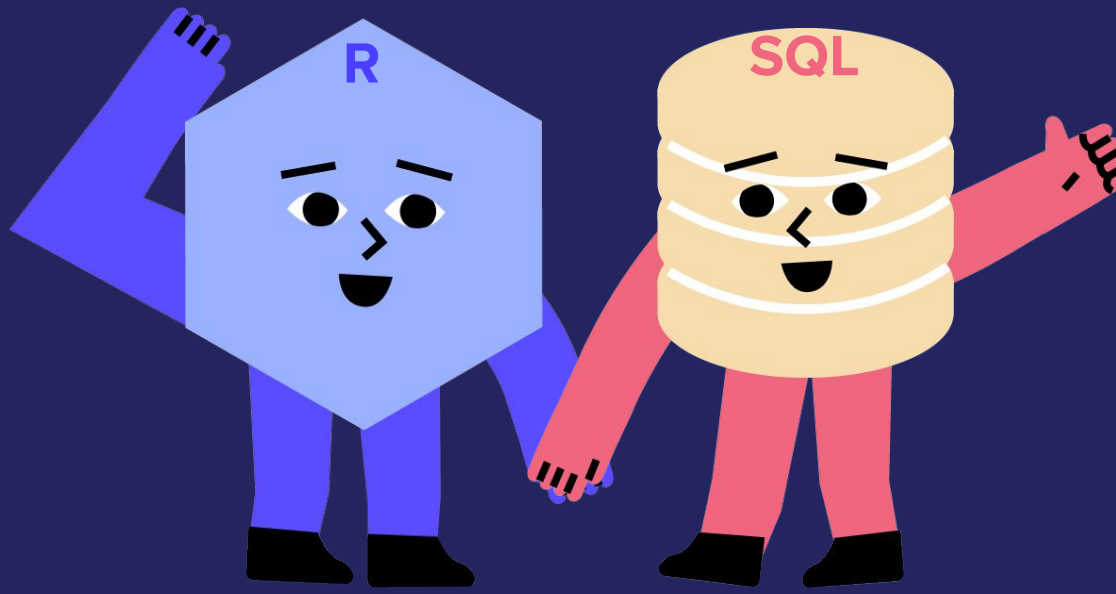
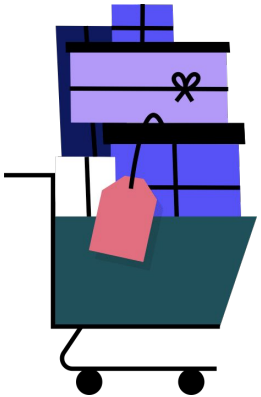
riskified technology;

The Dynamic duo



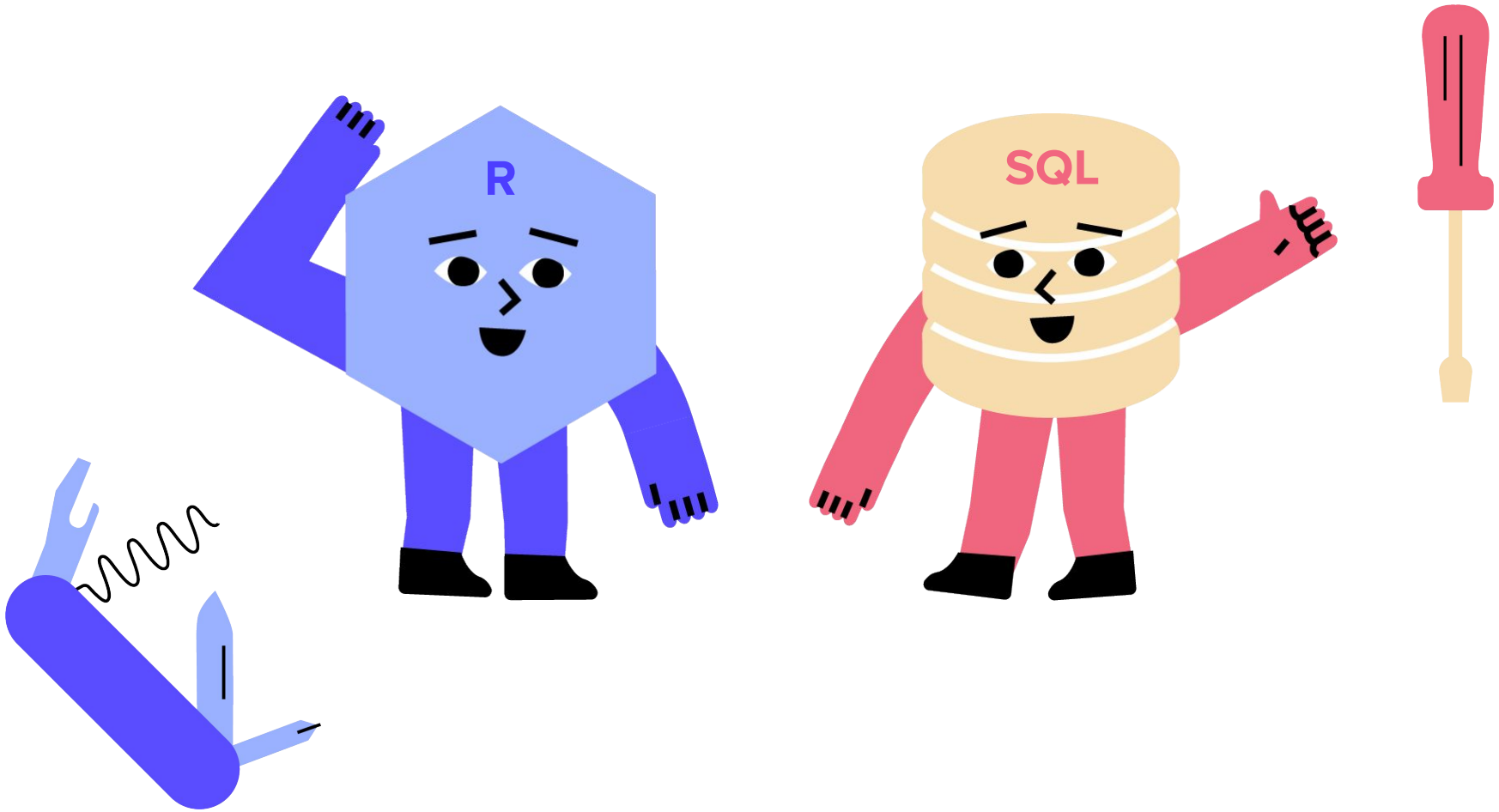
Irene Steves @i_steves https://rstd.io/global2021/irenesteves



Millions of orders a day

ETLs process incoming data and store in the data warehouse

Query data for analyses in R!


R
SQL

R the multitool

Why bother to learn SQL
if I can do it through R?

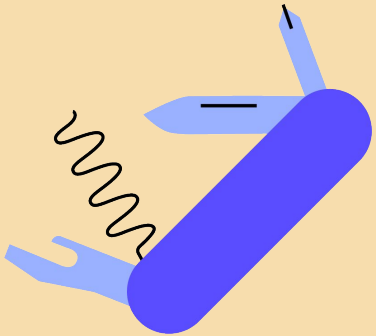
dbplyr

Extension of dplyr that runs on databases

Translates tidyverse to SQL for you


```r
mtcars_db %>%
  mutate(mpg_rnd = round(mpg)) %>%
  select(mpg_rnd, hp) %>%
  show_query()
```


```sql
SELECT ROUND(`mpg`, 0) AS `mpg_rnd`,
       `hp`
FROM `mtcars`
```

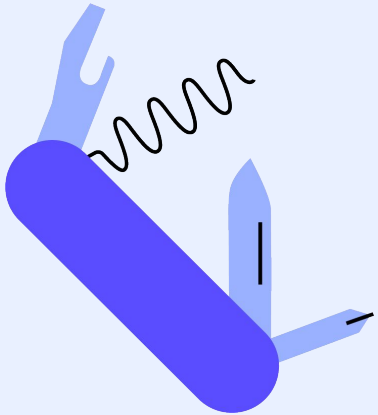



```r
mtcars_db %>%
  mutate(mpg_rnd = round(mpg)) %>%
  select(mpg_rnd, hp)
```


```sql
SELECT ROUND(mpg, 0) AS mpg_rnd,
       hp
FROM mtcars
```

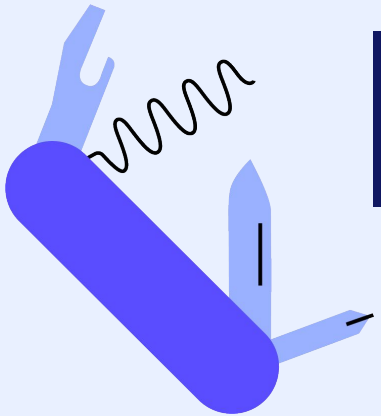
Query translations are often almost indistinguishable from hand-written queries


```r
mtcars_db %>%
  select(-hp) %>%
  show_query()
```




```sql
SELECT `mpg`, `cyl`, `disp`,
       `drat`, `wt`, `qsec`,
       `vs`, `am`, `gear`,
       `carb`
FROM `mtcars`
```

```
mtcars_db %>%
  summarize_all(max) %>%
  show_query()
```



dbplyr can save you from tedious typing

```
SELECT  MAX(`mpg`) AS `mpg`,
        MAX(`cyl`) AS `cyl`,
        MAX(`disp`)AS `disp`,
        MAX(`hp`)  AS `hp`,
        MAX(`drat`)AS `drat`,
        MAX(`wt`)  AS `wt`,
        MAX(`qsec`)AS `qsec`,
        MAX(`vs`)  AS `vs`,
        MAX(`am`)  AS `am`,
        MAX(`gear`)AS `gear`,
        MAX(`carb`)AS `carb`
FROM `mtcars`
```


```
mtcars_db %>%
  summarize_all(~sum(is.na(.x))))
```

Error: no such column: .x

```
mtcars_db %>%
  mutate_all(is.na) %>%
  summarize_all(sum)
```


```
mtcars %>%
  summarize_all(~sum(is.na(.x))))
```

Not all “native dplyr” can be translated


```
mtcars_db %>%
  summarize_all(~sum(is.na(.x))))
```

Error: no such column: .x

```
mtcars_db %>%
  mutate_all(is.na) %>%
  mutate_all(as.integer) %>%
  summarize_all(sum)
```

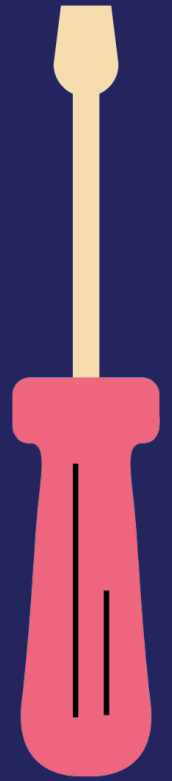
Not all “native dplyr” can be translated


```
mtcars %>%
  summarize_all(~sum(is.na(.x))))
```

dbplyr’s behavior is sometimes database-specific

SQL the screwdriver

Even screwdrivers have their moments



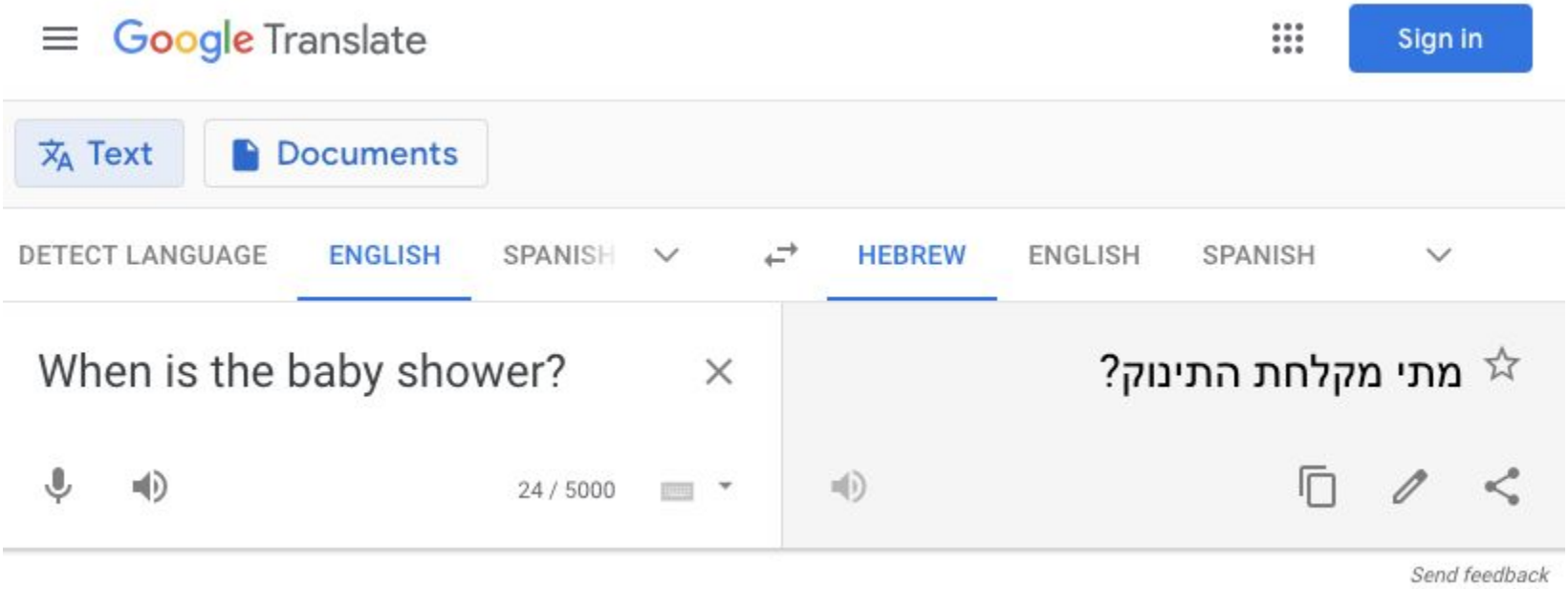

```r
mtcars_db %>%
  mutate(mpg_rnd = round(mpg)) %>%
  group_by(mpg_rnd) %>%
  summarize(hp_avg = mean(hp))
```


```sql
SELECT ROUND(mpg, 0) AS mpg_rnd,
       AVG(hp)   AS hp_avg
FROM mtcars
GROUP BY 1
```

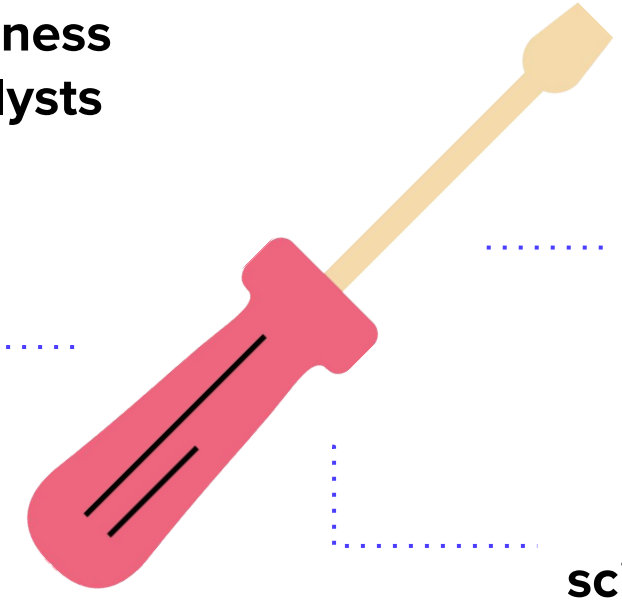

```r
mtcars_db %>%
  mutate(mpg_rnd = round(mpg)) %>%
  group_by(mpg_rnd) %>%
  summarize(hp_avg = mean(hp)) %>%
  show_query()
```

Need to dig into SQL to optimize for readability/speed

```sql
SELECT `mpg_rnd`,
       AVG(hp) AS hp_avg
FROM (
    SELECT `mpg`, `cyl`,
      `disp`, `hp`, `drat`,
      `wt`, `qsec`, `vs`,
      `am`, `gear`, `carb`,
      ROUND(`mpg`, 0) AS `mpg_rnd`
    FROM `mtcars`
    )
GROUP BY `mpg_rnd`
```


Google Translate
Sign in
Text
Documents
DETECT LANGUAGE
ENGLISH
SPANISH
HEBREW
ENGLISH
SPANISH
When is the baby shower?
24 / 5000
מתי מקלחת התינוק?
Send feedback

Universal tool



Business analysts

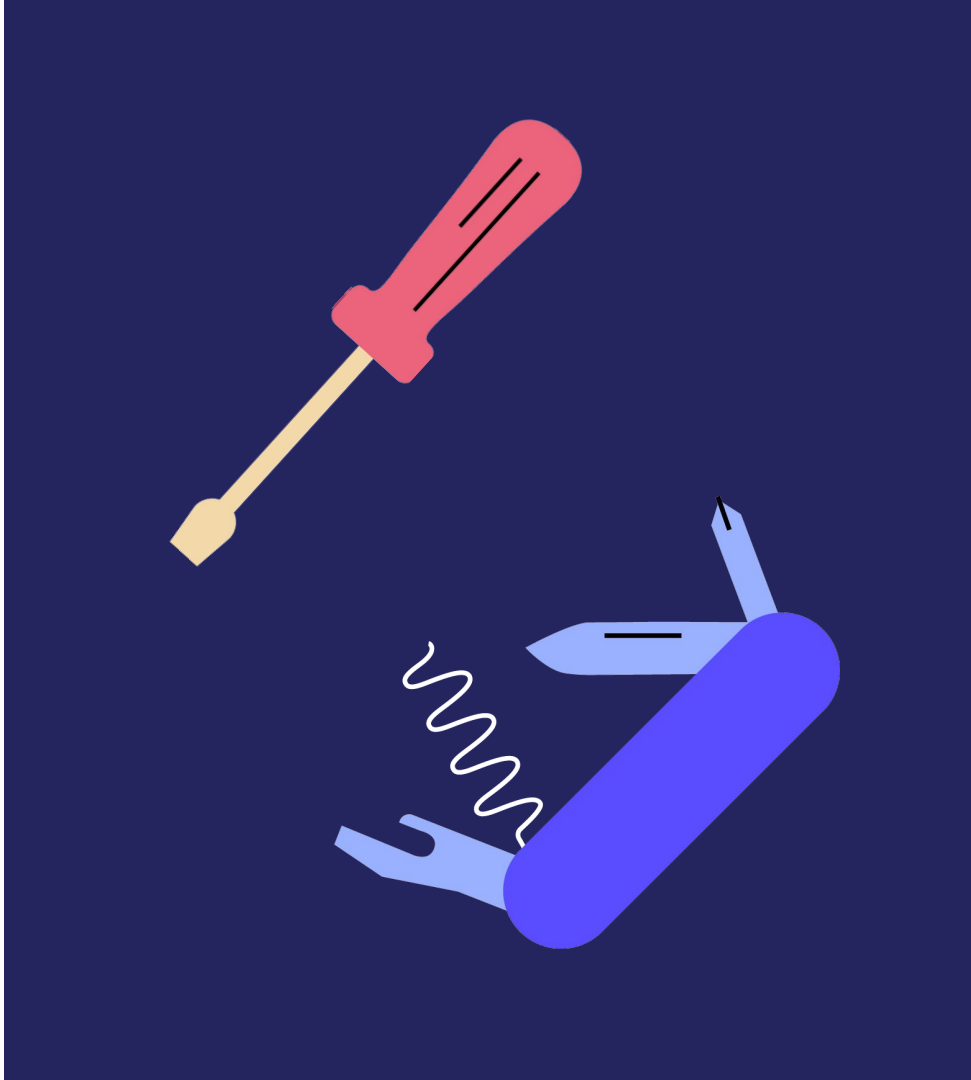
Software developers

Database administrators

Data scientists

R & SQL together in the same toolkit

Easy to become bilingual



`select()`	`SELECT`
`filter()`	`WHERE`
`filter()`	`HAVING`
`group_by()`	`GROUP BY`
`arrange()`	`ORDER BY`
`%>%`	`FROM`

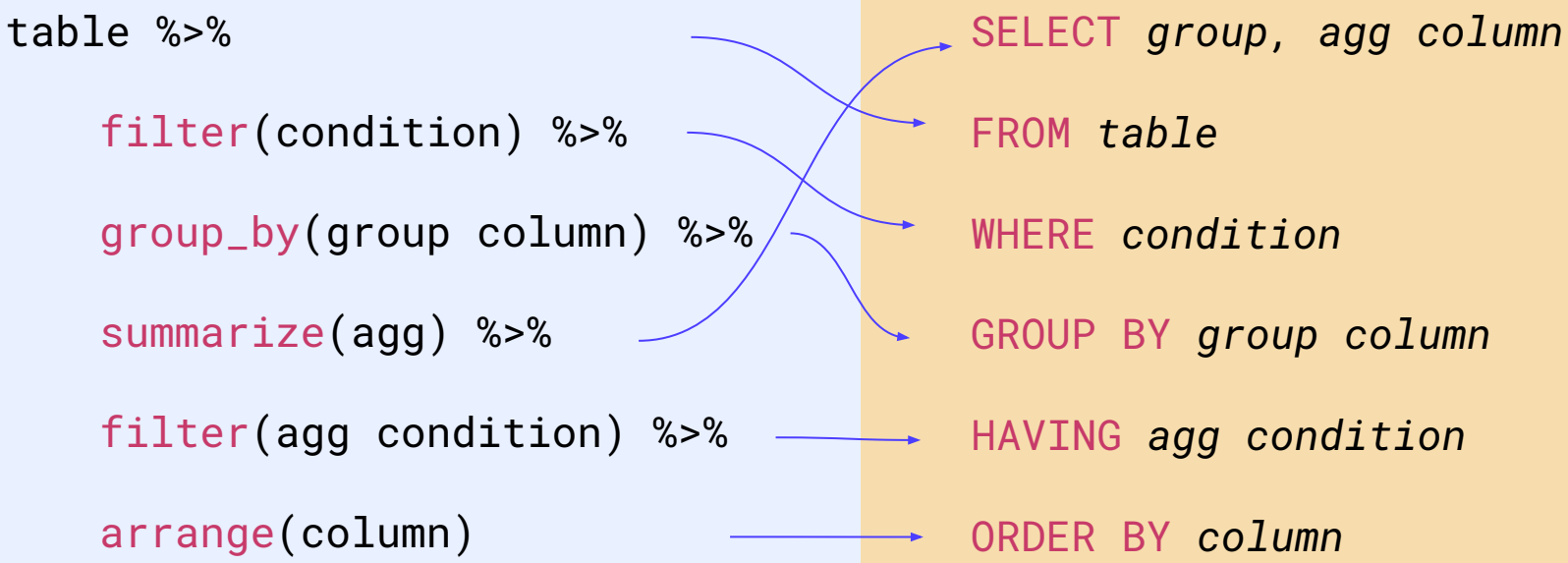
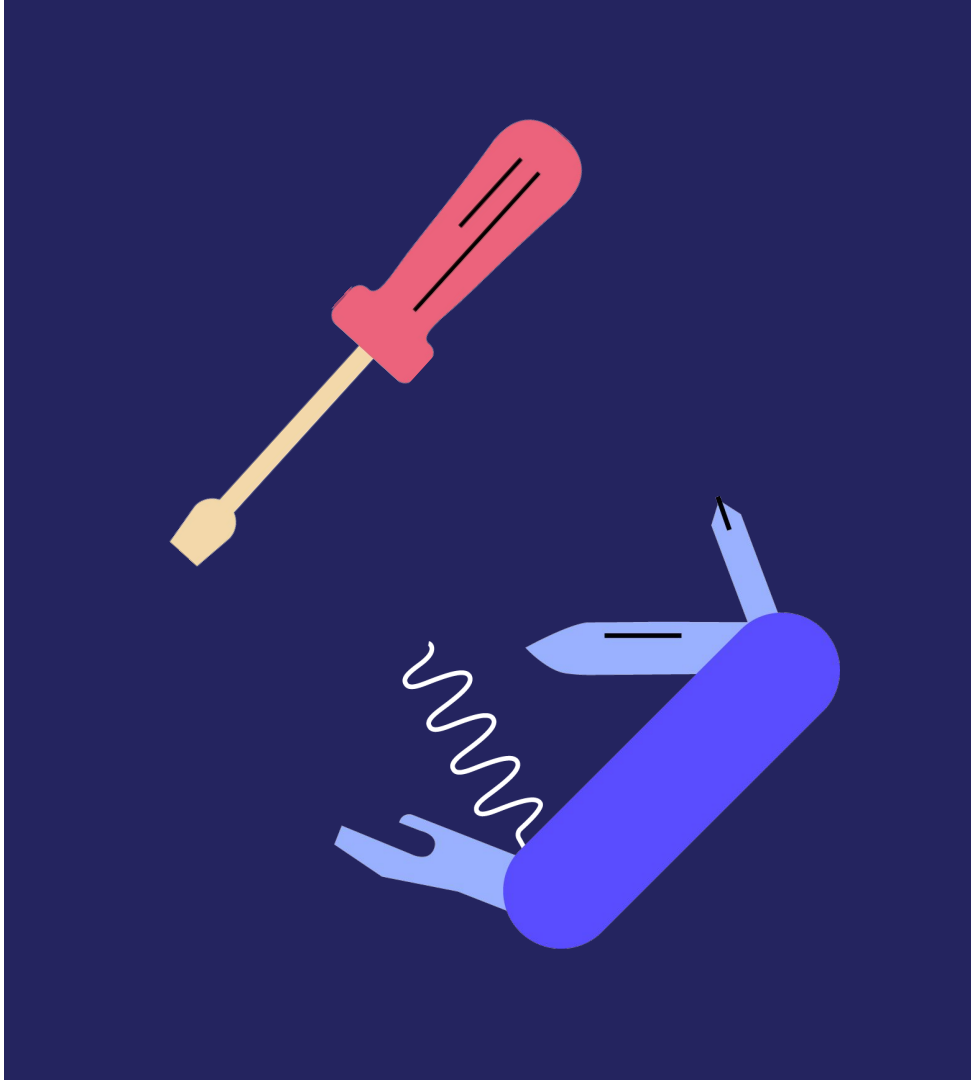
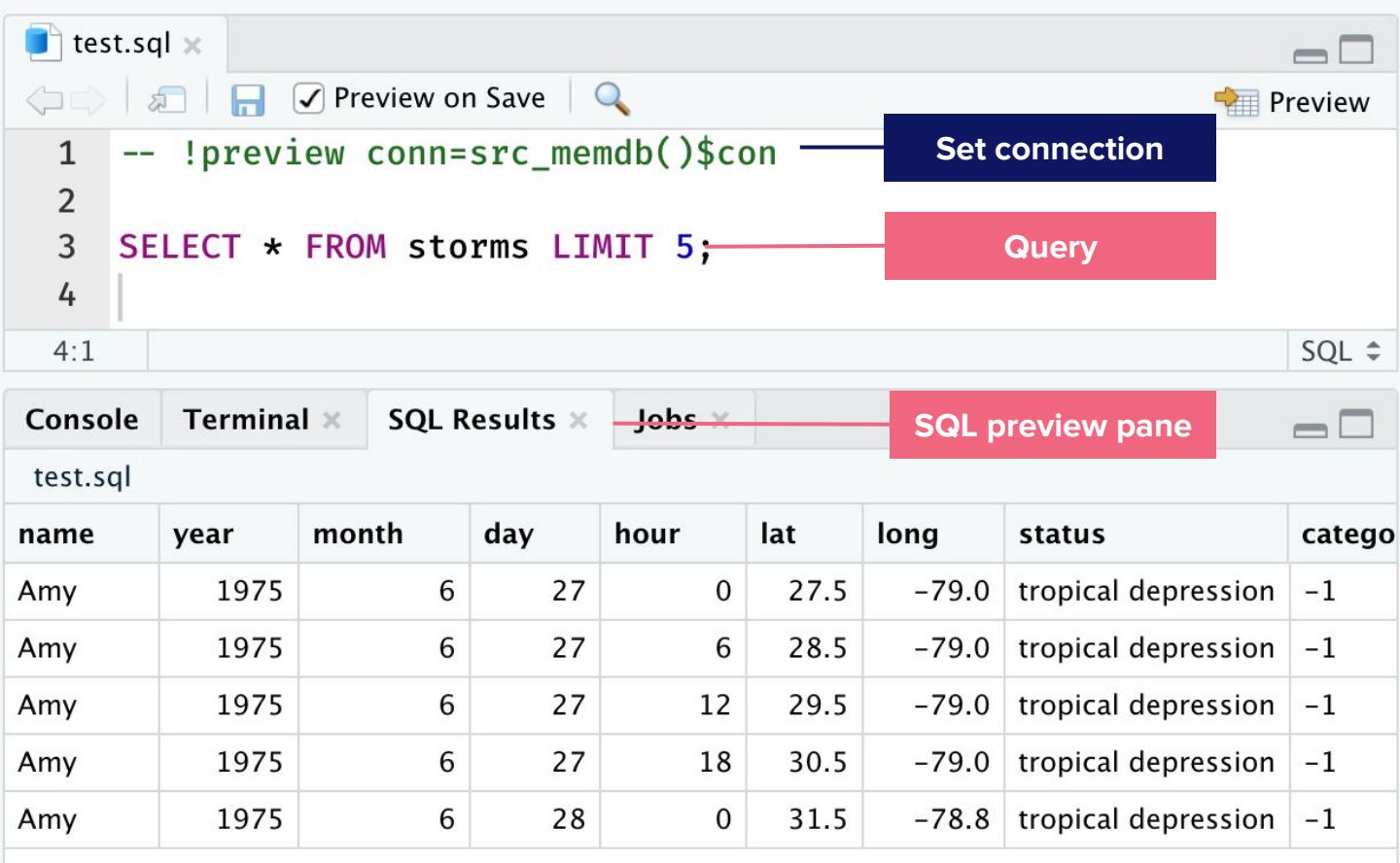


table %>%
filter(condition) %>%
group_by(group column) %>%
summarize(agg) %>%
filter(agg condition) %>%
arrange(column)
SELECT group, agg column
FROM table
WHERE condition
GROUP BY group column
HAVING agg condition
ORDER BY column

Familiar surroundings



SQL previews



name	year	month	day	hour	lat	long	status	catego
Amy	1975	6	27	0	27.5	-79.0	tropical depression	-1
Amy	1975	6	27	6	28.5	-79.0	tropical depression	-1
Amy	1975	6	27	12	29.5	-79.0	tropical depression	-1
Amy	1975	6	27	18	30.5	-79.0	tropical depression	-1
Amy	1975	6	28	0	31.5	-78.8	tropical depression	-1

SQL in RMarkdown

YAML header

R set-up chunk

SQL query

storms_db.Rmd

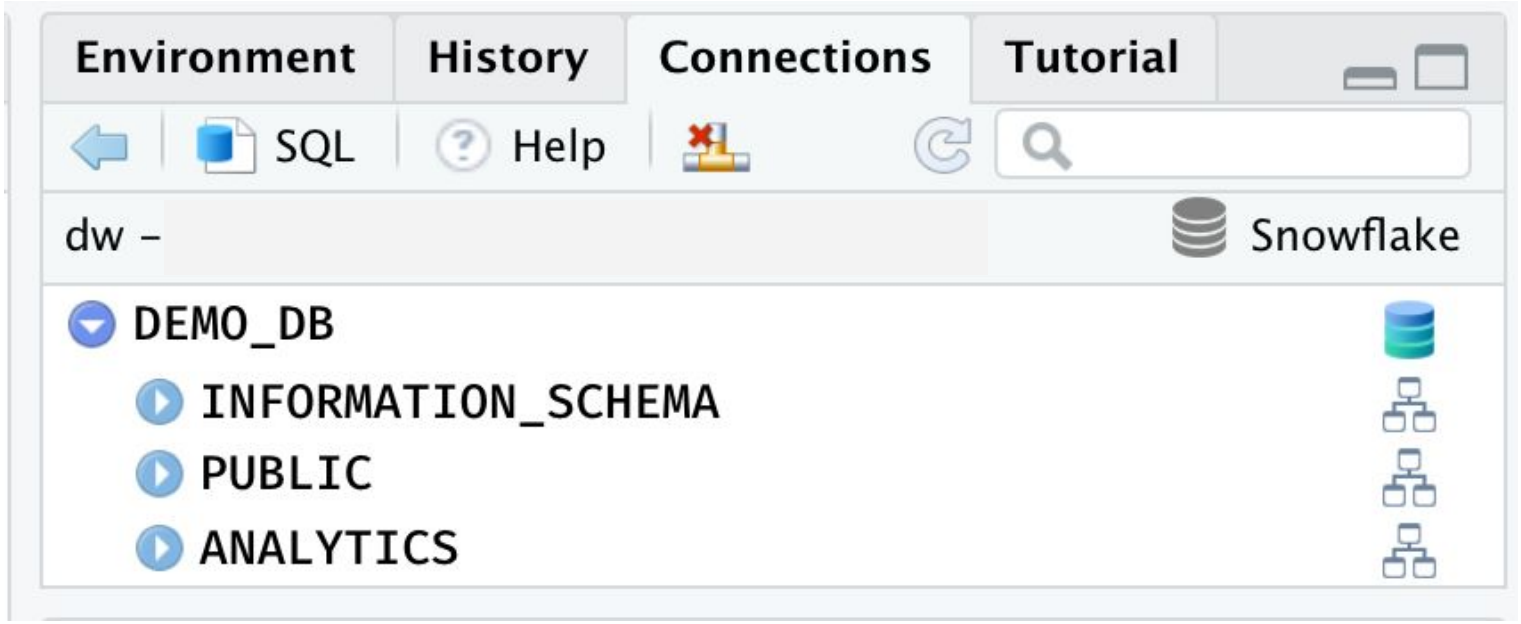
Knit | Insert | Run

```
1  ---
2  title: "Mixed document"
3  output: html_document
4  editor_options:
5    chunk_output_type: inline
6  ---
7
8  ```{r setup, include=FALSE}
9  library(dplyr)
10 library(dbplyr)
11
12 conn ← src_memdb()$con
13 storms_db ← tbl_memdb(storms)
14 ```
15
16 ```{sql, connection = conn}
17 SELECT * FROM storms LIMIT 5;
18 ```
```

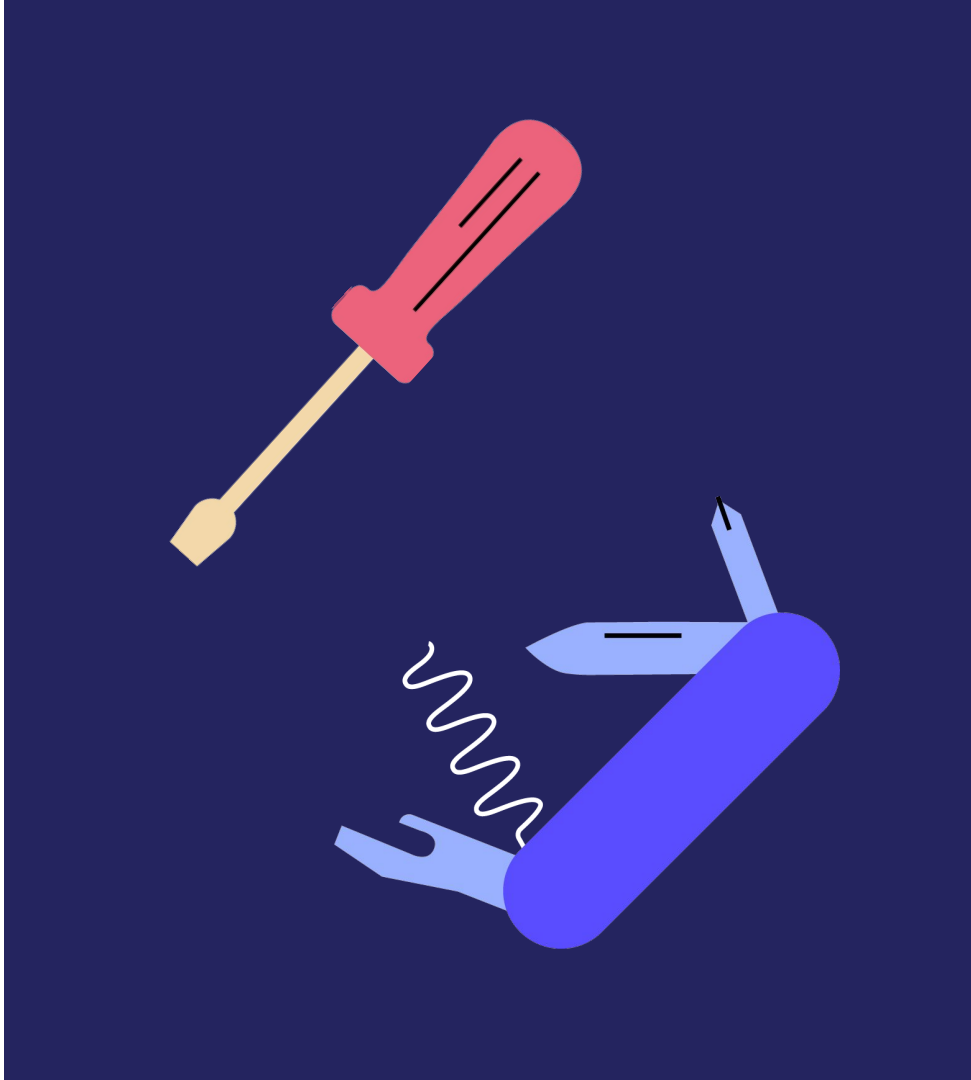
Enables inline query preview

name <chr>	year <dbl>	month <dbl>	day <int>	hour <dbl>	lat <dbl>	long <dbl>
Amy	1975	6	27	0	27.5	-79.0
Amy	1975	6	27	6	28.5	-79.0
Amy	1975	6	27	12	29.5	-79.0
Amy	1975	6	27	18	30.5	-79.0
Amy	1975	6	28	0	31.5	-78.8

Database navigation



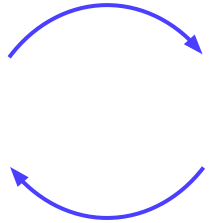
Day in the life



Workflow

Dedicated SQL IDE

- Auto-formatting + syntax highlighting for SQL
- Code/field suggestions
- Easy to stop bad queries



RStudio

- R formatting, suggestions, highlighting
- Visualizations, reporting
- Project organization + version control

Project structure

SQL files - read in relevant data & do initial data processing

R files - additional data wrangling + statistics/visualizations

RMarkdown report - results & conclusions to share with stakeholders

riskiconn

Internal package for handling database connections/queries

Configurations in one place

```
library(riskiconn)
```

Includes general DB configs like hostname, port, etc.

riskiconn

Internal package for handling database connections/queries

Configurations in one place

```
library(riskiconn)

get_query("select count(*)
from order_discounts")
```

Sets DB connection automatically unless explicitly provided

riskiconn

Internal package for handling database connections/queries

Configurations in one place

```
library(riskiconn)
options(riskidb = "snowflake")
get_query("select count(*)
from order_discounts")
```

During DB migration:
Set default database for R session

riskiconn

Internal package for handling database connections/queries

Configurations in one place

Caching query results

```
get_query("select count(*)
from order_discounts")
#> Querying DB...
#>  count(*)
#> 1 5717507
```

riskiconn

Internal package for handling database connections/queries

Configurations in one place

Caching query results

```
get_query("select count(*)
from order_discounts")
#> Reading from cache
#>  count(*)
#> 1 5717507
```

riskiconn

Internal package for handling database connections/queries

- Configurations in one place
- Caching query results
- Access to pipelines for moving large dataframes from R to the DB (& vice versa)

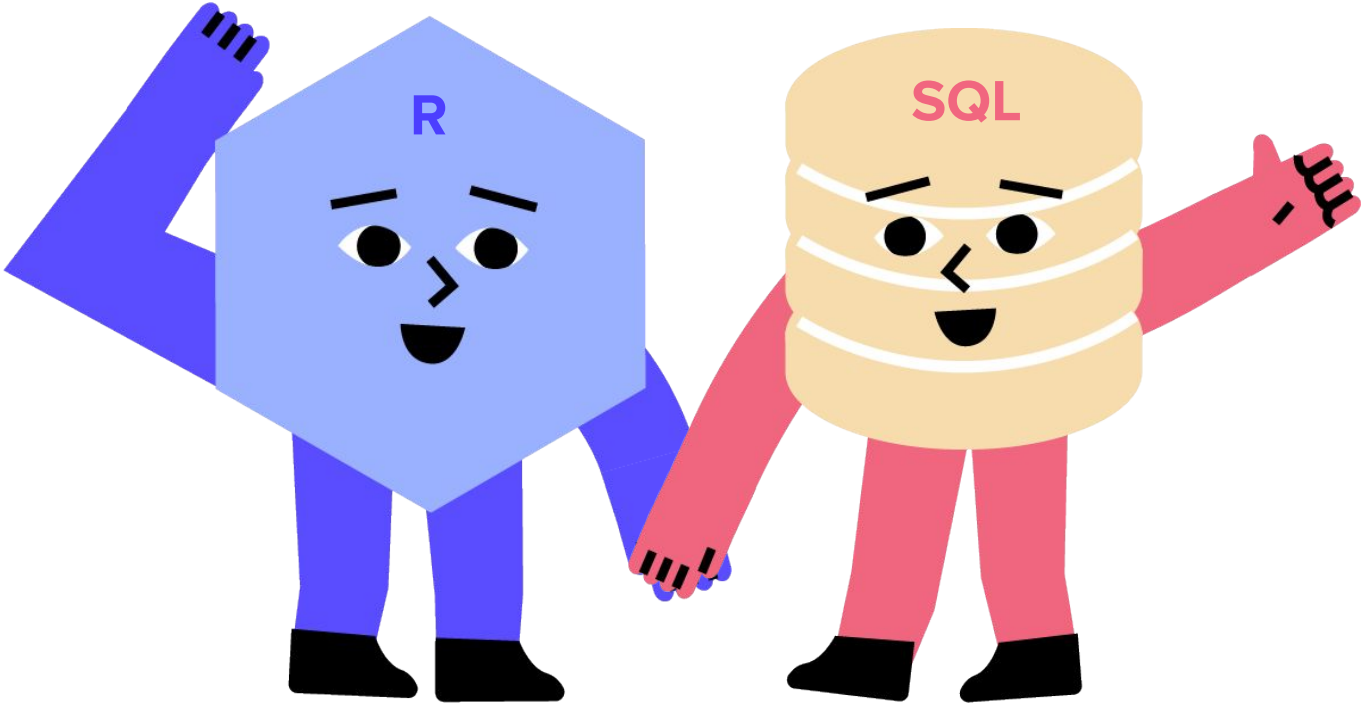
R & SQL work hand-in-hand

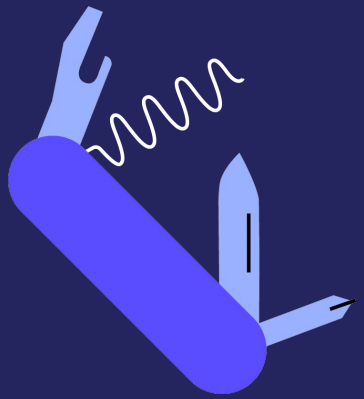
Summarizations across columns, advanced selection features

Translations to multiple dialects

Optimizing for speed/readability

Universal tool - used across many different technical roles


R
SQL



Thank you!

Irene Steves
@i_steves
irene.rbind.io
https://medium.com/riskified-technology