Assignment_4

BANDELLE

2023-09-14

1. The dataset ChickWeight tracks the weights of 48 baby chickens (chicks) feed

```
four different diets. *Feel free to complete all parts of the exercise in a*
*single R pipeline at the end of the problem.*
a.  Load the dataset using

    ```r
    data(ChickWeight)
    ```
b. Look at the help files for the description of the columns.
c) Remove all the observations except for observations from day 10 or day 20.
    The tough part in this instruction is distinguishing between "and" and "or".
    Obviously there are no observations that occur from both day 10 AND day 20.
    Google 'R logical operators' to get an introduction to those, but the
    short answer is that and is `&` and or is `|`.
d) Calculate the mean and standard deviation of the chick weights for each
    diet group on days 10 and 20.
```

2. The OpenIntro textbook on statistics includes a data set on body dimensions.

```
*Instead of creating an R chunk for each step of this problem, create a*
*single R pipeline that performs each of the following tasks.*
a)  Load the file using

    ```r
    Body <- read.csv('http://www.openintro.org/stat/data/bdims.csv')
    ```
b) The column `sex` is coded as a 1 if the individual is male and 0 if female.
    This is a non-intuitive labeling system. Create a new column `sex.MF`
    that uses labels Male and Female. Use this column for the rest of
    the problem. _Hint: The `ifelse()` command will be_
    very convenient here. It functions similarly to the same command in Excel.
c) The columns `wgt` and `hgt` measure weight and height in kilograms and
    centimeters (respectively). Use these to calculate the Body Mass Index
    (BMI) for each individual where
    $$BMI=\frac{Weight\,(kg)}{\left[Height\,(m)\right]^{2}}$$
d) Double check that your calculated BMI column is correct by examining the
    summary statistics of the column (e.g. `summary(Body)`). BMI values should
    be between 18 to 40 or so. Did you make an error in your calculation?
e) The function `cut` takes a vector of continuous numerical data and creates
    a factor based on your given cut-points.
```

```r
# Define a continuous vector to convert to a factor
x <- 1:10

# divide range of x into three groups of equal length
cut(x, breaks=3)
```

```
##  [1] (0.991,4] (0.991,4] (0.991,4] (0.991,4] (4,7]     (4,7]     (4,7]
##  [8] (7,10]    (7,10]    (7,10]
## Levels: (0.991,4] (4,7] (7,10]
```

```r
# divide x into four groups, where I specify all 5 break points
cut(x, breaks = c(0, 2.5, 5.0, 7.5, 10))
```

```
##  [1] (0,2.5]  (0,2.5]  (2.5,5]  (2.5,5]  (2.5,5]  (5,7.5]  (5,7.5]  (7.5,10]
##  [9] (7.5,10] (7.5,10]
## Levels: (0,2.5] (2.5,5] (5,7.5] (7.5,10]
```

```r
# (0,2.5] (2.5,5] means 2.5 is included in first group
# right=FALSE changes this to make 2.5 included in the second

# divide x into 3 groups, but give them a nicer
# set of group names
cut(x, breaks=3, labels=c('Low','Medium','High'))
```

```
##  [1] Low    Low    Low    Low    Medium Medium Medium High   High   High
## Levels: Low Medium High
```

Create a new column of in the data frame that divides the age into decades (10-19, 20-29, 30-39, etc). Notice the oldest person in the study is 67.

```r
Body <- Body %>%
  mutate( Age.Grp = cut(age,
                        breaks=c(10,20,30,40,50,60,70),
                        right=FALSE))
```

f) Find the average BMI for each `Sex.MF` by `Age.Grp` combination.

3. Suppose we have a data frame with the following two variables:

```r
df <- tribble(
```

```
  ~SubjectID, ~Outcome,
  1, 'good',
  1, 'good',
  1, 'good',
  2, 'good',
  2, 'bad',
  2, 'good',
  3, 'bad',
  4, 'good',
  4, 'good')
```

```
The `SubjectID` represents a particular individual that has had multiple
measurements. What we want to know is what proportion of individuals were
consistently `good` for all outcomes they had observed. So in our toy example
set, subjects `1` and `4` where consistently good, so our answer should be
$50\%$. *Hint: The steps below help understand the thinking, but this problem*
*can be done in two lines of code.*
a)  As a first step, we will summarize each subject with a column
    denotes if all the subject's observations were `good`. This should result
    in a column of TRUE/FALSE values with one row for each subject. *The `all()`*
    *function should be quite useful here. The corresponding `any()` function*
    *is also useful to know about.*
b)  Calculate the proportion of subjects that where consistently good by
    calculating the `mean()` of the TRUE/FALSE values. _This works because_
    _TRUE/FALSE values are converted to 1/0 values and then averaged._
```

1a)

```
data("ChickWeight")
```

1b

```
?ChickWeight
```

```
## starting httpd help server ... done
```

1c

```
ChickWeight %>% filter(Time==10 | Time==20)
```

```
##    weight Time Chick Diet
## 1      93   10     1    1
## 2     199   20     1    1
## 3     103   10     2    1
## 4     209   20     2    1
## 5      99   10     3    1
## 6     198   20     3    1
## 7      87   10     4    1
## 8     160   20     4    1
## 9     106   10     5    1
## 10    220   20     5    1
## 11    124   10     6    1
## 12    160   20     6    1
```

```
## 13     112   10     7     1
## 14     288   20     7     1
## 15      93   10     8     1
## 16     125   20     8     1
## 17      96   10     9     1
## 18     100   20     9     1
## 19      81   10    10     1
## 20     120   20    10     1
## 21     139   10    11     1
## 22     181   20    11     1
## 23      88   10    12     1
## 24     195   20    12     1
## 25      67   10    13     1
## 26      91   20    13     1
## 27     128   10    14     1
## 28     259   20    14     1
## 29      68   10    15     1
## 30      51   10    16     1
## 31      89   10    17     1
## 32     133   20    17     1
## 33      71   10    19     1
## 34     144   20    19     1
## 35      73   10    20     1
## 36     115   20    20     1
## 37     163   10    21     2
## 38     318   20    21     2
## 39      95   10    22     2
## 40     164   20    22     2
## 41     103   10    23     2
## 42     170   20    23     2
## 43      68   10    24     2
## 44      76   20    24     2
## 45     124   10    25     2
## 46     259   20    25     2
## 47     114   10    26     2
## 48     236   20    26     2
## 49     100   10    27     2
## 50     185   20    27     2
## 51     114   10    28     2
## 52     212   20    28     2
## 53     106   10    29     2
## 54     279   20    29     2
## 55      98   10    30     2
## 56     157   20    30     2
## 57     102   10    31     3
## 58     235   20    31     3
## 59     129   10    32     3
## 60     291   20    32     3
## 61     111   10    33     3
## 62     156   20    33     3
## 63     134   10    34     3
## 64     327   20    34     3
## 65     158   10    35     3
## 66     361   20    35     3
```

```
## 67     116   10    36    3
## 68     225   20    36    3
## 69      83   10    37    3
## 70     169   20    37    3
## 71     109   10    38    3
## 72     280   20    38    3
## 73     109   10    39    3
## 74     250   20    39    3
## 75     120   10    40    3
## 76     295   20    40    3
## 77     124   10    41    4
## 78     199   20    41    4
## 79     126   10    42    4
## 80     269   20    42    4
## 81     157   10    43    4
## 82     199   20    43    4
## 83     118   10    44    4
## 84     117   10    45    4
## 85     197   20    45    4
## 86     120   10    46    4
## 87     231   20    46    4
## 88     123   10    47    4
## 89     210   20    47    4
## 90     125   10    48    4
## 91     303   20    48    4
## 92     128   10    49    4
## 93     233   20    49    4
## 94     122   10    50    4
## 95     264   20    50    4
```