Assignment #6:

Marketing Biases in AI, a Pitch Deck, and a Video Ad

Behnam Mo

November 20, 2026

V Studio is a direct-to-consumer brand selling focus software and desk hardware. The company is about to relaunch its subscription plans, and the Chief Marketing Officer (CMO) has two questions. First, does adding a third plan to the menu lead more buyers to choose the top plan? Second, does an "Only 2 left" inventory label make buyers more likely to buy, and does that effect hold for human shoppers and AI shopping agents alike? You will test both questions on an AI model and pitch your findings to the CMO. You will also create a 30-second video ad for the new plans.

You must run every chat yourself. Use Antigravity, Claude Code, Codex, or your other favorite harness.

Your AI agent builds the slide deck and the video ad. You must run the experiments yourself, analyze the results, and write the analysis and explanations personally.

Submit one zip file on Canvas named `Assignment6_YourName.zip`. The zip must hold:

- `Assignment6_YourName.pdf`, containing all your answers, tables, and screenshots.
- A folder containing all website artifacts (e.g., HTML files, etc.), including your `AGENTS.md`.
- `pricing-ad.mp4`, and your video project folder minus its `node_modules` folder.

Q1 [60 pt]. Marketing biases in AI and an executive pitch.

A [4 pt]. Setup and brand rules. Make a folder for this assignment and open it as the project in your agent harness. Have your agent install two skills into the project, as you did in class: one design skill and one slide skill. Choose only from these approved lists:

- Design skills: frontend-design, theme-factory, impeccable, Hallmark, Taste Skill, or Vercel's agent skills.
- Slide skills: slide-wright, frontend-slides, or Stageshow.

(In Antigravity, project skills go in `.agents/skills/`; in Claude Code, they go in `.claude/skills/`.)

Next, write V Studio's brand rules in an `AGENTS.md` file in the project folder. Antigravity, Claude Code, and Codex all read it at the start of every conversation. Your rules file must contain:

1. Typography: One display font and one body font. Banned fonts: Inter, Roboto, Arial, and any system font such as Helvetica, San Francisco, or Segoe UI.
2. Color palette: Four labeled hex colors covering background, surfaces, text, and accent.
3. Spacing: One concrete spacing rule, such as margins and gaps on an 8-pixel grid.
4. AI slop bans: At least three named bans on common AI design patterns, such as purple-to-blue gradients, grids of identical centered cards, or slides that are only bullet lists. See 10 tells of a slop UI for examples of AI slop that your agent must avoid.

In your document, paste:

1. A screenshot of your agent's skills list showing both skills (for example, in Antigravity, open the Customizations panel; in Claude Code, paste the output of `/skills`). **[4 pt]**, 2 per skill.

B [28 pt]. Two marketing bias experiments. Review the class deck and follow the method. You will test two experimental versions that differ in one detail. Each individual run takes place in its own fresh chat, with 10 runs for Version A and 10 runs for Version B (20 valid runs per experiment, 40 runs total).

Follow these running rules:

- Use one chat model on OpenRouter or Google AI Studio for every run across both experiments. Record its exact name as the app shows it.
- Every run must be a new chat holding one message: the prompt. Never run two tests in one chat.

- Turn off memory and custom instructions before starting. Use regular chats, not temporary or incognito chats.
- Collect 10 valid answers per version. A valid reply contains only the single requested letter or number. If a reply is off-format, start one more fresh chat to replace it.

Writing your prompts:

- Write a short buyer context explaining who is buying and why to open each prompt. Use the same buyer context in both Version A and Version B of an experiment.
- Copy the plans, product details, availability lines, and closing instructions character for character from the text below.
- Banned in prompts: decoy, scarcity, urgency, FOMO, bias, experiment, test, version, and control.

Experiment 1: Decoy effect. You are testing whether adding an inferior third plan sends more buyers to the top plan. **Plus** costs the same as **Pro** and has fewer features, so no careful buyer picks it. It serves as an asymmetric decoy, like The Economist's print-only offer (decoy slide). The question is whether adding it sends more answers to Pro.

Copy these plans character for character:

- Essential, $9/month: focus soundscapes and a Pomodoro timer.
- Plus, $24/month: focus soundscapes, a Pomodoro timer, and an AI daily schedule builder.
- Pro, $24/month: focus soundscapes, a Pomodoro timer, an AI daily schedule builder, and calendar sync.

Construct your two versions:

- Version A offers two plans: A = Essential and B = Pro. End the prompt with this exact line: `Reply with only the letter A or B.`
- Version B offers three plans: A = Essential, B = Plus, and C = Pro. End the prompt with this exact line: `Reply with only the letter A, B, or C.`

Experiment 2: Scarcity. You are testing whether an inventory label raises purchase likelihood (scarcity slide). Both prompts open with your buyer context, followed by this exact product description: `V Chrono Desk Dial, $79: a tactile focus controller for`

`your desk. Rated 4.7 out of 5 from 310 verified reviews.`

Add the availability line:

- Version A adds: `Availability: In stock.`
- Version B adds: `Availability: Only 2 left in stock.`

End both prompts with this exact line: `How likely are you to buy it, from 1 (very unlikely) to 7 (very likely)? Reply with the number only.`

Proof of each run. Take a screenshot of each chat iteration.

In your document, paste:

1. The four prompts, word for word. **[4 pt]**, 1 per prompt.
2. Your run log: a screenshot of every chat, labeled with its experiment, version, and run number. **[6 pt]**, 3 per experiment.
3. Your results by counting. **[8 pt]**, 4 per experiment.
    - For Experiment 1: Count how many of the 10 answers chose Pro in each version. In Version B, also report the number choosing Plus. Calculate the gap (Version B count minus Version A count). A gap of 5 or more in either direction indicates a real effect; a smaller gap is too small to tell from chance. If Version A already has 6 or more Pro answers, an increase of 5 is impossible, so report "ceiling".
    - For Experiment 2: Calculate the average of the 10 ratings for each version. Calculate the gap (Version B average minus Version A average). A gap of 1 point or more in either direction indicates a real effect; a smaller gap is too small to tell from chance.
4. A comparison with people. **[4 pt]**, 2 per experiment.
    - State whether the AI moved in the same direction as people. A count out of 10 is a percentage (7 of 10 equals 70%).
    - People, decoy: 84% chose the top offer with the decoy on the menu versus 32% without; the gap was 52 percentage points (Ariely, 2008).
    - People, scarcity: A review of 131 studies showed that scarcity reliably raises purchase intentions (Barton, Zlatevska & Oppewal, 2022); compare direction only.

5. Fresh chats, in tokens.
    - Count the tokens of your Experiment 1 Version B prompt with the tokenizer in the course's language-model lesson.
    - Suppose hypothetically that you ran all 10 Version B trials in one chat, where each call resends the whole conversation so far (the growing conversation), like Bestie in Assignment 1.
    - Report the total input tokens of those 10 calls versus 10 fresh chats (ignore the one-letter replies). **[2 pt]** Then explain in one or two sentences why one long chat would also spoil the test. **[2 pt]**
6. What 10 runs measure. **[2 pt]**
    - Answer in two or three sentences: Each reply is a token drawn from the model's next-token probabilities (sampling), as in Assignment 2. Suppose Pro won 7 of your 10 Version B chats: what does that 7 in 10 tell you about this model? What would 10 runs at temperature 0 show, and how many independent answers would that be? Did your runs vary? Explain whether 10 answers from one model are like 10 different customers.

C [28 pt]. The pitch deck. Direct your agent, with both skills and the rules file, to build a deck for V Studio's CMO: one HTML file named `v-cmo-pitch.html`.

The file must open on double-click in a current Chrome browser with the internet on. Fonts and slide libraries may load from the web. It must require no install, build step, or local server. Any file it uses goes in the same website folder.

The deck must contain exactly 4 slides, and the arrow keys must move between them:

1. Summary: V Studio's relaunch goal, the questions tested, and the headline finding.
2. The menus: The two-plan and three-plan menus side by side, showing that Plus costs the same as Pro but gives less.
3. Results: Both experiments' counts or averages, with the number of runs, next to what people did (Part B, item 4), formatted as a chart or a table.
4. What to do: At least 2 concrete actions for human shoppers on V Studio's web store and at least 2 for AI shopping agents reading its product pages. Each action must tie to a result on slide 3 or a study from class.

Your deck is graded on these points:

1. `v-cmo-pitch.html` opens and moves through the four slides **[3 pt]**, and your PDF shows a screenshot of each slide **[2 pt]**.
2. The four slides hold the listed content **[8 pt]**, 2 per slide.
3. The deck follows the rules file: the two fonts **[2 pt]**, the four colors **[2 pt]**, and no AI slop, meaning none of the banned patterns and none of the ten tells in the linked article **[5 pt]**.
4. Accuracy and actions: every number on slides 2 and 3 matches the plans and the run log **[3 pt]**, and the slide-4 actions are concrete and tied to a result or a study **[3 pt]**.

Q2 [40 pt]. A video ad for the new plans. Paid-social teams make ads from code so they can change a price or a line of copy and render again, with nothing redrawn. You will build a programmatic video ad for V Studio's new plans.

Setup. Use Remotion or HyperFrames, whichever you prefer; both are free. As in class, ask your agent to install the Remotion skill (or the HyperFrames skill) and set up your video project; you do not need to type any commands. Don't worry; your agent will take care of all dependencies during installation. Then copy your Part A rules into the video project's `AGENTS.md`. State in your PDF which tool you used and why.

Design and video requirements. Look first at Remotion's prompt gallery to see videos next to the prompts that made them. Read at least the VVTerm promo, the Music CD store promo, and The Kinetic Marketing, noticing how precisely they set scenes, timing, colors, and fonts.

Your ad must meet these specifications:

- Length: 30 seconds. You may use any frame size.
- Structure: Exactly three scenes:
    1. Scene 1 (0 to 6 seconds, Hook): Animated text with V Studio's promise of deeper focus.
    2. Scene 2 (6 to 24 seconds, Offer): Essential $9, Plus $24, and Pro $24 side by side. The animation makes clear that Pro gives more than Plus for the same $24.
    3. Scene 3 (24 to 30 seconds, Call to action): Conclude with this exact text: `Start Your 14-Day Pro Trial`

- Animation and craft: Use the Part A fonts and colors. Follow Remocn's design defaults: no decorative letter-spacing, no all-caps by reflex, no gradient-filled text, and no glows or heavy shadows. Avoid two of Remocn's anti-patterns: do not have everything enter on the same frame (stagger entrances by 3 to 6 frames), and ensure no text is cut off because its scene ends too early.
- External data file: Put every tier name, price, feature, color, and the call-to-action text in one plain-text file that can be edited without touching code. Design the format yourself, as for the colleague in Assignment 3, and direct your agent to make the video read everything from it.

In your zip and your PDF:

1. `pricing-ad.mp4`, 30 seconds long **[4 pt]**, and a screenshot of the preview with its timeline in Remotion Studio or HyperFrames (ask your agent to open it) **[4 pt]**.
2. The three scenes, on time and with the content above **[12 pt]**, 4 per scene.
3. Brand and craft compliance: the Part A fonts and colors, none of the Remocn patterns above, and no text cut off or overlapping **[6 pt]**.
4. The data file in the zip, along with a short table in the PDF: each field, what it holds, and its limit (such as a tier name of at most 14 characters) **[6 pt]**.
5. Evidence of dynamic pricing: change Essential's price in the file by hand, have your agent render again, and paste the same Scene 2 frame from both renders plus the changed line in the file **[4 pt]**.
6. Edge case test: give one tier an oversized 30-character name in the file, have your agent render, and paste the frame. Report whether the ad coped (by shrinking the text, wrapping it, or stopping with a clear message) or broke, and explain how the file's limits would prevent it **[4 pt]**.