CT5132/CT5148 Week 12 Exercises

James McDermott

NUI Galway

Exercises

Again this week, our exercises are extracted from the lecture slides/videos, and solutions are given below.

We won't look at the exercises for `ggplot`, as the Data Visualisation module is coming up soon where you'll see that in much more detail.

Exercises (dplyr joins)

1. Read the three data files rentals.csv, movies.csv, customers.csv, all in the data/ directory, as tibbles.
2. Optional: get R to read the Date column correctly. Hint: https://readr.tidyverse.org/reference/parse_datetime.html
3. Using a dplyr join command, create a table showing the customer name and address for every rental.
4. Piping the result into another join command, recreate the full original table as shown under "Before Normalisation" above.
5. Notice the columns Name.x and Name.y which appear because there is a Name column in each of the Movies and Customers tables. Rename them.
6. Calculate the number of movies Frida watched of the Sci-fi genre.

Solutions (dplyr joins)

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------- tidyverse

## v ggplot2 3.1.0       v purrr   0.3.1
## v tibble  2.0.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0

## Warning: package 'tibble' was built under R version 3.5.2

## Warning: package 'tidyr' was built under R version 3.5.2

## Warning: package 'purrr' was built under R version 3.5.2

## Warning: package 'dplyr' was built under R version 3.5.2

## Warning: package 'stringr' was built under R version 3.5.2

## Warning: package 'forcats' was built under R version 3.5.2
```

Exercises 1 and 2:

```r
rentals <- read_csv("data/rentals.csv",
                    col_types=cols(Date=col_date(
                      format="%d-%b-%Y")))
movies <- read_csv("data/movies.csv")
```

```
## Parsed with column specification:
## cols(
##   MovieID = col_double(),
##   Name = col_character(),
##   Genre = col_character()
## )
```

```r
customers <- read_csv("data/customers.csv")
```

```
## Parsed with column specification:
## cols(
##   CustomerID = col_double(),
```

Customer name and address for each rental

```
inner_join(rentals, customers, by="CustomerID")
```

```
## # A tibble: 5 x 5
##   Date       MovieID CustomerID Name   Address
##   <date>       <dbl>      <dbl> <chr>  <chr>
## 1 2018-01-01     102          1 Bob    11, Haight St
## 2 2018-01-02     101          2 Frida  Oxford Circus
## 3 2018-01-02     102          3 Carrie 99, Fifth Ave
## 4 2018-01-05     103          1 Bob    11, Haight St
## 5 2018-01-05     104          2 Frida  Oxford Circus
```

Recreate original table

```r
inner_join(rentals, customers, by="CustomerID") %>%
  inner_join(movies, by="MovieID")
```

```
## # A tibble: 5 x 7
##   Date       MovieID CustomerID Name.x Address        Name.y
##   <date>       <dbl>      <dbl> <chr>  <chr>          <chr>
## 1 2018-01-01     102          1 Bob    11, Haight St  Amelie
## 2 2018-01-02     101          2 Frida  Oxford Circus  The Ma
## 3 2018-01-02     102          3 Carrie 99, Fifth Ave  Amelie
## 4 2018-01-05     103          1 Bob    11, Haight St  Skyfal
## 5 2018-01-05     104          2 Frida  Oxford Circus  Avenge
```

Rename columns

```
t = inner_join(rentals, customers, by="CustomerID") %>%
  inner_join(movies, by="MovieID") %>%
  rename(CustomerName=Name.x, MovieTitle=Name.y)
t
```

```
## # A tibble: 5 x 7
##   Date       MovieID CustomerID CustomerName Address      Mo
##   <date>       <dbl>      <dbl> <chr>        <chr>        <c
## 1 2018-01-01     102          1 Bob          11, Haight~ An
## 2 2018-01-02     101          2 Frida        Oxford Cir~ Th
## 3 2018-01-02     102          3 Carrie       99, Fifth ~ An
## 4 2018-01-05     103          1 Bob          11, Haight~ Sk
## 5 2018-01-05     104          2 Frida        Oxford Cir~ Av
```

Filter and count

```
t %>% filter(CustomerName=="Frida", Genre=="Sci-fi") %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     2
```

Filter and count

The following is a solution to the problem, but it requires the programmer to do all the work in their head. That's not scalable or flexible and it's error-prone, so don't do this.

```r
# Frida is CustomerID 2
# Movies 101 and 104 are Sci-fi
rentals %>% filter(CustomerID == 2,
                   MovieID %in% c(101, 104)) %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     2
```

Exercises

1. In the `mpg` dataset (part of the tidyverse), calculate the mean and standard deviation of the highway fuel efficiency.
2. Using `group_by`, calculate the mean and standard deviation of the highway fuel efficiency per manufacturer.
3. Calculate the correlation between highway fuel efficiency and engine size.
4. What was the average highway fuel efficiency in 1999 and in 2008?
5. Carry out a two-sample independent t-test between highway fuel efficiency in 1999 and 2008 and interpret the result.
6. Carry out a regression on highway fuel efficency by displacement.

Solution 1

library(tidyverse)

```
mean(mpg$hwy)
```

```
## [1] 23.44017
```

```
sd(mpg$hwy)
```

```
## [1] 5.954643
```

Solution 2

```r
mpg %>% group_by(manufacturer) %>%
  summarise(mean=mean(hwy), sd=sd(hwy))
```

```
## # A tibble: 15 x 3
##    manufacturer  mean    sd
##    <chr>        <dbl> <dbl>
##  1 audi          26.4  2.18
##  2 chevrolet     21.9  5.11
##  3 dodge         17.9  3.57
##  4 ford          19.4  3.33
##  5 honda         32.6  2.55
##  6 hyundai       26.9  2.18
##  7 jeep          17.6  3.25
##  8 land rover    16.5  1.73
##  9 lincoln       17     1
## 10 mercury       18     1.15
## 11 nissan        24.6  5.09
```

Solution 3

```
cor(mpg$hwy, mpg$displ)
```

```
## [1] -0.76602
```

Solution 4

```
mpg %>% group_by(year) %>%
  summarise(mean=mean(hwy), sd=sd(hwy))
```

```
## # A tibble: 2 x 3
##    year  mean    sd
##   <int> <dbl> <dbl>
## 1  1999  23.4  6.08
## 2  2008  23.5  5.85
```

Solution 5

```r
mpg1999 <- mpg %>% filter(year == 1999)
mpg2008 <- mpg %>% filter(year == 2008)
t.test(mpg1999$hwy, mpg2008$hwy)
```

```
##
##  Welch Two Sample t-test
##
## data:  mpg1999$hwy and mpg2008$hwy
## t = -0.032864, df = 231.64, p-value = 0.9738
## alternative hypothesis: true difference in means is not equ
## 95 percent confidence interval:
##  -1.562854  1.511572
## sample estimates:
## mean of x mean of y
##  23.42735  23.45299
```

Solution 6

```
res = lm(hwy ~ displ, data=mpg)
summary(res)
```

```
##
## Call:
## lm(formula = hwy ~ displ, data = mpg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1039 -2.1646 -0.2242  2.0589 15.0105
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.6977     0.7204   49.55   <2e-16 ***
## displ        -3.5306     0.1945  -18.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```