1. Installation

- Windows & Linux:
    1. Navigate to Duckdb Releases @github
    2. Download the appropriate distribution
    3. Unpack & Install

Additional instruction for Windows

- MacOS:
    1. Install homebrew
    2. ```
       brew install duckdb
       ```

2. Demo

1. Open terminal
2. ```
   duckdb demo1.duckdb
   ```
3. Lets have a look at available extensions

   ```
   DESCRIBE
       SELECT *
           FROM duckdb_extensions();
   ```
   ```
4. Which ones do we have installed & ready to be used?

   ```
   SELECT extension_name, loaded, installed
       from duckdb_extensions()
           ORDER BY installed DESC, loaded DESC;
   ```
5. Lets install & load `httpfs` so we can query data directly from the web

   ```
   INSTALL httpfs;
   LOAD httpfs;
   ```
   ```

6. What is the availale amount of memory to our duckdb instance?

```
select current_setting('memory_limit');
```

Lets increase it!

```
SET memory_limit='30GB';
```

7. Time to query some data!

```
SELECT count(*)
    FROM
read_csv_auto('https://raw.githubusercontent.com/JeffSackmann/tennis_a
tp/master/atp_matches_2023.csv');
```

8. Lets explore duckdbs output formats & change to a new one:

```
.help mode
```

```
.mode line
```

```
SELECT *
    FROM
read_csv_auto('https://raw.githubusercontent.com/JeffSackmann/tennis_a
tp/master/atp_matches_2023.csv')
    LIMIT 1;
```

9. Switching back to `duckbox` mode & querying a new data insight

```
.mode duckbox
```

```
select winner_ioc, loser_ioc, winner_hand, loser_hand,
count(distinct(tourney_id)) as num_winners, avg(winner_age) as
avg_winner_age, max(loser_age) as max_loser_age
    from
read_csv_auto('https://raw.githubusercontent.com/JeffSackmann/tennis_a
```

```
tp/master/atp_matches_2023.csv')
        group by all;
```

10. Lets go more interactive & automated!

-> Quit duckdb with Ctrl+D -> run:

```sh
    duckdb -csv \
    -s "select winner_ioc, loser_ioc, winner_hand, loser_hand,
count(distinct(tourney_id)) as num_winners, avg(winner_age) as
avg_winner_age, max(loser_age) as max_loser_age
    from
read_csv_auto('https://raw.githubusercontent.com/JeffSackmann/tennis_atp/m
aster/atp_matches_2023.csv')
        group by all;"\
      > results_2023.csv
```

```sh
head -n 5 results_2023.csv
```

10. Why write into `csv` when `parquet` is faster & takes less space?

```
    duckdb \
    -s "COPY (select winner_ioc, loser_ioc, winner_hand, loser_hand,
count(distinct(tourney_id)) as num_winners, avg(winner_age) as
avg_winner_age, max(loser_age) as max_loser_age
    from
read_csv_auto('https://raw.githubusercontent.com/JeffSackmann/tennis_a
tp/master/atp_matches_2023.csv')
        group by all) TO '/dev/stdout' (FORMAT PARQUET)"\
      > results_2023.parquet
```

Lets read the results:

```
duckdb -s "FROM 'results_2023.parquet' LIMIT 3"
```