ChickenData example

Andreia Carlos

12 August 2016

Introduction

This is an exercise with the ChickenWeight data from R Datasets Package.

The aim is to get data into a specific format to be used in a shinyApp. We want to present data by Time, Diet and the summary statistics for the chickens weight. For data processing and analysis we used dplyr package.

Data Processing

Steps:

1. Load `ChickWeight` data
2. Produce a new dataset with summaries of chicken weight, including `Number of cases (N)`, `Mean`, `Standard Deviation (SD)`, `Median`, `Minimum (Min)`and `Maximum (Max)`values.
3. Make the dataset into a "longer" format by converting the columns with statistical calculations into a column `Stats`and a column `Values`.
4. Move `Diet` groups from rows into columns.
5. Format data with *sprintf* function to present statistics into a consistent format, Mean as `Mean(SD)` and maximum and minimum values together as `Min-Max`.

1. Load ChickWeight data

```r
data("ChickWeight")
summary(ChickWeight)
```

```
##      weight           Time           Chick     Diet
##  Min.   : 35.0   Min.   : 0.00   13     : 12   1:220
##  1st Qu.: 63.0   1st Qu.: 4.00   9      : 12   2:120
##  Median :103.0   Median :10.00   20     : 12   3:120
##  Mean   :121.8   Mean   :10.72   10     : 12   4:118
##  3rd Qu.:163.8   3rd Qu.:16.00   17     : 12
##  Max.   :373.0   Max.   :21.00   19     : 12
##                                  (Other):506
```

Load required libraries

```r
library(dplyr)
library(tidyr) #for 'unite' function
library(knitr)
```

2. Produce a new dataset with summaries of chicken weight

```
DataStats <- ChickWeight %>%
        select(weight, Time, Diet) %>%
        group_by(Diet, Time) %>%
        summarise(
                N = n(),
                Mean = mean(weight),
                SD = sd(weight),
                Median = median(weight),
                Min = min(weight),
                Max = max(weight)
                ) %>%
        unite(Mean_SD, Mean, SD, sep = ".", remove = FALSE) %>% #Create new variable with MeanSD
        unite(Range, Min, Max, sep = "-", remove = FALSE)  #Create new column with Min-Max into the sam


kable(head(DataStats))
```

Diet	Time	N	Mean_SD	Mean	SD	Median	Range	Min	Max
1	0	20	41.4.0.99472291830968	41.40000	0.9947229	41	39-43	39	43
1	2	20	47.25.4.27815749814653	47.25000	4.2781575	49	35-51	35	51
1	4	19	56.4736842105263.4.12806682025795	56.47368	4.1280668	56	48-63	48	63
1	6	19	66.7894736842105.7.75728293905097	66.78947	7.7572829	67	51-84	51	84
1	8	19	79.6842105263158.13.7761977965981	79.68421	13.7761978	79	57-112	57	112
1	10	19	93.0526315789474.22.5424875000533	93.05263	22.5424875	93	51-139	51	139

DataStats has 48 observations and 10 variables

3. Make the dataset into a "longer" format by converting the columns with statistical calculations into a column

```
LongData <- DataStats  %>%
        gather("Statistics", "Value", 3:10) #selecting the columns that we want to rearrange

kable(head(LongData))
```

Diet	Time	Statistics	Value
1	0	N	20
1	2	N	20
1	4	N	19
1	6	N	19
1	8	N	19
1	10	N	19

LongData has 384 observations and 4 variables

4. Move `Diet` groups from rows into columns. This time using numbers attached to the columns of statistics so that we can reorder them afterwards

```r
TimesByDiet <- ChickWeight %>%
        select(weight, Time, Diet) %>%
        group_by(Diet, Time) %>%
        summarise( #Add numbers to the names of each column to order it afterwords
                N.1 = n(),
                Mean.2 = mean(weight),
                SD.3 = sd(weight),
                Median.4 = median(weight),
                Min.5 = min(weight),
                Max.6 = max(weight)
                ) %>%
        unite(Mean_SD.8, Mean.2, SD.3, sep = ".", remove = FALSE) %>%
        unite(Range.9, Min.5, Max.6, sep = "-", remove = FALSE)  %>%
        select(Time, Diet, N.1, Mean_SD.8, Median.4, Range.9) %>%
        gather("Statistics", "Value", 3:6) %>%
        spread(Diet, Value, sep = ".") %>% #Move diet groups from rows to columns
        separate(Statistics, c("Stats", "Order"), sep = "\\.", remove = TRUE) %>% #Separate Stats colum
        arrange(Time, Order) #reorder the dataset to have statistics by the appropriate order(N, Mean-S

kable(head(TimesByDiet))
```

Time	Stats	Order	Diet.1	Diet.2	Diet.3	Diet.4
0	N	1	20	10	10	10
0	Median	4	41	40.5	41	41
0	Mean_SD	8	41.4.0.99472291830968	40.7.1.49443411809733	40.8.1.03279555898864	41.1.0540925533894
0	Range	9	39-43	39-43	39-42	39-42
2	N	1	20	10	10	10
2	Median	4	49	48.5	49.5	51.5

This dataset has 48 observations and 7 variables

5. Format data with *sprintf* function to present statistics into a consistent format

```r
#By using mutate and sprintf, there is no longer need to attach numbers to the variable names to put th

Chick_print <- ChickWeight %>%
        select(weight, Time, Diet) %>%
        group_by(Diet, Time) %>%
        summarise(
                N = n(),
                Mean = mean(weight),
                SD = sd(weight),
                Median = median(weight),
                Min = min(weight),
                Max = max(weight)
                ) %>%
        mutate( #Creating new variables merging the previous ones and giving it the right format
                pN = as.character(N),
               pMeanSD = sprintf("%6.1f(%6.2f)", Mean, SD),
               pMedian = sprintf("%6.1f", Median),
               pMinMax = sprintf("%6.1f-%6.1f", Min, Max)
               ) %>%
        select(Diet, Time, N, pMeanSD, pMedian, pMinMax) %>% #select only the variables of interest
        gather("Statistics", "Value", 3:6) %>% #Get statistics into rows
        spread(Diet, Value, sep = ".") #Produce a dataset with diets in columns

kable(head(Chick_print))
```

Time	Statistics	Diet.1	Diet.2	Diet.3	Diet.4
0	N	20	10	10	10
0	pMeanSD	41.4(0.99)	40.7(1.49)	40.8(1.03)	41.0(1.05)
0	pMedian	41.0	40.5	41.0	41.0
0	pMinMax	39.0- 43.0	39.0- 43.0	39.0- 42.0	39.0- 42.0
2	N	20	10	10	10
2	pMeanSD	47.2(4.28)	49.4(2.88)	50.4(2.41)	51.8(1.93)

This dataset has 48 observations and 6 variables