Homework 4

rgk359 Rohit Kamat

This homework is due on Feb. 14, 2017 at 7:00pm. Please submit as a PDF file on Canvas.

Problem 1: (4 pts) Recall the `HairEyeColor` data set from an earlier in-class exercise. This data set contains counts of males and females with different combinations of hair and eye color.

```
HairEyeColor
```

```
## , , Sex = Male
##
##        Eye
## Hair    Brown Blue Hazel Green
##   Black    32   11    10     3
##   Brown    53   50    25    15
##   Red      10   10     7     7
##   Blond     3   30     5     8
##
## , , Sex = Female
##
##        Eye
## Hair    Brown Blue Hazel Green
##   Black    36    9     5     2
##   Brown    66   34    29    14
##   Red      16    7     7     7
##   Blond     4   64     5     8
```

I have split the data set into two data-frames, one `male` and one `female`. Using the `dplyr` and `tidyr` packages, make these data-frames tidy and then combine them into a single data-frame. Make sure that your final data-frame has a `sex` column indicating which data-frame the observations originally came from. **HINT**: You can use the `bind_rows` function to add rows from one data-frame onto another as long as both data-frames have identical column names.

```
male <- read.table(text="
Hair  Brown Blue Hazel Green
  Black    32   11    10     3
  Brown    53   50    25    15
  Red      10   10     7     7
  Blond     3   30     5     8
", head=T)

female <- read.table(text="
Hair    Brown Blue Hazel Green
  Black    36    9     5     2
  Brown    66   34    29    14
  Red      16    7     7     7
  Blond     4   64     5     8
", head=T)

male %>%gather(Eye,Count, Brown:Green) %>% mutate(Sex="Male")-> a
a
```

```
##     Hair   Eye Count  Sex
## 1  Black Brown    32 Male
## 2  Brown Brown    53 Male
## 3    Red Brown    10 Male
## 4  Blond Brown     3 Male
## 5  Black  Blue    11 Male
## 6  Brown  Blue    50 Male
## 7    Red  Blue    10 Male
## 8  Blond  Blue    30 Male
## 9  Black Hazel    10 Male
## 10 Brown Hazel    25 Male
## 11   Red Hazel     7 Male
## 12 Blond Hazel     5 Male
## 13 Black Green     3 Male
## 14 Brown Green    15 Male
## 15   Red Green     7 Male
## 16 Blond Green     8 Male
```

```
female %>% gather(Eye,Count, Brown:Green) %>% mutate(Sex="Female") -> b
b
```

```
##     Hair   Eye Count    Sex
## 1  Black Brown    36 Female
## 2  Brown Brown    66 Female
## 3    Red Brown    16 Female
## 4  Blond Brown     4 Female
## 5  Black  Blue     9 Female
## 6  Brown  Blue    34 Female
## 7    Red  Blue     7 Female
## 8  Blond  Blue    64 Female
## 9  Black Hazel     5 Female
## 10 Brown Hazel    29 Female
## 11   Red Hazel     7 Female
## 12 Blond Hazel     5 Female
## 13 Black Green     2 Female
## 14 Brown Green    14 Female
## 15   Red Green     7 Female
## 16 Blond Green     8 Female
```

```
bind_rows(a,b)-> final.table
final.table
```

```
##     Hair   Eye Count    Sex
## 1  Black Brown    32   Male
## 2  Brown Brown    53   Male
## 3    Red Brown    10   Male
## 4  Blond Brown     3   Male
## 5  Black  Blue    11   Male
## 6  Brown  Blue    50   Male
## 7    Red  Blue    10   Male
## 8  Blond  Blue    30   Male
## 9  Black Hazel    10   Male
## 10 Brown Hazel    25   Male
## 11   Red Hazel     7   Male
## 12 Blond Hazel     5   Male
## 13 Black Green     3   Male
## 14 Brown Green    15   Male
## 15   Red Green     7   Male
## 16 Blond Green     8   Male
## 17 Black Brown    36 Female
## 18 Brown Brown    66 Female
## 19   Red Brown    16 Female
## 20 Blond Brown     4 Female
## 21 Black  Blue     9 Female
## 22 Brown  Blue    34 Female
## 23   Red  Blue     7 Female
## 24 Blond  Blue    64 Female
## 25 Black Hazel     5 Female
## 26 Brown Hazel    29 Female
## 27   Red Hazel     7 Female
## 28 Blond Hazel     5 Female
## 29 Black Green     2 Female
## 30 Brown Green    14 Female
## 31   Red Green     7 Female
## 32 Blond Green     8 Female
```

Using the data-frame you created above, compute the total counts for each sex (i.e., the sum of the counts for each sex).

```
final.table %>% select(Count,Sex) %>% filter(Sex=="Male") %>% summarize(sum(Count))-> Ma
letotal
Maletotal
```

```
##   sum(Count)
## 1        279
```

```
final.table %>% select(Count,Sex) %>% filter(Sex=="Female") %>% summarize(sum(Count))->
Femaletotal
Femaletotal
```

```
##   sum(Count)
## 1        313
```

Problem 2: (3 pts) Recall that the `chickwts` data set contains information on the weight of chicks after being fed different feed supplements. The different feed supplements are labeled casein, horsebean, linseed, meatmeal, soybean, and sunflower in the `feed` column. I have created a new data-frame (`feed_names`), that contains the abbreviated names of different feed supplements. Using one of the `dplyr` join functions, combine the two data-frames so that there is an additional `feed_abbr` column and all of the original columns and rows in `chickwts` are retained. Which join function is most appropriate to use and why?

```
head(chickwts)
```

```
##   weight      feed
## 1    179 horsebean
## 2    160 horsebean
## 3    136 horsebean
## 4    227 horsebean
## 5    217 horsebean
## 6    168 horsebean
```

```
feed_names <- read.table(text="
feed feed_abbr
casein cs
whey wh
linseed  ls
meatmeal mm
fishmeal fm
soybean sb
sunflower sf
corn co
wheatbran wb
", head=T)

left_join(chickwts,feed_names) %>% head()
```

```
## Joining, by = "feed"
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factors with different levels, coercing to character vector
```

```
##   weight      feed feed_abbr
## 1    179 horsebean      <NA>
## 2    160 horsebean      <NA>
## 3    136 horsebean      <NA>
## 4    227 horsebean      <NA>
## 5    217 horsebean      <NA>
## 6    168 horsebean      <NA>
```

You want to use the left join function so that original columns of weight and feed do not change.

Problem 3: (3 pts) Make up your own data set which is **not** tidy and input it into R via the `text` option of `read.table()` . First, explain why it is not tidy. Then, using `dplyr` and/or `tidyr` convert it into a tidy data set.

```
love<- read.table(text = "
Gender     Single Relationship
 Male        30       70
 Female      50       50
", head=T)
```

It is not tidy because there needs to be columns of gender, relationship status, and the count rather than contigency table of the amount of a certain gender that are single or in a relationship.

```
love %>% gather(Relationship.Status, Count, Single:Relationship) %>% arrange(Gender) ->
real.love
real.love
```

```
##   Gender Relationship.Status Count
## 1 Female              Single    50
## 2 Female        Relationship    50
## 3   Male              Single    30
## 4   Male        Relationship    70
```