Race Strategy Simulator

A Python-based simulation tool to evaluate optimal race strategies in Formula 1 based on tyre compounds, track conditions, pit stop timing, and safety events. The code uses statistical modelling and parameter-based degradation curves to assess and compare race strategies, including 1-stop, 2-stop, and 3-stop options.

Features

- Calculates optimal tyre strategy for any number of laps.
- Models tyre degradation, fuel effects, and warm-up periods.
- Includes weather modes: Dry, Intermediate, Wet.
- Supports safety conditions: VSC, Safety Car, and Red Flags.
- Takes into account tyre set limitations and compound usage rules.
- Strategy output includes total race time, pit stop laps, and visualized lap-by-lap performance.

Inputs

The user is prompted to enter:

- Total race laps (e.g. 70)
- Pit stop time (in seconds)
- Current weather condition (Dry, Intermediate, or Wet)
- Base lap time on **Soft** tyres
- Optional: Laps of Virtual Safety Car (VSC), Safety Car (SC), or Red Flag events

---

Assumptions & Simplifications

General

- **Single-car simulation**: There is no multi-car interaction, blocking, traffic, or overcut/undercut modelling.
- **No external track evolution**: The grip level of the circuit remains constant throughout the race.
- **Perfect execution**: Pit stops are always perfectly timed and executed (except time loss).

Tyres

- Tyres degrade per lap based on type and compound-specific degradation rates.
- Each tyre compound has a preset life, degradation, and warm-up behaviour.
- There’s a sharp "falloff" in performance after a certain % of tyre life, which varies by compound.
- No tyre wear crossover (e.g., no "slicks becoming inters" or vice versa).

Safety Events

- VSC adds a fixed lap-time adjustment.
- SC adds a larger time adjustment.
- Red Flag resets the lap impact to near-zero, simulating a race pause.

Drivers

- **No “pushing” or pace management**: Drivers run all stints at consistent effort, within the statistical variation baked in.
- **No driving style effect**: There’s no distinction between aggressive or conservative driving.
- **No mistake modelling**: Drivers do not lock up, crash, or spin.

Outputs

- Top 10 race strategies ranked by total race time
- Breakdown of each stint (compound and length)
- Pit stop laps
- Graph showing lap-by-lap times and average pace

Strategy Depth

- 1-stop, 2-stop, and 3-stop strategies are all considered.
- The simulator exhaustively searches all legal combinations within stint-length windows and tyre availability rules.

Setup & Run

Requirements

```
pip install numpy matplotlib
```

Run

```
python f1_strategy_simulator.py
```

Example

```
Enter total number of laps (e.g. 70): 66
Enter pit stop time in seconds (e.g. 20.0): 22.5
Enter current weather (Dry, Intermediate, Wet): Dry
Enter base lap time on Soft tyres in seconds (e.g. 85.0): 90.0

Now enter the laps for safety events (if any):
Enter laps for VSC events separated by commas (or leave blank for none): 14, 38
Enter laps for SC events separated by commas (or leave blank for none):
Enter laps for Red events separated by commas (or leave blank for none):
```

Then the simulator outputs the top 10 strategies and plots the lap-by-lap time chart.

Potential Future Add-ons

Driver Behavior & Strategy

- Driver "pushing" mode: Allow stints with aggressive or conservative driving that affects degradation and lap times.
- Driver skill profiles: Simulate differences between drivers in tyre management, pace, or consistency.
- Error modelling: Add probabilities for lock-ups, spins, or mistakes (with time penalties or retirements).
- AI-controlled opponents: Simulate races against competing strategies for more dynamic comparisons.

Tyre & Stint Modeling

- Dynamic degradation curves: Use non-linear degradation models or real-world data fitting.
- Crossover conditions: Model when slicks outperform inters or when wets are better than inters.
- Tyre temperature management: Simulate overheating, graining, or underheating effects.
- Track evolution: Adjust tyre performance as grip improves throughout the race.

Environmental Variables

- Weather transitions: Add rain onset or drying track over the race duration.
- Track temperature effects: Influence degradation and grip based on a temperature profile.
- Wind or humidity impact: Simulate minor influences on lap time or tyre wear.

Strategic Complexity

- Undercut/overcut modelling: Time gains from pitting earlier or later than rivals.
- Pit crew variation: Randomized pit stops execution times.
- Mechanical reliability: Add chance of DNF due to technical failures.
- Grid penalties: Factor in starting position or in-race penalties.

Usability & Outputs

- Save/export results: Save results in CSV/JSON for analysis.
- Interactive UI: Build a GUI or web-based interface (e.g., using Streamlit or React).
- Monte Carlo simulation: Run thousands of iterations to assess strategy reliability.
- Stint visualization enhancements: Color-coded strategy charts, tyre wear curves, etc.
- Lap-by-lap strategy table: Exportable stint-by-stint summaries.

Technical Enhancements

- Custom parameter profiles: Let users define and load tyre/fuel/event profiles per track.

- API integration: Pull real F1 data or share simulation results via API.
- Multithreading: Speed up strategy simulation using concurrent execution.
- Strategy optimization algorithm: Use genetic algorithms or heuristics for smarter search.