SDS 4392 HW2

Washington University, SP 2025

Aidan Kardan SDS 4392 HW 2

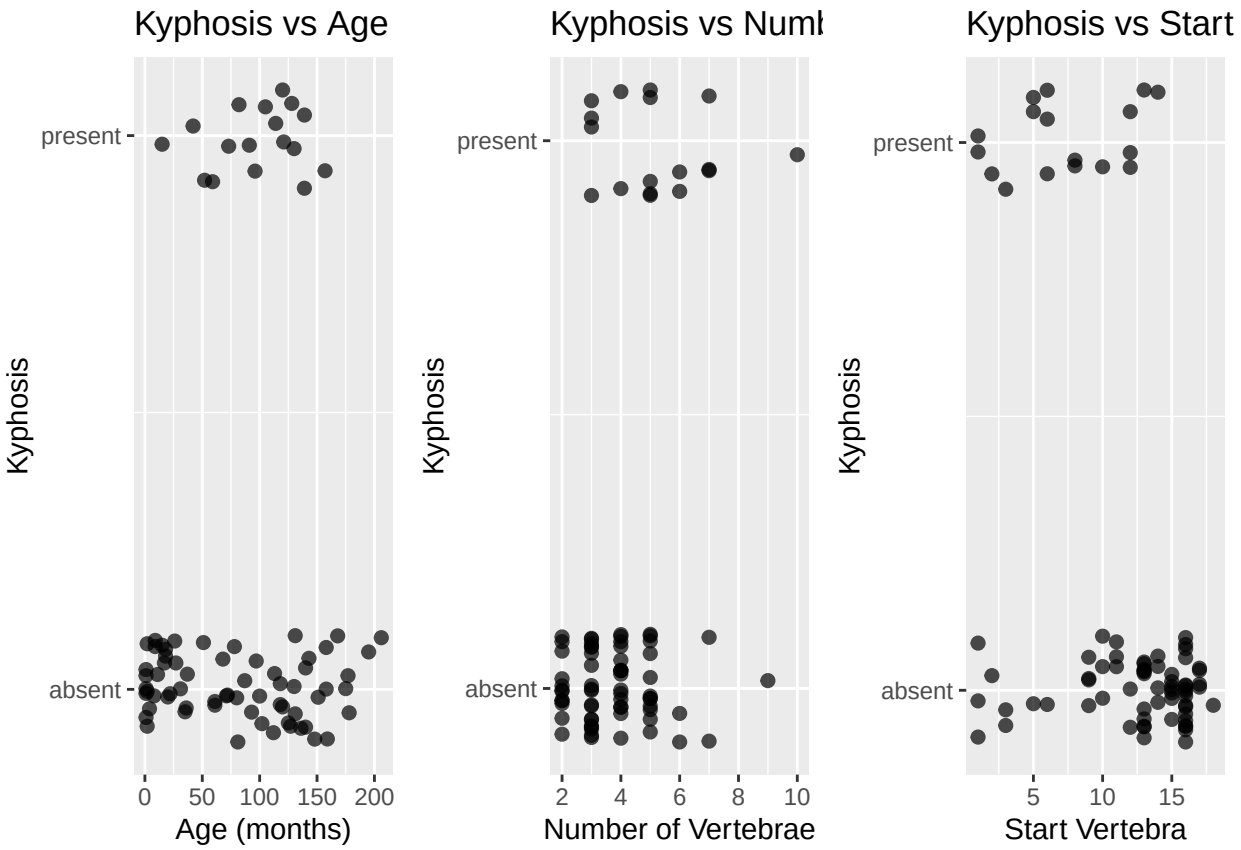
1.[5 pts] A study was conducted on children who had corrective spinal surgery. We are interested in factors that might result in kyphosis (a kind of deformation) after surgery. The data can be loaded by `data(kyphosis,package="rpart")`

```
data(kyphosis, package = "rpart")

# For plotting purposes, create a numeric version of the Kyphosis response:
# Map "absent" to 0 and "present" to 1
kyphosis$kyphosis_num <- ifelse(kyphosis$Kyphosis == 'present', 1, 0)
head(kyphosis)
```

```
##   Kyphosis Age Number Start kyphosis_num
## 1   absent  71      3     5            0
## 2   absent 158      3    14            0
## 3  present 128      4     5            1
## 4   absent   2      5     1            0
## 5   absent   1      4    15            0
## 6   absent   1      2    16            0
```

a) Make plots of the response as it relates to each of the three predictors. You may find a jittered scatterplot more effective than the interleaved histogram for a dataset of this size. Comment on how the predictors appear to be related to the response.

```
set.seed(123)
# Plot for Age
p1 <- ggplot(kyphosis, aes(x = Age, y = kyphosis_num)) +
  geom_jitter(width = 0, height = 0.1, size = 2, alpha = 0.7) +
  scale_y_continuous(breaks = c(0, 1), labels = c("absent", "present")) +
  labs(title = "Kyphosis vs Age", x = "Age (months)", y = "Kyphosis")

# Plot for Number
p2 <- ggplot(kyphosis, aes(x = Number, y = kyphosis_num)) +
  geom_jitter(width = 0, height = 0.1, size = 2, alpha = 0.7) +
  scale_y_continuous(breaks = c(0, 1), labels = c("absent", "present")) +
  labs(title = "Kyphosis vs Number", x = "Number of Vertebrae", y = "Kyphosis")

# Plot for Start
p3 <- ggplot(kyphosis, aes(x = Start, y = kyphosis_num)) +
  geom_jitter(width = 0, height = 0.1, size = 2, alpha = 0.7) +
  scale_y_continuous(breaks = c(0, 1), labels = c("absent", "present")) +
  labs(title = "Kyphosis vs Start", x = "Start Vertebra", y = "Kyphosis")
# Arrange the plots side-by-side
grid.arrange(p1, p2, p3, ncol = 3)
```



Age: It seems that points labeled present appear across all ages, no clear cutoff where kyphosis becomes more likely. Age alone does not look like it is strongly related to the response, kyphosis. However,

Number: Most observations are clustered around 3-6, both absent and present cases appear in this range. There is no obvious separation suggesting that Number itself is a dominant predictor of kyphosis.

Start: This variable shows a more noticeable pattern: lower "Start" values appear more frequently with "present" kyphosis, whereas higher "Start" values (above 10) are mostly "absent". This suggests "Start" may be the strongest predictor among the three for distinguishing kyphosis status.

Overall, from these initial visualizations, Start seems to have the clearest relationship to kyphosis, while Age and Number do not show as strong a pattern when viewed alone.

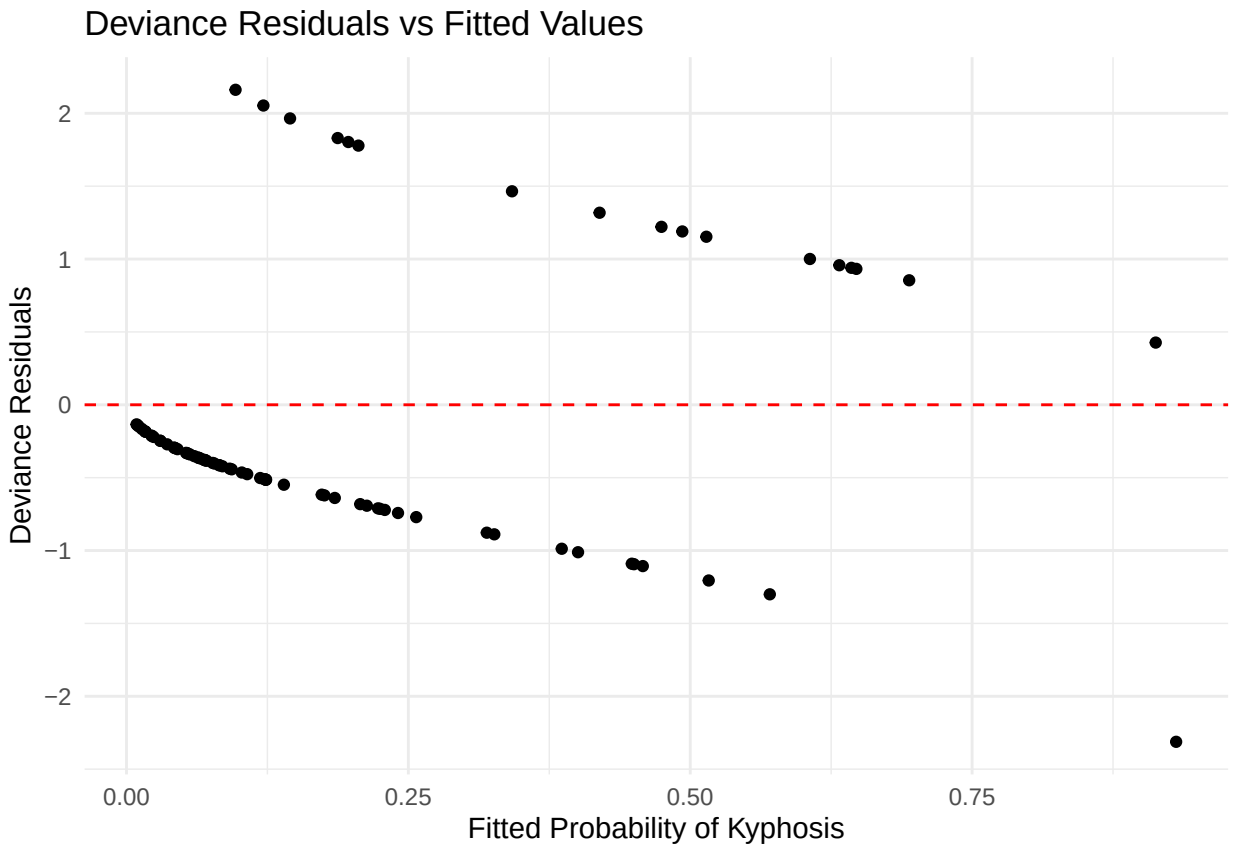
b) Fit a GLM with the kyphosis indicator as the response and the other three variables as predictors. Plot the deviance residuals against the fitted values. What can be concluded from this plot?

```
# Logistic regression model using the binary response
model1 <- glm(kyphosis_num ~ Age + Number + Start, data = kyphosis, family = binomial)
summary(model1)
```

```
##
## Call:
## glm(formula = kyphosis_num ~ Age + Number + Start, family = binomial,
##     data = kyphosis)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.036934   1.449575  -1.405  0.15996
## Age          0.010930   0.006446   1.696  0.08996 .
## Number       0.410601   0.224861   1.826  0.06785 .
```

```
## Start       -0.206510   0.067699  -3.050  0.00229 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83.234  on 80  degrees of freedom
## Residual deviance: 61.380  on 77  degrees of freedom
## AIC: 69.38
##
## Number of Fisher Scoring iterations: 5
```

```r
# Extract deviance residuals and fitted probabilities
dev_res     <- resid(model1, type = "deviance")
fitted_vals <- fitted(model1)

ggplot(kyphosis, aes(x = fitted_vals, y = dev_res)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(x = "Fitted Probability of Kyphosis",
       y = "Deviance Residuals",
       title = "Deviance Residuals vs Fitted Values") +
  theme_minimal()
```



Overall, the residuals do not show a clear systematic departure from zero—there's no strong curvature or funnel shape that would suggest a major model misspecification. While you can see a general slope (with negative residuals increasing as the fitted probability rises), that pattern is typical in logistic regression when the model underestimates some outcomes at low predicted probabilities and overestimates others at higher

probabilities. A few points do stand out as outliers with larger positive or negative residuals, but there's no evidence of a pervasive structural problem. In short, this plot does not indicate a serious lack of fit, though a few high-leverage points or additional predictors might be worth exploring.

c) Produce a binned residual plot as described in the text. You will need to select an appropriate amount of binning. Comment on the plot.

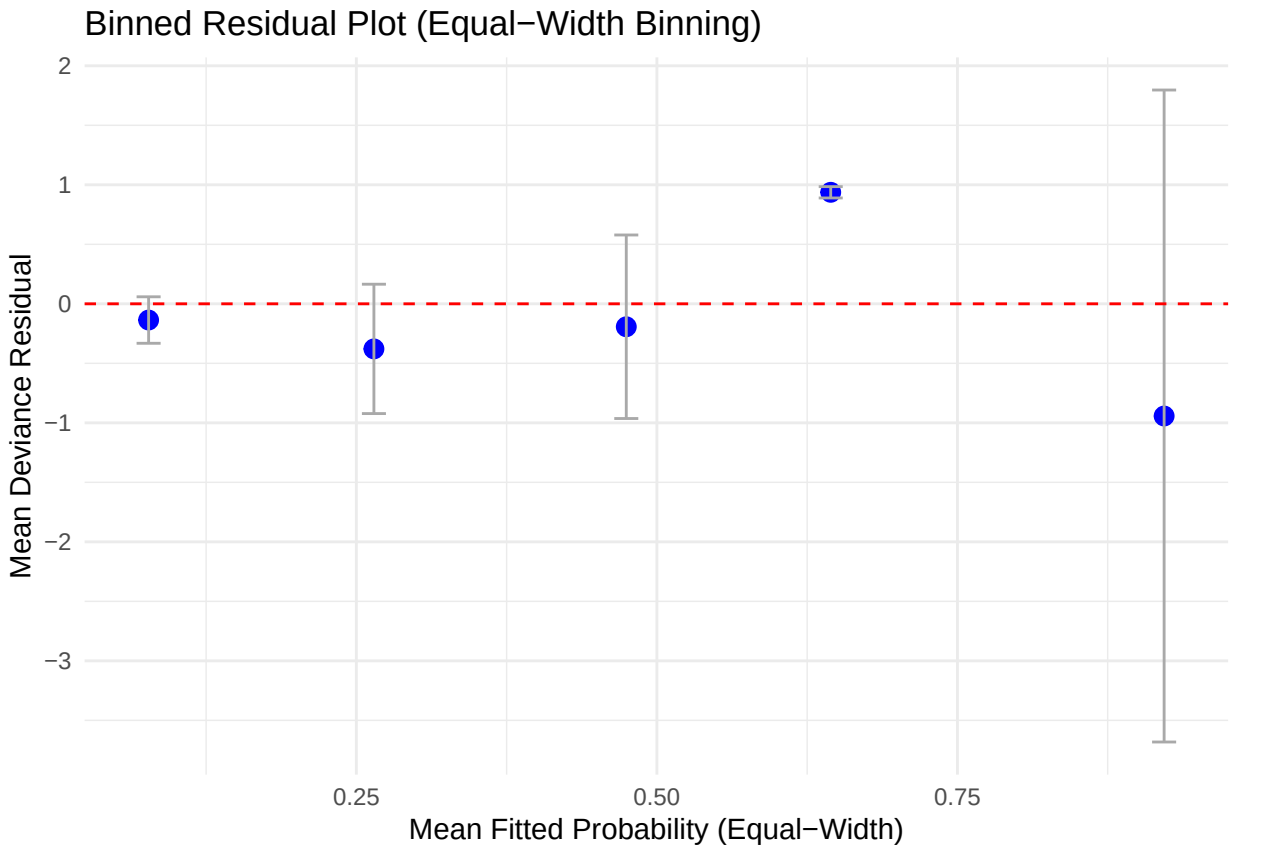
```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following object is masked from 'package:gridExtra':
##
##     combine
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

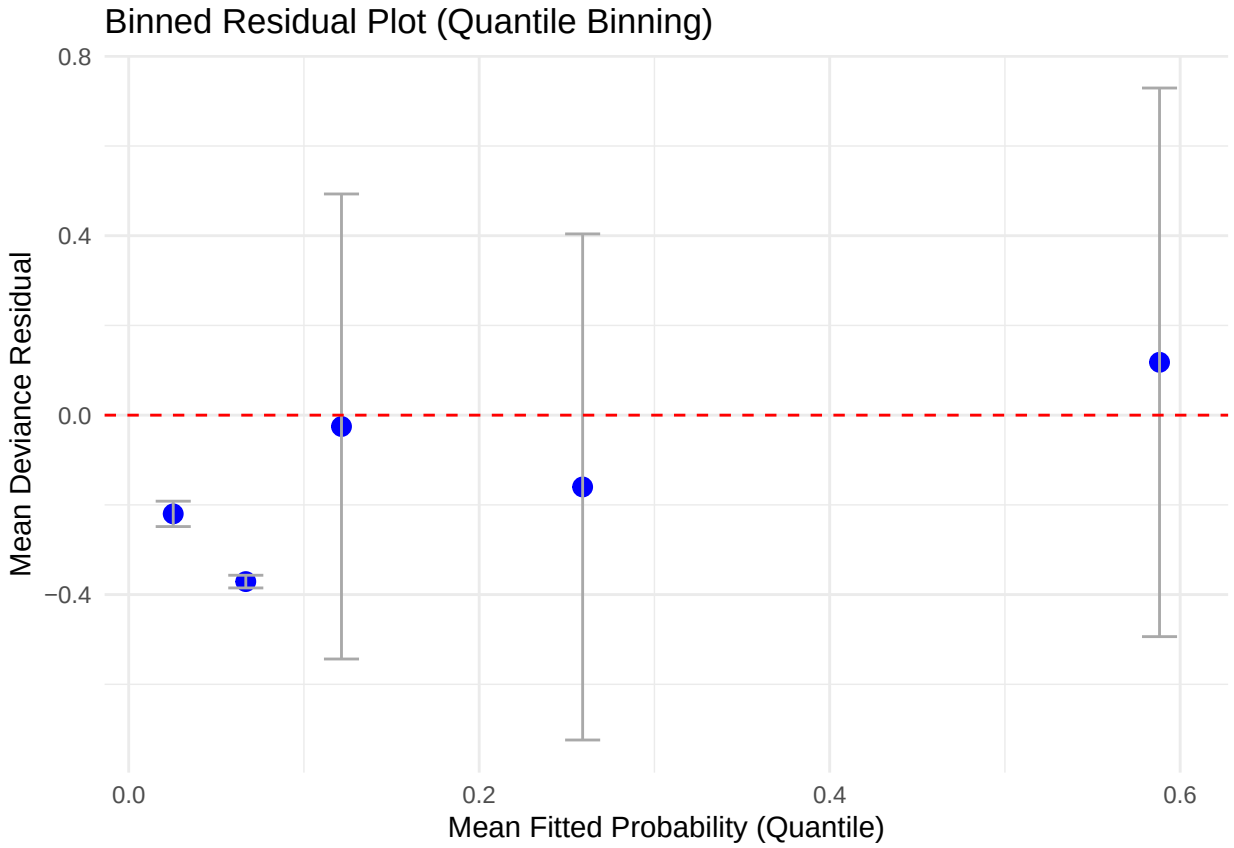
```
# Create data frame
df <- kyphosis %>%
  mutate(
    fitted  = fitted(model1),
    dev_res = resid(model1, type = "deviance")
  )

n_bins <- 5

# Create both types of bins in the same data frame:
df <- df %>%
  mutate(
    # Equal-width binning: divide the range [0,1] into n_bins equal intervals
    equal_bin = cut(fitted,
                    breaks = seq(0, 1, length.out = n_bins + 1),
                    include.lowest = TRUE),
    # Quantile-based binning: divide the data into bins with roughly equal counts
    quantile_bin = cut(fitted,
                       breaks = quantile(fitted, probs = seq(0, 1, length.out = n_bins + 1)),
                       include.lowest = TRUE)
  )

# Summarize the binned data for Equal-Width Binning
binned_equal <- df %>%
  group_by(equal_bin) %>%
  summarize(
    mean_fitted = mean(fitted),
    mean_res    = mean(dev_res),
    se_res      = sd(dev_res) / sqrt(n())
  )

# Summarize the binned data for Quantile-Based Binning
binned_quantile <- df %>%
```

```
  group_by(quantile_bin) %>%
  summarize(
    mean_fitted = mean(fitted),
    mean_res    = mean(dev_res),
    se_res      = sd(dev_res) / sqrt(n())
  )
```

```
# Plot for Equal-Width Binning
plot_equal <- ggplot(binned_equal, aes(x = mean_fitted, y = mean_res)) +
  geom_point(size = 3, color = "blue") +
  geom_errorbar(aes(ymin = mean_res - 2 * se_res, ymax = mean_res + 2 * se_res),
                width = 0.02, color = "darkgray") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(x = "Mean Fitted Probability (Equal-Width)",
       y = "Mean Deviance Residual",
       title = "Binned Residual Plot (Equal-Width Binning)") +
  theme_minimal()

# Plot for Quantile-Based Binning
plot_quantile <- ggplot(binned_quantile, aes(x = mean_fitted, y = mean_res)) +
  geom_point(size = 3, color = "blue") +
  geom_errorbar(aes(ymin = mean_res - 2 * se_res, ymax = mean_res + 2 * se_res),
                width = 0.02, color = "darkgray") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(x = "Mean Fitted Probability (Quantile)",
       y = "Mean Deviance Residual",
       title = "Binned Residual Plot (Quantile Binning)") +
  theme_minimal()

# Display the plots
plot_equal
```




```
plot_quantile
```



I chose to explore quantile binning and equal-width binning, for the bin length, I explored various lengths and concluded that 5 bins provides a good understanding of all residuals. I chose 5 bins because that allows for at least 16 observations in each bin. We typically want more than 30 observations in each bin, but this will take away from more detailed residual understanding across different probabilities for such a small dataset.

Interpretation

Good Fit at Low/Mid Ranges: The bins covering lower to mid-range fitted probabilities have mean residuals close to zero, and their error bars typically cross the zero line, indicating no consistent under- or over-prediction there.

Issues at High Probabilities: One or two bins at the higher end exhibit substantially negative mean residuals and wide confidence intervals.

This typically means:

Fewer Observations fall in that high-probability range, so each observation wields greater influence.

The model might be overestimating kyphosis in those few cases (leading to negative deviance residuals when the true outcome is 0 or the predicted probability is too high).

Investigate Outliers or Leverage Points: It would be worth identifying the specific cases in the high-probability range to see if they have unique characteristics or if the model simply can't capture their variability with the existing predictors.

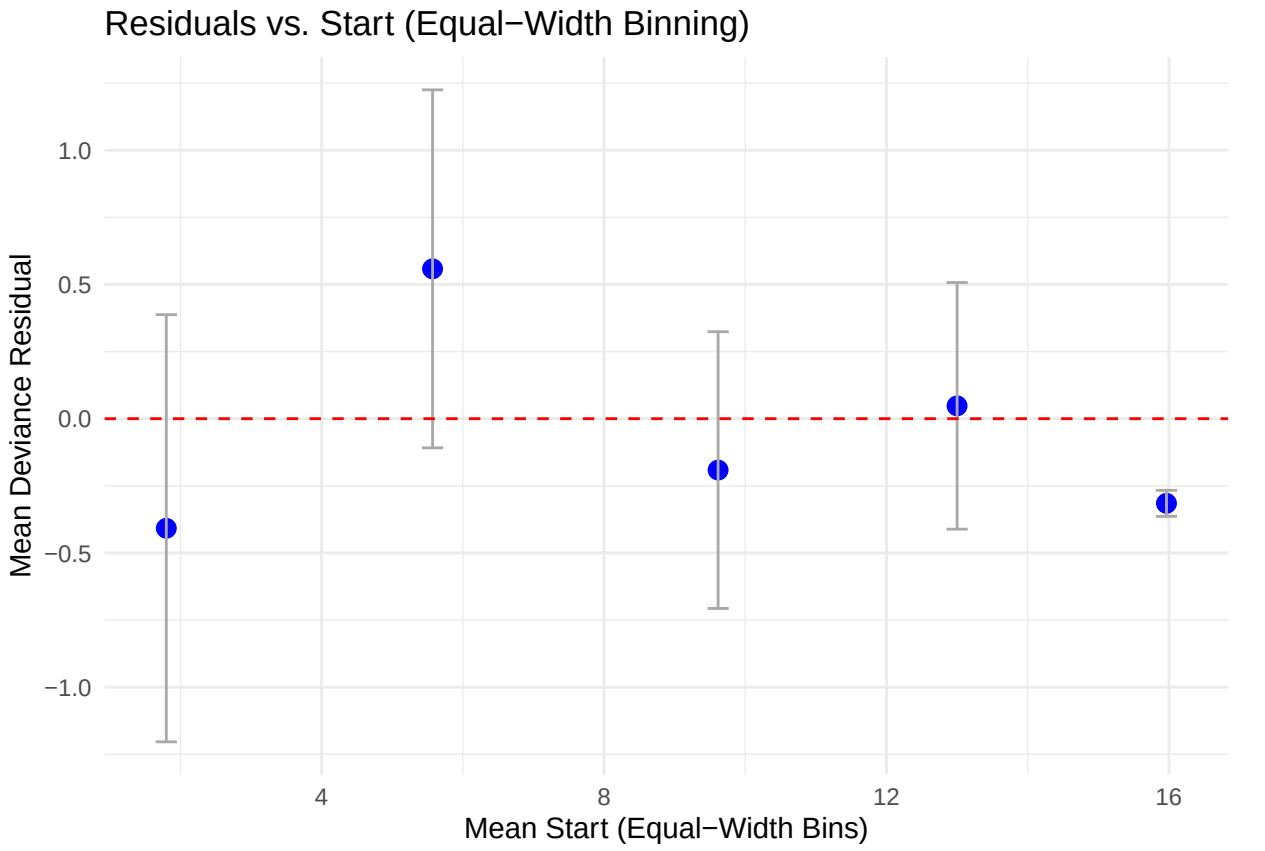
d) Plot the residuals against the Start predictor, using binning as appropriate. Comment on the plot.

```
library(dplyr)


# Create a data frame with 'Start' and deviance residuals
```

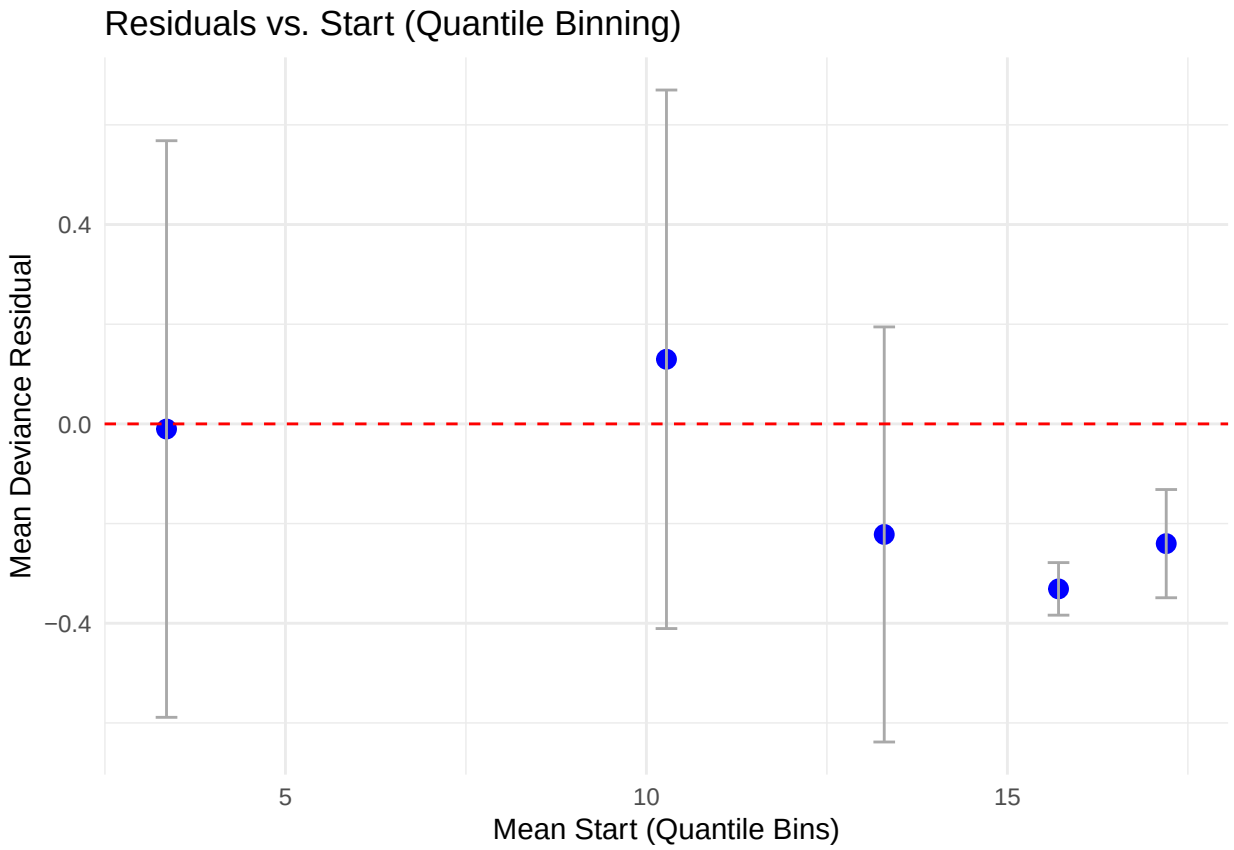
```r
resid_data <- data.frame(
  Start   = kyphosis$Start,
  dev_res = resid(model1, type = "deviance")
)


n_bins <- 5

# Equal-Width Binning
start_min <- min(resid_data$Start)
start_max <- max(resid_data$Start)

resid_data_ew <- resid_data %>%
  mutate(
    start_bin_ew = cut(
      Start,
      breaks = seq(start_min, start_max, length.out = n_bins + 1),
      include.lowest = TRUE
    )
  )

binned_ew <- resid_data_ew %>%
  group_by(start_bin_ew) %>%
  summarize(
    mean_start = mean(Start),
    mean_res   = mean(dev_res),
    se_res     = sd(dev_res) / sqrt(n())
  )
# Quantile-Based Binning
resid_data_q <- resid_data %>%
  mutate(
    start_bin_q = cut(
      Start,
      breaks = quantile(Start, probs = seq(0, 1, length.out = n_bins + 1)),
      include.lowest = TRUE
    )
  )

binned_q <- resid_data_q %>%
  group_by(start_bin_q) %>%
  summarize(
    mean_start = mean(Start),
    mean_res   = mean(dev_res),
    se_res     = sd(dev_res) / sqrt(n())
  )
```

```r
plot_ew <- ggplot(binned_ew, aes(x = mean_start, y = mean_res)) +
  geom_point(size = 3, color = "blue") +
  geom_errorbar(aes(ymin = mean_res - 2*se_res, ymax = mean_res + 2*se_res),
                width = 0.3, color = "darkgray") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(x = "Mean Start (Equal-Width Bins)",
       y = "Mean Deviance Residual",
       title = "Residuals vs. Start (Equal-Width Binning)") +
```

```
  theme_minimal()


plot_q <- ggplot(binned_q, aes(x = mean_start, y = mean_res)) +
  geom_point(size = 3, color = "blue") +
  geom_errorbar(aes(ymin = mean_res - 2*se_res, ymax = mean_res + 2*se_res),
                width = 0.3, color = "darkgray") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(x = "Mean Start (Quantile Bins)",
       y = "Mean Deviance Residual",
       title = "Residuals vs. Start (Quantile Binning)") +
  theme_minimal()

# Print both plots
plot_ew
```




```
plot_q
```



Equal-Width Binning

Bin Near Start 2–3: Mean residual is slightly below zero with a moderate negative error bar. This suggests the model may slightly overpredict kyphosis in that low Start range (residual < 0).

Bin Near Start 4–5: Shows a clearly positive mean residual (above 0), indicating some underprediction in that bin on average.

Bin Near Start 8: The mean residual again dips below zero, flipping from the prior bin's positive residual.

Bins Near Start 12 and 16: Hover closer to or slightly below zero, with moderately wide error bars but no strong signal of severe under- or overprediction.

From these ups and downs, there's no consistent pattern—the model oscillates around zero rather than steadily trending above or below it.

Quantile Binning

First Bin (Low Start Values): Slightly negative mean residual, indicating mild overprediction of kyphosis for these lowest Start values. The error bar is wide, likely reflecting a small sample in that bin or some outliers.

Second Bin (Mid-Low Start): Mean residual is near zero but with a tall error bar, again suggesting variability among these observations.

Middle Bin (Near Start 10): Positive mean residual with a very large error bar, pointing to potential outliers or relatively few data points in that range.

Higher Start Bins (13–16+): Mostly below zero, but again the negative offset is not extreme, and the error bars span zero for the last bin.

No Major Misspecification: Neither plot reveals a consistent upward or downward drift in mean residuals that would imply a missing non-linear term or strong misfit tied to Start.

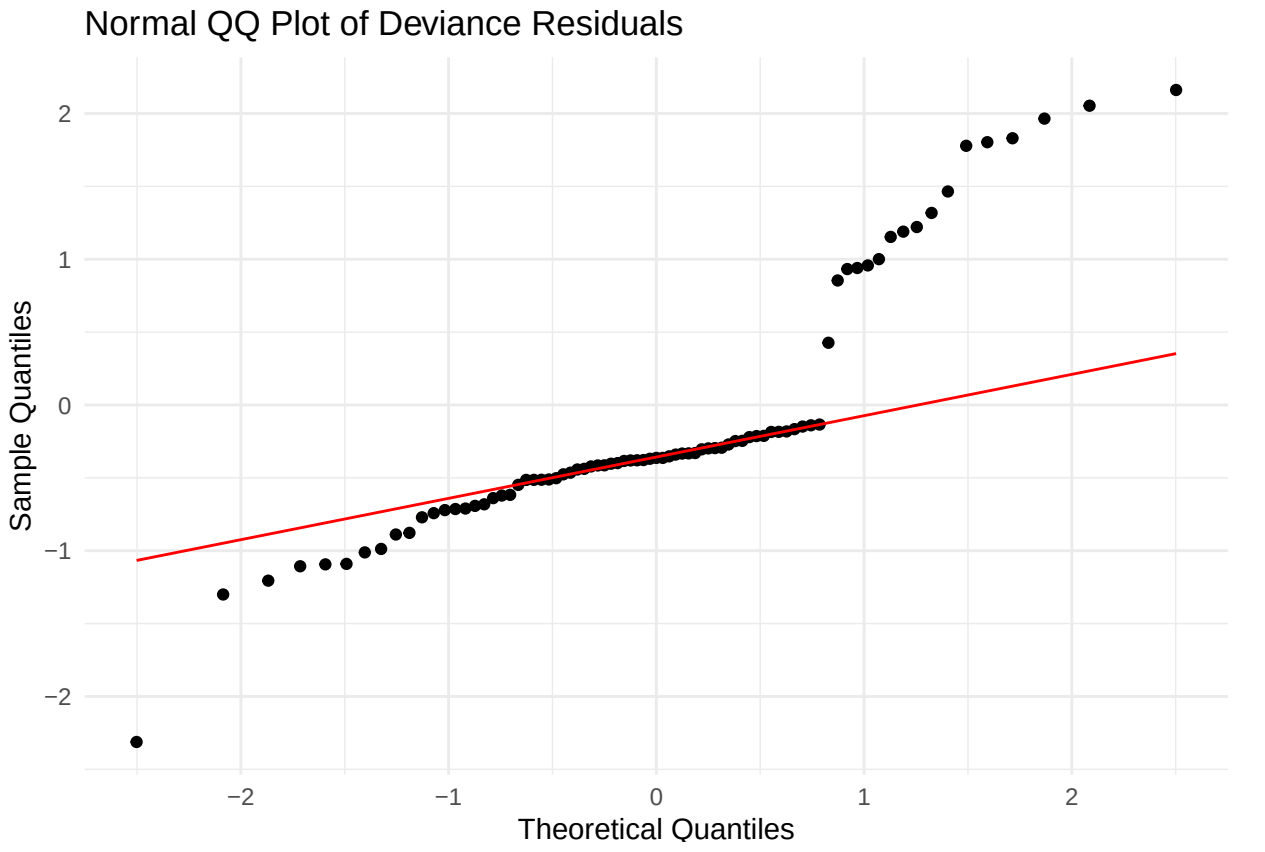
Binning Differences: Equal-width binning presents fixed Start intervals, leading to some bins with more points than others. Quantile binning balances the bin sizes but can produce less intuitive intervals.

Mild Fluctuations / Outliers: A few bins have wider error bars or deviate from zero, possibly due to small sample sizes in those intervals or outlier observations.

Given just 81 observations, 5 bins seems like a reasonable balance—enough granularity without sacrificing too much stability. Overall, these residual plots do not point to a major shortfall of the current model regarding the linear effect of Start.

e) Produce a normal QQ plot for the residuals. Interpret the plot.

```
qq_data <- data.frame(sample = dev_res)

ggplot(qq_data, aes(sample = sample)) + stat_qq () + stat_qq_line(color="red") +
  labs(title = "Normal QQ Plot of Deviance Residuals",
       x = "Theoretical Quantiles",
       y = "Sample Quantiles") +
  theme_minimal()
```



Central Fit:

Around the middle (-1 to 1 of the theoretical axis), points closely follow the diagonal, suggesting that for most observations the deviance residuals are not drastically off from a normal-like pattern.

Heavier Right Tail:

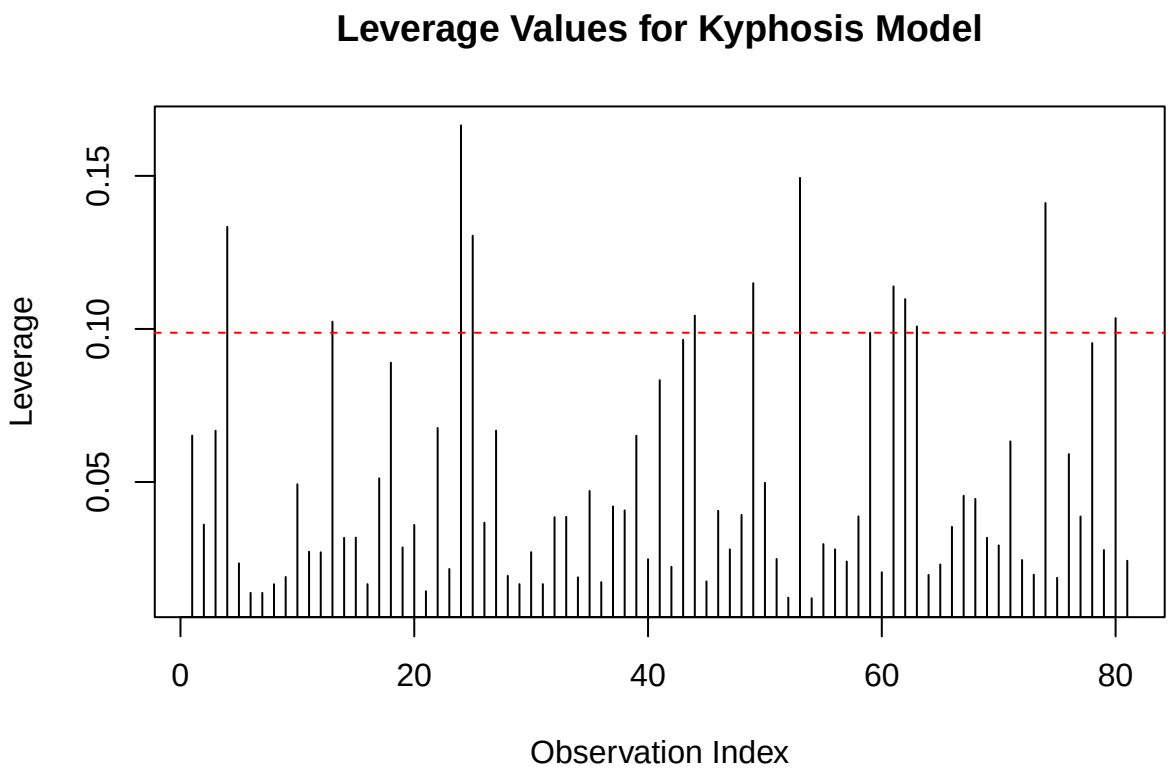
Many points deviate above the line once theoretical quantiles exceed 1, indicating a heavier-than-normal positive tail. In a logistic context, these are cases where the outcome is "present" yet the model assigns relatively low probabilities, leading to large positive residuals (i.e., under prediction).

Implication:

While logistic deviance residuals aren't strictly expected to be normal, this pronounced right-tail departure highlights a subset of observations the model consistently misfits. These likely warrant further scrutiny for potential outliers, additional predictors, or model adjustments.

f) Make a plot of the leverages. Interpret the plot.

```r
# Extract leverage (hat) values
lev <- hatvalues(model1)

p <- 4
n <- nrow(kyphosis)  # total observations

# A common rule of thumb for "high" leverage is 2*(p/n)
threshold <- 2 * (p / n)

# Plot leverage values
plot(lev,
     type = "h",
     main = "Leverage Values for Kyphosis Model",
     xlab = "Observation Index",
     ylab = "Leverage")
abline(h = threshold, col = "red", lty = 2)
```



Here, around 13 observations exceed the threshold. These cases aren't automatically outliers but this means that 13 observations are exerting a stronger pull on the fitted model parameters, because the model attempts to fit these unusual observations as well as the bulk of the data. However, having multiple leverage points can sometimes be less problematic than having a single extremely high leverage observation.

The next steps would be to examine these points individually, and assess whether the 13 points are also high-residual points.

g) Check the goodness of fit for this model. Create a plot like Figure 2.9. Compute the Hosmer-Lemeshow statistic and associated p-value. What do you conclude?

```
library(generalhoslem)
```

```
## Loading required package: reshape
```

```
##
## Attaching package: 'reshape'
```

```
## The following object is masked from 'package:dplyr':
##
##     rename
```

```
## The following object is masked from 'package:cowplot':
##
##     stamp
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
# Extract the predicted probabilities for 'present'
p_hat <- predict(model1, type = "response")

# Perform goodness-of-fit test with 3 bins
hl_results <- logitgof(obs = kyphosis$kyphosis_num, exp = p_hat, g = 3)

hl_results
```

```
##
##  Hosmer and Lemeshow test (binary model)
##
## data:  kyphosis$kyphosis_num, p_hat
## X-squared = 3.7616, df = 1, p-value = 0.05244
```

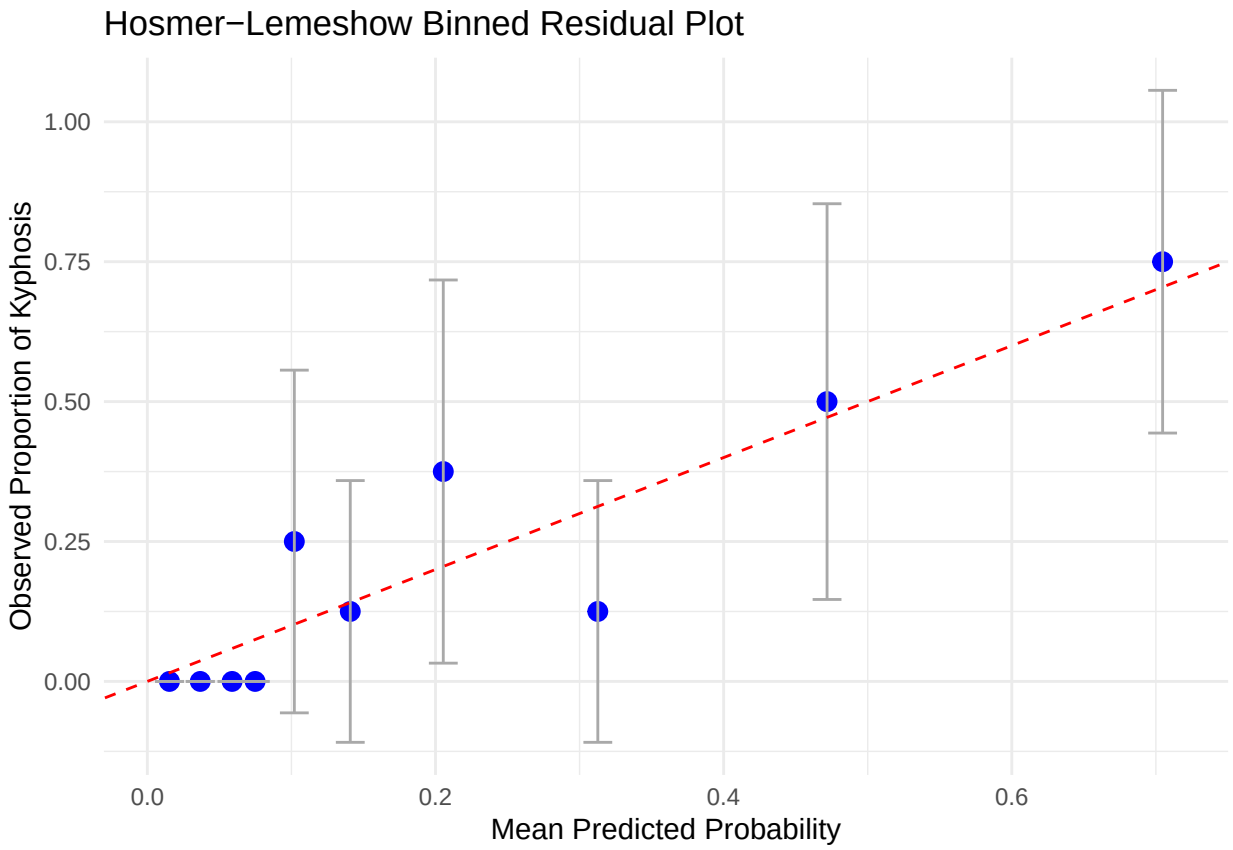
For the HL test, in order to have all of the bins with an expected frequency greater than 0, which typically is an issue with small datasets, the number of bins required is less than 4, so I chose 3 bins to compute the statistics. The HL test is sensitive to how you choose the bins. With 10 bins, the p-value is 0.60.

The result, p-value of 0.05244 is borderline, just above the conventional 0.05 threshold for statistical significance. The model isn't significantly misfitting but the result suggests the model's fit is not ideal either.

For the graphical representations, I will use 10 bins to get a more detailed understanding of the dispersion of the residuals overtime.

```
# Create a data frame with observed outcomes and predicted probabilities
df_plot <- data.frame(
  y = kyphosis$kyphosis_num,
  prob = p_hat
)

# Choose 10 bins based on predicted probabilities (quantile binning)
n_bins <- 10
df_plot$bin <- cut(df_plot$prob,
```

```
                  breaks = quantile(df_plot$prob, probs = seq(0, 1, length.out = n_bins + 1)),
                  include.lowest = TRUE)

# Summarize: observed proportion vs. mean predicted probability in each bin
bin_summary <- df_plot %>%
  group_by(bin) %>%
  summarize(
    mean_prob  = mean(prob),  # Mean predicted probability in each bin
    obs_rate   = mean(y),     # Observed proportion of kyphosis in each bin
    count      = n(),
    se         = sqrt((obs_rate * (1 - obs_rate)) / count)  # Standard error for binomial proportion
  )

ggplot(bin_summary, aes(x = mean_prob, y = obs_rate)) +
  geom_point(size = 3, color = "blue") +  # Observed proportions
  geom_errorbar(aes(ymin = obs_rate - 2 * se, ymax = obs_rate + 2 * se),
                width = 0.02, color = "darkgray") +  # Confidence intervals
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +  # Ideal fit line
  labs(x = "Mean Predicted Probability",
       y = "Observed Proportion of Kyphosis",
       title = "Hosmer-Lemeshow Binned Residual Plot") +
  theme_minimal()
```



Lower Probability Bins (0.0 - 0.2):

Most of these bins are below the red line, meaning the model overestimates kyphosis risk in this range.

The observed proportion of kyphosis is close to 0 in the lowest bins, even though the predicted probabilities

are slightly higher.

Mid-Range Probability Bins (0.2 - 0.4):

Some points lie close to the red line, meaning the model's predictions and actual kyphosis rates are roughly aligned in this range.

Error bars here are relatively moderate, indicating reasonable bin stability.

Higher Probability Bins (0.5 - 0.7):

The observed proportions are slightly above the red line, suggesting the model underestimates kyphosis probability at higher predicted values.

The error bars are quite large, meaning there is greater variability in these bins, likely due to fewer observations falling into this probability range.

Conclusion

The overall alignment is reasonable, with most points near the red line, indicating that the model generally fits well across probability ranges.

There is some miscalibration at the extremes:

The lowest bins (0.0 - 0.2) suggest slight overprediction of kyphosis probability.

The highest bins (0.5 - 0.7) suggest slight underprediction, but the large error bars indicate more uncertainty here.

No Major Systematic Misfit: The deviations are not extreme, and the p-value (0.60 with 10 bins) from the Hosmer-Lemeshow test supports this—no significant evidence of model misfit.

If refinement were needed, potential areas of improvement could involve investigating whether additional predictors or interaction terms might reduce miscalibration at the lowest and highest probability bins.

Final Verdict: Model fit appears reasonable, with small deviations in predicted vs. observed proportions mostly at the probability extremes.

h) Use the model to classify the subjects into predicted outcomes using a 0.5 cutoff. Produce cross-tabulation of these predicted outcomes with the actual outcomes. When kyphosis is actually present, what is the probability that this model would predict a present outcome? What is the name for this characteristic of the test?

```
# Predict probabilities
p_hat <- predict(model1, type = "response")

# Classify: predicted "present" if p_hat >= 0.5
pred_class <- ifelse(p_hat >= 0.5, 1, 0)

# Create a confusion matrix comparing predicted vs. actual
conf_mat <- table(
  Actual = kyphosis$kyphosis_num,
  Predicted = pred_class
)
conf_mat
```

```
##       Predicted
## Actual  0  1
##      0 61  3
##      1 10  7
```

```r
TP <- conf_mat["1", "1"]

# False Negatives
FN <- conf_mat["1", "0"]

# Sensitivity (True Positive Rate)
sensitivity <- TP / (TP + FN)
sensitivity
```

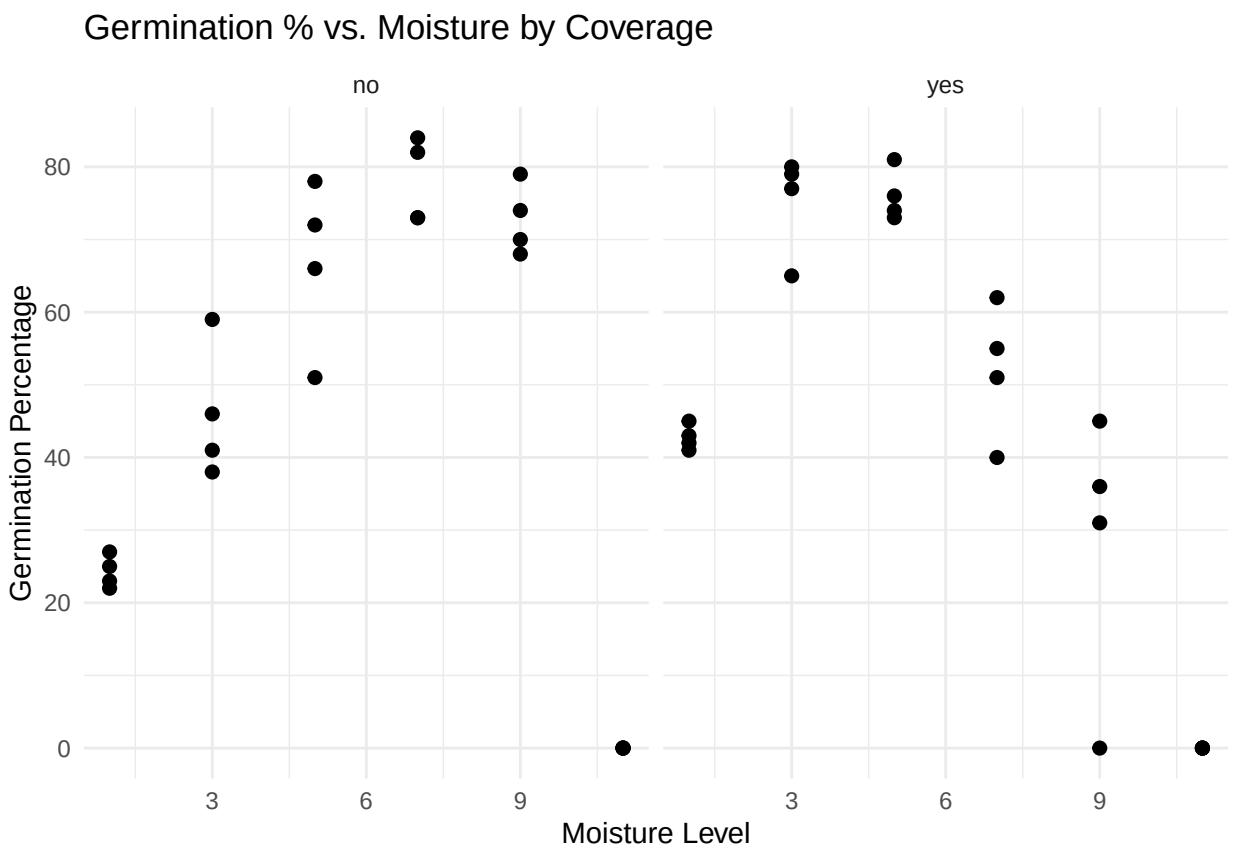
```
## [1] 0.4117647
```

The model's cross tabulation reveals that approximately **41.17%** of actual present cases are correctly predicted as present, and this metric is referred to as **Sensitivity, True Positive Rate, or Recall.**

2.[5 pts] A biologist analyzed an experiment to determine the effect of moisture content on seed germination. Eight boxes of 100 seeds each were treated with the same moisture level. Four boxes were covered and four left uncovered. The process was repeated at six different moisture levels.

a) Plot the germination percentage against the moisture level on two side-by-side plots according to the coverage of the box. What relationship do you see?

```r
library(faraway)
seeds <- faraway::seeds

# Convert any NA in 'germ' to 0
seeds$germ[is.na(seeds$germ)] <- 0
ggplot(seeds, aes(x = moisture, y = germ)) +
  geom_point(size = 2) +
  facet_wrap(~ covered, ncol = 2) +
  labs(
    title = "Germination % vs. Moisture by Coverage",
    x = "Moisture Level",
    y = "Germination Percentage"
  ) +
  theme_minimal()
```



From this plot, you'd normally expect higher moisture to boost germination overall—yet the data show a more irregular pattern, with some boxes performing well at mid-range moisture while others drop off at higher levels.

In particular:

No Simple Monotonic Trend: Germination doesn't consistently rise with moisture. Some points at low moisture are quite high, while at moisture close to 9 you see both high and very low percentages.

Coverage Not Universally Dominant: You don't see a uniform advantage for "yes" or "no"; some uncovered boxes achieve very high germination at around 6 or 9 moisture, while some covered boxes underperform at higher moisture.

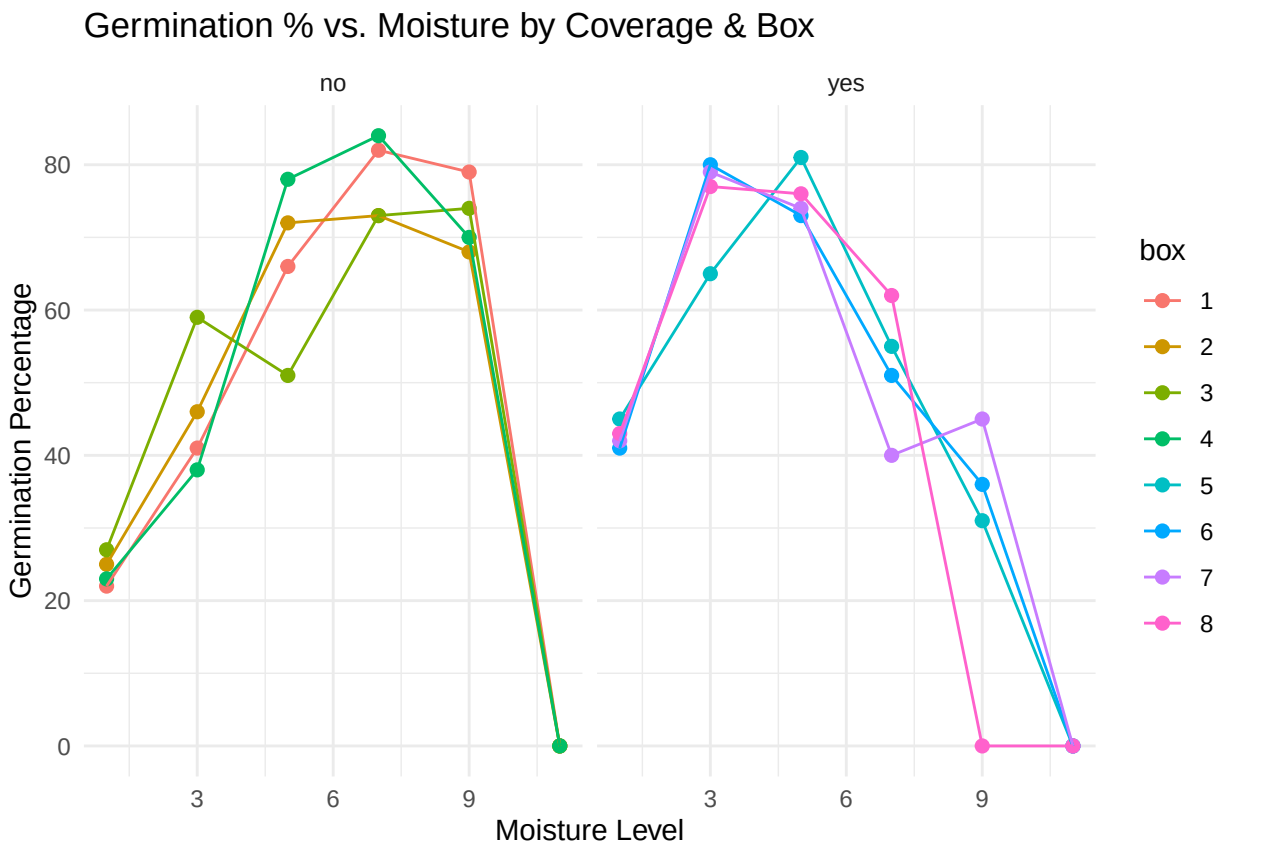
Likely Box Effect: The wide scatter at each moisture suggests variability from box to box is large enough to mask any smooth relationship. Some boxes thrive under certain moisture levels and coverage conditions, while others do poorly.

So the relationship between moisture and germination is complicated by both coverage and box-to-box differences, rather than a simple one-directional pattern.

b) Create a new factor describing the box (the data are ordered in blocks of 6 observations per box). Add lines to your previous plot that connect observations from the same box. Is there an indication of a box effect?

```
# Create the box factor:
# 8 boxes, each with 6 observations in consecutive order
seeds$box <- factor(rep(1:8, each = 6))

# Plot the germination vs. moisture, grouped by box
library(ggplot2)
ggplot(seeds, aes(x = moisture, y = germ, color = box, group = box)) +
```

```
  geom_point(size = 2) +
  geom_line() +
  facet_wrap(~ covered, ncol = 2) +
  labs(
    title = "Germination % vs. Moisture by Coverage & Box",
    x = "Moisture Level",
    y = "Germination Percentage"
  ) +
  theme_minimal()
```



From the plot, you can see that each box (line) traces a distinct trajectory of germination percentage across moisture levels, rather than overlapping into a single common pattern.

Some boxes consistently produce higher or lower germination than others, and this effect is not fully explained by differences in moisture alone or coverage alone.

For instance:

In the uncovered ("no") facet, the orange (box 2) and green (box 3) lines stay relatively high across moisture levels, while box 8 (dark green) dips lower.

In the covered ("yes") facet, certain boxes (e.g., pink for box 8) drop sharply at high moisture, while others remain somewhat higher.

These consistent shifts between boxes at each moisture level indicate an intrinsic "box effect"—some characteristic of each box (soil conditions, microclimate, etc.) that influences germination regardless of moisture or coverage.

c) Fit a binomial response model including the coverage, box and moisture predictors. Use the plots to determine an appropriate choice of model.

```r
model_binom <- glm(cbind(germ, 100 - germ) ~ covered + box + moisture,
                   data = seeds, family = binomial)
summary(model_binom)
```

```
##
## Call:
## glm(formula = cbind(germ, 100 - germ) ~ covered + box + moisture,
##     family = binomial, data = seeds)
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.652808   0.098433   6.632 3.31e-11 ***
## coveredyes  -0.224116   0.118471  -1.892   0.0585 .
## box2        -0.041756   0.117982  -0.354   0.7234
## box3        -0.041756   0.117982  -0.354   0.7234
## box4         0.020856   0.117916   0.177   0.8596
## box5         0.133533   0.118601   1.126   0.2602
## box6         0.161451   0.118550   1.362   0.1732
## box7         0.154476   0.118562   1.303   0.1926
## box8               NA         NA      NA       NA
## moisture    -0.120383   0.008784 -13.705  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1919.1  on 47  degrees of freedom
## Residual deviance: 1719.1  on 39  degrees of freedom
## AIC: 1926.9
##
## Number of Fisher Scoring iterations: 5
```

```r
mod_binom2 <- glm(cbind(germ, 100 - germ) ~ covered + box + moisture + I(moisture^2),
data = seeds, family = binomial)
summary(mod_binom2)
```

```
##
## Call:
## glm(formula = cbind(germ, 100 - germ) ~ covered + box + moisture +
##     I(moisture^2), family = binomial, data = seeds)
##
## Coefficients: (1 not defined because of singularities)
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.807034   0.135170 -13.369   <2e-16 ***
## coveredyes    -0.280031   0.132456  -2.114   0.0345 *
## box2          -0.052667   0.132502  -0.397   0.6910
## box3          -0.052667   0.132502  -0.397   0.6910
## box4           0.026389   0.132637   0.199   0.8423
## box5           0.166064   0.132269   1.255   0.2093
## box6           0.201077   0.132313   1.520   0.1286
## box7           0.192320   0.132301   1.454   0.1460
## box8                 NA         NA      NA       NA
## moisture       1.154925   0.044008  26.243   <2e-16 ***
## I(moisture^2) -0.111364   0.003898 -28.566   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1919.13  on 47  degrees of freedom
## Residual deviance:  653.63  on 38  degrees of freedom
## AIC: 863.37
##
## Number of Fisher Scoring iterations: 5
```

```
mod_binom3 <- glm(cbind(germ, 100 - germ) ~ covered * moisture + box,
data = seeds, family = binomial)
summary(mod_binom3)
```

```
##
## Call:
## glm(formula = cbind(germ, 100 - germ) ~ covered * moisture +
##     box, family = binomial, data = seeds)
##
## Coefficients: (1 not defined because of singularities)
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          0.03549    0.10876   0.326    0.744
## coveredyes           1.07196    0.16028   6.688 2.26e-11 ***
## moisture            -0.01704    0.01197  -1.423    0.155
## box2                -0.04011    0.11563  -0.347    0.729
## box3                -0.04011    0.11563  -0.347    0.729
## box4                 0.02003    0.11557   0.173    0.862
## box5                 0.14877    0.12520   1.188    0.235
## box6                 0.17990    0.12516   1.437    0.151
## box7                 0.17212    0.12517   1.375    0.169
## box8                      NA         NA      NA       NA
## coveredyes:moisture -0.22209    0.01822 -12.189  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1919.1  on 47  degrees of freedom
## Residual deviance: 1565.1  on 38  degrees of freedom
## AIC: 1774.8
##
## Number of Fisher Scoring iterations: 5
```

```
anova(model_binom, mod_binom2, mod_binom3)
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(germ, 100 - germ) ~ covered + box + moisture
## Model 2: cbind(germ, 100 - germ) ~ covered + box + moisture + I(moisture^2)
## Model 3: cbind(germ, 100 - germ) ~ covered * moisture + box
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        39    1719.13
## 2        38     653.63  1  1065.51 < 2.2e-16 ***
## 3        38    1565.06  0  -911.43
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

I chose to do ANOVA based on the summary statistics, the results show Model 2 is my best choice of model. Model 2 incorporates an additional quadratic moisture term which dramatically reduces the residual deviance, indicating the quadratic term in moisture provides a highly significant improvement over Model 1.

d) Test for the significance of a box effect in your model. Repeat the same test but using the Pearson's Chi-squared statistic instead of the deviance.

```
mod_binom2_no_box <- update(mod_binom2, . ~ . - box)
anova(mod_binom2_no_box, mod_binom2, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(germ, 100 - germ) ~ covered + moisture + I(moisture^2)
## Model 2: cbind(germ, 100 - germ) ~ covered + box + moisture + I(moisture^2)
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        44     657.22
## 2        38     653.63  6   3.5925   0.7316
```

```
pearson_full <- sum(resid(mod_binom2, type = "pearson")^2)
df_full <- df.residual(mod_binom2)

pearson_reduced <- sum(resid(mod_binom2_no_box, type = "pearson")^2)
df_reduced <- df.residual(mod_binom2_no_box)

# Difference in Pearson's X^2 and difference in df
chisq_diff <- pearson_reduced - pearson_full
df_diff <- df_reduced - df_full

pval_pearson <- 1 - pchisq(chisq_diff, df_diff)

chisq_diff
```

```
## [1] -0.07823378
```

```
df_diff
```

```
## [1] 6
```

```
pval_pearson
```

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

Conclusion from Deviance and Pearson Test are consistent. Both indicate that the effect of box is not statistically significant once you already have coverage and the non-linear moisture effects in the model.

As a result, I will use the model with box dropped.

e) At what value of moisture does the predicted maximum germination occur for noncovered boxes? For covered boxes?

```
# Create a moisture grid
moist_grid <- seq(min(seeds$moisture), max(seeds$moisture))

# Predict for noncovered boxes using the reduced model
pred_noncovered <- predict(mod_binom2_no_box,
                           newdata = data.frame(covered = "no", moisture = moist_grid),
                           type = "response")
```

```r
# Predict for covered boxes using the reduced model
pred_covered <- predict(mod_binom2_no_box,
                        newdata = data.frame(covered = "yes", moisture = moist_grid),
                        type = "response")

# Identify the moisture level at which predicted germination is maximized
max_moist_noncovered <- moist_grid[which.max(pred_noncovered)]
max_moist_covered    <- moist_grid[which.max(pred_covered)]

cat("Maximum predicted germination (noncovered) occurs at moisture level:",
    round(max_moist_noncovered, 2), "\n")
```

```
## Maximum predicted germination (noncovered) occurs at moisture level: 5
```

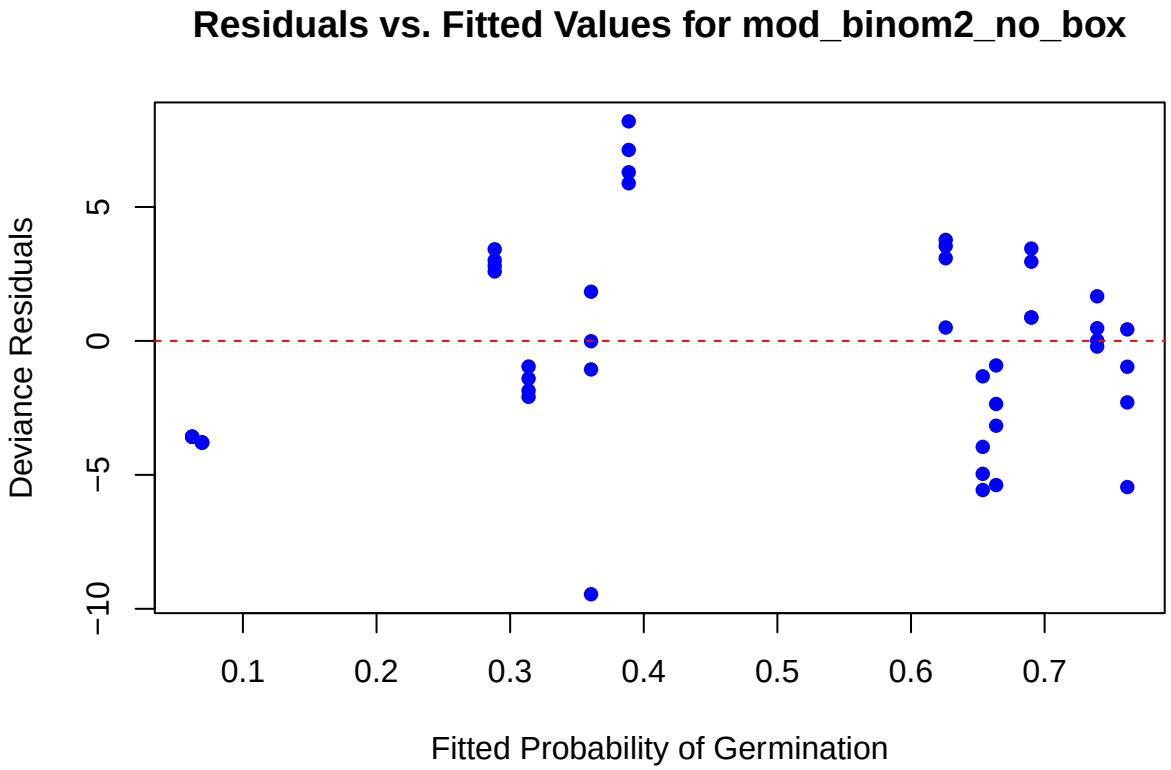
```r
cat("Maximum predicted germination (covered) occurs at moisture level:",
    round(max_moist_covered, 2), "\n")
```

```
## Maximum predicted germination (covered) occurs at moisture level: 5
```

Interestingly, the model produces maximum prediction germination to occur at the same moisture level for both noncovered and covered boxes.

f) Produce a plot of the residuals against the fitted values and interpret.

```r
# Extract deviance residuals and fitted values
dev_res <- resid(mod_binom2_no_box, type = "deviance")
fitted_vals <- fitted(mod_binom2_no_box)

# Plot the residuals vs. fitted values
plot(fitted_vals, dev_res,
     xlab = "Fitted Probability of Germination",
     ylab = "Deviance Residuals",
     main = "Residuals vs. Fitted Values for mod_binom2_no_box",
     pch = 16, col = "blue")
abline(h = 0, col = "red", lty = 2)
```



The plot of the residuals appear to show that my selected model is a good one.

There is no strong trend, there is a reasonable spread(variation) at the lower and higher ends which is normal in logistic models, but there is no obvious pattern suggesting over-or-under prediction in a specific region.

3.[5 pts] This problem concerns the modeling of the quantitative structure-activity relationships (QSAR) of the inhibition of dihydrofolate reductase (DHFR) by pyrimidines. We want to relate the physicochemical and/or structural properties as exhibited by the 26 predictors in pyrimidines with an activity level. We have structural information on 74 2,4-diamino- 5-(substituted benzyl) pyrimidines used as inhibitors of DHFR in E. coli. All the variables lie in [0,1].

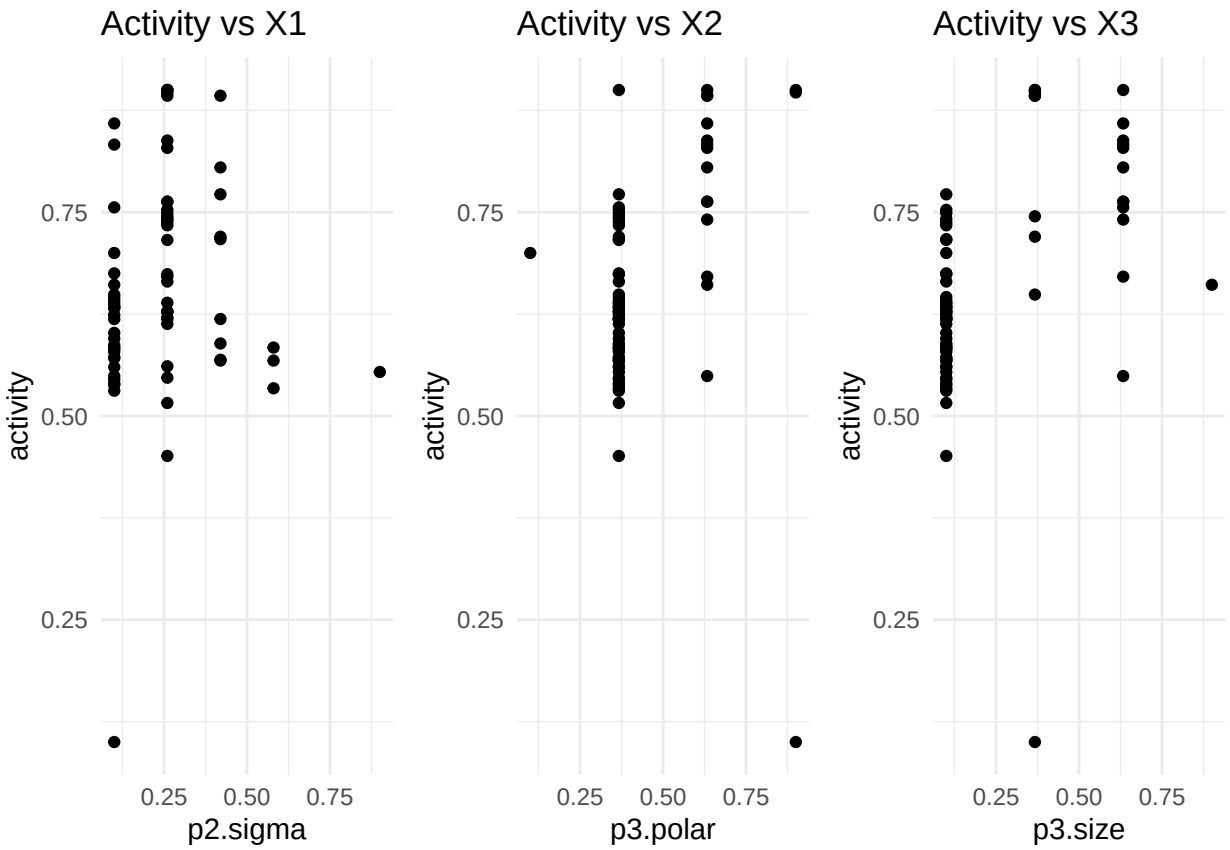
a) Plot the activity(response) against the first three predictors. Are any outliers in the response apparent? Remove any such cases.

```
pyrimidines <- faraway::pyrimidines
head(pyrimidines)
```

```
##   p1.polar p1.size p1.flex p1.h.doner p1.h.acceptor p1.pi.doner p1.pi.acceptor
## 1      0.5    0.26     0.1        0.9           0.9         0.9            0.1
## 2      0.5    0.26     0.1        0.1           0.1         0.5            0.1
## 3      0.3    0.42     0.1        0.1           0.5         0.5            0.1
## 4      0.1    0.74     0.7        0.1           0.5         0.1            0.1
## 5      0.1    0.42     0.4        0.1           0.5         0.1            0.1
## 6      0.3    0.42     0.3        0.9           0.9         0.1            0.1
##   p1.polarisable p1.sigma p2.polar p2.size p2.flex p2.h.doner p2.h.acceptor
## 1          0.367     0.42     0.26    0.10     0.1        0.1           0.1
## 2          0.367     0.58     0.42    0.26     0.1        0.9           0.1
## 3          0.367     0.26     0.26    0.10     0.1        0.1           0.1
## 4          0.367     0.10     0.26    0.10     0.1        0.1           0.1
## 5          0.367     0.10     0.26    0.10     0.1        0.1           0.1
## 6          0.367     0.10     0.26    0.10     0.1        0.1           0.1
##   p2.pi.doner p2.pi.acceptor p2.polarisable p2.sigma p3.polar p3.size p3.flex
```

```
## 1         0.1            0.1            0.1     0.10    0.367   0.100   0.100
## 2         0.9            0.1            0.1     0.26    0.900   0.367   0.100
## 3         0.1            0.1            0.1     0.10    0.633   0.633   0.100
## 4         0.1            0.1            0.1     0.10    0.367   0.100   0.100
## 5         0.1            0.1            0.1     0.10    0.367   0.100   0.100
## 6         0.1            0.1            0.1     0.10    0.633   0.633   0.633
##   p3.h.doner p3.h.acceptor p3.pi.doner p3.polarisable p3.sigma activity
## 1        0.1           0.1         0.1          0.100    0.100    0.571
## 2        0.1           0.1         0.5          0.367    0.900    0.900
## 3        0.1           0.5         0.5          0.367    0.367    0.833
## 4        0.1           0.1         0.1          0.100    0.100    0.582
## 5        0.1           0.1         0.1          0.100    0.100    0.587
## 6        0.9           0.9         0.1          0.367    0.100    0.549
```

```r
p1 <- ggplot(pyrimidines, aes(x=p2.sigma, y=activity)) +
  geom_point() +
  ggtitle("Activity vs X1") +
  theme_minimal()

p2 <- ggplot(pyrimidines, aes(x=p3.polar, y=activity)) +
  geom_point() +
  ggtitle("Activity vs X2") +
  theme_minimal()

p3 <- ggplot(pyrimidines, aes(x=p3.size, y=activity)) +
  geom_point() +
  ggtitle("Activity vs X3") +
  theme_minimal()

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



Yes, there seems to be 1 observation that has activity near 0 (< 0.10), this observation will be removed.

```
pyrimidines_clean <- subset(pyrimidines, activity > 0.10 & activity < 0.99)
```

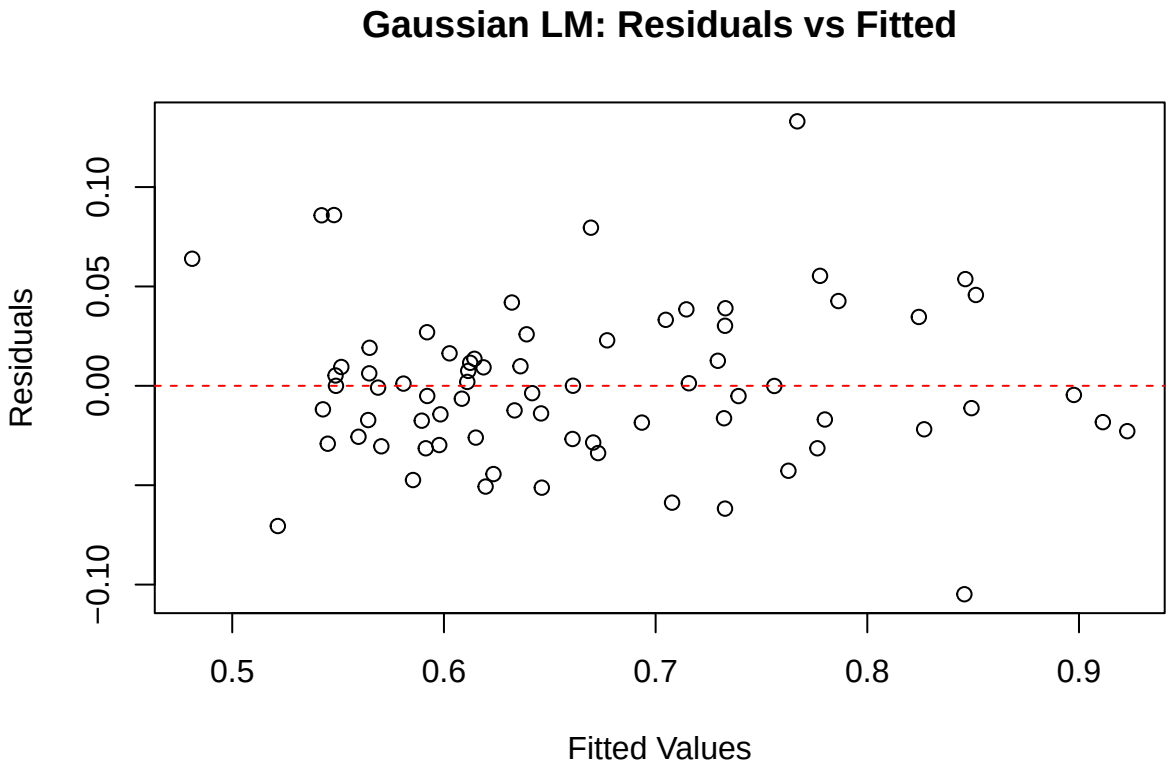
b) Fit a Gaussian linear model for the response with all 26 predictors. How well does this model fit the data in terms of R2? Plot the residuals against the fitted values. Is there any evidence of a violation of the standard assumptions?

```
model_gauss <- lm(activity ~ . , data = pyrimidines_clean)
summary(model_gauss)
```

```
##
## Call:
## lm(formula = activity ~ ., data = pyrimidines_clean)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.104856 -0.025626 -0.003679  0.019134  0.133071
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.524804   0.057478   9.131 6.77e-12 ***
## p1.polar       -0.281121   0.161738  -1.738  0.08888 .
## p1.size         0.161732   0.090760   1.782  0.08136 .
## p1.flex        -0.209947   0.071895  -2.920  0.00540 **
## p1.h.doner     -0.120697   0.060931  -1.981  0.05360 .
## p1.h.acceptor   0.051868   0.058282   0.890  0.37813
## p1.pi.doner     0.044287   0.059004   0.751  0.45672
```

```
## p1.pi.acceptor  0.118463   0.091183   1.299  0.20035
## p1.polarisable  0.148911   0.065750   2.265  0.02828 *
## p1.sigma        0.276042   0.156579   1.763  0.08455 .
## p2.polar       -0.032748   0.210570  -0.156  0.87709
## p2.size         0.156501   0.080072   1.955  0.05673 .
## p2.flex        -0.235798   0.067746  -3.481  0.00111 **
## p2.h.doner      0.018055   0.057312   0.315  0.75417
## p2.h.acceptor   0.023309   0.037575   0.620  0.53810
## p2.pi.doner    -0.014209   0.063834  -0.223  0.82483
## p2.pi.acceptor -0.009825   0.074666  -0.132  0.89589
## p2.polarisable  0.054130   0.050003   1.083  0.28465
## p2.sigma       -0.009310   0.185067  -0.050  0.96010
## p3.polar       -0.103299   0.180852  -0.571  0.57066
## p3.size         0.350239   0.131729   2.659  0.01075 *
## p3.flex        -0.100047   0.092197  -1.085  0.28351
## p3.h.doner      0.229888   0.222638   1.033  0.30721
## p3.h.acceptor  -0.284557   0.247213  -1.151  0.25565
## p3.pi.doner    -0.017501   0.367992  -0.048  0.96227
## p3.polarisable  0.021693   0.162123   0.134  0.89414
## p3.sigma        0.272428   0.218294   1.248  0.21835
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04913 on 46 degrees of freedom
## Multiple R-squared:  0.8744, Adjusted R-squared:  0.8034
## F-statistic: 12.32 on 26 and 46 DF,  p-value: 3.404e-13
```

```r
plot(fitted(model_gauss), resid(model_gauss),
     xlab = "Fitted Values",
     ylab = "Residuals",
     main = "Gaussian LM: Residuals vs Fitted")
abline(h=0, col="red", lty=2)
```



The adjusted R-squared is 0.8034, the R-squared is 0.8744, this is quite good, however considering it is a full model, it is likely prone to overfitting.

From the residuals vs fitted plot, there doesn't appear to be any strong pattern or shape, residuals are roughly scattered around zero without obvious curvature. That suggests no major violation of the linear model assumptions (normality, homoscedasticity) based on this plot alone.

c) Fit a quasi-binomial model for the activity response. Compare the predicted values for this model to those for the Gaussian linear model. Take care to compute the predicted values in the appropriate scale. Compare the fitted coefficients between the two models. Are there any substantial differences?

```
model_quasi <- glm(activity ~ ., data = pyrimidines_clean, family = quasibinomial)
summary(model_quasi)
```

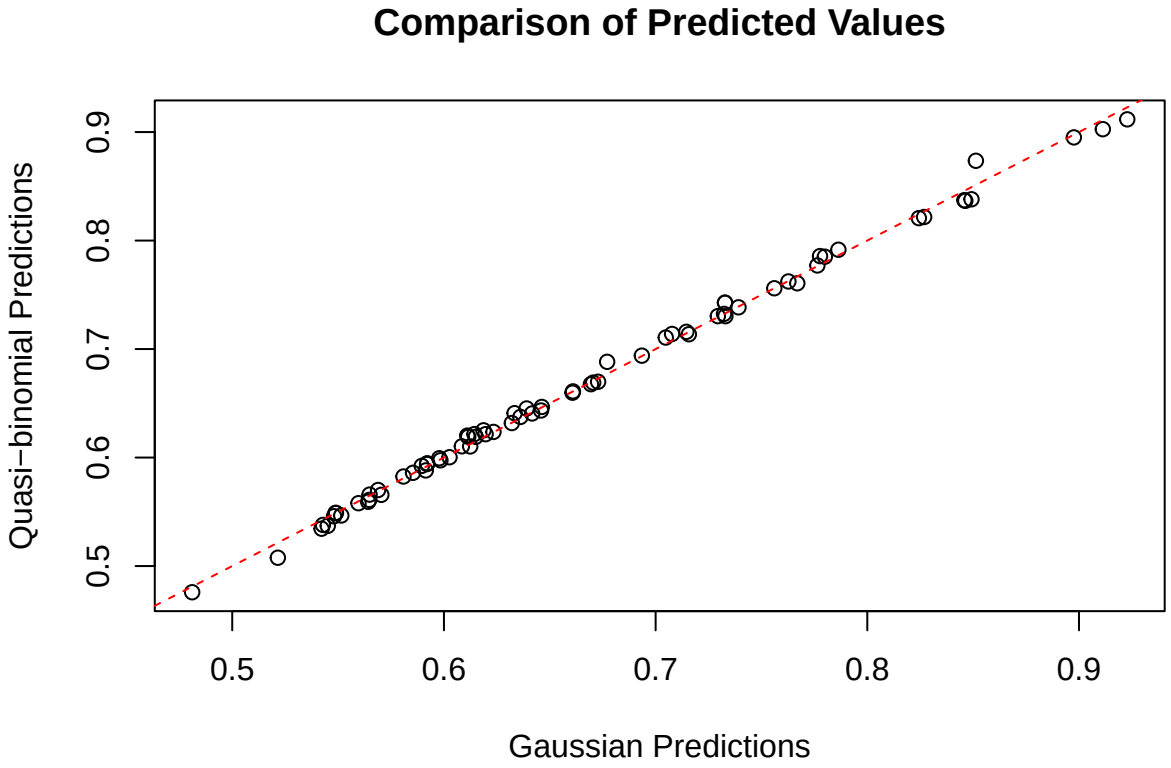
```
##
## Call:
## glm(formula = activity ~ ., family = quasibinomial, data = pyrimidines_clean)
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.048962   0.282100   0.174  0.86297
## p1.polar       -1.329158   0.779548  -1.705  0.09493 .
## p1.size         0.705941   0.431162   1.637  0.10839
## p1.flex        -0.921347   0.341241  -2.700  0.00967 **
## p1.h.doner     -0.505915   0.285662  -1.771  0.08318 .
## p1.h.acceptor   0.245498   0.272139   0.902  0.37170
## p1.pi.doner     0.146646   0.274156   0.535  0.59530
## p1.pi.acceptor  0.506579   0.429023   1.181  0.24376
## p1.polarisable  0.689160   0.313959   2.195  0.03324 *
## p1.sigma        1.309291   0.759034   1.725  0.09125 .
## p2.polar       -0.201515   0.994763  -0.203  0.84036
```

```
## p2.size         0.768765   0.397785   1.933  0.05945 .
## p2.flex        -1.177539   0.339639  -3.467  0.00115 **
## p2.h.doner      0.063053   0.279551   0.226  0.82255
## p2.h.acceptor   0.122573   0.184027   0.666  0.50870
## p2.pi.doner    -0.069150   0.308130  -0.224  0.82343
## p2.pi.acceptor -0.046187   0.352325  -0.131  0.89627
## p2.polarisable  0.246687   0.241786   1.020  0.31294
## p2.sigma        0.001547   0.868954   0.002  0.99859
## p3.polar       -0.674515   0.910651  -0.741  0.46264
## p3.size         1.424434   0.690750   2.062  0.04487 *
## p3.flex        -0.375631   0.479780  -0.783  0.43768
## p3.h.doner      1.400621   1.102696   1.270  0.21041
## p3.h.acceptor  -1.615232   1.247061  -1.295  0.20170
## p3.pi.doner     0.391883   1.920217   0.204  0.83919
## p3.polarisable  0.319654   0.840840   0.380  0.70558
## p3.sigma        1.726704   1.237534   1.395  0.16963
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 0.01216917)
##
##     Null deviance: 4.32379  on 72  degrees of freedom
## Residual deviance: 0.57852  on 46  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```
pred_gauss  <- predict(model_gauss)                    # on the linear model scale
pred_quasi  <- predict(model_quasi, type = "response")  # on the probability scale
```

The beta regression coefficients for each predictor in the two models are quite close to each other as expected.

```
plot(pred_gauss, pred_quasi,
     xlab = "Gaussian Predictions",
     ylab = "Quasi-binomial Predictions",
     main = "Comparison of Predicted Values")
abline(a = 0, b = 1, col = "red", lty = 2)
```



Both models produce nearly identical predictions for the activity variable in this data range, despite using different underlying assumptions (additive effects in the Gaussian model vs. log-odds in the quasi-binomial). This similarity arises because the activity values are well within (0,1) and do not approach the boundaries, making the simpler Gaussian approach practically equivalent to the logistic-based approach for these observations.

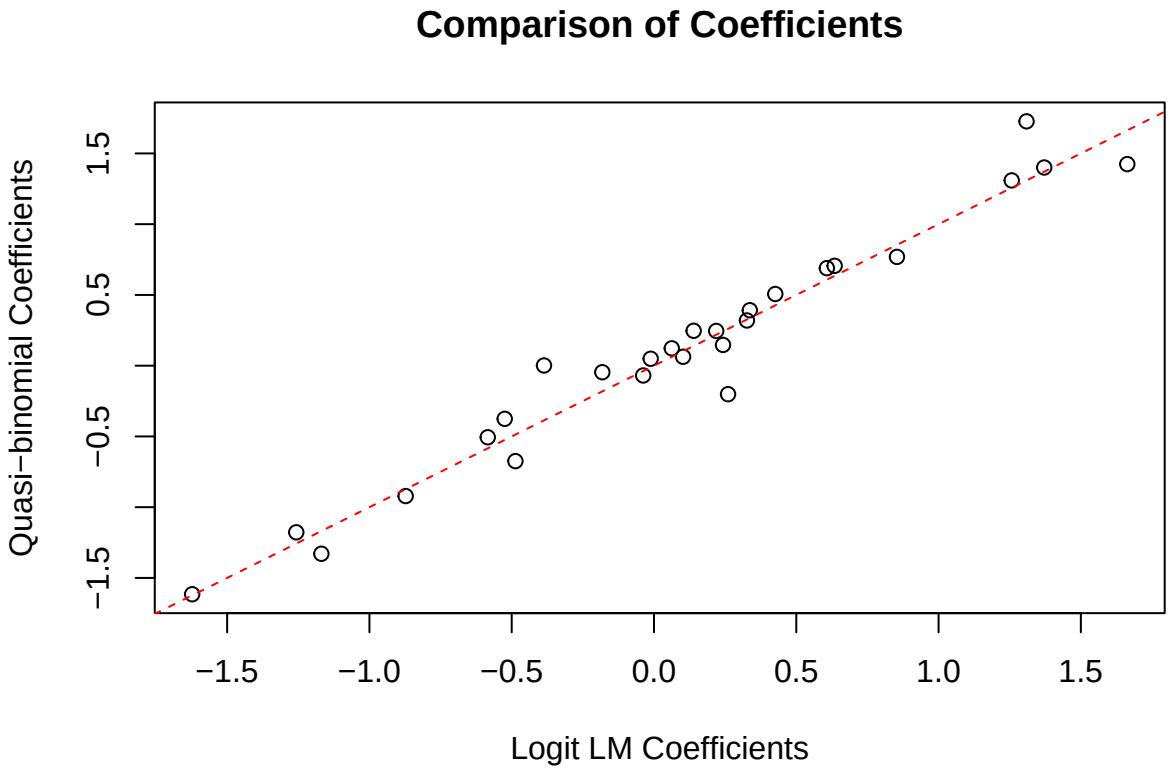
d) Fit a Gaussian linear model with the logit transformation applied to the response. Compare the coefficients of this model with the quasi-binomial model.

```
# Transform the response
pyrimidines_clean$logit_activity <- with(pyrimidines_clean, log(activity / (1 - activity)))


# Fit the Gaussian LM on logit_activity
model_logitlm <- lm(logit_activity ~ . - activity, data = pyrimidines_clean)
summary(model_logitlm)
```

```
##
## Call:
## lm(formula = logit_activity ~ . - activity, data = pyrimidines_clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.57575 -0.13307 -0.00969  0.08136  0.83277
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.01166    0.31584  -0.037   0.9707
## p1.polar       -1.16892    0.88876  -1.315   0.1949
## p1.size         0.63487    0.49873   1.273   0.2094
## p1.flex        -0.87269    0.39507  -2.209   0.0322 *
```

```
## p1.h.doner     -0.58420    0.33482  -1.745   0.0877 .
## p1.h.acceptor   0.21864    0.32026   0.683   0.4982
## p1.pi.doner     0.24249    0.32423   0.748   0.4583
## p1.pi.acceptor  0.42623    0.50105   0.851   0.3994
## p1.polarisable  0.60747    0.36130   1.681   0.0995 .
## p1.sigma        1.25707    0.86041   1.461   0.1508
## p2.polar        0.26019    1.15709   0.225   0.8231
## p2.size         0.85376    0.44000   1.940   0.0585 .
## p2.flex        -1.25711    0.37226  -3.377   0.0015 **
## p2.h.doner      0.10221    0.31493   0.325   0.7470
## p2.h.acceptor   0.06255    0.20647   0.303   0.7633
## p2.pi.doner    -0.03791    0.35077  -0.108   0.9144
## p2.pi.acceptor -0.18163    0.41029  -0.443   0.6601
## p2.polarisable  0.13935    0.27477   0.507   0.6145
## p2.sigma       -0.38654    1.01695  -0.380   0.7056
## p3.polar       -0.48698    0.99379  -0.490   0.6264
## p3.size         1.66321    0.72385   2.298   0.0262 *
## p3.flex        -0.52463    0.50663  -1.036   0.3058
## p3.h.doner      1.37136    1.22340   1.121   0.2681
## p3.h.acceptor  -1.62291    1.35844  -1.195   0.2383
## p3.pi.doner     0.33671    2.02213   0.167   0.8685
## p3.polarisable  0.32654    0.89087   0.367   0.7156
## p3.sigma        1.30914    1.19953   1.091   0.2808
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.27 on 46 degrees of freedom
## Multiple R-squared:  0.8666, Adjusted R-squared:  0.7912
## F-statistic: 11.49 on 26 and 46 DF,  p-value: 1.23e-12
```

```r
coef_quasi <- coef(model_quasi)
coef_logitlm <- coef(model_logitlm)

# Plot the common coefficients against each other
plot(coef_logitlm, coef_quasi,
     xlab = "Logit LM Coefficients",
     ylab = "Quasi-binomial Coefficients",
     main = "Comparison of Coefficients")
abline(a = 0, b = 1, col = "red", lty = 2)
```



These coefficients are very similar as expected based on the properties of transforming the gaussian LM response with the logit function.

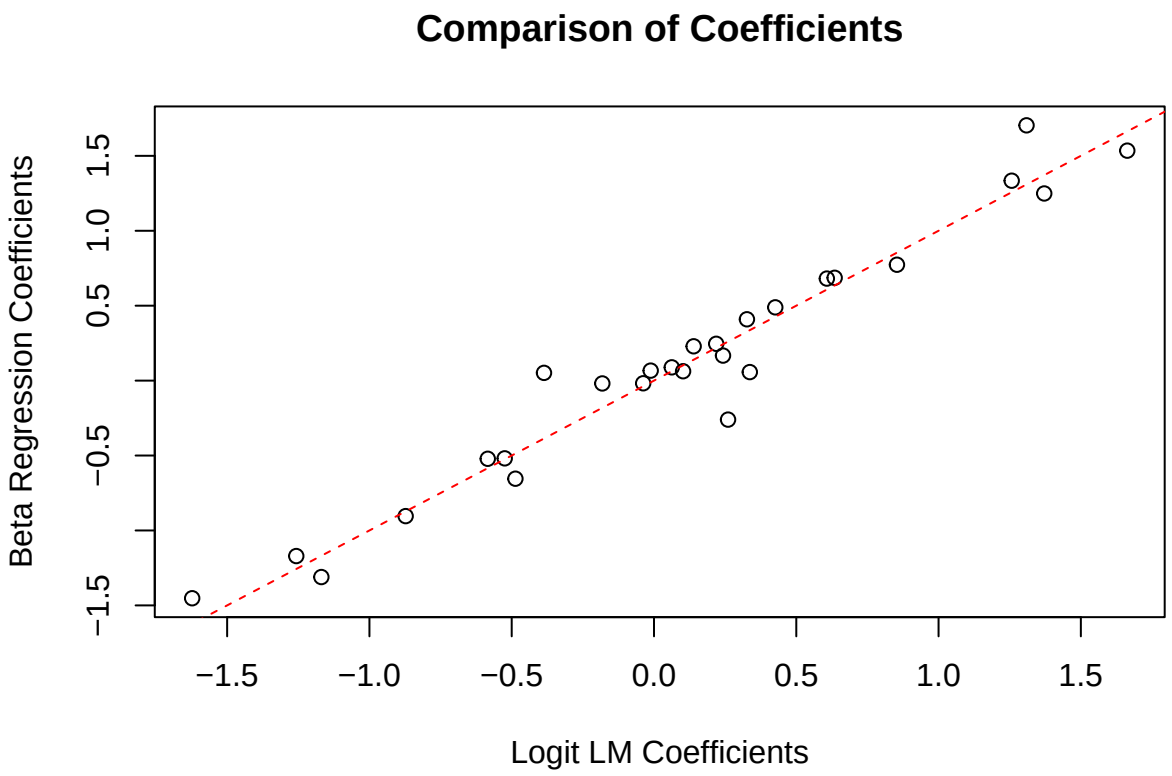
e) Fit a Beta regression model. Compare the coefficients of this model with that of logit response regression model.

```
library(betareg)

# Define a reduced formula using only 3 predictors:
form_beta <- activity ~ p1.polar + p1.size + p1.flex + p1.h.doner + p1.h.acceptor  + p1.pi.doner    + p
p1.sigma       + p2.polar      + p2.size       + p2.flex       + p2.h.doner    + p2.h.acceptor  + p2.pi.
p2.polarisable + p2.sigma       + p3.polar     + p3.size       + p3.flex       + p3.h.doner     + p3.h.a

# Fit the beta regression model on the reduced predictor set:
model_beta<- betareg(form_beta , data = pyrimidines_clean)

# Check the summary
summary(model_beta)
```

```
##
## Call:
## betareg(formula = form_beta, data = pyrimidines_clean)
##
## Quantile residuals:
##     Min      1Q  Median      3Q     Max
## -2.7450 -0.5777 -0.1151  0.3894  3.8300
##
## Coefficients (mean model with logit link):
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.06670    0.23105   0.289  0.77283
## p1.polar       -1.31096    0.63894  -2.052  0.04019 *
```

```
## p1.size          0.68620    0.35347   1.941  0.05222 .
## p1.flex         -0.90419    0.27976  -3.232  0.00123 **
## p1.h.doner      -0.52159    0.23435  -2.226  0.02604 *
## p1.h.acceptor    0.24574    0.22322   1.101  0.27095
## p1.pi.doner      0.16714    0.22504   0.743  0.45766
## p1.pi.acceptor   0.48896    0.35205   1.389  0.16486
## p1.polarisable   0.68094    0.25733   2.646  0.00814 **
## p1.sigma         1.33417    0.62223   2.144  0.03202 *
## p2.polar        -0.26022    0.81658  -0.319  0.74998
## p2.size          0.77340    0.32621   2.371  0.01775 *
## p2.flex         -1.17079    0.27853  -4.203 2.63e-05 ***
## p2.h.doner       0.06261    0.22926   0.273  0.78479
## p2.h.acceptor    0.08859    0.15129   0.586  0.55819
## p2.pi.doner     -0.01776    0.25285  -0.070  0.94402
## p2.pi.acceptor -0.01891    0.28925  -0.065  0.94787
## p2.polarisable   0.22905    0.19826   1.155  0.24796
## p2.sigma         0.05168    0.71324   0.072  0.94224
## p3.polar        -0.65463    0.74455  -0.879  0.37928
## p3.size          1.53465    0.56339   2.724  0.00645 **
## p3.flex         -0.51875    0.39352  -1.318  0.18742
## p3.h.doner       1.24891    0.90502   1.380  0.16759
## p3.h.acceptor   -1.45232    1.02218  -1.421  0.15537
## p3.pi.doner      0.05700    1.57468   0.036  0.97113
## p3.polarisable   0.40920    0.68884   0.594  0.55249
## p3.sigma         1.70353    1.00494   1.695  0.09004 .
##
## Phi coefficients (precision model with identity link):
##       Estimate Std. Error z value Pr(>|z|)
## (phi)   120.95      19.95   6.062 1.34e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Type of estimator: ML (maximum likelihood)
## Log-likelihood: 129.7 on 28 Df
## Pseudo R-squared: 0.8641
## Number of iterations: 43 (BFGS) + 3 (Fisher scoring)
```

```r
coef_logitlm <- coef(model_logitlm)
coef_beta <- coef(model_beta)

common_names <- intersect(names(coef_beta), names(coef_logitlm))
comparison_df <- data.frame(
  Predictor = common_names,
  Beta_Coeff = coef_beta[common_names],
  LogitLM_Coeff = coef_logitlm[common_names]
)

plot(coef_logitlm[common_names], coef_beta[common_names],
     xlab = "Logit LM Coefficients",
     ylab = "Beta Regression Coefficients",
     main = "Comparison of Coefficients")
abline(a = 0, b = 1, col = "red", lty = 2)
```



The coefficients of the beta regression and logit response regression are very similar as shown by the plot.

f) What property of the response leads to the similarity of the models considered thus far in this question?

The key property is that the response variable, activity, is a proportion that is strictly bounded in (0, 1) and, after removing extreme outliers, is mostly distributed in the mid-range rather than clustering at 0 or 1.

This means that when applying a logit transformation (or a logit link in a quasi-binomial or beta regression model), the transformed data do not suffer from severe boundary issues. As a result, models that assume a logit relationship—whether via a transformed linear model, a quasi-binomial GLM, or a beta regression—yield similar coefficient estimates and predictions.

In other words, because activity is well-behaved and lies comfortably away from the boundaries (since we removed the outlier), all these models essentially capture the same underlying relationship, leading to similar fits and similar interpretations of the predictors' effects.