Math 300 NTI Lesson 2

Scatterplot

Professor Bradley Warner

June, 2022

Contents

Objectives

1. Create a scatterplot using the `ggplot()` function.
2. Interpret the relationship between variables in a scatterplot.
3. Refine and improve scatterplots to illustrate relevant points by prepossessing the data or using functions such as `alpha()` and `geom_jitter()`.

Reading

Chapter 2 - 2.3

Lesson

Remember that you will be running this more like a lab than a lecture. You want them using `R` and answering questions.

Work through the learning checks LC2.1 - LC2.8.

- You want to emphasize that creating plots has two purposes. The first is exploratory, that is to learn about the data and, in particular for scatterplots, the relationship between two quantitative variables. Plots are also used for explanatory purposes. You want to convey to the audience an important or relevant feature of the data.
- For a scatterplot, we usually want to put the variable we believe to be the predictor or explanatory variable on the x-axis. On the y-axis we put the respone or dependent variable. Sometimes it is not clear which is which and in this case it does not matter which variable goes on which axis.

- Using the + symbol in ggplot() can be confusing, but emphasize that we are creating layers. Tell your students to ask themselves what the want R to do? And what does R need to do this? For a plot, we need data and then we need to tell R what the aesthetics and geoms are. The + symbol allows us to build in layers.
- You can show them how to insert an R code chunk into a markdown. There is the pulldown menu but you can also use Ctrl-Alt-I. Demo this in class.

Setup

```
library(nycflights13)
library(ggplot2)
library(dplyr)
```

We need to create the ***alaska_flights*** *data object. You may want to walk them through this code.*

```
alaska_flights <- flights %>%
  filter(carrier == "AS")
```

```
head(alaska_flights)
```

```
## # A tibble: 6 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
## 1  2013     1     1      724            725        -1     1020           1030
## 2  2013     1     1     1808           1815        -7     2111           2130
## 3  2013     1     2      722            725        -3      949           1030
## 4  2013     1     2     1818           1815         3     2131           2130
## 5  2013     1     3      724            725        -1     1012           1030
## 6  2013     1     3     1817           1815         2     2121           2130
## # ... with 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

We are using the ***str()*** *function in our solution. You should point this out to them.*

LC 2.1 (Objective 3)

(LC2.1) Take a look at both the flights and alaska_flights data frames by running View(flights) and View(alaska_flights) in the console. In what respect do these data frames differ? For example, think about the number of rows in each dataset.

Solution: flights contains all flight data, while alaska_flights contains only data from Alaskan carrier "AS". We can see that flights has 336776 rows while alaska_flights has only 714

```
str(flights)
```

```
## tibble [336,776 x 19] (S3: tbl_df/tbl/data.frame)
##  $ year          : int [1:336776] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
##  $ month         : int [1:336776] 1 1 1 1 1 1 1 1 1 1 ...
##  $ day           : int [1:336776] 1 1 1 1 1 1 1 1 1 1 ...
##  $ dep_time      : int [1:336776] 517 533 542 544 554 554 555 557 557 558 ...
##  $ sched_dep_time: int [1:336776] 515 529 540 545 600 558 600 600 600 600 ...
##  $ dep_delay     : num [1:336776] 2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
##  $ arr_time      : int [1:336776] 830 850 923 1004 812 740 913 709 838 753 ...
##  $ sched_arr_time: int [1:336776] 819 830 850 1022 837 728 854 723 846 745 ...
##  $ arr_delay     : num [1:336776] 11 20 33 -18 -25 12 19 -14 -8 8 ...
##  $ carrier       : chr [1:336776] "UA" "UA" "AA" "B6" ...
##  $ flight        : int [1:336776] 1545 1714 1141 725 461 1696 507 5708 79 301 ...
##  $ tailnum       : chr [1:336776] "N14228" "N24211" "N619AA" "N804JB" ...
##  $ origin        : chr [1:336776] "EWR" "LGA" "JFK" "JFK" ...
##  $ dest          : chr [1:336776] "IAH" "IAH" "MIA" "BQN" ...
##  $ air_time      : num [1:336776] 227 227 160 183 116 150 158 53 140 138 ...
##  $ distance      : num [1:336776] 1400 1416 1089 1576 762 ...
##  $ hour          : num [1:336776] 5 5 5 5 6 5 6 6 6 6 ...
##  $ minute        : num [1:336776] 15 29 40 45 0 58 0 0 0 0 ...
##  $ time_hour     : POSIXct[1:336776], format: "2013-01-01 05:00:00" "2013-01-01 05:00:00" ...
```

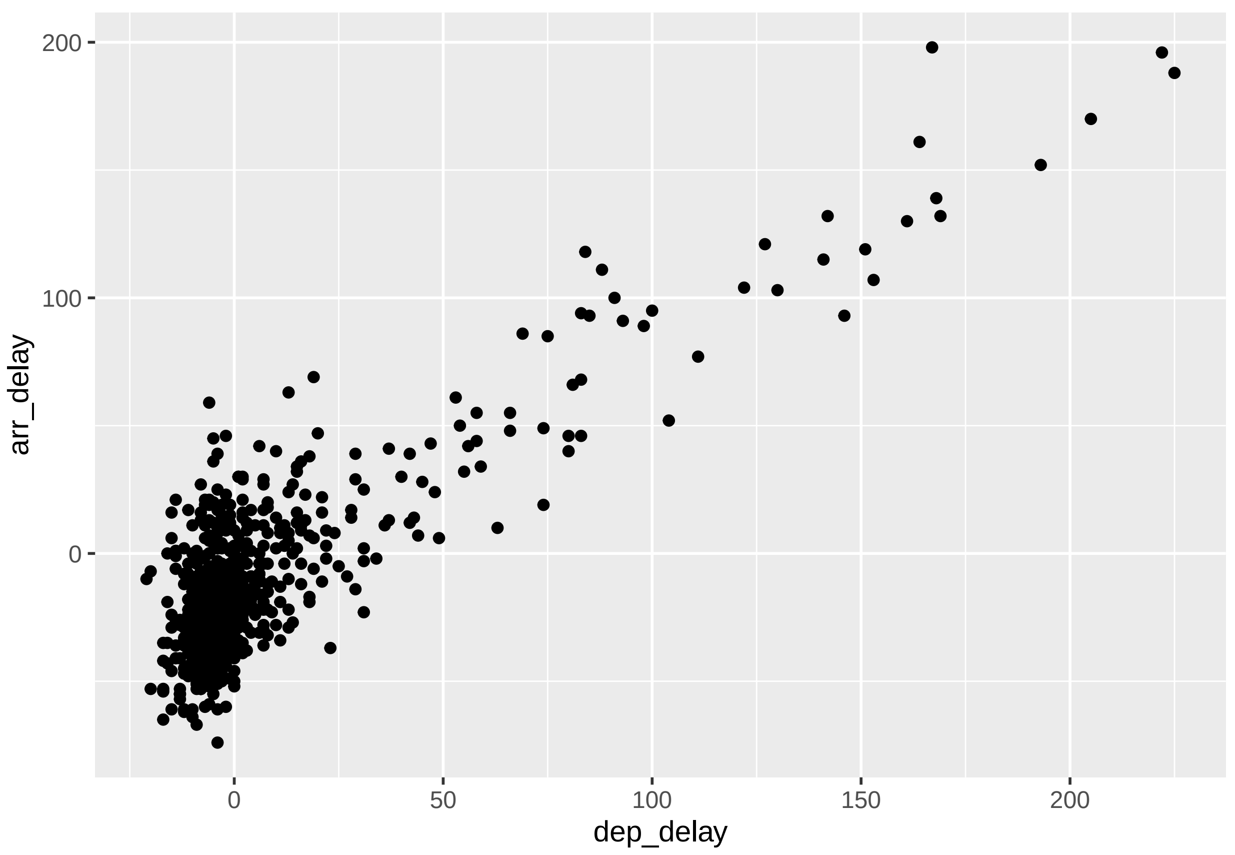
```r
str(alaska_flights)
```

```
## tibble [714 x 19] (S3: tbl_df/tbl/data.frame)
##  $ year          : int [1:714] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
##  $ month         : int [1:714] 1 1 1 1 1 1 1 1 1 1 ...
##  $ day           : int [1:714] 1 1 2 2 3 3 4 4 5 5 ...
##  $ dep_time      : int [1:714] 724 1808 722 1818 724 1817 725 1808 725 1803 ...
##  $ sched_dep_time: int [1:714] 725 1815 725 1815 725 1815 725 1815 725 1815 ...
##  $ dep_delay     : num [1:714] -1 -7 -3 3 -1 2 0 -7 0 -12 ...
##  $ arr_time      : int [1:714] 1020 2111 949 2131 1012 2121 1031 2101 1011 2118 ...
##  $ sched_arr_time: int [1:714] 1030 2130 1030 2130 1030 2130 1030 2130 1030 2130 ...
##  $ arr_delay     : num [1:714] -10 -19 -41 1 -18 -9 1 -29 -19 -12 ...
##  $ carrier       : chr [1:714] "AS" "AS" "AS" "AS" ...
##  $ flight        : int [1:714] 11 7 11 7 11 7 11 7 11 7 ...
##  $ tailnum       : chr [1:714] "N594AS" "N553AS" "N592AS" "N552AS" ...
##  $ origin        : chr [1:714] "EWR" "EWR" "EWR" "EWR" ...
##  $ dest          : chr [1:714] "SEA" "SEA" "SEA" "SEA" ...
##  $ air_time      : num [1:714] 338 336 314 332 325 327 345 338 330 343 ...
##  $ distance      : num [1:714] 2402 2402 2402 2402 2402 ...
##  $ hour          : num [1:714] 7 18 7 18 7 18 7 18 7 18 ...
##  $ minute        : num [1:714] 25 15 25 15 25 15 25 15 25 15 ...
##  $ time_hour     : POSIXct[1:714], format: "2013-01-01 07:00:00" "2013-01-01 18:00:00" ...
```

Build the plot for the next set of learning checks.

```r
ggplot(data = alaska_flights, mapping = aes(x = dep_delay, y = arr_delay)) +
  geom_point()
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```



LC 2.2 (Objective 2)

(LC2.2) What are some practical reasons why `dep_delay` and `arr_delay` have a positive relationship?

Solution: The later a plane departs, typically the later it will arrive.

LC 2.3 (Objective 2)

(LC2.3) What variables in the `weather` data frame would you expect to have a negative correlation (i.e. a negative relationship) with `dep_delay`? Why? Remember that we are focusing on numerical variables here. Hint: Explore the `weather` dataset by using the `View()` function.

Solution: An example in the `weather` dataset is `visib`, which measure visibility in miles. As visibility increases, we would expect departure delays to decrease.

LC 2.4 (Objective 2)

(LC2.4) Why do you believe there is a cluster of points near (0, 0)? What does (0, 0) correspond to in terms of the Alaskan flights?

Solution: The point (0,0) means no delay in departure nor arrival. From the point of view of Alaska airlines, this means the flight was on time. It seems most flights are at least close to being on time.

LC 2.5 (Objective 2)

(LC2.5) What are some other features of the plot that stand out to you?

Solution: Different people will answer this one differently. One answer is most flights depart and arrive less than an hour late.

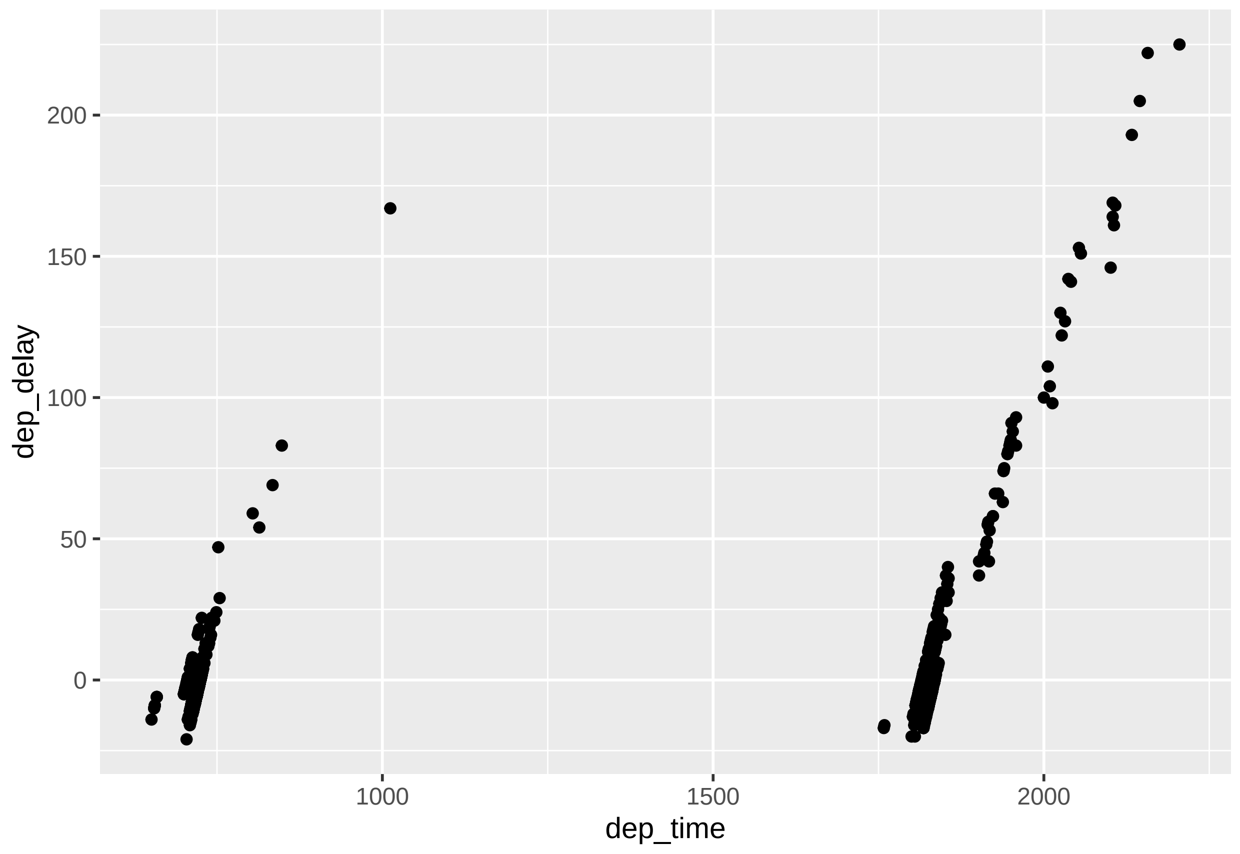
LC 2.6 (Objective 1)

(LC2.6) Create a new scatterplot using different variables in the `alaska_flights` data frame by modifying the example above.

Solution: Many possibilities for this one, see the plot below. Is there a pattern in departure delay depending on when the flight is scheduled to depart? Interestingly, there seems to be only two blocks of time where flights depart.

```
ggplot(data = alaska_flights, mapping = aes(x = dep_time, y = dep_delay)) +
  geom_point()
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



LC 2.7 (Objective 2)

(LC2.7) Why is setting the `alpha` argument value useful with scatterplots? What further information does it give you that a regular scatterplot cannot?

Solution: It thins out the points so we address overplotting. But more importantly it hints at the (statistical) **density** and **distribution** of the points: where are the points concentrated, where do they occur.

LC 2.8 (Objective 2, 3)

```r
ggplot(data = alaska_flights, mapping = aes(x = dep_delay, y = arr_delay)) +
  geom_point()
```

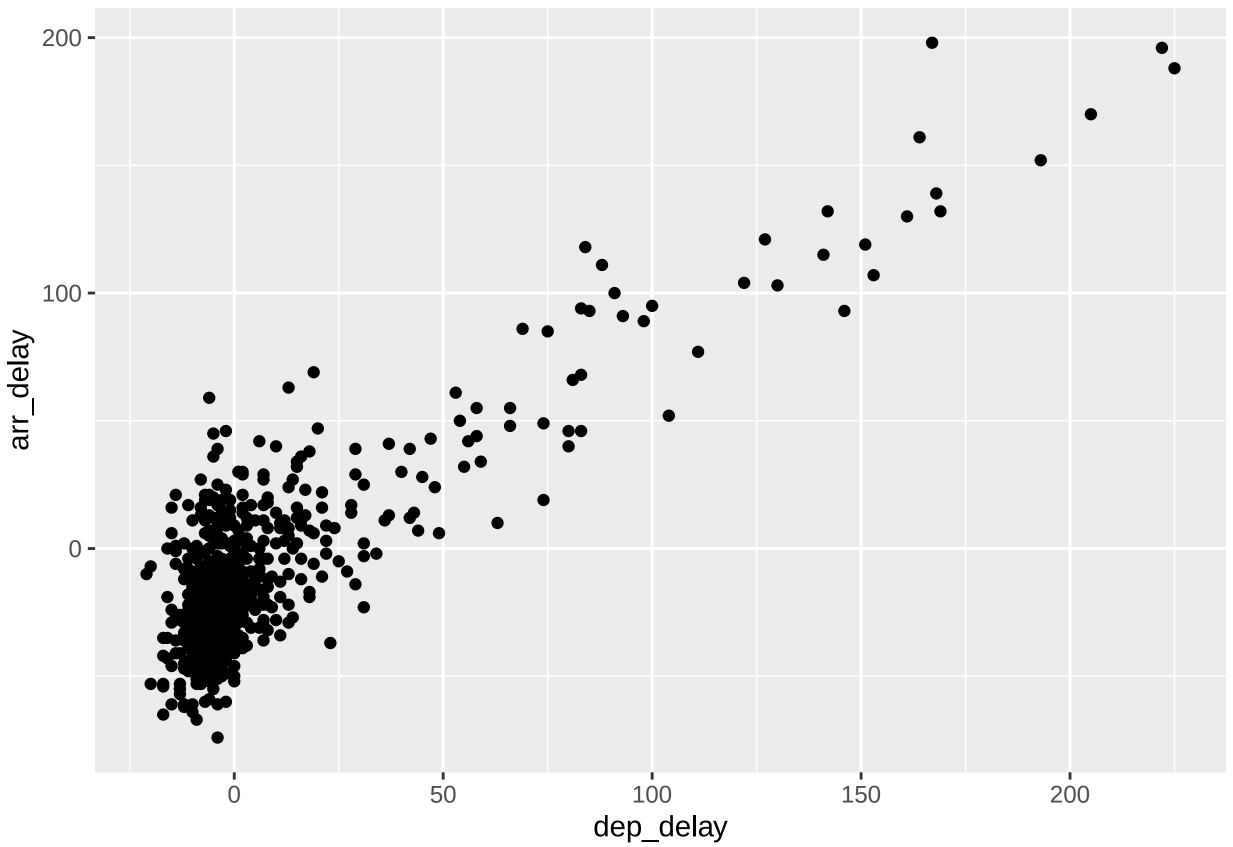


Figure 1: Figure 1: Arrival delays versus departure delays for Alaska Airlines flights from NYC in 2013.

```r
ggplot(data = alaska_flights, mapping = aes(x = dep_delay, y = arr_delay)) +
  geom_point(alpha = 0.2)
```

(LC2.8) After viewing the Figure 2 above, give an approximate range of arrival delays and departure delays that occur the most frequently. How has that region changed compared to when you observed the same plot without the `alpha = 0.2` set in Figure 1?

Solution: The lower plot suggests that most Alaska flights from NYC depart between 12 minutes early and on time and arrive between 50 minutes early and on time.

Documenting software

- File creation date: 2022-06-04

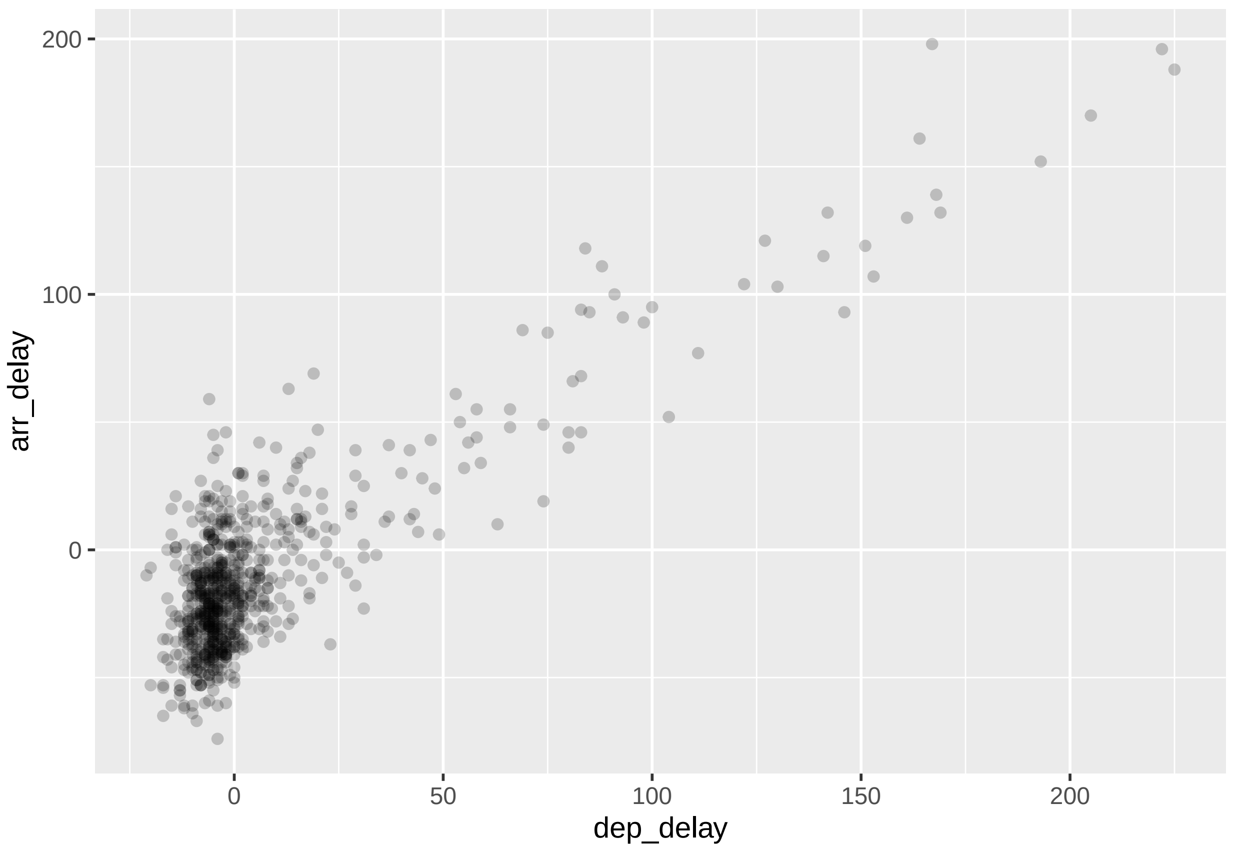


Figure 2: Figure 2: Arrival vs. departure delays scatterplot with alpha = 0.2

- R version 4.1.3 (2022-03-10)
- `ggplot2` package version: 3.3.6
- `dplyr` package version: 1.0.9
- `nycflights13` package version: 1.0.2