Documentation

2024-10-30

Table of contents

Preface

DICE combines the control of experimental research with more realistic stimuli that are inspired by social media platforms. The DICE app (which is based on the powerful oTree framework Chen, Schonger, and Wickens 2016) enables you to create realistic social media feeds with customizable content while supporting between-subjects experimental designs. It provides precise dwell time tracking for engagement measurement and integrates seamlessly with popular survey platforms like Qualtrics. For participant recruitment, DICE works smoothly with Prolific, and all collected data is available in convenient formats for analysis.

Quickstart

Getting started with the DICE app is straightforward. Visit dice-app.org to access the web interface. You'll need to create your stimulus configuration in a CSV file, for which we provide a template. After uploading your CSV to a public repository like GitHub, you can use DICE's web interface to configure and launch your study. The platform handles participant management and data collection automatically.

Documentation Overview

Our documentation follows the natural workflow of creating and running a DICE experiment:

The Stimuli chapter forms the foundation, showing you how to create and configure your social media feed content using a CSV file. This structured approach ensures precise control over your experimental manipulations while approaching an authentic feel of social media.

Once your stimuli are ready, the Launch a Study chapter guides you through the DICE web interface at dice-app.org. You'll learn how to configure your experiment, integrate with survey platforms like Qualtrics, and manage participant recruitment through Prolific.

The Data chapter then explains how to work with the data DICE collects, including dwell time measurements and user interactions. You'll learn about data structure, processing, and how to combine DICE data with your survey responses.

We've included two detailed Case Studies that demonstrate DICE's capabilities in real research scenarios: one investigating ad recall and position effects, and another examining brand safety in social media advertising. For researchers interested in advanced implementations, the Deployment chapter provides guidance for hosting DICE on your own servers.

Getting Help

If you need assistance with DICE, you have several options available:

- Explore the documentation chapters
- Review the case studies for practical examples
- Visit our GitHub repository for additional resources
- Contact the DICE team for technical support

1 Generate Stimuli

```
options(repos = c(CRAN = "https://cran.r-project.org"))


if (!requireNamespace("groundhog", quietly = TRUE)) {
    install.packages("groundhog")
    library("groundhog")
}

pkgs <- c("magrittr", "data.table", "knitr", "kableExtra", "stringr")

groundhog::groundhog.library(pkg = pkgs,
                             date = "2024-10-01")

rm(pkgs)
```

The DICE app was designed to fit into a consumer researcher's typical workflow where participants are recruited (e.g., via Prolific) before they are exposed to stimuli and survey items (e.g., in Qualtrics). The key procedural difference in using the DICE app (compared to software such as Adobe Photoshop or Microsoft Powerpoint) is that the stimuli are not configured graphically but tabularly: the DICE app requires researchers to configure a csv file that provides information on each post, such as the actual content, engagement metrics, and the corresponding username. The app then loops through that file (while filtering for conditions) to display each row as a separate post embedded in an interactive feed. The advantage of this procedural difference is that while it requires the same amount of information as the graphical configuration, it is less time consuming and less error prone because the software handles the graphical representation consistently. In addition, the tabular configuration is more accessible as researchers are trained to work with csv, as opposed to Photoshop files.

1.1 Overview

Here, we provide a configuration csv file that serves as a template for researchers who configure their first set of stimuli. In Table 1.1, we display and describe an exemplary row of this template containing social media posts on the Yosemite National Park in California. The first column in Table 1.1 describes a configuration column's scope, that is, whether it defines how, when and to whom a post is displayed (`design`), contains a post's actual content or engagement metrics such as the number of likes the post has received (`post`), or whether it describes a post's author (`user`). The second column lists all of the input csv file's required configuration columns.[1] We then describe the configuration columns' data types, provide examplary values, and describe them in more detail.

```r
input  <- data.table::fread(file = "input_data.csv")

kable(input)
```

Table 1.1: Description of Input csv File

Scope	Column	Type	Example	Description
design	condition	text	A	Unique identifier for the condition (e.g., group A or B). The app will create the same number of conditions and randomly assign participants to them.
design	sequence	numeric	1	Defines the order in which posts are displayed (ascending). If missing, random integers will be assigned, leading to between-subject randomization.
design	commented_post	boolean	0	If 1, this post will be displayed at the top of the feed while all other posts are displayed as comments of this post.
post	doc_id	numeric	1	Unique identifier for each post in each condition.
post	datetime	text	01.03.22 06:00	The time a post was published, formatted as dd.mm.yyyy hh:mm:ss.
post	text	text	Just experienced the most incredible sunrise… #Yosemite #NatureLovers	Content of the post. Can contain hashtags, emojis, and URLs.
post	media	text	https://images.unsplash.com/photo-1472396961693…	URL to a publicly available image or GIF. We recommend using compressed formats like .webp or hosting images on platforms such as Unsplash or GitHub.

[1] Researchers can change the order of these columns as they want. In addition, they can add additional columns for internal purposes. We recommend, to document a post's source (i.e., a url) if a post was scraped or copied from social media or an ad library.

Table 1.1: Description of Input csv File

Scope	Column	Type	Example	Description
post	alt_text	text	Sunset illuminates Half Dome…	Optional description of the media element for accessibility purposes.
post	likes	numeric	15	Number of likes the post has received.
post	reposts	numeric	6	Number of reposts the post has received.
post	replies	numeric	2	Number of replies the post has received.
post	sponsored	boolean	0	If 1, the post will be displayed as a sponsored post.
post	target	text	https://example.com	URL of the landing page for sponsored posts. If sponsored = 1, this URL will be displayed as the target for the advertisement.
user	username	text	NatureFanatic	The username of the post's author.
user	handle	text	NatureFanatic88	The handle of the post's author, automatically prefixed by an @ symbol.
user	user_description	text	Lover of all things nature…	A brief description or bio of the user, displayed when hovering over the user's profile.
user	user_image	text	https://images.unsplash.com/photo-1522506209496…	URL to the user's profile picture. Similar to media, we recommend using stock images or hosting them on platforms such as GitHub, preferably in compressed formats.
user	user_followers	numeric	4523	Number of followers the user has.

1.2 Design Columns

Before Video 1.1 describes the columns in more detail, we focus on the design columns and describe how researchers can configure them to implement their experimental designs.

1.2.1 Conditions

Researchers can leverage the `condition` configuration column to set up between-subjects designs by assigning the respective rows with N different values (e.g., “treatment 1”, “treatment 2”, ..., “treatment N”). The DICE app will then count the number on unique values and create N different treatment groups. When launching a study, participants will then be assigned to these groups randomly and uniformly such that the group sizes do not differ in expectation. In case study 2, we show how we leveraged this variable to create two conditions that contain two different sets of nineteen organic posts but share the same sponsored post. As each row can only be assigned to one condition, this required us to enter the sponsored post twice within the configuration file: both versions contained the same post and user configurations and only differed with respect to their `condition`. Similarly, if researchers want to display the same set of organic posts in N conditions, then they have to enter N copies of that set of posts and adjust the `condition` column accordingly.

1.2.2 Sequences

Another important configuration column is `sequence`. It defines the order in which posts are displayed and gives researchers control over the order in which posts are displayed. Explicit sequences can be useful to study ordering and ranking of social media posts. Researchers can, for instance, use the user interactions measured in a previous study to rank the by engagement to approximate a platform’s recommender systems. Importantly, this column is special as the DICE app replaces missing values with random numbers for each participant individually. This is a feature we leveraged in both of our case studies: in Case Study 2, we only defined the sequence of the sponsored post such that it was always displayed in fifth position. The `sequence` configuration column was not assigned to any of the organic posts. Hence, each participant experienced a different sequence of organic posts. In Case Study 1, we left that `sequence` column empty for every post to randomize the order of both sponsored and organic posts. This resulted in a diverse set of sequences that we exploited to study primacy effects in ad recall.

1.2.3 Threads

Finally, the `commented_post` configuration column is interesting for researches who want to investigate discussions as it changes the social media feed’s appearance slightly. If one post is assigned to a `1` in this column, this post will serve as a “parent post” whereas all other posts will be displayed a comments of that parent post.

1.3 Image Requirements

The DICE app does not host images directly. Instead, you must provide direct links (raw URLs) to your images hosted elsewhere. These URLs should point directly to the image file itself, not to a webpage containing the image.

What is a Raw Image URL? A raw image URL points directly to the image file and typically ends with a file extension like `.jpg`, `.png`, `.gif`, or preferably `.webp`. These URLs provide direct access to the image file without any surrounding webpage elements. For instance, a raw GitHub image URL might look like `https://raw.githubusercontent.com/username/repository/main/images/example` while an Imgur URL might be `https://i.imgur.com/abcd123.png`, and a Giphy URL could be `https://media.giphy.com/media/abc123/giphy.gif`.

1.3.1 Hosting Your Own Images

Several platforms can host your images for use with DICE. GitHub offers a straightforward approach: upload images to a public repository and use the raw URL that starts with `raw.githubusercontent.com`. Imgur provides another popular option as an image hosting service that readily provides direct image links. Cloud storage services like AWS S3 or Google Cloud Storage can also work well, though you'll need to ensure public access is enabled.

1.3.2 Getting Raw Image URLs

The process of obtaining a raw URL varies by platform. On GitHub, navigate to your uploaded image and click the "Raw" button - the resulting URL in your browser will start with raw.githubusercontent.com. When using services like Imgur, you can usually right-click on the uploaded image and select "Copy image address" or a similar option. The key is ensuring your URL ends with an image extension (`.jpg`, `.png`, etc.).

1.3.3 Verifying Your URLs

You can easily test whether your image URL is correct by pasting it directly into a browser's address bar. If the browser shows only the image itself, without any surrounding webpage elements, the URL is suitable for use with DICE. This simple verification step can save time troubleshooting later.

1.4 Video Tutorial

Here, we describe the configuration of the stimuli we used in our brand safety case study (see Section 4.1) in detail.

https://www.youtube.com/watch?v=lx_akVasq7I

Figure 1.1: DICE: csv file configuration

1.5 Best Practices

Image optimization plays a crucial role in your study's performance. Compressing your images before hosting them ensures faster loading times for participants. Web-optimized formats like .webp often provide the best balance of quality and performance. There are many online converters and even python modules available.[2]

It's crucial to understand that directly linking to images from social media platforms like X (formerly Twitter) may lead to problems: if a user updates their profile picture or deletes a post, the image URL will break and your study's stimuli will be incomplete. Instead, download any images you want to use from social media and host them on a platform you control, such as GitHub. This ensures your stimuli remain stable throughout your study's duration.

Remember to maintain a backup of all images and verify that your image URLs remain publicly accessible throughout your research period. This approach provides the most reliable way to ensure your study's integrity over time.

You will likely create your own images. To include them in your feed, you need to host them somewhere publicly. We usually use Github for these purposes. We learnt that it makes a lot of sense to compress the images you are using such that your stimuli are less affected by slow internet connections on your participants' side. In addition, it helps to use an image format that is optimized for web usage. We made good experiences with the .webp format.

1.6 Archiving for Review

For academic documentation and review purposes, it's valuable to create permanent archives of your experimental feeds. Services like perma.cc or the Internet Archive's Wayback Machine can capture and preserve your feeds exactly as they appeared during the study. These archives serve as reliable references for reviewers and future researchers, ensuring that your experimental stimuli remain accessible even if the original hosting platforms change or links break over time. When writing up your research, you can include these permanent archive links in your methodology section or supplementary materials.

[2] See our repository (url provided after review) for more details on the technical implementation.

1.7 Public Stimuli Resources & Databases

We created a set of synthetic users, where we matched actual usernames and handles (see McKelvey et al. 2017) with stock images (from unsplash.com) and some synthetic, LLM-generated information. Also see generated.photos and their academic data for user images and behindthename.com for user names.

In addition, the data science competition platform *kaggle* hosts a variety of annotated social media related datasets (e.g., the Social Media Sentiments Analysis Dataset, the Cyberbullying Dataset, Social Media Influencers in 2022, or Political Social Media Posts) which researchers can use to create their stimuli. The same applies to *Hugging Face*, which hosts a Reddit confessions dataset, for instance.

Finally, you can use Facebook's Ad Library, a publicly accessible database that archives advertisements run by advertisers across Meta's platforms to copy (or draw inspiration for) copy, creative as well as a landing page for sponsored posts.

2 Launch a Study

We developed the DICE web-app for researchers to launch a study without any coding and without hosting the oTree app yourself. This section guides you through the process step by step.

2.1 Meta Information

We ask you to provide some information about yourself and the study you intend to run. The DICE app uses these information to provide participants with contact information if any questions occur. In addition, the DICE app displays an external study name. The internal name only serves internal documentation purposes for yourself.

2.1.1 Configuration Files

When you successfully create a session, DICE automatically generates and downloads a JSON configuration file (e.g., `abc123_dice_config.json`). This file contains all your study settings and can be reused to quickly set up similar studies.

To load a saved configuration file:

1. Upload a previously downloaded JSON configuration file
2. Click "Load Configuration"
3. All form fields will be automatically populated with the saved settings from that session
4. You can then modify any settings as needed before creating a new session

This is useful for running follow-up studies or quickly reproducing a previous study's exact configuration.

2.2 Participant Recruitment

DICE supports multiple recruitment strategies to fit your research needs:

2.2.1 Study Type Options

You can choose from the following recruitment platforms:

Online Recruitment Platforms:

- **Prolific** — Optimized for Prolific's participant pool with automatic parameter handling for easy data linking
- **Connect (Cloud Research)** — Integration with Cloud Research's Connect platform for access to diverse participant populations

Lab Studies:

- **Lab Study (Single-Use Links)** — Run studies in a controlled lab environment where each participant receives a unique, single-use link

2.2.2 Study Slots (Participant Capacity)

You need to specify the number of participant slots available for your study. Each slot can be used by one participant. If there is attrition—that is, if a participant enters the study but leaves before finishing it—their slot in the database will be consumed and cannot be reused by other participants. Hence, you should account for expected attrition when specifying the number of study slots.

For example, if you expect a 20% dropout rate and want 100 completed responses, you should allocate approximately 125 slots.

Important considerations: - Once all slots are filled, no new participants can join the study - We enforce a maximum of 400 slots per study to maintain performance - If you need larger sample sizes, please contact us to discuss options - Alternatively, consider deploying a Digital In-Context Experiment yourself

2.2.3 Automatic Participant ID Parameter Handling

The participant ID parameter is automatically configured based on your selected recruitment platform:

- **Prolific** → Uses `PROLIFIC_PID` parameter
- **Connect** → Uses `participant_label` parameter
- **Lab Study** → Uses `participant_code` parameter

You don't need to manually configure these—they're set automatically when you select your study type. When integrating with survey platforms like Qualtrics, create an Embedded Data field that matches your platform's parameter name (e.g., `PROLIFIC_PID`, `participant_label`, or `participant_code`).

In addition to the participant ID, DICE also passes the participant's **experimental condition** as a URL parameter named `condition`. This allows you to:

- Create condition-specific survey questions in your survey platform
- Track which condition each participant experienced
- Analyze responses by experimental condition

For example, when redirecting to Qualtrics, participants will be sent to a URL like:

```
https://unisg.qualtrics.com/jfe/form/SV_0DnMoLpM0VxjhrM?PROLIFIC_PID=abc123&condition=A
```

To capture the condition in Qualtrics, create an Embedded Data field named `condition`, and it will be automatically populated with the participant's assigned condition.

Important Methodological Consideration: The condition label is visible in the redirect URL. If participants notice their condition assignment in the URL, they may infer details about your experimental design and hypothesis, which could create demand effects or bias their survey responses. To mitigate this risk, we recommend using abstract or non-descriptive condition labels in your CSV file (for example, "A" and "B" instead of "control" and "treatment", or acronyms like "FV" and "SV"). This way, even if participants see the URL parameter, they won't understand what condition they're in.

2.3 Stimulus Design

2.3.1 Social Media Platform

2.3.2 URL

In this section, you will provide the csv file you created to configure your stimuli as described in Chapter 1. The DICE app "ingests" these configurations if you provide raw URL to the file. This requires you to upload the csv file in a publicly available storage (we recommend GitHub as it is free and easy-to-use).

The DICE app needs access to your CSV file through what's called a "raw" URL. A raw URL points directly to the file's content, allowing DICE to read your configuration. Think of it as a direct link to the raw data, rather than a webpage showing the file.

2.3.2.1 What is a Raw URL?

When you share files through platforms like GitHub, the standard URL typically leads to a webpage that displays your file. However, DICE needs a special type of URL - a raw URL - that leads directly to the file's content. A raw URL usually follows this pattern:

```
https://raw.githubusercontent.com/username/repository/main/filename.csv
```

The key difference is that a raw URL starts with `raw.` in the domain name. This tells the server to send the pure file content rather than a webpage displaying the file.

2.3.2.2 Getting a Raw URL

If you're using GitHub, you can find the raw URL by navigating to your file and clicking the "Raw" button. The URL in your browser will then show the correct format that DICE needs. If you prefer using Google Drive, you can share your CSV file and get a direct link. First, share the file by setting it to "Anyone with the link can view." Then, instead of using the standard sharing link, modify it by replacing "file/d/" with "uc?export=download&id=" and removing everything after the file ID. Your Google Drive URL should look like this:

```
https://drive.google.com/uc?export=download&id=YOUR_FILE_ID
```

2.3.2.3 Using Your Raw URL

Once you have the raw URL, simply paste it into DICE where requested. The app will automatically fetch your configuration file and use it to set up your study.

2.3.3 CSV Delimiter

Your CSV file uses a special character to separate different pieces of data (like columns). While commas are the most common delimiter, some files use other characters. DICE can auto-detect the correct delimiter, but you can also manually select the character that matches your CSV file format—for example, if your data is separated by semicolons, choose "Semicolon" from the dropdown menu.

2.3.4 Search Term

Enter the term you want participants to see in the search bar of your simulated social media interface. Using a hashtag or a username creates a more authentic social media experience, as it mimics how users typically search for content on platforms like Twitter. This search term will appear pre-filled in the interface's search bar when participants start the study. For example, you might use `#myHashtag`, `@username`, or a specific topic like `climatechange`.

2.3.5 Validating Your CSV File (Recommended)

Before creating your session, you can test your CSV file to ensure it's properly formatted and accessible. Click the "Test CSV" button to:

- Verify that the CSV URL is accessible and working
- Auto-detect the correct delimiter if needed
- Check that all required columns are present
- Identify any potential issues with your data

This validation step helps catch configuration errors early and makes troubleshooting easier if anything goes wrong.

2.4 Participant Briefing

This section provides two rich text editors that allow you to create formatted content for your study participants.

2.4.1 Consent Form

The first editor allows you to create a consent form that participants must agree to before participating in your study. This is particularly important for research conducted through institutional review boards or when ethical compliance is required. You can format your consent text using the toolbar options, including headers, bold text, and other formatting features. The consent form typically includes information about the study's purpose, procedures, time commitment, voluntary participation, data privacy measures, and contact information for questions.

2.4.2 Study Instructions

The second editor is where you provide instructions and context for participants before they interact with the social media feed. These instructions could explain what participants will see, what they should do, and any specific guidance about how to engage with the content. Accordingly, your study instructions might include welcome text, task instructions, expectations about timing, and any special considerations for interacting with the social media posts.

If you leave the study instructions editor empty, participants will skip directly from the consent form to the social media feed. If both editors are empty, participants will go directly to the social media content without any preliminary pages.

The content you create in both editors is stored as part of your study configuration and will be displayed exactly as you format it to all participants in your study.

2.5 Measurement

2.5.1 Survey Integration

After participants interact with the social media feed, you can redirect them to a survey. Simply provide the base URL of your survey (for example, a Qualtrics survey URL) without any additional parameters. DICE will automatically:

- Add the participant's ID to track responses
- Handle the redirect process

For example, if you're using Qualtrics, your URL might look like this: `https://unisg.qualtrics.com/jfe/form/`

If you leave this field empty, participants will see a closing screen with your contact information instead of being redirected to a survey.

2.5.1.1 Completion Codes

Completion codes are used to verify that participants have completed your study and are eligible for compensation or credit. Different recruitment platforms provide completion codes in different ways:

- **Prolific**: Generates a unique completion code that participants must submit at the end of the study to confirm completion and receive payment
- **Connect (Cloud Research)**: May use completion codes or require a redirect back to their platform with specific parameters
- **Lab Studies**: You typically manage completion verification directly (e.g., via course credit systems or manual verification)

Best Practice for Data Integrity: The most secure approaches to handle completion codes are: 1. **Code-based**: Provide participants with a unique completion code at the end of your survey that they submit back to the recruitment platform 2. **Redirect-based**: Have your survey redirect participants back to the recruitment platform with a completion confirmation, which automatically verifies their participation

This ensures that participants who complete your DICE study and survey can properly confirm their participation without relying on self-reporting. Check your recruitment platform's documentation for their specific completion code requirements and implementation methods.

2.5.2 Dwell Time Tracking

The dwell time tracking threshold determines when DICE considers a social media post as "viewed" by the participant. This setting is expressed as a percentage of the post's visibility in the participant's browser window:

- A threshold of 75% (default) means a post must be 75% visible to be considered "in view"
- Lower values (closer to 1%) will count posts as viewed even when barely visible
- Higher values (closer to 100%) only count posts that are almost fully visible

Choose your threshold based on how strictly you want to measure post exposure. For most studies, the default value of 75% provides a good balance between capturing meaningful exposure while allowing for natural scrolling behavior.

2.6 Create Session

After clicking "Create Session," you'll see a confirmation screen that contains all essential information for running your study and collecting data.

2.6.1 Saving Your Session Information

The success screen displays your unique session code (such as `h4qfjieu`) and a corresponding session URL. These identifiers are crucial as they provide the only way to monitor your study's progress and download your data later. Make sure to save both the code and URL in a secure location for future reference.

2.6.2 Prolific Integration

This section is relevant only if you're using Prolific for participant recruitment.

The success screen provides a specially formatted URL that includes Prolific's parameter structure. It looks similar to this:

```
https://ibt-hsg.herokuapp.com/join/pifararu/?participant_label={{%PROLIFIC_PID%}}&prolific_s
```

Steps to integrate with Prolific:

1. Copy the entire URL from the success screen
2. Paste it (as is) into your Prolific study details
3. In Qualtrics, create an Embedded Data field named `PROLIFIC_PID` to capture the Prolific participant ID

The URL contains three important placeholders: - `{%PROLIFIC_PID%}` captures each participant's unique identifier - `{%STUDY_ID%}` tracks your specific study - `{%SESSION_ID%}` records individual participant sessions

You'll also find a field to enter your Prolific completion code. This code needs to be entered and submitted to allow participants to confirm their participation after completing your study. The system automatically handles the completion code distribution to eligible participants, ensuring proper compensation through Prolific's system.

2.6.3 Connect (Cloud Research) Integration

This section is relevant only if you're using Connect (Cloud Research) for participant recruitment.

After creating your session, you'll see information about how to set up your study in Connect:

1. Copy the session URL from the DICE success screen
2. In Connect, set the "Collecting Connect IDs" field to use `participant_label` as the parameter name
3. This ensures Connect will append `?participant_label=XXX` when redirecting participants to your study
4. In Qualtrics, create an Embedded Data field named `participant_label` to capture the Connect participant ID

This integration allows you to automatically track which Connect participants completed your study and link their DICE responses with their survey data.

2.6.4 Lab Study Integration (Single-Use Links)

This section is relevant only if you're running a lab study with single-use participant links.

When you select "Lab Study (Single-Use Links)" as your study type, DICE generates unique participant links for each participant slot:

1. After creating your session, navigate to the **Session Details** (accessible via the admin interface using your session code)
2. You'll find a list of **single-use participant links**, each containing a unique `participant_code`
3. Distribute these links to your participants individually (via email, on printed cards, etc.)
4. Each link can only be used once, ensuring each participant receives a unique identifier
5. In Qualtrics or your survey platform, create an Embedded Data field named `participant_code` to capture this identifier
6. This allows you to match DICE data with survey responses for each participant

This approach is ideal for in-person lab studies where you have direct control over participant recruitment and link distribution.

3 Data

```r
options(repos = c(CRAN = "https://cran.r-project.org"))


if (!requireNamespace("groundhog", quietly = TRUE)) {
    install.packages("groundhog")
    library("groundhog")
}

pkgs <- c("magrittr", "data.table", "knitr", "kableExtra", "stringr")

groundhog::groundhog.library(pkg = pkgs,
                             date = "2024-10-01")

rm(pkgs)
```

We created the shiny-app to pre-process the raw data (which is tabular but also contains columns with JSON formatted data to describe the participant×post-level observations).

The shiny-app requires you to upload your raw DICE data and transforms its wide to a long format. It then provides a download-button. Below, we briefly describe the raw data before we turn our attention to the output of the shiny-app, that is, the processed data. In both sections, we use the data collected in Case Study 1 and anonymize the Prolific IDs (called participant labels).

```r
raw <- fread(file = "../../oFeeds/studies/meme_feed/data/raw/all_apps_wide_2024-10-15.csv",
             na.strings = "",
             nrows = 7)
participants_in_raw_data <- raw[, unique(participant.label)]

processed <- fread(file = "../../oFeeds/studies/meme_feed/data/raw/DICE-processed-2024-10-15
                   na.strings = "")[participant_label %in% participants_in_raw_data]
```

3.1 Raw Data

While creating an experimental session, the DICE web app returns session code such as `r5fbdbr1`, for instance. It will also create a url containing that session code (e.g., `https://ibt-hsg.herokuapp.com/SessionStartLinks/r5fbdbr1`) directing you to you oTree interface, where you can monitor the session or download the data. After clicking on the `Data` tab, you can download the raw data by clicking on the `Plain` button in the bottom right of her screen. This will generate a csv file called `all_apps_wide_YYYY-MM-DD.csv`.

This data follows oTree's conventions as displayed in Table 3.1.[1]

```
raw[, participant.label := str_trunc(string = participant.label, width = 8, side = "left")]
raw[, participant.tweets := NULL]
setorder(x = raw, participant.label)
kable(raw[1:2])
```

[1] In Table 3.1, we removed a column called `participant.tweets` as it reports all the information displayed in the feed for every participant. We would advise you to delete that column too as it decreases the file size considerably (if you work with long feeds).

Table 3.1: Excerpt of Raw Data Analyzed in Case Study 1

[illegible]

Table 3.1: Excerpt of Raw Data Analyzed in Case Study 1

[illegible]															
5	jyri… 1737.846	5	DICE_Direct	2024-10-15 14:44:20	NA	0	1	z2q0As…	NA NA NA	0	Social Media Study	Two-50-hatter	https://raw.githubusercontent.com/Howquez/DICE/refs/heads/main/studies/meme_feed/stimuli/9gag.csv (9GAG)	sequence … Roggenkamp	LIFIC_PD NA	1 1 Mn… Fx… https://…lific.com/submissions/complete?cc=…

Table 3.1: Excerpt of Raw Data Analyzed in Case Study 1

[illegible]
6 [illegible] 5 DICE_[illegible] 2024-10-15 14:47:58 NA NA 0 1 [illegible] NA NA NA 0 Social Media Study [illegible] 50 [illegible] https://raw.githubusercontent.com/Howquez/DICE/refs/heads/main/studies/meme_feed/stimuli/9gag.csv [illegible] NA 1 1 [illegible] https://[illegible]sions/complete?cc=[illegible]

3.2 Processed Data

The data we show below has a nested structure as it shows observations on the participant×post-level.

```r
processed[, participant_label := str_trunc(string = participant_label, width = 8, side = "le
setorder(x = processed, participant_label)
processed[, touch_capability := as.logical(touch_capability)]
processed[, seconds_in_viewport := as.numeric(seconds_in_viewport)]

kable(processed[35:45])
```

session_code	participant_code	participant_label	participant_time_started	touch_capability	device_type	condition	doc_id	displayed_sequence	seconds_in_viewport	scroll_sequence	liked	reply	hasReply	height
z2q0bs	jysri7zu8	..37e56	2024-10-15 14:44:20	TRUE	Tablet	A	17	35	3.368	79	FALSE		FALSE	479
z2q0bs	jysri7zu8	..37e56	2024-10-15 14:44:20	TRUE	Tablet	A	8	36	2.417	80	FALSE		FALSE	456
z2q0bs	jysri7zu8	..37e56	2024-10-15 14:44:20	TRUE	Tablet	A	7	37	4.411	81	FALSE		FALSE	519
z2q0bs	jysri7zu8	..37e56	2024-10-15 14:44:20	TRUE	Tablet	A	12	38	5.179	82	FALSE		FALSE	454
z2q0bs	jysri7zu8	..37e56	2024-10-15 14:44:20	TRUE	Tablet	A	40	39	2.402	83	FALSE		FALSE	578
z2q0bs	jysri7zu8	..37e56	2024-10-15 14:44:20	TRUE	Tablet	A	38	40	1.968	84	FALSE		FALSE	578
z2q0bs	2budny1	.t4d647	2024-10-15 14:47:58	FALSE	Desktop	A	10	1	3.746	1	FALSE		FALSE	530
z2q0bs	2budny1	.t4d647	2024-10-15 14:47:58	FALSE	Desktop	A	34	2	13.087	3	FALSE		FALSE	169

session_code	participant_code	participant_label	participant_time_started_utc	touch_capability	device_type	condition	doc_id	displayed_sequence	seconds_in_viewport	scroll_sequence	liked	reply	hasReply	height
z2q0bsds	2bsudny1	t4d647	2024-10-15 14:47:58	FALSE	Desktop	A	24	3	5.362	2,4	FALSE		FALSE	520
z2q0bsds	2bsudny1	t4d647	2024-10-15 14:47:58	FALSE	Desktop	A	22	4	6.845	5	FALSE		FALSE	553
z2q0bsds	2bsudny1	t4d647	2024-10-15 14:47:58	FALSE	Desktop	A	39	5	3.642	6	FALSE		FALSE	410

Table 3.3 presents a code-book that describes the variables presented above.

```r
codebook_processed  <- data.table::fread(file = "codebook_processed.csv")

kable(codebook_processed)
```

Table 3.3: Codebook of Processed Data

Scope	Variable	Type	Example	Description
Session	session_code	text	z2q0bsds	Unique session identifier.
Participant	participant_code	text	jyri7zu8	Unique participant identifier generated by oTree.
Participant	participant_label	text	4d647	Unique participant identifier passed from Prolific (PID).
Participant	participant_time_started_utc	text	2024-10-15 14:47:58	Date and time the participant started the study.
Participant	touch_capability	boolean	FALSE	Describes whether the participant's device was recognized as a touch device.
Participant	device_type	boolean	Desktop	Describes whether the participant's device was recognized as a Desktop/Tablet/Mobile device.
Participant	condition	text	A	Condition name the researcher created in her stimuli csv file.

Table 3.3: Codebook of Processed Data

Scope	Variable	Type	Example	Description
Post	doc_id	text	34	Unique social media post identifier the researcher specified in her stimuli file.
Participant x Post	displayed_sequence	numeric	2	The position in which a feed was displayed within the feed.
Participant x Post	seconds_in_viewport	numeric	13.087	
Participant x Post	scroll_sequence	text	2, 4	Describes the sequence in which a participant browsed through the social media post. May contain multiple integers if a participant browsed back and forth.
Participant x Post	liked	boolean	TRUE	Describes wheter a participant liked a social media post.
Participant x Post	reply	text	This is a comment.	Describes what a participant commented on a social media post. NA if nothing.
Participant x Post	hasReply	boolean	TRUE	Describes wheter a participant commented on a social media post.
Participant x Post	height	numeric	520	Describes the vertical resolution (height measured in pixel) in which a post was rendered on the participant's screen.

3.3 Additional Processing

Importantly, none of the two datasets presented here contains any self-reports. To generate you final dataset which includes behavioral measures, the stimuli, as well as self-reports, we advise you to merge the processed data with two other data sources. First, merge the shiny app's output with your survey data using the `participant_label` as a key. Because the DICE app passes that ID to your survey tool, both datasets have these information. Second, merge that data with you stimuli csv (see Chapter 1) based on the `doc_id` and `condition`, if applicable.

Using R and the `{data.table}` package, this process looks as follows:

```
output <- data.table::merge.data.table(x = dice,
                                       y = qualtrics,
```

```
                                       by = "participant_label")

final <- data.table::merge.data.table(x = output,
                          y = stimuli,
                          by = c("doc_id", "condition"))

setorder(final, participant_code, displayed_sequence)
```

4 Case Studies

Two case studies illustrate the application of DICE. These studies are designed to replicate and expand classic context effects within a more ecologically valid social media feed environment using DICE. The focus of the case studies is to illustrate the usage and resulting data streams of DICE rather than advancing theory. This section aims to achieve two objectives. First, the studies demonstrate the ability to manipulate entire feed compositions and sequences, rather than just individual social media posts. Second, they demonstrate how post-level dwell time measurements can be used to approximate attention, complementing insights gained from traditional self-report measures. Substantively, Case Study 1 examines how content surrounding a sponsored post affects brand perceptions, while Case Study 2 explores how the specific position of a post within a feed influences attention and brand recall.

```
options(repos = c(CRAN = "https://cran.r-project.org"))


if (!requireNamespace("groundhog", quietly = TRUE)) {
    install.packages("groundhog")
}

pkgs <- c("magrittr", "data.table", "knitr", "stringr", "english", "moments", "devtools",
          "ggplot2", "patchwork", "scales",  "ggdist", "gghalves", "sjPlot", "gtsummary", "w
          "stargazer", "gt", "gtsummary", "flextable", "kableExtra", "MOTE", "dplyr",
          "multilevelmediation", "mediation", "lme4", "optimx")

groundhog::groundhog.library(pkg = pkgs,
                             date = "2024-10-01")

rm(pkgs)
```

```
set.seed(42)
```

```
layout <- theme(panel.background = element_rect(fill = "white"),
                legend.key = element_rect(fill = "white"),
                panel.grid.major.y = element_line(colour = "grey",
                                                  linewidth = 0.25),
```

```r
                 axis.ticks.y = element_blank(),
                 panel.grid.major.x = element_blank(),
                 axis.line.x.bottom = element_line(colour = "#000000",
                                                   linewidth = 0.5),
                 axis.line.y.left = element_blank(),
                 plot.title = element_text(size = rel(1))
)
```

```r
c_negative <- "#F0941F"
c_positive   <- "#196774"

# c_positive <- "#377E39"
# c_negative <- "#7D3756"
#
# c_positive <- "#009E73"
# c_negative <- "#CC79A7"

scale_color_custom_d <- function() {
  scale_color_manual(values = c(c_orange, c_teal))
}

scale_fill_custom_d <- function() {
  scale_fill_manual(values = c(c_orange, c_teal))
}

scale_color_custom_2d <- function() {
  scale_color_manual(values = c(c_positive, c_negative))
}

scale_fill_custom_2d <- function() {
  scale_fill_manual(values = c(c_positive, c_negative))
}
```

```r
# Create function to get effect size stats
get_effect_size <- function(data) {
  es <- effectsize::cohens_d(brand_attitude ~ condition,
                             data = data,
                             pooled_sd = FALSE)
  list(
    d = round(es$Cohens_d, 3),
    ci_low = round(es$CI_low, 3),
    ci_high = round(es$CI_high, 3),
```

```
    n = nrow(data)
  )
}
```

4.1 Case Study 1: Feed Composition and Context Effects

```
short <- fread(file = "../../oFeeds/studies/brand_safety/data/processed/brand-safety-short.cs
long  <- fread(file = "../../oFeeds/studies/brand_safety/data/processed/brand-safety-long.csv
stimuli_1  <- fread(file = "../templates/docs/data/case-study-2/brazil.csv", na.strings = "")
```

```
short[, condition := as.factor(ifelse(test = condition == "inappropriate", yes = "unsafe", no
long[, condition := as.factor(ifelse(test = condition == "inappropriate", yes = "unsafe", no
```

```
all_short <- copy(short)
short <- short[is.finite(log_dwell_pixel)] # [is_flood_aware == FALSE]
```

```
long[, displayed_sequence := as.integer(displayed_sequence)]
long[, seconds_in_viewport := as.numeric(seconds_in_viewport)]
long[, log_dwell_pixel := as.numeric(log_dwell_pixel)]
long[, log_dwell_time := as.numeric(log_dwell_time)]
```

```
subset <- long[is.finite(log_dwell_pixel) &
                     displayed_sequence < 19 &
                     displayed_sequence > 2]
```

```
short[, log_time_spent := log(time_spent_on_page - seconds_in_viewport)]
```

Case Study 1 demonstrates DICE's capability to study context effects with high experimental control and study realism. This study illustrates how researchers can systematically manipulate the broader context (i.e., the composition of the feed) in which users encounter a specific post (in this case, a sponsored post from a brand). Substantively, Case Study 1 examines the issue of brand safety in social media advertising. Brand safety refers to the idea that advertising should not appear in contexts that could harm a brand's reputation (Fournier and Srinivasan 2023). This concern is particularly relevant for social media advertising, where platforms use automated systems to place ads in dynamic, user-generated content environments. These systems often lack the nuanced understanding needed to identify potentially problematic contexts that could harm a brand. While industry reports suggest that up to 75% of brands have experienced such unsafe brand exposures (Ahmad et al. 2024; GumGum Inc. 2017), examining these effects in the field risks apparent brand damage.

4.1.1 Experimental Design

To test how brand (un)safe contexts affect brand perceptions, we created two social media feeds that were identical in structure but varied in their content surrounding a sponsored post (see Figure 4.1[1] for exemplary screenshots of the brand (un)safe feeds). The sponsored post in both conditions was an ad by the airline KLM promoting flights to Brazil.

In the brand-safe condition, the sponsored post was surrounded by actual organic posts covering Brazil scraped from the web. In the brand-unsafe condition, however, the sponsored post was surrounded by another set of scraped organic posts about the severe flooding that occurred during the time of the study. Such a situation is precisely the type of contextual mismatch that automated systems can create and managers fear due to the adverse consequences for brands (Ahmad et al. 2024; GumGum Inc. 2017). In both conditions, the sponsored post was always fixed in the fifth position, whereas the order of the organic posts varied randomly.

4.1.2 Procedure

We recruited 982 US-American participants on Prolific (M_{age} = 39 years; 56% female) to participate in the study. Participants browsed the simulated feed on their own devices (75% desktop, 21% mobile, and 4% tablet). After scrolling through the feed, participants were redirected to a Qualtrics survey in which they first provided demographic information as a filler task. Next, participants reported their brand attitude toward KLM using three seven-point scales (1 = "Negative/Unfavorable/Dislike" and 7 = "Positive/Favorable/Like"; $\alpha = 0.96$). Finally, we assessed participants' awareness of the Brazil flooding. For this and all studies, all stimuli, materials, data, and analysis code are available on the Open Science Framework (OSF): https://osf.io/2xs5c/?view_only=4bf95d2a2c8449218b5fa7cd288f626a.

4.1.3 Stimuli

Building on the CSV file structure introduced in Case Study 1, we created a file containing two distinct sets of content: nineteen organic posts for each experimental condition, plus one sponsored post that needed to appear in both feeds. To ensure the sponsored post would appear in both conditions while maintaining DICE's CSV structure, we entered the sponsored post twice in the file - once for each condition. This resulted in a file with forty rows total: nineteen organic posts for each condition plus the sponsored post appearing twice. Each post's content was specified in columns such as `<text>` and `<username>`. We used the `<condition>` parameter to distinguish between our brand-safebrand-unsafe feeds, assigning each row (i.e.,

[1]The figure shows how DICE enables the controlled manipulation of feed contexts. The identical sponsored post by KLM (highlighted) appears in the "brand-safe" feeds (left) surrounded by neutral Brazil content versus "brand-unsafe" feeds (right) where it appears alongside posts about the Brazil flooding disaster. The sponsored post always remains in the fifth position, while surrounding organic posts are fully randomized. An example feed for the brand-unsafe condition is accessible at https://tiny.cc/DICE1 .

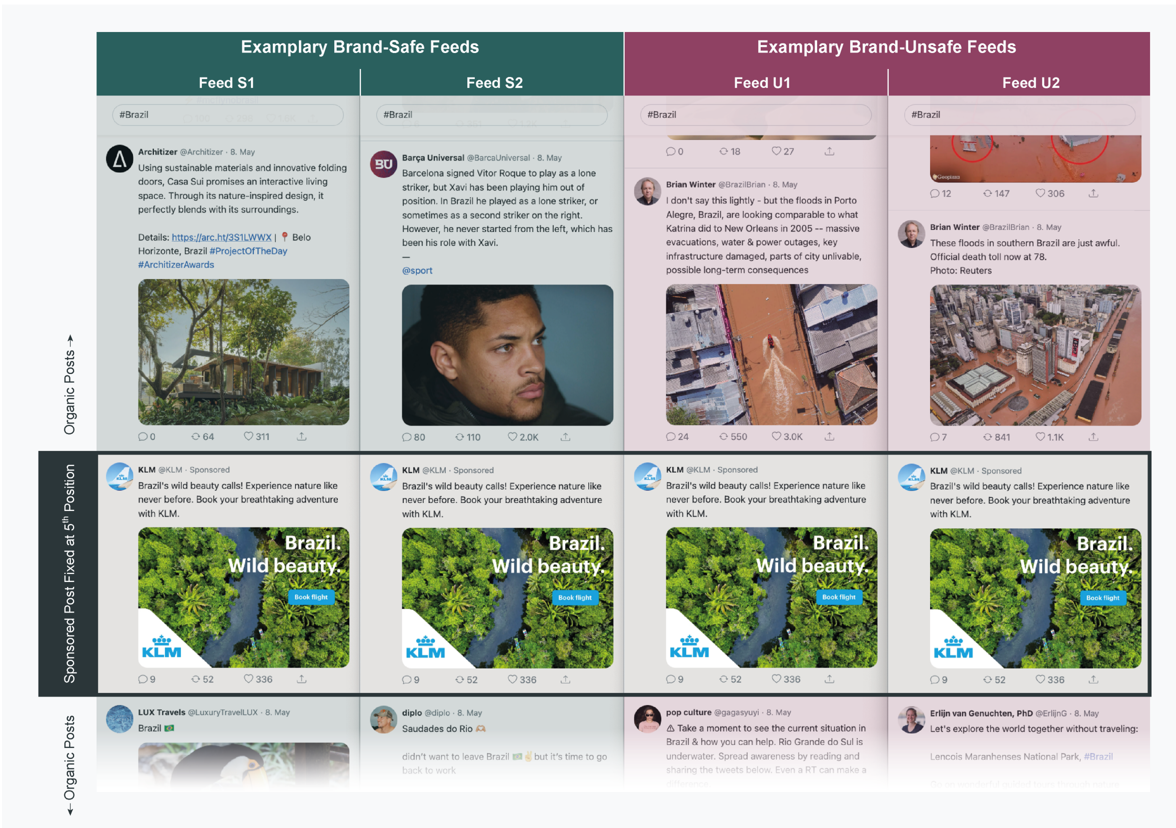


Figure 4.1: Exemplary DICE Feeds from Case Study 1

each post) to its respective condition. Similar to Case Study 1, we left the `<sequence>` column empty for organic posts to enable randomization, with one key exception: the KLM sponsored post was assigned a fixed `<sequence>` value of "5" to ensure consistent positioning across conditions. We marked this post as sponsored using the `<sponsored>` parameter and included a KLM landing page URL in the `<target>` column for participants who clicked on the ad. The resulting CSV file was uploaded to an online repository to generate a URL for the DICE app.

Table 4.1 shows an excerpt of the exact CSV files we used to create the stimuli for this study. You can download that file here.

```
text_columns <- names(stimuli_1)[sapply(stimuli_1, is.character)]

stimuli_1[, (text_columns) := lapply(.SD, function(x) {
  x <- str_replace_all(x, "\n(?=[^.,!?;:])", " ")
  x <- str_replace_all(x, "\n(?=[.,!?;:])", "")
  x <- str_replace_all(x, "@", "")
  str_replace_all(x, "<br>", " ")
}), .SDcols = text_columns]

kable(stimuli_1[c(5, 16:23)])
```

Table 4.1: CSV File Used in Study 1

doc_id	date_time	text	media	alt_text	likes	reposts	replies	user_name	user_handle	user_description	user_image	user_followers	sponsored	target	condition	sequence	commented_post	source
5	08.05.24 09:30	Brazil's wild beauty calls! Experience nature like never before. Book your breathtaking adventure with KLM.	https://i.postimg.cc/MGQtKsh2/brazil-wild-beauty.webp	NA	[illegible]	[illegible]	9	KLM	KLM	Official global account of KLM. FAQ about flying from AMS: http://klmf.ly/Schiphol. To learn how we can assist you, please check: http://klmf.ly/ContactCentre.	https://pbs.twimg.com/profile_images/1807736403385561088/HaxXwA-t_400x400.jpg	[illegible]	[illegible]	https://www.klm.com.br/en-br/flights-to-brazil	[illegible]	[illegible]	NA	
16	08.05.24 09:30	Madonna shines in a custom Jean Paul Gaultier Haute Couture once again as she took the stage for her legendary grand finale for her 'Celebration Tour' in Brazil	https://pbs.twimg.com/media/GM5Qx-cZX-MAA4nDR?format=jpg&name=large	NA	[illegible]	[illegible]	[illegible]	Jean Paul Gaultier	[illegible]	Jean Paul Gaultier	https://pbs.twimg.com/profile_images/1780899814722473984/[illegible]_400x400.[illegible]	[illegible]	[illegible]	NA	[illegible]	0	https://twitter.com/JPGaultier/status/178745502978949[illegible]	

Table 4.1: CSV File Used in Study 1

doc_id	date_time	text	media	alt_text	likes	reposts	replies	username	handle	user_description	user_image	user_followers	sponsored	target	condition	sequence	comment	source_post
17	08.05.24 09:30	Waved Woodpecker (Celeus undatus) in Brazil Schuler Franz © #Brazil #nature #wildlife #photography	https://pbs.twimg.com/media/GHOD_EjXkAACtmg?format=jpg&name=medium	NA	1198	206	[illegible]	Mbark Cherguia	[illegible]	Nature lover | Capturing nature's beauty | Open to collaborations | Follow for stunning landscapes! | Check out Geniustechw for art, tech & culture	https://pbs.twimg.com/profile_images/1748768033231798272/yxa0DbMM4_400x40	2329	NA	[illegible]	pro-private	0	[illegible]	https://twitter.com/MbarkCherguia/status/176189701668101

Table 4.1: CSV File Used in Study 1

doc_id	date_time	text	media	alt_text	likes	reposts	replies	username	handle	user_description	user_image	user_followers	sponsored	target	condition	sequence	commented	source	comment_post
18	08.05.24 09:30	Using sustainable materials and innovative folding doors, Casa Sui promises an interactive living space. Through its nature-inspired design, it perfectly blends with its surroundings. Details: https://arc.ht/3S1LWWX | Belo Horizonte, Brazil #ProjectOfTheDay #ArchitizerAwards	https://pbs.twimg.com/media/GEChwN4XQAA17m-?format=jpg&name=medium	NA	311	64	0	Architizer	Architizer	Architizer is the home of architectural inspiration.	https://pbs.twimg.com/profile_images/842415551955759109/cmi8egmBt_400x400.jpg	1203859	NA	no	promo-private	0	1		https://twitter.com/Architizer/status/174757451334302

Table 4.1: CSV File Used in Study 1

doc_id	datetime	text	media	alt_text	likes	reposts	replies	username	handle	user_description	user_image	user_followers	sponsored	target	condition	sequence	commented	source	post
19	08.05.24 09:30	The flora in Pará, Brazil, is essential for its economy, but preserving #biodiversity is equally important. Integrating it into value chains benefits the climate, people, and the economy. Learn more with this nature_org infographic!	https://pbs.twimg.com/media/GAvWidjb0AAMYBU?format=jpg&name=4096x4096	NA	142	60	3	UN Biodiversity	UNBiodiversity	Official account of the Secretariat of the Convention on Biological Diversity. Our Acting Executive Secretary hdavidcooper	https://pbs.twimg.com/profile_images/1737083733147070464/BigdpyhW_400x400.	203847		NA	appropriate	NA	0		https://twitter.com/UNBiodiversity/status/173516270346589

Table 4.1: CSV File Used in Study 1

doc_id	datetime	text	media	alt_text	likes	reposts	replies	username	handle	user_description	user_image	user_followers	sponsored	target	condition	sequence	comment	source	post
20	08.03.24 09:30	Brazil overlaid onto Europe	https://pbs.twimg.com/media/GNCfMeJXUAAn61W?format=jpg&name=large	NA	2899	33	37	Epic Maps	Locations	Educational and informative world Maps . Scroll Down & Get More Knowledge. We do not own any content posted. Dm for removal/credit.	https://pbs.twimg.com/profile_images/1369972073758564355/a7Pns4tx_400x400.jpg	850049	NA		appropriate	0			https://twitter.com/Locations/status/17881040925749

Table 4.1: CSV File Used in Study 1

doc_id	datetime	text	media	alt_text	likes	reposts	replies	user_name	user_handle	user_description	user_image	user_followers	sponsored	target	condition	sequence	comment	source_post
21	08.05.24 09:30	I don't say this lightly - but the floods in Porto Alegre, Brazil, are looking comparable to what Katrina did to New Orleans in 2005 – massive evacuations, water & power outages, key infrastructure damaged, parts of city unlivable, possible long-term consequences	https://pbs.twimg.com/media/GNCy3PWoAAGfil?format=jpg&name=900x900	NA	3025	624	[illegible]	Brian Winter	BrazilBrian	Editor-in-chief AmerQuarterly. Latin American politics & barbecue. " "O mais brasileiro dos texanos." " Opinions mine.	https://pbs.twimg.com/profile_images/1696273071898800128/Bcc439sta_400x400.jpg	71129	NA	0	inappropriate	[illegible]	[illegible]	https://twitter.com/BrazilBrian/status/17881409020676 3

Table 4.1: CSV File Used in Study 1

doc_id	datetime	text	media	alt_text	likes	reposts	replies	username	user_handle	user_description	user_image	user_followers	sponsor	target	condition	sequence	comment	source	commented_post
22	08.05.24 09:30	Sentinel-2 satellite imagery from earlier today shows the incredible extent of the historic flooding in Rio Grande do Sul, Brazil. Just terrible.	https://saltwire.[illegible]/2024/05/7/brazil-floods-death-toll-rises-to-90-dozens-still-stranded.jpg?cs=srgb&fit=crop&h=568&w=847&dpr=2&auto=enhance%2Cformat%2Cco[illegible]	NA	[illegible]	[illegible]	[illegible]	[illegible]hel Belgherze	[illegible]	[illegible] extreme weather events around the world. Also interested in climate dynamics, geohazards & satellite remote sensing. Views are my own.	https://pbs.twimg.com/profile_images/1729266199496736768/[illegible]	[illegible]	NA	[illegible]	inappropriate	[illegible]	NA	https://twitter.com/WxNB_/status/[illegible]	[illegible]

Table 4.1: CSV File Used in Study 1

doc_id	datetime	text	media	alt_text	likes	reposts	replies	username	handle	user_description	user_image	user_followers	sponsored	target	condition	sequence	comment	source_post
23	08.05.24 09:30	The sheer scale of the flooding in the metropolitan area of Porto Alegre, Brazil, which is home to over 4 million people, is hard to comprehend. Thousands of homes flooded, towns cut off, dozens killed, hundreds missing.	NA	NA	2231	1415	51	Nahel Belgherze	WxNB_	Covering extreme weather events around the world. Also interested in climate dynamics, geohazards & satellite remote sensing. Views are my own.	https://pbs.twimg.com/profile_images/1729266199496736768/[illegible]	51767	NA	NA	inappropriate	0	–	https://twitter.com/WxNB_/status/[illegible]

```r
# Omnibus test of joint orthogonality with randomization inference

# Full model with covariates
full_model <- lm(as.numeric(as.factor(condition)) ~ female + age, data = short)

# Null model without covariates
null_model <- lm(as.numeric(as.factor(condition)) ~ 1, data = short)

anova_result <- anova(null_model, full_model)

# Observed F-statistic from ANOVA
observed_f_stat <- anova_result$F[2]

# Set up randomization inference
n_simulations <- 1000
```

```r
simulated_f_stats <- numeric(length = n_simulations)

for (i in 1:n_simulations) {
  # Shuffle the treatment labels
  short[, shuffled_condition := sample(condition)]

  # Refit the full and null models with shuffled treatment
  shuffled_full_model <- lm(as.numeric(as.factor(shuffled_condition)) ~ female + age, data =
  shuffled_null_model <- lm(as.numeric(as.factor(shuffled_condition)) ~ 1, data = short)

  # Perform ANOVA on shuffled data
  shuffled_anova_result <- anova(shuffled_null_model, shuffled_full_model)

  # Store the F-statistic
  simulated_f_stats[i] <- shuffled_anova_result$F[2]
}

# Calculate the p-value based on randomization inference
balance_p <- sprintf(fmt = "%.2f", mean(simulated_f_stats >= observed_f_stat))
```

4.1.4 Participants and Randomization Checks

Participants were randomly assigned to view either the brand-safe feed (featuring general Brazil-related content) or the brand-unsafe feed (featuring flood coverage). A key advantage of DICE over observational and platform studies is its ability to implement true random assignment, allowing us to isolate the effect of context while canceling out other factors that might influence brand perception. To validate DICE's randomization functionality, we examined the balance between treatment groups: as illustrated in Table 4.2, the two treatment groups do not exhibit differences in observables. Following Kerwin, Rostom, and Sterck (2024), we also found support for balanced conditions in an omnibus test of joint orthogonality with randomization inference ($p = 0.39$).

```r
balance_fem <- lm(formula = female ~ condition, data = short)
balance_age <- lm(formula = age ~ condition, data = short)

balance_table <- data.table(variables = c("Mean Age (Years)", "Female (Percent)"),
                            safe = c(short[condition == "safe",
                                           mean(age, na.rm = TRUE)],
                                     short[condition == "safe",
                                           mean(female, na.rm = TRUE)]*100),
                            unsafe = c(short[condition == "unsafe",
```

```
                                          mean(age, na.rm = TRUE)],
                                    short[condition == "unsafe",
                                          mean(female, na.rm = TRUE)]*100),
                          difference = c(summary(balance_age)$coefficients[2, 1],
                                         summary(balance_fem)$coefficients[2, 1]*100),
                          pairwise_p = c(summary(balance_age)$coefficients[2, 4],
                                         summary(balance_fem)$coefficients[2, 4]))

balance_table %>% kable(digits = 3,
                        col.names = c("Covariate",
                                      "Safe",
                                      "Unsafe",
                                      "Difference",
                                      "p-value"))
```

Table 4.2: Covariate Balance Across Conditions

Covariate	Safe	Unsafe	Difference	p-value
Mean Age (Years)	38.848	38.522	-0.326	0.702
Female (Percent)	58.439	54.262	-4.177	0.194

4.1.5 Data

Our dataset comprises 955 participants and 15,343 observations at the participant × post level. Whereas Case Study 1 analyzed multiple sponsored posts across participants, here we focus on a single sponsored post (i.e., the KLM ad) viewed by all participants, which simplifies our analytical approach. Our final sample comprises 955 observations on the participant level, which is slightly less than the expected one observation per participant due to connectivity issues: no dwell time data were recorded for around 0.00% of the sponsored posts.

This simplified "short" data structure has two convenient methodological implications. First, because we analyze one observation per participant rather than nested data, we can apply simpler methods such as ordinary least squares (OLS) regressions in our analyses. Second, because we only focus on one sponsored post, we do not need to divide our dwell time measure by the post's height as we did in Case Study 1. Both aspects increase the interpretability of our results.

Table 4.3 shows an excerpt of the processed data to illustrate its nested (i.e., "long") structure.

```r
tmp <- short[,
            .(`Participant ID` = participant_label,
              `Position in Feed` = 5,
              Likes = liked,
              Replies = hasReply,
              `Seconds in Viewport` = seconds_in_viewport,
              `Dwell Time` = log_dwell_time,
              `Brand Attitudes` = brand_attitude,
              Age = age,
              Female = female,
              Desktop = as.logical(is_desktop),
              Recall = klm_uncued_recall)]

kable(tmp[35:45])
```

Table 4.3: Processed Data Analyzed in Study 2

Participant ID	Position in Feed	Likes	Replies	Seconds in Viewport	Dwell Time	Brand Attitudes	Age	Female	Desktop	Recall
59e833e72f63d30001c8fb56	5	FALSE	FALSE	1.783	0.5782971	4.000000	36	FALSE	TRUE	FALSE
59f226a9d6380600018b2923	5	FALSE	FALSE	6.393	1.8552035	5.333333	60	TRUE	TRUE	TRUE
5a0ef93279f96a0001c73f25	5	FALSE	FALSE	3.500	1.2527630	4.000000	31	TRUE	TRUE	FALSE
5a42c80ddaea400001ac24e0	5	FALSE	FALSE	10.239	2.3262045	5.000000	47	TRUE	TRUE	FALSE
5a542b96e0cf3d0001260df1	5	FALSE	FALSE	1.111	0.1052605	4.000000	38	TRUE	TRUE	FALSE
5a5c2c2eeedc320001429df1	5	FALSE	FALSE	20.422	3.0166124	4.000000	33	TRUE	TRUE	FALSE
5a7618b18fe2dc0001057243	5	FALSE	FALSE	3.866	1.3522206	4.000000	43	TRUE	TRUE	FALSE
5a78e410ae9a0b0001a97274	5	TRUE	FALSE	5.915	1.7774915	5.000000	32	TRUE	FALSE	FALSE
5a91756d6475f900019f9051	5	FALSE	FALSE	4.484	1.5005155	4.000000	43	TRUE	FALSE	FALSE
5a9fc2b66475f90001a0219a	5	FALSE	FALSE	3.165	1.1521531	1.333333	31	TRUE	TRUE	FALSE
5aaf4b0ce1546900019b015de	5	FALSE	FALSE	0.301	-1.2006450	4.000000	43	TRUE	TRUE	FALSE

4.1.6 Results and Discussion

```r
lm_main <- lm(brand_attitude ~ condition, data = short)

lm_x1 <- lm(brand_attitude ~ condition * log_dwell_time, data = short)
lm_x2 <- lm(brand_attitude ~ condition * log_dwell_time + log_time_spent, data = short)
```

```r
lm_x3 <- lm(brand_attitude ~ condition * log_time_spent, data = short)

lm_dwell <- lm(log_dwell_time ~ condition, data = short)
```

```r
model_summary <- summary(lm_main)
df_main <- model_summary$df[2]

beta <- model_summary$coefficients[2, 1]  # F-statistic value
se   <- model_summary$coefficients[2, 2]  # SE
t_value <- model_summary$coefficients[2, 3]
f_value <- model_summary$fstatistic[1]  # F-statistic value
f_df1 <- model_summary$fstatistic[2]    # degrees of freedom for the model
f_df2 <- model_summary$fstatistic[3]    # degrees of freedom for the residuals
p_value <- pf(f_value, f_df1, f_df2, lower.tail = FALSE)  # p-value from F-statistic
cohensD <- apa(effectsize::cohens_d(brand_attitude ~ condition, data = short, pooled_sd = FAl
```

```r
model_summary <- summary(lm_x1)

coefficients <- coef(model_summary)
df_moderation <- model_summary$df[2]  # Residual degrees of freedom

condition_beta <- coefficients[2, 1]
condition_se <- coefficients[2, 2]
condition_t <- coefficients[2, 3]
condition_p <- coefficients[2, 4]

dwell_beta <- coefficients[3, 1]
dwell_se <- coefficients[3, 2]
dwell_t <- coefficients[3, 3]
dwell_p <- coefficients[3, 4]

interaction_beta <- coefficients[4, 1]
interaction_se <- coefficients[4, 2]
interaction_t <- coefficients[4, 3]
interaction_p <- coefficients[4, 4]
```

```r
model_summary <- summary(lm_dwell)

coefficients <- coef(model_summary)
dwell_condition_df <- model_summary$df[2]
```

```r
dwell_condition_beta <- coefficients[2, 1]
dwell_condition_se <- coefficients[2, 2]
dwell_condition_t <- coefficients[2, 3]
dwell_condition_p <- coefficients[2, 4]
```

```r
model_summary <- summary(lm_x2)

coefficients <- coef(model_summary)
df_robustness <- model_summary$df[2]

robustness_beta <- coefficients[5, 1]
robustness_se <- coefficients[5, 2]
robustness_t <- coefficients[5, 3]
robustness_p <- coefficients[5, 4]
```

Brand attitudes toward KLM were significantly less positive in the brand-unsafe feed condition ($M_u = 4.310$, $SD_u = 1.366$) compared to the brand-safe feed condition ($M_s = 4.821$, $SD_s = 1.161$, $b = -.510$, $SE = .082$, $t(953) = -6.217$, $p = .000$, $d = .403$).

To further explore the interplay between the KLM ad's context and brand attitudes, we examined whether the dwell time of the ad moderated the previously reported main effect of context. An OLS regression revealed a statistically significant interaction between the context's brand safety and dwell time ($b = -.302$, $SE = .068$, $t(951) = -4.455$, $p = .000$), indicating that the lack of attention shapes how context affects brand attitudes (see Figure 4.2). This suggests that the negative effect of an unsafe context on brand attitude only emerged when participants spent a sufficient amount of time viewing the sponsored post. In contrast, among those participants with minimal dwell time, there was little difference in brand attitudes between safe and unsafe contexts. The main effects of brand safety ($b = -.020$, $SE = .137$, $t(951) = -.143$, $p = .886$) and dwell time ($b = .096$, $SE = .050$, $t(951) = 1.929$, $p = .054$) were not significant. The ad's dwell time did not vary across brand safety conditions ($b = -.002$, $SE = .078$, $t(953) = -4.662$, $p = .982$). Finally, this moderation is robust to alternative model specifications (see Section 4.1.7) where we repeated the same analysis while controlling for the dwell time allocated to all organic posts ($b = -.316$, $SE = .068$, $t(950) = -4.662$, $p = .000$).

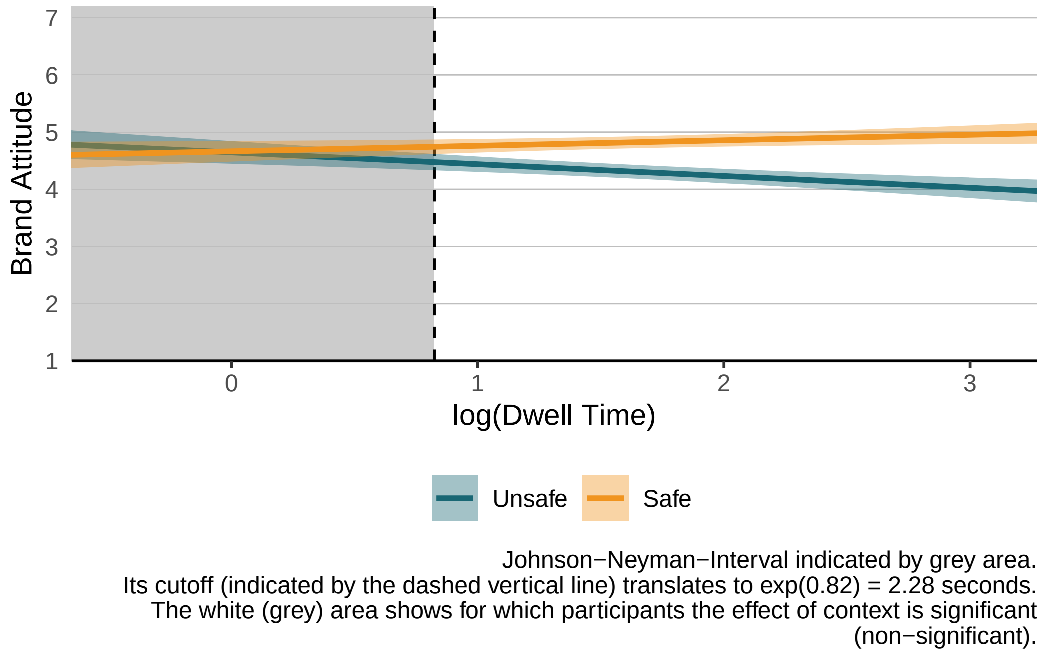
```r
q_95 <- short[, quantile(x = log_dwell_time, probs = 0.95, na.rm = TRUE)]
q_05 <- short[, quantile(x = log_dwell_time, probs = 0.05, na.rm = TRUE)]

# Find Johnson Neyman through simulation

# sub <- short[is.finite(log_dwell_time)]
sub <- copy(short)
tmp <- data.table(log_dwell_time = seq(from = min(sub$log_dwell_time,
```

```
                                                          na.rm = TRUE),
                                        to = max(sub$log_dwell_time,
                                                        na.rm = TRUE),
                                        length.out = 100),
                  condition = rep(x = c("safe", "unsafe"),
                                  each = 100))

lm_jn <- lm(brand_attitude ~ log_dwell_time * condition, data = sub)

predictions <- predict(lm_jn, newdata = tmp, interval = "confidence")

tmp[, c("fit", "lwr", "upr") := as.data.table(predictions)]

safe <- tmp[condition == "safe"]
unsafe <- tmp[condition == "unsafe"]

non_overlap_point <- which(safe$lwr > unsafe$upr)

jn_point <- safe$log_dwell_time[non_overlap_point[1]-1]
jn_seconds <- round(exp(jn_point), digits = 2)
```

```
sub[condition == "unsafe", condition := "Unsafe"]
sub[condition == "safe", condition := "Safe"]

ggplot(data = sub,
       mapping = aes(x = log_dwell_time,
                     y = brand_attitude,
                     fill = condition,
                     col = condition)) +
  geom_rect(aes(xmin = -Inf, xmax = jn_point, ymin = 1, ymax = 7.2),
            fill = "#cccccc", col = NA, alpha = 0.025) +
  geom_vline(xintercept = jn_point, lty = 2) +
  geom_smooth(method = "lm", formula = "y ~ x") +
  layout +
  labs(x = "log(Dwell Time)",
       y = "Brand Attitude",
       # title = "Moderating Effect of Dwell Time on Brand Attitude",
       caption = paste0("Johnson-Neyman-Interval indicated by grey area.
                        Its cutoff (indicated by the dashed vertical line) translates to exp
                        round(jn_point, digits = 2), ") = ",
       jn_seconds,
       " seconds.
```

```
      The white (grey) area shows for which participants the effect of context is significan
      (non-significant)."), #The x-axis ranges from the 5th to the 95th percentile of the d
      color = "",
      fill = "") +
  scale_y_continuous(expand = expansion(mult = c(0, 0)), breaks = 1:7) +
  scale_x_continuous(expand = expansion(mult = c(0, 0))) +
  coord_cartesian(xlim = c(q_05, q_95),
                  ylim = c(1, 7.2)) +
  scale_fill_custom_2d() +
  scale_color_custom_2d() +
  layout +
  theme(legend.position = "bottom")
```

Figure 4.2: Moderation of the Effect of Context on Brand Attitudes by Dwell Time



From a substantive perspective, the findings provide experimental support for brand safety concerns and reveal an intuitive nuance: contextual misplacements primarily harm brand perceptions when consumers pay sufficient attention to the content. When attention is minimal (indicated by minimal dwell times), the negative impact of unsafe contexts appears to be neutralized. From a methodological perspective, this study illustrates how researchers can manipulate entire feed compositions rather than just single social media posts. It also showcases how DICE's dwell time data can be interpreted as a proxy of attention. A lack of dwell time for a post, however, implies a lack of attention to this content.

4.1.7 Robustness Checks

To test the robustness of our findings, we conducted additional analyses controlling for participants' dwell time on organic posts that constituted our experimental manipulation. Since the brand safety manipulation involved varying the content of organic posts surrounding the sponsored ad (brand-safe vs. brand-unsafe), it was important to verify that our results were not simply driven by differential attention to these organic posts across conditions.

Models 1 and 2 in Table Table 4.4 feature the main findings reported in the manuscript. Model 3 adds the dwell time on all organic posts as a covariate, while Model 4 replaces the dwell time on the sponsored post with the dwell time on the organic posts. Both models demonstrate that our key findings (i.e., the main effects as well as the interaction between brand safety and dwell time) were robust. The interaction effect (Brand-Unsafe × Dwell Time KLM) remained significant, indicating that the moderating role of attention on context effects persists even after accounting for engagement with the manipulated organic content. Model 4 additionally shows that dwell time on organic posts also moderates the brand safety effect.

```r
n_obs <- format(nobs(lm_x1), big.mark = ",")

tbl_1 <- tbl_regression(
  lm_main,
  estimate_fun = ~ style_number(.x, digits = 3),
  pvalue_fun = label_style_pvalue(digits = 3, zero.print = "."),
  conf.int = FALSE,
  show_single_row = "condition",
  intercept = TRUE,
  add_estimate_to_reference_rows = FALSE
) |>
  modify_column_unhide(columns = std.error) |>
  add_glance_table(include = r.squared)

tbl_2 <- tbl_regression(
  lm_x1,
  estimate_fun = ~ style_number(.x, digits = 3),
  pvalue_fun = label_style_pvalue(digits = 3, zero.print = "."),
  conf.int = FALSE,
  show_single_row = "condition",
  intercept = TRUE,
  add_estimate_to_reference_rows = FALSE
) |>
  modify_column_unhide(columns = std.error) |>
  add_glance_table(include = r.squared)
```

```r
tbl_3 <- tbl_regression(
  lm_x2,
  estimate_fun = ~ style_number(.x, digits = 3),
  pvalue_fun = label_style_pvalue(digits = 3, zero.print = "."),
  conf.int = FALSE,
  show_single_row = "condition",
  intercept = TRUE,
  add_estimate_to_reference_rows = FALSE
) |>
  modify_column_unhide(columns = std.error) |>
  add_glance_table(include = r.squared)

tbl_4 <- tbl_regression(
  lm_x3,
  estimate_fun = ~ style_number(.x, digits = 3),
  pvalue_fun = label_style_pvalue(digits = 3, zero.print = "."),
  conf.int = FALSE,
  show_single_row = "condition",
  intercept = TRUE,
  add_estimate_to_reference_rows = FALSE
) |>
  modify_column_unhide(columns = std.error) |>
  add_glance_table(include = r.squared)

table <- tbl_merge(
  tbls = list(tbl_1, tbl_2, tbl_3, tbl_4),
  tab_spanner = c("Model 1", "Model 2", "Model 3", "Model 4")
) |>
  # Remove empty rows
  modify_table_body(
    ~ .x |>
      dplyr::filter(
        !(is.na(estimate_1) & is.na(estimate_2) & is.na(estimate_3) & is.na(estimate_4))
      )
  ) |>
  modify_table_body(
    ~ .x |>
      dplyr::mutate(
        label = case_when(
          label == "(Intercept)" ~ "Constant",
          label == "condition" ~ "Brand-Unsafe",
          label == "log_dwell_time" ~ "Dwell Time (KLM)",
```

```
        label == "log_time_spent" ~ "Dwell Time (Organic)",
        label == "unsafe * log_dwell_time" ~ "Brand-Unsafe × Dwell Time (KLM)",
        label == "unsafe * log_time_spent" ~ "Brand-Unsafe × Dwell Time (Organic)",
        label == "condition * log_dwell_time" ~ "Brand-Unsafe × Dwell Time (KLM)",
        label == "condition * log_time_spent" ~ "Brand-Unsafe × Dwell Time (Organic)",
        TRUE ~ label
      )
    )
  ) |>
  modify_table_body(
    ~ .x |>
      dplyr::arrange(
        factor(label, levels = c(
          "Brand-Unsafe",
          "Dwell Time (KLM)",
          "Dwell Time (Organic)",
          "Brand-Unsafe × Dwell Time (KLM)",
          "Brand-Unsafe × Dwell Time (Organic)",
          "Constant",
          "R²"
        ))
      )
  ) |>
  modify_header(
    label ~ "",
    estimate_1 ~ "b",
    estimate_2 ~ "b",
    estimate_3 ~ "b",
    estimate_4 ~ "b",
    std.error_1 ~ "SE",
    std.error_2 ~ "SE",
    std.error_3 ~ "SE",
    std.error_4 ~ "SE",
    p.value_1 ~ "p",
    p.value_2 ~ "p",
    p.value_3 ~ "p",
    p.value_4 ~ "p"
  )

table
```

Table 4.4: Estimates of Brand Attitude as a Function of Brand Safety and D

	Model 1			Model 2			
	b	SE[1]	p	b	SE[1]	p	b
Brand-Unsafe	-0.510	0.082	<0.001	-0.020	0.137	0.886	-0.006
Dwell Time (KLM)				0.096	0.050	0.054	0.057
Dwell Time (Organic)							0.175
Brand-Unsafe × Dwell Time (KLM)				-0.302	0.068	<0.001	-0.316
Brand-Unsafe × Dwell Time (Organic)							
Constant	4.821	0.058	<0.001	4.664	0.099	<0.001	3.901
R^2	0.039			0.062			0.070

[1]SE = Standard Error

4.2 Case Study 2: Position Effects and Competing Attention Between Brands

```
data <- fread(file = "../../oFeeds/studies/meme_feed/data/processed/meme-feed-data.csv", na.s
stimuli_2  <- fread(file = "../templates/docs/data/case-study-1/9gag.csv", na.strings = "")
```

```
subset <- data[is.finite(log_dwell_pixel) &
                     displayed_sequence < 39 &
                     displayed_sequence > 2]
```

Rather than focusing on a single post, Case Study 2 examines how multiple brands compete for user attention within the same feed—a common scenario in social media advertising where multiple advertisers target the same user and try to "cut through the clutter" of both organic content and competing ads (Ordenes et al. 2019). Case Study 2 also illustrates how to approximate attention patterns across an entire feed containing multiple sponsored posts from different advertisers in the same product category. By studying how users engage with multiple sponsored posts in a single feed, we can better understand the dynamics of attention allocation in social media environments where brands compete for attention simultaneously, building on related memory and context effects studied in traditional advertising environments (Pieters, Warlop, and Wedel 2002). Specifically, we examine how the position of sponsored posts affects brand recall when multiple brands are presented within the same feed, and how the dwell times captured through DICE help explain these effects.

4.2.1 Experimental Design

To investigate the relationship between ad placement and recall, we simulated social media feeds containing both organic and sponsored posts. Whereas the set of organic and sponsored posts was the same for all participants, the sequence in which the participants were exposed to these posts was unique for every participant as we randomized the sequence between subjects.

To investigate the relationship between ad placement and brand recall, we simulated a social media feed containing both organic and sponsored posts. Whereas the set of organic and sponsored posts was the same for all participants, we randomized the sequence in which participants were exposed to these posts between subjects.

The feed featured thirty-five organic posts and five consumer electronics ads (sponsored posts; see Figure 4.3[2] for example feeds). We selected multiple ads from established consumer electronics brands (i.e., Apple, Bose, Nintendo, Samsung, Whoop).[3] The sponsored posts were actual ads retrieved from Facebook's Ad Library, a publicly accessible database that archives advertisements run by advertisers across Meta's platforms. The selected sponsored posts shared similar basic characteristics, including the presence of a product image, brand logo, and brief text as part of the advertisement. The organic content featured a collection of memes taken from the platform 9gag, a popular internet meme collection with over sixteen million followers on X. We intentionally chose memes as organic content to offer a conservative test of how branded content "competes" for attention on social media, particularly among younger users (Malodia et al. 2022).

4.2.2 Procedure

```
N <- subset[, length(unique(participant_label))]
share_desktop <- round(100 * subset[device_type == "Desktop",
                                     length(unique(participant_label))] / N)
share_mobile  <- round(100 * subset[device_type == "Mobile",
                                     length(unique(participant_label))] / N)
share_tablet  <- round(100 * subset[device_type == "Tablet",
                                     length(unique(participant_label))] / N)
```

[2]The figure displays five exemplary feeds used in Case Study 2. The positioning of sponsored and organic content in the study was fully randomized. The sponsored posts are highlighted for expositional clarity. Each feed contained identical content: the same thirty-five organic posts and five sponsored posts, as shown in the second column, where both the BOSE and Apple ads are displayed. An example feed is accessible at https://tiny.cc/DICE2.

[3]This approach aligns with recent methodological work on stimulus sampling (Simonsohn, Montealegre, and Evangelidis 2024) which demonstrates that in studies focused on a single manipulated variable (a sponsored post's position in our case), using diverse stimuli helps ensure effects are not driven by idiosyncratic characteristics of any particular advertisement, increasing both internal and ecological validity.

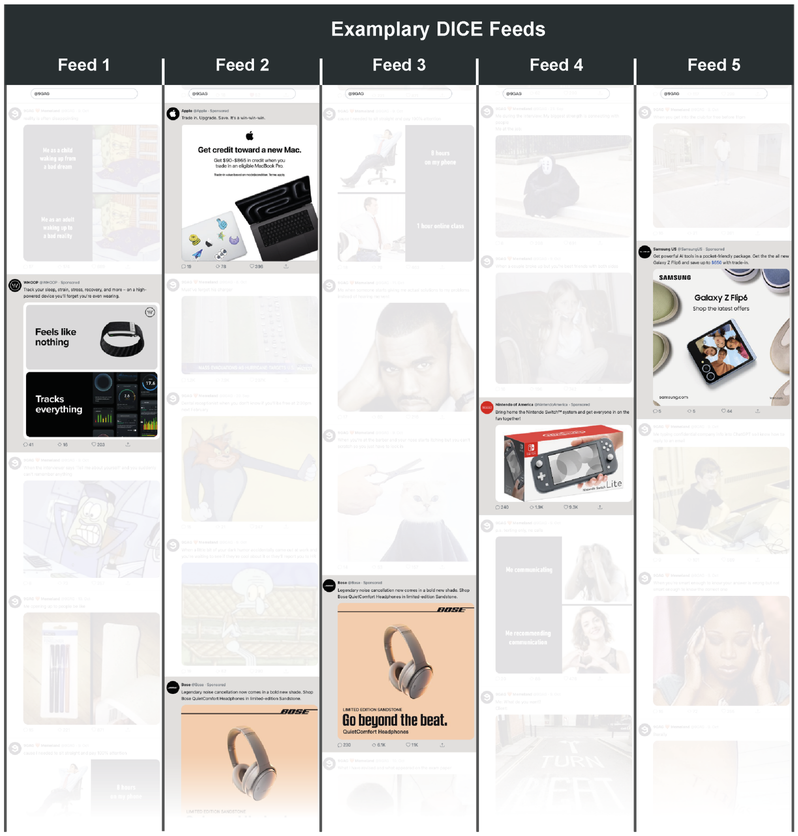


Figure 4.3: Exemplary DICE Feeds from Case Study 2

We recruited 300 younger American participants from Prolific (M_{age}=29.42 years; 53% female). Participants browsed the simulated feed on their own devices (72% desktop, 25% mobile, and 3% tablet). After scrolling through the feed, we redirected participants to a Qualtrics survey in which participants first provided demographic information as a filler task. Next, we measured whether participants recalled seeing the ads by the five brands in the feed. Specifically, we measured both free and cued recall. To measure cued recall, we showed participants a list of twenty brands from different categories and asked them to indicate whether they recalled seeing them (Campbell and Keller 2003; Simonov, Valletti, and Veiga 2025), including a no-recall option. The results for both recall measures were highly consistent; we report the cued recall results in the manuscript for parsimony (the results for unaided recall are reported in Section 4.2.7.1).

4.2.3 Stimuli

We next illustrate how we configured the feed to match our experimental design. Specifically, we created a CSV file that contains forty rows where each row represents one unique post. To guarantee that the order in which the posts were displayed was randomized between participants, we left the `<sequence>` column empty. Whenever the DICE app encounters missing values in that column, it assigns random numbers to that cell. Hence, leaving some, or in our case, *all* cells of this column empty, leads to random numbers and thus, random sequences in which the posts are displayed. Next, we specified the Boolean `<sponsored>` column and assigned a 0 to all thirty-five organic posts and a 1 to the five sponsored posts. For these sponsored posts, we also specified a `<target>` which is the URL of landing page participants are directed to if they click on the corresponding sponsored post. As a final step, we uploaded the CSV file to an online repository to create a URL that can be passed to DICE's web app.

Table 4.1 shows an excerpt of the exact CSV files we used to create the stimuli for this study. You can download that file here.

```r
text_columns <- names(stimuli_2)[sapply(stimuli_2, is.character)]

# Apply the line break replacement, and also remove @ symbols and <br> tags
stimuli_2[, (text_columns) := lapply(.SD, function(x) {
  x <- str_replace_all(x, "\n(?=[^.,!?;:])", " ")
  x <- str_replace_all(x, "\n(?=[.,!?;:])", "")
  x <- str_replace_all(x, "@", "")
  str_replace_all(x, "<br>", " ")
}), .SDcols = text_columns]

kable(tail(x = stimuli_2, n = 7))
```

Table 4.5: CSV File Used in Study 2

doc_id	date_time	text	media	alt_text	likes	reposts	replies	user_name	user_handle	user_description	user_image	user_followers	comments	sponsored	target_post	condition	sequence	source
34	20.09.24 12:01	Dental receptionist when you don't know if you'll be free at 2:30pm next February	https://pbs.twimg.com/media/GX3wITOX0AAEyhO?format=jpg&name=4096x4096	NA	24721	1569	0	9GAG: Memeland	**9GAG**	Building MEME-LAND one meme at a time • 9gag9gag.com • CEO 9gagceo.	https://pbs.twimg.com/profile_images/1695636573294125056/Jj9IWWu_400x400.jpg	16861230	12	NA		A	NA	https://twitter.com/9GAG/status/1836882519419989
35	19.09.24 06:01	Me: *sends "Good morning" text* Also me: *sleeps again*	https://pbs.twimg.com/media/GXz48AYX-MAA4-Wm?format=jpg&name=medium	NA	4751	189	0	9GAG: Memeland	**9GAG**	Building MEME-LAND one meme at a time • 9gag9gag.com • CEO 9gagceo.	https://pbs.twimg.com/profile_images/1695636573294125056/Jj9IWWu_400x400.jpg	16861230	12	NA		A	NA	https://twitter.com/9GAG/status/1836610460271030 57

Table 4.5: CSV File Used in Study 2

doc_id	date_time	text	media	alt_texts	likes	reposts	replies	user_name	user_handle	user_description	user_image	user_followers	comments	sponsored	target_post	condition	sequence	source
36	09.10.24 01:00	Trade in. Upgrade. Save. It's a win-win-win.	https://scontent-ham3-1.xx.fbcdn.net/v/t39.35426-6/462218991_2262824920752352_71707[illegible]jpg_s600x600&_nc_cat=105&ccb=1-7&_nc_sid=c53f8f&_nc_ohc=8iy6cVB[illegible]ham3-1.xx&_nc_gid=Au5_Py3fxlUQdue1av[illegible]hJD-AJ_RXQKg_afDs2aZ1kBcV–g&oe=67[illegible]	NA	396	78	19	Apple	Apple	Apple.com	https://substack-cdn.com/image/fetch/f_auto,q_auto:good,fl_progressive:steep/https%3A%2F%2Fsubstack-post-media.s3.amazonaws.com%2Fpublic%2Fimages%2F8ed3d547-94ff-48e1-9f20-8c14a7030a02_2000x2000.jpeg	9710	2532	1	https://www.apple.com/shop/buy-mac/macbook-[illegible]inch?cid=wwa-[illegible]mac-[illegible]socl-rbwfm-staf-infe-cpc-mos-mobtab-[illegible]gnt-broa-bnlp-lfnnl-usen-sngl-static-11ar-ban-macpro-trdn-rnge-buy-na-na-na-04807071	A	NA	https://www.facebook.com/ads/library/?active_status=active&ad_type=all&country=DE&id=5136347614[illegible]media_type=all&search_ty[illegible]

Table 4.5: CSV File Used in Study 2

doc_id	datetime	text	media	alt_text	likes	reposts	replies	user_handle	user_description	user_image	user_followers	comments	sponsored	target	condition	sequence	source
37	10.10.24 01:00	Track your sleep, strain, stress, recovery, and more – on a high-powered device you'll forget you're even wearing.	https://scontent-ham3-1.xx.fbcdn.net/v/t39.35426-6/456453687_1497228568338575_3460451067821947560_n.jpg?stp=dst-jpg_s600x600&_nc_cat=108&ccb=1-7&_nc_sid=c53f8f&_nc_ohc=sfzq2rnJIbYQ7kNvgEA5pdEc&_nc_ht=scontent-ham3-1.xx&_nc_gid=Ar8I66iuX4byWiBPP-Wa2iZK&oh=00_AYAcfU0N2dPoWf9iX2J78Z0qw-uFC0KjQS7_Si_G21NRDmg&oe=670EA829	NA	203	16	41	WHOOP	Your wearable health & fitness coach on a mission to unlock human performance. Join now and shop our latest styles	https://cdn-1.webcatalog.io/catalog/whoop/whoop-icon-filled-256.png?v=1727793667542	9840	45	1	https://www.whoop.com/	A	NA	https://www.facebook.com/ads/library/?active_status=active&ad_type=all&country=DE&id=884008727[illegible]dia_type=all&search_ty[illegible]

Table 4.5: CSV File Used in Study 2

doc_id	datetime	text	media	alt_text	likes	reposts	replies	user_handle	user_name	user_description	user_image	user_followers	comments	sponsored	target_post	condition	sequence	source
38	11.10.24 01:00	Get powerful AI tools in a pocket-friendly package. Get the the all new Galaxy Z Flip6 and save up to $650 with trade-in.	https://scontent-ham3-1.xx.fbcdn.net/v/t39.35426-6/461785311_849420040637530_[illegible].jpg_s600x600&_nc_cat=109&ccb=1-7&_nc_sid=c53f8f&_nc_ohc=_[illegible]XBev5O8Q7kNvgGd-3-L&_nc_ht=scontent-ham3-1.xx&_nc_gid=A6T___p18PiafwzhxKZvMoiE&oh=00_[illegible]	NA	44	5	5	SamsungUS	Samsung US	Official Twitter of Samsung USA.	https://i.pinimg.com/originals/[illegible].jpg	75937		1	https://www.samsung.com/us/smartphones/galaxy-z-flip6/buy/galaxy-z-flip6-512gb-unlocked-sm-f741uzsexaa/	A	NA	https://www.facebook.com/ads/library/?active_status=active&ad_type=all&country=DE&id=8007442920[illegible]&media_type=all&search_ty[illegible]

Table 4.5: CSV File Used in Study 2

doc_id	datetime	text	media	alt_text	likes	reposts	username	handle	user_description	user_image	user_followers	comments	sponsored	target_post	condition	sequence	source
39	12.10.24 01:00	Bring home the Nintendo Switch™ system and get everyone in on the fun together!	https://scontent-ham3-1.xx.fbcdn.net/v/t39.35426-6/461833323_1598276837567833_[illegible]_n.jpg?stp=dst-jpg_s600x600&_nc_cat=108&ccb=1-7&_nc_sid=c53f8f&_nc_ohc=YIdl5Mu9Ls54Q7kNvgGlM2&_nc_ht=scontent-ham3-1.xx&_nc_gid=AGzr0CP_zIG-20aQne5-Az6&oh=00_AYDry9MlNds6mSfmzOBiFatTpHy9qdqDPTMda84KJy3Gjw&oe=670E8CA6	NA	933	190	Nintendo	Nintendo of America	Welcome to the official Nintendo profile for gaming news! We're listening, too. For ESRB ratings go to http://esrb.org	https://pbs.twimg.com/profile_images/1539293929031421958/[illegible]_400x400.jpg	13991183	[illegible]	[illegible]	https://www.nintendo.com/us/store/products/nintendo-switch-lite-gray/	A	NA	https://www.facebook.com/ads/library/?active_status=active&ad_type=all&country=DE&id=105357370[illegible]&media_type=all&search_ty[illegible]

Table 4.5: CSV File Used in Study 2

doc_id	datetime	text	media	alt_text	likes	reposts	user_handle	user_name	user_description	user_image	user_followers	comments	sponsored	target_post	condition	sequence	source
40	13.10.24 01:00	Legendary noise cancellation now comes in a bold new shade. Shop Bose QuietComfort Headphones in limited-edition Sandstone.	https://scontent-ham3-1.xx.fbcdn.net/v/t39.35426-6/449312743_2744495332380149_8279471578607149378_n.jpg?stp=dst-jpg_s600x600&_nc_cat=102&ccb=1-7&_nc_sid=c53f8f&_nc_ohc=RZ2hDljuacQ7kNvgE-0db&_nc_ht=scontent-ham3-1.xx&_nc_gid=AVwJcHYfDMnOHYSQ-epUXZCO&oh=00_AYAFyDp3EHj3Gk_H1z_ApTeyOJuR	NA	11039	6392	Bose	Bose	Sound Is Power. For customer support, follow @BoseService.	https://logodownload.org/wp-content/uploads/2019/07/bose-logo-3.png	2339194	30	1	https://www.bose.com/p/noise-cancelling-headphones/quietcomfort-noise-cancelling-headphones/QC-HEADPHONEARN.html	A	NA	https://www.facebook.com/ads/library/?id=47104782579

4.2.4 Data and Randomization Checks

4.2.4.1 Final Sample

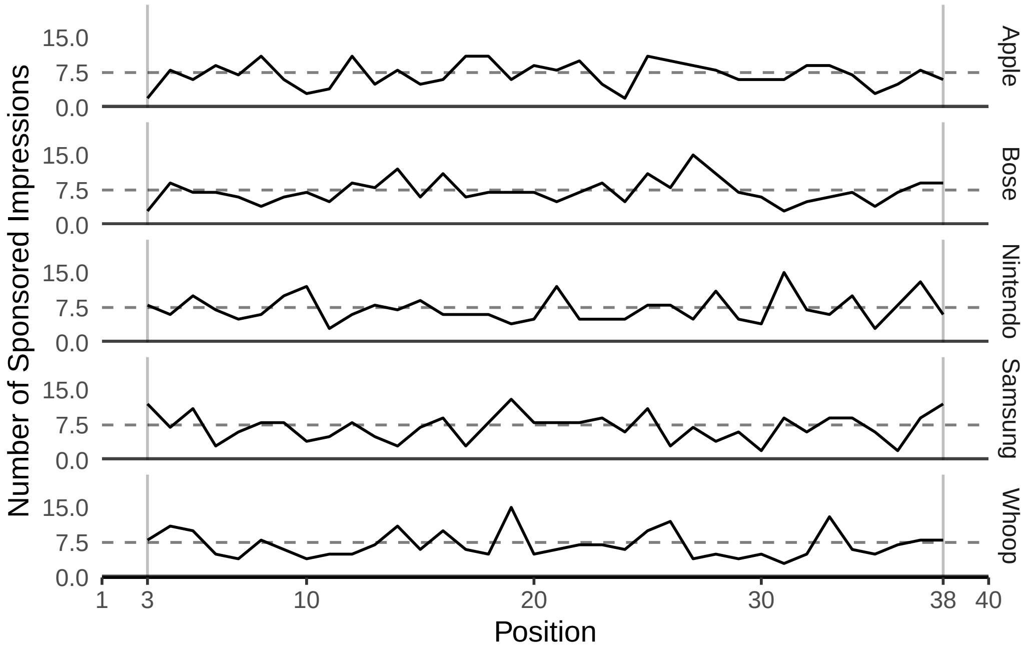
Our dataset comprises 300 participants and 10,377 observations at the participant × post level. In our analyses, we only focus on sponsored posts which is why our final sample comprises 1,283 observations on the participant × *sponsored* post level. This is less than the expected five observations per participant due to two reasons. First, due to connectivity issues: no dwell time data were recorded for around 3.67% of individual–sponsored post pairs. Second, we excluded the first and last two posts of each feed (i.e., 300×4 observations) from our analysis, as meaningful dwell times couldn't be determined for these. This is because participants were familiarizing themselves with the interface at the start and deciding whether to proceed to the next stage of the study at the end of the feeds.

4.2.4.2 Variables

Position in feed (subsequently referred to only as "position") is our main independent variable. Because we randomized the order in which the content appeared (i.e., position acts as a within-subject factor) and because we excluded observations positioned at the beginning and the end of the feed, we observe a sample mean of 20.58 as well as a minimum and maximum of 3 and 38, respectively. As, we randomly manipulated the position in which each post was displayed exogenously between subjects, each sequence in which participants browsed through ads was unique.

```
tmp <- subset[sponsored == TRUE,
            .(N = .N),
            by = c("displayed_sequence", "brand")][order(brand, displayed_sequence)]
ggplot(data = tmp,
       mapping = aes(x = displayed_sequence, y = N)) +
  facet_grid(rows = vars(brand)) +
  geom_line() +
  # geom_smooth(method = "loess", col = NA) +
  scale_y_continuous(limits = c(0, 20), expand = c(0, 0, 0.1, 0), breaks = c(0, 7.5, 15)) +
  scale_x_continuous(limits = c(1, 40), expand = c(0, 0, 0, 0), breaks = c(1, 3, 10, 20, 30,
  geom_hline(yintercept = 37.5/5, alpha = 0.5, lty = 2) +
  geom_hline(yintercept = 0, alpha = 0.75, linewidth = 1) +
  geom_vline(xintercept = 3, alpha = 0.25) +
  geom_vline(xintercept = 38, alpha = 0.25) +
  layout +
  labs(y = "Number of Sponsored Impressions", x = "Position") +
  theme(panel.grid.major.y = element_blank(),
        strip.background =element_rect(fill="#FFFFFF"),
        axis.line.x = element_blank())
```

Figure 4.4: Distribution of Sponsored Post Impressions by Position in Feed and Brand across All Participants



In Figure 4.4, each line represents the number of times a sponsored post for a specific brand appeared at each position across all participants. As the placements were fully randomized, we observed some random variability that naturally fluctuates around the expected value of 7.5 impressions per position (as indicated by dashed lines).[4] Taken together, this suggests that randomization within the DICE app was effective.

We measured dwell times as the number of seconds at least 50 percent of a post's pixels were visible on screen. We log-transformed the raw dwell times to reduce skewness. Unlike Case Study 2, where we focused on a single post (i.e., the KLM ad), the focal posts of interest (i.e., ads) in this Case Study vary in their post height. Thus, and as described in the "Behavioral Data" section of the DICE App Implementation, we normalized our dwell time measure by dividing it by post height to control for differently sized posts (i.e., the height in pixels of the corresponding sponsored post on a participant's screen).

We also tracked actual reactions to the content such as likes and replies to individual posts. In the full sample featuring both organic and sponsored posts, we observed 144 (30) participants who liked (replied to) any post in the feed. These numbers are obviously lower for sponsored posts (ads shown in the feed) with 32 (4) participants liked (replied to) at least one sponsored

[4]This expectation is based on each of the 5 sponsored posts being shown once per participant (N=300) in a feed of 40 positions, leading to an average of $\frac{300}{40} = 7.5$ for each brand's ad placement across all positions.

post or ad. Given the low incident rate, we did not analyze these likes and comments any further.

Table 4.6 shows an excerpt of the processed data to illustrate its nested (i.e., “long”) structure.

```
tmp <- data[,
            .(`Participant ID` = participant_label,
              `Post ID` = doc_id,
              `Sponsored Post` = sponsored,
              Brand = brand,
              `Position in Feed` = displayed_sequence,
              Likes = liked,
              Replies = hasReply,
              `Seconds in Viewport` = seconds_in_viewport,
              `log(Seconds in Viewport)` = log_dwell_time,
              `Post Height` = height,
              `Dwell Time` = log_dwell_pixel,
              Age = age,
              Female = female,
              Desktop = as.logical(is_desktop),
              `Recall Apple` = cued_apple,
              `Recall Bose` = cued_bose,
              `Recall Nintendo` = cued_nintendo,
              `Recall Samsung` = cued_samsung,
              `Recall Whoop` = cued_whoop)]

kable(tmp[35:45])
```

Table 4.6: Processed Data Analyzed in Study 1

Participant ID	Post ID	Sponsored Post	Brand	Position in Feed	Likes	Replies	Seconds in Viewport	log(Seconds in Viewport)	Post Height	Dwell Time	Age	Female	Desktop	Recall Apple	Recall Bose	Recall Nintendo	Recall Samsung	Recall Whoop
[illegible]	[illegible]	FALSE	NA	35	FALSE	FALSE	0.083	-2.4889147	383	-0.0064985	33	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
[illegible]	[illegible]	FALSE	NA	36	FALSE	FALSE	0.117	-2.1455813	420	-0.0051085	33	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
[illegible]	[illegible]	FALSE	NA	37	FALSE	FALSE	0.150	-1.8971200	416	-0.0045604	33	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE

Table 4.6: Processed Data Analyzed in Study 1

Participant ID	Post ID	Sponsored Post	Brand	Position in Feed	Likes	Replies	Seconds in Viewport	log(Seconds in Viewport)	Post Height	Dwell Time	Age	Female	Desktop	Recall Apple	Recall Bose	Recall Nintendo	Recall Samsung	Recall Whoop
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-1.7897615	402	-0.0044521	33	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-1.4524342	388	-0.0037434	33	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-0.9781661	507	-0.0019293	30	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
[illegible]	[illegible]	TRUE	Apple	[illegible]	[illegible]	[illegible]	[illegible]	-1.7147984	622	-0.0027569	30	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
[illegible]	[illegible]	TRUE	Whoop	[illegible]	[illegible]	[illegible]	[illegible]	-0.9623347	646	-0.0014897	30	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE

4.2.5 Empirical Model

To estimate the impact of ad positioning on brand recall in social media feeds, we employed a mixed-effects logistic regression model with brand fixed effects to account for the binary nature of recall outcome (recalled vs. not recalled) while considering the hierarchical structure of our data: multiple observations nested within participants and ads. We assume a binomial distribution as each observation represents a single trial with two possible outcomes (recalled vs. not recalled), where p_{ij} represents the probability of participant i recalling brand j:

$$\text{recall}_{ij} \sim \text{Binomial}(1, p_{ij})$$

We estimated the effect of ad positioning on recall and captured between-participant heterogeneity through random intercepts while controlling for brand fixed effects:

$$\text{logit}(p_{ij}) = a + a_i + \mathbf{x}_{ij}\mathbf{b} + \sum_{j=1}^{J-1} \gamma_j \text{Brand}_j$$

where a is the global intercept, a_i is the participant-specific random intercept, $\mathbf{x}_{ij}$ is a vector of continuous predictors (e.g., position and dwell time) with corresponding coefficient vector $\mathbf{b}$, and

γ_j represents the fixed effects for each brand j (with Apple serving as the reference category). The random participant effects a_i follows from our experimental design, given that the random assignment of ad positions ensures zero correlation between participant characteristics and the explanatory variables. Brand effects are treated as fixed parameters rather than random effects, allowing for potential correlation between brand characteristics and positioning.

The random participant effects a_i follows from our experimental design, given that the random assignment of ad positions ensures zero correlation between participant characteristics and the explanatory variables. Brand effects are treated as fixed parameters rather than random effects, allowing for potential correlation between brand characteristics and core explanatory variables such as dwell time.

Finally, for better comparability between models, we z-standardized all explanatory variables. The regression coefficients effects on the dependent variable are therefore quantified in standard deviations. This allowed us to compare the relative effect sizes between regression models. Because of the logit link, the odds ratio is $100 \times (e^{\beta} - 1)$, which gives the percentage change in the odds of recall.

4.2.6 Results and Discussion

```
tmp <- copy(subset[sponsored == 1])
tmp[, log_dwell_pixel := scale(log_dwell_pixel)]
tmp[, displayed_sequence := scale(displayed_sequence)]


glmer_1 <- glmer(recalled_brand_cued ~ displayed_sequence + I(displayed_sequence^2) +
                   (1|participant_label) + brand,
                 data = tmp,
                 family = binomial(link = "logit"))

glmer_2 <- glmer(recalled_brand_cued ~ log_dwell_pixel +
                   (1|participant_label) + brand,
                 data = tmp,
                 family = binomial(link = "logit"))

glmer_3 <- glmer(recalled_brand_cued ~ displayed_sequence + I(displayed_sequence^2) + log_dwe
                   (1|participant_label),
                 data = tmp,
                 family = binomial(link = "logit"),
                 control = glmerControl(optimizer = "optimx",
                                        optCtrl = list(method = "nlminb")))
```

```
model_1_output <- summary(glmer_1)
  model_1_coefficients <- coef(model_1_output)
  model_1_beta_1 <- apa(model_1_coefficients[2, 1], decimals = 3, leading = FALSE)
  model_1_se_1   <- apa(model_1_coefficients[2, 2], decimals = 3, leading = FALSE)
  model_1_z_1    <- apa(model_1_coefficients[2, 3], decimals = 3, leading = FALSE)
  model_1_p_1    <- apa(model_1_coefficients[2, 4], decimals = 3, leading = FALSE)

model_2_output <- summary(glmer_2)
  model_2_coefficients <- coef(model_2_output)
  model_2_beta_3 <- apa(model_2_coefficients[2, 1], decimals = 3, leading = FALSE)
  model_2_se_3   <- apa(model_2_coefficients[2, 2], decimals = 3, leading = FALSE)
  model_2_z_3    <- apa(model_2_coefficients[2, 3], decimals = 3, leading = FALSE)
  model_2_p_3    <- apa(model_2_coefficients[2, 4], decimals = 3, leading = FALSE)

model_3_output <- summary(glmer_3)
  model_3_coefficients <- coef(model_3_output)
  model_3_beta_1 <- apa(model_3_coefficients[2, 1], decimals = 3, leading = FALSE)
  model_3_se_1   <- apa(model_3_coefficients[2, 2], decimals = 3, leading = FALSE)
  model_3_z_1    <- apa(model_3_coefficients[2, 3], decimals = 3, leading = FALSE)
  model_3_p_1    <- apa(model_3_coefficients[2, 4], decimals = 3, leading = FALSE)
  model_3_beta_3 <- apa(model_3_coefficients[4, 1], decimals = 3, leading = FALSE)
  model_3_se_3   <- apa(model_3_coefficients[4, 2], decimals = 3, leading = FALSE)
  model_3_z_3    <- apa(model_3_coefficients[4, 3], decimals = 3, leading = FALSE)
  model_3_p_3    <- apa(model_3_coefficients[4, 4], decimals = 3, leading = FALSE)
```

The effect of *position* on recall was significant and negative (β_1 = -.207, SE = .070, z = -2.955, p = .003; see Model 1 in Table 4.7), suggesting a primacy effect such that the further up (down) an ad is displayed in a feed, the more (less) participants recall seeing the ad. We also examined potential non-linear effects of position (i.e., to assess whether especially the beginning and end of a feed promote ad recall) by adding a quadratic term, but found no statistically significant effect. These results are robust to alternative model specifications, using *uncued* recall (instead of cued recall) as our dependent variable (see Section 4.2.7.1).

```
tmp <- copy(subset[sponsored == 1])
tmp[, log_dwell_pixel := scale(log_dwell_pixel)]
tmp[, displayed_sequence := scale(displayed_sequence)]


med_model <- lmer(log_dwell_pixel ~ displayed_sequence + brand + (1|participant_label),
                  data = tmp)

out_model <- glmer(recalled_brand_cued ~ displayed_sequence + brand + log_dwell_pixel +
```

```
                    (1|participant_label),
                    data = tmp,
                    family = binomial(link = "logit"),
                    control = glmerControl(optimizer = "optimx",
                                           optCtrl = list(method = "nlminb")))


# check convergence
# summary(med_model)
# summary(out_model)

# mediation analysis
med_results <- mediate(med_model, out_model,
                       treat = "displayed_sequence",
                       mediator = "log_dwell_pixel",
                       boot = FALSE, sims = 1000)

# plot(med_results)
```

```
model_summary <- summary(med_results)

acme_estimate <- apa(model_summary$d.avg, decimals = 3, leading = FALSE)
acme_ci_lower <- apa(model_summary$d.avg.ci[1], decimals = 3, leading = FALSE)
acme_ci_upper <- apa(model_summary$d.avg.ci[2], decimals = 3, leading = FALSE)
prop_med <- apa(model_summary$n.avg, decimals = 3, leading = FALSE)
prop_med_ci_lower <- apa(model_summary$n.avg.ci[1], decimals = 3, leading = FALSE)
prop_med_ci_upper <- apa(model_summary$n.avg.ci[2], decimals = 3, leading = FALSE)
total_effect_p <- apa(model_summary$tau.p, decimals = 3, leading = FALSE)
```

As shown in Model 2 in Table 4.7, the post-level dwell time of a user significantly predicts ad recall (β_3 = .714, SE = .091, z = 7.832, p = .000). More importantly, we also found that the dwell time allocated to an ad was a stronger predictor of recall than the position of the ad in the feed. Specifically, Model 3, including both position and dwell time, shows that the effect of position becomes non-significant (β_1 = -.104, SE = .072, z = -1.434, p = .152), while the dwell time coefficient remains unchanged and significant (β_3 = .694, SE = .092, z = 7.550, p = .000). The minimal change in the Akaike information criterion when adding the position coefficients between Model 2 and Model 3 suggests that including ad position does not improve the model's fit to the data. A subsequent 1-1-1 hierarchical mediation analysis (Zhang, Zyphur, and Preacher 2008) indeed supports that post-level dwell time significantly mediates the relationship between ad position and recall ($a \times b = -.019$, 95% CI = [-.027, -.013], proportion mediated = .540, 95% CI = [.294, 1.439], p = .002). Section 4.2.7.2 reports additional analyses that suggest limited heterogeneity in these patterns across different sponsored posts. While brands may differ in their baseline dwell times due to potential familiarity differences, dwell time consistently predicts recall across brands.

```
n_obs <- format(nobs(glmer_1), big.mark = ",")

tbl_1 <- tbl_regression(
  glmer_1,
  intercept = TRUE,  # This ensures the intercept is included
  include = c(displayed_sequence, `I(displayed_sequence^2)`),
  estimate_fun = ~ style_number(.x, digits = 3),
  pvalue_fun = label_style_pvalue(digits = 3, zero.print = "."),
  conf.int = FALSE,
  label = list(
    displayed_sequence ~ "Position",
    `I(displayed_sequence^2)` ~ "Position²"
  ),
  add_estimate_to_reference_rows = FALSE
) |>
  modify_column_unhide(columns = std.error) |>
  add_glance_table(include = AIC)

tbl_2 <- tbl_regression(
  glmer_2,
  intercept = TRUE,
  include = c(log_dwell_pixel),
  estimate_fun = ~ style_number(.x, digits = 3),
  pvalue_fun = label_style_pvalue(digits = 3, zero.print = "."),
  conf.int = FALSE,
  label = list(
    log_dwell_pixel ~ "Dwell Time"
  ),
  add_estimate_to_reference_rows = FALSE
) |>
  modify_column_unhide(columns = std.error) |>
  add_glance_table(include = AIC)

tbl_3 <- tbl_regression(
  glmer_3,
  intercept = TRUE,
  include = c(displayed_sequence, `I(displayed_sequence^2)`, log_dwell_pixel),
  estimate_fun = ~ style_number(.x, digits = 3),
  pvalue_fun = label_style_pvalue(digits = 3, zero.print = "."),
  conf.int = FALSE,
```

```r
  label = list(
    displayed_sequence ~ "Position",
    `I(displayed_sequence^2)` ~ "Position²",
    log_dwell_pixel ~ "Dwell Time"
  ),
  add_estimate_to_reference_rows = FALSE
) |>
  modify_column_unhide(columns = std.error) |>
  add_glance_table(include = AIC)

table <- tbl_merge(
  tbls = list(tbl_1, tbl_2, tbl_3),
  tab_spanner = c("Model 1", "Model 2", "Model 3")
) |>
  modify_table_body(
    ~ .x |>
      dplyr::arrange(
        factor(label, levels = c("(Intercept)", "Position", "Position²", "Dwell Time", "AIC")
      )
  ) |>
  modify_header(
    label ~ "",
    estimate_1 ~ "beta",
    estimate_2 ~ "beta",
    estimate_3 ~ "beta",
    std.error_1 ~ "SE",
    std.error_2 ~ "SE",
    std.error_3 ~ "SE",
    p.value_1 ~ "p",
    p.value_2 ~ "p",
    p.value_3 ~ "p"
  )

table
```

Table 4.7: Estimates of Recall as a Function of Ad Position and Dwell Time

	Model 1			Model 2			Model 3		
	beta[1]	SE[1]	p	beta[1]	SE[1]	p	beta[1]	SE[1]	p
(Intercept)	-0.821	0.170	<0.001	-1.118	0.172	<0.001	-1.082	0.186	<0.001
Position	-0.207	0.070	0.003				-0.104	0.072	0.152

Position^2	0.028	0.077	0.719				-0.029	0.079	0.710
Dwell Time				0.714	0.091	<0.001	0.694	0.092	<0.001
AIC	1,476			1,405			1,407		

[1]OR = Odds Ratio, SE = Standard Error

From a substantive perspective, the findings demonstrate how memory formation operates when brands compete for limited attention in a social media environment, consistent with prior work on attention and memory under competitive exposure (Pieters, Warlop, and Wedel 2002). From a more methodological perspective, Case Study 2 demonstrates how to combine the behavioral data (i.e., post-level dwell time as a proxy of attention) with self-reports from conventional survey measures (e.g., recall of branded content).

4.2.7 Robustness Checks

4.2.7.1 Uncued Recall

Here, we reproduce Table 4.7 using uncued (or "free") instead of cued recall as our dependent variable, using the same empirical strategy as before.

Table 4.8 reproduces these findings and shows that they are robust to this alternative model specification: we still observe a primacy effect in out naïve Model 1 that vanishes as we control for dwell time in Model 3.

```
tmp <- copy(subset[sponsored == 1])
tmp[, log_dwell_pixel := scale(log_dwell_pixel)]
tmp[, displayed_sequence := scale(displayed_sequence)]


glmer_1 <- glmer(recalled_brand_uncued ~ displayed_sequence + I(displayed_sequence^2) +
                   (1|participant_label) + brand,
                 data = tmp,
                 family = binomial(link = "logit"))

glmer_2 <- glmer(recalled_brand_uncued ~ log_dwell_pixel +
                   (1|participant_label) + brand,
                 data = tmp,
                 family = binomial(link = "logit"))

glmer_3 <- glmer(recalled_brand_uncued ~ displayed_sequence + I(displayed_sequence^2) + log_
                   (1|participant_label),
                 data = tmp,
```

```r
                  family = binomial(link = "logit"),
                  control = glmerControl(optimizer = "optimx",
                                         optCtrl = list(method = "nlminb")))
```

```r
model_1_output <- summary(glmer_1)
  model_1_coefficients <- coef(model_1_output)
  model_1_beta_1 <- apa(model_1_coefficients[2, 1], decimals = 3, leading = FALSE)
  model_1_se_1   <- apa(model_1_coefficients[2, 2], decimals = 3, leading = FALSE)
  model_1_z_1    <- apa(model_1_coefficients[2, 3], decimals = 3, leading = FALSE)
  model_1_p_1    <- apa(model_1_coefficients[2, 4], decimals = 3, leading = FALSE)

model_2_output <- summary(glmer_2)
  model_2_coefficients <- coef(model_2_output)
  model_2_beta_3 <- apa(model_2_coefficients[2, 1], decimals = 3, leading = FALSE)
  model_2_se_3   <- apa(model_2_coefficients[2, 2], decimals = 3, leading = FALSE)
  model_2_z_3    <- apa(model_2_coefficients[2, 3], decimals = 3, leading = FALSE)
  model_2_p_3    <- apa(model_2_coefficients[2, 4], decimals = 3, leading = FALSE)

model_3_output <- summary(glmer_3)
  model_3_coefficients <- coef(model_3_output)
  model_3_beta_1 <- apa(model_3_coefficients[2, 1], decimals = 3, leading = FALSE)
  model_3_se_1   <- apa(model_3_coefficients[2, 2], decimals = 3, leading = FALSE)
  model_3_z_1    <- apa(model_3_coefficients[2, 3], decimals = 3, leading = FALSE)
  model_3_p_1    <- apa(model_3_coefficients[2, 4], decimals = 3, leading = FALSE)
  model_3_beta_3 <- apa(model_3_coefficients[4, 1], decimals = 3, leading = FALSE)
  model_3_se_3   <- apa(model_3_coefficients[4, 2], decimals = 3, leading = FALSE)
  model_3_z_3    <- apa(model_3_coefficients[4, 3], decimals = 3, leading = FALSE)
  model_3_p_3    <- apa(model_3_coefficients[4, 4], decimals = 3, leading = FALSE)
```

The effect of position on *uncued* recall was significant and negative (β_1 = -.180, SE = .075, z = -2.409, p = .016; see Model 1 in Table 4.8), suggesting a primacy effect such that the further up (down) an ad is displayed in a feed, the more (less) participants recall seeing the ad. We also examined potential non-linear effects of position (i.e., to assess whether especially the beginning and end of a feed promote uncued ad recall) by adding a quadratic term, but found no statistically significant effect.

```r
tmp <- copy(subset[sponsored == 1])
tmp[, log_dwell_pixel := scale(log_dwell_pixel)]
tmp[, displayed_sequence := scale(displayed_sequence)]
```

```r
med_model <- lmer(log_dwell_pixel ~ displayed_sequence + brand + (1|participant_label),
                  data = tmp)

out_model <- glmer(recalled_brand_uncued ~ displayed_sequence + brand + log_dwell_pixel +
                   (1|participant_label),
                   data = tmp,
                   family = binomial(link = "logit"),
                   control = glmerControl(optimizer = "optimx",
                                          optCtrl = list(method = "nlminb")))


# check convergence
# summary(med_model)
# summary(out_model)

# mediation analysis
med_results <- mediate(med_model, out_model,
                       treat = "displayed_sequence",
                       mediator = "log_dwell_pixel",
                       boot = FALSE, sims = 1000)

# plot(med_results)
```

```r
model_summary <- summary(med_results)

acme_estimate <- apa(model_summary$d.avg, decimals = 3, leading = FALSE)
acme_ci_lower <- apa(model_summary$d.avg.ci[1], decimals = 3, leading = FALSE)
acme_ci_upper <- apa(model_summary$d.avg.ci[2], decimals = 3, leading = FALSE)
prop_med <- apa(model_summary$n.avg, decimals = 3, leading = FALSE)
prop_med_ci_lower <- apa(model_summary$n.avg.ci[1], decimals = 3, leading = FALSE)
prop_med_ci_upper <- apa(model_summary$n.avg.ci[2], decimals = 3, leading = FALSE)
total_effect_p <- apa(model_summary$tau.p, decimals = 3, leading = FALSE)
```

As shown in Model 2 in Table 4.8, the post-level dwell time of a user significantly predicts uncued recall (β_3 = .596, SE = .090, z = 6.589, p = .000). More importantly, we also found that the dwell time allocated to an ad was a stronger predictor of uncued recall than the position of the ad in the feed. Specifically, Model 3, including both position and dwell time, shows that the effect of position becomes non-significant (β_1 = -.095, SE = .078, z = -1.221, p = .222), while the dwell time coefficient remains unchanged and significant (β_3 = .579, SE = .091, z = 6.348, p = .000). The minimal change in the Akaike information criterion when adding the position coefficients between Model 2 and Model 3 suggests that including ad position does not improve the model's fit to the data. A subsequent 1-1-1 hierarchical mediation analysis

(Zhang, Zyphur, and Preacher 2008) indeed supports that post-level dwell time significantly mediates the relationship between ad position and uncued recall ($a \times b = -.014$, 95% CI = [-.019, -.009], proportion mediated = .520, 95% CI = [.242, 2.366], p = .018).

```
n_obs <- format(nobs(glmer_1), big.mark = ",")

tbl_1 <- tbl_regression(
  glmer_1,
  intercept = TRUE,  # This ensures the intercept is included
  include = c(displayed_sequence, `I(displayed_sequence^2)`),
  estimate_fun = ~ style_number(.x, digits = 3),
  pvalue_fun = label_style_pvalue(digits = 3, zero.print = "."),
  conf.int = FALSE,
  label = list(
    displayed_sequence ~ "Position",
    `I(displayed_sequence^2)` ~ "Position²"
  ),
  add_estimate_to_reference_rows = FALSE
) |>
  modify_column_unhide(columns = std.error) |>
  add_glance_table(include = AIC)

tbl_2 <- tbl_regression(
  glmer_2,
  intercept = TRUE,
  include = c(log_dwell_pixel),
  estimate_fun = ~ style_number(.x, digits = 3),
  pvalue_fun = label_style_pvalue(digits = 3, zero.print = "."),
  conf.int = FALSE,
  label = list(
    log_dwell_pixel ~ "Dwell Time"
  ),
  add_estimate_to_reference_rows = FALSE
) |>
  modify_column_unhide(columns = std.error) |>
  add_glance_table(include = AIC)

tbl_3 <- tbl_regression(
  glmer_3,
  intercept = TRUE,
  include = c(displayed_sequence, `I(displayed_sequence^2)`, log_dwell_pixel),
  estimate_fun = ~ style_number(.x, digits = 3),
  pvalue_fun = label_style_pvalue(digits = 3, zero.print = "."),
```

```r
  conf.int = FALSE,
  label = list(
    displayed_sequence ~ "Position",
    `I(displayed_sequence^2)` ~ "Position²",
    log_dwell_pixel ~ "Dwell Time"
  ),
  add_estimate_to_reference_rows = FALSE
) |>
  modify_column_unhide(columns = std.error) |>
  add_glance_table(include = AIC)

table <- tbl_merge(
  tbls = list(tbl_1, tbl_2, tbl_3),
  tab_spanner = c("Model 1", "Model 2", "Model 3")
) |>
  modify_table_body(
    ~ .x |>
      dplyr::arrange(
        factor(label, levels = c("(Intercept)", "Position", "Position²", "Dwell Time", "AIC")
      )
  ) |>
  modify_header(
    label ~ "",
    estimate_1 ~ "beta",
    estimate_2 ~ "beta",
    estimate_3 ~ "beta",
    std.error_1 ~ "SE",
    std.error_2 ~ "SE",
    std.error_3 ~ "SE",
    p.value_1 ~ "p",
    p.value_2 ~ "p",
    p.value_3 ~ "p"
  )

table
```

Table 4.8: Estimates of Uncued Recall as a Function of Ad Position and Dwell Time

	Model 1			Model 2			Model 3		
	beta[1]	SE[1]	p	beta[1]	SE[1]	p	beta[1]	SE[1]	p
(Intercept)	-1.188	0.179	<0.001	-1.473	0.183	<0.001	-1.454	0.199	<0.001

Position	-0.180	0.075	0.016				-0.095	0.078	0.222
Position^2	0.033	0.082	0.690				-0.013	0.085	0.873
Dwell Time				0.596	0.090	<0.001	0.579	0.091	<0.001
AIC	1,274			1,225			1,227		

[1]OR = Odds Ratio, SE = Standard Error

4.2.7.2 Ad Position, Dwell Time, and Recall Across Brands

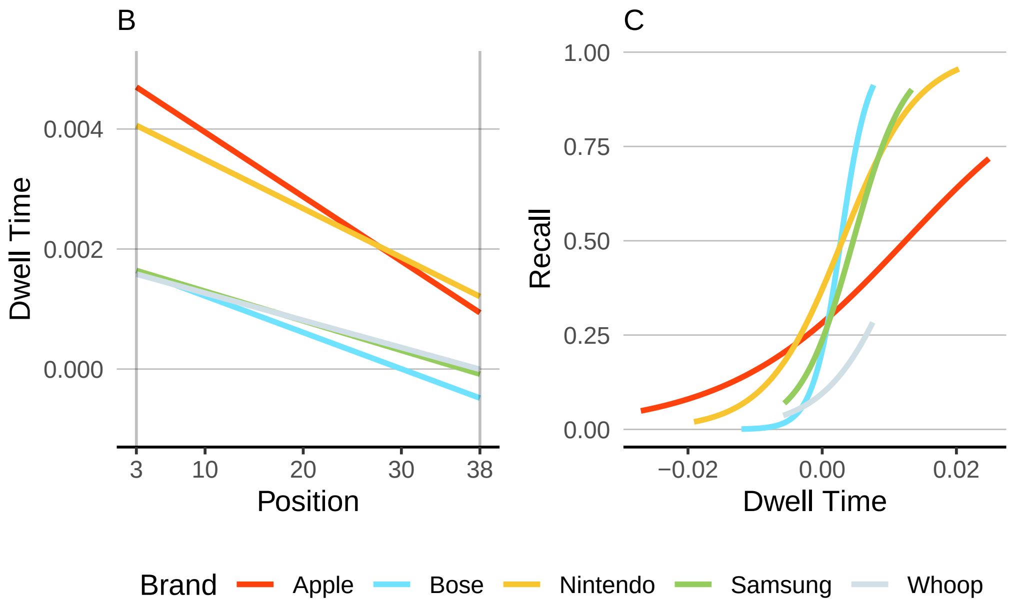
Ad recall significantly varied across brands. Figure 4.5 (Panel A) visualizes the relationship between dwell time and position across the five brands in our study. We observe a consistent negative relationship between position and dwell time across all brands, with posts placed later in the feed receiving less attention.

```
p4 <- ggplot(data = subset[sponsored == 1],
       mapping = aes(x = displayed_sequence, y = log_dwell_pixel,
                     col = brand)) +
  geom_smooth(method = "lm", formula = "y ~ x", se = FALSE) +
  coord_cartesian(ylim = c(-0.001, 0.005)) +
  scale_x_continuous(limits = c(1, 40), expand = c(0, NA), breaks = c(3, 10, 20, 30, 38)) +
  geom_vline(xintercept = 3, alpha = 0.25) +
  geom_vline(xintercept = 38, alpha = 0.25) +
  scale_colour_tron() +
  layout +
  labs(title = "B", y = "Dwell Time", x = "Position", col = "Brand")
```

```
p6 <- ggplot(data = subset[sponsored == 1],
       mapping = aes(x = log_dwell_pixel, y = as.numeric(recalled_brand_cued),
                     col = brand)) +
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"),
              formula = "y ~ x", se = FALSE) +
  scale_colour_tron() +
  scale_x_continuous(breaks = c(-0.02, 0, 0.02)) +
  layout +
  labs(title = "C", y = "Recall", x = "Dwell Time", col = "Brand")
```

```
(p4 + p6 + plot_layout(guides = "collect")) & theme(legend.position = "bottom")
```

Figure 4.5: Ad Position, Dwell Time, and Recall Across Brands



However, we also find that Apple and Nintendo generated significantly higher dwell time than all other brands (see the parallel upward shift of the regression line compared to Bose, Samsung, and Whoop). Panel B further shows how dwell time predicts recall probability consistently across all brands. The positive slope of all curves in Panel B indicates that increased dwell time enhances recall probability across all brands consistently, even for less familiar brands like Whoop. This suggests that while brands may differ in their baseline dwell time due to potential familiarity differences, we find a highly consistent empirical regularity of how dwell time ultimately predicts recall.

5 Deploy DICE yourself

The deployment process, while comprehensive, is designed to be accessible to researchers of all technical backgrounds. The step-by-step instructions that follow are structured to guide users through each stage of the deployment process, regardless of their familiarity with terminal or command-line interfaces. The commands are straightforward, and detailed guidance is provided throughout. It's worth noting that oTree has been successfully implemented by numerous researchers across various disciplines, many of whom began with limited technical expertise. Importantly, oTree has a large and active user community, which brings several advantages:

1. **Extensive Documentation**: oTree is well-documented, with comprehensive guides and tutorials available online.
2. **Community Support**: There's an active forum where you can ask questions and get help from experienced users.
3. **AI Assistance**: Because of its popularity and well-documented nature, generative AI tools can provide assistance with oTree-related queries.

The combination of detailed documentation, community resources, and technological support ensures that researchers can efficiently implement their experimental protocols. These resources are designed to facilitate a smooth deployment process, enabling researchers to focus on their experimental objectives rather than technical challenges.

5.1 Prerequisites

Before you begin, ensure you have the following installed:

1. **Python**: oTree requires Python 3.7 or higher. Download and install Python from python.org.

2. **pip**: This is Python's package installer. It usually comes with Python installation.

3. **Command Prompt or Terminal**: This is a text-based interface to interact with your computer. You'll use it to run commands for installing and running oTree.

    - On Windows: Search for "Command Prompt" or "PowerShell" in the Start menu.
    - On macOS: Open the "Terminal" application (found in Applications > Utilities).
    - On Linux: You can usually open a terminal with Ctrl+Alt+T or by searching for "Terminal" in your application menu.

5.2 Install oTree

Open a command prompt or terminal and run:

```
pip install otree
```

5.3 Extract the Provided oTree Project

Create and open a new directory (e.g. `my_DICE_experiment`) for your experiment (either manually or using the command prompt or terminal):

```
mkdir my_DICE_experiment
cd my_DICE_experiment
```

Now download the `DICE.otreezip` file (which you can find here) and store it in your new directory. Then use the command prompt or terminal and run the `otree unzip` command to extract the contents of the otreezip file:

```
otree unzip DICE.otreezip
```

If necessary, adjust the path to the otreezip file.

This will unpack the experiment such that you can browse and edit its raw files. One of them is called `requirements.txt`. You can find it in `my_DICE_experiment/` (or whatever name you chose for your project directory). You should have all the requirements installed but it may not hurt to re-install them using:

```
pip install -r requirements.txt
```

5.4 Running the experiment locally

Run the following command to set up the database:

```
otree resetdb
```

Start the oTree development server:

```
otree devserver
```

Open a web browser and go to:

`http://localhost:8000`

You should see the oTree admin interface.

5.5 Troubleshooting

If you encounter any issues while setting up or running the experiment, consider the following:

- If you see any "module not found" errors, make sure all required packages are installed.
- Check that you're in the correct directory when running commands.
- Ensure your Python version is compatible with the oTree version used in the provided experiment.

> **Note**
>
> If you encounter any specific errors or issues with the provided experiment, consult oTree's official documentation, oTree's official forum, generative AI or Hauke Roggenkamp.

6 Server setup

> **Note**
>
> The following text copies and summarizes oTree's official documentation.

If you are just testing your app on your personal computer, you can use otree devserver. You don't need a full server setup.

However, when you want to share your app with an audience, you must use a web server.

Choose which option you need:

- You want to launch your app to users on the internet: Use *Heroku*.
- You want the easiest setup: Again, we recommend *Heroku*.

6.1 Basic server setup with *Heroku*

Heroku is a commercial cloud hosting provider. It is the simplest way to deploy oTree.

The Heroku free plan is sufficient for testing your app, but once you are ready to launch a study, you should upgrade to a paid server, which can handle more traffic. However, Heroku is quite inexpensive, because you only pay for the time you actually use it. If you run a study for only 1 day, you can turn off your dynos and addons, and then you only pay 1/30 of the monthly cost. Often this means you can run a study for just a few dollars.

To deploy to Heroku, you should use oTree Hub, which automates your server setup and ensures your server is correctly configured.

oTree Hub also offers error/performance monitoring.

6.1.1 Server performance

Heroku offers different performance tiers for resources such as your dyno and database. What tier you need depends on how much traffic your app will get, and how it is coded.

Performance is a complicated subject since there are many factors that affect performance. oTree Hub's Pro plan has a "monitor" section that will analyze your logs to identify performance issues.

6.1.2 General tips

- Upgrade oTree to the latest version
- With the higher dyno tiers, Heroku provides a "Metrics" tab. Look at "Dyno load". If users are experiencing slow page load times and your dyno load stays above 1, then you should get a faster dyno. (But don't run more than 1 web dyno.)
- If your dyno load stays under 1 but page load times are still slow, the bottleneck might be something else like your Postgres database.

7 Extensions

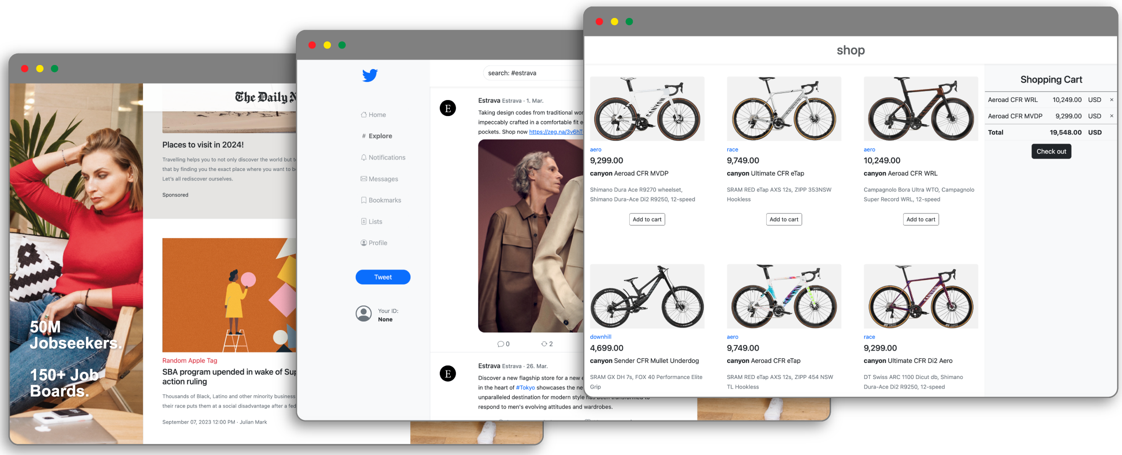


Figure 7.1: DICE extensions

Beyond social media (e.g., Twitter or LinkedIn-style) feeds, the app could be adapted to study various social media formats or, more broadly, other digital platforms. We have developed preliminary versions of news and shopping feeds using the DICE app's framework, demonstrating its flexibility and customizability.

You can download the corresponding `*.otreezip` files below. The Deployment chapter provides guidance for hosting these apps either locally or on your own server.

Download News Feed Download Shopping Feed

We hope to feature more DICE extensions (such as a review platform, for instance) developed by the research community in the future.

References

Ahmad, Wajeeha, Ananya Sen, Charles Eesley, and Erik Brynjolfsson. 2024. “Companies Inadvertently Fund Online Misinformation Despite Consumer Backlash.” *Nature* 630: 123–31. https://doi.org/10.1038/s41586-024-07404-1.

Campbell, Margaret C., and Kevin Lane Keller. 2003. “Brand Familiarity and Advertising Repetition Effects.” *Journal of Consumer Research* 30 (2): 292–304. https://doi.org/10.1086/376800.

Chen, Daniel L., Martin Schonger, and Chris Wickens. 2016. “oTree—an Open-Source Platform for Laboratory, Online, and Field Experiments.” *Journal of Behavioral and Experimental Finance* 9: 88–97. https://doi.org/doi.org/10.1016/j.jbef.2015.12.001.

Fournier, Susan, and Shuba Srinivasan. 2023. “Monitoring Marketing Sources of Brand Reputation Risk.” *NIM Marketing Intelligence Review* 15 (1): 46–53. https://doi.org/doi:10.2478/nimmir-2023-0007.

GumGum Inc. 2017. “The New Brand Safety Crisis: A Fractured Environment.” GumGum. https://web.archive.org/web/20220317063148/https://insights.gumgum.com/hubfs/Brand_Safety_GumGum.pdf.

Kerwin, Jason, Nada Rostom, and Olivier Sterck. 2024. “Striking the Right Balance: Why Standard Balance Tests over-Reject the Null, and How to Fix It.” IZA Discussion Paper 17217. Institute of Labor Economics (IZA). https://www.iza.org/publications/dp/17217/striking-the-right-balance-why-standard-balance-tests-over-reject-the-null-and-how-to-fix-it.

Malodia, Suresh, Amandeep Dhir, Anil Bilgihan, Pranao Sinha, and Tanishka Tikoo. 2022. “Meme Marketing: How Can Marketers Drive Better Engagement Using Viral Memes?” *Psychology & Marketing* 39 (9): 1775–1801. https://doi.org/10.1002/mar.21702.

McKelvey, Kevin, Peter Goutzounis, Stephen da Cruz, and Nathanael Chambers. 2017. “Aligning Entity Names with Online Aliases on Twitter.” In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, edited by Lun-Wei Ku and Cheng-Te Li, 25–35. Valencia, Spain: Association for Computational Linguistics. https://doi.org/10.18653/v1/W17-1104.

Ordenes, Francisco Villarroel, Dhruv Grewal, Stephan Ludwig, Ko De Ruyter, Dominik Mahr, and Martin Wetzels. 2019. “Cutting Through Content Clutter: How Speech and Image Acts Drive Consumer Sharing of Social Media Brand Messages.” *Journal of Consumer Research* 45 (5): 988–1012. https://doi.org/10.1093/jcr/ucy032.

Pieters, Rik, Luk Warlop, and Michel Wedel. 2002. “Breaking Through the Clutter: Benefits of Advertisement Originality and Familiarity for Brand Attention and Memory.” *Management Science* 48 (6): 765–81. https://doi.org/10.1287/mnsc.48.6.765.192.

Simonov, Andrey, Tommaso Valletti, and Andre Veiga. 2025. "Attention Spillovers from News to Ads: Evidence from an Eye-Tracking Experiment." *Journal of Marketing Research* 62 (2): 294–315. https://doi.org/10.1177/00222437241256900.

Simonsohn, Uri, Andres Montealegre, and Ioannis Evangelidis. 2024. "Stimulus Sampling Reimagined: Designing Experiments with Mix-and-Match, Analyzing Results with Stimulus Plots," February. https://doi.org/10.2139/ssrn.4716832.

Zhang, Zhen, Michael J. Zyphur, and Kristopher J. Preacher. 2008. "Testing Multilevel Mediation Using Hierarchical Linear Models: Problems and Solutions." *Organizational Research Methods* 12 (4): 695–719. https://doi.org/10.1177/1094428108327450.