

8. Exploring Data with dplyr (1)

CT1100 - J. Duggan

Overview

- Visualisation is an important tool for insight generation, but it's rare that you get the data in exactly the right form you need" (Wickham and Grolemund 2017)
 - Create new variables
 - Create summaries
 - Order data
- dplyr package is designed for data transformation

- All verbs (functions) work similarly
- The first argument is a data frame/tibble
- The subsequent arguments decide what to do with the data frame
- The result is a data frame (supports chaining of steps)

Function	Purpose
filter()	Pick observations by their values
arrange()	Reorder the rows
select()	Pick variables by their names
mutate()	Create new variables with functions of existing variables
summarise()	Collapse many values down to a single summary

Sample Data set ggplot2::mpg

```
## Observations: 234
## Variables: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi"
## $ model         <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4"
## $ displ         <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8
## $ year          <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008
## $ cyl           <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6
## $ trans         <chr> "auto(l5)", "manual(m5)", "manual(m6)"
## $ drv           <chr> "f", "f", "f", "f", "f", "f", "f", "f", "f", "f", "f", "f", "f"
## $ cty           <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 20, 20, 20
## $ hwy           <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 28, 28, 28
## $ fl            <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p"
## $ class         <chr> "compact", "compact", "compact", "compact", "compact", "compact", "compact", "compact", "compact", "compact", "compact", "compact", "compact"
```

(1) filter()

- Subset observations based on their values.
- First argument the name of the data frame
- Subsequent arguments are expressions that filter the data frame
- Only includes rows that have no missing values

```
filter(mpg,manufacturer=="audi",year==1999,model=="a4")
```

```
## # A tibble: 4 x 11
```

```
##   manufacturer model displ  year   cyl trans  drv    cty  
##   <chr>          <chr> <dbl> <int> <int> <chr> <chr> <int> <chr>  
## 1 audi          a4      1.8  1999     4 auto(~ f      18  
## 2 audi          a4      1.8  1999     4 manua~ f      21  
## 3 audi          a4      2.8  1999     6 auto(~ f      16  
## 4 audi          a4      2.8  1999     6 manua~ f      18
```

Cars with highest mpg, lowest mpg?

```
filter(mpg,hwy==max(hwy))
```

```
## # A tibble: 2 x 11
```

	manufacturer	model	displ	year	cyl	trans	drv	cty
	<chr>	<chr>	<dbl>	<int>	<int>	<chr>	<chr>	<int>
## 1	volkswagen	jetta	1.9	1999	4	manu~	f	33
## 2	volkswagen	new b~	1.9	1999	4	manu~	f	35

```
filter(mpg,hwy==min(hwy))
```

```
## # A tibble: 5 x 11
```

	manufacturer	model	displ	year	cyl	trans	drv	cty
	<chr>	<chr>	<dbl>	<int>	<int>	<chr>	<chr>	<int>
## 1	dodge	dakot~	4.7	2008	8	auto~	4	9
## 2	dodge	duran~	4.7	2008	8	auto~	4	9
## 3	dodge	ram 1~	4.7	2008	8	auto~	4	9
## 4	dodge	ram 1~	4.7	2008	8	manu~	4	9

Challenge 2.1

- List the cars with an average city mpg greater than the median.
- Show the cars with the maximum displacement

(2) arrange()

- Changes the order of rows.
- Takes a data frame and a set of column names to order by

```
arrange(mpg,displ)
```

```
## # A tibble: 234 x 11
```

##	manufacturer	model	displ	year	cyl	trans	drv	cty
##	<chr>	<chr>	<dbl>	<int>	<int>	<chr>	<chr>	<int>
##	1 honda	civic	1.6	1999	4	manu~	f	28
##	2 honda	civic	1.6	1999	4	auto~	f	24
##	3 honda	civic	1.6	1999	4	manu~	f	25
##	4 honda	civic	1.6	1999	4	manu~	f	23
##	5 honda	civic	1.6	1999	4	auto~	f	24
##	6 audi	a4	1.8	1999	4	auto~	f	18
##	7 audi	a4	1.8	1999	4	manu~	f	21
##	8 audi	a4 q~	1.8	1999	4	manu~	4	18
##	9 audi	a4 q~	1.8	1999	4	auto~	4	16

Show in descending order

```
arrange(mpg, desc(displ))
```

```
## # A tibble: 234 x 11
##   manufacturer model displ  year   cyl trans drv      cty
##   <chr>          <chr> <dbl> <int> <int> <chr> <chr> <int> <chr>
## 1 chevrolet      corv~    7   2008     8 manu~ r      15
## 2 chevrolet      k150~   6.5  1999     8 auto~ 4      14
## 3 chevrolet      corv~   6.2  2008     8 manu~ r      16
## 4 chevrolet      corv~   6.2  2008     8 auto~ r      15
## 5 jeep           gran~   6.1  2008     8 auto~ 4      11
## 6 chevrolet      c150~    6   2008     8 auto~ r      12
## 7 dodge          dura~   5.9  1999     8 auto~ 4      11
## 8 dodge          ram ~   5.9  1999     8 auto~ 4      11
## 9 chevrolet      c150~   5.7  1999     8 auto~ r      13
## 10 chevrolet      corv~   5.7  1999     8 manu~ r      16
## # ... with 224 more rows
```

Add an extra sort column

```
arrange(mpg, desc(year), desc(displ))
```

```
## # A tibble: 234 x 11
##   manufacturer model displ  year   cyl trans drv      cty
##   <chr>          <chr> <dbl> <int> <int> <chr> <chr> <int> <chr>
## 1 chevrolet      corv~    7   2008     8 manu~ r      15
## 2 chevrolet      corv~   6.2  2008     8 manu~ r      16
## 3 chevrolet      corv~   6.2  2008     8 auto~ r      15
## 4 jeep           gran~   6.1  2008     8 auto~ 4      11
## 5 chevrolet      c150~    6   2008     8 auto~ r      12
## 6 dodge          dura~   5.7  2008     8 auto~ 4      13
## 7 dodge          ram ~   5.7  2008     8 auto~ 4      13
## 8 jeep           gran~   5.7  2008     8 auto~ 4      13
## 9 toyota         land~   5.7  2008     8 auto~ 4      13
## 10 nissan         path~   5.6  2008     8 auto~ 4      12
## # ... with 224 more rows
```

The pipe operator

- The pipe `%>%` comes from the `magrittr` package (Stefan Milton Bache)
- Helps to write code that is easier to read and understand
 - `x %>% f(y)` turns into `f(x, y)`

```
mpg %>% select(model, displ, cty) %>% slice(1:2)
```

```
## # A tibble: 2 x 3
##   model displ  cty
##   <chr> <dbl> <int>
## 1 a4      1.8     18
## 2 a4      1.8     21
```

Summary One

- dplyr - a grammar of data manipulation
- Five verbs
 - **filter()**
 - **arrange()**
- Usefully combined with `%>%` operator