ITCS 6190 Assignment 1: Docker Containers Submission Report

Commands Used to Build and Run the Stack

Primary Commands

```
# Navigate to project directory
cd path/to/project

# Build and run the complete stack
make all

# Stop services when done
make down
```

Alternative Commands (if make not available)

```
# Direct Docker Compose commands
docker compose up --build
docker compose down -v
```

Verification Commands

```
# Check output file
cat out/summary.json

# View container status
docker compose ps

# Check logs if needed
docker compose logs app
```

Summary from stdout (Console Output)

Output from running `make all`:

```
=== Summary ===
{
  "total_trips": 6,
  "avg_fare_by_city": [
    {"city": "Charlotte", "avg_fare": 16.25},
    {"city": "New York", "avg_fare": 19.0},
    {"city": "San Francisco", "avg_fare": 20.25}
  ],
  "top_by_minutes": [
    {"city": "San Francisco", "minutes": 28, "fare": 29.3},
    {"city": "New York", "minutes": 26, "fare": 27.1},
    {"city": "Charlotte", "minutes": 21, "fare": 20.0},
    {"city": "Charlotte", "minutes": 12, "fare": 12.5},
    {"city": "San Francisco", "minutes": 11, "fare": 11.2},
    {"city": "New York", "minutes": 9, "fare": 10.9}
  ]
}
```

Container Status:

- Database container: Started successfully and initialized with sample data
- Application container: Connected to database, executed queries, and exited with code 0

Contents of out/summary.json

File Location: `out/summary.json`


```
{
  "total_trips": 6,
  "avg_fare_by_city": [
    {"city": "Charlotte", "avg_fare": 16.25},
    {"city": "New York", "avg_fare": 19.0},
    {"city": "San Francisco", "avg_fare": 20.25}
  ],
  "top_by_minutes": [
    {"city": "San Francisco", "minutes": 28, "fare": 29.3},
    {"city": "New York", "minutes": 26, "fare": 27.1},
    {"city": "Charlotte", "minutes": 21, "fare": 20.0},
    {"city": "Charlotte", "minutes": 12, "fare": 12.5},
```

```
    {"city": "San Francisco", "minutes": 11, "fare": 11.2},
    {"city": "New York", "minutes": 9, "fare": 10.9}
  ]
}
```


File Verification:

- File created successfully in mounted volume
- JSON format is valid and well-structured
- Data matches console output exactly

Reflection

Through this assignment, I learned the fundamentals of multi-container Docker applications and how services communicate within a Docker Compose environment. The most valuable lesson was understanding how to coordinate container dependencies using health checks and the `depends_on` configuration, ensuring the database is fully initialized before the application attempts to connect. I also gained practical experience with Docker networking, environment variable management, and volume mounting for persistent data storage.

The debugging process taught me how to effectively use Docker logs and container inspection tools to troubleshoot connectivity issues. If I were to improve this project, I would add more robust error handling in the Python application, implement connection pooling for better database performance, and create a more comprehensive test suite with additional sample data to validate the statistical calculations under different scenarios.

Technical Summary

Architecture: Two-container stack with PostgreSQL database and Python analytics service
Database: PostgreSQL 16 with automatic initialization via init.sql
Application: Python 3.11 with psycopg database connector
Orchestration: Docker Compose with health checks and service dependencies
Output: Console JSON display and persistent file storage

Repository URL: [purva115/ITCS6190assignment1: Assignment #1: Docker Containers]