**Take-Home Assessment: Inferential Data Analysis in Python**

Due Date: 24/11/2025

You can use this Kaggle dataset for multiple regression / inferential work:  
Student Performance (Multiple Linear Regression) — includes predictors for student performance. [Kaggle](https://www.kaggle.com/datasets/nikhil7280/student-performance-multiple-linear-regression?utm_source=chatgpt.com)

Alternatively, a House Prices / Regression / Income dataset works well. Eg: “Regression Dataset for Household Income Analysis” on Kaggle. [Kaggle](https://www.kaggle.com/datasets/stealthtechnologies/regression-dataset-for-household-income-analysis?utm_source=chatgpt.com)

**Project Tasks / Requirements**

Students must deliver a Python notebook (Jupyter) plus a formal report edited in latex(PDF). The notebook must be executable.

Below is a task breakdown. Each student must do all:

|  |  |
| --- | --- |
| **Task #** | **Task Description** |
| 1 | **Data ingestion & exploration** — load data, inspect structure, identify variable types, summarize data (mean, median, variance, quartiles). |
| 2 | **Assumption testing** — check normality (Shapiro-Wilk + Q-Q), detect heteroscedasticity, and test homogeneity of variance (if grouping exists). |
| 3 | **Data transformation & outlier handling** — apply log, sqrt, reciprocal, differences; handle outliers (trim / winsorize); compare before & after distributions (histograms, boxplots). |
| 4 | **Variable construction** — derive at least two new variables (e.g. ratios, interaction terms, indices), store appropriately, keep originals. |
| 5 | **Model fitting & selection** — fit multiple regression models (or ANOVA/sub-models), compare fit (R², adjusted R², AIC/BIC), select best. |
| 6 | **Hypothesis tests & confidence intervals** — for key coefficients or means / proportions, test hypotheses and compute CIs. |
| 7 | **Interpretation & reporting** — interpret parameter estimates, predictions, limitations, further research suggestions. |
| 8 | **Presentation & visualization** — produce well-labeled plots, summary tables, and a slide deck / summary report. |

**Deliverables**

1. **Python Notebook (.ipynb)**
   1. Clear, commented code
   2. Executable from start to finish
   3. Includes plots and diagnostics
2. **Written Report (PDF or Word)**
   1. Introduction & objectives
   2. Methods (assumptions, transformations, modeling)
   3. Results with interpretation
   4. Conclusions, limitations, and recommendations
   5. Appendices (variable dictionary, model summaries)
3. **Presentation Deck (PowerPoint / PDF slides)**
   1. 5–7 slides summarizing key findings for non-technical audience
4. **Data Files**
   1. Original data (read-only)
   2. Derived dataset versions with new variables

**Marking Rubric**

|  |  |
| --- | --- |
| **Component** | **Weight (%)** |
| Data exploration & summary correctness | 10 |
| Assumption checks & diagnostics | 10 |
| Transformations & outlier handling | 10 |
| Variable derivation & management | 10 |
| Model fitting & selection logic | 15 |
| Hypothesis tests & confidence intervals | 10 |
| Interpretation & reporting quality | 15 |
| Visualizations & presentation clarity | 10 |
| Code quality & reproducibility | 10 |
| Total | 100 |

**Guidance Notes**

1. Always preserve the original variables; when creating new ones, name them clearly (e.g. ln\_gdp, pop\_ratio).
2. Use diagnostic plots (histograms, Q-Q, residual plots) to validate transformations.
3. Compare models not just by R² but by parsimony (adjusted R², AIC/BIC).
4. Explain in words what your numbers mean (not just “β1 = 2.5” — “this means an increase of 1 unit in X is associated with 2.5 units in Y, controlling for others”).
5. Your slides should communicate *key insights* in non-technical language.