Module 5: Peer Reviewed Assignment

Outline:

The objectives for this assignment:

1. Understand what can cause violations in the linear regression assumptions.
2. Enhance your skills in identifying and diagnosing violated assumptions.
3. Learn some basic methods of addressing violated assumptions.

General tips:

1. Read the questions carefully to understand what is being asked.
2. This work will be reviewed by another human, so make sure that you are clear and concise in what your explanations and answers.

In [11]:
```
# Load Required Packages
library(ggplot2)
```

Problem 1: Let's Violate Some Assumptions!

When looking at a single plot, it can be difficult to discern the different assumptions being violated. In the following problem, you will simulate data that purposefully violates each of the four linear regression assumptions. Then we can observe the different diagnostic plots for each of those assumptions.

1. (a) Linearity

Generate SLR data that violates the linearity assumption, but maintains the other assumptions. Create a scatterplot for these data using ggplot.

Then fit a linear model to these data and comment on where you can diagnose nonlinearity in the diagnostic plots.

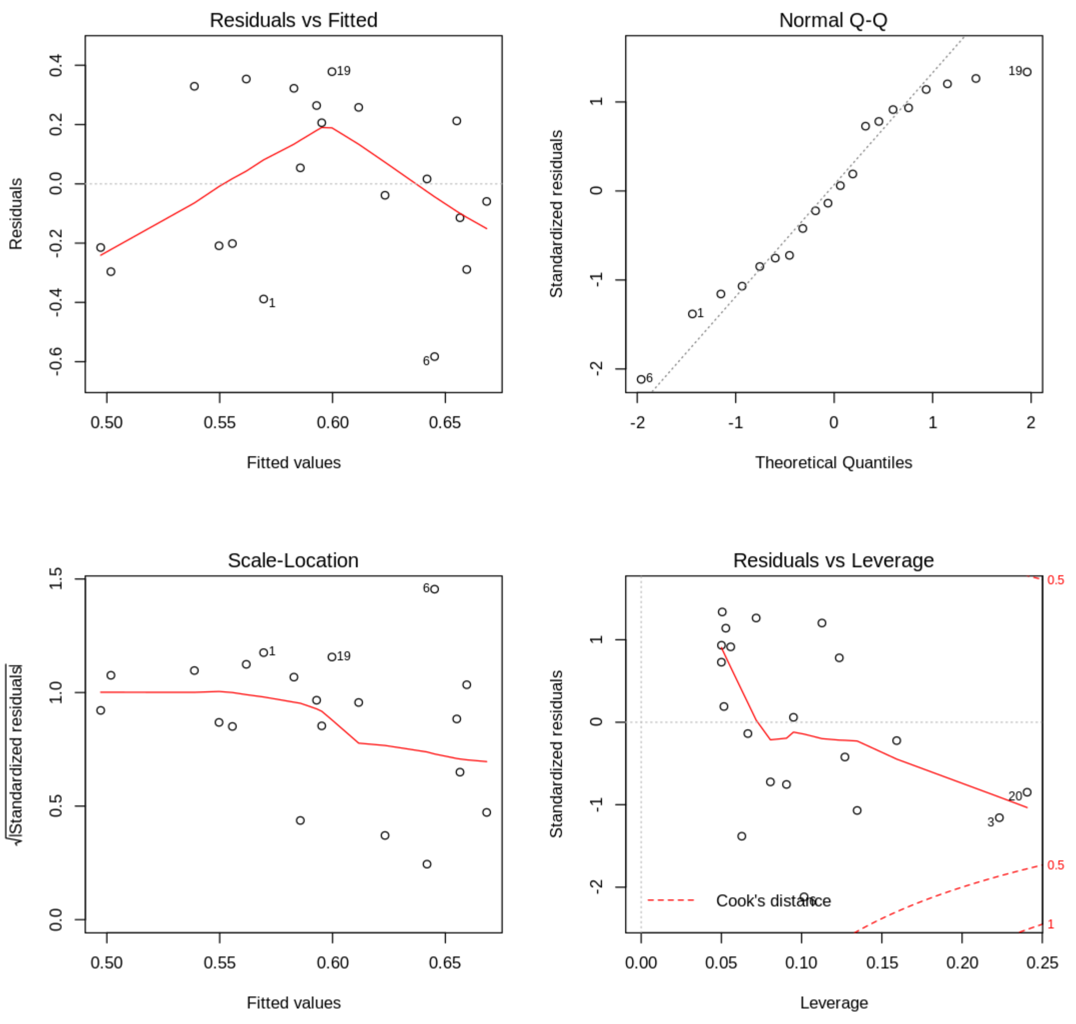
In [12]:
```
# Your Code Here
#set.seed(1123)
x1 = runif(20)
y1 = runif(20)
df = data.frame(x1, y1)
lmod = lm(y1 ~ x1, data = df)
summary(lmod)
plot(lmod)

options(repr.plot.width = 9, repr.plot.height = 9)
par(mfrow = c(2,2))
```

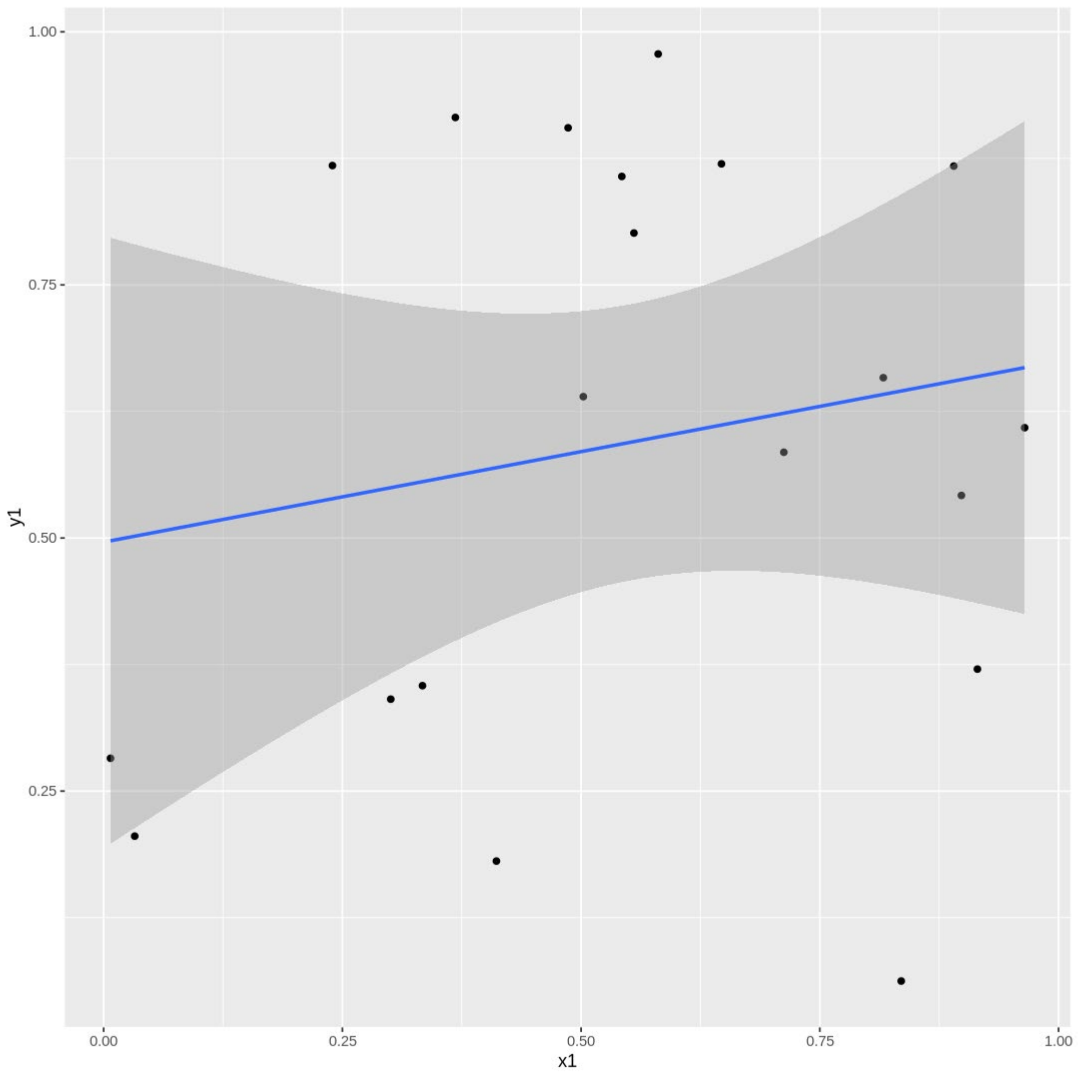
Processing math: 100%

```
ggplot(df, aes(x=x1, y=y1)) + geom_point() + geom_smooth(method='lm', se=T
RUE)
```

```
Call:
lm(formula = y1 ~ x1, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.58282 -0.21038 -0.01103  0.25954  0.37835 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)   0.4960     0.1439   3.446  0.00288 **
x1            0.1787     0.2327   0.768  0.45249   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2904 on 18 degrees of freedom
Multiple R-squared:  0.03172,	Adjusted R-squared:  -0.02207 
F-statistic: 0.5897 on 1 and 18 DF,  p-value: 0.4525
```

```
`geom_smooth()` using formula 'y ~ x'
```





In the diagnostic plots, a correctly formulated linear model should produce a flat residual vs fitted graph centered at 0, with the data points packed relatively tightly around the line. In the above diagnostic res vs fitted plot, we see that each point is relatively equally spaced from the line y=0, showing roughly equal variance for each data point. However, the fitted line itself does not show linearity. Also, the fitted plot does not imply a linear trend between the response and predictor variables.

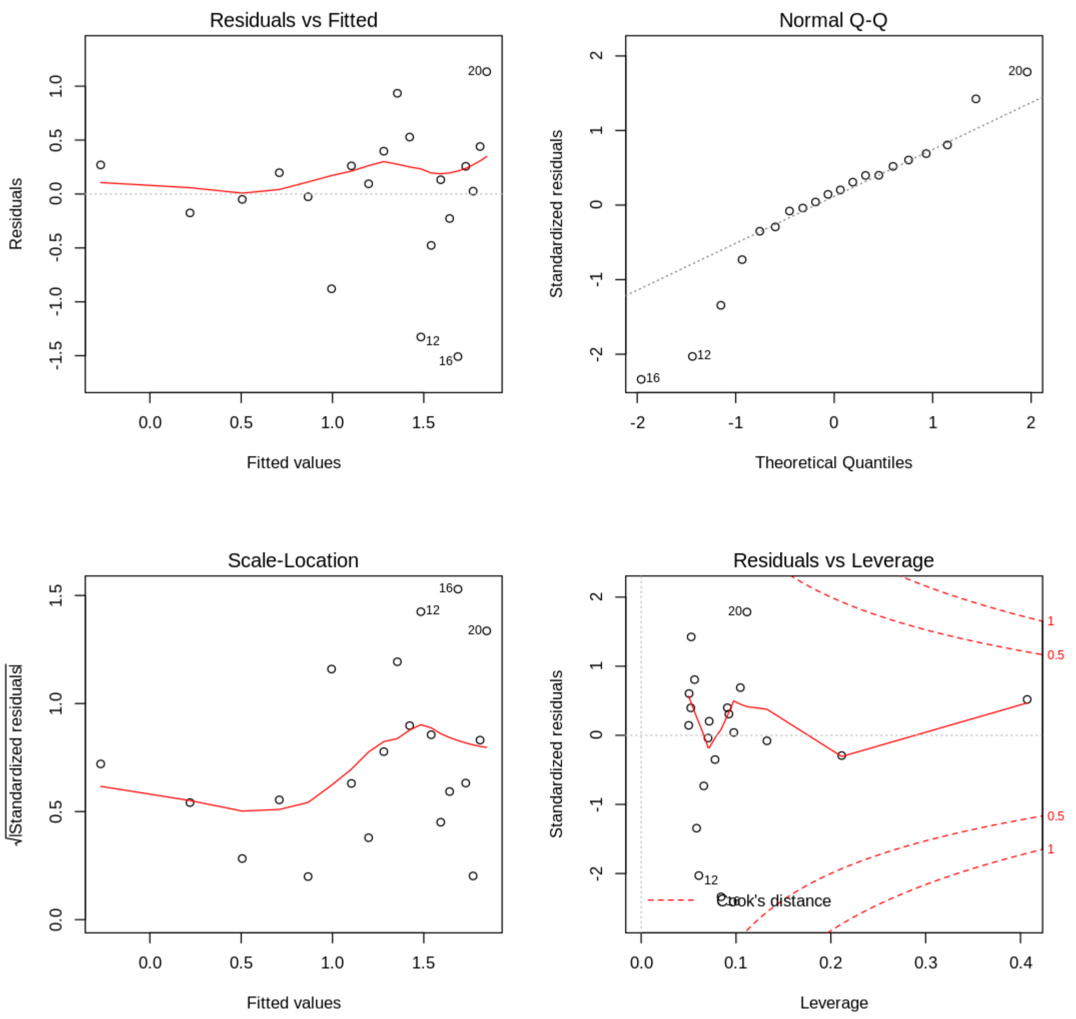
1. (b) Homoskedasticity

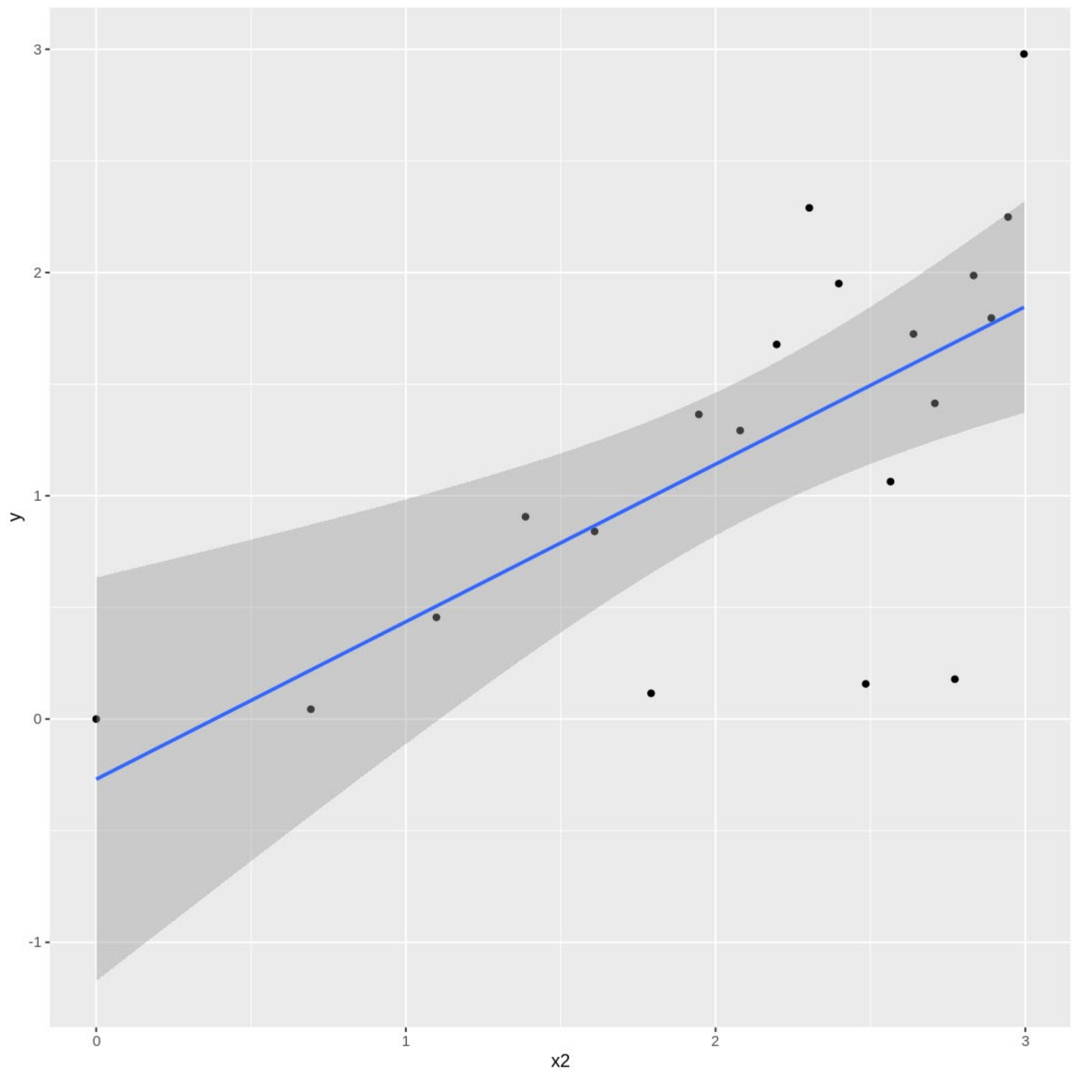
Simulate another SLR dataset that violates the constant variance assumption, but maintains the other assumptions. Then fit a linear model to these data and comment on where you can diagnose non-constant variance in the diagnostic plots.

In [21]:
```
# Your Code Here
x2 = log(seq(1, 20))
y = runif(10)*x2
df2 = data.frame(x2, y)
lmod2 = lm(y ~ x2, data = df2)
options(repr.plot.width = 9, repr.plot.height = 9)
par(mfrow = c(2,2))
plot(lmod2)
```

```
options(repr.plot.width = 9, repr.plot.height = 9)
par(mfrow = c(2,2))
ggplot(df2, aes(x=x2, y=y)) + geom_point() + geom_smooth(method='lm', se=T
RUE)
```

```
`geom_smooth()` using formula 'y ~ x'
```





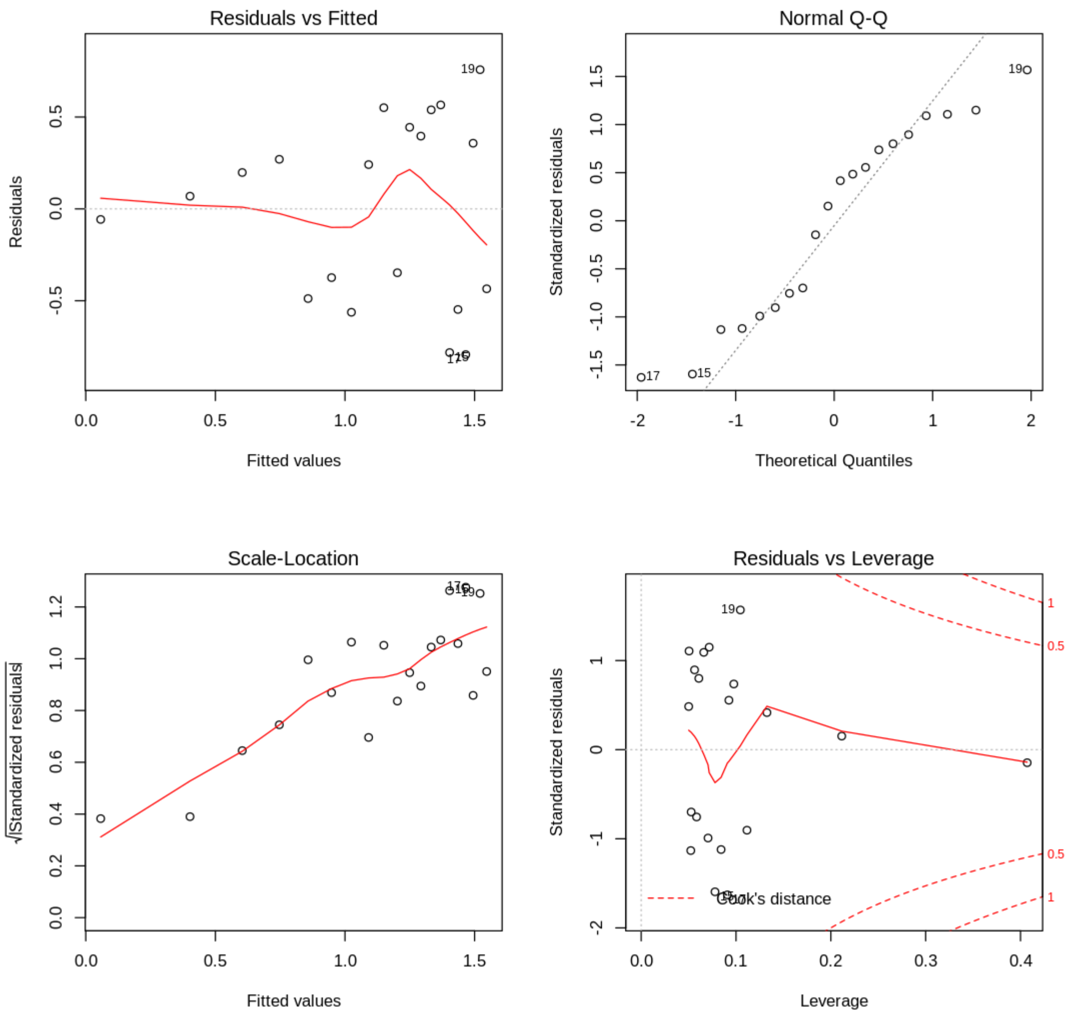
Here, we can look at the res vs fitted plot and see that the data points do not maintain a constant variance as we traverse along the x-axis. The data points spread out and show increasing variance as we move along the x-axis, showing the shape of a fan.

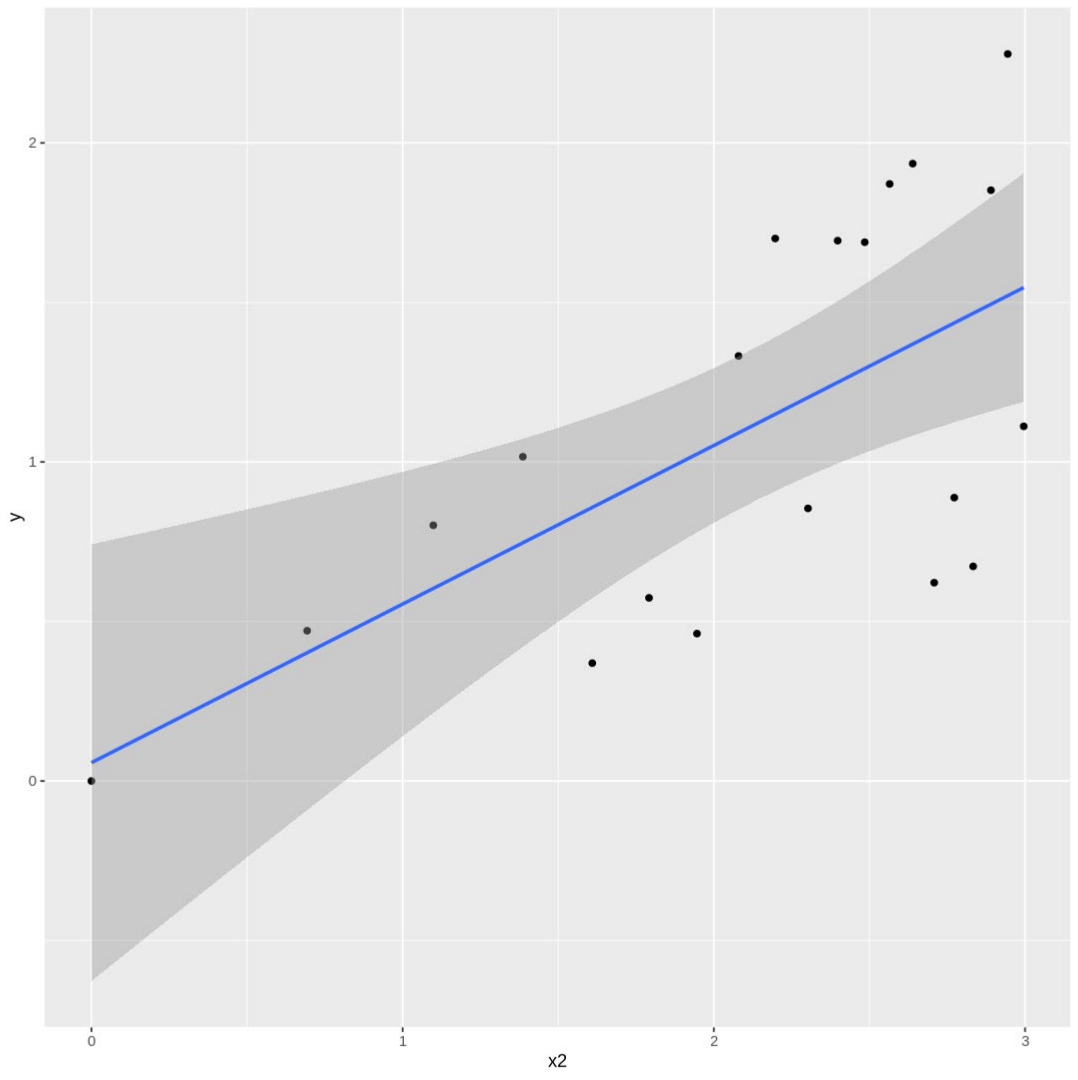
1. (c) Independent Errors

Repeat the above process with simulated data that violates the independent errors assumption.

In [22]:
```
# Your Code Here
x2 = log(seq(1, 20))
y = runif(10)*x2
df2 = data.frame(x2, y)
lmod2 = lm(y ~ x2, data = df2)
options(repr.plot.width = 9, repr.plot.height = 9)
par(mfrow = c(2,2))
plot(lmod2)
ggplot(df2, aes(x=x2, y=y)) + geom_point() + geom_smooth(method='lm', se=T
RUE)
```

```
`geom_smooth()` using formula 'y ~ x'
```


Residuals vs Fitted
Residuals
Fitted values
19
17
15
Normal Q-Q
Standardized residuals
Theoretical Quantiles
19
17
15
Scale-Location
√|Standardized residuals|
Fitted values
17
15
19
Residuals vs Leverage
Standardized residuals
Leverage
19
17
15
Cook's distance
1
0.5
0.5
1



Here we see again that data points on the res vs fitted plot are not constant. The points spread out like a fan as we move to the right, showing a non-independent relationship between the residuals and the y variables. Moreover, the Q-Q plot shows a non-linear trend as well, having a systematic over and under-fitted values with respect to the y=x line.

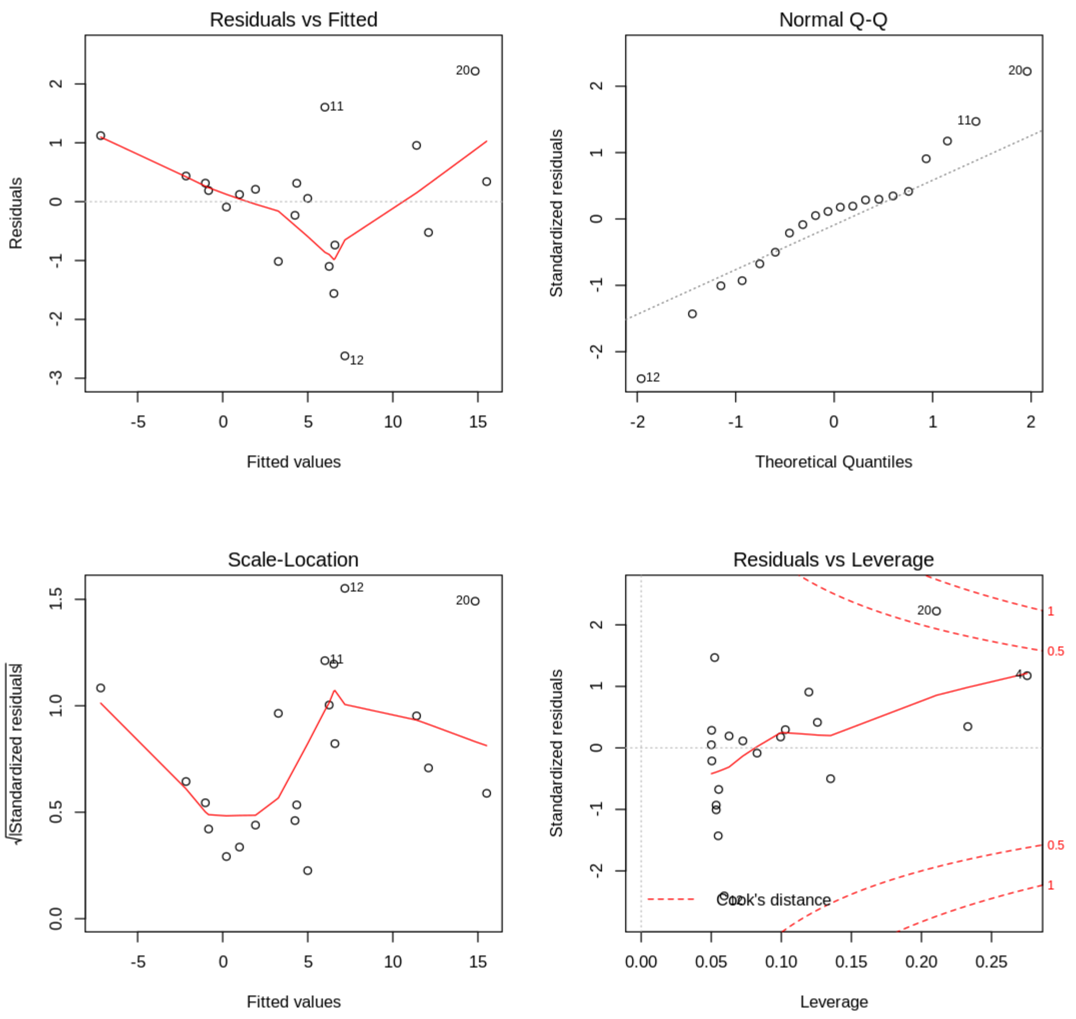
1. (d) Normally Distributed Errors

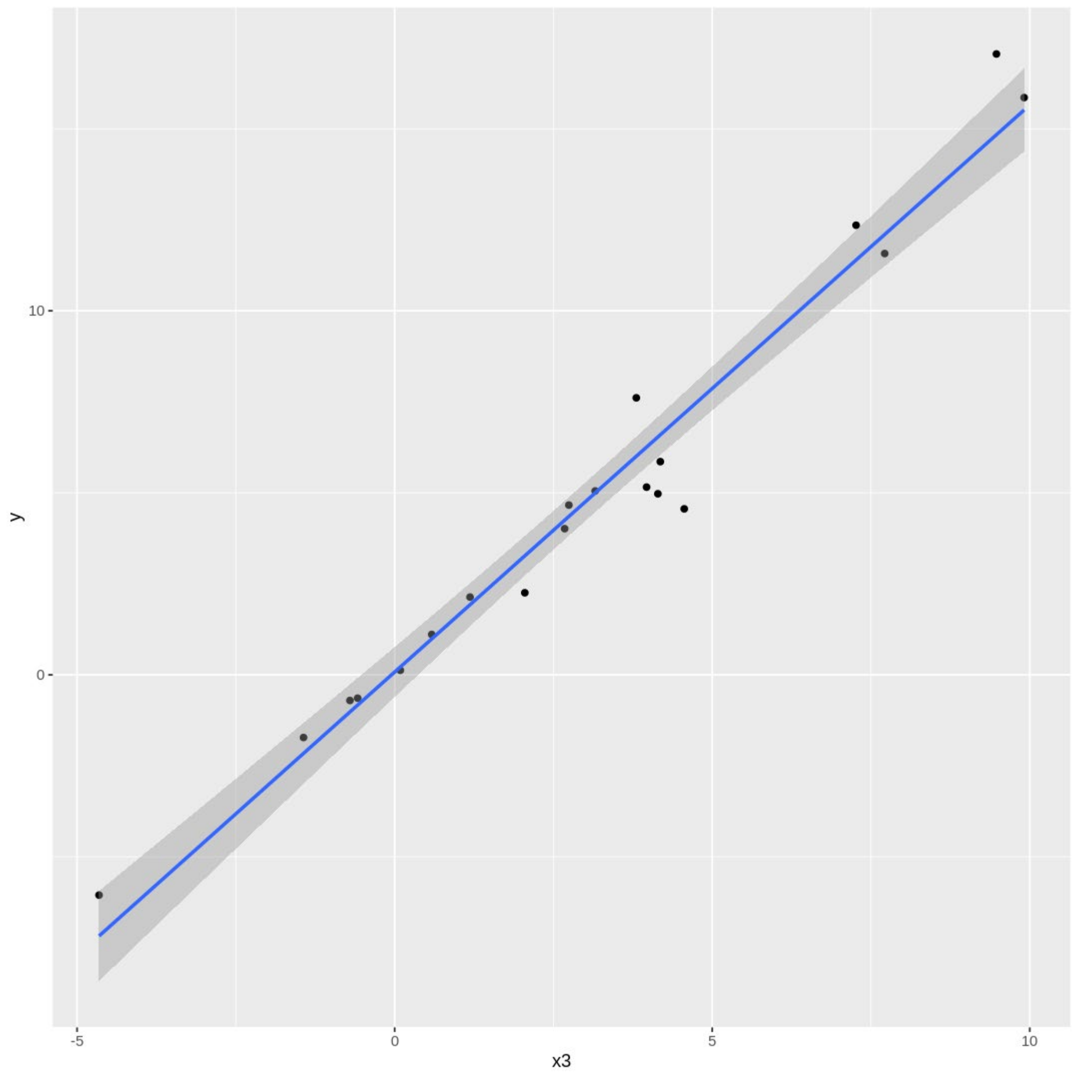
Only one more to go! Repeat the process again but simulate the data with non-normal errors.

In [39]:
```
# Your Code Here
x3 = c(runif(4, -5, 0), runif(12, 0, 5), runif(4, 6, 10))
y = seq(1, 2, 0.1) * x3
df3 = data.frame(x3, y)
lmod2 = lm(y ~ x3, data = df3)
options(repr.plot.width = 9, repr.plot.height = 9)
par(mfrow = c(2,2))
plot(lmod2)
ggplot(df3, aes(x=x3, y=y)) + geom_point() + geom_smooth(method='lm', se=T
RUE)
```

```
Warning message in seq(1, 2, 0.1) * x3:
“longer object length is not a multiple of shorter object length”
```

```
`geom_smooth()` using formula 'y ~ x'
```





Here, we see from the normal Q-Q plot that the errors show systematic deviations from the line of fit, with under-fitting and over-fitting at certain intervals. This implies that the error terms violate the normality assumption.

Problem 2: Hats for Sale

Recall that the *hat* or *projection* matrix is defined as

$$H = X(X^TX)^{-1}X^T.$$

The goal of this question is to use the hat matrix to prove that the fitted values, $\hat{\mathbf{Y}}$, and the residuals, $\hat{\boldsymbol{\varepsilon}}$, are uncorrelated. It's a bit of a process, so we will do it in steps.

2. (a) Show that $\hat{Y} = HY$. That is, H "puts a hat on" Y.

Given that $H = X(X^TX)^{-1}X^T$ and $Y = X\beta + \varepsilon$,

$$HY = X(X^TX)^{-1}X^TY$$

Since our ordinary least squares estimator is $\hat{\beta} = (X^T X)^{-1} X^T Y$, we can substitute and simplify:

$$HY = X\hat{\beta}$$

$$= \hat{Y}$$

2. (b) Show that H is symmetric: $H = H^T$.

Let's take the transpose of $H = X(X^T X)^{-1} X^T$ to prove symmetry:

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

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

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

$$= X(X^T X)^{-1} X^T$$

2. (c) Show that $H(I_n - H) = 0_n$, where 0_n is the zero matrix of size $n \times n$.**

A property of the hat matrix H is that it is indempotent. That is, the matrix multiplied by itself is equal to it self. So, $H(H) = H$.

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

$$= X(X^T X)^{-1} (X^T X)(X^T X)^{-1} X^T$$

We note that the middle terms $(X^T X)(X^T X)^{-1} = I$. Substitute and simplify:

$$= X(X^T X)^{-1} I X^T$$

$$= X(X^T X)^{-1} X^T$$

Given this property of the hat matrix, we can easily see that

$$H(I_n - H)$$

$$= HI_n - HH$$

$$= H - H$$

$$= 0_n$$

2. (d) Stating that $\hat{\mathbf{Y}}$ is uncorrelated with $\hat{\boldsymbol{\varepsilon}}$ is equivalent to showing that these vectors are orthogonal.* That is, we want their dot product to equal zero:

$$\hat{\mathbf{Y}}^T \hat{\boldsymbol{\varepsilon}} = 0.$$

Prove this result. Also explain why being uncorrelated, in this case, is equivalent to the being orthogonal.

Firstly, we have $\hat{\mathbf{Y}} = HY$. For the residuals,

$$\hat{\varepsilon} = Y - \hat{Y}$$

$$= Y - HY$$

$$= (I - H)Y$$

We substitute these values into $\hat{\mathbf{Y}}^T\hat{\boldsymbol{\varepsilon}} = 0$:

$$\hat{\mathbf{Y}}^T\hat{\boldsymbol{\varepsilon}}$$

$$= Y^T H^T (I - H)Y$$

From 2(b) we've shown that the hat matrix H is symmetric: $H = H^T$.
We've also shown that H is indempotent: $HH = H$. We can use these properties to simplify the middle term:

$$H^T(I - H)$$

$$= H^T I - H^T H$$

$$= H^T - H^T$$

$$= 0$$

Thus,

$$\hat{\mathbf{Y}}^T\hat{\boldsymbol{\varepsilon}} = 0.$$

By definition, uncorrelated means that change in one variable leads to, on average, no change on the other variable. Mathematically, two variables are uncorrelated if their covariance is 0: $cov[X, Y] = E[XY] - E[X][Y] = 0$. This implies that vectors representing each variables will show orthogonality when projected onto a plane. When following along the axis of one vector, the magnitude can vary without changing its component along the y-axis. On the other hand, varying the magnitude of a vector not along either axes will change both x and y components to vary. Thus, uncorrelation implies orthogonality.

2.(e) Why is this result important in the practical use of linear regression?

In the example of a sine/cosine function, the x (time) and y values are uncorrelated and are orthogonal to each other. However, having information of time enables us to make predictions of y. A non-zero correlation and orthogonality in this sense would affect our ability to make precise predictions, so they play a crucial part in the practical use of linear regression.

Problem 3: Model Diagnosis

We here at the University of Colorado's Department of Applied Math love Bollywood movies. So, let's analyze some data related to them!

We want to determine if there is a linear relation between the amount of money spent on a movie (it's budget) and the amount of money the movie makes. Any venture capitalists among you will certianly hope that there is at least some relation. So let's get to modelling!

3. (a) Initial Inspection

Load in the data from local directory and create a linear model with `Gross` as the response and `Budget` as the feature. The data is stored in the same local directory and is called `bollywood_boxoffice.csv`. Thank the University of Florida for this specific dataset.

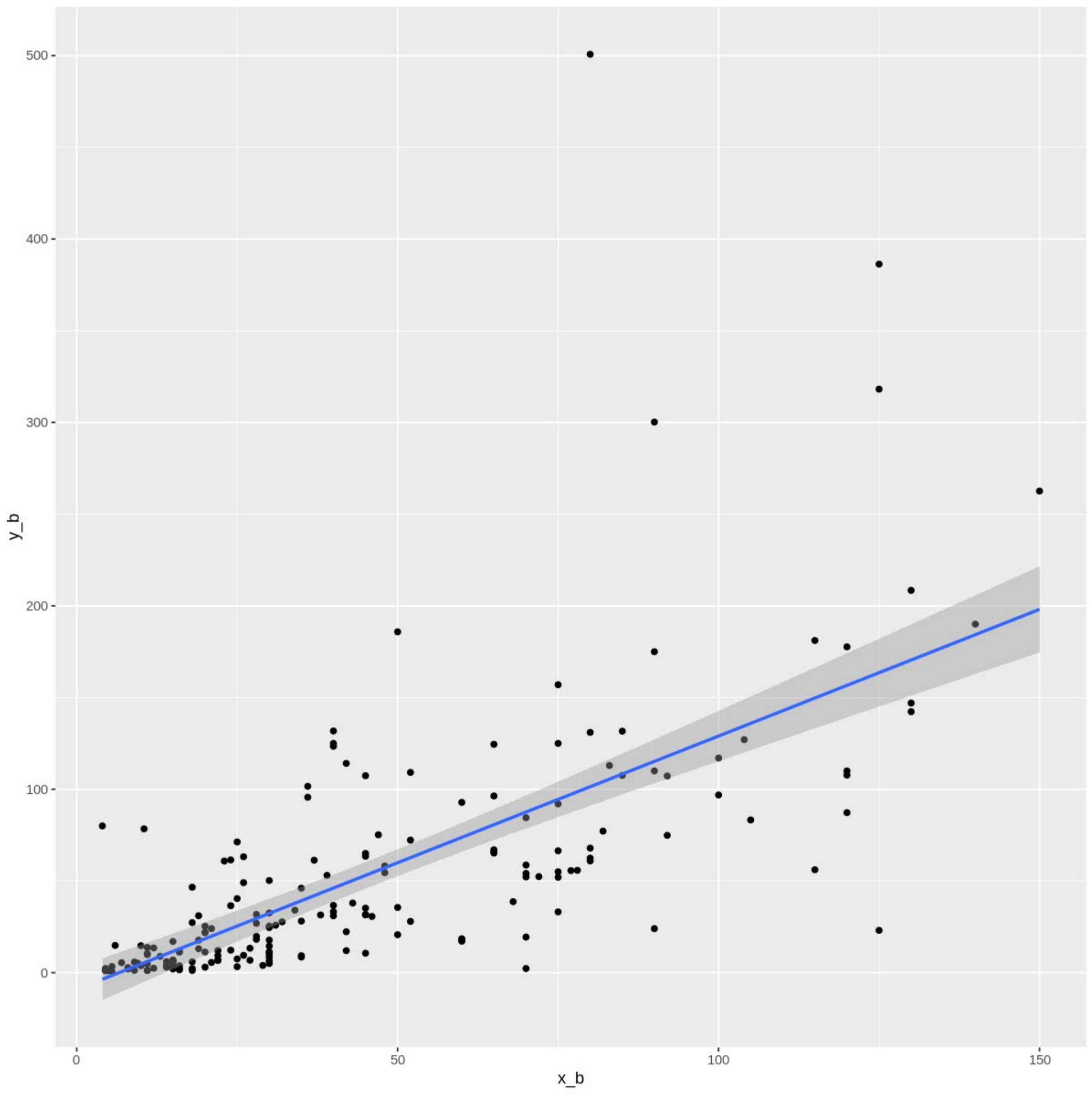
Specify whether each of the four regression model assumptions are being violated.

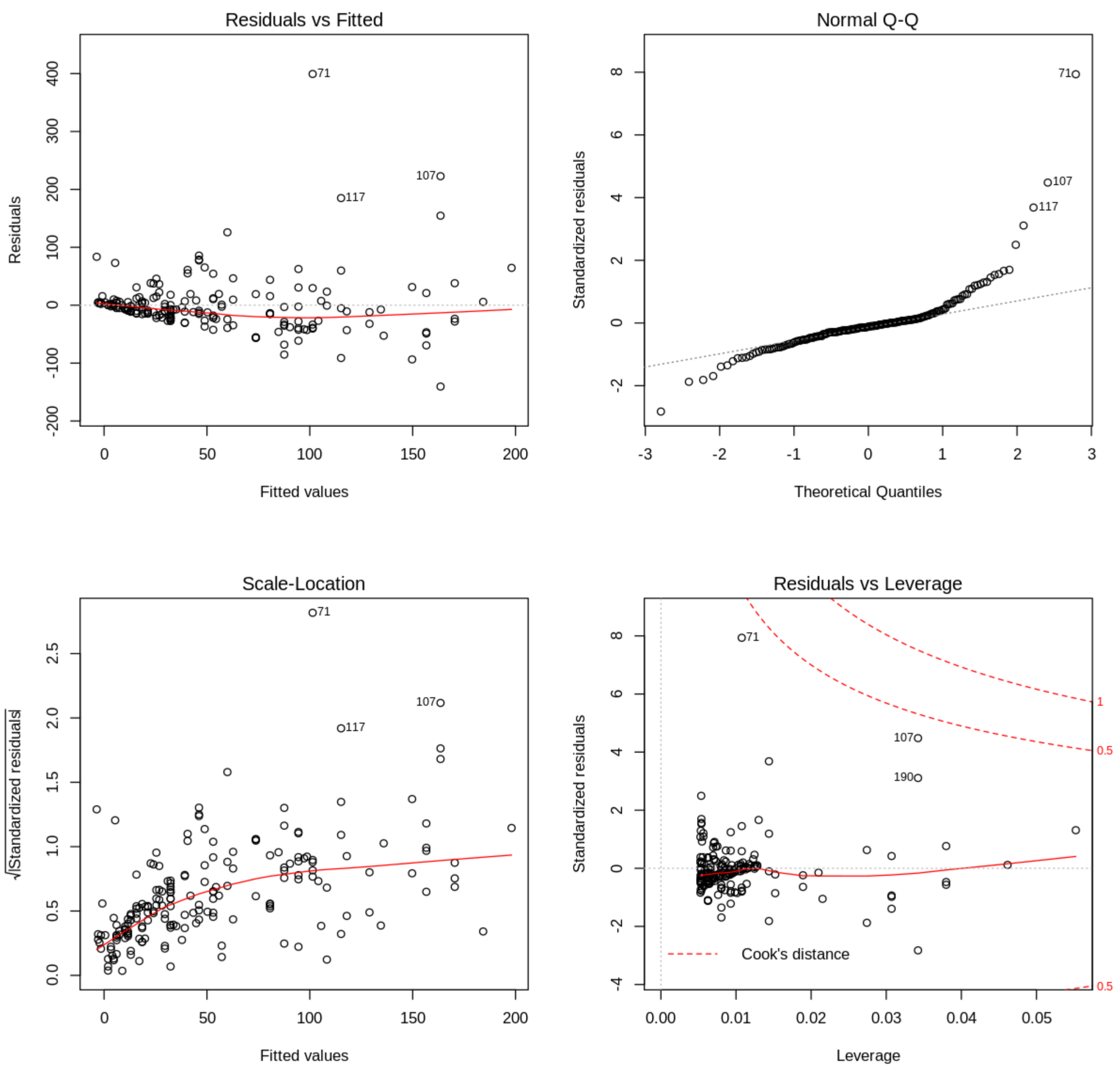
Data Source: http://www.bollymoviereviewz.com

In [42]:
```
# Load the data
bollywood = read.csv("bollywood_boxoffice.csv")
summary(bollywood)

# Your Code Here
x_b = bollywood$Budget
y_b = bollywood$Gross
df = data.frame(x_b, y_b)
lmod = lm(Gross ~ Budget, data = bollywood)
options(repr.plot.width = 10, repr.plot.height = 10)
par(mfrow = c(2,2))
ggplot(df, aes(x=x_b, y=y_b)) + geom_point() + geom_smooth(method='lm', se
=TRUE)
plot(lmod)
```

```
                   Movie          Gross             Budget
 1920London           :  1   Min.   :  0.63   Min.   :  4.00
 2 States\xa0         :  1   1st Qu.:  9.25   1st Qu.: 19.00
 24(Tamil,Telugu)     :  1   Median : 29.38   Median : 34.50
 Aashiqui 2           :  1   Mean   : 53.39   Mean   : 45.25
 AeDilHainMushkil\xa0:  1   3rd Qu.: 70.42   3rd Qu.: 70.00
 AGentleman           :  1   Max.   :500.75   Max.   :150.00
 (Other)              :184
```

```
`geom_smooth()` using formula 'y ~ x'
```


500
400
300
200
100
0
y_b
0
50
100
150
x_b



Looking at the scatterplot, the first linearity assumption is not met. The points are densely packed around lower values of x, spreading out as x increases. The constant variance assumption is also violated as shown in the res vs fitted plot, where the points show a similar pattern to the ggplot. This also leads to a violation of homoscedasticity. Lastly, the normal Q-Q plot shows that the error terms have a normal distriubtion.

3. (b) Transformations

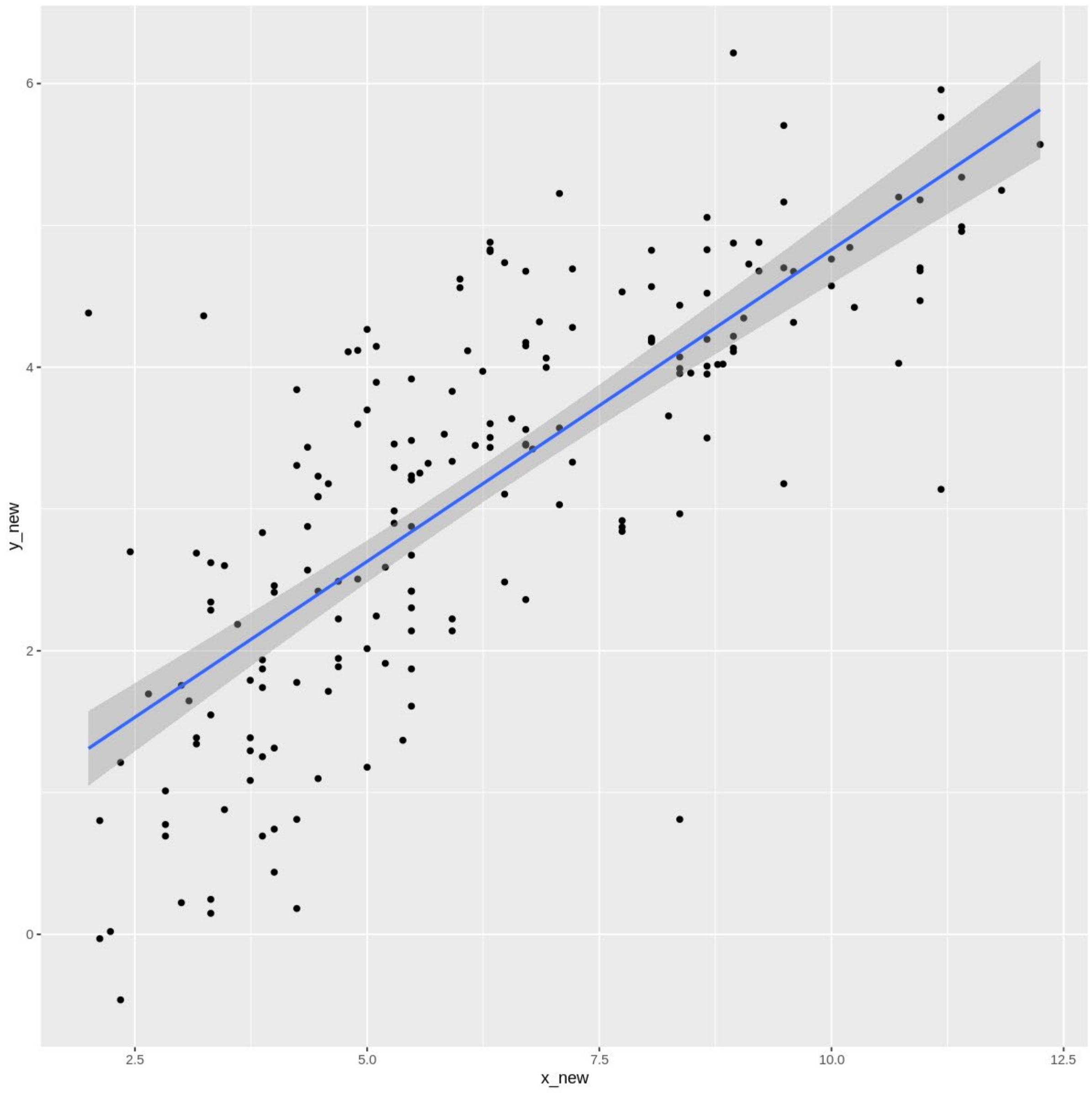
Notice that the Residuals vs. Fitted Values plot has a 'trumpet" shape to it, the points have a greater spread as the Fitted value increases. This means that there is not a constant variance, which violates the homoskedasticity assumption.

So how do we address this? Sometimes transforming the predictors or response can help stabilize the variance. Experiment with transfomrations on `Budget` and/or `Gross` so that, in the transformed scale, the relationship is approximately linear with a constant variance. Limit your transformations to square root, logarithms and exponentiation.

Note: There may be multiple transformations that fix this violation and give similar results. For the purposes of this problem, the transformed model doesn't have the be the "best" model, so long as it maintains both the linearity and homoskedasticity assumptions.

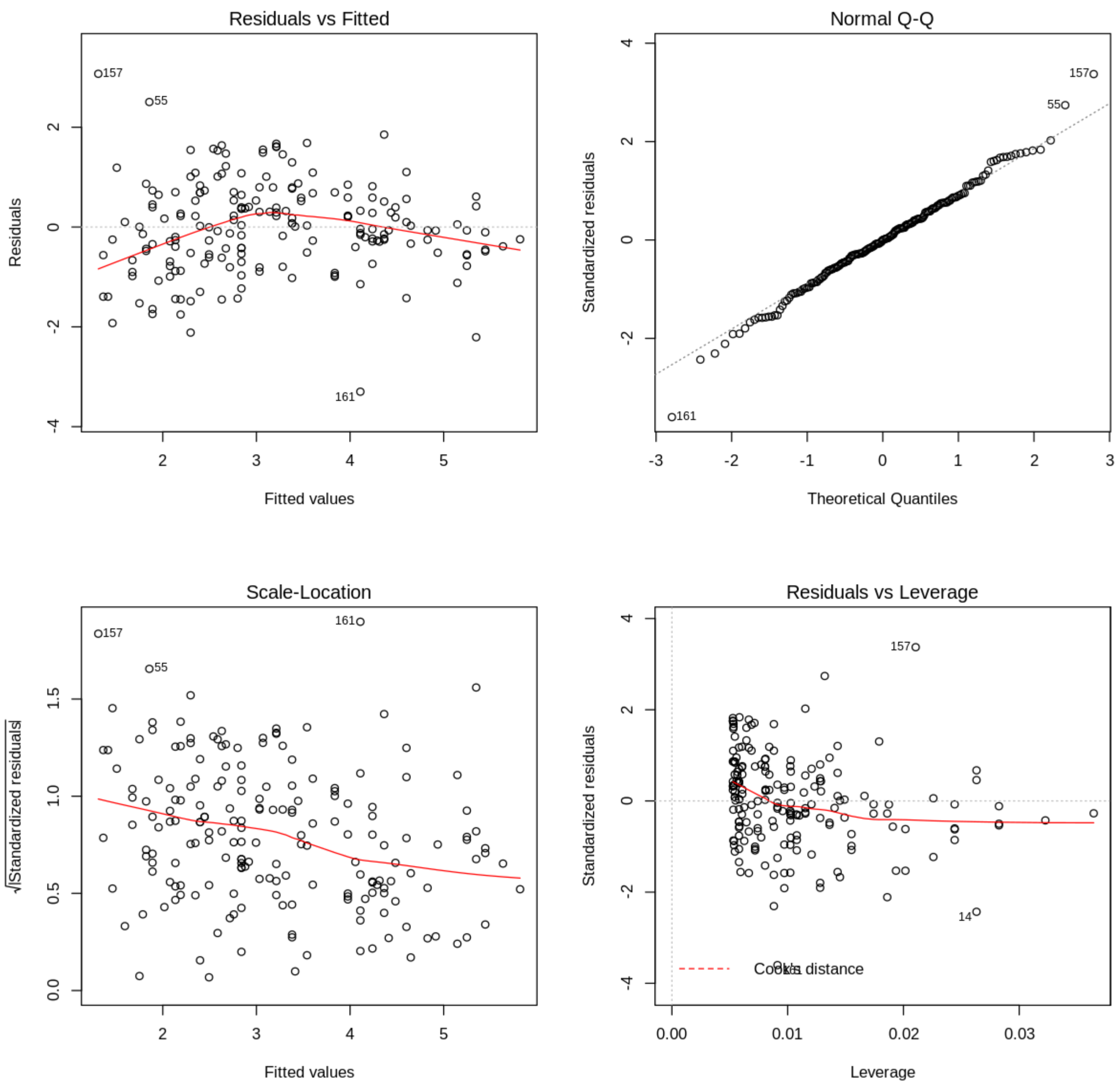
In [61]:
```
# Your Code Here
y_new = log(y_b)
x_new = sqrt(x_b)
df_new = data.frame(x_new, y_new)
lmod = lm(y_new ~ x_new, data = df_new)
options(repr.plot.width = 10, repr.plot.height = 10)
par(mfrow = c(2,2))
ggplot(df_new, aes(x=x_new, y=y_new)) + geom_point() + geom_smooth(method=
'lm', se=TRUE)
plot(lmod)
summary(lmod)
```

```
`geom_smooth()` using formula 'y ~ x'
```



```
Call:
lm(formula = y_new ~ x_new, data = df_new)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-3.2992 -0.5448  0.0004  0.5884  3.0706 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.43229    0.18262   2.367   0.0189 *  
x_new        0.43959    0.02715  16.193   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9205 on 188 degrees of freedom
Multiple R-squared:  0.5824,    Adjusted R-squared:  0.5802 
F-statistic: 262.2 on 1 and 188 DF,  p-value: < 2.2e-16
```



3. (c) Interpreting Your Transformation

You've fixed the nonconstant variance problem! Hurray! But now we have a transformed model, and it will have a different interpretation than a normal linear regression model. Write out the equation for your transformed model. Does this model have an interpretation similar to a standard linear model?

Transformed model equation: $log(y) = 0.43 + 0.44\sqrt{X} + \varepsilon, \varepsilon \sim N(0, 0.92^2)$ Now, the transformed model can be interpreted as quadrupling X leads to an expected increase of Y by a factor of exp(0.44 * 2).

```
In [ ]:
```