Setting the stage for data science: integration of data management skills in introductory and second courses in statistics (nycflights13)

Nicholas J. Horton, Benjamin S. Baumer, and Hadley Wickham

February 22, 2015

Many have argued that statistics students need additional facility to express statistical computations. By introducing students to commonplace tools for data management, visualization, and reproducible analysis in data science and applying these to real-world scenarios, we prepare them to think statistically. In an era of increasingly big data, it is imperative that students develop data-related capacities, beginning with the introductory course. We believe that the integration of these precursors to data science into our curricula—early and often—will help statisticians be part of the dialogue regarding *Big Data and Big Questions*.

Specifically, through our shared experience working in industry, government, private consulting, and academia we have identified five key elements which deserve greater emphasis in the undergraduate curriculum (in no particular order):

1. Thinking creatively, but constructively, about data. This "data tidying" includes the ability to move data not only between different file formats, but also different *shapes*. There are elements of data storage design (e.g. normal forms) and foresight into how data should arranged based on how it will likely be used.
2. Facility with data sets of varying sizes and some understanding of scalability issues when working with data. This includes an elementary understanding of basic computer architecture (e.g. memory vs. hard disk space), and the ability to query a relational database management system (RDBMS).
3. Statistical computing skills in a command-driven environment (e.g. R, Python, or Julia). Coding skills (in any language) are highly-valued and increasingly necessary. They provide freedom from the un-reproducible point-and-click application paradigm.
4. Experience wrestling with large, messy, complex, challenging data sets, for which there is no obvious goal or specially-curated statistical method (see SIDEBAR: What's in a name). While perhaps suboptimal for teaching specific statistical methods, these data are more similar to what analysts actually see in the wild.
5. An ethos of reproducibility. This is a major challenge for science in general, and we have the comparatively easy task of simply reproducing computations and analysis.

We illustrate how these five elements can be addressed in the undergraduate curriculum. While use of a database system gives full access to these data, those wanting to explore can undertake similar analyses using the `nycflights13` package on CRAN, which includes five dataframes that can be accessed within R (see http://www.amherst.edu/~nhorton/precursors for other example files and related resources).

```r
require(nycflights13)
airlines
```

```
## Source: local data frame [16 x 2]
##
##    carrier                        name
## 1       9E           Endeavor Air Inc.
## 2       AA      American Airlines Inc.
## 3       AS        Alaska Airlines Inc.
## 4       B6             JetBlue Airways
## 5       DL        Delta Air Lines Inc.
## 6       EV    ExpressJet Airlines Inc.
## 7       F9      Frontier Airlines Inc.
## 8       FL AirTran Airways Corporation
## 9       HA      Hawaiian Airlines Inc.
## 10      MQ                   Envoy Air
## 11      OO       SkyWest Airlines Inc.
## 12      UA       United Air Lines Inc.
## 13      US             US Airways Inc.
## 14      VX              Virgin America
## 15      WN      Southwest Airlines Co.
## 16      YV          Mesa Airlines Inc.
```

```
airports
```

```
## Source: local data frame [1,397 x 7]
##
##    faa                           name      lat        lon  alt tz dst
## 1  04G              Lansdowne Airport 41.13047  -80.61958 1044 -5   A
## 2  06A  Moton Field Municipal Airport 32.46057  -85.68003  264 -5   A
## 3  06C            Schaumburg Regional 41.98934  -88.10124  801 -6   A
## 4  06N                Randall Airport 41.43191  -74.39156  523 -5   A
## 5  09J          Jekyll Island Airport 31.07447  -81.42778   11 -4   A
## 6  0A9 Elizabethton Municipal Airport 36.37122  -82.17342 1593 -4   A
## 7  0G6        Williams County Airport 41.46731  -84.50678  730 -5   A
## 8  0G7  Finger Lakes Regional Airport 42.88356  -76.78123  492 -5   A
## 9  0P2   Shoestring Aviation Airfield 39.79482  -76.64719 1000 -5   U
## 10 0S9          Jefferson County Intl 48.05381 -122.81064  108 -8   A
## .. ...                            ...      ...        ...  ... .. ...
```

```
planes
```

```
## Source: local data frame [3,322 x 9]
##
##    tailnum year                    type     manufacturer     model engines
## 1   N10156 2004 Fixed wing multi engine          EMBRAER EMB-145XR       2
## 2   N102UW 1998 Fixed wing multi engine AIRBUS INDUSTRIE  A320-214       2
## 3   N103US 1999 Fixed wing multi engine AIRBUS INDUSTRIE  A320-214       2
## 4   N104UW 1999 Fixed wing multi engine AIRBUS INDUSTRIE  A320-214       2
## 5   N10575 2002 Fixed wing multi engine          EMBRAER EMB-145LR       2
## 6   N105UW 1999 Fixed wing multi engine AIRBUS INDUSTRIE  A320-214       2
## 7   N107US 1999 Fixed wing multi engine AIRBUS INDUSTRIE  A320-214       2
## 8   N108UW 1999 Fixed wing multi engine AIRBUS INDUSTRIE  A320-214       2
## 9   N109UW 1999 Fixed wing multi engine AIRBUS INDUSTRIE  A320-214       2
## 10  N110UW 1999 Fixed wing multi engine AIRBUS INDUSTRIE  A320-214       2
## ..     ...  ...                     ...              ...       ...     ...
## Variables not shown: seats (int), speed (int), engine (chr)
```

```
flights
```

```
## Source: local data frame [336,776 x 16]
##
##    year month day dep_time dep_delay arr_time arr_delay carrier tailnum
## 1  2013     1   1      517         2      830        11      UA  N14228
## 2  2013     1   1      533         4      850        20      UA  N24211
## 3  2013     1   1      542         2      923        33      AA  N619AA
## 4  2013     1   1      544        -1     1004       -18      B6  N804JB
## 5  2013     1   1      554        -6      812       -25      DL  N668DN
## 6  2013     1   1      554        -4      740        12      UA  N39463
## 7  2013     1   1      555        -5      913        19      B6  N516JB
## 8  2013     1   1      557        -3      709       -14      EV  N829AS
## 9  2013     1   1      557        -3      838        -8      B6  N593JB
## 10 2013     1   1      558        -2      753         8      AA  N3ALAA
## ..  ...   ... ...      ...       ...      ...       ...     ...     ...
## Variables not shown: flight (int), origin (chr), dest (chr), air_time
##   (dbl), distance (dbl), hour (dbl), minute (dbl)
```

```
weather
```

```
## Source: local data frame [8,719 x 14]
## Groups: month, day, hour
##
##    origin year month day hour  temp  dewp humid wind_dir wind_speed
## 1     EWR 2013     1   1    0 37.04 21.92 53.97      230   10.35702
## 2     EWR 2013     1   1    1 37.04 21.92 53.97      230   13.80936
## 3     EWR 2013     1   1    2 37.94 21.92 52.09      230   12.65858
## 4     EWR 2013     1   1    3 37.94 23.00 54.51      230   13.80936
## 5     EWR 2013     1   1    4 37.94 24.08 57.04      240   14.96014
## 6     EWR 2013     1   1    6 39.02 26.06 59.37      270   10.35702
## 7     EWR 2013     1   1    7 39.02 26.96 61.63      250    8.05546
## 8     EWR 2013     1   1    8 39.02 28.04 64.43      240   11.50780
## 9     EWR 2013     1   1    9 39.92 28.04 62.21      250   12.65858
## 10    EWR 2013     1   1   10 39.02 28.04 64.43      260   12.65858
## ..    ...  ...   ... ...  ...   ...   ...   ...      ...        ...
## Variables not shown: wind_gust (dbl), precip (dbl), pressure (dbl), visib
##   (dbl)
```

A framework for data-related skills The statistical data analysis cycle involves the formulation of questions, collection of data, analysis, and interpretation of results (see Figure 1). Data preparation and manipulation is not just a first step, but a key component of this cycle (which will often be nonlinear, see also http://www.jstatsoft.org/v59/i10/paper). When working with data, analysts must first determine what is needed, describe this solution in terms that a computer can understand, and execute the code.

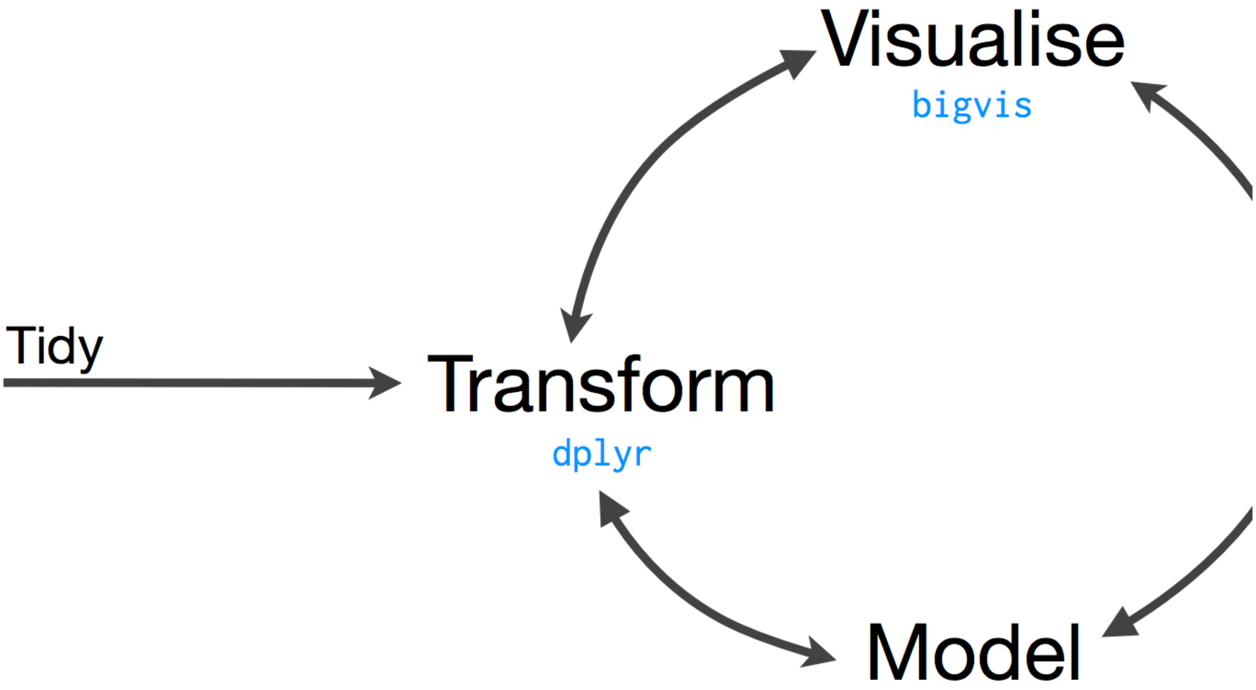


Figure 1: Statistical data analysis cycle (source: http://bit.ly/bigrdata4)

Here we illustrate how the `dplyr` package in R (http://cran.r-project.org/web/packages/dplyr) can be used to build a powerful and broadly accessible foundation for data manipulation. This approach is attractive because it provides simple functions that correspond to the most common data manipulation operations (or *verbs*) and uses efficient storage approaches so that the analyst can focus on the analysis. (Other systems could certainly be used in this manner, see for example http://iase-web.org/icots/9/proceedings/pdfs/ICOTS9_C134_CARVER.pdf.)

verb	meaning
select()	select variables (or columns)
filter()	subset observations (or rows)
mutate()	add new variables (or columns)
arrange()	re-order the observations
summarise()	reduce to a single row
group_by()	aggregate
left_join()	merge two data objects
distinct()	remove duplicate entries
collect()	force computation and bring data back into R

Table 1: Key verbs in `dplyr` and `tidyr` to support data management and manipulation (see http://bit.ly/bigrdata4 for more details)

Airline delays We demonstrate how to undertake analysis using the tools in the `dplyr` package. A smaller dataset is available for n=336,776 New York City flights in 2013 within the `nycflights13` package. The interface in R accessing the full database is almost identical in terms of the `dplyr` functionality, with the same functions being used.

Students can use this dataset to address questions that they find real and relevant. (It is not hard to find motivation for investigating patterns of flight delays. Ask students: have you ever been stuck in an airport because your flight was delayed or cancelled and wondered if you could have predicted the delay if you'd had more data?)

We begin by loading needed packages and connecting to a database containing the flight, airline, airport, and airplane data (see SIDEBAR: Databases).

Filtering observations We start with an analysis focused on three smaller airports in the Northeast. This illustrates the use of `filter()`, which allows the specification of a subset of rows of interest in the `airports` table (or dataset). We first start by exploring the `airports` table. Suppose we wanted to find out which airports certain codes belong to?

```
filter(airports, faa %in% c('ALB', 'BDL', 'BTV'))
```

```
## Source: local data frame [3 x 7]
##
##   faa            name      lat       lon alt tz dst
## 1 ALB     Albany Intl 42.74827 -73.80169 285 -5   A
## 2 BDL    Bradley Intl 41.93889 -72.68322 173 -5   A
## 3 BTV Burlington Intl 44.47186 -73.15328 335 -5   A
```

Aggregating observations Next we aggregate the counts of flights at all three of these airports at the monthly level (in the `ontime` flight-level table), using the `group_by()` and `summarise()` functions. The `collect()` function forces the evaluation. These functions are connected using the `%>%` operator. This pipes the results from one object or function as input to the next in an efficient manner.

```
airportcounts <- flights %>%
   filter(dest %in% c('ALB', 'BDL', 'BTV')) %>%
   group_by(year, month, dest) %>%
   summarise(count = n()) %>%
   collect()
```

Creating new derived variables Next we add a new column by constructing a date variable (using `mutate()` and helper functions from the `lubridate` package), then generate a time series plot.

```
library(lubridate)
airportcounts <- airportcounts %>%
  mutate(Date = ymd(paste(year, "-", month, "-01", sep="")))
head(airportcounts) # list only the first six observations
```

```
## Source: local data frame [6 x 5]
## Groups: year, month
##
##   year month dest count       Date
## 1 2013     1  ALB    64 2013-01-01
## 2 2013     1  BDL    37 2013-01-01
## 3 2013     1  BTV   223 2013-01-01
## 4 2013     2  ALB    58 2013-02-01
## 5 2013     2  BDL    46 2013-02-01
## 6 2013     2  BTV   189 2013-02-01
```

```
xyplot(count ~ Date, groups=dest, type=c("p","l"), lwd=2,
       auto.key=list(columns=3), xlab="Year",
       ylab="Number of flights per month", data=airportcounts)
```

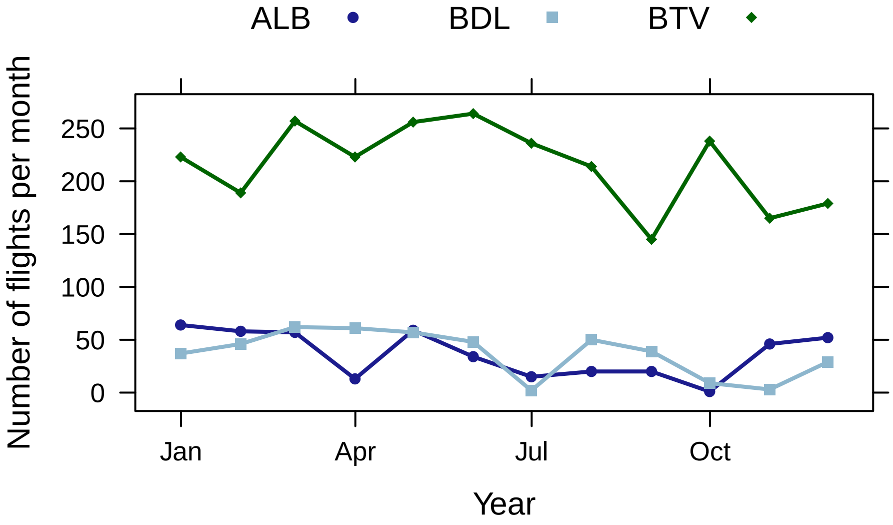


Figure 2: Comparison of the number of flights arriving at three airports by month in 2013.

We observe in Figure 2 that there are some interesting patterns over time for these airports. Burlington has the largest number of flights.

Sorting and selecting Another important verb is `arrange()`, which in conjunction with `head()` lets us display the months with the largest number of flights. Here we need to use `ungroup()`, since otherwise the data would remain aggregated by year, month, and destination.

```
airportcounts %>%
  ungroup() %>%
  arrange(desc(count)) %>%
  select(count, year, month, dest) %>%
  head()
```

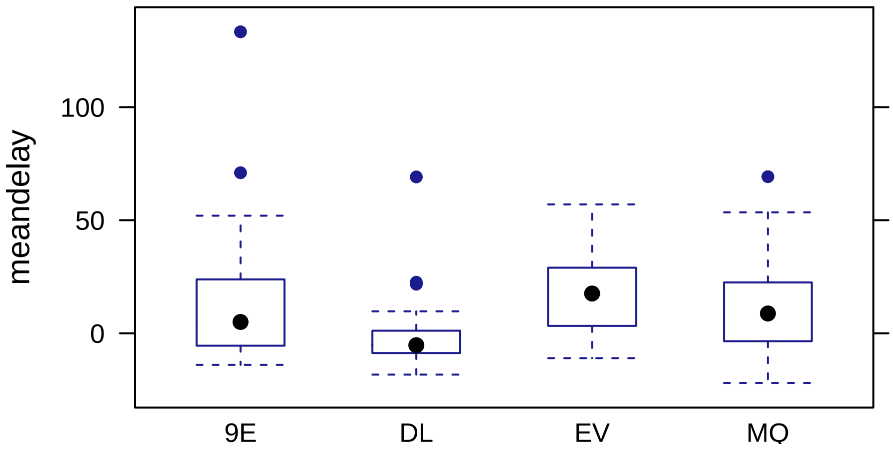
```
## Source: local data frame [6 x 4]
##
##   count year month dest
## 1   264 2013     6  BTV
## 2   257 2013     3  BTV
## 3   256 2013     5  BTV
## 4   238 2013    10  BTV
## 5   236 2013     7  BTV
## 6   223 2013     1  BTV
```

We can compare flight delays between two airlines serving a city pair. For example, which airline was most reliable flying from New York to Minneapolis/St. Paul (MSP) in January, 2013? Here we demonstrate how to calculate an average delay for each day . We create the analytic dataset through use of `select()` (to pick the variables to be included), `filter()` (to select a tiny subset of the observations), and then repeat the previous aggregation.

```
delays <- flights %>%
  select(origin, dest, year, month, day, carrier, arr_delay) %>%
  filter(dest == 'MSP' & month == 1) %>%
  group_by(year, month, day, carrier) %>%
  summarise(meandelay = mean(arr_delay), count = n())
options(digits=3)
favstats(~ meandelay, data=delays)
```

```
##  min    Q1 median   Q3 max mean   sd   n missing
##  -22 -5.51   4.12 22.6 133 11.1 23.8 112      12
```

```
bwplot(meandelay ~ carrier, data=delays)
```

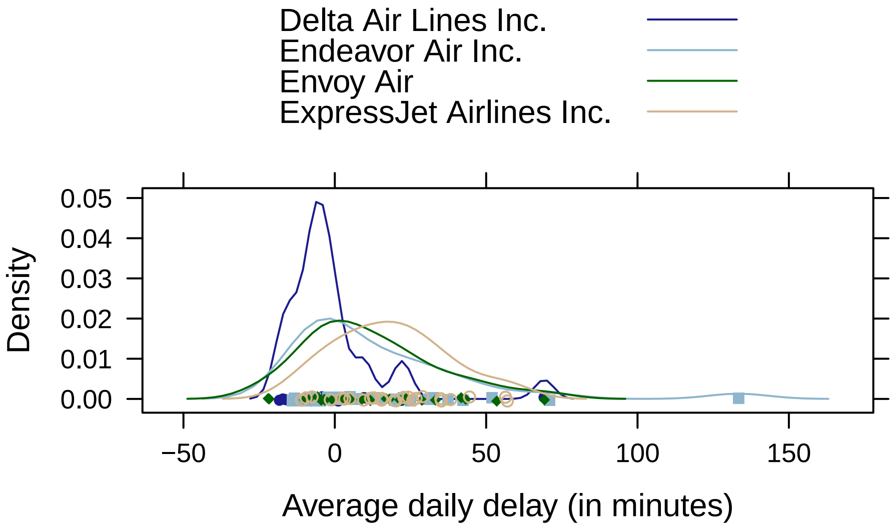


Merging Merging is another key capacity for students to master. Here, the full carrier names are merged (or joined, in database parlance) to facilitate the comparison, using the `left_join()` function to provide a less terse full name for the airlines in the legend of the figure.

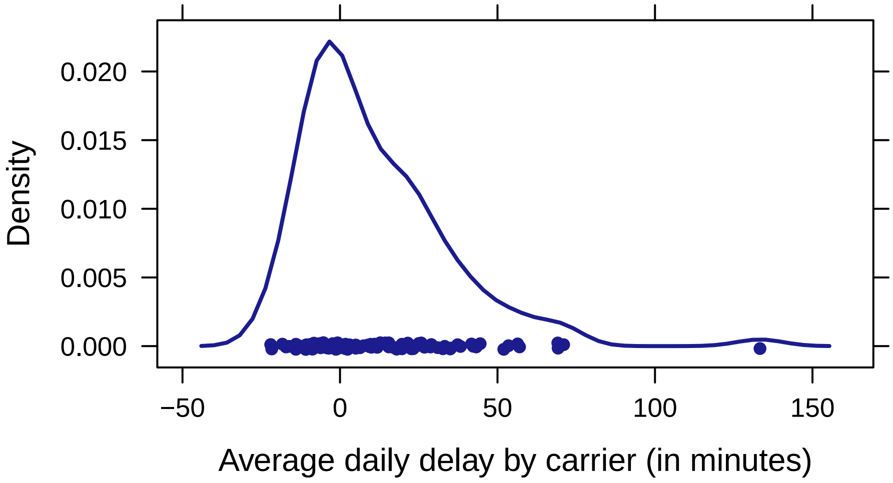
```
merged <- left_join(delays, airlines, by=c("carrier" = "carrier"))
merged <- mutate(merged, name = droplevels(name))
head(merged)
```

```
## Source: local data frame [6 x 7]
## Groups: year, month, day
##
##   year month day carrier meandelay count                    name
## 1 2013     1   1      9E     25.00     2       Endeavor Air Inc.
## 2 2013     1   1      DL     -5.29     7    Delta Air Lines Inc.
## 3 2013     1   1      EV     29.00     1 ExpressJet Airlines Inc.
## 4 2013     1   1      MQ     69.25     4               Envoy Air
## 5 2013     1   2      9E      7.50     2       Endeavor Air Inc.
## 6 2013     1   2      DL     -3.33     9    Delta Air Lines Inc.
```

```
densityplot(~ meandelay, group=name, auto.key=TRUE,
  xlab="Average daily delay (in minutes)", data=merged)
```



```
densityplot(~ meandelay, xlab="Average daily delay by carrier (in minutes)", data=merged)
```



```
favstats(meandelay ~ name, data=merged)
```

```
##                       .group   min    Q1 median     Q3   max   mean   sd  n
## 1     Delta Air Lines Inc. -18.3 -8.63  -5.27  0.571  69.1 -0.727 17.6 26
## 2        Endeavor Air Inc. -14.0 -5.50   5.00 23.833 133.3 14.261 30.4 31
## 3                Envoy Air -22.0 -3.50   8.75 22.500  69.2 11.563 21.7 29
## 4 ExpressJet Airlines Inc. -11.0  3.69  17.58 28.458  57.0 18.599 18.6 26
##   missing
## 1       5
## 2       0
## 3       2
## 4       5
```

Figure 3: Comparison of mean flight delays from New York to Minneapolis/St. Paul in January, 2013

We see in Figure 3 that the airlines are fairly reliable, though there were some days with average delays of 60 minutes or more.

```
filter(merged, meandelay > 60) %>% arrange(desc(meandelay))
```

```
## Source: local data frame [4 x 7]
## Groups: year, month, day
##
##   year month day carrier meandelay count                  name
## 1 2013     1   1      MQ      69.2     4             Envoy Air
## 2 2013     1  16      9E     133.3     3     Endeavor Air Inc.
## 3 2013     1  23      DL      69.1     7  Delta Air Lines Inc.
## 4 2013     1  30      9E      71.0     3     Endeavor Air Inc.
```

Integrating bigger questions and datasets into the curriculum This opportunity to make a complex and interesting dataset accessible to students in introductory statistics is quite compelling. In the introductory (or first) statistics course, we explored airline delays without any technology through use of the "Judging Airlines" model eliciting activity (MEA) developed by the CATALST Group (http://serc.carleton.edu/sp/library/mea/examples/example5.html). This MEA guides students to develop ideas regarding center and variability and the basics of informal inferences using small samples of data for pairs of airlines flying out of Chicago.

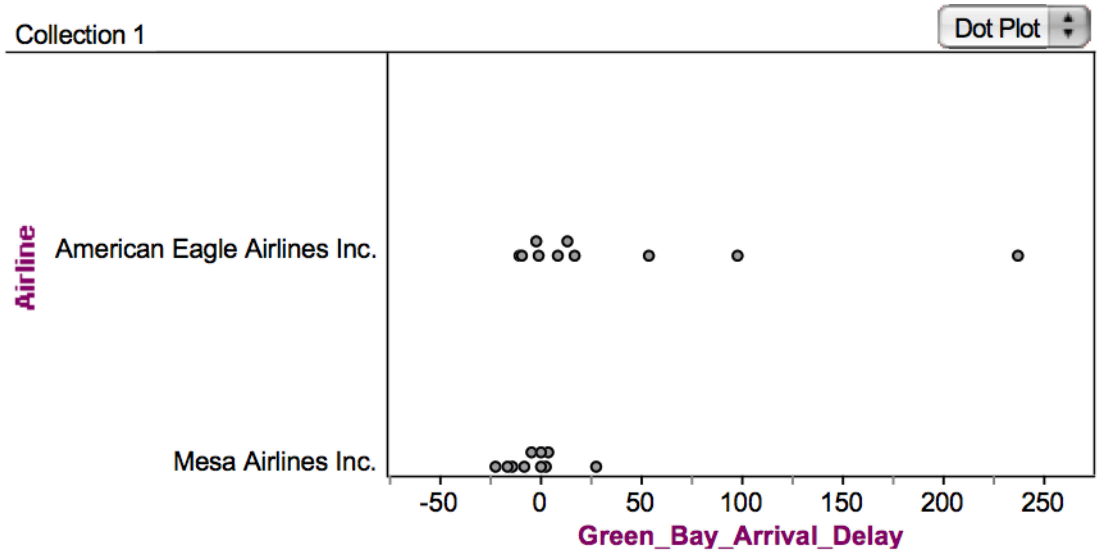


Figure 4: Fathom software display of sample airline delays data for a city pair used in the "Judging Airlines" MEA (model eliciting activity)

Figure 4 displays sample airline delays for ten flights each for American Eagle Airlines and Mesa Airlines flying from Chicago to Green Bay, Wisconsin. As part of this activity, students need to describe five possible sample statistics which could be used to compare the flight delays by airline. These might include the average, the maximum, the median, the 90th percentile, or the fraction that are late. Finally, they need to create a rule that incorporates at least two of those summary statistics that can be used to make a decision about whether one airline is more reliable. A possible rule might be to declare an airline is better than another if that airline has half an hour less average delay, and that same airline has 10% less delayed flights than the other (if the two measures of reliability differ in direction for the two airlines, no call is made).

To finish the assignment, students are provided with data for another four city pairs, asked to carry out their rule on these new "test" datasets, then summarize their results in a letter to the editor of *Chicago Magazine*.

Later in the course, the larger dataset can be reintroduced in several ways. It can be brought into class to illustrate univariate summaries or bivariate relationships (including more sophisticated visualization and graphical displays). Students can pose questions through projects or other extended assignments. A lab activity could have students explore their favorite airport or city pair (when comparing two airlines they will often find that only one airline services that connection, particularly for smaller airports.) Students could be asked to return to the informal "rule" they developed in an extension to assess its performance. Their rule can be programmed in R, and then carried out on a series of random samples from the flights from that city on that airline within that year. This allows them to see how often their rule picked an airline as being more reliable (using various subsets of the observed data as the "truth"). Finally, students can summarize the population of all flights, as a way to better understand sampling variability. This process reflects the process followed by analysts working with big data: sampling is used to generate hypotheses that are then tested against the complete dataset.

In a second course, more time is available to develop diverse statistical and computational skills. This includes more sophisticated data management and manipulation with explicit learning outcomes that are a central part of the course syllabus.

Other data wrangling and manipulation capacities can be introduced and developed using this example, including more elaborate data joins/merges (since there are tables providing additional (meta)data about planes). As an example, consider the many flights of plane N355NB, which flew out of Bradley airport in January, 2008.

```
filter(planes, tailnum=="N355NB")
```

```
## Source: local data frame [1 x 9]
##
##   tailnum year                    type manufacturer    model engines seats
## 1  N355NB 2002 Fixed wing multi engine       AIRBUS A319-114       2   145
## Variables not shown: speed (int), engine (chr)
```

We see that this is an Airbus 319.

```
singleplane <- filter(flights, tailnum=="N355NB") %>%
  select(year, month, day, dest, origin, distance)
head(singleplane)
```

```
## Source: local data frame [6 x 6]
##
##   year month day dest origin distance
## 1 2013     1   7  PIT    LGA      335
## 2 2013     1   8  FLL    LGA     1076
## 3 2013     1   9  PBI    LGA     1035
## 4 2013     1  10  MSP    LGA     1020
## 5 2013     1  21  PIT    LGA      335
## 6 2013     1  22  FLL    LGA     1076
```

```
sum(~ distance, data=singleplane)
```

```
## [1] 106914
```

This Airbus A319 has been very active, with 128 flights just in 2013 in the New York City area.

```
singleplane %>%
  group_by(dest) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  filter(count > 5)
```

```
## Source: local data frame [5 x 2]
##
##   dest count
## 1  ATL    46
## 2  DTW    19
## 3  MSP    10
## 4  FLL     9
## 5  TPA     7
```

Weather

Linkage to other data scraped from the Internet (e.g. detailed weather information for a particular airport or details about individual planes) may allow other questions to be answered.

```
head(weather)
```

```
## Source: local data frame [6 x 14]
## Groups: month, day, hour
##
##   origin year month day hour temp dewp humid wind_dir wind_speed wind_gust
## 1    EWR 2013     1   1    0 37.0 21.9  54.0      230       10.4      11.9
## 2    EWR 2013     1   1    1 37.0 21.9  54.0      230       13.8      15.9
## 3    EWR 2013     1   1    2 37.9 21.9  52.1      230       12.7      14.6
## 4    EWR 2013     1   1    3 37.9 23.0  54.5      230       13.8      15.9
## 5    EWR 2013     1   1    4 37.9 24.1  57.0      240       15.0      17.2
## 6    EWR 2013     1   1    6 39.0 26.1  59.4      270       10.4      11.9
## Variables not shown: precip (dbl), pressure (dbl), visib (dbl)
```

```
avgdelay <- flights %>%
  group_by(month, day) %>%
  filter(month < 13) %>%
  summarise(avgdelay = mean(arr_delay, na.rm=TRUE))
precip <- weather %>%
  group_by(month, day) %>%
  filter(month < 13) %>%
  summarise(totprecip = sum(precip), maxwind = max(wind_speed))
precip <- mutate(precip, anyprecip = ifelse(totprecip==0, "No", "Yes"))
merged <- left_join(avgdelay, precip, by=c("day", "month"))
head(merged)
```

```
## Source: local data frame [6 x 6]
## Groups: month
##
##   month day avgdelay totprecip maxwind anyprecip
## 1     1   1    12.65         0    16.1        No
## 2      1   2    12.69         0    18.4        No
## 3      1   3     5.73         0    11.5        No
## 4      1   4    -1.93         0    24.2        No
## 5      1   5    -1.53         0    18.4        No
## 6      1   6     4.24         0    15.0        No
```

A dramatic outlier emerges: windspeeds of 1000 mph are not common!

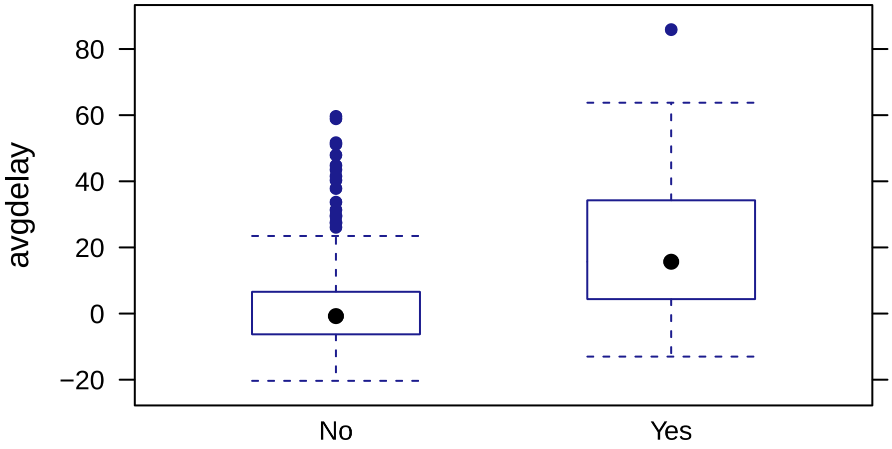
```
favstats(~ maxwind, data=merged)
```

```
##   min   Q1 median   Q3  max mean   sd   n missing
##  5.75 12.7   16.1 19.6 1048 19.3 54.4 363       2
```

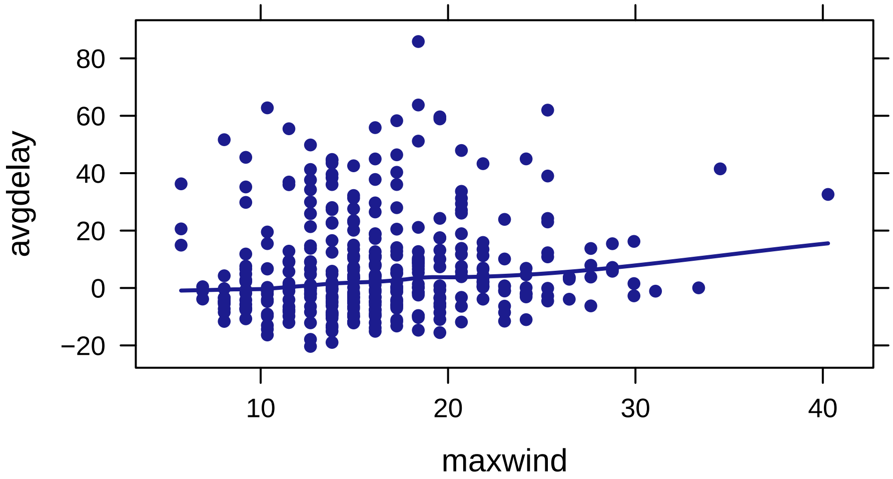
```
filter(merged, maxwind > 1000)
```

```
## Source: local data frame [1 x 6]
## Groups: month
##
##   month day avgdelay totprecip maxwind anyprecip
## 1     2  12    -2.14         0    1048        No
```

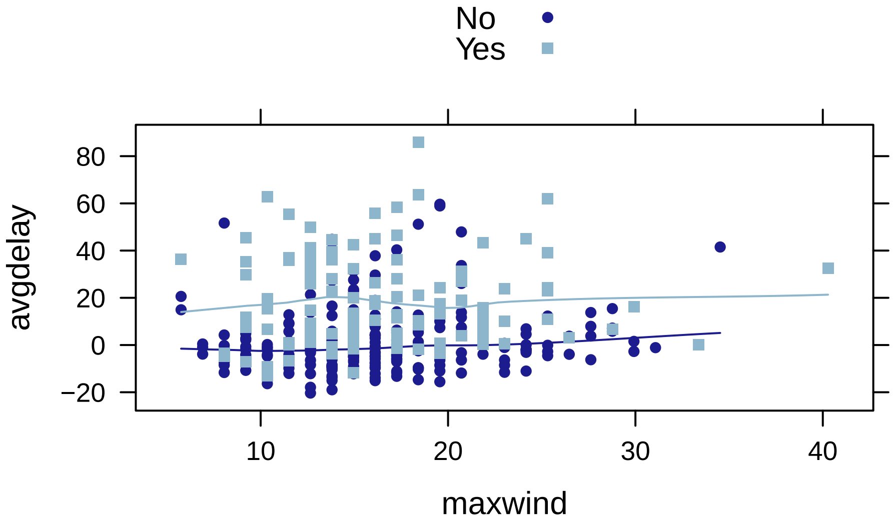
```
merged <- filter(merged, maxwind < 1000)
bwplot(avgdelay ~ anyprecip, data=merged)
```



```
xyplot(avgdelay ~ maxwind, type=c("p", "smooth"), data=merged)
```



```
xyplot(avgdelay ~ maxwind, groups=anyprecip, auto.key=TRUE, type=c("p", "smooth"), data=merged)
```



There is a modest relationship between average delay times and wind speed. The relationship is stronger between any precipitation (as seen in the last Figure)

Use of this rich dataset helps to excite students about the power of statistics, introduce tools that can help energize the next generation of data scientists, and build useful data-related skills.

Conclusion and next steps

Statistics students need to develop the capacity to make sense of the staggering amount of information collected in our increasingly data-centered world. In her 2013 book, Rachel Schutt succinctly summarized the challenges she faced as she moved into the workforce: "It was clear to me pretty quickly that the stuff I was working on at Google was different than anything I had learned at school." This anecdotal evidence is corroborated by the widely cited McKinsey report that called for the training of hundreds of thousands of workers with the skills to make sense of the rich and sophisticated data now available to make decisions (along with millions of new managers with the ability to comprehend these results). The disconnect between the complex analyses now demanded in industry and the instruction available in academia is a major challenge for the profession.

We agree that there are barriers and time costs to the introduction of reproducible analysis tools and more sophisticated data management and manipulation skills to our courses. Further guidance and research results are needed to guide our work in this area, along with illustrated examples, case studies, and faculty development. But these impediments must not slow down our adoption. As Schutt cautions in her book, statistics could be viewed as obsolete if this challenge is not embraced. We believe that the time to move forward in this manner is now, and believe that these these basic data-related skills provide a foundation for such efforts.

Copies of the R Markdown and formatted files for these analyses (to allow replication of the analyses) along with further background on databases and the Airline Delays dataset are available at http://www.amherst.edu/~nhorton/precursors. A previous version of this paper was presented in July, 2014 at the International Conference on Teaching Statistics (ICOTS9) in Flagstaff, AZ.

Further reading American Statistical Association Undergraduate Guidelines Workgroup (2014). 2014 Curriculum guidelines for undergraduate programs in statistical science. Alexandria, VA: American Statistical Association, http://www.amstat.org/education/curriculumguidelines.cfm.

Baumer, B., Cetinkaya-Rundel, M., Bray, A., Loi, L. and Horton, N.J. (2014). R Markdown: Integrating a reproducible analysis tool into introductory statistics, *Technology Innovations in Statistics Education*, http://escholarship.org/uc/item/90b2f5xh.

Finzer, W. (2013). The data science education dilemma. *Technology Innovations in Statistics Education*, http://escholarship.org/uc/item/7gv0q9dc.

Horton, N.J., Baumer, Ben S., and Wickham, H. (2014). Teaching precursors to data science in introductory and second courses in statistics, http://arxiv.org/abs/1401.3269.

Nolan, D. and Temple Lang, D. (2010). Computing in the statistics curricula, *The American Statistician*, 64, 97–107.

O'Neil, C. and Schutt R. (2013). Doing Data Science: Straight Talk from the Frontline, O'Reilly and Associates.

Wickham, H. (2011). ASA 2009 Data Expo, *Journal of Computational and Graphical Statistics*, 20(2):281-283.

SIDEBAR: What's in a word?

In their 2010 American Statistician paper, Deborah Nolan and Duncan Temple Lang describe the need for students to be able to "compute with data" to be able to answer statistical questions. Diane Lambert of Google calls this the capacity to "think with data". Statistics graduates need to be manage data, analyze it accurately, and communicate findings effectively. The Wikipedia data science entry states that "data scientists use the ability to find and interpret rich data sources, manage large amounts of data despite hardware, software, and bandwidth constraints, merge data sources, ensure consistency of datasets, create visualizations to aid in understanding data, build mathematical models using the data, present and communicate the data insights/findings to specialists and scientists in their team and if required to a non-expert audience." But what is the best word or phrase to describe these computational and data-related skills?

"Data wrangling" has been suggested as one possibility (and returned about 131,000 results on Google), though this connotes the idea of a long and complicated dispute, often involving livestock, which may not end well.

"Data grappling" is another option (about 7,500 results on Google), though this perhaps less attractive as it suggests pirates (and grappling hooks) or wrestling as combat sport or self defense.

"Data munging" (about 35,000 results on Google) is a common term in computer science used to describe changes to data (both constructive and destructive) or mapping from one format to another. A disadvantage of this term is that it has a somewhat pejorative sentiment.

"Data tidying" (about 900 results on Google) brings to mind the ideas of "bringing order to" or "arranging neatly".

"Data curation" (about 322,000 results on Google) is a term that focuses on a long-term time scale for use (and preservation). While important, this may be perceived a dusty and stale task.

"Data cleaning" (or "data cleansing", about 490,000 results on Google) is the process to identify and correct (or remove) invalid records from a dataset. Other related terms include "data standardization" and "data harmonization".

A search for "Data manipulation" yielded about 740,000 results on Google. Interestingly, this term on Wikipedia redirects to the "Misuse of statistics" page, implying the analyst might have malicious intentions and could torture the data to tell a particular story. The Wikipedia "Data manipulation language" page has no such negative connotations (and describes the Structured Query Language [SQL] as one such language). This dual meaning stems from the definition (from Merriam-Webster) of manipulate:

- To manage or utilize skillfully
- To control or play upon by artful, unfair, or insidious means especially to one's own advantage

"Data management" was the most common term, with more than 33,000,000 results on Google. The DAMA Data Management Body of Knowledge (DAMA-DMBOK, http://www.dama.org/files/public/DI_DAMA_DMBOK_Guide_Presentation_2007.pdf) provides a definition: "Data management is the development, execution and supervision of plans, policies, programs and practices that control, protect, deliver and enhance the value of data and information assets." While the term is somewhat clinical, does not necessarily capture the essential creativity required (and is decidedly non-sexy), data management may be the most appropriate phrase to describe the type of data-related skills students need to make sense of the information around them.

SIDEBAR: Making bigger datasets accessible through databases

This file used the `nycflights13` to access the data. But this is just a fraction of the available flight information. See http://www.amherst.edu/~nhorton/precursors for example code using SQLite.

Nolan and Temple Lang (2010) stress the importance of knowledge of information technologies, along with the ability to work with large datasets. Relational databases, first popularized in the 1970's, provide fast and efficient access to terabyte-sized files. These systems use a structured query language (SQL) to specify data operations. Surveys of graduates from statistics programs have noted that familiarity with databases and SQL would have been helpful as they moved to the workforce.

Database systems have been highly optimized and tuned since they were first invented. Connections between general purpose statistics packages such as R and database systems can be facilitated through use of SQL. Table 2 describes key operators for data manipulation in SQL.

verb	meaning
SELECT	create a new result set from a table
FROM	specify table
WHERE	subset observations
GROUP BY	aggregate
ORDER	re-order the observations
DISTINCT	remove duplicate values
JOIN	merge two data objects

Table 2: Key operators to support data management and manipulation in SQL (structured query language)

Use of a SQL interface to large datasets is attractive as it allow the exploration of datasets that would be impractical to analyze using general purpose statistical packages. In this application, much of the heavy lifting and data manipulation is done within the database system, with the results made available within the general purpose statistics package.

The ASA Data Expo 2009 website (http://stat-computing.org/dataexpo/2009) provides full details regarding how to download the Expo data (1.6 gigabytes compressed, 12 gigabytes uncompressed through 2008), set up a database using SQLite (http://www.sqlite.org), add indexing, and then access it from within R or RStudio. This is very straightforward to undertake (it took the first author less than 2 hours to set up using several years of data), though there are some limitations to the capabilities of SQLite.

MySQL (http://www.mysql.com, described as the world's most popular open source database) and PostgreSQL are more fully-featured systems (albeit with somewhat more complex installation and configuration).

The use of SQL within R (or other systems) is straightforward once the database has been created (either locally or remotely). An add-on package (such as `RMySQL` or `RSQLite`) must be installed and loaded, then a connection made to a local or remote database. In combination with tools such as R Markdown (which make it easy to provide a template and support code, described in detail in "Five Concrete Reasons Your Students Should Be Learning to Analyze Data in the Reproducible Paradigm", http://chance.amstat.org/2014/09/reproducible-paradigm) students can start to tackle more interesting and meatier questions using larger databases set up by their instructors. Instructors wanting to integrate databases into their repertoire may prefer to start with SQLite, then graduate to more sophisticated systems (which can be accessed remotely) using MySQL.

The `dplyr` package encapsulates and replaces the SQL interface for either system. It also features *lazy* evaluation, where operations are not undertaken until absolutely necessary.