Stat 5050: Introduction to R

Name: Peter Kurtz

Homework Assignment 04

70 Points

General Instructions

For this fourth homework assignment, you have to work with RMarkdown or knitr/Sweave. You can create your own RMarkdown (.Rmd) file, based on files from class and from Homework 1, copy the question numbers and the answer options into your .Rmd file, and knit that file into a pdf file. **Alternatively** (and much easier!!!), use this .Rnw file as a template, just fill in the answers into the provided spaces, and knit into a pdf file.

Only the final resulting pdf file (from .Rmd or .Rnw) has to be submitted via Canvas. As previously stated, I would like to encourage potential and current MS and PhD students to work with .Rnw and LaTeX instead of .Rmd.

You need to learn how to write R code that is easily readable for others. There exists the *Google's R Style Guide* (provided as a pdf here in Canvas) that summarizes rules for good R style. This style guide closely resembles the far more detailed *Tidyverse Style Guide*. These rules are accessible at `https://style.tidyverse.org/`. In particular, make sure that you always have a space after a comma and that you consistently use the same type of assignment operator, ideally `<-`. Look at the examples in these style guides and follow that style whenever you write your own R code from now on.

Do not forget to replace my name and include your name instead!

In all question parts, show your R code and the results!

(i) (20 Points) **Family Data Revisited:**

In the following exercises, try to write your code to be as general as possible so that it would still work if the family had 27 members in it or if the variables were in a different order in the data frame.

Show your R code and the final results produced from within R for all question parts!

(a) (3 Points) Copy the family data set for this homework from Canvas into your local folder for this homework. Then load the `hw04_familyDF.rda` data set into R. Show the "objects" that have been loaded.

Is the first "object" that is listed a data frame? The R output should be TRUE or FALSE. Search for help if you don't recall how to check whether something is a data frame.

Answer:

```
load(’hw04_familyDF.rda’)
names(family)

## [1] "firstName" "gender"    "age"       "height"
## [5] "weight"    "bmi"       "overWt"

is.data.frame(family[1])

## [1] TRUE
```

(b) (4 Points) The NHANES survey used different cut-off values for men and women when classifying them as overweight. Suppose that a man is classified as obese if his bmi exceeds 26 and a woman is classified as obese if her bmi exceeds 25. Write a logical expression to create a logical vector, called OW.NHANES, that is TRUE if a member of the family is obese and FALSE otherwise. Display its content.

Answer:

```
OW.NHAMES = family["bmi"] > 25 & family["gender"] == "f" |
  family["bmi"] > 26 & family["gender"] == "m"
OW.NHAMES

##         bmi
##  [1,] FALSE
##  [2,] FALSE
##  [3,] FALSE
##  [4,] FALSE
##  [5,] FALSE
##  [6,]  TRUE
##  [7,]  TRUE
##  [8,] FALSE
##  [9,]  TRUE
## [10,]  TRUE
## [11,]  TRUE
## [12,] FALSE
## [13,] FALSE
## [14,] FALSE
```

(c) (4 Points) Here is an alternative way to create the same vector that introduces some useful functions and ideas. We first create a numeric vector called OW.limit that is 26 for each male in the family and 25 for each female in the family. To do this, we create a vector of length 2, called OW.val, where the first element is 26 and second element is 25. Then we create the OW.limit vector by subsetting OW.val by position, where the positions are the numeric values in the gender variable (i.e., use as.numeric to coerce the factor vector to a numeric vector). Notice that we can "subset" a vector of length 2 by a much longer vector:

```
# Note that this code chunk is not executed because eval=FALSE.
# Change to eval=TRUE once you have answered the previous question parts.

OW.val <- 26:25
OW.limit <- OW.val[as.numeric(family$gender)]
OW.limit

##  [1] 26 25 26 26 25 25 26 25 26 26 25 26 26 25
```

Finally, use OW.limit and the bmi vector in family to create the desired logical vector, and call it OW.NHANES2. Display its content. Compare with your results from part (b) via the `any` function. Did you get the intended result? If not, check your R code again!

Answer:

```
OW.NHANES2 = family["bmi"] >= OW.limit
OW.NHANES2
```

```
##         bmi
##  [1,] FALSE
##  [2,] FALSE
##  [3,] FALSE
##  [4,] FALSE
##  [5,] FALSE
##  [6,]  TRUE
##  [7,]  TRUE
##  [8,] FALSE
##  [9,]  TRUE
## [10,]  TRUE
## [11,]  TRUE
## [12,] FALSE
## [13,] FALSE
## [14,] FALSE
```

```
setequal(OW.NHAMES, OW.NHANES2)
```

```
## [1] TRUE
```

(d) (4 Points) Use the vector OW.limit and each person's height to find the weight that they would have if their bmi was right at the limit (26 for men and 25 for women). Call this weight OW.weight and display its content. To do this, start with the formula

`bmi = (weight / 2.2) / (2.54 / 100 * height)^2`

and re-express it in terms of weight (i.e., `weight = ...`).

Answer:

```
heightFam = family[,c("height")]
OW.weight = OW.limit*(2.54/100*heightFam)**2*2.2
OW.weight
```

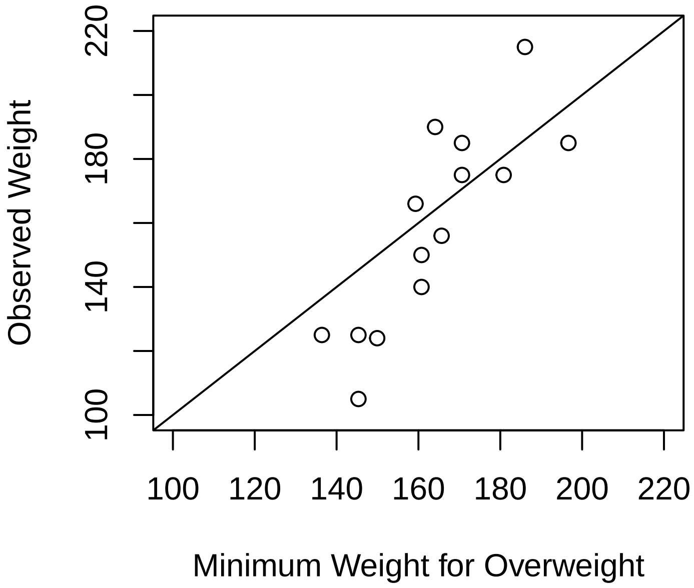
```
##  [1] 180.8254 145.3416 196.6569 165.6582 145.3416 164.0771
##  [7] 170.6402 149.9191 170.6402 186.0288 159.2868 160.7501
## [13] 160.7501 136.3997
```

(e) (5 Points) Create the following plot of actual weight (on the vertical axis) against the weight at which they would be overweight (on the horizontal axis). If you get an error when you run this code, check whether you are using the correct variable names in your code earlier on.

```
# Note that this code chunk is not executed because eval=FALSE.
# Change to eval=TRUE once you have answered the previous question parts.
#
# Make sure that your graph appears in your output!

plot(OW.weight, family$weight,
     xlab = "Minimum Weight for Overweight",
     xlim = c(100, 220), # !!!
     ylab = "Observed Weight",
     ylim = c(100, 220)) # !!!

abline(a = 0, b = 1)
```



`abline` adds a straight line (here with y-intercept $a = 0$ and slope $b = 1$) to the plot. Note that this is not the regression line! Thus, points that fall exactly on the line belong to individuals where the observed weight exactly qualifies to be overweight. Points above the line represent individuals who are overweight, and points below the line represent individuals who are not overweight.

We can easily count in the plot how many points are above the line and how many points are below the line, but we want that R does this counting for us! So, write two R expressions that do this counting for us and display their results.

Answer:

```r
# Number of points above the line
overWeight = sum(OW.NHAMES)
overWeight

## [1] 5

# Number of points below the line
notOverWeight = length(OW.NHAMES) - overWeight
```

```
notOverWeight

## [1] 9
```

(ii) (34 Points) **San Francisco Housing Data:**

In this question, you have to work with actual housing data from the San Francisco area.

Show your R code and the final results produced from within R for all question parts!

(a) (4 Points) Copy the San Francisco housing data set (`hw04_SFhousing.rda`) for this homework from Canvas into your local folder for this homework. Then load this data set into R. Show the "objects" that have been loaded. Are cities and housing both data frames? Let R answer this question! The R output should be TRUE or FALSE for each of these. Search for help if you don't recall how to check whether something is a data frame.

Answer:

```
load(’hw04_SFHousing.rda’)
names(housing)

##  [1] "county"  "city"    "zip"     "street"  "price"
##  [6] "br"      "lsqft"   "bsqft"   "year"    "date"
## [11] "long"    "lat"     "quality" "match"   "wk"

is.data.frame(housing[2])

## [1] TRUE

is.data.frame(housing)

## [1] TRUE
```

(b) (2 Points) What are the names of the vectors in housing?

Answer:

```
names(housing)

##  [1] "county"  "city"    "zip"     "street"  "price"
##  [6] "br"      "lsqft"   "bsqft"   "year"    "date"
## [11] "long"    "lat"     "quality" "match"   "wk"
```

(c) (2 Points) How many observations (i.e., rows) are in housing? Only report the number of rows, but not the number of columns!

Answer:

```
nrow(housing)

## [1] 281506
```

(d) (6 Points) Explore the housing data using the summary function. Describe in words at least three problems that you see with the data.

Answer:

```
summary(housing)

##                   county                city       
##  Santa Clara County :70424   Oakland      : 14730  
##  Alameda County     :60410   Santa Rosa   :  9917  
##  Contra Costa County:59381   Fremont      :  9414  
##  Solano County      :23404   San Francisco:  8137  
##  San Mateo County   :22558   Evergreen    :  7947  
##  Sonoma County      :21676   Antioch      :  7726  
##  (Other)            :23653   (Other)      :223635  
##       zip            street              price         
##  94565  :  4595   Length:281506      Min.   :   22000  
##  94509  :  4302   Class :character   1st Qu.:  400000  
##  95123  :  4023   Mode  :character   Median :  530000  
##  95687  :  3652                      Mean   :  602000  
##  94533  :  3472                      3rd Qu.:  700000  
##  (Other):261457                      Max.   :20000000  
##  NA's   :     5                                        
##        br            lsqft               bsqft        
##  Min.   :1.000   Min.   :       19   Min.   :    122  
##  1st Qu.:2.000   1st Qu.:     4000   1st Qu.:   1121  
##  Median :3.000   Median :     5760   Median :   1430  
##  Mean   :3.024   Mean   :    65939   Mean   :   1624  
##  3rd Qu.:4.000   3rd Qu.:     7701   3rd Qu.:   1882  
##  Max.   :8.000   Max.   :418611600   Max.   :1868120  
##                  NA's   :21687       NA's   :426      
##       year           date                        
##  Min.   :   0   Min.   :2003-04-27 01:00:00.00  
##  1st Qu.:1954   1st Qu.:2004-02-08 01:00:00.00  
##  Median :1971   Median :2004-10-24 01:00:00.00  
##  Mean   :1966   Mean   :2004-11-01 17:06:12.50  
##  3rd Qu.:1985   3rd Qu.:2005-07-24 01:00:00.00  
##  Max.   :3894   Max.   :2006-06-04 01:00:00.00  
##  NA's   :9202                                    
##       long              lat       
##  Min.   :-123.6   Min.   :36.98  
##  1st Qu.:-122.3   1st Qu.:37.50
##  Median :-122.1   Median :37.77
##  Mean   :-122.1   Mean   :37.78
##  3rd Qu.:-121.9   3rd Qu.:38.00
##  Max.   :-121.5   Max.   :38.85
##  NA's   :23316    NA's   :23316
##                                       quality
##  QUALITY_ADDRESS_RANGE_INTERPOLATION      :170719
##  gpsvisualizer                            : 31084
##  QUALITY_CITY_CENTROID                    : 20473
##  QUALITY_EXACT_PARCEL_CENTROID            : 17208
##  QUALITY_ZIP_CODE_TABULATION_AREA_CENTROID: 14980
##  (Other)                                  :  3726
##  NA's                                     : 23316
##               match              wk
##  Exact           :197044   Min.   :2003-04-21
##  Relaxed         : 30570   1st Qu.:2004-02-01
##  Relaxed; Soundex: 23338   Median :2004-10-18
##  Soundex         :  2573   Mean   :2004-10-26
##  1               :  2244   3rd Qu.:2005-07-18
##  (Other)         :  2421   Max.   :2006-05-29
##  NA's            : 23316
```

Problems:

i. There are many NA values for some of the variables. For example, there is no zip code for 5 values, no lsqft for 21687 values, no bsqft for 426, no year for 9202 etc.

ii. The year variable has a minimum of 0 and a maximum of 3894. These are not valid years.

iii. This is more of a subjective critique but I think the variable names should be more descriptive. For example, I do not understand what the bf variable is suppose to be.

(e) (4 Points) Motivated by a historic map from 1938, accessible at `https://www.davidrumsey.com/luna/servlet/detail/RUMSEY~8~1~248517~5515942:Map-of-Oakland,-Berkeley,-Alameda,-`, we will work with houses in the 7 nearby cities of Albany, Alameda, Berkeley, Emeryville, Oakland, Piedmont, and San Leandro, only. Subset the data frame so that we have only houses in these 7 cities, and keep only the variables city, zip, price, br, bsqft, and year. Call this new data frame BerkArea. This data frame should have 25,151 observations and 6 variables (check it!).

Answer:

```
chooseVars = c("city", "zip", "price", "br", "bsqft", "year")
nearCitiesVec = c("Albany", "Alameda", "Berkeley", "Emeryville", "Oakland", "Pi
ind = (housing$city %in% nearCitiesVec)
BerkArea = housing[ind,chooseVars]
nrow(BerkArea)

## [1] 25151
```

(f) (4 Points) We are interested in studying the relationship between price and size of house, but first we will further subset the data frame to remove the unusually large values. Use the quantile function to determine the 98th percentile of price and bsqft and eliminate all of those houses that are above either of these 98th percentiles. Call this new data frame BerkArea, as well. It should have 24,346 observations (check it!). Write your code so that it is very general and does not depend on the actual numeric value for these quantiles.

Answer:

```
quantile95P = quantile(BerkArea$price, .98)
indQ = BerkArea["price"] < quantile95P

quantile95Q = quantile(BerkArea$bsqft, .98, na.rm=TRUE)
indBs = BerkArea["bsqft"] < quantile95Q

indCom = indQ & indBs

BerkArea=BerkArea[indCom,]
nrow(BerkArea)

## [1] 24346
```

(g) (2 Points) Create a new vector that is called pricepsqft by dividing the sale price by the square footage of the house. Add this new variable to the BerkArea housing data frame and verify that it indeed has been added to the data frame.

Answer:

```
pricepsqft = BerkArea$price / BerkArea$bsqft
BerkArea$pricepsqft <- pricepsqft
names(BerkArea)
```

```
## [1] "city"       "zip"        "price"      "br"
## [5] "bsqft"      "year"       "pricepsqft"
```

(h) (4 Points) Create a vector called br6 that contains the number of bedrooms in the house, except when this number is greater than 6, it is set to 6. That is, if a house has 6 or more bedrooms then br6 will be 6. Otherwise it will be the number of bedrooms in the house. Note that there is no need for any “if”-statements or loops to create this vector — just basic R expressions discussed so far will be sufficient! Recall how TRUE and FALSE are represented numerically or how to reassign a different value to a subset!

Answer:

```
seven <- BerkArea$br >= 7
eight <- BerkArea$br >= 8

br6 <- BerkArea$br - seven - eight
BerkArea$br6 <- br6
```

(i) (6 Points) Recreate the following plot on your side. Then answer the question below. If you get an error when you run this code, check whether you are using the correct variable names in your code earlier on.

```
# Note that this code chunk is not executed because eval=FALSE.
# Change to eval=TRUE once you have answered the previous question parts.
#
# Make sure that your graph appears in your output!

rCols <- rainbow(6, alpha = 0.25)
brCols <- rCols[br6]

plot(pricepsqft ~ bsqft, data = BerkArea,
     main = "Housing Prices in the Berkeley Area",
     xlab = "Size of house (square ft)",
     ylab = "Price per square foot",
     col = brCols, pch = 19, cex = 0.5)
legend(legend = 1:6, fill = rCols, "topright")
```

What interesting feature do you see in the relationship between these variables that you may not have known before making this plot? Numerically quantify (use only 3 decimal digits!) and interpret this feature! HINT: If you aren't sure what measure to use, try searching via google (you should find a 3-letter function in R to calculate this measure of association).

Answer:

Place your answer here The relationship between size of house and price per square foot is slightly negative. Also, not surprisingly, the more rooms there are, the bigger the house. Using the correlation function in r outputs a 1.000. This is clearly not correct. Most likely do to something I don't understand or an error.

(iii) (16 Points) **Survival of Passengers on the Titanic:**

Work with the `Titanic` data set, a 4–dimensional array related to the survival of passengers and crew on board of the Titanic ocean liner. For further details, refer to the help page via `?Titanic`. Technically, the Titanic data set is a table, but we can access it similar to a multi–dimensional array.

Show your R code and the final results produced from within R for all question parts!

(a) (4 Points) Write an R expression that extracts the numbers of males in all three classes (but not crew) who survived the sinking of the Titanic. Provide data for children and adults. The result should look as follows:

```
     Age
Class Child Adult
  1st     5    57
  2nd    11    14
  3rd    13    75
```

Answer:

```
classNotCrew <- c("1st", "2nd", "3rd")
Titanic[classNotCrew,Sex = "Male",,Survived = "Yes"]

##      Age
## Class Child Adult
##   1st     5    57
##   2nd    11    14
##   3rd    13    75
```

(b) (4 Points) Write an R expression that extracts the numbers of female crew members (adults only) who survived or did not survive the sinking of the Titanic. The result should be a vector of length 2.

Answer:

```
Titanic[Class = "Crew", Sex = "Female", Age = "Adult",]

##  No Yes
##   3  20
```

(c) (4 Points) Write an R expression that extracts the following matrix from the Titanic data set:

```
      Sex
Class  Female Male
  Crew      3  670
  1st       4  118
  2nd      13  154
  3rd      89  387
```

Describe what this matrix represents, i.e., which subgroup(s) from the Titanic passengers and crew.

Answer:

```
Titanic[,, Age = "Adult",Survived="No"]

##        Sex
## Class  Male Female
##    1st  118      4
##    2nd  154     13
##    3rd  387     89
##    Crew 670      3
```

Description:

Place your answer here The subgroup is Adults that did not survive.

(d) (4 Points) Write an R expression that extracts the following vector from the Titanic data set:

```
[1]  13 14 75 76
```

Describe what this vector represents, i.e., which subgroup(s) from the Titanic passengers and crew. Hint: I first extracted a matrix and then transformed this into a vector using `as.vector`.

Answer:

```
as.vector(c(Titanic[Class="3rd", ,Age="Child", Survived = "Yes"],
            Titanic[Class="3rd", ,Age="Adult", Survived = "Yes"]))
```

```
## [1] 13 14 75 76
```

Description:

Place your answer here The first first two are male and female children in the third class that survived. The second two are the same but adults.