データ収集（Netlify + GitHub）の手順

**実験サイト公開 & データ収集セットアップ メモ**

**ゴール**

* GitHub でコードを管理し、Netlify に自動デプロイ
* 刺激（JSON）は後から差し替え可能
* （次の工程）Netlify Forms で回答データを受信

**今日やったこと（実施ログ）**

**1) GitHub リポジトリを作成**

* リポジトリ名：project\_005
* ローカルから初回コミット（方式2）

git init

git add .

git commit -m "initial commit"

git branch -M main

git remote add origin https://github.com/RenaH07/project\_005

git push -u origin main

**2) Netlify と GitHub を連携**

* Netlify ダッシュボード → **Add new project** → **Import an existing project** → **GitHub**
* 画面上部の **Configure the Netlify app on GitHub** をクリック
  + **All repositories** もしくは **Only select repositories** を選択
  + project\_004 を許可して保存
* Netlify のリポジトリ一覧に project\_004 が出現

**3) Netlify プロジェクトを作成（初回デプロイ）**

* **Build command**：空欄（ビルド無し）
* **Publish directory**：/（リポジトリ直下に index.html がある想定）
* Deploy を実行 → サイトURLが発行される

**刺激の差し替え（いつでも可能）**

* stimuli/ フォルダの JSON を置き換え / 追加
* stimuli/manifest.json の main 配列に使用ファイル名を列挙
* push すれば Netlify が自動で再デプロイ（fetch('stimuli/manifest.json', {cache:'no-store'}) なので最新が反映）

**次にやること：データ収集（Netlify Forms）**

**A) 「隠しフォーム」を index.html に入れる**

SPA（jsPsych）でも Netlify がフォームを検出できるように、**事前にフォームの定義だけ**を HTML に置く。

<!-- index.html の <body> 内（フッター付近）に設置 -->

<form name="experiment-data" method="POST" data-netlify="true" netlify netlify-honeypot="bot-field" hidden>

<input type="text" name="data" />

<input type="text" name="bot-field" /> <!-- ハニーポット（スパム対策） -->

</form>

**B) 送信コード（experiment.js は既にOK）**

// jsPsych on\_finish でまとめてPOST

fetch("/", {

method: "POST",

headers: { "Content-Type": "application/x-www-form-urlencoded" },

body: new URLSearchParams({

"form-name": "experiment-data",

"data": JSON.stringify(payload) // { id, when, data:[trials...] }

})

});

**C) Netlify 側設定**

* 初回デプロイ後に **Forms の自動検出**を有効化（必要な場合）
* 1件テスト送信 → Netlify ダッシュボードの **Forms → experiment-data** にレコードが出ることを確認
* **Download as CSV** で回答を取得可能

**D) R で読み込む最小テンプレ**

# install.packages(c("tidyverse","jsonlite"))

library(tidyverse); library(jsonlite)

subs <- readr::read\_csv("experiment-data.csv") # NetlifyからDLしたCSV

rows <- subs %>% mutate(payload = purrr::map(data, ~ fromJSON(.x)))

trials <- rows %>%

transmute(pid = purrr::map\_chr(payload, "id"),

when = purrr::map\_chr(payload, "when"),

trials = purrr::map(payload, "data")) %>%

tidyr::unnest(trials)

survey <- trials %>%

filter(trial\_type == "survey-html-form") %>%

mutate(answers = purrr::map(responses, ~ fromJSON(.x))) %>%

tidyr::unnest\_wider(answers)

# 例：主要列の確認

survey %>%

select(participant\_id, stimulus\_file, KAWAII, ANIMACY, INTENT,

APPROACH, VALENCE, PREDICT, SMOOTH, free\_text) %>% head()

**トラブルシューティング（今回ハマりかけた点）**

* **Netlify のリストにリポジトリが出ない**
  + リポジトリが**空** → 先に初回コミット（index.html を入れて push）
  + Netlify App の**リポジトリ権限未付与** → 「Configure the Netlify app on GitHub」で project\_004 を許可
  + アカウント切替の見落とし → Netlify で**正しい GitHub アカウント**を選んでいるか確認
* **Forms が出てこない**
  + index.html に**隠しフォーム**があるか
  + フォーム検出を**有効化**したか／再デプロイしたか
  + 送信時の "form-name": "experiment-data" がフォーム名と一致しているか

**付録：運用チートシート**

* **更新 → 公開**：
  1. 変更をコミット → 2) git push → 3) Netlify が自動デプロイ
* **丸の色やライン太さなどの見た目**：experiment.js の <style> 内を編集（SELECT\_COLOR/SELECT\_RING で選択色）
* **IMC の正解位置を変えたい**：imc\_silent\_pass = (v === '4') ? 1 : 0; の数値を希望位置に
* **ローカル確認**：VS Code の Live Server などで index.html を開く  
  （fetch のパスは相対参照なのでそのまま動く）

**備考**

* このチャットにアップされた**一部の添付ファイルは期限切れ**の可能性があります。必要なファイルがあれば、**再アップロード**をお願いします。
* Netlify の**フォーム送信数**はプランで上限があります。大規模収集前にダッシュボードで確認しておくと安心です。

必要なら、このメモをそのまま DEPLOY\_NOTES.md としてプッシュできるように整形してお渡しします。  
次は **隠しフォームの追加→テスト送信** を一緒にやりましょう。

データを修正したらPUSH

Set-Location -LiteralPath "G:\マイドライブ\☆刺激\project\_004"

git add experiment.js

git commit -m "Add bridge screen before main; clean duplicates"

git push

↓これでもいける

git add experiment.js

git commit -m "Fix: add resend utilities & define attemptResendPendingOnLoad"

git push

git add index.html

git commit -m "Fix: add resend utilities & define attemptResendPendingOnLoad"

git push

git status

git add stimuli/02\_g0o0\_j80\_jp197\_n0p4\_np619\_\_rot0\_\_cut0-1025.json

git commit -m "Fix: update JSON"

git push

git status

git add stimuli/01\_g0o0\_j0\_jp0\_n0\_np0\_\_rot0\_\_cut0-618.json

git commit -m "Fix: update JSON"

git push

**サイト名の変え方（Netlify）**

UIが少し変わることがあるので**どちらか使える方**で進めてください。

**方法 A（Production domains から）**

1. Netlifyで対象サイトを開く → 左メニュー **Domain management**。
2. **Production domains** の xxxxx.netlify.app の **Options** → **Edit site name**。
3. 新しい名前を入力して保存（世界で一意の名前が必要です）。 [Netlify Support Forums](https://answers.netlify.com/t/how-to-change-netlify-user-name/119891?utm_source=chatgpt.com)

**方法 B（Site settings から）**

1. サイトの **Site settings** を開く。
2. **Site information / Site details** の **Change site name** をクリック。
3. 新しい名前を入力して保存。  
   *「サイト名はグローバルに一意である必要がある」と明記されています。* [SST Guide](https://guide.sst.dev/archives/custom-domain-in-netlify.html?utm_source=chatgpt.com)

**ポイント**：survey01 のような短い一般名は既に埋まっていることが多いです。末尾に1–2文字足す（例：survey01a）と通りやすいです。[SST Guide](https://guide.sst.dev/archives/custom-domain-in-netlify.html?utm_source=chatgpt.com" \t "_blank)

**変えた後に起きること・注意点**

* **URLも一緒に変わります。**  
  旧 \*.netlify.app は**自動リダイレクトされません**（そのまま 404）。共有リンクは新URLに差し替えてください。 [Netlify Support Forums](https://answers.netlify.com/t/rename-site-will-the-old-url-redirect/87916?utm_source=chatgpt.com)
* もし**ドメインを安定させたい**場合は、独自ドメインを追加しておくと、サイト名を変えても**独自ドメインはそのまま**使い続けられます。  
  手順：**Domain management → Add a domain → Add a domain you already own** → 案内に従ってDNSを設定。完了したら **Set as primary domain**。 [Netlify Docs](https://docs.netlify.com/manage/domains/manage-domains/assign-a-domain-to-your-site-app/?utm_source=chatgpt.com)
* リダイレクト（**同一ドメイン内のパス**移動）は \_redirects で設定できます。サブドメイン自体の変更（旧 \*.netlify.app → 新 \*.netlify.app）は自動ではできません。 [Netlify Docs+1](https://docs.netlify.com/manage/routing/redirects/redirect-options/?utm_source=chatgpt.com)
* あなたの実験コードは \*\*相対パス（fetch("/") など）\*\*なので、**サイト名を変えても動作は基本そのまま**です。外部に配布したURLだけ差し替えればOK。

**Rで読み込む最小テンプレ（今回の形式に合わせて修正済み）**

survey-html-form は response カラム（= ネストしたリスト）に各項目が入ります。  
下のコードで**1つの大きな表**に展開できます。

# install.packages(c("tidyverse","jsonlite"))

library(tidyverse)

library(jsonlite)

# NetlifyからDLしたCSV

subs <- readr::read\_csv("experiment-data.csv")

# data列（JSON文字列）→ payload(list)

rows <- subs %>%

mutate(payload = purrr::map(data, ~ fromJSON(.x)))

# trials 展開（各 trial は data の配列に入っている）

trials <- rows %>%

transmute(pid = purrr::map\_chr(payload, "id"),

when = purrr::map\_chr(payload, "when"),

trials = purrr::map(payload, "data")) %>%

tidyr::unnest(trials)

# アンケートページだけ取り出し、response（list列）をフラット化

survey <- trials %>%

filter(trial\_type == "survey-html-form")

answers <- purrr::map\_dfr(survey$response, ~ as\_tibble(.x))

survey\_flat <- survey %>%

select(pid, when, participant\_id, stimulus\_file, block, trial\_index1,

imc\_silent, imc\_silent\_pass, free\_text) %>%

bind\_cols(answers) %>%

mutate(across(c(KAWAII, ANIMACY, INTENT, VALENCE, APPROACH, PREDICT, SMOOTH, IMC\_silent),

~ suppressWarnings(as.numeric(.))))

# ざっと確認

glimpse(survey\_flat)

summary(select(survey\_flat, KAWAII, ANIMACY, INTENT, VALENCE, APPROACH, PREDICT, SMOOTH))

mean(survey\_flat$imc\_silent\_pass %||% 0, na.rm = TRUE)

# 必要パッケージ（初回のみ）

# install.packages(c("tidyverse","jsonlite","janitor","afex","emmeans","broom","lavaan","semTools"))

library(tidyverse)

library(jsonlite)

library(janitor)

# 1) 読み込み

raw <- readr::read\_csv("experiment-data.csv", show\_col\_types = FALSE)

# 2) Netlify形式かどうかを判定して展開

is\_netlify <- "data" %in% names(raw) && grepl("^\\s\*\\{", raw$data[1] %||% "")

trials <-

if (is\_netlify) {

# Netlify: 1行=1参加者、data 列に全試行が JSON で入っている

raw %>%

mutate(payload = map(data, ~ tryCatch(fromJSON(.x), error = function(e) NULL))) %>%

filter(!map\_lgl(payload, is.null)) %>%

mutate(pid = map\_chr(payload, ~ .x$id),

when = map\_chr(payload, ~ .x$when),

order = map(payload, ~ .x$meta$stim\_order %||% NA),

df = map(payload, ~ as\_tibble(.x$data))) %>%

select(pid, when, order, df) %>%

unnest(df)

} else {

# すでに試行レベル：そのまま使う（列名をきれいに）

raw %>% clean\_names()

}

# 3) 評定ページ（survey-html-form）だけを抽出し、本番 block=main のみを解析対象に

dat\_main <- trials %>%

clean\_names() %>%

filter(trial\_type == "survey-html-form", block == "main")

# 4) responses(JSON文字列) をフラット化して列に展開（1..5 を numeric に）

pull\_responses <- function(s) {

if (is.na(s) || s == "") return(tibble())

x <- tryCatch(fromJSON(s), error=function(e) NULL)

if (is.null(x)) return(tibble())

as\_tibble(as.list(x))

}

dat\_resp <- dat\_main %>%

mutate(.resp = map(responses, pull\_responses)) %>%

unnest(.resp)

# 5) 刺激ID（001..012）をファイル名から抽出

dat\_resp <- dat\_resp %>%

mutate(cond\_id = str\_extract(stimulus\_file, "(?<=/|^)\\d{2,3}(?=\_)") %>% as.integer())

# 6) スケールの向き：左=ポジティブ（1=肯定）なので、

# 「数が大きいほどポジティブ」にしたい場合は反転（5->1, 4->2, …）

rev5 <- function(x) ifelse(is.na(x), NA, 6 - as.numeric(x))

dat\_resp <- dat\_resp %>%

mutate(

ANIMACY\_r = rev5(ANIMACY),

INTENT\_r = rev5(INTENT),

KAWAII\_r = rev5(KAWAII),

VALENCE\_r = rev5(VALENCE),

APPROACH\_r= rev5(APPROACH),

WARMTH\_r = rev5(WARMTH),

# SMOOTH は「左=洗練/右=ぎこちない」なので「有能さ」を作るなら反転して COMPETENCE

COMPETENCE = rev5(SMOOTH)

)

CSVの開き方

});