Big Data Analytics- Homework 1
Shivani Tayade

Summary

Colab link: [big data homework1.ipynb]

Q1a. Are the test and control groups probabilistically equivalent? We compared average income, gender, and gamer status across test and control groups. The p-values for all three variables were above 0.05, and the percentage differences in means were negligible. This confirms that the groups are probabilistically equivalent on observed characteristics.

Variable	Test Mean	Control Mean	% Diff	P-Value
income	54.9382	55.1660	-0.41%	0.1276
gender	0.6473	0.6479	-0.10%	0.9060
gamer	0.6013	0.6018	-0.08%	0.9267

Q1b. Interpretation of Group Differences The similarity in income, gender, and gamer status between test and control groups shows that random assignment worked effectively. Any differences in outcomes (like purchases) can therefore be more confidently attributed to the treatment effect (i.e., the promotion). The statistical comparison between the test group and control group for income, gender and gamer variables shows that there is equivalence between these two groups. The p-value for all the three variables is above 0.05, which makes it statistically insignificant

Q1c. If large group differences existed before running the experiment If we had observed big differences before launching the experiment, then it indicates that the randomisation has failed,leading to potential confounding bias in estimating the treatment effect.

We should have re-randomized the groups or used stratified random assignment to ensure comparability. Alternatively, covariate adjustment in analysis could partially correct for imbalance.

Q1d. Significance testing in big data In large datasets, classic p-values may flag even tiny, meaningless differences as "significant." Instead, we should use effect sizes (like Cohen's d), confidence intervals, Bayesian methods, and bootstrap resampling methods to focus on practical significance.

Q2. Purchase Rates by Group We calculated test and control group purchase rates across different segments:

- **All Customers**: Test = 7.68%, Control = 3.62%, Diff = +4.06%
- **Male Customers**: Test = 7.46%, Control = 3.72%, Diff = +3.74%
- **Female Customers**: Test = 8.09%, Control = 3.44%, Diff = +4.65%

- **Gamers**: Test = 10.45%, Control = 3.54%, Diff = +6.91%
- **Non-Gamers**: Test = 3.51%, Control = 3.74%, Diff = –0.23%
- **Female Gamers**: Test = 11.01%, Control = 3.20%, Diff = +7.81%
- **Male Gamers**: Test = 10.14%, Control = 3.73%, Diff = +6.41%

Snapshot from code for reference-

Group	Test Purchase Rate	Control Purchase Rate	Absolute Difference
All Customers	0.076822	0.036213	0.040609
Male Customers	0.074575	0.037176	0.037399
Female Customers	0.080945	0.034442	0.046503
Gamers	0.104487	0.035436	0.069051
Non-Gamers	0.035092	0.037387	-0.002295
Female Gamers	0.110092	0.032041	0.078051
Male Gamers	0.101404	0.037275	0.064129

Interpretation: Female gamers respond the most positively to the promotion. Non-gamers show no uplift.

Q3. Expected Revenue

- **All Customers**: Test = $0.96, Control = $1.36
- **Female Gamers**: **Test = $1.38, Control = $1.20**
- **Male Gamers**: Test = $1.27, Control = $1.40

Group	Test Revenue (Net)	Control Revenue (Gross)
All Customers	0.960272	1.357991
Female Gamers	1.376147	1.201543
Male Gamers	1.267551	1.397815

Interpretation: The promotion only increases net revenue for **female gamers**. The control group still generates more revenue overall.

The overall conversion rate for the test group(all customers) is higher, i.e. 4.1 percentage points (0.076822- 0.036213) higher than the control group, but the net revenue is lower than the control group. Gamers, particularly female gamers, showed the strongest lift in both conversion rates and revenue:Female gamers had the highest net revenue gain ($1.38 vs $1.20). Non-gamers showed no meaningful lift and even slightly worse performance.

Q4. Recommendation to Management Game-Fun should not offer the promotion to all users. Instead, it should be targeted to **female gamers** and possibly other gamer segments who showed strong purchase lift and net revenue gains. Non-gamers should not receive the offer, as they showed little to no benefit.

Big Data Analytics- Homework 1
Shivani Tayade

Exercise 2: Oregon Health Insurance Experiment (OHIE)

Q1. ITT Estimate (Lottery Winners vs Losers)

- Winners: 0.90% negative outcomes
- Losers: 1.16%
- Difference: -0.26 percentage points

Interpretation: This difference suggests a potential benefit from winning the lottery. To interpret causally, we must assume lottery assignment affects health outcomes **only** through Medicaid enrollment.

Q2. Actual Enrollment Comparison

- Enrolled: 0.48% negative outcomes
- Not Enrolled: 1.40%
- Difference: -0.91 percentage points

Interpretation: This difference is likely **biased** due to self-selection. Sicker or more proactive individuals may be more likely to enroll.

Q3. Comparison Among Lottery Winners

- Enrolled: 0.48%
- Not Enrolled: 2.54%
- Difference: -2.06 percentage points

Interpretation: Among lottery winners, enrolled individuals had better outcomes. This is more credible, but still assumes compliance behavior is random.

Q4a. Wald Estimator

- Outcome diff: -0.0026
- Enrollment rate: 79.89%
- Wald Estimate: -0.0032

Q4b. Wald Assumptions To treat Wald as causal:

- Winning lottery affects enrollment (relevance)
- No other path from lottery to outcomes (exclusion)
- Random assignment of lottery (independence)
- No defiers (monotonicity)

Q4c. Standard Errors

- ITT SE = 0.00131
- Wald SE = 0.00164

- Wald SE is larger because it is divided by enrollment rate.

Q4d. Two-Sided Non-Compliance Check We checked for people who lost the lottery but enrolled. Found **0 cases**, so we confirm **only one-sided non-compliance**.

Seminal AI Paper Reflection: Pandemonium (Selfridge, 1959)

In “Pandemonium: A Paradigm for Learning”, Oliver Selfridge introduces a model for how machines can recognize patterns and learn from feedback. The system is built from layers of "demons" — small agents that each look for specific features in input data.

1. Key Insights: The big idea here is that learning can come from competing layers of “demons” — each one trying to recognize a pattern and “shout” when it sees something useful. These demons are basically like little functions or feature detectors. The system improves by reinforcing the demons that make correct guesses. This idea matters because it introduces layered learning, modular pattern detection, and the idea of credit assignment — which are central to modern deep learning models today.

2. Impact & Evolution: Pandemonium was one of the first models to show how hierarchical pattern recognition could work. It influenced neural networks, especially convolutional networks that also build up patterns from low to high levels. The way it assigns “blame” or “credit” to different levels also inspired early ideas in backpropagation and learning through feedback. It’s also kind of a mental ancestor of things like attention mechanisms — where the system decides which signals to prioritize.

3. Limitations & Critiques: One limitation is that the “demons” don’t really learn anything new — they’re chosen and adjusted, but not discovered. There’s no deep feature invention or abstraction. Also, the structure is a bit rigid — there’s no flexibility in how demons combine or change roles. Finally, it assumes the system already has a way to judge correctness (the teacher), which may not always be available in real-world settings.

4. Connections: This connects with what we learned about feature engineering and early neural nets. Pandemonium is like a manual version of what we now do with layers in deep learning — starting with small detectors and building up to complex patterns. It also reminds me of ensemble methods, where multiple “experts” vote, and only some get rewarded or tuned.

5. Application & Experimentation: If I had to build on this, I’d experiment with a system where demons evolve or merge over time — like if a group of low-level detectors could form a new high-level one automatically. That could lead to more self-organizing feature hierarchies without being manually hardcoded.

6. Unanswered Questions: One thing I wondered is: what happens when the wrong demons keep getting reinforced early on? Is there a way for the system to backtrack or “forget”? Also, can this model handle ambiguous or noisy input, or does it break when signals are unclear?