Health Informatics Group Project: Athletics Performance Analytics

Assignment Overview

Your team will work with a real-world collegiate athletics database containing performance metrics from multiple tracking systems (Hawkins force plates, Kinexon GPS/accelerometry, and Vald strength testing). This multi-part group assignment will challenge you to extract insights from health/performance data using Python and SQL.

Learning Objectives:

- Connect to and query relational databases using Python
- Clean and transform data
- Perform exploratory data analysis on longitudinal (repeat) performance metrics
- Create data visualizations to communicate findings
- Work collaboratively

Database Overview

The athletics database contains performance data from multiple tracking systems integrated into a single unified table.

Main Table:

`research_experiment_refactor_test` - Single unified table containing all performance metrics from three data sources:

- Hawkins (force plates)
- Kinexon (GPS/accelerometry)
- Vald (strength testing)

Table Schema:

Column	Type	Description
id	BIGINT	Unique record identifier (auto-increment)
playername	VARCHAR(255)	Anonymized player identifier (e.g., PLAYER_001, PLAYER_002)
timestamp	DATETIME	Date and time of the measurement/session
device	VARCHAR(50)	Specific device/equipment used for measurement
metric	VARCHAR(255)	Name of the performance metric being measured

Column	Type	Description
value	DECIMAL(20,6)	Numeric value of the metric
team	VARCHAR(255)	Sport/team affiliation (e.g., Football, Soccer, Basketball)
session_type	VARCHAR(255)	Type of session (e.g., Practice, Game, Training) - only relevant for Kinexon
session_description	TEXT	Detailed description of the session
function_description	VARCHAR(255)	Movement or exercise description
data_source	VARCHAR(50)	Original data source (Hawkins, Kinexon, or Vald)
created_at	TIMESTAMP	Record creation timestamp

Key Features:

- **Long Format Data**: Each row represents one metric measurement for one player at one point in time
- **Anonymized Players**: All player names are replaced with codes (PLAYER_001, etc.) to protect privacy
- **Multi-Source Integration**: Combines data from three different sports performance systems
- **Longitudinal Data**: Multiple measurements per player over time enable trend analysis

Understanding the Data Structure:

This table uses **long format** (also called "narrow" or "tidy" format), which means:

- Each metric gets its own row
- A single player test session may generate dozens or hundreds of rows
- Example: If PLAYER_001 does a jump test that measures 5 different metrics, there will be 5 rows with the same timestamp

Example:

```
playername   | timestamp           | device  | metric           | value
| team
-------------|---------------------|---------|------------------|-------
-|----------
PLAYER_001   | 2024-09-15 10:30:00 | Hawkins | mRSI             | 2.45
| Football
PLAYER_001   | 2024-09-15 10:30:00 | Hawkins | Jump Height(m)   | 0.52
| Football
PLAYER_001   | 2024-09-15 10:30:00 | Hawkins | Peak Force(N)    | 2350.5
| Football
PLAYER_002   | 2024-09-15 10:35:00 | Hawkins | mRSI             | 2.12
| Football
```

Why long format?

- Efficient storage when you have many metrics
- Flexible - easy to add new metrics without changing table structure
- Common in healthcare/research databases
- You'll practice transforming this to "wide format" for analysis

---

Part 1: Database Connection & Data Exploration

Objectives:

- Establish database connection
- Understand data structure and quality
- Generate summary statistics

Tasks:

1.1 Database Setup (Individual)

Each team member should:

- Set up Python environment with required libraries (`sqlalchemy`, `pandas`, `pymysql`)
- Successfully connect to the database (credentials will be provided)
- Test connection by querying the `research_experiment_refactor_test` table

Deliverable: Screenshot showing successful connection and first 10 rows from the table

1.2 Data Quality Assessment (Group)

As a team, create a Python script that answers:

1. How many unique athletes are in the database?
2. How many different sports/teams are represented?
3. What is the date range of available data?
4. Which data source (Hawkins/Kinexon/Vald) has the most records?
5. Are there any athletes with missing or invalid names?
6. How many athletes have data from multiple sources (2 or 3 systems)?

Deliverable: Python script (`part1_exploration.py`) with documented code and a 1-page summary report

1.3 Metric Discovery & Selection (Group)

Create a query/script that:

- Lists the top 10 most common metrics for Hawkins data (filter by `data_source = 'Hawkins'`)
- Lists the top 10 most common metrics for Kinexon data (filter by `data_source = 'Kinexon'`)
- Lists the top 10 most common metrics for Vald data (filter by `data_source = 'Vald'`)
- Identifies how many unique metrics exist across all data sources

- For each data source, show the date range and record count for the top metrics

Deliverable: Add to `part1_exploration.py` with printed output

1.4 Brief Review Of Literature & Metric Selection (Group)

Based on your metric discovery, conduct a very brief literature review (does not need to follow and abide by PRISMA framework due to limited time allowed):

1. **Select 5 metrics** that are well-represented in your data (good sample size, multiple teams)
2. **Research each metric** using Google Scholar, PubMed, or sports science databases:
    - What does this metric measure?
    - Why is it important for athletic performance?
    - What are normal/elite values?
    - What does existing research say about this metric?
3. **Identify a research gap or question**:
    - Is there limited research on this metric in certain sports?
    - Are there unexplored relationships (e.g., metric X vs. injury risk)?
    - Could you compare across teams/positions in novel ways?

Deliverable:

- Short report (2-3 pages) in `part1_literature_review.pdf` that includes:
    - Summary of the selected metrics and why you chose them
    - Brief review for each (3-5 key citations per metric)
    - Potential research question or hypothesis based on identified gaps
    - Justification for why your analysis will be valuable

Note: The rest of the assignment should focus on YOUR selected metrics, not necessarily mRSI (which is used as an example in the instructions).

Part 2: Data Cleaning & Transformation

IMPORTANT: From this point forward, all analyses should focus on **YOUR selected metrics** from Part 1. The examples that mention specific metrics (like mRSI) are just illustrations - replace them with your chosen metrics.

Objectives:

- Handle missing data and outliers for your selected metrics
- Transform data from long to wide format
- Create derived metrics based on your research question

Tasks:

2.1 Missing Data Analysis (Group)

Focusing on **your selected metrics** from Part 1:

1. Identify which of your selected metrics have the most NULL or zero values
2. For each sport/team, calculate what percentage of athletes have at least 5 measurements for your selected metrics
3. Identify athletes who haven't been tested in the last 6 months (for your selected metrics)
4. Determine if you have sufficient data to answer your research question

Deliverable: Python script (`part2_cleaning.py`) with summary statistics

2.2 Data Transformation Challenge (Group)

The data is in "long format" (one row per metric per timestamp). Transform it to "wide format" for easier analysis.

Create a function that:

- Takes a player name (e.g., "PLAYER_001") and your selected metrics as input
- Returns a pandas DataFrame with:
  - Columns: timestamp, [your selected metrics]
  - One row per test session
  - Properly handles missing values

Test your function on at least 3 different athletes from different teams.

Deliverable: Add transformation function to `part2_cleaning.py` with example outputs

2.3 Create a Derived Metric (Group)

Athletes are often compared to team averages. Using **your selected metric(s)**, create code that:

1. Calculates the mean value for each team (using the `team` column)
2. For each athlete measurement, calculates their percent difference from their team's average
3. Identifies the top 5 and bottom 5 performers relative to their team mean
4. Optional: Create z-scores or percentile rankings
   - Z-score tutorial: SciPy stats.zscore
   - Percentile tutorial: NumPy percentile
   - Example: Calculating z-scores in pandas

Deliverable: Extend `part2_cleaning.py` with this analysis

Part 3: Longitudinal Analysis & Visualization

REMINDER: Continue using **YOUR selected metrics** and relate all findings back to your literature review and research question.

Objectives:

- Analyze trends over time for your selected metrics
- Create meaningful visualizations that support your research question
- Identify performance patterns and compare to published literature

Tasks:

3.1 Individual Athlete Timeline (Pair Work)

Select 2 athletes from a team of your choice and **use your selected metrics**:

1. Create line plots showing their metric values over time (recommended: last 6-12 months)
2. Identify their best and worst performance dates
3. Calculate if they show improvement or decline trend (simple linear regression acceptable)
4. Relate your findings to your literature review - are the trends expected? Surprising?

Deliverable: Jupyter notebook (`part3_viz_individual.ipynb`) with plots and interpretation

3.2 Team Comparison Analysis (Pair Work)

Compare two different teams/sports using the `team` column and **your selected metrics**:

1. Create box plots or violin plots comparing your selected metric(s) between teams
2. Calculate statistical significance (t-test or ANOVA as appropriate)
3. Create a visualization showing testing frequency by team over time
4. Interpret results in context of your literature review:
    - Do differences make sense given sport demands?
    - How do values compare to published norms (if available)?
    - What might explain the differences or similarities?

Deliverable: Jupyter notebook (`part3_viz_comparison.ipynb`) with plots and statistical tests

3.3 Dashboard Metric (Full Group)

Create a summary visualization that shows:

- Total number of tests per month (all systems combined)
- Breakdown by data source (stacked bar chart recommended)
- Identify any gaps or unusual patterns in data collection

Deliverable: Add to `part3_viz_comparison.ipynb`

Part 4: Research Synthesis & Application

REMINDER: Your deliverables should demonstrate how your analysis addresses the research gap you identified in Part 1.

Objectives:

- Create evidence-based performance monitoring tools
- Synthesize findings into a research report
- Communicate practical implications for coaches/trainers
- Present findings professionally

Tasks:

4.1 Performance Monitoring Flag System (Group)

Design a flagging system based on **your selected metrics** and literature review:

Your Task:

1. Based on your literature review, define 2-3 clinically/performance-relevant thresholds:
    - Examples: metric declined by X% compared to baseline
    - Metric below/above published risk threshold
    - Athlete hasn't been tested in >30 days
    - Left/right asymmetry if using bilateral metrics
    - Deviation from team norms
2. Justify your thresholds using evidence from your literature review
3. Create a script that identifies athletes meeting your flag criteria
4. Output a CSV with: playername, team, flag reason, metric value, last test date

Deliverable:

- Python script (`part4_flags.py`)
- Output CSV (`part4_flagged_athletes.csv`)
- Brief justification document (1 page) explaining your thresholds

4.2 Research Synthesis & Recommendations (Group)

Synthesize your findings into a research report (3-4 pages) that includes:

1. **Introduction**: Your research question and why it matters (based on literature gaps)
2. **Methods**:
    - Description of your selected metrics
    - Data filtering and cleaning approach
    - Statistical methods used
3. **Results**:
    - Key findings from your analyses
    - Tables and figures with appropriate captions
    - Statistical test results
4. **Discussion**:
    - How do your findings relate to existing literature?
    - What gaps did your analysis address?
    - What are the practical implications for coaches/trainers?
    - What surprised you? What confirmed existing knowledge?
5. **Limitations & Future Directions**:
    - Data limitations (missing values, sample size, etc.)
    - What additional data would be helpful?
    - Recommendations for future research
6. **References**: Full citations for your literature review

Deliverable: PDF report (`part4_research_synthesis.pdf`)

4.3 Final Presentation (Group)

Create a 10-12 minute presentation covering:

1. **Introduction** (2 min):
    - Your research question/hypothesis
    - Why it matters (the gap you're addressing)
    - Your selected metrics and why
2. **Methods** (2 min):
    - Data overview and quality assessment
    - Analysis approach
3. **Key Findings** (4 min):
    - Main results with visualizations
    - Statistical findings
    - Comparison to literature
4. **Practical Applications** (2 min):
    - Your performance monitoring flag system
    - Recommendations for coaches/trainers
    - How your findings fill the identified gap
5. **Limitations & Future Work** (1 min):
    - Data challenges you encountered
    - What additional research is needed
6. **Q&A** (1-2 min)

Deliverable: Presentation slides (PowerPoint/PDF) uploaded to your GitHub repo

---

Technical Requirements

Required Python Libraries:

```
import pandas as pd
import sqlalchemy
from sqlalchemy import create_engine, text
import pymysql
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from scipy import stats
```

Database Connection Template:

```
from sqlalchemy import create_engine
import pandas as pd

# Connection string (credentials will be provided)
engine = create_engine(
    "mysql+pymysql://username:password@host:port/database_name"
)

# Example query
query = "SELECT * FROM research_experiment_refactor_test LIMIT 10"
df = pd.read_sql(query, engine)

# Close connection when done
engine.dispose()
```

Code Quality Expectations:

- Use meaningful variable names
- Comment complex logic
- Don't hardcode secret/database connection string values - use variables/parameters that load from .env files

---

Submission Guidelines

GitHub Repository Setup (Required)

IMPORTANT: All work must be submitted via GitHub.

1. **Repository Name**: `507_groupproject_2025`
2. **Owner**: One team member creates the repository
3. **Collaborators**: Owner must add:
    - All team members with **Admin** rights
    - Instructor: `hantswilliams` with **Admin** rights
4. **Repository Settings**:
    - Public
    - Must include a comprehensive README.md

To add collaborators:

1. Go to your repository on GitHub
2. Click `Settings` → `Collaborators and teams`
3. Click `Add people`
4. Search for GitHub username

5. Select role: `Admin`
6. Send invitation

Required collaborators:

- All team members
- `hantswilliams` (instructor)

File Structure:

```
507_groupproject_2025/
├── README.md (with group member names, roles, and contributions)
├── references.md (full bibliography in APA or similar format)
├── .env.example (template for database credentials - DO NOT include
actual credentials)
├── .gitignore (exclude .env, data files, etc.)
├── part1_exploration.py
├── part1_summary.pdf
├── part1_literature_review.pdf (NEW - your metric selection and lit
review)
├── part2_cleaning.py
├── part3_viz_individual.ipynb
├── part3_viz_comparison.ipynb
├── part4_flags.py
├── part4_flagged_athletes.csv
├── part4_flag_justification.pdf (NEW - explain your thresholds)
├── part4_research_synthesis.pdf (NEW - replaces sport_analysis.pdf)
└── final_presentation.pdf
```

Grading Rubric (100 points):

Component	Points	Details
Part 1: Exploration & Literature Review	25	Data exploration (10), Metric selection & lit review (10), Research question (5)
Part 2: Cleaning & Transformation	20	Missing data analysis (8), Transformation (7), Derived metrics (5)
Part 3: Visualization & Analysis	20	Individual timeline (7), Team comparison (8), Statistical rigor (5)
Part 4: Synthesis & Application	25	Flag system with justification (10), Research synthesis report (15)
Code Quality & Documentation	5	Clean code, comments, GitHub organization, reproducibility
Presentation	5	Clarity, professionalism, time management, Q&A responses

Component	Points	Details
TOTAL	**100**	

Important Notes:

- **GitHub Repository:** You MUST create a repository named `507_groupproject_2025` and add `hantswilliams` as a collaborator with Admin rights. Failure to do so will result in point deductions.
- **Data Privacy:** All athlete identifiers are anonymized (PLAYER_001, etc.). Do not attempt to identify individuals. This data is for research/educational purposes only.
- **Collaboration:** Work is expected to be collaborative, but each member should contribute code. When you submit to github, I should see multiple collaborators with their own commits. I do not want to see one single contributor to the repo.
- **Resources:** You may use any resources, but please cite any external code/algorithms used in a markdown file called `references.md`
- **Security:** Never commit database credentials or `.env` files to GitHub. Use `.gitignore` to exclude sensitive files.

Helpful Hints

Setting Up Your GitHub Repository

1. **Create .gitignore file** (to protect sensitive data):

```
# Environment variables and credentials
.env
*.env

# Python
__pycache__/
*.py[cod]
*$py.class
*.so

# Jupyter Notebook
.ipynb_checkpoints
*.ipynb_checkpoints/

# Data files (if you download data locally)
*.csv
*.pkl
*.xlsx
!*_example.csv  # Allow example files

# IDEs
.vscode/
.idea/
```

```
*.swp
*.swo

# OS
.DS_Store
Thumbs.db
```

2. **Create .env.example file** (template for teammates):

```
# Database Configuration
# Copy this file to .env and fill in your actual credentials
DB_HOST=your_database_host
DB_USER=your_username
DB_PASSWORD=your_password
DB_NAME=database_name
DB_TABLE=research_experiment_refactor_test
```

3. **README.md should include**:

- Project title and description
- Team member names and roles
- Setup instructions (how to install dependencies)
- How to run each script
- Database connection instructions
- Project structure overview

Common Challenges & Solutions:

1. **Challenge:** Timestamp is stored as DATETIME but may need addition formatting
    - **Solution:** Use `pd.to_datetime()`
2. **Challenge:** Some metrics have parentheses and special characters
    - **Solution:** Use backticks in SQL: `` `Jump Height(m)` `` or use `metric = 'Jump Height(m)'` in WHERE clause
3. **Challenge:** Data in long format is hard to analyze
    - **Solution:** Use `pandas.pivot_table()` or `pandas.pivot()` to transform to wide format
4. **Challenge:** Too much data to load at once
    - **Solution:** Filter by team, data_source, or date range in SQL query first

```
query = """
SELECT * FROM research_experiment_refactor_test
WHERE team = 'Football'
```

```
AND timestamp >= '2024-01-01'
"""
```

5. **Challenge:** Values stored as DECIMAL but may have NULLs
    - **Solution:** Use `pd.to_numeric(df['value'], errors='coerce')` or `df['value'].fillna(0)`
6. **Challenge:** Filtering by data source
    - **Solution:** Use the `data_source` column:

```
hawkins_data = df[df['data_source'] == 'Hawkins']
kinexon_data = df[df['data_source'] == 'Kinexon']
vald_data = df[df['data_source'] == 'Vald']
```

Sample Queries to Get Started:

Example 1: Explore available metrics by data source

```
-- See what metrics are available from each system
SELECT data_source, metric, COUNT(*) as record_count
FROM research_experiment_refactor_test
GROUP BY data_source, metric
ORDER BY data_source, record_count DESC;
```

Example 2: Find metrics with good sample sizes

```
-- Find metrics that have been measured frequently
SELECT metric,
       COUNT(DISTINCT playername) as num_athletes,
       COUNT(*) as total_measurements,
       MIN(timestamp) as earliest_date,
       MAX(timestamp) as latest_date
FROM research_experiment_refactor_test
WHERE value IS NOT NULL
GROUP BY metric
HAVING num_athletes >= 10  -- At least 10 different athletes
ORDER BY total_measurements DESC
LIMIT 20;
```

Example 3: Get data for YOUR selected metric

```sql
-- Replace 'YOUR_METRIC' with your chosen metric
SELECT playername, team, timestamp, metric, value, data_source
FROM research_experiment_refactor_test
WHERE metric = 'YOUR_METRIC'  -- e.g., 'Jump Height(m)', 'Peak
Force(N)', etc.
  AND value IS NOT NULL
ORDER BY playername, timestamp;
```

Example 4: Compare metric across teams

```sql
-- Replace 'YOUR_METRIC' with your chosen metric
SELECT team,
       COUNT(DISTINCT playername) as num_athletes,
       AVG(value) as mean_value,
       STDDEV(value) as std_value,
       MIN(value) as min_value,
       MAX(value) as max_value
FROM research_experiment_refactor_test
WHERE metric = 'YOUR_METRIC'
  AND team IS NOT NULL
  AND value IS NOT NULL
GROUP BY team
ORDER BY mean_value DESC;
```

Example 5: Get longitudinal data for one athlete

```sql
SELECT timestamp, metric, value, device, session_type
FROM research_experiment_refactor_test
WHERE playername = 'PLAYER_001'
  AND metric IN ('YOUR_METRIC_1', 'YOUR_METRIC_2')  -- Your selected
metrics
ORDER BY timestamp, metric;
```

Extension Opportunities (Optional - Extra Credit)

For groups wanting an additional challenge:

1. **Interactive Dashboard:** Create a simple web dashboard using Dash or Streamlit
2. **Advanced Statistics:** Implement mixed-effects models to account for individual repeated measures

---

Resources that you might find helpful

Literature Search Resources:

Databases for Sports Science & Performance:

- Google Scholar - Broad academic search
- PubMed - Biomedical and health sciences
- SPORTDiscus - Sports & sports medicine (via library)
- ResearchGate - Access to researchers and papers

Useful Search Terms:

- Your metric name + "athletic performance"
- Your metric + "reliability" or "validity"
- Your metric + specific sport (e.g., "football", "soccer")
- "Force plate assessment" / "GPS tracking" / "Strength testing"
- "Injury risk" + your metric
- "Normative values" + your metric

Key Journals:

- Journal of Strength and Conditioning Research
- Sports Medicine
- International Journal of Sports Physiology and Performance
- British Journal of Sports Medicine
- Medicine & Science in Sports & Exercise

Data Visualization:

- Matplotlib Tutorials
- Seaborn Gallery

Statistics:

- SciPy Stats Module
- Statistics How To - Plain language stats explanations