exploration (3 variables selection)

2025-12-15

```r
install.packages("haven")
```

```
##
## The downloaded binary packages are in
##  /var/folders/g_/ljm_0wb9519gm4d8zgwd_s300000gn/T//RtmpvRbswa/downloaded_packages
```

```r
library(haven)

gss <- read_dta("/Users/joyqu/Desktop/PLSC/GSS2024.dta")
```

```r
head(gss)
```

```
## # A tibble: 6 x 813
##   year         id wrkstat hrs1        hrs2        evwork      wrkslf  occ10
##   <dbl+lbl> <dbl> <dbl+l> <dbl+lbl>   <dbl+lbl>   <dbl+lbl>   <dbl+l> <dbl+lbl>
## 1 2024          1 1 [wor~    43       NA(i) [iap] NA(i) [iap] 2 [som~  230 [edu~
## 2 2024          2 5 [ret~ NA(i) [iap] NA(i) [iap]     1 [yes] 2 [som~  800 [acc~
## 3 2024          3 5 [ret~ NA(i) [iap] NA(i) [iap]     1 [yes] 2 [som~  430 [man~
## 4 2024          4 2 [wor~    20       NA(i) [iap] NA(i) [iap] 2 [som~ 4760 [ret~
## 5 2024          5 5 [ret~ NA(i) [iap] NA(i) [iap]     1 [yes] 2 [som~ 5860 [off~
## 6 2024          6 4 [une~ NA(i) [iap] NA(i) [iap] NA(i) [iap] 1 [sel~ 4000 [che~
## # i 805 more variables: prestg10 <dbl+lbl>, prestg105plus <dbl+lbl>,
## #   indus10 <dbl+lbl>, marital <dbl+lbl>, martype <dbl+lbl>, divorce <dbl+lbl>,
## #   widowed <dbl+lbl>, spwrksta <dbl+lbl>, sphrs1 <dbl+lbl>, sphrs2 <dbl+lbl>,
## #   spevwork <dbl+lbl>, cowrksta <dbl+lbl>, cowrkslf <dbl+lbl>,
## #   coevwork <dbl+lbl>, cohrs1 <dbl+lbl>, cohrs2 <dbl+lbl>, spwrkslf <dbl+lbl>,
## #   spocc10 <dbl+lbl>, sppres10 <dbl+lbl>, sppres105plus <dbl+lbl>,
## #   spind10 <dbl+lbl>, coocc10 <dbl+lbl>, coind10 <dbl+lbl>, ...
```

```r
dim(gss)
```

```
## [1] 3309  813
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)

# Keep only needed variables
```

```r
gss_clean <- gss %>%
  select(polviews, age, race, sex) %>%
  # remove "Don't Know / NA / Refused / No answer"
  filter(!polviews %in% c(8, 9),    # GSS missing codes for polviews
         !is.na(polviews)) %>%
  # Convert categorical vars to factors
  mutate(
    polviews = as.integer(polviews),         # 1=ext lib ... 7=ext cons
    race = factor(race),
    sex = factor(sex)
  )
head(gss_clean)
```

```
## # A tibble: 6 x 4
##   polviews age       race  sex
##      <int> <dbl+lbl> <fct> <fct>
## 1        4 33        2     1
## 2        3 64        1     1
## 3        1 69        1     2
## 4        4 70        1     2
## 5        2 48        1     2
## 6        4 30        1     2
```

```r
set.seed(123)   # makes the sample reproducible

sample100 <- gss_clean %>%
  drop_na() %>%        # removes any row with ANY missing value
  sample_n(100)

head(sample100)
```

```
## # A tibble: 6 x 4
##   polviews age       race  sex
##      <int> <dbl+lbl> <fct> <fct>
## 1        3 59        1     1
## 2        4 52        1     2
## 3        6 61        1     1
## 4        4 45        1     2
## 5        4 28        3     1
## 6        4 62        1     2
```

```r
sample100_nolabel <- sample100 %>%
  select(-polviews)      # remove the numeric ideology variable
head(sample100_nolabel)
```

```
## # A tibble: 6 x 3
##   age       race  sex
##   <dbl+lbl> <fct> <fct>
## 1 59        1     1
## 2 52        1     2
## 3 61        1     1
## 4 45        1     2
## 5 28        3     1
## 6 62        1     2
```

```r
#used same 100 person sample as in 7 variable prediction
var <- read.csv("/Users/joyqu/Desktop/PLSC/3_var/3_var_gss_gpt5_predictions.csv")
head(var)
```

```
##   age race sex pred_polview
## 1  67    1   1            6
## 2  56    3   2            4
## 3  33    1   2            4
## 4  24    1   2            3
## 5  46    1   2            5
## 6  25    1   1            4
```

```r
# Extract variables
y_true <- as.numeric(sample100$polviews)
y_pred <- as.numeric(var$pred_polview)

# Compute metrics
MAE <- mean(abs(y_true - y_pred))
MSE <- mean((y_true - y_pred)^2)
Accuracy <- mean(y_true == y_pred)
Within1 <- mean(abs(y_true - y_pred) <= 1)

cat("Mean Absolute Error:", MAE, "\n")
```

```
## Mean Absolute Error: 1.7
```

```r
cat("Mean Squared Error:", MSE, "\n")
```

```
## Mean Squared Error: 4.28
```

```r
cat("Exact Match Accuracy:", round(Accuracy*100, 1), "%\n")
```

```
## Exact Match Accuracy: 12 %
```

```r
cat("Within ±1 Accuracy:", round(Within1*100, 1), "%\n")
```

```
## Within ±1 Accuracy: 51 %
```

```r
narrative <- read.csv("/Users/joyqu/Desktop/PLSC/3_var/3_var_gss_gpt5_narrative_predictions.csv")
head(narrative)
```

```
##
## 1                       67 years old, this white man has settled into a steady rhythm of daily life.
## 2 56 years old, this from a diverse background woman has settled into a steady rhythm of daily life.
## 3                     33 years old, this white woman has settled into a steady rhythm of daily life.
## 4                     24 years old, this white woman has settled into a steady rhythm of daily life.
## 5                     46 years old, this white woman has settled into a steady rhythm of daily life.
## 6                       25 years old, this white man has settled into a steady rhythm of daily life.
##   pred_polview_narr
## 1                 5
## 2                 4
## 3                 4
## 4                 4
## 5                 4
## 6                 4
```

```r
# Extract variables
y_true <- as.numeric(sample100$polviews)
```

```r
y_pred <- as.numeric(narrative$pred_polview_narr)

# Compute metrics
MAE <- mean(abs(y_true - y_pred))
MSE <- mean((y_true - y_pred)^2)
Accuracy <- mean(y_true == y_pred)
Within1 <- mean(abs(y_true - y_pred) <= 1)

cat("Mean Absolute Error:", MAE, "\n")
```

```
## Mean Absolute Error: 1.17
```

```r
cat("Mean Squared Error:", MSE, "\n")
```

```
## Mean Squared Error: 2.47
```

```r
cat("Exact Match Accuracy:", round(Accuracy*100, 1), "%\n")
```

```
## Exact Match Accuracy: 32 %
```

```r
cat("Within ±1 Accuracy:", round(Within1*100, 1), "%\n")
```

```
## Within ±1 Accuracy: 65 %
```

```r
library(dplyr)
library(readr)
library(caret)    # for confusionMatrix
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library(MLmetrics)  # for f1
```

```
##
## Attaching package: 'MLmetrics'

## The following objects are masked from 'package:caret':
##
##     MAE, RMSE

## The following object is masked from 'package:base':
##
##     Recall
```

```r
library(purrr)
```

```
##
## Attaching package: 'purrr'

## The following object is masked from 'package:caret':
##
##     lift
```

```r
library(dplyr)

df <- sample100 %>%
  mutate(row_id = row_number()) %>%
  select(
    row_id,
    POLVIEWS_TRUE = polviews,
```

```r
    age, sex, race  # <- keep whatever predictors you want
  ) %>%
  inner_join(
    var %>%
      mutate(row_id = row_number()) %>%
      select(row_id, pred_var = pred_polview),
    by = "row_id"
  ) %>%
  inner_join(
    narrative %>%
      mutate(row_id = row_number()) %>%
      select(row_id, pred_narr = pred_polview_narr),
    by = "row_id"
  )
```

```r
library(dplyr)
f1_macro <- function(true, pred) {
  true <- as.character(true)
  pred <- as.character(pred)

  f1_scores <- sapply(unique(true), function(cls) {
    MLmetrics::F1_Score(
      y_pred = pred == cls,
      y_true = true == cls
    )
  })
  mean(f1_scores, na.rm = TRUE)
}

f1_weighted <- function(true, pred) {
  true <- as.character(true)
  pred <- as.character(pred)

  classes <- unique(true)
  weights <- prop.table(table(true))

  f1_scores <- sapply(classes, function(cls) {
    MLmetrics::F1_Score(
      y_pred = pred == cls,
      y_true = true == cls
    )
  })

  sum(f1_scores * weights[names(f1_scores)], na.rm = TRUE)
}

# 1. Build df and KEEP ALL predictors from sample100
df <- sample100 %>%
  mutate(row_id = row_number()) %>%
  select(
    row_id,
    POLVIEWS_TRUE = polviews,
    # keep ALL predictors here:
    age,
```

```r
    sex,
    race
  ) %>%
  inner_join(
    var %>%
      mutate(row_id = row_number()) %>%
      select(row_id, pred_var = pred_polview),
    by = "row_id"
  ) %>%
  inner_join(
    narrative %>%
      mutate(row_id = row_number()) %>%
      select(row_id, pred_narr = pred_polview_narr),
    by = "row_id"
  )

df <- df %>%
  mutate(
    # Factor version for F1
    POLVIEWS_TRUE_fac = factor(POLVIEWS_TRUE),
    pred_var_fac      = factor(pred_var,  levels = levels(POLVIEWS_TRUE_fac)),
    pred_narr_fac     = factor(pred_narr, levels = levels(POLVIEWS_TRUE_fac)),

    # Numeric version for bias / error
    polviews_num = as.numeric(as.character(POLVIEWS_TRUE)),
    pred_var_num = as.numeric(as.character(pred_var)),
    pred_narr_num = as.numeric(as.character(pred_narr)),

    # Signed errors
    error_var  = pred_var_num  - polviews_num,
    error_narr = pred_narr_num - polviews_num
  )
results <- tibble(
  Model = c("Variable Model", "Narrative Model"),
  Macro_F1 = c(
    f1_macro(df$POLVIEWS_TRUE_fac, df$pred_var_fac),
    f1_macro(df$POLVIEWS_TRUE_fac, df$pred_narr_fac)
  ),
  Weighted_F1 = c(
    f1_weighted(df$POLVIEWS_TRUE_fac, df$pred_var_fac),
    f1_weighted(df$POLVIEWS_TRUE_fac, df$pred_narr_fac)
  )
)

print(results)
```

```
## # A tibble: 2 x 3
##   Model           Macro_F1 Weighted_F1
##   <chr>              <dbl>       <dbl>
## 1 Variable Model     0.838       0.756
## 2 Narrative Model    0.847       0.696
```

```r
mislabeled_comparison <- df %>%
  mutate(
```

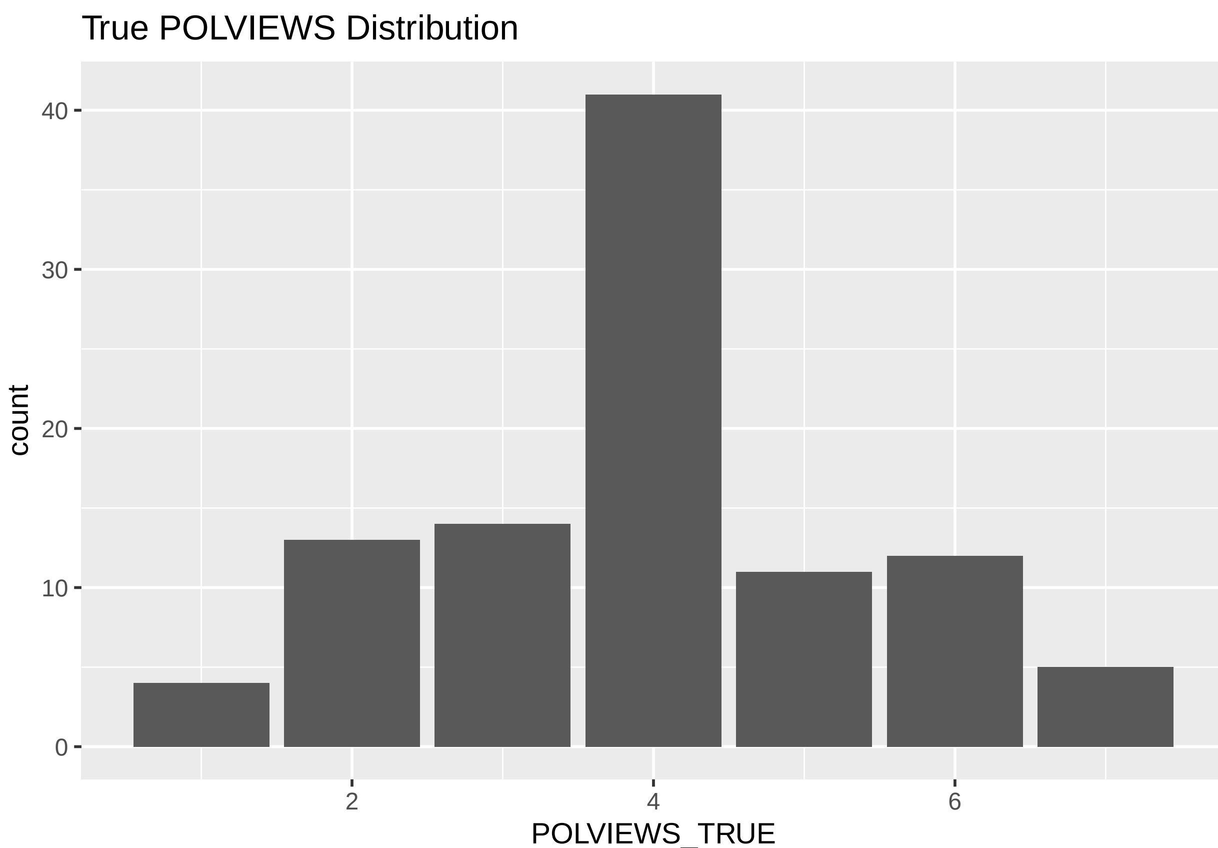
```r
    # Wrong / right flags
    var_wrong  = pred_var  != POLVIEWS_TRUE,
    narr_wrong = pred_narr != POLVIEWS_TRUE,

    # Case types with only two models
    case_type = case_when(
      var_wrong  & !narr_wrong ~ "Only Variable Model Wrong",
      !var_wrong & narr_wrong  ~ "Only Narrative Model Wrong",
      var_wrong  & narr_wrong  ~ "Both Wrong",
      TRUE                     ~ "Both Correct"
    ),

    # Differences vs true (numeric scale 1-7)
    diff_var  = as.numeric(pred_var)  - as.numeric(POLVIEWS_TRUE),
    diff_narr = as.numeric(pred_narr) - as.numeric(POLVIEWS_TRUE),

    # Bias direction for each model (only label as too lib/con if it's wrong)
    bias_var = dplyr::case_when(
      !var_wrong        ~ "Correct",
      diff_var  > 0     ~ "Too Conservative",
      diff_var  < 0     ~ "Too Liberal",
      TRUE              ~ NA_character_
    ),
    bias_narr = dplyr::case_when(
      !narr_wrong       ~ "Correct",
      diff_narr  > 0    ~ "Too Conservative",
      diff_narr  < 0    ~ "Too Liberal",
      TRUE              ~ NA_character_
    )
  ) %>%
  select(
    row_id, POLVIEWS_TRUE,
    pred_var, pred_narr,
    var_wrong, narr_wrong,
    case_type,
    bias_var, bias_narr
  )

# Save to CSV
write.csv(mislabeled_comparison,
          "3_var_mislabeled_cases_comparison.csv",
          row.names = FALSE)
```

```r
bias_table <- mislabeled_comparison %>%
  select(bias_var, bias_narr) %>%
  tidyr::pivot_longer(
    cols      = everything(),
    names_to  = "model",
    values_to = "bias"
  ) %>%
  dplyr::filter(bias != "Correct") %>%   # only mislabeled cases
  dplyr::group_by(model, bias) %>%
  dplyr::summarise(count = dplyr::n(), .groups = "drop_last") %>%
  dplyr::mutate(
```

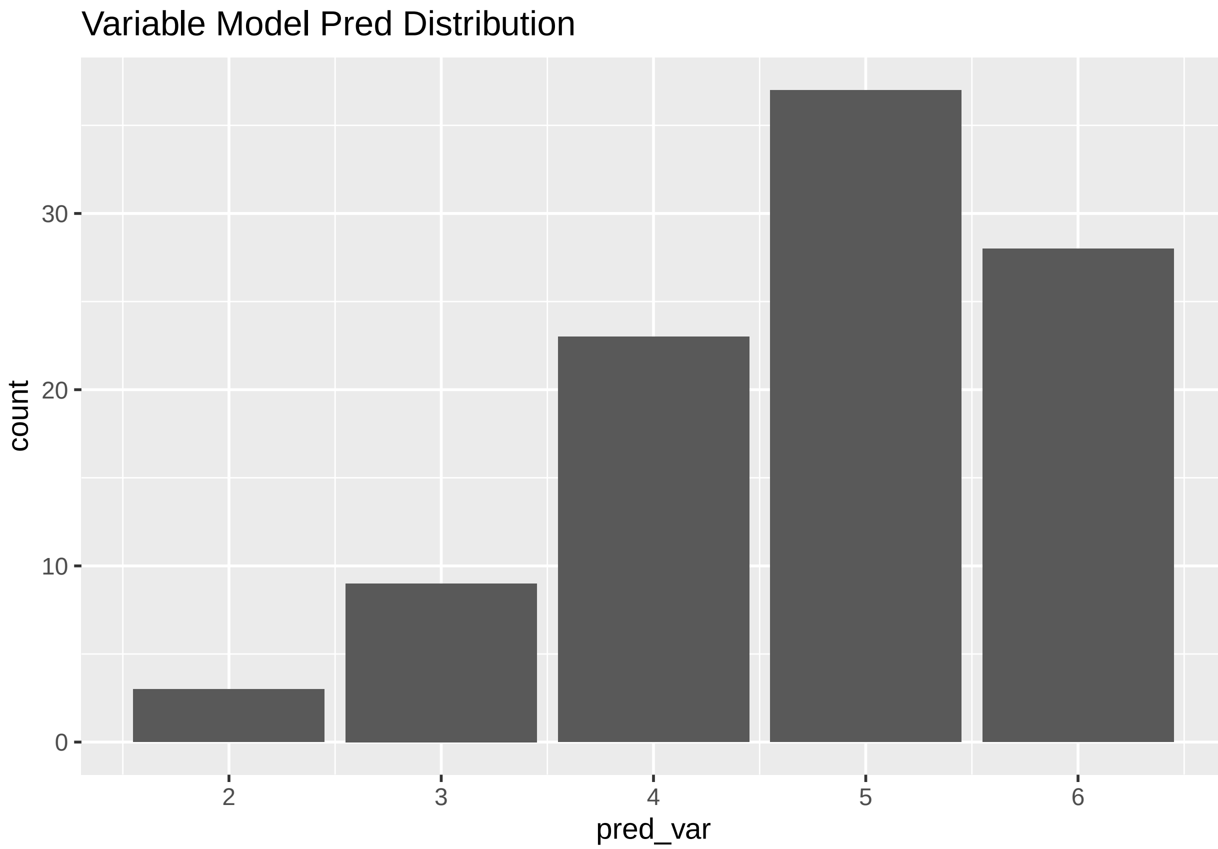
```r
    percent = count / sum(count) * 100
  ) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(
    model = dplyr::recode(
      model,
      bias_var  = "Variable Model",
      bias_narr = "Narrative Model"
    )
  ) %>%
  dplyr::arrange(model, bias)
bias_table
```

```
## # A tibble: 4 x 4
##   model           bias              count percent
##   <chr>           <chr>             <int>   <dbl>
## 1 Narrative Model Too Conservative     39    57.4
## 2 Narrative Model Too Liberal          29    42.6
## 3 Variable Model  Too Conservative     60    68.2
## 4 Variable Model  Too Liberal          28    31.8
```

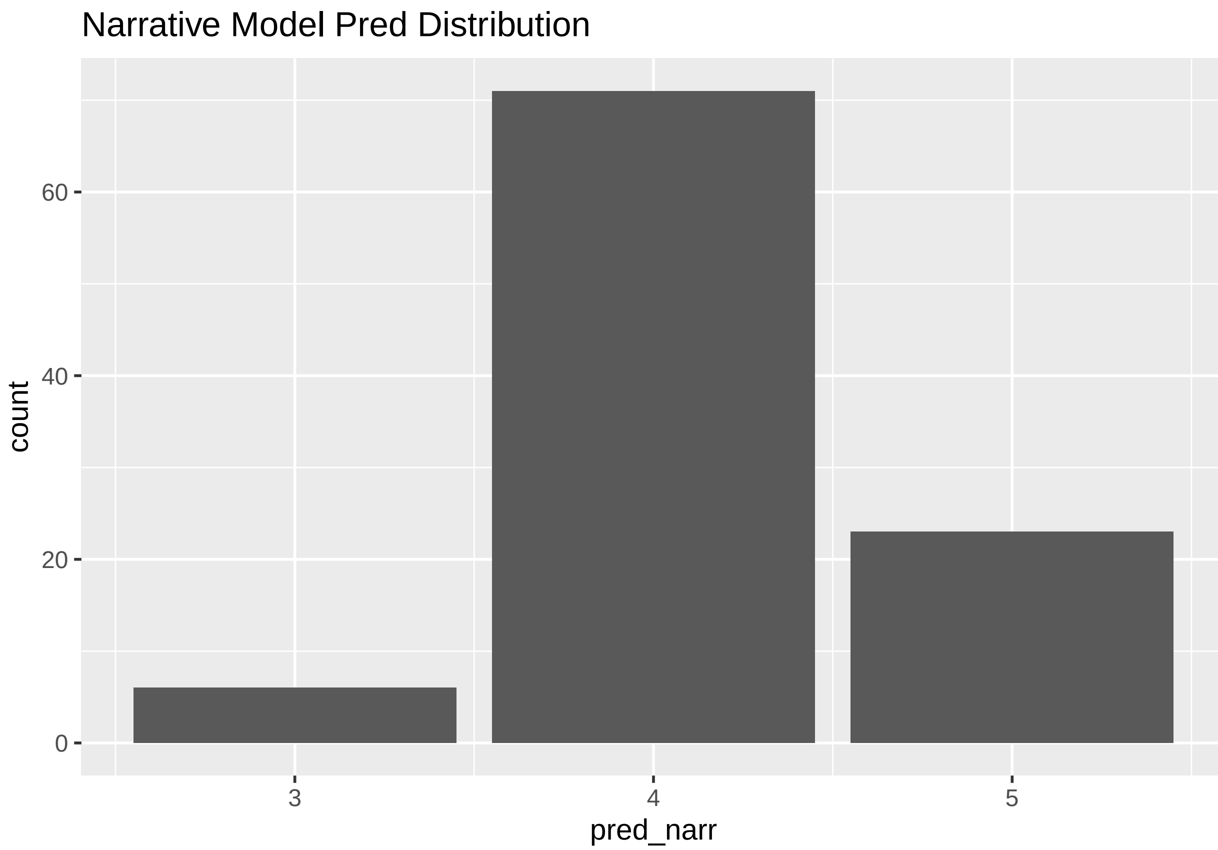
```r
#true polviews distribution
library(ggplot2)

ggplot(df, aes(x = POLVIEWS_TRUE)) +
  geom_bar() +
  ggtitle("True POLVIEWS Distribution")
```



```r
ggplot(df, aes(x = pred_var)) +
  geom_bar() +
  ggtitle("Variable Model Pred Distribution")
```




```r
ggplot(df, aes(x = pred_narr)) +
  geom_bar() +
  ggtitle("Narrative Model Pred Distribution")
```




```r
library(dplyr)

df <- df %>%
  mutate(
    POLVIEWS_TRUE = as.numeric(as.character(POLVIEWS_TRUE)),
    pred_var      = as.numeric(as.character(pred_var)),
    pred_narr     = as.numeric(as.character(pred_narr))
  )
df <- df %>%
  mutate(
    error_var  = pred_var  - POLVIEWS_TRUE,
    error_narr = pred_narr - POLVIEWS_TRUE
  )

summary(df$error_var)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    -4.0    -1.0     1.0     0.8     2.0     5.0
```

```r
summary(df$error_narr)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -3.00   -1.00    0.00    0.19    1.00    4.00
```

```r
mean(df$error_var,  na.rm = TRUE)  # > 0 => too conservative on average
```

```
## [1] 0.8
```

```r
mean(df$error_narr, na.rm = TRUE)
```

```
## [1] 0.19
```

```r
bias_by_predictor <- function(data, predictor) {
  data %>%
    group_by({{ predictor }}) %>%
    summarise(
      n = n(),
      mean_error_var  = mean(error_var, na.rm = TRUE),
      mean_error_narr = mean(error_narr, na.rm = TRUE),

      prop_too_cons_var = mean(error_var  > 0, na.rm = TRUE),
      prop_too_lib_var  = mean(error_var  < 0, na.rm = TRUE),

      prop_too_cons_narr = mean(error_narr > 0, na.rm = TRUE),
      prop_too_lib_narr  = mean(error_narr < 0, na.rm = TRUE)
    ) %>%
    arrange(desc(mean_error_var))
}
bias_by_predictor(df, age)
```

```
## # A tibble: 53 x 8
##      age     n mean_error_var mean_error_narr prop_too_cons_var prop_too_lib_var
##    <dbl> <int>          <dbl>           <dbl>             <dbl>            <dbl>
##  1 20        1            4              3                  1                  0
##  2 26        1            4              3                  1                  0
##  3 42        1            3              2                  1                  0
##  4 47        3            3              2.33               1                  0
##  5 58        1            3              2                  1                  0
##  6 75        1            3              2                  1                  0
##  7 44        2            2.5            1.5                1                  0
##  8 30        1            2              0                  1                  0
##  9 35        2            2              1                  1                  0
## 10 36        2            2              1.5                0.5                0
## # i 43 more rows
## # i 2 more variables: prop_too_cons_narr <dbl>, prop_too_lib_narr <dbl>
```

```r
bias_by_predictor(df, sex)
```

```
## # A tibble: 2 x 8
##   sex       n mean_error_var mean_error_narr prop_too_cons_var prop_too_lib_var
##   <fct> <int>          <dbl>           <dbl>             <dbl>            <dbl>
## 1 1        48          0.854           0.188             0.688            0.25
## 2 2        52          0.75            0.192             0.519            0.308
## # i 2 more variables: prop_too_cons_narr <dbl>, prop_too_lib_narr <dbl>
```

```r
bias_by_predictor(df, race)
```

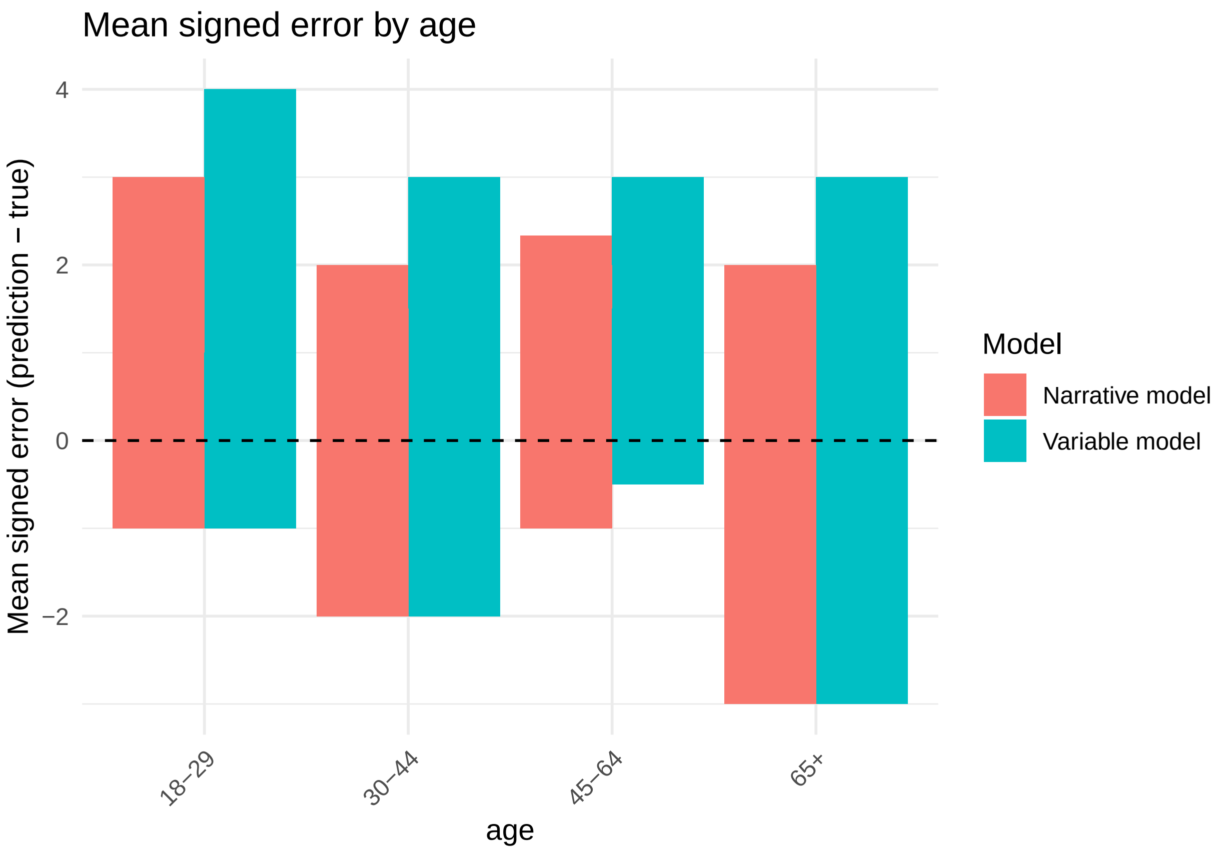
```
## # A tibble: 3 x 8
##   race      n mean_error_var mean_error_narr prop_too_cons_var prop_too_lib_var
##   <fct> <int>          <dbl>           <dbl>             <dbl>            <dbl>
## 1 2        11          1               0.273             0.818           0.0909
## 2 1        78          0.795           0.179             0.577           0.308
## 3 3        11          0.636           0.182             0.545           0.273
## # i 2 more variables: prop_too_cons_narr <dbl>, prop_too_lib_narr <dbl>
```

```
#when mean error > 0, this predictor is more conservative on average
#prop_too_cons_var: proportion of cases where variable model is too conservative
```

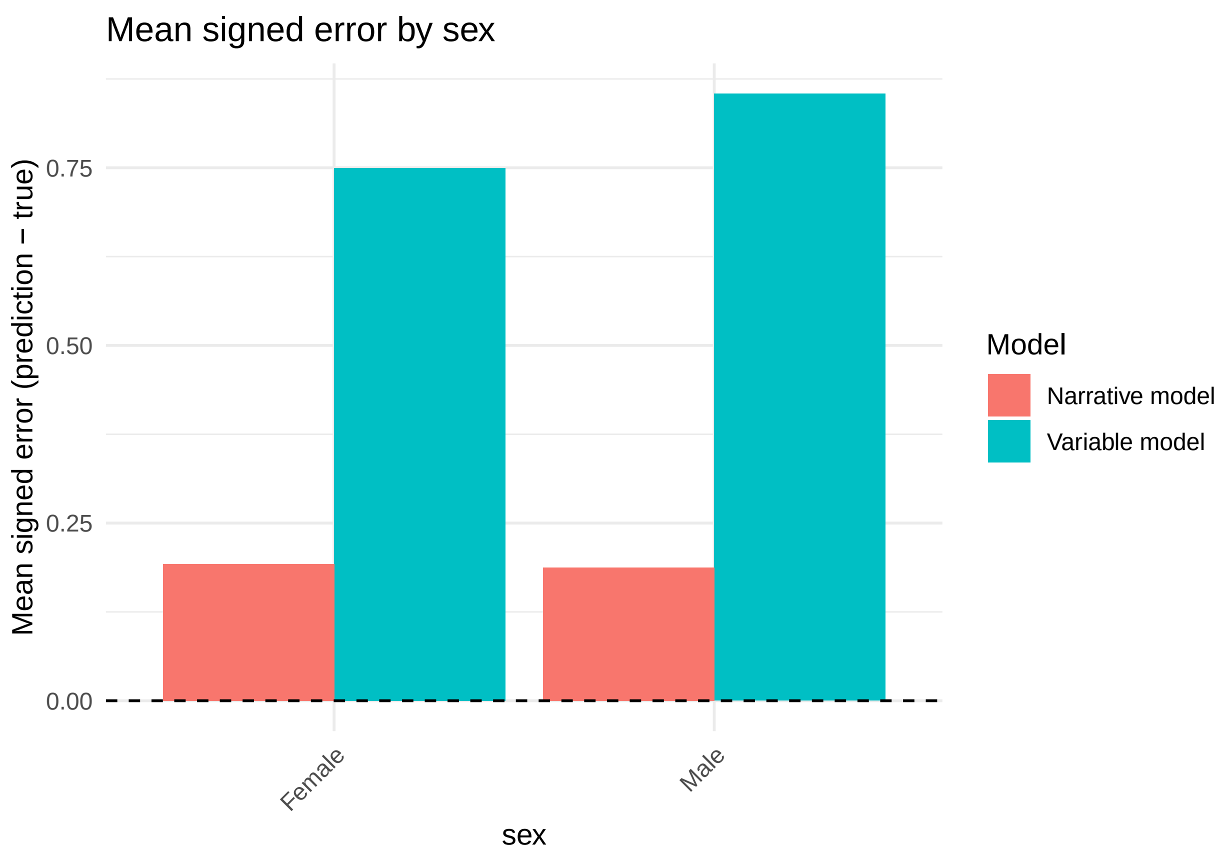
```
label_maps <- list(

  # ---- Gender ----
  sex = c(
    "1" = "Male",
    "2" = "Female"
  ),

  # ---- Race ----
  race = c(
    "1" = "White",
    "2" = "Black",
    "3" = "Other"
  )
)

bucket_age <- function(a) {
  dplyr::case_when(
    is.na(a)           ~ NA_character_,
    a < 30             ~ "18-29",
    a >= 30 & a < 45   ~ "30-44",
    a >= 45 & a < 65   ~ "45-64",
    a >= 65            ~ "65+",
    TRUE ~ NA_character_
  )
}

plot_mean_error_by_predictor <- function(data, predictor) {

  pred_sym  <- rlang::ensym(predictor)
  pred_name <- rlang::as_name(pred_sym)

  summary_df <- data %>%
    dplyr::group_by(!!pred_sym) %>%
    dplyr::summarise(
      n = dplyr::n(),
      mean_error_var  = mean(error_var,  na.rm = TRUE),
      mean_error_narr = mean(error_narr, na.rm = TRUE),
      .groups = "drop"
    ) %>%
    tidyr::pivot_longer(
      cols = c(mean_error_var, mean_error_narr),
      names_to = "model",
      values_to = "mean_error"
    ) %>%
    dplyr::mutate(
      model = dplyr::recode(
        model,
        mean_error_var  = "Variable model",
        mean_error_narr = "Narrative model"
      )
```

```
  )

# Now add human-readable labels
if (pred_name == "occ10") {

  summary_df <- summary_df %>%
    dplyr::mutate(
      predictor_label = vapply(.data[[pred_name]], map_occ10, character(1))
    )

} else if (pred_name == "age") {

  # use age buckets instead of raw ages
  summary_df <- summary_df %>%
    dplyr::mutate(
      predictor_label = bucket_age(.data[[pred_name]])
    )
} else if (pred_name == "educ") {

  summary_df <- summary_df %>%
    dplyr::mutate(
      predictor_label = factor(
        as.numeric(.data[[pred_name]]),
        levels = sort(unique(as.numeric(.data[[pred_name]])))
      )
    )

} else if (pred_name %in% names(label_maps)) {

  map_vec <- label_maps[[pred_name]]

  summary_df <- summary_df %>%
    dplyr::mutate(
      predictor_label = map_vec[as.character(.data[[pred_name]])]
    )

} else {

  summary_df <- summary_df %>%
    dplyr::mutate(
      predictor_label = as.character(.data[[pred_name]])
    )
}

ggplot(summary_df,
       aes(x = predictor_label,
           y = mean_error,
           fill = model)) +
  geom_col(position = "dodge") +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(
    title = paste("Mean signed error by", pred_name),
    x = pred_name,
```

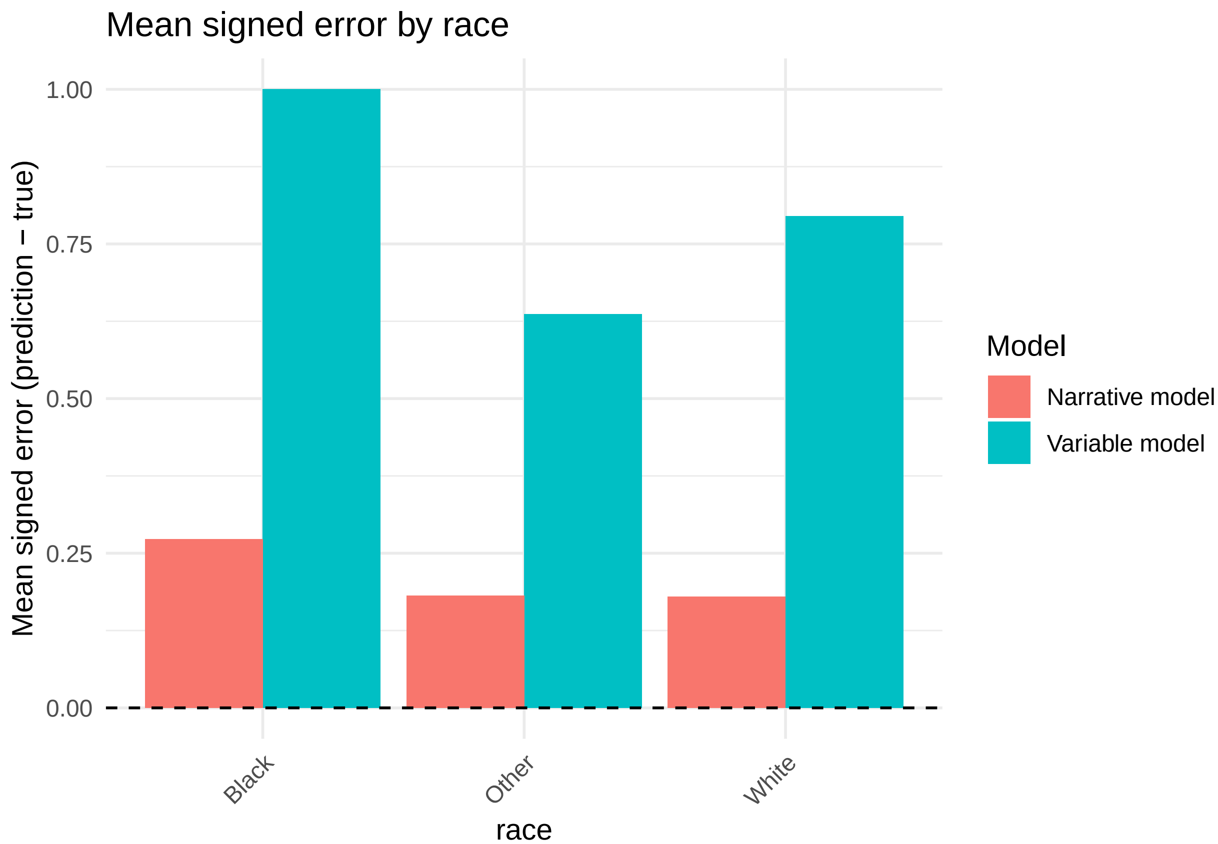
```r
      y = "Mean signed error (prediction - true)",
      fill = "Model"
    ) +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}

plot_mean_error_by_predictor(df, age)
```




```r
plot_mean_error_by_predictor(df, sex)
```




```
plot_mean_error_by_predictor(df, race)
```




```
#collapse POLVIEWS into two categories: conservative or not conservative
sample100_binary <- sample100 %>%
  mutate(
    polviews_binary = case_when(
      polviews %in% c(1, 2, 3, 4) ~ 0,    # Not conservative
      polviews %in% c(5, 6, 7) ~ 1,    # Conservative
    )
  ) %>%
  filter(!is.na(polviews_binary))
head(sample100_binary)
```

```
## # A tibble: 6 x 5
##   polviews age       race  sex   polviews_binary
##      <int> <dbl+lbl> <fct> <fct>           <dbl>
## 1        3 59        1     1                   0
## 2        4 52        1     2                   0
## 3        6 61        1     1                   1
## 4        4 45        1     2                   0
## 5        4 28        3     1                   0
## 6        4 62        1     2                   0
```

```
sample100_nolabel_bin <- sample100_binary %>%
  select(-polviews_binary) %>% # remove the binary ideology variable)
  select(-polviews) # remove the numeric ideology variable


head(sample100_nolabel_bin)
```

```
## # A tibble: 6 x 3
##   age       race  sex
##   <dbl+lbl> <fct> <fct>
## 1 59        1     1
## 2 52        1     2
## 3 61        1     1
## 4 45        1     2
## 5 28        3     1
## 6 62        1     2
```

```
write.csv(sample100_nolabel_bin, "3_var_gss_sample_100_unlabeled_bin.csv", row.names = FALSE)
```

```
var_bin <- read.csv("/Users/joyqu/Desktop/PLSC/3_var/3_var_gss_gpt5_var_predictions_bin.csv")
head(var_bin)
```

```
##   age race sex pred_polview
## 1  59    1   1            1
## 2  52    1   2            1
## 3  61    1   1            1
## 4  45    1   2            1
## 5  28    3   1            0
## 6  62    1   2            1
```

```
# Extract variables
y_true_bin <- as.numeric(sample100_binary$polviews_binary)
y_pred_bin <- as.numeric(var_bin$pred_polview)

# Compute metrics
MAE <- mean(abs(y_true_bin - y_pred_bin))
MSE <- mean((y_true_bin - y_pred_bin)^2)
Accuracy <- mean(y_true_bin == y_pred_bin)
Within1 <- mean(abs(y_true_bin - y_pred_bin) <= 1)

cat("Mean Absolute Error:", MAE, "\n")
```

```
## Mean Absolute Error: 0.6
```

```
cat("Mean Squared Error:", MSE, "\n")
```

```
## Mean Squared Error: 0.6
```

```
cat("Exact Match Accuracy:", round(Accuracy*100, 1), "%\n")
```

```
## Exact Match Accuracy: 40 %
```

```
cat("Within ±1 Accuracy:", round(Within1*100, 1), "%\n")
```

```
## Within ±1 Accuracy: 100 %
```

```
narrative_bin <- read.csv("/Users/joyqu/Desktop/PLSC/3_var/3_var_gss_gpt5_narrative_predictions_bin.csv
head(narrative_bin)
```

```
##
## 1                          67 years old, this white man has settled into a steady rhythm of daily life.
## 2 56 years old, this from a diverse background woman has settled into a steady rhythm of daily life.
## 3                        33 years old, this white woman has settled into a steady rhythm of daily life.
## 4                        24 years old, this white woman has settled into a steady rhythm of daily life.
## 5                        46 years old, this white woman has settled into a steady rhythm of daily life.
## 6                          25 years old, this white man has settled into a steady rhythm of daily life.
```

```
##   pred_polview_narr
## 1                 1
## 2                 0
## 3                 0
## 4                 0
## 5                 0
## 6                 1
```

```
# Extract variables
y_true_bin <- as.numeric(sample100_binary$polviews_binary)
y_pred_bin <- as.numeric(narrative_bin$pred_polview_narr)

# Compute metrics
MAE <- mean(abs(y_true_bin - y_pred_bin))
MSE <- mean((y_true_bin - y_pred_bin)^2)
Accuracy <- mean(y_true_bin == y_pred_bin)
Within1 <- mean(abs(y_true_bin - y_pred_bin) <= 1)

cat("Mean Absolute Error:", MAE, "\n")
```

```
## Mean Absolute Error: 0.58
```

```
cat("Mean Squared Error:", MSE, "\n")
```

```
## Mean Squared Error: 0.58
```

```
cat("Exact Match Accuracy:", round(Accuracy*100, 1), "%\n")
```

```
## Exact Match Accuracy: 42 %
```

```
cat("Within ±1 Accuracy:", round(Within1*100, 1), "%\n")
```

```
## Within ±1 Accuracy: 100 %
```

```
df_bin <- sample100_binary %>%
  mutate(row_id = row_number()) %>%
  select(
    row_id,
    POLVIEWS_TRUE = polviews_binary,
    age, sex, race   # <- keep whatever predictors you want
  ) %>%
  inner_join(
    var_bin %>%
      mutate(row_id = row_number()) %>%
      select(row_id, pred_var = pred_polview),
    by = "row_id"
  ) %>%
  inner_join(
    narrative_bin %>%
      mutate(row_id = row_number()) %>%
      select(row_id, pred_narr = pred_polview_narr),
    by = "row_id"
  )
head(df_bin)
```

```
## # A tibble: 6 x 7
##   row_id POLVIEWS_TRUE age       sex   race  pred_var pred_narr
##    <int>         <dbl> <dbl+lbl> <fct> <fct>    <int>     <int>
```

```
## 1      1              0 59        1     1            1          1
## 2      2              0 52        2     1            1          0
## 3      3              1 61        1     1            1          0
## 4      4              0 45        2     1            1          0
## 5      5              0 28        1     3            0          0
## 6      6              0 62        2     1            1          1
```

```r
df_bin <- df_bin %>%
  mutate(
    # Factor version for F1
    POLVIEWS_TRUE_fac = factor(POLVIEWS_TRUE),
    pred_var_fac      = factor(pred_var,  levels = levels(POLVIEWS_TRUE_fac)),
    pred_narr_fac     = factor(pred_narr, levels = levels(POLVIEWS_TRUE_fac)),

    # Numeric version for bias / error
    polviews_num = as.numeric(as.character(POLVIEWS_TRUE)),
    pred_var_num = as.numeric(as.character(pred_var)),
    pred_narr_num = as.numeric(as.character(pred_narr)),

    # Signed errors
    error_var  = pred_var_num  - polviews_num,
    error_narr = pred_narr_num - polviews_num
  )
results <- tibble(
  Model = c("Variable Model", "Narrative Model"),
  Macro_F1 = c(
    f1_macro(df_bin$POLVIEWS_TRUE_fac, df_bin$pred_var_fac),
    f1_macro(df_bin$POLVIEWS_TRUE_fac, df_bin$pred_narr_fac)
  ),
  Weighted_F1 = c(
    f1_weighted(df_bin$POLVIEWS_TRUE_fac, df_bin$pred_var_fac),
    f1_weighted(df_bin$POLVIEWS_TRUE_fac, df_bin$pred_narr_fac)
  )
)

print(results)
```

```
## # A tibble: 2 x 3
##   Model           Macro_F1 Weighted_F1
##   <chr>              <dbl>       <dbl>
## 1 Variable Model     0.394       0.421
## 2 Narrative Model    0.405       0.363
```

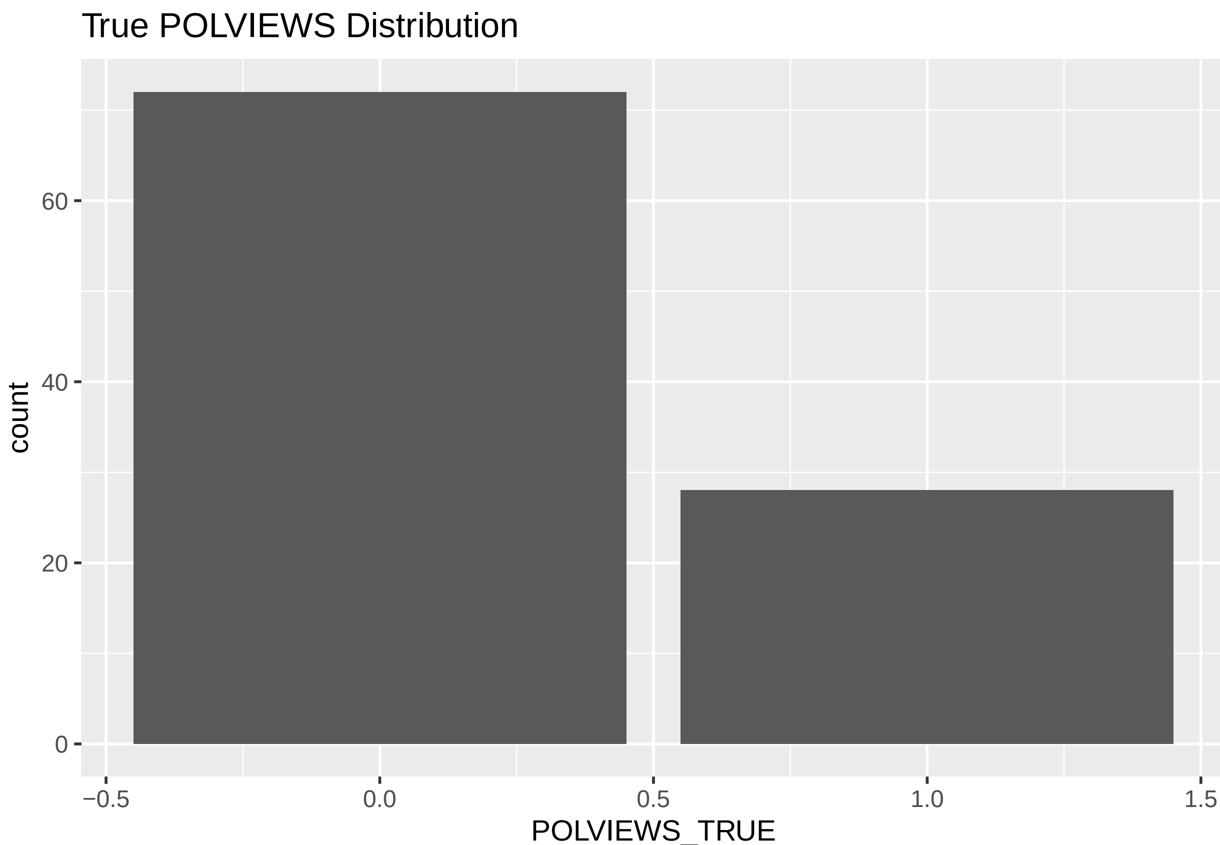
```r
mislabeled_comparison <- df_bin %>%
  mutate(
    # Wrong / right flags
    var_wrong  = pred_var  != POLVIEWS_TRUE,
    narr_wrong = pred_narr != POLVIEWS_TRUE,

    # Case types with only two models
    case_type = case_when(
      var_wrong  & !narr_wrong ~ "Only Variable Model Wrong",
      !var_wrong & narr_wrong  ~ "Only Narrative Model Wrong",
      var_wrong  & narr_wrong  ~ "Both Wrong",
      TRUE                     ~ "Both Correct"
```

```r
    ),

    # Differences vs true (numeric scale 1-7)
    diff_var  = as.numeric(pred_var)  - as.numeric(POLVIEWS_TRUE),
    diff_narr = as.numeric(pred_narr) - as.numeric(POLVIEWS_TRUE),

    # Bias direction for each model (only label as too lib/con if it's wrong)
    bias_var = dplyr::case_when(
      !var_wrong         ~ "Correct",
      diff_var  > 0      ~ "Too Conservative",
      diff_var  < 0      ~ "Too Liberal",
      TRUE               ~ NA_character_
    ),
    bias_narr = dplyr::case_when(
      !narr_wrong        ~ "Correct",
      diff_narr  > 0     ~ "Too Conservative",
      diff_narr  < 0     ~ "Too Liberal",
      TRUE               ~ NA_character_
    )
  ) %>%
  select(
    row_id, POLVIEWS_TRUE,
    pred_var, pred_narr,
    var_wrong, narr_wrong,
    case_type,
    bias_var, bias_narr
  )

# Save to CSV
write.csv(mislabeled_comparison,
          "3_var_mislabeled_cases_comparison_bin.csv",
          row.names = FALSE)

bias_table <- mislabeled_comparison %>%
  select(bias_var, bias_narr) %>%
  tidyr::pivot_longer(
    cols      = everything(),
    names_to  = "model",
    values_to = "bias"
  ) %>%
  dplyr::filter(bias != "Correct") %>%   # only mislabeled cases
  dplyr::group_by(model, bias) %>%
  dplyr::summarise(count = dplyr::n(), .groups = "drop_last") %>%
  dplyr::mutate(
    percent = count / sum(count) * 100
  ) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(
    model = dplyr::recode(
      model,
      bias_var  = "Variable Model",
      bias_narr = "Narrative Model"
    )
```

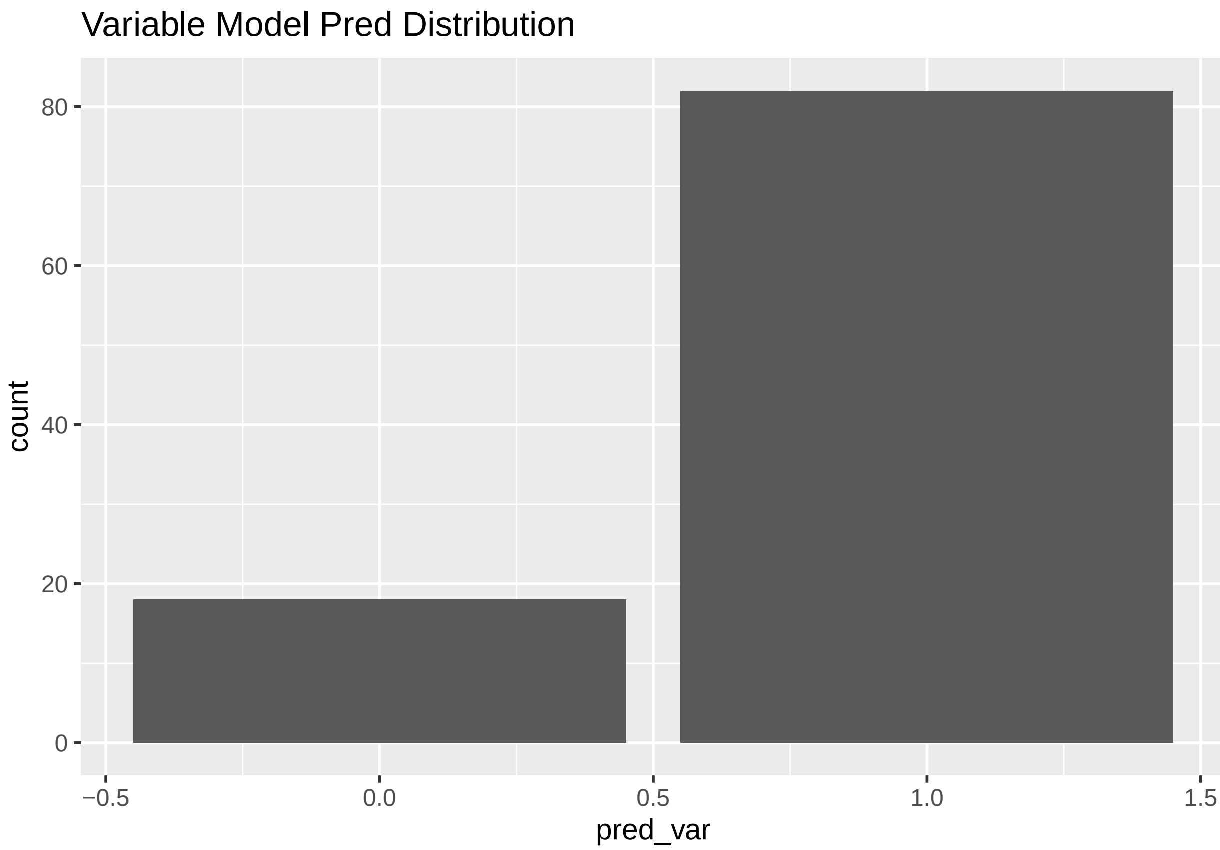
```r
  ) %>%
  dplyr::arrange(model, bias)
bias_table
```

```
## # A tibble: 4 x 4
##   model           bias              count percent
##   <chr>           <chr>             <int>   <dbl>
## 1 Narrative Model Too Conservative     43    74.1
## 2 Narrative Model Too Liberal          15    25.9
## 3 Variable Model  Too Conservative     57    95
## 4 Variable Model  Too Liberal           3     5
```

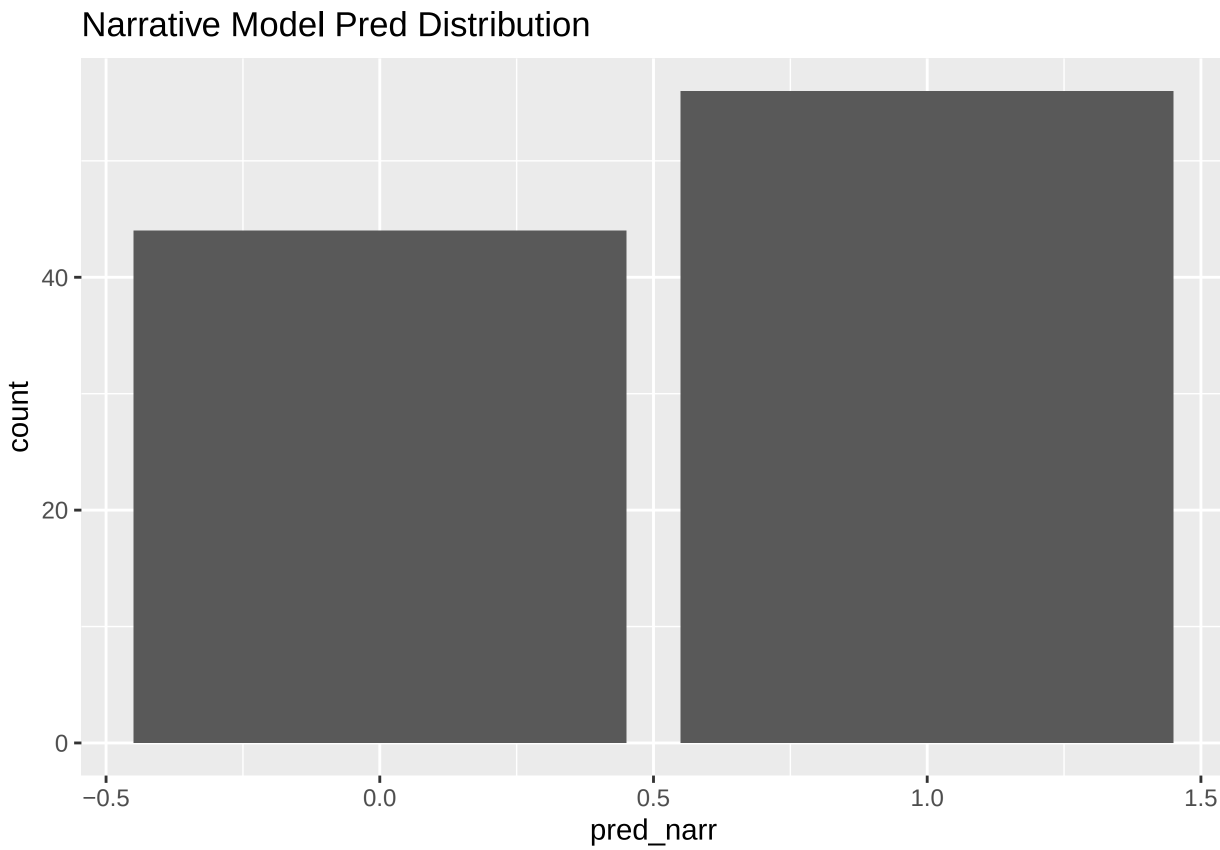
```r
#true polviews distribution

ggplot(df_bin, aes(x = POLVIEWS_TRUE)) +
  geom_bar() +
  ggtitle("True POLVIEWS Distribution")
```




```r
ggplot(df_bin, aes(x = pred_var)) +
  geom_bar() +
  ggtitle("Variable Model Pred Distribution")
```




```r
ggplot(df_bin, aes(x = pred_narr)) +
  geom_bar() +
  ggtitle("Narrative Model Pred Distribution")
```




```
bias_by_predictor(df_bin, age)
```

```
## # A tibble: 53 x 8
##      age     n mean_error_var mean_error_narr prop_too_cons_var prop_too_lib_var
##    <dbl> <int>          <dbl>           <dbl>             <dbl>            <dbl>
##  1 18        1              1           1                     1                0
##  2 26        1              1           1                     1                0
##  3 30        1              1           1                     1                0
##  4 31        2              1           0.5                   1                0
##  5 32        2              1           0                     1                0
##  6 39        1              1           0                     1                0
##  7 42        1              1           0                     1                0
##  8 47        3              1           0.667                 1                0
##  9 49        2              1           0.5                   1                0
## 10 52        1              1           0                     1                0
## # i 43 more rows
## # i 2 more variables: prop_too_cons_narr <dbl>, prop_too_lib_narr <dbl>
```

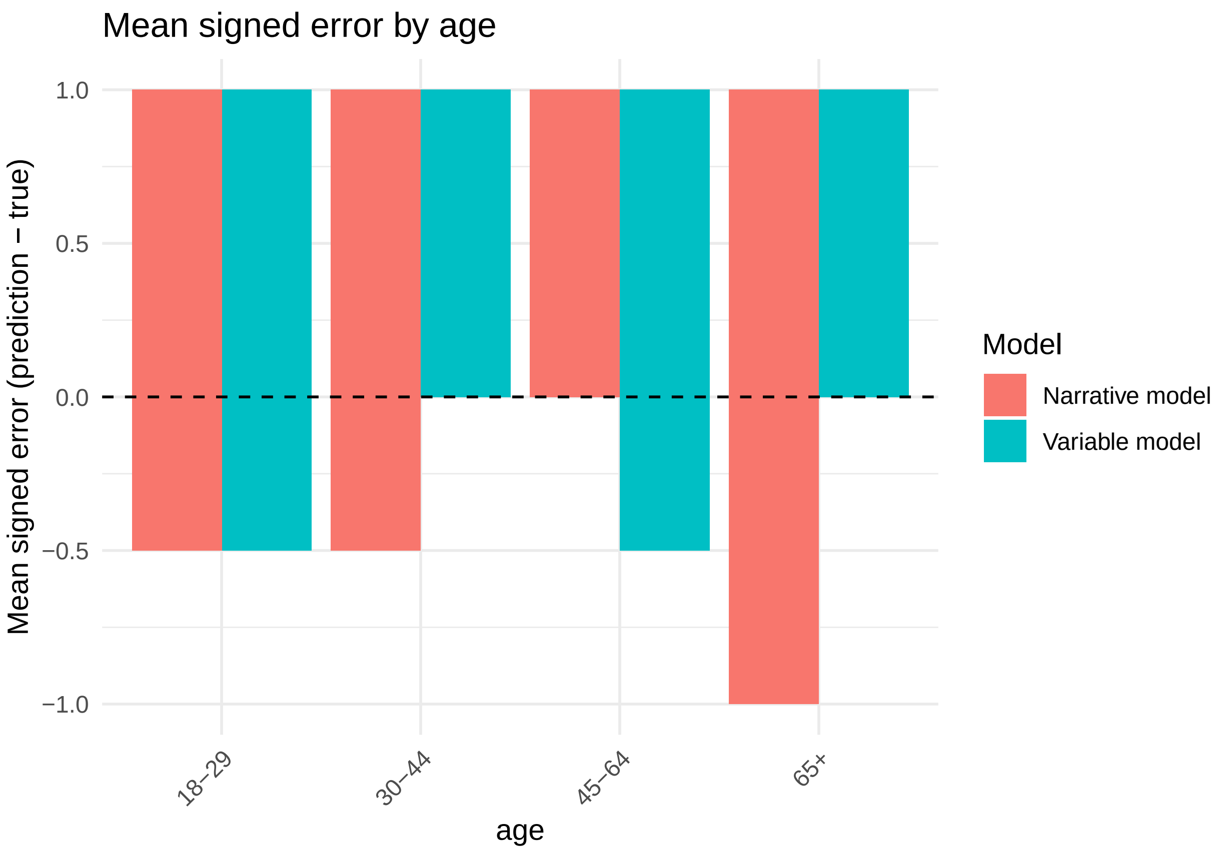
```
bias_by_predictor(df_bin, sex)
```

```
## # A tibble: 2 x 8
##   sex       n mean_error_var mean_error_narr prop_too_cons_var prop_too_lib_var
##   <fct> <int>          <dbl>           <dbl>             <dbl>            <dbl>
## 1 1        48          0.625           0.333             0.667           0.0417
## 2 2        52          0.462           0.231             0.481           0.0192
## # i 2 more variables: prop_too_cons_narr <dbl>, prop_too_lib_narr <dbl>
```

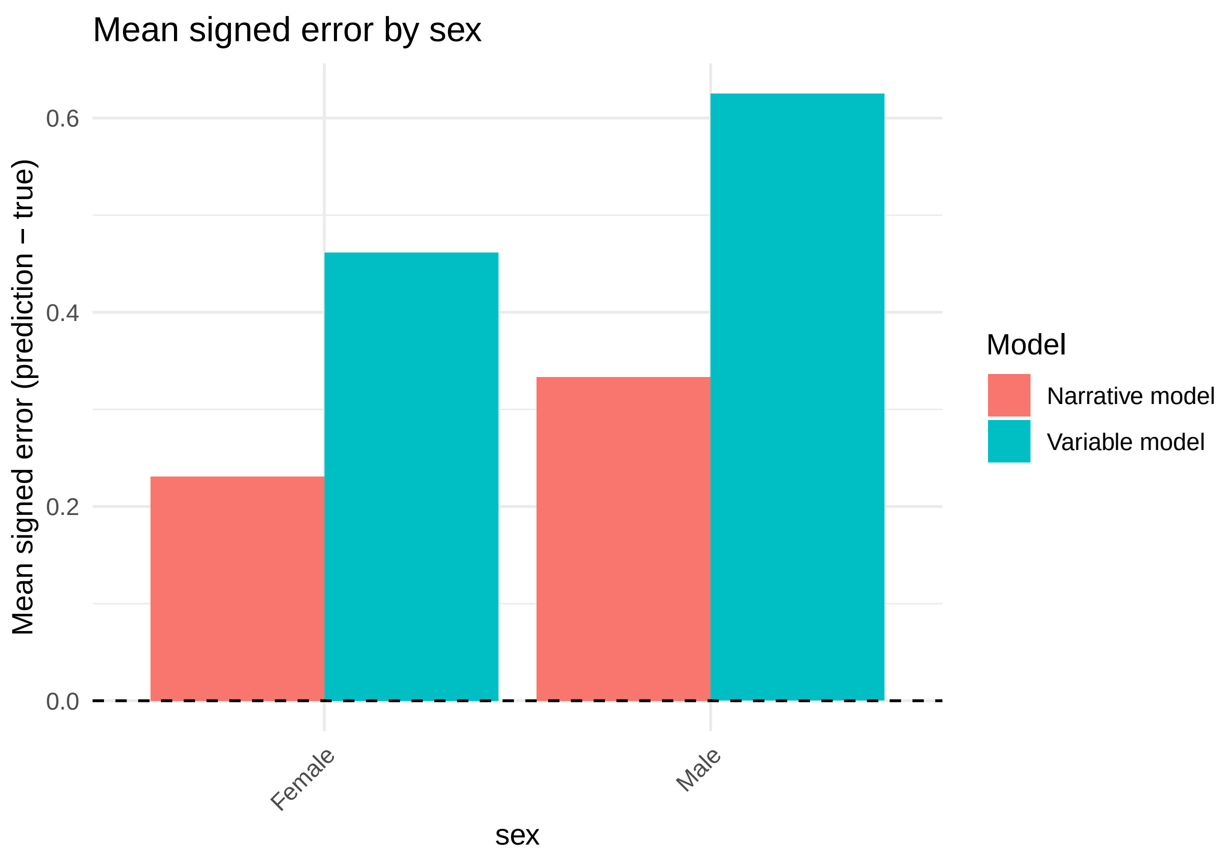
```r
bias_by_predictor(df_bin, race)
```

```
## # A tibble: 3 x 8
##   race      n mean_error_var mean_error_narr prop_too_cons_var prop_too_lib_var
##   <fct> <int>          <dbl>           <dbl>             <dbl>            <dbl>
## 1 1        78          0.641           0.244             0.654           0.0128
## 2 3        11          0.182           0.455             0.273           0.0909
## 3 2        11          0.182           0.364             0.273           0.0909
## # i 2 more variables: prop_too_cons_narr <dbl>, prop_too_lib_narr <dbl>
```

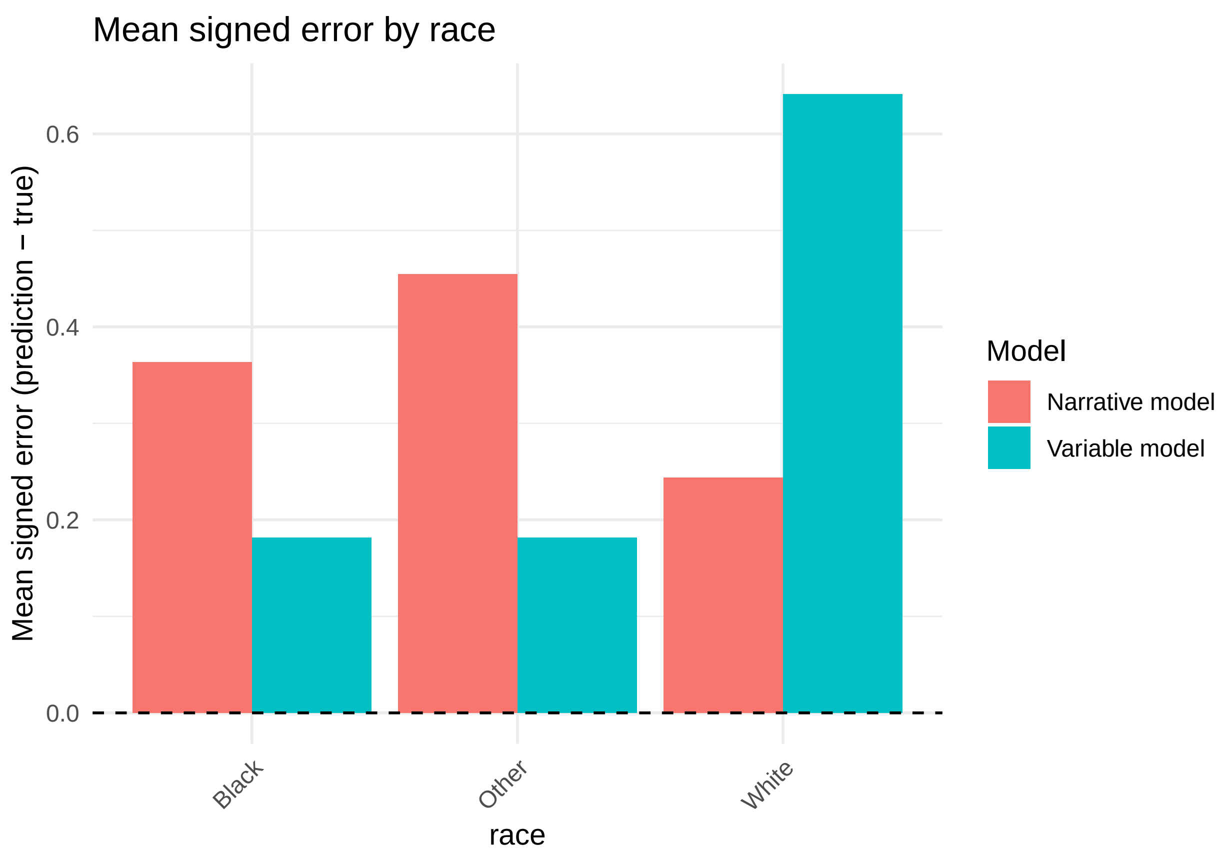
```r
plot_mean_error_by_predictor(df_bin, age)
```




```r
plot_mean_error_by_predictor(df_bin, sex)
```




```r
plot_mean_error_by_predictor(df_bin, race)
```




```
#collapse POLVIEWS into three categories: 1 = Liberal, 2 = Moderate, 3 = Conservative
sample100_3 <- sample100 %>%
  mutate(
    polviews_3 = case_when(
      polviews %in% c(1, 2, 3) ~ 1,    # liberal
      polviews %in% c(4) ~ 2,    # moderate
      polviews %in% c(5, 6, 7) ~ 3     # conservative
    )
  ) %>%
  filter(!is.na(polviews_3))
head(sample100_3)
```

```
## # A tibble: 6 x 5
##   polviews age       race  sex   polviews_3
##      <int> <dbl+lbl> <fct> <fct>      <dbl>
## 1        3 59        1     1              1
## 2        4 52        1     2              2
## 3        6 61        1     1              3
## 4        4 45        1     2              2
## 5        4 28        3     1              2
## 6        4 62        1     2              2
```

```
sample100_nolabel_3 <- sample100_3 %>%
  select(-polviews_3) %>% # remove the binary ideology variable)
  select(-polviews) # remove the numeric ideology variable


head(sample100_nolabel_3)
```

```
## # A tibble: 6 x 3
##   age       race  sex
##   <dbl+lbl> <fct> <fct>
## 1 59        1     1
## 2 52        1     2
## 3 61        1     1
## 4 45        1     2
## 5 28        3     1
## 6 62        1     2
```

```
write.csv(sample100_nolabel_3, "3_var_gss_sample_100_unlabeled_3.csv", row.names = FALSE)
```

```
var_3 <- read.csv("/Users/joyqu/Desktop/PLSC/3_var/3_var_gss_gpt5_var_predictions_3.csv")
head(var_3)
```

```
##   age race sex pred_polview
## 1  59    1   1            3
## 2  52    1   2            3
## 3  61    1   1            3
## 4  45    1   2            3
## 5  28    3   1            2
## 6  62    1   2            3
```

```
# Extract variables
y_true_3 <- as.numeric(sample100_3$polviews_3)
y_pred_3 <- as.numeric(var_3$pred_polview)

# Compute metrics
MAE <- mean(abs(y_true_3 - y_pred_3))
MSE <- mean((y_true_3 - y_pred_3)^2)
Accuracy <- mean(y_true_3 == y_pred_3)
Within1 <- mean(abs(y_true_3 - y_pred_3) <= 1)

cat("Mean Absolute Error:", MAE, "\n")
```

```
## Mean Absolute Error: 0.94
```

```
cat("Mean Squared Error:", MSE, "\n")
```

```
## Mean Squared Error: 1.5
```

```
cat("Exact Match Accuracy:", round(Accuracy*100, 1), "%\n")
```

```
## Exact Match Accuracy: 34 %
```

```
cat("Within ±1 Accuracy:", round(Within1*100, 1), "%\n")
```

```
## Within ±1 Accuracy: 72 %
```

```
narrative_3 <- read.csv("/Users/joyqu/Desktop/PLSC/3_var/3_var_gss_gpt5_narrative_predictions_3.csv")
head(narrative_3)
```

```
##
## 1                        67 years old, this white man has settled into a steady rhythm of daily life.
## 2 56 years old, this from a diverse background woman has settled into a steady rhythm of daily life.
## 3                      33 years old, this white woman has settled into a steady rhythm of daily life.
## 4                      24 years old, this white woman has settled into a steady rhythm of daily life.
## 5                      46 years old, this white woman has settled into a steady rhythm of daily life.
## 6                        25 years old, this white man has settled into a steady rhythm of daily life.
```

```
##   pred_polview_narr
## 1                 2
## 2                 2
## 3                 2
## 4                 2
## 5                 2
## 6                 2
```

```r
# Extract variables
y_true_3 <- as.numeric(sample100_3$polviews_3)
y_pred_3 <- as.numeric(narrative_3$pred_polview_narr)

# Compute metrics
MAE <- mean(abs(y_true_3 - y_pred_3))
MSE <- mean((y_true_3 - y_pred_3)^2)
Accuracy <- mean(y_true_3 == y_pred_3)
Within1 <- mean(abs(y_true_3 - y_pred_3) <= 1)

cat("Mean Absolute Error:", MAE, "\n")
```

```
## Mean Absolute Error: 0.61
```

```r
cat("Mean Squared Error:", MSE, "\n")
```

```
## Mean Squared Error: 0.63
```

```r
cat("Exact Match Accuracy:", round(Accuracy*100, 1), "%\n")
```

```
## Exact Match Accuracy: 40 %
```

```r
cat("Within ±1 Accuracy:", round(Within1*100, 1), "%\n")
```

```
## Within ±1 Accuracy: 99 %
```

```r
df_3 <- sample100_3 %>%
  mutate(row_id = row_number()) %>%
  select(
    row_id,
    POLVIEWS_TRUE = polviews_3,
    age, sex, race   # <- keep whatever predictors you want
  ) %>%
  inner_join(
    var_3 %>%
      mutate(row_id = row_number()) %>%
      select(row_id, pred_var = pred_polview),
    by = "row_id"
  ) %>%
  inner_join(
    narrative_3 %>%
      mutate(row_id = row_number()) %>%
      select(row_id, pred_narr = pred_polview_narr),
    by = "row_id"
  )
head(df_3)
```

```
## # A tibble: 6 x 7
##   row_id POLVIEWS_TRUE age       sex   race  pred_var pred_narr
##    <int>         <dbl> <dbl+lbl> <fct> <fct>    <int>     <int>
```

```
## 1      1             1 59        1     1            3         2
## 2      2             2 52        2     1            3         2
## 3      3             3 61        1     1            3         2
## 4      4             2 45        2     1            3         2
## 5      5             2 28        1     3            2         2
## 6      6             2 62        2     1            3         2
```

```
df_3 <- df_3 %>%
  mutate(
    # Factor version for F1
    POLVIEWS_TRUE_fac = factor(POLVIEWS_TRUE),
    pred_var_fac      = factor(pred_var,  levels = levels(POLVIEWS_TRUE_fac)),
    pred_narr_fac     = factor(pred_narr, levels = levels(POLVIEWS_TRUE_fac)),

    # Numeric version for bias / error
    polviews_num = as.numeric(as.character(POLVIEWS_TRUE)),
    pred_var_num = as.numeric(as.character(pred_var)),
    pred_narr_num = as.numeric(as.character(pred_narr)),

    # Signed errors
    error_var  = pred_var_num  - polviews_num,
    error_narr = pred_narr_num - polviews_num
  )
results <- tibble(
  Model = c("Variable Model", "Narrative Model"),
  Macro_F1 = c(
    f1_macro(df_3$POLVIEWS_TRUE_fac, df_3$pred_var_fac),
    f1_macro(df_3$POLVIEWS_TRUE_fac, df_3$pred_narr_fac)
  ),
  Weighted_F1 = c(
    f1_weighted(df_3$POLVIEWS_TRUE_fac, df_3$pred_var_fac),
    f1_weighted(df_3$POLVIEWS_TRUE_fac, df_3$pred_narr_fac)
  )
)

print(results)
```

```
## # A tibble: 2 x 3
##   Model           Macro_F1 Weighted_F1
##   <chr>              <dbl>       <dbl>
## 1 Variable Model     0.644       0.660
## 2 Narrative Model    0.567       0.509
```

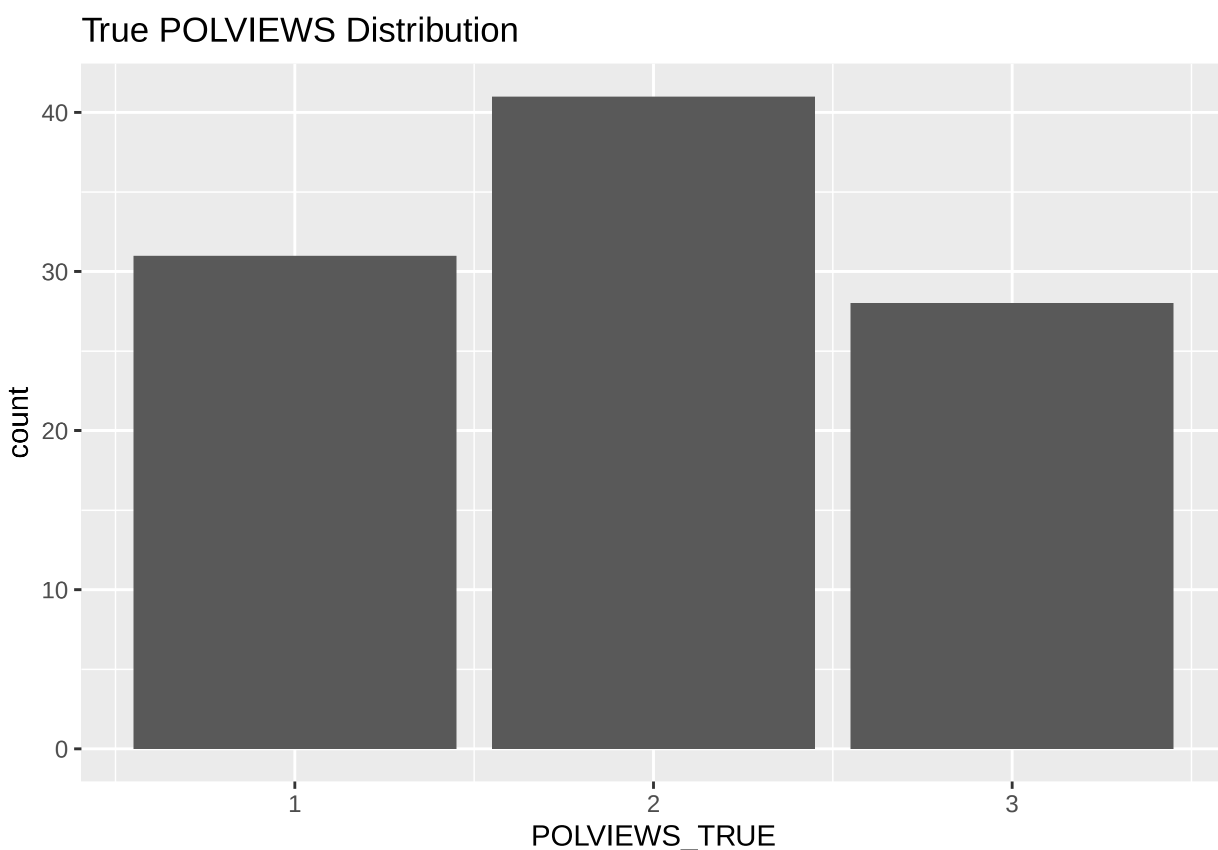
```
mislabeled_comparison <- df_3 %>%
  mutate(
    # Wrong / right flags
    var_wrong  = pred_var  != POLVIEWS_TRUE,
    narr_wrong = pred_narr != POLVIEWS_TRUE,

    # Case types with only two models
    case_type = case_when(
      var_wrong  & !narr_wrong ~ "Only Variable Model Wrong",
      !var_wrong & narr_wrong  ~ "Only Narrative Model Wrong",
      var_wrong  & narr_wrong  ~ "Both Wrong",
      TRUE                     ~ "Both Correct"
```

```
    ),

    # Differences vs true (numeric scale 1-7)
    diff_var  = as.numeric(pred_var)  - as.numeric(POLVIEWS_TRUE),
    diff_narr = as.numeric(pred_narr) - as.numeric(POLVIEWS_TRUE),

    # Bias direction for each model (only label as too lib/con if it's wrong)
    bias_var = dplyr::case_when(
      !var_wrong         ~ "Correct",
      diff_var  > 0      ~ "Too Conservative",
      diff_var  < 0      ~ "Too Liberal",
      TRUE               ~ NA_character_
    ),
    bias_narr = dplyr::case_when(
      !narr_wrong        ~ "Correct",
      diff_narr  > 0     ~ "Too Conservative",
      diff_narr  < 0     ~ "Too Liberal",
      TRUE               ~ NA_character_
    )
  ) %>%
  select(
    row_id, POLVIEWS_TRUE,
    pred_var, pred_narr,
    var_wrong, narr_wrong,
    case_type,
    bias_var, bias_narr
  )

# Save to CSV
write.csv(mislabeled_comparison,
          "3_var_mislabeled_cases_comparison_3.csv",
          row.names = FALSE)

bias_table <- mislabeled_comparison %>%
  select(bias_var, bias_narr) %>%
  tidyr::pivot_longer(
    cols      = everything(),
    names_to  = "model",
    values_to = "bias"
  ) %>%
  dplyr::filter(bias != "Correct") %>%   # only mislabeled cases
  dplyr::group_by(model, bias) %>%
  dplyr::summarise(count = dplyr::n(), .groups = "drop_last") %>%
  dplyr::mutate(
    percent = count / sum(count) * 100
  ) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(
    model = dplyr::recode(
      model,
      bias_var  = "Variable Model",
      bias_narr = "Narrative Model"
    )
```

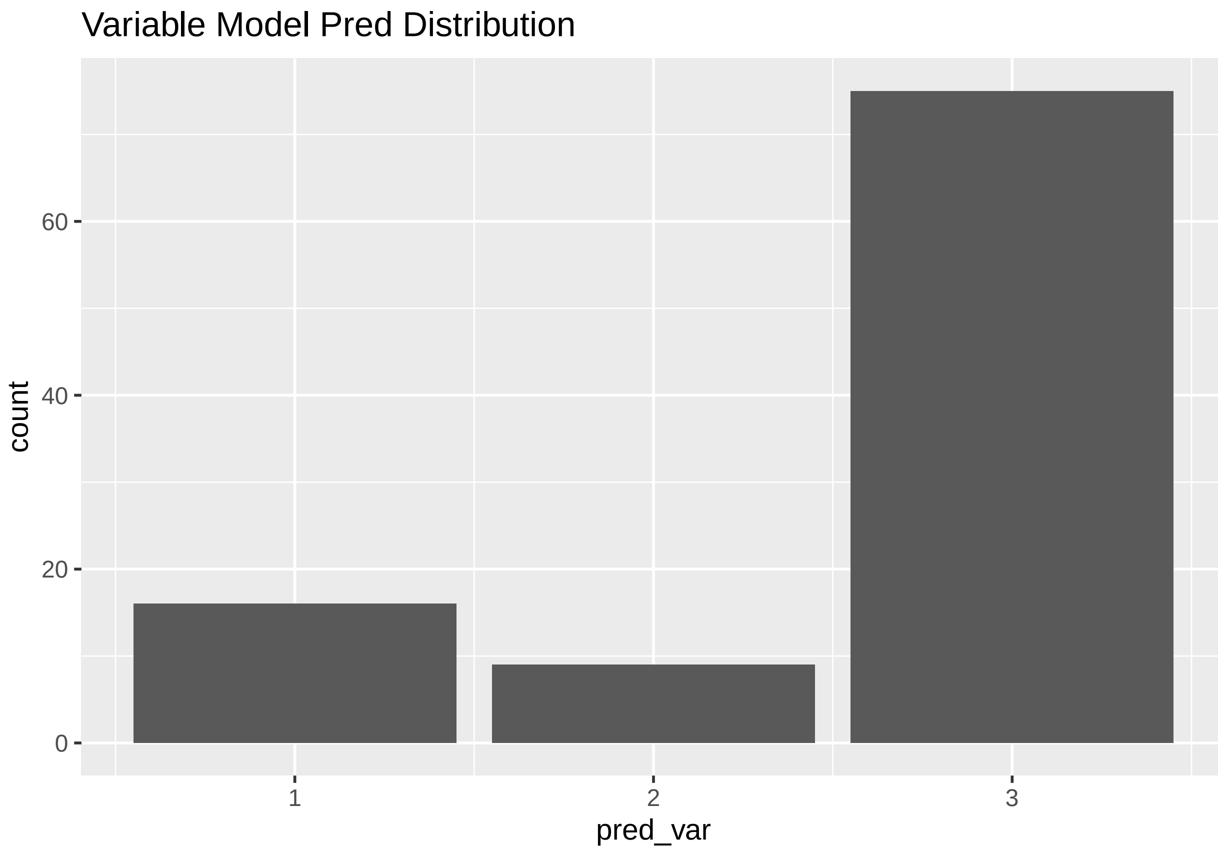
```r
  ) %>%
  dplyr::arrange(model, bias)
bias_table
```

```
## # A tibble: 4 x 4
##   model           bias             count percent
##   <chr>           <chr>            <int>   <dbl>
## 1 Narrative Model Too Conservative    33    55
## 2 Narrative Model Too Liberal         27    45
## 3 Variable Model  Too Conservative    52    78.8
## 4 Variable Model  Too Liberal         14    21.2
```

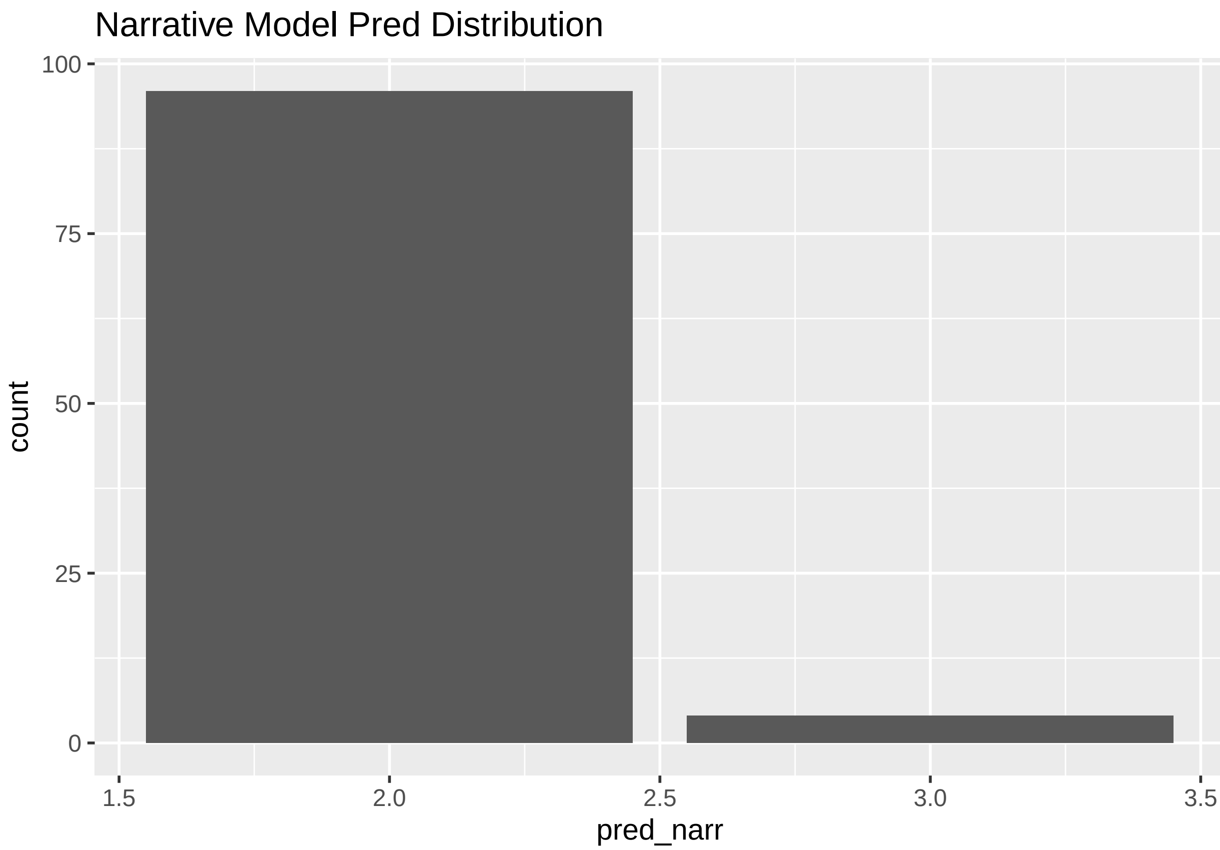
```r
#true polviews distribution

ggplot(df_3, aes(x = POLVIEWS_TRUE)) +
  geom_bar() +
  ggtitle("True POLVIEWS Distribution")
```




```r
ggplot(df_3, aes(x = pred_var)) +
  geom_bar() +
  ggtitle("Variable Model Pred Distribution")
```




```
ggplot(df_3, aes(x = pred_narr)) +
  geom_bar() +
  ggtitle("Narrative Model Pred Distribution")
```




```r
bias_by_predictor(df_3, age)
```

```
## # A tibble: 53 x 8
##      age       n mean_error_var mean_error_narr prop_too_cons_var prop_too_lib_var
##    <dbl> <int>          <dbl>           <dbl>             <dbl>            <dbl>
##  1 26        1            2               1                   1                0
##  2 39        1            2               1                   1                0
##  3 42        1            2               1                   1                0
##  4 47        3            2               1                   1                0
##  5 56        3            2               1                   1                0
##  6 57        1            2               1                   1                0
##  7 58        1            2               1                   1                0
##  8 75        1            2               1                   1                0
##  9 85        1            2               1                   1                0
## 10 49        2            1.5             0.5                 1                0
## # i 43 more rows
## # i 2 more variables: prop_too_cons_narr <dbl>, prop_too_lib_narr <dbl>
```

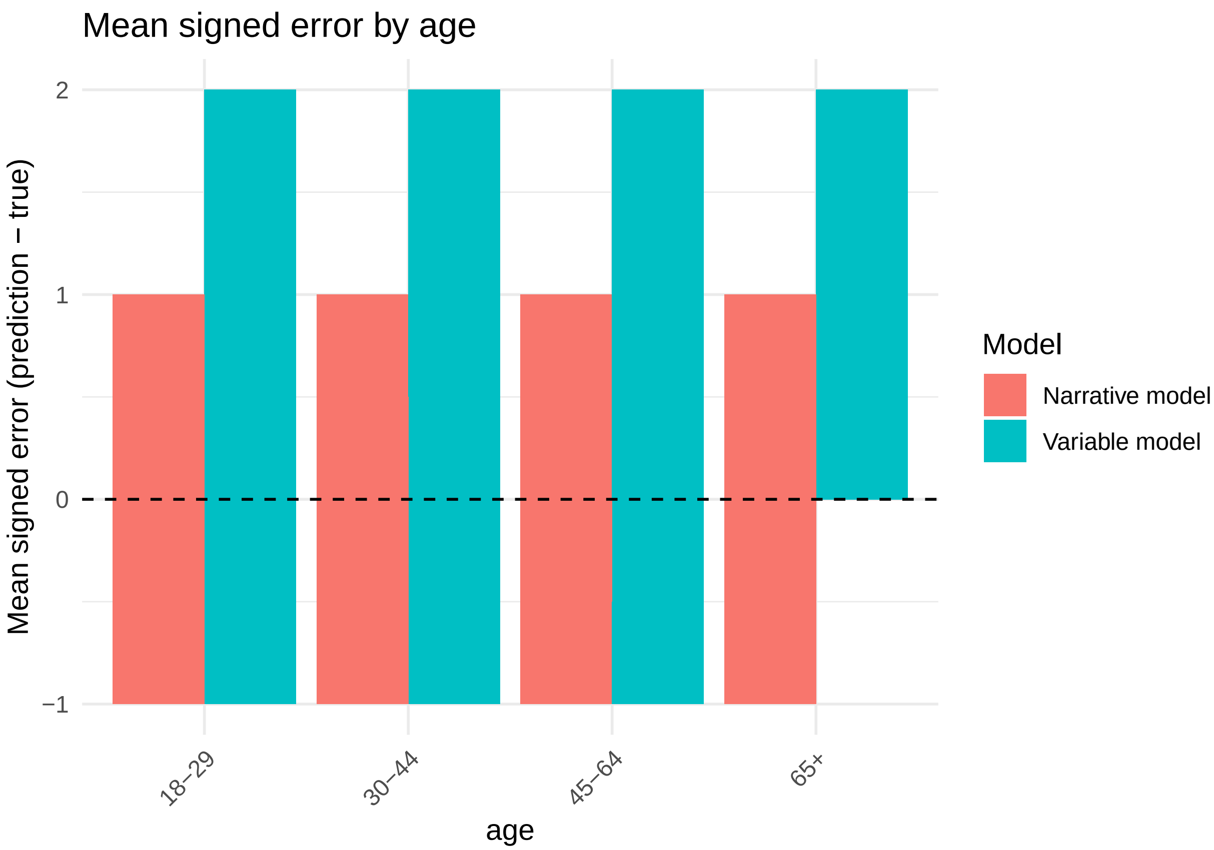
```r
bias_by_predictor(df_3, sex)
```

```
## # A tibble: 2 x 8
##   sex       n mean_error_var mean_error_narr prop_too_cons_var prop_too_lib_var
##   <fct> <int>          <dbl>           <dbl>             <dbl>            <dbl>
## 1 1        48           0.75          0.104              0.583            0.125
## 2 2        52           0.5           0.0385             0.462            0.154
## # i 2 more variables: prop_too_cons_narr <dbl>, prop_too_lib_narr <dbl>
```

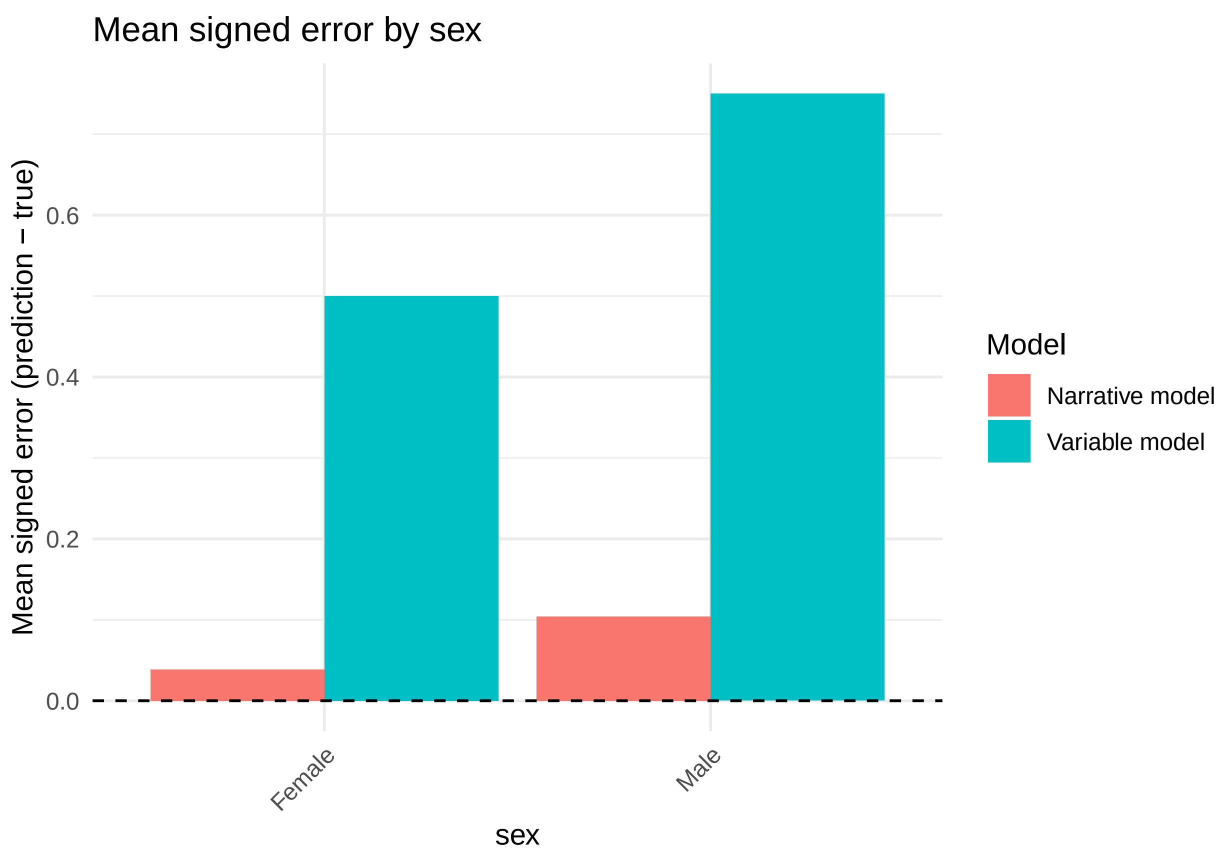
```r
bias_by_predictor(df_3, race)
```

```
## # A tibble: 3 x 8
##   race      n mean_error_var mean_error_narr prop_too_cons_var prop_too_lib_var
##   <fct> <int>          <dbl>           <dbl>             <dbl>            <dbl>
## 1 1        78          0.949          0.0641             0.641           0.0256
## 2 3        11         -0.182          0.0909             0.182           0.273
## 3 2        11         -0.909          0.0909             0                0.818
## # i 2 more variables: prop_too_cons_narr <dbl>, prop_too_lib_narr <dbl>
```

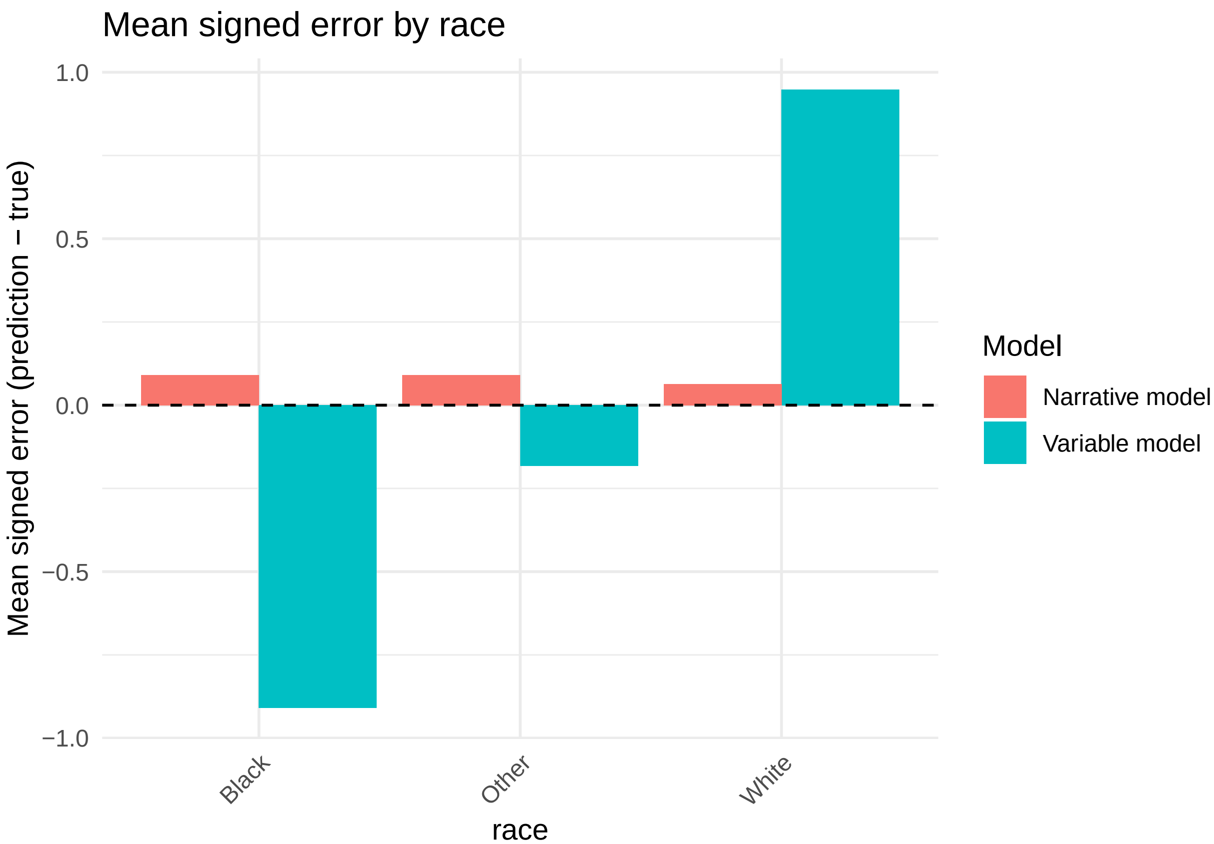
```r
plot_mean_error_by_predictor(df_3, age)
```




```r
plot_mean_error_by_predictor(df_3, sex)
```




```
plot_mean_error_by_predictor(df_3, race)
```




```
#collapse POLVIEWS into four categories:
sample100_4 <- sample100 %>%
  mutate(
    polviews_4= case_when(
      polviews %in% c(1, 2) ~ 1,    # extremely liberal
      polviews %in% c(3) ~ 2,    # slightly liberal
      polviews %in% c(4) ~ 3,     # moderate
      polviews %in% c(5, 6, 7) ~ 4     # conservative
    )
  ) %>%
  filter(!is.na(polviews_4))
head(sample100_4)
```

```
## # A tibble: 6 x 5
##   polviews age        race  sex   polviews_4
##      <int> <dbl+lbl> <fct> <fct>      <dbl>
## 1        3 59         1     1              2
## 2        4 52         1     2              3
## 3        6 61         1     1              4
## 4        4 45         1     2              3
## 5        4 28         3     1              3
## 6        4 62         1     2              3
```

```
sample100_nolabel_4 <- sample100_4 %>%
  select(-polviews_4) %>% # remove the ideology variable)
  select(-polviews) # remove the numeric ideology variable
```

```r
head(sample100_nolabel_4)
```

```
## # A tibble: 6 x 3
##   age       race  sex
##   <dbl+lbl> <fct> <fct>
## 1 59        1     1
## 2 52        1     2
## 3 61        1     1
## 4 45        1     2
## 5 28        3     1
## 6 62        1     2
```

```r
write.csv(sample100_nolabel_4, "3_var_gss_sample_100_unlabeled_4.csv", row.names = FALSE)
```

```r
var_4 <- read.csv("/Users/joyqu/Desktop/PLSC/3_var/3_var_gss_gpt5_var_predictions_4.csv")
head(var_4)
```

```
##   age race sex pred_polview
## 1  59    1   1            4
## 2  52    1   2            4
## 3  61    1   1            4
## 4  45    1   2            4
## 5  28    3   1            2
## 6  62    1   2            4
```

```r
# Extract variables
y_true_4 <- as.numeric(sample100_4$polviews_4)
y_pred_4 <- as.numeric(var_4$pred_polview)

# Compute metrics
MAE <- mean(abs(y_true_4 - y_pred_4))
MSE <- mean((y_true_4 - y_pred_4)^2)
Accuracy <- mean(y_true_4 == y_pred_4)
Within1 <- mean(abs(y_true_4 - y_pred_4) <= 1)

cat("Mean Absolute Error:", MAE, "\n")
```

```
## Mean Absolute Error: 1.16
```

```r
cat("Mean Squared Error:", MSE, "\n")
```

```
## Mean Squared Error: 2.38
```

```r
cat("Exact Match Accuracy:", round(Accuracy*100, 1), "%\n")
```

```
## Exact Match Accuracy: 30 %
```

```r
cat("Within ±1 Accuracy:", round(Within1*100, 1), "%\n")
```

```
## Within ±1 Accuracy: 69 %
```

```r
narrative_4 <- read.csv("/Users/joyqu/Desktop/PLSC/3_var/3_var_gss_gpt5_narrative_predictions_4.csv")
head(narrative_4)
```

```
##
## 1                         67 years old, this white man has settled into a steady rhythm of daily life.
## 2 56 years old, this from a diverse background woman has settled into a steady rhythm of daily life.
## 3                       33 years old, this white woman has settled into a steady rhythm of daily life.
```

```
## 4                     24 years old, this white woman has settled into a steady rhythm of daily life.
## 5                     46 years old, this white woman has settled into a steady rhythm of daily life.
## 6                       25 years old, this white man has settled into a steady rhythm of daily life.
##   pred_polview_narr
## 1                 3
## 2                 3
## 3                 3
## 4                 3
## 5                 3
## 6                 3
```

```
# Extract variables
y_true_4 <- as.numeric(sample100_4$polviews_4)
y_pred_4 <- as.numeric(narrative_4$pred_polview_narr)

# Compute metrics
MAE <- mean(abs(y_true_4 - y_pred_4))
MSE <- mean((y_true_4 - y_pred_4)^2)
Accuracy <- mean(y_true_4 == y_pred_4)
Within1 <- mean(abs(y_true_4 - y_pred_4) <= 1)

cat("Mean Absolute Error:", MAE, "\n")
```

```
## Mean Absolute Error: 0.83
```

```
cat("Mean Squared Error:", MSE, "\n")
```

```
## Mean Squared Error: 1.19
```

```
cat("Exact Match Accuracy:", round(Accuracy*100, 1), "%\n")
```

```
## Exact Match Accuracy: 35 %
```

```
cat("Within ±1 Accuracy:", round(Within1*100, 1), "%\n")
```

```
## Within ±1 Accuracy: 82 %
```

```
df_4 <- sample100_4 %>%
  mutate(row_id = row_number()) %>%
  select(
    row_id,
    POLVIEWS_TRUE = polviews_4,
    age, sex, race   # <- keep whatever predictors you want
  ) %>%
  inner_join(
    var_4 %>%
      mutate(row_id = row_number()) %>%
      select(row_id, pred_var = pred_polview),
    by = "row_id"
  ) %>%
  inner_join(
    narrative_4 %>%
      mutate(row_id = row_number()) %>%
      select(row_id, pred_narr = pred_polview_narr),
    by = "row_id"
  )
head(df_4)
```

```
## # A tibble: 6 x 7
##   row_id POLVIEWS_TRUE age       sex   race  pred_var pred_narr
##    <int>         <dbl> <dbl+lbl> <fct> <fct>    <int>     <int>
## 1      1             2 59        1     1            4         3
## 2      2             3 52        2     1            4         3
## 3      3             4 61        1     1            4         3
## 4      4             3 45        2     1            4         3
## 5      5             3 28        1     3            2         3
## 6      6             3 62        2     1            4         3
```

```r
df_4 <- df_4 %>%
  mutate(
    # Factor version for F1
    POLVIEWS_TRUE_fac = factor(POLVIEWS_TRUE),
    pred_var_fac      = factor(pred_var,  levels = levels(POLVIEWS_TRUE_fac)),
    pred_narr_fac     = factor(pred_narr, levels = levels(POLVIEWS_TRUE_fac)),

    # Numeric version for bias / error
    polviews_num = as.numeric(as.character(POLVIEWS_TRUE)),
    pred_var_num = as.numeric(as.character(pred_var)),
    pred_narr_num = as.numeric(as.character(pred_narr)),

    # Signed errors
    error_var  = pred_var_num  - polviews_num,
    error_narr = pred_narr_num - polviews_num
  )
results <- tibble(
  Model = c("Variable Model", "Narrative Model"),
  Macro_F1 = c(
    f1_macro(df_4$POLVIEWS_TRUE_fac, df_4$pred_var_fac),
    f1_macro(df_4$POLVIEWS_TRUE_fac, df_4$pred_narr_fac)
  ),
  Weighted_F1 = c(
    f1_weighted(df_4$POLVIEWS_TRUE_fac, df_4$pred_var_fac),
    f1_weighted(df_4$POLVIEWS_TRUE_fac, df_4$pred_narr_fac)
  )
)

print(results)
```

```
## # A tibble: 2 x 3
##   Model           Macro_F1 Weighted_F1
##   <chr>              <dbl>       <dbl>
## 1 Variable Model     0.733       0.701
## 2 Narrative Model    0.687       0.567
```

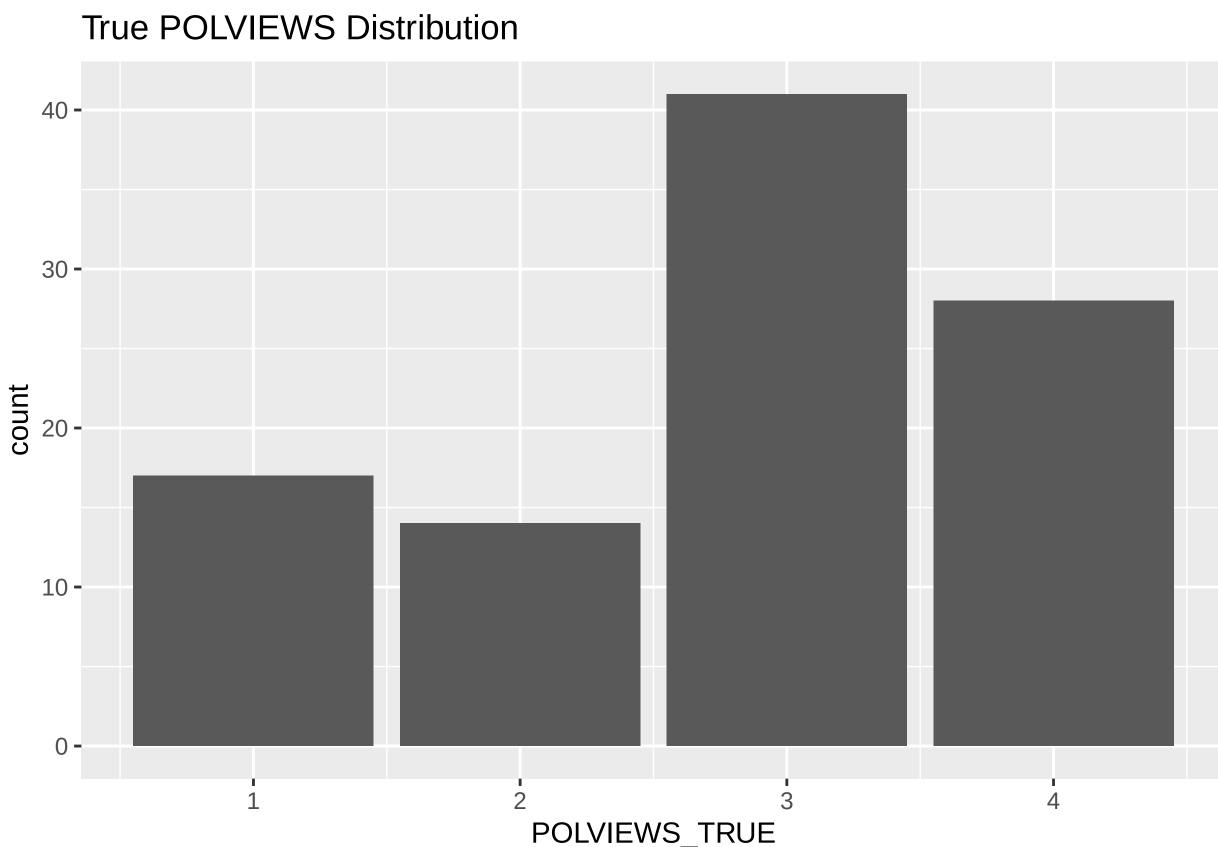
```r
mislabeled_comparison <- df_4 %>%
  mutate(
    # Wrong / right flags
    var_wrong  = pred_var  != POLVIEWS_TRUE,
    narr_wrong = pred_narr != POLVIEWS_TRUE,

    # Case types with only two models
    case_type = case_when(
      var_wrong  & !narr_wrong ~ "Only Variable Model Wrong",
```

```
      !var_wrong & narr_wrong  ~ "Only Narrative Model Wrong",
      var_wrong  & narr_wrong  ~ "Both Wrong",
      TRUE                     ~ "Both Correct"
    ),

    # Differences vs true (numeric scale 1-7)
    diff_var  = as.numeric(pred_var)  - as.numeric(POLVIEWS_TRUE),
    diff_narr = as.numeric(pred_narr) - as.numeric(POLVIEWS_TRUE),

    # Bias direction for each model (only label as too lib/con if it's wrong)
    bias_var = dplyr::case_when(
      !var_wrong          ~ "Correct",
      diff_var  > 0       ~ "Too Conservative",
      diff_var  < 0       ~ "Too Liberal",
      TRUE                ~ NA_character_
    ),
    bias_narr = dplyr::case_when(
      !narr_wrong         ~ "Correct",
      diff_narr  > 0      ~ "Too Conservative",
      diff_narr  < 0      ~ "Too Liberal",
      TRUE                ~ NA_character_
    )
  ) %>%
  select(
    row_id, POLVIEWS_TRUE,
    pred_var, pred_narr,
    var_wrong, narr_wrong,
    case_type,
    bias_var, bias_narr
  )

# Save to CSV
write.csv(mislabeled_comparison,
          "3_var_mislabeled_cases_comparison_4.csv",
          row.names = FALSE)

bias_table <- mislabeled_comparison %>%
  select(bias_var, bias_narr) %>%
  tidyr::pivot_longer(
    cols      = everything(),
    names_to  = "model",
    values_to = "bias"
  ) %>%
  dplyr::filter(bias != "Correct") %>%   # only mislabeled cases
  dplyr::group_by(model, bias) %>%
  dplyr::summarise(count = dplyr::n(), .groups = "drop_last") %>%
  dplyr::mutate(
    percent = count / sum(count) * 100
  ) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(
    model = dplyr::recode(
      model,
```

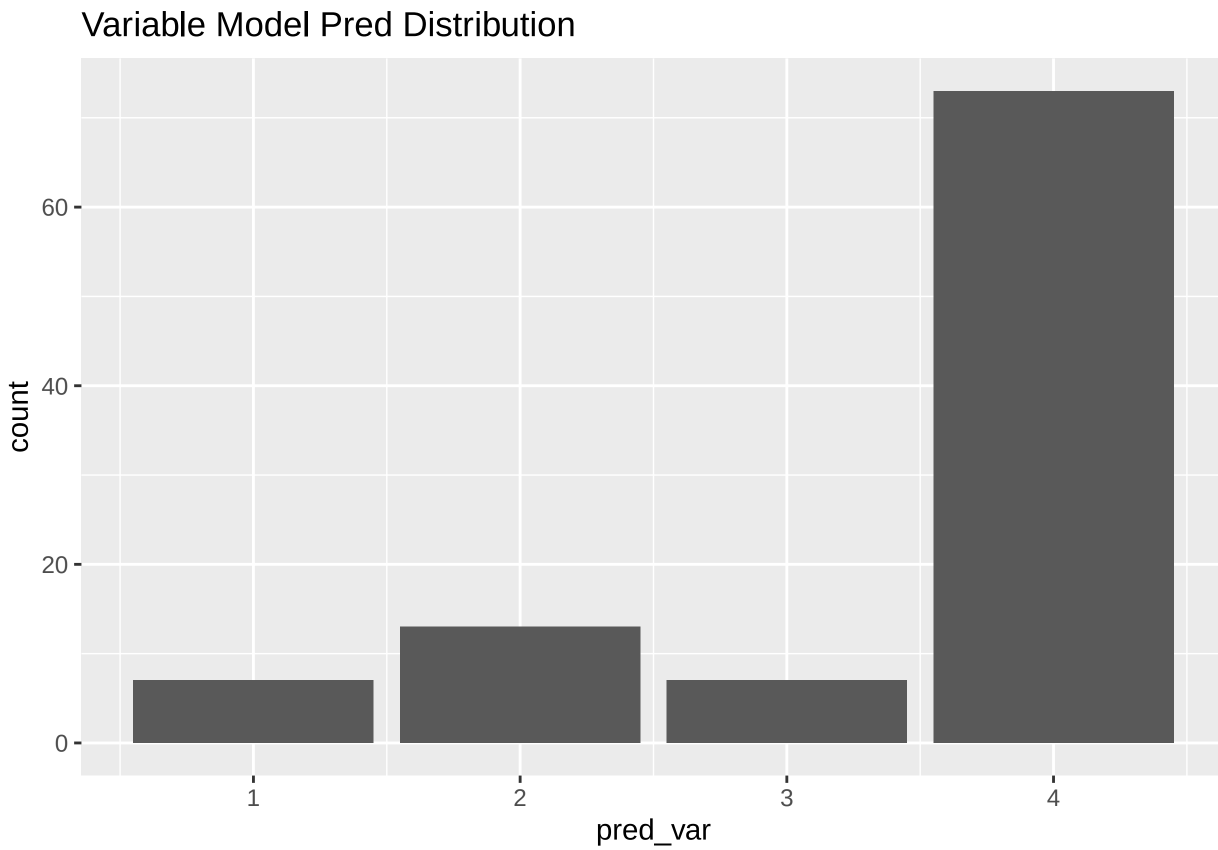
```r
      bias_var  = "Variable Model",
      bias_narr = "Narrative Model"
    )
  ) %>%
  dplyr::arrange(model, bias)
bias_table
```

```
## # A tibble: 4 x 4
##   model           bias              count percent
##   <chr>           <chr>             <int>   <dbl>
## 1 Narrative Model Too Conservative     31    47.7
## 2 Narrative Model Too Liberal          34    52.3
## 3 Variable Model  Too Conservative     51    72.9
## 4 Variable Model  Too Liberal          19    27.1
```

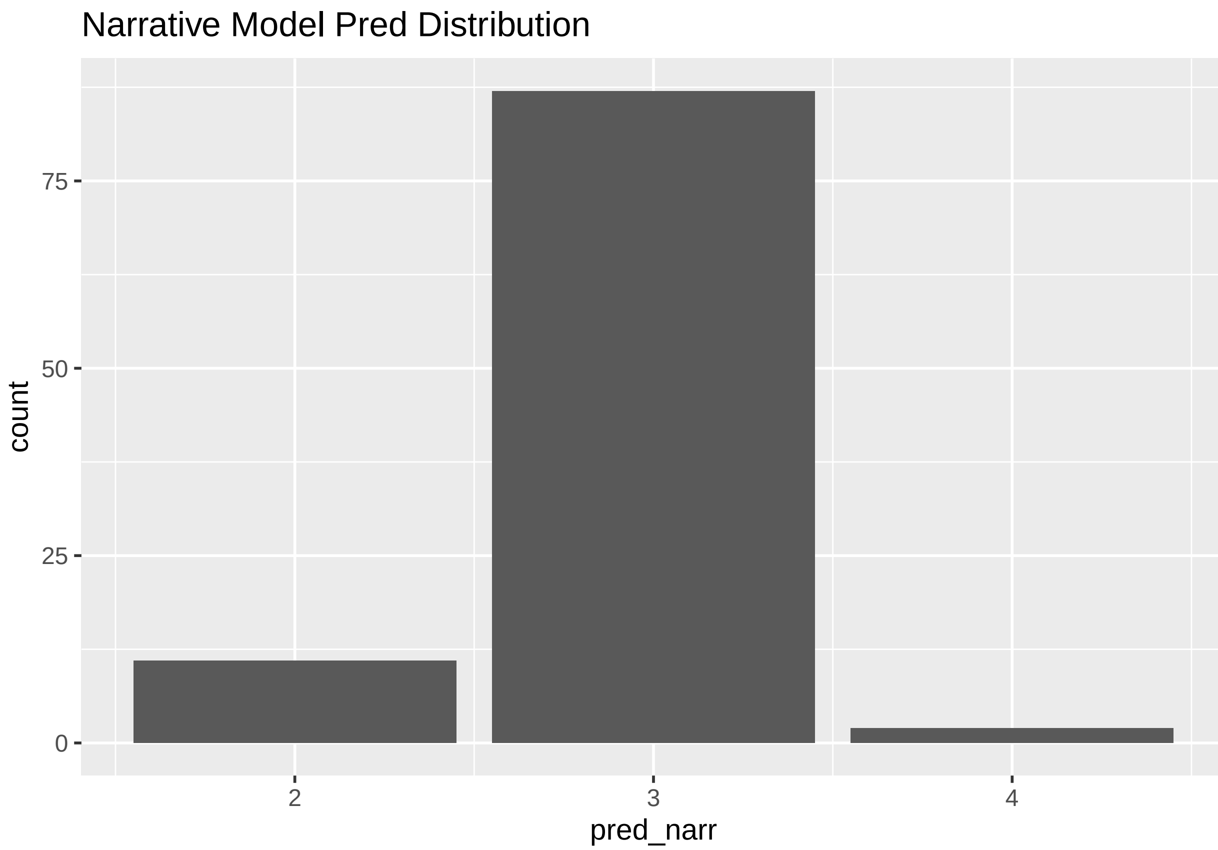
```r
#true polviews distribution

ggplot(df_4, aes(x = POLVIEWS_TRUE)) +
  geom_bar() +
  ggtitle("True POLVIEWS Distribution")
```




```r
ggplot(df_4, aes(x = pred_var)) +
  geom_bar() +
  ggtitle("Variable Model Pred Distribution")
```




```r
ggplot(df_4, aes(x = pred_narr)) +
  geom_bar() +
  ggtitle("Narrative Model Pred Distribution")
```




```r
bias_by_predictor(df_4, age)
```

```
## # A tibble: 53 x 8
##      age     n mean_error_var mean_error_narr prop_too_cons_var prop_too_lib_var
##    <dbl> <int>          <dbl>           <dbl>             <dbl>            <dbl>
##  1 39        1           3              2                     1                0
##  2 42        1           3              2                     1                0
##  3 57        1           3              2                     1                0
##  4 58        1           3              2                     1                0
##  5 75        1           3              2                     1                0
##  6 47        3           2.67           1.67                  1                0
##  7 49        2           2              0.5                   1                0
##  8 56        3           2              0.667                 1                0
##  9 69        2           2              0.5                   1                0
## 10 85        1           2              1                     1                0
## # i 43 more rows
## # i 2 more variables: prop_too_cons_narr <dbl>, prop_too_lib_narr <dbl>
```

```r
bias_by_predictor(df_4, sex)
```

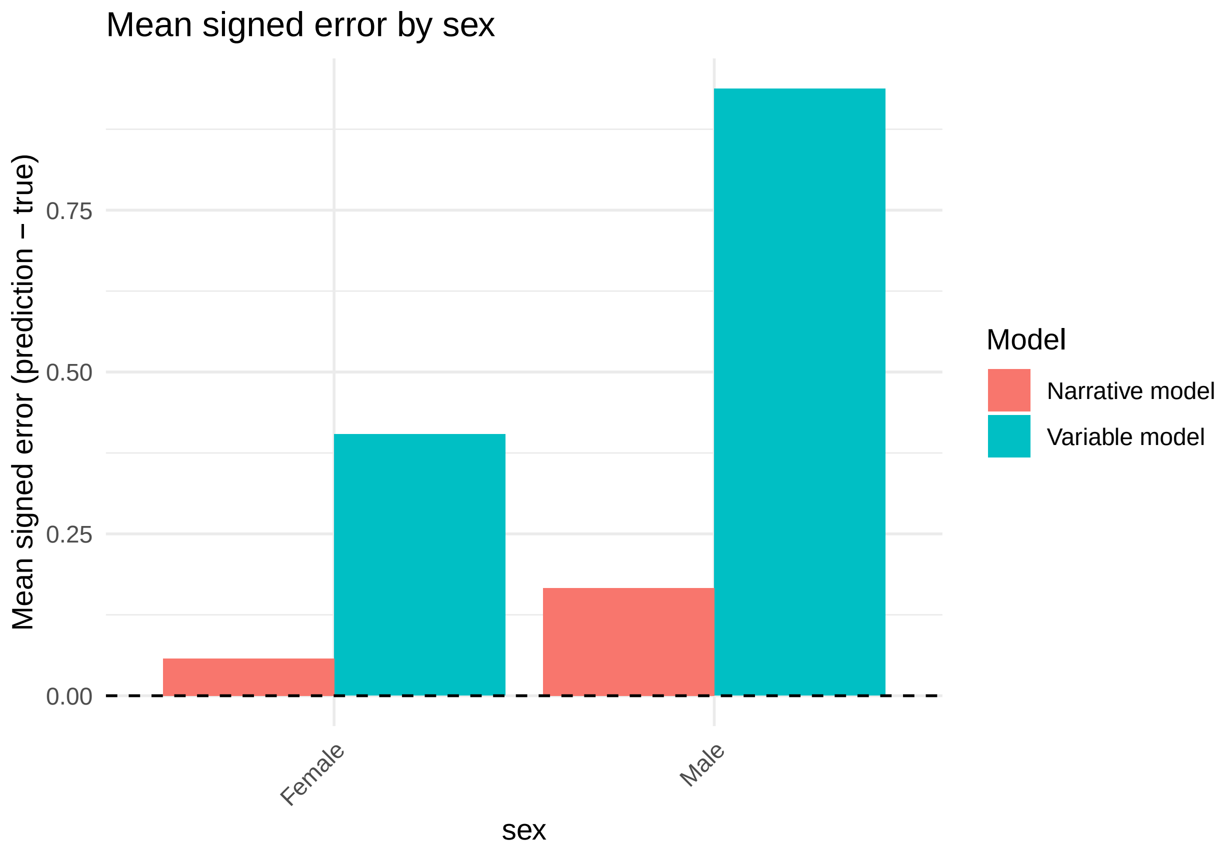
```
## # A tibble: 2 x 8
##   sex       n mean_error_var mean_error_narr prop_too_cons_var prop_too_lib_var
##   <fct> <int>          <dbl>           <dbl>             <dbl>            <dbl>
## 1 1        48          0.938          0.167              0.604            0.125
## 2 2        52          0.404          0.0577             0.423            0.25
## # i 2 more variables: prop_too_cons_narr <dbl>, prop_too_lib_narr <dbl>
```

```
bias_by_predictor(df_4, race)
```

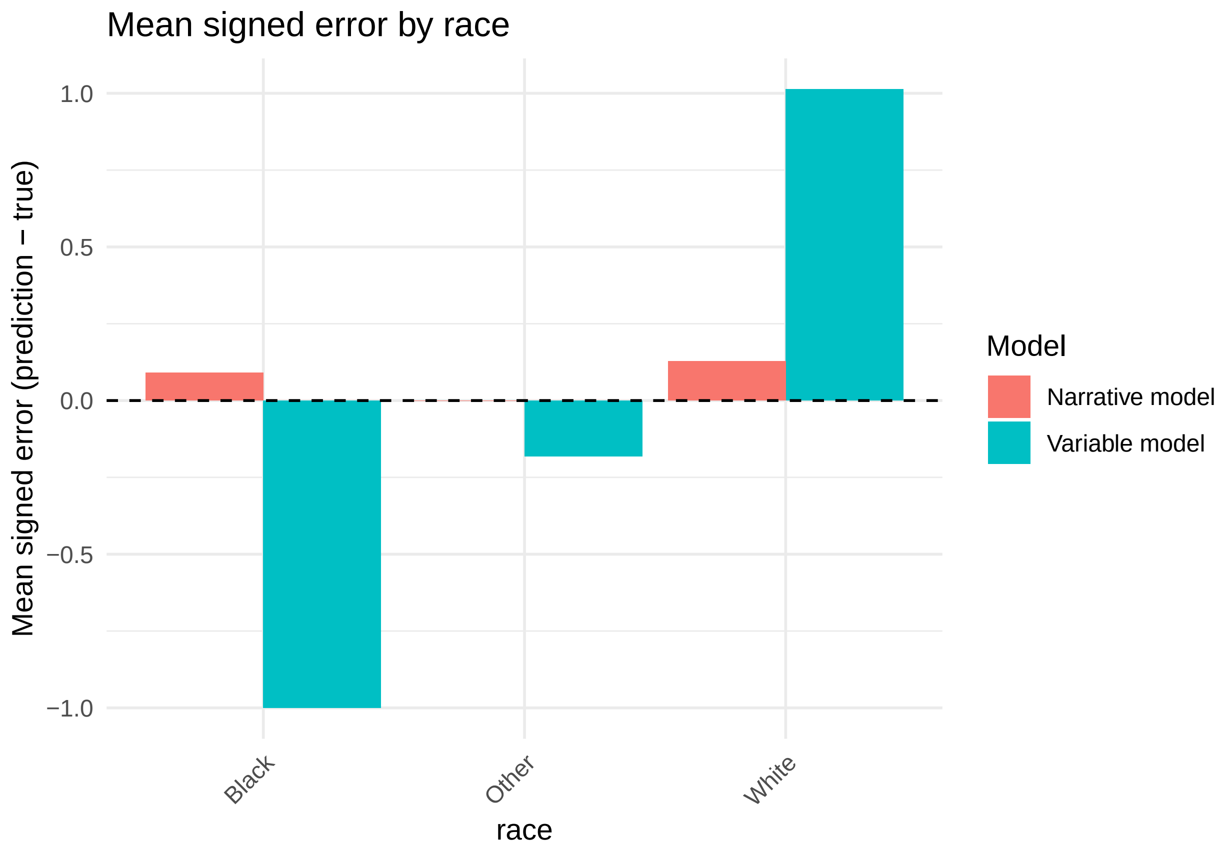
```
## # A tibble: 3 x 8
##   race      n mean_error_var mean_error_narr prop_too_cons_var prop_too_lib_var
##   <fct> <int>          <dbl>           <dbl>             <dbl>            <dbl>
## 1 1        78          1.01           0.128             0.603            0.0641
## 2 3        11         -0.182          0                 0.273            0.545
## 3 2        11         -1              0.0909            0.0909           0.727
## # i 2 more variables: prop_too_cons_narr <dbl>, prop_too_lib_narr <dbl>
```

```
plot_mean_error_by_predictor(df_4, age)
```


```
plot_mean_error_by_predictor(df_4, sex)
```




```
plot_mean_error_by_predictor(df_4, race)
```




```
#collapse POLVIEWS into five categories
sample100_5 <- sample100 %>%
  mutate(
    polviews_5= case_when(
      polviews %in% c(1) ~ 1,    # extremely liberal
      polviews %in% c(2,3) ~ 2,    #  liberal
      polviews %in% c(4) ~ 3,     # moderate
      polviews %in% c(5,6) ~ 4,     # conservative
      polviews %in% c(7) ~ 5     # extremely conservative
    )
  ) %>%
  filter(!is.na(polviews_5))
head(sample100_5)
```

```
## # A tibble: 6 x 5
##   polviews age        race  sex   polviews_5
##      <int> <dbl+lbl> <fct> <fct>      <dbl>
## 1        3 59        1     1              2
## 2        4 52        1     2              3
## 3        6 61        1     1              4
## 4        4 45        1     2              3
## 5        4 28        3     1              3
## 6        4 62        1     2              3
```

```
sample100_nolabel_5 <- sample100_5 %>%
  select(-polviews_5) %>% # remove the ideology variable)
  select(-polviews) # remove the numeric ideology variable
```

```r
head(sample100_nolabel_5)
```

```
## # A tibble: 6 x 3
##   age       race  sex
##   <dbl+lbl> <fct> <fct>
## 1 59        1     1
## 2 52        1     2
## 3 61        1     1
## 4 45        1     2
## 5 28        3     1
## 6 62        1     2
```

```r
write.csv(sample100_nolabel_5, "3_var_gss_sample_100_unlabeled_5.csv", row.names = FALSE)
```

```r
var_5 <- read.csv("/Users/joyqu/Desktop/PLSC/3_var/3_var_gss_gpt5_var_predictions_5.csv")
head(var_5)
```

```
##   age race sex pred_polview
## 1  59    1   1            4
## 2  52    1   2            4
## 3  61    1   1            4
## 4  45    1   2            4
## 5  28    3   1            3
## 6  62    1   2            4
```

```r
# Extract variables
y_true_5 <- as.numeric(sample100_5$polviews_5)
y_pred_5 <- as.numeric(var_5$pred_polview)

# Compute metrics
MAE <- mean(abs(y_true_5 - y_pred_5))
MSE <- mean((y_true_5 - y_pred_5)^2)
Accuracy <- mean(y_true_5 == y_pred_5)
Within1 <- mean(abs(y_true_5 - y_pred_5) <= 1)

cat("Mean Absolute Error:", MAE, "\n")
```

```
## Mean Absolute Error: 1.03
```

```r
cat("Mean Squared Error:", MSE, "\n")
```

```
## Mean Squared Error: 1.75
```

```r
cat("Exact Match Accuracy:", round(Accuracy*100, 1), "%\n")
```

```
## Exact Match Accuracy: 29 %
```

```r
cat("Within ±1 Accuracy:", round(Within1*100, 1), "%\n")
```

```
## Within ±1 Accuracy: 72 %
```

```r
narrative_5 <- read.csv("/Users/joyqu/Desktop/PLSC/3_var/3_var_gss_gpt5_narrative_predictions_5.csv")
head(narrative_5)
```

```
##
## 1                       67 years old, this white man has settled into a steady rhythm of daily life.
## 2 56 years old, this from a diverse background woman has settled into a steady rhythm of daily life.
## 3                     33 years old, this white woman has settled into a steady rhythm of daily life.
```

```
## 4                     24 years old, this white woman has settled into a steady rhythm of daily life.
## 5                     46 years old, this white woman has settled into a steady rhythm of daily life.
## 6                       25 years old, this white man has settled into a steady rhythm of daily life.
##   pred_polview_narr
## 1                 3
## 2                 3
## 3                 3
## 4                 3
## 5                 3
## 6                 3
```

```r
# Extract variables
y_true_5 <- as.numeric(sample100_5$polviews_5)
y_pred_5 <- as.numeric(narrative_5$pred_polview_narr)

# Compute metrics
MAE <- mean(abs(y_true_5 - y_pred_5))
MSE <- mean((y_true_5 - y_pred_5)^2)
Accuracy <- mean(y_true_5 == y_pred_5)
Within1 <- mean(abs(y_true_5 - y_pred_5) <= 1)

cat("Mean Absolute Error:", MAE, "\n")
```

```
## Mean Absolute Error: 0.8
```

```r
cat("Mean Squared Error:", MSE, "\n")
```

```
## Mean Squared Error: 1.12
```

```r
cat("Exact Match Accuracy:", round(Accuracy*100, 1), "%\n")
```

```
## Exact Match Accuracy: 34 %
```

```r
cat("Within ±1 Accuracy:", round(Within1*100, 1), "%\n")
```

```
## Within ±1 Accuracy: 88 %
```

```r
df_5 <- sample100_5 %>%
  mutate(row_id = row_number()) %>%
  select(
    row_id,
    POLVIEWS_TRUE = polviews_5,
    age, sex, race  # <- keep whatever predictors you want
  ) %>%
  inner_join(
    var_5 %>%
      mutate(row_id = row_number()) %>%
      select(row_id, pred_var = pred_polview),
    by = "row_id"
  ) %>%
  inner_join(
    narrative_5 %>%
      mutate(row_id = row_number()) %>%
      select(row_id, pred_narr = pred_polview_narr),
    by = "row_id"
  )
head(df_5)
```

```
## # A tibble: 6 x 7
##   row_id POLVIEWS_TRUE age       sex   race  pred_var pred_narr
##    <int>         <dbl> <dbl+lbl> <fct> <fct>    <int>     <int>
## 1      1             2 59        1     1            4         3
## 2      2             3 52        2     1            4         3
## 3      3             4 61        1     1            4         3
## 4      4             3 45        2     1            4         3
## 5      5             3 28        1     3            3         3
## 6      6             3 62        2     1            4         3
```

```
df_5 <- df_5 %>%
  mutate(
    # Factor version for F1
    POLVIEWS_TRUE_fac = factor(POLVIEWS_TRUE),
    pred_var_fac      = factor(pred_var,  levels = levels(POLVIEWS_TRUE_fac)),
    pred_narr_fac     = factor(pred_narr, levels = levels(POLVIEWS_TRUE_fac)),

    # Numeric version for bias / error
    polviews_num = as.numeric(as.character(POLVIEWS_TRUE)),
    pred_var_num = as.numeric(as.character(pred_var)),
    pred_narr_num = as.numeric(as.character(pred_narr)),

    # Signed errors
    error_var  = pred_var_num  - polviews_num,
    error_narr = pred_narr_num - polviews_num
  )
results <- tibble(
  Model = c("Variable Model", "Narrative Model"),
  Macro_F1 = c(
    f1_macro(df_5$POLVIEWS_TRUE_fac, df_5$pred_var_fac),
    f1_macro(df_5$POLVIEWS_TRUE_fac, df_5$pred_narr_fac)
  ),
  Weighted_F1 = c(
    f1_weighted(df_5$POLVIEWS_TRUE_fac, df_5$pred_var_fac),
    f1_weighted(df_5$POLVIEWS_TRUE_fac, df_5$pred_narr_fac)
  )
)

print(results)
```

```
## # A tibble: 2 x 3
##   Model           Macro_F1 Weighted_F1
##   <chr>              <dbl>       <dbl>
## 1 Variable Model     0.782       0.702
## 2 Narrative Model    0.761       0.587
```

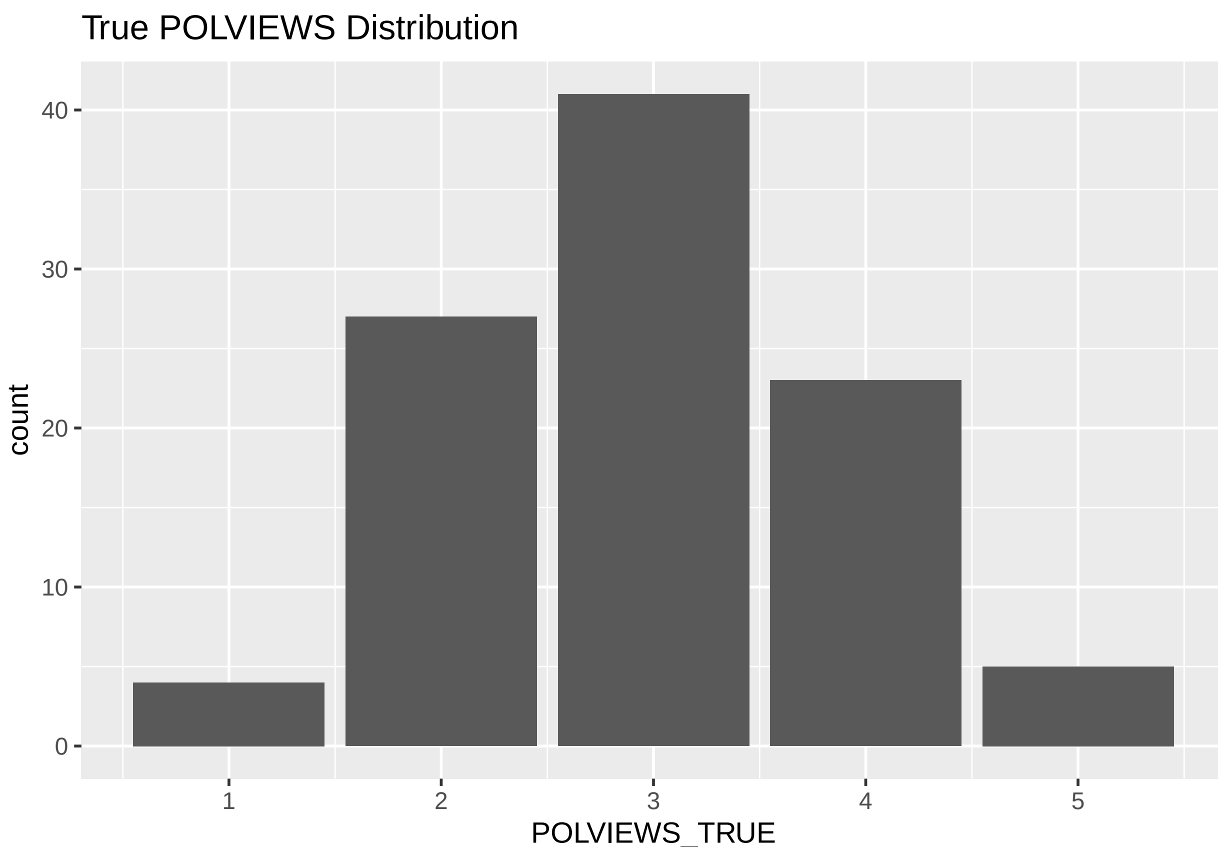
```
mislabeled_comparison <- df_5 %>%
  mutate(
    # Wrong / right flags
    var_wrong  = pred_var  != POLVIEWS_TRUE,
    narr_wrong = pred_narr != POLVIEWS_TRUE,

    # Case types with only two models
    case_type = case_when(
      var_wrong  & !narr_wrong ~ "Only Variable Model Wrong",
```

```r
      !var_wrong & narr_wrong  ~ "Only Narrative Model Wrong",
      var_wrong  & narr_wrong  ~ "Both Wrong",
      TRUE                     ~ "Both Correct"
    ),

    # Differences vs true (numeric scale 1-7)
    diff_var  = as.numeric(pred_var)  - as.numeric(POLVIEWS_TRUE),
    diff_narr = as.numeric(pred_narr) - as.numeric(POLVIEWS_TRUE),

    # Bias direction for each model (only label as too lib/con if it's wrong)
    bias_var = dplyr::case_when(
      !var_wrong          ~ "Correct",
      diff_var  > 0       ~ "Too Conservative",
      diff_var  < 0       ~ "Too Liberal",
      TRUE                ~ NA_character_
    ),
    bias_narr = dplyr::case_when(
      !narr_wrong         ~ "Correct",
      diff_narr  > 0      ~ "Too Conservative",
      diff_narr  < 0      ~ "Too Liberal",
      TRUE                ~ NA_character_
    )
  ) %>%
  select(
    row_id, POLVIEWS_TRUE,
    pred_var, pred_narr,
    var_wrong, narr_wrong,
    case_type,
    bias_var, bias_narr
  )

# Save to CSV
write.csv(mislabeled_comparison,
          "3_var_mislabeled_cases_comparison_5.csv",
          row.names = FALSE)

bias_table <- mislabeled_comparison %>%
  select(bias_var, bias_narr) %>%
  tidyr::pivot_longer(
    cols      = everything(),
    names_to  = "model",
    values_to = "bias"
  ) %>%
  dplyr::filter(bias != "Correct") %>%   # only mislabeled cases
  dplyr::group_by(model, bias) %>%
  dplyr::summarise(count = dplyr::n(), .groups = "drop_last") %>%
  dplyr::mutate(
    percent = count / sum(count) * 100
  ) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(
    model = dplyr::recode(
      model,
```

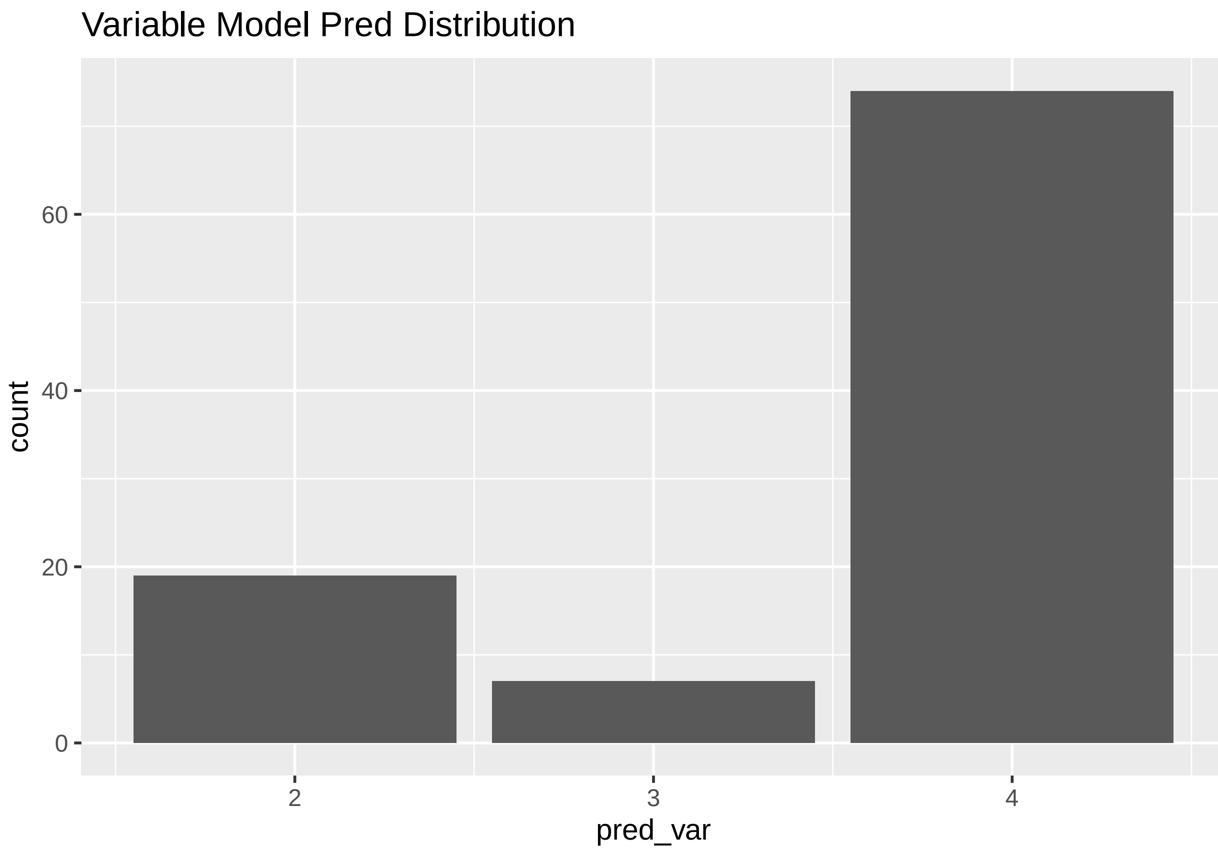
```r
      bias_var  = "Variable Model",
      bias_narr = "Narrative Model"
    )
  ) %>%
  dplyr::arrange(model, bias)
bias_table
```

```
## # A tibble: 4 x 4
##   model           bias              count percent
##   <chr>           <chr>             <int>   <dbl>
## 1 Narrative Model Too Conservative     35    53.0
## 2 Narrative Model Too Liberal          31    47.0
## 3 Variable Model  Too Conservative     52    73.2
## 4 Variable Model  Too Liberal          19    26.8
```

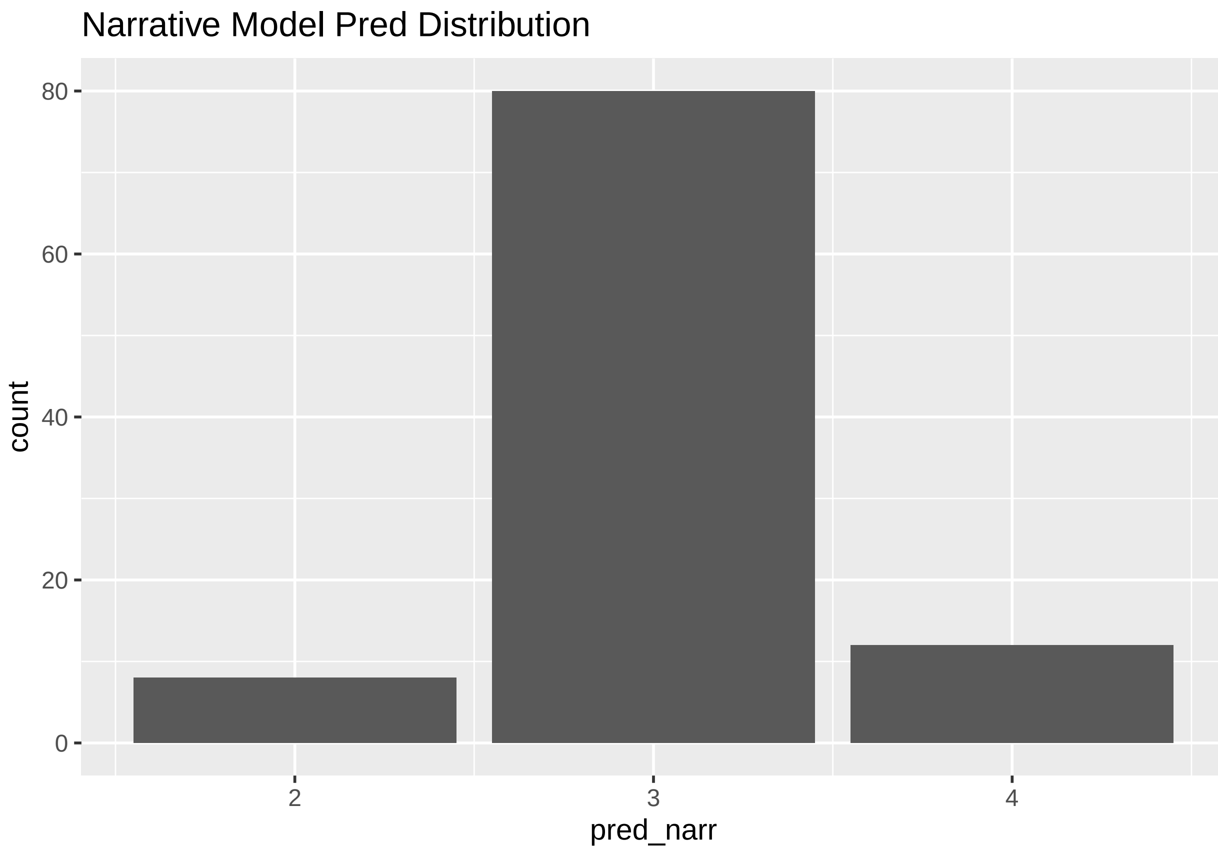
```r
#true polviews distribution

ggplot(df_5, aes(x = POLVIEWS_TRUE)) +
  geom_bar() +
  ggtitle("True POLVIEWS Distribution")
```




```r
ggplot(df_5, aes(x = pred_var)) +
  geom_bar() +
  ggtitle("Variable Model Pred Distribution")
```




```
ggplot(df_5, aes(x = pred_narr)) +
  geom_bar() +
  ggtitle("Narrative Model Pred Distribution")
```




```
bias_by_predictor(df_5, age)
```

```
## # A tibble: 53 x 8
##      age     n mean_error_var mean_error_narr prop_too_cons_var prop_too_lib_var
##    <dbl> <int>          <dbl>           <dbl>             <dbl>            <dbl>
##  1 47        3           2.33            1.67                 1                0
##  2 39        1           2               1                    1                0
##  3 42        1           2               1                    1                0
##  4 49        2           2               0.5                  1                0
##  5 56        3           2               1                    1                0
##  6 57        1           2               1                    1                0
##  7 58        1           2               1                    1                0
##  8 75        1           2               1                    1                0
##  9 85        1           2               1                    1                0
## 10 69        2           1.5             0                    1                0
## # i 43 more rows
## # i 2 more variables: prop_too_cons_narr <dbl>, prop_too_lib_narr <dbl>
```

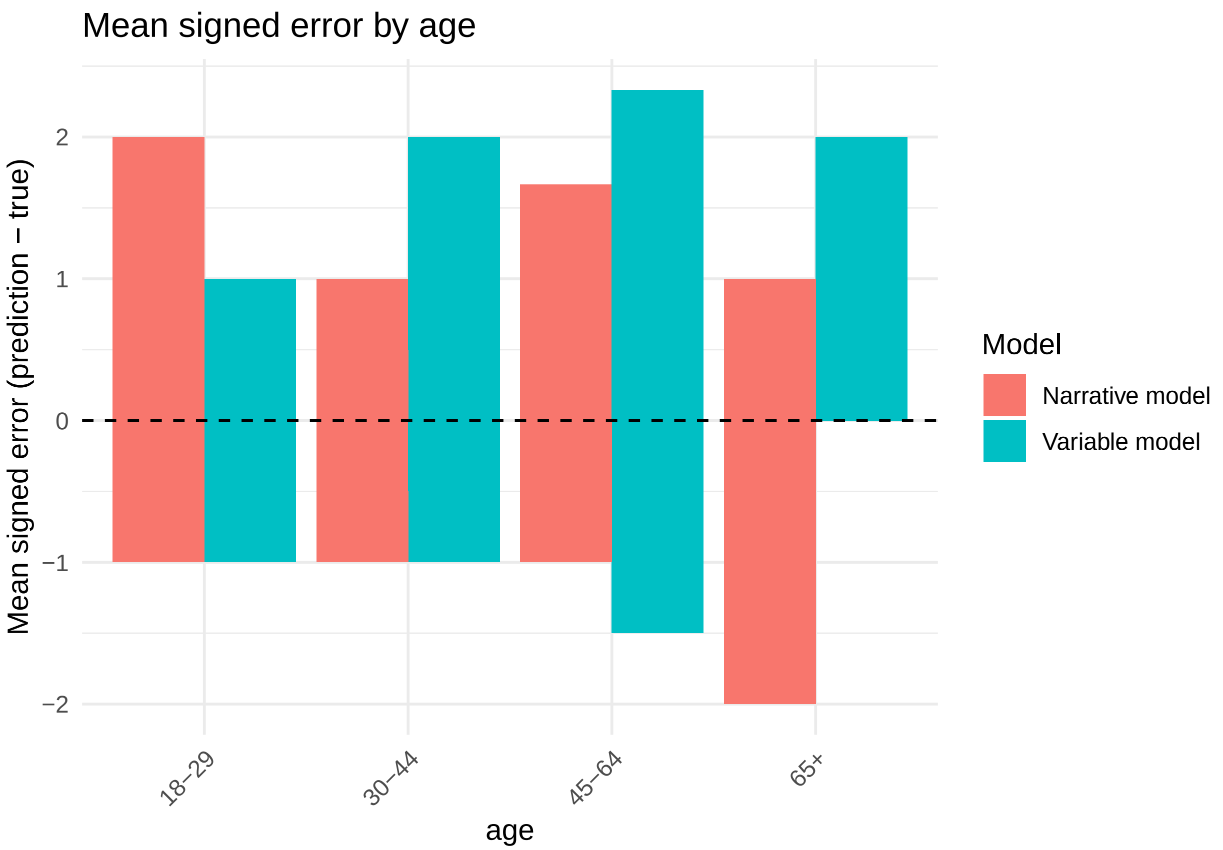
```
bias_by_predictor(df_5, sex)
```

```
## # A tibble: 2 x 8
##   sex       n mean_error_var mean_error_narr prop_too_cons_var prop_too_lib_var
##   <fct> <int>          <dbl>           <dbl>             <dbl>            <dbl>
## 1 1        48          0.708          0.0417             0.604            0.167
## 2 2        52          0.442          0.0769             0.442            0.212
## # i 2 more variables: prop_too_cons_narr <dbl>, prop_too_lib_narr <dbl>
```

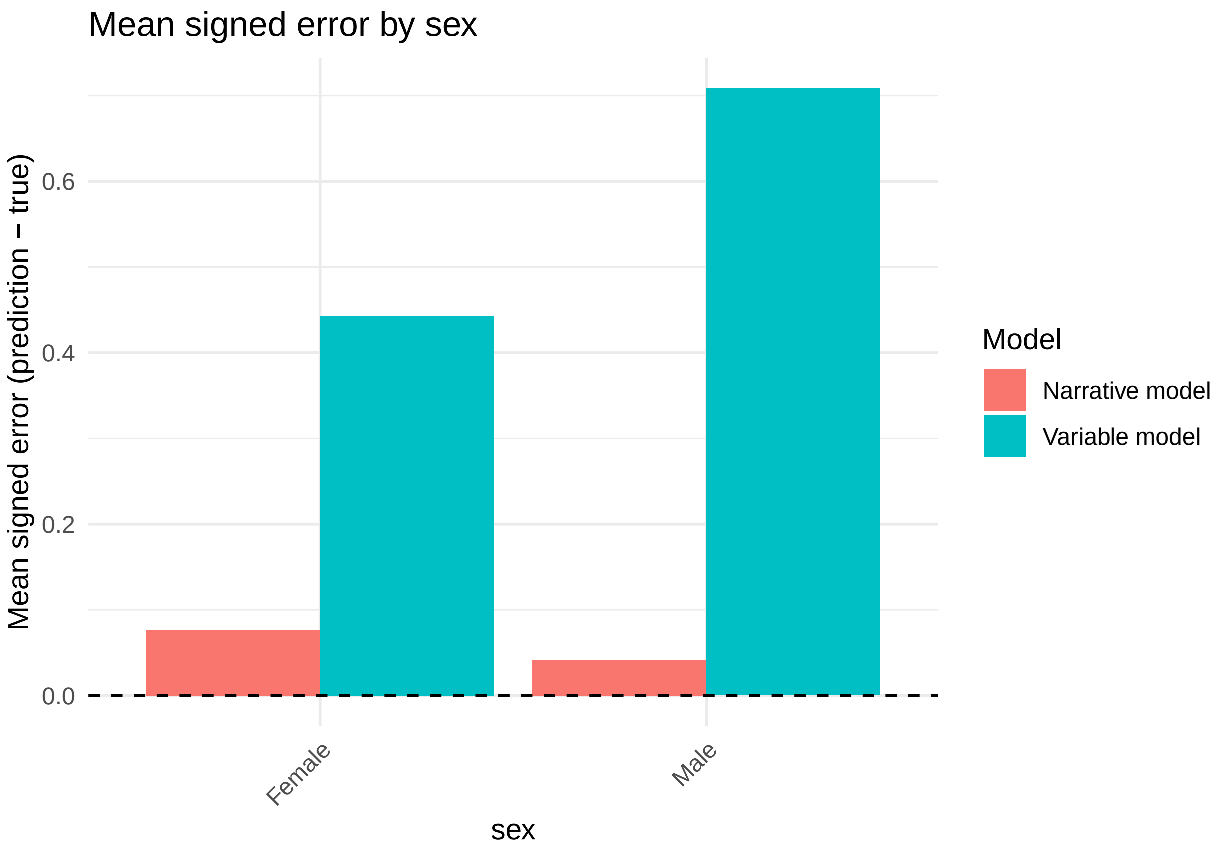
```r
bias_by_predictor(df_5, race)
```

```
## # A tibble: 3 x 8
##   race      n mean_error_var mean_error_narr prop_too_cons_var prop_too_lib_var
##   <fct> <int>          <dbl>           <dbl>             <dbl>            <dbl>
## 1 1        78         0.846           0.0769             0.603           0.0897
## 2 3        11        -0.0909         -0.0909             0.273           0.364
## 3 2        11        -0.727           0.0909             0.182           0.727
## # i 2 more variables: prop_too_cons_narr <dbl>, prop_too_lib_narr <dbl>
```

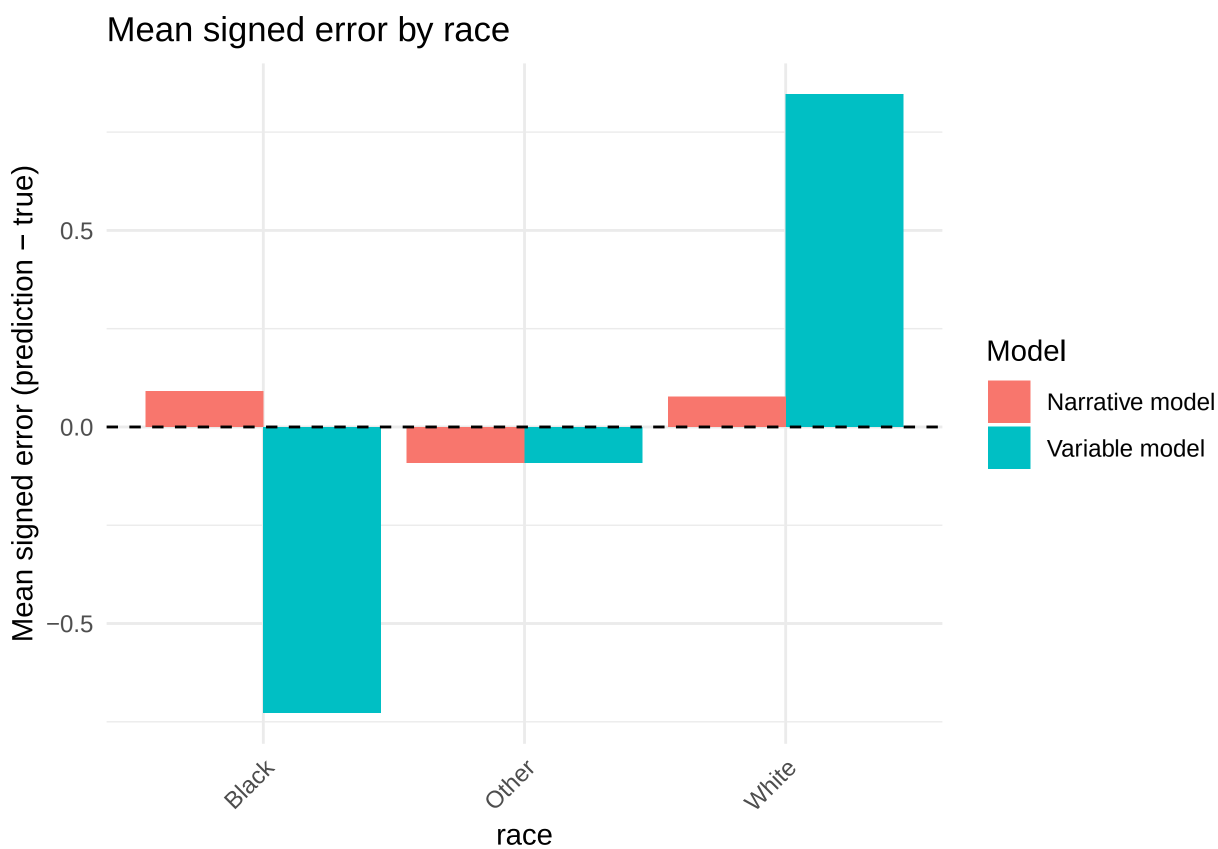
```r
plot_mean_error_by_predictor(df_5, age)
```




```r
plot_mean_error_by_predictor(df_5, sex)
```




```
plot_mean_error_by_predictor(df_5, race)
```



Mean signed error by race
Mean signed error (prediction − true)
0.5
0.0
−0.5
Black
Other
White
race
Model
Narrative model
Variable model