1. Import demographic and laboratory data as a dataframes:

dm\_data <- haven::read\_xpt("Data/dm.xpt")  
lab\_data <- haven::read\_xpt("Data/lb.xpt")  
  
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 <chr>, DMDTC <chr>,  
## # DMDY <dbl>

if (is.data.frame(lab\_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… 39 ALB Album… CHEM… 3.9 g/dL 3.3   
## 3 CDISCPIL… LB 01-701… 74 ALB Album… CHEM… 3.8 g/dL 3.3   
## 4 CDISCPIL… LB 01-701… 104 ALB Album… CHEM… 3.7 g/dL 3.3   
## 5 CDISCPIL… LB 01-701… 134 ALB Album… CHEM… 3.8 g/dL 3.3   
## 6 CDISCPIL… LB 01-701… 164 ALB Album… CHEM… 3.8 g/dL 3.3   
## # 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>

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("The dataset has 306 rows and 25 columns, as expected.")  
} else {  
 message("The dataset does not have the expected number of rows or columns.")  
}

## 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(lab\_data) == 59580 & ncol(lab\_data) == 23) {  
 message("The dataset has 59580 rows and 23 columns, as expected.")  
} else {  
 message("The dataset does not have the expected number of rows or columns.")  
}

## 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 <- dm\_data %>%  
 #select(-DOMAIN, -STUDYID) %>%  
 right\_join(lab\_data, by = "USUBJID") %>%   
 select(-c(`STUDYID.y`, `DOMAIN.y`, `STUDYID.x`, `DOMAIN.x`))  
  
# check if join is correct  
if (nrow(data) == nrow(lab\_data) & ncol(data) == ncol(dm\_data) + ncol(lab\_data) - 5) {  
 message("The join was successful.")  
} else {  
 message("The join was unsuccessful.")  
}

## The join was successful.

1. Baseline summary table is generated correctly by arm:

baseline\_summary <- dm\_data %>%  
 select(ARM, SEX, RACE, ETHNIC, COUNTRY) %>%  
 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")

## The following errors were returned during `as\_flex\_table()`:  
## ✖ For variable `COUNTRY` (`ARM`) and "statistic", "p.value", and "parameter"  
## statistics: 'x' and 'y' must have at least 2 levels

baseline\_summary

| **Variable** | **Placebo  N = 861** | **Screen Failure  N = 521** | **Xanomeline High Dose  N = 841** | **Xanomeline Low Dose  N = 841** | **p-value2** |
| --- | --- | --- | --- | --- | --- |
| **Sex** |  |  |  |  | 0.074 |
| F | 53 (62%) | 36 (69%) | 40 (48%) | 50 (60%) |  |
| M | 33 (38%) | 16 (31%) | 44 (52%) | 34 (40%) |  |
| **Race** |  |  |  |  | 0.2 |
| AMERICAN INDIAN OR ALASKA NATIVE | 0 (0%) | 1 (1.9%) | 1 (1.2%) | 0 (0%) |  |
| ASIAN | 0 (0%) | 2 (3.8%) | 0 (0%) | 0 (0%) |  |
| BLACK OR AFRICAN AMERICAN | 8 (9.3%) | 6 (12%) | 9 (11%) | 6 (7.1%) |  |
| WHITE | 78 (91%) | 43 (83%) | 74 (88%) | 78 (93%) |  |
| **Ethnicity** |  |  |  |  | 0.3 |
| HISPANIC OR LATINO | 3 (3.5%) | 5 (9.6%) | 3 (3.6%) | 6 (7.1%) |  |
| NOT HISPANIC OR LATINO | 83 (97%) | 47 (90%) | 81 (96%) | 78 (93%) |  |
| **Country** |  |  |  |  |  |
| USA | 86 (100%) | 52 (100%) | 84 (100%) | 84 (100%) |  |
| 1n (%) | | | | | |
| 2Pearson's Chi-squared test; Fisher's exact test | | | | | |

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

data\_clean <- data %>%   
 na.omit()  
  
mixed\_model <- lm(  
 LBORRES ~ ARM,  
 data = data\_clean  
)  
  
# 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) | 44.4 | 0.5 | 89.7 | 0.0 |
| ARMXanomeline High Dose | 0.1 | 0.8 | 0.1 | 0.9 |
| ARMXanomeline Low Dose | -0.3 | 0.8 | -0.4 | 0.7 |
| n: 3 | | | | |

1. Group comparison summary for radiologic outcomes by arm is generated:

comparison\_results <- data\_clean %>%  
 group\_by(ARM) %>%  
 summarise(  
 mean\_rad\_num = mean(LBORRES, na.rm = TRUE),  
 sd\_rad\_num = sd(LBORRES, 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

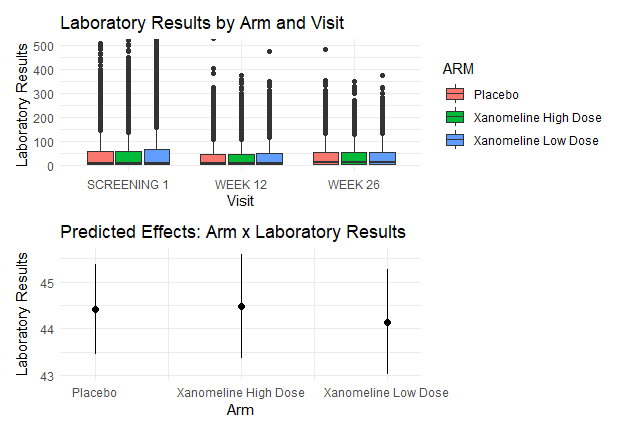
| **ARM** | **mean\_rad\_num** | **sd\_rad\_num** |
| --- | --- | --- |
| **character** | **numeric** | **numeric** |
| Placebo |  | 74.8 |
| Xanomeline High Dose |  | 73.7 |
| Xanomeline Low Dose |  | 71.7 |
| n: 3 | | |

1. Laboratory results by arm and visit and predicted effects of arm on laboratory results plot is generated:

plot1 <- data\_clean %>%  
 mutate(LBORRES = as.numeric(LBORRES)) %>%  
 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(ylim = c(0, 500))  
  
interaction\_effects <- ggeffects::ggpredict(mixed\_model, terms = c("ARM"))

## Some of the focal terms are of type `character`. This may lead to  
## unexpected results. It is recommended to convert these variables to  
## factors before fitting the model.  
## The following variables are of type character: `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



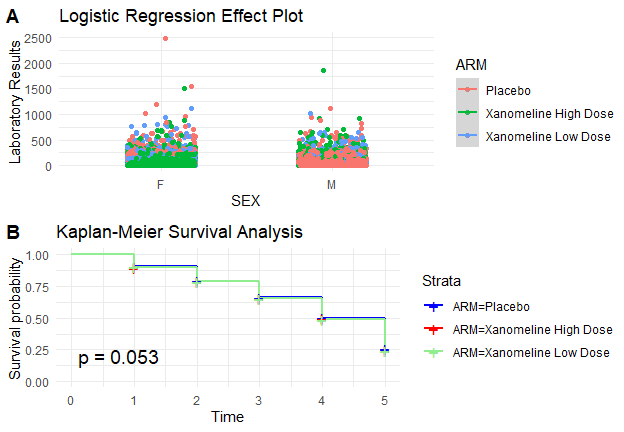
Combined plot of laboratory results by arm and visit and predicted effects of arm on laboratory results.

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

# Logistic regression effect plot  
log\_plot <- ggplot(data\_clean, aes(x = SEX, y = as.numeric(LBORRES), color = ARM)) +  
 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 = "Laboratory Results")   
  
# Kaplan-Meier survival plot  
if (!"time" %in% colnames(data\_clean)) {  
 set.seed(123)  
 data\_clean$time <- sample(1:5, nrow(data\_clean), replace = TRUE) # Random time-to-event data  
 data\_clean$status <- sample(0:1, nrow(data\_clean), replace = TRUE) # Random censoring data  
}  
surv\_fit <- survfit(Surv(time, status) ~ ARM, data = data\_clean)  
  
km\_plot <- ggsurvplot(surv\_fit, data = data\_clean,   
 palette = c("blue", "red", "lightgreen"),   
 pval = TRUE,   
 title = "Kaplan-Meier Survival Analysis")   
km\_plot$plot <- km\_plot$plot + theme\_minimal()   
  
# Combine all plots  
combined\_plot <- ggarrange(  
 log\_plot, km\_plot$plot,  
 ncol = 1, labels = c("A", "B")  
)

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

combined\_plot



Combined plot of logistic regression effect plot and Kaplan-Meier survival analysis.

1. Measure system response time for queries:

start\_time <- Sys.time()  
# Run multiple logistic regression iterations for performance testing  
for (i in 1:1000) {  
 lm(  
 LBORRES ~ ARM,  
 data = data\_clean,  
 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: 29.7784249782562"

9.1 Demographic Data remains unaltered during analysis:

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

## Data integrity is intact.

9.2. Laboratory Data remains unaltered during analysis:

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

## Data integrity is intact.