Inner_join

Ni

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Joining Tables

The inner_join

```
setwd("~/Dropbox/Coursera/RStudio/Data/LEGOs")
sets<-read.csv("sets.csv")
glimpse(sets)
```

```
## Rows: 15,425
## Columns: 5
## $ set_num   <chr> "001-1", "0011-2", "0011-3", "0012-1", "0013-1", "0014-1"...
## $ name      <chr> "Gears", "Town Mini-Figures", "Castle 2 for 1 Bonus Offer...
## $ year      <int> 1965, 1978, 1987, 1979, 1979, 1979, 1979, 1978, 1965, 196...
## $ theme_id  <int> 1, 84, 199, 143, 143, 143, 143, 186, 1, 366, 366, 366, 67...
## $ num_parts <int> 43, 12, 0, 12, 12, 12, 18, 15, 3, 403, 35, 0, 0, 57, 77, ...
```

The *theme_id* variable in the sets table links to the *id* variable in the themes table. To see the theme that each set is associated with, we will need to join the two tables. We used the inner join()

```
setwd("~/Dropbox/Coursera/RStudio/Data/LEGOs")
themes<-read.csv("themes.csv")
glimpse(themes)
```

```
## Rows: 658
## Columns: 3
## $ id        <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17...
## $ name      <chr> "Technic", "Arctic Technic", "Competition", "Expert Build...
## $ parent_id <int> NA, 1, 1, 1, 1, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 1, 1, 1, 1,...
```

```r
inJoinSetsAndThemes1<-sets %>%
      # Linking theme_id in the first table to id in the second table
      inner_join(themes,by = c("theme_id"="id"))
head(inJoinSetsAndThemes1)
```

```
##   set_num                     name.x year theme_id num_parts       name.y
## 1   001-1                      Gears 1965        1        43      Technic
## 2  0011-2          Town Mini-Figures 1978       84        12 Supplemental
## 3  0011-3 Castle 2 for 1 Bonus Offer 1987      199         0 Lion Knights
## 4  0012-1         Space Mini-Figures 1979      143        12 Supplemental
## 5  0013-1         Space Mini-Figures 1979      143        12 Supplemental
## 6  0014-1         Space Mini-Figures 1979      143        12 Supplemental
##   parent_id
## 1        NA
## 2        67
## 3       186
## 4       126
## 5       126
## 6       126
```

We get the output of joining two tables, combining each set with its theme. But because both tables have variable *name*, we have *name.x* and *name.y* You can not have two variable with the same name.

```r
inJoinSetsAndThemes2<-sets %>%
      # Customizing the join by adding suffix
      inner_join(themes,by = c("theme_id"="id"),suffix=c("_sets","_themes"))
head(inJoinSetsAndThemes2)
```

```
##   set_num                  name_sets year theme_id num_parts  name_themes
## 1   001-1                      Gears 1965        1        43      Technic
## 2  0011-2          Town Mini-Figures 1978       84        12 Supplemental
## 3  0011-3 Castle 2 for 1 Bonus Offer 1987      199         0 Lion Knights
## 4  0012-1         Space Mini-Figures 1979      143        12 Supplemental
## 5  0013-1         Space Mini-Figures 1979      143        12 Supplemental
## 6  0014-1         Space Mini-Figures 1979      143        12 Supplemental
##   parent_id
## 1        NA
## 2        67
## 3       186
## 4       126
## 5       126
## 6       126
```

```r
# Finding the most common themes
inJoinSetsAndThemes3<-sets %>%
      inner_join(themes,by = c("theme_id"="id"),suffix=c("_sets","_themes")) %>%
      count(name_themes,sort=TRUE)
head(inJoinSetsAndThemes3)
```

```
##    name_themes   n
## 1    Star Wars 759
## 2         Gear 585
## 3    Basic Set 547
## 4 Supplemental 535
## 5      Technic 452
## 6      Friends 378
```

Joining with a one-to-many relationship

```r
setwd("~/Dropbox/Coursera/RStudio/Data/LEGOs")
parts<-read.csv("parts.csv")
glimpse(parts)
```

```
## Rows: 35,995
## Columns: 4
## $ part_num         <chr> "004229", "004284", "004285", "004490", "004590", ...
## $ name             <chr> "Sticker Sheet for Set 295-1", "Sticker Sheet for ...
## $ part_cat_id      <int> 58, 58, 58, 58, 58, 58, 58, 58, 58, 17, 1, 1, 1, 1...
## $ part_material_id <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
```

```r
setwd("~/Dropbox/Coursera/RStudio/Data/LEGOs")
part_categories<-read.csv("part_categories.csv")
glimpse(part_categories)
```

```
## Rows: 65
## Columns: 2
## $ id   <int> 1, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20...
## $ name <chr> "Baseplates", "Bricks Sloped", "Duplo, Quatro and Primo", "Bri...
```

```r
inJoinParts<-parts %>%
  inner_join(part_categories,by=c("part_cat_id"="id"), suffix=c("_part","_category"))
head(inJoinParts)
```

```
##   part_num                      name_part part_cat_id part_material_id
## 1   004229    Sticker Sheet for Set 295-1          58                1
## 2   004284    Sticker Sheet for Set 723-2          58                1
## 3   004285    Sticker Sheet for Set 725-2          58                1
## 4   004490    Sticker Sheet for Set 365-1          58                1
## 5   004590    Sticker Sheet for Set 182-1          58                1
## 6   004591 Sticker Sheet 1 for Set 1650-1          58                1
##   name_category
## 1      Stickers
## 2      Stickers
## 3      Stickers
## 4      Stickers
## 5      Stickers
## 6      Stickers
```

```r
setwd("~/Dropbox/Coursera/RStudio/Data/LEGOs")
inventories<-read.csv("inventories.csv")
glimpse(inventories)
```

```
## Rows: 25,766
## Columns: 3
## $ id      <int> 1, 3, 4, 15, 16, 17, 19, 21, 22, 25, 26, 27, 28, 30, 31, 33...
## $ version <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ set_num <chr> "7922-1", "3931-1", "6942-1", "5158-1", "903-1", "850950-1"...
```

An inventory represents a product that's made up of some combination of parts. The variable set_num link to the set_num in sets table.

```
setsAndInventory<-sets%>%
  inner_join(inventories,by="set_num")
dim(setsAndInventory)
```

```
## [1] 15826     7
```

```
dim(inventories)
```

```
## [1] 25766     3
```

```
dim(sets)
```

```
## [1] 15425     5
```

```
setsAndInventory_filter<-sets %>%
  inner_join(inventories,by="set_num") %>%
  filter(version==1)
dim(setsAndInventory_filter)
```

```
## [1] 15422     7
```

Notice that after filtering, there are 4976 observations, compared to 4977 in sets table, meaning there is one set that doesn't have version 1, which is probabily a data issue. An inner join keeps an observation only if it has an exact match between the first and the second table.

```
setwd("~/Dropbox/Coursera/RStudio/Data/LEGOs")
inventory_parts<-read.csv("inventory_parts.csv")
glimpse(inventory_parts)
```

```
## Rows: 825,203
## Columns: 5
## $ inventory_id <int> 1, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ...
## $ part_num     <chr> "48379c01", "48395", "mcsport6", "paddle", "2343", "30...
## $ color_id     <int> 72, 7, 25, 0, 47, 29, 2, 15, 15, 15, 29, 15, 15, 29, 2...
## $ quantity     <int> 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 4, 1, 1, 1, 5, 2, 1, 3, ...
## $ is_spare     <chr> "f", "f", "f", "f", "f", "f", "f", "f", "f", "f", "f",...
```

The inventory_parts table combines a part and a colord. The combination describes a single legal piece

```r
partsAndInvpart<-parts %>%
  inner_join(inventory_parts,by="part_num")
```

```r
setwd("~/Dropbox/Coursera/RStudio/Data/LEGOs")
colors<-read.csv("colors.csv")
glimpse(colors)
```

```
## Rows: 184
## Columns: 4
## $ id       <int> -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ...
## $ name     <chr> "[Unknown]", "Black", "Blue", "Green", "Dark Turquoise", "...
## $ rgb      <chr> "0033B2", "05131D", "0055BF", "237841", "008F9B", "C91A09"...
## $ is_trans <chr> "f", "f", "f", "f", "f", "f", "f", "f", "f", "f", "f", "f"...
```

Joining three or more tables

```r
sets_inv_themes<-sets %>%
  inner_join(inventories,by="set_num")  %>%
  inner_join(themes,by=c("theme_id" = "id"),suffix=c("_set","_theme"))
head(sets_inv_themes)
```

```
##   set_num                   name_set year theme_id num_parts    id version
## 1   001-1                      Gears 1965        1        43 24696       1
## 2  0011-2          Town Mini-Figures 1978       84        12  5087       1
## 3  0011-3 Castle 2 for 1 Bonus Offer 1987      199         0  2216       1
## 4  0012-1         Space Mini-Figures 1979      143        12  1414       1
## 5  0013-1         Space Mini-Figures 1979      143        12  4609       1
## 6  0014-1         Space Mini-Figures 1979      143        12  5004       1
##     name_theme parent_id
## 1      Technic        NA
## 2 Supplemental        67
## 3 Lion Knights       186
## 4 Supplemental       126
## 5 Supplemental       126
## 6 Supplemental       126
```

```r
sets_inv_invParts<-sets %>%
    inner_join(inventories,by="set_num") %>%
  inner_join(inventory_parts,by=c("id"="inventory_id")) %>%
  inner_join(colors,by=c("color_id"="id"),suffix=c("_set","_color")) %>%
  # count the name_color column
  count(name_color,sort=TRUE)
head(sets_inv_invParts)
```

```
##          name_color      n
## 1             Black 145403
## 2             White  88369
## 3 Light Bluish Gray  83618
## 4               Red  64787
## 5  Dark Bluish Gray  63221
## 6            Yellow  45084
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

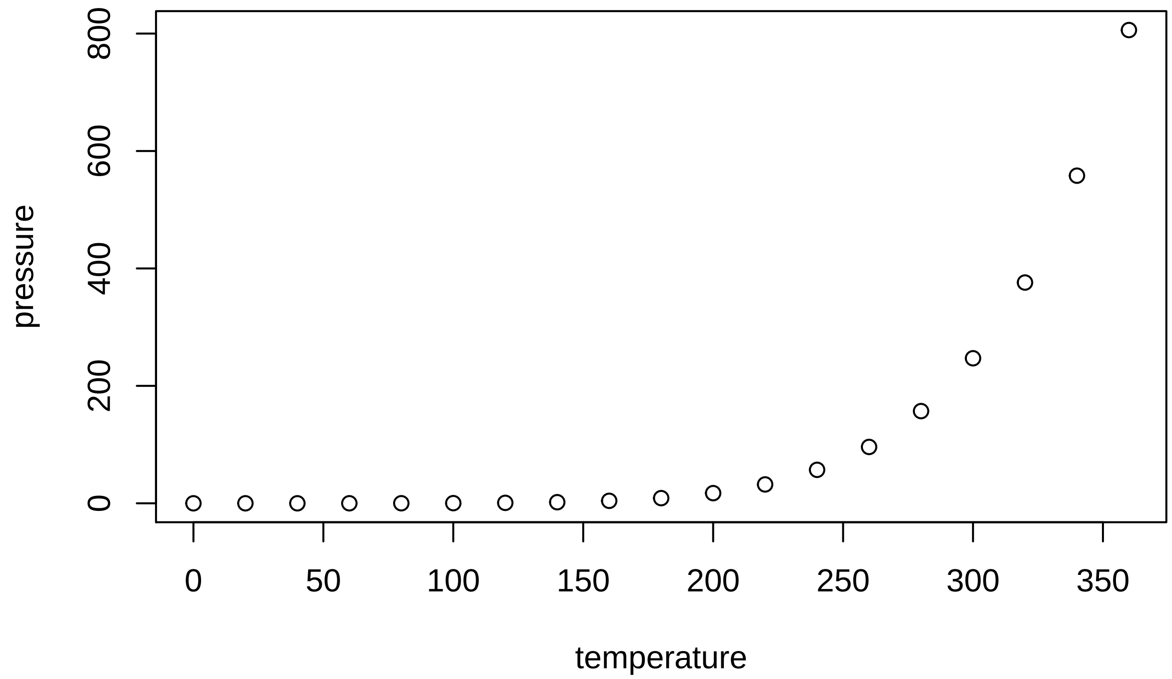
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.