Analysis

```
library(tidyverse)
```

```
## Warning: package 'purrr' was built under R version 4.5.1
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.2.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
load("strokeStudy.RData")
```

```
x$siteID <- as.character(x$siteID)
unique(x$siteID)
```

```
## [1] "110" "100" "120" "130" "140" "150" "160" "170" "180"
```

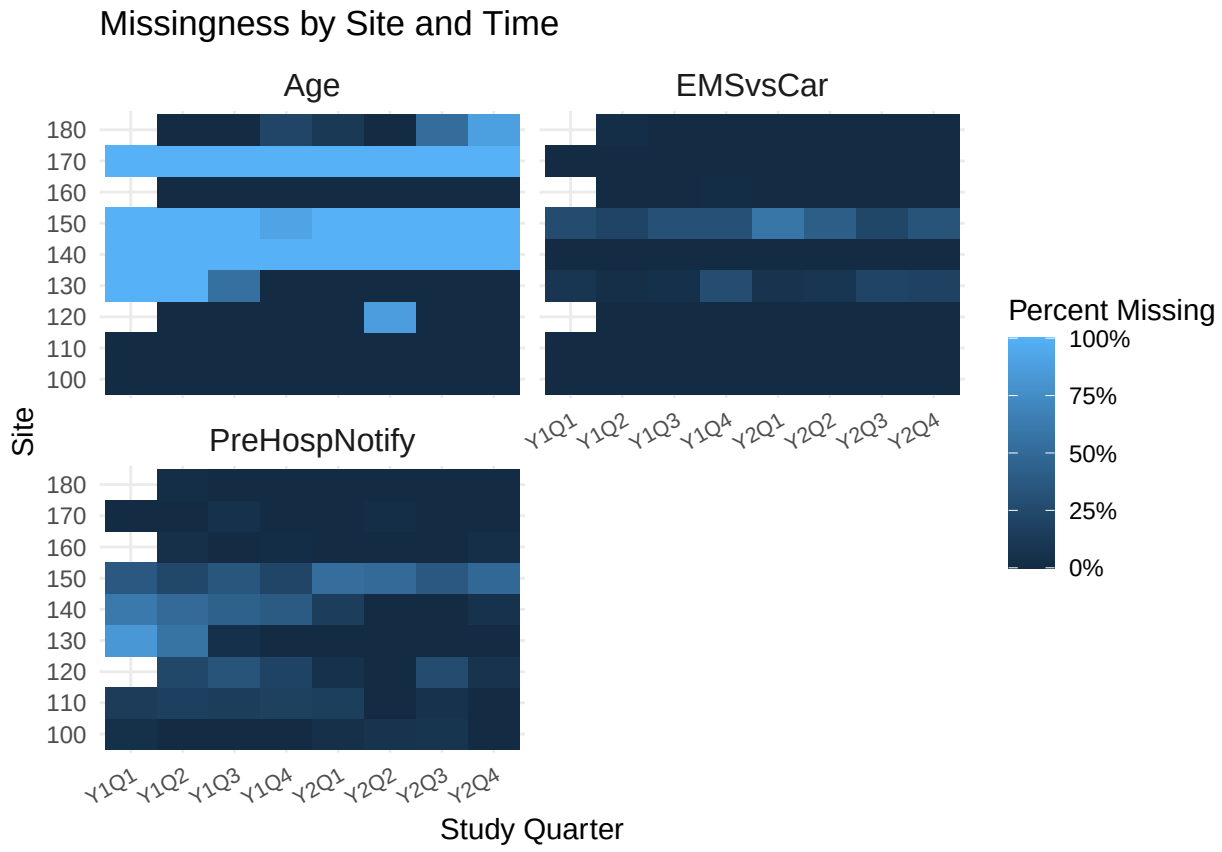
EDA (MISSINGNESS)

```
# Remove NAs from outcome
x <- x |>
  filter(!is.na(homeOrRehab))

missing_vars <- c("Age", "PreHospNotify", "EMSvsCar")

x_missing_outcome <- x |>
  mutate(
    homeOrRehab = factor(homeOrRehab),
    across(all_of(missing_vars), as.character)
  ) |>
  pivot_longer(
    cols = all_of(missing_vars),
    names_to = "variable",
    values_to = "value"
  ) |>
  mutate(is_missing = is.na(value)) |>
```

```
  group_by(variable, homeOrRehab) |>
  summarize(prop_missing = mean(is_missing), .groups = "drop")

ggplot(x_missing_outcome,
       aes(x = homeOrRehab, y = prop_missing, fill = homeOrRehab)) +
  geom_col() +
  facet_wrap(~ variable) +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(
    title = "Missingness by Outcome",
    x = "Outcome",
    y = "Percent Missing"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```


```
missing_by_site_time <- x |>
  select(siteID, Time2, all_of(missing_vars)) |>
  mutate(across(all_of(missing_vars), as.character)) |>
  pivot_longer(
    cols = all_of(missing_vars),
    names_to = "variable",
    values_to = "value"
  ) |>
  mutate(is_missing = is.na(value)) |>
```

```
  group_by(siteID, Time2, variable) |>
  summarize(prop_missing = mean(is_missing), .groups = "drop")

ggplot(missing_by_site_time,
       aes(x = Time2, y = siteID, fill = prop_missing)) +
  geom_tile() +
  facet_wrap(~ variable, nrow = 2) +
  scale_fill_continuous(labels = scales::percent_format()) +
  labs(
    title = "Missingness by Site and Time",
    x = "Study Quarter",
    y = "Site",
    fill = "Percent Missing"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 30, hjust = 1, size = 8),
    strip.text = element_text(size = 12)
  )
```

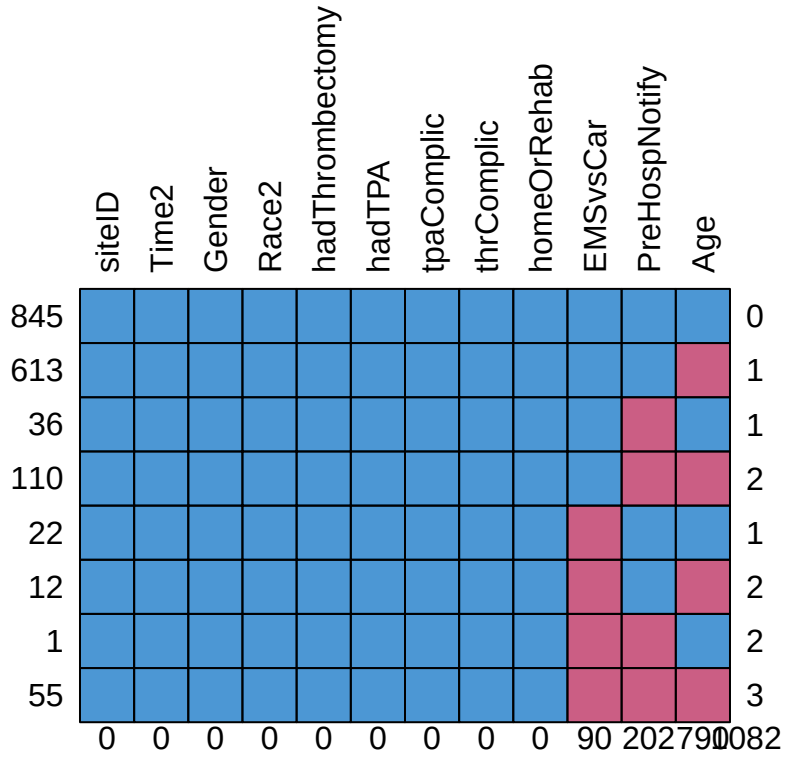



```
library(mice)
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
md.pattern(x,rotate.names = TRUE)
```




```
##     siteID Time2 Gender Race2 hadThrombectomy hadTPA tpaComplic thrComplic
## 845      1     1      1     1               1      1          1          1
## 613      1     1      1     1               1      1          1          1
## 36       1     1      1     1               1      1          1          1
## 110      1     1      1     1               1      1          1          1
## 22       1     1      1     1               1      1          1          1
## 12       1     1      1     1               1      1          1          1
## 1        1     1      1     1               1      1          1          1
## 55       1     1      1     1               1      1          1          1
##          0     0      0     0               0      0          0          0
##     homeOrRehab EMSvsCar PreHospNotify Age
## 845           1        1             1   1    0
## 613           1        1             1   0    1
## 36            1        1             0   1    1
## 110           1        1             0   0    2
```

```
## 22            1        0             1   1    1
## 12            1        0             1   0    2
## 1             1        0             0   1    2
## 55            1        0             0   0    3
##              0       90           202 790 1082
```

DATA CLEANING / IMPUTATION

Packages

```
library(dplyr)
```

Data Cleaning

```
strokes <- x
strokes$EMSvsCar <- factor(strokes$EMSvsCar)
```

```
strokes |>
  group_by(hadTPA, tpaComplic) |>
  count()
```

```
## # A tibble: 3 x 3
## # Groups:   hadTPA, tpaComplic [3]
##   hadTPA tpaComplic     n
##   <lgl>  <lgl>      <int>
## 1 FALSE  FALSE        469
## 2 TRUE   FALSE       1182
## 3 TRUE   TRUE          43
```

```
strokes <- strokes %>%
  mutate(TPA_group = case_when(
    hadTPA == FALSE ~ "NoTPA",
    hadTPA == TRUE & tpaComplic == FALSE ~ "TPA_NoComp",
    hadTPA == TRUE & tpaComplic == TRUE  ~ "TPA_Comp"
  ))

# Optional: make it a factor with ordered levels
strokes <- strokes %>%
  mutate(TPA_group = factor(TPA_group, levels = c("NoTPA", "TPA_NoComp", "TPA_Comp")))

# Check result
table(strokes$TPA_group)
```

```
##
##      NoTPA TPA_NoComp   TPA_Comp
##        469       1182         43
```

```
strokes <- strokes |>
  select(-hadTPA) |>
  select(-tpaComplic)
```

There is some correlation with hadTP and tpaComplication, so we made a new variable and removed the other two.

Missingness

First impute age, then EMS vs Car, then Notify

I chose this order because of the dependency thing (unless I misunderstood that).

PreHospNotify cannot be imputed without knowing EMSvsCar first. –> this is because when EMSvsCar = 1, PreHospNotify has be No.

See

```
strokes |> group_by(strokes$EMSvsCar, PreHospNotify) |> count()
```

```
## # A tibble: 7 x 3
## # Groups:   strokes$EMSvsCar, PreHospNotify [7]
##   `strokes$EMSvsCar` PreHospNotify     n
##   <fct>              <fct>         <int>
## 1 0                  No              190
## 2 1                  Yes            1099
## 3 1                  No              169
## 4 1                  <NA>            146
## 5 <NA>               Yes              32
## 6 <NA>               No                2
## 7 <NA>               <NA>             56
```

We need to maintain the constraint thought that if it's car, there won't be notify. –> I asked and because there's not a forced rule for when EMS vs Car equals 1, i.e. EMS could be PreHospNotify = YES or NO when it's 1. It recommended resolving using post-processing.

```
library(mice)

# Select relevant variables or full dataset
#data <- strokes |>
  #select(-siteID)


strokes$siteID <-factor(strokes$siteID)
data <- strokes

# STEP 1: Specify imputation methods
# "" means: do NOT impute the variable
method <- make.method(data)

method["Age"] <- "pmm"
method["EMSvsCar"] <- "logreg"          # assuming it's binary
```

```r
method["PreHospNotify"] <- "logreg"    # assuming it's binary

# Make sure NO OTHER variables are imputed
for (v in names(method)) {
  if (!v %in% c("Age", "EMSvsCar", "PreHospNotify")) {
    method[v] <- ""
  }
}

# STEP 2: Build predictor matrix
pred <- make.predictorMatrix(data)

# Remove self-prediction
pred["Age", "Age"] <- 0
pred["EMSvsCar", "EMSvsCar"] <- 0
pred["PreHospNotify", "PreHospNotify"] <- 0

# (OPTIONAL) If you want all variables to predict the three variables,
# keep predictor matrix as is.
# Or customize:
# pred["EMSvsCar", ] <- 0
# pred["EMSvsCar", c("Age")] <- 1

# STEP 3: Set the order of imputation
visitSequence <- c("Age", "EMSvsCar", "PreHospNotify")

# STEP 4: Run MICE
imp <- mice(
  data,
  m = 20,
  method = method,
  predictorMatrix = pred,
  visitSequence = visitSequence,
  maxit = 20,
  print = FALSE
)
```

```
## Warning: Number of logged events: 400
```

my post-processing

holds that all cases where EMS vs Car = 0, there won't be a prehospital notify

```r
completed_list <- mice::complete(imp, action = "all")

# Define the categories exactly as they appear in your data
fix_rule <- function(df) {
  df$PreHospNotify[df$EMSvsCar == 0] <- "No"
  return(df)
}

completed_fixed <- lapply(completed_list, fix_rule)
```

Rule out MNAR

Need to validate we can make the MAR assumption. These are diagnostics.

```
##### SENSITIVITY ANALYSIS - Age + EMSvsCar + PreHospNotify #####

library(dplyr)
library(mice)
library(broom)
```

```
## Warning: package 'broom' was built under R version 4.5.1
```

```
# 1. Define delta values
delta_age <- c(-2, 0, 2, 5)                          # numeric adjustments
delta_ems <- c(-1, 0, 1)                             # 0/1 flips
delta_pre <- c("flip_to_no", "none", "flip_to_yes")# categorical flips

# Storage
sensitivity_results <- list()

# 2. Loop through all effects
for(a in delta_age){
  for(e in delta_ems){
    for(pn in delta_pre){

      adjusted_dfs <- lapply(completed_fixed, function(df){
        df_new <- df

        ##### AGE adjustment #####
        df_new$Age <- df_new$Age + a

        ##### EMSvsCar flips #####
        if(e == -1){
          df_new$EMSvsCar[df_new$EMSvsCar == 1] <- 0
        } else if(e == 1){
          df_new$EMSvsCar[df_new$EMSvsCar == 0] <- 1
        }

        ##### PreHospNotify flips #####
        if(pn == "flip_to_no"){
          df_new$PreHospNotify[df_new$PreHospNotify == "Yes"] <- "No"
        } else if(pn == "flip_to_yes"){
          df_new$PreHospNotify[df_new$PreHospNotify == "No"] <- "Yes"
        }

        ##### SAFETY CHECKS #####
        # EMSvsCar must have 2 levels
        if(length(unique(df_new$EMSvsCar)) < 2){
          return(NULL)
        }

        # PreHospNotify must have 2 levels
        if(length(unique(df_new$PreHospNotify)) < 2){
```

```
          return(NULL)
        }

        df_new
      })

      # Remove NULL datasets created by safety checks
      adjusted_dfs <- Filter(Negate(is.null), adjusted_dfs)

      # If ALL datasets collapsing → skip scenario
      if(length(adjusted_dfs) == 0){
        scenario_name <- paste0("Age_", a, "_EMS_", e, "_Pre_", pn)
        sensitivity_results[[scenario_name]] <- "SKIPPED (variable collapsed)"
        next
      }

      ##### Fit logistic models #####
      model_list <- lapply(adjusted_dfs, function(df){
        glm(
          homeOrRehab ~ Age + EMSvsCar + PreHospNotify,
          data = df,
          family = binomial
        )
      })

      ##### Pool results #####
      pooled <- pool(model_list)
      pooled_summary <- summary(pooled)

      ##### Save scenario #####
      scenario_name <- paste0("Age_", a, "_EMS_", e, "_Pre_", pn)
      sensitivity_results[[scenario_name]] <- pooled_summary
    }
  }
}

##### 3. Example: view a scenario #####
sensitivity_results[["Age_0_EMS_0_Pre_none"]]
```

```
##              term    estimate   std.error   statistic        df      p.value
## 1     (Intercept)  5.63028315 0.612644473  9.19013130  69.30380 1.295560e-13
## 2             Age -0.05185686 0.007775077 -6.66962719  38.40706 6.567213e-08
## 3       EMSvsCar1 -1.13386838 0.313168982 -3.62062799 566.14790 3.202523e-04
## 4 PreHospNotifyNo -0.01474296 0.190486225 -0.07739647 424.57042 9.383446e-01
```

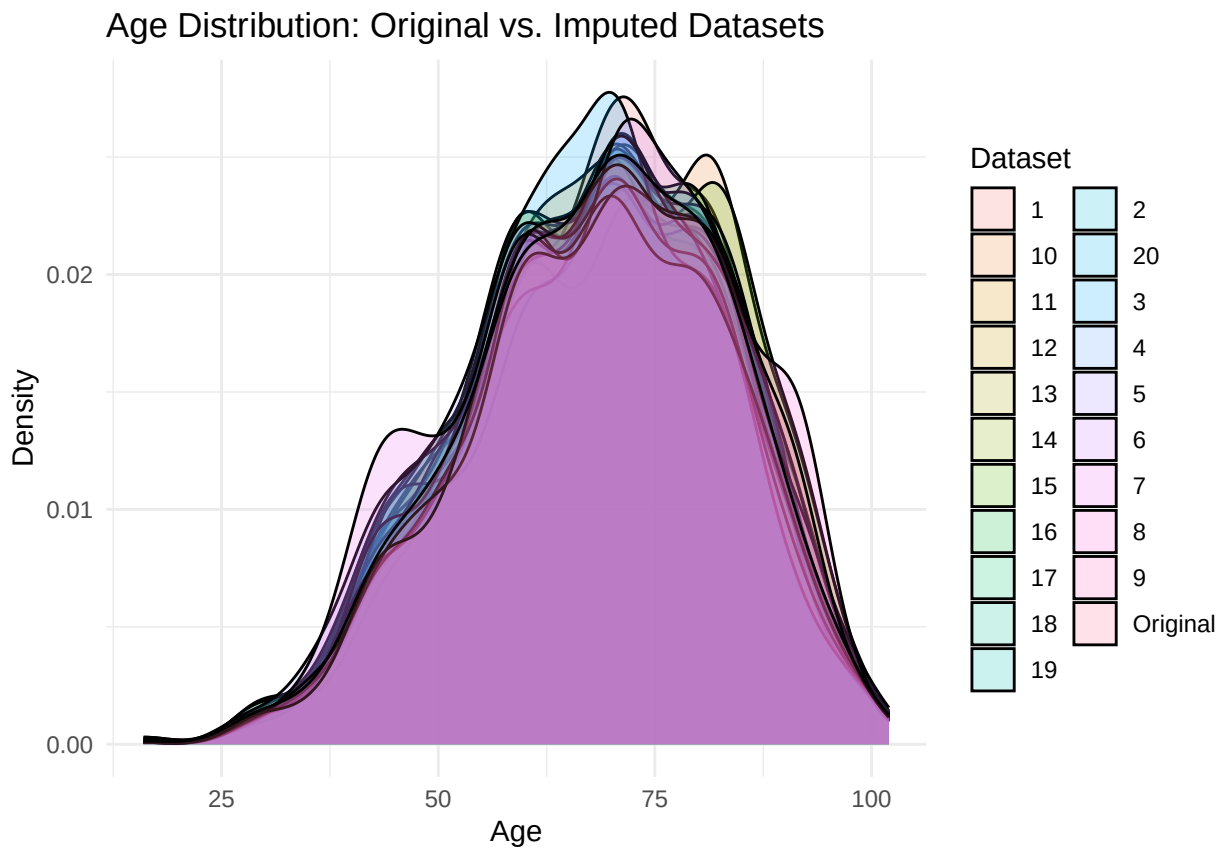
```
model_results <- lapply(sensitivity_age, function(s) {
  with(mice::as.mids(s$data), glm(StrokeOutcome ~ Age + EMSvsCar + PreHospNotify, family=binomial))
})

pooled_results <- lapply(model_results, pool)
```

EDA (POST IMPUTATION)

First is just checking the distributions pre and post imputation (they should be approximately the same):

```
# Combining all completed datasets into a long dataframe
post_imp_long <- bind_rows(
  lapply(seq_along(completed_fixed), function(i) {
    completed_fixed[[i]] |> mutate(.imp = i)
  })
)

# Age
orig <- x |> mutate(.imp = "Original")

post_imp_long2 <- post_imp_long |>
  mutate(.imp = as.character(.imp))

age_compare <- bind_rows(
  orig |> select(Age, .imp),
  post_imp_long2 |> select(Age, .imp)
)

ggplot(age_compare, aes(x = Age, fill = factor(.imp))) +
  geom_density(alpha = 0.2) +
  labs(
    title = "Age Distribution: Original vs. Imputed Datasets",
    x = "Age",
    y = "Density",
    fill = "Dataset"
  ) +
  theme_minimal()
```

```
## Warning: Removed 790 rows containing non-finite outside the scale range
## (‘stat_density()‘).
```




```r
# EMSvsCar, PreHospNotify
orig_summary <- x |>
  summarize(
    EMSvsCar = mean(EMSvsCar == 1, na.rm = TRUE),
    PreHospNotify = mean(PreHospNotify == "Yes", na.rm = TRUE)
  ) |>
  mutate(.imp = "Original") |>
  pivot_longer(
    cols = c("EMSvsCar", "PreHospNotify"),
    names_to = "variable",
    values_to = "prop_yes"
  )

imp_summary <- post_imp_long |>
  mutate(
    .imp = as.character(.imp),
    EMSvsCar = as.numeric(as.character(EMSvsCar)),
    PreHospNotify = as.character(PreHospNotify)
  ) |>
  group_by(.imp) |>
  summarize(
    EMSvsCar = mean(EMSvsCar == 1, na.rm = TRUE),
    PreHospNotify = mean(PreHospNotify == "Yes", na.rm = TRUE)
  ) |>
  pivot_longer(
    cols = c("EMSvsCar", "PreHospNotify"),
```

```
    names_to = "variable",
    values_to = "prop_yes"
  )

binary_summary_all <- binary_summary_all |>
  mutate(
    .imp = factor(.imp, levels = c("Original", as.character(1:20)))
  )

ggplot(binary_summary_all, aes(x = .imp, y = prop_yes, fill = (.imp == "Original"))) +
  geom_col() +
  facet_wrap(~ variable, nrow = 1) +
  scale_fill_manual(values = c("grey70", "tomato"), guide = "none") +
  labs(
    title = "Proportion of Binary Variables: Original vs. Imputed Datasets",
    x = "Dataset",
    y = "Proportion 'Yes'"
  ) +
  coord_cartesian(ylim = c(0, 1)) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

Now onto EDA for association:

```
# just doing eda with 1st bc distributions roughly the same - obviously we would want to use all for pr
assoc_df <- eda_df |>
  select(
    Disposition,
    AgeGroup,
    Gender,
    Race2,
    ArrivalMode,
    PreHospitalNotification,
    Thrombectomy,
    TPAComplication,
    ThrombectomyComplication,
    StudyQuarter
  ) |>
  pivot_longer(
    cols = -Disposition,
    names_to = "Variable",
    values_to = "Level"
  ) |>
  group_by(Variable, Level) |>
  summarize(
    n = n(),
    prop_favorable = mean(Disposition == "Favorable Discharge"),
    se = sqrt(prop_favorable * (1 - prop_favorable) / n),
    .groups = "drop"
  ) |>
  mutate(
    Variable = recode(
```

```
      Variable,
      AgeGroup = "Age Group",
      Gender = "Gender",
      Race2 = "Race",
      ArrivalMode = "Arrival Mode",
      PreHospitalNotification = "Pre-Hospital Notification",
      Thrombectomy = "Thrombectomy",
      TPAComplication = "TPA Complication",
      ThrombectomyComplication = "Thrombectomy Complication",
      StudyQuarter = "Study Quarter"
    )
  )

ggplot(assoc_df, aes(x = Level, y = prop_favorable)) +
  geom_col(fill = "#4472C4") +
  geom_errorbar(
    aes(
      ymin = prop_favorable - 1.96 * se,
      ymax = prop_favorable + 1.96 * se
    ),
    width = 0.2
  ) +
  facet_wrap(~ Variable, scales = "free_x", nrow = 2) +
  scale_y_continuous(labels = scales::percent_format(), limits = c(0, 1)) +
  labs(
    title = "Association Between Predictors and Favorable Discharge",
    x = "",
    y = "Percent Favorable Discharge"
  ) +
  theme_minimal(base_size = 15) +
  theme(
    strip.text = element_text(face = "bold", size = 16),
    axis.text.x = element_text(angle = 45, hjust = 1, size = 12),
    axis.text.y = element_text(size = 12),
    plot.title = element_text(size = 20, face = "bold")
  )
```

FREQUENTIST ANALYSIS

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.5.1

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```r
results = NULL
AUCs = NULL
for (i in 1:20){
  data <- completed_fixed[[i]]
  data$homeOrRehab <- if_else(data$homeOrRehab,1,0)
  modelFit <- glm(factor(homeOrRehab) ~ ., data = data, family = "binomial")
  AUCs <- c(AUCs,auc(data$homeOrRehab,predict(modelFit,data)))
  results <- c(results, list(summary(modelFit)))
}
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

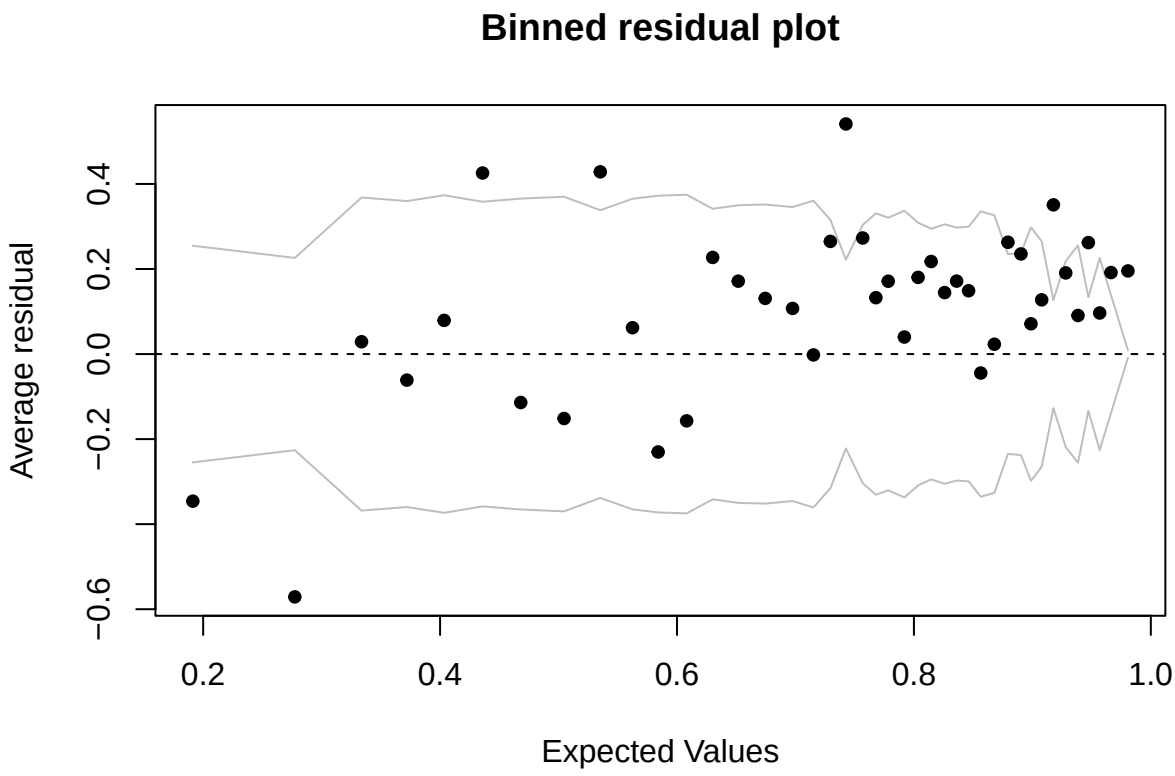
```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

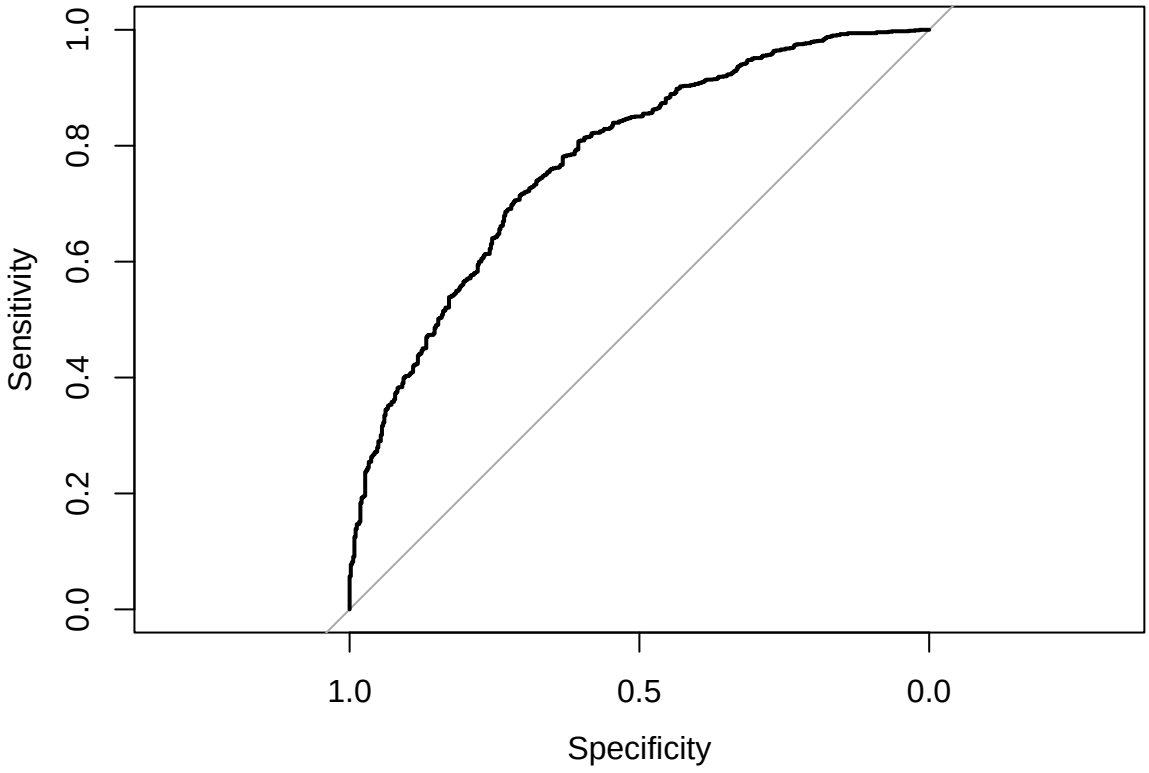
```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
arm::binnedplot(fitted(modelFit),residuals(modelFit))
```




```
plot.roc(data$homeOrRehab,predict(modelFit,data))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```




```
results[[1]]$coefficients |>
  as.data.frame() |>
  select(-`z value`) |>
  kableExtra::kable()
```

	Estimate	Std. Error	Pr(>|z|)
(Intercept)	5.7799301	0.5705023	0.0000000
siteID110	-0.1232194	0.3026655	0.6839242
siteID120	-0.6621678	0.3282635	0.0436763
siteID130	-0.7932584	0.2582793	0.0021311
siteID140	-0.1548561	0.2803877	0.5807478
siteID150	0.0994261	0.2713844	0.7140916
siteID160	-0.3518889	0.2685118	0.1900215
siteID170	-0.1204110	0.2531221	0.6342857
siteID180	-0.3552787	0.2820829	0.2078557
Time2Y1Q2	0.2446337	0.2577567	0.3425761
Time2Y1Q3	0.3187541	0.2729834	0.2429404
Time2Y1Q4	0.1261808	0.2617033	0.6296979
Time2Y2Q1	0.2551876	0.2620325	0.3301173
Time2Y2Q2	0.5919624	0.2757708	0.0318272
Time2Y2Q3	0.4661083	0.2695416	0.0837620
Time2Y2Q4	0.3254628	0.2669935	0.2228474
Age	-0.0465877	0.0048068	0.0000000
GenderFemale	-0.4761476	0.1198258	0.0000708

	Estimate	Std. Error	Pr(>|z|)
Race2African American	-0.3710837	0.1479807	0.0121537
Race2Other	-0.3388274	0.2713401	0.2117681
Race2Missing	-0.1073636	0.3138726	0.7323049
EMSvsCar1	-1.2298700	0.3026863	0.0000484
PreHospNotifyNo	-0.1110070	0.1815570	0.5409237
hadThrombectomyTRUE	-0.5497037	0.1574632	0.0004812
thrComplicTRUE	-1.2652359	0.3734873	0.0007050
TPA_groupTPA_NoComp	0.4767152	0.1639096	0.0036328
TPA_groupTPA_Comp	-0.9265502	0.3702244	0.0123260

```
data.frame(iter = seq(1,20), AUC = AUCs) |>
  kableExtra::kable()
```

iter	AUC
1	0.7494305
2	0.7705442
3	0.7814613
4	0.7767622
5	0.7803915
6	0.7715149
7	0.7807510
8	0.7597893
9	0.7819744
10	0.7661823
11	0.7839799
12	0.7961077
13	0.7810967
14	0.7676226
15	0.7562897
16	0.7639377
17	0.7964039
18	0.7877032
19	0.8062316
20	0.7757607

BAYESIAN ANALYSIS

{{< pagebreak >}}

```
results[[1]]$coefficients |>
  as.data.frame() |>
  select(-`z value`) |>
  kableExtra::kable()
```

	Estimate	Std. Error	Pr(>|z|)
(Intercept)	5.7799301	0.5705023	0.0000000
siteID110	-0.1232194	0.3026655	0.6839242

	Estimate	Std. Error	Pr(>|z|)
siteID120	-0.6621678	0.3282635	0.0436763
siteID130	-0.7932584	0.2582793	0.0021311
siteID140	-0.1548561	0.2803877	0.5807478
siteID150	0.0994261	0.2713844	0.7140916
siteID160	-0.3518889	0.2685118	0.1900215
siteID170	-0.1204110	0.2531221	0.6342857
siteID180	-0.3552787	0.2820829	0.2078557
Time2Y1Q2	0.2446337	0.2577567	0.3425761
Time2Y1Q3	0.3187541	0.2729834	0.2429404
Time2Y1Q4	0.1261808	0.2617033	0.6296979
Time2Y2Q1	0.2551876	0.2620325	0.3301173
Time2Y2Q2	0.5919624	0.2757708	0.0318272
Time2Y2Q3	0.4661083	0.2695416	0.0837620
Time2Y2Q4	0.3254628	0.2669935	0.2228474
Age	-0.0465877	0.0048068	0.0000000
GenderFemale	-0.4761476	0.1198258	0.0000708
Race2African American	-0.3710837	0.1479807	0.0121537
Race2Other	-0.3388274	0.2713401	0.2117681
Race2Missing	-0.1073636	0.3138726	0.7323049
EMSvsCar1	-1.2298700	0.3026863	0.0000484
PreHospNotifyNo	-0.1110070	0.1815570	0.5409237
hadThrombectomyTRUE	-0.5497037	0.1574632	0.0004812
thrComplicTRUE	-1.2652359	0.3734873	0.0007050
TPA_groupTPA_NoComp	0.4767152	0.1639096	0.0036328
TPA_groupTPA_Comp	-0.9265502	0.3702244	0.0123260