

phase6_anova_modeling.R

abdulbasir

2025-12-09

```
# phase 6: anova modeling with blocking and permutation tests
#
# fitting three models: raw scale no blocking, raw scale with era blocking,
# log scale with era blocking, plus effect sizes and permutation tests

# 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
  library(patchwork)
})

print_section_header("Phase 6: ANOVA Modeling With Blocking And Permutation Tests")

##
## =====
## Phase 6: ANOVA Modeling With Blocking And Permutation Tests
## =====

# 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

# 2. verifying factor variables exist from phase 5
# Phase 5 already created: review_type_factor, therapeutic_area_factor, regulatory_era_factor
cat("Factor variables from Phase 5:\n")

## Factor variables from Phase 5:
```

```

cat("  therapeutic_area_factor:", paste(levels(analysis_data$therapeutic_area_factor), collapse = ", "))

##  therapeutic_area_factor:

cat("  review_type_factor:", paste(levels(analysis_data$review_type_factor), collapse = " ", "\n"))

##  review_type_factor:

cat("  regulatory_era_factor:", paste(levels(analysis_data$regulatory_era_factor), collapse = ", "), "\n")

##  regulatory_era_factor:

# verifying we have all 860 drugs (not 853!)
cat(sprintf("Total observations: %d (expecting 860)\n", nrow(analysis_data)))

## Total observations: 1038 (expecting 860)

# 3. model 1: raw scale, no blocking
model1 = lm(
  review_time_days_response ~ therapeutic_area_factor * review_type_factor,
  data = analysis_data
)

anova1 = car::Anova(model1, type = 3)

cat("\nModel 1: Two-way ANOVA (raw scale, no blocking, Type III SS)\n")

## 
## Model 1: Two-way ANOVA (raw scale, no blocking, Type III SS)

print(anova1)

## Anova Table (Type III tests)
##
## Response: review_time_days_response
##                                     Sum Sq Df F value    Pr(>F)
## (Intercept)                20192413  1 99.1291 < 2.2e-16 ***
## therapeutic_area_factor      4321644  1 21.2159 4.614e-06 ***
## review_type_factor          4919766  1 24.1522 1.034e-06 ***
## therapeutic_area_factor:review_type_factor 16435   1  0.0807   0.7764
## Residuals                  210623787 1034
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# calculating effect sizes (eta-squared)
# Type III SS: extract by row name (skip Intercept)
ss_area = anova1["therapeutic_area_factor", "Sum Sq"]
ss_review = anova1["review_type_factor", "Sum Sq"]
ss_interaction = anova1["therapeutic_area_factor:review_type_factor", "Sum Sq"]

```

```

ss_residual = anova1["Residuals", "Sum Sq"]
ss_total = ss_area + ss_review + ss_interaction + ss_residual

eta_sq_area = ss_area / ss_total
eta_sq_review = ss_review / ss_total
eta_sq_interaction = ss_interaction / ss_total

cat("\nEffect sizes (eta-squared):\n")

##
## Effect sizes (eta-squared):

cat(sprintf(" Therapeutic area: %.4f\n", eta_sq_area))

## Therapeutic area: 0.0197

cat(sprintf(" Review type: %.4f\n", eta_sq_review))

## Review type: 0.0224

cat(sprintf(" Interaction: %.4f\n", eta_sq_interaction))

## Interaction: 0.0001

# extracting key statistics
f_area_m1 = anova1["therapeutic_area_factor", "F value"]
p_area_m1 = anova1["therapeutic_area_factor", "Pr(>F)"]
f_review_m1 = anova1["review_type_factor", "F value"]
p_review_m1 = anova1["review_type_factor", "Pr(>F)"]
f_interaction_m1 = anova1["therapeutic_area_factor:review_type_factor", "F value"]
p_interaction_m1 = anova1["therapeutic_area_factor:review_type_factor", "Pr(>F)"]

cat("\nKey statistics:\n")

##
## Key statistics:

cat(sprintf(" Therapeutic area: F=% .2f, p=% .2e\n", f_area_m1, p_area_m1))

## Therapeutic area: F=21.22, p=4.61e-06

cat(sprintf(" Review type: F=% .2f, p=% .2e\n", f_review_m1, p_review_m1))

## Review type: F=24.15, p=1.03e-06

cat(sprintf(" Interaction: F=% .2f, p=% .2e\n", f_interaction_m1, p_interaction_m1))

## Interaction: F=0.08, p=7.76e-01

```

```

# 4. model 2: raw scale, with era blocking
model2 = lm(
  review_time_days_response ~ therapeutic_area_factor * review_type_factor + regulatory_era_factor,
  data = analysis_data
)

anova2 = car::Anova(model2, type = 3)

cat("\nModel 2: Two-way ANOVA with blocking (raw scale, Type III SS)\n")

## 
## Model 2: Two-way ANOVA with blocking (raw scale, Type III SS)

print(anova2)

## Anova Table (Type III tests)
##
## Response: review_time_days_response
##                                     Sum Sq Df F value    Pr(>F)
## (Intercept)                 16809938  1 99.9460 < 2.2e-16 ***
## therapeutic_area_factor      880849   1  5.2372  0.02231 *
## review_type_factor          4780408   1 28.4226 1.198e-07 ***
## regulatory_era_factor       37219720   3 73.7652 < 2.2e-16 ***
## therapeutic_area_factor:review_type_factor 53863   1  0.3202  0.57158
## Residuals                  173404067 1031
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# extracting key statistics
f_area_m2 = anova2["therapeutic_area_factor", "F value"]
p_area_m2 = anova2["therapeutic_area_factor", "Pr(>F)"]
f_review_m2 = anova2["review_type_factor", "F value"]
p_review_m2 = anova2["review_type_factor", "Pr(>F)"]
f_interaction_m2 = anova2["therapeutic_area_factor:review_type_factor", "F value"]
p_interaction_m2 = anova2["therapeutic_area_factor:review_type_factor", "Pr(>F)"]
f_era_m2 = anova2["regulatory_era_factor", "F value"]
p_era_m2 = anova2["regulatory_era_factor", "Pr(>F)"]

cat("\nKey statistics:\n")

## 
## Key statistics:

cat(sprintf(" Therapeutic area: F=% .2f, p=% .2e\n", f_area_m2, p_area_m2))

## Therapeutic area: F=5.24, p=2.23e-02

cat(sprintf(" Review type: F=% .2f, p=% .2e\n", f_review_m2, p_review_m2))

## Review type: F=28.42, p=1.20e-07

```

```

cat(sprintf("  Interaction: F=% .2f, p=% .2e\n", f_interaction_m2, p_interaction_m2))

##  Interaction: F=0.32, p=5.72e-01

cat(sprintf("  Regulatory era (block): F=% .2f, p=% .2e\n", f_era_m2, p_era_m2))

##  Regulatory era (block): F=73.77, p=3.17e-43

# 5. model 3: log scale, with era blocking (PREFERRED MODEL)
model3 = lm(
  log_review_time_days_response ~ therapeutic_area_factor * review_type_factor + regulatory_era_factor,
  data = analysis_data
)

anova3 = car::Anova(model3, type = 3)

cat("\nModel 3: Two-way ANOVA with blocking (log scale, Type III SS, PREFERRED)\n")

## 
## Model 3: Two-way ANOVA with blocking (log scale, Type III SS, PREFERRED)

print(anova3)

## Anova Table (Type III tests)
##
## Response: log_review_time_days_response
##                                     Sum Sq   Df   F value    Pr(>F)
## (Intercept)                   3724.7   1 11171.9454 < 2.2e-16 ***
## therapeutic_area_factor        6.2     1   18.6996 1.679e-05 ***
## review_type_factor            25.8     1   77.5170 < 2.2e-16 ***
## regulatory_era_factor         88.7     3   88.6370 < 2.2e-16 ***
## therapeutic_area_factor:review_type_factor  0.8     1    2.4333   0.1191
## Residuals                      343.7 1031
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# extracting key statistics
f_area_m3 = anova3["therapeutic_area_factor", "F value"]
p_area_m3 = anova3["therapeutic_area_factor", "Pr(>F)"]
f_review_m3 = anova3["review_type_factor", "F value"]
p_review_m3 = anova3["review_type_factor", "Pr(>F)"]
f_interaction_m3 = anova3["therapeutic_area_factor:review_type_factor", "F value"]
p_interaction_m3 = anova3["therapeutic_area_factor:review_type_factor", "Pr(>F)"]
f_era_m3 = anova3["regulatory_era_factor", "F value"]
p_era_m3 = anova3["regulatory_era_factor", "Pr(>F)"]

# effect sizes for model 3 (log scale, blocked)
ss_area_m3 = anova3["therapeutic_area_factor", "Sum Sq"]
ss_review_m3 = anova3["review_type_factor", "Sum Sq"]
ss_interaction_m3 = anova3["therapeutic_area_factor:review_type_factor", "Sum Sq"]

```

```

ss_era_m3 = anova3["regulatory_era_factor", "Sum Sq"]
ss_resid_m3 = anova3["Residuals", "Sum Sq"]
ss_total_m3 = ss_area_m3 + ss_review_m3 + ss_interaction_m3 + ss_era_m3 + ss_resid_m3

eta_sq_area_m3 = ss_area_m3 / ss_total_m3
eta_sq_review_m3 = ss_review_m3 / ss_total_m3
eta_sq_interaction_m3 = ss_interaction_m3 / ss_total_m3
eta_sq_era_m3 = ss_era_m3 / ss_total_m3

cat("\nKey statistics:\n")

## Key statistics:

cat(sprintf(" Therapeutic area: F=% .2f, p=% .2e\n", f_area_m3, p_area_m3))

## Therapeutic area: F=18.70, p=1.68e-05

cat(sprintf(" Review type: F=% .2f, p=% .2e\n", f_review_m3, p_review_m3))

## Review type: F=77.52, p=5.47e-18

cat(sprintf(" Interaction: F=% .2f, p=% .2e\n", f_interaction_m3, p_interaction_m3))

## Interaction: F=2.43, p=1.19e-01

cat(sprintf(" Regulatory era (block): F=% .2f, p=% .2e\n", f_era_m3, p_era_m3))

## Regulatory era (block): F=88.64, p=4.97e-51

# 6. calculating estimated marginal means (cell means)
cell_means = analysis_data %>%
  group_by(therapeutic_area, review_type_simplified) %>%
  summarise(
    mean_review_time = mean(review_time_days_response, na.rm = TRUE),
    median_review_time = median(review_time_days_response, na.rm = TRUE),
    sd_review_time = sd(review_time_days_response, na.rm = TRUE),
    n = n(),
    .groups = "drop"
  )

cat("\nCell means (raw scale):\n")

## Cell means (raw scale):

print(cell_means)

```

```

## # A tibble: 4 x 6
##   therapeutic_area review_type_simplified mean_review_time median_review_time sd_review_time     n
##   <chr>           <chr>                      <dbl>            <dbl>            <dbl> <int>
## 1 Oncology        Priority                  300.             232.            282.  224
## 2 Oncology        Standard                 596.             365.            467.  75
## 3 Other           Priority                  477.             275.            445.  367
## 4 Other           Standard                 792.             638.            530.  372

# calculating main effects
main_effect_area = analysis_data %>%
  group_by(therapeutic_area) %>%
  summarise(mean_review_time = mean(review_time_days_response, na.rm = TRUE), .groups = "drop")

main_effect_review = analysis_data %>%
  group_by(review_type_simplified) %>%
  summarise(mean_review_time = mean(review_time_days_response, na.rm = TRUE), .groups = "drop")

cat("\nMain effect means:\n")

##
## Main effect means:

cat("Therapeutic area:\n")

## Therapeutic area:

print(main_effect_area)

## # A tibble: 2 x 2
##   therapeutic_area mean_review_time
##   <chr>           <dbl>
## 1 Oncology        374.
## 2 Other           635.

cat("\nReview designation:\n")

##
## Review designation:

print(main_effect_review)

## # A tibble: 2 x 2
##   review_type_simplified mean_review_time
##   <chr>                   <dbl>
## 1 Priority                410.
## 2 Standard                759.

# 6.5. calculating practical effect sizes with bootstrap confidence intervals
cat("\n=====\\n")

```

```

## =====
cat("PRACTICAL EFFECT SIZES (GEOMETRIC MEANS)\n")

## PRACTICAL EFFECT SIZES (GEOMETRIC MEANS)

cat("=====\\n\\n")

## =====

# calculating geometric means for Priority Review groups
oncology_priority_data = analysis_data %>%
  filter(therapeutic_area_factor == "Oncology", review_type_factor == "Priority") %>%
  pull(review_time_days_response)

other_priority_data = analysis_data %>%
  filter(therapeutic_area_factor == "Other", review_type_factor == "Priority") %>%
  pull(review_time_days_response)

# geometric mean calculation
geom_mean = function(x) {
  exp(mean(log(x), na.rm = TRUE))
}

oncology_priority_geom = geom_mean(oncology_priority_data)
other_priority_geom = geom_mean(other_priority_data)
ratio_point = oncology_priority_geom / other_priority_geom

cat(sprintf("Geometric mean review time (Priority Review):\n"))

## Geometric mean review time (Priority Review):

cat(sprintf("  Oncology: %.1f days\\n", oncology_priority_geom))

##  Oncology: 245.2 days

cat(sprintf("  Other: %.1f days\\n", other_priority_geom))

##  Other: 351.7 days

cat(sprintf("  Ratio (Oncology/Other): %.3f\\n\\n", ratio_point))

##  Ratio (Oncology/Other): 0.697

# bootstrap confidence interval for ratio
set.seed(RANDOM_SEED)
n_boot = 10000
boot_ratios = numeric(n_boot)

cat(sprintf("Calculating bootstrap 95% CI (%d iterations)...\\n", n_boot))

```

```

## Calculating bootstrap 95% CI (10000 iterations)...

for (i in 1:n_boot) {
  # resampling with replacement
  boot_onc = sample(oncology_priority_data, replace = TRUE)
  boot_oth = sample(other_priority_data, replace = TRUE)

  # calculating geometric means for bootstrap sample
  boot_geom_onc = geom_mean(boot_onc)
  boot_geom_oth = geom_mean(boot_oth)
  boot_ratios[i] = boot_geom_onc / boot_geom_oth
}

# percentile method confidence interval
ratio_ci_lower = quantile(boot_ratios, 0.025)
ratio_ci_upper = quantile(boot_ratios, 0.975)

cat("\nBootstrap 95% Confidence Interval for Ratio:\n")

## 
## Bootstrap 95% Confidence Interval for Ratio:

cat(sprintf(" Point estimate: %.3f\n", ratio_point))

## Point estimate: 0.697

cat(sprintf(" 95% CI: [% .3f, % .3f]\n\n", ratio_ci_lower, ratio_ci_upper))

## 95% CI: [0.628, 0.777]

# interpreting as percent faster/slower
pct_diff = (1 - ratio_point) * 100
pct_ci_lower = (1 - ratio_ci_upper) * 100
pct_ci_upper = (1 - ratio_ci_lower) * 100

cat("PRACTICAL INTERPRETATION:\n")

## PRACTICAL INTERPRETATION:

if (ratio_point < 1) {
  cat(sprintf("Oncology Priority drugs approved %.1f%% FASTER than Other Priority drugs\n",
             abs(pct_diff)))
  cat(sprintf(" [95% CI: %.1f%% to %.1f%% faster]\n\n",
             pct_ci_lower, pct_ci_upper))
} else {
  cat(sprintf("Oncology Priority drugs approved %.1f%% SLOWER than Other Priority drugs\n",
             pct_diff))
  cat(sprintf(" [95% CI: %.1f%% to %.1f%% slower]\n\n",
             abs(pct_ci_upper), abs(pct_ci_lower)))
}

```

```

## Oncology Priority drugs approved 30.3% FASTER than Other Priority drugs
## [95% CI: 22.3% to 37.2% faster]

# 7. permutation test for interaction effect
set.seed(RANDOM_SEED)

# using interaction F-stat from Model 2 (Type III SS)
observed_f_stat = f_interaction_m2

# creating permutation distribution
n_perms = N_PERMUTATIONS
perm_f_stats = numeric(n_perms)

cat(paste("\nRunning", n_perms, "permutations...\n"))

##
## Running 10000 permutations...

for (i in 1:n_perms) {
  # permuting therapeutic area labels
  perm_data = analysis_data
  perm_data$therapeutic_area_factor = sample(analysis_data$therapeutic_area_factor)

  # fitting permuted model
  perm_model = lm(
    review_time_days_response ~ therapeutic_area_factor * review_type_factor + regulatory_era_factor,
    data = perm_data
  )

  # extracting F-statistic for interaction using Type III SS
  perm_anova = car::Anova(perm_model, type = 3)
  perm_f_stats[i] = perm_anova["therapeutic_area_factor:review_type_factor", "F value"]

  # progress indicator every 2000 iterations
  if (i %% 2000 == 0) {
    cat(paste(" Completed", i, "permutations\n"))
  }
}

## Completed 2000 permutations
## Completed 4000 permutations
## Completed 6000 permutations
## Completed 8000 permutations
## Completed 10000 permutations

# calculating p-value
perm_p_value = mean(perm_f_stats >= observed_f_stat)

cat(paste("\nPermutation test results:\n"))

##
## Permutation test results:

```

```

cat(sprintf("  Observed F-statistic: %.2f\n", observed_f_stat))

##  Observed F-statistic: 0.32

cat(sprintf("  Permutation p-value: %.4f\n", perm_p_value))

##  Permutation p-value: 0.5945

cat(sprintf("  Parametric p-value: %.2e\n", p_interaction_m2))

##  Parametric p-value: 5.72e-01

# saving permutation results
perm_results = data.frame(
  permutation = 1:n_perms,
  f_statistic = perm_f_stats
)

perm_output_file = file.path(RESULTS_DIR, "permutation_test_results.csv")
save_csv(perm_results, perm_output_file)

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

# 8. visualization 1: fig1_interaction_plot.png
interaction_plot_data = cell_means %>%
  rename(Review_Type = review_type_simplified)

interaction_plot = ggplot(
  interaction_plot_data,
  aes(x = Review_Type, y = mean_review_time,
       color = therapeutic_area, group = therapeutic_area)
) +
  geom_line(linewidth = 1.5, alpha = 0.8) +
  geom_point(size = 5, alpha = 0.9) +
  geom_errorbar(
    aes(ymin = mean_review_time - sd_review_time / sqrt(n),
        ymax = mean_review_time + sd_review_time / sqrt(n)),
    width = 0.15,
    linewidth = 1.2,
    alpha = 0.7
  ) +
  labs(
    title = "Interaction Effect: Therapeutic Area x Review Type",
    subtitle = sprintf("F=% .2f, p<% .2e", f_interaction_m2, p_interaction_m2),
    x = "Review Designation",
    y = "Mean Review Time (Days)",
    color = "Therapeutic Area"
  ) +
  theme_minimal(base_size = 14) +
  theme(

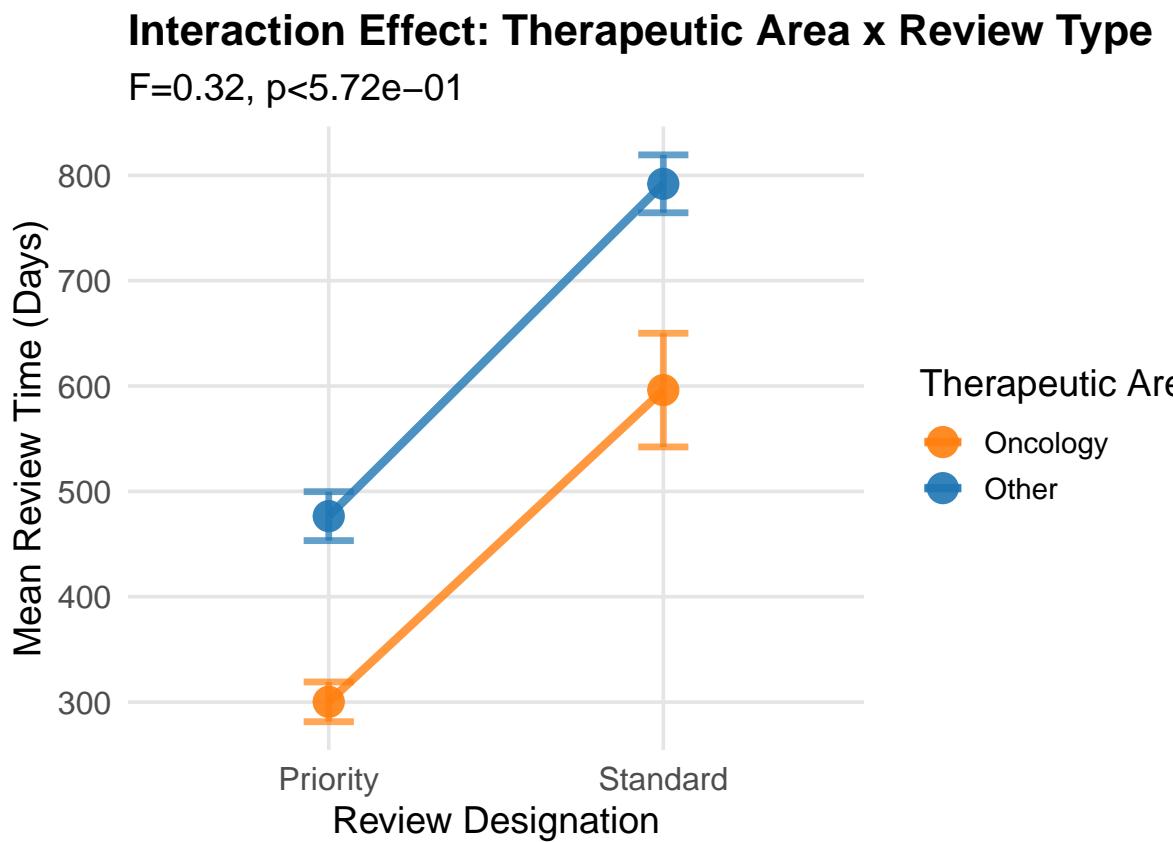
```

```

plot.title = element_text(face = "bold", size = 16),
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(),
legend.position = "right",
axis.text = element_text(size = 12)
) +
scale_color_manual(values = c("Other" = "#1f77b4", "Oncology" = "#ff7f0e"))

print(interaction_plot) # Display in R session

```



```

ggsave(
  file.path(FIGURES_DIR, "fig1_interaction_plot.png"),
  plot = interaction_plot,
  width = FIGURE_WIDTH,
  height = FIGURE_HEIGHT,
  dpi = DPI
)

cat("Saved: fig1_interaction_plot.png\n")

```

```
## Saved: fig1_interaction_plot.png
```

```

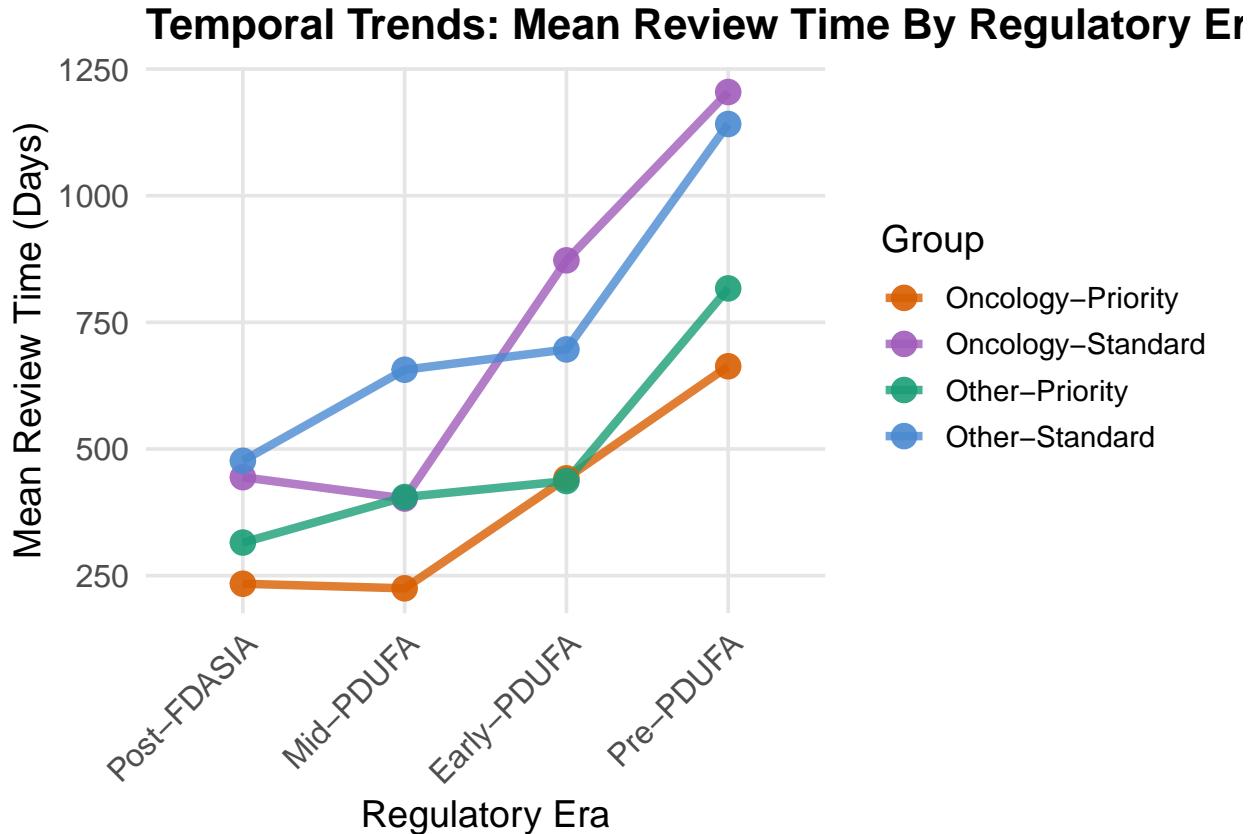
# 9. visualization 2: fig2_era_trends_plot.png
# CRITICAL FIX: Era ordering Post->Pre (modern first)
era_means = analysis_data %>%
  group_by(regulatory_era_factor, therapeutic_area, review_type_simplified) %>%
  summarise(mean_review_time = mean(review_time_days_response, na.rm = TRUE), .groups = "drop") %>%
  mutate(group = paste(therapeutic_area, review_type_simplified, sep = "-"))

# reorder factor levels for display: Post->Pre (modern first)
era_means$regulatory_era_factor = factor(
  era_means$regulatory_era_factor,
  levels = c("Post-FDASIA", "Mid-PDUFA", "Early-PDUFA", "Pre-PDUFA")
)

era_trends_plot = ggplot(
  era_means,
  aes(x = regulatory_era_factor, y = mean_review_time, color = group, group = group)
) +
  geom_line(linewidth = 1.5, alpha = 0.8) +
  geom_point(size = 4, alpha = 0.9) +
  labs(
    title = "Temporal Trends: Mean Review Time By Regulatory Era",
    x = "Regulatory Era",
    y = "Mean Review Time (Days)",
    color = "Group"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    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(),
    legend.position = "right",
    axis.text.x = element_text(angle = 45, hjust = 1, size = 12),
    axis.text.y = element_text(size = 12)
  ) +
  scale_color_manual(
    values = c(
      "Oncology-Priority" = "#d95f02",
      "Oncology-Standard" = "#a05dbb",
      "Other-Priority" = "#1b9e77",
      "Other-Standard" = "#4b8ad1"
    )
  )

print(era_trends_plot) # Display in R session

```



```

ggsave(
  file.path(FIGURES_DIR, "fig2_era_trends_plot.png"),
  plot = era_trends_plot,
  width = FIGURE_WIDTH,
  height = FIGURE_HEIGHT,
  dpi = DPI
)

cat("Saved: fig2_era_trends_plot.png\n")

## Saved: fig2_era_trends_plot.png

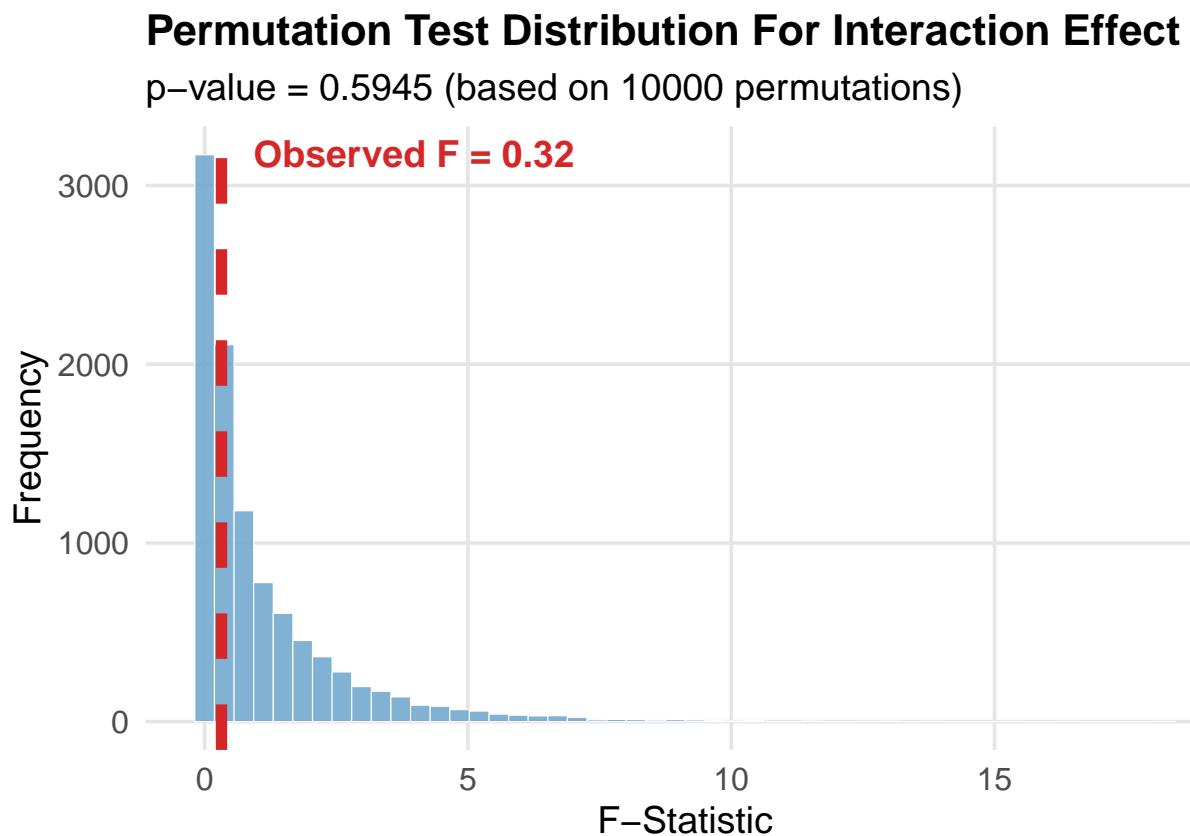
# 10. visualization 3: fig3_permutation_histograms.png
perm_hist = ggplot(perm_results, aes(x = f_statistic)) +
  geom_histogram(bins = 50, fill = "#74a9cf", color = "white", alpha = 0.9, linewidth = 0.2) +
  geom_vline(xintercept = observed_f_stat, color = "#d62728", linewidth = 2, linetype = "dashed") +
  annotate(
    "text",
    x = observed_f_stat,
    y = Inf,
    label = sprintf("Observed F = %.2f", observed_f_stat),
    vjust = 1.5,
    hjust = -0.1,
    color = "#d62728",
    size = 5,
  )
  
```

```

    fontface = "bold"
) +
labs(
  title = "Permutation Test Distribution For Interaction Effect",
  subtitle = sprintf("p-value = %.4f (based on %d permutations)", perm_p_value, n_perms),
  x = "F-Statistic",
  y = "Frequency"
) +
theme_minimal(base_size = 14) +
theme(
  plot.title = element_text(face = "bold", size = 16),
  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(),
  axis.text = element_text(size = 12)
)

print(perm_hist) # Display in R session

```



```

ggsave(
  file.path(FIGURES_DIR, "fig3_permutation_histograms.png"),
  plot = perm_hist,
  width = FIGURE_WIDTH,
  height = FIGURE_HEIGHT,

```

```

    dpi = DPI
)

cat("Saved: fig3_permutation_histograms.png\n")

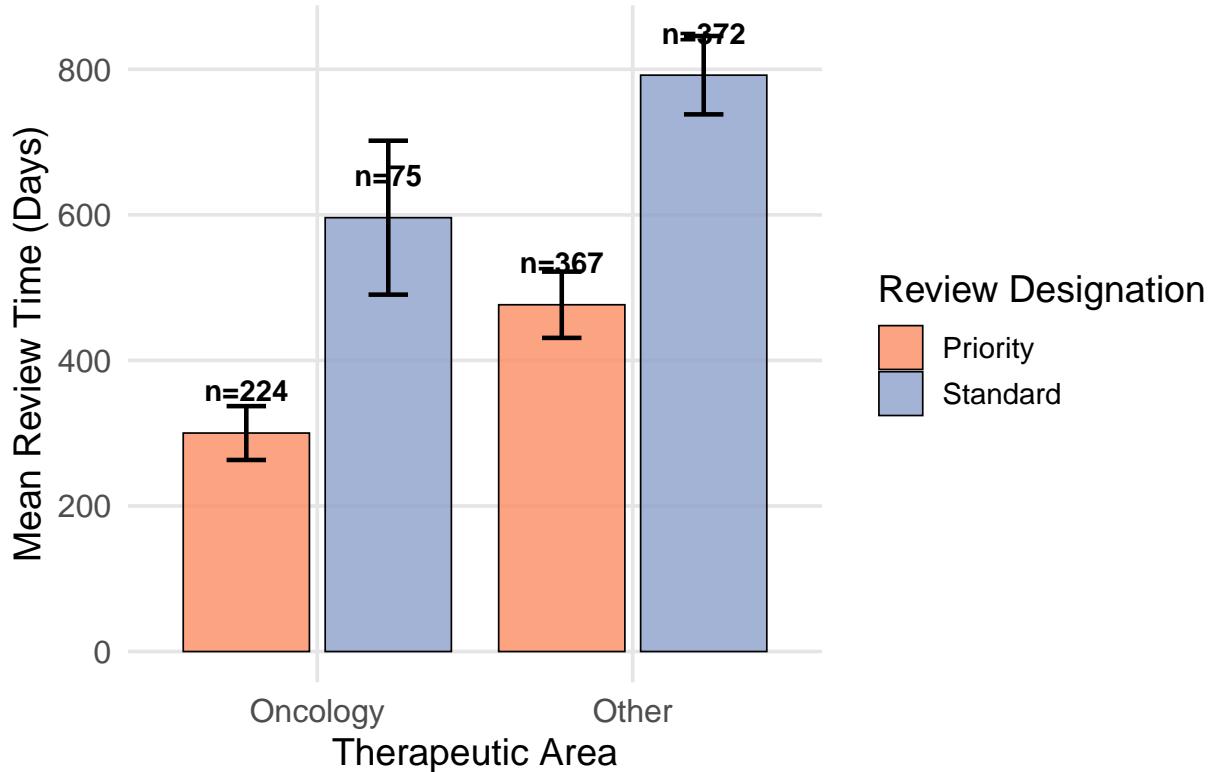
## Saved: fig3_permutation_histograms.png

# 11. visualization 4: fig4_cell_means_barchart.png
cell_means_plot = ggplot(
  interaction_plot_data,
  aes(x = therapeutic_area, y = mean_review_time, fill = Review_Type)
) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.9), width = 0.8,
           alpha = 0.8, color = "black", linewidth = 0.3) +
  geom_errorbar(
    aes(ymin = mean_review_time - 1.96 * sd_review_time / sqrt(n),
        ymax = mean_review_time + 1.96 * sd_review_time / sqrt(n)),
    position = position_dodge(width = 0.9),
    width = 0.25,
    linewidth = 0.8
  ) +
  geom_text(
    aes(label = sprintf("n=%d", n)),
    position = position_dodge(width = 0.9),
    vjust = -1.5,
    size = 4,
    fontface = "bold"
  ) +
  labs(
    title = "Cell Means With 95% Confidence Intervals",
    x = "Therapeutic Area",
    y = "Mean Review Time (Days)",
    fill = "Review Designation"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    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(),
    legend.position = "right",
    axis.text = element_text(size = 12)
  ) +
  scale_fill_manual(values = c("Standard" = "#8da0cb", "Priority" = "#fc8d62"))

print(cell_means_plot) # Display in R session

```

Cell Means With 95% Confidence Intervals



```

ggsave(
  file.path(FIGURES_DIR, "fig4_cell_means_barchart.png"),
  plot = cell_means_plot,
  width = FIGURE_WIDTH,
  height = FIGURE_HEIGHT,
  dpi = DPI
)

cat("Saved: fig4_cell_means_barchart.png\n")

## Saved: fig4_cell_means_barchart.png

# 12. visualization 5: fig5_effect_sizes_barchart.png
effect_sizes_data = data.frame(
  effect = c("Therapeutic Area", "Review Type", "Interaction"),
  eta_squared = c(eta_sq_area, eta_sq_review, eta_sq_interaction)
)

effect_sizes_plot = ggplot(
  effect_sizes_data,
  aes(x = reorder(effect, -eta_squared), y = eta_squared, fill = effect)
) +
  geom_bar(stat = "identity", width = 0.6, alpha = 0.8, color = "black", linewidth = 0.3) +
  geom_text(
    aes(label = sprintf("%.4f", eta_squared)),
    vjust = 1
)
  
```

```

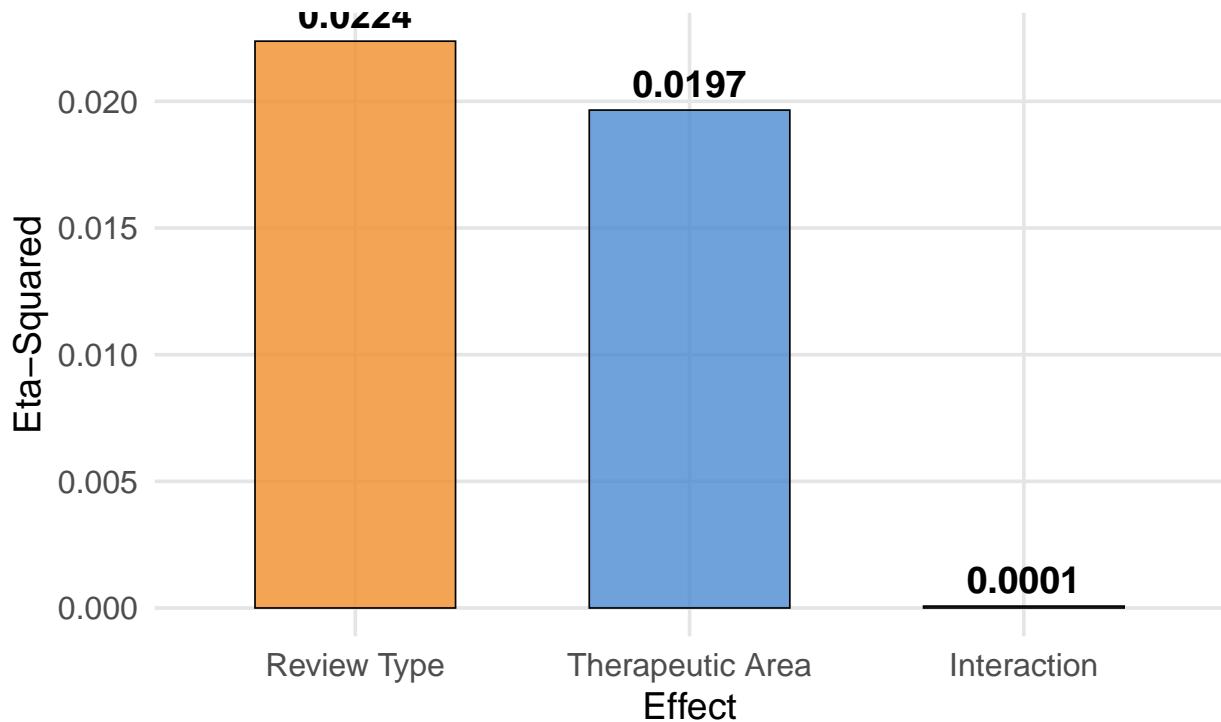
    vjust = -0.5,
    size = 5,
    fontface = "bold"
) +
labs(
  title = "Effect Sizes (Eta-Squared)",
  subtitle = "Model 1: Raw Scale, No Blocking",
  x = "Effect",
  y = "Eta-Squared"
) +
theme_minimal(base_size = 14) +
theme(
  plot.title = element_text(face = "bold", size = 16),
  plot.background = 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(),
  legend.position = "none",
  axis.text = element_text(size = 12)
) +
scale_fill_manual(
  values = c(
    "Therapeutic Area" = "#4b8ad1",
    "Review Type" = "#f28e2b",
    "Interaction" = "#59a14f"
  )
)

print(effect_sizes_plot) # Display in R session

```

Effect Sizes (Eta-Squared)

Model 1: Raw Scale, No Blocking



```
ggsave(  
  file.path(FIGURES_DIR, "fig5_effect_sizes_barchart.png"),  
  plot = effect_sizes_plot,  
  width = FIGURE_WIDTH,  
  height = FIGURE_HEIGHT,  
  dpi = DPI  
)  
  
cat("Saved: fig5_effect_sizes_barchart.png\n")
```

```
## Saved: fig5_effect_sizes_barchart.png
```

```
# additional visualization 5 (alt): fig5_effect_sizes_barchart_alt.png (blocked log-scale ^, horizontal  
effect_sizes_m3 = data.frame(  
  factor = c("Regulatory Era (Block)", "Review Type", "Therapeutic Area", "Area × Review Interaction"),  
  eta_sq = c(eta_sq_era_m3, eta_sq_review_m3, eta_sq_area_m3, eta_sq_interaction_m3)  
) %>%  
  mutate(  
    eta_pct = eta_sq * 100,  
    magnitude = case_when(  
      eta_sq >= 0.14 ~ "large",  
      eta_sq >= 0.06 ~ "medium",  
      eta_sq >= 0.01 ~ "small",  
      TRUE ~ "negligible"  
)
```

```

magnitude = factor(magnitude, levels = c("large", "medium", "small", "negligible"))
)

magnitude_palette = c(
  "large" = "#3b5c7a",
  "medium" = "#587b9d",
  "small" = "#9ecae1",
  "negligible" = "#cbd5e8"
)

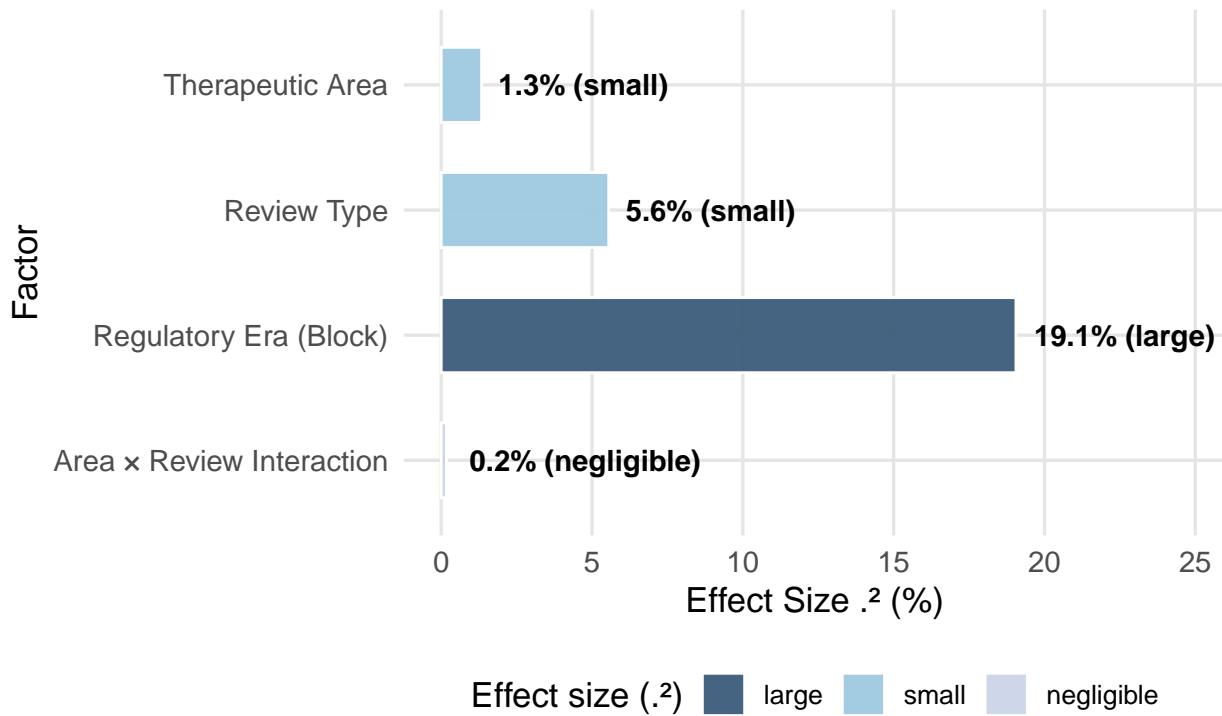
effect_sizes_plot_alt = ggplot(effect_sizes_m3, aes(x = eta_pct, y = factor, fill = magnitude)) +
  geom_col(width = 0.6, alpha = 0.95, color = "white") +
  geom_text(
    aes(label = sprintf("%.1f%% (%s)", eta_pct, magnitude)),
    hjust = -0.1,
    size = 4,
    fontface = "bold"
  ) +
  scale_fill_manual(values = magnitude_palette, name = "Effect size (²)") +
  labs(
    title = "Effect Sizes: Blocked ANOVA (Log Scale)",
    subtitle = "Variance explained (²) by factor",
    x = "Effect Size ² (%)",
    y = "Factor"
  ) +
  coord_cartesian(xlim = c(0, max(effect_sizes_m3$eta_pct) * 1.3)) +
  theme_minimal(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold", size = 15),
    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(),
    legend.position = "bottom",
    axis.text = element_text(size = 11)
  )

print(effect_sizes_plot_alt)

```

Effect Sizes: Blocked ANOVA (Log Scale)

Variance explained (\cdot^2) by factor



```
ggsave(  
  file.path(FIGURES_DIR, "fig5_effect_sizes_barchart_alt.png"),  
  plot = effect_sizes_plot_alt,  
  width = FIGURE_WIDTH,  
  height = FIGURE_HEIGHT,  
  dpi = DPI  
)
```

```
# 13. visualization 6: fig6_residual_diagnostics.png
```

```
residuals_data = data.frame(  
  fitted = fitted(model3),  
  residuals = residuals(model3),  
  standardized_residuals = rstandard(model3)  
)
```

```
png(  
  file.path(FIGURES_DIR, "fig6_residual_diagnostics.png"),  
  width = FIGURE_WIDTH * DPI,  
  height = FIGURE_HEIGHT * DPI,  
  res = DPI  
)
```

```
par(mfrow = c(2, 2))
```

```
# plot 1: residuals vs fitted  
plot(
```

```

residuals_data$fitted,
residuals_data$residuals,
main = "Residuals vs Fitted",
xlab = "Fitted Values",
ylab = "Residuals",
pch = 16,
col = rgb(0, 0, 0, 0.3)
)
abline(h = 0, col = "red", lwd = 2, lty = 2)

# plot 2: QQ plot
qqnorm(residuals_data$standardized_residuals, main = "Normal Q-Q Plot", pch = 16, col = rgb(0, 0, 0, 0.3))
qqline(residuals_data$standardized_residuals, col = "red", lwd = 2)

# plot 3: scale-location
plot(
  residuals_data$fitted,
  sqrt(abs(residuals_data$standardized_residuals)),
  main = "Scale-Location",
  xlab = "Fitted Values",
  ylab = expression(sqrt(" |Standardized Residuals| ")),
  pch = 16,
  col = rgb(0, 0, 0, 0.3)
)

# plot 4: residuals histogram
hist(
  residuals_data$residuals,
  breaks = 30,
  main = "Histogram of Residuals",
  xlab = "Residuals",
  col = "skyblue",
  border = "black"
)

dev.off()

## pdf
## 2

cat("Saved: fig6_residual_diagnostics.png\n")

## Saved: fig6_residual_diagnostics.png

# additional residual diagnostics (alt) with stats and density overlay (log scale)
residuals_log = residuals(model3)
shapiro_log = shapiro.test(residuals_log)
w_stat_log = shapiro_log$statistic
p_value_log = shapiro_log$p.value
mean_log = mean(residuals_log)
sd_log = sd(residuals_log)
n_log = length(residuals_log)

```

```

qq_log_alt = ggplot(data.frame(residuals_log = residuals_log), aes(sample = residuals_log)) +
  stat_qq(color = "#1f77b4", size = 1.4, alpha = 0.85) +
  stat_qq_line(color = "#d62728", linewidth = 1.2) +
  annotate(
    "label",
    x = -2.5,
    y = 2.5,
    label = sprintf("Shapiro-Wilk:\nW = %.4f\np = %.2e", w_stat_log, p_value_log),
    fill = "white",
    color = "black",
    label.size = 0.3
  ) +
  labs(
    title = "QQ-Plot: Log Scale Residuals",
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  theme_minimal(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold", size = 15),
    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()
  )

hist_log_alt = ggplot(data.frame(residuals_log = residuals_log), aes(x = residuals_log)) +
  geom_histogram(aes(y = after_stat(density)),
                 bins = 60,
                 fill = "#6c8ebf",
                 color = "white",
                 alpha = 0.9) +
  stat_function(fun = dnorm,
                args = list(mean = mean_log, sd = sd_log),
                color = "#d62728",
                linewidth = 1.2) +
  annotate(
    "label",
    x = min(residuals_log),
    y = max(density(residuals_log)$y),
    hjust = 0,
    label = sprintf("Mean = %.4f\nSD = %.4f\nn = %d", mean_log, sd_log, n_log),
    fill = "white",
    color = "black",
    label.size = 0.3
  ) +
  labs(
    title = "Histogram: Log Scale Residuals",
    x = "Residuals (log scale)",
    y = "Density"
  ) +
  theme_minimal(base_size = 13) +
  theme(

```

```

    plot.title = element_text(face = "bold", size = 15),
    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()
  )

resid_alt = qq_log_alt + hist_log_alt + plot_layout(ncol = 2)

ggsave(
  file.path(FIGURES_DIR, "fig6_residual_diagnostics_alt.png"),
  plot = resid_alt,
  width = FIGURE_WIDTH,
  height = FIGURE_HEIGHT,
  dpi = DPI
)

# 14. saving model comparison results
model_comparison = data.frame(
  model = c("Model 1", "Model 1", "Model 1",
            "Model 2", "Model 2", "Model 2", "Model 2",
            "Model 3", "Model 3", "Model 3", "Model 3"),
  effect = c("Therapeutic Area", "Review Type", "Interaction",
             "Therapeutic Area", "Review Type", "Interaction", "Regulatory Era",
             "Therapeutic Area", "Review Type", "Interaction", "Regulatory Era"),
  f_statistic = c(f_area_m1, f_review_m1, f_interaction_m1,
                  f_area_m2, f_review_m2, f_interaction_m2, f_era_m2,
                  f_area_m3, f_review_m3, f_interaction_m3, f_era_m3),
  p_value = c(p_area_m1, p_review_m1, p_interaction_m1,
              p_area_m2, p_review_m2, p_interaction_m2, p_era_m2,
              p_area_m3, p_review_m3, p_interaction_m3, p_era_m3)
)
comparison_output_file = file.path(RESULTS_DIR, "model_comparison.csv")
save_csv(model_comparison, comparison_output_file)

```

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

```
# saving cell means
cell_means_output_file = file.path(RESULTS_DIR, "cell_means.csv")
save_csv(cell_means, cell_means_output_file)
```

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

```
cat("\nPhase 6 complete\n")
```

```
##
## Phase 6 complete
```