Data Science 2 Midterm

Jasmin Martinez (JRM2319), Ixtaccihuatl Obregon, Elliot Kim

03/24/2025

Exploratory Analysis

For datasets **dat1** and **dat2** initial exploration of the data structure, descriptive statistics of continuous variables, correlation analysis, and various visualization techniques were conducted.

Looking at **dat1**, the distribution of **log_antibody** is approximately normal, as seen in Figure @ref(fig:d1-log-antibody-hist) and Figure @ref(fig:d1-log-antibody-qq). High and low outliers can be observed in the Figure @ref(fig:d1-log-antibody-box). Most covariates have weak or low correlation with each other. There is a high positive correlation between **bmi** and **weight** ($\rho = 0.72$) and a moderate negative correlation between **bmi** and **height** ($\rho = -0.50$). There is a mild negative correlation between **log_antibody** and **bmi** ($\rho = -0.23$), **weight** ($\rho = -0.17$), and **age** ($\rho = -0.15$). There is a mild positive correlation between **log_antibody** and **height** ($\rho = 0.10$). The correlation between **log_antibody** and **SBP** ($\rho = -0.06$), **LDL** ($\rho = -0.04$), and **time** ($\rho = -0.01$) are near zero, indicating no linear relationship. Linear relationships can be seen in Figure @ref(fig:d1-log-antibody-lin).

Exploring **dat2**, the distribution of **log_antibody** is also approximately normal, as shown Figure @ref(fig:d2-log-antibody-hist) and Figure @ref(fig:d2-log-antibody-qq). High and low outliers are again visible in the Figure @ref(fig:d2-log-antibody-box). Most covariates exhibit weak or low correlation with each other. There is a high positive correlation between **bmi** and **weight** ($\rho = 0.72$) and a moderate negative correlation between **bmi** and **height** ($\rho = -0.53$). There is a mild negative correlation between **log_antibody** and **bmi** ($\rho = -0.16$), **weight** ($\rho = -0.11$), and **age** ($\rho = -0.08$). A mild positive correlation exists between **log_antibody** and **height** ($\rho = 0.084$). The correlations between **log_antibody** and **SBP** ($\rho = -0.01$), **LDL** ($\rho = -0.00$), and **time** ($\rho = -0.25$) are near zero or weak, again suggesting no strong linear relationship. Linear relationships can be seen in Figure @ref(fig:d2-log-antibody-lin).

The exploration and evaluation of **dat1** is used to help build a prediction model specific ally using GAM to understand how demographic and clinical factors influence antibody responses and how antibody levels change over time following vaccination. Considering the researcher collects a new and independent dataset, **dat2**, we discover the correlation between **time** and **log_antibody** is stronger than in **dat1** and similarly has weak linear relationships with most covariates. **Dat2** allows us to evaluate the robustness and generalizability of our prediction model.

Model Trainging

Multiple Linear Regression

A multiple linear regression model is simple and easy to use when working with linearly dependent data but works under the assumption of homoscedasticity and no multicollinearity. Fitting this model required ** method = lm ** in the ** train()** function, resulting in $R^2 = 0.1470498 $ and $R^2_{adj} = 0.03320516$ indicating a poor fit.

Ridge and Lasso Regression

Generalized Additive Model (GAM)

A Generalized Additive Model (GAM) was created to model log-transformed antibody level. Predictors included age, gender, race, smoking, height, weight, BMI, diabetes, hypertension, SBP, LDL, and time since vaccination. To prepare the data for modeling, the model matrix, x, was created using the outcome and all predictors listed above and extracted the response variable, y. Given that race and smoking were categorical variables with multiple categories, some re-coding was necessary to use these variables in the model building. Therefore, race and smoking were converted into factor variables with "White" being the reference group for race and "Never" being the reference group for smoking.

The GAM models were then fitted. The first model, gam.m1, is a standard linear model with no smoothing terms. The second model, gam.m2, uses smoothing on age, bmi, SBP, LDL, and time since vaccination. There was reason to believe these variables were non-linear, therefore the smoothing allowed the variables to be used in the model. Finally, model 3, gam.m3, includes a tensor product to model the interaction between height and weight.

The three GAM models were then compared using an ANOVA test that provides the f-test; this was used to determine which model provides the best fit. Cross-validation for GAM tuning was then conducted using a 10-fold cross-validation to tune to GAM model. The best hyperparameters and the final fitted model were then retrieved. The same model training procedure was used with the new, independent dataset, dat2.

Model 2 (gam.m2) was chosen to be the final model based on the improvements it made upon Model 1 (gam.m1), (pval=<0.001).

Multivariate Adaptive Regression Splines Model

The Multivariate Adaptive Regression Splines (MARS) model is a flexible non-linear model technique used to evaluate the relationship between an outcome and multiple covariates. MARS divides the data into segments and fits a piecewise linear regressions for each segment. MARS is used to model **log_antibody** levels based on clinical and demographic predictors. The model is tuned on two hyperparameters: **nprune** for the most number of terms and **degree** for the most amount of degree interactions between predictors.

Re-sampling and final model selection

Final Model Results

The prediction model's robustness and generalizability is mostly acceptable for **dat2**. Prediction accuracy was determined by the mean squared error (MSE) at 0.325 showing a low average on unseen data. The prediction model shows model stability with generalized cross-validation (GCV) score of 0.279. Looking at @ref(fig:dx-plots) we evaluate Predicted vs Actual log_antibody Levels, Residuals vs Predicted, and the Distribution of Residuals. Predictions are mostly aligned with the observed values especially in the 9.5-10.5 predicted range. There is mild under-prediction below value 9, which indicates that the model may tend to underestimate lower antibody levels but perform well in the 9.5-10.5 predicted range. The predicted vs residual plot shows residuals mostly centered and near 0. Right-skewness can be observed in the predicted vs residual plots, which could suggest some non-linearity. The distribution of residuals looks approximately normal and does not have extreme outliers or multi-modality.

Table 1: Clean Summary Statistics for Numeric Variables

Variable	Min	Q1	Median	Mean	Q3	Max
id	1.000000	1250.750000	2500.50000	2500.50000	3750.25000	5000.00000
age	44.000000	57.000000	60.00000	59.96840	63.00000	75.00000
gender	0.000000	0.000000	0.00000	0.48540	1.00000	1.00000
height	150.200000	166.100000	170.10000	170.12634	174.22500	192.90000
weight	56.700000	75.400000	80.10000	80.10908	84.90000	106.00000
bmi	18.200000	25.800000	27.60000	27.74040	29.50000	38.80000
diabetes	0.000000	0.000000	0.00000	0.15440	0.00000	1.00000
hypertension	0.000000	0.000000	0.00000	0.45960	1.00000	1.00000
SBP	101.000000	124.000000	130.00000	129.90040	135.00000	155.00000
LDL	43.000000	96.000000	110.00000	109.90860	124.00000	185.00000
time	30.000000	76.000000	106.00000	108.86260	138.00000	270.00000
log	7.765405	9.681635	10.08908	10.06434	10.47758	11.96137

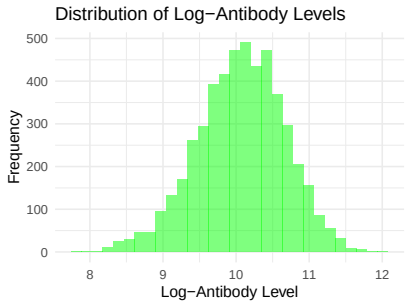


Figure 1: Histogram of log_antibody levels for dat1

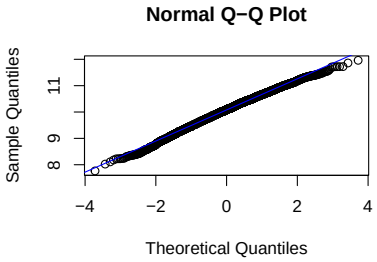


Figure 2: Q-Q plot of log_antibody levels for dat1

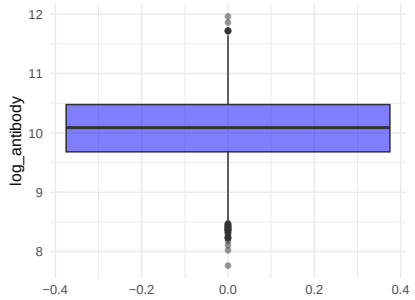


Figure 3: Boxplot of log_antibody levels for dat1

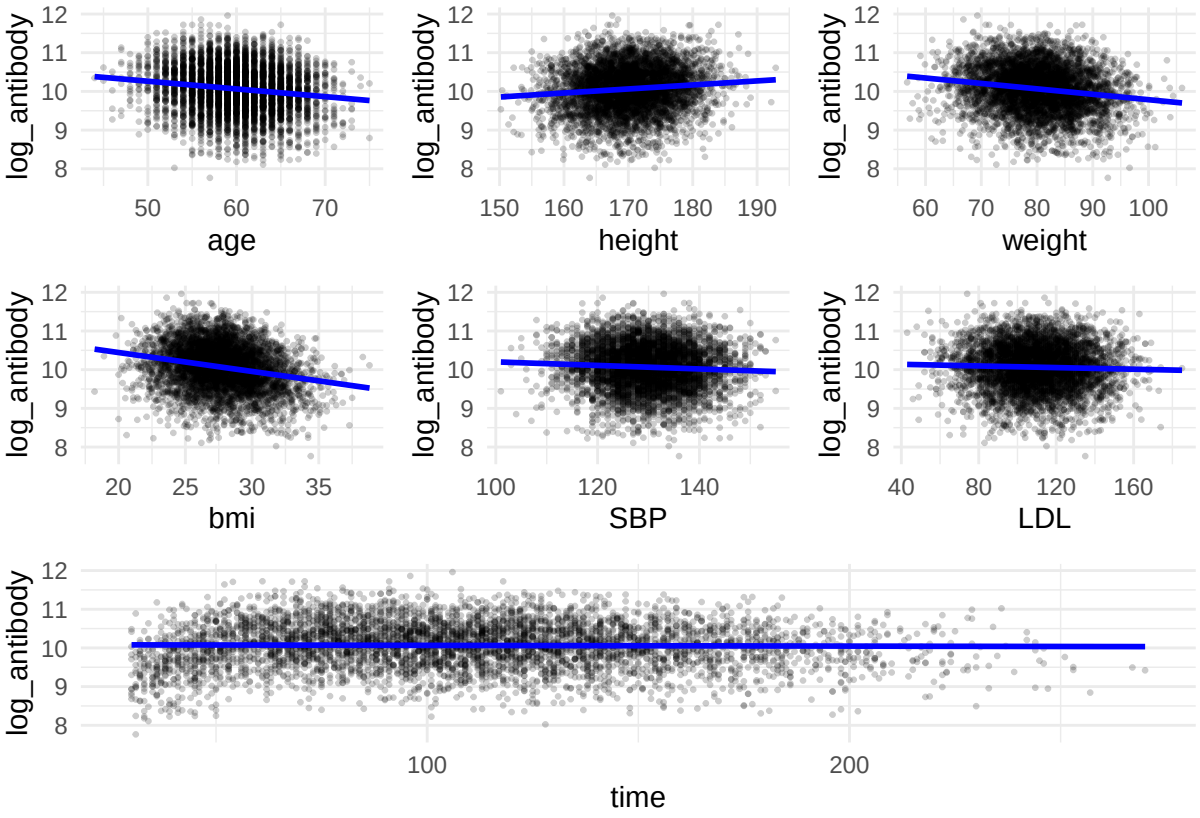


Figure 4: Linearity between log_antibody levels and covariates for dat1

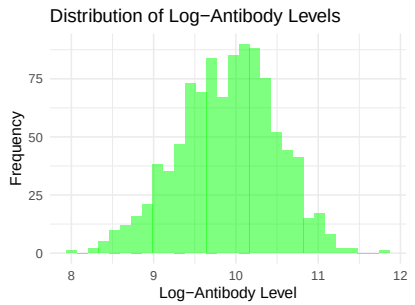


Figure 5: Histogram of log_antibody levels for dat2

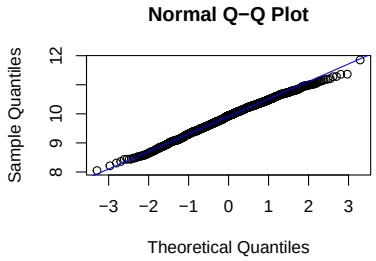


Figure 6: Q-Q plot of log_antibody levels for dat2

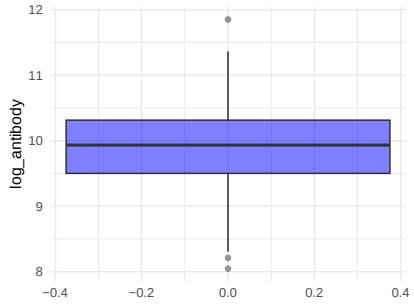


Figure 7: Boxplot of log_antibody levels for dat2

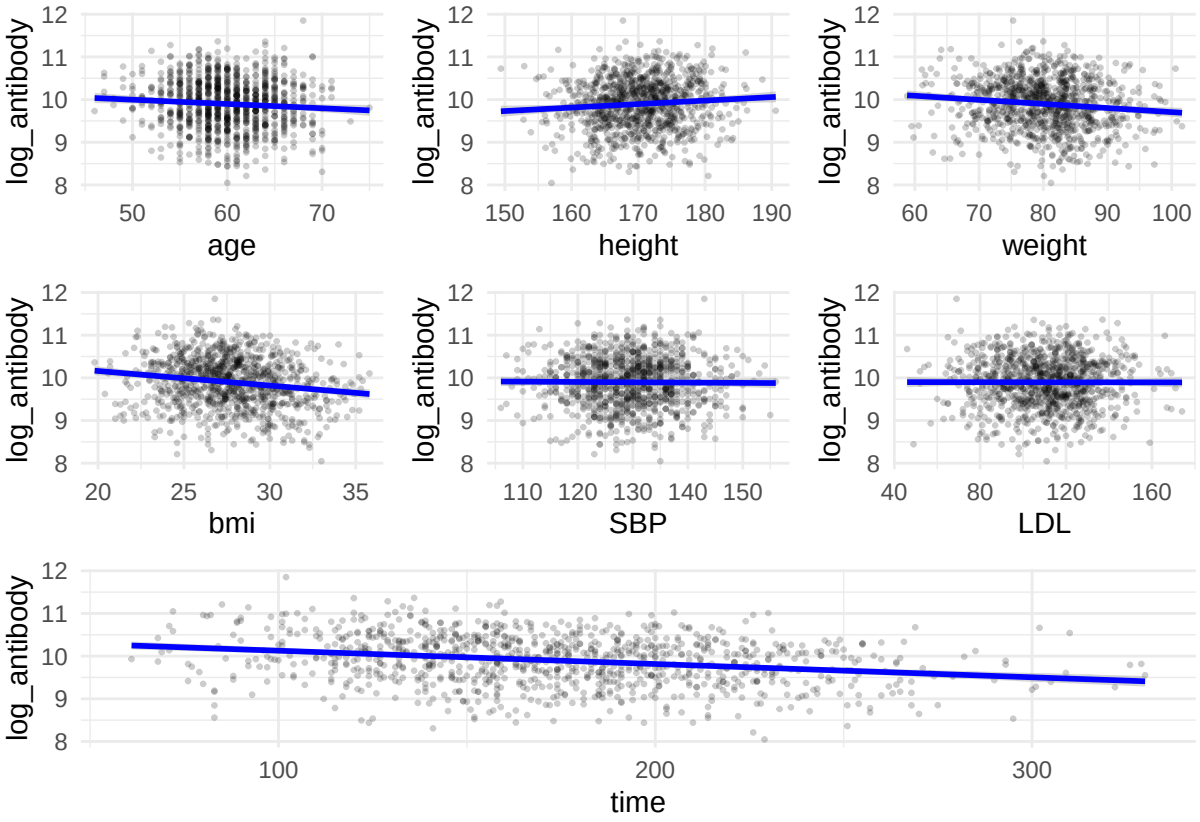
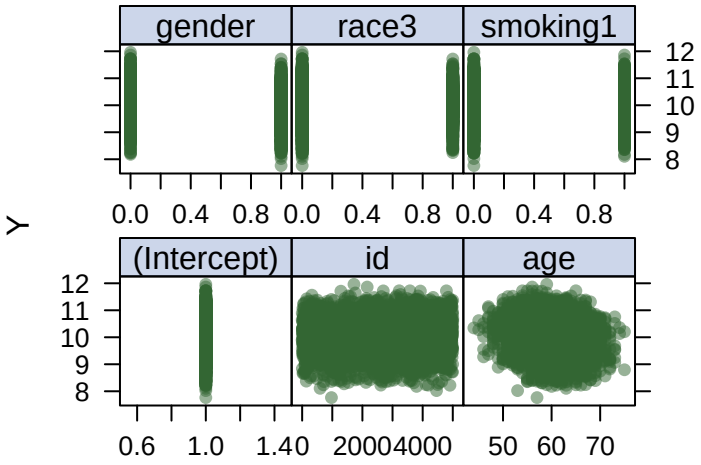
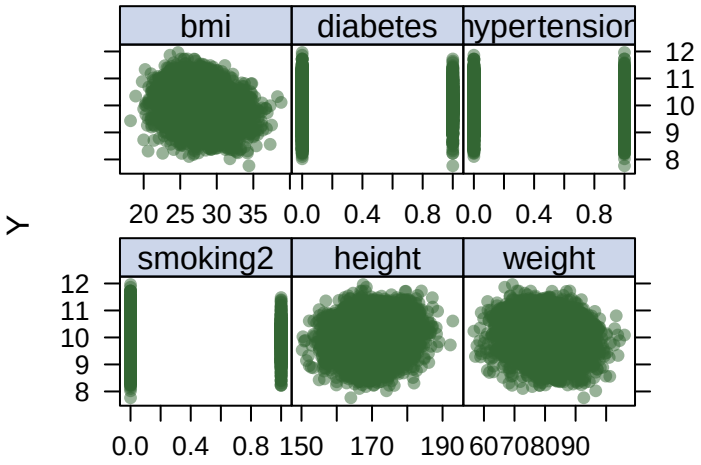
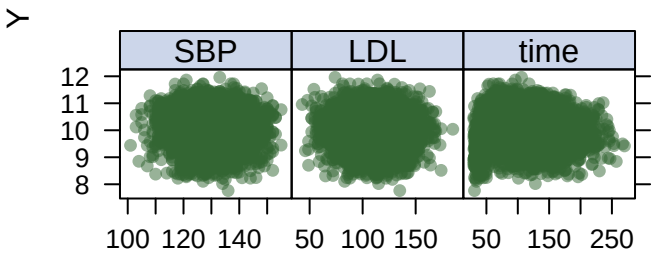


Figure 8: Linearity between log_antibody levels and covariates for dat2


gender
race3
smoking1
(Intercept)
id
age
Y


bmi
diabetes
hypertension
smoking2
height
weight
Y




```
### LM Modeling
set.seed(2)
datSplit <- initial_split(data = dat1, prop = 0.8)
trainData <- training(datSplit)
testData <- testing(datSplit)
head(trainData)
```

```
##     id age gender  race smoking height weight  bmi diabetes hypertension SBP
## 1 3925  58      1 White   Never  178.0   85.3 26.9        0            1 133
## 2 4806  52      1 White   Never  173.2   82.1 27.4        0            0 114
## 3 2822  69      1 White  Former  175.7   82.2 26.6        0            1 135
## 4 4512  56      1 White Current  175.9   78.4 25.4        0            0 125
## 5 4488  65      0 White Current  173.9   77.3 25.5        0            0 126
## 6  273  63      1 White   Never  166.9   84.1 30.2        1            1 141
##   LDL time log_antibody
## 1 119  103     9.448527
## 2  81   45     9.805080
## 3 125  106    10.686421
## 4 121   77     9.939505
## 5  95   71    10.376005
## 6 143   64    10.057639
```

```
fit.lm <- train(log_antibody ~ age + gender + race + smoking + height + weight + bmi + diabetes + hyper
                data = dat1,
                method = "lm",
                trControl = trainControl(method = "cv", number = 10))
fit.lm$results$Rsquared
```

```
## [1] 0.1481854
```

```
fit.lm$results$RsquaredSD
```

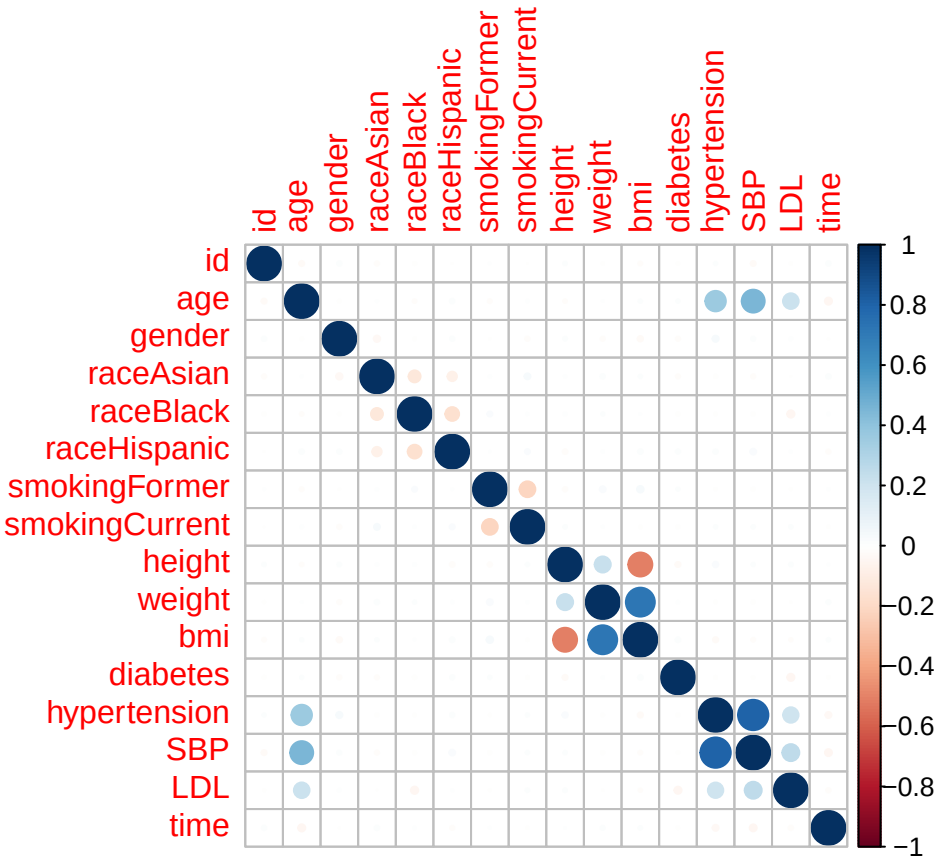
```
## [1] 0.03320516
```

```
#r2= 0.1470498
# adjusted r2=0.03320516
# result: model is poor fit
```

```
###  Ridge Modeling
set.seed(2)
# Split data into training and testing datasets 80/20 split
dat1_split = initial_split(dat1, prop = 0.8)

# Extract training and test data
dat1_train = training(dat1_split)
dat1_test = testing(dat1_split)
```

Ridge regression

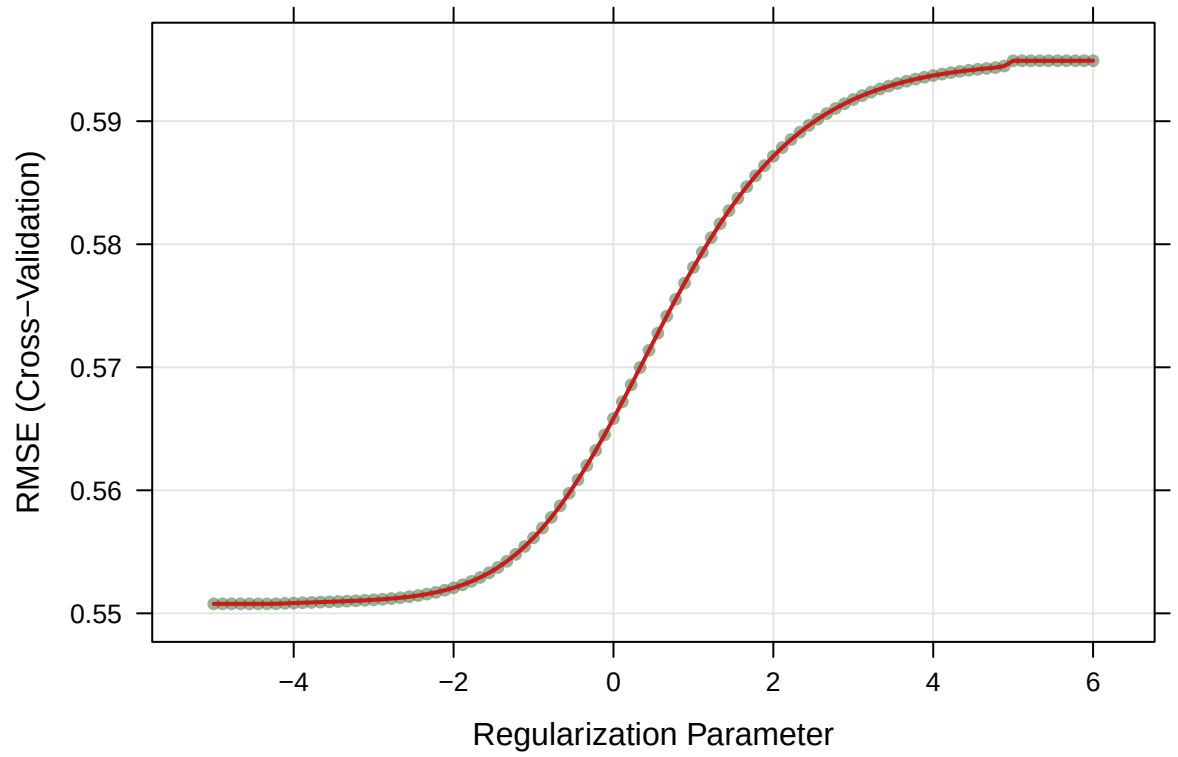
```
# matrix of predictors
x = model.matrix(log_antibody ~., trainData)[,-1]
y = trainData[, "log_antibody"]

corrplot(cor(x), method = "circle", type = "full")
```



```
# fitting Ridge regression using caret
ctrl1 = trainControl(method = "cv", number = 10)
set.seed(2)
ridge.fit <- train(log_antibody ~.,
                   data = trainData,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 0,
                                          lambda = exp(seq(6, -5, length = 100))),
                   trControl = ctrl1)
```

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

```
plot(ridge.fit, xTrans = log)
```




```
ridge.fit$bestTune
```

```
##   alpha     lambda
## 7     0 0.01312373
```

```
coef(ridge.fit$finalModel, ridge.fit$bestTune$lambda)
```

```
## 17 x 1 sparse Matrix of class "dgCMatrix"
```

```
##                           s1
## (Intercept)     1.253674e+01
## id             -1.434239e-07
## age            -2.012203e-02
## gender         -2.907560e-01
## raceAsian       1.167977e-02
## raceBlack       9.923777e-03
## raceHispanic   -2.240479e-02
## smokingFormer   2.333491e-02
## smokingCurrent -2.147113e-01
## height          9.684341e-05
## weight         -7.058063e-04
## bmi            -4.645545e-02
## diabetes        1.390199e-04
## hypertension   -3.062993e-02
## SBP             1.893603e-03
## LDL             4.463429e-05
## time           -1.461308e-04
```

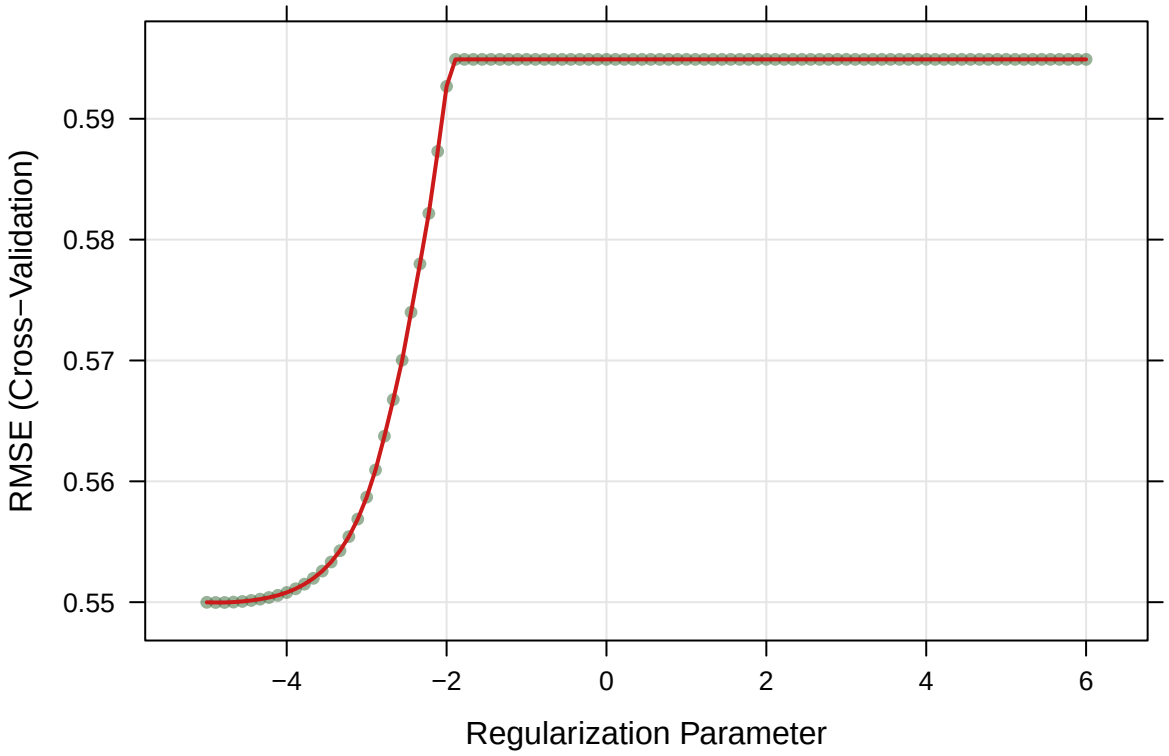
```r
### Lasso Modeling
set.seed(2)
# fitting lasso using caret
lasso.fit = train(log_antibody ~.,
                  data = trainData,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = 1,
                                         lambda = exp(seq(6, -5, length = 100))),
                  trControl = ctrl1)
```

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

```r
lasso.fit$bestTune
```

```
##   alpha      lambda
## 2     1 0.007529784
```

```r
plot(lasso.fit, xTrans = log)
```

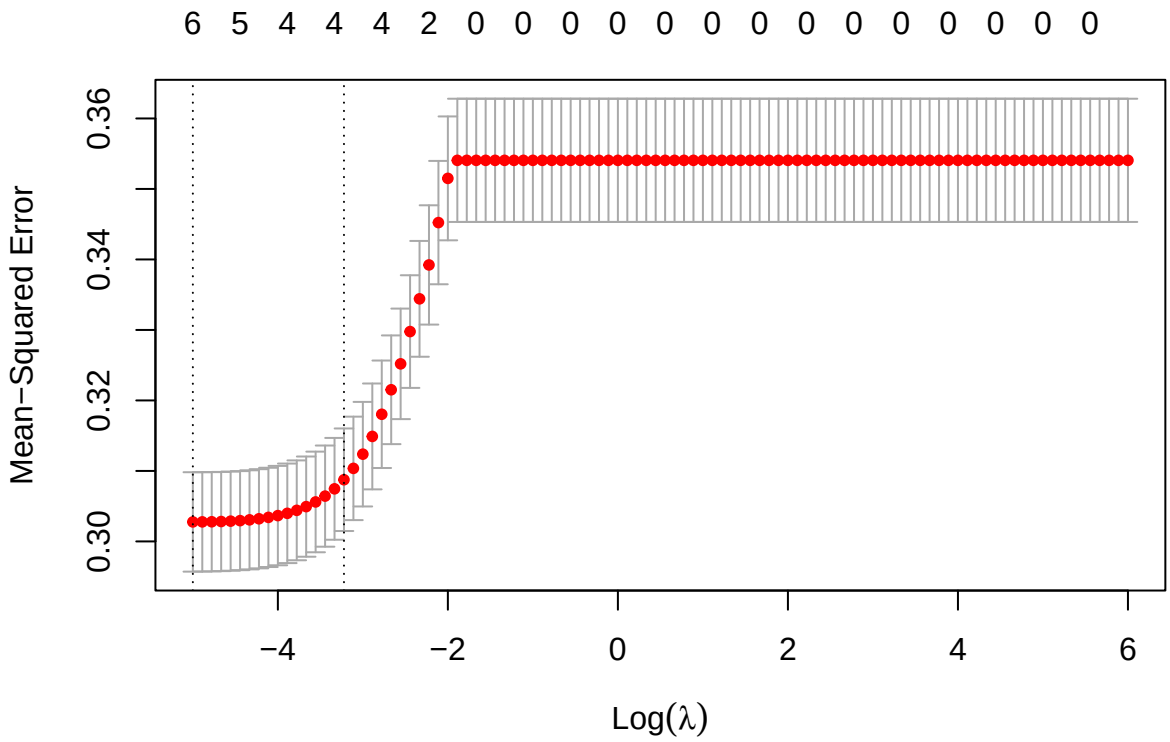



```
coef(lasso.fit$finalModel, lasso.fit$bestTune$lambda)
```

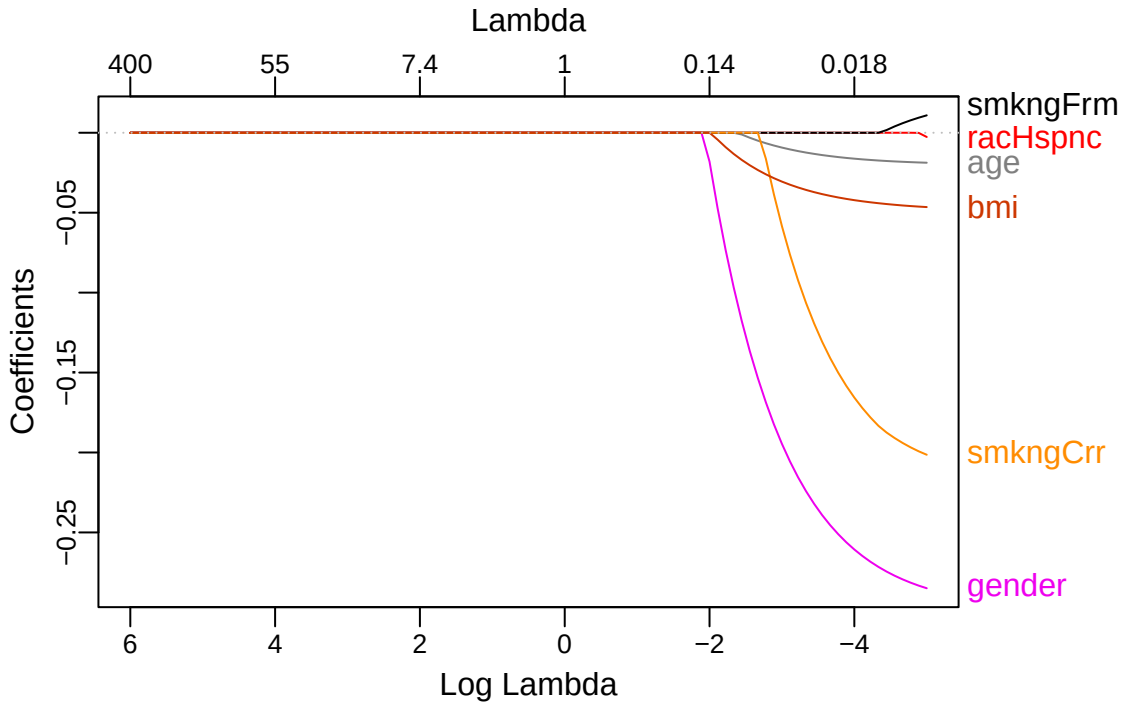
```
## 17 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)    12.6165394572
## id              .
## age            -0.0185746747
## gender         -0.2832854842
## raceAsian       .
## raceBlack       .
## raceHispanic   -0.0004898182
## smokingFormer   0.0094256969
## smokingCurrent -0.1992544787
## height          .
## weight          .
## bmi            -0.0461866744
## diabetes        .
## hypertension    .
## SBP             .
## LDL             .
## time            .
```

```
# OR
```

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




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




```
coef(cv.lasso, s = "lambda.min")
```

```
## 17 x 1 sparse Matrix of class "dgCMatrix"
##                            s1
## (Intercept)     12.635694838
## id               .
## age             -0.018744168
## gender          -0.284914273
## raceAsian        .
## raceBlack        .
## raceHispanic    -0.002628035
## smokingFormer    0.010892162
## smokingCurrent -0.201419004
## height           .
## weight           .
## bmi             -0.046482646
## diabetes         .
## hypertension     .
## SBP              .
## LDL              .
## time             .
```

```
head(predict(cv.lasso, newx = x, s = "lambda.min"))
```

```
##   lambda.min
```

```
## 1  10.013236
## 2  10.102459
## 3   9.831887
## 4   9.919029
## 5  10.030597
## 6   9.766122
```

Table 2: ANOVA Comparison of GAM Models (Dat1)

Model	Resid. Df	Resid. Dev	Df	Deviance	F	Pr(>F)
gam.m1	4984.000	1509.442	NA	NA	NA	NA
gam.m2	4971.177	1380.053	12.82350	129.389611	36.37749	0.0000000
gam.m3	4968.540	1378.818	2.63652	1.234568	1.68820	0.1738572

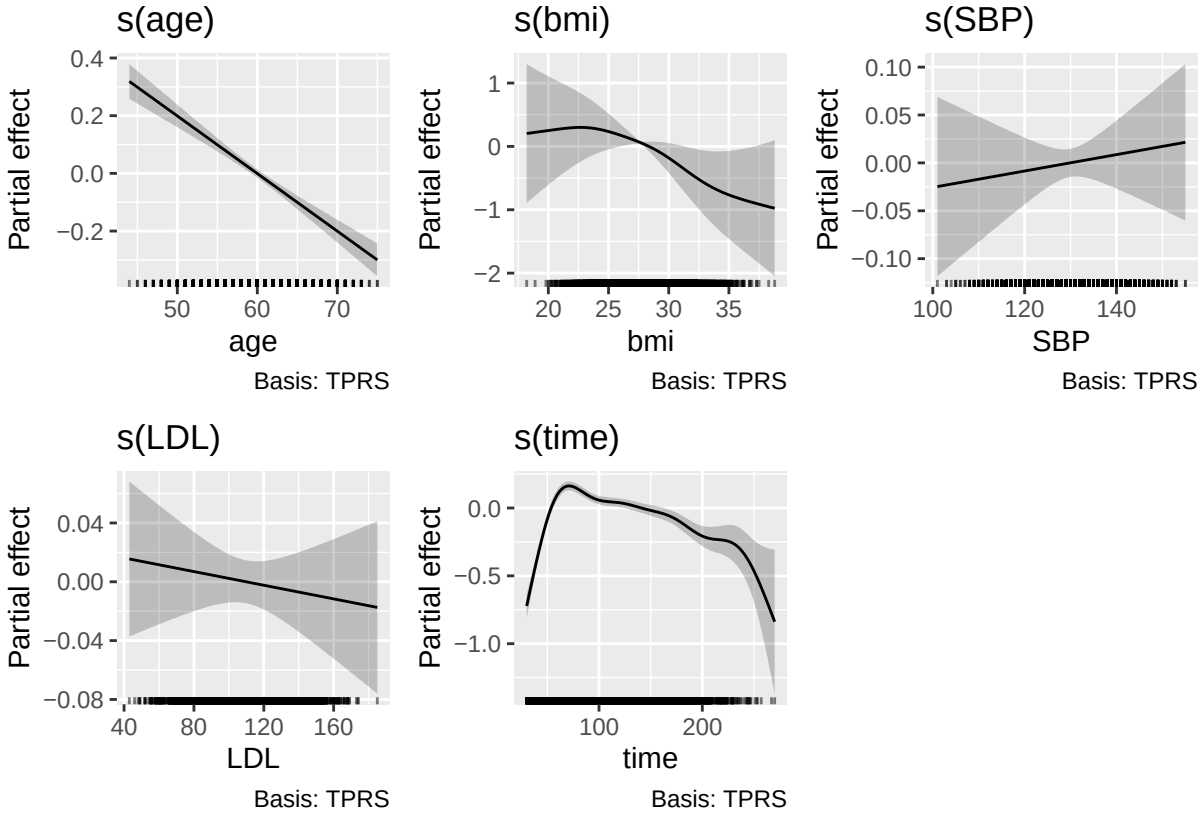


Figure 9: Smooth terms for GAM model (gam.m2)

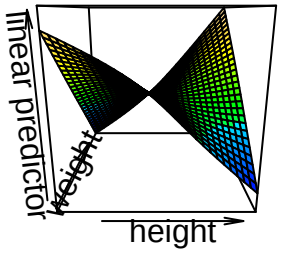
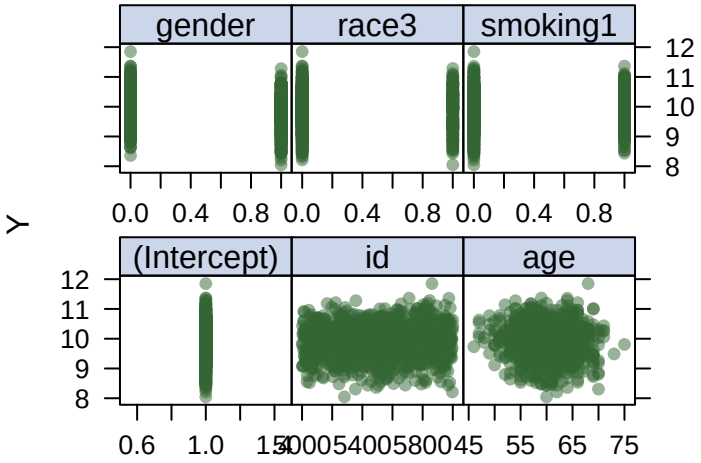
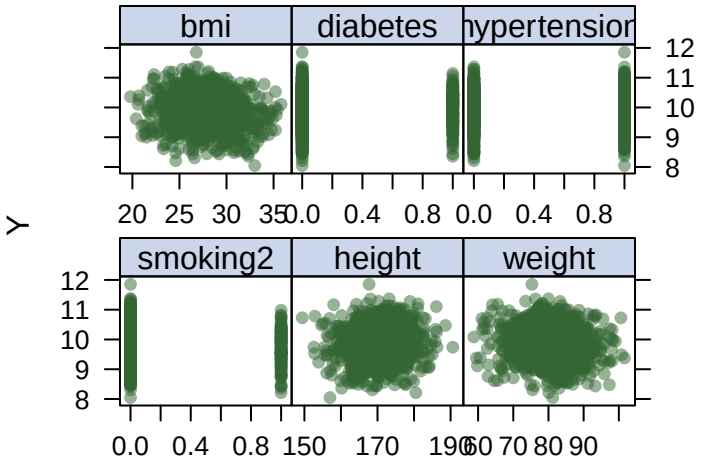


Figure 10: Visualization of Model 3 (dat1)





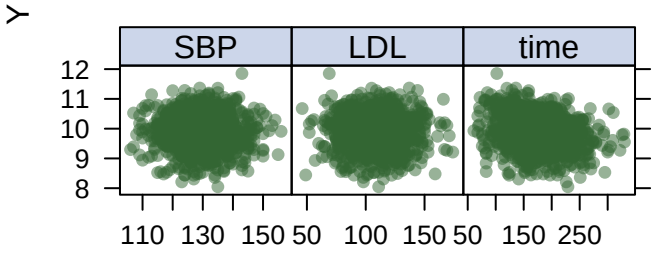
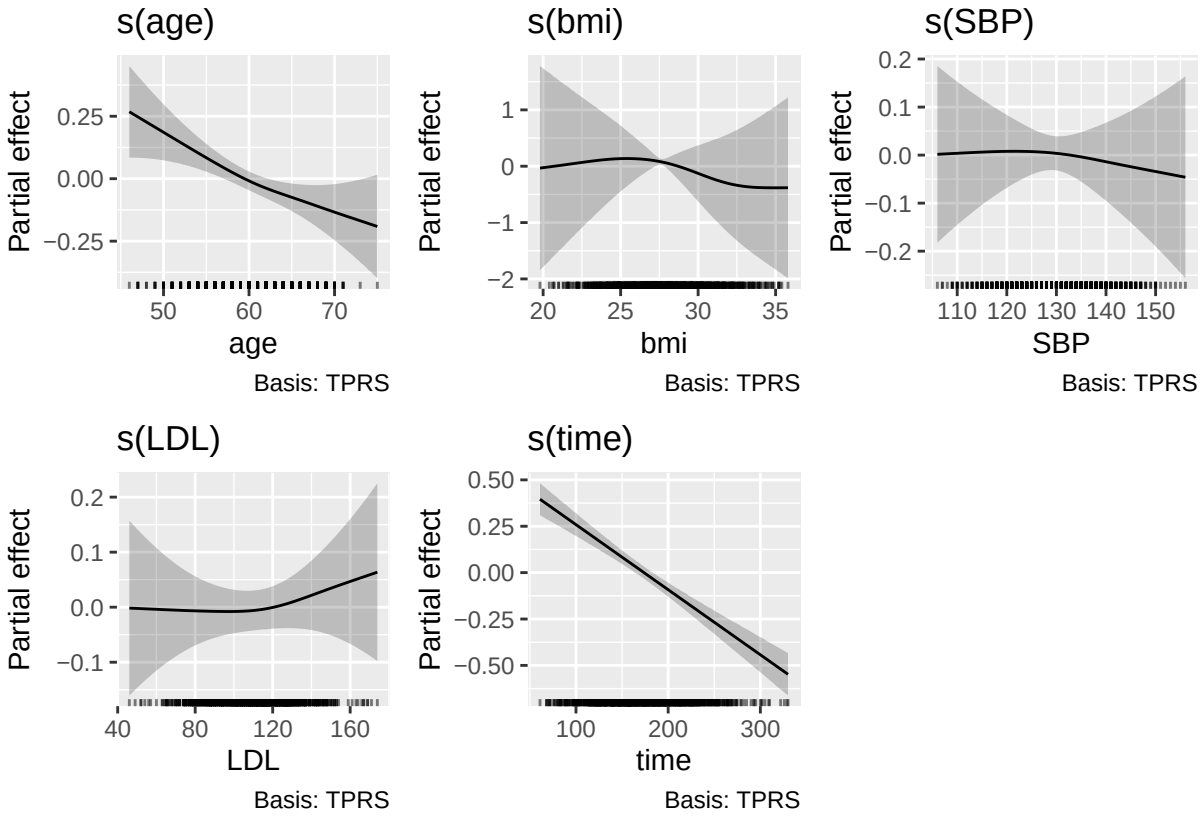
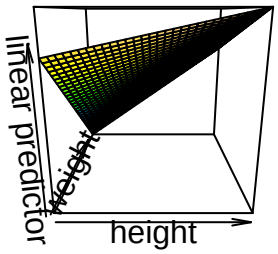


Table 3: ANOVA Comparison of GAM Models (Dat2)

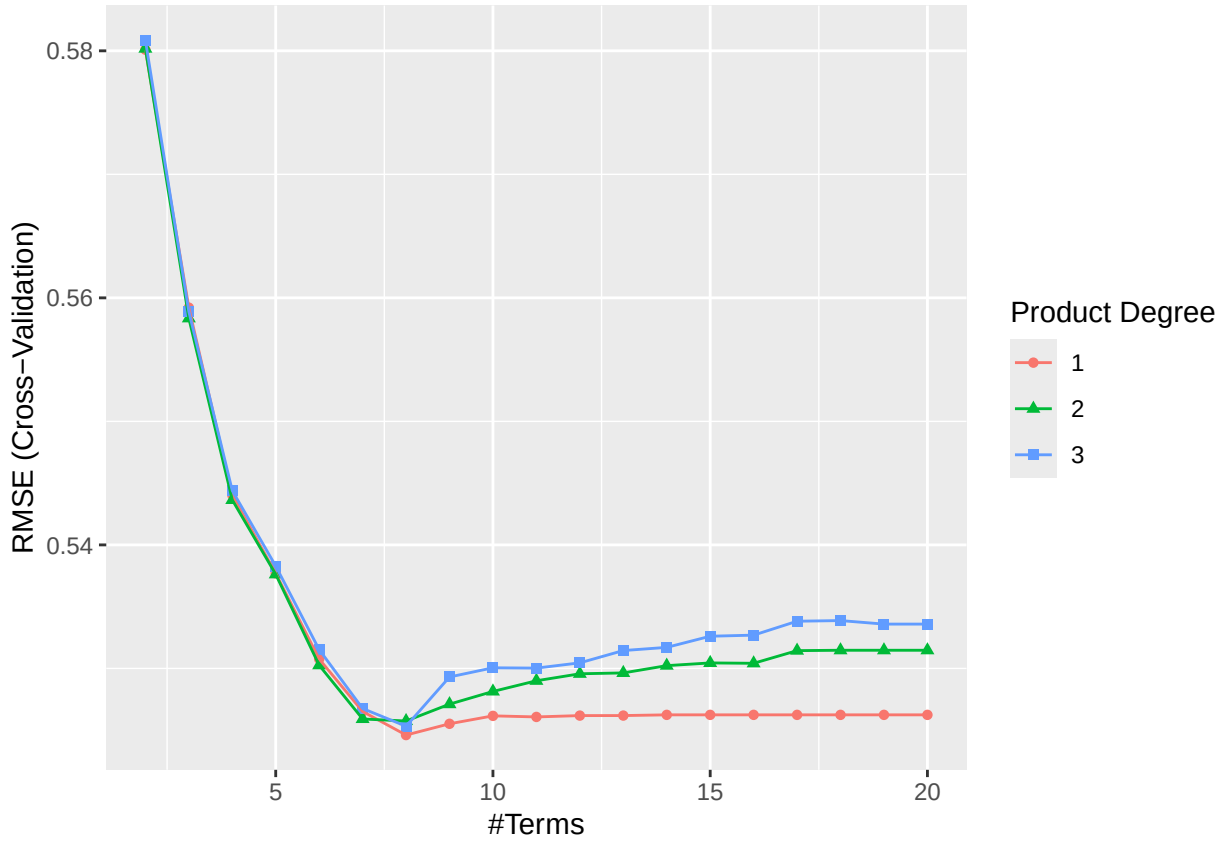
Model	Resid. Df	Resid. Dev	Df	Deviance	F	Pr(>F)
gam.m1	984.0000	275.4546	NA	NA	NA	NA
gam.m2	977.7820	268.5783	6.217997	6.8762407	4.0341782	0.0004416
gam.m3	976.6323	268.2745	1.149679	0.3038207	0.9640406	0.3384056






```
### MARS Modeling
set.seed(2)
x = model.matrix(log_antibody ~ age + gender + race + smoking + height + weight + bmi +
                   diabetes + hypertension + SBP + LDL + time, data = trainData)[, -1]

y = trainData$log_antibody
```

```
ctrl1 = trainControl(method = "cv", number = 10)
mars_grid = expand.grid(degree = 1:3,
                        nprune = 2:20)
set.seed(2)
mars.fit = train(x, y, method = "earth",
                 tuneGrid = mars_grid, trControl = ctrl1)
ggplot(mars.fit)
```




```
mars.fit$bestTune
```

```
##   nprune degree
## 7      8      1
```

```
coef(mars.fit$finalModel)
```

```
##    (Intercept)     h(time-57)     h(57-time)    h(27.9-bmi)         gender
##   10.631298637   -0.002153431   -0.033562469   -0.062977631   -0.299216256
##      h(70-age) smokingCurrent    h(bmi-23.6)
##    0.020111911   -0.228383083   -0.084778772
```

```
x_test = model.matrix(log_antibody ~ age + gender + race + smoking + height + weight + bmi +
                        diabetes + hypertension + SBP + LDL + time, data = testData)[, -1]
y_test = testData$log_antibody
```

```r
predictions = predict(mars.fit, newdata = x_test)
mse = mean((predictions- y_test)^2)
cat("MSE:", format(mse, nsmall = 3), "\n")
```

```
## MSE: 0.2915797
```

```r
dim(x)
```

```
## [1] 4000   15
```

```r
### Resampling

#Lm model for the resampling:
set.seed(2)
fit.lm <- train(log_antibody ~ age + gender + race + smoking + height + weight + bmi + diabetes + hyper
data = trainData,
method = "lm",
trControl = trainControl(method = "cv", number = 10))
```

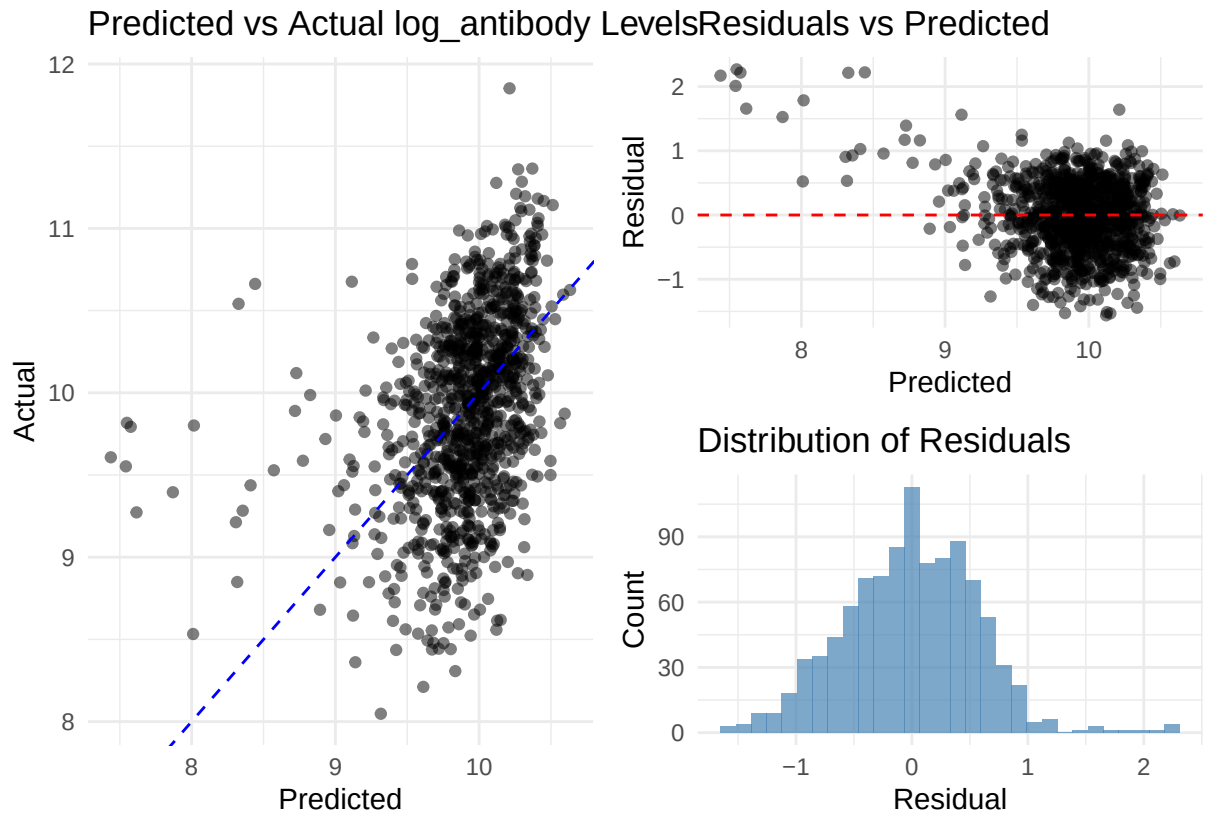


Figure 11: Diagnostic Plots for Prediction Model on dat2