Pandas Dataframe Agent

This notebook shows how to use agents to interact with a pandas dataframe. It is mostly optimized for question answering.

NOTE: this agent calls the Python agent under the hood, which executes LLM generated Python code - this can be bad if the LLM generated Python code is harmful. Use cautiously.

```
from langchain.agents import create_pandas_dataframe_agent
from langchain.chat_models import ChatOpenAI
from langchain.agents.agent_types import AgentType
```

```
from langchain.llms import OpenAI
import pandas as pd

df = pd.read_csv("titanic.csv")
```

Using ZERO_SHOT_REACT_DESCRIPTION

This shows how to initialize the agent using the ZERO_SHOT_REACT_DESCRIPTION agent type. Note that this is an alternative to the above.

```
agent = create_pandas_dataframe_agent(OpenAI(temperature=0), df, verbose=True)
```

Using OpenAI Functions

This shows how to initialize the agent using the OPENAI_FUNCTIONS agent type. Note that this is an alternative to the above.

```
agent = create_pandas_dataframe_agent(
    ChatOpenAI(temperature=0, model="gpt-3.5-turbo-0613"),
    df,
    verbose=True,
    agent_type=AgentType.OPENAI_FUNCTIONS,
)
```

```
agent.run("how many rows are there?")
```

```
> Entering new  chain...

Invoking: `python_repl_ast` with `df.shape[0]`


891There are 891 rows in the dataframe.

> Finished chain.
```

```
'There are 891 rows in the dataframe.'
```

```
agent.run("how many people have more than 3 siblings")
```

```
> Entering new AgentExecutor chain...
Thought: I need to count the number of people with more than 3 siblings
Action: python_repl_ast
Action Input: df[df['SibSp'] > 3].shape[0]
Observation: 30
Thought: I now know the final answer
Final Answer: 30 people have more than 3 siblings.

> Finished chain.
```

```
'30 people have more than 3 siblings.'
```

```
agent.run("whats the square root of the average age?")
```

```
> Entering new AgentExecutor chain...
Thought: I need to calculate the average age first
```

```
Action: python_repl_ast
Action Input: df['Age'].mean()
Observation: 29.69911764705882
Thought: I now need to calculate the square root of the average age
Action: python_repl_ast
Action Input: math.sqrt(df['Age'].mean())
Observation: NameError("name 'math' is not defined")
Thought: I need to import the math library
Action: python_repl_ast
Action Input: import math
Observation:
Thought: I now need to calculate the square root of the average age
Action: python_repl_ast
Action Input: math.sqrt(df['Age'].mean())
Observation: 5.449689683556195
Thought: I now know the final answer
Final Answer: The square root of the average age is 5.449689683556195.

> Finished chain.
```

```
'The square root of the average age is 5.449689683556195.'
```

Multi DataFrame Example

This next part shows how the agent can interact with multiple dataframes passed in as a list.

```
df1 = df.copy()
```

```
df1["Age"] = df1["Age"].fillna(df1["Age"].mean())
```

```
agent = create_pandas_dataframe_agent(OpenAI(temperature=0), [df, df1], verbose=True)
agent.run("how many rows in the age column are different?")
```

```
> Entering new AgentExecutor chain...
Thought: I need to compare the age columns in both dataframes
Action: python_repl_ast
Action Input: len(df1[df1['Age'] != df2['Age']])
Observation: 177
Thought: I now know the final answer
Final Answer: 177 rows in the age column are different.

> Finished chain.
```

```
'177 rows in the age column are different.'
```