A closer look at R package dplyr

Diksha Radhakrishnan

Introduction and Motivation

Manipulating datasets is a skill that we can all agree is incontestably useful. We were first taught how to wrangle with data in R and pull it apart about two weeks into the course. I don't know about you, but when we started using bracket notation to extract relevant information from the NBA datasets, my head was spinning. It was a cumbersome approach in my opinion to reshape and aggregate the data sets, and I realized during the lab assignments that I was definitely guessing more than I would care to admit, copying my answers from earlier and editing them to get to where I wanted, and sometimes finding code difficult to decipher, especially for complicated operations. [1]

It was about a week later when we were introduced to the R package dplyr. It took me by surprise: it was uncomplicated and incredibly intuitive. The functions used to extract relevant information from data sets were clearly designed with efficiency and brevity in mind. This was my motivation for making this post - I wished to find out more about dplyr and its usage, beyond what we learned in lab and lecture.

Background

Dplyr is a grammar of data manipulation created by Hadley Wickham, providing a consistent set of verbs that help you solve the most common data manipulation challenges. [2] The most common functions or verbs that are used are:
- **select()**: picks variables or columns based on their names
- **filter()**: picks cases or rows based on a certain criteria / their value
- **arrange()**: reorders the rows
- **mutate()**: adds new variables or columns that are functions of existing ones
- **slice()**: selects/subsets rows
- **summarise()**: reduces multiple values of a variable down to a single summary value
- **group_by()**: enables us to perform any operation by grouping by one or more variable. [3]

In addition to these functions which we worked with in class, I was interested to find out more about [4]:
- sample_n() and sample_frac(): randomly select a random number or random fraction of rows, respectively
- count(): tallies observations based on a group or a variable.

The pipe operator in R, represented by "%>%", which is used to chain code together, proving to be useful when multiple operations are carried out at once on a dataset.
We will explore the proper usage and functioning of these verbs in the examples.

Examples and Visualization

For the purposes of this post, and because we've maybe had our fair share of NBA datasets thus far in the course, the dataset that I will be using comes from a paper [5] by three researchers, Dezhbakhsh, Rubin, and Shepherd. The dataset contains rates of various violent crimes for every year 1960-2003 (44 years) in every US state. The researchers compiled the data from the FBI's Uniform Crime Reports.

Demo

Loading the dplyr package, as well as ggplot2 for potential graphs:

```
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
```

```
library(ggplot2)
```

The following table gives us crime rates for all 50 states of the U.S. from 1960 to 2003.

```
crimes = read.csv('crime_rates.csv', stringsAsFactors = FALSE)
slice(crimes, 1:200) #show the first 200 rows
```

```
## # A tibble: 200 x 12
##     State  Year Population Violent.Crime.Rate Murder.Rate
##     <chr> <int>      <int>              <dbl>       <dbl>
##  1 Alaska  1960     226167              104.3        10.2
##  2 Alaska  1961     234000               88.9        11.5
##  3 Alaska  1962     246000               91.5         4.5
##  4 Alaska  1963     248000              109.7         6.5
##  5 Alaska  1964     250000              150.0        10.4
##  6 Alaska  1965     253000              149.0         6.3
##  7 Alaska  1966     272000              150.4        12.9
##  8 Alaska  1967     272000              160.7         9.6
##  9 Alaska  1968     277000              175.5        10.5
## 10 Alaska  1969     282000              221.3        10.6
## # ... with 190 more rows, and 7 more variables: Forcible.Rape.Rate <dbl>,
## #   Robbery.Rate <dbl>, Aggraveted.Assault.Rate <dbl>,
## #   Property.Crime.Rate <dbl>, Burglary.Rate <dbl>,
## #   Larceny...Theft.Rate <dbl>, Motor.Vehicle.Theft.Rate <dbl>
```

What if we only cared about murder rates?

```
murder = select(crimes, State, Year, Population, Murder.Rate)
slice(murder, 1:200) #show the first 200 rows
```

```
## # A tibble: 200 x 4
##     State  Year Population Murder.Rate
##     <chr> <int>      <int>       <dbl>
##  1 Alaska  1960     226167        10.2
##  2 Alaska  1961     234000        11.5
##  3 Alaska  1962     246000         4.5
##  4 Alaska  1963     248000         6.5
##  5 Alaska  1964     250000        10.4
##  6 Alaska  1965     253000         6.3
##  7 Alaska  1966     272000        12.9
##  8 Alaska  1967     272000         9.6
##  9 Alaska  1968     277000        10.5
## 10 Alaska  1969     282000        10.6
## # ... with 190 more rows
```

What if we wanted to know the exact number of murders in a state in a particular year?
Note:
Murder Rates are calculated as follows:

$$\frac{NumberOfMurdersInStateXinYearY}{PopulationInStateXinYearY} * 100000$$

(Murder is rare, so we multiply by 100,000 to avoid handling small numbers).

```
murder = mutate(murder, TotalMurders = (Population * Murder.Rate)/100000)
slice(murder, 1:200) #show the first 200 rows
```

```
## # A tibble: 200 x 5
##     State  Year Population Murder.Rate TotalMurders
##     <chr> <int>      <int>       <dbl>        <dbl>
##  1 Alaska  1960     226167        10.2     23.06903
##  2 Alaska  1961     234000        11.5     26.91000
##  3 Alaska  1962     246000         4.5     11.07000
##  4 Alaska  1963     248000         6.5     16.12000
##  5 Alaska  1964     250000        10.4     26.00000
##  6 Alaska  1965     253000         6.3     15.93900
##  7 Alaska  1966     272000        12.9     35.08800
##  8 Alaska  1967     272000         9.6     26.11200
##  9 Alaska  1968     277000        10.5     29.08500
## 10 Alaska  1969     282000        10.6     29.89200
## # ... with 190 more rows
```

What if we wanted to know the top five states who had the highest murder rate in 2000?

```
slice(arrange(filter(murder, Year == 2000), desc(Murder.Rate)), 1:5)
```

```
## # A tibble: 5 x 5
##         State  Year Population Murder.Rate TotalMurders
##         <chr> <int>      <int>       <dbl>        <dbl>
## 1   Louisiana  2000    4468976        12.5     558.6220
## 2 Mississippi  2000    2844658         9.0     256.0192
## 3    Maryland  2000    5296486         8.1     429.0154
## 4     Georgia  2000    8186453         8.0     654.9162
## 5     Alabama  2000    4447100         7.4     329.0854
```

We could do the identical operation above using the pipe operator:

```
murder  %>%
  filter(Year == 2000)  %>%
  arrange(desc(Murder.Rate))  %>%
  slice(1:5)
```

```
## # A tibble: 5 x 5
##          State  Year Population Murder.Rate TotalMurders
##          <chr> <int>      <int>       <dbl>        <dbl>
## 1   Louisiana  2000    4468976        12.5     558.6220
## 2 Mississippi  2000    2844658         9.0     256.0192
## 3    Maryland  2000    5296486         8.1     429.0154
## 4     Georgia  2000    8186453         8.0     654.9162
## 5     Alabama  2000    4447100         7.4     329.0854
```

What if we wanted to know what the largest population across the states in 1960?

```
summarise(filter(murder, Year == 1960), max_pop = max(Population))
```

```
##    max_pop
## 1 16838000
```

What if we wanted to know which State this was?

```
select(filter(murder, Year == 1960 & Population == 16838000), State)
```

```
##      State
## 1 New York
```

Now let's use some grouping operations, by Year, to consolidate this information to find out the average murder rate across the US for each year!

```
summarise(group_by(murder, Year), avg_murder_rate = mean(Murder.Rate))
```

```
## # A tibble: 44 x 2
##     Year avg_murder_rate
##    <int>           <dbl>
##  1  1960           5.072
##  2  1961           4.646
##  3  1962           4.342
##  4  1963           4.390
##  5  1964           4.554
##  6  1965           4.630
##  7  1966           5.322
##  8  1967           5.618
##  9  1968           6.086
## 10  1969           6.290
## # ... with 34 more rows
```

Let's compare some of the crime rates in Louisiana - a state with one of the consistently highest crime rates in the United States - over time.

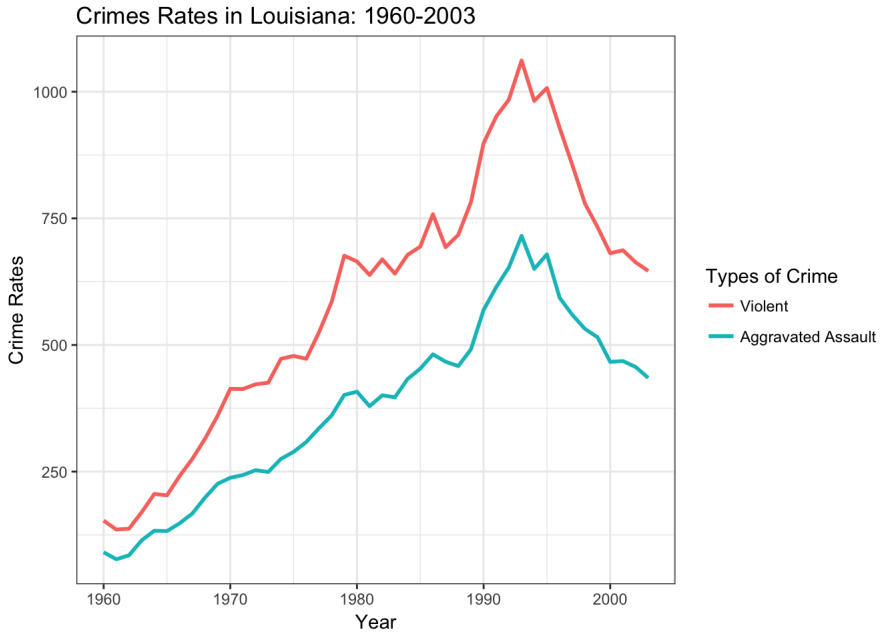
```
louisiana = select(filter(crimes, State == "Louisiana"), Year, Violent.Crime.Rate, Aggraveted.Assault.Rate, Murder
.Rate, Forcible.Rape.Rate )
louisiana
```

```
##    Year Violent.Crime.Rate Aggraveted.Assault.Rate Murder.Rate
## 1  1960              153.2                   90.8         8.3
## 2  1961              135.7                   76.9         6.4
## 3  1962              137.2                   84.7         6.8
## 4  1963              169.6                  114.2         6.9
## 5  1964              205.9                  133.2         8.3
## 6  1965              203.1                  132.6         8.1
## 7  1966              241.2                  147.9         9.9
## 8  1967              275.0                  167.2         9.3
## 9  1968              314.6                  198.4         9.5
## 10 1969              360.3                  226.0         9.5
## 11 1970              413.5                  237.8        11.7
## 12 1971              413.1                  243.2        11.1
## 13 1972              422.4                  252.8        13.2
## 14 1973              425.6                  249.3        15.4
## 15 1974              472.7                  275.1        16.0
## 16 1975              478.4                  289.0        12.6
## 17 1976              472.8                  308.5        13.2
## 18 1977              524.8                  335.5        15.5
## 19 1978              584.9                  360.7        15.8
## 20 1979              676.3                  401.4        16.9
## 21 1980              665.0                  407.8        15.7
## 22 1981              638.3                  379.2        15.6
## 23 1982              669.1                  400.6        16.0
## 24 1983              640.9                  396.8        14.2
## 25 1984              678.0                  432.8        12.9
## 26 1985              694.2                  453.3        10.9
```

```
## 27 1986              758.2                  481.6        12.8
## 28 1987              693.0                  467.0        11.1
## 29 1988              717.4                  458.3        11.6
## 30 1989              781.8                  491.4        14.9
## 31 1990              898.4                  569.2        17.2
## 32 1991              951.0                  614.3        16.9
## 33 1992              984.6                  653.4        17.4
## 34 1993             1061.7                  715.4        20.3
## 35 1994              981.9                  650.3        19.8
## 36 1995             1007.4                  679.0        17.0
## 37 1996              929.1                  593.5        17.5
## 38 1997              855.9                  559.7        15.7
## 39 1998              779.5                  531.9        12.8
## 40 1999              732.7                  515.2        10.7
## 41 2000              681.1                  466.6        12.5
## 42 2001              687.0                  468.3        11.2
## 43 2002              663.3                  456.7        13.2
## 44 2003              646.3                  435.0        13.0
##    Forcible.Rape.Rate
## 1                 8.6
## 2                 8.0
## 3                 6.8
## 4                 6.2
## 5                11.1
## 6                11.1
## 7                16.6
## 8                16.5
## 9                16.4
## 10               22.1
## 11               23.1
## 12               23.7
## 13               23.0
## 14               22.2
## 15               25.2
## 16               23.7
## 17               26.8
## 18               30.9
## 19               34.8
## 20               38.6
## 21               44.5
## 22               41.4
## 23               39.9
## 24               39.9
## 25               41.8
## 26               39.8
## 27               40.1
## 28               35.9
## 29               38.5
## 30               38.2
## 31               42.2
## 32               40.9
## 33               42.3
## 34               42.3
## 35               44.6
## 36               42.7
## 37               41.5
## 38               41.3
## 39               36.8
## 40               33.1
## 41               33.5
## 42               31.4
## 43               34.2
## 44               41.1
```

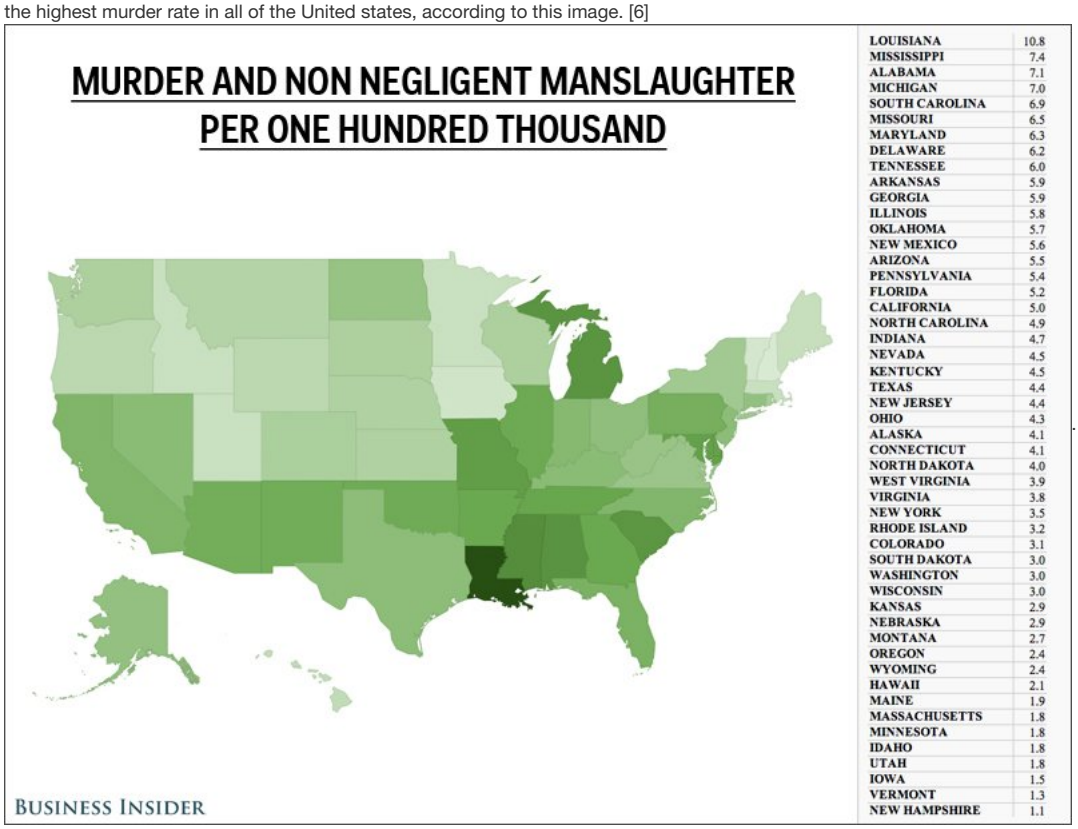
We now want to visualise (using ggplot2) how Violent Crime and Aggravated Assault rates varied in Louisiana over the years:

```
ggplot() + geom_line(data=louisiana, aes(x = Year, y = Violent.Crime.Rate, colour = "darkblue"), size = 1) +geom_l
ine(data=louisiana, aes(x = Year, y = Aggraveted.Assault.Rate, colour = "red"), size =1) + scale_color_discrete(na
me = "Types of Crime", labels = c("Violent", "Aggravated Assault")) + ylab("Crime Rates") + theme_bw() + ggtitle("
Crimes Rates in Louisiana: 1960-2003")
```



Discussion

Note that the violent crime rates and aggravated assault rates in Louisiana move in tandem - which suggests that on the whole, all crime rates per state move roughly in step with one another. We can see that in Louisiana, crime rates had more or less been steadily and rapidly on the rise from 1960 onwards, and reached a peak in roughly 1992, before declining rather quickly again. We know that Louisiana has had historically high crime rates, and a rather troubling past. In fact, we know that certain cities like Baton Rouge and New Orleans have astronomically high crime rates, and much research is being done with regards to curbing these, and promoting institutional reform. In fact, as of 2012, Louisiana still had the highest murder rate in all of the United states, according to this image. [6]



If you are more interested in the reasons behind the high crime/murder rates in Louisiana, I found this article [7] intriguing. Overall, using the dplyr package, we were able able to make some interesting abstractions from a few simple operations - and this is only one of the many ways in which dplyr is useful.

Conclusion

I think that the dplyr package certainly helps us carry out rigorous data analysis, incredibly easily. The brief examples we just walked through were uncomplicated and with the help of the ggplot2 package, we were able to draw some high-level conclusions using the data. Despite being a fairly new package [8] - it was released in 2014, and based off of an older package plyr - dplyr has revolutionalized the manner in which we deal with data using R. With these ease of data analysis, it is no doubt that in the forthcoming years, we will be far more efficient in our manipulation of datasets, with the current version of dplyr and further iterations of the grammar. I know I have loved learning more about this package, and I hope this post has been equally informative for those of you reading this.

References

1. Karl Broman's page on dplyr

2. The tidyverse page on dplyr
3. Stat 133 Github - Lab 05 on dplyr/ggplot2 basics
4. The R-bloggers' blog page on dplyr
5. Dezhbakhsh, Hashem, Paul H. Rubin, and Joanna M. Shepherd. "Does capital punishment have a deterrent effect? New evidence from postmoratorium panel data." American Law and Economics Review 5, no. 2 (2003): 344-376.
6. Image depicting the murder rates in the United States according to the FBI Uniform Crime Report, 2012
7. Business Insider: Why Louisiana is the Murder Capital of America
8. [illegible] on dplyr