

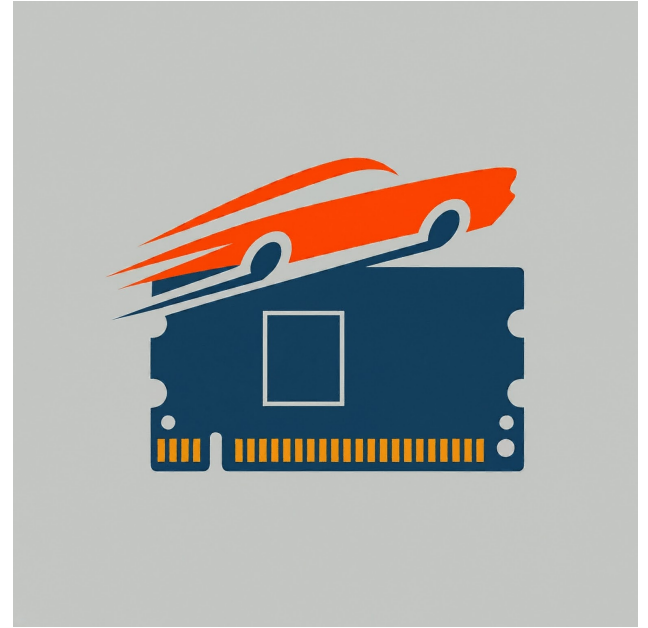
Project Report

Team Name:

char_junkies

Members:

- Adriean Lemoine
- Chris Smith
- Nicholas Burgo



GitHub: https://github.com/arlemoine/CS340_F_24_char_junkies

Goal of the Project

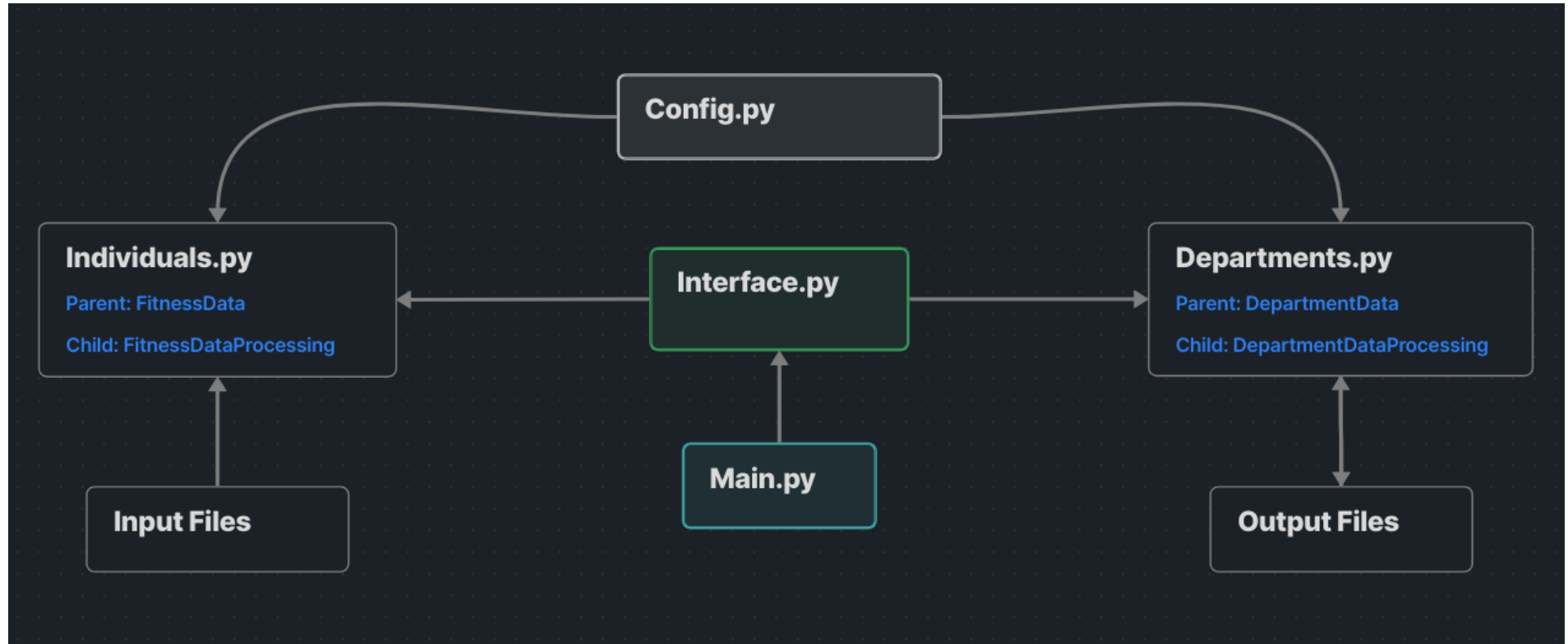
It is believed that the methods of ensuring police officer fitness level are inadequate to guarantee a proper police force. More specifically, it is not enough to give a fitness test to each officer once a year, as an annual fitness test does not guarantee an active lifestyle and overall indicator of officer health.

We attempt to solve this problem by the utilization of **Fitbits** in the department. Each officer has their **steps**, **heart rate variability** (HRV), and **heart rate** continuously monitored and assessed to calculate a **fitness score**. The heart rate is used to clean the heart rate variability data. The average steps and HRV are used to calculate a fitness score for each officer.

The fitness score determines whether or not the officer meets fitness standards, lands in probation, or fails fitness standards. We can also examine the officer's data as well as graphs to visualize this data, as the data and graphs are generated automatically.

This information can be used to determine how far from the fitness standard each officer is. It can also be used to project future fitness statistics, including the likelihood that a new officer would meet the fitness standards.

Module Communication Flow



Class Diagrams: Individuals.py

Individuals

FitnessData

Attributes:

- config: dict
- name: string
- age: int
- df_steps: DataFrame
- df_hrv: DataFrame
- departureAngle: float

Methods

- initLog()
- view_table()
- view_steps_table()
- view_hrv_table()
- get_line_graph()
- get_steps_line_graph()
- get_hrv_line_graph()
- data_search()
- print()

FitnessDataProcessing

Attributes

- dirOut: string
- step_score: int
- hrv_score: int
- fitness_score: int

Methods

- writeOutputs()
- updateDirectory()
- importAll()
- importAge()
- importSteps()
- importHrv()
- calc_call()
- calc_step_score()
- calc_hrv_score()
- calc_fitness_score()
- writeAllToFile()
- genPlots()
- get_violin_plot()
- query_data()
- show_stats_for_month()

Class Diagrams: Departments.py

Departments

DepartmentData

Attributes

- departmentName: string
- individuals: dict
- daysInMonth: int
- df_dept_steps: DataFrame
- df_dept_hrv: DataFrame
- df_dept_fitness_scores: DataFrame
- df_dept_age: DataFrame
- logging: Logger obj

Methods

- initLog()
- addIndividual()
- dropIndividual()
- calcDays()
- getDataFrames()
- getSteps()
- getHRV()
- getFitnessScores()
- getAges()

DepartmentDataProcessing

Attributes

- config: dict
- dirPickle: string
- df_stats_steps: DataFrame
- df_stats_hrv: DataFrame
- df_stats_fitness_score: DataFrame
- df_stats_age: DataFrame
- df_jointCounts: DataFrame
- df_jointProbs: DataFrame
- totalCounts: int
- condProbs: dict
- uniqueAgeGroup: List
- uniqueFitnessGroup: List
- combAgeGroup: List
- combFitnessGroup: List
- permAgeGroup: List
- permFitnessGroup: List

Methods

- getAll()
- updateDirectory()
- getStats_all()
- getStats_steps()
- getStats_hrv()
- getStats_fitness_score()
- getStats_age()
- gen_dept_hist()
- getJointCounts()
- getJointProbs()
- getCondProbs()
- getUniqueValues()
- gen_dept_vectors()
- gen_vectors()
- writeStats()
- pickleSave()
- pickleLoad()
- writeOutputs()
- printIndividuals()

Class Diagrams:

Data Formats and Other

Input/

- individualName/
 - age.csv
 - steps.csv
 - hrv/
 - hrvDD.csv
(where DD = day of month)

Output

- departmentName/
 - departmentName.log
 - departmentName.pkl
 - data.txt
 - histograms.png
- individualName/
 - individualName.log
 - data.txt
 - lineGraphs.png
 - violinPlots.png
 - vectorPlots.png

Configurations

Constants

- DIR_INPUT
- DIR_OUTPUT
- FILENAME_STEPS
- FILENAME_AGE
- MAX_HEART_RATE
- MAX_DEPARTMENT_SIZE
- STEP_WEIGHT
- HRV_WEIGHT
- MIN_FITNESS_SCORE
- PROB_FITNESS_SCORE

UI

Coordinates user input with other modules

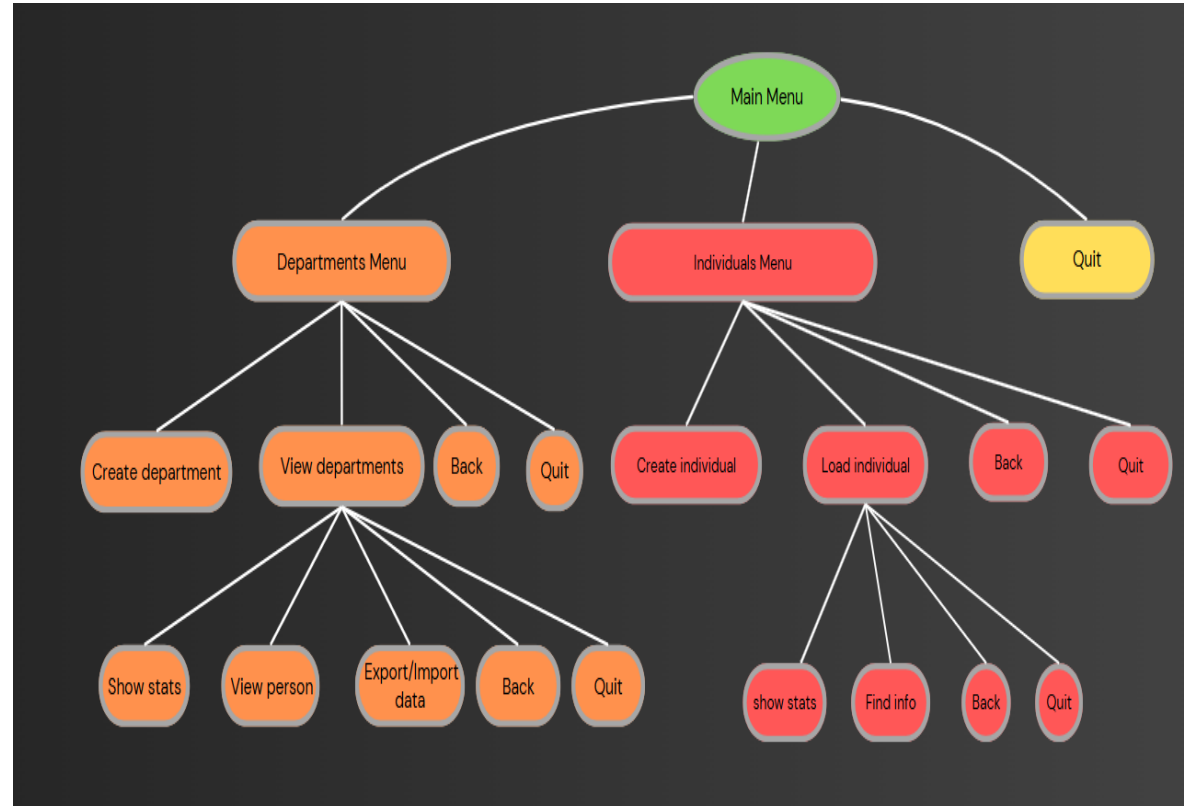
Main

Directs user to the UI module

User Manual

Main Menu

1. Individuals - Access the **Individuals Menu** to create or manage individual profiles.
2. Departments - Access the **Departments Menu** to create or manage department data.
3. Quit - Exits the application.



User Manual

Individual Menu

1. Create Individual - *Use this option to create a new individual profile.*
2. Load Individual - *Lists all created individuals. Goes to the sub menu*
3. Back - *Returns to the *Main Menu*.*
4. Quit - *Exits the application.*

Individual Sub-Menu (intfInd3)

1. Show Stats - *Displays monthly statistics for the selected individual.*
2. Find Info - *Allows queries to find specific information.*
3. Back - *Returns to the *Individuals Menu*.*
4. Quit - *Exits the application.*

User Manual

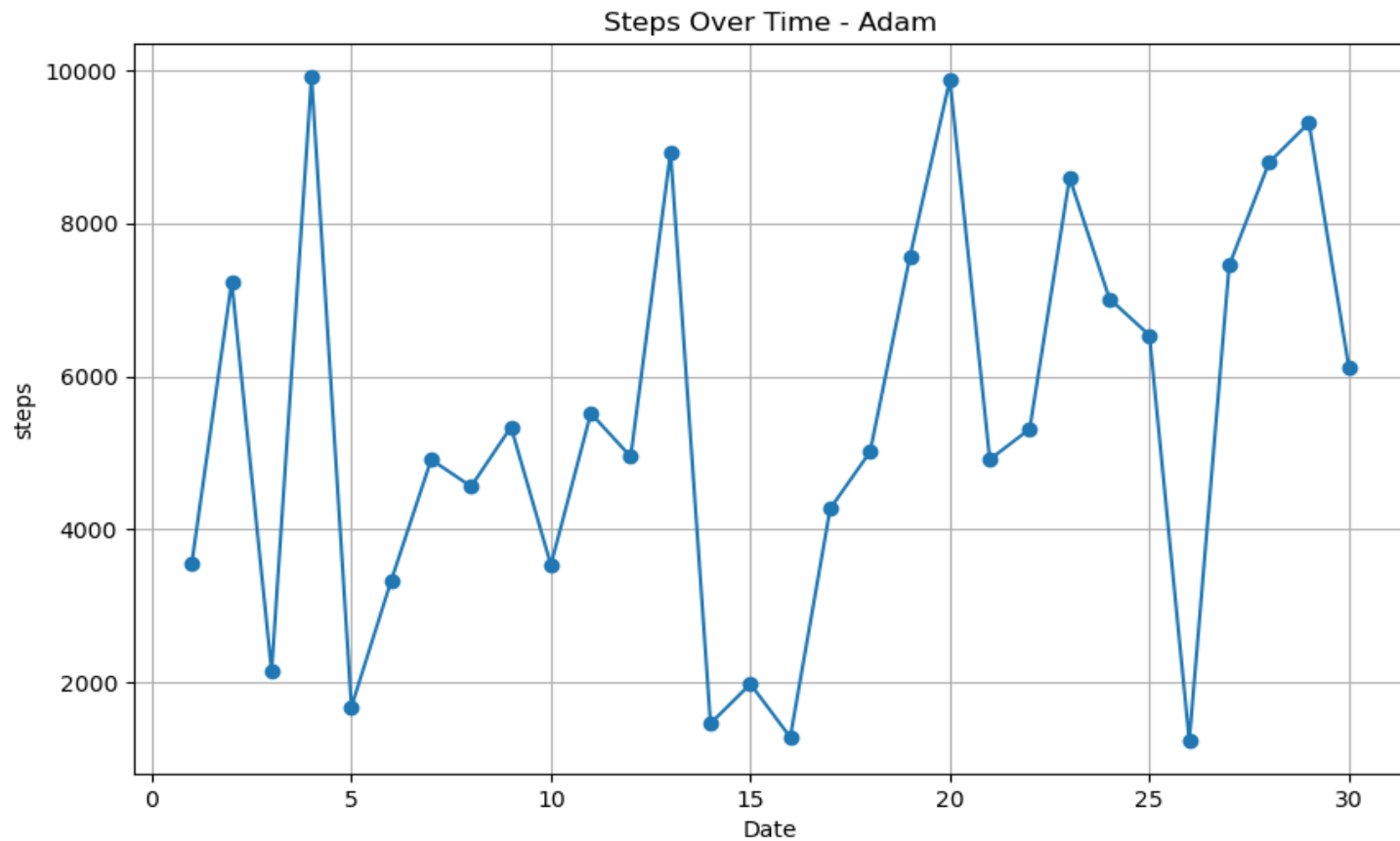
Department Menu

1. Create Department – *Use this option to create a new department.*
2. View Departments - *Lists all created departments. Directs you to the *Department Sub-Menu*.*
3. Back - *Returns to the *Main Menu*.*
4. Quit- *Exits the application.*

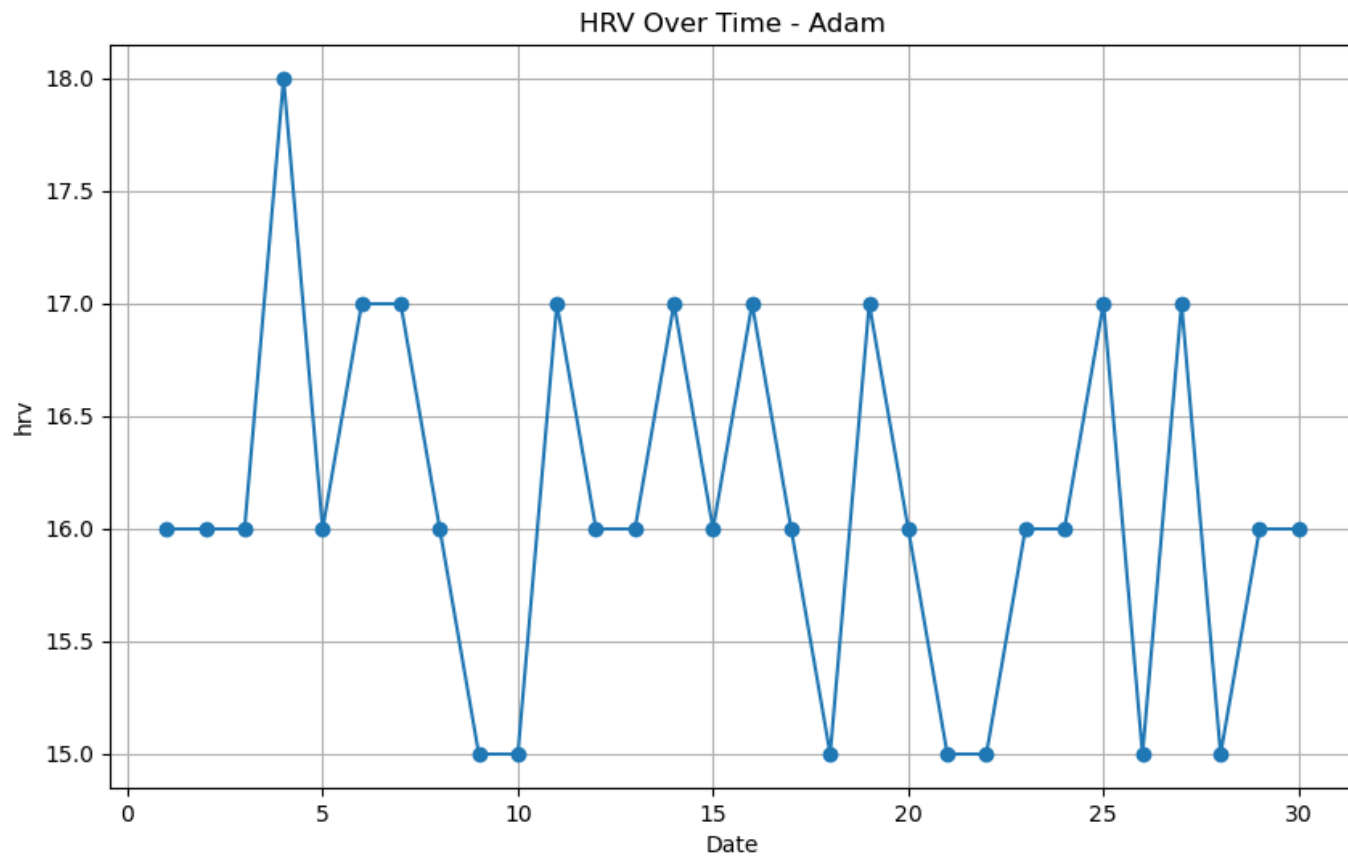
Department Sub-Menu (intfDept3)

1. Show Stats - *Displays department statistics.*
2. View Person – *Navigates to the * Individual Sub-Menu*.*
3. Export/Import Data - *Provides options for saving or loading department data using Pickle.*
4. Back - *Returns to the *Departments Menu*.*
5. Quit - *Exits the application.*

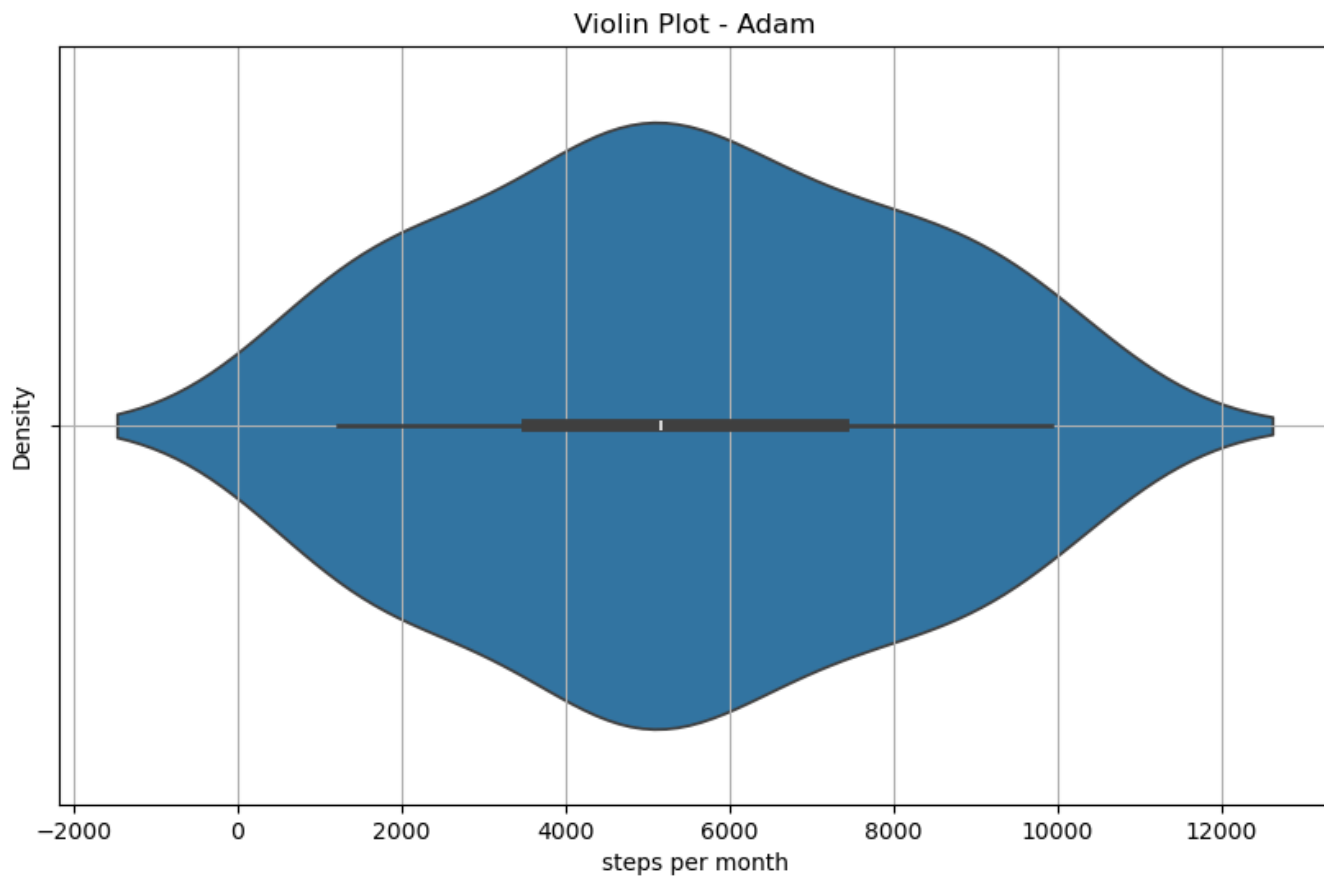
Plots: Line



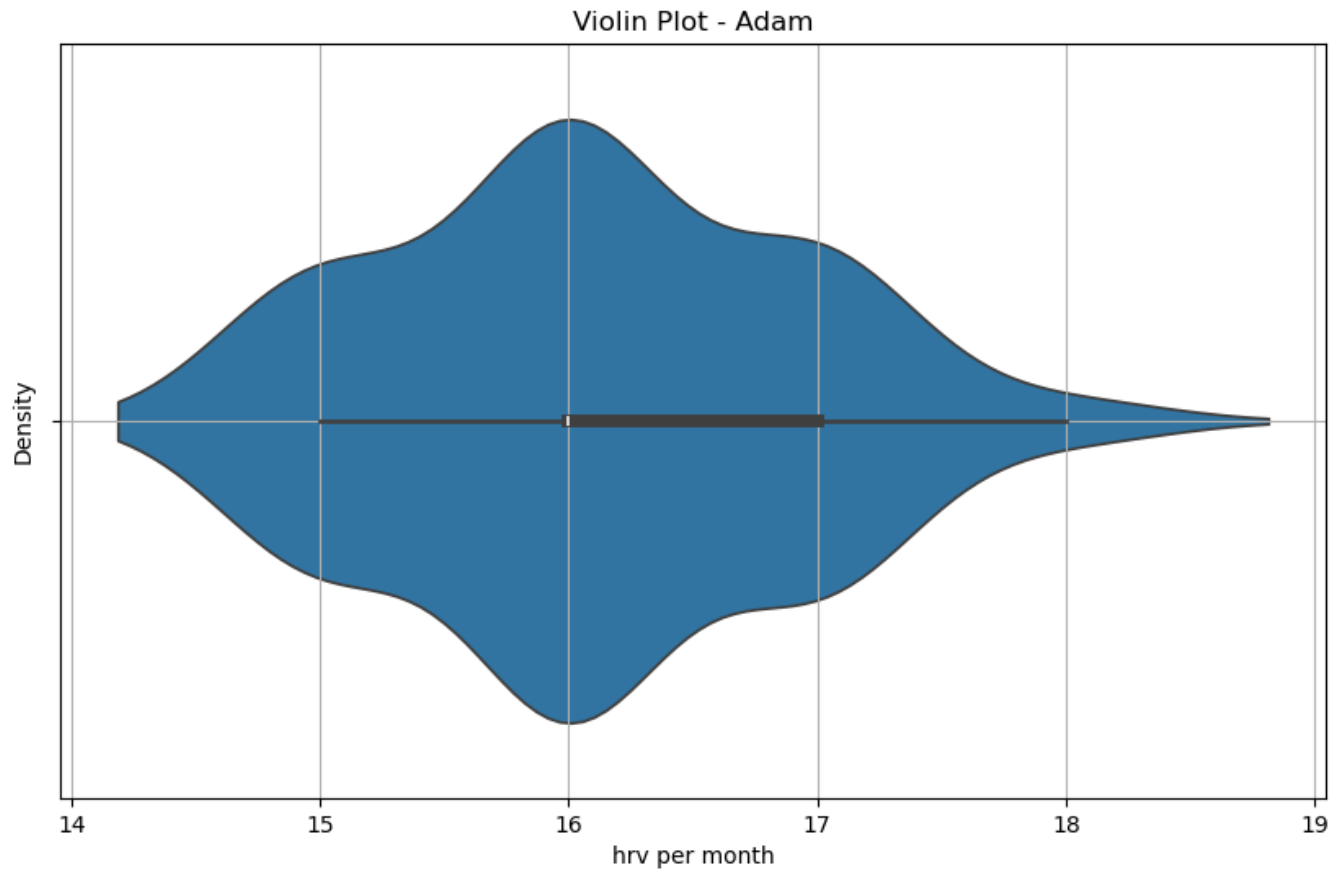
Plots: Line



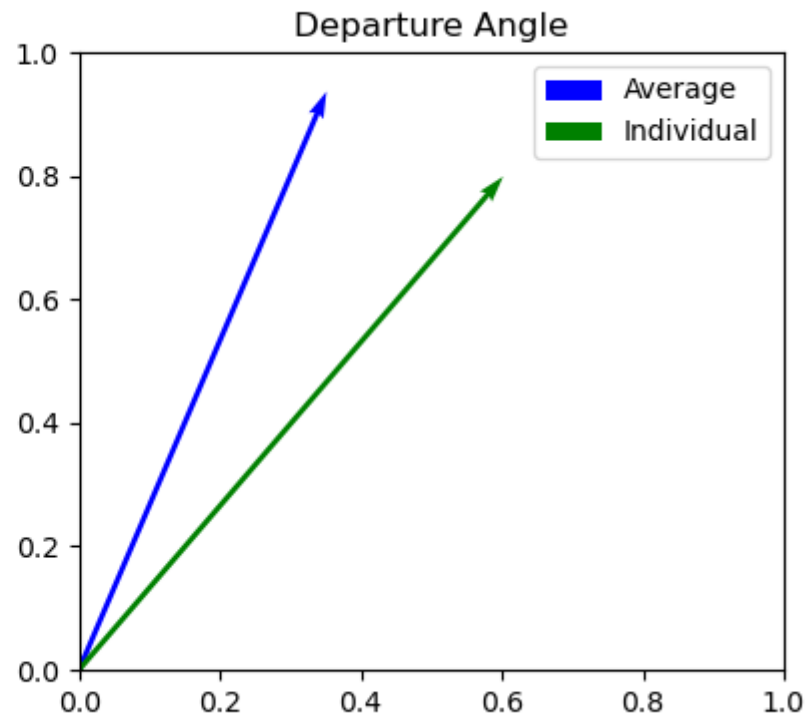
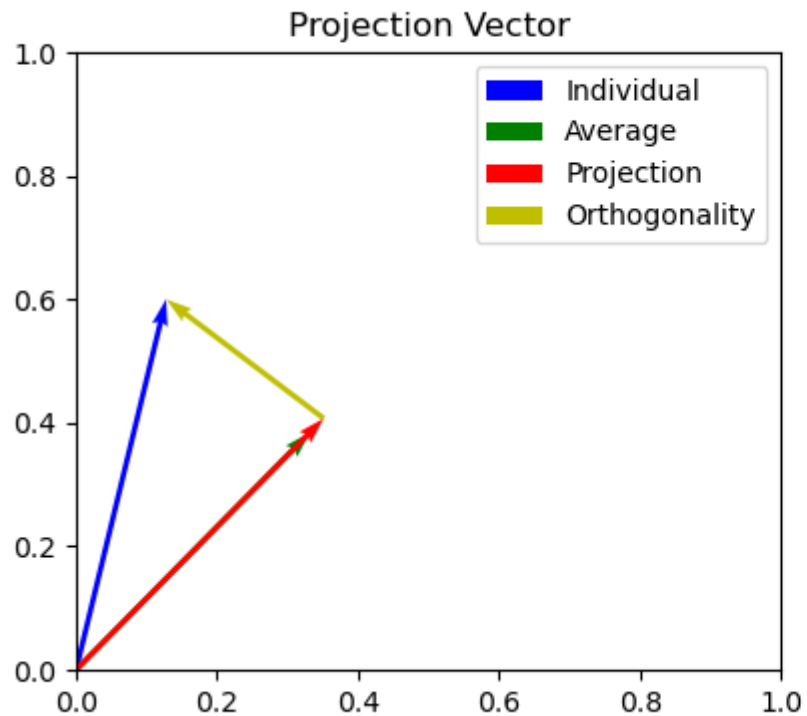
Plots: Violin



Plots: Violin

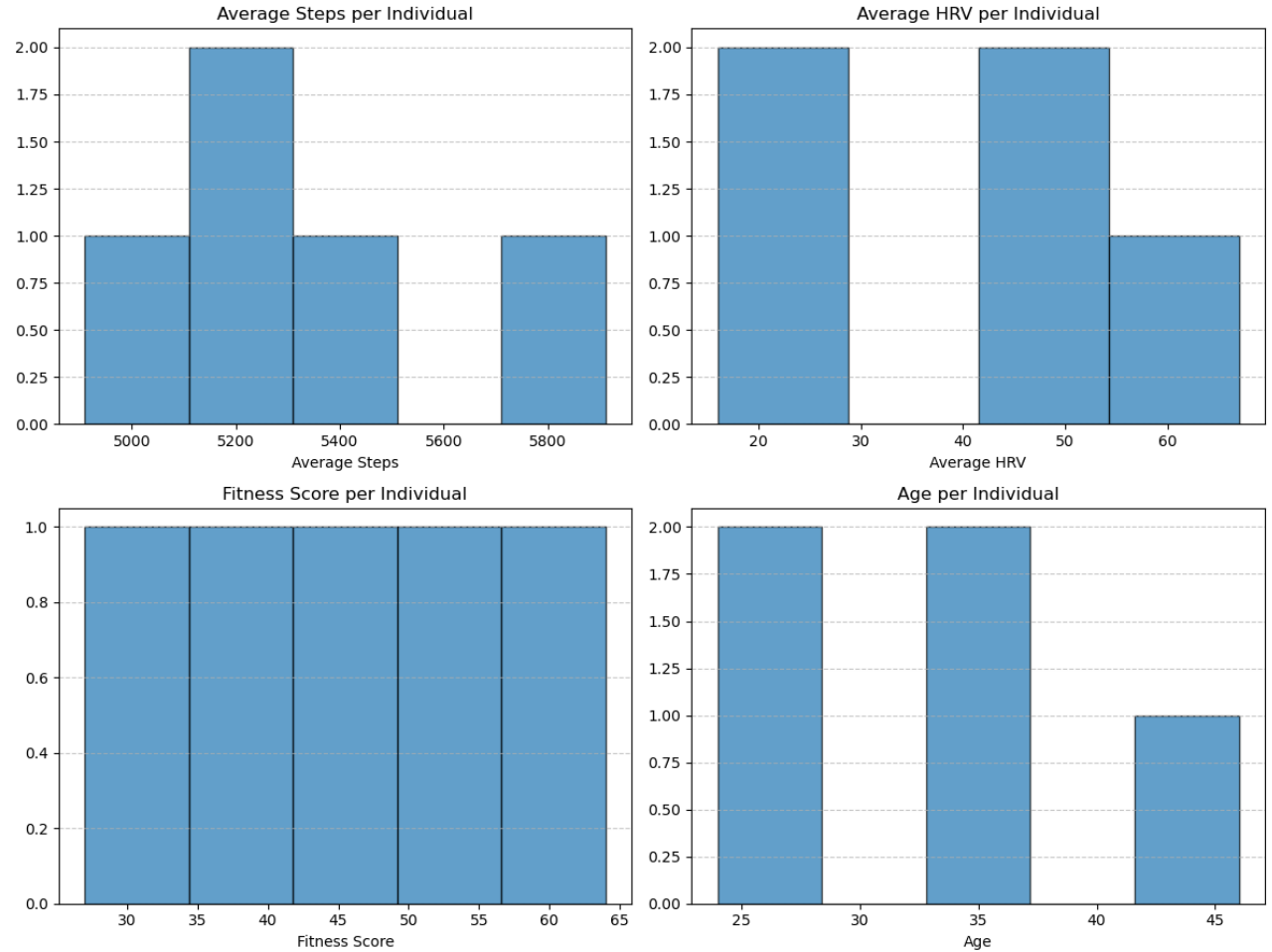


Plots: Vector



Plots: Histogram

Histograms for Department: CMPSPD 340



Team Issues

- None