Is Margherita Better than Quattro Stagioni?

Pandas vs Polars API

Jan Pipek

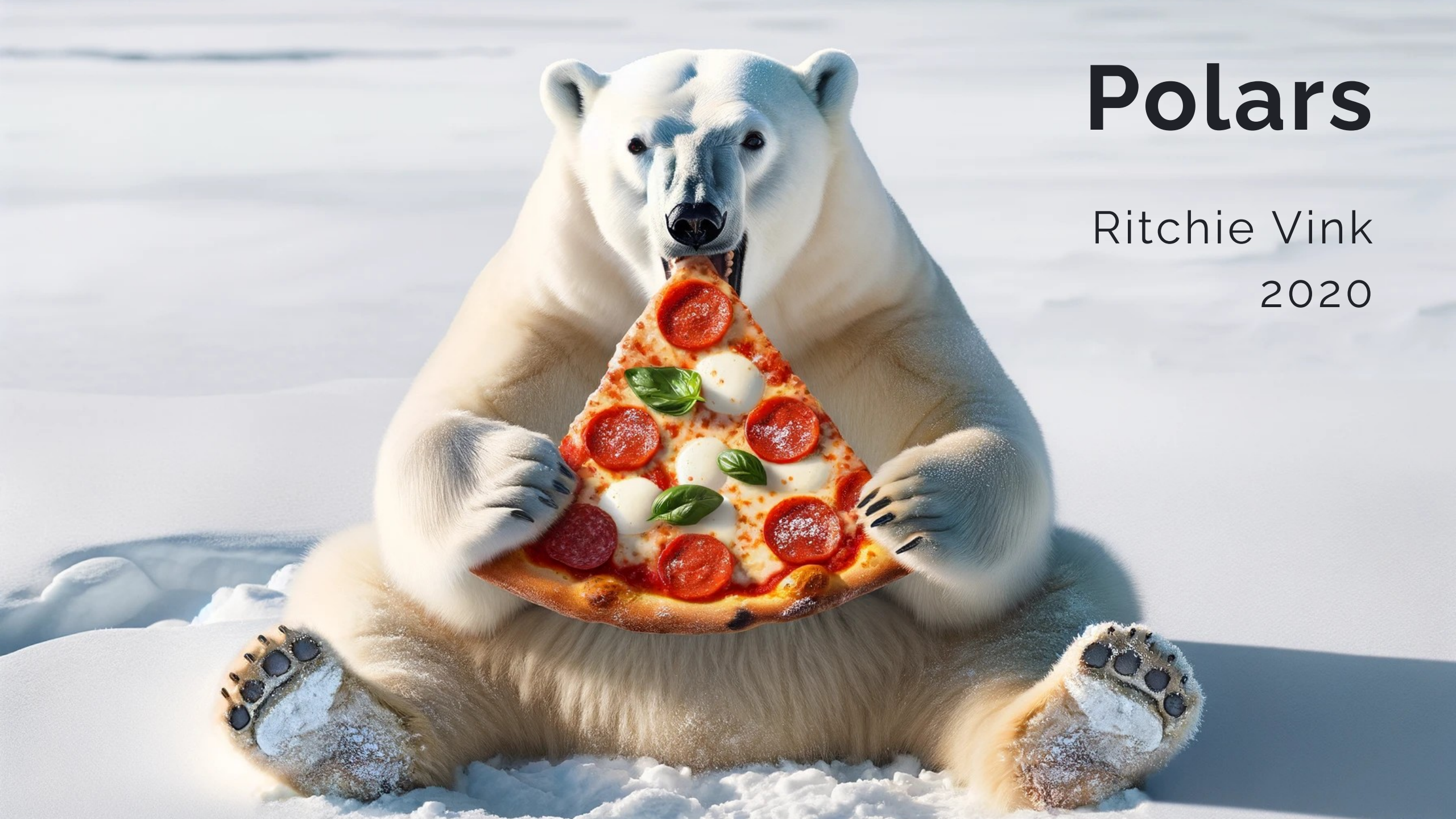
~~Contents~~

performance
backend
lazy/eager evaluation
love

Pandas

Wes McKinney

2008



Polars

Ritchie Vink

2020

Nice 🎉

but

Can I use polars in production code?

than complicated. Flat is better than nested. Sparse is better than dense. Readability counts. Special cases aren't special enough to break the rules. Although practicality beats purity. Errors should never pass silently. Unless explicitly silenced. In the face of ambiguity, refuse the temptation to guess. **There should be one-- and preferably only one -- obvious way to do it. Although that way may not be obvious at first unless you're Dutch.** Now is better than never. Although never is often better than *right* now. If the implementation is hard to explain, it's a bad idea. If the implementation is easy to explain, it may be a good idea. Namespaces are one honking

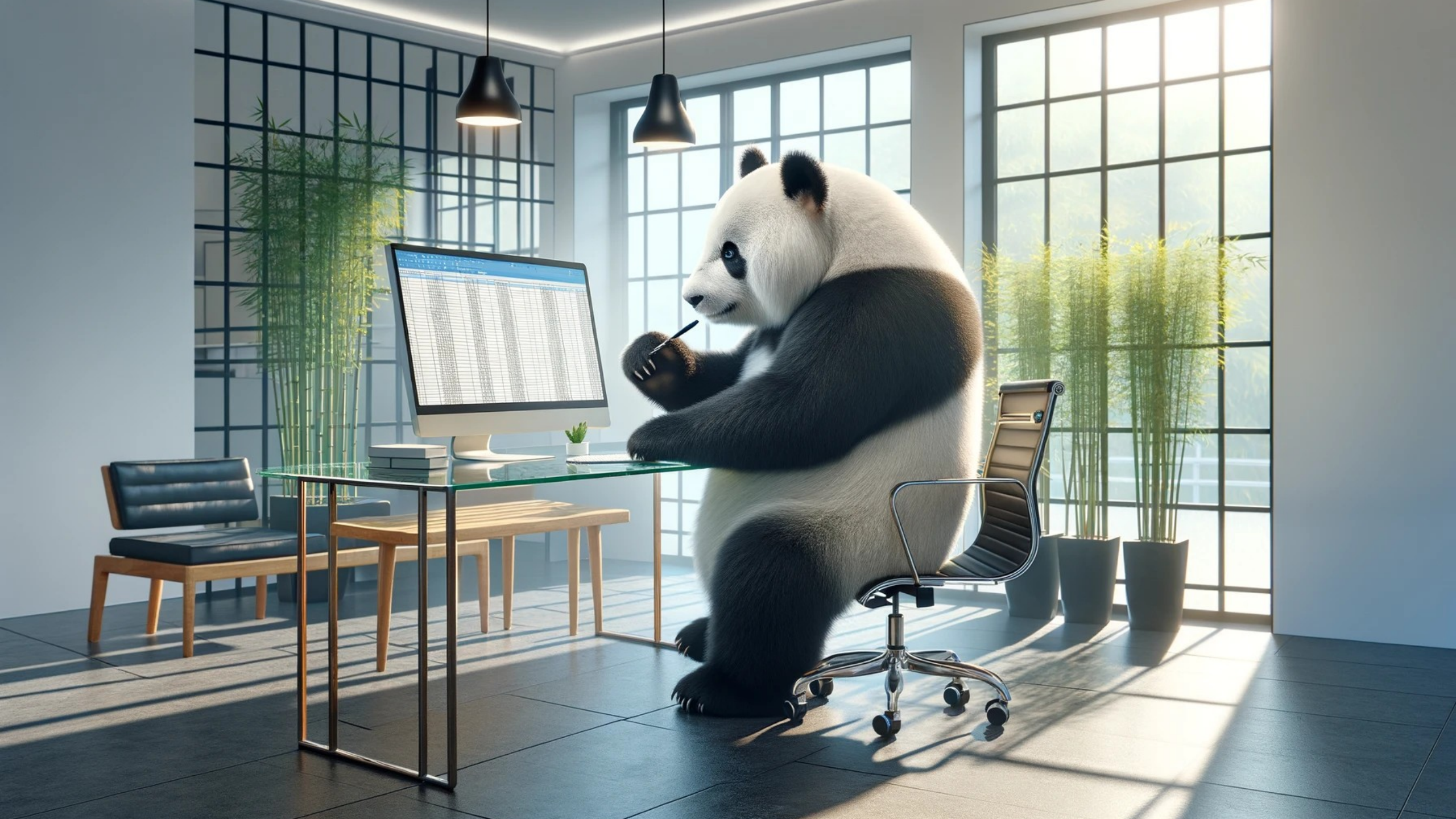
do not include

do not allow

do not guess

Do not include

e.g. indices



name	price	vegetarian
Margherita	150	True
Quattro formaggi	200	True
Quattro stagioni	200	False
Capricciosa	210	False
Prosciutto	170	False

Index

name	price	vegetarian
Margherita	150	True
Quattro formaggi	200	True
Quattro stagioni	200	False
Capricciosa	210	False
Prosciutto	170	False

Automatic index

	name	price	vegetarian
0	Margherita	150	True
1	Quattro formaggi	200	True
2	Quattro stagioni	200	False
3	Capricciosa	210	False
4	Prosciutto	170	False

Hierarchical indices

vegetarian	name	main_ingredient	price	
			25 cm	33 cm
True	Margherita	tomato	150	180
	Quattro formaggi	cheese	200	240
False	Quattro stagioni	cheese	200	250
	Capricciosa	ham	210	270
	Prosciutto	ham	170	210

How do I find my rows?

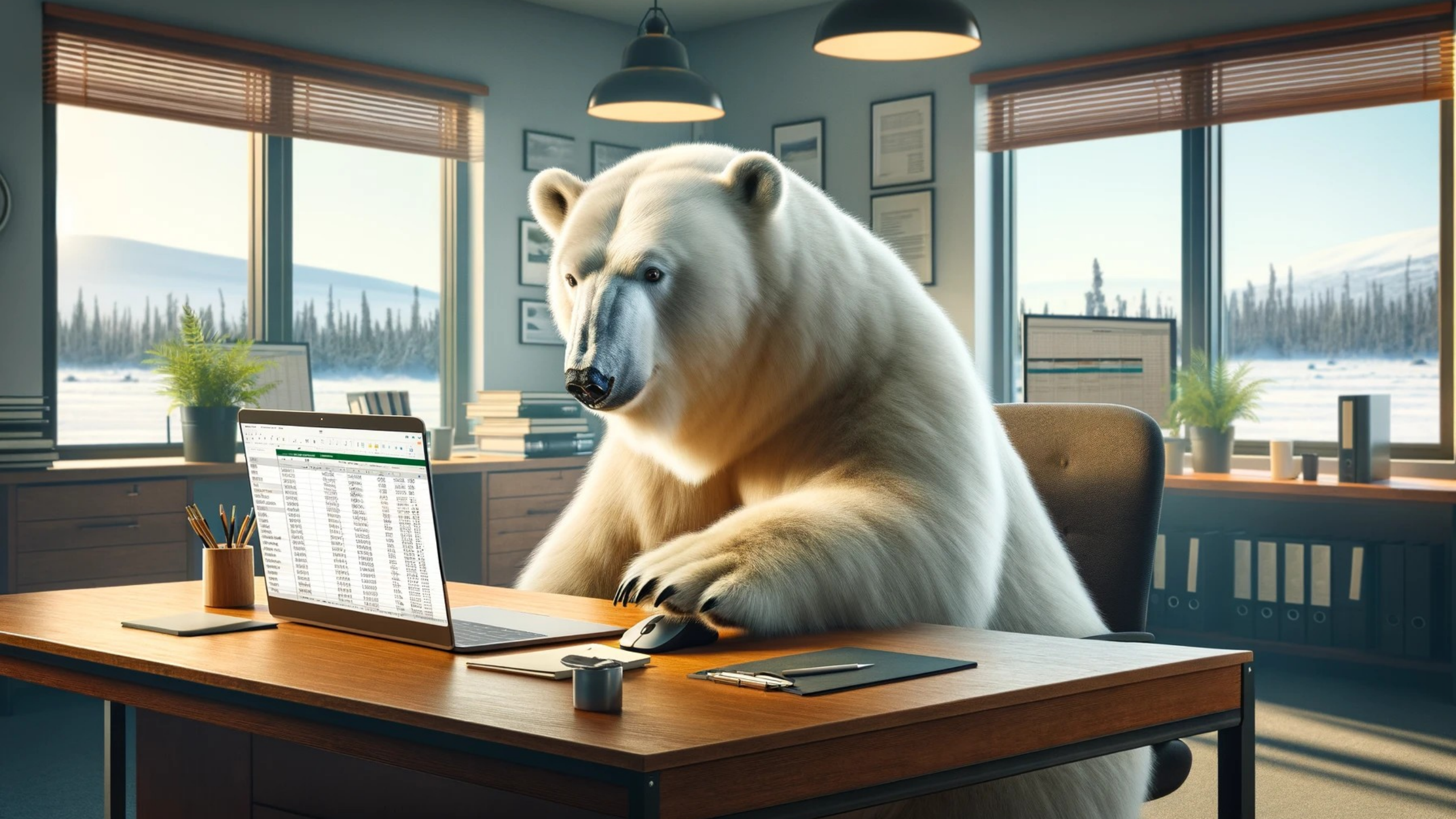
No index

```
df[df["name"] == x]                    # bracket and "df" gymnastics
df.loc[df["name"] == x]                # same
df[df.name == x]                       # believe in the dot!
df.query(f"name='{x}'")                # safe?
```

Yes index

```
df.loc[x]                                          # single (or first-level)
df.loc[(x, ...), :]                                # first-level index (fancy)
df.loc[(slice(None), x)]                           # second-level index (simple)
df.loc[pd.IndexSlice[:, x]]                        # second-level index (fancy)
df.loc(axis=0)[:, x]                               # second-level index (fancier)
df.xs(x, level="name")                             # what???
df[df.index.get_level_values("name") == x]         # why not?
```

And then there is .iloc too 😱



vegetarian	name	main_ingredient	price 25 cm	price 33 cm
bool	*str*	*str*	*i64*	*i64*
true	Margherita	tomato	150	180
true	Quattro formaggi	cheese	200	240
false	Quattro stagioni	cheese	200	250
false	Capricciosa	ham	210	270
false	Prosciutto	ham	170	210

```
df.filter(pl.col("name") == "margherita")

# Shortcut!
dl.filter(name="margherita")
```

Do not allow

e.g. mutability

Pandas

possible but discouraged

```
# Mutable
df["number_of_ingredients"] = [1, 2, 3]
df.loc["margherita", "number_of_ingredients"] = 3

# Immutable
df.assign(number_of_ingredients=[1, 2, 3])
```

Polars

almost impossible*

```
# Only immutable
df.with_columns(number_of_ingredients=[1, 2, 3])

df.with_columns(
    number_of_ingredients=pl.when(
        pl.col("name") == "margherita"
    )
    .then(3)
    .otherwise(
        pl.col("number_of_ingredients")
    )
)
```

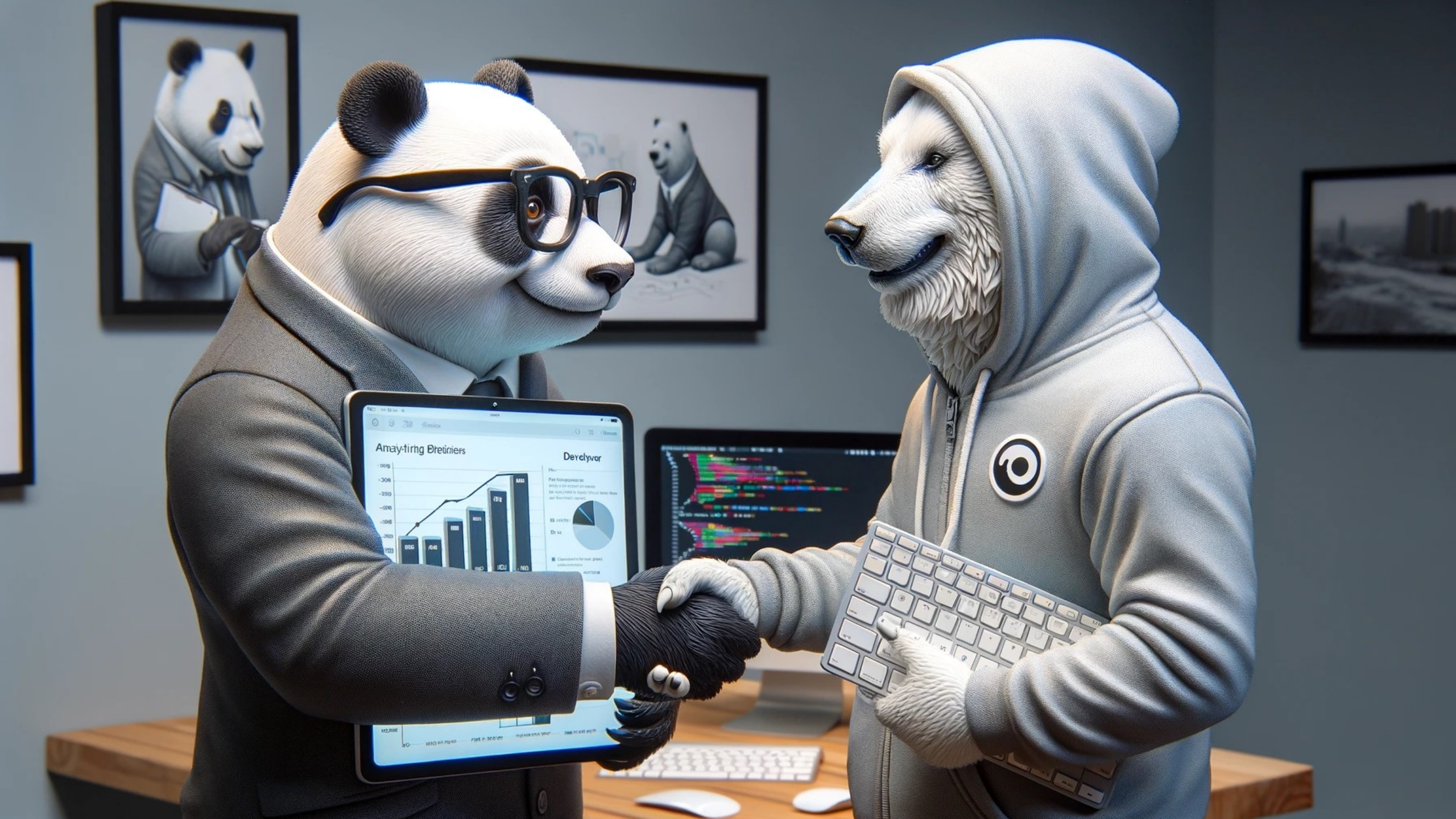
Do not guess

e.g. data dimensionality

e.g. data types

e.g. date format

Conclusion

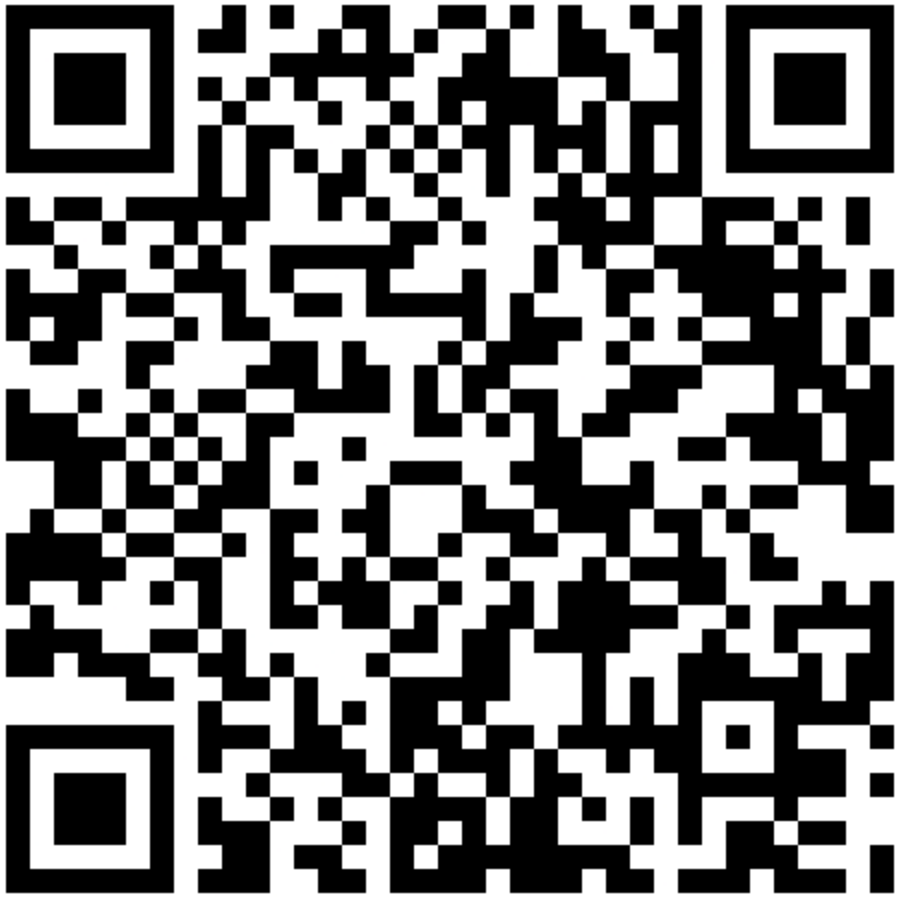


Pandas

less typing
intuition usually works
interactive use

Polars

more explicit
no surprises
data pipelines



Thank you!