Initialize a new Expectation Suite by profiling a batch of your data.

This process helps you avoid writing lots of boilerplate when authoring suites by allowing you to select columns and other factors that you care about and letting a profiler write some candidate expectations for you to adjust.

Expectation Suite Name: `exp_suite`

In [1]:

```
import datetime

import pandas as pd

import great_expectations as ge
import great_expectations.jupyter_ux
from great_expectations.profile.user_configurable_profiler import (
    UserConfigurableProfiler,
)
from great_expectations.core.batch import BatchRequest
from great_expectations.checkpoint import SimpleCheckpoint
from great_expectations.exceptions import DataContextError

context = ge.data_context.DataContext()

batch_request = {
    "datasource_name": "my_pandas_datasource",
    "data_connector_name": "default_inferred_data_connector_name",
    "data_asset_name": "first_data.csv",
    "limit": 1000,
}

expectation_suite_name = "exp_suite"

validator = context.get_validator(
    batch_request=BatchRequest(**batch_request),
    expectation_suite_name=expectation_suite_name,
)
column_names = [f'"{column_name}"' for column_name in validator.columns()]
print(f"Columns: {', '.join(column_names)}.")
validator.head(n_rows=5, fetch_all=False)
```

```
2022-07-14T00:06:58+0530 - INFO - Great Expectations logging enabled at 20 level by J
upyterUX module.
Calculating Metrics:   0%|          | 0/2 [00:00<?, ?it/s]
Columns: "TV", "Radio", "Newspaper", "Sales".
Calculating Metrics:   0%|          | 0/1 [00:00<?, ?it/s]
```

Out[1]:

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9

Select columns

Select the columns on which you would like to set expectations and those which you would like to ignore.

Great Expectations will choose which expectations might make sense for a column based on the **data type** and **cardinality** of the data in each selected column.

Simply comment out columns that are important and should be included. You can select multiple lines and use a Jupyter keyboard shortcut to toggle each line: **Linux/Windows**: `Ctrl-/`, **macOS**: `Cmd-/`

In [2]:

```
exclude_column_names = [
    "TV",
    "Radio",
    "Newspaper",
    "Sales",
]
```

```
C:\Users\rosha\AppData\Roaming\Python\Python38\site-packages\ipykernel\ipkernel.py:28
3: DeprecationWarning: `should_run_async` will not call `transform_cell` automaticall
y in the future. Please pass the result to `transformed_cell` argument and any except
ion that happen during thetransform in `preprocessing_exc_tuple` in IPython 7.17 and
above.
  and should_run_async(code)
```

Run the UserConfigurableProfiler

The suites generated here are **not meant to be production suites** -- they are **a starting point to build upon**.

To get to a production-grade suite, you will definitely want to edit this suite after this initial step gets you started on the path towards what you want.

This is highly configurable depending on your goals. You can ignore columns or exclude certain expectations, specify a threshold for creating value set expectations, or even specify semantic types for a given column. You can find more information about how to configure this profiler, including a list of the expectations that it uses, here.

In [3]:

```
profiler = UserConfigurableProfiler(
    profile_dataset=validator,
    excluded_expectations=None,
    ignored_columns=exclude_column_names,
    not_null_only=False,
    primary_or_compound_key=None,
    semantic_types_dict=None,
    table_expectations_only=False,
    value_set_threshold="MANY",
)
suite = profiler.build_suite()
validator.expectation_suite = suite
```

```
Calculating Metrics:   0%|          | 0/2 [00:00<?, ?it/s]
2022-07-14T00:07:59+0530 - INFO -       0 expectation(s) included in expectation_suit
e.
Calculating Metrics:   0%|          | 0/2 [00:00<?, ?it/s]
Calculating Metrics:   0%|          | 0/1 [00:00<?, ?it/s]
Calculating Metrics:   0%|          | 0/1 [00:00<?, ?it/s]
2022-07-14T00:07:59+0530 - INFO -       2 expectation(s) included in expectation_suit
e.
Creating an expectation suite with the following expectations:

Table-Level Expectations
expect_table_columns_to_match_ordered_list
expect_table_row_count_to_be_between
```

Save & review your new Expectation Suite

Let's save the draft expectation suite as a JSON file in the `great_expectations/expectations` directory of your project and rebuild the Data Docs site to make it easy to review your new suite.

In [4]:

```
print(validator.get_expectation_suite(discard_failed_expectations=False))
validator.save_expectation_suite(discard_failed_expectations=False)

checkpoint_config = {
    "class_name": "SimpleCheckpoint",
    "validations": [
        {
            "batch_request": batch_request,
            "expectation_suite_name": expectation_suite_name,
        }
    ],
}
checkpoint = SimpleCheckpoint(
    f"_tmp_checkpoint_{expectation_suite_name}", context, **checkpoint_config
)
checkpoint_result = checkpoint.run()

context.build_data_docs()

validation_result_identifier = checkpoint_result.list_validation_result_identifiers()[
context.open_data_docs(resource_identifier=validation_result_identifier)
```

```
2022-07-14T00:08:06+0530 - INFO -       2 expectation(s) included in expectation_suit
e.
{
  "expectation_suite_name": "exp_suite",
  "expectations": [
    {
      "meta": {},
      "expectation_type": "expect_table_columns_to_match_ordered_list",
      "kwargs": {
        "column_list": [
          "TV",
          "Radio",
          "Newspaper",
          "Sales"
        ]
      }
    },
    {
      "meta": {},
      "expectation_type": "expect_table_row_count_to_be_between",
      "kwargs": {
        "min_value": 99,
        "max_value": 99
      }
    }
  ],
  "meta": {
    "citations": [
      {
        "batch_request": {
          "data_asset_name": "first_data.csv",
          "data_connector_name": "default_inferred_data_connector_name",
          "datasource_name": "my_pandas_datasource",
          "limit": 1000
        },
        "citation_date": "2022-07-13T18:35:45.127312Z",
        "comment": "Created suite added via CLI"
      }
    ],
    "great_expectations_version": "0.15.12",
    "columns": {
      "TV": {
        "description": ""
      },
      "Radio": {
        "description": ""
      },
      "Newspaper": {
        "description": ""
      },
      "Sales": {
        "description": ""
      }
    }
  },
  "data_asset_type": null,
  "ge_cloud_id": null
}
2022-07-14T00:08:06+0530 - INFO -       2 expectation(s) included in expectation_suit
e.
```

```
2022-07-14T00:08:06+0530 - INFO -       2 expectation(s) included in expectation_suit
e.
Calculating Metrics:   0%|          | 0/3 [00:00<?, ?it/s]
```

Next steps

After you review this initial Expectation Suite in Data Docs you should edit this suite to make finer grained adjustments to the expectations. This can be done by running `great_expectations suite edit exp_suite`.