Tidyverse - Dplyr

Joshua Hummell

4/8/2021

Contents

Tidyverse Vignette

```
library(dplyr)
```

Hands down my favorite R package in Tidyverse is Dplyr Dplyr allows for easy data manipulation and, therefore, is highly useful for everyday work!

```
murders <- read.csv('https://raw.githubusercontent.com/fivethirtyeight/data/master/murder_2016/murder_2
```

Select data columns with ease

```
murders %>% select(state)
```

```
##             state
## 1        Maryland
## 2        Illinois
## 3           Texas
## 4            Ohio
## 5            D.C.
## 6       Wisconsin
## 7    Pennsylvania
## 8        Missouri
## 9       Tennessee
## 10       Missouri
## 11       Oklahoma
## 12       Kentucky
## 13       Colorado
## 14     California
## 15          Texas
## 16       New York
## 17        Florida
## 18      Minnesota
## 19       Nebraska
## 20     California
## 21         Alaska
```

```
## 22 North Carolina
## 23      Louisiana
## 24     New Mexico
## 25       Colorado
## 26        Indiana
## 27     California
## 28 North Carolina
## 29        Indiana
## 30     New Jersey
## 31       Oklahoma
## 32         Oregon
## 33     California
## 34           Ohio
## 35        Florida
## 36     California
## 37       Colorado
## 38         Nevada
## 39     California
## 40     California
## 41      Minnesota
## 42     California
## 43 North Carolina
## 44     New Jersey
## 45        Arizona
## 46          Texas
## 47       Virginia
## 48     California
## 49        Georgia
## 50         Nevada
## 51        Florida
## 52 North Carolina
## 53         Kansas
## 54        Arizona
## 55          Texas
## 56     California
## 57           Ohio
## 58     California
## 59        Arizona
## 60     California
## 61     California
## 62       Michigan
## 63     Washington
## 64          Texas
## 65        Arizona
## 66          Texas
## 67       Kentucky
## 68      Tennessee
## 69        Florida
## 70           Ohio
## 71         Hawaii
## 72          Texas
## 73       Nebraska
## 74        Florida
## 75     California
```

```
## 76        Alabama
## 77     California
## 78          Texas
## 79          Texas
## 80          Texas
## 81   Pennsylvania
## 82  Massachusetts
## 83       New York
```

easily filter data

```r
murders %>%
  filter(city == 'Baltimore')
```

```
##        city    state X2014_murders X2015_murders change
## 1 Baltimore Maryland           211           344    133
```

Easily Aggregate Date

```r
state <- murders %>%
  select(state, change) %>%
  group_by(state) %>%
  summarise(state_totals = sum(change)) %>%
  arrange(desc(state_totals))
state
```

```
## # A tibble: 34 x 2
##    state      state_totals
##    <chr>             <int>
##  1 Maryland            133
##  2 Illinois             67
##  3 California           60
##  4 Missouri             60
##  5 D.C.                 57
##  6 Ohio                 57
##  7 Wisconsin            55
##  8 Colorado             40
##  9 Texas                40
## 10 Oklahoma             37
## # ... with 24 more rows
```

and even join data

```r
states_pop <- read.csv('https://raw.githubusercontent.com/jhumms/DATA607/main/state_populations.csv')
colnames(states_pop) <- tolower(colnames(states_pop))

murders_state <- left_join(state, states_pop, by='state')

murders_state
```

```
## # A tibble: 34 x 5
```

```
##    state      state_totals  rank population percent.of.total
##    <chr>             <int> <int>      <dbl> <chr>
##  1 Maryland            133    19    6045680 1.82%
##  2 Illinois             67     5   12671821 3.86%
##  3 California           60     1   39512223 11.91%
##  4 Missouri             60    18    6137428 1.85%
##  5 D.C.                 57    NA         NA <NA>
##  6 Ohio                 57     7   11689100 3.52%
##  7 Wisconsin            55    20    5822434 1.75%
##  8 Colorado             40    21    5758736 1.74%
##  9 Texas                40     2   28995881 8.74%
## 10 Oklahoma             37    28    3956971 1.19%
## # ... with 24 more rows
```

and, if that weren't enough, you can even make aggregations across columns very easily!

```
murders_state$population <- as.numeric(murders_state$population)

murders_state %>%
  mutate(murder_rate_by_pop = (state_totals / population) *100) %>%
  arrange(desc(murder_rate_by_pop))
```

```
## # A tibble: 34 x 6
##    state      state_totals  rank population percent.of.total murder_rate_by_pop
##    <chr>             <int> <int>      <dbl> <chr>                         <dbl>
##  1 Maryland            133    19    6045680 1.82%                       0.00220
##  2 Alaska               14    48     731545 0.22%                       0.00191
##  3 Missouri             60    18    6137428 1.85%                       0.000978
##  4 Wisconsin            55    20    5822434 1.75%                       0.000945
##  5 Oklahoma             37    28    3956971 1.19%                       0.000935
##  6 Colorado             40    21    5758736 1.74%                       0.000695
##  7 New Mexico           13    36    2096829 0.63%                       0.000620
##  8 Illinois             67     5   12671821 3.86%                       0.000529
##  9 Nebraska             10    37    1934408 0.58%                       0.000517
## 10 Ohio                 57     7   11689100 3.52%                       0.000488
## # ... with 24 more rows
```

Tidyverse Extend

Josh's vignette and his dataset caught my attention. It looks like the **murders** dataset contains 83 observations with 5 variables: City, State, 2014 murder count (X2014_murders), 2015 murder count (X2015_murders), and the difference between the two years (change).

Using dplyr's glimpse[1] function, we can take a look at the data as well as their types.

```
glimpse(murders)
```

```
## Rows: 83
## Columns: 5
## $ city          <chr> "Baltimore", "Chicago", "Houston", "Cleveland", "Washing~
## $ state         <chr> "Maryland", "Illinois", "Texas", "Ohio", "D.C.", "Wiscon~
```

[1]https://www.rdocumentation.org/packages/dplyr/versions/0.3/topics/glimpse

```
## $ X2014_murders <int> 211, 411, 242, 63, 105, 90, 248, 78, 41, 159, 45, 56, 31~
## $ X2015_murders <int> 344, 478, 303, 120, 162, 145, 280, 109, 72, 188, 73, 81,~
## $ change        <int> 133, 67, 61, 57, 57, 55, 32, 31, 31, 29, 28, 25, 22, 22,~
```

From Josh's work, we can see there are 34 distinct states in `murder` dataset. We can also see the distinct cities listed using dplyr's distinct [2] function

```
murders %>%
  distinct(city)
```

```
##                      city
## 1               Baltimore
## 2                 Chicago
## 3                 Houston
## 4               Cleveland
## 5              Washington
## 6               Milwaukee
## 7            Philadelphia
## 8             Kansas City
## 9               Nashville
## 10             St. Louis
## 11         Oklahoma City
## 12            Louisville
## 13                Denver
## 14           Los Angeles
## 15                Dallas
## 16              New York
## 17               Orlando
## 18           Minneapolis
## 19                 Omaha
## 20            Sacramento
## 21             Anchorage
## 22 Charlotte-Mecklenburg
## 23           New Orleans
## 24           Albuquerque
## 25                Aurora
## 26            Fort Wayne
## 27            Long Beach
## 28                Durham
## 29          Indianapolis
## 30                Newark
## 31                 Tulsa
## 32              Portland
## 33         San Francisco
## 34            Cincinnati
## 35                 Tampa
## 36           Bakersfield
## 37      Colorado Springs
## 38             Las Vegas
## 39               Oakland
## 40             San Diego
## 41              St. Paul
```

[2] https://dplyr.tidyverse.org/reference/distinct_all.html?q=distinct

```
## 42              Anaheim
## 43           Greensboro
## 44          Jersey City
## 45                 Mesa
## 46           Fort Worth
## 47       Virginia Beach
## 48               Irvine
## 49              Atlanta
## 50            Henderson
## 51         Jacksonville
## 52              Raleigh
## 53              Wichita
## 54             Chandler
## 55                Plano
## 56             Stockton
## 57               Toledo
## 58          Chula Vista
## 59              Phoenix
## 60            Riverside
## 61             San Jose
## 62              Detroit
## 63              Seattle
## 64              El Paso
## 65               Tucson
## 66            Arlington
## 67            Lexington
## 68              Memphis
## 69       St. Petersburg
## 70             Columbus
## 71             Honolulu
## 72               Laredo
## 73              Lincoln
## 74                Miami
## 75            Santa Ana
## 76               Mobile
## 77               Fresno
## 78               Austin
## 79          San Antonio
## 80       Corpus Christi
## 81           Pittsburgh
## 82               Boston
## 83              Buffalo
```

Next, suppose you wanted the top 10 cities with the most murders in the year 2015. In order to extract the top 10 cities, we would use the top_n()[3] function as shown below. Note, if a variable is not specified in the function as we have below with X2015_murders, then the top_n() function will automatically extract the top n specifed by the last column in the dataset.

```
murders %>%
  top_n(10, X2015_murders)
```

```
##             city        state X2014_murders X2015_murders change
```

[3]https://dplyr.tidyverse.org/reference/top_n.html

```
## 1     Baltimore     Maryland            211            344    133
## 2       Chicago     Illinois            411            478     67
## 3       Houston        Texas            242            303     61
## 4    Washington         D.C.            105            162     57
## 5  Philadelphia Pennsylvania            248            280     32
## 6     St. Louis     Missouri            159            188     29
## 7   Los Angeles   California            260            282     22
## 8      New York     New York            333            352     19
## 9   New Orleans    Louisiana            150            164     14
## 10      Detroit     Michigan            298            295     -3
```

Lets say now, instead of the top ten, you actually want the bottom 5 cities with the least murders from the year 2014. We can still use the top_n() function, the only difference will be is that we will add a minus (-) sign to the input.

```
murders %>%
  top_n(-5, X2014_murders)
```

```
##          city      state X2014_murders X2015_murders change
## 1      Irvine California             0             2      2
## 2   Henderson     Nevada             3             4      1
## 3    Chandler    Arizona             1             1      0
## 4       Plano      Texas             4             4      0
## 5 Chula Vista California             7             6     -1
## 6     Lincoln   Nebraska             7             1     -6
```

Which cities had the greatest percent change in murder counts? The dataset already comes with a chnage column that give us the murders 2015 - murders 2014 value, however we would like to see this in a percentage. We can perform such calculation and add it as a new column using the mutate()[4] function.
Note, the round()[5] from baseR is used to round our percent_change value to the 2 decimal places by wrapping our percent change function and specifing the two decimal places.

```
murders %>%
  mutate(percent_change = round(((X2015_murders-X2014_murders)/X2014_murders)*100,2))
```

```
##                    city          state X2014_murders X2015_murders change
## 1             Baltimore       Maryland           211           344    133
## 2               Chicago       Illinois           411           478     67
## 3               Houston          Texas           242           303     61
## 4             Cleveland           Ohio            63           120     57
## 5            Washington           D.C.           105           162     57
## 6             Milwaukee      Wisconsin            90           145     55
## 7          Philadelphia   Pennsylvania           248           280     32
## 8           Kansas City       Missouri            78           109     31
## 9             Nashville      Tennessee            41            72     31
## 10            St. Louis       Missouri           159           188     29
## 11        Oklahoma City       Oklahoma            45            73     28
## 12           Louisville       Kentucky            56            81     25
## 13               Denver       Colorado            31            53     22
## 14          Los Angeles     California           260           282     22
```

[4]https://www.rdocumentation.org/packages/plyr/versions/1.8.6/topics/mutate
[5]https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/Round

```
## 15                   Dallas          Texas           116             136      20
## 16                 New York       New York           333             352      19
## 17                  Orlando        Florida            15              32      17
## 18              Minneapolis      Minnesota            31              47      16
## 19                    Omaha       Nebraska            32              48      16
## 20               Sacramento     California            28              43      15
## 21                Anchorage         Alaska            12              26      14
## 22 Charlotte-Mecklenburg North Carolina            47              61      14
## 23              New Orleans      Louisiana           150             164      14
## 24              Albuquerque     New Mexico            30              43      13
## 25                   Aurora       Colorado            11              24      13
## 26               Fort Wayne        Indiana            12              25      13
## 27               Long Beach     California            23              36      13
## 28                   Durham North Carolina            21              34      13
## 29             Indianapolis        Indiana           136             148      12
## 30                   Newark     New Jersey            93             104      11
## 31                    Tulsa       Oklahoma            46              55       9
## 32                 Portland         Oregon            26              34       8
## 33            San Francisco     California            45              53       8
## 34               Cincinnati           Ohio            60              66       6
## 35                    Tampa        Florida            28              34       6
## 36              Bakersfield     California            17              22       5
## 37         Colorado Springs       Colorado            20              25       5
## 38                Las Vegas         Nevada           122             127       5
## 39                  Oakland     California            80              85       5
## 40                San Diego     California            32              37       5
## 41                 St. Paul      Minnesota            11              16       5
## 42                  Anaheim     California            14              18       4
## 43               Greensboro North Carolina            23              26       3
## 44              Jersey City     New Jersey            24              27       3
## 45                     Mesa        Arizona            13              16       3
## 46               Fort Worth          Texas            54              56       2
## 47           Virginia Beach       Virginia            17              19       2
## 48                   Irvine     California             0               2       2
## 49                  Atlanta        Georgia            93              94       1
## 50                Henderson         Nevada             3               4       1
## 51             Jacksonville        Florida            96              97       1
## 52                  Raleigh North Carolina            16              17       1
## 53                  Wichita         Kansas            26              27       1
## 54                 Chandler        Arizona             1               1       0
## 55                    Plano          Texas             4               4       0
## 56                 Stockton     California            49              49       0
## 57                   Toledo           Ohio            24              24       0
## 58              Chula Vista     California             7               6      -1
## 59                  Phoenix        Arizona           114             112      -2
## 60                Riverside     California            12              10      -2
## 61                 San Jose     California            32              30      -2
## 62                  Detroit       Michigan           298             295      -3
## 63                  Seattle     Washington            26              23      -3
## 64                  El Paso          Texas            21              17      -4
## 65                   Tucson        Arizona            35              31      -4
## 66                Arlington          Texas            13               8      -5
## 67                Lexington       Kentucky            20              15      -5
## 68                  Memphis      Tennessee           140             135      -5
```

```
## 69       St. Petersburg       Florida            19            14     -5
## 70             Columbus          Ohio            83            77     -6
## 71             Honolulu        Hawaii            21            15     -6
## 72               Laredo         Texas            14             8     -6
## 73              Lincoln      Nebraska             7             1     -6
## 74                Miami       Florida            81            75     -6
## 75            Santa Ana    California            18            12     -6
## 76               Mobile       Alabama            31            24     -7
## 77               Fresno    California            47            39     -8
## 78               Austin         Texas            32            23     -9
## 79          San Antonio         Texas           103            94     -9
## 80       Corpus Christi         Texas            27            17    -10
## 81           Pittsburgh  Pennsylvania            69            57    -12
## 82               Boston Massachusetts            53            38    -15
## 83              Buffalo      New York            60            41    -19
##    percent_change
## 1           63.03
## 2           16.30
## 3           25.21
## 4           90.48
## 5           54.29
## 6           61.11
## 7           12.90
## 8           39.74
## 9           75.61
## 10          18.24
## 11          62.22
## 12          44.64
## 13          70.97
## 14           8.46
## 15          17.24
## 16           5.71
## 17         113.33
## 18          51.61
## 19          50.00
## 20          53.57
## 21         116.67
## 22          29.79
## 23           9.33
## 24          43.33
## 25         118.18
## 26         108.33
## 27          56.52
## 28          61.90
## 29           8.82
## 30          11.83
## 31          19.57
## 32          30.77
## 33          17.78
## 34          10.00
## 35          21.43
## 36          29.41
## 37          25.00
## 38           4.10
```

```
## 39            6.25
## 40           15.62
## 41           45.45
## 42           28.57
## 43           13.04
## 44           12.50
## 45           23.08
## 46            3.70
## 47           11.76
## 48             Inf
## 49            1.08
## 50           33.33
## 51            1.04
## 52            6.25
## 53            3.85
## 54            0.00
## 55            0.00
## 56            0.00
## 57            0.00
## 58          -14.29
## 59           -1.75
## 60          -16.67
## 61           -6.25
## 62           -1.01
## 63          -11.54
## 64          -19.05
## 65          -11.43
## 66          -38.46
## 67          -25.00
## 68           -3.57
## 69          -26.32
## 70           -7.23
## 71          -28.57
## 72          -42.86
## 73          -85.71
## 74           -7.41
## 75          -33.33
## 76          -22.58
## 77          -17.02
## 78          -28.12
## 79           -8.74
## 80          -37.04
## 81          -17.39
## 82          -28.30
## 83          -31.67
```

Using our top_n() function as well as leveraging the pipe operator multiple times, we can see which top 5 cities had the highest percent increases, and arrange them in descending order using the arrange()[6] function.

```
murders %>%
  mutate(percent_change = round(((X2015_murders-X2014_murders)/X2014_murders)*100,2)) %>%
  top_n(5,percent_change) %>%
  arrange(desc(percent_change))
```

[6]https://www.rdocumentation.org/packages/dplyr/versions/0.7.8/topics/arrange

```
##         city      state X2014_murders X2015_murders change percent_change
## 1     Irvine California             0             2      2            Inf
## 2     Aurora   Colorado            11            24     13         118.18
## 3  Anchorage     Alaska            12            26     14         116.67
## 4    Orlando    Florida            15            32     17         113.33
## 5 Fort Wayne    Indiana            12            25     13         108.33
```