pipeline Documentation

- **Created at:** 2017-04-10 16:46:13.277706
- **Corral Version:** 0.2.6

`<EMPTY DOC>` ## Models

Model Statistics

- **Table:** `Statistics`

`<EMPTY DOC>`

Model Name

- **Table:** `Name`

`<EMPTY DOC>`

Model Observation

- **Table:** `Observation`

`<EMPTY DOC>`

Loader:

- **Python Path** `pipeline.load.Loader`

`<EMPTY DOC>`

Steps

Step SetosaStatistics

- **Python Path** `pipeline.steps.SetosaStatistics`

`<EMPTY DOC>`

Step StatisticsCreator

- **Python Path** `pipeline.steps.StatisticsCreator`

`<EMPTY DOC>`

Step VersicolorStatistics

- **Python Path** `pipeline.steps.VersicolorStatistics`

`<EMPTY DOC>`

Step VirginicaStatistics

- **Python Path** `pipeline.steps.VirginicaStatistics`

`<EMPTY DOC>`

Alerts

Alert StatisticsAlert

- **Python Path** `pipeline.alerts.StatisticsAlert`

`<EMPTY DOC>`

Command Line Interface

run 'python in_corral.py –help'

Available subcommands

CORRAL

- `alembic`: Execute all the Alembic migration tool commands under Corral enviroment
- `check-alerts`: Run the alerts and announce to the endpoint if something is found
- `create`: Create a new corral pipeline
- `create-doc`: Generate a Markdown documentation for your pipeline
- `create-models-diagram`: Generates a class diagram in 'dot' format of the models classes
- `createdb`: Create all the database structure
- `dbshell`: Run an SQL shell throught sqlalchemy
- `exec`: Execute file inside corral environment
- `groups`: List all existent groups for Steps and Alerts
- `load`: Excecute the loader class
- `lsalerts`: List all available alert classes
- `lssteps`: List all available step classes
- `makemigrations`: Generate a database migration script for your current pipeline

- `migrate`: Synchronizes the database state with the current set of models and migrations
- `notebook`: Run the Jupyter notebook inside Corral enviroment
- `qareport`: Run the QA test for your pipeline and make a reports of errors, maintanability, coverage and a full QA index.
- `run`: Excecute the steps in order or one step in particular
- `run-all`: Shortcut command to run the loader, steps and alerts asynchronous. For more control check the commands 'load', 'run' and 'check-alerts'.
- `shell`: Run the Python shell inside Corral enviroment
- `test`: Run all unittests for your pipeline

pipeline

- `ls`: hace el ls macho