Init

In [1]:
```
import pandas as pd
import pyarrow as pa
```

In [2]:
```
pd.show_versions()
```

```
INSTALLED VERSIONS
------------------
commit                : 0691c5cf90477d3503834d983f69350f250a6ff7
python                : 3.12.8
python-bits           : 64
OS                    : Windows
OS-release            : 2019Server
Version               : 10.0.17763
machine               : AMD64
processor             : Intel64 Family 6 Model 165 Stepping 2, GenuineIntel
byteorder             : little
LC_ALL                : None
LANG                  : None
LOCALE                : English_United States.1252

pandas                : 2.2.3
numpy                 : 2.1.2
pytz                  : 2024.2
dateutil              : 2.9.0.post0
pip                   : 24.3.1
Cython                : None
sphinx                : None
IPython               : 8.29.0
adbc-driver-postgresql: None
adbc-driver-sqlite    : None
bs4                   : 4.12.3
blosc                 : None
bottleneck            : 1.4.2
dataframe-api-compat  : None
fastparquet           : None
fsspec                : None
html5lib              : None
hypothesis            : None
gcsfs                 : None
jinja2                : 3.1.4
lxml.etree            : 5.3.0
matplotlib            : 3.9.2
numba                 : None
numexpr               : 2.10.1
odfpy                 : None
openpyxl              : 3.1.5
pandas_gbq            : None
psycopg2              : None
pymysql               : None
pyarrow               : 18.1.0
pyreadstat            : None
pytest                : None
python-calamine       : None
pyxlsb                : None
s3fs                  : None
scipy                 : None
sqlalchemy            : None
tables                : None
tabulate              : 0.9.0
xarray                : None
xlrd                  : None
xlsxwriter            : None
zstandard             : 0.23.0
tzdata                : 2024.2
qtpy                  : None
pyqt5                 : None
```

With NumPy

In [3]:
```
pd.DataFrame( { 'A': [ 'a1', 'a2' ] }, dtype = 'category' ).value_counts( dropna = False )
```

Out[3]:
```
A
a1    1
a2    1
Name: count, dtype: int64
```

In [4]:
```
pd.DataFrame( { 'A': [ 'a1', pd.NA ] }, dtype = 'category' ).value_counts( dropna = False )
```

Out[4]:
```
A
a1     1
nan    1
Name: count, dtype: int64
```

Same with PyArrow

In [5]:
```
pd.DataFrame( { 'A': [ 'a1', 'a2' ] }, dtype = pd.ArrowDtype( pa.dictionary( pa.int32(), pa.utf8() ) ) ).value_counts( dropna = False )
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[5], line 1
----> 1 pd.DataFrame( { 'A': [ 'a1', 'a2' ] }, dtype = pd.ArrowDtype( pa.dictionary( pa.int32(), pa.utf8() ) ) ).value_counts( dropna = False )

File C:\Python\Lib\site-packages\pandas\core\frame.py:7519, in DataFrame.value_counts(self, subset, normalize, sort, ascending, dropna)
   7517 # Force MultiIndex for a list_like subset with a single column
   7518 if is_list_like(subset) and len(subset) == 1:  # type: ignore[arg-type]
-> 7519     counts.index = MultiIndex.from_arrays(
   7520         [counts.index], names=[counts.index.name]
   7521     )
   7523 return counts

File C:\Python\Lib\site-packages\pandas\core\indexes\multi.py:533, in MultiIndex.from_arrays(cls, arrays, sortorder, names)
    530     if len(arrays[i]) != len(arrays[i - 1]):
    531         raise ValueError("all arrays must be same length")
--> 533 codes, levels = factorize_from_iterables(arrays)
    534 if names is lib.no_default:
    535     names = [getattr(arr, "name", None) for arr in arrays]

File C:\Python\Lib\site-packages\pandas\core\arrays\categorical.py:3069, in factorize_from_iterables(iterables)
   3065 if len(iterables) == 0:
   3066     # For consistency, it should return two empty lists.
   3067     return [], []
-> 3069 codes, categories = zip(*(factorize_from_iterable(it) for it in iterables))
   3070 return list(codes), list(categories)

File C:\Python\Lib\site-packages\pandas\core\arrays\categorical.py:3069, in <genexpr>(.0)
   3065 if len(iterables) == 0:
   3066     # For consistency, it should return two empty lists.
   3067     return [], []
-> 3069 codes, categories = zip(*(factorize_from_iterable(it) for it in iterables))
   3070 return list(codes), list(categories)

File C:\Python\Lib\site-packages\pandas\core\arrays\categorical.py:3042, in factorize_from_iterable(values)
   3037     codes = values.codes
   3038 else:
   3039     # The value of ordered is irrelevant since we don't use cat as such,
   3040     # but only the resulting categories, the order of which is independent
   3041     # from ordered. Set ordered to False as default. See GH #15457
-> 3042     cat = Categorical(values, ordered=False)
   3043     categories = cat.categories
   3044     codes = cat.codes

File C:\Python\Lib\site-packages\pandas\core\arrays\categorical.py:451, in Categorical.__init__(self, values, categories, ordered, dtype, fastpath, copy)
    447 if dtype.categories is None:
    448     if isinstance(values.dtype, ArrowDtype) and issubclass(
    449         values.dtype.type, CategoricalDtypeType
    450     ):
--> 451         arr = values._pa_array.combine_chunks()
    452         categories = arr.dictionary.to_pandas(types_mapper=ArrowDtype)
    453         codes = arr.indices.to_numpy()

AttributeError: 'Index' object has no attribute '_pa_array'
```

In [6]:
```
pd.DataFrame( { 'A': [ 'a1', pd.NA ] }, dtype = pd.ArrowDtype( pa.dictionary( pa.int32(), pa.utf8() ) ) ).value_counts( dropna = False )
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[6], line 1
----> 1 pd.DataFrame( { 'A': [ 'a1', pd.NA ] }, dtype = pd.ArrowDtype( pa.dictionary( pa.int32(), pa.utf8() ) ) ).value_counts( dropna = False )

File C:\Python\Lib\site-packages\pandas\core\frame.py:7519, in DataFrame.value_counts(self, subset, normalize, sort, ascending, dropna)
   7517 # Force MultiIndex for a list_like subset with a single column
   7518 if is_list_like(subset) and len(subset) == 1:  # type: ignore[arg-type]
-> 7519     counts.index = MultiIndex.from_arrays(
   7520         [counts.index], names=[counts.index.name]
   7521     )
   7523 return counts

File C:\Python\Lib\site-packages\pandas\core\indexes\multi.py:533, in MultiIndex.from_arrays(cls, arrays, sortorder, names)
    530     if len(arrays[i]) != len(arrays[i - 1]):
    531         raise ValueError("all arrays must be same length")
--> 533 codes, levels = factorize_from_iterables(arrays)
    534 if names is lib.no_default:
    535     names = [getattr(arr, "name", None) for arr in arrays]

File C:\Python\Lib\site-packages\pandas\core\arrays\categorical.py:3069, in factorize_from_iterables(iterables)
   3065 if len(iterables) == 0:
   3066     # For consistency, it should return two empty lists.
   3067     return [], []
-> 3069 codes, categories = zip(*(factorize_from_iterable(it) for it in iterables))
   3070 return list(codes), list(categories)

File C:\Python\Lib\site-packages\pandas\core\arrays\categorical.py:3069, in <genexpr>(.0)
   3065 if len(iterables) == 0:
   3066     # For consistency, it should return two empty lists.
   3067     return [], []
-> 3069 codes, categories = zip(*(factorize_from_iterable(it) for it in iterables))
   3070 return list(codes), list(categories)

File C:\Python\Lib\site-packages\pandas\core\arrays\categorical.py:3042, in factorize_from_iterable(values)
   3037     codes = values.codes
   3038 else:
   3039     # The value of ordered is irrelevant since we don't use cat as such,
   3040     # but only the resulting categories, the order of which is independent
   3041     # from ordered. Set ordered to False as default. See GH #15457
-> 3042     cat = Categorical(values, ordered=False)
   3043     categories = cat.categories
   3044     codes = cat.codes

File C:\Python\Lib\site-packages\pandas\core\arrays\categorical.py:451, in Categorical.__init__(self, values, categories, ordered, dtype, fastpath, copy)
    447 if dtype.categories is None:
    448     if isinstance(values.dtype, ArrowDtype) and issubclass(
    449         values.dtype.type, CategoricalDtypeType
    450     ):
--> 451         arr = values._pa_array.combine_chunks()
    452         categories = arr.dictionary.to_pandas(types_mapper=ArrowDtype)
    453         codes = arr.indices.to_numpy()

AttributeError: 'Index' object has no attribute '_pa_array'
```

With NumPy

In [7]:
```
pd.concat( [
    pd.DataFrame( { 'A': [ 'a1', 'a2' ], 'B': [ 'b1', pd.NA ] }, dtype = 'str' ),
    pd.DataFrame( { 'C': [ 'c1', 'c2' ], 'D': [ 'd1', pd.NA ] }, dtype = 'category' )
], axis = 1 ).value_counts( dropna = False )
```

Out[7]:
```
A   B    C   D
a1  b1   c1  d1    1
a2  NaN  c2  NaN   1
Name: count, dtype: int64
```

Note that in second line both B and D are `NaN` : correct!

Same with PyArrow

In [8]:
```
pd.concat( [
    pd.DataFrame( { 'A': [ 'a1', 'a2' ], 'B': [ 'b1', pd.NA ] }, dtype = pd.ArrowDtype( pa.string() ) ),
    pd.DataFrame( { 'C': [ 'c1', 'c2' ], 'D': [ 'd1', pd.NA ] }, dtype = pd.ArrowDtype( pa.dictionary( pa.int32(), pa.utf8() ) ) )
], axis = 1 ).value_counts( dropna = False )
```

Out[8]:
```
A   B     C   D
a1  b1    c1  d1    1
a2  <NA>  c2  d1    1
Name: count, dtype: int64
```

Note that in second line D is `d1` and not `<NA>` : wrong!