phase7_era_stratification.R

abdulbasir

2025-12-09

```r
# phase 7: era-specific interaction testing
#
# investigating if the main effects vary across different time periods through
# stratified ANOVA analyses for each regulatory era

# sourcing configuration and utilities
source("config.R")
source("utils.R")

# loading required libraries
suppressPackageStartupMessages({
  library(dplyr)
  library(readr)
  library(ggplot2)
  library(car)  # for Type III SS (Marginal) on unbalanced designs
})

print_section_header("Phase 7: Era-Specific Interaction Testing")
```

```
##
## ======================================================================
## Phase 7: Era-Specific Interaction Testing
## ======================================================================
```

```r
# 1. loading analysis-ready data from phase 5
input_file = file.path(RESULTS_DIR, "analysis_ready_dataset.csv")
analysis_data = load_csv(input_file)

cat(paste("Analysis-ready sample: n =", nrow(analysis_data), "\n"))
```

```
## Analysis-ready sample: n = 1038
```

```r
# 2. extracting era subsets
era_list = c("Pre-PDUFA", "Early-PDUFA", "Mid-PDUFA", "Post-FDASIA")
era_subsets = list()

for (era_name in era_list) {
  era_subsets[[era_name]] = analysis_data %>%
    filter(regulatory_era_factor == era_name)
}

cat("Era-stratified subsets created:\n")
```

```
## Era-stratified subsets created:
```

```r
for (era_name in era_list) {
  cat(sprintf("  %s: n = %3d\n", era_name, nrow(era_subsets[[era_name]])))
}
```

```
##   Pre-PDUFA: n = 207
##   Early-PDUFA: n = 336
##   Mid-PDUFA: n = 181
##   Post-FDASIA: n = 314
```

```r
# 3. running ANOVAs for all eras
era_results = data.frame(
  era = character(),
  n = integer(),
  interaction_F = numeric(),
  interaction_df_num = integer(),
  interaction_df_denom = integer(),
  interaction_p = numeric(),
  singular = logical(),
  stringsAsFactors = FALSE
)

for (era_name in era_list) {
  era_df = era_subsets[[era_name]]
  n_total = nrow(era_df)

  cat(sprintf("\n%s (n=%d):\n", era_name, n_total))

  if (n_total < 20) {
    cat("  Warning: insufficient sample size for ANOVA\n")
    next
  }

  # checking for singularity (Post-FDASIA has 0 "Other" drugs)
  n_therapeutic = length(unique(era_df$therapeutic_area_factor))
  n_review = length(unique(era_df$review_type_factor))

  if (n_therapeutic < 2 || n_review < 2) {
    cat(sprintf("  Insufficient factor variation (%d areas, %d review types) - interaction undefined\n"
                n_therapeutic, n_review))
    era_results = rbind(era_results, data.frame(
      era = era_name,
      n = n_total,
      interaction_F = NA,
      interaction_df_num = NA,
      interaction_df_denom = NA,
      interaction_p = NA,
      singular = TRUE
    ))
    next
  }
  # fitting ANOVA model
  model = lm(
    log_review_time_days_response ~ therapeutic_area_factor * review_type_factor,
    data = era_df
  )

  anova_result = car::Anova(model, type = 3)

  # extracting interaction statistics
  f_interaction = anova_result["therapeutic_area_factor:review_type_factor", "F value"]
  df_num = anova_result["therapeutic_area_factor:review_type_factor", "Df"]
  df_denom = anova_result["Residuals", "Df"]
  p_interaction = anova_result["therapeutic_area_factor:review_type_factor", "Pr(>F)"]

  cat(sprintf("  Interaction: F(%d,%d) = %.2f, p = %.4f\n",
              df_num, df_denom, f_interaction, p_interaction))

  era_results = rbind(era_results, data.frame(
    era = era_name,
    n = n_total,
    interaction_F = f_interaction,
    interaction_df_num = df_num,
    interaction_df_denom = df_denom,
    interaction_p = p_interaction,
    singular = FALSE
  ))
}
```

```
##
## Pre-PDUFA (n=207):
##   Interaction: F(1,203) = 2.03, p = 0.1560
##
## Early-PDUFA (n=336):
##   Interaction: F(1,332) = 0.00, p = 0.9562
##
## Mid-PDUFA (n=181):
##   Interaction: F(1,177) = 0.03, p = 0.8556
##
## Post-FDASIA (n=314):
##   Interaction: F(1,310) = 1.99, p = 0.1592
```

```r
cat("\nEra-stratified interaction tests:\n")
```

```
##
## Era-stratified interaction tests:
```

```r
print(era_results)
```

```
##           era   n interaction_F interaction_df_num interaction_df_denom interaction_p singular
## 1   Pre-PDUFA 207   2.027876941                  1                  203     0.1559715    FALSE
## 2 Early-PDUFA 336   0.003027732                  1                  332     0.9561518    FALSE
## 3   Mid-PDUFA 181   0.033221647                  1                  177     0.8555809    FALSE
## 4 Post-FDASIA 314   1.991363085                  1                  310     0.1592012    FALSE
```

```
# 4. computing cell means per era
era_cell_means = list()

for (era_name in era_list) {
  era_df = era_subsets[[era_name]]

  if (nrow(era_df) > 0) {
    means = era_df %>%
      group_by(therapeutic_area_factor, review_type_factor) %>%
      summarise(
        mean_log_time = mean(log_review_time_days_response, na.rm = TRUE),
        n = n(),
        .groups = "drop"
      )

    era_cell_means[[era_name]] = means
    cat(sprintf("\n%s cell means (log scale):\n", era_name))
    print(means)
  }
}
```

```
##
## Pre-PDUFA cell means (log scale):
## # A tibble: 4 x 4
##   therapeutic_area_factor review_type_factor mean_log_time     n
##   <chr>                   <chr>                      <dbl> <int>
## 1 Oncology                Priority                    6.25    20
## 2 Oncology                Standard                    7.08     5
## 3 Other                   Priority                    6.52    81
## 4 Other                   Standard                    6.90   101
##
## Early-PDUFA cell means (log scale):
## # A tibble: 4 x 4
##   therapeutic_area_factor review_type_factor mean_log_time     n
##   <chr>                   <chr>                      <dbl> <int>
## 1 Oncology                Priority                    5.84    32
## 2 Oncology                Standard                    6.53    19
## 3 Other                   Priority                    5.71   106
## 4 Other                   Standard                    6.39   179
##
## Mid-PDUFA cell means (log scale):
## # A tibble: 4 x 4
##   therapeutic_area_factor review_type_factor mean_log_time     n
##   <chr>                   <chr>                      <dbl> <int>
## 1 Oncology                Priority                    5.25    45
## 2 Oncology                Standard                    5.90    13
## 3 Other                   Priority                    5.72    63
## 4 Other                   Standard                    6.32    60
##
## Post-FDASIA cell means (log scale):
## # A tibble: 4 x 4
##   therapeutic_area_factor review_type_factor mean_log_time     n
##   <chr>                   <chr>                      <dbl> <int>
## 1 Oncology                Priority                    5.39   127
## 2 Oncology                Standard                    6.00    38
## 3 Other                   Priority                    5.63   117
## 4 Other                   Standard                    6.08    32
```

```r
# 5. computing simple effects per era
simple_effects = data.frame(
  era = character(),
  other_benefit_pct = numeric(),
  oncology_benefit_pct = numeric(),
  benefit_difference = numeric(),
  stringsAsFactors = FALSE
)

for (era_name in era_list) {
  if (era_name %in% names(era_cell_means)) {
    means_df = era_cell_means[[era_name]]

    # extracting cell means
    other_priority = means_df %>%
      filter(therapeutic_area_factor == "Other", review_type_factor == "Priority") %>%
      pull(mean_log_time)
    other_standard = means_df %>%
      filter(therapeutic_area_factor == "Other", review_type_factor == "Standard") %>%
      pull(mean_log_time)
    onc_priority = means_df %>%
      filter(therapeutic_area_factor == "Oncology", review_type_factor == "Priority") %>%
      pull(mean_log_time)
    onc_standard = means_df %>%
      filter(therapeutic_area_factor == "Oncology", review_type_factor == "Standard") %>%
      pull(mean_log_time)

    # calculating percentage benefit (on log scale)
    if (length(other_priority) > 0 && length(other_standard) > 0) {
      other_benefit = ((other_standard - other_priority) / other_standard) * 100
    } else {
      other_benefit = NA
    }

    if (length(onc_priority) > 0 && length(onc_standard) > 0) {
      onc_benefit = ((onc_standard - onc_priority) / onc_standard) * 100
    } else {
      onc_benefit = NA
    }

    # differential benefit
    if (!is.na(other_benefit) && !is.na(onc_benefit)) {
      benefit_diff = onc_benefit - other_benefit
    } else {
      benefit_diff = NA
    }

    simple_effects = rbind(simple_effects, data.frame(
      era = era_name,
      other_benefit_pct = other_benefit,
      oncology_benefit_pct = onc_benefit,
      benefit_difference = benefit_diff
    ))
  }
}

cat("\nSimple effects (priority benefit %):\n")
```

```
##
## Simple effects (priority benefit %):
```

```r
print(simple_effects)
```

```
##           era other_benefit_pct oncology_benefit_pct benefit_difference
## 1   Pre-PDUFA          5.559333             11.76688         6.20754220
## 2 Early-PDUFA         10.725941             10.66799        -0.05794745
## 3   Mid-PDUFA          9.555837             10.91594         1.36009983
## 4 Post-FDASIA          7.451984             10.12117         2.66918852
```

```r
# 6. testing temporal trend
valid_eras = era_results %>% filter(!is.na(interaction_F) & !singular)

if (nrow(valid_eras) >= 3) {
  # mapping era to numeric
  era_to_numeric = c("Pre-PDUFA" = 1, "Early-PDUFA" = 2, "Mid-PDUFA" = 3, "Post-FDASIA" = 4)

  era_numeric = sapply(valid_eras$era, function(x) era_to_numeric[x])
  f_values = valid_eras$interaction_F

  # spearman correlation
  cor_test = cor.test(era_numeric, f_values, method = "spearman")
  rho = cor_test$estimate
  p_temporal = cor_test$p.value

  cat(sprintf("\nTemporal trend test (Spearman  ):\n"))
  cat(sprintf("  Using %d valid eras\n", nrow(valid_eras)))
  cat(sprintf("    = %.4f, p = %.4f\n", rho, p_temporal))

  if (p_temporal < ALPHA) {
    if (rho > 0) {
      cat("  Significant: increasing trend (interaction strengthens over time)\n")
    } else {
      cat("  Significant: decreasing trend (interaction weakens over time)\n")
    }
  } else {
    cat("  Non-significant: no temporal trend detected\n")
  }
} else {
  cat(sprintf("\nInsufficient valid eras for temporal trend test (only %d eras)\n", nrow(valid_eras)))
  rho = NA
  p_temporal = NA
}
```

```
##
## Temporal trend test (Spearman  ):
##   Using 4 valid eras
##     = -0.2000, p = 0.9167
##   Non-significant: no temporal trend detected
```

```
# 7. combining era results
combined_results = era_results %>%
  left_join(simple_effects, by = "era") %>%
  mutate(
    significant = interaction_p < ALPHA,
    interpretation = case_when(
      singular ~ "Undefined (singular)",
      is.na(interaction_F) ~ "Insufficient data",
      significant ~ sprintf("Significant (p=%.4f)", interaction_p),
      TRUE ~ sprintf("Non-significant (p=%.4f)", interaction_p)
    )
  )

cat("\nCombined era-stratified results:\n")
```

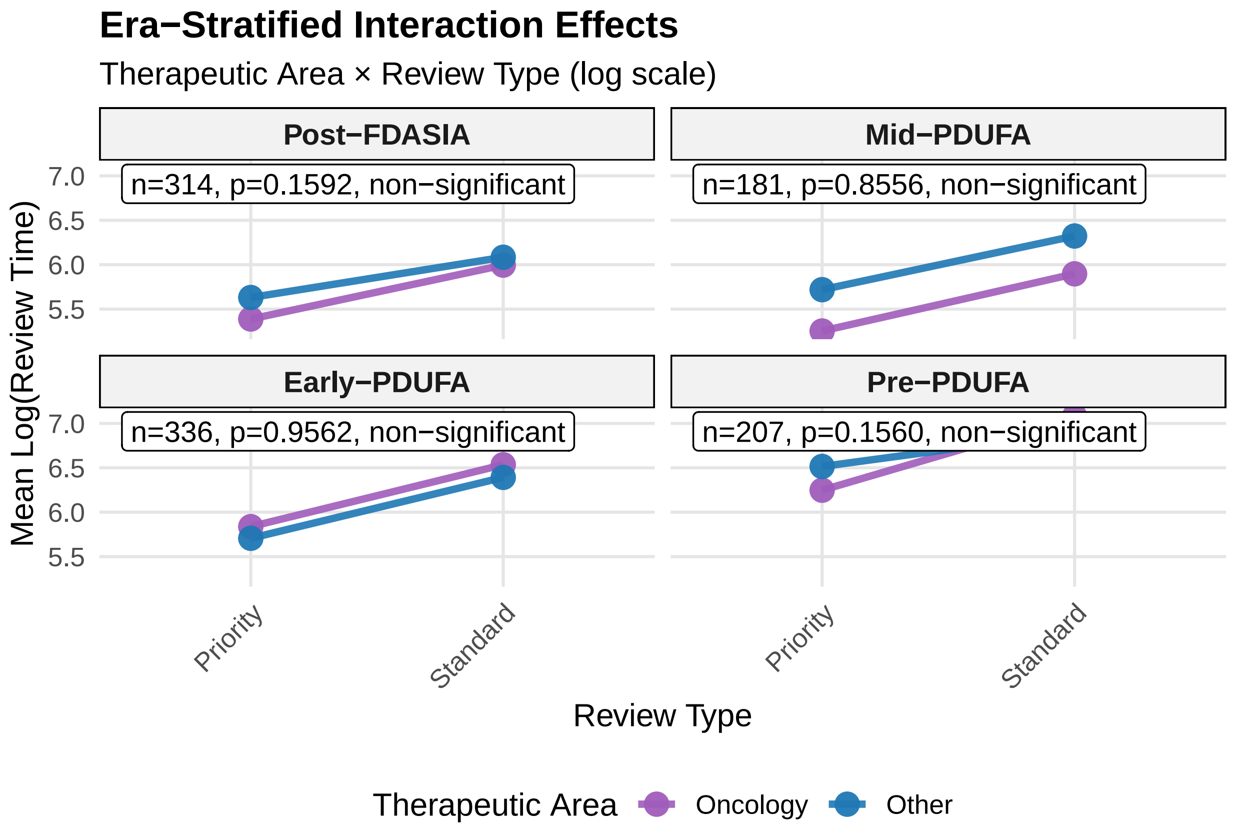
```
##
## Combined era-stratified results:
```

```
print(combined_results)
```

```
##           era   n interaction_F interaction_df_num interaction_df_denom interaction_p singular
## 1   Pre-PDUFA 207   2.027876941                  1                  203     0.1559715    FALSE
## 2 Early-PDUFA 336   0.003027732                  1                  332     0.9561518    FALSE
## 3   Mid-PDUFA 181   0.033221647                  1                  177     0.8555809    FALSE
## 4 Post-FDASIA 314   1.991363085                  1                  310     0.1592012    FALSE
##   other_benefit_pct oncology_benefit_pct benefit_difference significant            interpretation
## 1          5.559333             11.76688         6.20754220       FALSE Non-significant (p=0.1560)
## 2         10.725941             10.66799        -0.05794745       FALSE Non-significant (p=0.9562)
## 3          9.555837             10.91594         1.36009983       FALSE Non-significant (p=0.8556)
## 4          7.451984             10.12117         2.66918852       FALSE Non-significant (p=0.1592)
```

```
# 8. visualization: era_stratified_interaction_plot.png
# prepare data for plotting
plot_data = data.frame()

for (era_name in era_list) {
  if (era_name %in% names(era_cell_means)) {
    means_df = era_cell_means[[era_name]] %>%
      mutate(era = era_name)
    plot_data = rbind(plot_data, means_df)
  }
}

# CRITICAL: Quadrant layout ordering
# Q2 (top-left): Post-FDASIA
# Q1 (top-right): Mid-PDUFA
```

```r
# Q3 (bottom-left): Early-PDUFA
# Q4 (bottom-right): Pre-PDUFA
plot_data$era = factor(
  plot_data$era,
  levels = c("Post-FDASIA", "Mid-PDUFA", "Early-PDUFA", "Pre-PDUFA")
)

# creating metadata for labels
era_meta = era_results %>%
  mutate(
    n_label = if_else(is.na(n), "n=0", paste0("n=", n)),
    p_label = if_else(is.na(interaction_p), "p=NA", sprintf("p=%.4f", interaction_p)),
    sig_label = case_when(
      singular ~ "insufficient data",
      is.na(interaction_p) ~ "insufficient data",
      interaction_p < ALPHA ~ "significant",
      TRUE ~ "non-significant"
    )
  )

# creating alt plot data with labels
plot_data_alt = plot_data %>%
  left_join(era_meta %>% select(era, n_label, p_label, sig_label), by = "era")

# re-apply factor ordering after join to ensure quadrant layout
plot_data_alt$era = factor(
  plot_data_alt$era,
  levels = c("Post-FDASIA", "Mid-PDUFA", "Early-PDUFA", "Pre-PDUFA")
)

era_interaction_plot = ggplot(
  plot_data_alt,
  aes(x = review_type_factor, y = mean_log_time,
      color = therapeutic_area_factor, group = therapeutic_area_factor)
) +
  geom_line(linewidth = 1.3, alpha = 0.9) +
  geom_point(size = 3.8, alpha = 0.95) +
  facet_wrap(~ era, ncol = 2, drop = FALSE) +
  labs(
    title = "Era-Stratified Interaction Effects",
    subtitle = "Therapeutic Area × Review Type (log scale)",
    x = "Review Type",
    y = "Mean Log(Review Time)",
    color = "Therapeutic Area"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    plot.title = element_text(face = "bold", size = 14),
    plot.background = element_rect(fill = "white", color = NA),
    panel.background = element_rect(fill = "white", color = NA),
    panel.grid.major = element_line(color = "gray90"),
    panel.grid.minor = element_blank(),
    strip.background = element_rect(fill = "gray95", color = "black"),
    strip.text = element_text(face = "bold", size = 11),
    legend.position = "bottom",
    axis.text = element_text(size = 10),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  scale_color_manual(values = c("Other" = "#1f78b4", "Oncology" = "#a05dbb")) +
  geom_label(
    data = plot_data_alt %>% group_by(era) %>% slice(1),
    aes(x = -Inf, y = Inf, label = paste0(n_label, ", ", p_label, ", ", sig_label)),
    hjust = -0.05,
    vjust = 1.1,
    fill = "white",
    color = "black",
    label.size = 0.3,
    inherit.aes = FALSE
  )

print(era_interaction_plot)
```



```
ggsave(
  file.path(FIGURES_DIR, "era_stratified_interaction_plot.png"),
  plot = era_interaction_plot,
  width = 10,
  height = 8,
  dpi = DPI
```

```r
)

cat("Saved: era_stratified_interaction_plot.png\n")
```

```
## Saved: era_stratified_interaction_plot.png
```

```r
# 9. saving results
combined_output = file.path(RESULTS_DIR, "era_stratified_results.csv")
save_csv(combined_results, combined_output)
```

```
## saving results to: /Users/abdulbasir/Downloads/Experimental AI/fda-oncology-approval-analysis/results
```

```r
cat("\nPhase 7 complete\n")
```

```
##
## Phase 7 complete
```