Homework 2

David Lewis

February 20, 2025

Load the data

```
data <- USArrests
```

1. What is the size of the data? 2 points

```
dim(data)
```

```
## [1] 50  4
```

There are 50 rows and 4 columns

2. Describe the data. 2 points

```
library(printr)
```

```
## Registered S3 method overwritten by 'printr':
##   method                from
##   knit_print.data.frame rmarkdown
```

```
?USArrests
```

```
## Violent Crime Rates by US State
##
## Description:
##
##      This data set contains statistics, in arrests per 100,000
##      residents for assault, murder, and rape in each of the 50 US
##      states in 1973.  Also given is the percent of the population
##      living in urban areas.
##
## Usage:
##
##      USArrests
##
## Format:
##
##      A data frame with 50 observations on 4 variables.
##
##        [,1]  'Murder'    numeric  Murder arrests (per 100,000)
##        [,2]  'Assault'   numeric  Assault arrests (per 100,000)
##        [,3]  'UrbanPop'  numeric  Percent urban population
##        [,4]  'Rape'      numeric  Rape arrests (per 100,000)
```

```
##
## Note:
##
##      'USArrests' contains the data as in McNeil's monograph.  For the
##      'UrbanPop' percentages, a review of the table (No. 21) in the
##      Statistical Abstracts 1975 reveals a transcription error for
##      Maryland (and that McNeil used the same "round to even" rule that
##      R's 'round()' uses), as found by Daniel S Coven (Arizona).
##
##      See the example below on how to correct the error and improve
##      accuracy for the '<n>.5' percentages.
##
## Source:
##
##      World Almanac and Book of facts 1975.  (Crime rates).
##
##      Statistical Abstracts of the United States 1975, p.20, (Urban
##      rates), possibly available as
##      <https://books.google.ch/books?id=zl9qAAAAMAAJ&pg=PA20>.
##
## References:
##
##      McNeil, D. R. (1977) _Interactive Data Analysis_.  New York:
##      Wiley.
##
## See Also:
##
##      The 'state' data sets.
##
## Examples:
##
##      summary(USArrests)
##
##      require(graphics)
##      pairs(USArrests, panel = panel.smooth, main = "USArrests data")
##
##      ## Difference between 'USArrests' and its correction
##      USArrests["Maryland", "UrbanPop"] # 67 -- the transcription error
##      UA.C <- USArrests
##      UA.C["Maryland", "UrbanPop"] <- 76.6
##
##      ## also +/- 0.5 to restore the original  <n>.5  percentages
##      s5u <- c("Colorado", "Florida", "Mississippi", "Wyoming")
##      s5d <- c("Nebraska", "Pennsylvania")
##      UA.C[s5u, "UrbanPop"] <- UA.C[s5u, "UrbanPop"] + 0.5
##      UA.C[s5d, "UrbanPop"] <- UA.C[s5d, "UrbanPop"] - 0.5
##
##      ## ==> UA.C  is now a *C*orrected version of  USArrests
```

3. Show the top nine rows of the data. 2 points

```r
head(data, 9)
```

	Murder	Assault	UrbanPop	Rape
Alabama	13.2	236	58	21.2
Alaska	10.0	263	48	44.5
Arizona	8.1	294	80	31.0
Arkansas	8.8	190	50	19.5
California	9.0	276	91	40.6
Colorado	7.9	204	78	38.7
Connecticut	3.3	110	77	11.1
Delaware	5.9	238	72	15.8
Florida	15.4	335	80	31.9

4. Write a function that calculates and returns length, mean and standard deviation of a vector. 2 points

```
stats <- function(vector) {
    if (!is.vector(vector) || !is.atomic(vector)) {
        stop("Input is not a vector")
    }
    if (sum(is.na(vector)) > 0) {
        message("WARNING: input vector contains NA values, removing")
        vector <- na.omit(vector)
    }

    stat <- list()

    stat$length <- length(vector)
    stat$sd <- sd(vector)
    stat$mean <- mean(vector)

    return(stat)
}
```

5. Does you function deal with missing values? 2 points

Yes, the function prints a warning if the input vector contains missing values and removes them before calculating the mean standard deviation and length.

6. How can you modify it so that missing values are ignored? 2 points

There is a check in the function before the calculations that detects if the input vector contains any NA values via `is.na`

7. Apply the function from 6 to each column of the USArrests data. Report the means and standard deviations. 2 points

```
print_only_mean_and_sd <- function(l) {
    list(mean = l$mean, standard_deviation = l$sd)
}
statistics <- apply(data, 2, stats)
sapply(statistics, print_only_mean_and_sd)
```

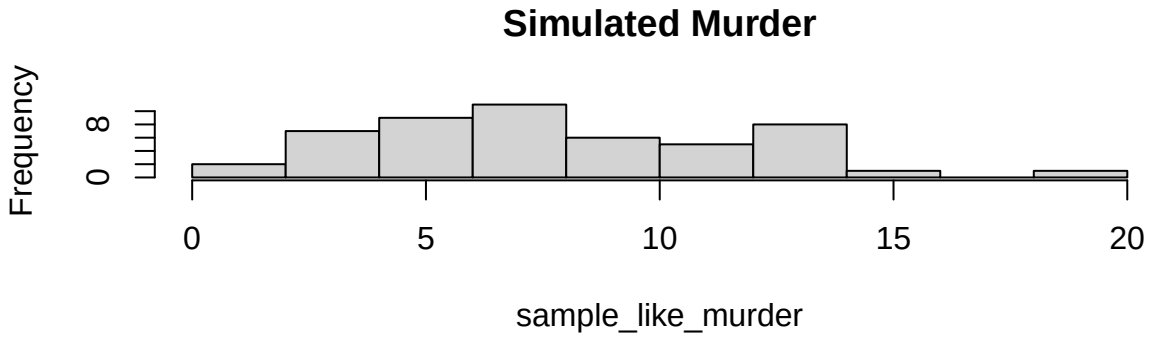
	Murder	Assault	UrbanPop	Rape
mean	7.788	170.76	65.54	21.232
standard_deviation	4.35550976420929	83.3376608400171	14.4747634008368	9.36638453105965

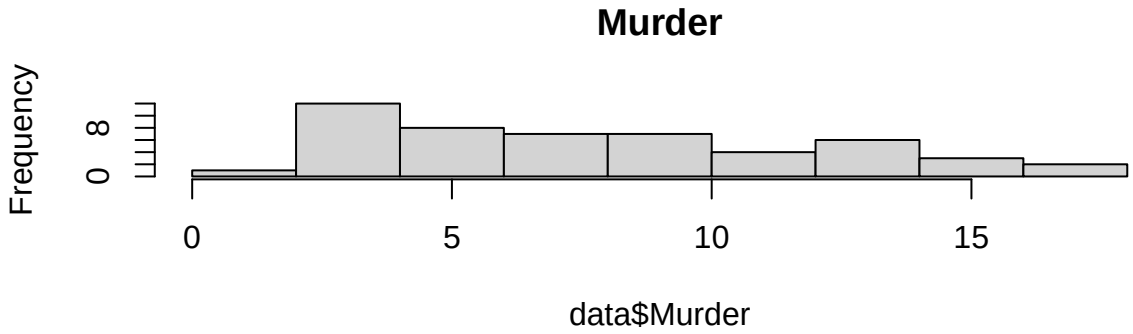
8. Generate a random sample from a normal distribution with characteristics similar to the murder column in USArrests data (same size, same mean, same sd). 2 points

```
set.seed(0) # so my commentatary on the histograms make sense
sample_like_murder <- rnorm(statistics$Murder$length, statistics$Murder$mean, statistics$Murder$sd)
```

9. Create two histograms - for the simulated and the original murder data. Use par(mfrow = c(2,1)) command to plot two histograms in one window. Do the histograms look similar? 2 points

```
par(mfrow = c(2, 1))
hist(sample_like_murder, main = "Simulated Murder")
hist(data$Murder, main = "Murder")
```





They do not look that similar, the `rnorm` sample resembles a normal distribution, while the actual murder histogram is much more irregular when compared to a normal distribution. However, there are not enough data points (50) to really make a valid assumption about the "normal-ness" of the data.

10. Create a new column called MurderRatio – the ratio of murder to urban population percent. Use with() or within() functions. 2 points

```
data <- within(data, {
    MurderRatio <- Murder / UrbanPop
})
```

11. Define a new variable called MurderGroup as follows - if MurderRatio is higher than 0.25, the state is considered "High", if MurderRatio is between 0.05 and 0.25, the state is considered "Medium", and the state is "Low" otherwise. 2 points

```
MurderGroup <- cut(data$MurderRatio, breaks = c(-Inf, 0.05, 0.25, Inf), labels = c("Low", "Medium", "Hig
```

12. What percent of the states is in High, Low, Medium groups? 2 points

```
percent <- function(vector) {
    (table(vector) / length(vector)) * 100
}
percent(MurderGroup)
```

Low	Medium	High
20	72	8

13. Generate a random sample from the set of labels c("High", "Low", "Medium") with probabilities from Question 12. Use the documentation command to learn how to specify the probabilities of selection. What percent of the states are in High, Low, and Medium groups in simulated data? Why are these percentages not equal to percentages in 11? 4 points

```
s <- sample(c("High", "Low", "Medium"), size = 50, replace = TRUE, prob = c(0.2, 0.72, 0.08))
percent(s)
```

High	Low	Medium
8	80	12

Because the sample size is small, the sample is likely to not exactly represent the distribution (given by the probabilities.) Increasing the sample size should better fit the probabilities.

```
set.seed(0)
s <- sample(c("High", "Low", "Medium"), size = 5000, replace = TRUE, prob = c(0.2, 0.72, 0.08))
percent(s)
```

High	Low	Medium
20.06	71.26	8.68

When the sample size is increased to 5000, the percentages are a much closer to the given probabilities.

14. Summarize all variables in the USArrests data by MurderGroup by group means? Do you see a pattern in different crime counts by groups? 2 points

```
aggregate(. ~ MurderGroup, data = data, FUN = mean)
```

MurderGroup	Murder	Assault	UrbanPop	Rape	MurderRatio
Low	2.500000	89.0000	63.80000	11.81000	0.0385844
Medium	8.430556	182.2778	67.83333	23.94444	0.1266435
High	15.225000	271.5000	49.25000	20.37500	0.3111995

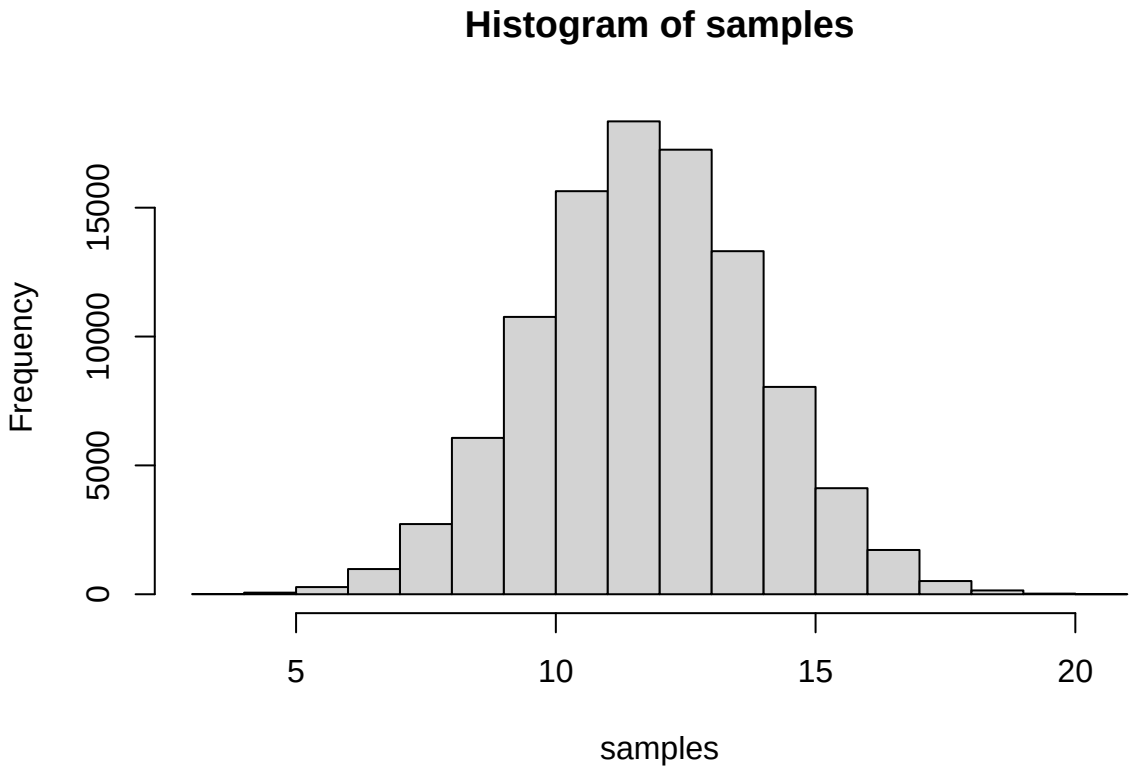
The highest rate of murders occur in populations with a low ratio of urban population. There is a similar pattern that occurs with the other crimes in relation to UrbanPopulation There is a similar pattern that occurs with the other crimes in relation to UrbanPopulation. The only contrary variable is Rape, which occurs most often at the Medium murder ratio (the highest urban population).

15. Optional problem for extra points: 5 points

Results of 100 coin toss experiments are recorded as follows: 0 = "Tail", 1 = "Head". The output, X, is a string of 0's and 1 of length 100. Then, the number of times when we see 1-0-0 in X is calculated and equal to 21. (Example: if X = c(0001001110100), 1-0-0 occurs 2 times). Do you believe that this is a fair coin (probability of 1 = 0.5)? Justify your answer. HINTS: You may use other functions, or functions below:

- You can use paste(x, collapse = " ") to convert x to a string;
- You can use unlist(strsplit(x, split = " ")) to convert string x to a vector of characters;
- gsub("100", "X", x) can be used to replace a "100" pattern with "X" pattern.

```
matches100 <- function(seed = 0) {
    set.seed(seed)
    s <- sample(c("0", "1"), size = 100, replace = TRUE, prob = c(0.5, 0.5))
    s <- paste(s, collapse = "")
    l <- gregexpr("100", as.vector(s))[[1]] ## we can use gregexpr to count all "100" matches
    length(l)
}
samples <- sapply(1:100000, matches100)
par(mfrow = c(1, 1))
hist(samples)
```



As you can see from the histogram, 21 "100" matches is extremely unlikely given 100000 trials of 100 coin tosses.

To see just how unlikely, we can count the max number of "100" matches.K

```
max(samples)
```

```
## [1] 21
```

```
length(which(samples == max(samples)))
```

```
## [1] 4
```

As we can see, the number of "100" matches is only 21 times out of 100000 trials.

Verdict: While it is possible that it is a fair coin, it is very unlikely.