

```

# -----
# DECAF reanalysis – aggregated binomial with informative priors
# Author: Dr. James Brophy (cleaned & annotated by M365 Copilot)
# -----

# Reproducibility & packages -----
set.seed(20251112)

suppressPackageStartupMessages({
  library(brms)      # Bayesian GLMs via Stan
  library(cmdstanr)   # Backend for brms
  library(posterior)  # Tidy posterior draws
  library(bayesplot)  # PPC & MCMC viz helpers (we'll mainly use ggplot for
Prior/Posterior)
  library(ggplot2)
  library(scales)     # percent format
})

# Data: aggregated counts -----
# We keep "abstinence" (decaf) as the reference level – as in the DECAF report.
agg2 <- data.frame(
  group = factor(c("caffeinated", "abstinence"),
                 levels = c("abstinence", "caffeinated")), # reference first
  y      = c(47, 64),
  n      = c(100, 100)
)

stopifnot(all(levels(agg2$group) == c("abstinence", "caffeinated")))

# -----
# PRIOR CONSTRUCTION (Your scientific belief)
# -----
# Belief 1: Decaf (reference) reduces risk by 41% vs caffeinated.
RR_decaf_vs_caff <- 0.59 # RR(decaf / caffeinated)

# Belief 2: Baseline risk you want to anchor is caffeinated = 0.50.
p_caff_baseline <- 0.50

# Helper: given RR(decaf vs caff) and p_caff, compute implied:
#   p_decaf, OR(caff vs decaf), logOR(caff vs decaf), and Intercept mean = logit(p_decaf).
from_RR_and_p_caff <- function(RR_decaf_vs_caff, p_caff) {
  stopifnot(RR_decaf_vs_caff > 0, p_caff > 0, p_caff < 1)
  p_decaf <- RR_decaf_vs_caff * p_caff # decaf risk = RR * caffeinated risk
  if (p_decaf <= 0 || p_decaf >= 1) {
    warning(sprintf("Implied p_decaf = %.3f outside (0,1). Adjust RR or p_caff.",
p_decaf))
  }
  odds_caff <- p_caff / (1 - p_caff)
  odds_decaf <- p_decaf / (1 - p_decaf)
  OR_caff_vs_decaf <- odds_caff / odds_decaf
  logOR_caff_vs_decaf <- log(OR_caff_vs_decaf)
  mu_intercept <- qlogis(p_decaf) # baseline (decaf) on logit scale
  list(
    p_caff      = p_caff,
    p_decaf     = p_decaf,
    OR          = OR_caff_vs_decaf,
    logOR       = logOR_caff_vs_decaf, # prior mean for class="b"
    mu_intercept = mu_intercept        # prior mean for class="Intercept"
  )
}

conv <- from_RR_and_p_caff(RR_decaf_vs_caff, p_caff_baseline)
mu_b <- conv$logOR # prior mean for b_groupcaffeinated (log-OR: caff vs decaf)
≈ +0.871

```

```

mu_intercept <- conv$mu_intercept # prior mean for Intercept = logit(p_decaf) ≈ -0.871
p_decaf_prior_center <- plogis(mu_intercept)

message(sprintf("Prior centers: logOR_b ≈ %.3f (OR ≈ %.2f), Intercept ≈ %.3f => p_decaf ≈
%.3f",
               mu_b, exp(mu_b), mu_intercept, p_decaf_prior_center))

# Prior SDs: tune to reflect how strongly you want to encode the belief.
sd_b <- 0.5 # narrower = stronger prior on treatment effect
sd_intercept <- 1.5 # weakly-informative baseline for decaf

# Inject numeric constants (prevents Stan scope issues)
priors <- c(
  set_prior(sprintf("normal(%g, %g)", mu_b, sd_b), class = "b"),
  set_prior(sprintf("normal(%g, %g)", mu_intercept, sd_intercept), class = "Intercept")
)

# -----
# MODEL FIT – aggregated binomial with trials(n)
# -----
fit_binom2 <- brm(
  formula = brms::bf(y | trials(n) ~ 1 + group), # groupcaffeinated is log-OR(caff vs decaf)
  data = agg2,
  family = binomial(), # logit link
  prior = priors,
  chains = 4, iter = 4000, warmup = 1000, refresh = 0,
  backend = "cmdstanr",
  control = list(adapt_delta = 0.95)
)

print(summary(fit_binom2), digits = 3)

# -----
# POSTERIOR PROBABILITIES PER ARM
# IMPORTANT: With trials(n), posterior_epred() gives expected COUNTS.
# Use posterior_linpred(..., transform=TRUE) or fitted(..., scale="linear") for
# probabilities.
# -----
nd <- data.frame(
  group = factor(c("abstinence", "caffeinated"), levels = levels(agg2$group)),
  n = c(100, 100) # required for trials(n) even if we only want probabilities
)

p_post <- posterior_linpred(fit_binom2, newdata = nd, transform = TRUE) # draws x 2
p_decaf_post <- p_post[, 1] # probabilities in [0,1]
p_caff_post <- p_post[, 2]

stopifnot(all(p_decaf_post >= 0 & p_decaf_post <= 1))
stopifnot(all(p_caff_post >= 0 & p_caff_post <= 1))

# -----
# PRIOR-vs-POSTERIOR OVERLAYS (parameters & clinical scales)
# Use ggplot (not mcmc*_chains) so legend is "Prior"/"Posterior", not "Chain".
# -----

# Posterior draws for the coefficient (log-OR: caff vs decaf)
draws <- as_draws_df(fit_binom2)
post_b <- draws$b_groupcaffeinated

# Prior draws (on log-odds scale)
S <- length(post_b)
prior_b <- rnorm(S, mean = mu_b, sd = sd_b)

# (A) log-OR overlay

```

```

df_b <- rbind(
  data.frame(value = prior_b, which = "Prior"),
  data.frame(value = post_b, which = "Posterior")
)

ggplot(df_b, aes(x = value, colour = which, fill = which)) +
  geom_density(alpha = 0.15) +
  labs(x = "log-OR (caffeinated vs decaf)", y = NULL,
       title = "Prior vs Posterior for log-OR (caffeinated vs decaf)") +
  theme_bw()

# (B) Odds ratio overlay
df_or <- rbind(
  data.frame(value = exp(prior_b), which = "Prior"),
  data.frame(value = exp(post_b), which = "Posterior")
)

ggplot(df_or, aes(x = value, colour = which, fill = which)) +
  geom_density(alpha = 0.15) +
  geom_vline(xintercept = 1, linetype = 2) +
  labs(x = "Odds ratio (caffeinated / decaf)", y = NULL,
       title = "Prior vs Posterior for OR") +
  theme_bw()

# (C) Arm risks (probability scale)
# Rebuild PRIOR samples on the logit scale -> probabilities
prior_intercept <- rnorm(S, mean = mu_intercept, sd = sd_intercept) # ~logit(p_decaf)
prior_beta <- rnorm(S, mean = mu_b, sd = sd_b) # ~log-OR

inv_logit <- function(x) 1 / (1 + exp(-x))
p_decaf_prior <- inv_logit(prior_intercept)
p_caff_prior <- inv_logit(prior_intercept + prior_beta)

stopifnot(all(p_decaf_prior >= 0 & p_decaf_prior <= 1))
stopifnot(all(p_caff_prior >= 0 & p_caff_prior <= 1))

# Plot: Decaf risk
df_decaf <- rbind(
  data.frame(value = p_decaf_prior, which = "Prior"),
  data.frame(value = p_decaf_post, which = "Posterior")
)

ggplot(df_decaf, aes(x = value, colour = which, fill = which)) +
  geom_density(alpha = 0.15) +
  scale_x_continuous(labels = percent, limits = c(0, 1)) +
  labs(x = "Decaf (reference) risk", y = NULL,
       title = "Decaf risk: Prior vs Posterior") +
  theme_bw()

# Plot: Caffeinated risk
df_caff <- rbind(
  data.frame(value = p_caff_prior, which = "Prior"),
  data.frame(value = p_caff_post, which = "Posterior")
)

ggplot(df_caff, aes(x = value, colour = which, fill = which)) +
  geom_density(alpha = 0.15) +
  scale_x_continuous(labels = percent, limits = c(0, 1)) +
  labs(x = "Caffeinated risk", y = NULL,
       title = "Caffeinated risk: Prior vs Posterior") +
  theme_bw()

# (D) Derived effects (RD & RR)
rd_prior <- p_caff_prior - p_decaf_prior
rr_prior <- p_caff_prior / p_decaf_prior

```

```

rd_post <- p_caff_post - p_decaf_post
rr_post <- p_caff_post / p_decaf_post

# RD overlay
df_rd <- rbind(
  data.frame(value = rd_prior, which = "Prior"),
  data.frame(value = rd_post, which = "Posterior")
)
ggplot(df_rd, aes(x = value, colour = which, fill = which)) +
  geom_density(alpha = 0.15) +
  geom_vline(xintercept = 0, linetype = 2) +
  labs(x = "Risk difference (caffeinated - decaf)", y = NULL,
       title = "RD: Prior vs Posterior") +
  theme_bw()

# RR overlay
df_rr <- rbind(
  data.frame(value = rr_prior, which = "Prior"),
  data.frame(value = rr_post, which = "Posterior")
)
ggplot(df_rr, aes(x = value, colour = which, fill = which)) +
  geom_density(alpha = 0.15) +
  geom_vline(xintercept = 1, linetype = 2) +
  labs(x = "Risk ratio (caffeinated / decaf)", y = NULL,
       title = "RR: Prior vs Posterior") +
  theme_bw()

# -----
# POSTERIOR PREDICTIVE CHECKS (COUNTS)
# With 2 aggregated rows, use arm-specific bar PPCs for clarity.
# -----
# Include n in newdata because we used trials(n)
nd_abst <- data.frame(group = factor("abstinence", levels = levels(agg2$group)), n = 100)
nd_caff <- data.frame(group = factor("caffeinated", levels = levels(agg2$group)), n = 100)

yrep_abst <- posterior_predict(fit_binom2, newdata = nd_abst, ndraws = 400)[, 1]
yrep_caff <- posterior_predict(fit_binom2, newdata = nd_caff, ndraws = 400)[, 1]

y_abst <- agg2$y[agg2$group == "abstinence"]
y_caff <- agg2$y[agg2$group == "caffeinated"]

bayesplot::ppc_bars(y = y_abst, yrep = matrix(yrep_abst, ncol = 1)) +
  ggplot2::ggtitle("PPC - Abstinence (decaf) counts")

bayesplot::ppc_bars(y = y_caff, yrep = matrix(yrep_caff, ncol = 1)) +
  ggplot2::ggtitle("PPC - Caffeinated counts")

# -----
# SUMMARIES (posterior means/CrIs for risks and contrasts)
# -----
summ <- function(x) c(mean = mean(x), sd = sd(x),
                      q025 = quantile(x, 0.025),
                      q50 = median(x),
                      q975 = quantile(x, 0.975))

out_tab <- rbind(
  p_decaf = summ(p_decaf_post),
  p_caffeinated = summ(p_caff_post),
  RD = summ(rd_post),
  RR = summ(rr_post),
  OR = summ(exp(post_b))
)
round(out_tab, 3)

```

```

# Posterior predictive checks – “continuous” overlays
# 1A) Density overlay of counts (y vs yrep)

y_obs <- agg2$y # observed counts: c(47, 64)

# Replicated counts from the posterior (draws × 2)
yrep <- posterior_predict(fit_binom2, newdata = nd, ndraws = 20)

nd <- data.frame(
  group = factor(c("abstinence","caffeinated"), levels = levels(agg2$group)),
  n = c(100, 100)
)
y_obs <- agg2$y # c(47, 64)

bayesplot::ppc_dens_overlay(y = y_obs, yrep = yrep) +
  ggplot2::ggtitle("Posterior predictive – density overlay (counts)")

bayesplot::ppc_ecdf_overlay(y = y_obs, yrep = yrep) +
  ggplot2::ggtitle("Posterior predictive – ECDF overlay (counts)")

bayesplot::ppc_intervals(y = y_obs, yrep = yrep) +
  ggplot2::ggtitle("Posterior predictive – predictive intervals (counts)")

bayesplot::ppc_ribbon(y = y_obs, yrep = yrep) +
  ggplot2::ggtitle("Posterior predictive – ribbons (counts)")

# Abstinence (decaf): column 1
bayesplot::ppc_hist(y = y_obs[1], yrep = yrep[, 1, drop = FALSE], binwidth = 1) +
  ggplot2::ggtitle("PPC – Abstinence (decaf): histogram of replicated counts")

# Caffeinated: column 2
bayesplot::ppc_hist(y = y_obs[2], yrep = yrep[, 2, drop = FALSE], binwidth = 1) +
  ggplot2::ggtitle("PPC – Caffeinated: histogram of replicated counts")

# Abstinence (decaf)
ggplot(data.frame(yrep = yrep[, 1]), aes(x = yrep)) +
  geom_density(fill = "skyblue", alpha = 0.2) +
  geom_vline(xintercept = y_obs[1], linetype = 2) +
  labs(title = "PPC – Abstinence (decaf): density of replicated counts", x = "count", y =
NULL) +
  theme_bw()

# Caffeinated
ggplot(data.frame(yrep = yrep[, 2]), aes(x = yrep)) +
  geom_density(fill = "salmon", alpha = 0.2) +
  geom_vline(xintercept = y_obs[2], linetype = 2) +
  labs(title = "PPC – Caffeinated: density of replicated counts", x = "count", y = NULL) +
  theme_bw()

```