In class bonus exercise Biostat 200C

Apr 16th, 2024

AUTHOR
Hanbei Xiong

Simulate data for a logistic regression model with a quadratic term (e.g., X_1^2) as the true model and check the linearity assumption using the following plots:

We first simulate the data.

```
        library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.0     ✔ tibble    3.2.1
✔ lubridate 1.9.3     ✔ tidyr     1.3.1
✔ purrr     1.0.2
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to
become errors
```

```
        library(scales)
```

```
Attaching package: 'scales'

The following object is masked from 'package:purrr':

    discard

The following object is masked from 'package:readr':

    col_factor
```

```
        set.seed(123)

        # Number of observations
        n <- 1000

        # Coefficients
```

```
beta_0 <- -1
beta_1 <- 0.5
beta_2 <- -0.3

# Simulate X1
X1 <- rnorm(n, 0, 1)

X2 <- X1^2

# Calculate log-odds for Y
log_odds <- beta_0 + beta_1 * X1 + beta_2 * X2

# Convert log-odds to probability using logistic function
p <- 1 / (1 + exp(-log_odds))

# Simulate Y as a binary outcome
Y <- rbinom(n, 1, p)
```

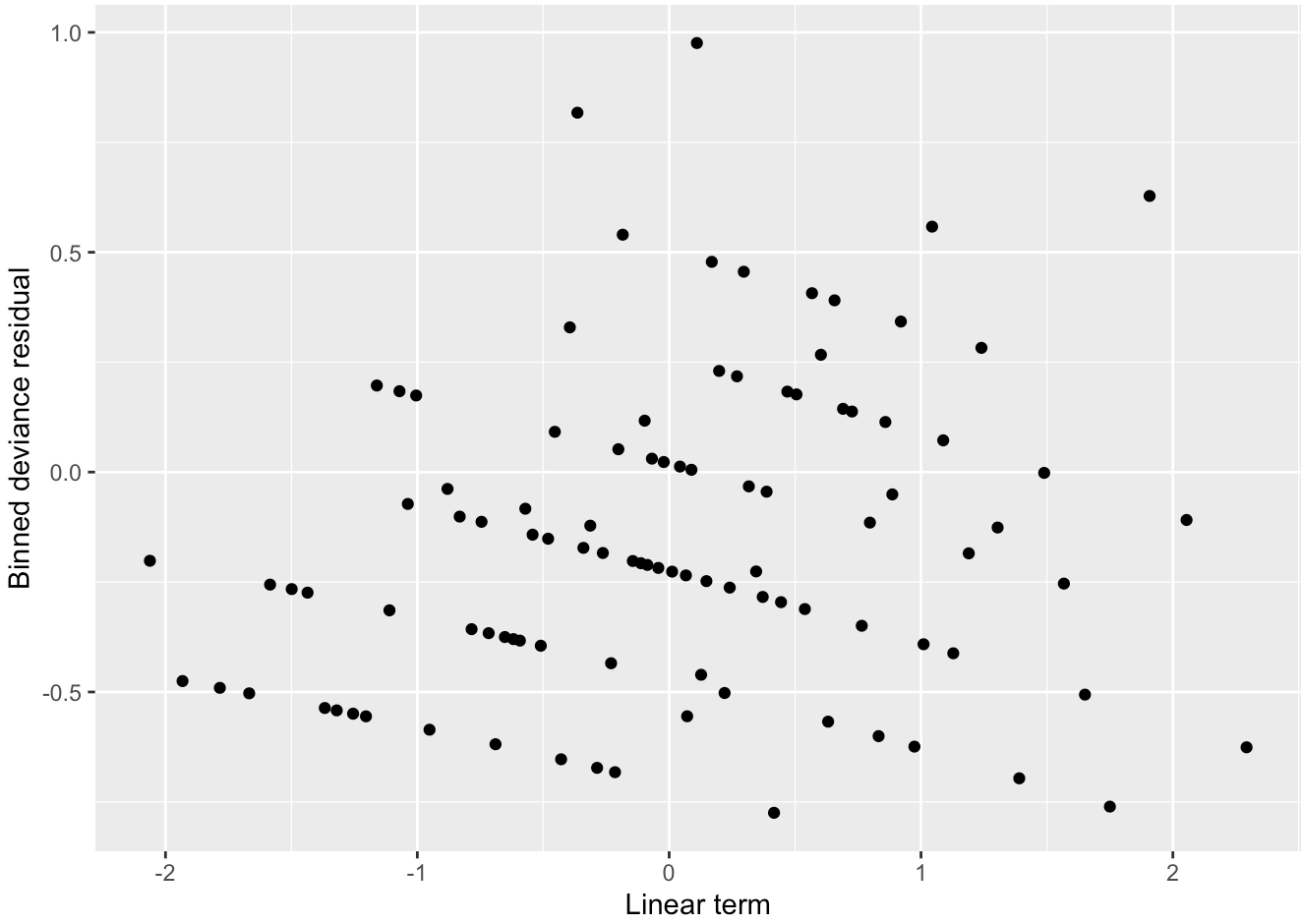
```
df = data.frame(Y = Y, X1 = X1, X2 = X2)
```

- Binned deviance residuals against linear predictor X_1 when you model the systematic component as a linear function of the predictors

```
mod1 <- glm(Y ~ X1, family = binomial(link = "logit"))
```

```
devres1 <- residuals(mod1)
```

```
df %>%
  mutate(devres = devres1) %>%
  group_by(cut(X1, breaks = unique(quantile(X1, (1:100)/101)))) %>%
  summarize(devres = mean(devres),
            X1 = mean(X1)) %>%
  ggplot() +
  geom_point(mapping = aes(x = X1, y = devres)) +
  labs(x = "Linear term", y = "Binned deviance residual")
```

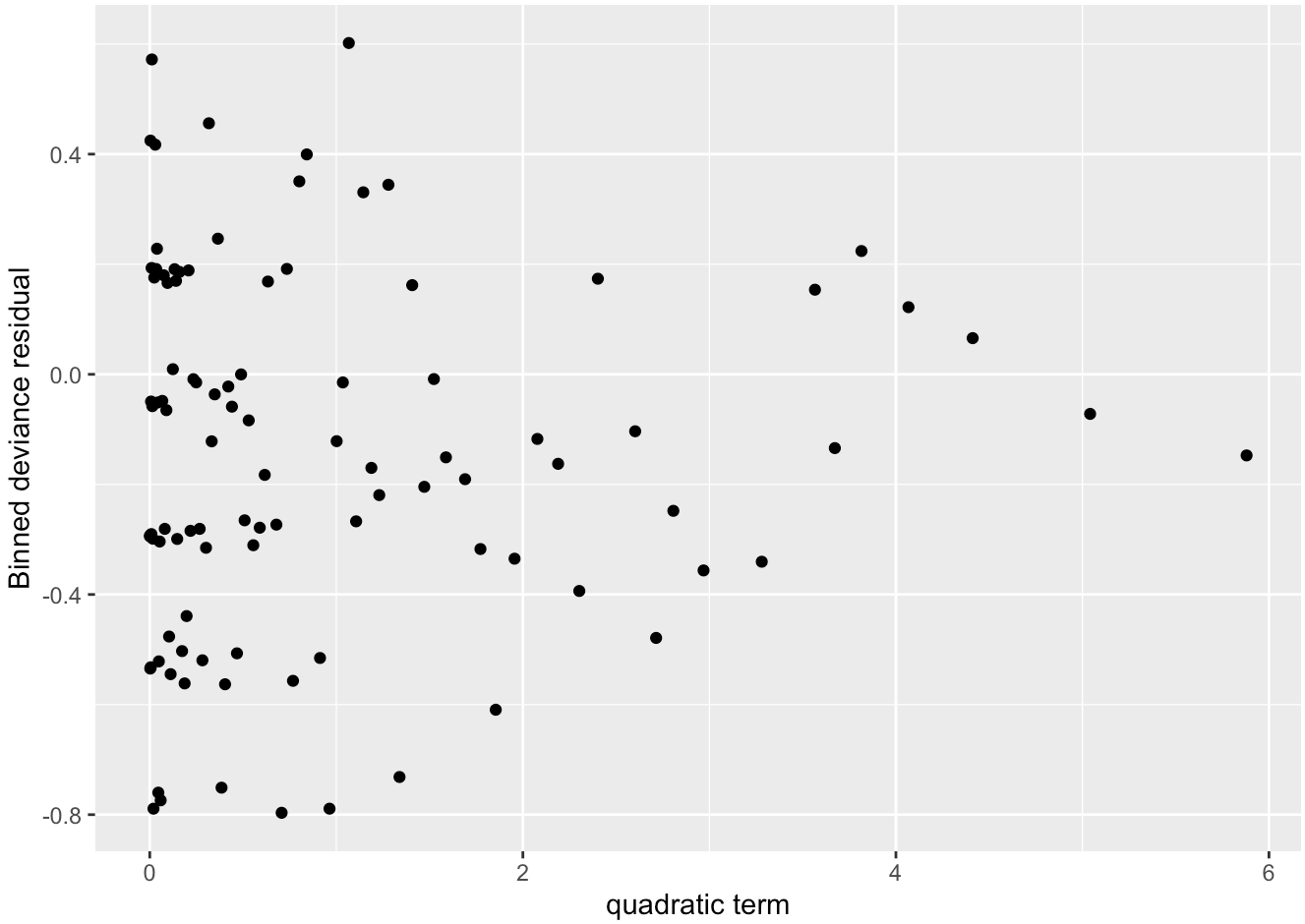


- Binned deviance residuals against the quadratic term X_1^2 when you model the systematic component as a quadratic function of the predictors

```
mod2 <- glm(Y ~ X1 + X2, family = binomial(link = "logit"))
```

```
devres2 <- residuals(mod2)
```

```
df %>%
  mutate(devres = devres2) %>%
  group_by(cut(X2, breaks = unique(quantile(X2, (1:100)/101)))) %>%
  summarize(devres = mean(devres),
            X2 = mean(X2)) %>%
  ggplot() +
  geom_point(mapping = aes(x = X2, y = devres)) +
  labs(x = "quadratic term", y = "Binned deviance residual")
```

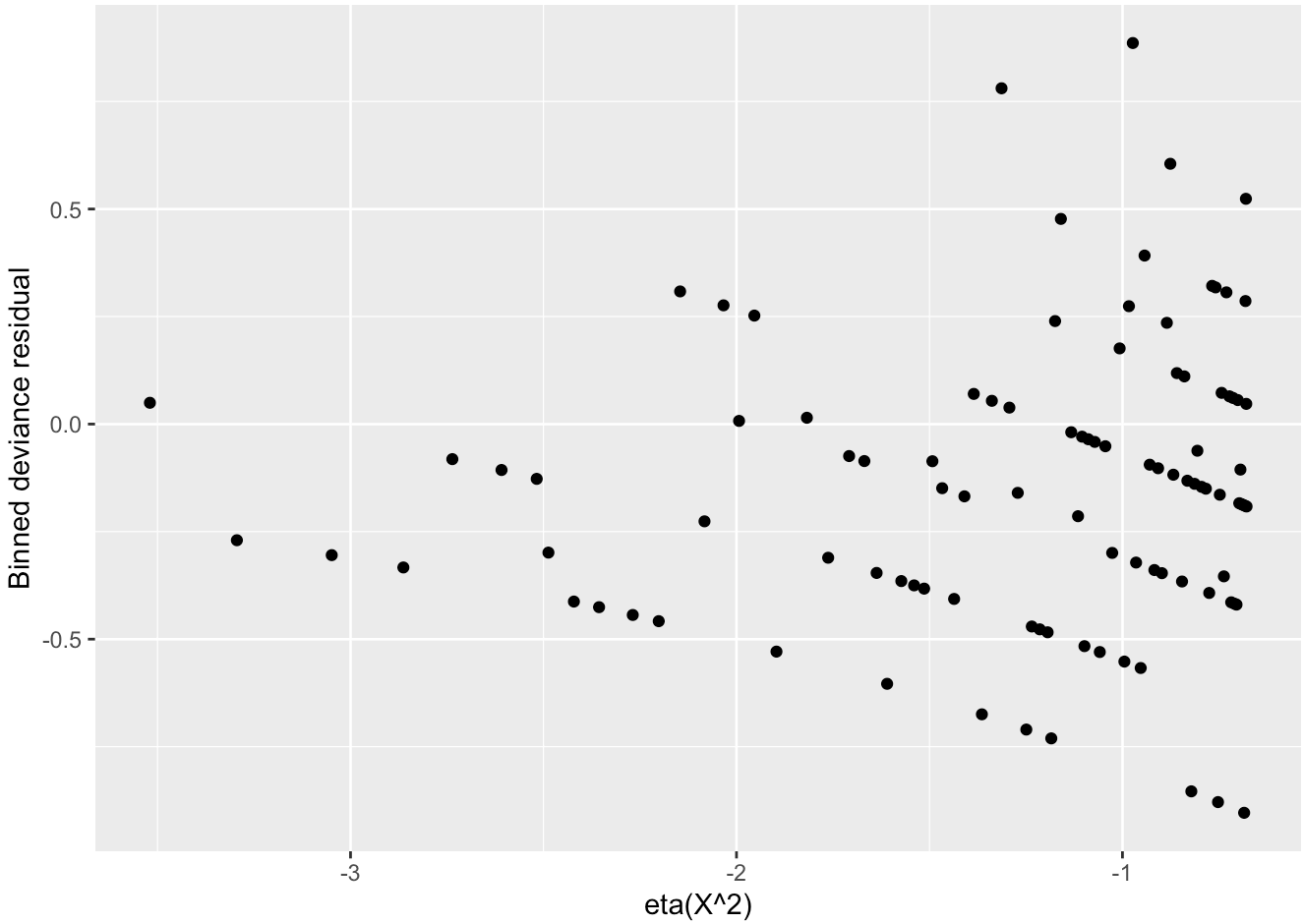


- Binned deviance residuals against fitted value ($\hat{\eta}$) when you model the systematic component as a quadratic function of the predictors

```
mod3 <- glm(Y ~ X1 + X2, family = binomial(link = "logit"))
```

```
devres3 <- residuals(mod3)
linpred3  <- predict(mod3)
```

```
df %>%
  mutate(devres = devres3, linpred = linpred3) %>%
  group_by(cut(linpred, breaks = unique(quantile(linpred, (1:100)/101)))) %>%
  summarize(devres = mean(devres),
            linpred = mean(linpred)) %>%
  ggplot() +
  geom_point(mapping = aes(x = linpred, y = devres)) +
  labs(x = "eta(X^2)", y = "Binned deviance residual")
```

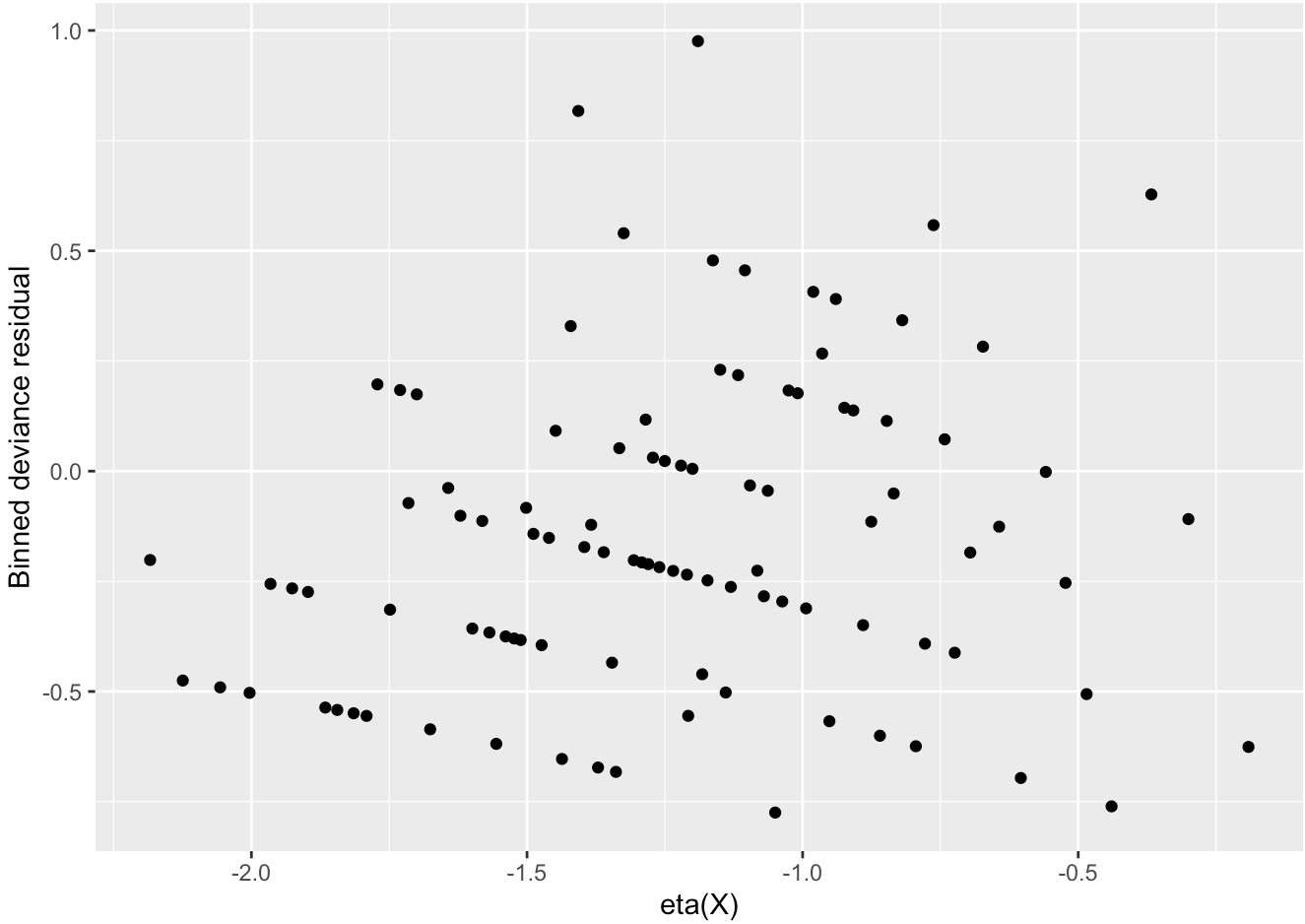


- Binned deviance residuals against fitted value ($\hat{\eta}$) when you model the systematic component as a linear function of the predictors

```
mod4 <- glm(Y ~ X1, family = binomial(link = "logit"))
```

```
devres4 <- residuals(mod4)
linpred4  <- predict(mod4)
```

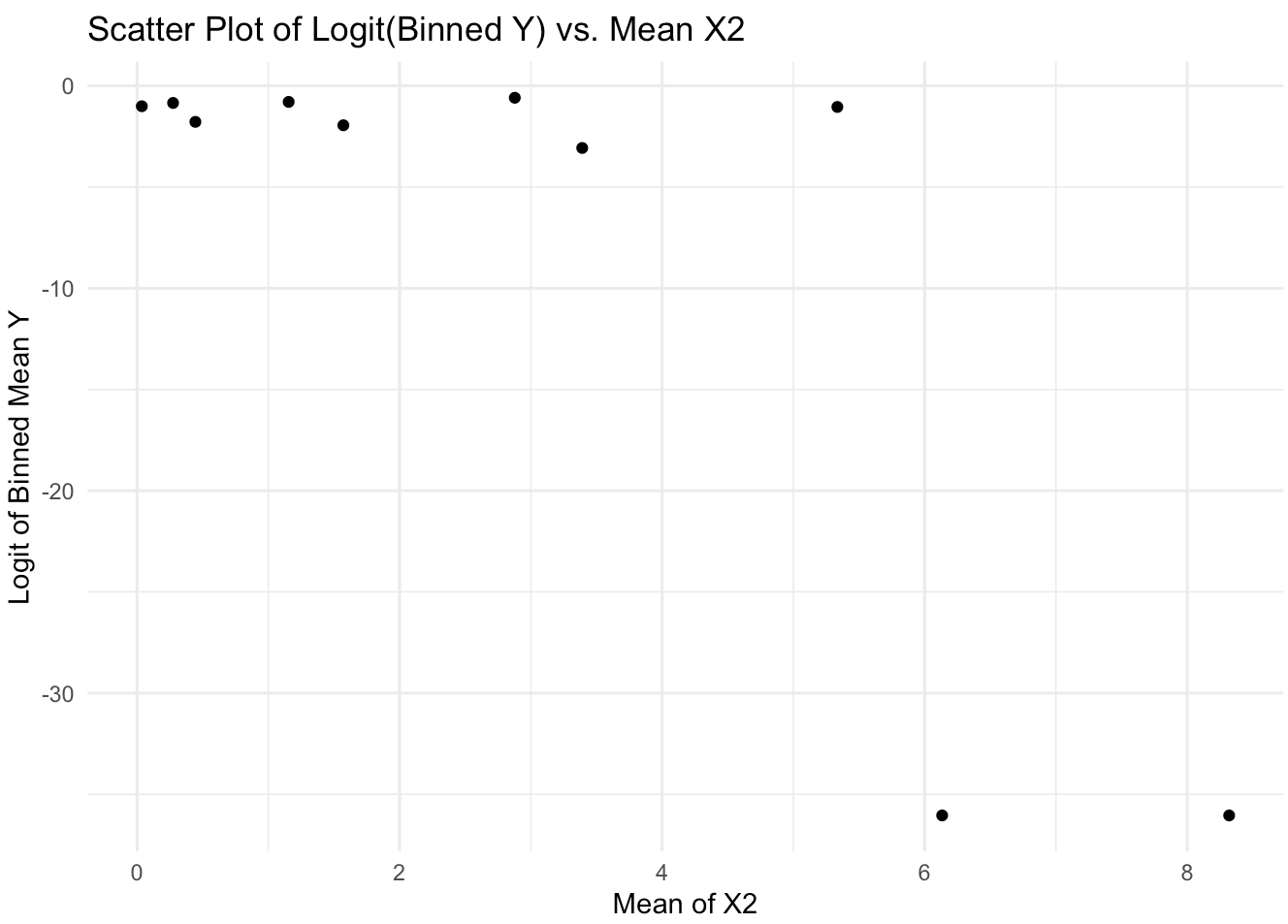
```
df %>%
  mutate(devres = devres4, linpred = linpred4) %>%
  group_by(cut(linpred, breaks = unique(quantile(linpred, (1:100)/101)))) %>%
  summarize(devres = mean(devres),
            linpred = mean(linpred)) %>%
  ggplot() +
  geom_point(mapping = aes(x = linpred, y = devres)) +
  labs(x = "eta(X)", y = "Binned deviance residual")
```



- Scatter plot of logit(binned Y) and X_1^2: break the range of X_1 into bins, and within each bin, calculate the mean value of X_1^2 and Y for observations in that bin. We then transform the mean of Y through the link function

```
df$binned_X1 <- cut(df$X1, breaks = 10)

binned_data <- df %>%
  group_by(binned_X1) %>%
  summarise(mean_Y = mean(Y),
            mean_X2 = mean(X2)) %>%
  ungroup()

binned_data$logit_mean_Y <- log(binned_data$mean_Y / (1 - binned_data$mean_Y))

# Handle cases where mean_Y is 0 or 1
binned_data$logit_mean_Y <- ifelse(binned_data$mean_Y == 0, log(.Machine$double.e
                                   ifelse(binned_data$mean_Y == 1, -log(.Machine$

# Create the scatter plot
ggplot(binned_data, aes(x = mean_X2, y = logit_mean_Y)) +
  geom_point() +
  scale_y_continuous(labels = comma) +
  labs(x = "Mean of X2", y = "Logit of Binned Mean Y") +
```

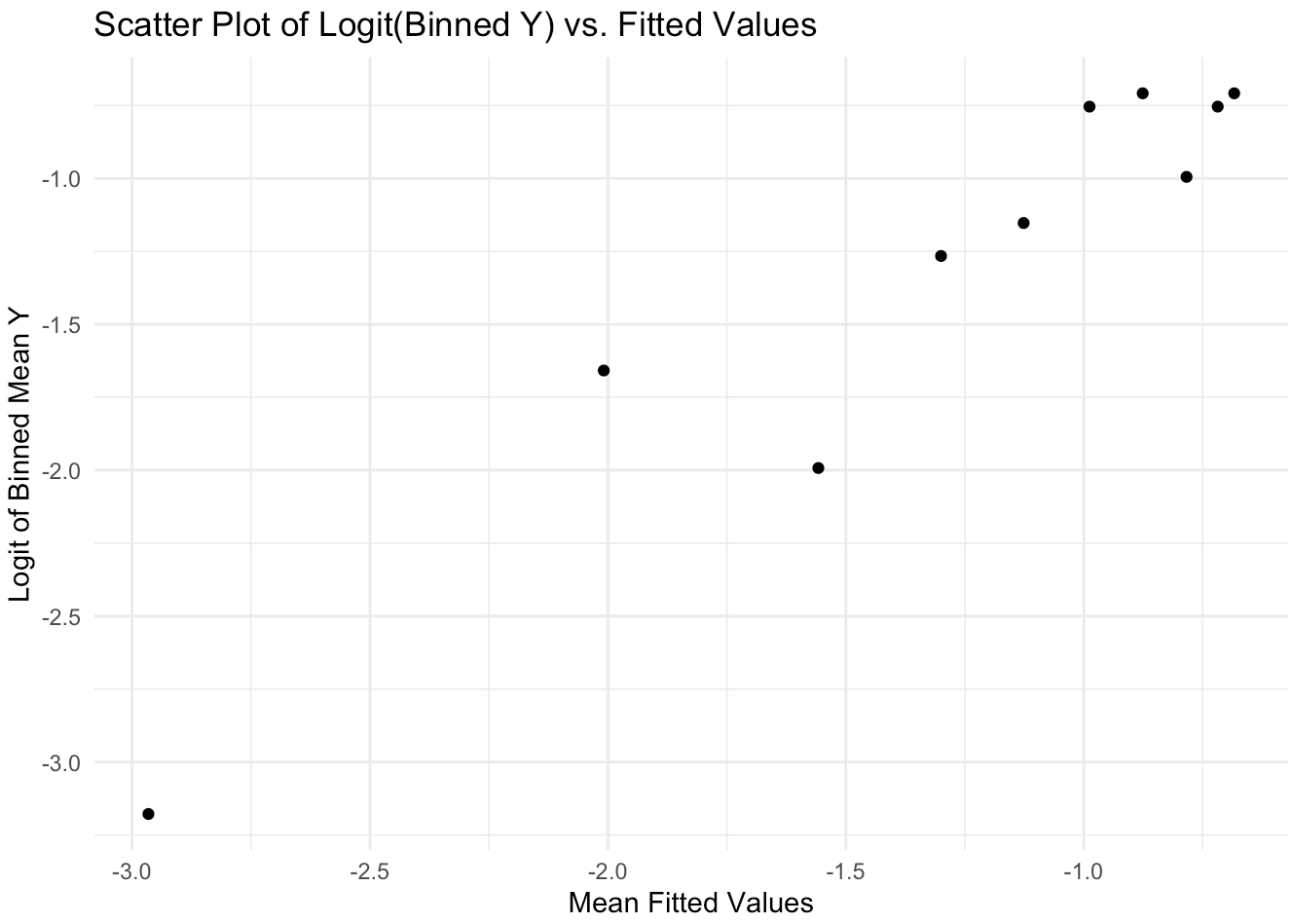
```
  ggtitle("Scatter Plot of Logit(Binned Y) vs. Mean X2") +
  theme_minimal()
```



- Scatter plot of logit(binned Y) and $\hat{\eta}$

```
linpred2 <- predict(mod2, type = "link")

df <- df %>% mutate(fitted_values = linpred2)

# Create bins based on the quantiles of the fitted values
df <- df %>% mutate(bin = cut(fitted_values, breaks = quantile(fitted_values, prc

# Calculate the mean of Y and the mean of fitted values in each bin
binned_stats <- df %>% group_by(bin) %>% summarise(
  mean_Y = mean(Y),
  mean_fitted_values = mean(fitted_values)
)

# Transform mean_Y using the logit function
binned_stats <- binned_stats %>% mutate(logit_mean_Y = log(mean_Y / (1 - mean_Y))

# Handle cases where mean_Y is 0 or 1 to avoid computation errors in logit transf
binned_stats$logit_mean_Y <- ifelse(binned_stats$mean_Y == 0, log(.Machine$double
                                    ifelse(binned_stats$mean_Y == 1, -log(.Machir
```

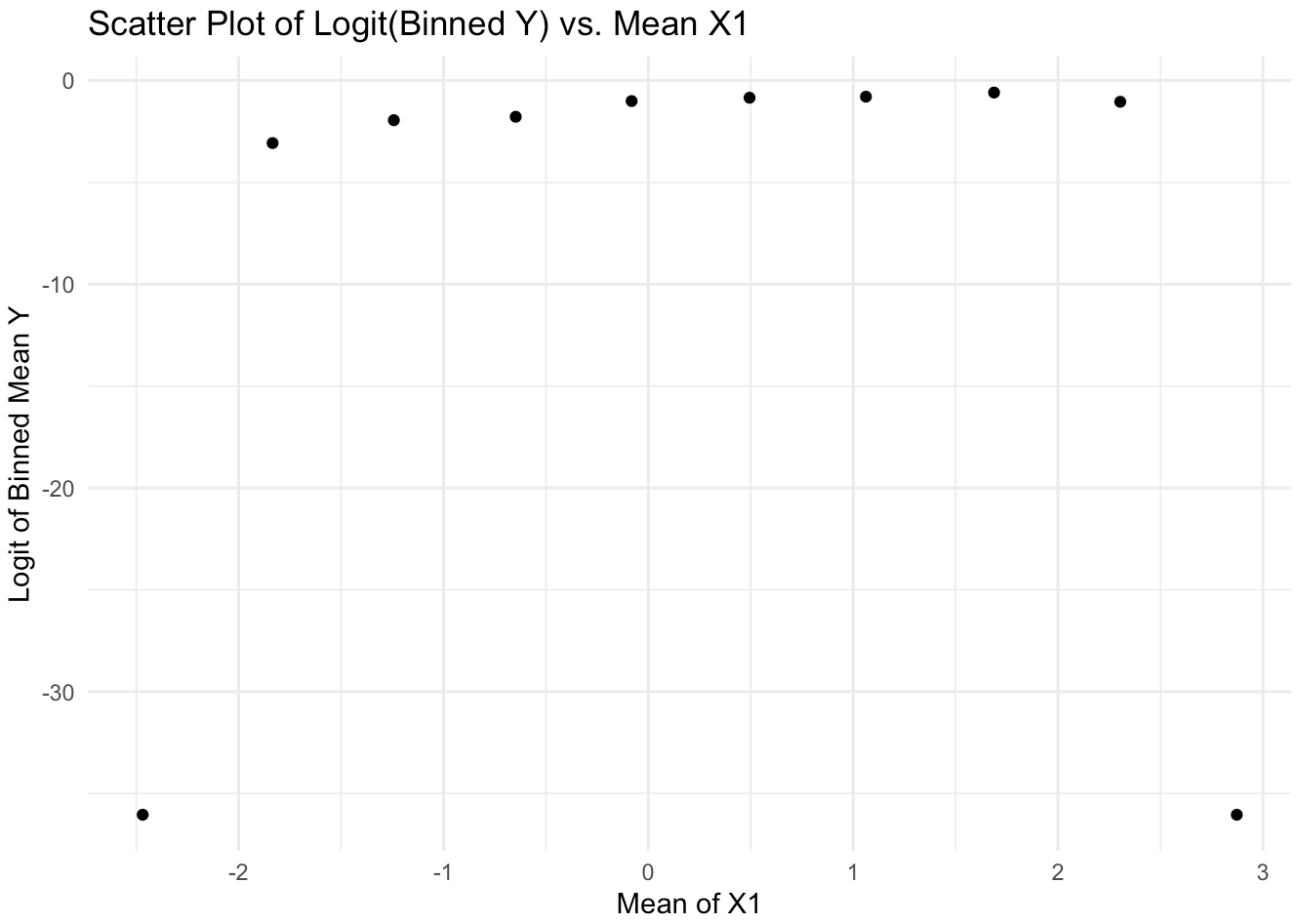
```
ggplot(binned_stats, aes(x = mean_fitted_values, y = logit_mean_Y)) +
  geom_point() +
  labs(x = "Mean Fitted Values", y = "Logit of Binned Mean Y") +
  ggtitle("Scatter Plot of Logit(Binned Y) vs. Fitted Values") +
  theme_minimal()
```



- Scatter plot of logit(binned Y) and X_1

```
binned_data <- df %>%
  group_by(binned_X1) %>%
  summarise(mean_Y = mean(Y),
            mean_X1 = mean(as.numeric(as.character(X1)))) %>%  # Ensure numeric c
  ungroup()

# Transform mean_Y through the logit link function
binned_data$logit_mean_Y <- log(binned_data$mean_Y / (1 - binned_data$mean_Y))

# Handle cases where mean_Y is 0 or 1 to avoid computation errors in logit transf
binned_data$logit_mean_Y <- ifelse(binned_data$mean_Y == 0, log(.Machine$double.e
                                   ifelse(binned_data$mean_Y == 1, -log(.Machine

# Create the scatter plot
ggplot(binned_data, aes(x = mean_X1, y = logit_mean_Y)) +
  geom_point() +
```

```
  scale_y_continuous(labels = comma) +
  labs(x = "Mean of X1", y = "Logit of Binned Mean Y") +
  ggtitle("Scatter Plot of Logit(Binned Y) vs. Mean X1") +
  theme_minimal()
```



Conclusion: By observing the last three plots, I think linearity assumption is met since majority of points center around 0 with only a few outlier far away from 0. However, the deviance vs fitted value plot shows a systematic monotonic patter which should raise awareness.