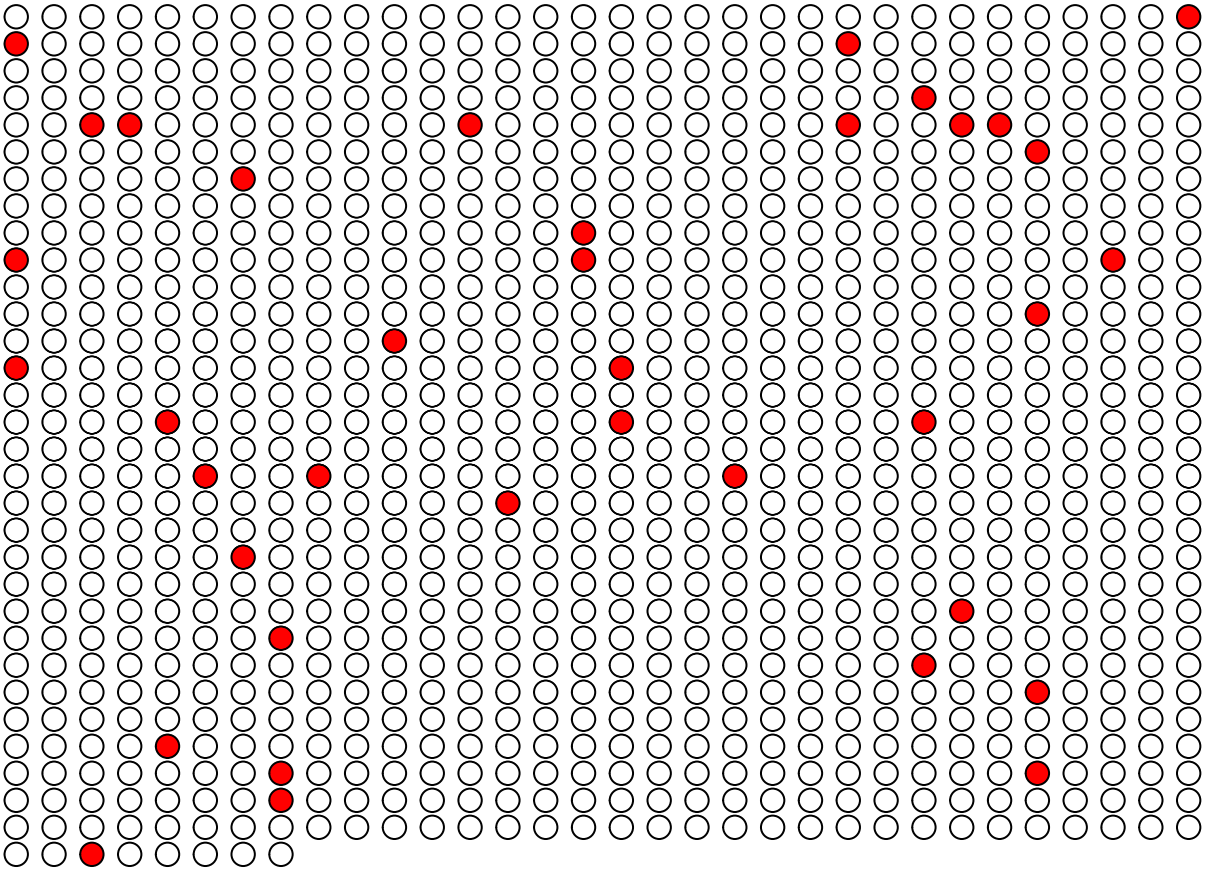
```
visualFraction(3,100, random=TRUE)
```



```r
# results for 37/1000
visualFraction(37,1000)
```



```
visualFraction(37,1000, random=TRUE)
```



(c) (2 marks) Explain why the case `random` = `TRUE` might be of interest.

For larger denominators, having the overplotted circles be randomly spread out makes it easier to quickly visually evaluate/approximate what ratio the original fraction was than having them all grouped out at the top of the diagram.

2. In class, a time scale was used to indicate the average time it would take to first win Lotto 649, purchasing 1 ticket per weekly draw. Assume that a winning ticket is one which matches the 6 numbers drawn from 1 to 49.

(a) (1 mark) Suppose p is the probability of winning the grand prize. Write down the value for p for Lotto 649. The numbers are being drawn without replacement and their order does not matter, so the number of possible combinations is:

$$\begin{aligned} \tbinom{49}{6} &= \tfrac{49!}{6!(49-6)!} \\ &= \tfrac{49!}{6!43!} \\ &= 13,983,816 \end{aligned}$$

$\therefore p = \frac{1}{13,983,816}$

(b) (1 mark) Write down the probability of winning (**for the first time**) on the nth draw (i.e. losing on the first $n-1$ draws). Let X be a random variable that represents the number of draws up to and including the draw in which the player wins for the first time. Then:

$$X \sim Geo(p) \qquad where \;\; p = \frac{1}{13,983,816}$$

Therefore, the probability of winning (**for the first time**) on the nth draw is:

$$\begin{aligned} P(X=n) &= p(1-p)^{n-1} \\ &= \tfrac{13{,}983{,}815^{n-1}}{13{,}983{,}816^{n}} \\ &= \tfrac{1}{13{,}983{,}815} \cdot (\tfrac{13{,}983{,}815}{13{,}983{,}816})^{n} \end{aligned}$$

(c) (1 mark) Determine the expected number of draws you must play (1 ticket each draw) before winning for the first time. As in (b), let X be a random variable that represents the number of draws up to and including the draw in which you wins for the first time. Then, the expected number of draws you must play (1 ticket each draw) before winning for the first time is $E(X)$. Since X follows a geometric distribution with probablity p,

$$\begin{aligned} E(X) &= \tfrac{1}{p} \\ &= 13,983,816 \end{aligned}$$

$\therefore$ $13,983,816$ is the expected number of draws you must play before winning for the first time.

(d) (1 mark) Show how the average time to win Lotto 649 when playing 1 ticket per weekly 649 draw turns into the long wait given for the Homo sapiens example (as described in the slides).

3. Colour blindness:

a. (3 marks) How well do each of these palettes work for those unable to see colour at all? Why?

The image below shows the monochromatic view of the original palettes (generated by the given website).

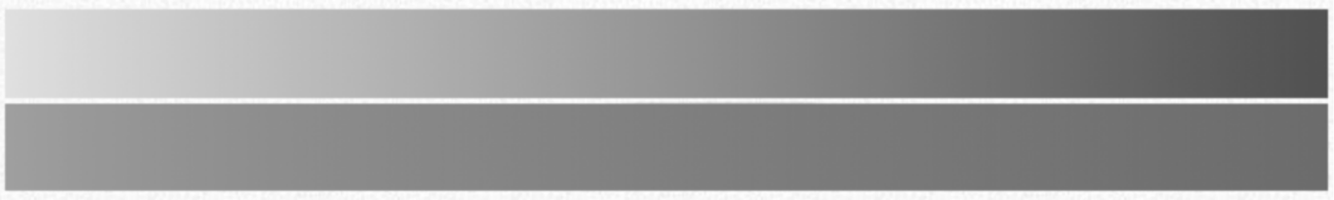


Figure 1: Monochromatic View of Palettes

Though both are grey palettes with no varying hue, the **yellow to blue** palette works better for those unable to see color.

This is because this palette has opposing, distinguishable **saturation**, which increases from the start to the end of the palette. Hence, those unable to see color can use saturation instead of hue to distinguish between categories.

The second palette (the green to red one) however, has very little, almost-unnoticeable variation in the saturation of the grey, so there is no way that anyone with monochromatic vision will be able to tell apart various categories.

b. (3 marks) Which oppositional colour pair seems best over all – yellow-blue or green-red? Explain your answer.

Overall, to benefit the most amount of people, it would be best to use the **green-red** palette.

- No color blindness: both are good as the opposing hues are distinguishable.
- Protanopia: 2nd palette is worse (becomes a dark yellow palette with very little saturation variety); first is still a yellow blue palette with distinguishable hues
- Deuteranopia: Same result as with protanopia, but 2nd palette is a dark orange palette and 1st is now an orange blue palette
- Tritanopia: Both palettes have opposing colors on their opposite ends, but 2nd is better as there is a higher range of constrasting hue along the palette because the red opposes the blue more than the lavender does.

While the green-red palette is the worse option of the two palettes for people with monochromatic vision, it is an extremely rare condition - it occurs 1 in 33000 people according to colourblindawareness.org. Hence, unless a substantial subset of the target viewers of the graphics have monochromatic vision, it is overall better to use the **green-red** palette.

c. (4 marks) Given what we discussed in class about the photo-receptors in the human retina, what characteristic of the photo-receptors might explain your choice in part (b)? The medium and long cones are both able to absorb green and red light to different degrees, whereas the short cones are able to absorb purple, blue and green light. Hence, most of the cones in the human eye can absorb green and red color, and the colors in between those wavelengths, so humans would be better able to perceive the colors in the green-red palette.

4. Visual representations.

(a) The following diagram was produced by the World Health Organization few years back.

i. (4 marks) Suppose the percentages were not actually displayed. What visual features of the display are available to the reader to decode the numbers presented? Which of these correspond to the percentages?

The **size** of the drops and their **colors/designs** can be used by the reader to decode the presented numbers.
The color of the drops tells the reader what category of water (satisfactory or unsatisfactory) that the percentages would be representing. Water is typically depicted as being a vibrant blue, so readers would likely be able to use these differences in saturation and design to distinguish between the categories that each drop represented. The more-saturated blue drop would be decoded as satisfactory water, and the less-saturated, pink-dotted drop would be decoded as unsatisfactory water.
As can be seen in the original diagram, the size of the drop positively correlates with the percentage displayed. A reader can infer that a bigger drop indicates a higher percentage, especially since the scale is the same for both years and for both categories displayed.

ii. (4 marks) From Cleveland and McGill's ordering of elementary tasks identify which of these tasks are used where in this diagram. Which of these elementary tasks are most likely to be used in decoding the numbers by the reader? Comment on the likely accuracy of this decoding.

- **Position on identical but nonaligned scales**: this is used to judge the size of the drops, and hence the magnitude of the numbers/percentages. It is somewhat accuragte for this decoding, as the relative positioning of the drop tells the user how large the magnitude of the numbers are (the higher positioned the drop is, the larger its corresponding percentage's magnitiude.) However, this is only clear when used with the elementary task of judging area.
- **Area**: this is used with the size of drops. It is the most accurate decoding for the magnitude of the numbers, as it is easy to quickly percieve the difference in the area of the drops, and thus decode what percentage the drop may represent.
- **Color hue**: this is used in the design of the drops. It is the most accurate decoding for the category/type of water source being described, as it is the colors being used - blue vs brown and blue with pink dots - have respectively positive and negative connotation to them. Hence, it is easy for readers to quickly decode what category the drops and their numbers correspond to.

iii. (4 marks) Create a table representing the same numerical information. Give the pros and cons of the table compared to the original diagram. (See `Tables.Rmd` in the folder for this assignment)

```r
library(kableExtra)
cleanWater = c(82, 66)
waterData = data.frame(c('Satisfactory', 'Unsatisfactory'), rbind(cleanWater, 100-cleanWater))
colnames(waterData) = c('', "(%)", "(%)")
rownames(waterData) = NULL
kable(waterData, "latex", booktabs = T) %>%
  kable_styling() %>%
  add_header_above(c("Water Source" = 1, "2000" = 1, "2015" = 1)) %>%
  add_header_above(c(" " = 1, "Year" = 2))
```

	Year	
Water Source	2000	2015
	(%)	(%)
Satisfactory	82	66
Unsatisfactory	18	34

The table is easier and faster to interpret for readers as there are no visual cues to decode. On the other hand, it is less aesthetically pleasing and the information at the top of the original diagram is lost.

(b) The following diagram was produced with considerable enthusiasm from a site called premiermicrosoft.wordpress.com https://premiermicrosoft.wordpress.com/2012/02/12/how-to-make-a-graphchart-in-microsoft-word/index.htm (not to be confused with Microsoft's https://premier.microsoft.com) where it is called a "cone chart".

i. (5 marks) Write out all of the values that appear in the diagram which are categorical and say how they are encoded in the diagram.

- **Activity during start up**: the activities that occur during program start up are categorical values. They are also encoded by their relative position along the scale at the back of the plot. Each column of cones corresponds, from left to right, to the ”Launching”, ”Splash Screen”, ”Loading”, and ”Processing” activities respectively.
- **Programs**: the type of program is a categorical value. The program type is encoded by the **saturation** of the purple hue of the cones. They are also encoded by their relative position along the scale at the left of the plot. Each row of cones corresponds, from bottom to top, to Program A, B, and C respectively, with the saturation increasing as you go down the rows from the top.

ii. (3 marks) Consider how the values of `Time for Programs to Start Up` encoded. From Cleveland and McGill's ordering of elementary tasks identify which of these tasks are used in the encoding of these values. Which elementary tasks are most likely to be used in decoding the values by the reader? Comment on the likely accuracy of this decoding.

- **Position along a common scale**: Each category of activity, and each program type is positioned along common scales - the left and back scales respectively. For both the activities and program types, this is the most likely and most accurate task to be used by readers for decoding. This is due to the clear spacing and separation of the cones in each row and category.
- **Length**: The length of time that each of the program activities takes is encoded in the diagram as the **height**, or vertical length of the cones. This is the most likely task to be used in decoding the values, as well as the most accurate. A higher height clearly indicates a longer run time to the reader, and it is easy to differentiate between the lengths of the cones.
- **Volume**: The length of time that each of the program activities takes is also encoded in the diagram as the **volume** of the cones. This is the least likely task to be used in decoding the values, as well as the least accurate of the two options. A higher volume does indicate a longer run time to the reader, but the area of the base of the cones is not clear as the only quantitative scale available is for the height of the cone. Hence, it is difficult for readers to accurately determine what the volumes are and which cones have a higher volume than others.
- **Color Saturation**: Each program is encoded by a certain saturation of purple in the cones. Compared to positioning along the scale, it is the less likely and accurate way of decoding the program category that the cones represent. The saturation for program B and A are very similar, so a reader would be unlikely to quickly recognize that as a way to differentiate between the program categories, or they would perceive the last two rows of cones as corresponding to the same program type.

iii. (2 marks) Critically assess the contribution of the scales appearing at the left and back of the plot.

The scales to the left are useful for the user as they provide a way for users to determine what the height of the cones, and hence the program run times, as this scale is a numerical one. The scale at the back contributes nothing to the plot however, as there are no values on the scale.

5. Tables are an important way to display symbolic numbers. In the document `Tables.pdf` (and perhaps more importantly its source file `Tables.Rmd`) you will find some examples of manipulating tables using the `knitr` `R` package. Please consult those files (downloading them and opening them from RStudio) and familiarize yourself with the material found there. It will be very helpful to you in undertaking the analysis in this questions.

(a) (10 marks) Reformat this table to make whatever patterns it contains more easily apprehended. Show each step that you choose to follow by displaying the table that results from each step. Say why you chose to make that step by referencing the rules we had for reformatting tables. Write down a summary of whatever patterns you have uncovered.

```r
setwd("/Users/azoachaforcheh/Documents/Waterloo/F17/stat442/a2/data")
data = read.csv("aboriginal.csv")
new_table = data
rownames(new_table)[1] = "Aboriginal.identity"

# Rule: Numbers that vary the least should appear in columns.
kable(t(new_table))
```

	Aboriginal.identity	North.American.Indian	Métis	Inuit	Non.aboriginal
CA	3.753989	2.2343213	1.2476541	0.1615824	96.24599
NL	4.685284	1.5511076	1.2924232	0.9418509	95.31571
PE	1.289073	0.9127827	0.2868746	0.0223539	98.71093
NS	2.676920	1.6875394	0.8504136	0.0359876	97.32363
NB	2.452581	1.7209755	0.5933440	0.0257069	97.54672
PQ	1.458128	0.8752801	0.3762824	0.1472585	98.54180
ON	2.015938	1.3167876	0.6119016	0.0169176	97.98410
MB	15.473549	8.8785768	6.3347199	0.0498450	84.52601
SK	14.875504	9.5822194	5.0448184	0.0225402	85.12450
AB	5.784535	2.9872357	2.6254816	0.0494418	94.21546
BC	4.812383	3.1803573	1.4589932	0.0195121	95.18762
YT	25.107651	20.8015899	2.6498841	0.8446505	74.90891
NT	50.255723	30.7842182	8.7189479	10.1315149	49.73210
NU	84.961637	0.3410060	0.4433078	84.0068201	15.02131

```r
# Rule: Use memorable self-explanatory labels and names.
# Change from abbreviated province name to full name
nprovinces = ncol(new_table)
for (i in 1:nprovinces) {
  abbr = colnames(new_table)[i]
  if (abbr == "CA") {colnames(new_table)[i] = "Canada"}
  if (abbr == "NL") {colnames(new_table)[i] = "Newfoundland and Labrador"}
  if (abbr == "PE") {colnames(new_table)[i] = "Prince Edward Island"}
  if (abbr == "NS") {colnames(new_table)[i] = "Nova Scotia"}
  if (abbr == "NB") {colnames(new_table)[i] = "New Brunswick"}
  if (abbr == "PQ") {colnames(new_table)[i] = "Quebec"}
  if (abbr == "ON") {colnames(new_table)[i] = "Ontario"}
  if (abbr == "MB") {colnames(new_table)[i] = "Manitoba"}
  if (abbr == "SK") {colnames(new_table)[i] = "Saskatchewan"}
  if (abbr == "AB") {colnames(new_table)[i] = "Alberta"}
  if (abbr == "BC") {colnames(new_table)[i] = "British Columbia"}
  if (abbr == "YT") {colnames(new_table)[i] = "Yukon"}
  if (abbr == "NT") {colnames(new_table)[i] = "Northwest Territories"}
  if (abbr == "NU") {colnames(new_table)[i] = "Nunavut"}
}
```

```r
kable(t(new_table))
```

	Aboriginal.identity	North.American.Indian	Métis	Inuit	Non.aboriginal
Canada	3.753989	2.2343213	1.2476541	0.1615824	96.24599
Newfoundland and Labrador	4.685284	1.5511076	1.2924232	0.9418509	95.31571
Prince Edward Island	1.289073	0.9127827	0.2868746	0.0223539	98.71093
Nova Scotia	2.676920	1.6875394	0.8504136	0.0359876	97.32363
New Brunswick	2.452581	1.7209755	0.5933440	0.0257069	97.54672
Quebec	1.458128	0.8752801	0.3762824	0.1472585	98.54180
Ontario	2.015938	1.3167876	0.6119016	0.0169176	97.98410
Manitoba	15.473549	8.8785768	6.3347199	0.0498450	84.52601
Saskatchewan	14.875504	9.5822194	5.0448184	0.0225402	85.12450
Alberta	5.784535	2.9872357	2.6254816	0.0494418	94.21546
British Columbia	4.812383	3.1803573	1.4589932	0.0195121	95.18762
Yukon	25.107651	20.8015899	2.6498841	0.8446505	74.90891
Northwest Territories	50.255723	30.7842182	8.7189479	10.1315149	49.73210
Nunavut	84.961637	0.3410060	0.4433078	84.0068201	15.02131

```r
# Rule: Reduce number of digits.
new_tab2 = round(new_table, digits = 1)
kable(t(new_tab2))
```

	Aboriginal.identity	North.American.Indian	Métis	Inuit	Non.aboriginal
Canada	3.8	2.2	1.2	0.2	96.2
Newfoundland and Labrador	4.7	1.6	1.3	0.9	95.3
Prince Edward Island	1.3	0.9	0.3	0.0	98.7
Nova Scotia	2.7	1.7	0.9	0.0	97.3
New Brunswick	2.5	1.7	0.6	0.0	97.5
Quebec	1.5	0.9	0.4	0.1	98.5
Ontario	2.0	1.3	0.6	0.0	98.0
Manitoba	15.5	8.9	6.3	0.0	84.5
Saskatchewan	14.9	9.6	5.0	0.0	85.1
Alberta	5.8	3.0	2.6	0.0	94.2
British Columbia	4.8	3.2	1.5	0.0	95.2
Yukon	25.1	20.8	2.6	0.8	74.9
Northwest Territories	50.3	30.8	8.7	10.1	49.7
Nunavut	85.0	0.3	0.4	84.0	15.0

```r
# Rule: Use averages (or medians) to help focus the eye over the array.
# adding the row averages
new_tab3 = rbind(new_tab2, colMeans(new_tab2))
rownames(new_tab3) = c(rownames(new_tab2), "Ave.")
kable(t(new_tab3))
```

	Aboriginal.identity	North.American.Indian	Métis	Inuit	Non.aboriginal	Ave.
Canada	3.8	2.2	1.2	0.2	96.2	20.72
Newfoundland and Labrador	4.7	1.6	1.3	0.9	95.3	20.76
Prince Edward Island	1.3	0.9	0.3	0.0	98.7	20.24
Nova Scotia	2.7	1.7	0.9	0.0	97.3	20.52
New Brunswick	2.5	1.7	0.6	0.0	97.5	20.46
Quebec	1.5	0.9	0.4	0.1	98.5	20.28
Ontario	2.0	1.3	0.6	0.0	98.0	20.38
Manitoba	15.5	8.9	6.3	0.0	84.5	23.04

	Aboriginal.identity	North.American.Indian	Métis	Inuit	Non.aboriginal	Ave.
Saskatchewan	14.9	9.6	5.0	0.0	85.1	22.92
Alberta	5.8	3.0	2.6	0.0	94.2	21.12
British Columbia	4.8	3.2	1.5	0.0	95.2	20.94
Yukon	25.1	20.8	2.6	0.8	74.9	24.84
Northwest Territories	50.3	30.8	8.7	10.1	49.7	29.92
Nunavut	85.0	0.3	0.4	84.0	15.0	36.94

```r
# adding the column averages
new_tab4 = cbind(new_tab3, rowMeans(new_tab3))
colnames(new_tab4) = c(colnames(new_tab3), "Average")
new_tab4 = round(new_tab4, digits = 1)
kable(t(new_tab4))
```

	Aboriginal.identity	North.American.Indian	Métis	Inuit	Non.aboriginal	Ave.
Canada	3.8	2.2	1.2	0.2	96.2	20.7
Newfoundland and Labrador	4.7	1.6	1.3	0.9	95.3	20.8
Prince Edward Island	1.3	0.9	0.3	0.0	98.7	20.2
Nova Scotia	2.7	1.7	0.9	0.0	97.3	20.5
New Brunswick	2.5	1.7	0.6	0.0	97.5	20.5
Quebec	1.5	0.9	0.4	0.1	98.5	20.3
Ontario	2.0	1.3	0.6	0.0	98.0	20.4
Manitoba	15.5	8.9	6.3	0.0	84.5	23.0
Saskatchewan	14.9	9.6	5.0	0.0	85.1	22.9
Alberta	5.8	3.0	2.6	0.0	94.2	21.1
British Columbia	4.8	3.2	1.5	0.0	95.2	20.9
Yukon	25.1	20.8	2.6	0.8	74.9	24.8
Northwest Territories	50.3	30.8	8.7	10.1	49.7	29.9
Nunavut	85.0	0.3	0.4	84.0	15.0	36.9
Average	15.7	6.2	2.3	6.9	84.3	23.1

```r
# Rule: Rearrange columns so that averages are strictly decreasing (or increasing) from left to right.
roworder = c(order(rowMeans(new_tab3[1:5,]), decreasing = TRUE),6)
new_tab4 = new_tab4[roworder,]
kable(t(new_tab4), align="rrrrrc")
```

	Non.aboriginal	Aboriginal.identity	Inuit	North.American.Indian	Métis	Ave.
Canada	96.2	3.8	0.2	2.2	1.2	20.7
Newfoundland and Labrador	95.3	4.7	0.9	1.6	1.3	20.8
Prince Edward Island	98.7	1.3	0.0	0.9	0.3	20.2
Nova Scotia	97.3	2.7	0.0	1.7	0.9	20.5
New Brunswick	97.5	2.5	0.0	1.7	0.6	20.5
Quebec	98.5	1.5	0.1	0.9	0.4	20.3
Ontario	98.0	2.0	0.0	1.3	0.6	20.4
Manitoba	84.5	15.5	0.0	8.9	6.3	23.0
Saskatchewan	85.1	14.9	0.0	9.6	5.0	22.9
Alberta	94.2	5.8	0.0	3.0	2.6	21.1
British Columbia	95.2	4.8	0.0	3.2	1.5	20.9
Yukon	74.9	25.1	0.8	20.8	2.6	24.8
Northwest Territories	49.7	50.3	10.1	30.8	8.7	29.9
Nunavut	15.0	85.0	84.0	0.3	0.4	36.9
Average	84.3	15.7	6.9	6.2	2.3	23.1

```
# Rule: Note dramatically exceptional values and exclude them from pattern summary calculations.
new_tab5 = new_tab4
new_tab5['Non.aboriginal','Average'] = (sum(new_tab4['Non.aboriginal',])-49.70-15)/
  (length(new_tab4['Non.aboriginal',])-2)
new_tab5['Inuit','Average'] = (sum(new_tab4['Inuit',], 1, 0)-10.1-84)/
  (length(new_tab4['Inuit',])-2)
new_tab5['North.American.Indian','Average'] = (sum(new_tab4['North.American.Indian',])-20.8-30.8)/
  (length(new_tab4['North.American.Indian',])-2)
new_tab5['Aboriginal.identity','Average'] = (sum(new_tab4['Aboriginal.identity',])-50.3-85.0)/
  (length(new_tab4['Aboriginal.identity',])-2)

newRowOrder = c(order(new_tab5[1:5,'Average'], decreasing = TRUE),6)
new_tab5 = round(new_tab5[newRowOrder,], digits = 1)

final_tab = new_tab5
final_tab['Non.aboriginal','Average'] =
  paste0(final_tab['Non.aboriginal','Average'], '*')
final_tab['Inuit','Average'] =
  paste0(final_tab['Inuit','Average'], '*')
final_tab['North.American.Indian','Average'] =
  paste0(final_tab['North.American.Indian','Average'], '*')
final_tab['Aboriginal.identity','Average'] =
  paste0(final_tab['Aboriginal.identity','Average'], '*')
final_tab['Ave.','Average'] =
  paste0(final_tab['Ave.','Average'], '*')

final_tab['Non.aboriginal', 'Northwest Territories'] =
  paste0('(', final_tab['Non.aboriginal', 'Northwest Territories'] ,')')
final_tab['Non.aboriginal', 'Nunavut'] =
  paste0('(', final_tab['Non.aboriginal', 'Nunavut'] ,'.0)')
final_tab['Inuit', 'Northwest Territories'] =
  paste0('(', final_tab['Inuit', 'Northwest Territories'] ,')')
final_tab['Inuit', 'Nunavut'] =
  paste0('(', final_tab['Inuit', 'Nunavut'] ,')')
final_tab['North.American.Indian', 'Northwest Territories'] =
  paste0('(', final_tab['North.American.Indian', 'Northwest Territories'] ,')')

final_tab['North.American.Indian', 'Yukon'] =
  paste0('(', final_tab['North.American.Indian', 'Yukon'] ,')')
final_tab['Aboriginal.identity', 'Northwest Territories'] =
  paste0('(', final_tab['Aboriginal.identity', 'Northwest Territories'] ,')')
final_tab['Aboriginal.identity', 'Nunavut'] =
  paste0('(', final_tab['Aboriginal.identity', 'Nunavut'] ,'.0)')

final_tab['Ave.', 'Northwest Territories'] =
  paste0('(', final_tab['Ave.', 'Northwest Territories'] ,')')
final_tab['Ave.', 'Nunavut'] =
  paste0('(', final_tab['Ave.', 'Nunavut'] ,')')
final_tab['Ave.', 'Yukon'] =
  paste0('(', final_tab['Ave.', 'Yukon'] ,')')

kable(t(final_tab), align="rrrrrc")
```

	Non.aboriginal	Aboriginal.identity	North.American.Indian	Métis	Inuit	Ave.
Canada	96.2	3.8	2.2	1.2	0.2	20.7
Newfoundland and Labrador	95.3	4.7	1.6	1.3	0.9	20.8

	Non.aboriginal	Aboriginal.identity	North.American.Indian	Métis	Inuit	Ave.
Prince Edward Island	98.7	1.3	0.9	0.3	0.0	20.2
Nova Scotia	97.3	2.7	1.7	0.9	0.0	20.5
New Brunswick	97.5	2.5	1.7	0.6	0.0	20.5
Quebec	98.5	1.5	0.9	0.4	0.1	20.3
Ontario	98.0	2.0	1.3	0.6	0.0	20.4
Manitoba	84.5	15.5	8.9	6.3	0.0	23.0
Saskatchewan	85.1	14.9	9.6	5.0	0.0	22.9
Alberta	94.2	5.8	3.0	2.6	0.0	21.1
British Columbia	95.2	4.8	3.2	1.5	0.0	20.9
Yukon	74.9	25.1	(20.8)	2.6	0.8	(24.8)
Northwest Territories	(49.7)	(50.3)	(30.8)	8.7	(10.1)	(29.9)
Nunavut	(15.0)	(85.0)	0.3	0.4	(84)	(36.9)
Average	92.3*	7.7*	3.2*	2.3	0.8*	23.1*

In summary, there are:

- 2 exceptionally low values for Non-aboriginal people in the Northwest Territories and Nunavut (differing from their average by about 43 and 77
- 2 exceptionally high values for the Aboriginal identity population in the Northwest Territories and Nunavut (differing from their average by about 43 and 77
- 2 exceptionally high values for North American Indian people in Yukon and the Northwest Territories (differing from their average by about 18 and 28
- 2 exceptionally high values for Inuit people in the Northwest Territories and Nunavut (differing from their average by about 9 and 83

Overall, the Northwest Territories and Nunavut have exceptionally high percentages of Aboriginal people and exceptionally low percentages of Non-Aboriginal people compared to other provinces, and to the rest of the country.

(b) (4 marks) Note that the category `Aboriginal.identity.population` includes the "Aboriginal groups (North American Indian, Métis and Inuit), multiple Aboriginal responses and Aboriginal responses not included elsewhere". Replace the data on `Aboriginal.identity.population` by `Other.aboriginal` that is the difference between `Aboriginal.identity.population` and the North American Indian, Métis and Inuit groups. Again, give the table the best presentation and summarize whatever pattern exists.

```r
data3 = data
other = NULL
for (i in 1:ncol(data3)) {
  other = append(other, data3[,i][1] - sum(data3[,i][c(2:4)]))
}
data3[1,] = other
rownames(data3)[1] = "Other.aborginal"

# Change from abbreviated province name to full name
nprovinces = ncol(data3)
for (i in 1:nprovinces) {
  abbr = colnames(data3)[i]
  if (abbr == "CA") {colnames(data3)[i] = "Canada"}
  if (abbr == "NL") {colnames(data3)[i] = "Newfoundland and Labrador"}
  if (abbr == "PE") {colnames(data3)[i] = "Prince Edward Island"}
  if (abbr == "NS") {colnames(data3)[i] = "Nova Scotia"}
  if (abbr == "NB") {colnames(data3)[i] = "New Brunswick"}
  if (abbr == "PQ") {colnames(data3)[i] = "Quebec"}
  if (abbr == "ON") {colnames(data3)[i] = "Ontario"}
  if (abbr == "MB") {colnames(data3)[i] = "Manitoba"}
  if (abbr == "SK") {colnames(data3)[i] = "Saskatchewan"}
  if (abbr == "AB") {colnames(data3)[i] = "Alberta"}
  if (abbr == "BC") {colnames(data3)[i] = "British Columbia"}
  if (abbr == "YT") {colnames(data3)[i] = "Yukon"}
  if (abbr == "NT") {colnames(data3)[i] = "Northwest Territories"}
  if (abbr == "NU") {colnames(data3)[i] = "Nunavut"}
}

data3 = round(data3, digits = 1)

# adding the row averages
data_avgs = rbind(data3, colMeans(data3))
rownames(data_avgs) = c(rownames(data3), "Ave.")

# adding the column averages
data_avg = cbind(data_avgs, rowMeans(data_avgs))
colnames(data_avg) = c(colnames(data_avgs), "Average")
roworder = c(order(rowMeans(data_avgs[1:5,]), decreasing = TRUE),6)
data_avg = round(data_avg[roworder,], digits = 1)

data_avg['Non.aboriginal','Average'] = (sum(data_avg['Non.aboriginal',])-49.70-15)/
  (length(data_avg['Non.aboriginal',])-2)
data_avg['Inuit','Average'] = (sum(data_avg['Inuit',], 1, 0)-10.1-84)/
  (length(data_avg['Inuit',])-2)
data_avg['North.American.Indian','Average'] = (sum(data_avg['North.American.Indian',])-20.8-30.8)/
  (length(data_avg['North.American.Indian',])-2)

newRowOrder = c(order(data_avg[1:5,'Average'], decreasing = TRUE),6)
data_avg = round(data_avg[newRowOrder,], digits = 1)

final_tab2 = data_avg
final_tab2['Non.aboriginal','Average'] =
  paste0(final_tab2['Non.aboriginal','Average'], '*')
final_tab2['Inuit','Average'] =
  paste0(final_tab2['Inuit','Average'], '*')
final_tab2['North.American.Indian','Average'] =
  paste0(final_tab2['North.American.Indian','Average'], '*')
final_tab2['Ave.','Average'] =
  paste0(final_tab2['Ave.','Average'], '.0*')

final_tab2['Non.aboriginal', 'Northwest Territories'] =
  paste0('(', final_tab2['Non.aboriginal', 'Northwest Territories'] ,')')
final_tab2['Non.aboriginal', 'Nunavut'] =
  paste0('(', final_tab2['Non.aboriginal', 'Nunavut'] ,'.0)')
final_tab2['Inuit', 'Northwest Territories'] =
  paste0('(', final_tab2['Inuit', 'Northwest Territories'] ,')')
final_tab2['Inuit', 'Nunavut'] =
  paste0('(', final_tab2['Inuit', 'Nunavut'] ,'.0)')
final_tab2['North.American.Indian', 'Northwest Territories'] =
  paste0('(', final_tab2['North.American.Indian', 'Northwest Territories'] ,')')
final_tab2['North.American.Indian', 'Yukon'] =
  paste0('(', final_tab2['North.American.Indian', 'Yukon'] ,')')

final_tab2['Ave.', 'Northwest Territories'] =
  paste0('(', final_tab2['Ave.', 'Northwest Territories'] ,'.0)')
final_tab2['Ave.', 'Nunavut'] =
  paste0('(', final_tab2['Ave.', 'Nunavut'] ,'.0)')
final_tab2['Ave.', 'Yukon'] =
  paste0('(', final_tab2['Ave.', 'Yukon'] ,'.0)')

kable(t(final_tab2),  align="rrrrrc")
```

	Non.aboriginal	North.American.Indian	Métis	Inuit	Other.aborginal	Ave.
Canada	96.2	2.2	1.2	0.2	0.1	20.0
Newfoundland and Labrador	95.3	1.6	1.3	0.9	0.9	20.0
Prince Edward Island	98.7	0.9	0.3	0.0	0.1	20.0
Nova Scotia	97.3	1.7	0.9	0.0	0.1	20.0
New Brunswick	97.5	1.7	0.6	0.0	0.1	20.0
Quebec	98.5	0.9	0.4	0.1	0.1	20.0
Ontario	98.0	1.3	0.6	0.0	0.1	20.0
Manitoba	84.5	8.9	6.3	0.0	0.2	20.0
Saskatchewan	85.1	9.6	5.0	0.0	0.2	20.0
Alberta	94.2	3.0	2.6	0.0	0.1	20.0
British Columbia	95.2	3.2	1.5	0.0	0.2	20.0
Yukon	74.9	(20.8)	2.6	0.8	0.8	(20.0)
Northwest Territories	(49.7)	(30.8)	8.7	(10.1)	0.6	(20.0)
Nunavut	(15.0)	0.3	0.4	(84.0)	0.2	(20.0)
Average	92.3*	3.2*	2.3	0.8*	0.3	20.0*

In summary, there are:

- 2 exceptionally low values for Non-aboriginal people in the Northwest Territories and Nunavut (differing from their average by about 43 and 77
- 2 exceptionally high values for North American Indian people in Yukon and the Northwest Territories (differing from their average by about 18 and 28
- 2 exceptionally high values for Inuit people in the Northwest Territories and Nunavut (differing from their average by about 9 and 83

Overall, the Northwest Territories and Nunavut have exceptionally high percentages of Aboriginal people and exceptionally low percentages of Non-Aboriginal people compared to other provinces, and to the rest of the country.

(c) (4 marks) Whatever marginal (row or column) pattern you identified in the previous part, build the table of deviations from that pattern, display it, and comment on what you see.

```r
data_deviations = data_avg
for (col in colnames(data_avg)) {
  data_deviations[,col] = data_avg[,col] - data_avg[,'Average']
}

final_tab3 = data_deviations
final_tab3['Non.aboriginal','Average'] =
  paste0(final_tab3['Non.aboriginal','Average'], '*')
final_tab3['Inuit','Average'] =
  paste0(final_tab3['Inuit','Average'], '*')
final_tab3['North.American.Indian','Average'] =
  paste0(final_tab3['North.American.Indian','Average'], '*')
final_tab3['Ave.','Average'] =
  paste0(final_tab3['Ave.','Average'], '*')

final_tab3['Non.aboriginal', 'Northwest Territories'] =
  paste0('(', final_tab3['Non.aboriginal', 'Northwest Territories'] ,')')
```

```r
final_tab3['Non.aboriginal', 'Nunavut'] =
  paste0('(', final_tab3['Non.aboriginal', 'Nunavut'] ,')')
final_tab3['Inuit', 'Northwest Territories'] =
  paste0('(', final_tab3['Inuit', 'Northwest Territories'] ,')')
final_tab3['Inuit', 'Nunavut'] =
  paste0('(', final_tab3['Inuit', 'Nunavut'] ,')')
final_tab3['North.American.Indian', 'Northwest Territories'] =
  paste0('(', final_tab3['North.American.Indian', 'Northwest Territories'] ,')')

final_tab3['North.American.Indian', 'Yukon'] =
  paste0('(', final_tab3['North.American.Indian', 'Yukon'] ,')')

final_tab3['Ave.', 'Northwest Territories'] =
  paste0('(', final_tab3['Ave.', 'Northwest Territories'] ,')')
final_tab3['Ave.', 'Nunavut'] =
  paste0('(', final_tab3['Ave.', 'Nunavut'] ,')')
final_tab3['Ave.', 'Yukon'] =
  paste0('(', final_tab3['Ave.', 'Yukon'] ,')')

kable(t(final_tab3), digits=1, align="rrrrrc")
```

	Non.aboriginal	North.American.Indian	Métis	Inuit	Other.aborginal	Ave.
Canada	3.9	-1.0	-1.1	-0.6	-0.2	0.0
Newfoundland and Labrador	3.0	-1.6	-1.0	0.1	0.6	0.0
Prince Edward Island	6.4	-2.3	-2.0	-0.8	-0.2	0.0
Nova Scotia	5.0	-1.5	-1.4	-0.8	-0.2	0.0
New Brunswick	5.2	-1.5	-1.7	-0.8	-0.2	0.0
Quebec	6.2	-2.3	-1.9	-0.7	-0.2	0.0
Ontario	5.7	-1.9	-1.7	-0.8	-0.2	0.0
Manitoba	-7.8	5.7	4.0	-0.8	-0.1	0.0
Saskatchewan	-7.2	6.4	2.7	-0.8	-0.1	0.0
Alberta	1.9	-0.2	0.3	-0.8	-0.2	0.0
British Columbia	2.9	0.0	-0.8	-0.8	-0.1	0.0
Yukon	-17.4	(17.6)	0.3	0	0.5	(0)
Northwest Territories	(-42.6)	(27.6)	6.4	(9.3)	0.3	(0)
Nunavut	(-77.3)	-2.9	-1.9	(83.2)	-0.1	(0)
Average	0*	0*	0	0*	0	0*

The row averages are almost all equal to each other, and hence there was no deviation from the total average.

6. **Graduate students** (bonus undergraduates): Suppose we have n-dimensional real and linearly independent vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$ and $\mathbf{y}$. The vector $\mathbf{y}$ is the sum of two n-dimensional real vectors μ and $\mathbf{r}$

$$\mathbf{y} = \mu + \mathbf{r}$$

where μ is restricted to be a linear combination of the vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$. That is

$$\mu = \theta_1 \times \mathbf{x}_1 + \theta_2 \times \mathbf{x}_2 + \cdots + \theta_p \times \mathbf{x}_p$$

for some unknown real constants $\theta_1, \theta_2, \ldots, \theta_p$, or equivalently

$$\mu = \mathbf{X}\theta$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_p]$ is an $n \times p$ matrix and $\theta = (\theta_1, \theta_2, \ldots, \theta_p)^T$ is a $p \times 1$ vector.

(a) (5 marks) For any $\mathbf{y}$, neither μ nor $\mathbf{r}$ are uniquely defined. Suppose we choose particular vectors $\widehat{\mu}$, and $\widehat{\mathbf{r}}$ (with $\mathbf{y} = \widehat{\mu} + \widehat{\mathbf{r}}$) to be such that they are orthogonal to one another (whatever values any θ_i take). That is, $\widehat{\mu}^T\widehat{\mathbf{r}} = 0$.

Prove that this additional constraint implies that

$$\widehat{\mu} = \mathbf{P}\mathbf{y}$$

where $\mathbf{P} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ and hence show that $\widehat{\mathbf{r}} = (\mathbf{I}_n - \mathbf{P})\mathbf{y}$.

$\widehat{\mu}^T\widehat{\mathbf{r}} = 0$ (whatever values any θ_i take) $\implies$ $\widehat{\mathbf{r}}$ is orthogonal to $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$. Hence:

$$\begin{aligned}
\mathbf{X}^T\widehat{\mathbf{r}} &= \begin{bmatrix} \mathbf{x}_1^T\widehat{\mathbf{r}} \\ \cdots \\ \mathbf{x}_p^T\widehat{\mathbf{r}} \end{bmatrix} \\
&= \mathbf{0} \\
\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\widehat{\mathbf{r}} &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{0} \\
\mathbf{P}\widehat{\mathbf{r}} &= \mathbf{0} \\
\mathbf{P}\mathbf{y} - \mathbf{P}\widehat{\mathbf{r}} &= \mathbf{P}\mathbf{y} \\
\mathbf{P}\widehat{\mu} &= \mathbf{P}\mathbf{y}
\end{aligned}$$

But we can simplify $\mathbf{Pu}$ to:

$$\begin{aligned}
\mathbf{P}\mu &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mu \\
&= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\theta \\
&= \mathbf{X}\mathbf{I}_p\theta \\
&= \mathbf{X}\theta \\
&= \widehat{\mu}
\end{aligned}$$

$\therefore \widehat{\mu} = \mathbf{P}\mathbf{y}$. For the second proof:

$$\begin{aligned}
\mathbf{P}\mathbf{y} &= \widehat{\mu} \\
&= \mathbf{y} - \widehat{\mathbf{r}} \\
\widehat{\mathbf{r}} &= \mathbf{y} - \mathbf{P}\mathbf{y} \\
&= (\mathbf{I}_n - \mathbf{P})\mathbf{y}
\end{aligned}$$

(b) (2 marks) Show that $\mathbf{P}$ is an idempotent matrix, that is that $\mathbf{P}^2 = \mathbf{P}$.

$$\begin{aligned}
\mathbf{P}^2 &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \\
&= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{I}_p\mathbf{X}^T \\
&= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \\
&= \mathbf{P}
\end{aligned}$$

$\therefore$ $\mathbf{P}$ is an idempotent matrix.

(c) (2 marks) Show that if **P** is an idempotent matrix, then so must be $ **({I}__n -{P})$.**

Assume that **P** is an idempotent matrix.

$$\begin{aligned}
(\mathbf{I}_n - \mathbf{P})^2 &= (\mathbf{I}_n - \mathbf{P})(\mathbf{I}_n - \mathbf{P}) \\
&= \mathbf{I}_n^2 - \mathbf{I}_n\mathbf{P} - \mathbf{P}\mathbf{I}_n + \mathbf{P}^2 \\
&= \mathbf{I}_n - \mathbf{P} - \mathbf{P} + \mathbf{P} \qquad \text{since } \mathbf{P} \text{ is idempotent} \\
&= \mathbf{I}_n - \mathbf{P}
\end{aligned}$$

$\therefore (\mathbf{I}_n - \mathbf{P})$ is an idempotent matrix.

(d) (2 marks) Show that $\widehat{\mathbf{r}}$ is in fact orthogonal to $\widehat{\mu}$.

We first show that P is symmetric:

$$\begin{aligned}
\mathbf{P}^T &= (\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)^T \\
&= (\mathbf{X}^T)^T(\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1})^T \\
&= \mathbf{X}((\mathbf{X}^T\mathbf{X})^{-1})^T\mathbf{X}^T \\
&= \mathbf{X}((\mathbf{X}^T\mathbf{X})^T)^{-1}\mathbf{X}^T \\
&= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \\
&= \mathbf{P}
\end{aligned}$$

$\therefore$ $\mathbf{P}$ is symmetric.

$$\begin{array}{rcl}\widehat{\mu}^T\widehat{\mathbf{r}} & = & \widehat{\mu}\cdot\widehat{\mathbf{r}} \\ & = & \mathbf{Py}\cdot(\mathbf{I}_n-\mathbf{P})\mathbf{y} \\ & = & \mathbf{y}\cdot\mathbf{P}(\mathbf{I}_n-\mathbf{P})\mathbf{y} \\ & = & \mathbf{y}^T(\mathbf{P}-\mathbf{P}^2)\mathbf{y} \\ & = & \mathbf{y}^T\mathbf{0}\mathbf{y} \\ & = & \mathbf{0}\end{array}$$

$\therefore$ $\widehat{\mu}$ is orthogonal to $\widehat{\mathbf{r}}$.

(e) (5 marks) Show that $\widehat{\mu}=\mathbf{Py}$ is the choice of μ which minimizes the squared length of $\mathbf{r}$. That is it minimizes $\mathbf{r}^T\mathbf{r}=(\mathbf{y}-\mu)^T(\mathbf{y}-\mu)$.