Set 6

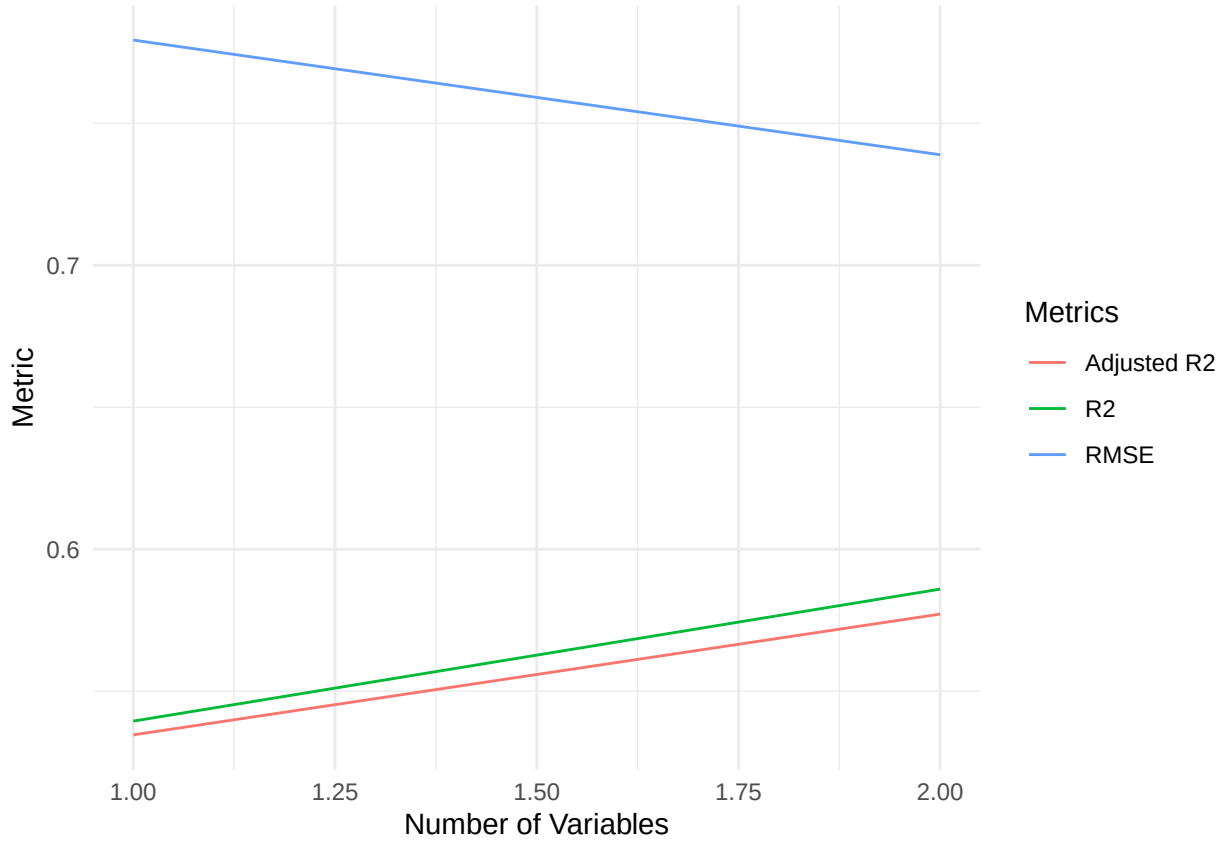
Oluwanifemi

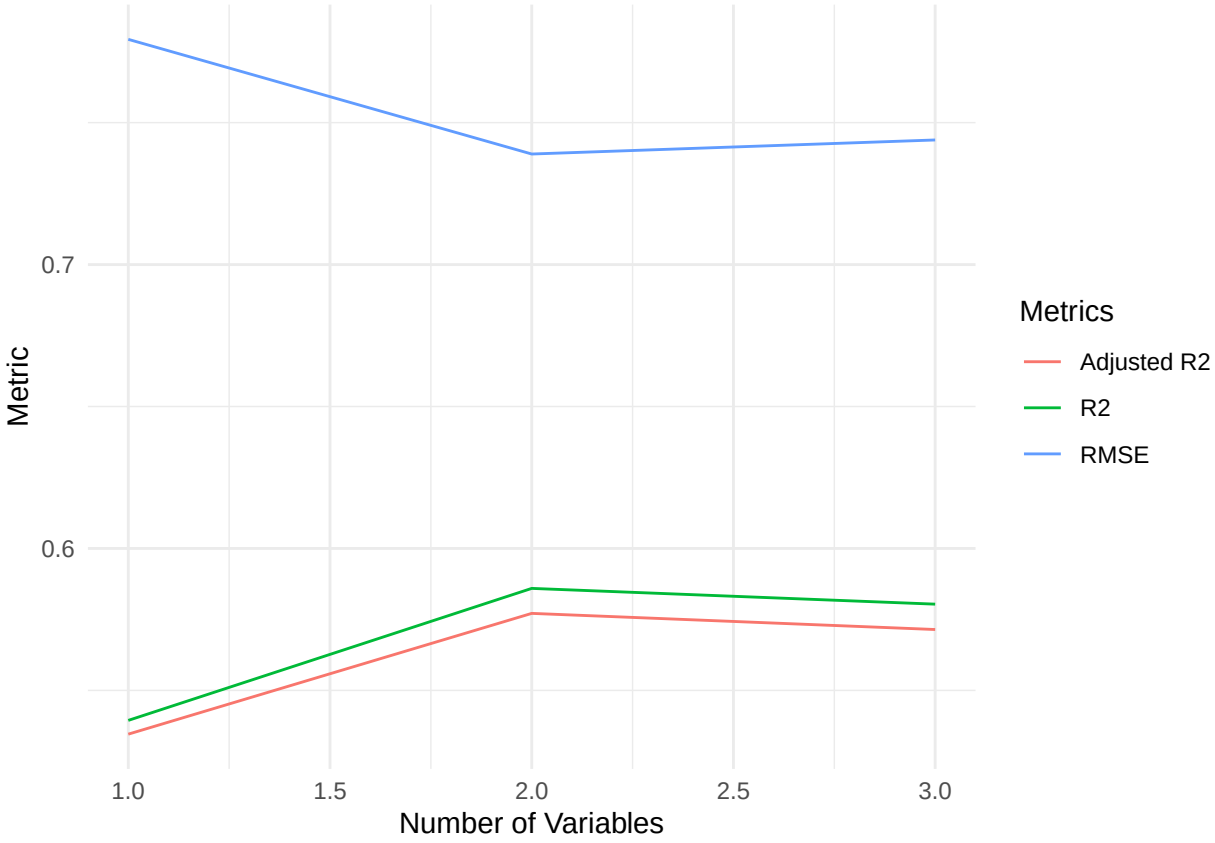
2024-11-01

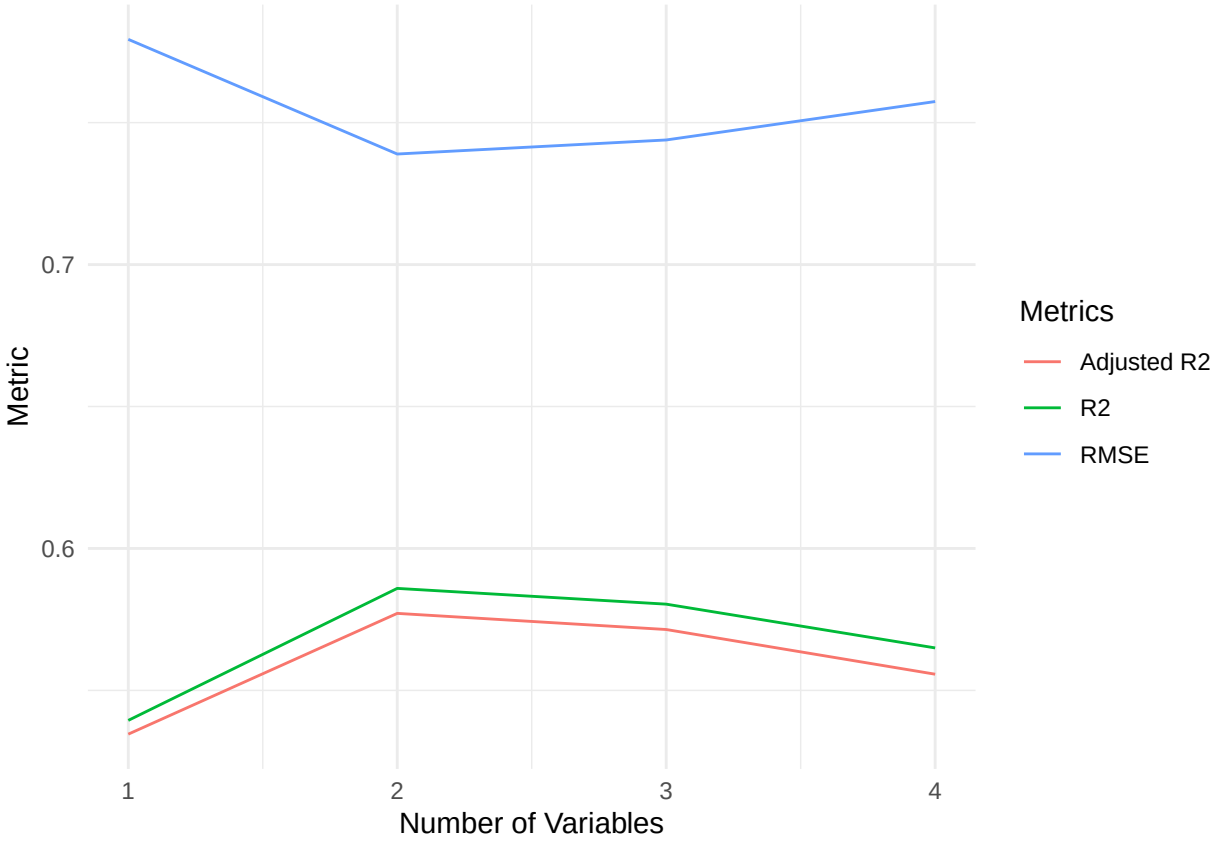
```r
# Load required libraries
library(faraway)
library(ggplot2)

# Load data
data(prostate)
```

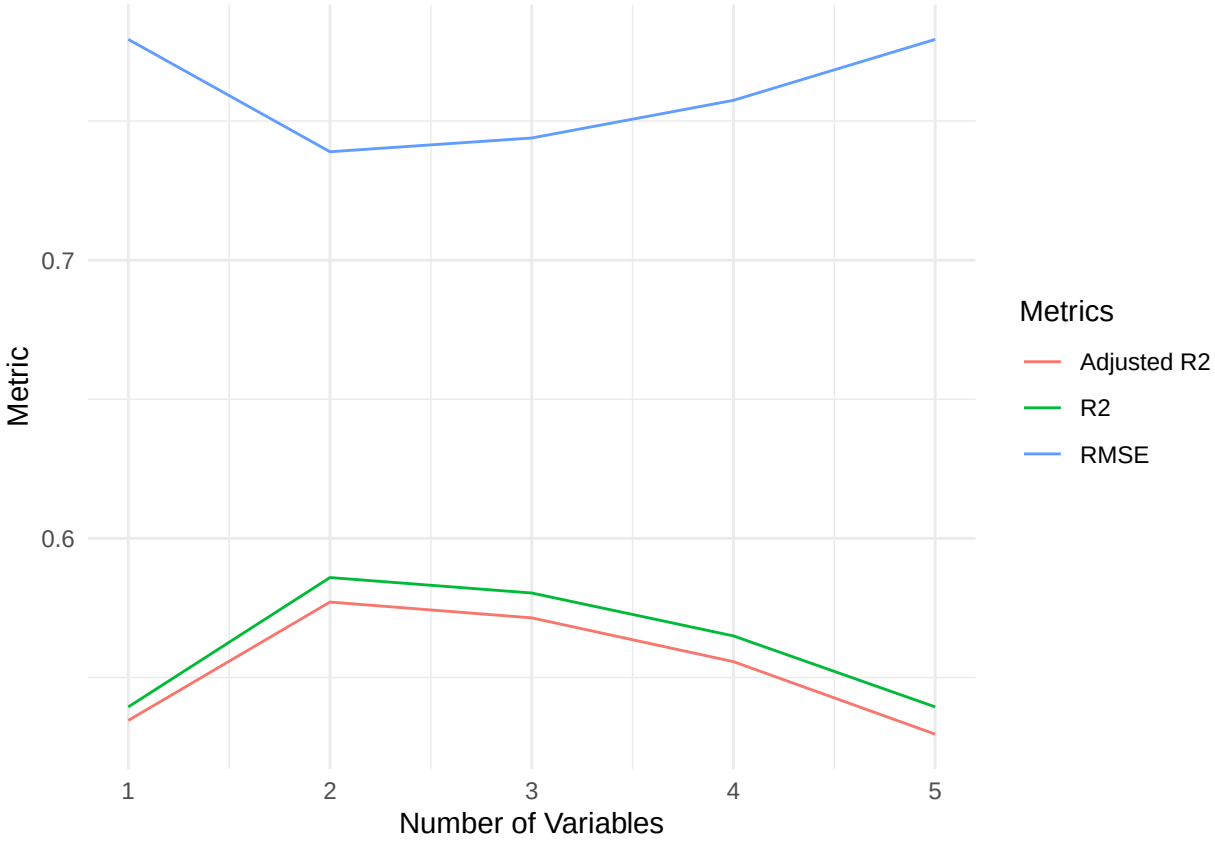
```r
model_1 <- lm(lpsa ~ lcavol, data = prostate)
View(prostate)
summary_1 <- summary(model_1)
r2_1 <- summary_1$r.squared
adj_r2_1 <- summary_1$adj.r.squared
rmse_1 <- sqrt(mean(model_1$residuals^2))

# Store results for plotting
results <- data.frame(
  Variables = 1,
  R2 = r2_1,
  Adj_R2 = adj_r2_1,
  RMSE = rmse_1
)

predictors <- c("lweight", "svi", "lbph", "age", "lcp", "pgg45", "gleason")
model_current <- model_1

for (i in 1:length(predictors)) {

  model_current <- update(model_current, . ~ . + get(predictors[i]))

  # Calculate metrics
  model_summary <- summary(model_current)
  r2 <- model_summary$r.squared
  adj_r2 <- model_summary$adj.r.squared
  rmse <- sqrt(mean(model_current$residuals^2))

  # Store metrics
  results <- rbind(results, data.frame(Variables = i + 1, R2 = r2, Adj_R2 = adj_r2, RMSE = rmse))
   #plotting each
 gg <- ggplot(results, aes(x = Variables)) +
  geom_line(aes(y = R2, color = "R2")) +
  geom_line(aes(y = Adj_R2, color = "Adjusted R2")) +
  geom_line(aes(y = RMSE, color = "RMSE")) +
```

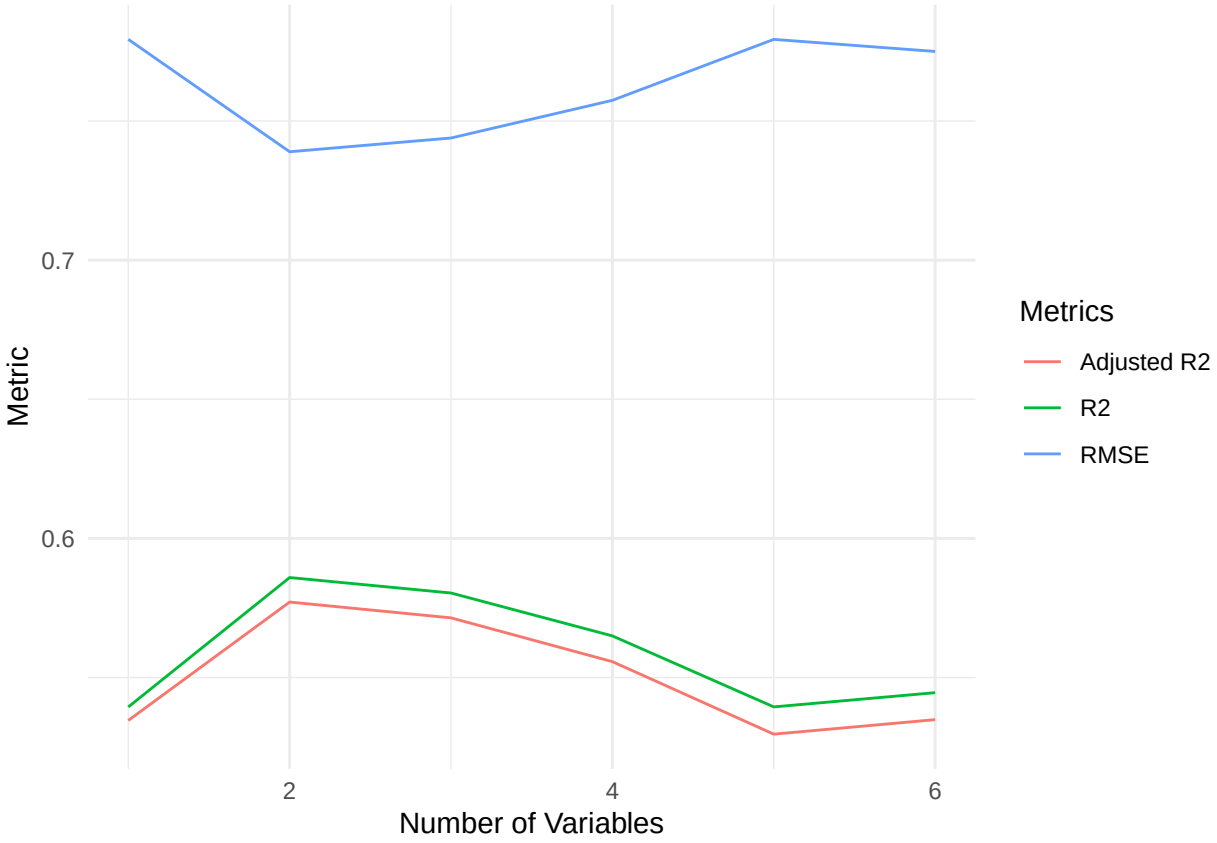
```
  labs(x = "Number of Variables", y = "Metric", color = "Metrics") +
  theme_minimal()
 print(gg)
}
```

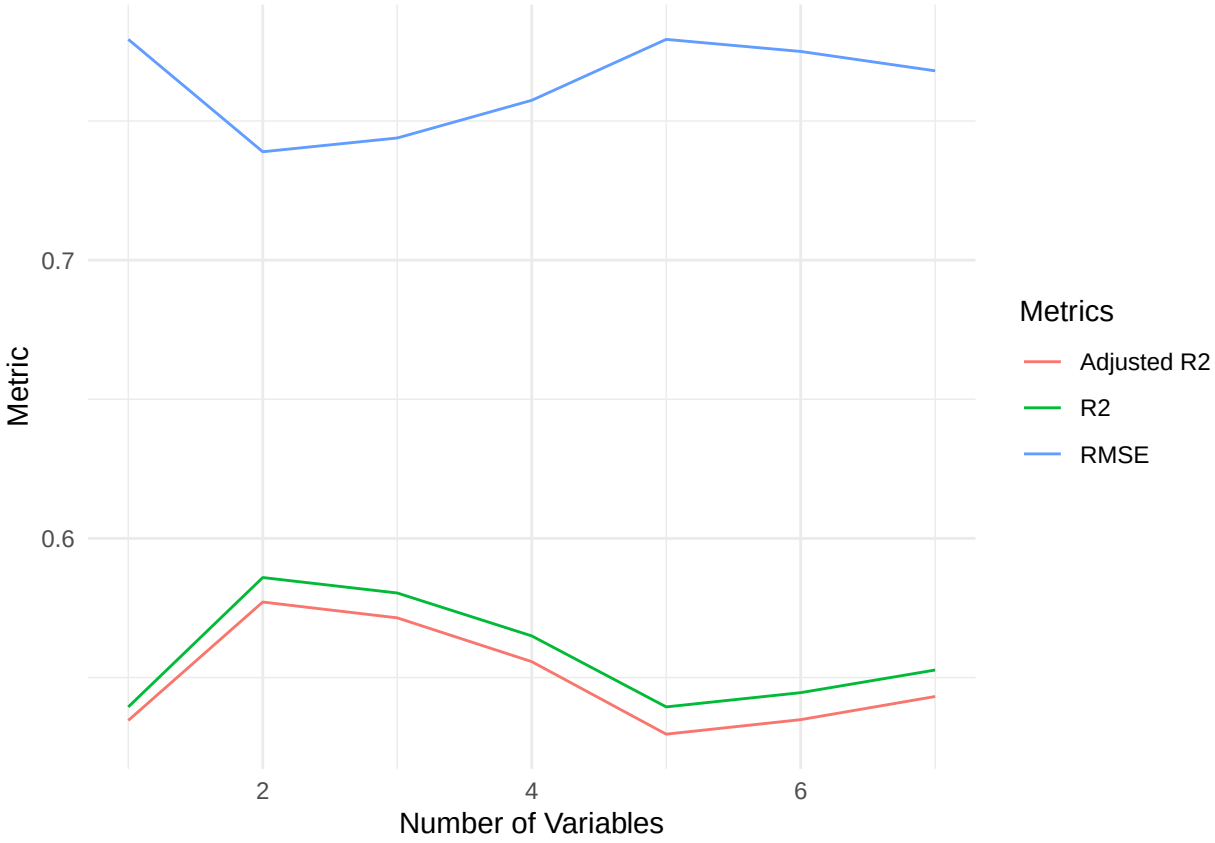


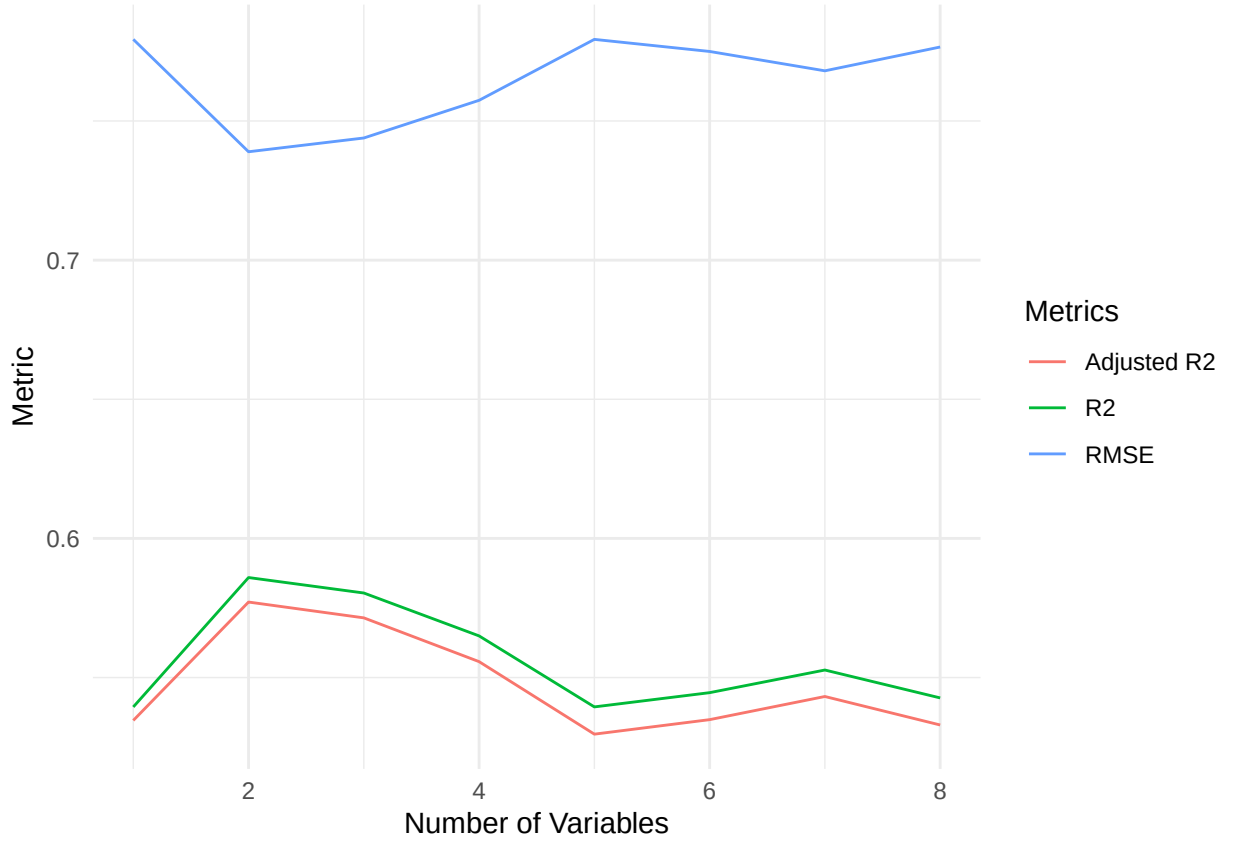

0.7
0.6
Metric
1.0
1.5
2.0
2.5
3.0
Number of Variables
Metrics
Adjusted R2
R2
RMSE


0.7
0.6
Metric
1
2
3
4
Number of Variables
Metrics
Adjusted R2
R2
RMSE


0.7
0.6
Metric
1
2
3
4
5
Number of Variables
Metrics
Adjusted R2
R2
RMSE


0.7
0.6
Metric
2
4
6
Number of Variables
Metrics
Adjusted R2
R2
RMSE

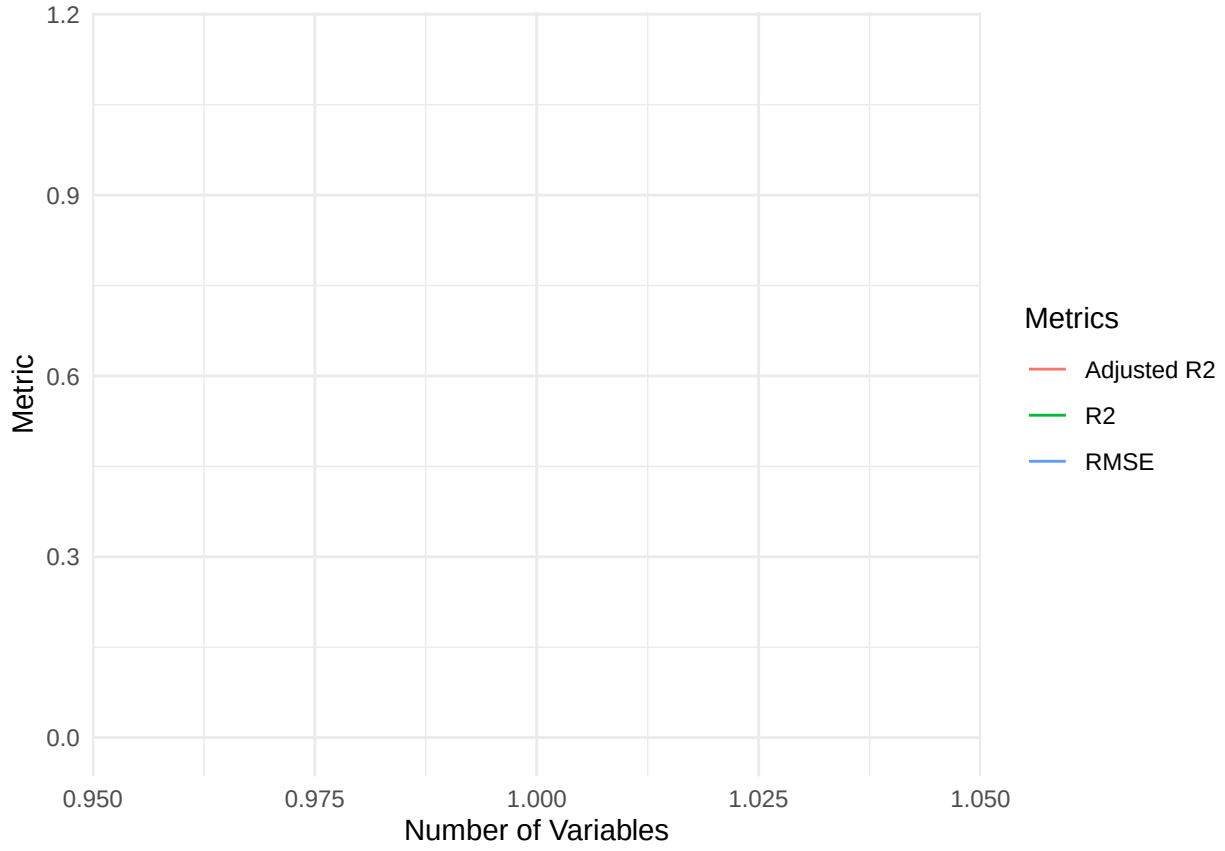

0.7
0.6
Metric
2
4
6
Number of Variables
Metrics
Adjusted R2
R2
RMSE

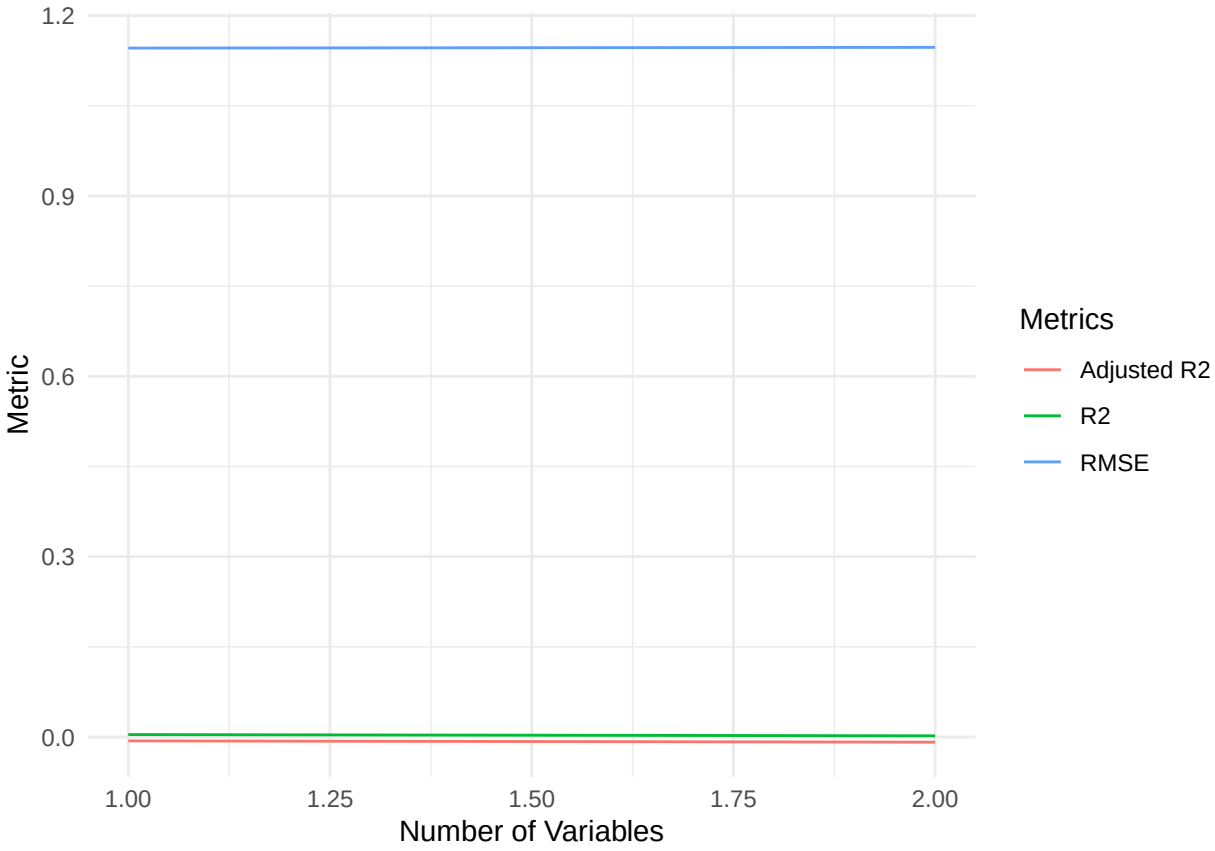


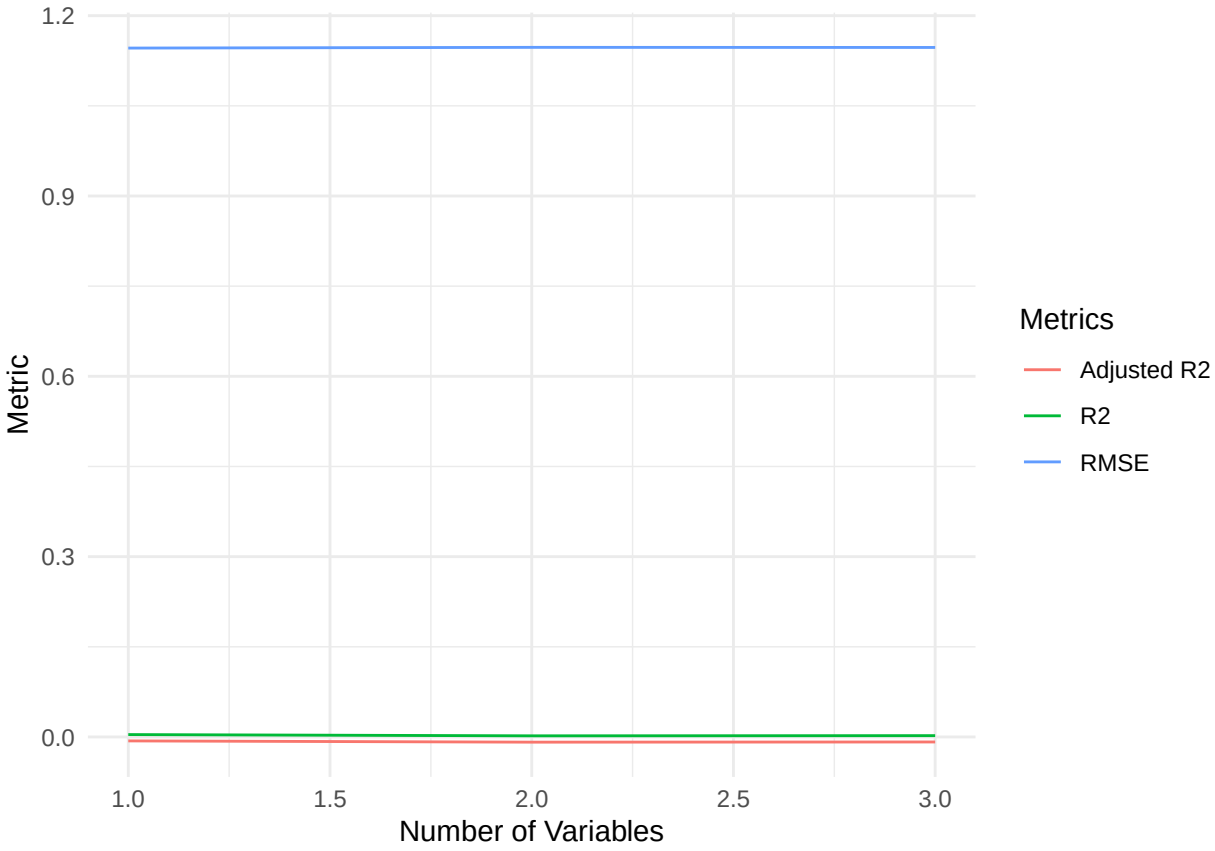

```r
set.seed(42)
results1 <- data.frame(Variables = integer(), R2 = double(), Adj_R2 = double(), RMSE = double())
model_norm <- lm(lpsa ~ 1, data = prostate)
for(i in 1:10){
  n <- nrow(prostate)
  x_1 <- rnorm(n,0,1)
  #model_norm <- lm(lpsa ~ x_1, data = prostate)
  #x_2 <- rnorm(n,0,1)
  model_norm <- update(model_norm, . ~ . + x_1)

  # Calculate metrics
  model_summary1 <- summary(model_norm)
  r2 <- model_summary1$r.squared
  adj_r2 <- model_summary1$adj.r.squared
  rmse <- sqrt(mean(model_norm$residuals^2))

   # Store metrics
  results1 <- rbind( results1,data.frame(Variables = i , R2 = r2, Adj_R2 = adj_r2, RMSE = rmse))

   gg <- ggplot(results1, aes(x = Variables)) +
  geom_line(aes(y = R2, color = "R2")) +
  geom_line(aes(y = Adj_R2, color = "Adjusted R2")) +
  geom_line(aes(y = RMSE, color = "RMSE")) +
  labs(x = "Number of Variables", y = "Metric", color = "Metrics") +
  theme_minimal()
 print(gg)
```

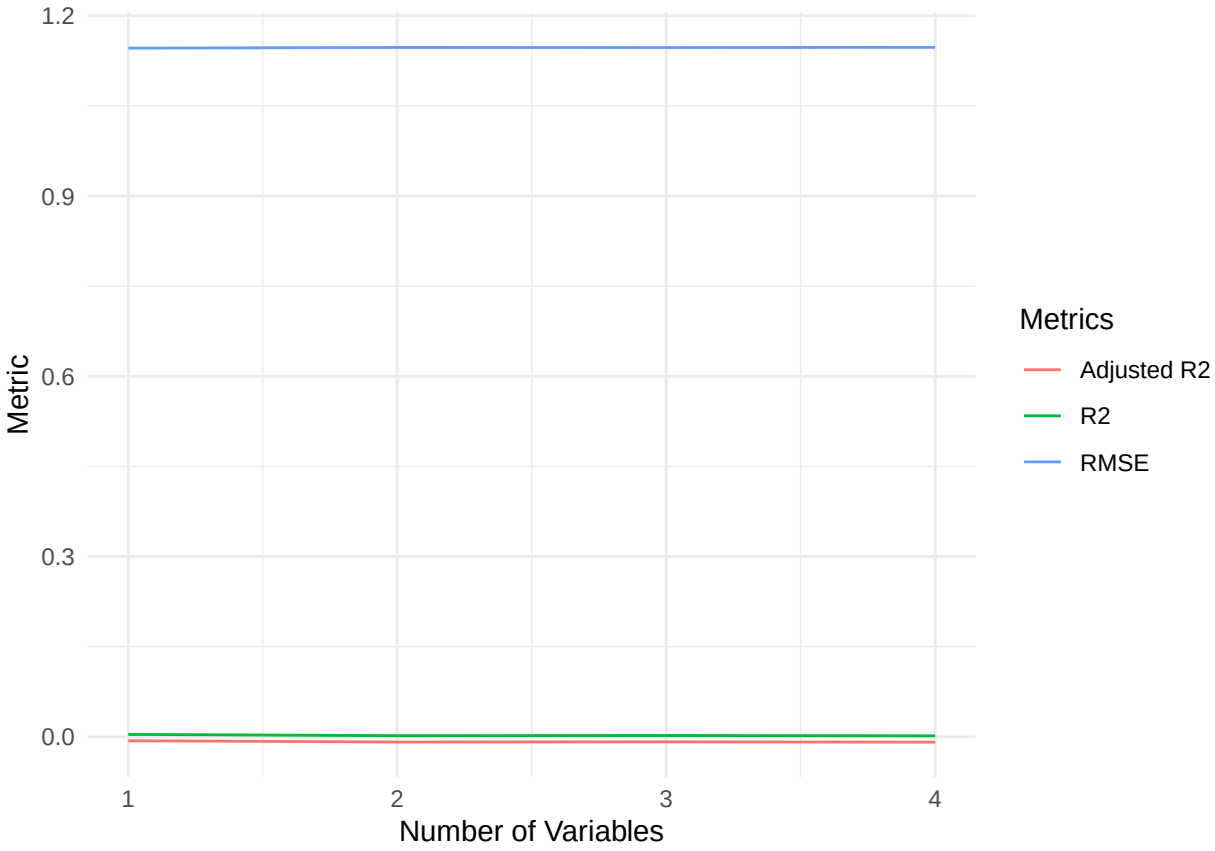
```
}
```

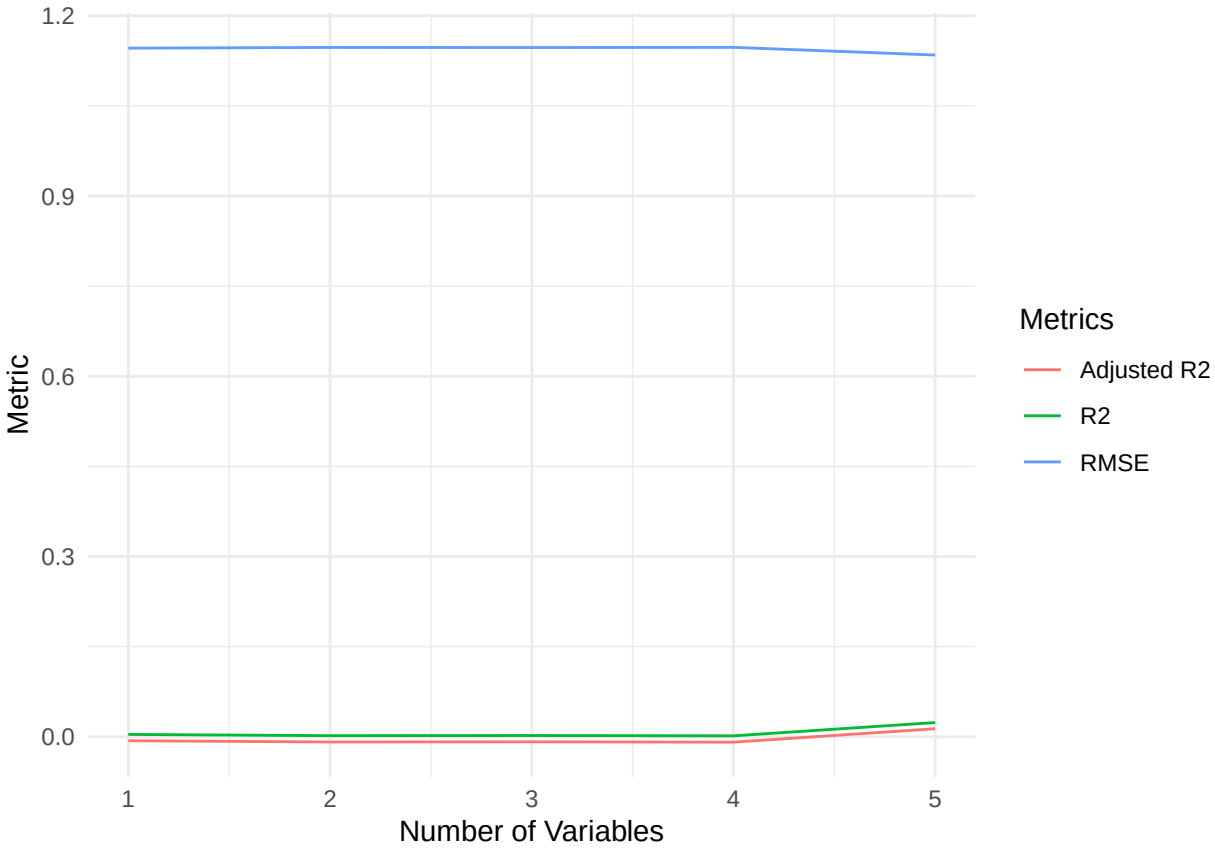
```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```

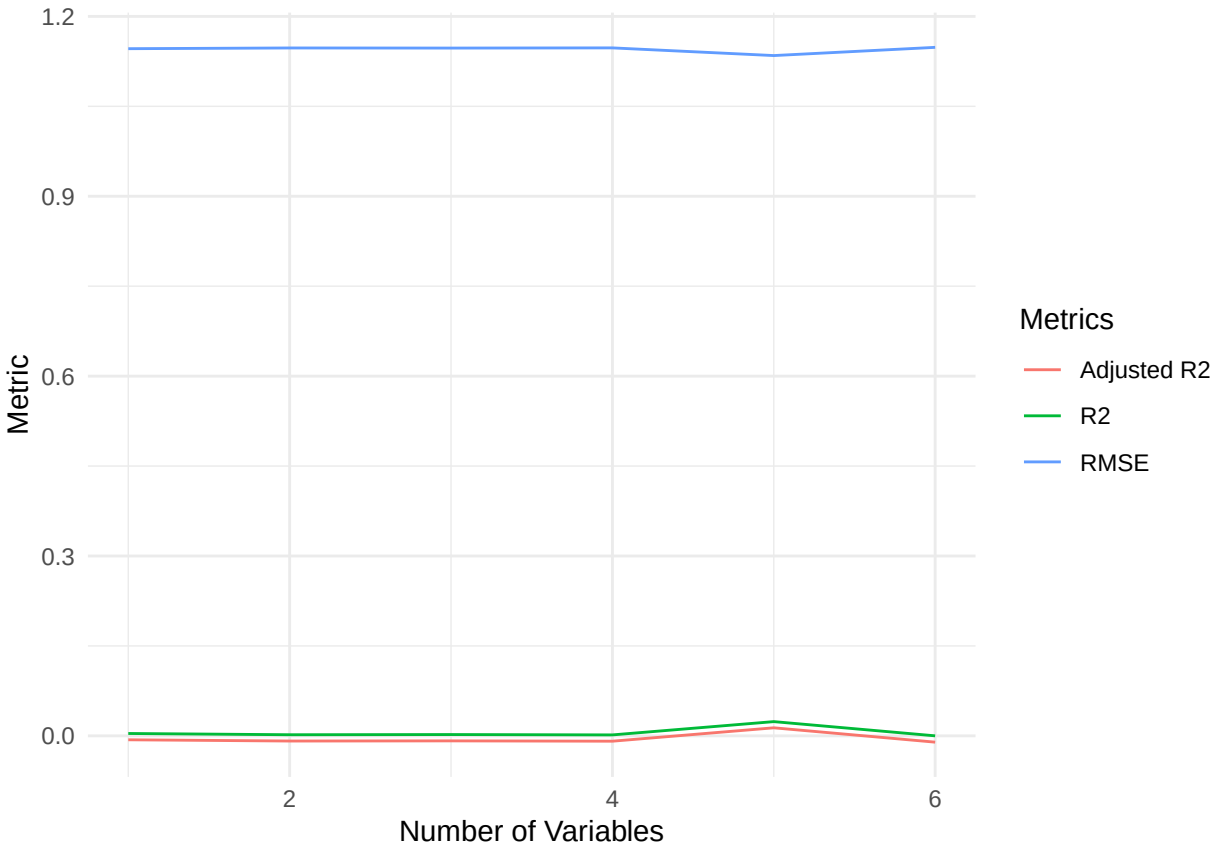


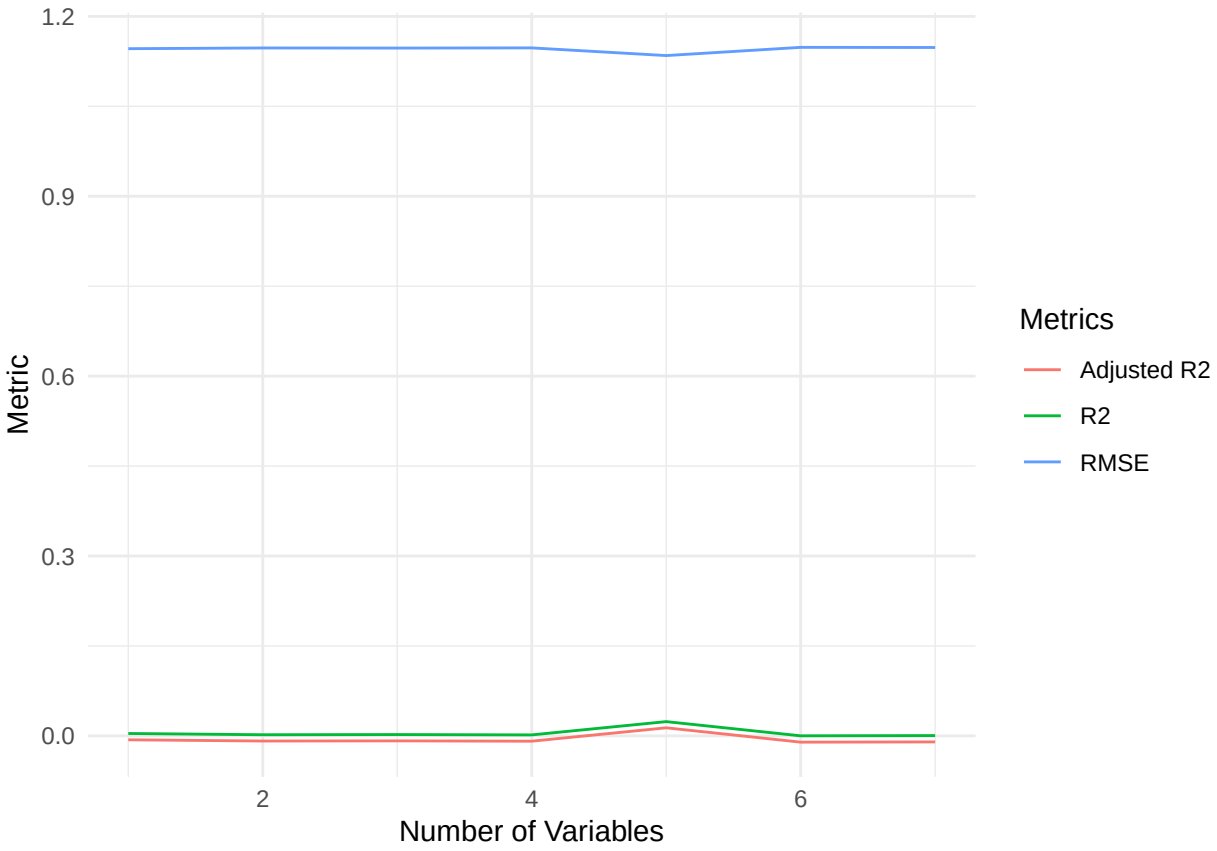

1.2
0.9
0.6
0.3
0.0
Metric
1.00
1.25
1.50
1.75
2.00
Number of Variables
Metrics
Adjusted R2
R2
RMSE

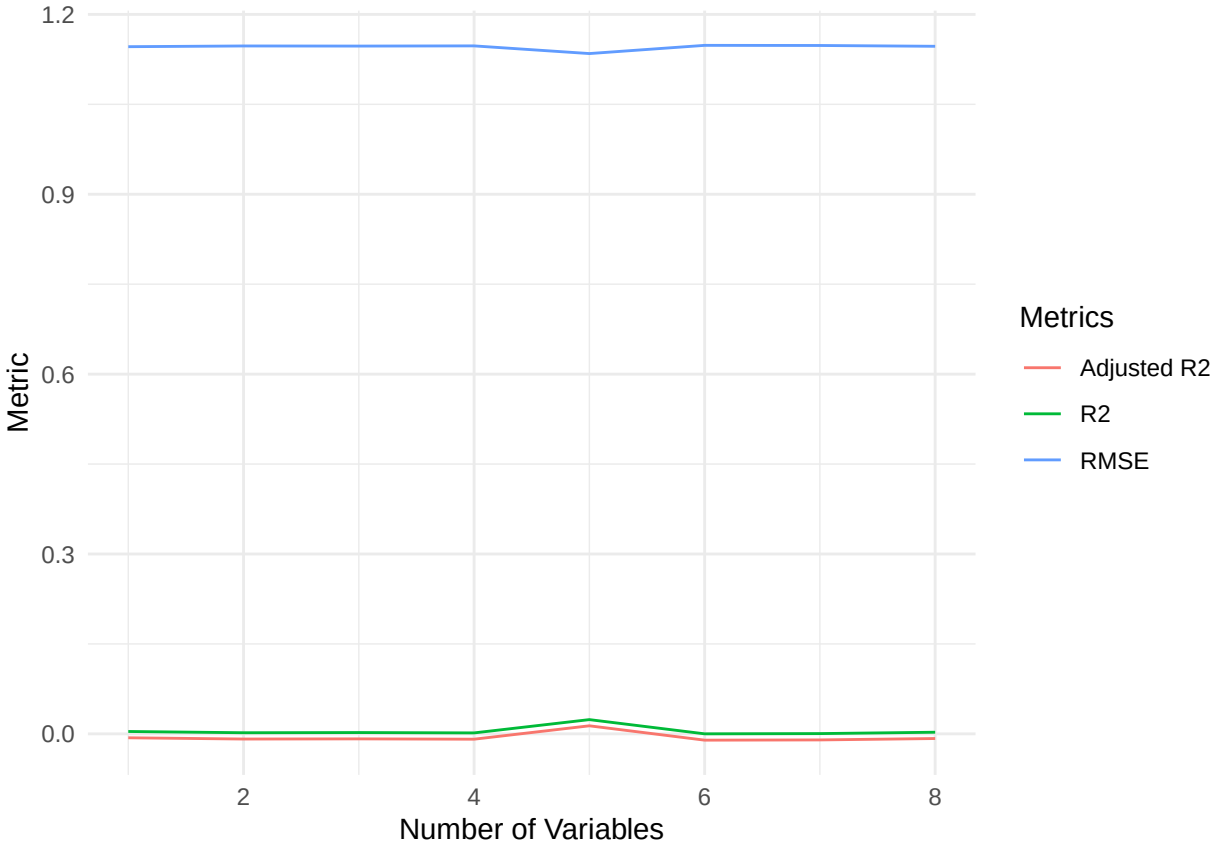

1.2
0.9
0.6
0.3
0.0
Metric
1.0
1.5
2.0
2.5
3.0
Number of Variables
Metrics
Adjusted R2
R2
RMSE

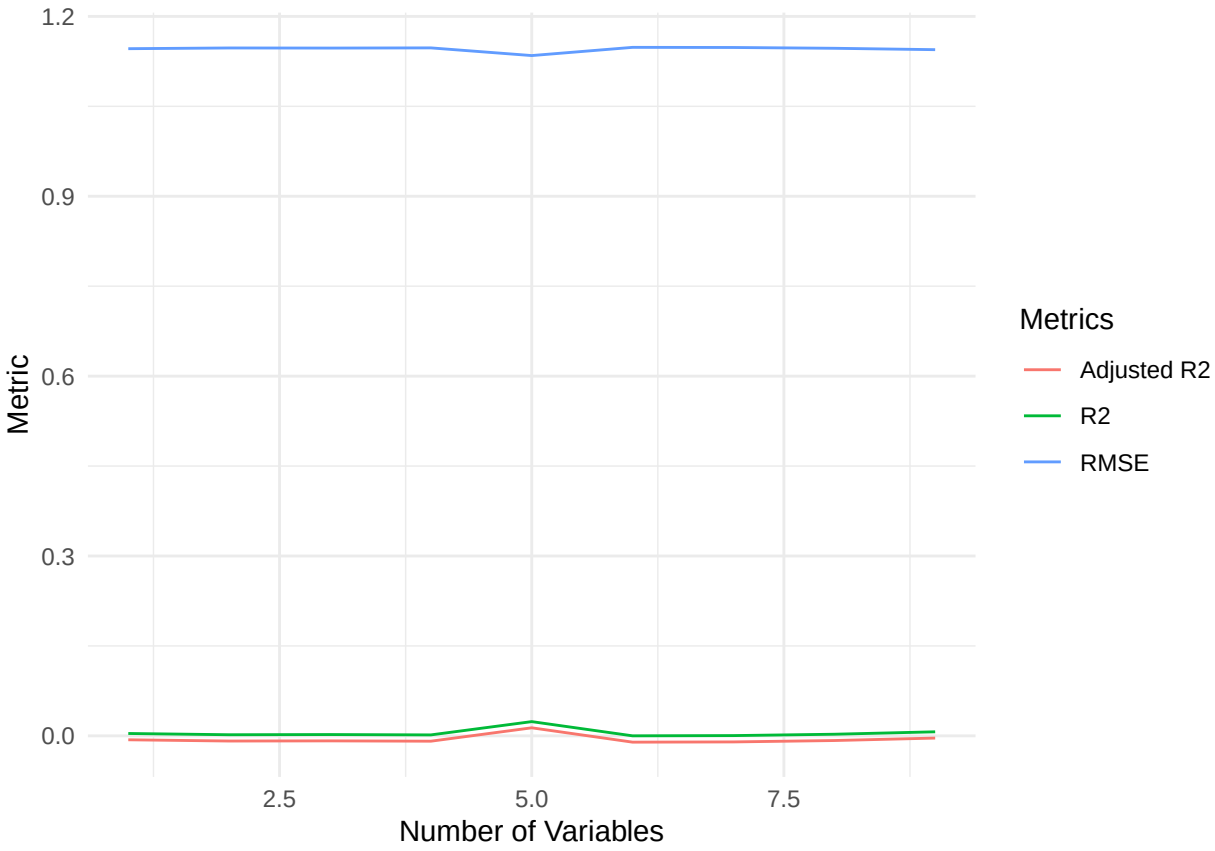

1.2
0.9
0.6
0.3
0.0
Metric
1
2
3
4
Number of Variables
Metrics
Adjusted R2
R2
RMSE

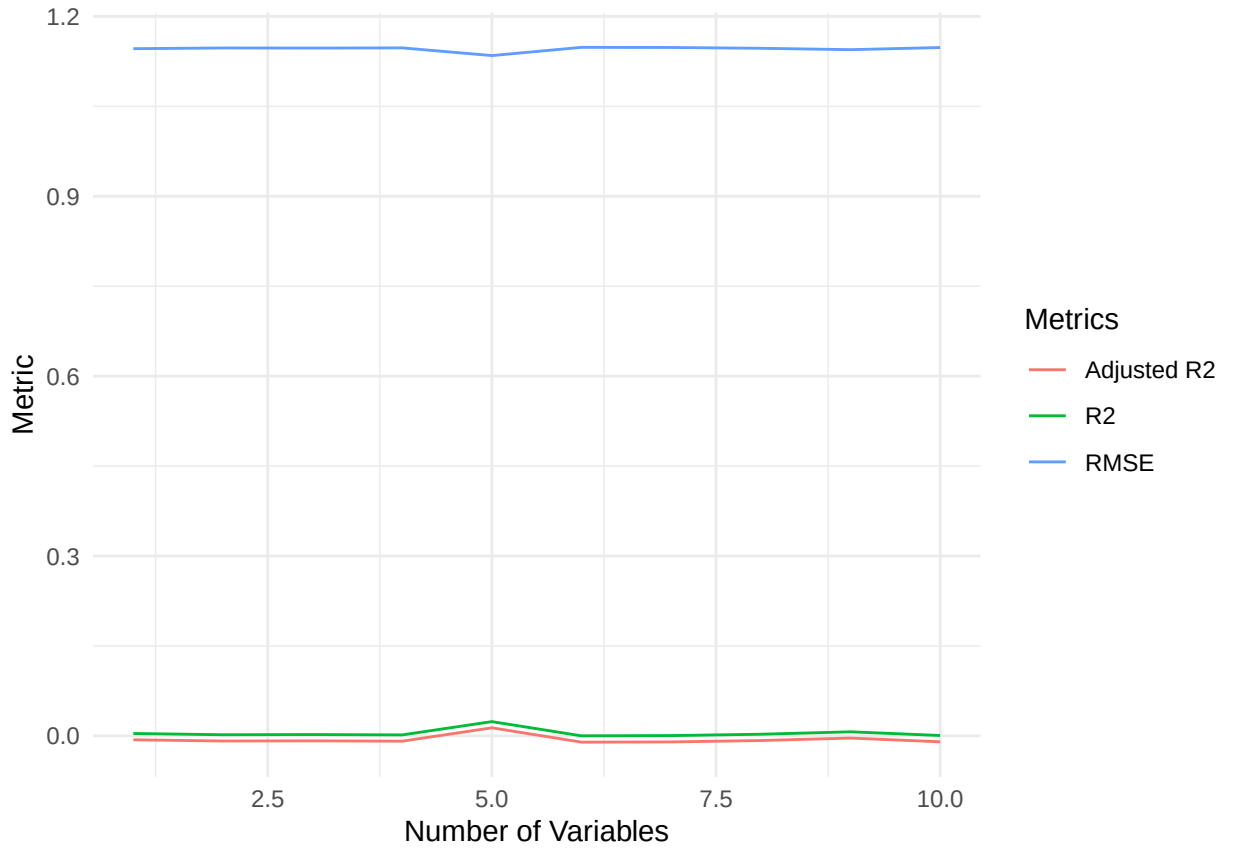

1.2
0.9
0.6
0.3
0.0
Metric
1
2
3
4
5
Number of Variables
Metrics
Adjusted R2
R2
RMSE


1.2
0.9
0.6
0.3
0.0
Metric
2
4
6
Number of Variables
Metrics
Adjusted R2
R2
RMSE


1.2
0.9
0.6
0.3
0.0
Metric
2
4
6
Number of Variables
Metrics
Adjusted R2
R2
RMSE


1.2
0.9
0.6
0.3
0.0
Metric
2
4
6
8
Number of Variables
Metrics
Adjusted R2
R2
RMSE


1.2
0.9
0.6
0.3
0.0
Metric
2.5
5.0
7.5
Number of Variables
Metrics
Adjusted R2
R2
RMSE



d) i)I noticed that the metrics that is R^2, adjusted R^2, etc changed as more covariates were added, this is expecially noted in the graphs

ii)compared to the actual covariates, the junk variates looked more similar even as more covariates were added until about 5 junk variates were added, then it started looking a little different

iii) The metrics with 4 to 5 covariates for the non-junk covariates did not look very different from each other in the sense of the graphical visualization.

Problem 2: Matrix Proofs

Part (a): Show that $(X'X)' = X'X$

Given the property that for matrices $A \in \mathbb{R}^{m \times l}$ and $B \in \mathbb{R}^{l \times k}$, $(AB)' = B'A'$, we can proceed as follows:

1. Set $A = X'$ and $B = X$, so that $X'X$ is the product of an $n \times p$ matrix X' and a $p \times n$ matrix X.
2. By the transpose property:
$$(X'X)' = X'(X')' = X'X$$
3. Since $X'X$ is symmetric, it follows that $(X'X)' = X'X$.

Part (b): Show that $[(X'X)^{-1}]' = (X'X)^{-1}$

To show that the inverse of $X'X$ is symmetric, we use the following property: if a matrix M is invertible, then $(M')^{-1} = (M^{-1})'$.

1. Let $M = X'X$, which is symmetric from part (a), so $M' = M$.
2. Since $X'X$ is invertible, we have:

$$[(X'X)^{-1}]' = (M^{-1})' = (M')^{-1} = (X'X)^{-1}$$

3. Therefore, $[(X'X)^{-1}]' = (X'X)^{-1}$, showing that the inverse of $X'X$ is also symmetric.

Part (c): Show that the Hat Matrix H is Symmetric, $H = H'$

The hat matrix H is defined as $H = X(X'X)^{-1}X'$.

1. Using the property for the transpose of a product of matrices, $(ABC)' = C'B'A'$, we get:

$$H' = \left(X(X'X)^{-1}X'\right)' = X'\left((X'X)^{-1}\right)'X$$

2. Since $(X'X)^{-1}$ is symmetric from part (b), we have $\left((X'X)^{-1}\right)' = (X'X)^{-1}$.
3. Therefore:

$$H' = X(X'X)^{-1}X' = H$$

4. Thus, H is symmetric, i.e., $H = H'$.

Part (d): Show that $HH' = H$

Since we have shown in part (c) that $H = H'$, it follows that:

$$HH' = HH = H$$

This shows that H is idempotent, meaning that applying H twice gives the same result as applying H once.

Part (e): Show that $(I_n - H)'(I_n - H) = I_n - H$

To show this, we expand $(I_n - H)'(I_n - H)$ using the distributive property of the transpose:

$$(I_n - H)'(I_n - H) = I_n' - H' - H + H'H$$

1. Since I_n is the identity matrix, we have $I_n' = I_n$.
2. Also, from part (c), $H' = H$, and from part (d), $HH' = H$.
3. Substituting these results:

$$(I_n - H)'(I_n - H) = I_n - H - H + H = I_n - H$$

Thus, $(I_n - H)'(I_n - H) = I_n - H$, as required.

```
data(teengamb)
model <- lm(gamble ~ sex + status + income + verbal, data = teengamb)
summary(model)
```

```
##
## Call:
## lm(formula = gamble ~ sex + status + income + verbal, data = teengamb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -51.082 -11.320  -1.451   9.452  94.252
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.55565   17.19680   1.312   0.1968
## sex         -22.11833    8.21111  -2.694   0.0101 *
## status        0.05223    0.28111   0.186   0.8535
## income        4.96198    1.02539   4.839 1.79e-05 ***
## verbal       -2.95949    2.17215  -1.362   0.1803
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.69 on 42 degrees of freedom
## Multiple R-squared:  0.5267, Adjusted R-squared:  0.4816
## F-statistic: 11.69 on 4 and 42 DF,  p-value: 1.815e-06
```

```
r_squared <- summary(model)$r.squared
r_squared * 100
```

```
## [1] 52.67234
```

```
residuals <- residuals(model)
max_residual <- max(residuals)
case_number <- which(residuals == max_residual)
case_number
```

```
## 24
## 24
```

```
max_residual
```

```
## [1] 94.25222
```

```
mean_residual <- mean(residuals)
median_residual <- median(residuals)
mean_residual
```

```
## [1] -1.556914e-16
```

```
median_residual
```

```
## [1] -1.451392
```

```
fitted_values <- fitted(model)
cor_residuals_fitted <- cor(residuals, fitted_values)
cor_residuals_fitted
```

```
## [1] -6.215823e-17
```

```r
cor_residuals_income <- cor(residuals, teengamb$income)
cor_residuals_income
```

```
## [1] 3.247058e-17
```

```r
sex_coef <- coef(model)["sex"]
sex_coef
```

```
##       sex
## -22.11833
```