

Midterm Project

Minghe Wang (mw3845), Zebang Zhang (zz3309), Xuanyu Guo (xg2451)

2025-03-30

Exploratory Data Analysis

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

# no missing data
all(is.na(dat1))

## [1] FALSE
all(is.na(dat2))

## [1] FALSE
ifelse(all(names(dat1) == names(dat2)), "train and test data have same structure", "train and test data")

## [1] "train and test data have same structure"
str(dat1)

## 'data.frame': 5000 obs. of 14 variables:
## $ id : int 1 2 3 4 5 6 7 8 9 10 ...
## $ age : num 50 71 58 63 56 59 67 62 60 64 ...
## $ gender : int 0 1 1 0 1 1 0 1 0 1 ...
## $ race : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 3 4 1 4 1 ...
## $ smoking : Factor w/ 3 levels "0","1","2": 1 1 2 1 1 1 1 1 1 1 ...
## $ height : num 176 176 169 167 163 ...
## $ weight : num 68.3 69.6 76.9 90 83.9 86.8 91.4 87.7 85.7 76.6 ...
## $ bmi : num 22 22.6 27 32.1 31.7 30.8 29.7 28.1 29 31.5 ...
## $ diabetes : int 0 0 0 0 0 0 0 0 0 0 ...
## $ hypertension: num 0 1 0 1 0 1 1 0 0 1 ...
## $ SBP : num 130 149 127 138 123 132 133 130 129 134 ...
## $ LDL : num 82 129 101 93 97 108 89 96 120 135 ...
## $ time : num 76 82 168 105 193 143 63 78 61 88 ...
## $ log_antibody: num 10.65 9.89 10.9 9.91 9.56 ...

# The 'id' column has no actual meaning, so we remove it.
dat1 <- dat1 %>%
  select(-id)

dat2 <- dat2 %>%
  select(-id)

# Convert categorical variables to labeled factors
convert_factors <- function(df) {
```

```

df %>%
  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"))
  )
}

dat1 <- convert_factors(dat1)
dat2 <- convert_factors(dat2)

```

Univariate analysis(continous & categorical)

```

continuous_var <- dat1 %>%
  select(age, height, weight, bmi, SBP, LDL, time)

categorical_var <- dat1 %>%
  select(gender, race, smoking, diabetes, hypertension)

# ---- Continuous Variables ----
# summary
summary(continuous_var)

```

##	age	height	weight	bmi
##	Min. :44.00	Min. :150.2	Min. : 56.70	Min. :18.20
##	1st Qu.:57.00	1st Qu.:166.1	1st Qu.: 75.40	1st Qu.:25.80
##	Median :60.00	Median :170.1	Median : 80.10	Median :27.60
##	Mean :59.97	Mean :170.1	Mean : 80.11	Mean :27.74
##	3rd Qu.:63.00	3rd Qu.:174.2	3rd Qu.: 84.90	3rd Qu.:29.50
##	Max. :75.00	Max. :192.9	Max. :106.00	Max. :38.80

##	SBP	LDL	time
##	Min. :101.0	Min. : 43.0	Min. : 30.0
##	1st Qu.:124.0	1st Qu.: 96.0	1st Qu.: 76.0
##	Median :130.0	Median :110.0	Median :106.0
##	Mean :129.9	Mean :109.9	Mean :108.9
##	3rd Qu.:135.0	3rd Qu.:124.0	3rd Qu.:138.0
##	Max. :155.0	Max. :185.0	Max. :270.0

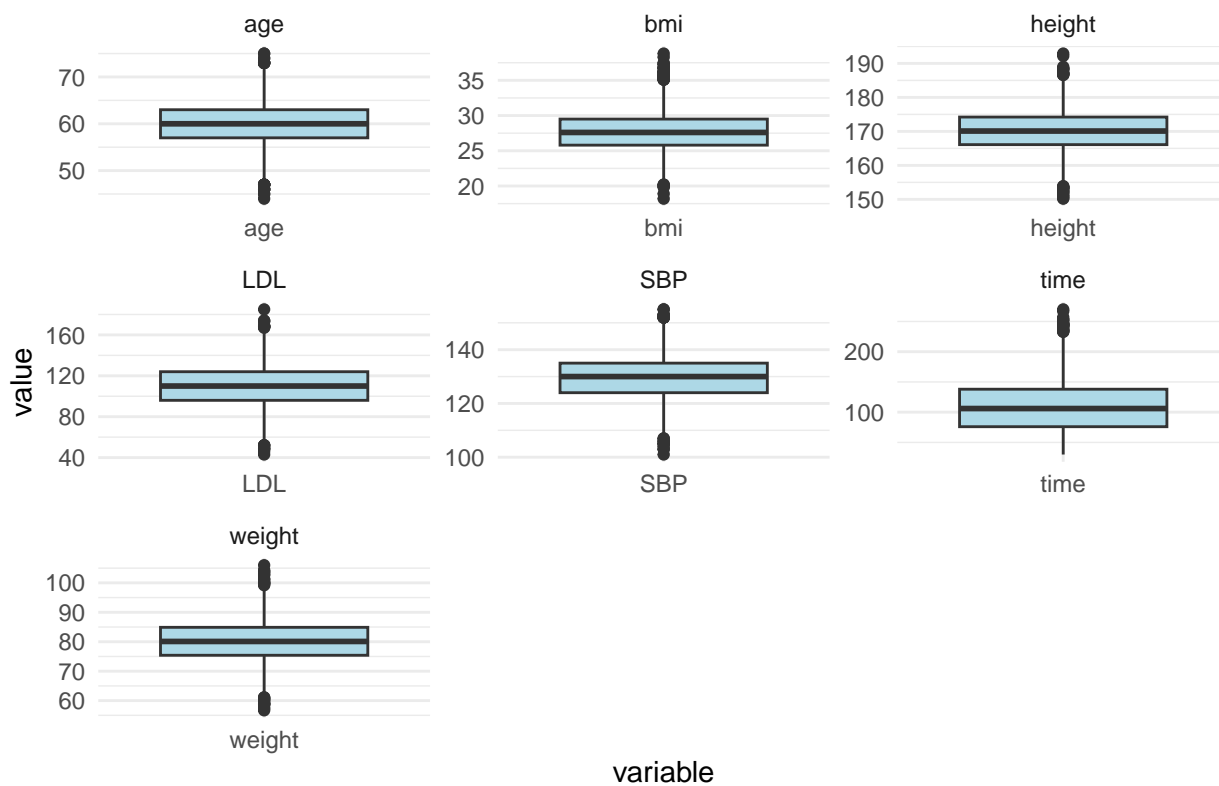
```

# boxplots
continuous_var_long <- continuous_var %>%
  tidyr::pivot_longer(cols = everything(), names_to = "variable", values_to = "value")

ggplot(continuous_var_long, aes(x = variable, y = value)) +
  geom_boxplot(fill = "lightblue") +
  facet_wrap(~variable, scales = "free", ncol = 3) +
  theme_minimal() +
  labs(title = "Boxplots of Continuous Variables")

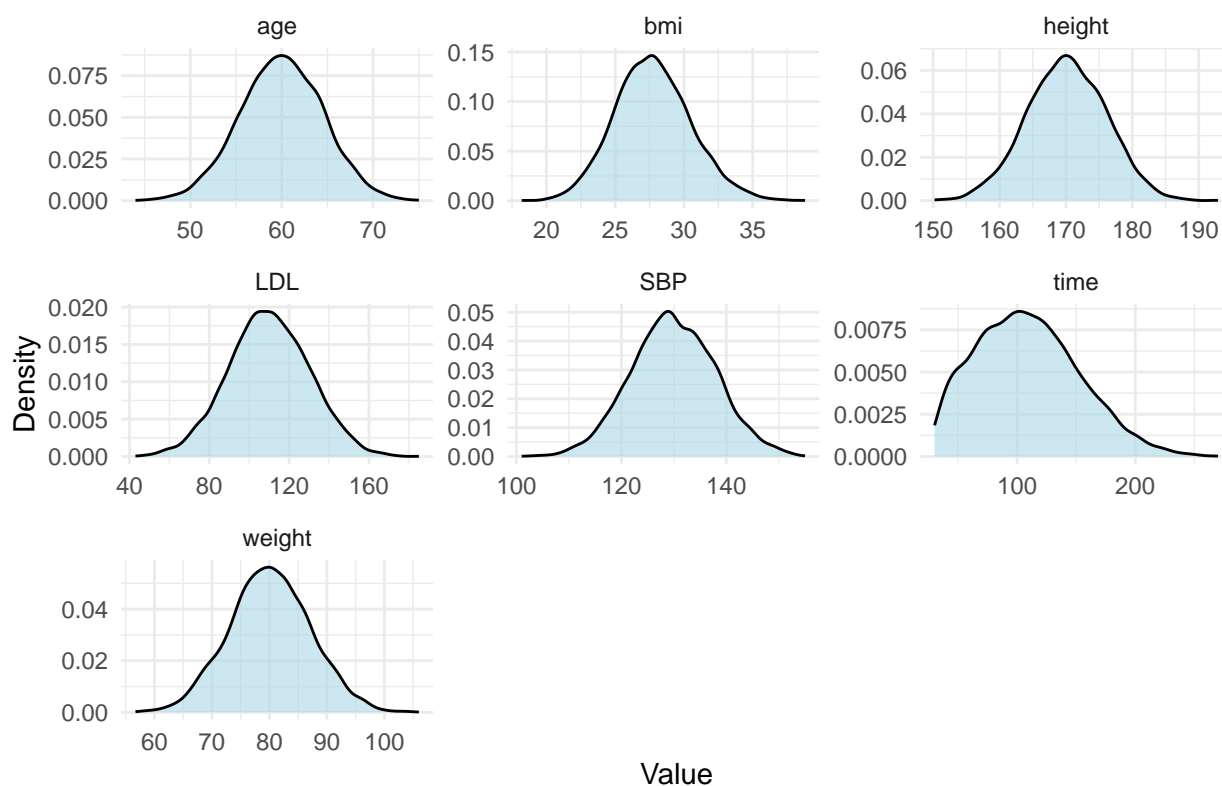
```

Boxplots of Continuous Variables



```
# density plots
ggplot(continuous_var_long, aes(x = value)) +
  geom_density(fill = "lightblue", alpha = 0.6) +
  facet_wrap(~variable, scales = "free", ncol = 3) +
  theme_minimal() +
  labs(title = "Density Plots of Continuous Variables", x = "Value", y = "Density")
```

Density Plots of Continuous Variables



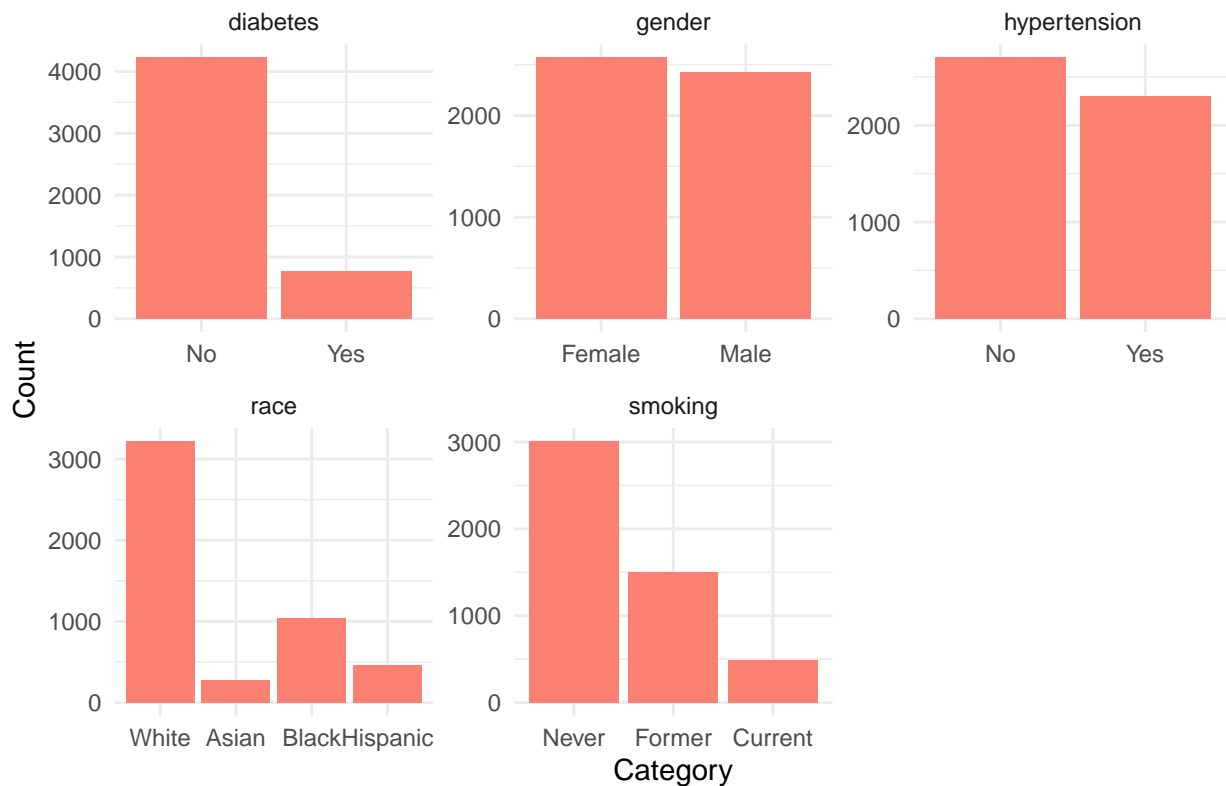
```
# ---- Categorical Variables ----
# summary
summary(categorical_var)
```

```
##      gender      race      smoking      diabetes      hypertension
## Female:2573   White   :3221   Never   :3010   No :4228   No :2702
## Male  :2427   Asian   : 278   Former :1504   Yes: 772   Yes:2298
##                                     Black  :1036   Current: 486
##                                     Hispanic: 465
```

```
# bar plots
categorical_var_long <- categorical_var %>%
  tidyr::pivot_longer(cols = everything(), names_to = "variable", values_to = "value")

ggplot(categorical_var_long, aes(x = value)) +
  geom_bar(fill = "salmon") +
  facet_wrap(~variable, scales = "free", ncol = 3) +
  theme_minimal() +
  labs(title = "Bar Plots of Categorical Variables", x = "Category", y = "Count")
```

Bar Plots of Categorical Variables



According to the box plot for continuous variables:

- **Age**, **BMI**, and **SBP** appear reasonably normally distributed, with expected ranges for an adult population; **LDL cholesterol** and **time** since vaccination show a wider range, right-skewness and some outliers, which may impact linear models.

According to the bar plot for categorical variables:

- **Gender** is fairly balanced between Female and Male;
- **Race** is skewed, with a majority of participants identifying as White (Category 1). Other racial/ethnic groups are underrepresented;
- **Smoking** status shows that the majority are never smokers (Category 0), with fewer current and former smokers;
- A large proportion of participants do not have **diabetes**;
- A moderate split exists for **hypertension**, which may contribute meaningfully to clinical outcome variation
- Demographically, the population is balanced by gender but skewed by race and smoking status.

```
# response variable `log_antibody`

# Density plot
p1 <- ggplot(dat1, aes(x = log_antibody)) +
  geom_density(fill = "skyblue", alpha = 0.5) +
  ggtitle("Density Plot of log_antibody") +
  xlab("log_antibody") +
  theme_minimal()
```

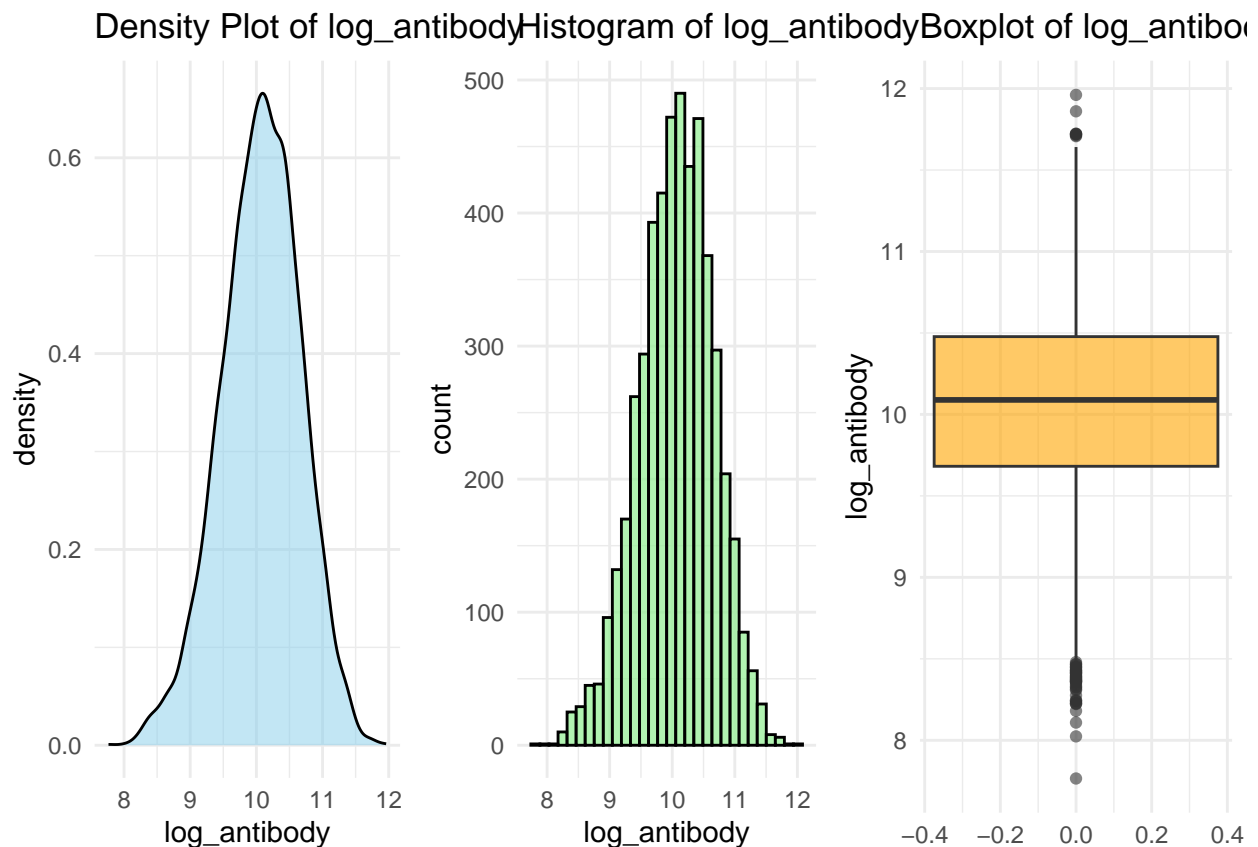
```

# Histogram
p2 <- ggplot(dat1, aes(x = log_antibody)) +
  geom_histogram(bins = 30, fill = "lightgreen", color = "black", alpha = 0.7) +
  ggtitle("Histogram of log_antibody") +
  xlab("log_antibody") +
  theme_minimal()

# Boxplot
p3 <- ggplot(dat1, aes(y = log_antibody)) +
  geom_boxplot(fill = "orange", alpha = 0.6) +
  ggtitle("Boxplot of log_antibody") +
  ylab("log_antibody") +
  theme_minimal()

grid.arrange(p1, p2, p3, ncol = 3)

```



`log_antibody` (response) appears fairly symmetrical, which supports its use as a continuous response in linear or GAM models.

Overall, we believe the response variable `log_antibody` is well-behaved, and further correlation analysis(eg. bivariate) is needed.

Next, we assess correlations and non-linear trends to guide model form.

Correlation Analysis

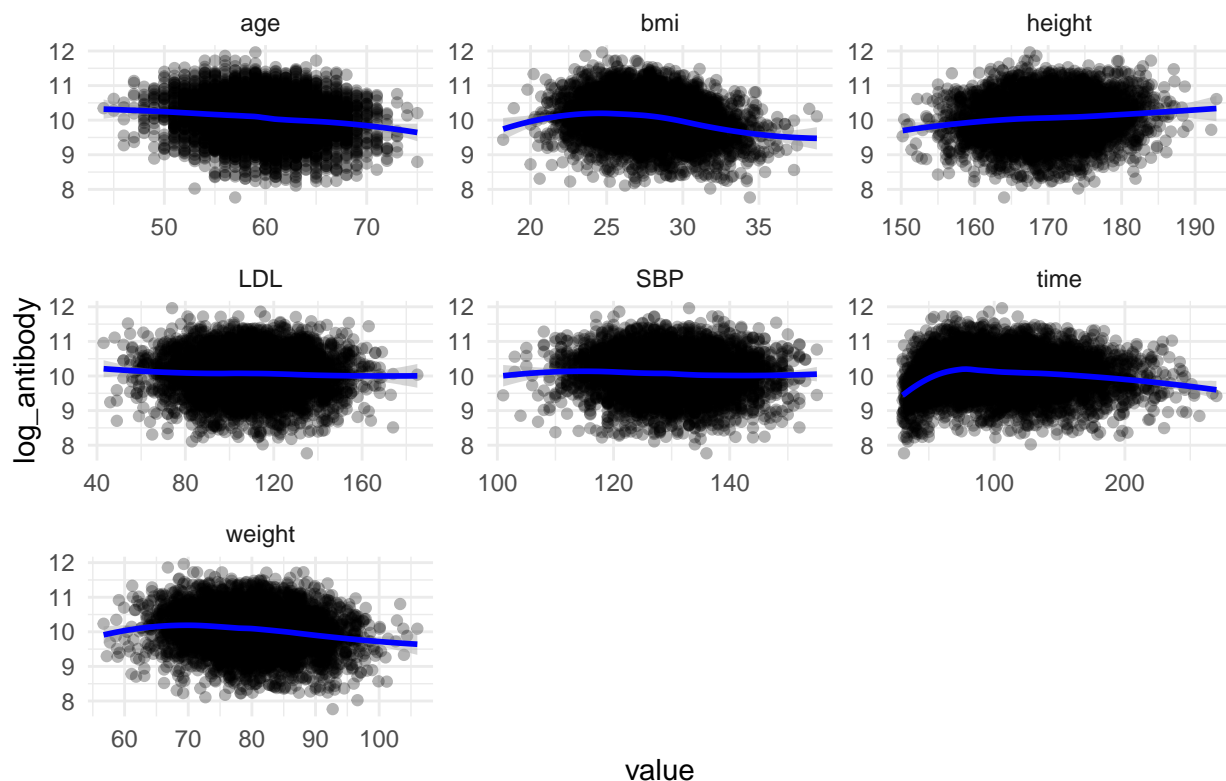
We first analyze the relationship between `log_antibody` (response variable) and continuous variables & correlations among continuous variables themselves.

```
continuous_var_long <- dat1 %>%
  select(age, height, weight, bmi, SBP, LDL, time, log_antibody) %>%
  tidyr::pivot_longer(cols = -log_antibody, names_to = "variable", values_to = "value")

# Scatterplots with smoothing lines
ggplot(continuous_var_long, aes(x = value, y = log_antibody)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "loess", color = "blue") +
  facet_wrap(~variable, scales = "free", ncol = 3) +
  theme_minimal() +
  labs(title = "Continuous Predictors vs. log_antibody")
```

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

Continuous Predictors vs. log_antibody



Using LOESS method, we observe linearity between predictors and the response. The plot shows that `bmi`, `time`, and `weight` has clear non linear trend against response `log_antibody`, indicating potential need to use GAM or non linear model.

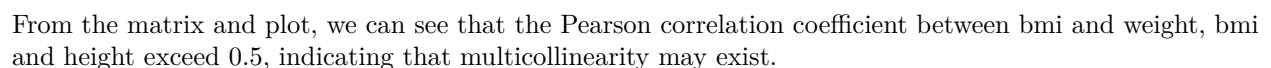
```
continuous_name <- c("age", "height", "weight", "bmi", "SBP", "LDL", "time", "log_antibody")
dat_cont <- dat1[, continuous_name]

# coefficient matrix
cor_matrix <- cor(dat_cont, use = "complete.obs", method = "pearson")
print(round(cor_matrix, 2))
```

```
##           age height weight  bmi  SBP  LDL  time log_antibody
## age       1.00  -0.01  0.00  0.00  0.44  0.21 -0.03      -0.15
## height   -0.01   1.00  0.23 -0.50  0.00  0.02  0.01       0.10
```

```
cor_melt <- melt(cor_matrix)
```

Correlation Heatmap of Continuous Variables



8


```
vif(lm_full)
```

```
##              GVIF Df GVIF^(1/(2*Df))
## age          1.258104 1      1.121652
## gender       1.002988 1      1.001493
## smoking      1.002682 2      1.000670
## height      107.111548 1     10.349471
## weight      169.112707 1     13.004334
## bmi         213.764468 1     14.620686
## diabetes     1.001898 1      1.000949
## hypertension 2.791341 1      1.670731
## SBP          3.070211 1      1.752202
## LDL          1.085268 1      1.041762
## time         1.002242 1      1.001120
```

The VIF of bmi, weight and height exceed 10, indicating serious multicollinearity among these variables. Since BMI is a function of weight and height, it is recommended to retain only one of them (e.g., BMI) in the model to avoid redundancy and unstable coefficient estimates.

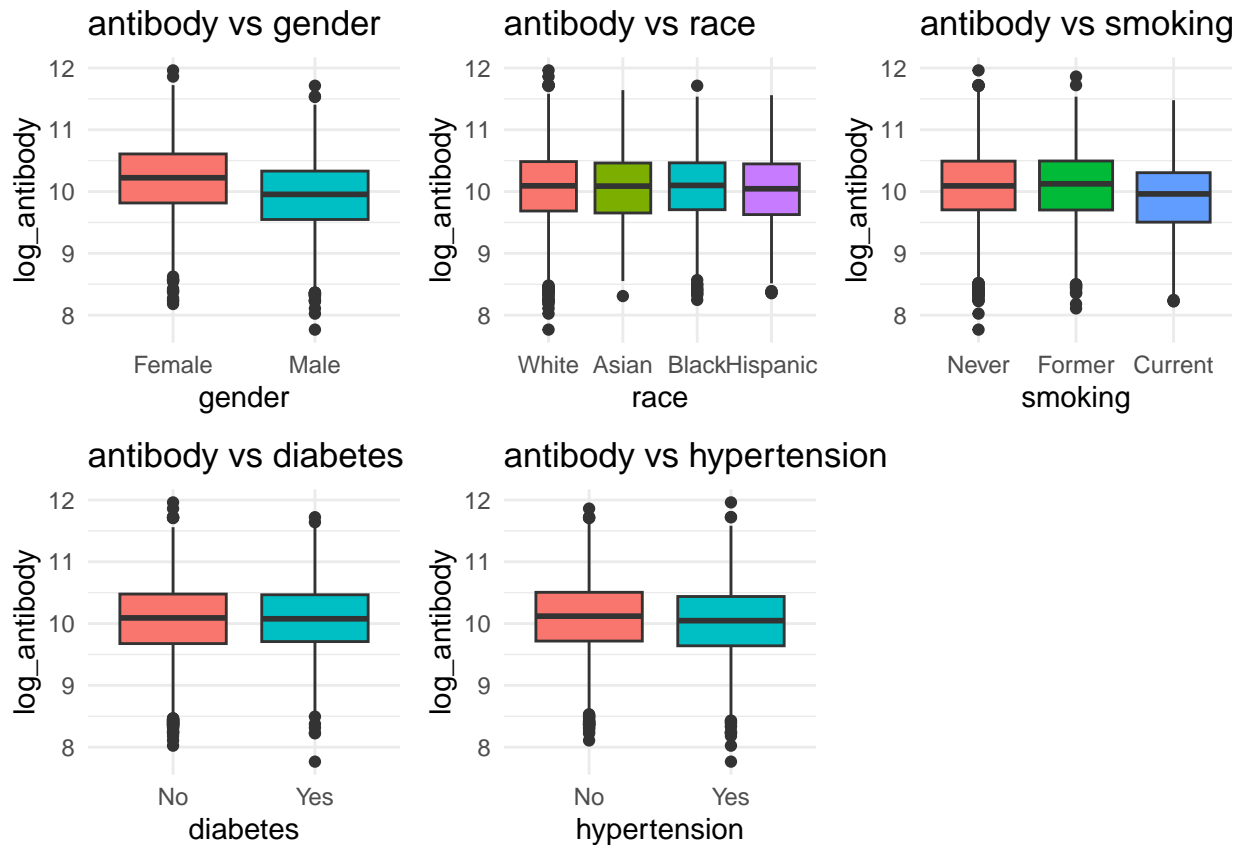
Then for categorical variable, generate boxplots to visualize the distribution of log_antibody across levels of each categorical variable.

```
categorical_name <- c("gender", "race", "smoking", "diabetes", "hypertension")

plot_list <- list()

for (name in categorical_name) {
  p <- ggplot(dat1, aes_string(x = name, y = "log_antibody", fill = name)) +
    geom_boxplot() +
    ggtitle(paste("antibody vs", name)) +
    theme_minimal() +
    theme(legend.position = "none")
  plot_list[[name]] <- p
}

do.call("grid.arrange", c(plot_list, nrow = 2, ncol = 3))
```



The boxplots suggest that the distribution of log_antibody does not differ substantially across the categories of each categorical variable, indicating limited evidence of strong group-level effects.

Model Training

We first fit a Lasso regression model to select important predictors and address multicollinearity. This is particularly useful here, as previous VIF analysis indicated strong multicollinearity among BMI, height, and weight.

```
x <- model.matrix(log_antibody ~ ., data = dat1)[, -1]
y <- dat1$log_antibody

set.seed(2)
cv.lasso <- cv.glmnet(x, y,
                      alpha = 1,
                      lambda = exp(seq(-5, 6, length = 100)))

cv.lasso$lambda.min

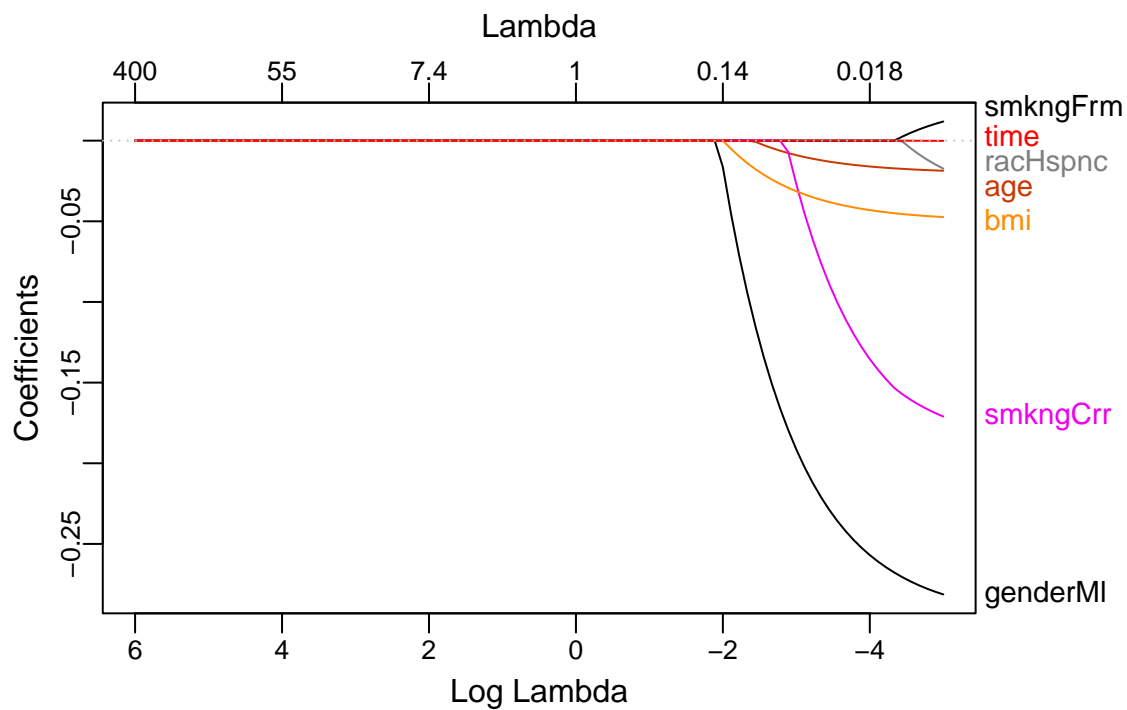
## [1] 0.006737947

lasso_coef <- coef(cv.lasso, s = "lambda.min")
lasso_coef

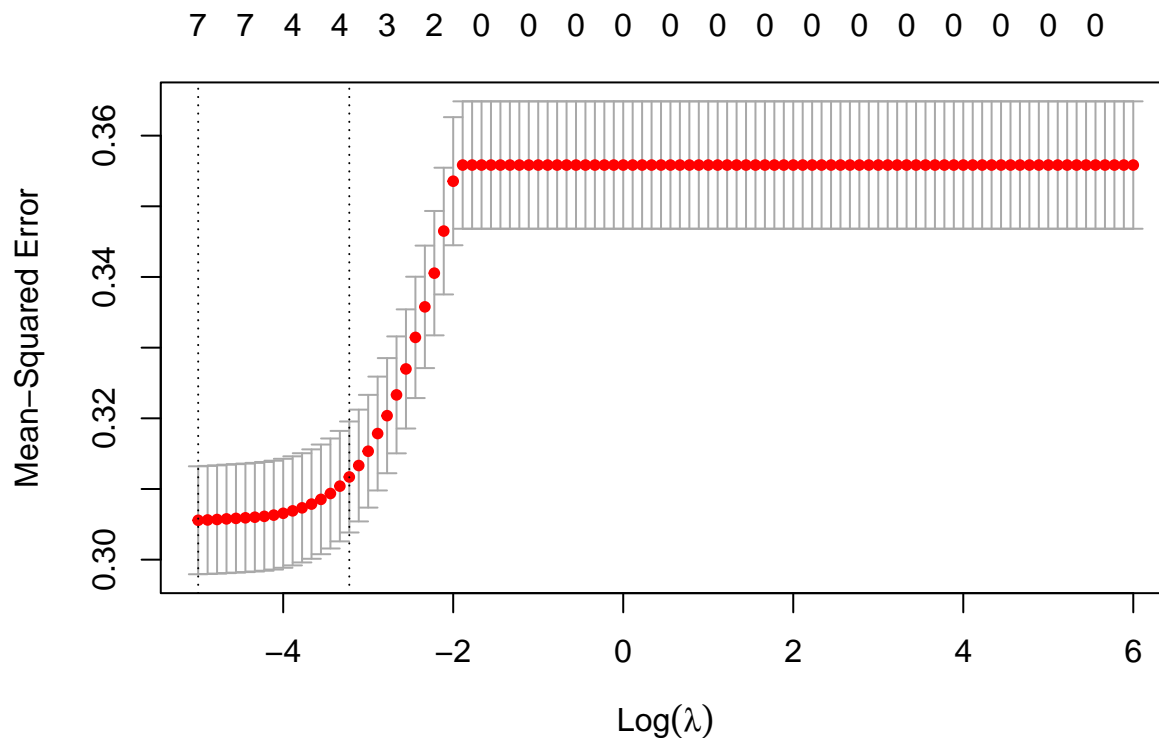
## 16 x 1 sparse Matrix of class "dgCMatrix"
##              s1
## (Intercept) 12.662438353
## age        -0.018645486
```

```
## genderMale      -0.281300893
## raceAsian       .
## raceBlack       .
## raceHispanic    -0.017347987
## smokingFormer   0.011848697
## smokingCurrent  -0.171036138
## height          .
## weight          .
## bmi             -0.047381778
## diabetesYes     .
## hypertensionYes .
## SBP             .
## LDL             .
## time            -0.000131821
```

```
plot_glmnet(cv.lasso$glmnet.fit)
```



```
plot(cv.lasso)
```



From the coefficients under `lambda.min`, we can see that height and weight are excluded from the Lasso model and bmi is retained. The lasso model helps solve multicollinearity.

```
y_pred_train <- predict(cv.lasso, newx = x, s = "lambda.min")

rmse_train <- sqrt(mean((y_pred_train - y)^2))

print(paste("RMSE on training set (lambda.min):", round(rmse_train, 4)))

## [1] "RMSE on training set (lambda.min): 0.5518"
```

Then we use the test dataset (`dat2`) to compute the test RMSE of the lasso model and evaluate model generalizability.

```
x_test <- model.matrix(log_antibody ~ ., data = dat2)[, -1]
y_test <- dat2$log_antibody

y_pred_lasso <- predict(cv.lasso, newx = x_test, s = "lambda.min")

rmse_lasso <- sqrt(mean((y_pred_lasso - y_test)^2))

print(paste("RMSE on test set (lambda.min):", round(rmse_lasso, 4)))

## [1] "RMSE on test set (lambda.min): 0.5749"
```

Since Lasso is a linear model that assumes additive and linear relationships between predictors and the outcome, we next explore two nonlinear modeling approaches (GAM and MARS) to capture potential nonlinearities and interaction effects in the data.

```
# Define trainControl
ctrl1 <- trainControl(method = "cv", number = 10)

train_y <- dat1$log_antibody
```

```

train_x <- dat1 %>%
  select(-log_antibody)

# GAM model
set.seed(2)
gam.fit <- train(train_x, train_y,
  method = "gam",
  trControl = ctrl1)

gam.fit$bestTune

## select method
## 2 TRUE GCV.Cp
gam.fit$finalModel

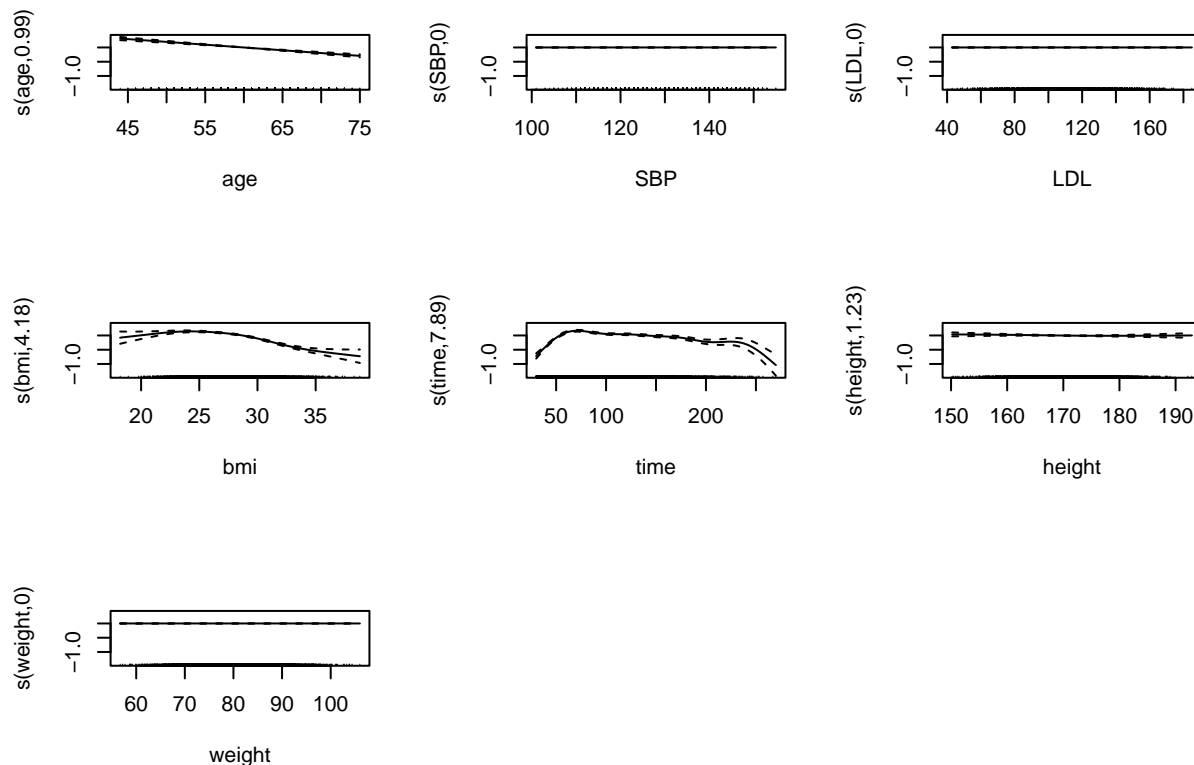
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender + diabetes + hypertension + smoking + race +
## s(age) + s(SBP) + s(LDL) + s(bmi) + s(time) + s(height) +
## s(weight)
##
## Estimated degrees of freedom:
## 0.991 0.000 0.000 4.179 7.892 1.234 0.000
## total = 23.3
##
## GCV score: 0.2786734

summary(gam.fit$finalModel)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ gender + diabetes + hypertension + smoking + race +
## s(age) + s(SBP) + s(LDL) + s(bmi) + s(time) + s(height) +
## s(weight)
##
## Parametric coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.228177 0.015328 667.269 < 2e-16 ***
## genderMale -0.297837 0.014933 -19.945 < 2e-16 ***
## diabetesYes 0.014230 0.020640 0.689 0.491
## hypertensionYes -0.007678 0.015995 -0.480 0.631
## smokingFormer 0.022219 0.016660 1.334 0.182
## smokingCurrent -0.193175 0.025834 -7.478 8.9e-14 ***
## raceAsian -0.003296 0.033009 -0.100 0.920
## raceBlack -0.010509 0.018837 -0.558 0.577
## raceHispanic -0.037424 0.026176 -1.430 0.153
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
##
## Approximate significance of smooth terms:
##          edf Ref.df      F p-value
## s(age)    9.908e-01    9 13.733 <2e-16 ***
## s(SBP)    6.175e-07    9  0.000  0.765
## s(LDL)    6.648e-07    9  0.000  0.639
## s(bmi)    4.179e+00    9 41.897 <2e-16 ***
## s(time)   7.892e+00    9 44.960 <2e-16 ***
## s(height) 1.234e+00    9  0.278  0.121
## s(weight) 2.262e-06    9  0.000  0.666
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.22   Deviance explained = 22.4%
## GCV = 0.27867   Scale est. = 0.27738    n = 5000
par(mfrow = c(3,3)) # Set plotting layout to 3x3 grid
plot(gam.fit$finalModel)
par(mfrow = c(1,1))
```



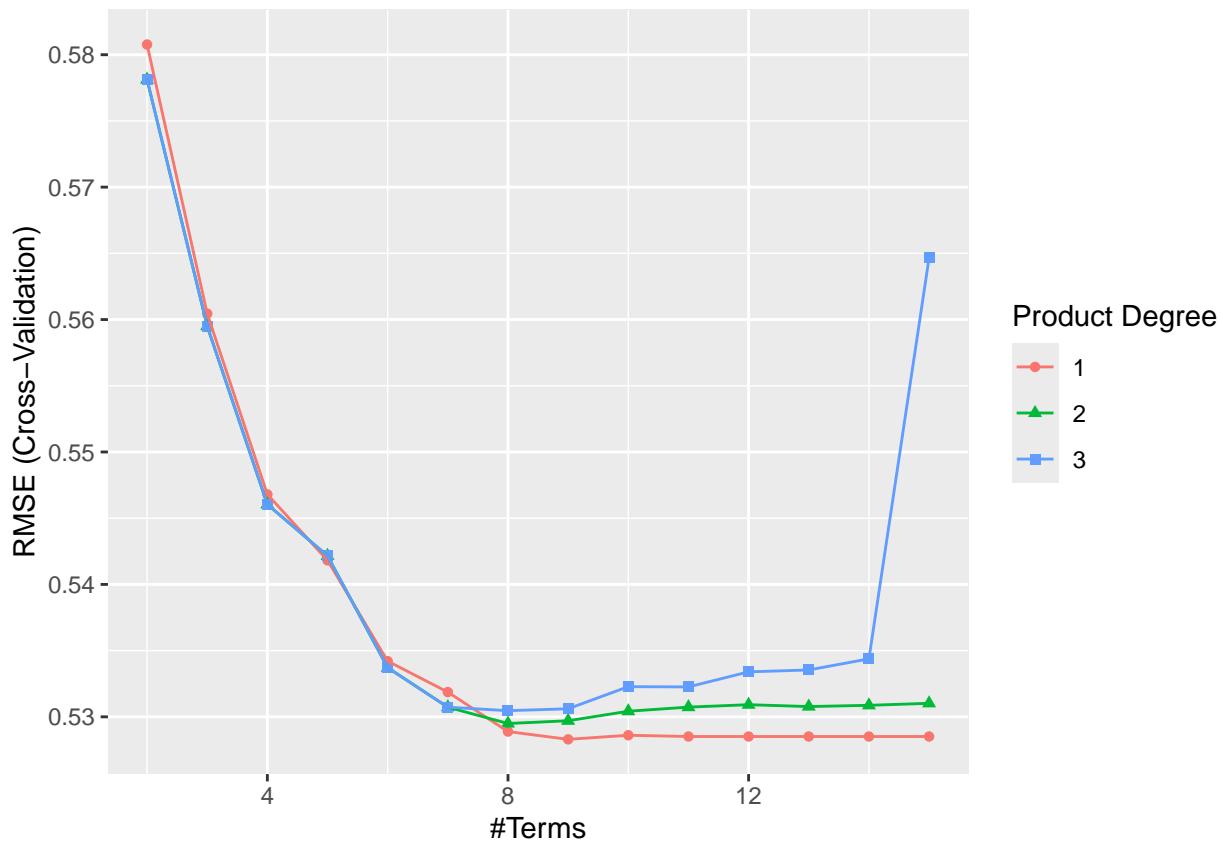
We fit a Generalized Additive Model (GAM) to examine both linear and nonlinear relationships between antibody levels and a set of demographic, clinical, and time-related predictors. The model achieved an adjusted R^2 of 0.22 and explained 22.4% of the deviance, indicating a moderate fit that captures key patterns in the data. Among the categorical predictors, male gender and current smoking were significantly associated with lower antibody responses, while race, diabetes, and hypertension did not show statistically significant effects. Several continuous variables exhibited nonlinear effects: notably, BMI and time since vaccination demonstrated strong and statistically significant nonlinear associations with antibody levels. Antibody responses peaked at a moderate BMI and declined at higher levels, while time since vaccination showed a characteristic waning immunity curve, with predicted antibody levels decreasing nonlinearly over time. These findings support the relevance of GAM for capturing complex decay behavior in vaccine-induced immune

responses against both demographic and clinical factors.

```
# MARS model
mars_grid <- expand.grid(degree = 1:3,
                        nprune = 2:15)

set.seed(2)
mars.fit <- train(train_x, train_y,
                  method = "earth",
                  tuneGrid = mars_grid,
                  trControl = ctrl1)

ggplot(mars.fit)
```



```
mars.fit$bestTune
```

```
##   nprune degree
## 8      9      1
```

```
summary(mars.fit)
```

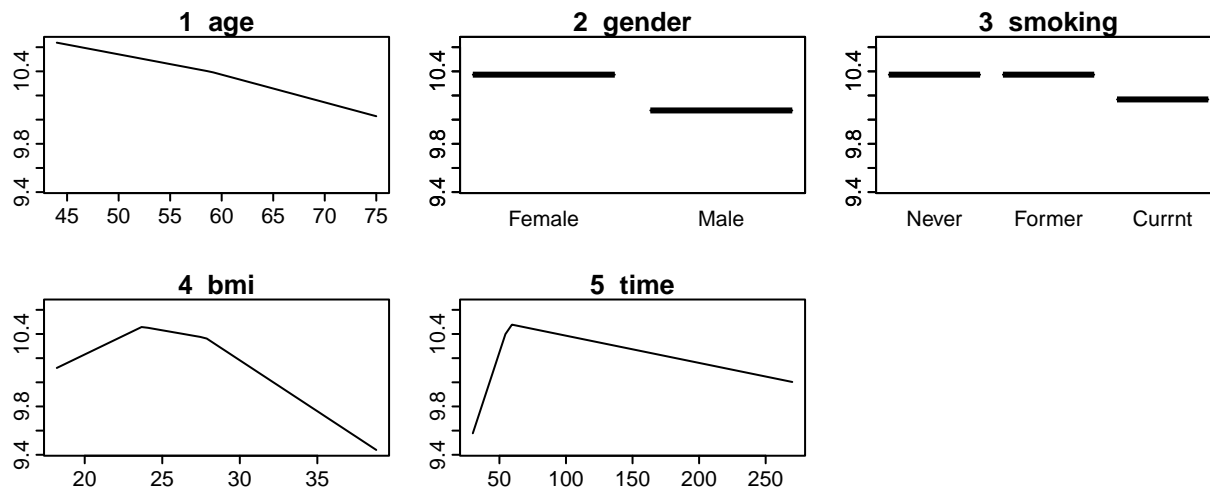
```
## Call: earth(x=data.frame[5000,12], y=c(10.65,9.889,1...), keepxy=TRUE,
##           degree=1, nprune=9)
##
##               coefficients
## (Intercept)    10.8474469
## genderMale     -0.2962905
## smokingCurrent -0.2051269
## h(59-age)       0.0161385
```

```
## h(age-59)          -0.0229576
## h(bmi-23.7)        -0.0843802
## h(27.8-bmi)        -0.0619974
## h(57-time)         -0.0335293
## h(time-57)         -0.0022542
##
## Selected 9 of 10 terms, and 5 of 15 predictors (nprune=9)
## Termination condition: RSq changed by less than 0.001 at 10 terms
## Importance: bmi, genderMale, time, age, smokingCurrent, raceAsian-unused, ...
## Number of terms at each degree of interaction: 1 8 (additive model)
## GCV 0.2787787    RSS 1384.431    GRSq 0.2166152    RSq 0.2216218
```

```
plotmo(mars.fit,
nresponse = 1,
degree2 = FALSE)
```

```
## plotmo grid:   age gender  race smoking height weight  bmi diabetes
##                60 Female White  Never  170.1   80.1  27.6      No
## hypertension SBP LDL time
##                No 130 110 106
```

```
type=raw train.default(x=train_x, y=train_y, method="earth", trControl=ctr...
```



We fit a Multivariate Adaptive Regression Splines (MARS) model to explore nonlinear and piecewise linear associations between antibody levels and a set of demographic, clinical, and time-related predictors. The best model selected through cross-validation used an additive structure (degree = 1, indicating no interaction effects among predictors) with 9 basis functions (nprune=9), explaining 22.2% of the deviance.

Among categorical variables, male gender and current smoking were associated with lower antibody levels, while other predictors like race, diabetes, and hypertension were excluded from the final model, indicating limited contribution to prediction.

Several continuous variables demonstrated nonlinear threshold effects via hinge functions. In particular, BMI exhibited a peaked relationship: antibody levels increased up to a moderate BMI (~23.7), then declined at higher BMI levels. Similarly, time since vaccination exhibited a nonlinear trend, with antibody levels increasing up to approximately 57 days post-vaccination and then declining thereafter.

The MARS model effectively identified and modeled piecewise linear patterns and threshold effects without explicitly specifying them, confirming its usefulness in capturing key nonlinear structures in immune response data.

To show the necessity of using nonlinear models (GAM and MARS), we also fit a simple linear model to

make comparison.

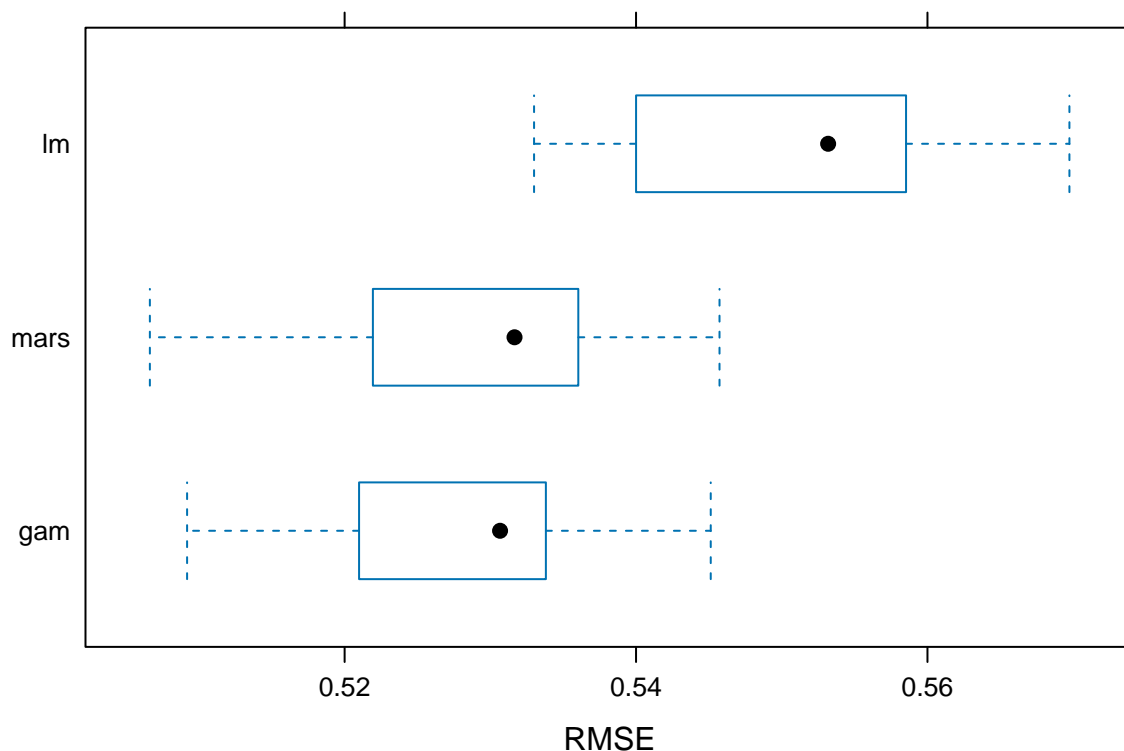
```
# Simple linear model
lm.fit <- train(train_x, train_y,
               method = "lm",
               trControl = ctrl1)
```

Model selection

To select the best model, we can compare the cross-validated metrics of the models using resampling, this helps evaluate and visualize the relative performance of the models.

```
resamp <- resamples(list(mars = mars.fit, gam = gam.fit, lm=lm.fit))
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: mars, gam, lm
## Number of resamples: 10
##
## MAE
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max. NA's
## mars 0.4120189 0.4180233 0.4203065 0.4224208 0.4285348 0.4360995    0
## gam  0.4127242 0.4190074 0.4202804 0.4224455 0.4273258 0.4352565    0
## lm   0.4216320 0.4324849 0.4427639 0.4393366 0.4470054 0.4500495    0
##
## RMSE
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max. NA's
## mars 0.5066327 0.5230870 0.5316602 0.5282995 0.5354905 0.5457286    0
## gam  0.5091877 0.5223781 0.5306669 0.5279212 0.5336806 0.5451253    0
## lm   0.5330013 0.5407617 0.5531813 0.5513093 0.5581304 0.5697423    0
##
## Rsquared
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max. NA's
## mars 0.1766328 0.1941155 0.2028183 0.2159220 0.2369173 0.2730827    0
## gam  0.1795042 0.1955026 0.2071224 0.2170568 0.2376473 0.2735385    0
## lm   0.1100305 0.1214327 0.1395787 0.1477937 0.1579194 0.2093478    0
bwplot(resamp, metric = "RMSE")
```



Apparently, the mean and standard error of RMSE of the simple linear model across cross-validation is quite bad, indicating that the simple linear is definitely not a good choice, so in the following we only need to discuss GAM and MARS model

The distribution of RMSE (of GAM and MARS model) across cross-validation is also similar, with GAM model showing a slightly lower median RMSE compared to MARS model.

Moreover, both models achieve lower average RMSE on the training set (dat1) compared to the Lasso model (Lasso model train RMSE: 0.5518, test RMSE: 0.5749), suggesting better performance. This may be attributed to the presence of nonlinear relationships between the response variable (log_antibody) and some predictors (such as bmi and time), which cannot be effectively captured by the linear structure of Lasso regression.

We should also evaluate the generalizability of the trained MARS and GAM models by computing RMSE on the test set (dat2).

```
# test RMSE of MARS model
mars.pred <- predict(mars.fit, newdata = dat2)
mars_test_rmse = sqrt(mean((mars.pred - dat2[, "log_antibody"])^2))
print(paste("RMSE on test set (MARS model):", round(mars_test_rmse, 4)))
```

```
## [1] "RMSE on test set (MARS model): 0.5328"
```

```
# test RMSE of GAM model
gam.pred <- predict(gam.fit, newdata = dat2)
gam_test_rmse = sqrt(mean((gam.pred - dat2[, "log_antibody"])^2))
print(paste("RMSE on test set (GAM model):", round(gam_test_rmse, 4)))
```

```
## [1] "RMSE on test set (GAM model): 0.5701"
```

```
# test RMSE of GAM model
lm.pred <- predict(lm.fit, newdata = dat2)
lm_test_rmse = sqrt(mean((lm.pred - dat2[, "log_antibody"])^2))
print(paste("RMSE on test set (simple linear model):", round(lm_test_rmse, 4)))
```

```
## [1] "RMSE on test set (simple linear model): 0.5683"
```

Both MARS and GAM models show very similar performance in cross-validation, with nearly identical mean RMSE values (0.5283 vs 0.5279).

However, on the test set, the MARS model achieves a lower RMSE (0.5328) compared to the GAM model (0.5701), suggesting better generalization.

Given this gap in test performance while maintaining comparable training performance, the MARS model appears to be the better choice in this case.

Results and Discussion

Influence of Demographic and Clinical Factors on Antibody Responses

```
dat2$pred_mars <- predict(mars.fit, newdata = dat2)

dat2 %>%
  group_by(gender) %>%
  summarise(mean_pred = mean(pred_mars), sd_pred = sd(pred_mars))
```

```
## # A tibble: 2 x 3
##   gender mean_pred sd_pred
##   <fct>      <dbl>   <dbl>
## 1 Female    10.1    0.215
## 2 Male      9.83    0.204
```

```
dat2 %>%
  group_by(smoking) %>%
  summarise(mean_pred = mean(pred_mars), sd_pred = sd(pred_mars))
```

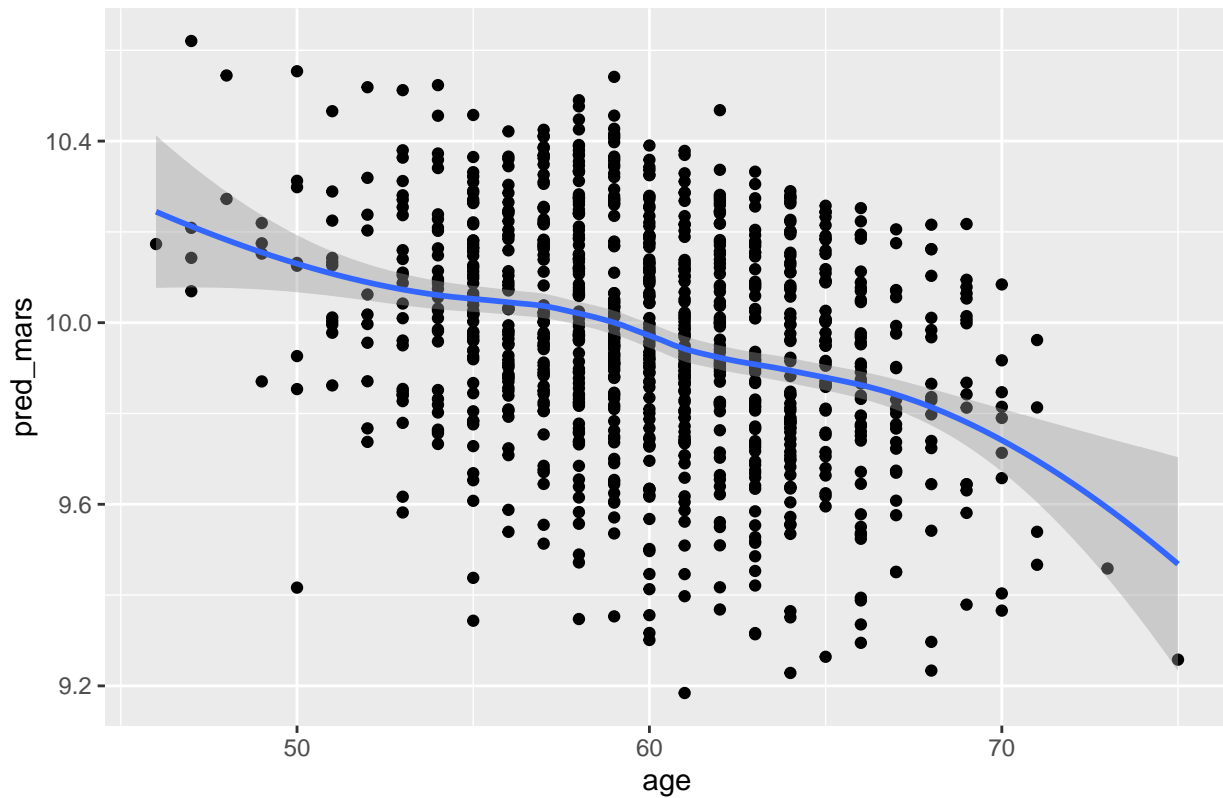
```
## # A tibble: 3 x 3
##   smoking mean_pred sd_pred
##   <fct>      <dbl>   <dbl>
## 1 Never      9.99    0.238
## 2 Former     9.98    0.260
## 3 Current    9.80    0.222
```

For categorical predictors, the predicted antibody levels from the MARS model on dat2 are consistent with the trends observed during model training on dat1. Specifically, females show higher predicted antibody levels than males (10.10 vs. 9.83), reflecting potential gender-related differences in immune response. Similarly, current smokers exhibit the lowest predicted antibody levels (mean = 9.80) compared to former and never smokers (means = 9.98 and 9.99, respectively), aligning with immunological evidence that smoking impairs vaccine response. These results reinforce the reliability of the model and suggest meaningful differences in antibody responses across demographic and behavioral subgroups.

```
# 1. PDP-like plots in dat2 using predicted values
ggplot(dat2, aes(x = age, y = pred_mars)) +
  geom_point() +
  geom_smooth(method = "loess") +
  labs(title = "Predicted Antibody Level vs Age (dat2)")
```

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

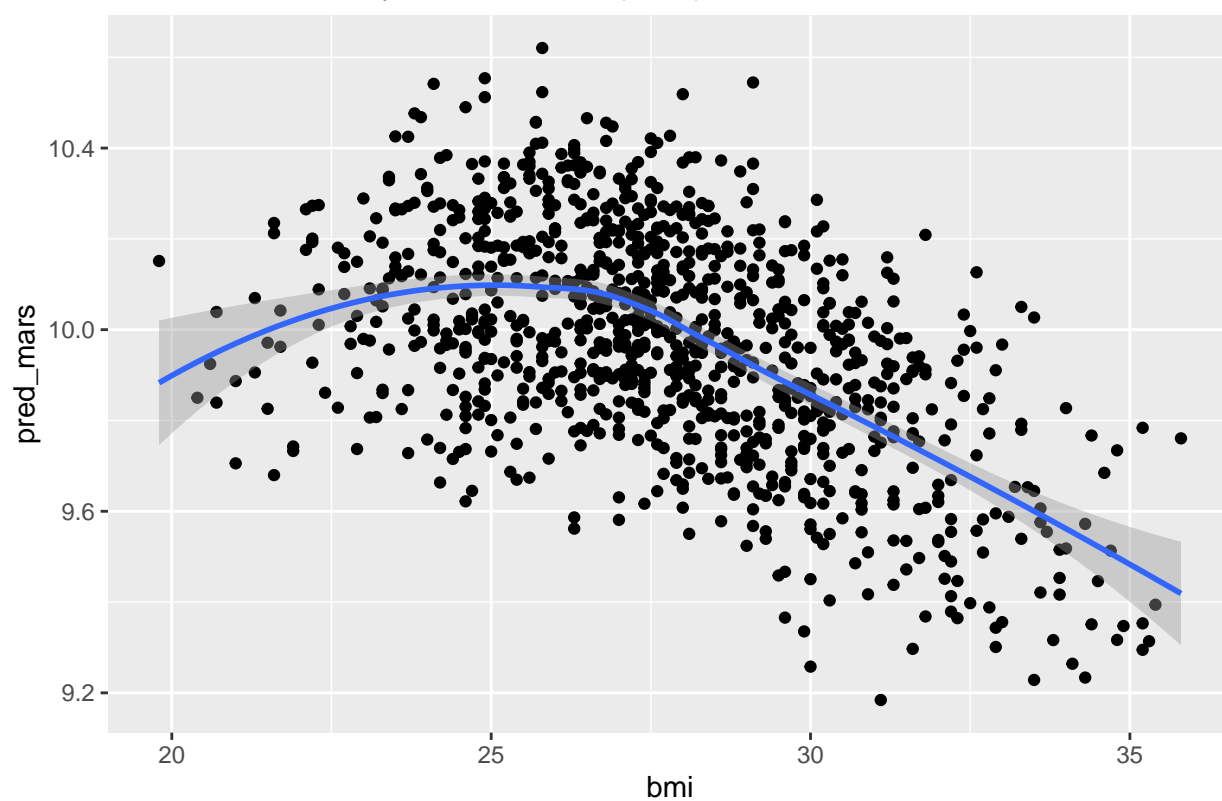
Predicted Antibody Level vs Age (dat2)



```
ggplot(dat2, aes(x = bmi, y = pred_mars)) +  
  geom_point() +  
  geom_smooth(method = "loess") +  
  labs(title = "Predicted Antibody Level vs BMI (dat2)")
```

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

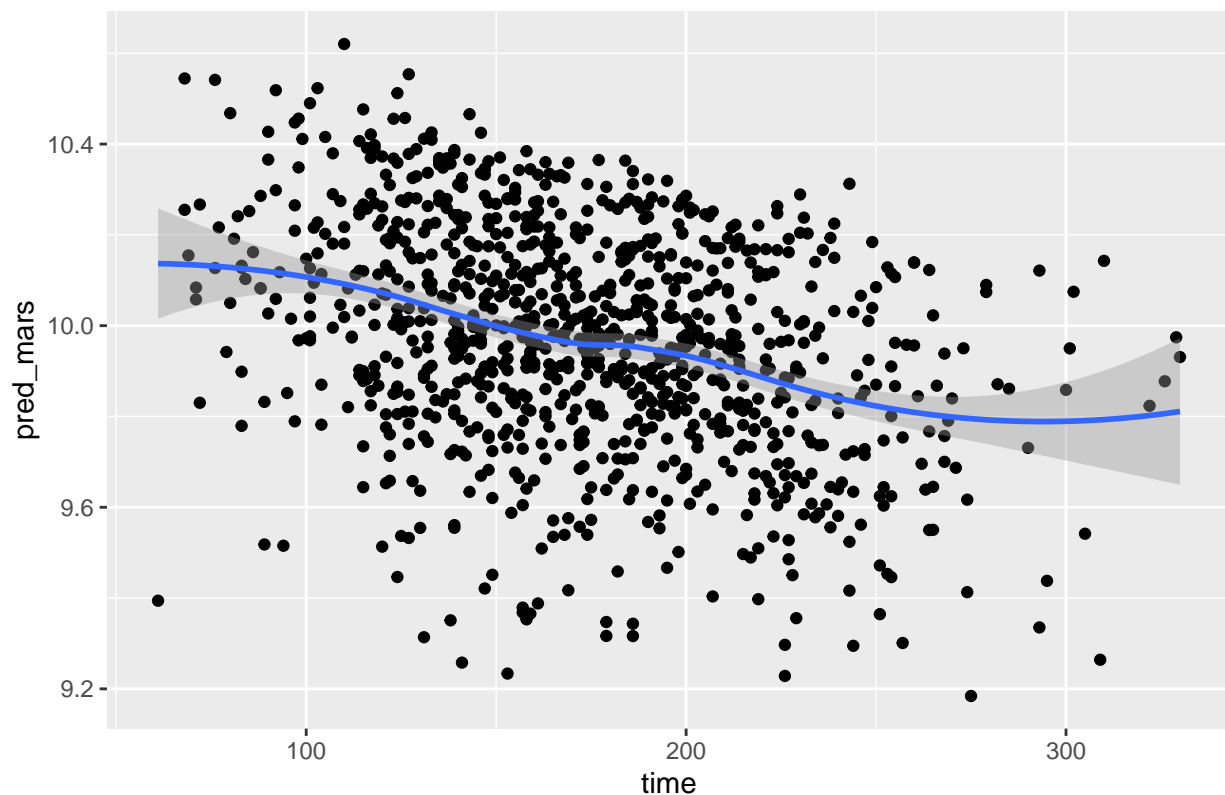
Predicted Antibody Level vs BMI (dat2)



```
ggplot(dat2, aes(x = time, y = pred_mars)) +  
  geom_point() +  
  geom_smooth(method = "loess") +  
  labs(title = "Predicted Antibody Level vs Time (dat2)")
```

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

Predicted Antibody Level vs Time (dat2)



```
summary(dat1$time)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      30.0   76.0   106.0   108.9   138.0   270.0
```

```
summary(dat2$time)
```

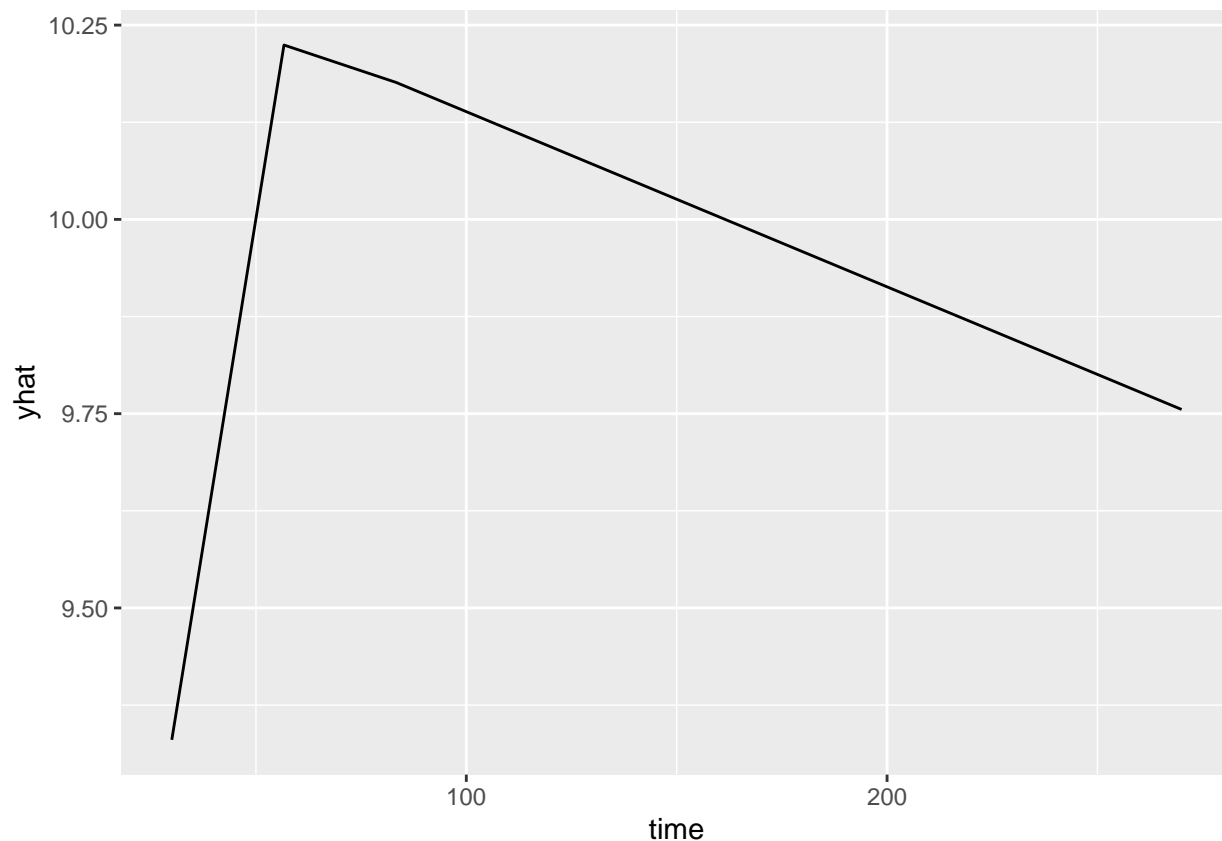
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      61.0  140.0  171.0  173.8  205.0   330.0
```

For continuous variables, we observed consistent trends in the marginal effects of BMI and age on predicted antibody levels between dat1 and dat2, indicating stable model behavior across datasets. However, the relationship between time since vaccination and predicted antibody levels differs noticeably: while dat1 shows an initial sharp rise followed by a gradual decline (consistent with typical post-vaccination antibody dynamics), the prediction on dat2 displays a more uniformly decreasing trend. Upon examining the distribution of the time variable, we found that this shift in behavior is due to the absence of early time points in dat2 — specifically, no observations exist for time < 61. As a result, the model cannot capture the early rise phase in dat2, and instead predicts a smoother, gradual decline.

How Antibody Levels Change Over Time

Since the both best GAM and best MARS model select age, gender, smoking, bmi, time as predictor, we will use them to analyze time decay behavior of antibody level.

```
p_time <- pdp::partial(mars.fit, pred.var = c("time"), grid.resolution = 10) |> autoplot()
p_time
```



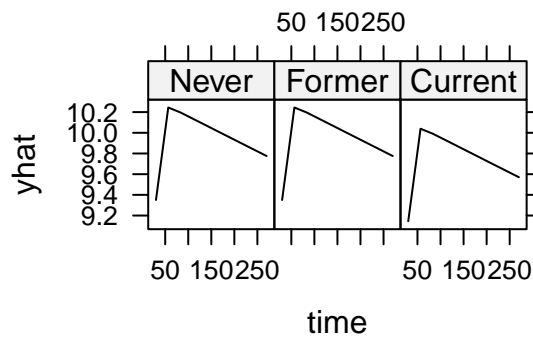
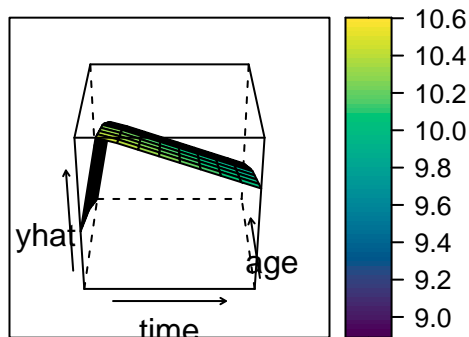
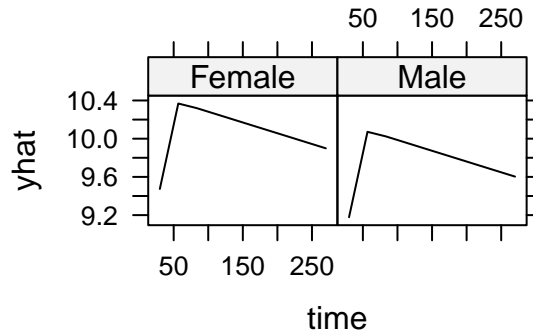
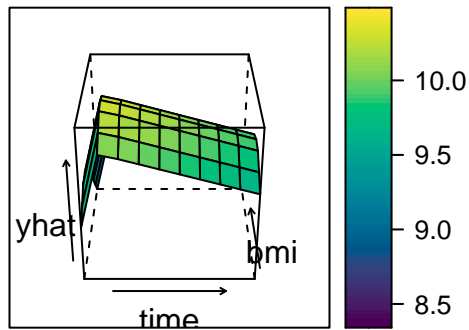
```
p_time_bmi <- pdp::partial(mars.fit, pred.var = c("time", "bmi"),
  grid.resolution = 10) |>
  pdp::plotPartial(levelplot = FALSE, zlab = "yhat", drape = TRUE,
    screen = list(z = 0, x = -60))

p_time_gender <- pdp::partial(mars.fit, pred.var = c("time", "gender"),
  grid.resolution = 10) |>
  pdp::plotPartial(levelplot = FALSE, zlab = "yhat", drape = TRUE,
    screen = list(z = 0, x = -60))

p_time_age <- pdp::partial(mars.fit, pred.var = c("time", "age"),
  grid.resolution = 10) |>
  pdp::plotPartial(levelplot = FALSE, zlab = "yhat", drape = TRUE,
    screen = list(z = 0, x = -60))

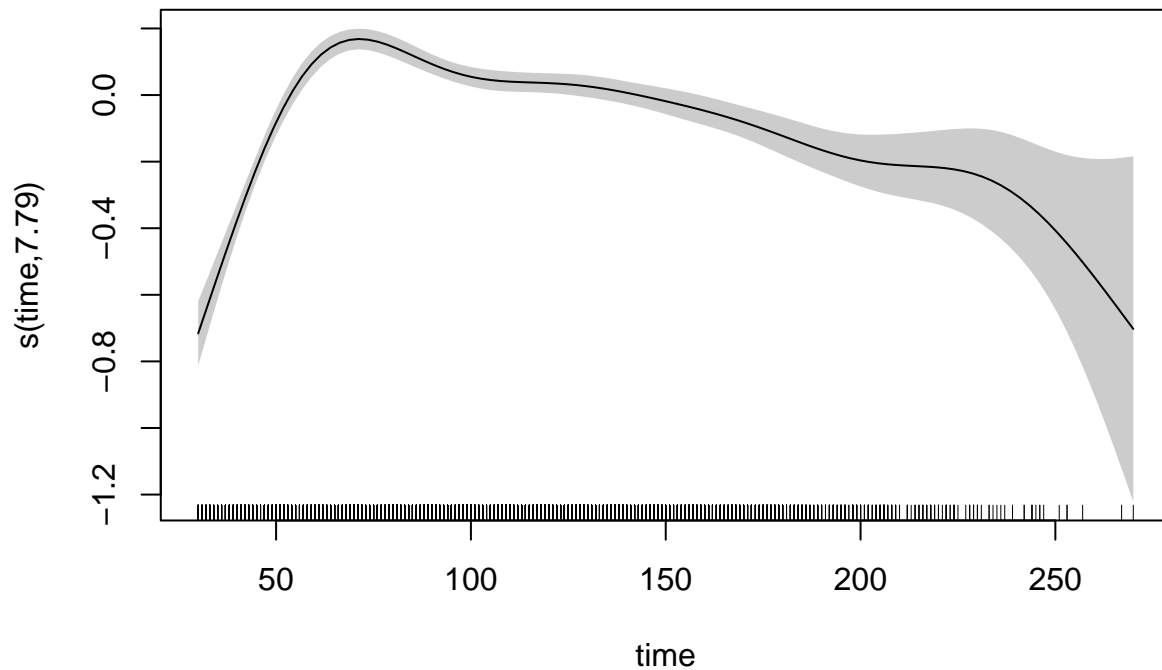
p_time_smoking <- pdp::partial(mars.fit, pred.var = c("time", "smoking"),
  grid.resolution = 10) |>
  pdp::plotPartial(levelplot = FALSE, zlab = "yhat", drape = TRUE,
    screen = list(z = 0, x = -60))

grid.arrange(p_time_bmi, p_time_gender, p_time_age, p_time_smoking,
  nrow = 2, ncol = 2)
```



With MARS model, we observe the antibody level have a sharp rise before time approximately less than 50 and then gradually decay, which aligns with the natural behavior regarding antibody level after the vaccination. Then we observe a consistent time-dependent pattern in the behavior of antibody levels, which also aligns with the overall trend of time decay behavior of antibody level. So we conclude that the time-dependent behavior of antibody levels is generally similar across these different and significant demographic and clinical factors.

```
dat1_ageGroup <- dat1 %>%
  mutate(age_group = ntile(age, 3)) %>%
  mutate(age_group = factor(age_group, labels = c("Young", "Middle", "Older"))) %>%
  mutate(
    gender = factor(gender, labels = c("Female", "Male")),
    diabetes = factor(diabetes, labels = c("No", "Yes")),
    hypertension = factor(hypertension, labels = c("No", "Yes"))
  )
gam_decay <- gam(log_antibody ~ s(time) + age + gender + diabetes + hypertension,
  data = dat1, method = "REML")
plot(gam_decay, select = 1, shade = TRUE)
```

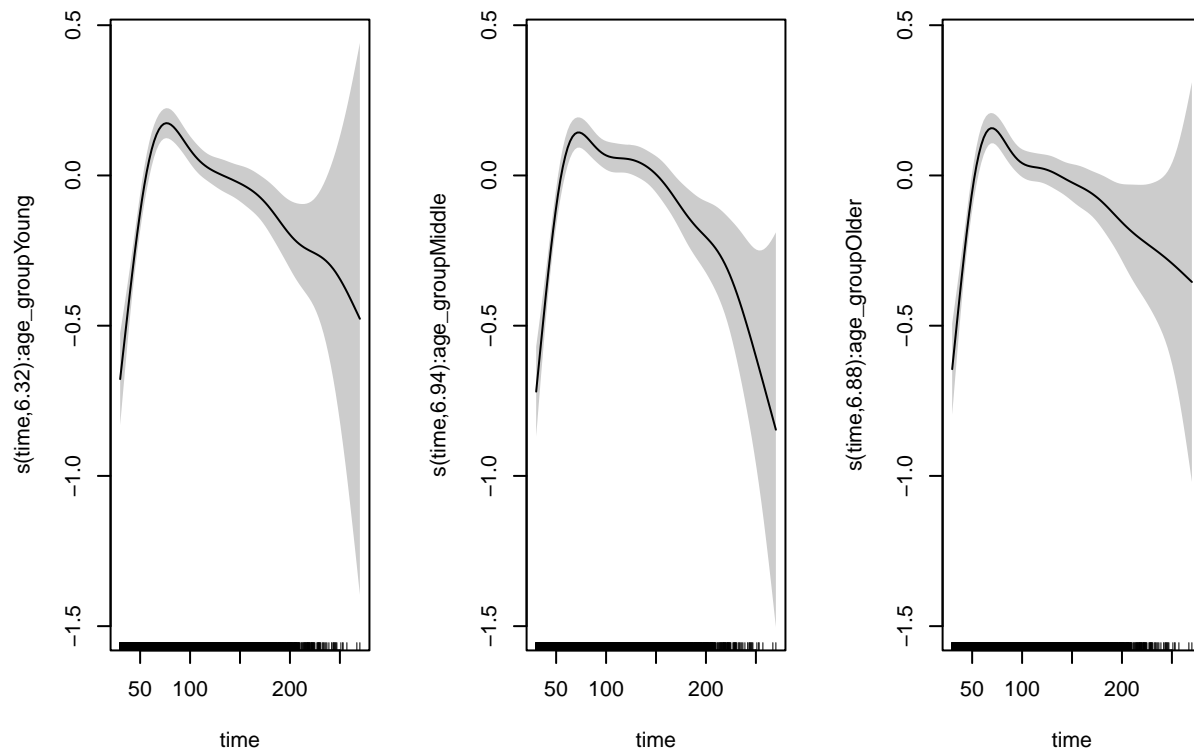
```
gam_age_interact <- gam(
  log_antibody ~ s(time, by = age_group) + age_group + gender + bmi + SBP + LDL +
    race + smoking + diabetes + hypertension,
  data = dat1_ageGroup,
  method = "GCV.Cp"
)

gam_gender_interact <- gam(
  log_antibody ~ s(time, by = gender) + gender + age + bmi + SBP + LDL +
    race + smoking + diabetes + hypertension,
  data = dat1_ageGroup,
  method = "GCV.Cp"
)

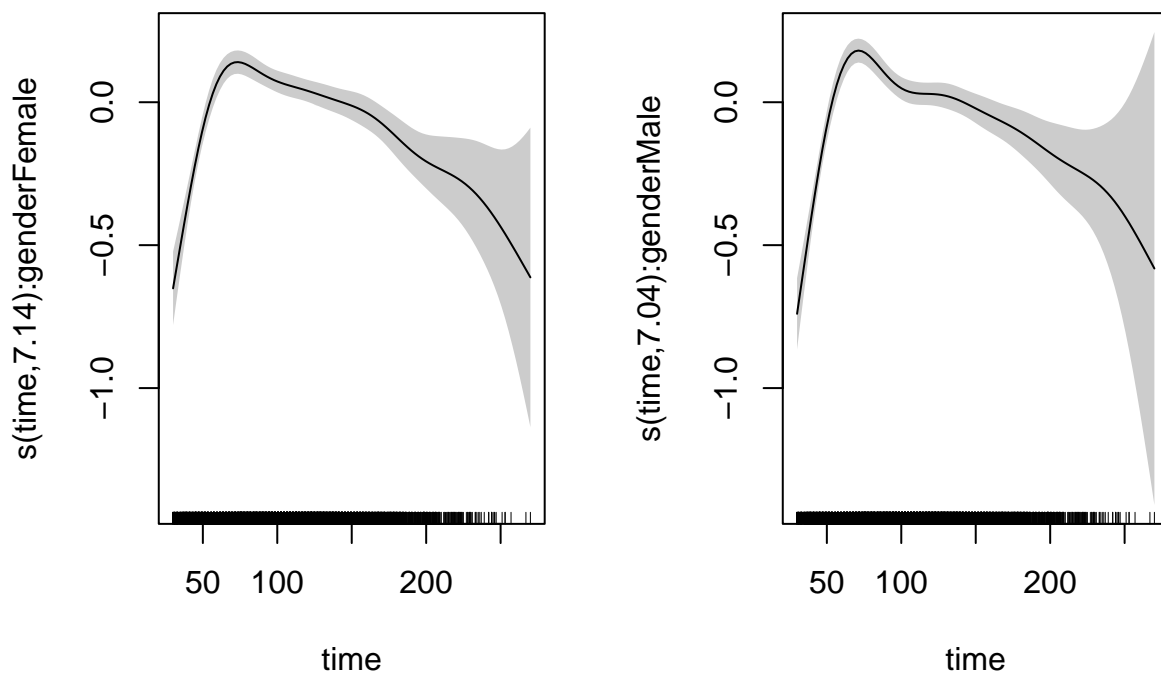
gam_diabetes_interact <- gam(
  log_antibody ~ s(time, by = diabetes) + diabetes + age + bmi + SBP + LDL +
    race + gender + smoking + hypertension,
  data = dat1_ageGroup,
  method = "GCV.Cp"
)

gam_hypertension_interact <- gam(
  log_antibody ~ s(time, by = hypertension) + diabetes + age + bmi + SBP + LDL +
    race + gender + smoking + hypertension,
  data = dat1_ageGroup,
  method = "GCV.Cp"
)

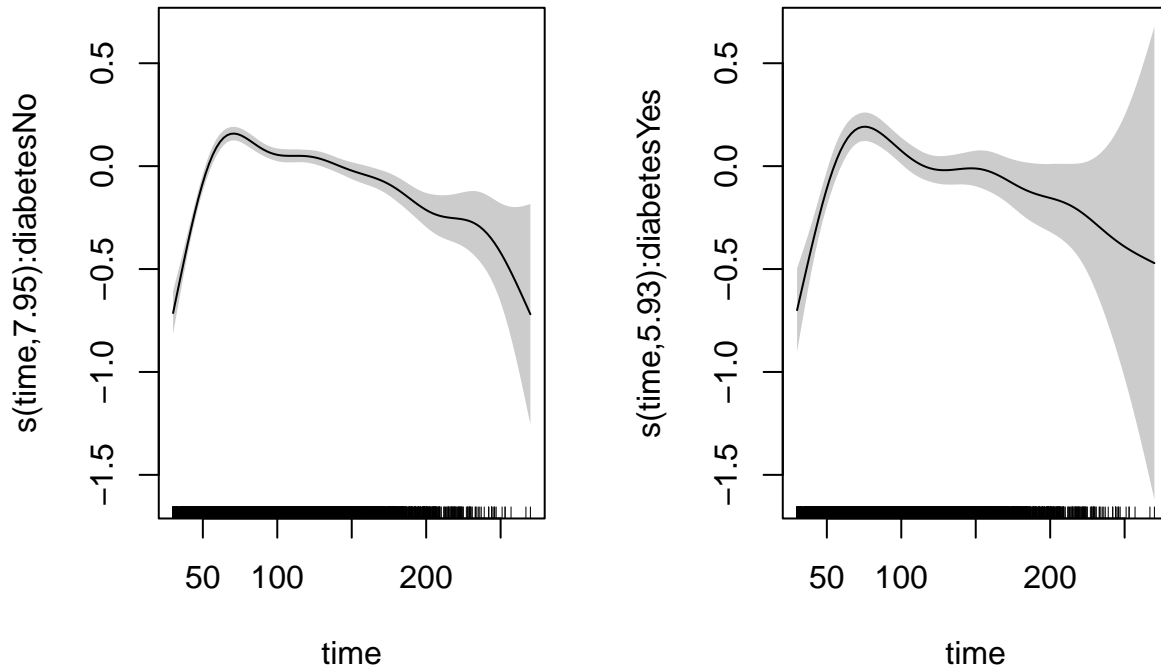
# Plotting smooth terms
par(mfrow = c(1, 3))
plot(gam_age_interact, shade = TRUE)
```



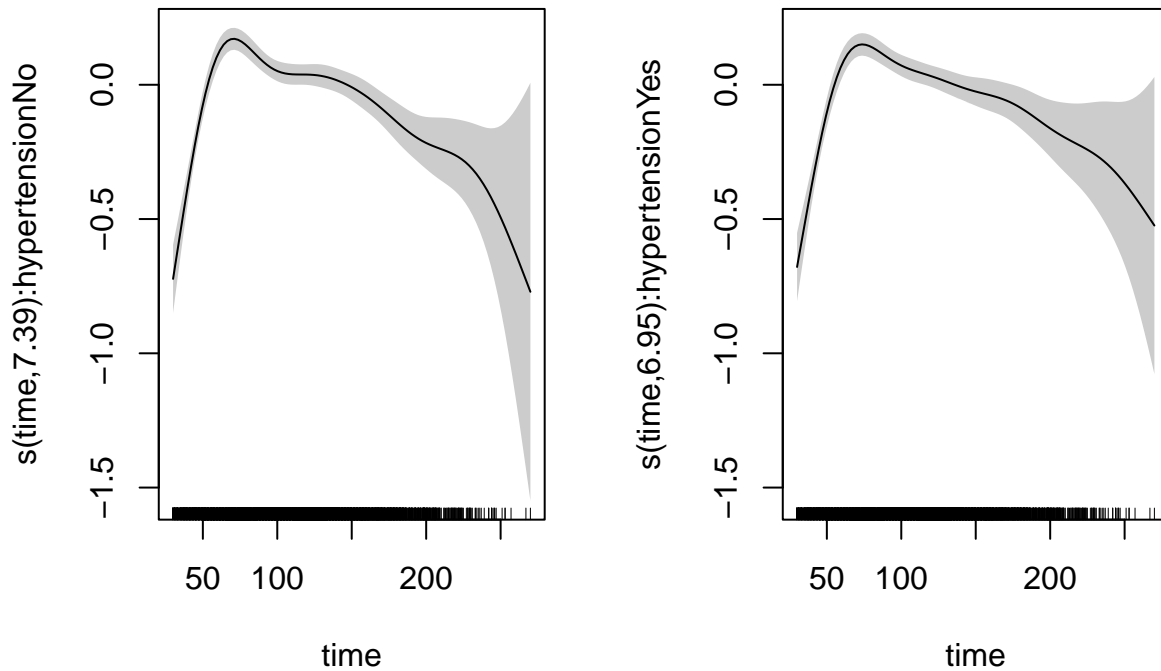
```
par(mfrow = c(1, 2))
plot(gam_gender_interact, shade = TRUE)
```



```
plot(gam_diabetes_interact, shade = TRUE)
```



```
plot(gam_hypertension_interact, shade = TRUE)
```



Our another aim is to understand how the effects of time on antibody levels might vary across different demographic and clinical factors. Therefore, we examined the time decay behavior of antibody levels across different subgroups defined by categorical factors(demographic and clinical).

We observe that while the general pattern of antibody decay is similar across all subgroups, certain groups (such as the middle-aged, no hypertension, no diabetes subgroups) exhibit a sharper decline in antibody levels at the tail end of the time period. These factors may be important factors in shaping the rate of antibody decay who deserve more in-depth investigation.