Day 2: Regression and ANOVA Exercise

```r
knitr::opts_chunk$set(warning = FALSE)
knitr::opts_chunk$set(message = FALSE)
```

```r
library(tidyverse)
```

Part 1

```r
set.seed(100)
n <- 10000

# True Regression Coefficients
beta_0 <- 3
beta_1 <- 2
beta_2 <- 7.33
beta_3 <- 5
beta_4 <- 1.25
beta_5 <- 0
sigma2 <- 6

# Predictor Variables
x0 <- rep(1, n)
x1 <- rexp(n, rate = 2)
x2 <- rgeom(n, prob = 0.2)
x3 <- rnorm(n, mean = -2, sd = 5)
x4 <- rpois(n, 3)
x5 <- rbinom(n, 20, 0.5)

pred_mat <- data.frame(x0, x1, x2, x3, x4, x5) %>%
  as.matrix(ncol = 6)
```

```r
# Error Terms
epsilon <- rnorm(n, sd = sqrt(sigma2))
epsilon2 <- rnorm(n, sd = (2*x2 + 1))
epsilon3 <- rgamma(n, 2, 1)

y <- beta_0 +
  (beta_1 * x1) +
  (beta_2 * x2) +
  (beta_3 * x3) +
  (beta_4 * x4) +
  (beta_5 * x5) +
  epsilon

y2 <- beta_0 +
  (beta_1 * x1) +
  (beta_2 * x2) +
  (beta_3 * x3) +
  (beta_4 * x4) +
  (beta_5 * x5) +
  epsilon2

y3 <- beta_0 +
  (beta_1 * x1) +
  (beta_2 * x2) +
  (beta_3 * x3) +
  (beta_4 * x4) +
  (beta_5 * x5) +
  epsilon3

df <- tibble(x1, x2, x3, x4, x5, y)
df2 <- tibble(x1, x2, x3, x4, x5, y2)
df3 <- tibble(x1, x2, x3, x4, x5, y3)
```

First, run the code above to generate the three simulated datasets that we will be using for this exercise. Note that for every dataset, all the true regression coefficients used to generate the data are nonzero except for `beta_5`.

Fit a regression model on `df` using `x1`, `x2`, and `x3` as the predictor variables and run the appropriate regression diagnostics. Which OLS assumption is violated here? How do you think this affects our model estimates?

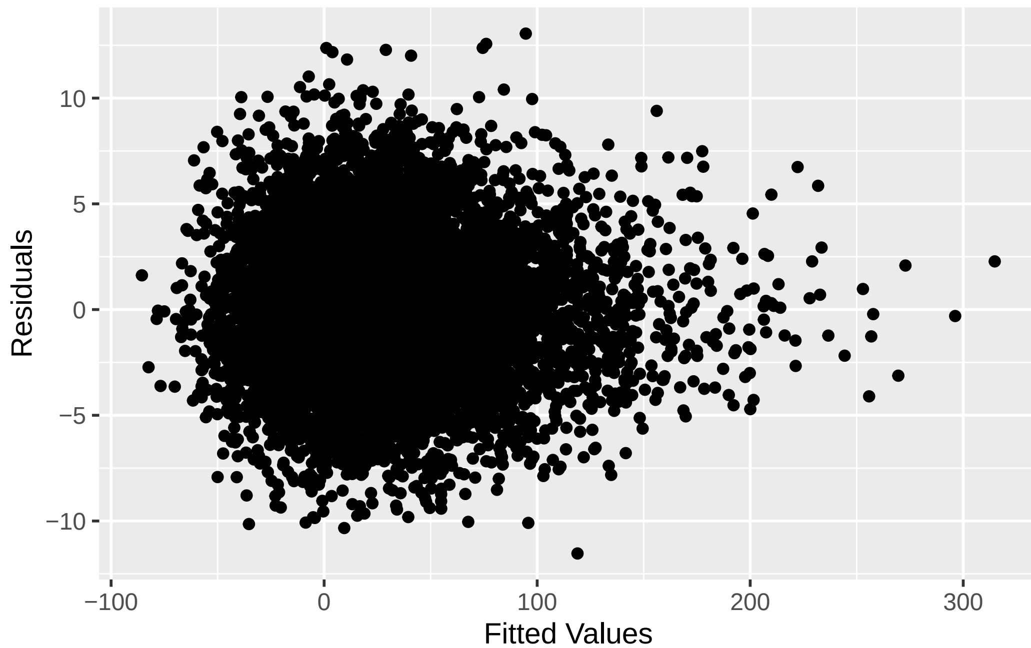
```r
lm1 <- lm(y ~ x1 + x2 + x3, data = df)
```

```r
summary(lm1)
```

```

Call:
lm(formula = y ~ x1 + x2 + x3, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-11.535  -2.201  -0.107   2.154  13.052 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  6.74395    0.05600  120.43   <2e-16 ***
x1           1.98222    0.06405   30.95   <2e-16 ***
x2           7.33433    0.00720 1018.60   <2e-16 ***
x3           5.00049    0.00651  768.18   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.266 on 9996 degrees of freedom
Multiple R-squared:  0.994,	Adjusted R-squared:  0.994 
F-statistic: 5.485e+05 on 3 and 9996 DF,  p-value: < 2.2e-16
```

Based on the fact that all predictor variables have true nonzero coefficients, this model is an example of underfitting.

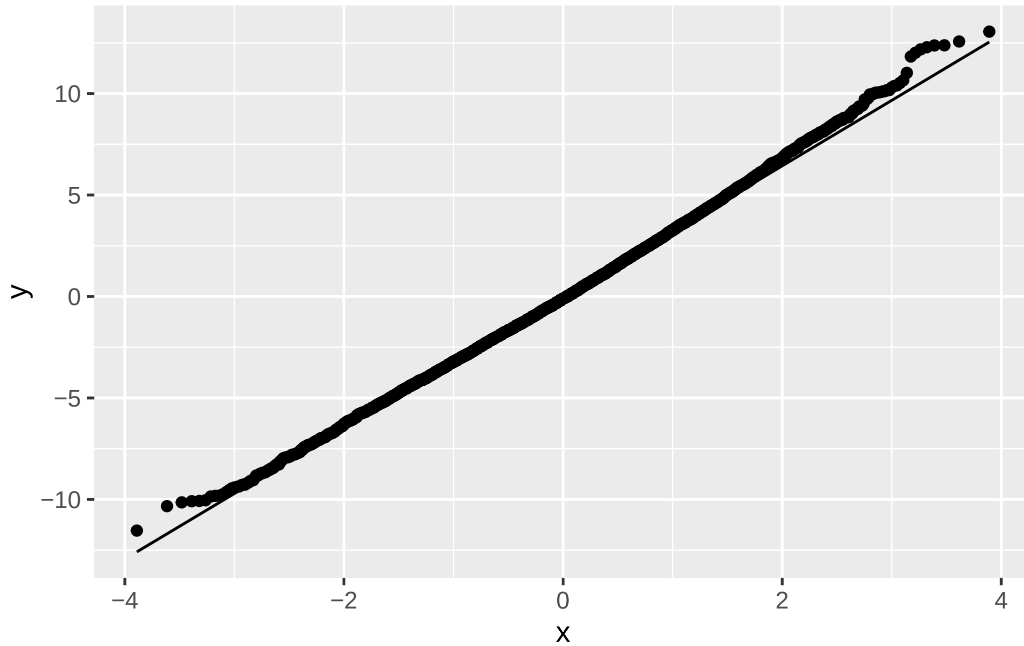
```r
ggplot() +
  geom_point(aes(x = lm1$fitted.values, y = lm1$residuals)) +
  xlab("Fitted Values") +
  ylab("Residuals")
```



The residual plot above does not indicate any critical concerns. The residuals at larger fitted values appear to have less variation around 0 than at lower fitted values, though this may be due to the presence of a smaller number of observations at this level of fitted values.

Additionally, the Q-Q plot below also seems to show that the residuals are relatively normally distributed, with slight deviations at the tails.

```
ggplot() +
  geom_qq(aes(sample = lm1$residuals)) +
  geom_qq_line(aes(sample = lm1$residuals))
```



Underfitting suggests that we would see biased coefficient estimates and a biased estimate of the variance of the error terms. Looking at the true coefficients from the simulation code, the coefficient estimates appear to be very accurate. However, we can see that the estimated residual standard error of 3.266, when squared, suggests that the estimated variance of the error terms is equal to 10.67, higher than the true value of 6.

Part 2

Fit a regression model on `df` using all possible predictor variables and run the appropriate regression diagnostics. Which OLS assumption is violated here? How do you think this affects our model estimates?

```r
lm2 <- lm(y ~ x1 + x2 + x3 + x4 + x5, data = df)
```

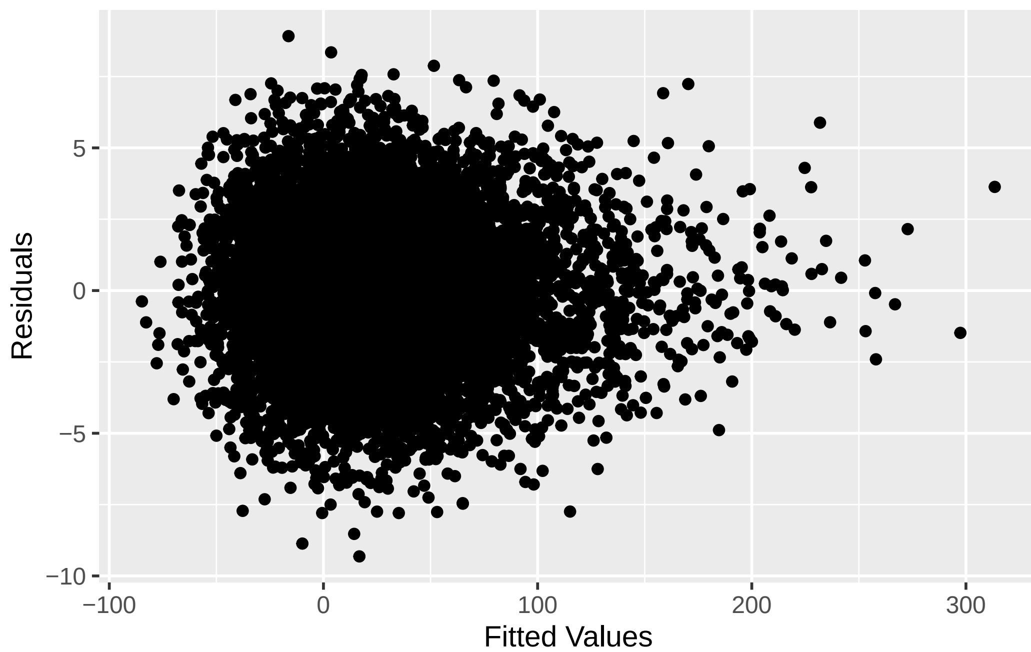
```r
summary(lm2)
```

```
Call:
lm(formula = y ~ x1 + x2 + x3 + x4 + x5, data = df)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-9.3152 -1.6423 -0.0449  1.6335  8.9158 

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)    
(Intercept) 2.9544041  0.1226706   24.084   <2e-16 ***
x1          2.0027035  0.0475017   42.161   <2e-16 ***
x2          7.3312096  0.0053393 1373.058   <2e-16 ***
x3          4.9913133  0.0048279 1033.841   <2e-16 ***
x4          1.2666792  0.0140001   90.477   <2e-16 ***
x5          0.0008448  0.0106403    0.079    0.937    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.422 on 9994 degrees of freedom
Multiple R-squared:  0.9967,    Adjusted R-squared:  0.9967 
F-statistic: 6.002e+05 on 5 and 9994 DF,  p-value: < 2.2e-16
```

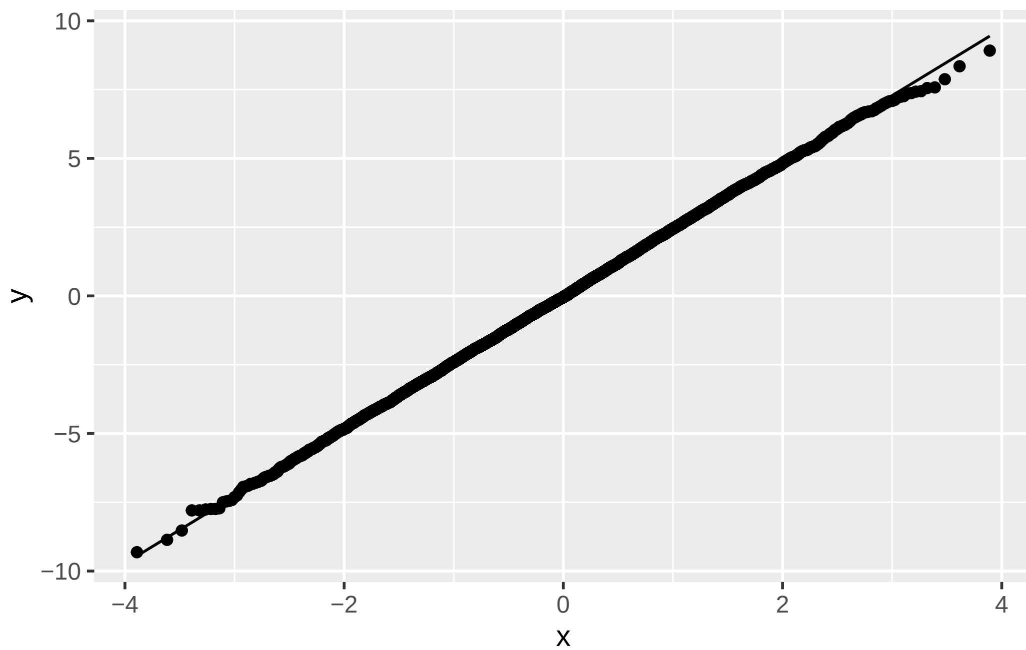
Because this model includes the `x5` variable, which we know truly has a coefficient equal to 0, this model is overfit. We expect the estimated coefficients to be unbiased, and the model summary shows that they are very close to the true coefficients. Additionally, squaring the residual standard error shows an estimated variance of 5.87, which is close to the true variance of 6.

```r
ggplot() +
  geom_point(aes(x = lm2$fitted.values, y = lm2$residuals)) +
  xlab("Fitted Values") +
  ylab("Residuals")
```



The residual plot above again shows little indication of problems with the model fit, and the Q-Q plot below also shows that the residuals are also normally distributed.

```
ggplot() +
  geom_qq(aes(sample = lm2$residuals)) +
  geom_qq_line(aes(sample = lm2$residuals))
```



Somewhat surprisingly for an overfit model, the estimated standard errors for the coefficients also align closely to the values seen in the true coefficient covariance matrix. It is possible that this would change with a smaller sample size.

```
sigma2 * solve(t(pred_mat) %*% pred_mat)
```

```
               x0            x1            x2            x3            x4
x0  1.539436e-02 -1.276229e-03 -1.221421e-04  5.016711e-05 -6.157183e-04
x1 -1.276229e-03  2.308335e-03  1.985989e-06 -4.205199e-07  3.371634e-06
x2 -1.221421e-04  1.985989e-06  2.916449e-05 -2.831522e-07 -4.864862e-07
x3  5.016711e-05 -4.205199e-07 -2.831522e-07  2.384530e-05 -1.447150e-06
x4 -6.157183e-04  3.371634e-06 -4.864862e-07 -1.447150e-06  2.005127e-04
x5 -1.171232e-03  9.478909e-06  5.057887e-07  3.426903e-07  1.707012e-06
               x5
x0 -1.171232e-03
x1  9.478909e-06
x2  5.057887e-07
x3  3.426903e-07
x4  1.707012e-06
x5  1.158204e-04
```

Part 3

Fit a regression model on `df2` using all predictor variables except `x5`, and run the appropriate regression diagnostics. Which OLS assumption is violated? How does this affect our model estimates?

```
lm3 <- lm(y2 ~ x1 + x2 + x3 + x4, data = df2)
```

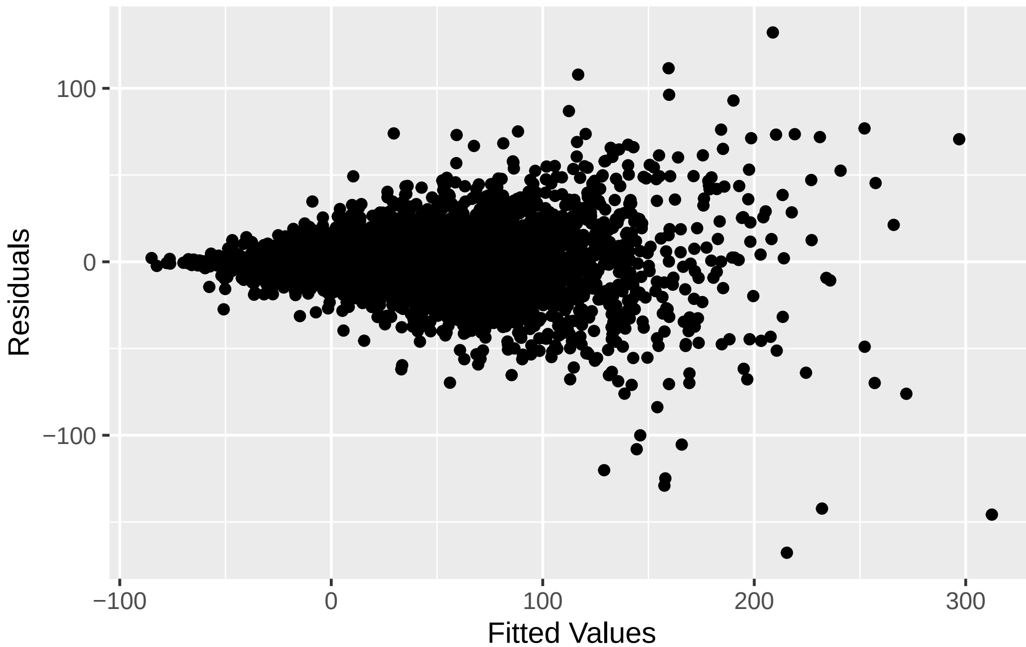
```
summary(lm3)
```

```
Call:
lm(formula = y2 ~ x1 + x2 + x3 + x4, data = df2)

Residuals:
     Min       1Q   Median       3Q      Max
-167.733   -3.339   -0.057    3.352  132.201

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.61539    0.31289   8.359   <2e-16 ***
x1           2.08112    0.25225   8.250   <2e-16 ***
x2           7.30736    0.02836 257.687   <2e-16 ***
x3           4.98937    0.02564 194.579   <2e-16 ***
x4           1.38471    0.07435  18.623   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.86 on 9995 degrees of freedom
Multiple R-squared:  0.9138,    Adjusted R-squared:  0.9138
F-statistic: 2.65e+04 on 4 and 9995 DF,  p-value: < 2.2e-16
```

Examining the model summary, we see that again, the estimated coefficients are very close to the true values used in the data generation process. However, there are signs of higher error in the process, with the intercept estimate being further from the true value than in the previous model, and much higher standard errors than in the previous models. We can also see that the estimated residual standard error of 12.86 is much higher than what we observed in the other models.

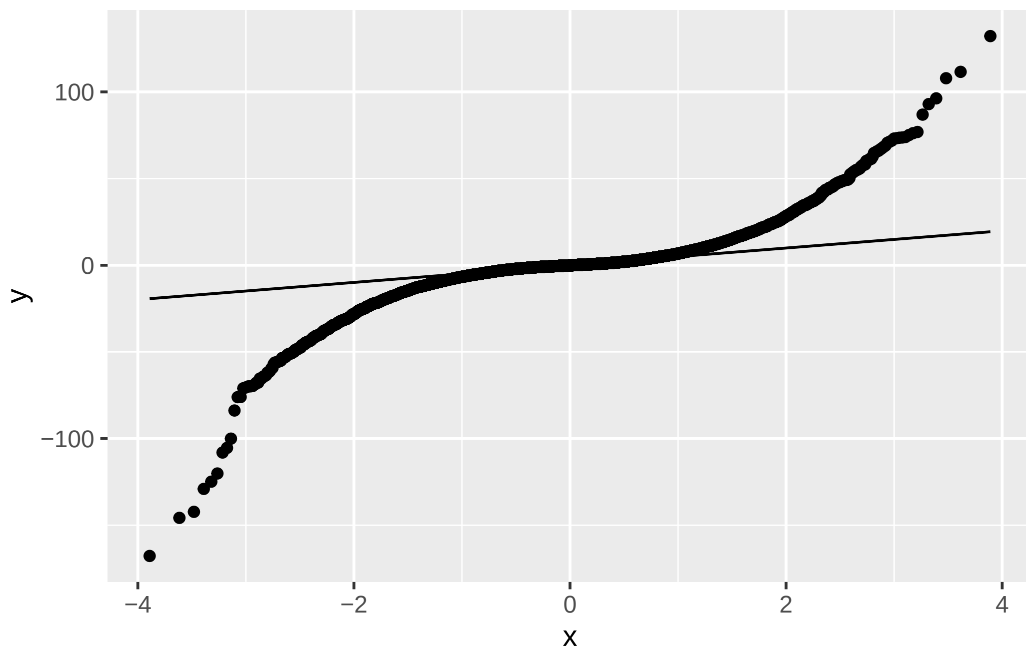
```
ggplot() +
  geom_point(aes(x = lm3$fitted.values, y = lm3$residuals)) +
```

```r
  xlab("Fitted Values") +
  ylab("Residuals")
```



The residual plot above shows the reason for this discrepancy: where in other models, the residuals generally showed less variation at higher fitted values, in this case the variation in the residuals increases dramatically as the fitted values rise. This tells us that homoscedasticity is violated, and we can expect to have biased variance estimates.

```r
ggplot() +
  geom_qq(aes(sample = lm3$residuals)) +
  geom_qq_line(aes(sample = lm3$residuals))
```



From the Q-Q plot above, we can see a departure from normality.

Part 4

Fit a regression model on `df3` using all predictor variables except `x5`, and run the appropriate regression diagnostics. Which OLS assumption is violated? How does this affect our model estimates?

```
lm4 <- lm(y3 ~ x1 + x2 + x3 + x4, data = df3)
```

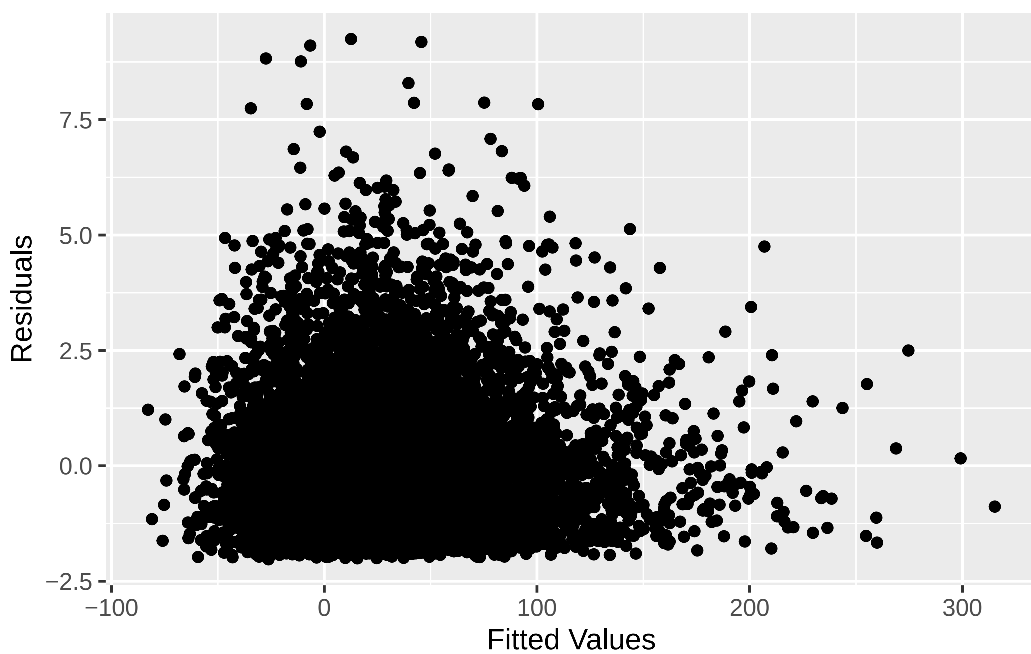
```
summary(lm4)
```

```
Call:
lm(formula = y3 ~ x1 + x2 + x3 + x4, data = df3)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.0257 -1.0461 -0.3255  0.6869  9.2475 
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.054685   0.034667  145.81   <2e-16 ***
x1          1.969312   0.027949   70.46   <2e-16 ***
x2          7.326882   0.003142 2331.96   <2e-16 ***
x3          4.999625   0.002841 1759.78   <2e-16 ***
x4          1.244230   0.008238  151.03   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.425 on 9995 degrees of freedom
Multiple R-squared:  0.9988,    Adjusted R-squared:  0.9988
F-statistic: 2.167e+06 on 4 and 9995 DF,  p-value: < 2.2e-16
```

Looking at the model summary, we can see that the estimated coefficients are very close to the true coefficients, but the estimated standard errors tend to be different from the true regression coefficient covariance matrix.

```r
ggplot() +
  geom_point(aes(x = lm4$fitted.values, y = lm4$residuals)) +
  xlab("Fitted Values") +
  ylab("Residuals")
```



The residual plot is showing an uncommon pattern, with the residual values not symmetrically distributed around 0. This is an early indication that the residuals may not be normally (or symmetrically) distributed. This is confirmed by the Q-Q plot below.

```
ggplot() +
  geom_qq(aes(sample = lm4$residuals)) +
  geom_qq_line(aes(sample = lm4$residuals))
```