ZOOLOGY 101:

pandas, polars, and ducks in the data wilderness

Jaroslav Bezdek, Machine Learning Engineer at STRV

STRV

MOTIVATION 01

MY_TOOLBOX


pandas


matplotlib


scikit
learn


NumPy


seaborn


K
Keras


dbt


Streamlit


TensorFlow

DUCKDB_INTEREST_OVER_TIME

Source: Google Trends

MY_TOOLBOX_WITH_DUCKDB

DuckDB

DATA

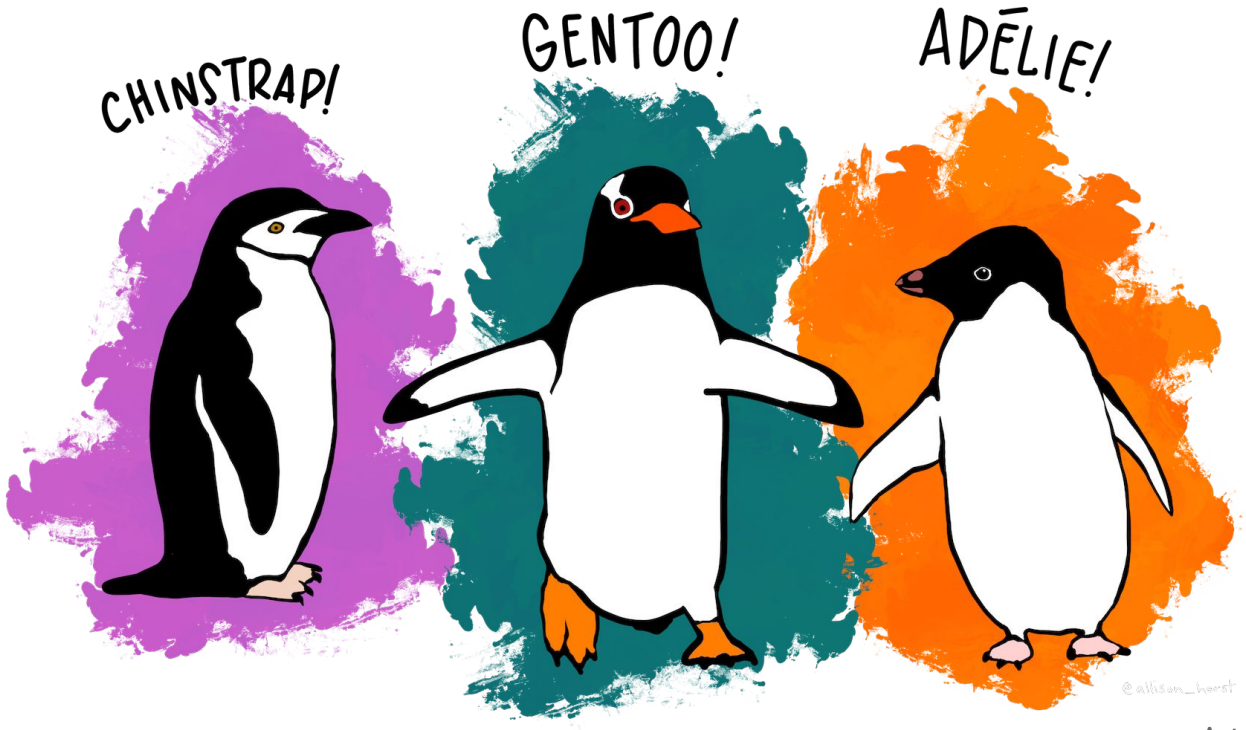
02

palmer_penguins_data.py

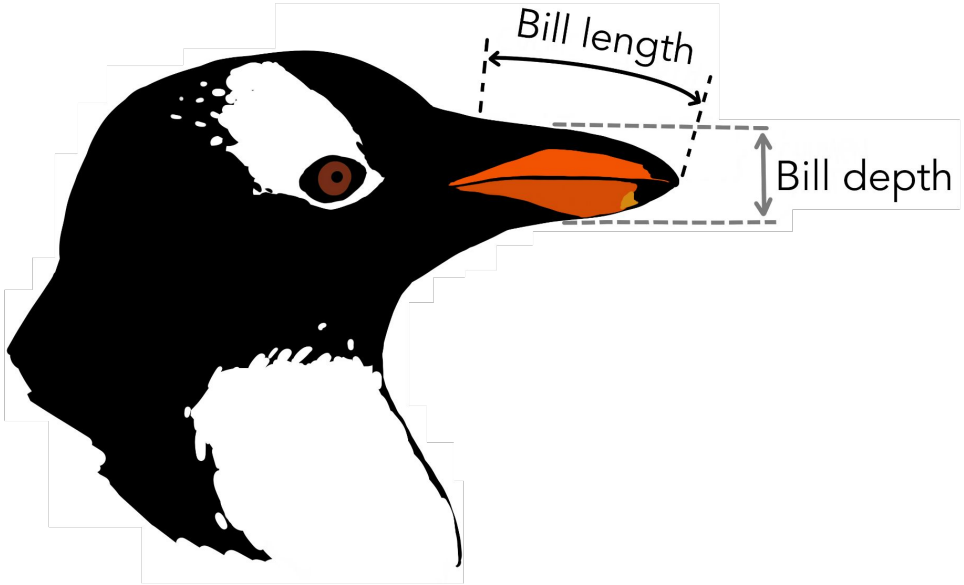
```
>> print(pinguins)

shape: (344, 6)
```

pinguin_id	date	species	bill_length	bill_depth	sex
str	str	str	f64	f64	str
N1A1	2007-11-11	adelie	39.1	18.7	MALE
N1A2	2007-11-11	adelie	39.5	17.4	FEMALE
N2A1	2007-11-16	adelie	40.3	18.0	FEMALE
N2A2	2007-11-16	adelie	null	null	null
N3A1	2007-11-16	adelie	36.7	19.3	FEMALE
…	…	…	…	…	…
N38A2	2009-12-01	gentoo	null	null	null
N39A1	2009-11-22	gentoo	46.8	14.3	FEMALE
N39A2	2009-11-22	gentoo	50.4	15.7	MALE
N43A1	2009-11-22	gentoo	45.2	14.8	FEMALE
N43A2	2009-11-22	gentoo	49.9	16.1	MALE



Artwork by: Allison Horst


Bill length
Bill depth

IMPORT PANDAS AS PD 03

features_pandas_dataframe.py

```
>> import pandas as pd
>>
>> df = pd.read_csv(
..     filepath_or_buffer="/path/to/palmer_penguins.csv"
.. )
>>
>> print(df)

    pinguin_id        date species  bill_length  bill_depth     sex
0         N1A1  2007-11-11  adelie         39.1        18.7    MALE
1         N1A2  2007-11-11  adelie         39.5        17.4  FEMALE
2         N2A1  2007-11-16  adelie         40.3        18.0  FEMALE
3         N2A2  2007-11-16  adelie          NaN         NaN     NaN
4         N3A1  2007-11-16  adelie         36.7        19.3  FEMALE
..         ...         ...     ...          ...         ...     ...
339      N38A2  2009-12-01  gentoo          NaN         NaN     NaN
340      N39A1  2009-11-22  gentoo         46.8        14.3  FEMALE
341      N39A2  2009-11-22  gentoo         50.4        15.7    MALE
342      N43A1  2009-11-22  gentoo         45.2        14.8  FEMALE
343      N43A2  2009-11-22  gentoo         49.9        16.1    MALE

[344 rows x 6 columns]
```

features_pandas_data_transformation.py

```
>> print(
..     df
..     .assign(
..         sex=lambda _df: _df.sex.str.lower(),
..     )
..     .groupby(by="species")
..     .agg(
..         samples_cnt=("pinguin_id", "count"),
..         male_cnt=("sex", lambda x: x[x == "male"].count()),
..         bill_length_avg=("bill_length", "mean"),
..     )
..     .round(1)
.. )

           samples_cnt  male_cnt  bill_length_avg
species
adelie             152        73             38.8
chinstrap           68        34             48.8
gentoo             124        61             47.5
```

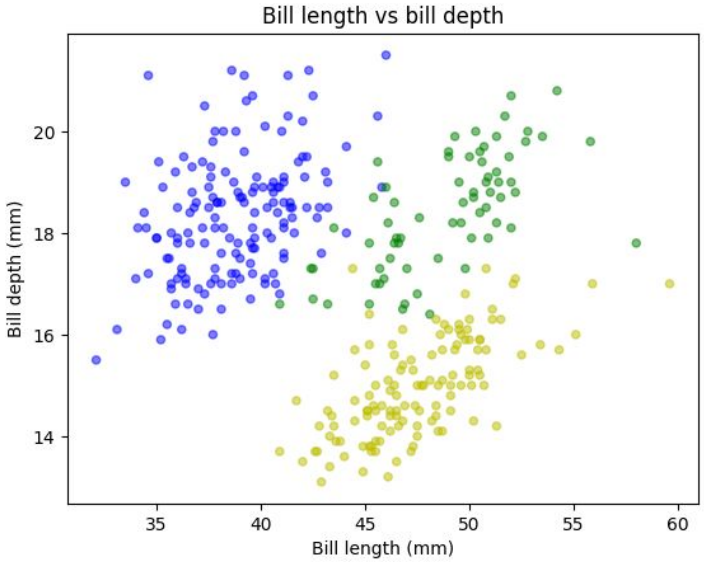
features_pandas_time_series_analysis.py

```
>> # convert string to datetime
>> df["date"] = pd.to_datetime(df.date)
>>
>> # get unique sample dates from year 2007
>> unique_dates_2007 = (
..     df
..     .date
..     .loc[df.date.dt.year == 2007]
..     .drop_duplicates()
..     .sort_values()
..     .reset_index(drop=True)
.. )
>>
>> # get the difference between sample days
>> print(unique_dates_2007 - unique_dates_2007.shift())

0       NaT
1    1 days
2    1 days
...
14   1 days
15   1 days
16   3 days
```

features_pandas_plot.py

```
>> (
..     df
..     .plot(
..         kind="scatter",
..         x="bill_length",
..         y="bill_depth",
..         color=df.species.map({
..             "adelie": "b",
..             "chinstrap": "g",
..             "gentoo": "y",
..         }),
..         alpha=0.5,
..         xlabel="Bill length (mm)",
..         ylabel="Bill depth (mm)",
..         title="Bill length vs bill depth",
..     )
.. )
```



STRENGTHS_AND_WEAKNESSES

👍

- Flexibility and easy to use
- Versatility
- Integration with other Python libraries
- Community and documentation

👎

- Performance with large datasets
- Focus on structured data
- Steeper learning curve for advanced features

NOTABLE_USE_CASES

- Data analysis & exploration
- Data visualization
- Time series analysis
- Data preparation for ML

IMPORT POLARS AS PL 04

features_polars_no_index.py

```
>> import polars as pl
>>
>> df = pl.read_csv(
..     source="/path/to/palmer_penguins.csv"
.. )
>>
>> print(df)

shape: (344, 6)
```

pinguin_id	date	species	bill_length	bill_depth	sex
str	str	str	f64	f64	str
N1A1	2007-11-11	adelie	39.1	18.7	MALE
N1A2	2007-11-11	adelie	39.5	17.4	FEMALE
N2A1	2007-11-16	adelie	40.3	18.0	FEMALE
N2A2	2007-11-16	adelie	null	null	null
N3A1	2007-11-16	adelie	36.7	19.3	FEMALE
…	…	…	…	…	…
N38A2	2009-12-01	gentoo	null	null	null
N39A1	2009-11-22	gentoo	46.8	14.3	FEMALE
N39A2	2009-11-22	gentoo	50.4	15.7	MALE
N43A1	2009-11-22	gentoo	45.2	14.8	FEMALE
N43A2	2009-11-22	gentoo	49.9	16.1	MALE

features_polars_missing_values.py

```
>> print(
..     df
..     .with_columns(
..         bill_depth_int=pl.col("bill_depth").cast(pl.Int64),
..         missing_values=None,
..     )
..     .select("pinguin_id", "bill_depth", "bill_depth_int", "missing_values")
..     .head()
.. )

shape: (5, 4)
```

pinguin_id	bill_depth	bill_depth_int	missing_values
str	f64	i64	null
N1A1	18.7	18	null
N1A2	17.4	17	null
N2A1	18.0	18	null
N2A2	null	null	null
N3A1	19.3	19	null

features_polars_eager_vs_lazy_execution.py

```
>> # eager execution
>> print(
..     pl
..     .read_csv("/path/to/palmer_penguins.csv")
..     .group_by("species")
..     .agg(
..         pl.col("bill_length").mean(),
..         pl.col("bill_depth").mean(),
..     )
.. )
>>
>> # lazy execution
>> print(
..     pl
..     .scan_csv("/path/to/palmer_penguins.csv")
..     .group_by("species")
..     .agg(
..         pl.col("bill_length").mean(),
..         pl.col("bill_depth").mean(),
..     )
..     .collect()
.. )
```

features_polars_sequential_vs_parallel_operations.py

```
>> df.with_columns(
..     # bill length
..     bill_length_nm=pl.col("bill_length") * 1e+6,
..     bill_length_um=pl.col("bill_length") * 1e+3,
..     bill_length_mm=pl.col("bill_length"),
..     bill_length_m=pl.col("bill_length") * 1e-3,
..     bill_length_km=pl.col("bill_length") * 1e-6,
..
..     # bill depth
..     bill_depth_nm=pl.col("bill_depth") * 1e+6,
..     bill_depth_um=pl.col("bill_depth") * 1e+3,
..     bill_depth_mm=pl.col("bill_depth"),
..     bill_depth_m=pl.col("bill_depth") * 1e-3,
..     bill_depth_km=pl.col("bill_depth") * 1e-6,
.. )
```

STRENGTHS_AND_WEAKNESSES

👍

- Performance
  - Lazy evaluation
  - Parallel execution
- Intuitive missing values definition

👎

- Dataframe indices
- Learning curve (when transferring from pandas)
- Limited functionality

NOTABLE_USE_CASES

- Data analysis & exploration
- Data visualization
- Data preparation for ML

IMPORT DUCKDB 05

“DuckDB is fast in-process analytical database.”

duckdb.org

DATABASE_WORKLOAD_TYPES

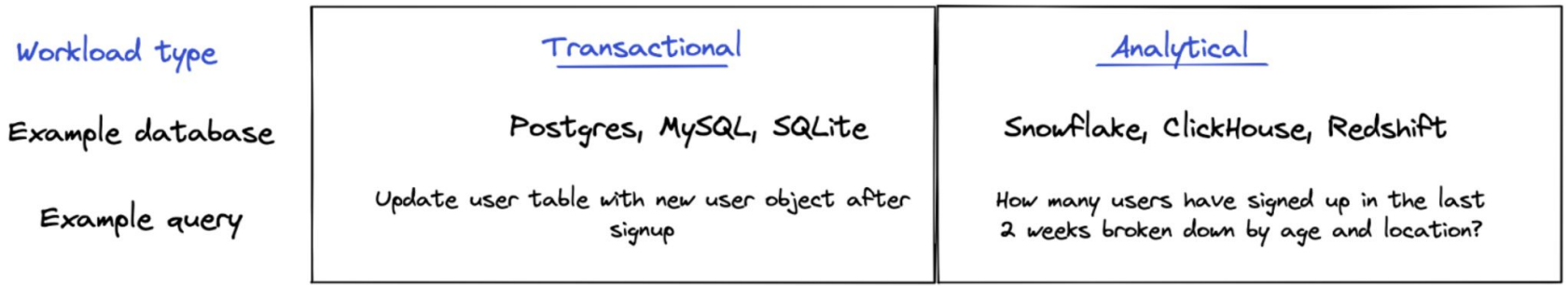


Image by Kojo Osei

DATABASE_DEPLOYMENT_TYPES

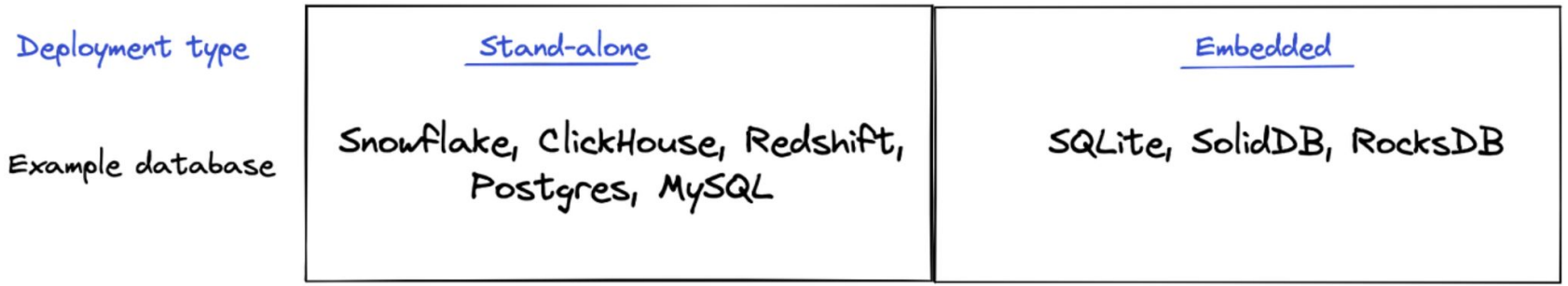


Image by Kojo Osei

DATABASE_TYPES

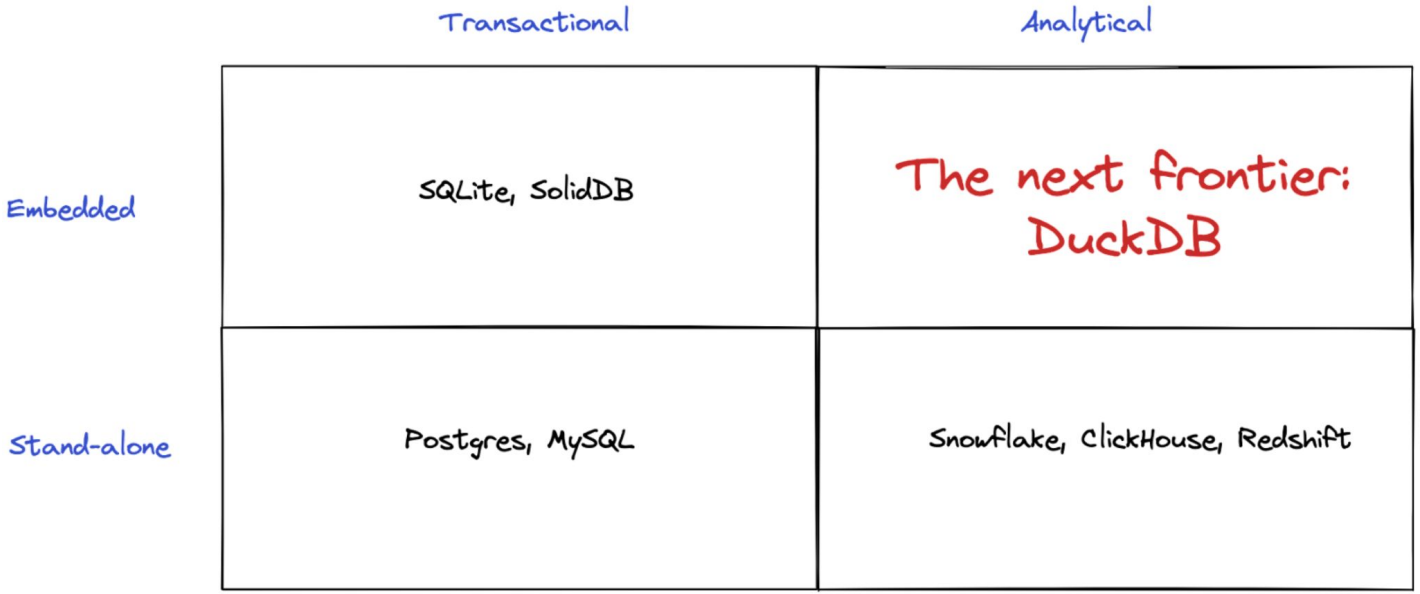


Image by Kojo Osei

features_duckdb_read_data.py

```
>> import duckdb
>>
>> data_raw = (
..     duckdb
..     .read_csv("/path/to/palmer_penguins/*.csv")
..     .select("species", "bill_length", "bill_depth", "sex")
.. )
>>
>> print(data_raw)
```

species varchar	bill_length double	bill_depth double	sex varchar
adelie	39.1	18.7	MALE
adelie	42.0	20.2	NULL
·	·	·	·
gentoo	46.2	14.1	FEMALE
gentoo	49.9	16.1	MALE

344 rows (4 shown) 4 columns

features_duckdb_sql_syntax_using_cte.py

```
>> duckdb.sql(query=f"""
..     with cleaned as (
..         select
..             species,
..             lower(sex) as sex,
..             bill_length,
..             bill_depth
..         from data_raw
..         where sex is not null
..     ),
..
..     agg as (
..         select
..             species,
..             avg(bill_length)::int as bill_length_avg,
..             avg(bill_depth)::int as bill_depth_avg
..         from cleaned
..         where sex = 'male'
..         group by species
..         order by species
..     )
..
..     select * from agg;
..     """
.. )
```

species varchar	bill_length_avg int32	bill_depth_avg int32
adelie	40	19
chinstrap	51	19
gentoo	49	16

features_duckdb_relational_api.py

```
>> (
..     data_raw
..     .project("""
..         species,
..         lower(sex) as sex,
..         bill_length,
..         bill_depth
..     """)
..     .filter("sex = 'male'")
..     .aggregate("""
..         species,
..         avg(bill_length)::int as bill_length_avg,
..         avg(bill_depth)::int as bill_depth_avg
..     """)
..     .order("species")
.. )
```

species varchar	bill_length_avg int32	bill_depth_avg int32
adelie	40	19
chinstrap	51	19
gentoo	49	16

features_duckdb_to_pandas.py

```
>> df_agg = agg.to_df()
>> print(df_agg)


     species  bill_length_avg  bill_depth_avg
0     adelie               40              19
1  chinstrap               51              19
2     gentoo               49              16
```

features_duckdb_from_pandas.py

```
>> agg = duckdb.sql("select * from df_agg")
>> print(agg)
```

species varchar	bill_length_avg int32	bill_depth_avg int32
adelie	40	19
chinstrap	51	19
gentoo	49	16

STRENGTHS_AND_WEAKNESSES

👍

- Reading multiple files at once
- SQL syntax
- APIs for major programming languages
- Performance

👎

- “Only” analytical workloads
- Community support
- Immaturity

NOTABLE_USE_CASES

- Data analysis using SQL syntax
- Working with datasets larger than memory
- Reading files from S3 comfortably

COMPARISON 06

COMPARISON

			
Speed	• Fast	• Faster	• Fastest
Size	• Big	• Medium	• Small
Maturity	• Adult	• Teenager	• Child
Syntax	• Complex	• Simpler	• Simple

THANK YOU!

www.strv.com / @strvcom

STRV