dplyr_session2

Cody Flagg

Monday, May 04, 2015

Outline

1. rename() : rename columns in a data frame
2. arrange() : order rows
3. group_by() : summary statistics by groups you define, like ddply(data, grp, transform)
4. Piping/Chaining %>% : connect manipulations into a single call

- http://www.sharpsightlabs.com/dplyr-intro-data-manipulation-with-r/
- https://rpubs.com/justmarkham/dplyr-tutorial
- https://rpubs.com/mkapur/ModelingProject

Future Topics

4. SQL functions
    - Connecting to databases (http://cran.r-project.org/web/packages/dplyr/vignettes/databases.html) https://github.com/datacarpentry/R-dplyr-ecology/blob/gh-pages/03-data-analysis.Rmd https://github.com/justmarkham/dplyr-tutorial/blob/master/dplyr-tutorial.Rmd
    - Manipulating data from DBs
5. Two-table joins : multi-table data manipulation (http://cran.r-project.org/web/packages/dplyr/vignettes/two-table.html)
6. Data Cleaning (http://stackoverflow.com/questions/23642811/replace-parts-of-a-variable-using-numeric-indices-in-dplyr-rq=1)

1. rename() - rename columns

```
# Let's play with the iris dataset for a bit, RA Fisher first analyzed these data in 1936
# first print the names of the columns

data(iris)
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
## 7            4.6         3.4          1.4         0.3     setosa
## 8            5.0         3.4          1.5         0.2     setosa
## 9            4.4         2.9          1.4         0.2     setosa
```

```
## 10          4.9         3.1          1.5         0.1     setosa
## 11          5.4         3.7          1.5         0.2     setosa
## 12          4.8         3.4          1.6         0.2     setosa
## 13          4.8         3.0          1.4         0.1     setosa
## 14          4.3         3.0          1.1         0.1     setosa
## 15          5.8         4.0          1.2         0.2     setosa
## 16          5.7         4.4          1.5         0.4     setosa
## 17          5.4         3.9          1.3         0.4     setosa
## 18          5.1         3.5          1.4         0.3     setosa
## 19          5.7         3.8          1.7         0.3     setosa
## 20          5.1         3.8          1.5         0.3     setosa
## 21          5.4         3.4          1.7         0.2     setosa
## 22          5.1         3.7          1.5         0.4     setosa
## 23          4.6         3.6          1.0         0.2     setosa
## 24          5.1         3.3          1.7         0.5     setosa
## 25          4.8         3.4          1.9         0.2     setosa
## 26          5.0         3.0          1.6         0.2     setosa
## 27          5.0         3.4          1.6         0.4     setosa
## 28          5.2         3.5          1.5         0.2     setosa
## 29          5.2         3.4          1.4         0.2     setosa
## 30          4.7         3.2          1.6         0.2     setosa
## 31          4.8         3.1          1.6         0.2     setosa
## 32          5.4         3.4          1.5         0.4     setosa
## 33          5.2         4.1          1.5         0.1     setosa
## 34          5.5         4.2          1.4         0.2     setosa
## 35          4.9         3.1          1.5         0.2     setosa
## 36          5.0         3.2          1.2         0.2     setosa
## 37          5.5         3.5          1.3         0.2     setosa
## 38          4.9         3.6          1.4         0.1     setosa
## 39          4.4         3.0          1.3         0.2     setosa
## 40          5.1         3.4          1.5         0.2     setosa
## 41          5.0         3.5          1.3         0.3     setosa
## 42          4.5         2.3          1.3         0.3     setosa
## 43          4.4         3.2          1.3         0.2     setosa
## 44          5.0         3.5          1.6         0.6     setosa
## 45          5.1         3.8          1.9         0.4     setosa
## 46          4.8         3.0          1.4         0.3     setosa
## 47          5.1         3.8          1.6         0.2     setosa
## 48          4.6         3.2          1.4         0.2     setosa
## 49          5.3         3.7          1.5         0.2     setosa
## 50          5.0         3.3          1.4         0.2     setosa
## 51          7.0         3.2          4.7         1.4 versicolor
## 52          6.4         3.2          4.5         1.5 versicolor
## 53          6.9         3.1          4.9         1.5 versicolor
## 54          5.5         2.3          4.0         1.3 versicolor
## 55          6.5         2.8          4.6         1.5 versicolor
## 56          5.7         2.8          4.5         1.3 versicolor
## 57          6.3         3.3          4.7         1.6 versicolor
## 58          4.9         2.4          3.3         1.0 versicolor
## 59          6.6         2.9          4.6         1.3 versicolor
## 60          5.2         2.7          3.9         1.4 versicolor
## 61          5.0         2.0          3.5         1.0 versicolor
## 62          5.9         3.0          4.2         1.5 versicolor
## 63          6.0         2.2          4.0         1.0 versicolor
```

```
## 64          6.1         2.9          4.7         1.4 versicolor
## 65          5.6         2.9          3.6         1.3 versicolor
## 66          6.7         3.1          4.4         1.4 versicolor
## 67          5.6         3.0          4.5         1.5 versicolor
## 68          5.8         2.7          4.1         1.0 versicolor
## 69          6.2         2.2          4.5         1.5 versicolor
## 70          5.6         2.5          3.9         1.1 versicolor
## 71          5.9         3.2          4.8         1.8 versicolor
## 72          6.1         2.8          4.0         1.3 versicolor
## 73          6.3         2.5          4.9         1.5 versicolor
## 74          6.1         2.8          4.7         1.2 versicolor
## 75          6.4         2.9          4.3         1.3 versicolor
## 76          6.6         3.0          4.4         1.4 versicolor
## 77          6.8         2.8          4.8         1.4 versicolor
## 78          6.7         3.0          5.0         1.7 versicolor
## 79          6.0         2.9          4.5         1.5 versicolor
## 80          5.7         2.6          3.5         1.0 versicolor
## 81          5.5         2.4          3.8         1.1 versicolor
## 82          5.5         2.4          3.7         1.0 versicolor
## 83          5.8         2.7          3.9         1.2 versicolor
## 84          6.0         2.7          5.1         1.6 versicolor
## 85          5.4         3.0          4.5         1.5 versicolor
## 86          6.0         3.4          4.5         1.6 versicolor
## 87          6.7         3.1          4.7         1.5 versicolor
## 88          6.3         2.3          4.4         1.3 versicolor
## 89          5.6         3.0          4.1         1.3 versicolor
## 90          5.5         2.5          4.0         1.3 versicolor
## 91          5.5         2.6          4.4         1.2 versicolor
## 92          6.1         3.0          4.6         1.4 versicolor
## 93          5.8         2.6          4.0         1.2 versicolor
## 94          5.0         2.3          3.3         1.0 versicolor
## 95          5.6         2.7          4.2         1.3 versicolor
## 96          5.7         3.0          4.2         1.2 versicolor
## 97          5.7         2.9          4.2         1.3 versicolor
## 98          6.2         2.9          4.3         1.3 versicolor
## 99          5.1         2.5          3.0         1.1 versicolor
## 100         5.7         2.8          4.1         1.3 versicolor
## 101         6.3         3.3          6.0         2.5  virginica
## 102         5.8         2.7          5.1         1.9  virginica
## 103         7.1         3.0          5.9         2.1  virginica
## 104         6.3         2.9          5.6         1.8  virginica
## 105         6.5         3.0          5.8         2.2  virginica
## 106         7.6         3.0          6.6         2.1  virginica
## 107         4.9         2.5          4.5         1.7  virginica
## 108         7.3         2.9          6.3         1.8  virginica
## 109         6.7         2.5          5.8         1.8  virginica
## 110         7.2         3.6          6.1         2.5  virginica
## 111         6.5         3.2          5.1         2.0  virginica
## 112         6.4         2.7          5.3         1.9  virginica
## 113         6.8         3.0          5.5         2.1  virginica
## 114         5.7         2.5          5.0         2.0  virginica
## 115         5.8         2.8          5.1         2.4  virginica
## 116         6.4         3.2          5.3         2.3  virginica
## 117         6.5         3.0          5.5         1.8  virginica
```

```
## 118          7.7         3.8          6.7         2.2  virginica
## 119          7.7         2.6          6.9         2.3  virginica
## 120          6.0         2.2          5.0         1.5  virginica
## 121          6.9         3.2          5.7         2.3  virginica
## 122          5.6         2.8          4.9         2.0  virginica
## 123          7.7         2.8          6.7         2.0  virginica
## 124          6.3         2.7          4.9         1.8  virginica
## 125          6.7         3.3          5.7         2.1  virginica
## 126          7.2         3.2          6.0         1.8  virginica
## 127          6.2         2.8          4.8         1.8  virginica
## 128          6.1         3.0          4.9         1.8  virginica
## 129          6.4         2.8          5.6         2.1  virginica
## 130          7.2         3.0          5.8         1.6  virginica
## 131          7.4         2.8          6.1         1.9  virginica
## 132          7.9         3.8          6.4         2.0  virginica
## 133          6.4         2.8          5.6         2.2  virginica
## 134          6.3         2.8          5.1         1.5  virginica
## 135          6.1         2.6          5.6         1.4  virginica
## 136          7.7         3.0          6.1         2.3  virginica
## 137          6.3         3.4          5.6         2.4  virginica
## 138          6.4         3.1          5.5         1.8  virginica
## 139          6.0         3.0          4.8         1.8  virginica
## 140          6.9         3.1          5.4         2.1  virginica
## 141          6.7         3.1          5.6         2.4  virginica
## 142          6.9         3.1          5.1         2.3  virginica
## 143          5.8         2.7          5.1         1.9  virginica
## 144          6.8         3.2          5.9         2.3  virginica
## 145          6.7         3.3          5.7         2.5  virginica
## 146          6.7         3.0          5.2         2.3  virginica
## 147          6.3         2.5          5.0         1.9  virginica
## 148          6.5         3.0          5.2         2.0  virginica
## 149          6.2         3.4          5.4         2.3  virginica
## 150          5.9         3.0          5.1         1.8  virginica
```

```
# list the column names first, so you know what you're working with
names(iris)
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"
## [5] "Species"
```

```
# The periods in each column name are kind of annoying, let's fix that
rename(iris, petLength = Petal.Length,
  petWidth = Petal.Width,
  sepWidth = Sepal.Width,
  sepLength = Sepal.Length) -> iris2 # turn the assignment around, remember this for piping later

# rename and select have special arguments for finding columns
select(iris, matches("sepal", ignore.case = FALSE)) %>% head # case sensitive
```

```
## data frame with 0 columns and 6 rows
```

```
select(iris, matches("sepal")) %>% head # not case sensitive
```

```
##   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
```

```
# another example - notice that it is not case sensitive
select(iris, contains("l.l")) %>% head
```

```
##   Sepal.Length Petal.Length
## 1          5.1          1.4
## 2          4.9          1.4
## 3          4.7          1.3
## 4          4.6          1.5
## 5          5.0          1.4
## 6          5.4          1.7
```

```
# similar to a regex function i.e. stringr::sub_str()
select(iris, ends_with("th")) %>% head # the regex starts at the end of a string; if end of string does
```

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

```
# what if you have lots of columns, named/numbered something?
# create a silly example
df1 = data.frame(x05 = rnorm(5), x06= runif(5), x07 = rpois(5, 2), type = sample(letters, 5))
# now select a few columns
select(df1, num_range("x", 5:6, width=2)) # the width argument specifies that the string is at least 2
```

```
##           x05       x06
## 1  0.06081869 0.2043535
## 2  0.14850527 0.1432178
## 3 -0.94231104 0.8971633
## 4  1.02005376 0.5503973
## 5 -0.84575074 0.2712184
```

2. arrange() - re-order rows by column

```
# arrange(dataframe, column1, column2, ...)
head(arrange(iris2, desc(Species), sepLength))
```

```
##   sepLength sepWidth petLength petWidth   Species
## 1       4.9      2.5       4.5      1.7 virginica
## 2       5.6      2.8       4.9      2.0 virginica
## 3       5.7      2.5       5.0      2.0 virginica
## 4       5.8      2.7       5.1      1.9 virginica
## 5       5.8      2.8       5.1      2.4 virginica
## 6       5.8      2.7       5.1      1.9 virginica
```

```
# the piping way:
arrange(iris2, desc(Species), sepLength) %>% head
```

```
##   sepLength sepWidth petLength petWidth   Species
## 1       4.9      2.5       4.5      1.7 virginica
## 2       5.6      2.8       4.9      2.0 virginica
## 3       5.7      2.5       5.0      2.0 virginica
## 4       5.8      2.7       5.1      1.9 virginica
## 5       5.8      2.8       5.1      2.4 virginica
## 6       5.8      2.7       5.1      1.9 virginica
```

3. group_by() - equivalent to plyr::ddply()

```
library(plyr)
```

```
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
# mutate iris2, replace iris2 with mutated data
iris2 %>% mutate(sepalRatio = sepWidth/sepLength, # a simple ratio
                 avgSepLen = mean(sepLength)) -> iris2 # can use mutate to do column-wise operations to

# transumute only outputs new columns
transmute(iris2, sepalRatio = sepWidth/sepLength)
```

```
##     sepalRatio
## 1    0.6862745
## 2    0.6122449
## 3    0.6808511
## 4    0.6739130
## 5    0.7200000
## 6    0.7222222
```

```
## 7    0.7391304
## 8    0.6800000
## 9    0.6590909
## 10   0.6326531
## 11   0.6851852
## 12   0.7083333
## 13   0.6250000
## 14   0.6976744
## 15   0.6896552
## 16   0.7719298
## 17   0.7222222
## 18   0.6862745
## 19   0.6666667
## 20   0.7450980
## 21   0.6296296
## 22   0.7254902
## 23   0.7826087
## 24   0.6470588
## 25   0.7083333
## 26   0.6000000
## 27   0.6800000
## 28   0.6730769
## 29   0.6538462
## 30   0.6808511
## 31   0.6458333
## 32   0.6296296
## 33   0.7884615
## 34   0.7636364
## 35   0.6326531
## 36   0.6400000
## 37   0.6363636
## 38   0.7346939
## 39   0.6818182
## 40   0.6666667
## 41   0.7000000
## 42   0.5111111
## 43   0.7272727
## 44   0.7000000
## 45   0.7450980
## 46   0.6250000
## 47   0.7450980
## 48   0.6956522
## 49   0.6981132
## 50   0.6600000
## 51   0.4571429
## 52   0.5000000
## 53   0.4492754
## 54   0.4181818
## 55   0.4307692
## 56   0.4912281
## 57   0.5238095
## 58   0.4897959
## 59   0.4393939
## 60   0.5192308
```

```
## 61   0.4000000
## 62   0.5084746
## 63   0.3666667
## 64   0.4754098
## 65   0.5178571
## 66   0.4626866
## 67   0.5357143
## 68   0.4655172
## 69   0.3548387
## 70   0.4464286
## 71   0.5423729
## 72   0.4590164
## 73   0.3968254
## 74   0.4590164
## 75   0.4531250
## 76   0.4545455
## 77   0.4117647
## 78   0.4477612
## 79   0.4833333
## 80   0.4561404
## 81   0.4363636
## 82   0.4363636
## 83   0.4655172
## 84   0.4500000
## 85   0.5555556
## 86   0.5666667
## 87   0.4626866
## 88   0.3650794
## 89   0.5357143
## 90   0.4545455
## 91   0.4727273
## 92   0.4918033
## 93   0.4482759
## 94   0.4600000
## 95   0.4821429
## 96   0.5263158
## 97   0.5087719
## 98   0.4677419
## 99   0.4901961
## 100  0.4912281
## 101  0.5238095
## 102  0.4655172
## 103  0.4225352
## 104  0.4603175
## 105  0.4615385
## 106  0.3947368
## 107  0.5102041
## 108  0.3972603
## 109  0.3731343
## 110  0.5000000
## 111  0.4923077
## 112  0.4218750
## 113  0.4411765
## 114  0.4385965
```

```
## 115  0.4827586
## 116  0.5000000
## 117  0.4615385
## 118  0.4935065
## 119  0.3376623
## 120  0.3666667
## 121  0.4637681
## 122  0.5000000
## 123  0.3636364
## 124  0.4285714
## 125  0.4925373
## 126  0.4444444
## 127  0.4516129
## 128  0.4918033
## 129  0.4375000
## 130  0.4166667
## 131  0.3783784
## 132  0.4810127
## 133  0.4375000
## 134  0.4444444
## 135  0.4262295
## 136  0.3896104
## 137  0.5396825
## 138  0.4843750
## 139  0.5000000
## 140  0.4492754
## 141  0.4626866
## 142  0.4492754
## 143  0.4655172
## 144  0.4705882
## 145  0.4925373
## 146  0.4477612
## 147  0.3968254
## 148  0.4615385
## 149  0.5483871
## 150  0.5084746
```

```
# group the data by column "Species" - summarise by mean sepal and petal lengths
iris2 %>% group_by(Species) %>% summarise(mean(sepLength), mean(petLength))
```

```
##   mean(sepLength) mean(petLength)
## 1        5.843333           3.758
```

```
# how you would do this in plyr; I think it makes more sense to use a single function call
# syntax: ddply(dataframe, groups, special function, function calls)
head(ddply(iris2, .(Species), summarize,
      avgSepLen = mean(sepLength)))
```

```
##      Species avgSepLen
## 1     setosa     5.006
## 2 versicolor     5.936
## 3  virginica     6.588
```

```r
# transform adds column(s) all at once
head(ddply(iris2, .(Species), transform,
      avgSepLen = mean(sepLength)))
```

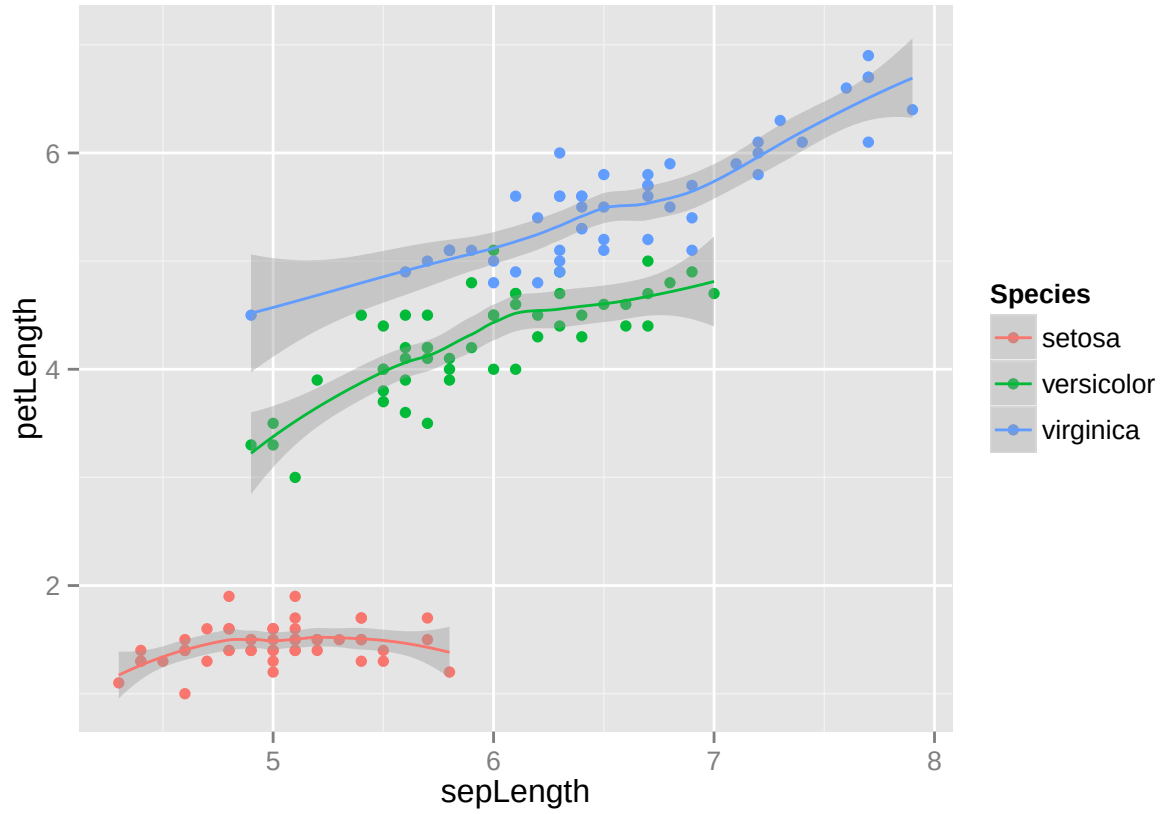
```
##   sepLength sepWidth petLength petWidth Species sepalRatio avgSepLen
## 1       5.1      3.5       1.4      0.2  setosa  0.6862745     5.006
## 2       4.9      3.0       1.4      0.2  setosa  0.6122449     5.006
## 3       4.7      3.2       1.3      0.2  setosa  0.6808511     5.006
## 4       4.6      3.1       1.5      0.2  setosa  0.6739130     5.006
## 5       5.0      3.6       1.4      0.2  setosa  0.7200000     5.006
## 6       5.4      3.9       1.7      0.4  setosa  0.7222222     5.006
```

```r
# mutate lets you adds column(s) sequentially, so they can be referenced internally within the function
head(ddply(iris2, .(Species), mutate,
      avgSepLen = mean(sepLength), ratko = avgSepLen*10)) # ratko references the previous column: avgSe
```

```
##   sepLength sepWidth petLength petWidth Species sepalRatio avgSepLen ratko
## 1       5.1      3.5       1.4      0.2  setosa  0.6862745     5.006 50.06
## 2       4.9      3.0       1.4      0.2  setosa  0.6122449     5.006 50.06
## 3       4.7      3.2       1.3      0.2  setosa  0.6808511     5.006 50.06
## 4       4.6      3.1       1.5      0.2  setosa  0.6739130     5.006 50.06
## 5       5.0      3.6       1.4      0.2  setosa  0.7200000     5.006 50.06
## 6       5.4      3.9       1.7      0.4  setosa  0.7222222     5.006 50.06
```

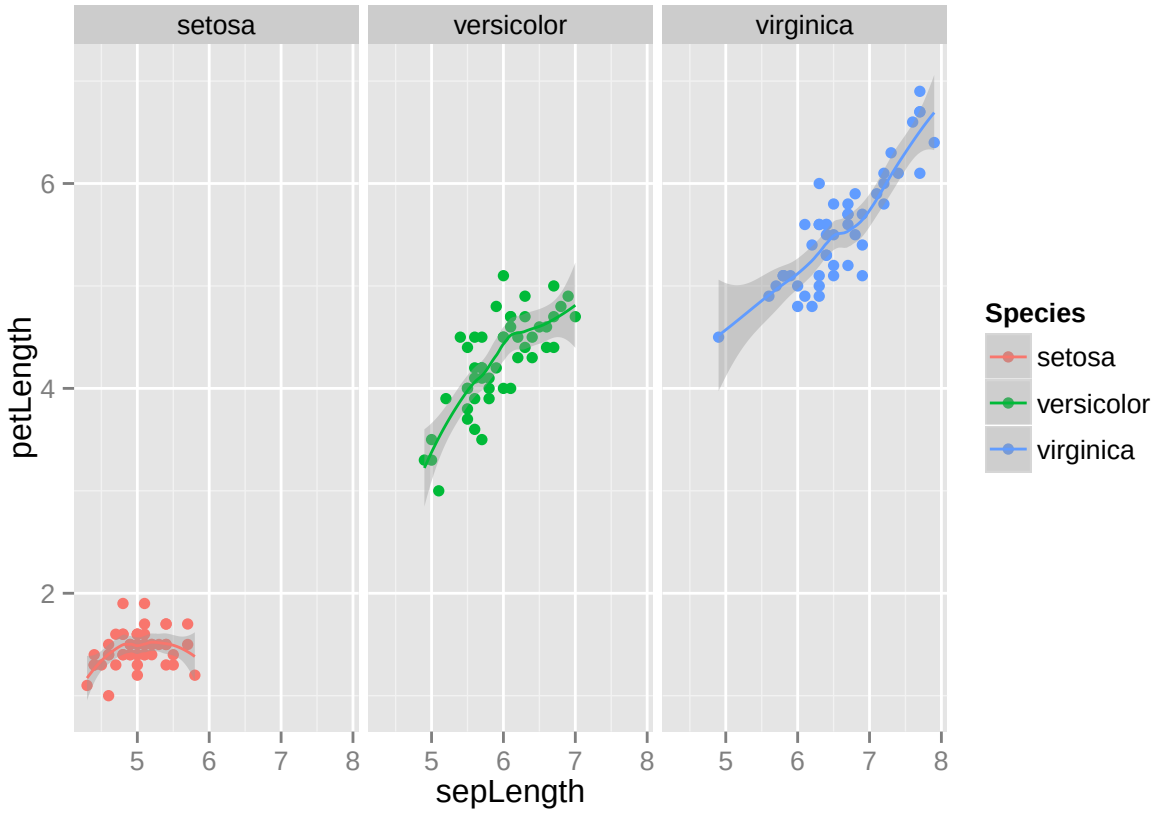
```r
# can we use a pipe to plot? Si se puede, a few examples below:
# plot the groups on the same panel, using color to identify groups
iris2 %>% group_by(Species) %>% ggplot(aes(sepLength,petLength, color = Species)) + geom_point() + geom
```

```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to cl
```




```
# facet_wrap the different groups onto different panels
iris2 %>% group_by(Species) %>% ggplot(aes(sepLength,petLength, color = Species)) + geom_point() + geom_
```

```
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to ch
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to ch
## geom_smooth: method="auto" and size of largest group is <1000, so using loess. Use 'method = x' to ch
```




```
# sfsmisc as an alternative for plotting?

# what about modelling; Can we fit a linear model to each group, then examine the summary data? - not s
iris2 %>% group_by(Species) %>% do(model = lm(sepLength ~ petLength, data = .)) -> iris.mods

# look at the structure - HOW DO I GET TO YOU?
str(iris.mods[1,2])
```

```
## Classes 'tbl_df' and 'data.frame':   1 obs. of  1 variable:
##  $ model:List of 1
##   ..$ :List of 12
##   .. ..$ coefficients : Named num  4.213 0.542
##   .. .. ..- attr(*, "names")= chr  "(Intercept)" "petLength"
##   .. ..$ residuals    : Named num  0.1276 -0.0724 -0.2181 -0.4266 0.0276 ...
##   .. .. ..- attr(*, "names")= chr  "1" "2" "3" "4" ...
##   .. ..$ effects      : Named num  -35.3978 0.6592 -0.2232 -0.4456 0.0156 ...
##   .. .. ..- attr(*, "names")= chr  "(Intercept)" "petLength" "" "" ...
##   .. ..$ rank         : int 2
##   .. ..$ fitted.values: Named num  4.97 4.97 4.92 5.03 4.97 ...
##   .. .. ..- attr(*, "names")= chr  "1" "2" "3" "4" ...
##   .. ..$ assign       : int  0 1
##   .. ..$ qr           :List of 5
##   .. .. ..$ qr   : num [1:50, 1:2] -7.071 0.141 0.141 0.141 0.141 ...
##   .. .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. .. ..$ : chr  "1" "2" "3" "4" ...
##   .. .. .. .. ..$ : chr  "(Intercept)" "petLength"
```

```
##   .. .. .. ..- attr(*, "assign")= int  0 1
##   .. .. ..$ qraux: num  1.14 1.04
##   .. .. ..$ pivot: int  1 2
##   .. .. ..$ tol  : num 1e-07
##   .. .. ..$ rank : int 2
##   .. .. ..- attr(*, "class")= chr "qr"
##   .. ..$ df.residual  : int 48
##   .. ..$ xlevels      : Named list()
##   .. ..$ call         : language lm(formula = sepLength ~ petLength, data = .)
##   .. ..$ terms        :Classes 'terms', 'formula' length 3 sepLength ~ petLength
##   .. .. .. ..- attr(*, "variables")= language list(sepLength, petLength)
##   .. .. .. ..- attr(*, "factors")= int [1:2, 1] 0 1
##   .. .. .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. .. .. ..$ : chr [1:2] "sepLength" "petLength"
##   .. .. .. .. .. ..$ : chr "petLength"
##   .. .. .. ..- attr(*, "term.labels")= chr "petLength"
##   .. .. .. ..- attr(*, "order")= int 1
##   .. .. .. ..- attr(*, "intercept")= int 1
##   .. .. .. ..- attr(*, "response")= int 1
##   .. .. .. ..- attr(*, ".Environment")=<environment: 0x000000000a0f2ff0>
##   .. .. .. ..- attr(*, "predvars")= language list(sepLength, petLength)
##   .. .. .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
##   .. .. .. .. ..- attr(*, "names")= chr [1:2] "sepLength" "petLength"
##   .. ..$ model        :'data.frame': 50 obs. of  2 variables:
##   .. .. ..$ sepLength: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##   .. .. ..$ petLength: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##   .. .. ..- attr(*, "terms")=Classes 'terms', 'formula' length 3 sepLength ~ petLength
##   .. .. .. .. ..- attr(*, "variables")= language list(sepLength, petLength)
##   .. .. .. .. ..- attr(*, "factors")= int [1:2, 1] 0 1
##   .. .. .. .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. .. .. .. ..$ : chr [1:2] "sepLength" "petLength"
##   .. .. .. .. .. .. ..$ : chr "petLength"
##   .. .. .. .. ..- attr(*, "term.labels")= chr "petLength"
##   .. .. .. .. ..- attr(*, "order")= int 1
##   .. .. .. .. ..- attr(*, "intercept")= int 1
##   .. .. .. .. ..- attr(*, "response")= int 1
##   .. .. .. .. ..- attr(*, ".Environment")=<environment: 0x000000000a0f2ff0>
##   .. .. .. .. ..- attr(*, "predvars")= language list(sepLength, petLength)
##   .. .. .. .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
##   .. .. .. .. .. ..- attr(*, "names")= chr [1:2] "sepLength" "petLength"
##   .. ..- attr(*, "class")= chr "lm"
```