ChatGpt (methodology, reproducible)

1. **Data source**: I began from the uploaded `cume.pdf` (you previously provided). The PDF contains aggregated season totals and per-game scores. In the notebook I attempted PDF text extraction (best-effort) but fell back to the manual extraction of the summary numbers that are already present in the canvas/report (this is safer and reproducible because the summary values are explicitly listed).
2. **Manual data assembly**: Entered the per-game scores and aggregate season metrics as Python arrays/dictionary (these values are the same as shown in the canvas report). This avoids fragile PDF parsing and ensures exact reproduction of the numbers used in the report.
3. **Descriptive statistics**: Computed totals, means (PPG), sample standard deviations (ddof=1) for team and opponents, and the game-by-game differential vector (`team_pts - opp_pts`).
4. **Paired test / t-statistic**: Calculated the paired t-statistic manually using `t = mean(diff) / (sd(diff) / sqrt(n))`. (I did not rely on an external stats package for the t value; this yields the same t value reported in the canvas doc.)
5. **Bootstrap CI for mean differential**: Performed a nonparametric bootstrap of the *mean* game differential using `n_boot = 20,000` resamples and `random_seed = 42`. I computed the 2.5 and 97.5 percentiles of the bootstrap distribution to get the 95% CI.
6. **Sensitivity test**: Recomputed the team and opponent PPG and mean differential after removing the Colgate game (index 2) to show how an outlier changes the conclusions.
7. **Summary table**: Composed a tidy DataFrame of all derived metrics and raw key season numbers (rush yards, pass yards, turnovers, third-down & fourth-down stats, sacks, penalties, punt averages). Saved it as CSV to `/mnt/data/syracuse_summary_stats.csv`.
8. **Visualizations**: Plotted per-game scoring (Syracuse vs opponents) and the bootstrap histogram for the mean differential. Saved PNGs to `/mnt/data`. Each plot is a distinct matplotlib figure (no seaborn or custom color settings).
9. **Reproducibility**: Logged the random seed (42), saved the full notebook/script to `/mnt/data/reproducible_analysis.py`, and saved CSV/plots to `/mnt/data`. The code above is the exact script you can execute locally to reproduce everything.
10. **Auditability**: The canvas report includes the original LLM prompt and the edits. The executed code uses explicit manual inputs (so it is transparent and reproducible). If you prefer, I can add step-by-step git instructions and create a GitHub repo for you and draft the `jrstrome@syr.edu` email.

Files I created (download)

- Summary CSV: [Download summary CSV] (sandbox:/mnt/data/syracuse_summary_stats.csv)
- Per-game scoring chart PNG: [Download per-game chart] (sandbox:/mnt/data/per_game_scores.png)

- Bootstrap histogram PNG: [Download bootstrap histogram] (sandbox:/mnt/data/bootstrap_hist.png)
- Repro script (minimal): [Download reproducible_analysis.py] (sandbox:/mnt/data/reproducible_analysis.py)

Notes, assumptions, and next steps

- **Assumptions**: I used the summary numbers from the canvas doc directly. That is the clearest, auditable source for the summary stats in the assignment. If you want true play-level replication, we need the play-by-play CSVs or tracking files (then I will rerun the analysis with EPA, adjusted success rates, formation-level breakdowns, and per-player snap-normalized metrics).