HW6_key

```r
library(tidyverse)
library(mdsr)
library(dbplyr)
library(DBI)
```

```r
# connect to the database which lives on a remote server maintain by
#   St. Olaf's IT department
library(RMariaDB)
con <- dbConnect(
  MariaDB(), host = "mdb.stolaf.edu",
  user = "ruser", password = "ruserpass",
  dbname = "flight_data"
)
```

On Your Own - Adapting 164 Code

These problems are based on class exercises from SDS 164, so you've already solved them in R! Now we're going to try to duplicate those solutions in SQL (but with 2023 data instead of 2013).

```r
# Read in 2013 NYC flights data
library(nycflights13)
flights_nyc13 <- nycflights13::flights
planes_nyc13 <- nycflights13::planes
```

1. Summarize carriers flying to MSP by number of flights and proportion that are cancelled (assuming that a missing arrival time indicates a cancelled flight). [This was #4 in 17_longer_pipelines.Rmd.]

```r
# Original solution from SDS 164
flights_nyc13 |>
  mutate(carrier = fct_collapse(carrier, "Delta +" = c("DL", "9E"),
                                       "American +"= c("AA", "MQ"),
                                      "United +" = c("EV", "OO", "UA"))) |>
  filter(dest == "MSP") |>
  group_by(origin, carrier) |>
  summarize(n_flights = n(),
            num_cancelled = sum(is.na(arr_time)),
            prop_cancelled = mean(is.na(arr_time)))
```

```
# A tibble: 5 x 5
# Groups:   origin [3]
  origin carrier    n_flights num_cancelled prop_cancelled
  <chr>  <fct>          <int>         <int>          <dbl>
1 EWR    Delta +          598            10         0.0167
2 EWR    United +        1779           105         0.0590
3 JFK    Delta +         1095            41         0.0374
4 LGA    Delta +         2420            25         0.0103
5 LGA    American +      1293            62         0.0480
```

First duplicate the output above, then check trends in 2023 across all origins. Here are a few hints:

- use flightdata instead of flights_nyc13
- remember that flights_nyc13 only contained 2013 and 3 NYC origin airports (EWR, JFK, LGA)
- is.na can be replaced with CASE WHEN ArrTime IS NULL THEN 1 ELSE 0 END or with CASE WHEN cancelled = 1 THEN 1 ELSE 0 END
- CASE WHEN can also be used replace fct_collapse

Duplicate 2013 NYC analysis for 2023:

```sql
SELECT Reporting_Airline,
  SUM(1) AS n_flights
FROM flightdata
WHERE year = 2023
GROUP BY Reporting_Airline
ORDER BY n_flights DESC;
```

Table 1: Displaying records 1 - 10

Reporting_Airline	n_flights
WN	1438465
DL	984986
AA	940531
UA	732212
OO	675163
YX	295275
B6	274852
NK	263871
AS	245344
MQ	227488

```sql
SELECT Reporting_Airline, dest, origin, Year,
  SUM(1) AS n_flights,
  SUM(cancelled) AS num_cancelled,
  AVG(cancelled) AS prop_cancelled,
  CASE WHEN (Reporting_Airline = "DL" OR Reporting_Airline = "9E") THEN 'Delta +'
    WHEN (Reporting_Airline = "AA" OR Reporting_Airline = "MQ") THEN 'American +'
    WHEN (Reporting_Airline = "EV" OR Reporting_Airline = "OO" OR Reporting_Airline = "UA") T
    ELSE 'Other' END AS new_carrier
FROM flightdata
WHERE dest = "MSP" AND year = 2023 AND (origin = "EWR" OR origin = "JFK" OR origin = "LGA")
GROUP BY origin, new_carrier
ORDER BY prop_cancelled DESC;
```

Table 2: 8 records

Reporting_Airline	dest	origin	Year	n_flights	num_cancelled	prop_cancelled	new_carrier
OO	MSP	JFK	2023	63	3	0.0476	United +
OO	MSP	EWR	2023	859	26	0.0303	United +
B6	MSP	JFK	2023	84	2	0.0238	Other
DL	MSP	LGA	2023	1729	35	0.0202	Delta +
9E	MSP	JFK	2023	1049	20	0.0191	Delta +
YX	MSP	LGA	2023	632	12	0.0190	Other
YX	MSP	EWR	2023	214	4	0.0187	Other
9E	MSP	EWR	2023	1308	23	0.0176	Delta +

See trends in 2023 across all origins (similar for other two problems - just remove origin from WHERE and re-run):

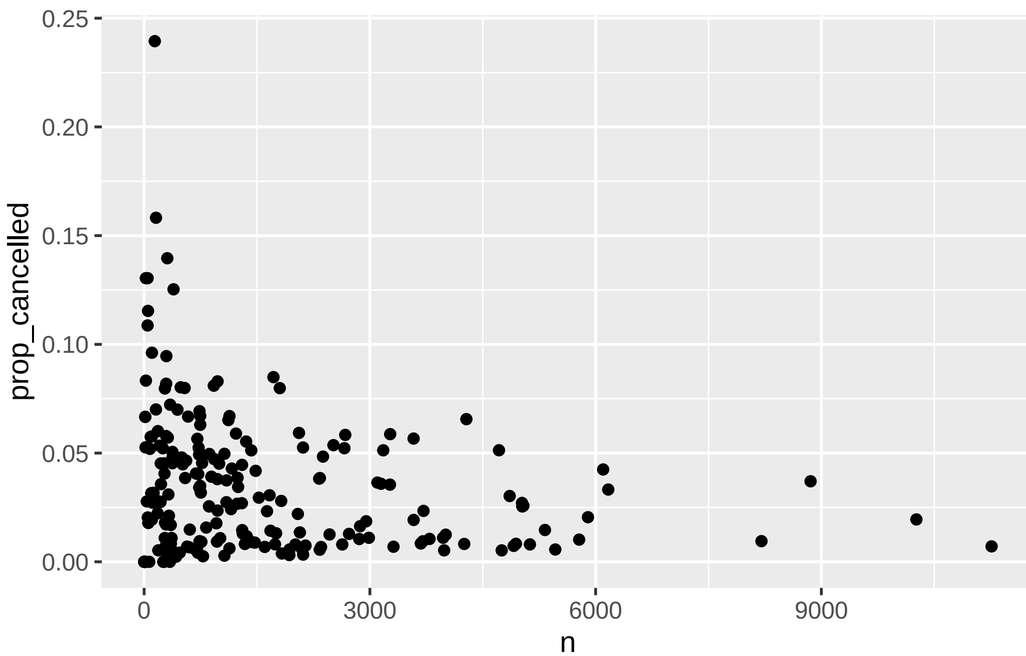
```
SELECT Reporting_Airline, dest, ArrTime, origin, Year,
  SUM(1) AS n_flights,
  SUM(cancelled) AS num_cancelled,
  AVG(cancelled) AS prop_cancelled,
  CASE WHEN (Reporting_Airline = "DL" OR Reporting_Airline = "9E") THEN 'Delta +'
    WHEN (Reporting_Airline = "AA" OR Reporting_Airline = "MQ") THEN 'American +'
    WHEN (Reporting_Airline = "EV" OR Reporting_Airline = "OO" OR Reporting_Airline = "UA") T
    ELSE 'Other' END AS new_carrier
FROM flightdata
WHERE dest = "MSP" AND year = 2023
GROUP BY origin, new_carrier
ORDER BY prop_cancelled DESC;
```

Table 3: Displaying records 1 - 10

Reporting_Airline	dest	ArrTime	origin	Year	n_flights	num_cancelled	prop_cancelled	new_carrier
G4	MSP	NA	PBI	2023	17	3	0.1765	Other
9E	MSP	1348	MOT	2023	8	1	0.1250	Delta +
OO	MSP	1039	BNA	2023	12	1	0.0833	United +
DL	MSP	1819	HDN	2023	17	1	0.0588	Delta +
OO	MSP	1940	STL	2023	21	1	0.0476	United +
OO	MSP	1754	JFK	2023	63	3	0.0476	United +
OO	MSP	1212	IND	2023	87	4	0.0460	United +
9E	MSP	612	GFK	2023	228	10	0.0439	Delta +
OO	MSP	1431	MKE	2023	115	5	0.0435	United +
9E	MSP	1845	RST	2023	404	16	0.0396	Delta +

2. Plot number of flights vs. proportion cancelled for every origin-destination pair (assuming that a missing arrival time indicates a cancelled flight). [This was #7 in 17_longer_pipelines.Rmd.]

```
# Original solution from SDS 164
flights_nyc13 |>
  group_by(origin, dest) |>
  summarize(n = n(),
            prop_cancelled = mean(is.na(arr_time))) |>
  filter(prop_cancelled < 1) |>
  ggplot(aes(n, prop_cancelled)) +
  geom_point()
```



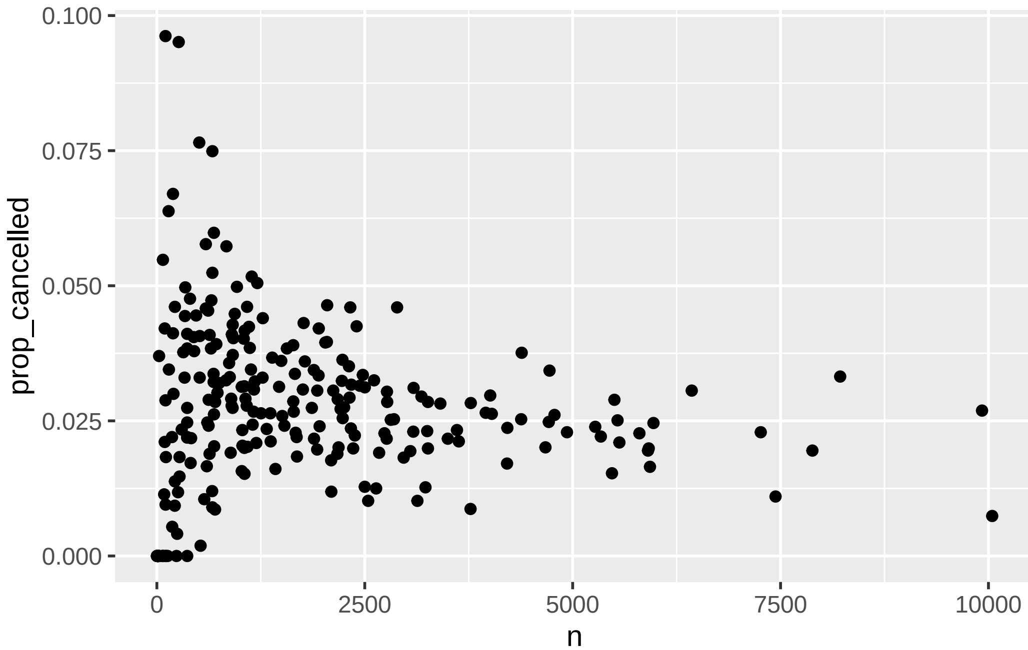
First duplicate the plot above for 2023 data, then check trends across all origins. Do all of the data wrangling in SQL. Here are a few hints:

- use flightdata instead of flights_nyc13
- remember that flights_nyc13 only contained 2013 and 3 NYC origin airports (EWR, JFK, LGA)
- use an `sql` chunk and an `r` chunk
- include `connection =` and `output.var =` in your sql chunk header (this doesn't seem to work with dbGetQuery()…)

Duplicate 2013 NYC analysis for 2023:

```sql
SELECT origin, dest,
  SUM(1) AS n,
  AVG(cancelled) AS prop_cancelled
FROM flightdata
WHERE year = 2023 AND (origin = "EWR" OR origin = "JFK" OR origin = "LGA")
GROUP BY origin, dest
HAVING prop_cancelled < 1
```

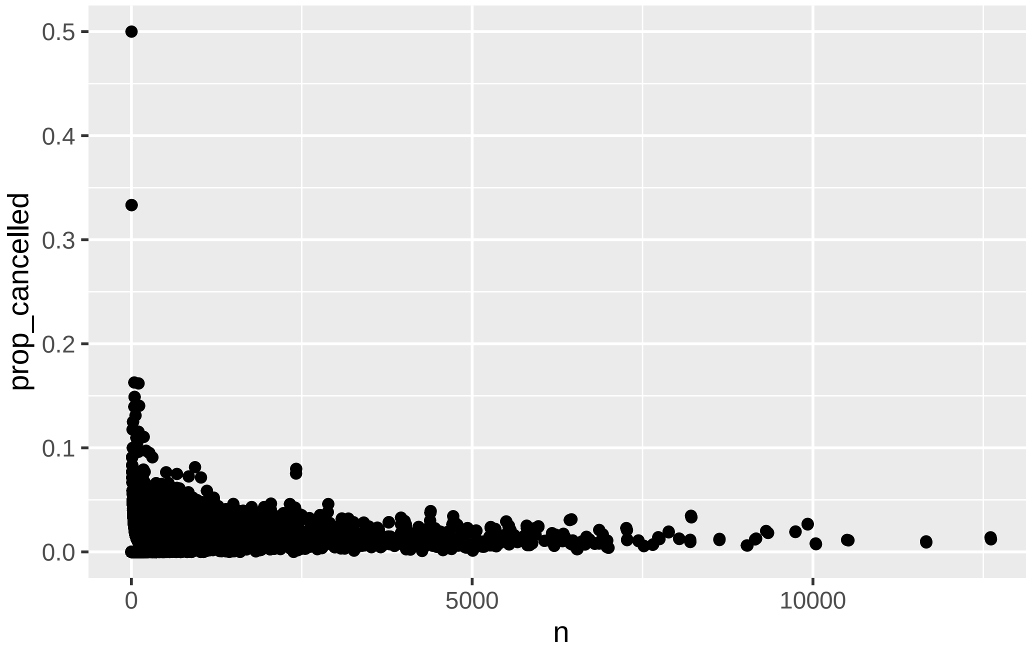
```r
plot_data |>
  ggplot(aes(n, prop_cancelled)) +
  geom_point()
```



See trends in 2023 across all origins:

```
SELECT origin, dest,
  SUM(1) AS n,
  AVG(cancelled) AS prop_cancelled
FROM flightdata
WHERE year = 2023
GROUP BY origin, dest
HAVING prop_cancelled < 1
```

```
plot_data2 |>
  ggplot(aes(n, prop_cancelled)) +
  geom_point()
```



3. Produce a table of weighted plane age by carrier, where weights are based on number of flights per plane. [This was #6 in 26_more_joins.Rmd.]

```
# Original solution from SDS 164
flights_nyc13 |>
  left_join(planes_nyc13, join_by(tailnum)) |>
  mutate(plane_age = 2013 - year.y) |>
  group_by(carrier) |>
  summarize(unique_planes = n_distinct(tailnum),
            mean_weighted_age = mean(plane_age, na.rm =TRUE),
            sd_weighted_age = sd(plane_age, na.rm =TRUE)) |>
  arrange(mean_weighted_age)
```

```
# A tibble: 16 x 4
   carrier unique_planes mean_weighted_age sd_weighted_age
   <chr>           <int>             <dbl>           <dbl>
 1 HA                 14              1.55            1.14
 2 AS                 84              3.34            3.07
 3 VX                 53              4.47            2.14
 4 F9                 26              4.88            3.67
 5 B6                193              6.69            3.29
 6 OO                 28              6.84            2.41
 7 9E                204              7.10            2.67
```

```
 8 US                290              9.10            4.88
 9 WN                583              9.15            4.63
10 YV                 58              9.31            1.93
11 EV                316             11.3             2.29
12 FL                129             11.4             2.16
13 UA                621             13.2             5.83
14 DL                629             16.4             5.49
15 AA                601             25.9             5.42
16 MQ                238             35.3             3.13
```

First duplicate the output above for 2023, then check trends across all origins. Do all of the data wrangling in SQL. Here are a few hints:

- use flightdata instead of flights_nyc13
- remember that flights_nyc13 only contained 2013 and 3 NYC origin airports (EWR, JFK, LGA)
- you'll have to merge the flights dataset with the planes dataset
- you can use DISTINCT inside a COUNT()
- investigate SQL clauses for calculating a standard deviation
- you cannot use a derived variable inside a summary clause in SELECT

For bonus points, also merge the airlines dataset and include the name of each carrier and not just the abbreviation!

Duplicate 2013 NYC analysis for 2023:

```sql
SELECT Reporting_Airline AS carrier,
  a.name AS carrier_name,
  COUNT(DISTINCT o.TAIL_NUMBER) AS unique_planes,
  AVG(o.year - p.year) AS mean_weighted_age,
  STDDEV_SAMP(o.year - p.year) AS sd_weighted_age
FROM flightdata AS o
LEFT JOIN planes p ON o.TAIL_NUMBER = p.tailnum
LEFT JOIN airlines a ON o.Reporting_Airline = a.carrier
WHERE o.year = 2023 AND origin IN ("EWR", "JFK", "LGA")
GROUP BY carrier_name
ORDER BY mean_weighted_age ASC
```

```
test
```

```
   carrier            carrier_name unique_planes mean_weighted_age
1       G4           Allegiant Air            94                NA
2       F9 Frontier Airlines Inc.           136          4.135947
```

```
3       OO  SkyWest Airlines Inc.           188           5.850972
4       AS   Alaska Airlines Inc.           215           6.404048
5       NK       Spirit Air Lines           218           6.738930
6       HA Hawaiian Airlines Inc.            24           9.728571
7       MQ              Envoy Air           102          10.690840
8       WN Southwest Airlines Co.           841          10.848831
9       YX       Republic Airline           229          11.993201
10      9E      Endeavor Air Inc.           127          12.218725
11      AA American Airlines Inc.           890          12.943699
12      DL   Delta Air Lines Inc.           844          12.966209
13      B6         JetBlue Airways          296          14.032734
14      UA  United Air Lines Inc.           920          16.083366
   sd_weighted_age
1               NA
2         2.365734
3         4.212619
4         4.900640
5         3.613592
6         1.938314
7         5.861740
8         6.883556
9         4.826594
10        3.968461
11        6.999990
12       10.325917
13        5.634976
14        9.824684
```

See trends in 2023 across all origins:

```sql
SELECT Reporting_Airline AS carrier,
  a.name AS carrier_name,
  COUNT(DISTINCT o.TAIL_NUMBER) AS unique_planes,
  AVG(o.year - p.year) AS mean_weighted_age,
  STDDEV_SAMP(o.year - p.year) AS sd_weighted_age
FROM flightdata AS o
LEFT JOIN planes p ON o.TAIL_NUMBER = p.tailnum
LEFT JOIN airlines a ON o.Reporting_Airline = a.carrier
WHERE o.year = 2023
GROUP BY carrier_name
ORDER BY mean_weighted_age ASC
```

```
test2
```

```
   carrier           carrier_name unique_planes mean_weighted_age
1       G4          Allegiant Air           131                NA
2       F9 Frontier Airlines Inc.           140          4.205092
3       NK       Spirit Air Lines           222          6.045380
4       MQ              Envoy Air           170          7.378282
5       OH      PSA Airlines Inc.           125         10.137503
6       AS   Alaska Airlines Inc.           250         10.334035
7       OO  SkyWest Airlines Inc.           501         10.996546
8       WN Southwest Airlines Co.           855         11.184157
9       YX       Republic Airline           229         12.383440
10      9E      Endeavor Air Inc.           154         13.032144
11      AA American Airlines Inc.           954         13.156226
12      B6        JetBlue Airways           297         13.479612
13      DL   Delta Air Lines Inc.           948         14.832476
14      UA  United Air Lines Inc.           948         16.072381
15      HA Hawaiian Airlines Inc.            60         17.720354
   sd_weighted_age
1               NA
2         2.405261
3         4.272343
4         4.457980
5         5.175366
6         7.160113
7         6.982704
8         7.093490
9         4.554182
10        4.052969
11        7.259300
12        6.094246
13        9.965276
14        9.020514
15        6.682312
```