In-class worksheet 5

Jan 30, 2018

1. Tidy data

Is the `iris` dataset tidy? Explain why or why not.

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

It is tidy. The dataset contains five variables: sepal length, sepal width, petal length, petal width, and species name. All of these variables correspond to one column each, and each row in the data set corresponds to one observational unit (flower).

Is the `HairEyeColor` dataset tidy? Explain why or why not.

```
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
```

It is not. Columns correspond to different values of eye color. In a tidy data set, there would be one column listing eye colors and one listing hair colors, with values brown, blue, hazel, green (for eye color) and black, brown, red, blong (for hair color). Also, the two tables should be combined, and sex should be stored in an additional column.

2. Selecting rows and columns

All subsequent code will be based on the dplyr library, which is part of the tidyverse. So we first have to load this library:

```
library(tidyverse)
```

Now, using the dplyr function `filter()` , pick all the rows in the `iris` dataset that pertain to species setosa, and store them in a new table called `iris.setosa` .

```
iris.setosa <- filter(iris, Species=='setosa')
head(iris.setosa)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Pick all the rows in the `iris` dataset where species virginica has a sepal length > 7.

```
filter(iris, Sepal.Length>7 & Species=='virginica')
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 1           7.1         3.0          5.9         2.1 virginica
## 2           7.6         3.0          6.6         2.1 virginica
## 3           7.3         2.9          6.3         1.8 virginica
## 4           7.2         3.6          6.1         2.5 virginica
## 5           7.7         3.8          6.7         2.2 virginica
## 6           7.7         2.6          6.9         2.3 virginica
## 7           7.7         2.8          6.7         2.0 virginica
## 8           7.2         3.2          6.0         1.8 virginica
## 9           7.2         3.0          5.8         1.6 virginica
## 10          7.4         2.8          6.1         1.9 virginica
## 11          7.9         3.8          6.4         2.0 virginica
## 12          7.7         3.0          6.1         2.3 virginica
```

Are there any cases in the `iris` dataset for which the ratio of sepal length to sepal width exceeds the ratio of petal length to petal width? Use `filter()` to find out.

```
filter(iris, Sepal.Length/Sepal.Width > Petal.Length/Petal.Width)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 1          6.9         3.1          5.1         2.3 virginica
```

There is exactly one such observation, for species virginica.

Create a pared-down table which contains only data for species setosa and which only has the columns `Sepal.Length` and `Sepal.Width`. Store the result in a table called `iris.pared`.

```
# first extract all data for species setosa
iris.setosa <- filter(iris, Species=='setosa')
# now select sepal length and width
iris.pared <- select(iris.setosa, Sepal.Length, Sepal.Width)
head(iris.pared)
```

```
##   Sepal.Length Sepal.Width
## 1          5.1         3.5
## 2          4.9         3.0
## 3          4.7         3.2
## 4          4.6         3.1
## 5          5.0         3.6
## 6          5.4         3.9
```

3. Creating new data, arranging

Using the function `mutate()`, create a new data column that holds the ratio of sepal length to sepal width. Store the resulting table in a variable called `iris.ratio`.

```
iris.ratio <- mutate(iris, sepal.length.to.width = Sepal.Length/Sepal.Width)
head(iris.ratio)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
##   sepal.length.to.width
## 1              1.457143
## 2              1.633333
## 3              1.468750
## 4              1.483871
## 5              1.388889
## 6              1.384615
```

Order the `iris.ratio` table by species name and by increasing values of sepal length-to-width ratio.

```
iris.ratio.ordered <- arrange(iris.ratio, Species, sepal.length.to.width)
head(iris.ratio.ordered)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.2         4.1          1.5         0.1  setosa
## 2          4.6         3.6          1.0         0.2  setosa
## 3          5.7         4.4          1.5         0.4  setosa
## 4          5.5         4.2          1.4         0.2  setosa
## 5          5.1         3.8          1.5         0.3  setosa
## 6          5.1         3.8          1.9         0.4  setosa
##   sepal.length.to.width
## 1              1.268293
## 2              1.277778
## 3              1.295455
## 4              1.309524
## 5              1.342105
## 6              1.342105
```

4. Grouping and summarizing

Calculate the mean and standard deviation of the sepal lengths for each species. Do this by first creating a table grouped by species, which you call `iris.grouped` . Then run `summarize()` on that table.

```
iris.grouped <- group_by(iris, Species)
head(iris.grouped)
```

```
## # A tibble: 6 x 5
## # Groups:   Species [1]
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##          <dbl>       <dbl>        <dbl>       <dbl>  <fctr>
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
summarize(iris.grouped,
            mean.sepal.length = mean(Sepal.Length),
            sd.sepal.length = sd(Sepal.Length))
```

```
## # A tibble: 3 x 3
##      Species mean.sepal.length sd.sepal.length
##       <fctr>             <dbl>           <dbl>
## 1     setosa             5.006       0.3524897
## 2 versicolor             5.936       0.5161711
## 3  virginica             6.588       0.6358796
```

Use the function `n()` to count the number of observations for each species.

```
summarize(iris.grouped, count = n())
```

```
## # A tibble: 3 x 2
##      Species count
##       <fctr> <int>
## 1     setosa    50
## 2 versicolor    50
## 3  virginica    50
```

For each species, calculate the percentage of cases with sepal length > 5.5.

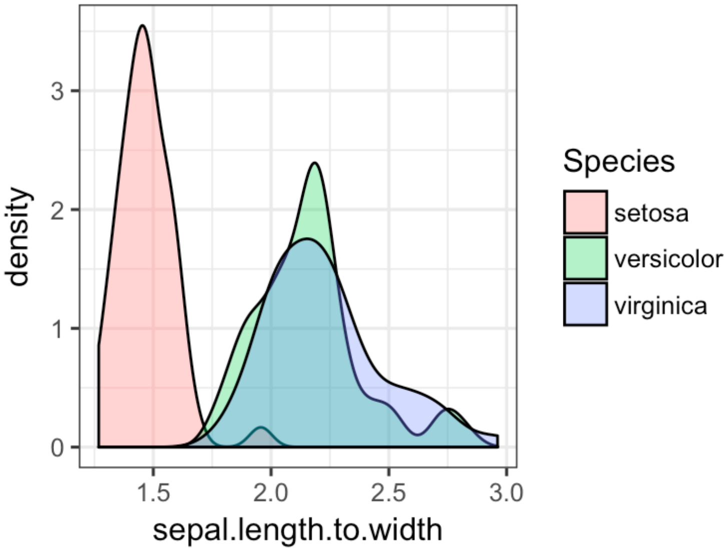
```
summarize(iris.grouped, percent = sum(Sepal.Length>5.5)/n())
```

```
## # A tibble: 3 x 2
##      Species percent
##       <fctr>   <dbl>
## 1     setosa    0.06
## 2 versicolor    0.78
## 3  virginica    0.98
```

5. If this was easy

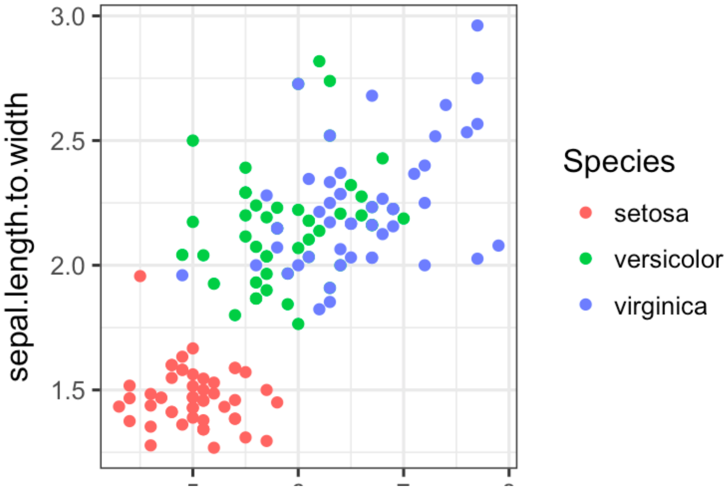
Take the `iris.ratio` data set you have created and plot the distribution of sepal length-to-width ratios for the three species.

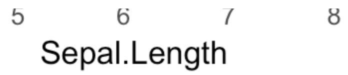
```
# ggplot2 is part of tidyverse, so we don't need to load it separately
theme_set(theme_bw(base_size=12)) # change the ggplot2 theme
ggplot(iris.ratio, aes(x=sepal.length.to.width, fill=Species)) + geom_density(a
lpha=.3)
```



Now plot sepal length-to-width ratios vs. sepal lengths. Does it look like there is a relationship between the length-to-width ratios and the lengths? Does it matter whether you consider each species individually or all together? How could you find out?

```
ggplot(iris.ratio, aes(y=sepal.length.to.width, x=Sepal.Length, color=Species))
+ geom_point()
```





There seems to be an overall trend of increasing length-to-width ratio with increasing length, but it seems that within each species there is little correlation between these values. We can check this by running a correlation analysis for each species:

```
# Setosa
setosa.ratio <- filter(iris.ratio, Species=='setosa')
cor.test(setosa.ratio$Sepal.Length, setosa.ratio$sepal.length.to.width)
```

```
##
##  Pearson's product-moment correlation
##
## data:  setosa.ratio$Sepal.Length and setosa.ratio$sepal.length.to.width
## t = -1.1218, df = 48, p-value = 0.2675
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4195161  0.1240336
## sample estimates:
##        cor
## -0.1598322
```

```
# Versicolor
versicolor.ratio <- filter(iris.ratio, Species=='versicolor')
cor.test(versicolor.ratio$Sepal.Length, versicolor.ratio$sepal.length.to.width)
```

```
##
##  Pearson's product-moment correlation
##
## data:  versicolor.ratio$Sepal.Length and versicolor.ratio$sepal.length.to.wi
dth
## t = 1.7522, df = 48, p-value = 0.08613
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.03559303  0.49008499
## sample estimates:
##       cor
## 0.2451838
```

```
# Virginica
virginica.ratio <- filter(iris.ratio, Species=='virginica')
cor.test(virginica.ratio$Sepal.Length, virginica.ratio$sepal.length.to.width)
```

```
## 
##  Pearson's product-moment correlation
## 
## data:  virginica.ratio$Sepal.Length and virginica.ratio$sepal.length.to.widt
h
## t = 3.5015, df = 48, p-value = 0.001011
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1975161 0.6480484
## sample estimates:
##       cor 
## 0.4510651
```

Virginica shows a significant correlation between sepal length and length-to-width ratio, the other two species do not.