Midterm

Group 4

2025-03-24

Load data

```
load("dat1.RData")
load("dat2.RData")

dat1 =
  dat1 |>
  select(-id)

dat2 =
  dat2 |>
  select(-id)
```

Explorey Data Analysis and Visualization

Summary statistics

dat1

```
summarize_cat = function(data, var_name) {

  var_label = deparse(substitute(var_name))

  out_df =
    data |>
    count({{ var_name }}) |>
    mutate(
      Percent = round(100 * n / sum(n), 1),
      Variable = var_label,
      Level = as.character({{ var_name }})
    ) |>
    rename(N = n) |>
    select(Variable, Level, N, Percent)

  return(out_df)
}

gender <- summarize_cat(dat1, gender)
race <- summarize_cat(dat1, race)
smoking <- summarize_cat(dat1, smoking)
```

```r
diabetes <- summarize_cat(dat1, diabetes)
hypertension <- summarize_cat(dat1, hypertension)
bind_rows(gender, race, smoking, diabetes, hypertension) %>%
  knitr::kable()
```

Variable	Level	N	Percent
gender	0	2573	51.5
gender	1	2427	48.5
race	1	3221	64.4
race	2	278	5.6
race	3	1036	20.7
race	4	465	9.3
smoking	0	3010	60.2
smoking	1	1504	30.1
smoking	2	486	9.7
diabetes	0	4228	84.6
diabetes	1	772	15.4
hypertension	0	2702	54.0
hypertension	1	2298	46.0

```r
summarize_cont = function(data, var_name) {

  var_label = deparse(substitute(var_name))

  out_df =
    data |>
    summarize(
      Variable = var_label,
      Median = round(median({{ var_name }}, na.rm = TRUE), 1),
      Q1 = round(quantile({{ var_name }}, 0.25, na.rm = TRUE), 1),
      Q3 = round(quantile({{ var_name }}, 0.75, na.rm = TRUE), 1)
    ) |>
    mutate(
      IQR = paste0("[", Q1, ", ", Q3, "]")
    ) |>
    select(Variable, Median, IQR)

  return(out_df)
}

age = summarize_cont(dat1, age)
bmi = summarize_cont(dat1, bmi)
height = summarize_cont(dat1, height)
weight = summarize_cont(dat1, weight)
SBP = summarize_cont(dat1, SBP)
LDL = summarize_cont(dat1, LDL)
time = summarize_cont(dat1, time)
log_anti = summarize_cont(dat1, log_antibody)
bind_rows(age, bmi, height, weight, SBP, LDL, time, log_anti) %>%
  knitr::kable()
```

Variable	Median	IQR
age	60.0	[57, 63]
bmi	27.6	[25.8, 29.5]
height	170.1	[166.1, 174.2]
weight	80.1	[75.4, 84.9]
SBP	130.0	[124, 135]
LDL	110.0	[96, 124]
time	106.0	[76, 138]
log_antibody	10.1	[9.7, 10.5]

dat2

```
gender <- summarize_cat(dat2, gender)
race <- summarize_cat(dat2, race)
smoking <- summarize_cat(dat2, smoking)
diabetes <- summarize_cat(dat2, diabetes)
hypertension <- summarize_cat(dat2, hypertension)
bind_rows(gender, race, smoking, diabetes, hypertension) %>%
  knitr::kable()
```

Variable	Level	N	Percent
gender	0	509	50.9
gender	1	491	49.1
race	1	663	66.3
race	2	55	5.5
race	3	199	19.9
race	4	83	8.3
smoking	0	601	60.1
smoking	1	296	29.6
smoking	2	103	10.3
diabetes	0	843	84.3
diabetes	1	157	15.7
hypertension	0	544	54.4
hypertension	1	456	45.6

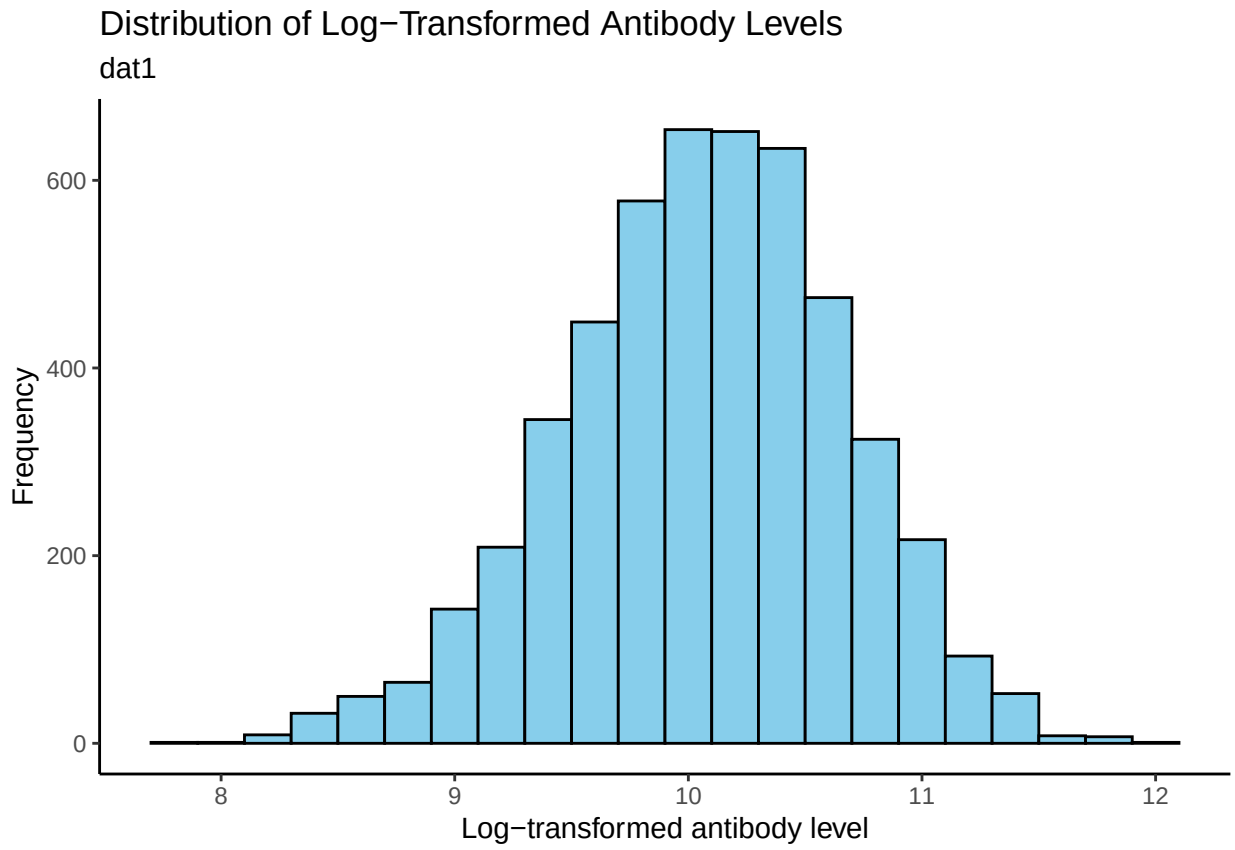
```
age = summarize_cont(dat2, age)
bmi = summarize_cont(dat2, bmi)
height = summarize_cont(dat2, height)
weight = summarize_cont(dat2, weight)
SBP = summarize_cont(dat2, SBP)
LDL = summarize_cont(dat2, LDL)
time = summarize_cont(dat2, time)
log_anti = summarize_cont(dat2, log_antibody)
bind_rows(age, bmi, height, weight, SBP, LDL, time, log_anti) %>%
  knitr::kable()
```

Variable	Median	IQR
age	60.0	[57, 63]
bmi	27.6	[25.8, 29.6]

Variable	Median	IQR
height	170.2	[166.1, 174.2]
weight	80.2	[75.3, 84.4]
SBP	130.0	[124, 135]
LDL	112.0	[96, 124]
time	171.0	[140, 205]
log_antibody	9.9	[9.5, 10.3]

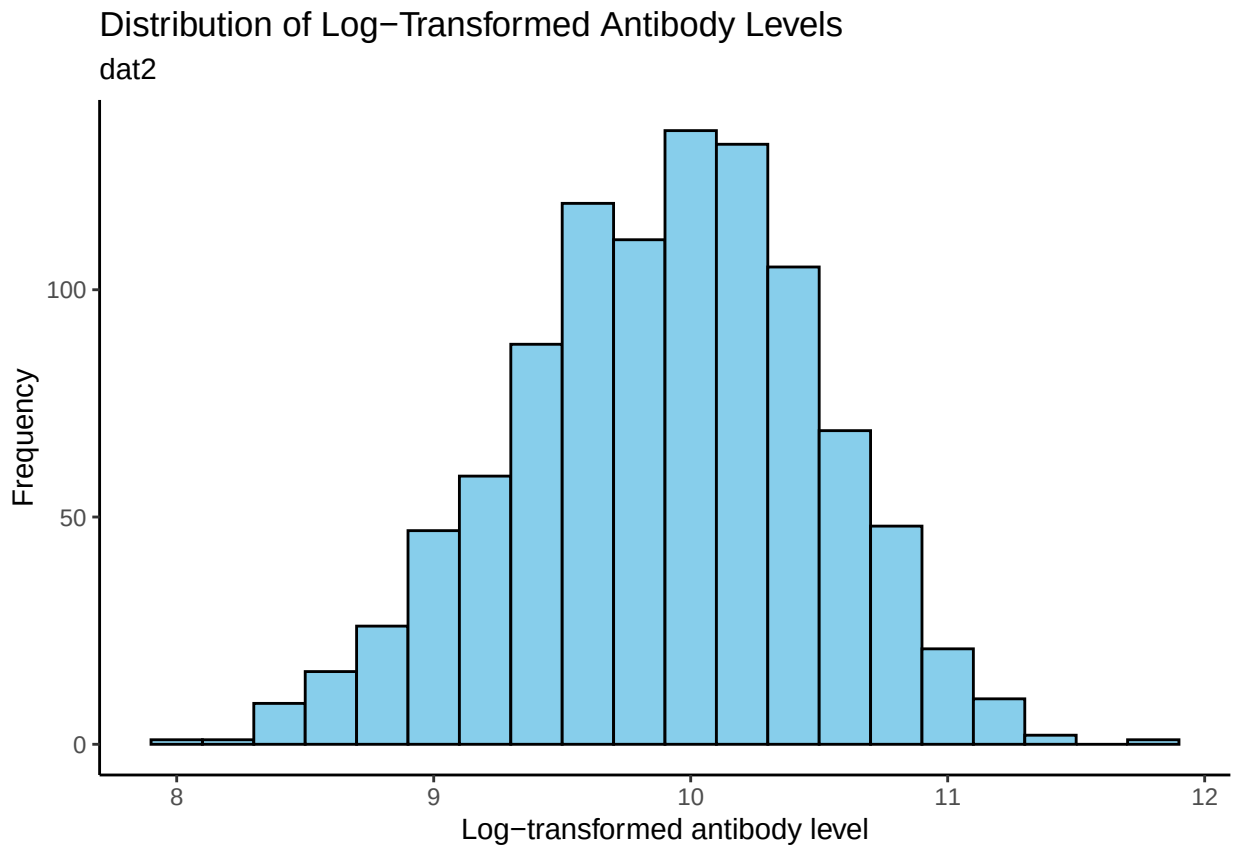
Distribution of antibody levels

```
ggplot(dat1, aes(x = log_antibody)) +
  geom_histogram(binwidth = 0.2, fill = 'skyblue', color = 'black') +
  labs(x = "Log-transformed antibody level",
       y = "Frequency",
       title = "Distribution of Log-Transformed Antibody Levels",
       subtitle = "dat1") +
  theme_classic()
```




```
ggplot(dat2, aes(x = log_antibody)) +
  geom_histogram(binwidth = 0.2, fill = 'skyblue', color = 'black') +
  labs(x = "Log-transformed antibody level",
       y = "Frequency",
```

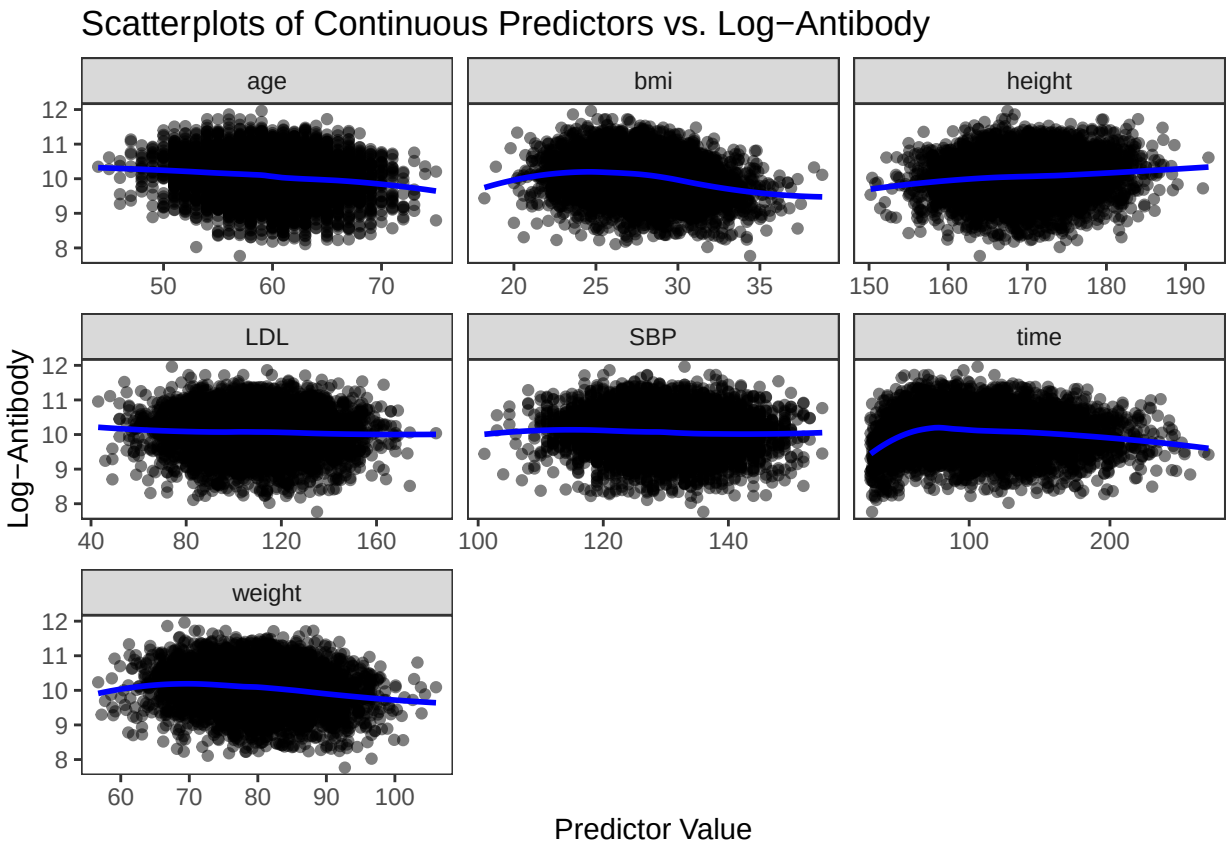
```
       title = "Distribution of Log-Transformed Antibody Levels",
       subtitle = "dat2") +
  theme_classic()
```



Scatterplots of Continuous Predictors vs. Log-Antibody

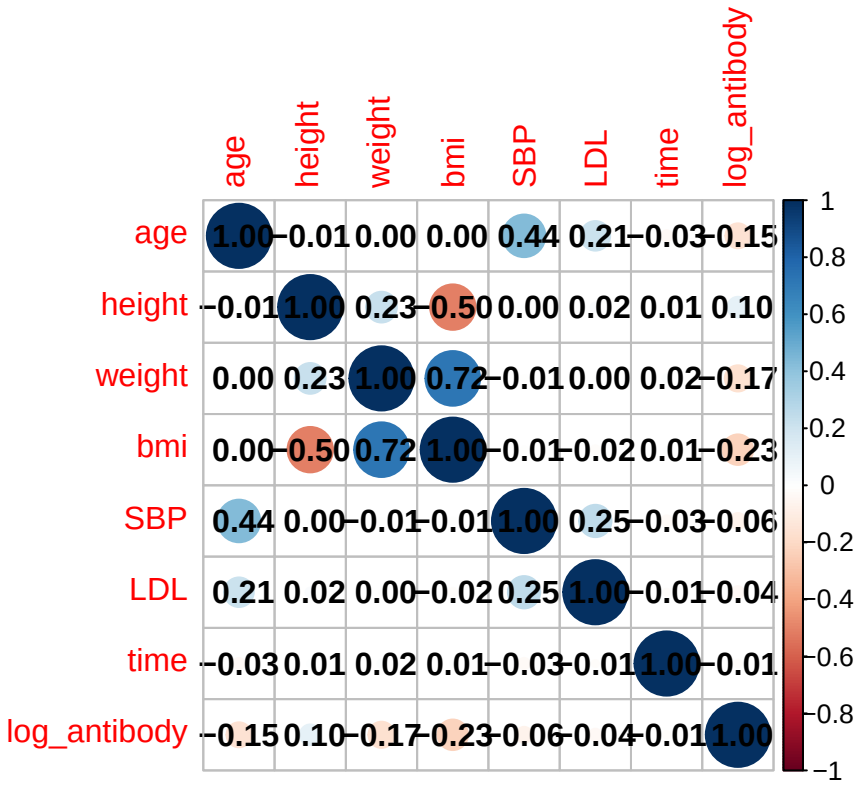
```
dat1 %>%
  select(age, height, weight, bmi, SBP, LDL, time, log_antibody) %>%
  pivot_longer(
    cols = c(age,  height, weight, bmi, SBP, LDL, time),
    names_to = "predictor",
    values_to = "value"
  ) %>%
  ggplot(aes(x = value, y = log_antibody)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "loess", se = FALSE, color = "blue") +
  facet_wrap(~ predictor, scales = "free_x") +
  labs(
    x = "Predictor Value",
    y = "Log-Antibody",
    title = "Scatterplots of Continuous Predictors vs. Log-Antibody"
  ) + theme_test()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Correlation Matrix of Continuous Variables

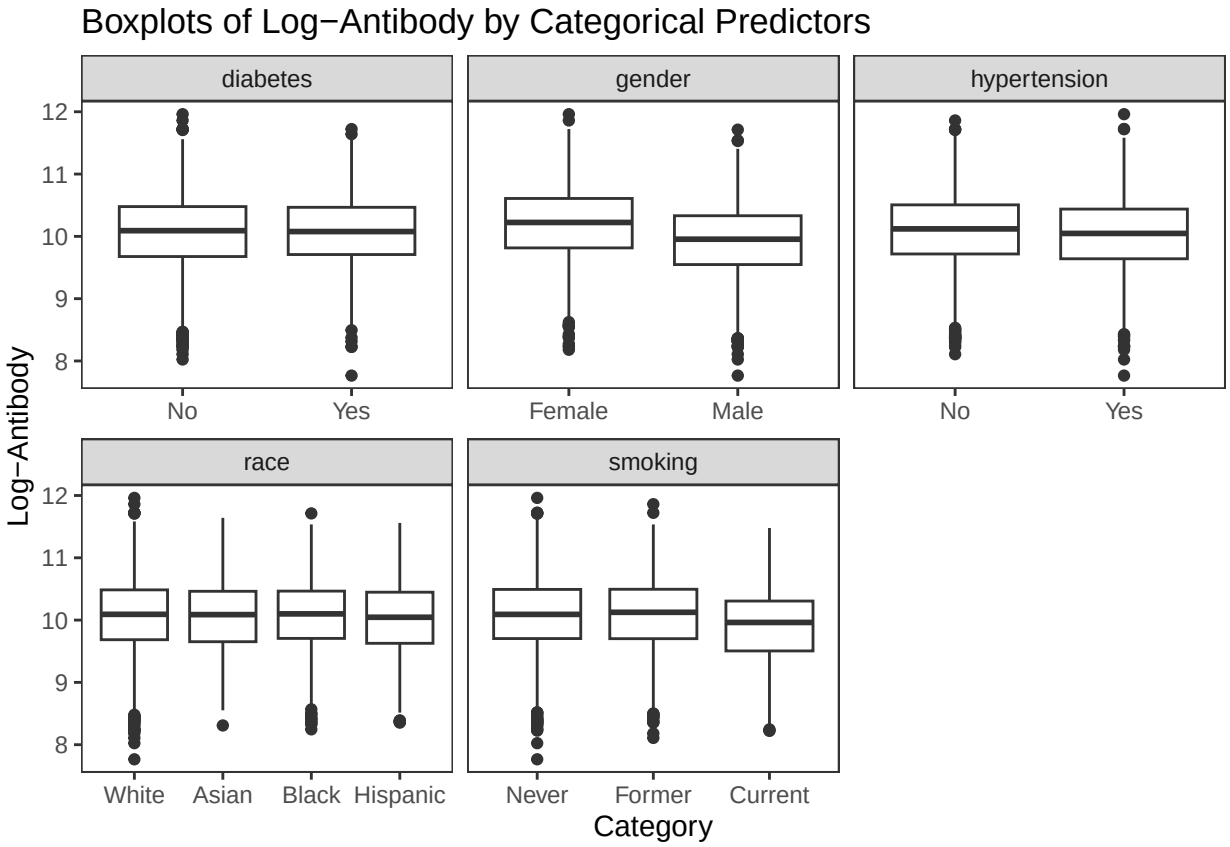
```
dat1 %>%
  select(age, height, weight, bmi, SBP, LDL, time, log_antibody) %>%
  cor(use = "complete.obs") %>%
  corrplot(type = "full",
           title = "Correlation Matrix of Continuous Variables",
           addCoef.col = "black",
           mar = c(0,0,2,0))
```



Boxplots of log_antibody by categorical variables

```
dat1 %>%
  select(log_antibody, gender, race, smoking, diabetes, hypertension) %>%
  mutate(
    gender = factor(gender,
                    levels = c(0, 1),
                    labels = c("Female", "Male")),
    race = factor(race,
                  levels = c(1, 2, 3, 4),
                  labels = c("White", "Asian", "Black", "Hispanic")),
    smoking = factor(smoking,
                     levels = c(0, 1, 2),
                     labels = c("Never", "Former", "Current")),
    diabetes = factor(diabetes,
                      levels = c(0, 1),
                      labels = c("No", "Yes")),
    hypertension = factor(hypertension,
                          levels = c(0, 1),
                          labels = c("No", "Yes"))
  ) %>%
  pivot_longer(
    cols = c(gender, race, smoking, diabetes, hypertension),
    names_to = "predictor",
```

```
    values_to = "category"
  ) %>%
  ggplot(aes(x = category, y = log_antibody)) +
  geom_boxplot() +
  facet_wrap(~ predictor, scales = "free_x") +
  labs(
    x = "Category",
    y = "Log-Antibody",
    title = "Boxplots of Log-Antibody by Categorical Predictors"
  ) + theme_test()
```

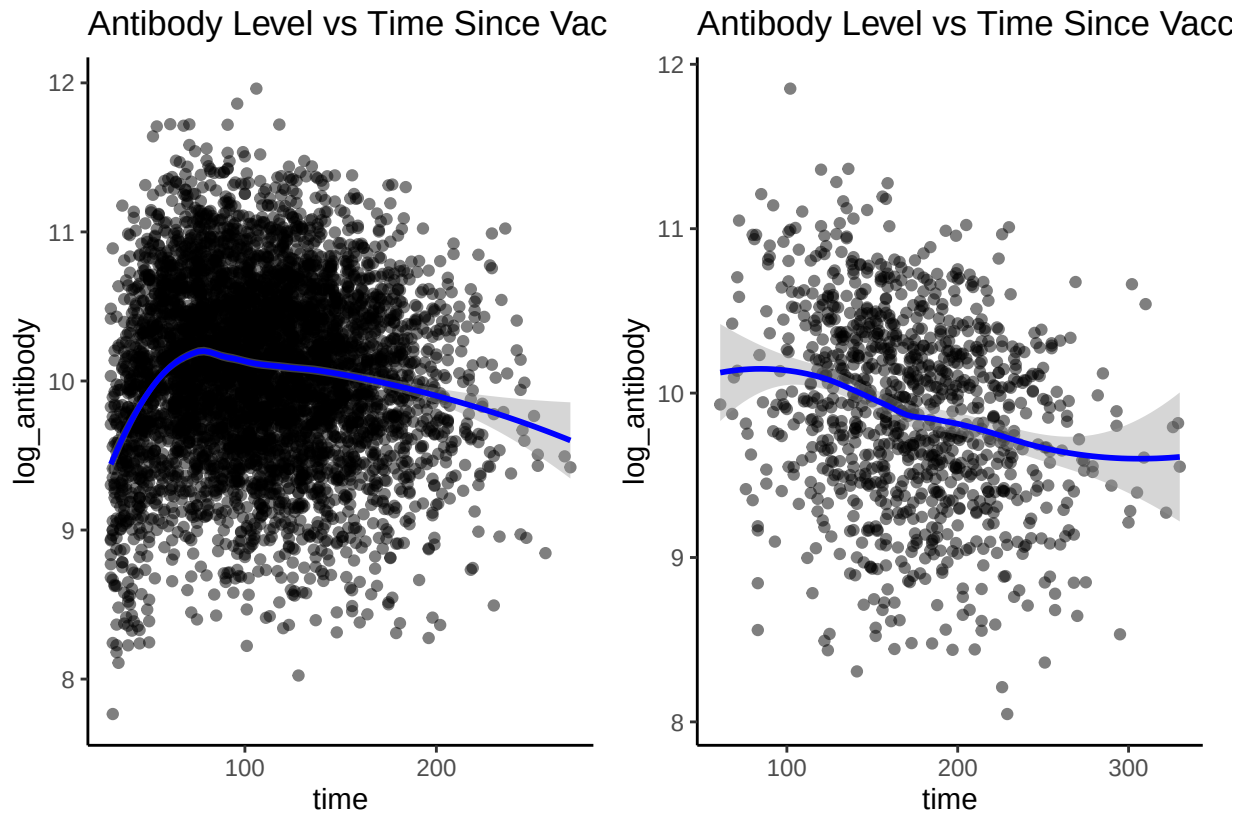


Antibody level and time

```
p1 <- ggplot(dat1, aes(x = time, y = log_antibody)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "loess", color = "blue") +
  labs(title = "Antibody Level vs Time Since Vaccination (dat1)") +
  theme_classic()

p2 <- ggplot(dat2, aes(x = time, y = log_antibody)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "loess", color = "blue") +
  labs(title = "Antibody Level vs Time Since Vaccination (dat2)") +
```

```
  theme_classic()

p1+p2
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



New datasets after adjustment

```
dat1 =
  dat1 |>
  select(-bmi)

dat2 =
  dat2 |>
  select(-bmi)
```

Models Building

cross-validation

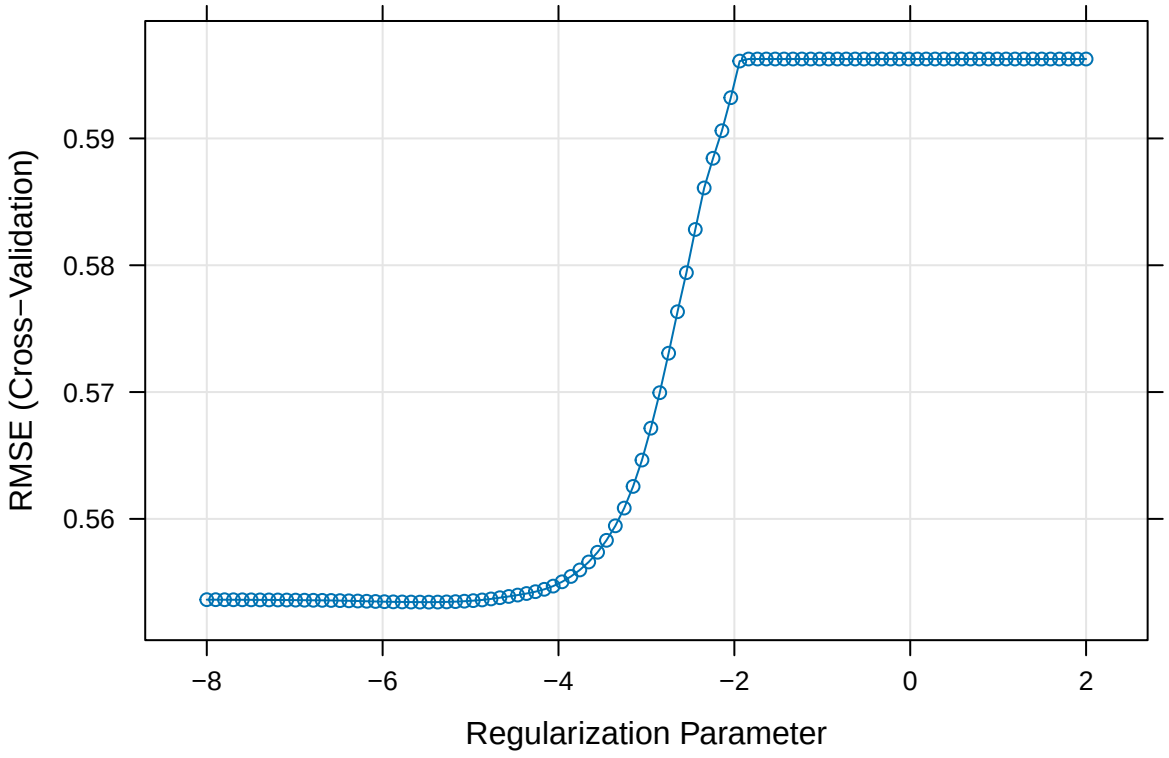
```r
set.seed(123)
ctrl = trainControl(method = "cv", number = 10)
```

LASSO

```r
set.seed(123)
model_lasso <- train(
  log_antibody ~ .,
  data = dat1,
  method = "glmnet",
  trControl = ctrl,
  tuneGrid = expand.grid(alpha = 1,
                         lambda = exp(seq(2, -8, length = 100))

))
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```r
plot(model_lasso, xTrans = log)
```



```r
model_lasso$bestTune
```

```
##    alpha      lambda
## 26     1 0.004191287
```

```r
coef(model_lasso$finalModel, model_lasso$bestTune$lambda)
```

```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                            s1
## (Intercept)  10.2861226400
## age          -0.0192317177
## gender       -0.2860853991
## race2         .
## race3         .
## race4        -0.0266918715
## smoking1      0.0166767858
## smoking2     -0.1773402273
## height        0.0142828656
## weight       -0.0165288840
## diabetes      0.0001036127
## hypertension  .
## SBP           .
## LDL           .
## time         -0.0001902478
```

GAM

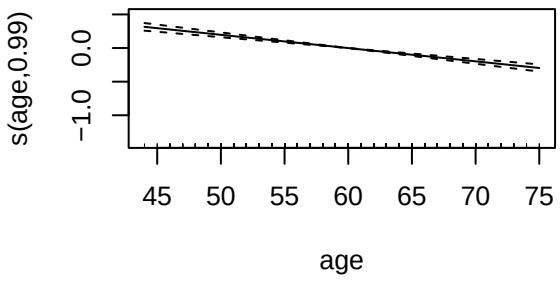
```r
set.seed(123)
model_gam <- train(
  log_antibody ~ .,
  data = dat1,
  method = "gam",
  trControl = ctrl
)

model_gam$finalModel
```

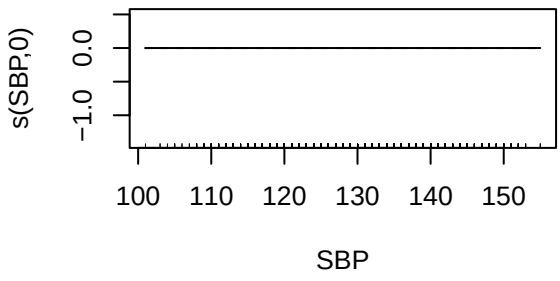
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender + race2 + race3 + race4 + smoking1 + smoking2 +
##     diabetes + hypertension + s(age) + s(SBP) + s(LDL) + s(time) +
##     s(height) + s(weight)
##
## Estimated degrees of freedom:
## 0.99 0.00 0.00 7.87 2.30 3.20  total = 23.37
##
## GCV score: 0.2837087
```

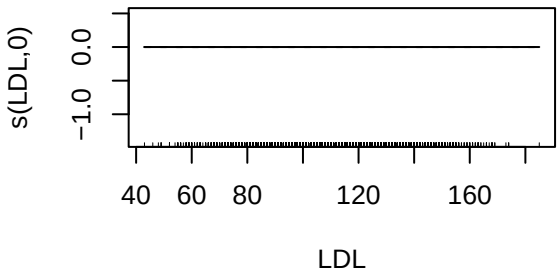
```
model_gam$bestTune
```

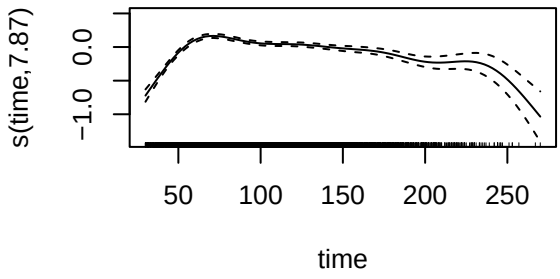
```
##   select method
## 2   TRUE GCV.Cp
```

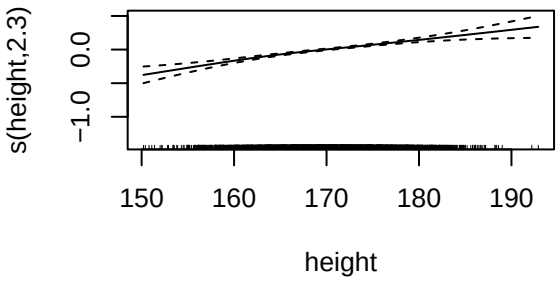
```
par(mfrow = c(2,2))
plot(model_gam$finalModel)
```

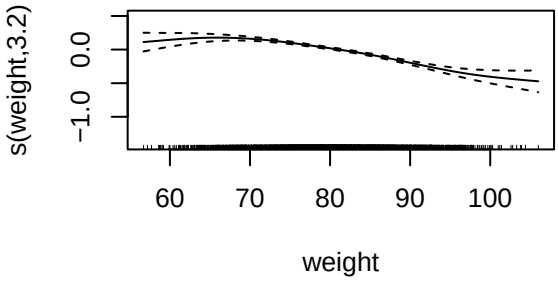






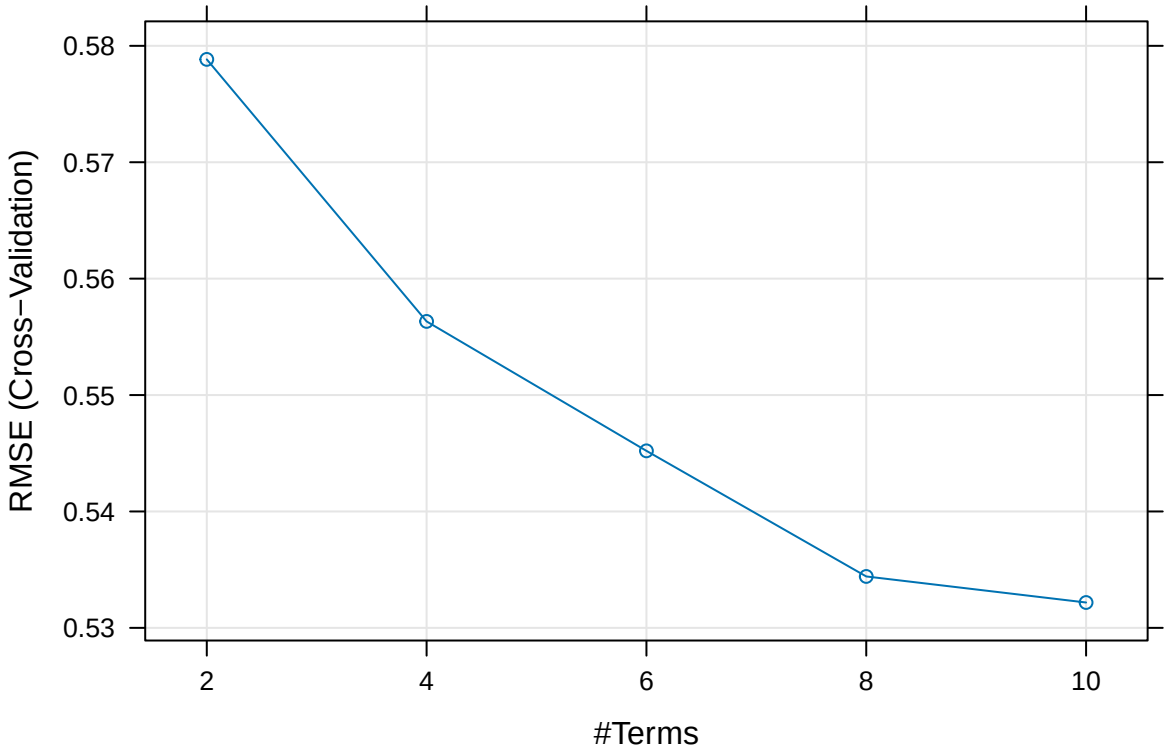






MARS

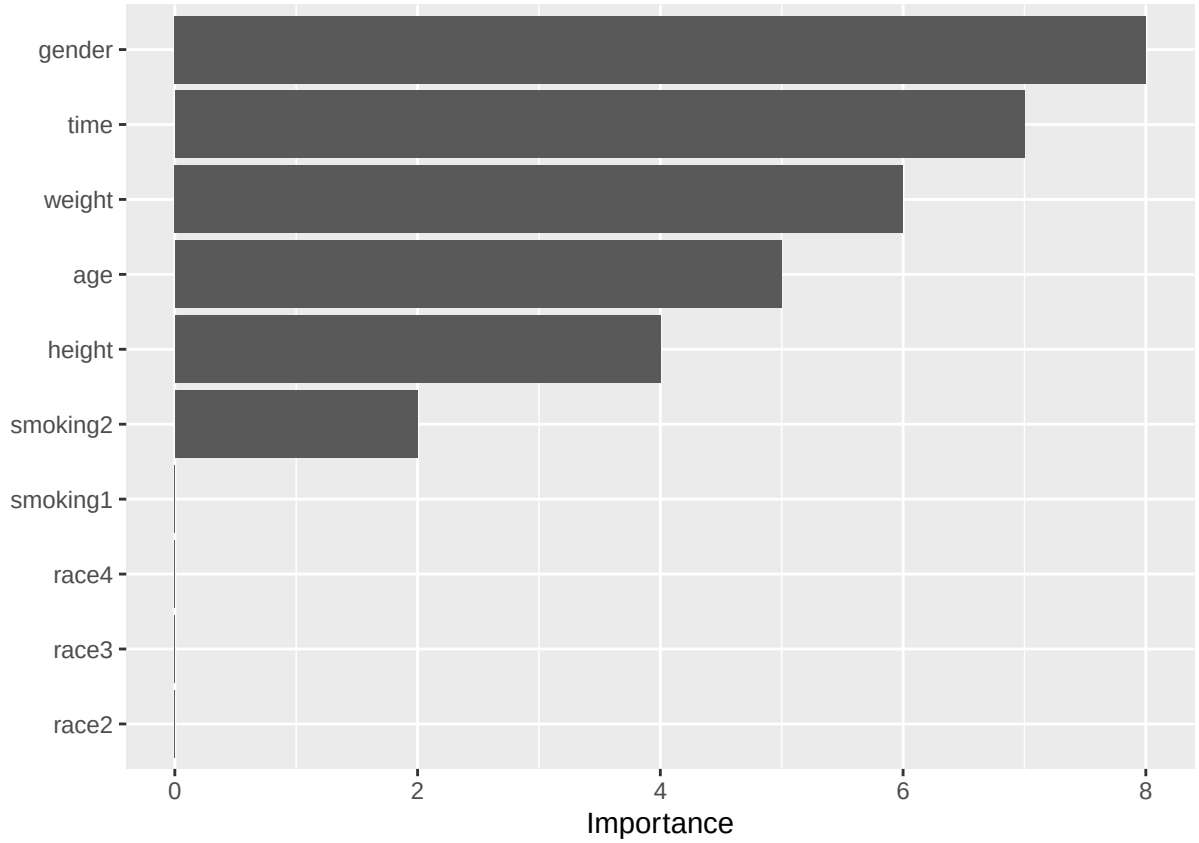
```
set.seed(123)
model_mars <- train(
  log_antibody ~ .,
  data = dat1,
  method = "earth",
  trControl = ctrl,
  tuneLength = 5
)

plot(model_mars)
```



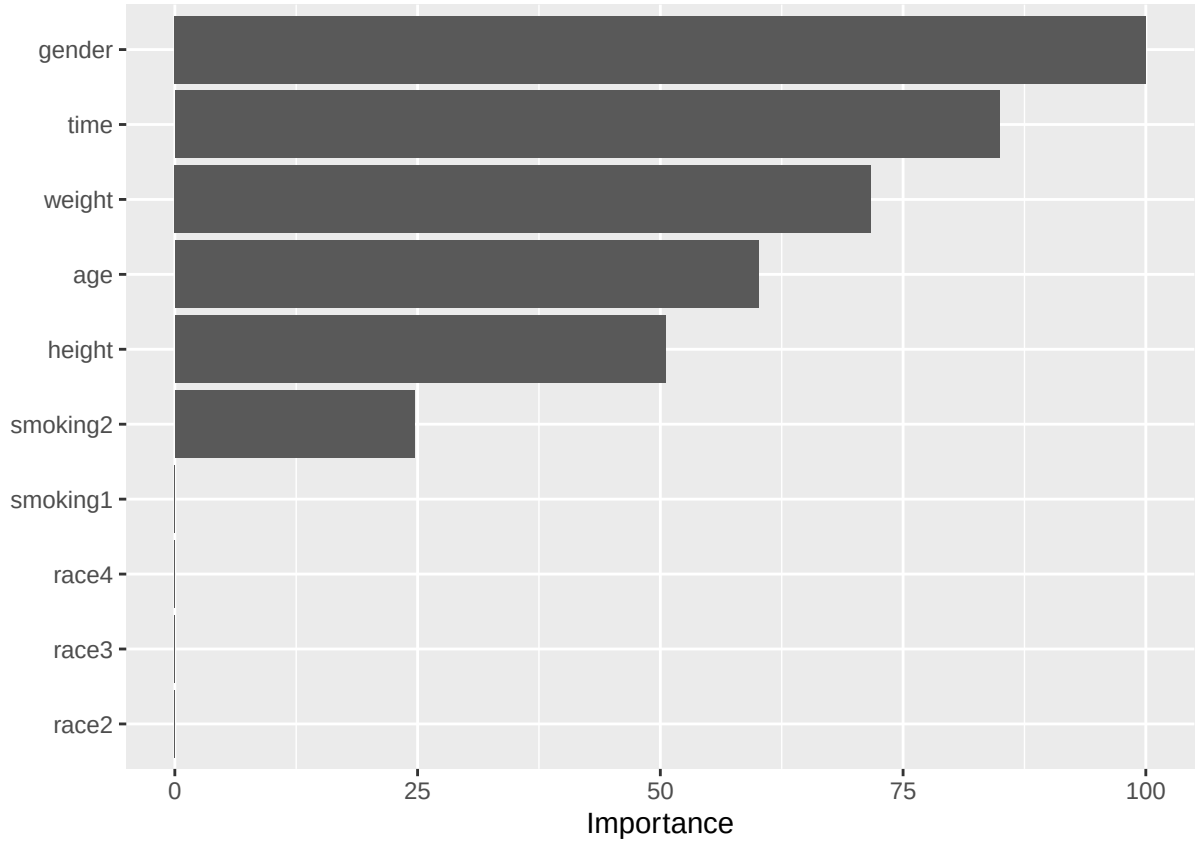

```
coef(model_mars$finalModel)
```

```
##     (Intercept)      h(time-57)      h(57-time)          gender h(weight-71.3)
##    10.265741995    -0.002273742    -0.033751205    -0.293358075    -0.019491531
##       h(70-age)  h(height-162)  h(162-height)        smoking2
##     0.020385000     0.013720849    -0.032693796    -0.201394423
```

```
vip(model_mars$finalModel, type = "nsubsets")
```




```
vip(model_mars$finalModel, type = "rss")
```



Predictions and Model Evaluation

```
set.seed(123)

pred_lasso <- predict(model_lasso, newdata = dat2)
pred_mars  <- predict(model_mars, newdata = dat2)
pred_gam   <- predict(model_gam, newdata = dat2)

resample = resamples(list(lasso = model_lasso, gam = model_gam, mars = model_mars))
summary(resample)
```

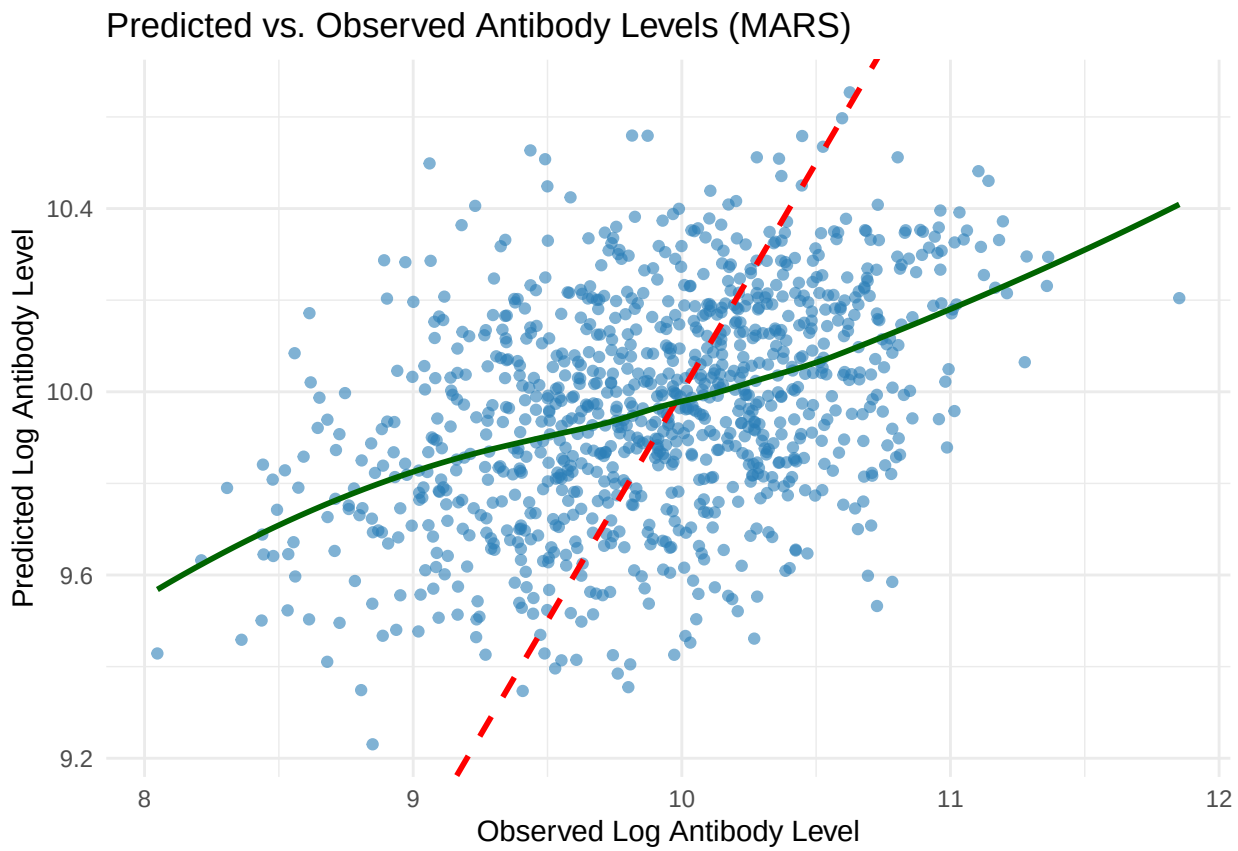
```
##
## Call:
## summary.resamples(object = resample)
##
## Models: lasso, gam, mars
## Number of resamples: 10
##
## MAE
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lasso 0.4285576 0.4336399 0.4376162 0.4411608 0.4454918 0.4726699    0
## gam   0.4097911 0.4166471 0.4221812 0.4260111 0.4276756 0.4654718    0
## mars  0.4105375 0.4170560 0.4228003 0.4256732 0.4262580 0.4660933    0
```

```
##
## RMSE
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lasso 0.5371499 0.5432117 0.5537788 0.5534342 0.5582848 0.5842641    0
## gam   0.5110780 0.5219839 0.5290748 0.5325277 0.5398018 0.5766493    0
## mars  0.5122879 0.5201509 0.5292147 0.5321795 0.5392713 0.5780741    0
##
## Rsquared
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lasso 0.09175865 0.1322362 0.1405894 0.1400107 0.1471445 0.1875308    0
## gam   0.14410267 0.1854427 0.2103004 0.2037073 0.2261907 0.2463613    0
## mars  0.14040983 0.1912325 0.2135016 0.2048437 0.2257523 0.2487932    0
```

```r
# Visualization on predicteed vs. observed
ggplot(data.frame(Observed = dat2$log_antibody, Predicted = pred_mars),
       aes(x = dat2$log_antibody, y = pred_mars)) +
  geom_point(alpha = 0.6, color = "#2c7fb8") +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed", size = 1) +
  geom_smooth(method = "loess", se = FALSE, color = "darkgreen") +
  labs(
    title = "Predicted vs. Observed Antibody Levels (MARS)",
    x = "Observed Log Antibody Level",
    y = "Predicted Log Antibody Level"
  ) +
  theme_minimal()
```

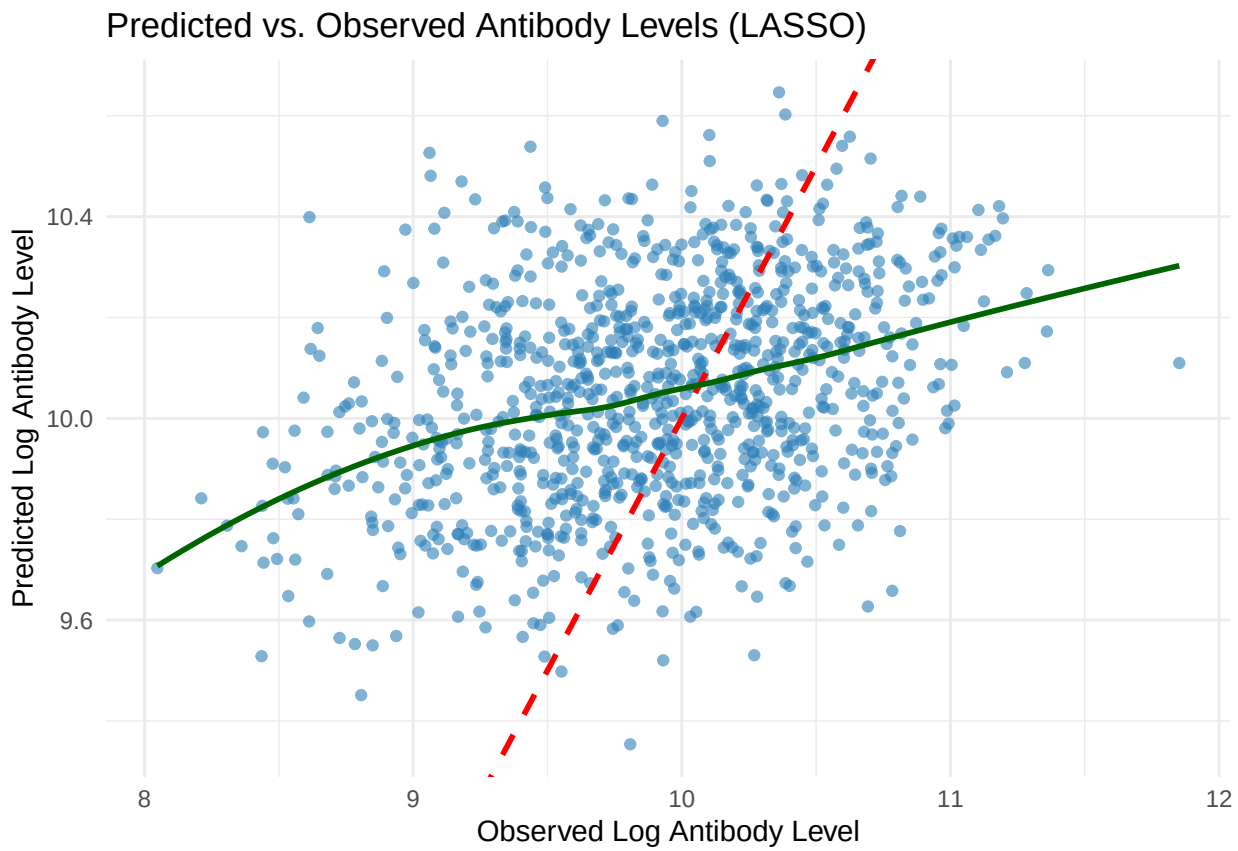
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



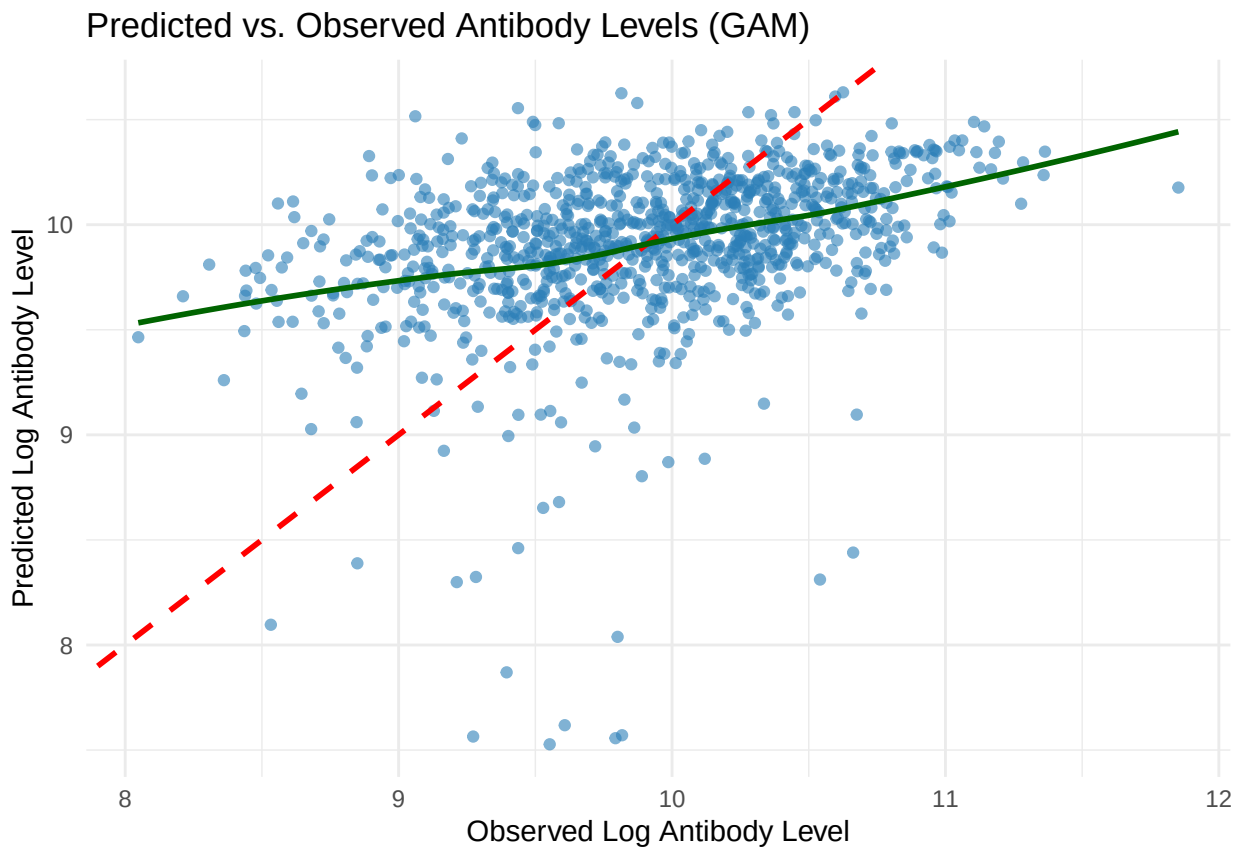

```
ggplot(data.frame(Observed = dat2$log_antibody, Predicted = pred_lasso),
       aes(x = dat2$log_antibody, y = pred_lasso)) +
  geom_point(alpha = 0.6, color = "#2c7fb8") +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed", size = 1) +
  geom_smooth(method = "loess", se = FALSE, color = "darkgreen") +
  labs(
    title = "Predicted vs. Observed Antibody Levels (LASSO)",
    x = "Observed Log Antibody Level",
    y = "Predicted Log Antibody Level"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



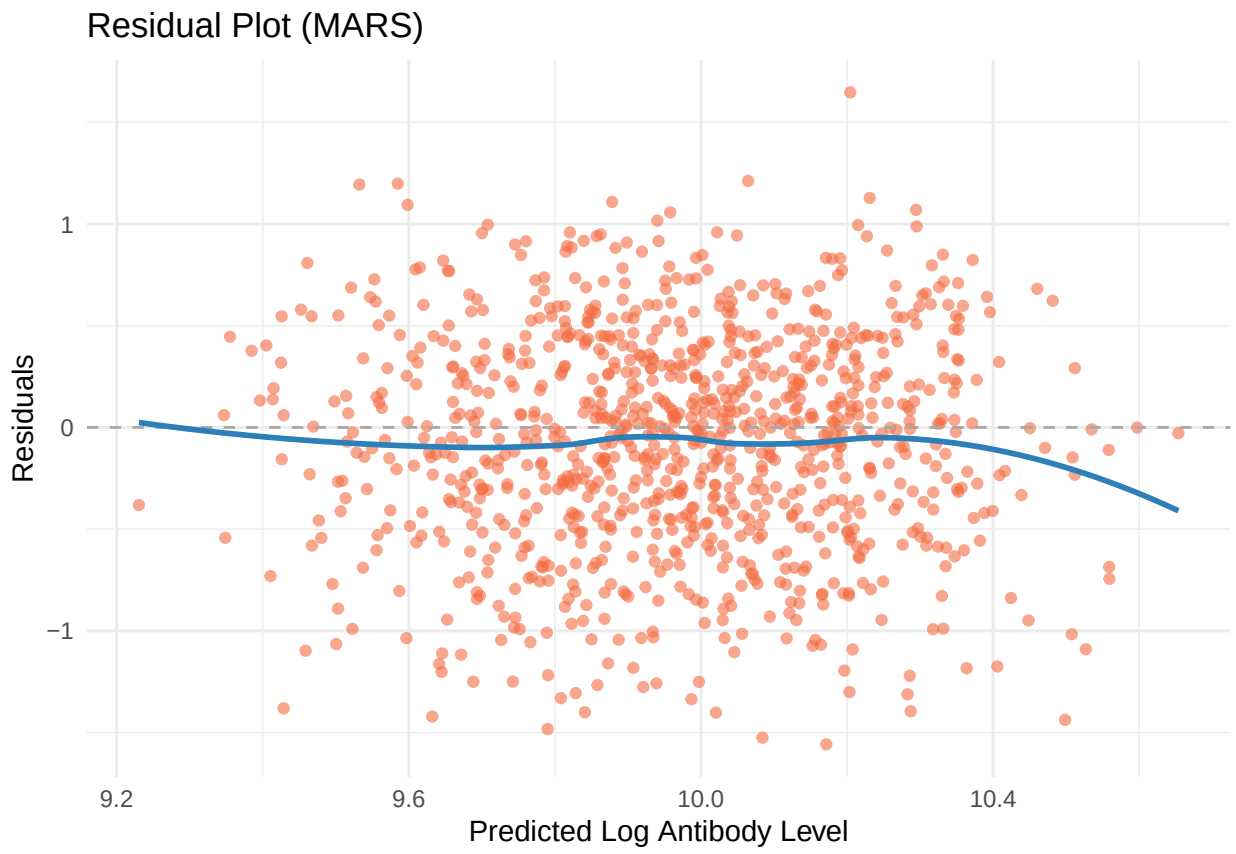

```
ggplot(data.frame(Observed = dat2$log_antibody, Predicted = pred_gam),
       aes(x = dat2$log_antibody, y = pred_gam)) +
  geom_point(alpha = 0.6, color = "#2c7fb8") +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed", size = 1) +
  geom_smooth(method = "loess", se = FALSE, color = "darkgreen") +
  labs(
    title = "Predicted vs. Observed Antibody Levels (GAM)",
    x = "Observed Log Antibody Level",
    y = "Predicted Log Antibody Level"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



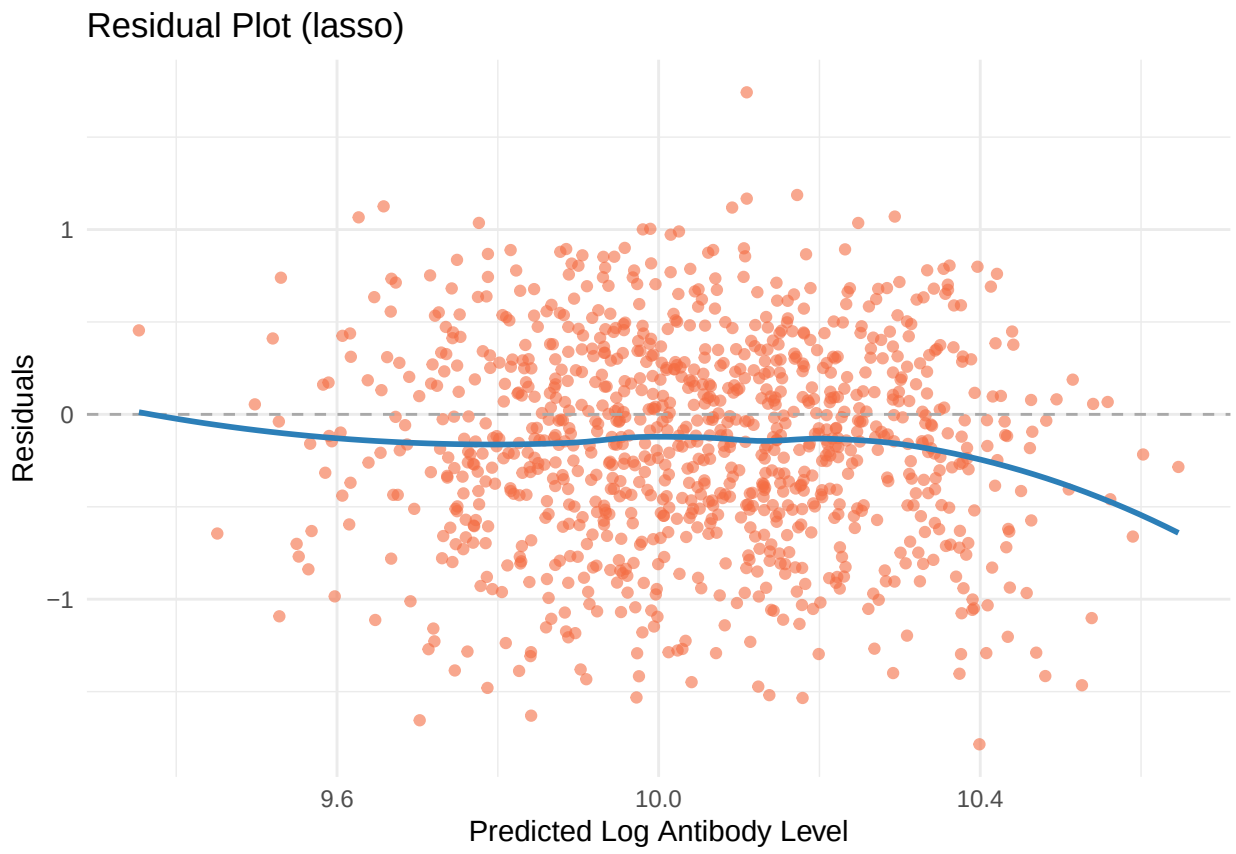

```
# Residual Plot
residual_mars <- dat2$log_antibody - pred_mars
residual_lasso <- dat2$log_antibody - pred_lasso
residual_gam <- dat2$log_antibody - pred_gam

ggplot(data.frame(Predicted = pred_mars, Residuals = residual_mars),
       aes(x = pred_mars, y = residual_mars)) +
  geom_point(alpha = 0.6, color = "#f46d43") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "darkgray") +
  geom_smooth(method = "loess", se = FALSE, color = "#2c7fb8") +
  labs(
    title = "Residual Plot (MARS)",
    x = "Predicted Log Antibody Level",
    y = "Residuals"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



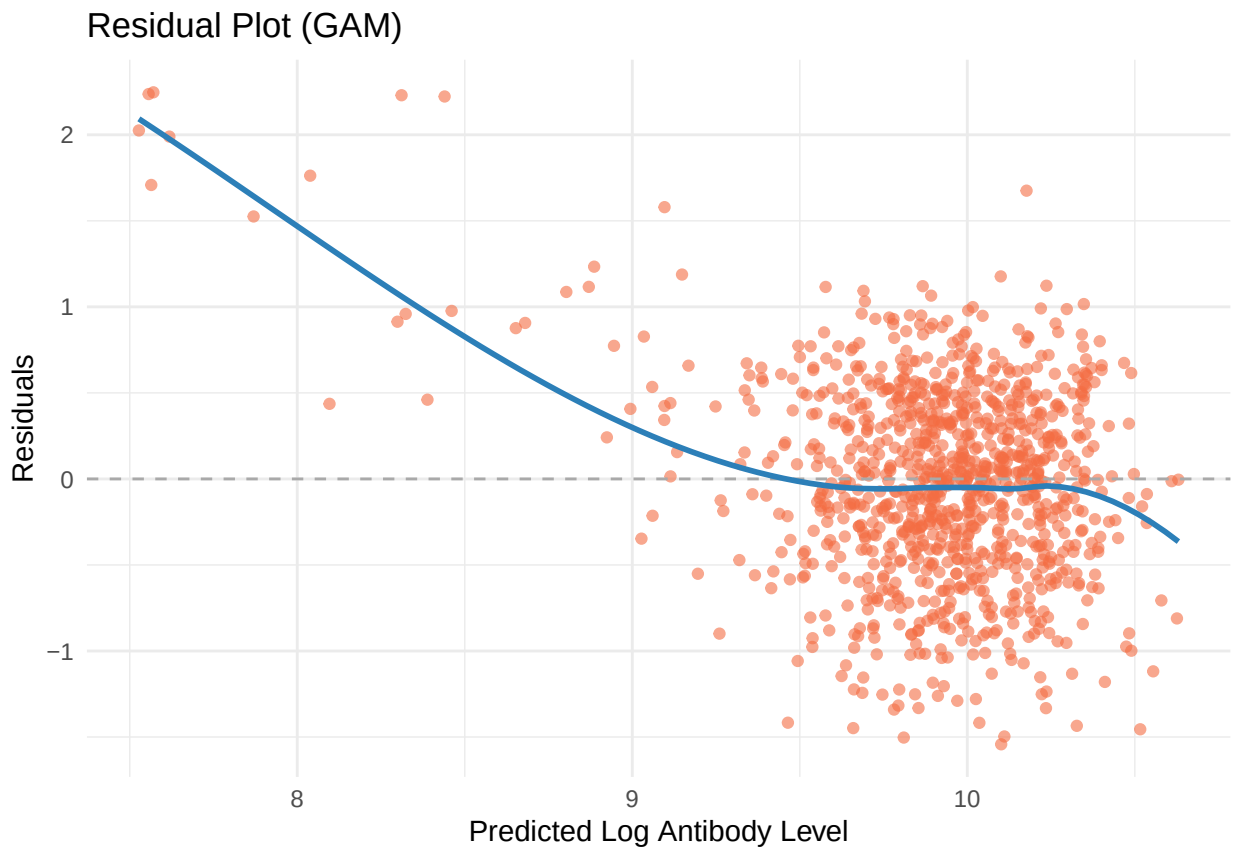

```
ggplot(data.frame(Predicted = pred_lasso, Residuals = residual_lasso),
       aes(x = pred_lasso, y = residual_lasso)) +
  geom_point(alpha = 0.6, color = "#f46d43") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "darkgray") +
  geom_smooth(method = "loess", se = FALSE, color = "#2c7fb8") +
  labs(
    title = "Residual Plot (lasso)",
    x = "Predicted Log Antibody Level",
    y = "Residuals"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(data.frame(Predicted = pred_gam, Residuals = residual_gam),
       aes(x = pred_gam, y = residual_gam)) +
  geom_point(alpha = 0.6, color = "#f46d43") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "darkgray") +
  geom_smooth(method = "loess", se = FALSE, color = "#2c7fb8") +
  labs(
    title = "Residual Plot (GAM)",
    x = "Predicted Log Antibody Level",
    y = "Residuals"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



lasso model has the lowest mean RMSE (0.5534342)