

## Ramki danych cz. 1

- Wszystkie zadania rozwiążemy bez używania pętli.
- Zbiory danych można pobrać ze strony <http://gagolewski.com/resources/>.
- W zadaniach 6.2 – 6.6 przeanalizuj podane polecenia SQL (por. `?sqldf::sqldf`), a następnie odtwórz wynik korzystając z bazowych funkcji R-owych.

Consider the `nycflights13` database.

```
install.packages("nycflights13")  
library("nycflights13")
```

The database contains:

- `flights` – information about all flights that departed from NYC (e.g., EWR, JFK, and LGA) in 2013
- `airports` – airport names and locations
- `planes` – construction information about each plane
- `airlines` – translation between two letter carrier codes and names
- `weather` – hourly meteorological data for each airport

### Zadanie 6.1.

Break the ice: `?datasetname` (e.g. `?planes`), `nrow()`, `ncol()`, `head()`, `tail()`.

### Zadanie 6.2.

Fetch the results corresponding to the following SQL commands. Output a data frame in each case.

- `SELECT DISTINCT engine FROM planes`
- `SELECT DISTINCT type, manufacturer FROM planes`
- `SELECT COUNT(*), engine FROM planes GROUP BY engine`
- `SELECT COUNT(*), engine, type FROM planes  
GROUP BY engine, type`
- `SELECT MIN(year), AVG(year), MAX(year), engine, manufacturer FROM planes  
GROUP BY engine, manufacturer`

### Zadanie 6.3.

Fetch the results corresponding to the following SQL commands:

- `SELECT * FROM planes WHERE speed IS NOT NULL`
- `SELECT tailnum FROM planes WHERE seats BETWEEN 150 AND 190 AND year >= 2012`
- `SELECT * FROM planes WHERE manufacturer IN ("BOEING", "AIRBUS", "EMBRAER")  
AND seats>390`
- `SELECT DISTINCT year, seats FROM planes WHERE year >= 2012  
ORDER BY year ASC, seats DESC`

- (e) `SELECT DISTINCT year, seats FROM planes WHERE year >= 2012  
ORDER BY seats DESC, year ASC`

**Zadanie 6.4.**

Fetch the results corresponding to the following SQL commands:

- (a) `SELECT manufacturer, COUNT(*) FROM planes WHERE seats > 200  
GROUP BY manufacturer`
- (b) `SELECT manufacturer, COUNT(*) FROM planes  
GROUP BY manufacturer HAVING COUNT(*) > 10`
- (c) `SELECT manufacturer, COUNT(*) FROM planes WHERE seats > 200  
GROUP BY manufacturer HAVING COUNT(*) > 10`
- (d) `SELECT manufacturer, COUNT(*) AS howmany FROM planes  
GROUP BY manufacturer ORDER BY howmany DESC LIMIT 5`

**Zadanie 6.5.**

Fetch the results corresponding to the following SQL commands:

- (a) `SELECT * FROM flights LEFT JOIN planes ON flights.tailnum=planes.tailnum`
- (b) `SELECT planes.*, airlines.* FROM  
(SELECT DISTINCT carrier, tailnum FROM flights) AS cartail  
INNER JOIN planes ON cartail.tailnum=planes.tailnum  
INNER JOIN airlines ON cartail.carrier=airlines.carrier`
- (c) `SELECT flights2.*, weather2.atemp, weather2.ahumid, weather2.apressure FROM  
(SELECT * FROM flights WHERE origin='EWR') AS flights2  
LEFT JOIN  
(SELECT year, month, day, AVG(temp) AS atemp,  
AVG(humid) AS ahumid, AVG(pressure) AS apressure  
FROM weather WHERE origin='EWR' GROUP BY year, month, day) AS weather2  
ON flights2.year=weather2.year  
AND flights2.month=weather2.month  
AND flights2.day=weather2.day`

**Zadanie 6.6.**

Let A – rows 1,...,10 from airports. B – rows 6,...,15.

- (a) `SELECT * FROM A UNION SELECT * FROM B`
- (b) `SELECT * FROM A UNION ALL SELECT * FROM B`
- (c) `SELECT * FROM A INTERSECT SELECT * FROM B`
- (d) `SELECT * FROM A EXCEPT SELECT * FROM B`
- (e) `SELECT * FROM B EXCEPT SELECT * FROM A`