###Name: Hui Lyu

#PART 1. Family Data

Load the data from the Web into your R session with the following command:

load(url("http://courseweb.lis.illinois.edu/~jguo24/family.rda"))

In the following exercises try to write your code to be as general as possible

so that it could still work if the family had 27 members in it or if the

variables were in a different order in the data frame.

Q1. (2 pts.)

The NHANES survey (the source of the family data) 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 family is obese and FALSE otherwise

```
> OW_NHANES = ((family$gender=="m") & (family$bmi>26)) | ((family$gender=="
f") & (family$bmi>25))
> OW_NHANES
 [1] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALS
E FALSE FALSE
```

Q2. (4 pts.)

Here is an alternative way to create the same vector that introduces

some useful functions and ideas

We will begin by creating 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, first create a vector of length 2 called OWval whose first element

is 26 and second element is 25.

```
> OWval = c(26,25)
> OWval
[1] 26 25
```

Create the OW_limit vector by subsetting OWval 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)

```
> OW_limit = OWval[as.numeric(family$gender)]
> OW_limit
 [1] 26 25 26 26 25 25 26 25 26 26 25 26 26 25
```

Finally, us OW_limit and bmi to create the desired logical vector, and

call it OW_NHANES2.

```
> OW_NHANES2 = family$bmi > OW_limit
> OW_NHANES2
 [1] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALS
E FALSE FALSE
```

Q3. (2 pts.)

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

To do this, start with the formula:

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

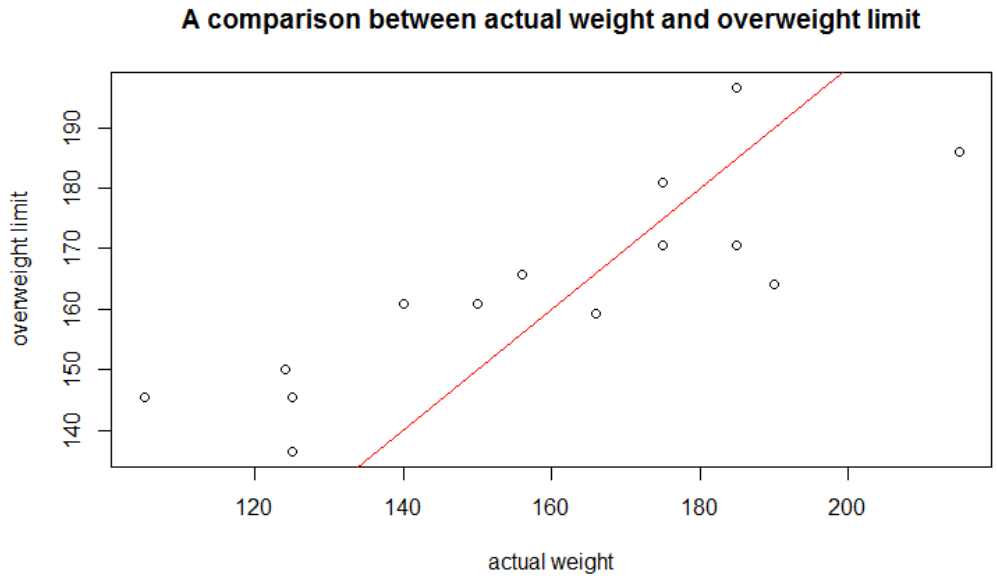
and find re-express it in terms of weight.

```
> OW_weight = OW_limit * 2.2 * (2.54/100 * family$height)^2
> OW_weight
 [1] 180.8254 145.3416 196.6569 165.6582 145.3416 164.0771 170.6402 149.91
91 170.6402 186.0288
[11] 159.2868 160.7501 160.7501 136.3997
```

Make a plot of the weight at which they would

be over weight against actual weight

```
> plot(family$weight, OW_weight, main = "A comparison between actual weigh
t and overweight limit", xlab = "actual weight", ylab = "overweight limit
")
> abline(a=0,b=1, col='red')
```



In the figure above, spots below the red line mean overweight, and spots above the red line mean not overweight. As can be seen clearly, five people are overweight among the all.

#PART 2. Baseball data

#Load the data into R.

#In order to access this data set we will install the relevant package and use the following code to do so:

install.packages("vcd")

library(vcd)

attach(Baseball)

#This means that the dataset Baseball was in the vcd package.

Q4. (4 pts.)

How many variables are in the dataset Baseball?

Your code below

length(Baseball)

Your answer here

25 variables

How many observations are in Baseball?

Your code below

dim(Baseball)

Your answer here

322 observations

Run the summary function and answer the following questions:

For the variable team87, which state had the most baseball players in the dataset?

> summary(Baseball$team87)

Atl Bal Bos Cal Chi Cin Cle Det Hou KC LA Mil Min Mon NY Oak Phi Pit SD SF Sea StL Tex Tor

12 15 11 12 22 11 13 13 12 14 13 14 14 13 24 13 13 16 10 15 11 9 12 10

alternative way:

> summary(Baseball)

Your answer here

New York

Make an observation about the variable, sal87, which is the yearly salary of the selected

baseball players in the dataset.

Who is the highest paid player in the data set?

Your code below

na.omit(Baseball[Baseball$sal87==max(Baseball$sal87, na.rm = TRUE),c("name1","name2")])

Your answer here

Eddie Murray

```
> na.omit(Baseball[Baseball$sal87==max(Baseball$sal87, na.rm = TRUE),c("na
me1","name2")])
    name1  name2
100 Eddie Murray
```

Q5. (2 pts.)

Now, we only want to use the baseball players in the National League.

This information is found through the variable, league86. The letter N indicates that the

player is in the National League. The letter A indicates that the player is in the American League.

Subset the new data frame so that all of the baseball players are in the National League,

and only keep the following variables: name1, name2, years, hits86, homer86, homeruns,rbi, and sal87.

To clarify, the variable, homer86 are the homeruns in that the player hit in '86, and the

variable homeruns are career homeruns for each player.

Call the new data Baseball1 (your code below)

```
> Baseball1 = Baseball[Baseball$league86=="N",c("name1","name2","years","h
its86","homer86","homeruns","rbi","sal87")]
> dim(Baseball1)
[1] 147   8
```

Q6. (2 pts.)

We want to remove unusually large values in order to further subset the data.

Use the quantile function to determine the 99% of variable sal87 (the salaries of the players in '87).

Then remove those baseball players that are above the 99th percentile.

Call this new dataset Baseball1 as well.

sal99 = quantile(Baseball1$sal87, 0.99, na.rm = TRUE)

which(Baseball1$sal87>sal99)

Baseball1 = Baseball1[-which(Baseball1$sal87>sal99),]

Baseball1 = na.omit(Baseball1)

```
> sal99 = quantile(Baseball1$sal87, 0.99, na.rm = TRUE)
> which(Baseball1$sal87>sal99)
[1]  97 104
> Baseball1 = Baseball1[-which(Baseball1$sal87>sal99),]
> Baseball1 = na.omit(Baseball1)
> dim(Baseball1)
[1] 122   8
```

Q7. (2 pts.)

Create a new vector called hitsperhome.

Divide hits86 by homer86, and this will create our new vector.

Now add this new variable to the data frame.

hitsperhome = Baseball1$hits86/Baseball1$homer86

Baseball1$hitsperhome = hitsperhome

hitsperhome == Inf means that its homer86 == 0

I think value Inf can be retained in Baseball1.

```
> hitsperhome = Baseball1$hits86/Baseball1$homer86
> Baseball1$hitsperhome = hitsperhome
> length(Baseball1)
[1] 9
```

Q8. (2 pts.)

Create a vector called hr15, this will be the number of homeruns hit in the year 1986

(NOT total) so use the variable, homer86, if this number is greater than 15, it is set to 15.

So if a player has 15 or more homeruns in that year, then hr15 will be 15, otherwise

it will be the actual number of homeruns.

hr15 = Baseball1$homer86

hr15[which(Baseball1$homer86>15)]=15

> hr15

```
 [1]  1  7  4 15 10  6 13 15  7  6 10  0  6  6  1 10 15  4 15  5  8 15 15  9 13  7  6  6  2 15
[31]  4  4  5  8 15  1  3 15  7 13  3 15  2 15 15  2 15 15 11 14 15  4 15 15 15  5  4 10 14  9
[61]  6  6  5  5 15 15  2  5  4 13  7 10  3 15  6 13 12  4  5 15 12  0  8  9  0  8  7  2  4 12
[91]  6  8  3 15  0  5  6  8  9 12  4  0 15  0  9  0  8  1  7 11  4  8  7 13  8  5  3 14 15 15
[121] 15  0  6  4  8  1  1  3  3  9  4 15  2  1  2 14 10  2  5  0 15  1  2 11  7
```

Q9. (2 pts.)

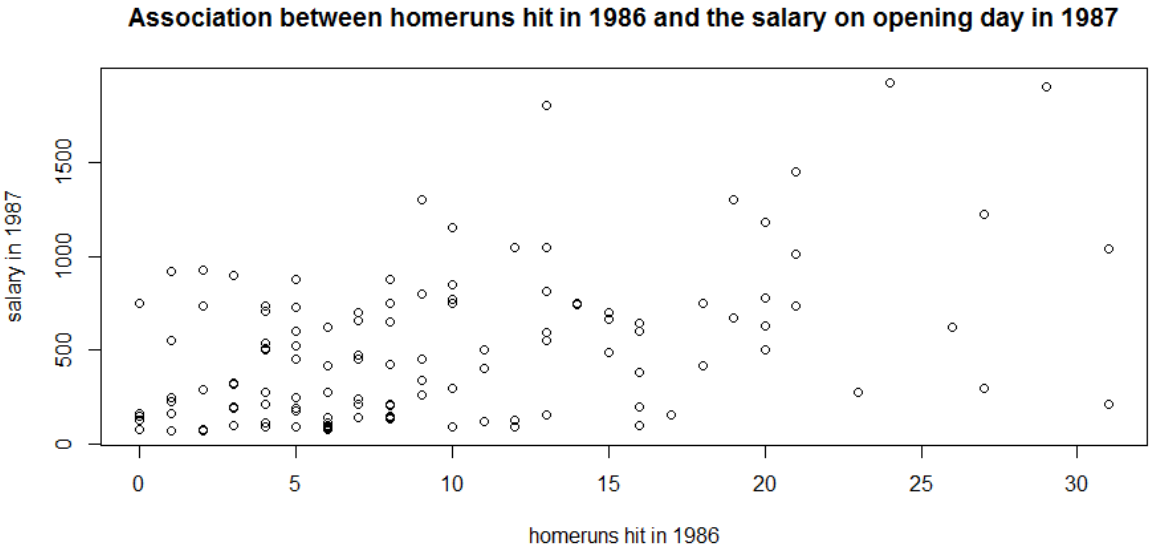
Find out if there is a significant association between homeruns hit in 1986, variable homer86,

#and the salary of the players on opening day in 1987, variable sal87 (which is USD 1000).

Answer this using several functions, including the plot function.

Make 3 observations below.

```
> plot(Baseball1$homer86,Baseball1$sal87, main = "Association between home
runs hit in 1986 and the salary on opening day in 1987", xlab = "homeruns
hit in 1986", ylab = "salary in 1987")
```



From the plot we get we see that when we plot the variable sal87 with homer86,

the points form some kind of line especially when homer86 is less than 20,

when the value of homer86 gets bigger the value of sal87 gets somehow proportionally bigger too,

we can suspect a positive correlation between homer86 and sal87. But there exists some

exceptions. Several players whose homer86 is less than 5 still have nearly 1000 salaries.

And several players whose homer86 is greater than 25 only have less than 500 salaries.

But on the whole, the association between homer86 and sal87 is positive correlation.

```
> df = data.frame(Baseball1$homer86,Baseball1$sal87)
> cor(na.omit(df),method="pearson")
                  Baseball1.homer86 Baseball1.sal87
Baseball1.homer86         1.0000000       0.4473575
Baseball1.sal87           0.4473575       1.0000000
```

Pearson coefficient is a good measure to calculate the coefficient of correlation.

Function cor() is used for this problem.

Since it is scaled between 1 (for a perfect positive correlation) to

-1 (for a perfect negative correlation), 0 would be complete randomness.

0.4473575 reflects some degree of positive correlation.

```
> model = lm(Baseball1$sal87~1+Baseball1$homer86)
> summary(model)
Call:
lm(formula = Baseball1$sal87 ~ 1 + Baseball1$homer86)

Residuals:
    Min      1Q  Median      3Q     Max
-805.48 -251.56  -43.98  216.38 1215.87

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        268.991     53.647   5.014 1.86e-06 ***
Baseball1$homer86   24.242      4.424   5.479 2.39e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 354.7 on 120 degrees of freedom
  (23 observations deleted due to missingness)
Multiple R-squared:  0.2001,   Adjusted R-squared:  0.1935
F-statistic: 30.02 on 1 and 120 DF,  p-value: 2.392e-07
```

The basic function to build linear model (linear regression) in R is to use the

lm() function. P-value is much smaller than 0.05, so there exists some significant association,

though multiple R-squared is not high enough. It is basically positive correlation.