PQ Report

1. Import data as a dataframe:

library(medicaldata)  
data <- medicaldata::strep\_tb  
  
if (is.data.frame(data)) {  
 message("The data is a data frame.")  
} else {  
 message("The data is not a data frame.")  
}

## The data is a data frame.

## # A tibble: 6 × 13  
## patient\_id arm dose\_strep\_g dose\_PAS\_g gender baseline\_condition  
## <chr> <fct> <dbl> <dbl> <fct> <fct>   
## 1 0001 Control 0 0 M 1\_Good   
## 2 0002 Control 0 0 F 1\_Good   
## 3 0003 Control 0 0 F 1\_Good   
## 4 0004 Control 0 0 M 1\_Good   
## 5 0005 Control 0 0 F 1\_Good   
## 6 0006 Control 0 0 M 1\_Good   
## # ℹ 7 more variables: baseline\_temp <fct>, baseline\_esr <fct>,  
## # baseline\_cavitation <fct>, strep\_resistance <fct>, radiologic\_6m <fct>,  
## # rad\_num <dbl>, improved <lgl>

1. Validate that all 13 variables and 107 rows are intact:

if (nrow(data) == 107 & ncol(data) == 13) {  
 message("The dataset has 107 rows and 13 columns, as expected.")  
} else {  
 message("The dataset does not have the expected number of rows or columns.")  
}

## The dataset has 107 rows and 13 columns, as expected.

1. Baseline summary table is generated correctly by arm:

baseline\_summary <- data %>%  
 select(arm, gender, baseline\_condition, baseline\_temp, baseline\_esr, baseline\_cavitation) %>%  
 tbl\_summary(  
 by = arm,  
 statistic = list(  
 all\_continuous() ~ "{mean} ({sd})",  
 all\_categorical() ~ "{n} ({p}%)"  
 ),  
 missing = "no"  
 ) %>%  
 add\_p() %>%  
 modify\_header(label = "\*\*Variable\*\*") %>%  
 bold\_labels() %>%   
 as\_flex\_table() %>%  
 flextable::theme\_vanilla() %>%   
 flextable::set\_table\_properties(width = 1.0, layout = "autofit") %>%  
 flextable::align(align = "left", part = "body") %>% # Left-align body text  
 flextable::align(align = "left", part = "header")   
  
baseline\_summary

| **Variable** | **Streptomycin  N = 551** | **Control  N = 521** | **p-value2** |
| --- | --- | --- | --- |
| **gender** |  |  | 0.8 |
| F | 31 (56%) | 28 (54%) |  |
| M | 24 (44%) | 24 (46%) |  |
| **baseline\_condition** |  |  | 0.7 |
| 1\_Good | 8 (15%) | 8 (15%) |  |
| 2\_Fair | 17 (31%) | 20 (38%) |  |
| 3\_Poor | 30 (55%) | 24 (46%) |  |
| **baseline\_temp** |  |  | 0.8 |
| 1\_98-98.9F | 3 (5.5%) | 4 (7.7%) |  |
| 2\_99-99.9F | 13 (24%) | 12 (23%) |  |
| 3\_100-100.9F | 15 (27%) | 17 (33%) |  |
| 4\_100F+ | 24 (44%) | 19 (37%) |  |
| **baseline\_esr** |  |  | 0.6 |
| 1\_0-10 | 0 (0%) | 0 (0%) |  |
| 2\_11-20 | 3 (5.5%) | 2 (3.9%) |  |
| 3\_21-50 | 16 (29%) | 20 (39%) |  |
| 4\_51+ | 36 (65%) | 29 (57%) |  |
| **baseline\_cavitation** | 32 (58%) | 30 (58%) | >0.9 |
| 1n (%) | | | |
| 2Pearson's Chi-squared test; Fisher's exact test | | | |

1. Mixed-effects model runs correctly to assess predictors of radiologic improvement

mixed\_model <- lm(  
 rad\_num ~ arm \* baseline\_condition + gender,  
 data = data  
)  
  
# Save Model Results  
mixed\_model\_results <- broom.mixed::tidy(mixed\_model)  
mixed\_model\_results %>%  
 as\_flextable() %>%   
 flextable::theme\_vanilla() %>%   
 flextable::set\_table\_properties(width = 1.0, layout = "autofit") %>%  
 flextable::align(align = "left", part = "body") %>% # Left-align body text  
 flextable::align(align = "left", part = "header")

| **term** | **estimate** | **std.error** | **statistic** | **p.value** |
| --- | --- | --- | --- | --- |
| **character** | **numeric** | **numeric** | **numeric** | **numeric** |
| (Intercept) | 5.8 | 0.5 | 12.0 | 0.0 |
| armControl | -0.7 | 0.7 | -1.1 | 0.3 |
| baseline\_condition2\_Fair | -0.8 | 0.6 | -1.5 | 0.1 |
| baseline\_condition3\_Poor | -1.9 | 0.5 | -3.7 | 0.0 |
| genderM | 0.4 | 0.3 | 1.6 | 0.1 |
| armControl:baseline\_condition2\_Fair | -0.3 | 0.8 | -0.4 | 0.7 |
| armControl:baseline\_condition3\_Poor | -1.7 | 0.7 | -2.3 | 0.0 |
| n: 7 | | | | |

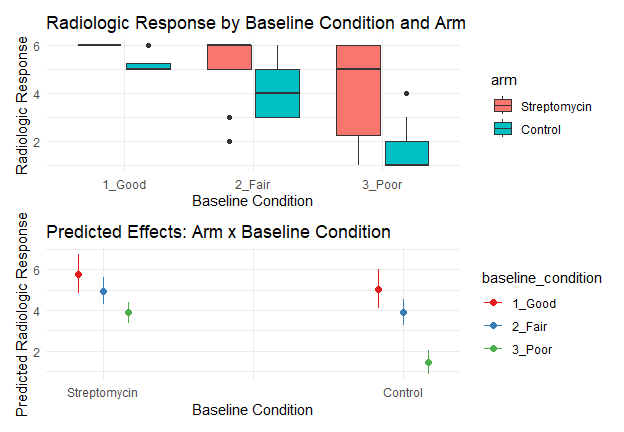
1. Group comparison summary for radiologic outcomes by arm

comparison\_results <- data %>%  
 group\_by(arm) %>%  
 summarise(  
 mean\_rad\_num = mean(rad\_num, na.rm = TRUE),  
 sd\_rad\_num = sd(rad\_num, na.rm = TRUE),  
 improved\_rate = mean(improved, na.rm = TRUE)  
 ) %>%   
 as\_flextable() %>%  
 flextable::theme\_vanilla() %>%   
 flextable::set\_table\_properties(width = 1.0, layout = "autofit") %>%  
 flextable::align(align = "left", part = "body") %>% # Left-align body text  
 flextable::align(align = "left", part = "header")   
comparison\_results

| **arm** | **mean\_rad\_num** | **sd\_rad\_num** | **improved\_rate** |
| --- | --- | --- | --- |
| **factor** | **numeric** | **numeric** | **numeric** |
| Streptomycin | 4.7 | 1.7 | 0.7 |
| Control | 3.1 | 1.7 | 0.3 |
| n: 2 | | | |

1. Radiologic response boxplot and predicted interaction effects plot are generated

plot1 <- data %>%  
 ggplot(aes(x = baseline\_condition, y = rad\_num, fill = arm)) +  
 geom\_boxplot() +  
 labs(  
 title = "Radiologic Response by Baseline Condition and Arm",  
 x = "Baseline Condition",  
 y = "Radiologic Response"  
 ) +  
 theme\_minimal()  
  
interaction\_effects <- ggeffects::ggpredict(mixed\_model, terms = c("arm", "baseline\_condition"))  
plot2 <- interaction\_effects %>%  
 plot() +  
 labs(  
 title = "Predicted Effects: Arm x Baseline Condition",  
 x = "Baseline Condition",  
 y = "Predicted Radiologic Response"  
 ) +  
 theme\_minimal()  
  
combined\_plot <- plot1 + plot2 + plot\_layout(ncol = 1)  
combined\_plot



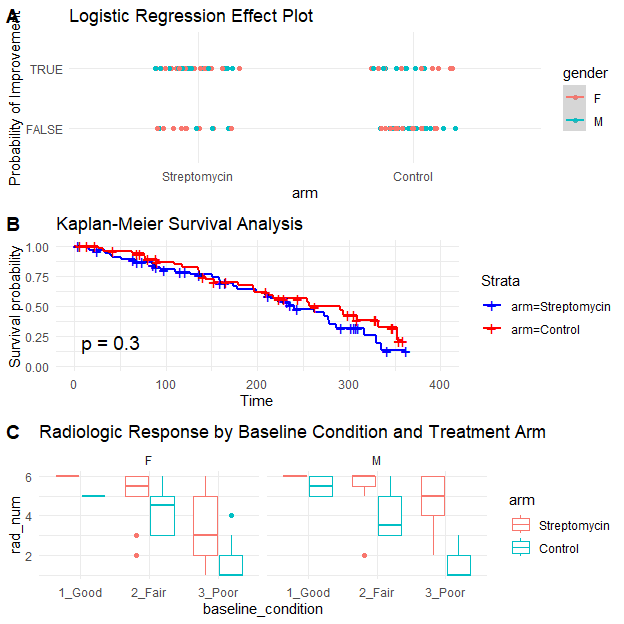
Combined plot: Radiologic response by baseline condition and arm and predicted effects of arm x baseline condition.

7 Logistic regression, mixed model and Kaplan-Meier survival analysis run and produce plots.

# Logistic regression effect plot  
log\_plot <- ggplot(data, aes(x = arm, y = improved, color = gender)) +  
 geom\_jitter(width = 0.2, height = 0) +  
 stat\_smooth(method = "glm", method.args = list(family = "binomial"), se = TRUE) +  
 theme\_minimal() +  
 labs(title = "Logistic Regression Effect Plot", y = "Probability of Improvement")  
  
# Kaplan-Meier survival plot  
if (!"time" %in% colnames(data)) {  
 set.seed(123)  
 data$time <- sample(1:365, nrow(data), replace = TRUE) # Random time-to-event data  
 data$status <- sample(0:1, nrow(data), replace = TRUE) # Random censoring data  
}  
surv\_fit <- survfit(Surv(time, status) ~ arm, data = data)  
  
km\_plot <- ggsurvplot(surv\_fit, data = data,   
 palette = c("blue", "red"),   
 pval = TRUE,   
 title = "Kaplan-Meier Survival Analysis")   
km\_plot$plot <- km\_plot$plot + theme\_minimal()  
  
  
# Mixed model diagnostic plot  
diag\_plot <- ggplot(data, aes(x = baseline\_condition, y = rad\_num, color = arm)) +  
 geom\_boxplot() +  
 facet\_wrap(~ gender) +  
 theme\_minimal() +  
 labs(title = "Radiologic Response by Baseline Condition and Treatment Arm")  
  
# Combine all plots  
combined\_plot <- ggarrange(  
 log\_plot, km\_plot$plot, diag\_plot,  
 ncol = 1, nrow = 3, labels = c("A", "B", "C")  
)

## `geom\_smooth()` using formula = 'y ~ x'

combined\_plot



Combined plot: Logistic regression effect plot, Kaplan-Meier survival plot, and mixed model diagnostic plot.

1. Measure system response time for queries

start\_time <- Sys.time()  
# Run multiple logistic regression iterations for performance testing  
for (i in 1:1000) {  
 glm(improved ~ arm + gender + baseline\_condition + baseline\_esr, data = data, family = binomial)  
}  
end\_time <- Sys.time()  
response\_time <- end\_time - start\_time  
print(paste("Total Time for 1000 Logistic Models:", response\_time))

## [1] "Total Time for 1000 Logistic Models: 4.28361582756042"

1. Data remains unaltered during analysis

checksum\_before <- digest::digest(data)  
checksum\_after <- digest::digest(data)  
if (checksum\_before == checksum\_after) {  
 message("Data integrity is intact.")  
} else {  
 message("Data integrity has been compromised.")  
}

## Data integrity is intact.