Here is the code that will need to be run in your Synapse Analytics workspace notebook to recreate Figure 16-4.

from azureml.opendatasets import NycTlcYellow

from datetime import datetime

from dateutil import parser

end\_date = parser.parse('2018-05-08 00:00:00')

start\_date = parser.parse('2018-05-01 00:00:00')

nyc\_tlc = NycTlcYellow(start\_date=start\_date, end\_date=end\_date)

nyc\_tlc\_df = nyc\_tlc.to\_spark\_dataframe()

After the job succeeds, run a select on the VendorID column to confirm that you have approximately 2.1M records using the following code: SELECT COUNT(vendorID) FROM nyc\_tlc\_df

Here is the code shown in Figure 16-6 that you will need to run.

nyc\_tlc\_df.createOrReplaceTempView('nyc\_tlc\_df')

nyc\_tlc\_df.write.parquet('nyc\_tlc\_df\_parquet', mode='overwrite')

Run the following query on the data frame to obtain a benchmark on how long the aggregate query takes to complete.

from pyspark.sql import functions as F

df = nyc\_tlc\_df.groupBy("passengerCount").agg(F.avg('tripDistance').alias('AvgTripDistance'), F.sum('tripDistance').alias('SumTripDistance'))

display(df)

Run the following code to import Hyperspace.

from hyperspace import Hyperspace

hs = Hyperspace(spark)

Here is the code that you will need to run to re-create the results shown in Figure 16-10.

df=spark.read.parquet("/user/trusted-service-user/nyc\_tlc\_df\_parquet/")

Here is the code that you will need to run to re-create the results shown in Figure 16-11.

from hyperspace import IndexConfig

hs.createIndex(df, IndexConfig("vendorID", ["passengerCount"], ["tripDistance"]))

Now that you have created a Hyperspace Index, re-run your original query from earlier to explore the execution time and query details.

from pyspark.sql import functions as F

df = nyc\_tlc\_df.groupBy("passengerCount").agg(F.avg('tripDistance').alias('AvgTripDistance'), F.sum('tripDistance').alias('SumTripDistance'))

display(df)

The following scripts list out other APIs to refresh, delete(soft-delete), restore and vaccum(hard-delete) the Hyperspace Indexes.

// Refreshes the given index if the source data changes.

hs.refreshIndex("index")

// Soft-deletes the given index and does not physically remove it from filesystem.

hs.deleteIndex("index")

// Restores the soft-deleted index.

hs.restoreIndex("index")

// Hard-delete the given index and physically remove it from filesystem.

hs.vacuumIndex("index")

Website References:

<https://microsoft.github.io/hyperspace/>

<https://docs.microsoft.com/en-us/azure/synapse-analytics/quickstart-create-workspace>

<https://docs.microsoft.com/en-us/azure/synapse-analytics/quickstart-create-apache-spark-pool-portal>

<https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>

<https://docs.microsoft.com/en-us/azure/synapse-analytics/spark/apache-spark-performance-hyperspace>