r4ds Ex 13

MW

2019/07/03

13.1

1

> Imagine you wanted to draw (approximately) the route each plane flies from its origin to its destination. What variables would you need? What tables would you need to combine?

We need to combine `flights` and `airport` due to requiring latitude and longitude of airports of destination.

2

> I forgot to draw the relationship between `weather` and `airports`. What is the relationship and how should it appear in the diagram?

`airports$faa` and `weather$origin` are relationships as foreign keys.

3

> `weather` only contains information for the origin (NYC) airports. If it contained weather records for all airports in the USA, what additional relation would it define with `flights`?

There are weather of destinations.

4

> We know that some days of the year are "special", and fewer people than usual fly on them. How might you represent that data as a data frame? What would be the primary keys of that table? How would it connect to the existing tables?

```
holiday <- tribble(
    ~year, ~month, ~day, ~holiday,
    2013, 01, 01, "New Year",
    2013, 12, 25, "Christmas Day"
)
```

13.3

1

Add a surrogate key to flights.

```
flights %>% mutate(id=1:nrow(.))
```

```
## # A tibble: 336,776 x 20
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 336,766 more rows, and 13 more variables: sched_arr_time <int>,
## #   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>, id <int>
```

2

Identify the keys in the following datasets > 1. Lahman::Batting, > 2. babynames::babynames > 3. nasaweather::atmos > 4. fueleconomy::vehicles > 5. ggplot2::diamonds (You might need to install some packages and read some documentation.)

1.

In 1.-4. there are primary keys of each dataset.

```
Lahman::Batting %>% as_tibble() %>%
    group_by(playerID, yearID, stint) %>% #grouping by primary keys
    mutate(count=n()) %>%
    filter(count>1)
```

```
## # A tibble: 0 x 23
## # Groups:   playerID, yearID, stint [0]
## # ... with 23 variables: playerID <chr>, yearID <int>, stint <int>,
## #   teamID <fct>, lgID <fct>, G <int>, AB <int>, R <int>, H <int>,
```

```
## #   X2B <int>, X3B <int>, HR <int>, RBI <int>, SB <int>, CS <int>,
## #   BB <int>, SO <int>, IBB <int>, HBP <int>, SH <int>, SF <int>,
## #   GIDP <int>, count <int>
```

Basically, 2.-4. are same as 1.

5

There are no primary keys.

```
ggplot2::diamonds %>%
    distinct() %>%
    nrow()
```

```
## [1] 53794
```

```
ggplot2::diamonds %>%
    nrow()
```

```
## [1] 53940
```

3

> Draw a diagram illustrating the connections between the `Batting`, `Master`, and `Salaries` tables in the Lahman package. Draw another diagram that shows the relationship between `Master`, `Managers`, `AwardsManagers`. How would you characterise the relationship between the `Batting`, `Pitching`, and `Fielding` tables?

We can create diagram in R using `DiagrameR` or `datamodelr`.

```
DiagrammeR::grViz("13Ex/1.dot")
DiagrammeR::grViz("13Ex/2.dot")
```

1.dot

```
digraph subgraph_label {
  rankdir = TB
  subgraph cluster0{
    yearID_S[label="yearID"]
    teamID_S[label="teamID"]
    playerID_S[label="playerID"]
    label = "Salaries"
    {rank = same; yearID; teamID; playerID;}
  }
  subgraph cluster1{
    playerID_M[label="playerID"]
    label = "Master"
  }
```

```
  subgraph cluster2{
    playerID_B[label="playerID"]
    yearID_B[label="yearID"]
    stint_B[label="stint"]
    label = "Batting"
    {rank = same; yearID; stint; playerID;}
  }
  playerID_M -> playerID_S
  playerID_M -> playerID_B
}
```

2.dot

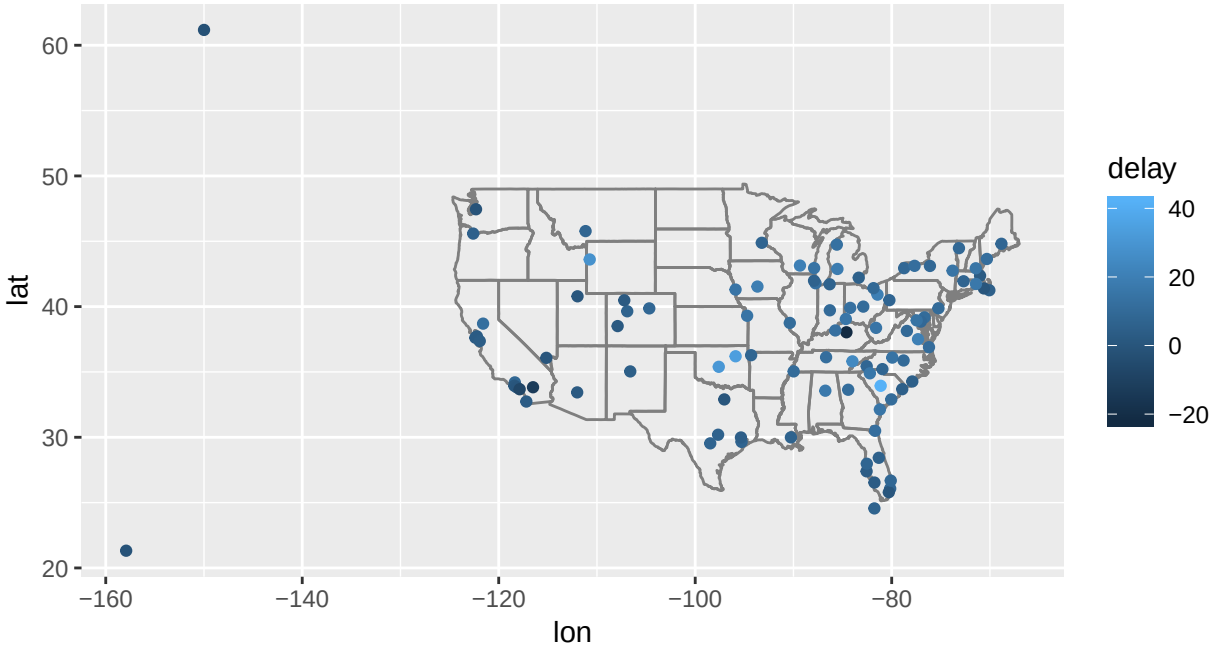
```
digraph subgraph_label {
  rankdir = TB
  subgraph cluster0{
    yearID_MN[label="yearID"]
    teamID_MN[label="teamID"]
    playerID_MN[label="playerID"]
    inseason_MN[label="inseason"]
    label = "Managers"
    {rank = same; yearID_MN; teamID_MN; inseason_MN; playerID_MN}
  }
  subgraph cluster1{
    playerID_M[label="playerID"]
    label = "Master"
  }
  subgraph cluster2{
    playerID_A[label="playerID"]
    awardID_A[label="awardID"]
    yearID_A[label="yearID"]
    lgID_A[label="lgID"]
    label = "AwardManagers"
    {rank = same; yearID; stint; playerID;}
  }
  playerID_M -> playerID_A
  playerID_M -> playerID_MN
}
```

13.4

1

Compute the average delay by destination, then join on the airports data frame so you can show the spatial distribution of delays. Here's an easy way to draw a map of the United States:

```
flights %>%
    group_by(dest) %>%
    summarise(delay = mean(arr_delay, na.rm = TRUE)) %>%
    inner_join(airports, by = c(dest = "faa")) %>%
    ggplot(aes(lon, lat, colour = delay)) +
    borders("state") +
    geom_point() +
    coord_quickmap()
```



2

Add the location of the origin and destination (i.e. the `lat` and `lon`) to `flights`.

```
flights %>% select(year:day, hour, origin, dest) %>%
    left_join(
          airports,
          by = c("origin" = "faa")
          ) %>%
    left_join(
          airports,
          by = c("dest" = "faa")
    ) %>%
    select(year:day, origin, dest, lat.x, lon.x, lat.y, lon.y)
```

```
## # A tibble: 336,776 x 9
##     year month   day origin dest  lat.x lon.x lat.y lon.y
##    <int> <int> <int> <chr>  <chr> <dbl> <dbl> <dbl> <dbl>
##  1  2013     1     1 EWR    IAH    40.7 -74.2  30.0 -95.3
```

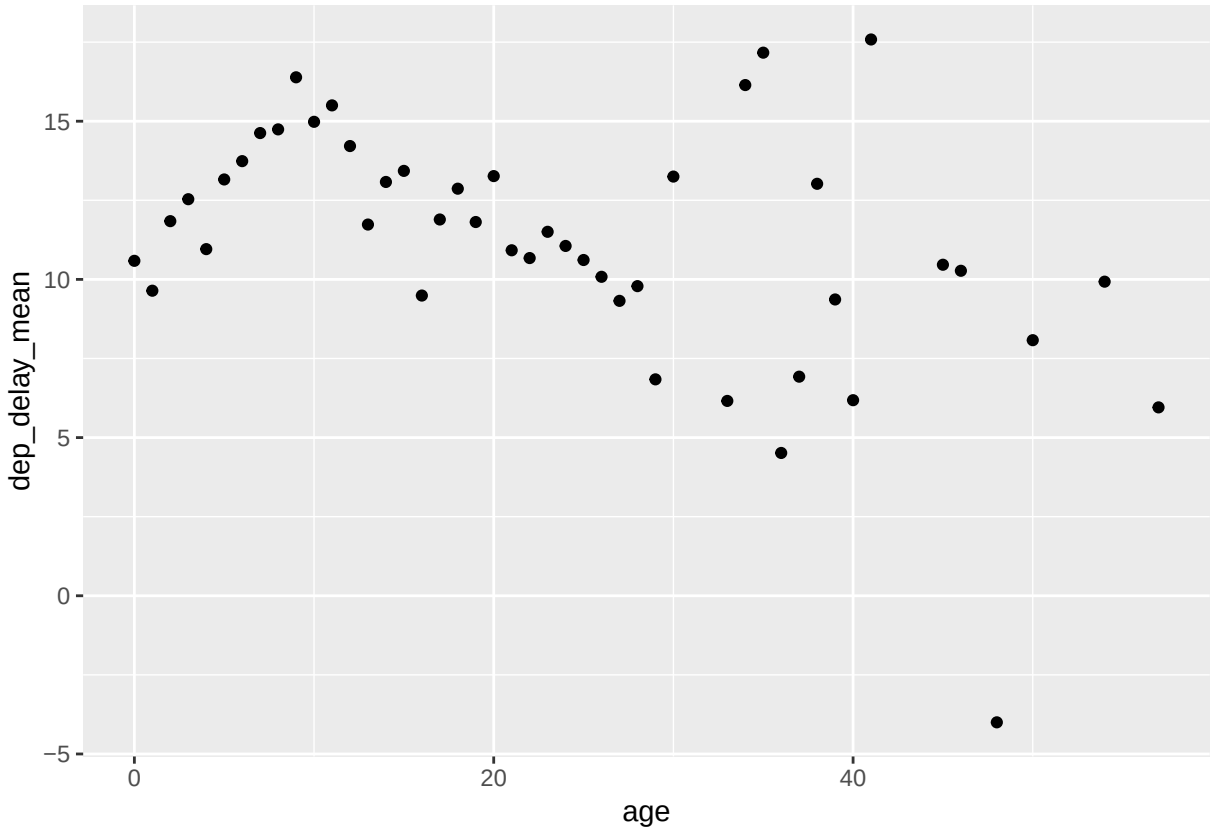
```
##  2  2013     1     1 LGA    IAH    40.8 -73.9  30.0 -95.3
##  3  2013     1     1 JFK    MIA    40.6 -73.8  25.8 -80.3
##  4  2013     1     1 JFK    BQN    40.6 -73.8  NA    NA
##  5  2013     1     1 LGA    ATL    40.8 -73.9  33.6 -84.4
##  6  2013     1     1 EWR    ORD    40.7 -74.2  42.0 -87.9
##  7  2013     1     1 EWR    FLL    40.7 -74.2  26.1 -80.2
##  8  2013     1     1 LGA    IAD    40.8 -73.9  38.9 -77.5
##  9  2013     1     1 JFK    MCO    40.6 -73.8  28.4 -81.3
## 10  2013     1     1 LGA    ORD    40.8 -73.9  42.0 -87.9
## # ... with 336,766 more rows
```

3

Is there a relationship between the age of a plane and its delays?

```
flights %>% inner_join(planes %>% select(tailnum, plane_year = year), by = "tailnum") %>%
    mutate(age = year - plane_year) %>%
    group_by(age) %>%
    summarise(
          dep_delay_mean = mean(dep_delay, na.rm = TRUE),
          arr_delay_mean = mean(arr_delay, na.rm = TRUE),
          n_arr_delay = sum(!is.na(arr_delay)),
          n_dep_delay = sum(!is.na(arr_delay))
          ) %>%
    ggplot(aes(x=age, y=dep_delay_mean)) +
    geom_point()
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



4

What weather conditions make it more likely to see a delay?

```
flights %>%
    inner_join(weather, by = c("origin" = "origin",
                    "year" = "year",
                    "month" = "month",
                    "day" = "day",
                    "hour" = "hour"
                    )
    )
```

```
## # A tibble: 335,220 x 29
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <dbl> <dbl> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
```

```
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 335,210 more rows, and 22 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour.x <dttm>, temp <dbl>, dewp <dbl>, humid <dbl>,
## #   wind_dir <dbl>, wind_speed <dbl>, wind_gust <dbl>, precip <dbl>,
## #   pressure <dbl>, visib <dbl>, time_hour.y <dttm>
```

GLM…?

5

> What happened on June 13, 2013? Display the spatial pattern of delays, and then use Google to cross-reference with the weather.

There were storms.

13.5.1

1

> What does it mean for a flight to have a missing `tailnum`? What do the tail numbers that don't have a matching record in planes have in common? (Hint: one variable explains ~90% of the problems.)

```
flights %>%
    anti_join(planes, by = "tailnum") %>%
    group_by(carrier) %>%
    summarize(count=n()) %>%
    select(carrier, count)
```

```
## # A tibble: 10 x 2
##    carrier count
##    <chr>   <int>
##  1 9E       1044
##  2 AA      22558
##  3 B6        830
##  4 DL        110
##  5 F9         50
##  6 FL        187
##  7 MQ      25397
##  8 UA       1693
##  9 US        699
## 10 WN         38
```

2

Filter flights to only show flights with planes that have flown at least 100 flights.

```
flights %>%
    group_by(tailnum) %>%
    count() %>%
    filter(n >= 100)
```

```
## # A tibble: 1,218 x 2
## # Groups:   tailnum [1,218]
##    tailnum     n
##    <chr>   <int>
##  1 <NA>     2512
##  2 N0EGMQ    371
##  3 N10156    153
##  4 N10575    289
##  5 N11106    129
##  6 N11107    148
##  7 N11109    148
##  8 N11113    138
##  9 N11119    148
## 10 N11121    154
## # ... with 1,208 more rows
```

3

Combine `fueleconomy::vehicles` and `fueleconomy::common` to find only the records for the most common models.

```
fueleconomy::vehicles %>%
    semi_join(fueleconomy::common, by = c("make", "model")) %>%
    distinct(model, make) %>%
    group_by(model) %>%
    mutate(count=n()) %>%
    filter(count > 1) %>%
    arrange(count)
```

```
## # A tibble: 8 x 3
## # Groups:   model [3]
##   model     make       count
##   <chr>     <chr>      <int>
## 1 Colt      Dodge          2
## 2 Colt      Plymouth       2
```

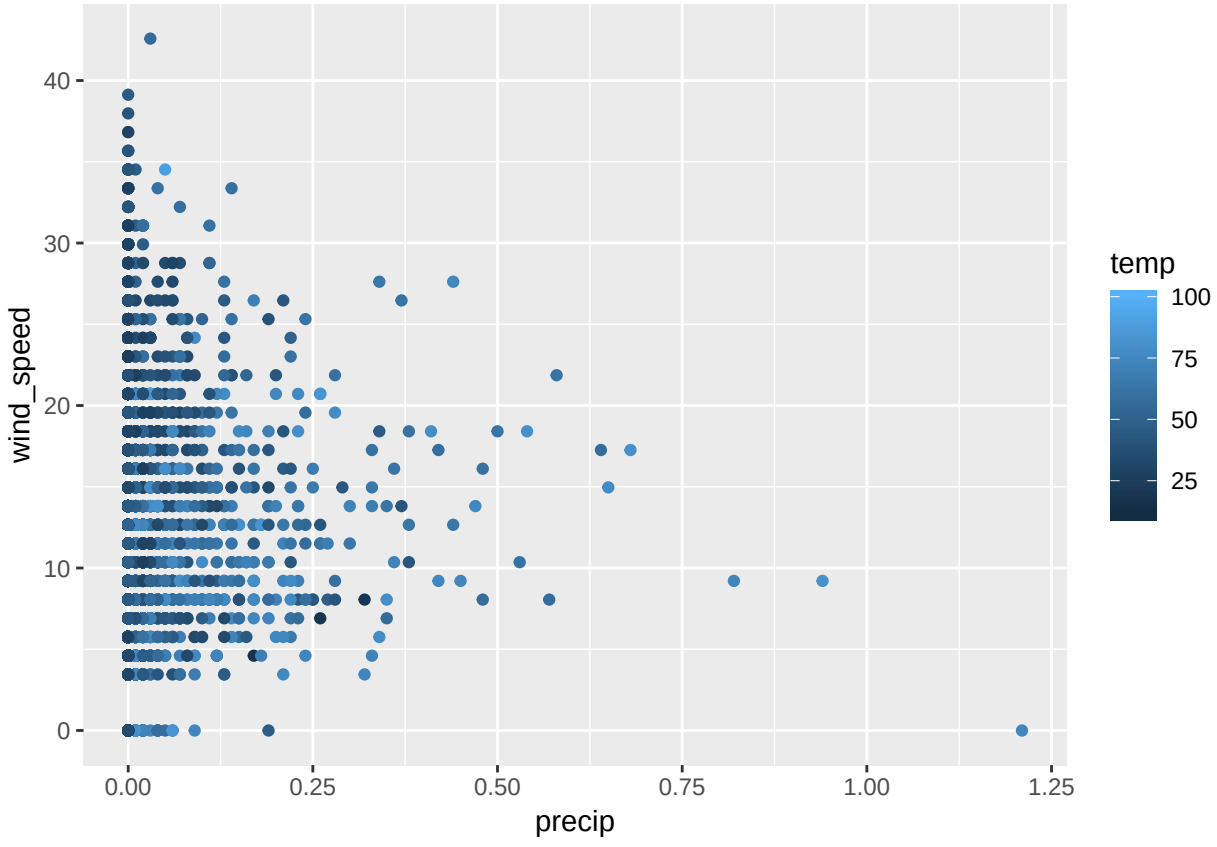
```
## 3 Truck 2WD Mitsubishi     3
## 4 Truck 4WD Mitsubishi     3
## 5 Truck 2WD Nissan         3
## 6 Truck 4WD Nissan         3
## 7 Truck 2WD Toyota         3
## 8 Truck 4WD Toyota         3
```

4

Find the 48 hours (over the course of the whole year) that have the worst delays. Cross-reference it with the weather data. Can you see any patterns?

```
flights %>%
    mutate(hour=sched_dep_time %/% 100) %>%
    group_by(origin, year, month, day, hour) %>%
    summarise(dep_delay = mean(dep_delay, na.rm = TRUE)) %>%
    arrange(desc(dep_delay)) %>%
    slice(1:48) %>%
    inner_join(weather, by=c("origin", "year", "month", "day", "hour")) %>%
    ggplot(aes(x = precip, y = wind_speed, color = temp)) +
    geom_point()
```

```
## Warning: Removed 4 rows containing missing values (geom_point).
```



5

What does `anti_join(flights, airports, by = c("dest" = "faa"))` tell you? What does `anti_join(airports, flights, by = c("faa" = "dest"))` tell you?