Spatial statistics - Assignment #1

Edgar Manrique

3/14/2022

1. Generate a random, normally distributed sample of size n = 1000 (Mean = 50; Standard Deviation = 10)

```
set.seed(1)
x <- rnorm(1000, mean = 50, sd = 10)

(q1 <- quantile(x, .25))
```

```
##      25%
## 43.02627
```

```
(q3 <- quantile(x, .75))
```

```
##      75%
## 56.88428
```

```
(range <- range(x)[2] - range(x)[1])
```

```
## [1] 68.18325
```

```
(range <- max(x) - min(x))
```

```
## [1] 68.18325
```

```
(iqr <- IQR(x))
```

```
## [1] 13.85801
```

```
(iqr <- q3-q1)
```

```
##      75%
## 13.85801
```

1st quartile and 3rd quartile were generated using the function `quantile` and setting it to 0.25 and 0.75 respectively. Also, the `range` function has as an output the min a max values of a vector, thus, you can select the items accordingly to compute the range size, the same result would be accomplished by calculating max a min values independently and substracting them. Finally, the inter quartile range can be computed with the function `IQR`, the same result can be computed by substracting the 3rd and 1st quartiles.

2. Find the mean, median and range.

```r
x <- c(29, 35, 17, 30, 231, 6, 27, 35, 23, 29, 13)
(summ <- summary(x))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    6.00   20.00   29.00   43.18   32.50  231.00
```

```r
(range <- max(x) - min(x))
```

```
## [1] 225
```

The function `summary` computes a series of summary values from a vector, including the mean, median, 1st and 3rd quartiles and min and max values.

3. Find the grouped mean of the following data.

```r
((10*5) + (30+15) + (50*10) + (70*12)) / (5+15+10+12)
```

```
## [1] 34.16667
```

4. Ten migration distances corresponding to the distances moved by recent migrants are observed (in km): 43, 6, 7, 11, 122, 41, 21, 17, 1, 3. Find the mean and standard deviation. Interpret the result in your own words.

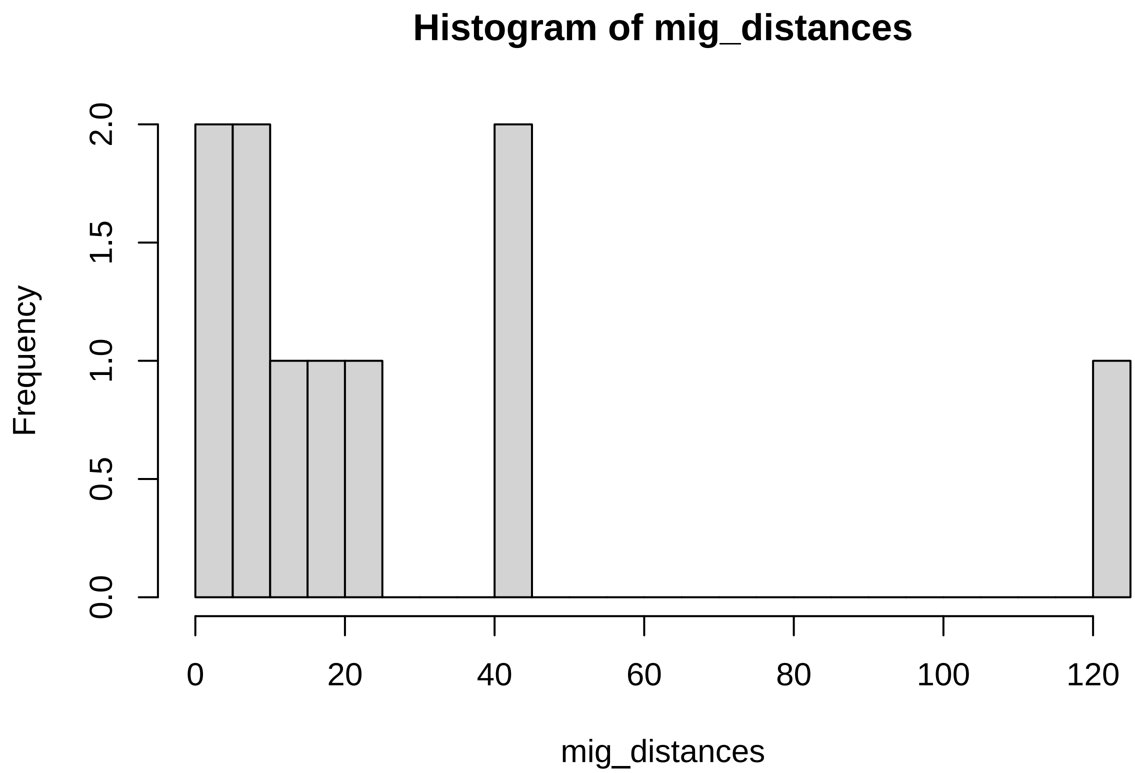
```r
mig_distances <- c(43, 6, 7, 11, 122, 41, 21, 17, 1, 3)
mean(mig_distances)
```

```
## [1] 27.2
```

```r
sd(mig_distances)
```

```
## [1] 36.45637
```

```r
hist(mig_distances, breaks = 20)
```



Migration distances are very dispersed in their distribution, making it that the standard deviation is greater than the mean. For this type of data, mobility in terms of distance, it could mean that the source of the migration comes from many different places. Although, if the source is the same place (e.g. same country) it could mean that the destination is very close to the borders of the country.

5. Calculate descriptive statistics.

```
(x <- read.csv('~/Desktop/Data_SimpleExercises1.csv'))
```

```
##    Item   X_i
## 1     1  96.8
## 2     2  95.9
## 3     3  92.6
## 4     4  98.7
## 5     5 106.4
## 6     6 124.5
## 7     7 107.1
## 8     8  58.1
## 9     9 101.8
## 10   10  96.0
## 11   11 102.6
## 12   12  64.8
## 13   13 117.3
## 14   14 104.4
## 15   15  98.4
```

```
## 16   16 126.0
## 17   17 115.3
## 18   18  91.3
## 19   19  99.9
## 20   20 115.0
```

```r
mean(x$X_i)
```

```
## [1] 100.645
```

```r
median(x$X_i)
```

```
## [1] 100.85
```

```r
quantile(x$X_i, .25)
```

```
##    25%
## 95.975
```

```r
quantile(x$X_i, .5)
```

```
##    50%
## 100.85
```

```r
quantile(x$X_i, .75)
```

```
##     75%
## 109.075
```

```r
IQR(x$X_i)
```

```
## [1] 13.1
```

```r
min(x$X_i)
```

```
## [1] 58.1
```

```r
max(x$X_i)
```

```
## [1] 126
```

```r
var(x$X_i)
```

```
## [1] 279.971
```

```
sd(x$X_i)
```

```
## [1] 16.73233
```

6. Describe the following data by using common data categories (nominal, ordinal, dichotomous, discrete, etc.):

- Intelligence scores: ordinal
- Temperature in Kelvin or Temperature in ^{a}C: continuous
- Rainy days in September: discrete
- Hair color: nominal
- Rankings of teams in a tournament: ordinal
- Body size classified as tiny, small, medium, tall, and giant: ordinal
- Customer satisfaction (satisfied or not satisfied): dichotomous

7. Given the following data:

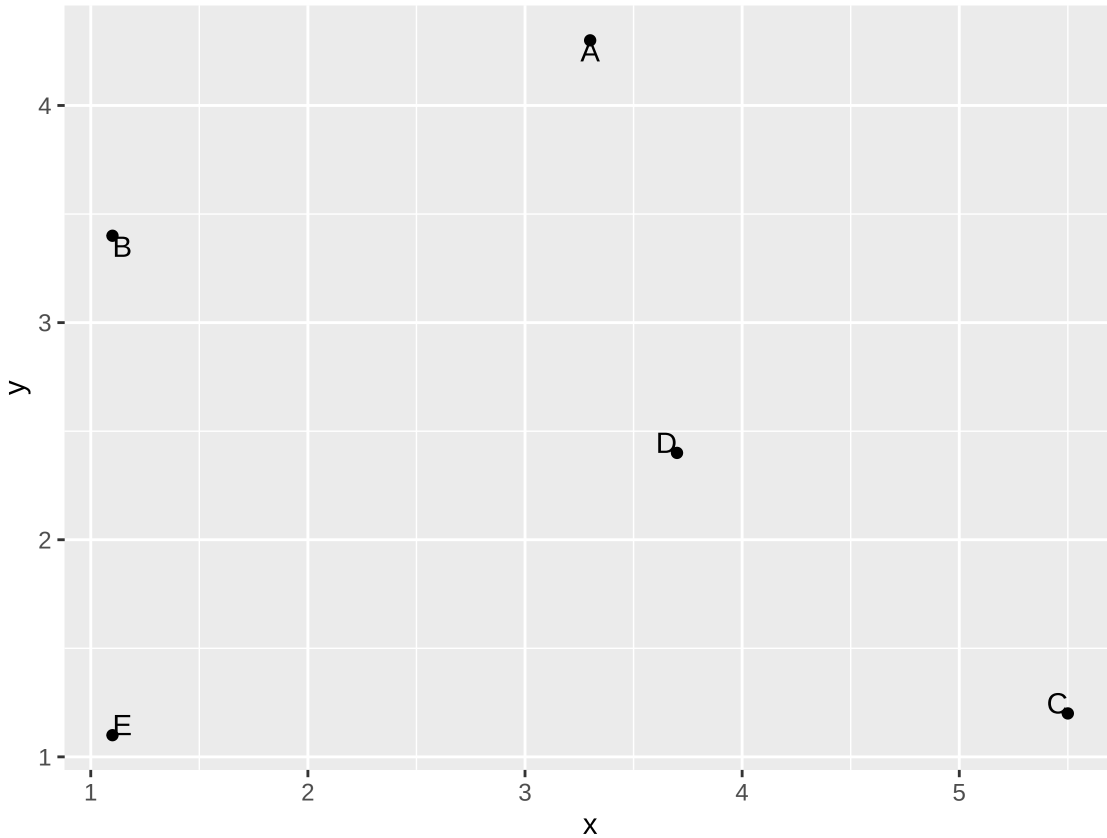
```
city <- c('A', 'B', 'C', 'D', 'E')
x <- c(3.3, 1.1, 5.5, 3.7, 1.1)
y <- c(4.3, 3.4, 1.2, 2.4, 1.1)
pop <- c(34000, 6500, 8000, 5000, 1500)

(db <- data.frame(city = city,
                  x = x,
                  y = y,
                  pop = pop)
)
```

```
##   city   x   y   pop
## 1    A 3.3 4.3 34000
## 2    B 1.1 3.4  6500
## 3    C 5.5 1.2  8000
## 4    D 3.7 2.4  5000
## 5    E 1.1 1.1  1500
```

```
(g <- ggplot(db, aes(x = x, y = y)) +
    geom_point()+
    geom_text(aes(label = city),  vjust = "inward", hjust = "inward") +
    coord_equal()

)
```



a. Find the weighted mean center of population.

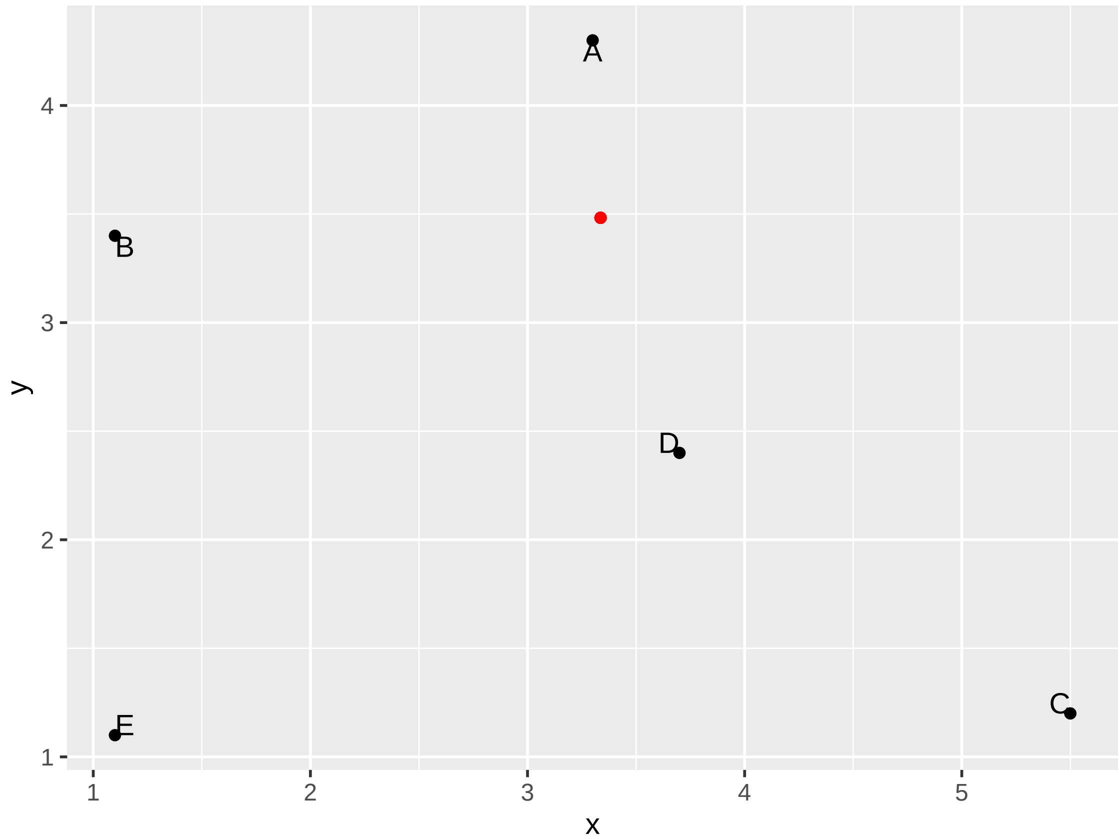
```
(w_x <- weighted.mean(db$x, w = db$pop))
```

```
## [1] 3.336364
```

```
(w_y <- weighted.mean(db$y, w = db$pop))
```

```
## [1] 3.482727
```

```
g + geom_point(x = w_x, y = w_y, color = 'red')
```

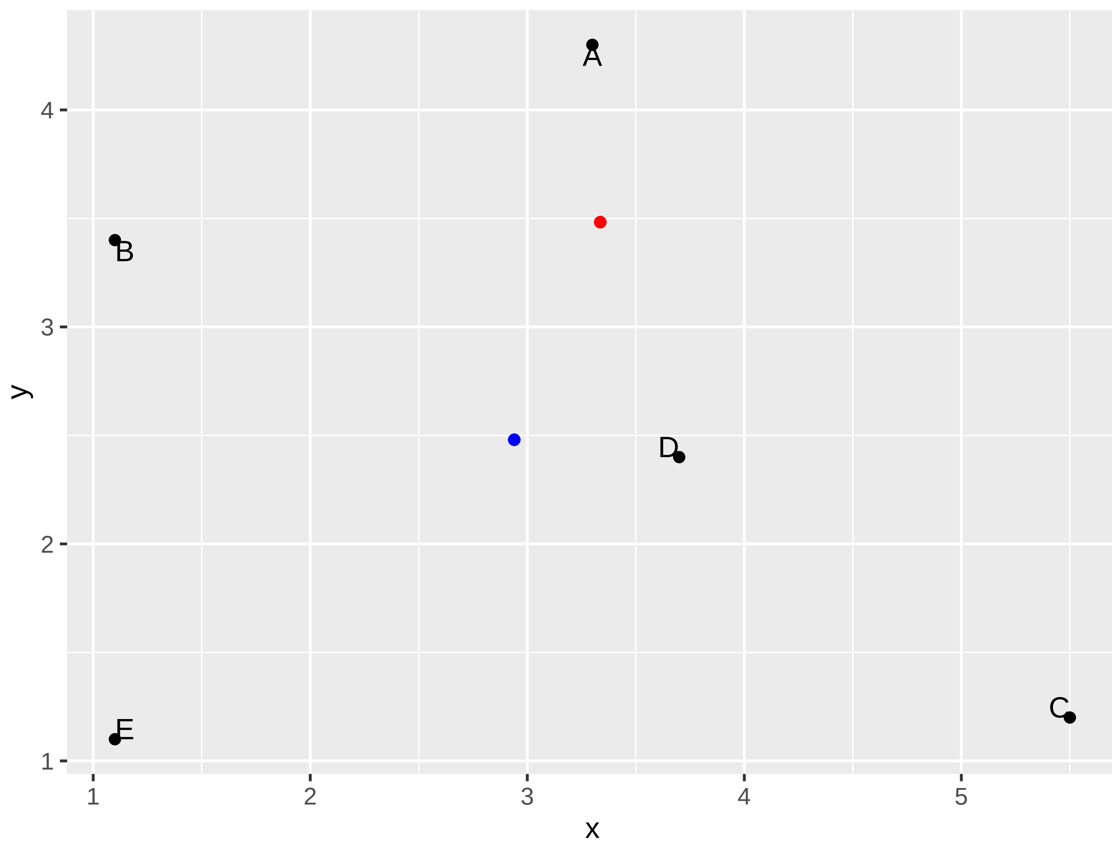


The weighted mean center can be visualized on the plot above relative to the position of the cities.

b.Find the unweighted mean center, and comment on the differences between your two answers.

```
m_x <- mean(db$x)
m_y <- mean(db$y)

g + geom_point(x = w_x, y = w_y, color = 'red') +
 geom_point(x = m_x, y = m_y, color = 'blue')
```



The position of both weighted (red) and un-weighted (blue) mean centers can be seen in the plot above. It is noticeable the effect that the population has on the position of the red dot, the city A has 62% of the total population and makes the center be very close to A. While in the un-weighted position D is closer to the center.

c. Find the distance of each city to the weighted mean center of population.

```
(db <- db %>%
  mutate(distance = sqrt((x - w_x)^2 + (y - w_y)^2))
)
```

```
##   city   x   y   pop  distance
## 1    A 3.3 4.3 34000 0.8180813
## 2    B 1.1 3.4  6500 2.2378932
## 3    C 5.5 1.2  8000 3.1451814
## 4    D 3.7 2.4  5000 1.1421601
## 5    E 1.1 1.1  1500 3.2678298
```

As mentioned earlier, city A is the closest to the weighted center with a distance of 0.8, while city E is the farthest with a distance of 3.3.

d. Find the standard distance (weighted) for the five cities.

```r
sdist <- sqrt(sum((db$x - w_x)^2) + sum((db$y - w_y)^2)) / (nrow(db)-2)
```

e. Find the relative standard distance assuming that the study area has a size of 10.

```r
(r <- sqrt(10/pi))
```

```
## [1] 1.784124
```

```r
(sdistrel <- sdist/r)
```

```
## [1] 0.9807014
```

f. Repeat part (e), this time assuming that the size of the study area is 15,000m^2.

```r
(r <- sqrt(15000/pi))
```

```
## [1] 69.09883
```

```r
(sdistrel <- sdist/r)
```

```
## [1] 0.0253216
```