|  |  |
| --- | --- |
| Title | **A two-part, Phase 1b/2a study to investigate the safety and efficacy of topical Medi‑Solfen for wound analgesia and antisepsis, when used prior to Standard of Care (SOC) (including suture repair) of small to medium, simple skin lacerations in adults** |
| Author: | Anna Zaremba  Statistical Programmer |
|  |  |
| Reviewed: | Marius Ardelean  Senior Statistician |
|  |  |
| Approved: | Oleksandr Savuliak  Junior Statistician |

1. Import demographic and laboratory data as dataframes:

url <- "https://raw.githubusercontent.com/RConsortium/submissions-pilot3-adam-to-fda/main/m5/datasets/rconsortiumpilot3/tabulations/sdtm/"  
  
# Read the DM data from public github  
dm\_url <- glue::glue(url, "dm.xpt")  
dm\_data <- haven::read\_xpt(readr::read\_file\_raw(dm\_url))  
  
dm\_data <- dm\_data %>%   
 dplyr::mutate(COUNTRY = factor(COUNTRY, levels = c("USA", "NOT USA"))) %>%  
 dplyr::arrange(USUBJID)  
  
  
# Read the LB data from public github  
lb\_url <- glue::glue(url, "lb.xpt")  
lb\_data <- haven::read\_xpt(readr::read\_file\_raw(lb\_url))  
  
lb\_data <- lb\_data %>%   
 arrange(USUBJID, LBSEQ)  
  
rm(url, dm\_url, lb\_url)  
  
if (is.data.frame(dm\_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 × 25  
## STUDYID DOMAIN USUBJID SUBJID RFSTDTC RFENDTC RFXSTDTC RFXENDTC RFICDTC  
## <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>   
## 1 CDISCPILOT01 DM 01-701-1… 1015 2014-0… 2014-0… 2014-01… 2014-07… ""   
## 2 CDISCPILOT01 DM 01-701-1… 1023 2012-0… 2012-0… 2012-08… 2012-09… ""   
## 3 CDISCPILOT01 DM 01-701-1… 1028 2013-0… 2014-0… 2013-07… 2014-01… ""   
## 4 CDISCPILOT01 DM 01-701-1… 1033 2014-0… 2014-0… 2014-03… 2014-03… ""   
## 5 CDISCPILOT01 DM 01-701-1… 1034 2014-0… 2014-1… 2014-07… 2014-12… ""   
## 6 CDISCPILOT01 DM 01-701-1… 1047 2013-0… 2013-0… 2013-02… 2013-03… ""   
## # ℹ 16 more variables: RFPENDTC <chr>, DTHDTC <chr>, DTHFL <chr>, SITEID <chr>,  
## # AGE <dbl>, AGEU <chr>, SEX <chr>, RACE <chr>, ETHNIC <chr>, ARMCD <chr>,  
## # ARM <chr>, ACTARMCD <chr>, ACTARM <chr>, COUNTRY <fct>, DMDTC <chr>,  
## # DMDY <dbl>

if (is.data.frame(lb\_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 × 23  
## STUDYID DOMAIN USUBJID LBSEQ LBTESTCD LBTEST LBCAT LBORRES LBORRESU LBORNRLO  
## <chr> <chr> <chr> <dbl> <chr> <chr> <chr> <chr> <chr> <chr>   
## 1 CDISCPIL… LB 01-701… 1 ALB Album… CHEM… 3.8 g/dL "3.3"   
## 2 CDISCPIL… LB 01-701… 2 ALP Alkal… CHEM… 34 U/L "35"   
## 3 CDISCPIL… LB 01-701… 3 ALT Alani… CHEM… 27 U/L "6"   
## 4 CDISCPIL… LB 01-701… 4 ANISO Aniso… HEMA… 1 NO UNITS ""   
## 5 CDISCPIL… LB 01-701… 5 AST Aspar… CHEM… 40 U/L "9"   
## 6 CDISCPIL… LB 01-701… 6 BASO Basop… HEMA… 0.05 THOU/uL "0"   
## # ℹ 13 more variables: LBORNRHI <chr>, LBSTRESC <chr>, LBSTRESN <dbl>,  
## # LBSTRESU <chr>, LBSTNRLO <dbl>, LBSTNRHI <dbl>, LBNRIND <chr>,  
## # LBBLFL <chr>, VISITNUM <dbl>, VISIT <chr>, VISITDY <dbl>, LBDTC <chr>,  
## # LBDY <dbl>

1. Data integrity checks

2.1. Validate that all 25 variables and 306 rows are intact for demographic data:

if (nrow(dm\_data) == 306 & ncol(dm\_data) == 25) {  
 message("OK: The dataset has 306 rows and 25 columns, as expected.")  
} else {  
 message("ERROR: The dataset does not have the expected number of rows or columns.")  
}

## OK: The dataset has 306 rows and 25 columns, as expected.

2.2. Validate that all 23 variables and 59580 rows are intact for laboratory data:

if (nrow(lb\_data) == 59580 & ncol(lb\_data) == 23) {  
 message("OK: The dataset has 59580 rows and 23 columns, as expected.")  
} else {  
 message("ERROR: The dataset does not have the expected number of rows or columns.")  
}

## OK: The dataset has 59580 rows and 23 columns, as expected.

2.3. Validate that the join between demographic and laboratory data is successful:

# Merge demographic and laboratory data  
data <- lb\_data %>% select(-DOMAIN) %>%  
 left\_join(dm\_data %>% select(-DOMAIN, -SUBJID), by = c("USUBJID", "STUDYID")) %>%  
 dplyr::mutate(ARM = factor(ARM)) %>%  
 dplyr::select(!starts\_with("LB"), starts\_with("LB"))  
  
# check if join is correct  
if (nrow(data) == nrow(lb\_data) & ncol(data) == ncol(dm\_data) + ncol(lb\_data) - 5) {  
 message("The join was successful.")  
} else {  
 message("The join was unsuccessful.")  
}

## The join was successful.

1. Data cleaning and summaries

3.1. Data cleaning

# Define allowed visit levels  
visits <- unique(data$VISIT)  
ordered\_visits <- visits[grepl("SCREENING|WEEK", visits)]  
  
# Store original labels before modification  
labels <- lapply(data, attr, "label")  
  
# Data filtering and cleaning  
data\_clean <- data %>%  
 select(USUBJID, ARM, VISIT, AGE, SEX, LBORRES, LBTEST) %>%  
 filter(LBTEST == "Albumin", VISIT %in% ordered\_visits) %>%  
 mutate(  
 LBORRES = as.numeric(LBORRES),  
 VISIT = factor(VISIT, levels = unique(c(ordered\_visits, VISIT))),  
 ARM = factor(ARM, levels = unique(ARM))  
 )  
  
# Restore labels  
for (col in names(labels)) {  
 if (!is.null(labels[[col]]) && col %in% names(data\_clean)) {  
 attr(data\_clean[[col]], "label") <- labels[[col]]  
 }  
 }

3.2. Baseline summary table is generated correctly by arm:

baseline\_summary <- data %>%  
 dplyr::distinct(USUBJID, ARM, SEX, RACE, ETHNIC, COUNTRY) %>%   
 dplyr::mutate(RACE = fct\_infreq(RACE),  
 ETHNIC = fct\_infreq(ETHNIC)) %>%  
 gtsummary::tbl\_summary(  
 by = ARM,  
 include = -USUBJID,  
 digits = list(  
 all\_continuous() ~ c(1, 1, 1),  
 all\_categorical() ~ c(0,1 )),  
 statistic = list(  
 all\_continuous() ~ "{mean} ({sd})",  
 all\_categorical() ~ "{n} ({p}%)"  
 ),  
 missing = "no"  
 ) %>%  
 gtsummary::add\_p(pvalue\_fun = label\_style\_pvalue(digits = 3)) %>%  
 gtsummary::add\_overall() %>%  
 gtsummary::modify\_header(label = "\*\*Variable\*\*") %>%  
 gtsummary::bold\_labels() %>%   
 gtsummary::as\_flex\_table() %>%  
 flextable::theme\_vanilla() %>%   
 flextable::set\_table\_properties(width = 1.0, layout = "autofit") %>%  
 flextable::align(align = "left", part = "body") %>%   
 flextable::align(align = "left", part = "header")   
  
baseline\_summary

| **Variable** | **Overall  N = 2541** | **Placebo  N = 861** | **Xanomeline High Dose  N = 841** | **Xanomeline Low Dose  N = 841** | **p-value2** |
| --- | --- | --- | --- | --- | --- |
| **Sex** |  |  |  |  | 0.141 |
| F | 143 (56.3%) | 53 (61.6%) | 40 (47.6%) | 50 (59.5%) |  |
| M | 111 (43.7%) | 33 (38.4%) | 44 (52.4%) | 34 (40.5%) |  |
| **RACE** |  |  |  |  | 0.680 |
| WHITE | 230 (90.6%) | 78 (90.7%) | 74 (88.1%) | 78 (92.9%) |  |
| BLACK OR AFRICAN AMERICAN | 23 (9.1%) | 8 (9.3%) | 9 (10.7%) | 6 (7.1%) |  |
| AMERICAN INDIAN OR ALASKA NATIVE | 1 (0.4%) | 0 (0.0%) | 1 (1.2%) | 0 (0.0%) |  |
| **ETHNIC** |  |  |  |  | 0.570 |
| NOT HISPANIC OR LATINO | 242 (95.3%) | 83 (96.5%) | 81 (96.4%) | 78 (92.9%) |  |
| HISPANIC OR LATINO | 12 (4.7%) | 3 (3.5%) | 3 (3.6%) | 6 (7.1%) |  |
| **COUNTRY** |  |  |  |  | >0.999 |
| USA | 254 (100.0%) | 86 (100.0%) | 84 (100.0%) | 84 (100.0%) |  |
| NOT USA | 0 (0.0%) | 0 (0.0%) | 0 (0.0%) | 0 (0.0%) |  |
| 1n (%) | | | | | |
| 2Pearson's Chi-squared test; Fisher's exact test | | | | | |

3.3. Summaries by ARM and VISIT:

ALB\_summary <- data\_clean %>%   
 dplyr::group\_by(ARM, VISIT) %>%  
 dplyr::summarise(n = n(),  
 mean = round(mean(LBORRES, na.rm = TRUE), 2),  
 sd = round(sd(LBORRES, na.rm = TRUE), 2),  
 median = median(LBORRES, na.rm = TRUE),  
 min = min(LBORRES, na.rm = TRUE),  
 max = max(LBORRES, na.rm = TRUE),  
 .groups = "drop") %>%  
 flextable::flextable() %>%  
 flextable::theme\_booktabs() %>%  
 flextable::autofit()  
  
ALB\_summary

| ARM | Visit Name | n | mean | sd | median | min | max |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Placebo | SCREENING 1 | 86 | 3.98 | 0.28 | 4.00 | 3.2 | 4.6 |
| Placebo | WEEK 2 | 83 | 3.89 | 0.31 | 3.90 | 3.1 | 4.6 |
| Placebo | WEEK 4 | 79 | 3.88 | 0.33 | 3.90 | 2.9 | 4.7 |
| Placebo | WEEK 6 | 73 | 3.91 | 0.26 | 3.90 | 3.4 | 4.5 |
| Placebo | WEEK 8 | 72 | 3.98 | 0.35 | 4.00 | 2.8 | 4.9 |
| Placebo | WEEK 12 | 67 | 3.95 | 0.35 | 4.00 | 2.8 | 4.7 |
| Placebo | WEEK 16 | 68 | 4.04 | 0.30 | 4.00 | 3.3 | 4.8 |
| Placebo | WEEK 20 | 65 | 3.96 | 0.35 | 4.00 | 2.6 | 4.6 |
| Placebo | WEEK 24 | 57 | 3.97 | 0.33 | 4.00 | 2.9 | 4.6 |
| Placebo | WEEK 26 | 57 | 3.98 | 0.30 | 4.00 | 3.3 | 4.7 |
| Xanomeline High Dose | SCREENING 1 | 84 | 4.03 | 0.28 | 4.00 | 3.2 | 4.9 |
| Xanomeline High Dose | WEEK 2 | 78 | 3.89 | 0.28 | 3.90 | 3.3 | 4.6 |
| Xanomeline High Dose | WEEK 4 | 72 | 3.91 | 0.30 | 3.90 | 3.0 | 4.8 |
| Xanomeline High Dose | WEEK 6 | 66 | 3.95 | 0.28 | 3.90 | 3.4 | 4.5 |
| Xanomeline High Dose | WEEK 8 | 56 | 3.98 | 0.23 | 4.00 | 3.6 | 4.5 |
| Xanomeline High Dose | WEEK 12 | 50 | 3.98 | 0.24 | 3.95 | 3.5 | 4.4 |
| Xanomeline High Dose | WEEK 16 | 37 | 3.99 | 0.19 | 4.00 | 3.6 | 4.4 |
| Xanomeline High Dose | WEEK 20 | 31 | 3.96 | 0.19 | 4.00 | 3.5 | 4.4 |
| Xanomeline High Dose | WEEK 24 | 30 | 4.05 | 0.21 | 4.10 | 3.5 | 4.4 |
| Xanomeline High Dose | WEEK 26 | 27 | 4.00 | 0.23 | 4.00 | 3.5 | 4.5 |
| Xanomeline Low Dose | SCREENING 1 | 82 | 3.98 | 0.26 | 4.00 | 3.2 | 4.6 |
| Xanomeline Low Dose | WEEK 2 | 80 | 3.87 | 0.32 | 3.90 | 3.1 | 4.6 |
| Xanomeline Low Dose | WEEK 4 | 72 | 3.86 | 0.28 | 3.85 | 3.1 | 4.5 |
| Xanomeline Low Dose | WEEK 6 | 62 | 3.84 | 0.26 | 3.80 | 3.2 | 4.4 |
| Xanomeline Low Dose | WEEK 8 | 60 | 3.91 | 0.29 | 3.90 | 3.3 | 5.1 |
| Xanomeline Low Dose | WEEK 12 | 51 | 3.89 | 0.22 | 3.90 | 3.1 | 4.4 |
| Xanomeline Low Dose | WEEK 16 | 42 | 3.91 | 0.30 | 3.90 | 3.2 | 4.5 |
| Xanomeline Low Dose | WEEK 20 | 30 | 3.86 | 0.27 | 3.90 | 2.8 | 4.2 |
| Xanomeline Low Dose | WEEK 24 | 26 | 4.04 | 0.25 | 4.05 | 3.2 | 4.5 |
| Xanomeline Low Dose | WEEK 26 | 25 | 3.92 | 0.24 | 3.90 | 3.5 | 4.4 |

1. Statistical modeling

4.1 LM model

# LM model ----  
lm\_model <- lm(LBORRES ~ ARM \* VISIT + AGE,   
 data = data\_clean)  
  
lm\_model\_results <- broom.mixed::tidy(lm\_model) %>%   
 mutate(across(where(is.numeric), ~ round(.x, 3)))  
  
lm\_numeric\_cols <- names(lm\_model\_results)[sapply(lm\_model\_results, is.numeric)]  
  
lm\_model\_table <- lm\_model\_results %>%  
 flextable() %>%  
 colformat\_num(j = lm\_numeric\_cols, digits = 3) %>% # Ensure 3-digit precision  
 theme\_booktabs() %>% # Optional: Clean table style  
 autofit() # Adjust column widths  
lm\_model\_table

| term | estimate | std.error | statistic | p.value |
| --- | --- | --- | --- | --- |
| (Intercept) | 4.512 | 0.069 | 65.619 | 0.000 |
| ARMXanomeline High Dose | 0.039 | 0.043 | 0.903 | 0.367 |
| ARMXanomeline Low Dose | -0.004 | 0.044 | -0.102 | 0.919 |
| VISITWEEK 2 | -0.096 | 0.043 | -2.207 | 0.027 |
| VISITWEEK 4 | -0.109 | 0.044 | -2.489 | 0.013 |
| VISITWEEK 6 | -0.078 | 0.045 | -1.735 | 0.083 |
| VISITWEEK 8 | -0.010 | 0.045 | -0.212 | 0.832 |
| VISITWEEK 12 | -0.040 | 0.046 | -0.872 | 0.383 |
| VISITWEEK 16 | 0.052 | 0.046 | 1.128 | 0.259 |
| VISITWEEK 20 | -0.030 | 0.046 | -0.646 | 0.518 |
| VISITWEEK 24 | -0.021 | 0.048 | -0.436 | 0.663 |
| VISITWEEK 26 | -0.005 | 0.048 | -0.109 | 0.914 |
| AGE | -0.007 | 0.001 | -8.561 | 0.000 |
| ARMXanomeline High Dose:VISITWEEK 2 | -0.042 | 0.062 | -0.675 | 0.500 |
| ARMXanomeline Low Dose:VISITWEEK 2 | -0.013 | 0.062 | -0.217 | 0.828 |
| ARMXanomeline High Dose:VISITWEEK 4 | -0.014 | 0.063 | -0.218 | 0.827 |
| ARMXanomeline Low Dose:VISITWEEK 4 | -0.007 | 0.063 | -0.104 | 0.917 |
| ARMXanomeline High Dose:VISITWEEK 6 | -0.008 | 0.065 | -0.120 | 0.904 |
| ARMXanomeline Low Dose:VISITWEEK 6 | -0.056 | 0.065 | -0.860 | 0.390 |
| ARMXanomeline High Dose:VISITWEEK 8 | -0.039 | 0.066 | -0.584 | 0.559 |
| ARMXanomeline Low Dose:VISITWEEK 8 | -0.050 | 0.066 | -0.763 | 0.446 |
| ARMXanomeline High Dose:VISITWEEK 12 | -0.010 | 0.068 | -0.150 | 0.881 |
| ARMXanomeline Low Dose:VISITWEEK 12 | -0.054 | 0.068 | -0.800 | 0.424 |
| ARMXanomeline High Dose:VISITWEEK 16 | -0.093 | 0.072 | -1.291 | 0.197 |
| ARMXanomeline Low Dose:VISITWEEK 16 | -0.118 | 0.070 | -1.683 | 0.093 |
| ARMXanomeline High Dose:VISITWEEK 20 | -0.031 | 0.075 | -0.408 | 0.683 |
| ARMXanomeline Low Dose:VISITWEEK 20 | -0.093 | 0.076 | -1.231 | 0.219 |
| ARMXanomeline High Dose:VISITWEEK 24 | 0.042 | 0.077 | 0.551 | 0.582 |
| ARMXanomeline Low Dose:VISITWEEK 24 | 0.064 | 0.080 | 0.798 | 0.425 |
| ARMXanomeline High Dose:VISITWEEK 26 | -0.034 | 0.079 | -0.437 | 0.662 |
| ARMXanomeline Low Dose:VISITWEEK 26 | -0.069 | 0.080 | -0.863 | 0.388 |

4.2 LMM model

# LMM model ----  
# fixed effects: ARM, VISIT, AGE  
# random effect: USUBJID  
lmm\_model <- lmer(LBORRES ~ ARM \* VISIT + AGE + (1 | USUBJID),   
 data = data\_clean)  
  
lmm\_model\_results <- broom.mixed::tidy(lmm\_model) %>%  
 mutate(across(where(is.numeric), ~ round(.x, 1))) %>%   
 select(-group) %>%   
 filter(term %in% c("VISITWEEK 2", "VISITWEEK 4", " VISITWEEK 6", "VISITWEEK 8", "VISITWEEK 12", "VISITWEEK 16", "VISITWEEK 20", "VISITWEEK 24", "VISITWEEK 26", "AGE"))  
  
lmm\_numeric\_cols <- names(lmm\_model\_results)[sapply(lmm\_model\_results, is.numeric)]  
  
lmm\_model\_table <- lmm\_model\_results %>%  
 flextable() %>%  
 colformat\_num(j = lmm\_numeric\_cols, digits = 1) %>% # Ensure 3-digit precision  
 theme\_booktabs() %>% # Optional: Clean table style  
 autofit() # Adjust column widths  
lmm\_model\_table

| effect | term | estimate | std.error | statistic | df | p.value |
| --- | --- | --- | --- | --- | --- | --- |
| fixed | VISITWEEK 2 | -0.1 | 0 | -3.4 | 1,502.6 | 0.0 |
| fixed | VISITWEEK 4 | -0.1 | 0 | -3.7 | 1,504.8 | 0.0 |
| fixed | VISITWEEK 8 | 0.0 | 0 | -1.3 | 1,512.8 | 0.2 |
| fixed | VISITWEEK 12 | -0.1 | 0 | -2.1 | 1,514.3 | 0.0 |
| fixed | VISITWEEK 16 | 0.0 | 0 | 1.0 | 1,514.4 | 0.3 |
| fixed | VISITWEEK 20 | 0.0 | 0 | -1.6 | 1,514.7 | 0.1 |
| fixed | VISITWEEK 24 | 0.0 | 0 | -0.7 | 1,515.0 | 0.5 |
| fixed | VISITWEEK 26 | 0.0 | 0 | -0.1 | 1,515.1 | 0.9 |
| fixed | AGE | 0.0 | 0 | -4.3 | 251.0 | 0.0 |

4.3. LMM model with nesting and map function

data\_perf <- data %>%  
 select(USUBJID, ARM, VISIT, AGE, LBORRES, LBTEST) %>%  
 filter(VISIT %in% ordered\_visits) %>%  
 mutate(  
 LBORRES = as.numeric(LBORRES),  
 VISIT = factor(VISIT, levels = unique(c(ordered\_visits, VISIT))),  
 ARM = factor(ARM, levels = unique(ARM))  
 ) %>%  
 filter(!is.na(LBORRES))

## Warning: There was 1 warning in `mutate()`.  
## ℹ In argument: `LBORRES = as.numeric(LBORRES)`.  
## Caused by warning:  
## ! NAs introduced by coercion

tests <- unique(data\_perf$LBTEST)[1:5]  
   
data\_perf\_filtered <- data\_perf %>% filter(LBTEST %in% tests)  
   
# Analysis with nesting and map function (replacement for for loop)  
data\_analysis <- data\_perf\_filtered %>%  
 group\_by(LBTEST) %>%  
 nest() %>%  
 mutate(  
 model = map(.x = data, ~ lmer(LBORRES ~ ARM \* VISIT + AGE + (1 | USUBJID), data = .)), # Fit the model once  
 test\_results = map(model, tidy), # Extract tidy results  
 tables = map(model, ~ tbl\_regression(., exponentiate = FALSE) # Generate regression table  
 )  
 )

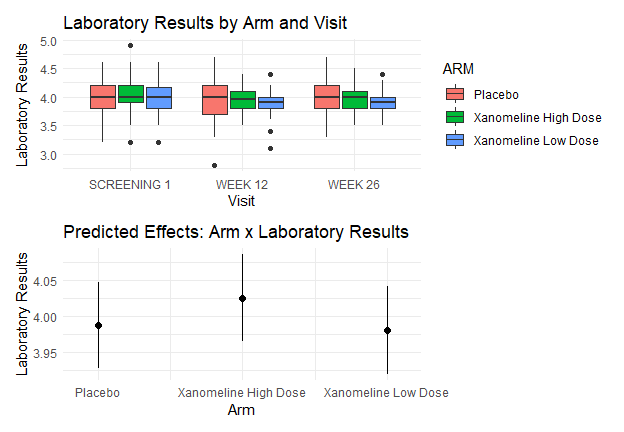
## Warning: There were 2 warnings in `mutate()`.  
## The first warning was:  
## ℹ In argument: `model = map(...)`.  
## ℹ In group 4: `LBTEST = "Anisocytes"`.  
## Caused by warning in `checkConv()`:  
## ! Model failed to converge with max|grad| = 13.7337 (tol = 0.002, component 1)  
## ℹ Run `dplyr::last\_dplyr\_warnings()` to see the 1 remaining warning.

# unnest the 'test\_results'  
test\_results <- data\_analysis %>%  
 select(LBTEST, test\_results) %>%  
 unnest(test\_results)  
test\_results

## # A tibble: 165 × 9  
## # Groups: LBTEST [5]  
## LBTEST effect group term estimate std.error statistic df p.value  
## <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Albumin fixed <NA> (Intercept) 4.56 0.136 33.6 260. 1.59e-96  
## 2 Albumin fixed <NA> ARMXanomeli… 0.0385 0.0434 0.889 553. 3.74e- 1  
## 3 Albumin fixed <NA> ARMXanomeli… -0.00623 0.0435 -0.143 559. 8.86e- 1  
## 4 Albumin fixed <NA> VISITWEEK 2 -0.0973 0.0282 -3.45 1503. 5.80e- 4  
## 5 Albumin fixed <NA> VISITWEEK 4 -0.105 0.0286 -3.66 1505. 2.65e- 4  
## 6 Albumin fixed <NA> VISITWEEK 6 -0.0991 0.0294 -3.37 1511. 7.71e- 4  
## 7 Albumin fixed <NA> VISITWEEK 8 -0.0382 0.0296 -1.29 1513. 1.97e- 1  
## 8 Albumin fixed <NA> VISITWEEK 12 -0.0641 0.0302 -2.12 1514. 3.42e- 2  
## 9 Albumin fixed <NA> VISITWEEK 16 0.0309 0.0301 1.03 1514. 3.05e- 1  
## 10 Albumin fixed <NA> VISITWEEK 20 -0.0487 0.0305 -1.60 1515. 1.11e- 1  
## # ℹ 155 more rows

1. Laboratory results by ARM and VISIT and predicted effects of ARM

plot1 <- data\_clean %>%  
 filter(VISIT %in% c("SCREENING 1", "WEEK 12", "WEEK 26")) %>%   
 ggplot(aes(x = VISIT, y = LBORRES, fill = ARM)) +  
 geom\_boxplot() +  
 labs(  
 title = "Laboratory Results by Arm and Visit",  
 x = "Visit",  
 y = "Laboratory Results"  
 ) +  
 theme\_minimal() +  
 coord\_cartesian()  
  
interaction\_effects <- ggeffects::ggpredict(lmm\_model, terms = c("ARM"))  
  
plot2 <- interaction\_effects %>%  
 plot() +  
 labs(  
 title = "Predicted Effects: Arm x Laboratory Results",  
 x = "Arm",  
 y = "Laboratory Results"  
 ) +  
 theme\_minimal()  
  
combined\_plot <- plot1 + plot2 + plot\_layout(ncol = 1)  
combined\_plot



Combined plot of laboratory results by arm and visit and predicted effects of arm on laboratory results.

1. Measure system response time for queries:

data\_perf <- data %>%  
 select(USUBJID, ARM, VISIT, AGE, LBORRES, LBTEST) %>%  
 filter(VISIT %in% ordered\_visits) %>%  
 mutate(  
 LBORRES = as.numeric(LBORRES),  
 VISIT = factor(VISIT, levels = unique(c(ordered\_visits, VISIT))),  
 ARM = factor(ARM, levels = unique(ARM))  
 ) %>%  
 filter(!is.na(LBORRES))  
  
tests <- unique(data\_perf$LBTEST)[1:30]  
  
start\_time <- Sys.time()  
  
data\_perf\_filtered <- data\_perf %>% filter(LBTEST %in% tests)  
  
# Analysis with nesting and map function (replacement for for loop)  
data\_analysis <- data\_perf\_filtered %>%  
 group\_by(LBTEST) %>%  
 nest() %>%  
 mutate(test\_results = map(.x=data, ~ lmer(LBORRES ~ ARM \* VISIT + AGE + (1 | USUBJID),  
 data = .) %>% tidy()  
 )  
 ) %>%  
 unnest(test\_results) %>%  
 select(-data)  
  
  
for (i in tests) {  
 lmer(LBORRES ~ ARM \* VISIT + AGE + (1 | USUBJID),  
 data = data\_perf %>% filter(LBTEST == i)) %>% na.omit()  
 #print(i)  
 }  
end\_time <- Sys.time()  
response\_time <- end\_time - start\_time  
print(paste("Total Time:", response\_time))

## [1] "Total Time: 6.88285207748413"