Set_Nine

Oluwanifemi

2024-12-01

```
library(faraway)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

```
library(alr4)
```

```
## Loading required package: car

## Loading required package: carData

##
## Attaching package: 'car'

## The following objects are masked from 'package:faraway':
##
##     logit, vif

## Loading required package: effects

## lattice theme set by effectsTheme()
## See ?effectsTheme for details.

##
## Attaching package: 'alr4'

## The following objects are masked from 'package:faraway':
##
##     cathedral, pipeline, twins
```

```
data("sat")
View(sat)
```

Question 1

(a)

Label leakage happens when information from the target variable leaks into the independent variables used to build the predictive model.This often occurs when features contain data that are directly or indirectly derived from the label which leads to an inflated model performance during training and testing, as the model is essentially being fed information it shouldn't have access to in a true predictive setting.

Training example leakage happens when information from the test set leaks into the training process, either directly or indirectly. This can happen due to improper data splitting. It can be caused by performing normalization on the entire dataset before splitting into training and test sets, thereby "contaminating" the test data. This causes the model to perform better on the test set because it has seen or indirectly used parts of the test data during training, leading to overconfidence in its performance.

(b)

P-hacking refers to the practice of manipulating data analysis or selectively reporting results to achieve statistically significant outcomes ($p < 0.05$). This can include practices like testing multiple hypotheses and only reporting those with significant results, or tweaking model specifications to produce a desired outcome. The problem lies in inflating the likelihood of false positives, undermining the reliability of findings.P-hacking is a big issue in many fields that rely on quantitative resaerch like data science/data analysis because it could create false confidence in conclusions, which might skew perception of underlying factors in data, and make people trust the wrong conclusion leading to ignorance and miseducation.

Multiple comparison : The article explores the issue of multiple comparisons in data analysis, specifically in neuroscience, where numerous statistical tests are performed on the same data. It highlights the risk of Type I errors (false positives) and stresses the importance of applying corrections, such as the Bonferroni correction etc. , to control for this. Without these adjustments, findings can be misleading, making it crucial to ensure the reliability and validity of results in complex studies.The issue of multiple comparisons is highly important in data science, especially when handling large datasets or performing numerous tests. Failing to adjust for multiple comparisons can lead to false positives, which can mislead conclusions and impact the reliability of the analysis. Correcting for this issue ensures that findings are statistically valid and not driven by chance, making it crucial for drawing accurate and trustworthy insights from data.

(c)

####(i) I learnt more about the importance of Data Analysis to break into Machine Learning and the importance of clustering in Machine Learning. (ii) I learnt about "Cheating" in data training (data leakages) with the way some Etsy shop owners were naming their brands e.g. the glasses shop and how it affects data being collated for analysis in general

```
model_1 <- lm(total~expend+ratio+salary+takers, data = sat)
summary(model_1)
```

```
##
## Call:
## lm(formula = total ~ expend + ratio + salary + takers, data = sat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -90.531 -20.855  -1.746  15.979  66.571
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1045.9715    52.8698  19.784  < 2e-16 ***
```

```
## expend         4.4626    10.5465   0.423    0.674
## ratio         -3.6242     3.2154  -1.127    0.266
## salary         1.6379     2.3872   0.686    0.496
## takers        -2.9045     0.2313 -12.559 2.61e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.7 on 45 degrees of freedom
## Multiple R-squared:  0.8246, Adjusted R-squared:  0.809
## F-statistic: 52.88 on 4 and 45 DF,  p-value: < 2.2e-16
```

```
new_state <- data.frame(expend = 6, ratio = 16, salary = 35, takers = 30)

pred_interval <- predict(model_1, newdata = new_state, interval = "prediction", level = 0.99)

pred_interval
```

```
##        fit      lwr      upr
## 1 984.9521 895.7647 1074.139
```

The fit is going to be the same, which is the center of the interval, but the width will change. The width of the interval will decrease when constructing a confidence interval compared to a prediction interval. This is because ,a confidence interval estimates the mean response (average SAT score) for the specified predictors, which has less uncertainty because it considers only the variability of the model's predictions.

```
conf_interval <- predict(model_1, newdata = new_state, interval = "confidence", level = 0.99)

conf_interval
```

```
##        fit      lwr      upr
## 1 984.9521 970.1753 999.7288
```

If we use 95%, the center of the interval will still be the same as it determined solely by the regression equation, but there will be a change in the width. 95 percent would mean less certainty to encompass the true values, a smaller critical values will be used, which would reduce the margin of error and make the interval narrower.

Changing the Pupil/Teacher ratio to 25 changes on the variables used in the regression equation also it alters the center (fit),the coefficent for ratio is negative, therefore the predicted SAT score will decrease. The width of the interval would also likely change, if I am to get the average ratio and 25 is far from it, the confidence interval is likely to widen due to increased uncertainty.

```
pred_interval_1 <- predict(model_1, newdata = new_state, interval = "prediction", level = 0.95)

pred_interval_1
```

```
##        fit      lwr     upr
## 1 984.9521 918.1638 1051.74
```

```r
conf_interval_1 <- predict(model_1, newdata = new_state, interval = "confidence", level = 0.95)

conf_interval_1
```

```
##        fit      lwr      upr
## 1 984.9521 973.8864 996.0177
```

```r
new_state_1 <- data.frame(expend = 6, ratio = 25, salary = 35, takers = 30)

pred_interval_2 <- predict(model_1, newdata = new_state_1, interval = "prediction", level = 0.99)

pred_interval_2
```

```
##       fit      lwr      upr
## 1 952.334 838.6646 1066.003
```

```r
conf_interval_2 <- predict(model_1, newdata = new_state_1, interval = "confidence", level = 0.99)

conf_interval_2
```

```
##       fit      lwr      upr
## 1 952.334 880.3292 1024.339
```

```r
# ADDITIONAL CODE
pred_interval_3 <- predict(model_1, newdata = new_state_1, interval = "prediction", level = 0.95)

pred_interval_3
```

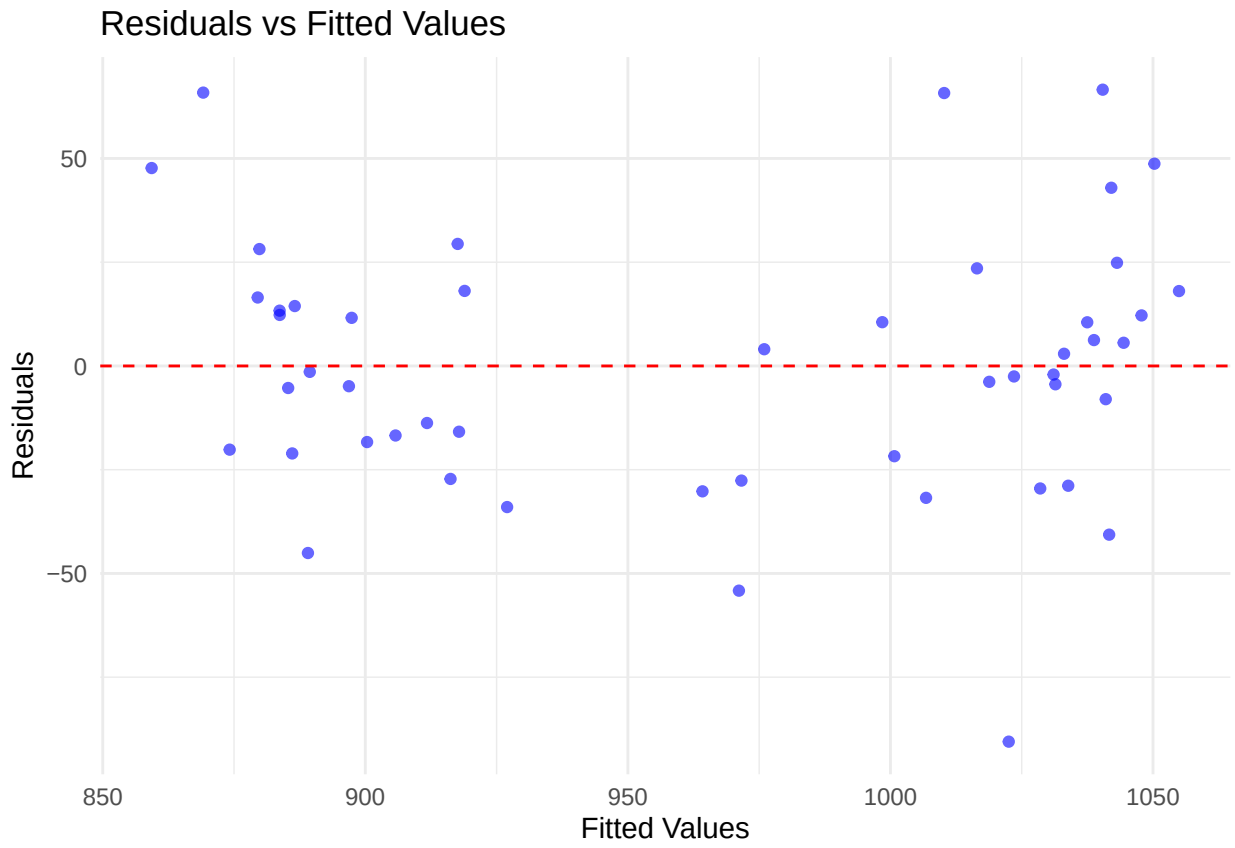
```
##       fit      lwr      upr
## 1 952.334 867.2123 1037.456
```

```r
conf_interval_3 <- predict(model_1, newdata = new_state_1, interval = "confidence", level = 0.95)

conf_interval_3
```

```
##       fit     lwr      upr
## 1 952.334 898.413 1006.255
```

```r
# Create the residuals vs fitted values plot
#plot(model_1$fitted.values, resid(model),
    # xlab = "Fitted Values",
    # ylab = "Residuals",
    # main = "Residuals vs Fitted Values",
    # pch = 20)
#abline(h = 0, col = "red")  # Add a horizontal line at 0 for reference


residuals <- resid(model_1)
fitted_values <- fitted(model_1)
```

```r
diagnostic_data <- data.frame(Fitted = fitted_values, Residuals = residuals)

ggplot(diagnostic_data, aes(x = Fitted, y = Residuals)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Fitted Values",
       x = "Fitted Values",
       y = "Residuals") +
  theme_minimal()
```



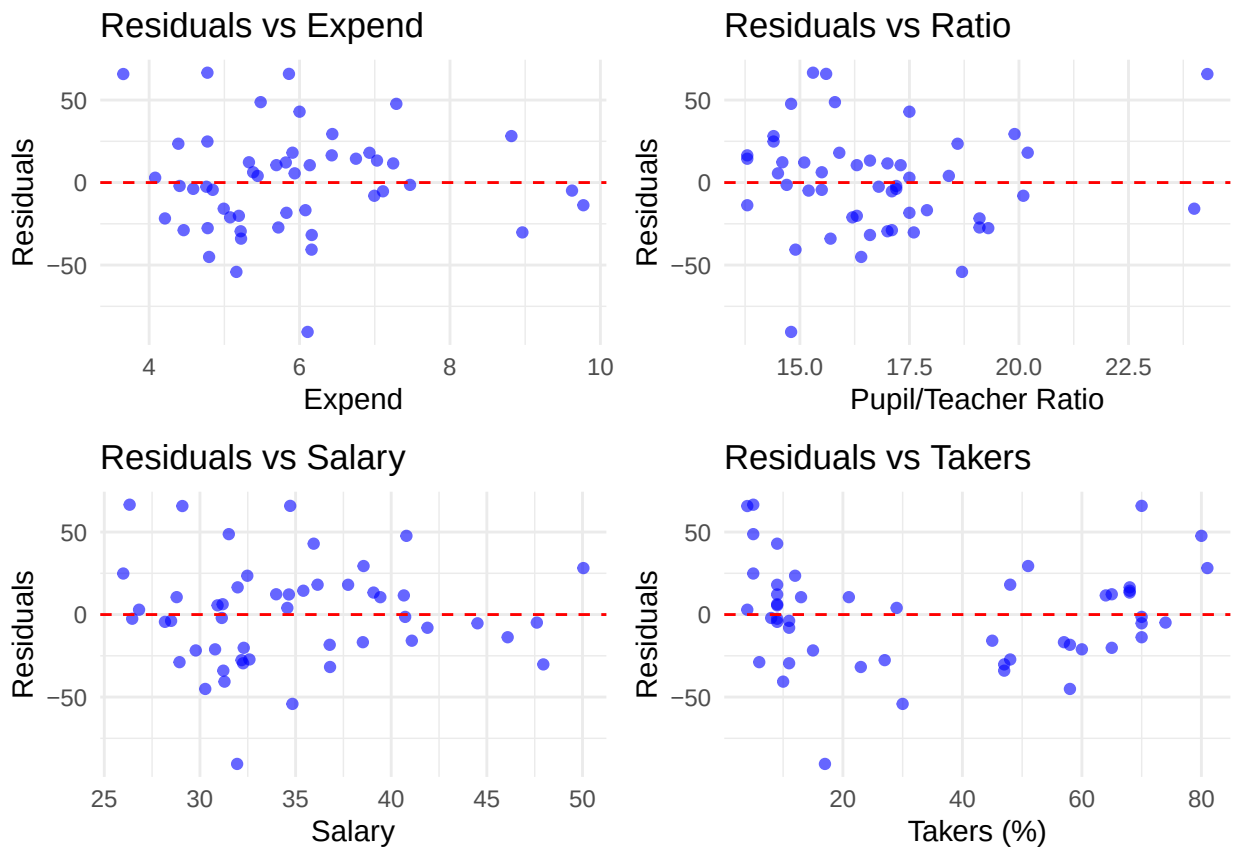
Based on the plot, the residuals seems scattered and a little congested on the opposing sides, but all in all, I would not say it violated equal variance badly. But it gives an illusion of an almost quadratic shape forming.

```r
# Extract residuals
residuals <- resid(model_1)

# Add residuals to the original data for plotting
data_with_residuals <- sat
data_with_residuals$residuals <- residuals

# Create residual plots for each covariate
plot_expend <- ggplot(data_with_residuals, aes(x = expend, y = residuals)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Expend", x = "Expend", y = "Residuals") +
```

```
  theme_minimal()

plot_ratio <- ggplot(data_with_residuals, aes(x = ratio, y = residuals)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Ratio", x = "Pupil/Teacher Ratio", y = "Residuals") +
  theme_minimal()

plot_salary <- ggplot(data_with_residuals, aes(x = salary, y = residuals)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Salary", x = "Salary", y = "Residuals") +
  theme_minimal()

plot_takers <- ggplot(data_with_residuals, aes(x = takers, y = residuals)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Takers", x = "Takers (%)", y = "Residuals") +
  theme_minimal()

# Display all plots
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.4.2
```

```
grid.arrange(plot_expend, plot_ratio, plot_salary, plot_takers, ncol = 2)
```



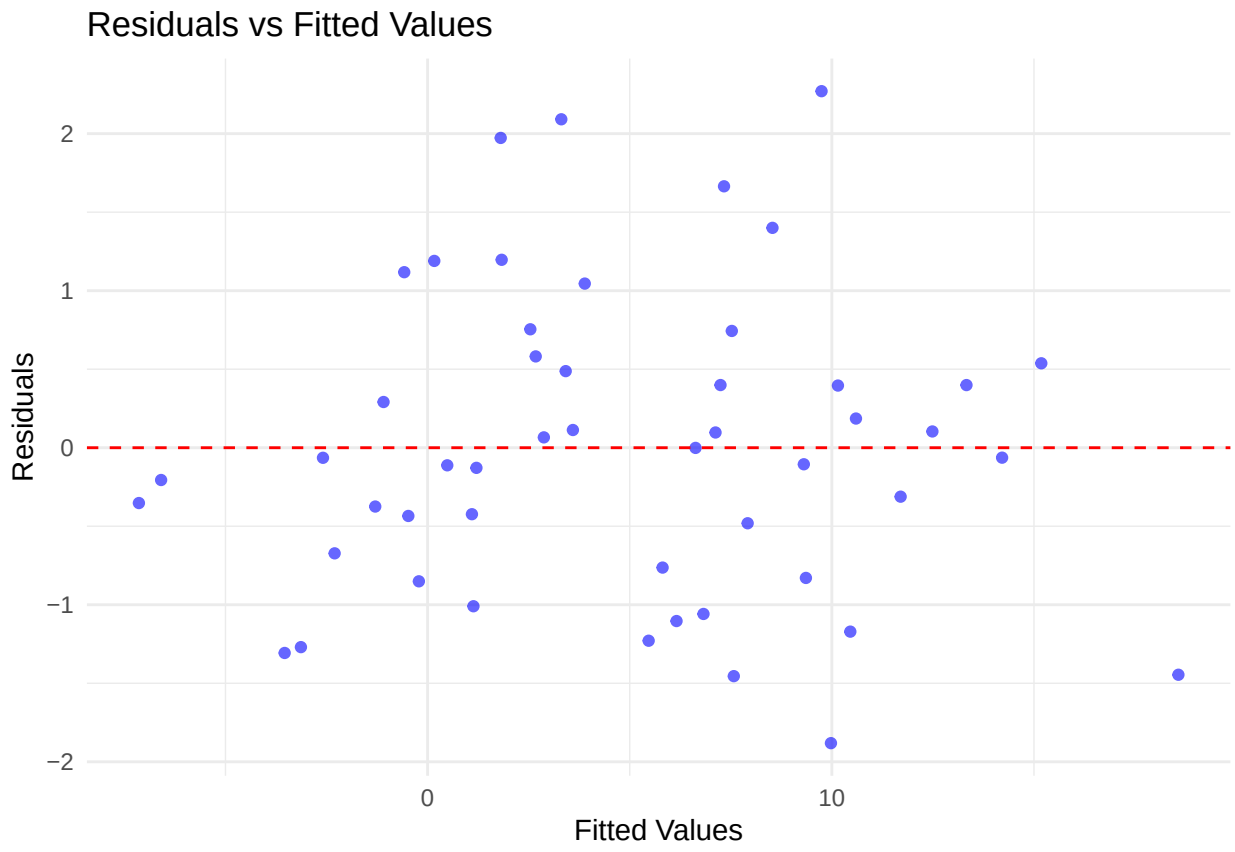
The equal variance for expend is scattered but seems to have a little concentration in the center, the graph for Ratio shows a skew to the right side, that violated equal variance.Both the Expend and Ratio graph show subtle signs of coning. Both violate equal variance slightly with Ratio showing more violation. Salary's graph seems very scattered with little discernable pattern, while Takers has some points congested at its opposite sides, that seems to defy equal variance(slightly quadratic). Using Salary alone or with Expend as x variables might not neccesarily violate equal variance badly.

```r
# Simulate the data
set.seed(123)  # Set seed for reproducibility
x1 <- rnorm(50)  # Draw 50 values for x1 from standard normal distribution
x2 <- rnorm(50)  # Draw 50 values for x2 from standard normal distribution
y <- 5 + 6 * x1 - x2 + rnorm(50)  # Generate y based on the specified model

# Fit the linear model
model <- lm(y ~ x1 + x2)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.88137 -0.74056 -0.06374  0.52516  2.27045
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   4.7701     0.1431  33.341  < 2e-16 ***
## x1            6.0252     0.1540  39.122  < 2e-16 ***
## x2           -1.1696     0.1575  -7.427 1.85e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9975 on 47 degrees of freedom
## Multiple R-squared:  0.9716, Adjusted R-squared:  0.9704
## F-statistic: 804.3 on 2 and 47 DF,  p-value: < 2.2e-16
```

```r
# Extract residuals and fitted values
residuals <- resid(model)
fitted_values <- fitted(model)

# Create the residuals vs fitted values plot
ggplot(data = data.frame(Fitted = fitted_values, Residuals = residuals),
       aes(x = Fitted, y = Residuals)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Fitted Values", x = "Fitted Values", y = "Residuals") +
  theme_minimal()
```



The graph shows scatterd points with no clear pattern so there might be evidence of homoscedacity, that is, equal variance,

```r
# Simulate the data
set.seed(123)  # Set seed for reproducibility
x1 <- rnorm(50)  # Draw 50 values for x1 from standard normal distribution
x2 <- rnorm(50)  # Draw 50 values for x2 from standard normal distribution

# Generate y with error term variance dependent on x1
y <- 5 + 6 * x1 - x2 + rnorm(50, 0, x1^2)

# Fit the linear model
model <- lm(y ~ x1 + x2)

# Output the model summary
summary(model)
```

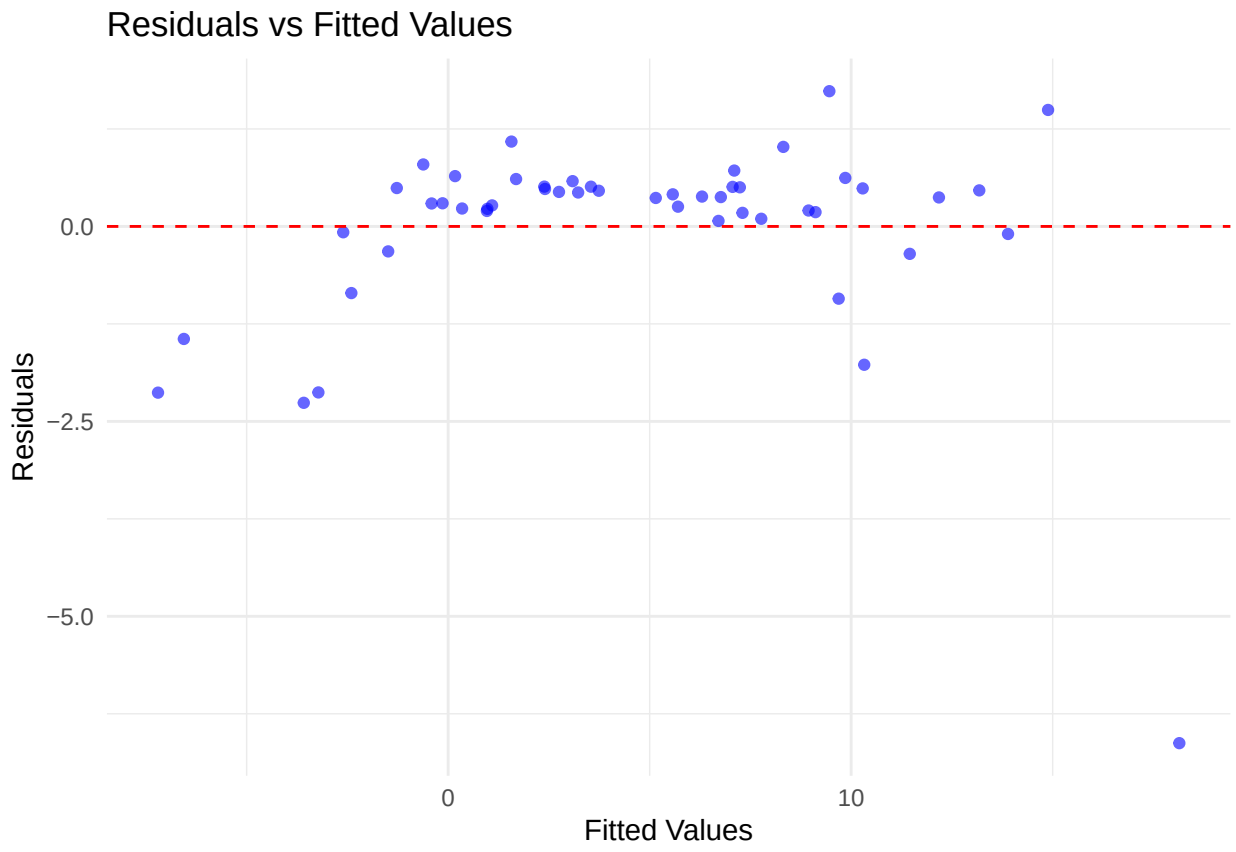
```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6265  0.0773  0.3683  0.5056  1.7342
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.5664     0.1861  24.537  < 2e-16 ***
## x1            5.9541     0.2003  29.721  < 2e-16 ***
## x2           -1.0543     0.2048  -5.147 5.11e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.298 on 47 degrees of freedom
## Multiple R-squared:  0.9515, Adjusted R-squared:  0.9494
## F-statistic:   461 on 2 and 47 DF,  p-value: < 2.2e-16
```

```r
model <- lm(y ~ x1 + x2)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6265  0.0773  0.3683  0.5056  1.7342
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.5664     0.1861  24.537  < 2e-16 ***
## x1            5.9541     0.2003  29.721  < 2e-16 ***
```

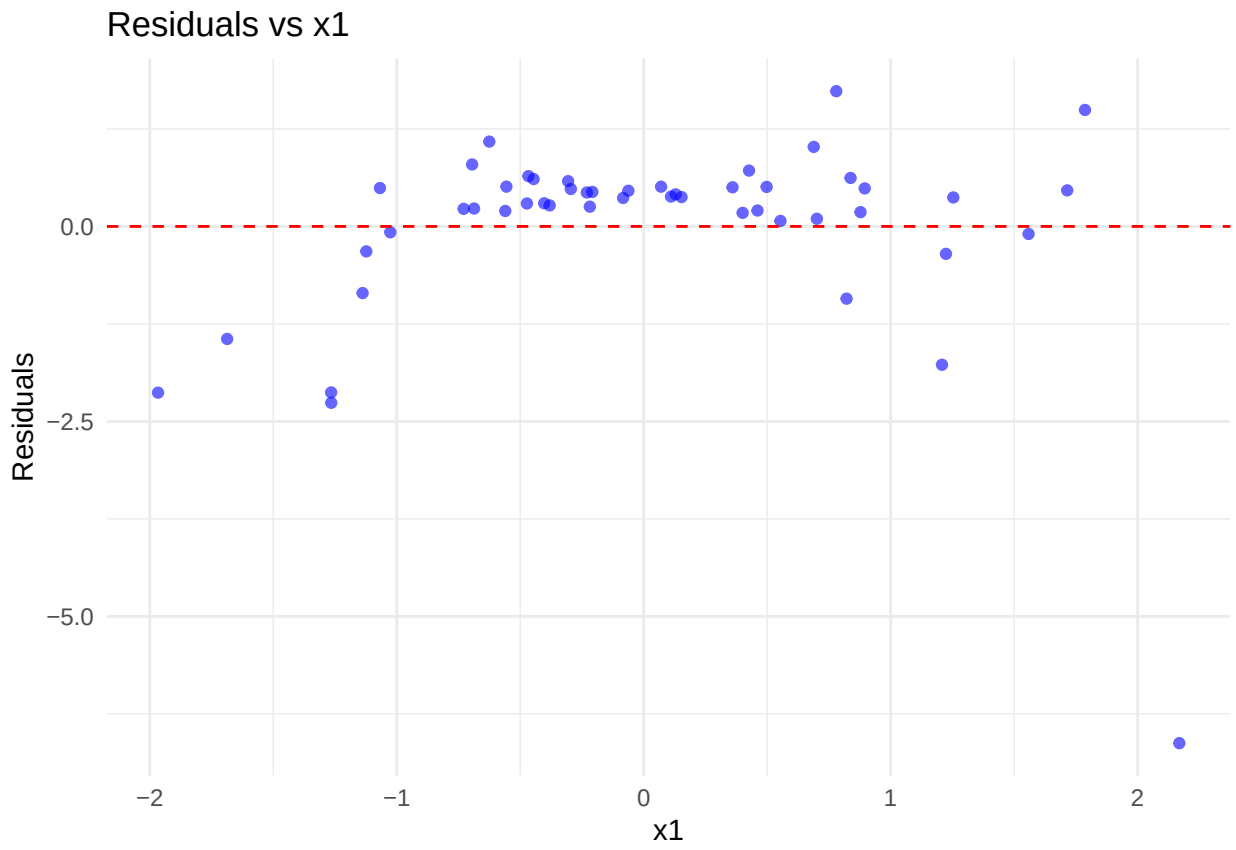
```
## x2           -1.0543     0.2048  -5.147 5.11e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.298 on 47 degrees of freedom
## Multiple R-squared:  0.9515, Adjusted R-squared:  0.9494
## F-statistic:   461 on 2 and 47 DF,  p-value: < 2.2e-16
```

```r
# Extract residuals and fitted values
residuals <- resid(model)
fitted_values <- fitted(model)

data_plot <- data.frame(x1 = x1, x2 = x2, Residuals = residuals)

# Create the residuals vs fitted values plot
ggplot(data = data.frame(Fitted = fitted_values, Residuals = residuals),
       aes(x = Fitted, y = Residuals)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Fitted Values", x = "Fitted Values", y = "Residuals") +
  theme_minimal()
```

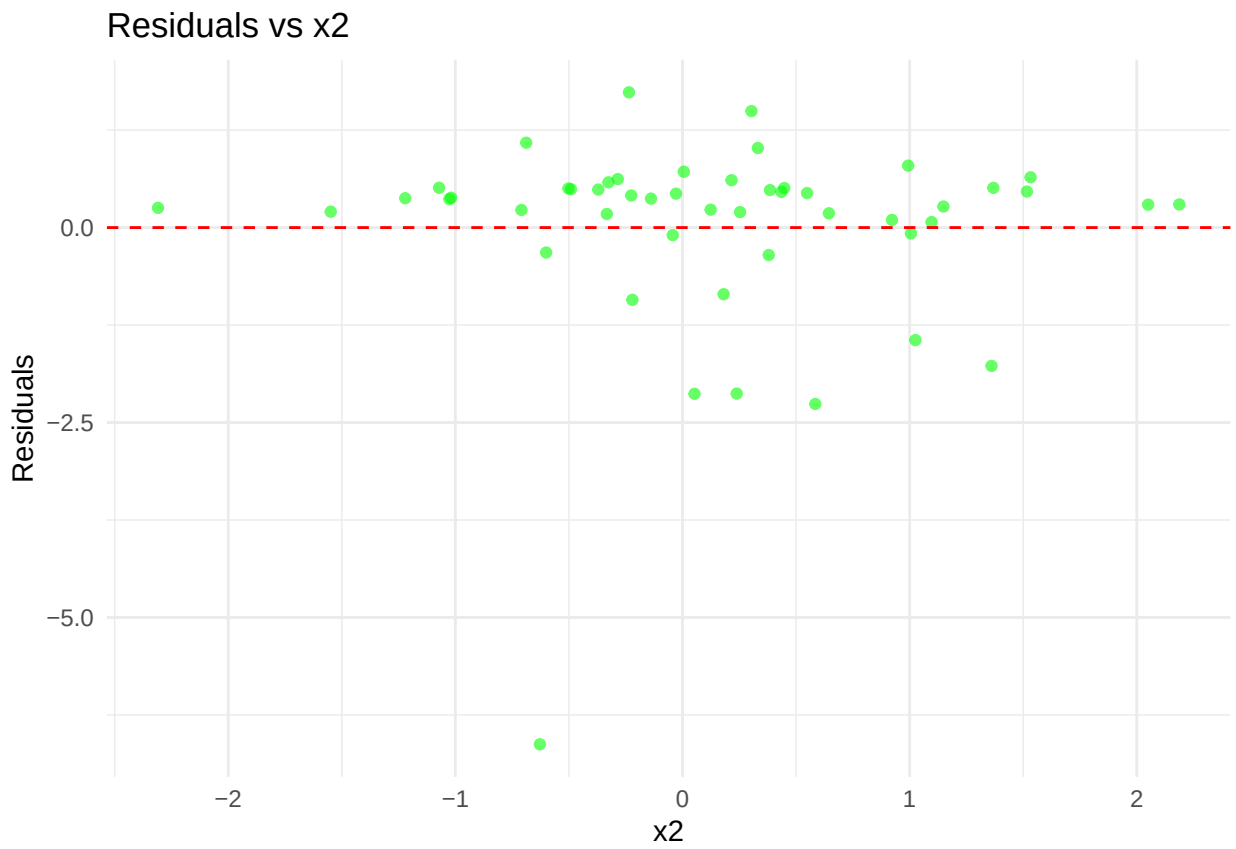



```r
# Plot residuals vs x1
ggplot(data_plot, aes(x = x1, y = Residuals)) +
  geom_point(color = "blue", alpha = 0.6) +
```

```
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs x1", x = "x1", y = "Residuals") +
  theme_minimal()
```




```
# Plot residuals vs x2
ggplot(data_plot, aes(x = x2, y = Residuals)) +
  geom_point(color = "green", alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs x2", x = "x2", y = "Residuals") +
  theme_minimal()
```



This graph does not look as spread out as the Residuals vs Fitted, you could almost see a quadratic shape forming but that does not mean it fully invalidates the equal variance assumption. Residual with x2 seems to be less patterned compared to the rest so I would say it has a higher equality of variance compared to the others. Residuals vs x2 seems to have better equal variance. I would like to believe the x^2 used for y affects the shape and makes it a little quadratic which affects the model. It might mean that for some values the linear estimate is not ver accurate