STAT 210
Applied Statistics and Data Analysis:
Homework 2 - Solution

Due on Sep. 21/2025

You cannot use artificial intelligence tools to solve this homework.
Show complete solutions to get full credit. Writing code is not enough to answer a question. Your comments are more important than the code. Do not write comments in chunks. Label your graphs appropriately

Question 1

You will need the file `weights.txt`. This file has (simulated) data on an experiment to test the effects of two different types of diet. The set has 100 observations (50 males and 50 females), and there are three types of `diet`, coded as 1, 2, and 3. The first type (coded 1) corresponds to the control group and has subjects who did not change their usual diet during the eight weeks that the experiment lasted. The other two groups (coded 2 and 3) correspond to the diets that are being tested. The weight is measured in kilograms and the height in centimeters.

(a) Read the file `weights.txt` and store it in an object called `data1`. Look at the structure of `data1` using the function `str`. Check whether there are missing values in the file using the function `is.na`.

(b) The columns `pre_weight` and `post_weight` have the weight for the subjects before starting the diet and eight weeks after following the diet, respectively. Create a new column in `data1` called `diff` that has the difference in weights (final weight minus initial weight).

(c) Using the function `subset`, create a new data frame named `data1b` that has the variables `gender`, `age`, `height`, `diet`, and `diff` in `data1`. Using `data1b` and the function `tapply`, calculate mean and standard deviation for `diff` according to `diet`. Compare these results and comment.

(d) Using `data1b` and the function `tapply`, calculate mean and standard deviation for `diff` according to `diet` and `gender`. Compare these results and comment.

(e) Use the function `split` on the file `data1` with argument `gender` and store the result in an object called `d1`. What type of object is `d1`? Use the function `quantile` on the variable `height` on each of the components of `d1` to get a summary of the height for the different genders classes. Store the results in two vectors named `ql1` and `ql2`. Calculate `ql2/ql1` and interpret the result.

(f) The body mass index (BMI) is defined as a person's weight in kilograms divided by the square of height in meters. Add a column named `bmi` to the data frame `data1` with the value of this index for each subject using the weight before the experiment started. Count how many subjects have BMI above 30.

Solution:

```
data1 <- read.table('weights.txt', header = T)
str(data1)
```

```
## 'data.frame':    100 obs. of  7 variables:
##  $ subject    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ gender     : chr  "F" "F" "F" "F" ...
##  $ age        : int  50 30 43 36 43 41 24 24 57 45 ...
##  $ height     : int  169 170 175 162 173 157 170 153 161 168 ...
##  $ pre_weight : int  76 70 73 65 75 42 68 41 43 66 ...
##  $ diet       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ post_weight: num  74.4 73.6 75.3 65.6 77.7 ...
```

There are seven variables in the data frame. The first, `subject` is just an index. The we have `gender`, `age`, `height`, `pre_weight`, which is the weight before the experiment started, `diet`, and `post_weight`, which is the weight after dieting for eight weeks.

We check for `NA`'s:

```
sum(is.na(data1))
```

```
## [1] 0
```

There are none.

(b) We create the new column in the data frame and use `str` to chack:

```
data1$diff <- data1$post_weight - data1$pre_weight
str(data1)
```

```
## 'data.frame':    100 obs. of  8 variables:
##  $ subject    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ gender     : chr  "F" "F" "F" "F" ...
##  $ age        : int  50 30 43 36 43 41 24 24 57 45 ...
##  $ height     : int  169 170 175 162 173 157 170 153 161 168 ...
##  $ pre_weight : int  76 70 73 65 75 42 68 41 43 66 ...
##  $ diet       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ post_weight: num  74.4 73.6 75.3 65.6 77.7 ...
##  $ diff       : num  -1.6 3.645 2.259 0.648 2.666 ...
```

(c) New data frame:

```
data1b <- subset(data1, select = c('gender', 'age', 'height','diet',  'diff'))
str(data1b)
```

```
## 'data.frame':    100 obs. of  5 variables:
##  $ gender: chr  "F" "F" "F" "F" ...
##  $ age   : int  50 30 43 36 43 41 24 24 57 45 ...
##  $ height: int  169 170 175 162 173 157 170 153 161 168 ...
##  $ diet  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ diff  : num  -1.6 3.645 2.259 0.648 2.666 ...
```

We create the vectors with mean and standard deviation.

```
(diff_mn <- tapply(data1b$diff, data1$diet, mean))
```

```
##          1          2          3
##  0.3076043  0.2291316 -1.2873649
```

```
(diff_sd <- tapply(data1b$diff, data1$diet, sd))
```

```
##         1         2         3
## 2.5301875 3.6408409 0.9576384
```

We see that the average difference for diets 1 and 2 is positive and small, while the difference for diet 3 is

negative and above one kilogram. Regarding the standard deviation, it is smaller for subjects that followed diat 3 than for the other two diets.

(d) The means and standard deviations according to diet and gender are

```
tapply(data1b$diff, data1[,c(2,6)], mean)
```

```
##       diet
## gender          1         2         3
##      F  1.0094958 0.2901713 -1.328099
##      M -0.3529995 0.1642768 -1.246631
```

```
tapply(data1b$diff, data1[,c(2,6)], sd)
```

```
##       diet
## gender        1        2         3
##      F 2.076769 4.058366 1.0064372
##      M 2.793558 3.271363 0.9354238
```

With diet 1, females have an average positive gain of weight of about one kilogram while men lose about 350 grams. For diet 2, the results for both genders are similar, they are positive and small. Finally, for diet 3 both values are negative and similar, with a loss of about 1.3 kilograms.

Regarding the standard deviation, the values for males and females are similar for all diets, but there are diffeences between diets, with diet 3 having the smalled standard deviation, while diet 2 has the largest.

(e) We use `split`:

```
d1 <- split(data1, data1$gender)
str(d1)
```

```
## List of 2
##  $ F:'data.frame':   50 obs. of  8 variables:
##   ..$ subject    : int [1:50] 1 2 3 4 5 6 7 8 9 10 ...
##   ..$ gender     : chr [1:50] "F" "F" "F" "F" ...
##   ..$ age        : int [1:50] 50 30 43 36 43 41 24 24 57 45 ...
##   ..$ height     : int [1:50] 169 170 175 162 173 157 170 153 161 168 ...
##   ..$ pre_weight : int [1:50] 76 70 73 65 75 42 68 41 43 66 ...
##   ..$ diet       : int [1:50] 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ post_weight: num [1:50] 74.4 73.6 75.3 65.6 77.7 ...
##   ..$ diff       : num [1:50] -1.6 3.645 2.259 0.648 2.666 ...
##  $ M:'data.frame':   50 obs. of  8 variables:
##   ..$ subject    : int [1:50] 51 52 53 54 55 56 57 58 59 60 ...
##   ..$ gender     : chr [1:50] "M" "M" "M" "M" ...
##   ..$ age        : int [1:50] 50 39 48 46 54 44 42 27 42 49 ...
##   ..$ height     : int [1:50] 193 172 182 167 188 181 181 177 163 179 ...
##   ..$ pre_weight : int [1:50] 100 62 92 43 112 81 88 84 41 66 ...
##   ..$ diet       : int [1:50] 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ post_weight: num [1:50] 97.3 61.4 91.9 42.7 108.3 ...
##   ..$ diff       : num [1:50] -2.691 -0.6187 -0.0856 -0.3228 -3.7141 ...
```

Object `d1` is a list with two components, which correspond to the two values for `gender`: `F` and `M`. Each component is a data frame with all the information included in `data1` for the corresponding value of `gender`. Hence, each component has 50 observations.

```
ql1 <- quantile(d1$'F'$height)
ql2 <- quantile(d1$'M'$height)
ql2/ql1
```

```
##       0%      25%      50%      75%     100%
## 1.053691 1.037267 1.066265 1.076923 1.066298
```

The function `quantile` gives the minimum, maximum, and quartile values for the sample. We observe that the ratio of the male to female values is always greater than one by between 3.7 and 7.7 %.

(f) Since the height is in centimeters, we need to divide by 100:

```
data1$bmi <- data1$pre_weight/(data1$height/100)^2
str(data1)
```

```
## 'data.frame':    100 obs. of  9 variables:
##  $ subject    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ gender     : chr  "F" "F" "F" "F" ...
##  $ age        : int  50 30 43 36 43 41 24 24 57 45 ...
##  $ height     : int  169 170 175 162 173 157 170 153 161 168 ...
##  $ pre_weight : int  76 70 73 65 75 42 68 41 43 66 ...
##  $ diet       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ post_weight: num  74.4 73.6 75.3 65.6 77.7 ...
##  $ diff       : num  -1.6 3.645 2.259 0.648 2.666 ...
##  $ bmi        : num  26.6 24.2 23.8 24.8 25.1 ...
```

How many are above 30:

```
sum(data1$bmi> 30)
```

```
## [1] 10
```

Ten out of 100 have BMI over 30.

Question 2

(a) Create two matrices. The first, called `m1`, has dimension 3×5 and the components are values simulated from a binomial distribution with size 15 and probability 0.25. The second, called `m2`, has dimension 5×3 and the components are values simulated from a Poisson distribution with parameter 3. Create also a vector `v1` of length three from a negative binomial distribution with parameters `size = 2` and `prob = 0.25`.

(b) Create a list named `hwlist` that has as a first component `m1`, second component `m2`, and third component `v1`. The names of these components should be `item1`, `item2`, and `item3`, respectively. Use the function `rm` to remove `m1`, `m2`, and `v1` from the working environment.

(c) Using matrix multiplication, multiply `item1` times `item2` and store the result in `hwlist` under the name `item4`. Multiply `item1` and `item2` also in the reverse order, but do not store the outcome.

(d) Denote the transpose of a matrix M by M^t. Verify that

$$(\texttt{item1} \star \texttt{item2})^t = (\texttt{item1})^t \star (\texttt{item2})^t$$

where $\star$ denotes standard matrix multiplication, and verify this relation also for the product of the matrices in the reverse order.

(e) The matrix `item4` has dimension 3×3. Add an identity matrix of dimension 3 to `item4` and store it in the same position.

(f) Solve the system of equations `item4` $\star$`x` = `item3`. Verify that you have obtained the correct solution.

(g) Find the inverse of `item4` and call it `item4_inv`. Verify that `item4_inv` is indeed the inverse of `item4`, and that multiplying `item4_inv` $\times$ `item3` gives the solution to the system of equations in (f).

Solution:

(a) Matrices and vector:

```
set.seed(2468)
(m1 <- matrix(rbinom(15, size = 15, prob = 0.25), ncol = 5))
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    4    6    4    3    2
## [2,]    4    2    2    1    5
## [3,]    3    3    5    1    1
```

```
(m2 <- matrix(rpois(15, 3), ncol = 3))
```

```
##      [,1] [,2] [,3]
## [1,]    5    2    3
## [2,]    2    4    6
## [3,]    1    1    4
## [4,]    2    2    1
## [5,]    9    2    5
```

```
(v1 <- rnbinom(3, size = 2, prob = 0.25))
```

```
## [1] 11  4 15
```

(b)

```
hwlist <- list(item1 = m1, item2 = m2, item3 = v1)
str(hwlist)
```

```
## List of 3
##  $ item1: int [1:3, 1:5] 4 4 3 6 2 3 4 2 5 3 ...
##  $ item2: int [1:5, 1:3] 5 2 1 2 9 2 4 1 2 2 ...
##  $ item3: int [1:3] 11 4 15
```

Remove original objects. We check with `ls` that the objects are no longer in the working directory.

```
rm(m1, m2, v1)
ls()
```

```
## [1] "d1"      "data1"   "data1b"  "diff_mn" "diff_sd" "hwlist"  "ql1"
## [8] "ql2"
```

(c) Matrix multiplications:

```
(hwlist$item4 <- hwlist$item1 %*% hwlist$item2)
```

```
##      [,1] [,2] [,3]
## [1,]   60   46   77
## [2,]   73   30   58
## [3,]   37   27   53
```

In the reverse order

```
(hwlist$item2 %*% hwlist$item1)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   37   43   39   20   23
## [2,]   42   38   46   16   30
## [3,]   20   20   26    8   11
## [4,]   19   19   17    9   15
## [5,]   59   73   65   34   33
```

(d) Verify the relation:

```
t(hwlist$item1 %*% hwlist$item2)
```

```
##      [,1] [,2] [,3]
## [1,]   60   73   37
## [2,]   46   30   27
## [3,]   77   58   53
```

```
t(hwlist$item2) %*% t(hwlist$item1)
```

```
##      [,1] [,2] [,3]
## [1,]   60   73   37
## [2,]   46   30   27
## [3,]   77   58   53
```

In the reverse order:

```
t(hwlist$item2 %*% hwlist$item1)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   37   42   20   19   59
## [2,]   43   38   20   19   73
## [3,]   39   46   26   17   65
## [4,]   20   16    8    9   34
## [5,]   23   30   11   15   33
```

```
t(hwlist$item1) %*% t(hwlist$item2)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   37   42   20   19   59
## [2,]   43   38   20   19   73
## [3,]   39   46   26   17   65
## [4,]   20   16    8    9   34
## [5,]   23   30   11   15   33
```

(e)

```
(hwlist$item4 <- hwlist$item4 + diag(3))
```

```
##      [,1] [,2] [,3]
## [1,]   61   46   77
## [2,]   73   31   58
## [3,]   37   27   54
```

(f) We use `solve` to find the solution

```
(x <- solve(hwlist$item4, hwlist$item3))
```

```
## [1] -0.3007154 -1.0547695  1.0112083
```

To verify that this is the solution, we multiply `item4` by `x`

```
hwlist$item4 %*% x
```

```
##      [,1]
## [1,]   11
## [2,]    4
## [3,]   15
```

(g) The inverse is also obtained using the function `solve`:

```
(item4_inv <- solve(hwlist$item4))
```

```
##              [,1]         [,2]        [,3]
## [1,] -0.008585056  0.032193959 -0.02233704
## [2,]  0.142766296 -0.035373609 -0.16558029
## [3,] -0.065500795 -0.004372019  0.11661367
```

We check that this is the inverse. To make the result clearer, we round the value to 14 digits

```
round(hwlist$item4 %*% item4_inv, 14)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

```
round(item4_inv %*% hwlist$item4, 14)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

Verify solution to system of equations:

```
item4_inv %*% hwlist$item3
```

```
##            [,1]
## [1,] -0.3007154
## [2,] -1.0547695
## [3,]  1.0112083
```

Recall the `x` is

```
x
```

```
## [1] -0.3007154 -1.0547695  1.0112083
```

and we see that they are equal.