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 [1]:
```r
# Load Required Packages
library(ggplot2)
library(dplyr)
library(lmtest)
```

```
Attaching package: ‘dplyr’


The following objects are masked from ‘package:stats’:

    filter, lag


The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union


Loading required package: zoo


Attaching package: ‘zoo’


The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric
```

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.

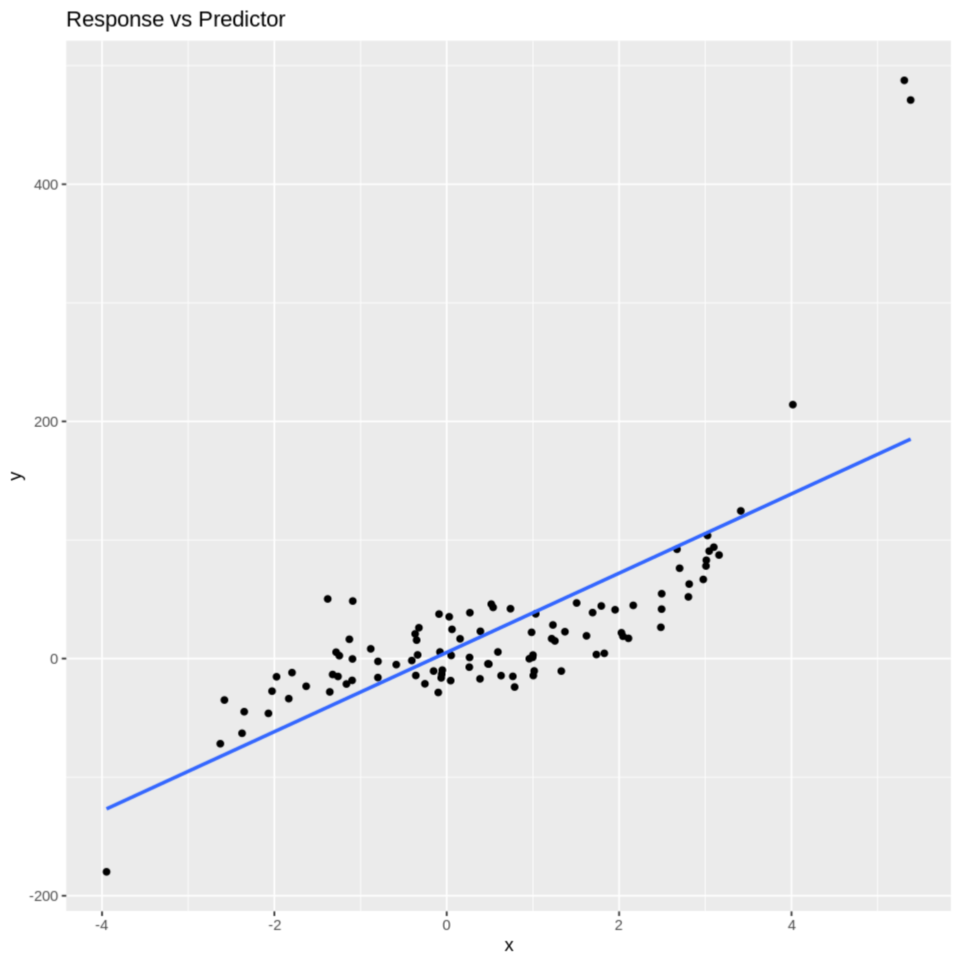
1. (a) Linearity

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

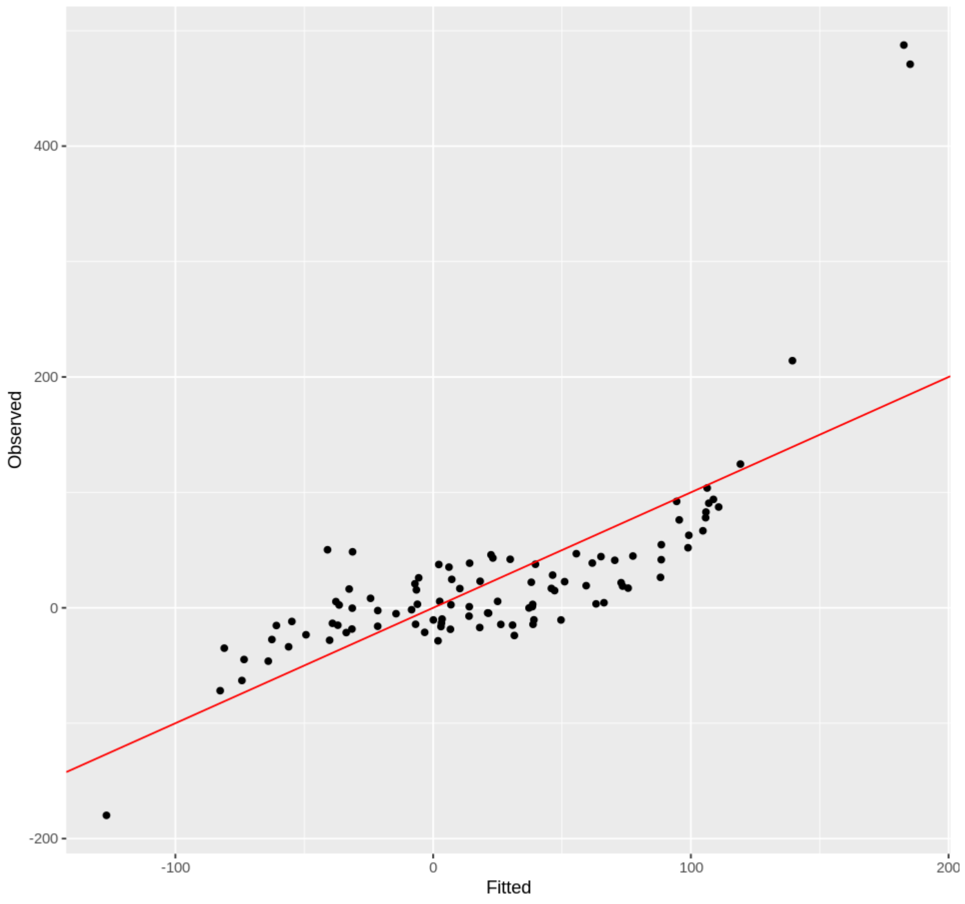
Then fit a linear model to these data and comment on where you can diagnose nonlinearity in the diagnostic plots.

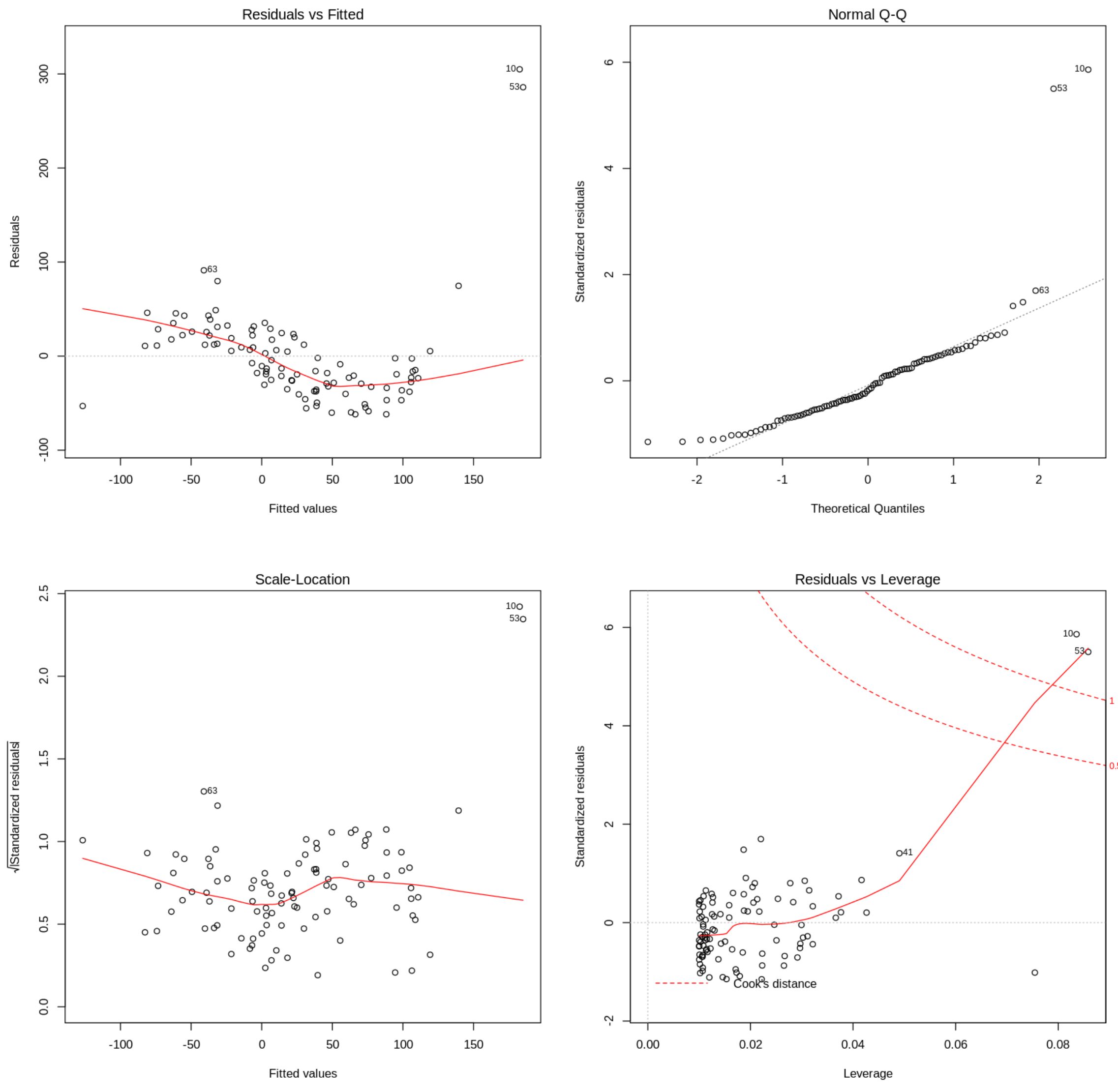
```
In [2]: # Your Code Here

seed = 0

set.seed(seed)
n = 100
x = 2 * rnorm(n) + 0.5
e = 20 * rnorm(n)
y = 3 * x ** 3 + 5 + e
data = data.frame(x=x, y=y)

p = ggplot(data, aes(x=x, y=y)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE) +
    ggtitle("Response vs Predictor")
options(repr.plot.width = 8, repr.plot.height = 8)
print(p)

model = lm(y ~ x, data)
summary(model)
data$y_hat = fitted(model)
p = ggplot(data, aes(x=y_hat, y=y)) +
    geom_point() +
    geom_abline(slope=1, intercept=0, col='red') +
    xlab('Fitted') +
    ylab('Observed') +
    ggtitle("Observed vs Fitted\nRed line represents Observed = Fit
ted")
options(repr.plot.width = 8, repr.plot.height = 8)
print(p)

options(repr.plot.width = 13, repr.plot.height = 13)
par(mfrow = c(2, 2))
plot(model)
```

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


Call:
lm(formula = y ~ x, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-61.898 -30.940  -9.663  21.914 304.926 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)    5.188      5.691   0.912    0.364    
x             33.420      3.094  10.801   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 54.35 on 98 degrees of freedom
Multiple R-squared:  0.5435,	Adjusted R-squared:  0.5388 
F-statistic: 116.7 on 1 and 98 DF,  p-value: < 2.2e-16
```



Observed vs Fitted
Red line represents Observed = Fitted





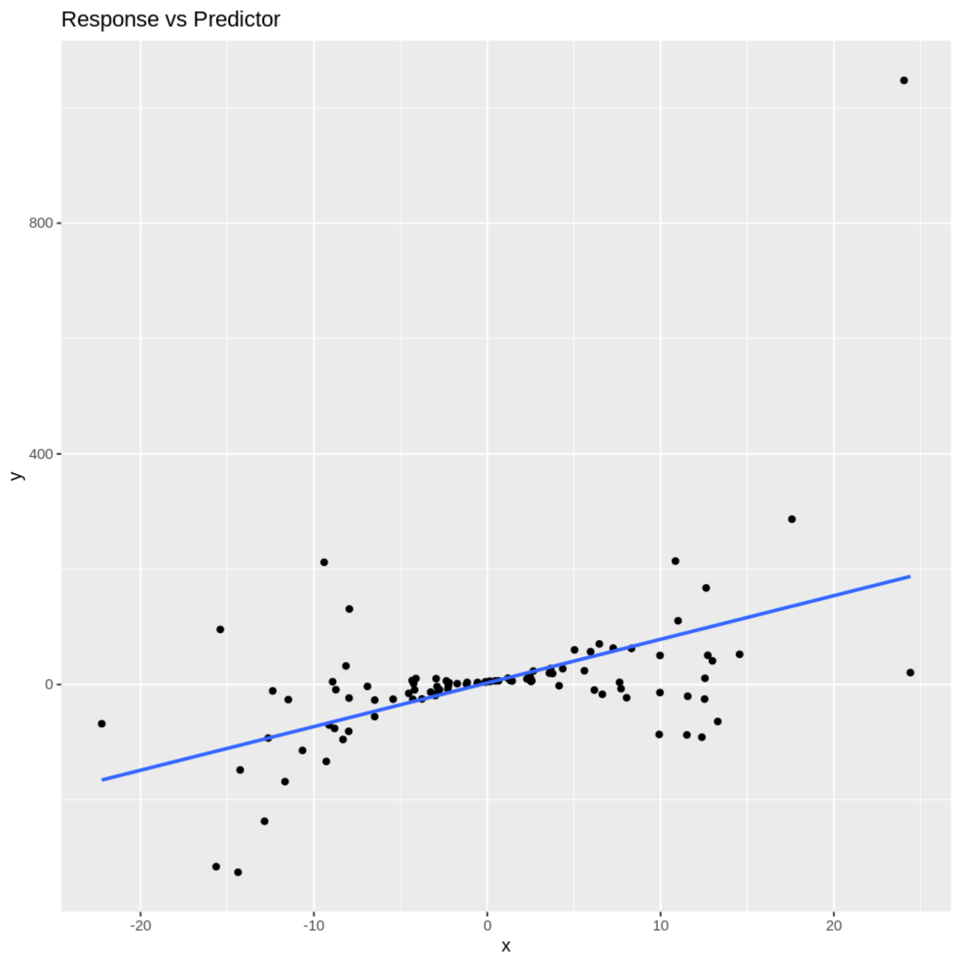
We can see clear structure that was not captured by linear relationship is showing up when looking at Observed vs Fitted and Residuals vs Fitted plots.

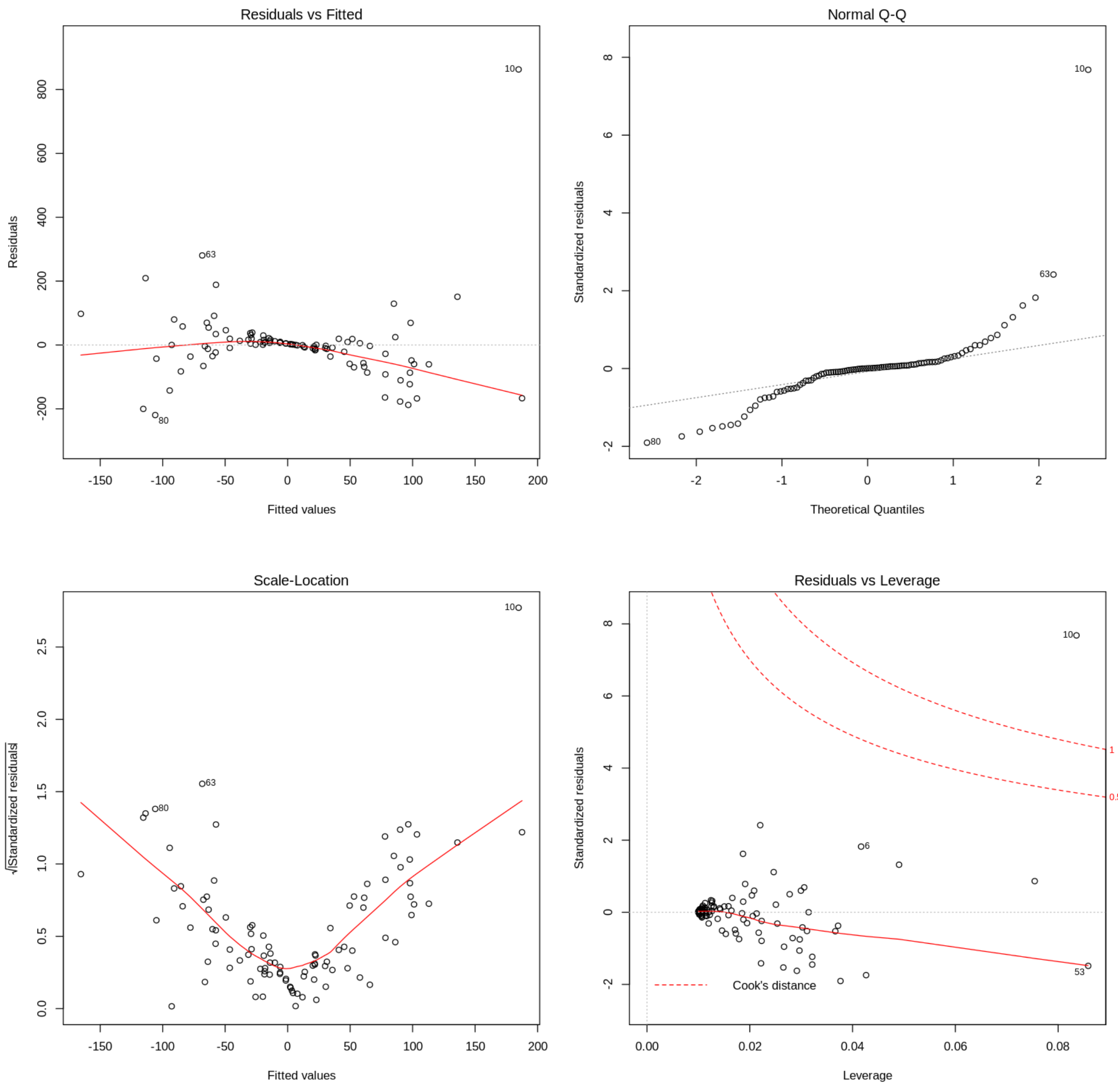
1. (b) Homoskedasticity

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 [3]:
```
# Your Code Here

set.seed(seed)
n = 100
x = 10 * rnorm(n)
e = rnorm(n) * abs(x) ** 2
y = 3 * x + 5 + e
data = data.frame(x=x, y=y)

p = ggplot(data, aes(x=x, y=y)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE) +
    ggtitle("Response vs Predictor")
options(repr.plot.width = 8, repr.plot.height = 8)
print(p)

model = lm(y ~ x, data)
options(repr.plot.width = 13, repr.plot.height = 13)
par(mfrow = c(2, 2))
plot(model)
```

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





We can see clear signs of heteroscedasticity on Residuals vs Fitted plot. Instead of gradual scattering around the horizontal line (where residuals $\hat{\varepsilon} = 0$) we see a clear structure where residuals' variance is increasing when move away from fitted $\hat{y} = 0$ in both positive and negative directions.

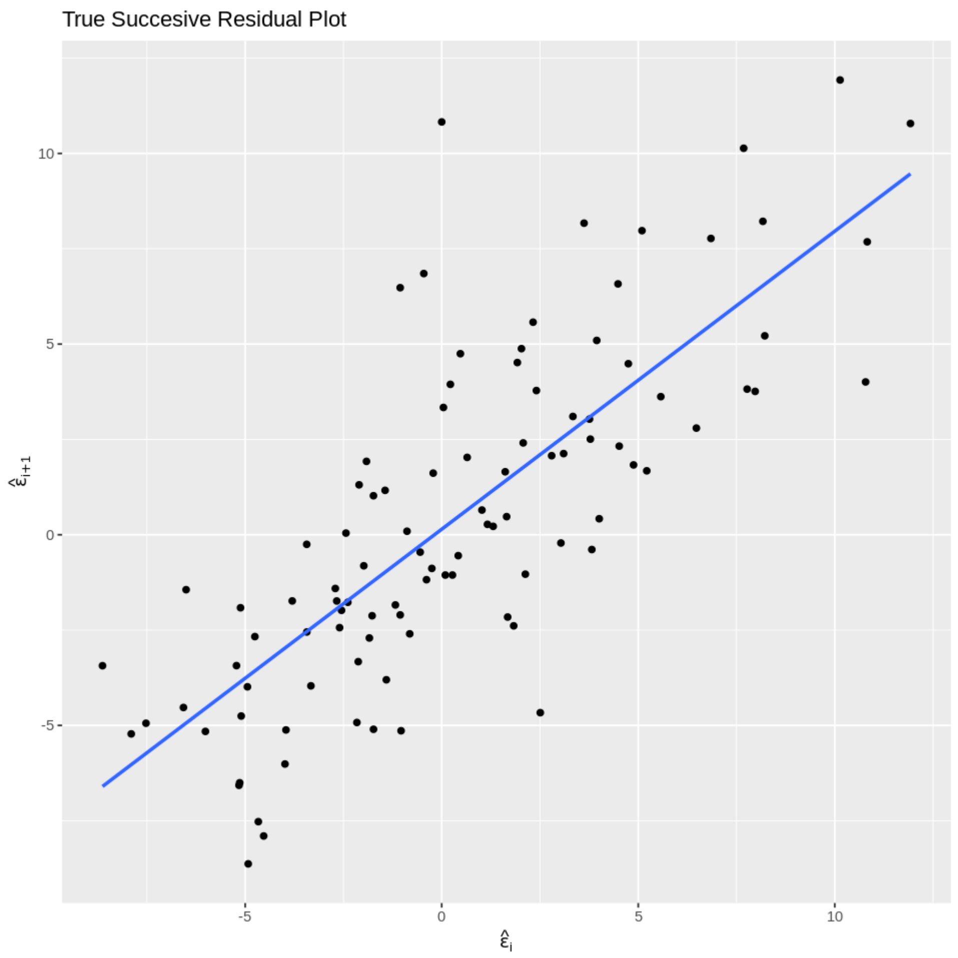
1. (c) Independent Errors

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

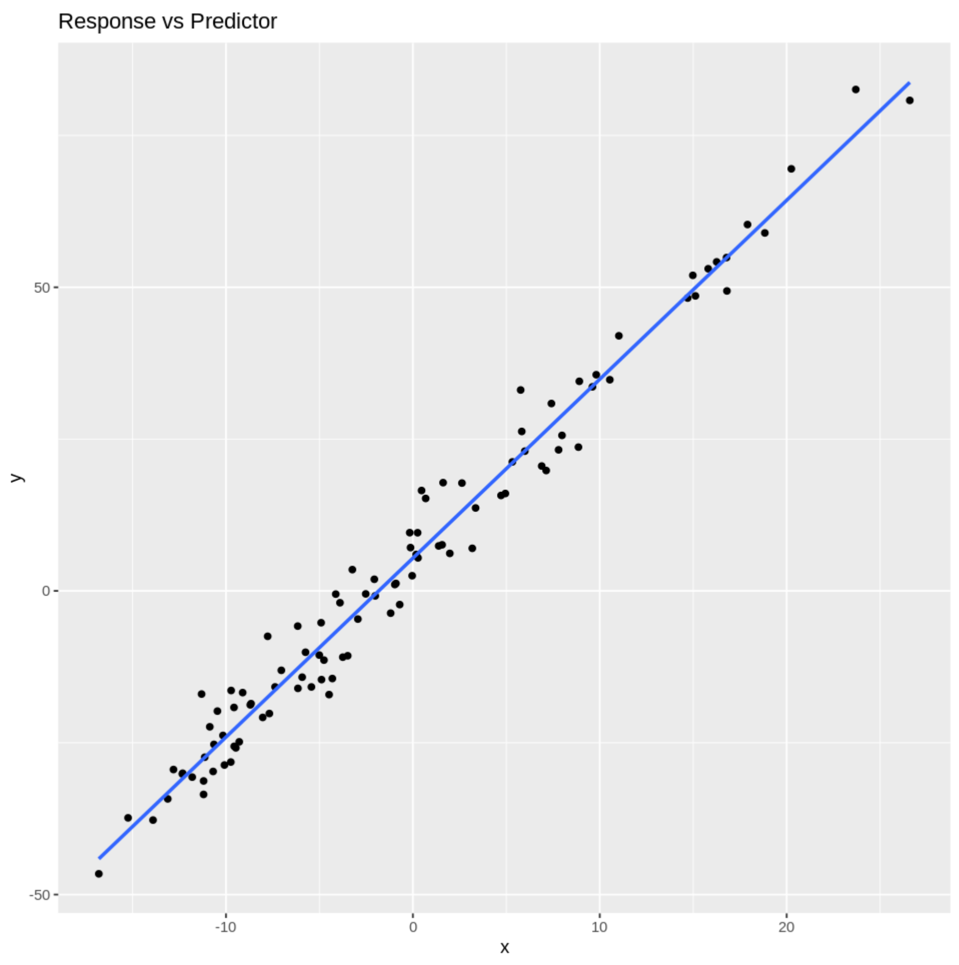
In [4]:
```
# Your Code Here

set.seed(seed)
n = 100

rho = 0.8
gamma = rnorm(n)
e = rep(0, n)
e[1] = 3 * rnorm(1) + gamma[1]
for (i in 2:n){
    e[i] = rho * e[i - 1] + gamma[i]
}
e = 3 * e
df = data.frame(e=e, e_prev=lag(e, 1))
df[is.na(df$e_prev), "e_prev"] = 0
p = ggplot(df, aes(x=e_prev, y=e)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE) +
    xlab(expression(hat(epsilon)[i])) +
    ylab(expression(hat(epsilon)[i + 1])) +
    ggtitle("True Succesive Residual Plot")
options(repr.plot.width = 8, repr.plot.height = 8)
print(p)

x = 10 * rnorm(n)
y = 3 * x + 5 + e
data = data.frame(x=x, y=y)
p = ggplot(data, aes(x=x, y=y)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE) +
    ggtitle("Response vs Predictor")
options(repr.plot.width = 8, repr.plot.height = 8)
print(p)

model = lm(y ~ x, data)
summary(model)

data$index = 1:dim(data)[1]
data$y_hat = fitted(model)
data$e_hat = residuals(model)
p = ggplot(data, aes(x=index, y=e_hat)) +
    geom_point() +
    geom_smooth(se = FALSE, color='grey', size=3, alpha=0.5) +
    xlab('Index') +
    ylab('Residual') +
    ggtitle("Residual vs Index")
options(repr.plot.width = 8, repr.plot.height = 8)
print(p)

print(dwtest(model))

my_lag = 1
sr = na.omit(data.frame(x=lag(data$e_hat, my_lag), y=data$e_hat))
p = ggplot(sr, aes(x=x, y=y)) +
    geom_point() +
```

```r
    geom_vline(xintercept = 0) +
    geom_hline(yintercept = 0) +
    xlab(expression(hat(epsilon)[i])) +
    ylab(expression(hat(epsilon)[i + 1])) +
    ggtitle("Succesive Residual Plot")
print(p)
sr_model = lm(y ~ x, sr)
summary(sr_model)
```

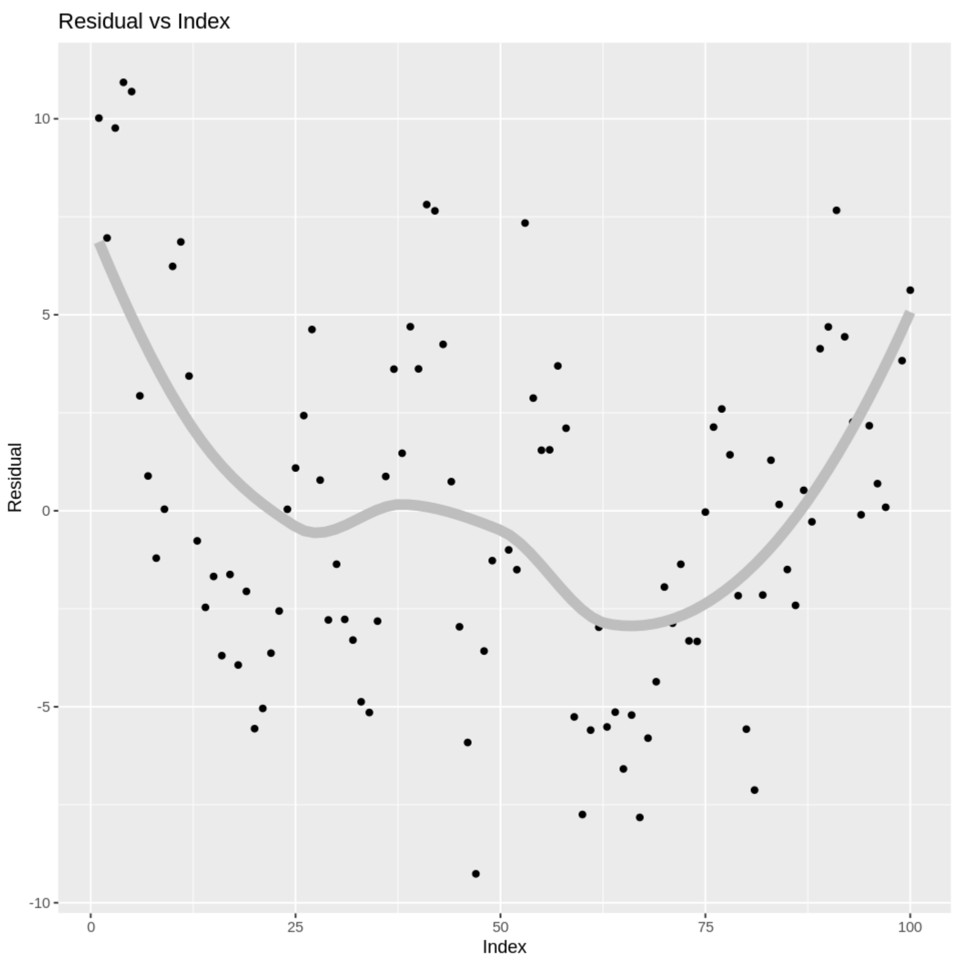
```
`geom_smooth()` using formula 'y ~ x'

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



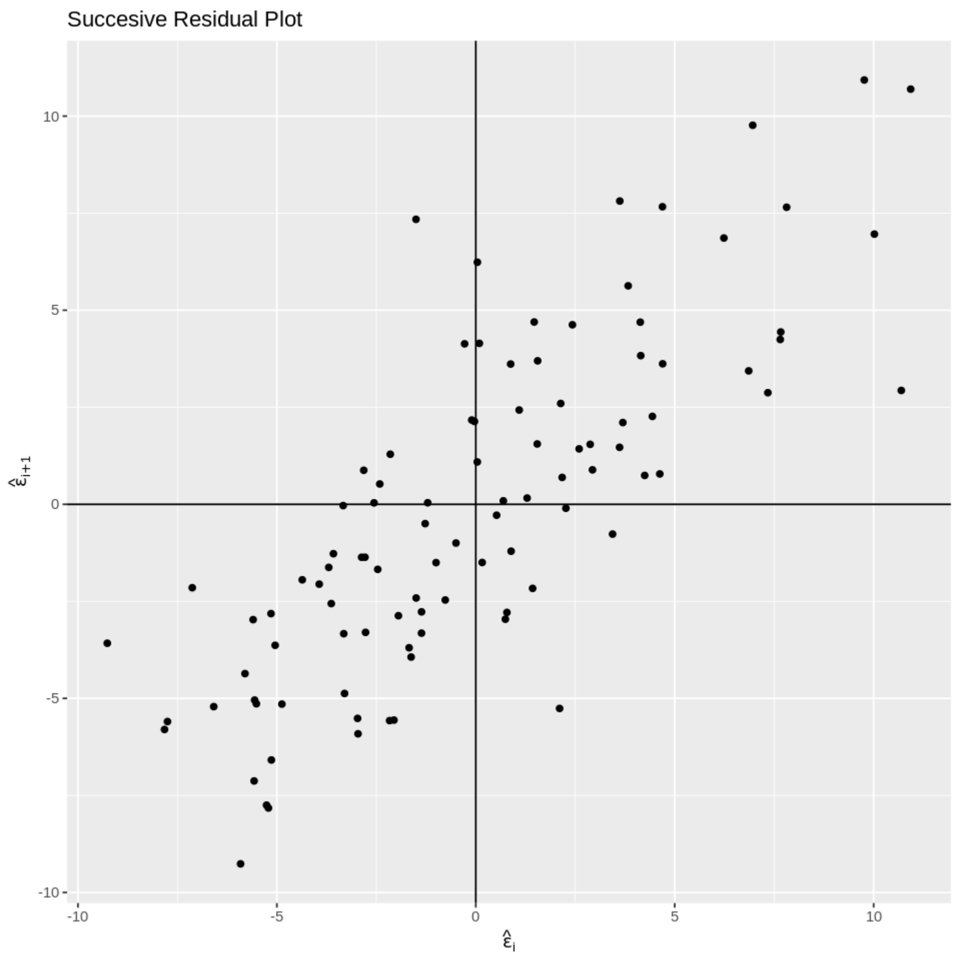
```
Call:
lm(formula = y ~ x, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.2625 -3.0534 -0.1915  2.8893 10.9305 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  5.39488    0.44781   12.05   <2e-16 ***
x            2.94694    0.04644   63.45   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.468 on 98 degrees of freedom
Multiple R-squared:  0.9762,    Adjusted R-squared:  0.976 
F-statistic:  4026 on 1 and 98 DF,  p-value: < 2.2e-16

`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
        Durbin-Watson test

data:  model
DW = 0.40177, p-value = 3.273e-16
alternative hypothesis: true autocorrelation is greater than 0
```



```
Call:
lm(formula = y ~ x, data = sr)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.8405 -1.8082 -0.3921  1.6349  8.5670 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.05698    0.26825  -0.212    0.832    
x            0.77780    0.06083  12.786   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.669 on 97 degrees of freedom
Multiple R-squared:  0.6276,    Adjusted R-squared:  0.6238 
F-statistic: 163.5 on 1 and 97 DF,  p-value: < 2.2e-16
```



We model residuals as $\varepsilon_i = \rho\varepsilon_{i-1} + \gamma_i$ where iid γ is sampled from standard normal and $\rho = 0.8$.

1. Looking at Residuals vs Index plot we see a pattern of residuals going down and then up as index increases. This indicates potential deviation from independence.
2. Looking at Successive Residuals plot we see a clear linear pattern indicating deviation from independence with estimated $\hat{\rho} = 0.78$ which is very close to true $\rho = 0.8$.
3. The way we generated the residuals is suited for the Durbin-Watson test. Running the test we reject the null ($\rho = 0$) at any significance level as the p-value is practically zero. Hence the residuals are independent.

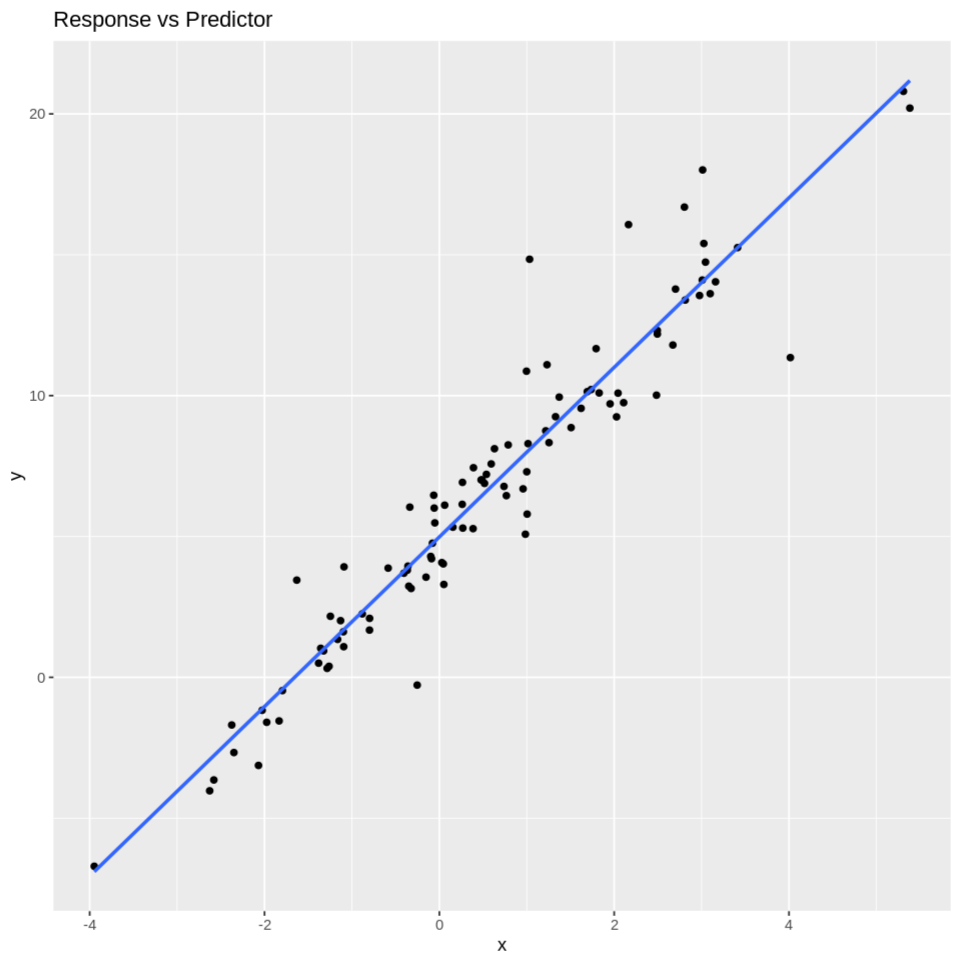
We see that all three methods work great for this way of generating residuals.

1. (d) Normally Distributed Errors

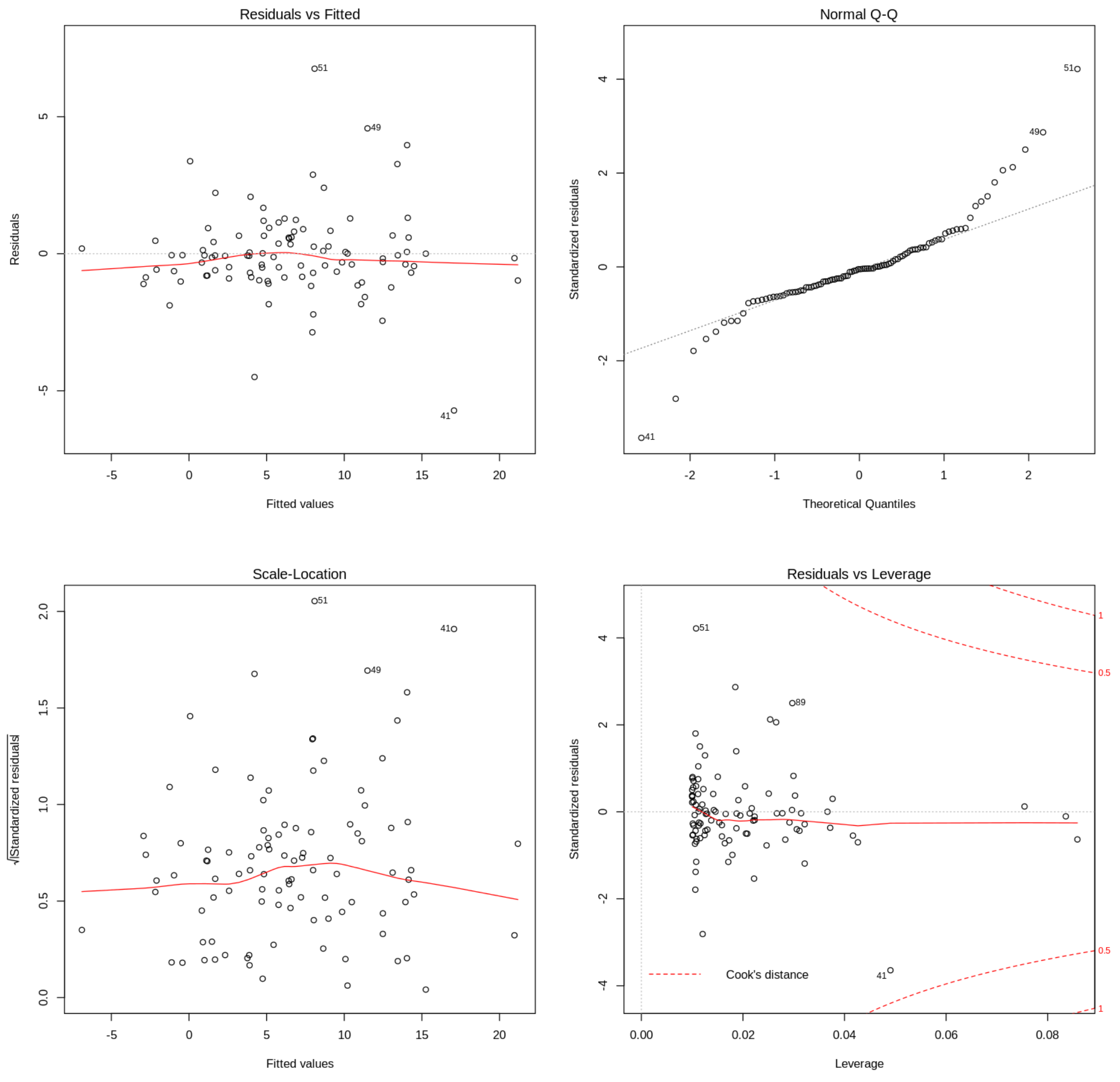
Only one more to go! Repeat the process again but simulate the data with non-normal errors.

In [5]:
```r
# Your Code Here

set.seed(seed)
n = 100
x = 2 * rnorm(n) + 0.5
e = 1 * rt(n, df=4)
y = 3 * x + 5 + e
data = data.frame(x=x, y=y)

p = ggplot(data, aes(x=x, y=y)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE) +
    ggtitle("Response vs Predictor")
options(repr.plot.width = 8, repr.plot.height = 8)
print(p)

model = lm(y ~ x, data)
summary(model)

options(repr.plot.width = 13, repr.plot.height = 13)
par(mfrow = c(2, 2))
plot(model)

print(shapiro.test(resid(model)))
```

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


Call:
lm(formula = y ~ x, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.7195 -0.7970 -0.0775  0.5937  6.7510 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.98279    0.16853   29.57   <2e-16 ***
x            3.00961    0.09163   32.84   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.61 on 98 degrees of freedom
Multiple R-squared:  0.9167,	Adjusted R-squared:  0.9159 
F-statistic:  1079 on 1 and 98 DF,  p-value: < 2.2e-16
```



```
	Shapiro-Wilk normality test

data:  resid(model)
W = 0.89575, p-value = 8.975e-07
```



We model errors with t-distribution with 4 degrees of freedom instead of standard normal.

1. QQ-plot clearly indicates deviations from normality when we look at the tails
2. Residual vs Fitted plot hints at the possibility of rejecting normality: The picture shows a reasonable distribution along $y = 0$ with few notable outliers.
3. Shapiro-Wilk test rejects normality with p-value = `8.975e-07`

Problem 2: Hats for Sale

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

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

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

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

As it was shown $\widehat{Y} = X\widehat{\boldsymbol{\beta}}$. But $\widehat{\boldsymbol{\beta}} = (X^TX)^{-1}X^TY$ hence $\widehat{Y} = X(X^TX)^{-1}X^TY = HY$

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

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

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

$HH = X(X^TX)^{-1}X^TX(X^TX)^{-1}X^T = X(X^TX)^{-1}X^T = H$. Then
$H(I_n - H) = HI_n - HH = H - H = 0$

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

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

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

We have $\widehat{\mathbf{Y}} = H\mathbf{Y}$ hence $\widehat{\mathbf{Y}}^T = \mathbf{Y}^TH^T = \mathbf{Y}^TH$. Next $\widehat{\boldsymbol{\varepsilon}} = \mathbf{Y} - \widehat{\mathbf{Y}} = \mathbf{Y} - H\mathbf{Y} = (I_n - H)\mathbf{Y}$. Finally $\widehat{\mathbf{Y}}^T\widehat{\boldsymbol{\varepsilon}} = \mathbf{Y}^TH(I_n - H)\mathbf{Y} = \mathbf{Y}^T0\mathbf{Y} = 0$

To show that $\widehat{\mathbf{Y}}$ and $\widehat{\boldsymbol{\varepsilon}}$ are uncorrelated, we need to show that their covariance is zero. By definition (we'll use $\mathbb{E}[\widehat{\boldsymbol{\varepsilon}}] = 0$)

$$\operatorname{cov}(\widehat{\mathbf{Y}}, \widehat{\boldsymbol{\varepsilon}}) = \mathbb{E}\left[\left(\widehat{\mathbf{Y}} - \mathbb{E}[\widehat{\mathbf{Y}}]\right)\left(\widehat{\boldsymbol{\varepsilon}} - \mathbb{E}[\widehat{\boldsymbol{\varepsilon}}]\right)\right] = \mathbb{E}\left[\widehat{\mathbf{Y}}\widehat{\boldsymbol{\varepsilon}}\right] - \mathbb{E}\left[\widehat{\mathbf{Y}}\right]\mathbb{E}\left[\widehat{\boldsymbol{\varepsilon}}\right] = \mathbb{E}\left[\widehat{\mathbf{Y}}\widehat{\boldsymbol{\varepsilon}}\right]$$

which in practice turns into sample average $\frac{1}{n}\widehat{\mathbf{Y}}^T\widehat{\boldsymbol{\varepsilon}}$.

Hence, $\widehat{\mathbf{Y}}^T\widehat{\boldsymbol{\varepsilon}} = 0$ implies $\operatorname{cov}(\widehat{\mathbf{Y}}, \widehat{\boldsymbol{\varepsilon}}) = 0$ which in turn imply zero correlation.

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

Objective function of linear regression is minimization of RSS $= ||\boldsymbol\varepsilon||_2 = \boldsymbol\varepsilon^T \boldsymbol\varepsilon$.

Expanding optimal (min) RSS we get $$

\widehat{\boldsymbol\varepsilon}^T \widehat{\boldsymbol\varepsilon}

({\bf Y} - \widehat{\bf Y})^T \widehat{\boldsymbol \varepsilon}

{\bf Y}^T\widehat{\boldsymbol\varepsilon}

\widehat{\bf Y}^T \widehat{\boldsymbol \varepsilon}

{\bf Y}^T\widehat{\boldsymbol\varepsilon} $$ which means that that our estimator is the best we can do to approximate true response $\mathbf{Y}$ using hyperplane spanning predictors $\mathbf{x}_1, \ldots, \mathbf{x}_p$ as only part of the response that is orthogonal to this hyperplane contribute to the min value RSS.

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

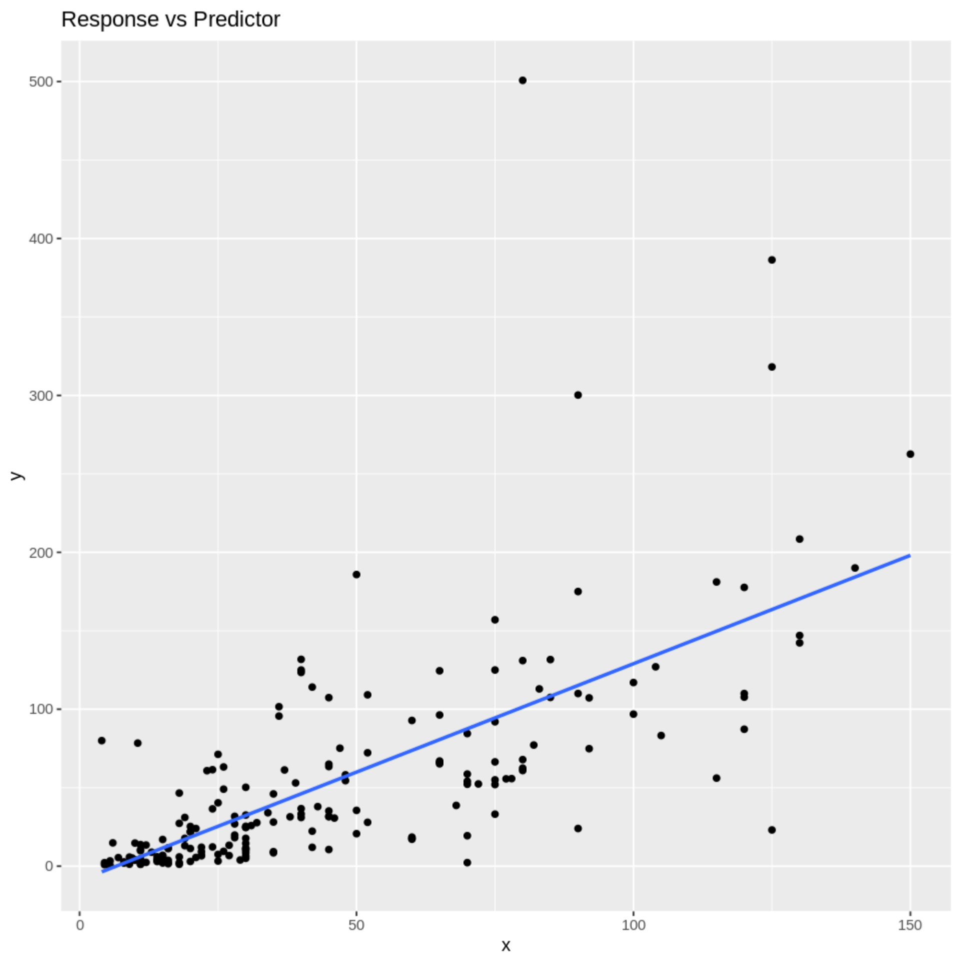
Load in the data from local directory and create a linear model with `Gross` as the response and `Budget` as the feature. The data is stored in the same local directory and is called `bollywood_boxoffice.csv`. Thank the University of Florida for this specific dataset.

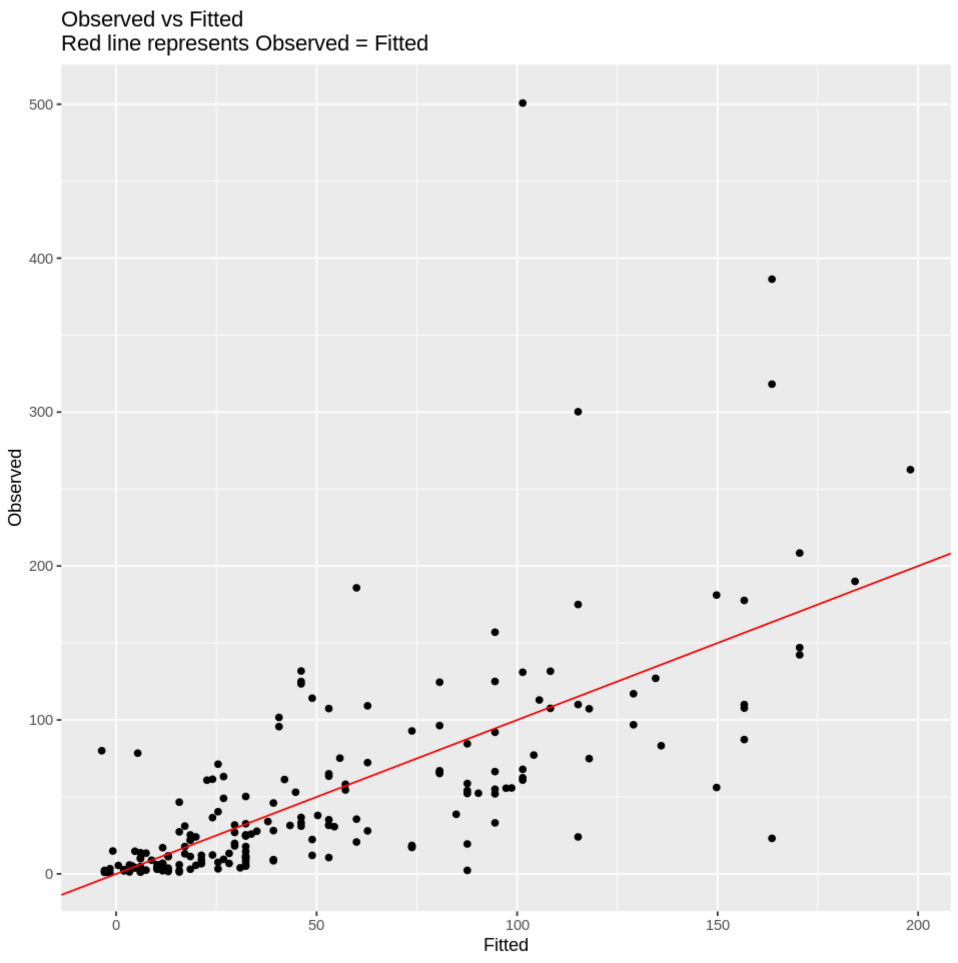
Specify whether each of the four regression model assumptions are being violated.

Data Source: [http://www.bollymoviereviewz.com (http://www.bollymoviereviewz.com)](http://www.bollymoviereviewz.com)

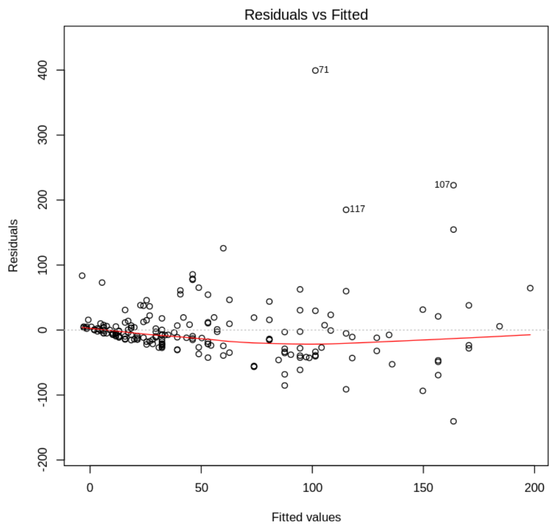
```
In [6]: # Load the data
bollywood = read.csv("bollywood_boxoffice.csv")
# summary(bollywood)

# Your Code Here

data = data.frame(x=bollywood$Budget, y=bollywood$Gross)

p = ggplot(data, aes(x=x, y=y)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE) +
    ggtitle("Response vs Predictor")
options(repr.plot.width = 8, repr.plot.height = 8)
print(p)

model = lm(y ~ x, data)
summary(model)
data$y_hat = fitted(model)
data$e_hat = resid(model)

p = ggplot(data, aes(x=y_hat, y=y)) +
    geom_point() +
    geom_abline(slope=1, intercept=0, col='red') +
    xlab('Fitted') +
    ylab('Observed') +
    ggtitle("Observed vs Fitted\nRed line represents Observed = Fit
ted")
options(repr.plot.width = 8, repr.plot.height = 8)
print(p)

options(repr.plot.width = 13, repr.plot.height = 13)
par(mfrow = c(2, 2))
plot(model)

data.order = arrange(data, x)
p = ggplot(data.order, aes(x=x, y=e_hat)) +
    geom_point() +
    geom_smooth(se=FALSE, color='grey', size=3, alpha=0.5) +
    xlab('Ordered Budget') +
    ylab('Residual') +
    ggtitle("Residual vs Ordered Budget")
options(repr.plot.width = 8, repr.plot.height = 8)
print(p)

my_lag = 1
sr = na.omit(data.frame(x=lag(data$e_hat, my_lag), y=data$e_hat))

p = ggplot(sr, aes(x=x, y=y)) +
    geom_point() +
    geom_vline(xintercept = 0) +
    geom_hline(yintercept = 0) +
    xlab(expression(hat(epsilon)[i])) +
    ylab(expression(hat(epsilon)[i + 1])) +
#     geom_smooth(se=FALSE, color='grey', size=3, alpha=0.5) +
#     geom_abline(slope = 0, intercept = 0, linetype = "solid", col
or = "blue", size=3, alpha=0.4) +
    ggtitle("Succesive Residual Plot", round(cor(sr$x, sr$y), 3))
print(p)

print(dwtest(model))
print(shapiro.test(resid(model)))
```

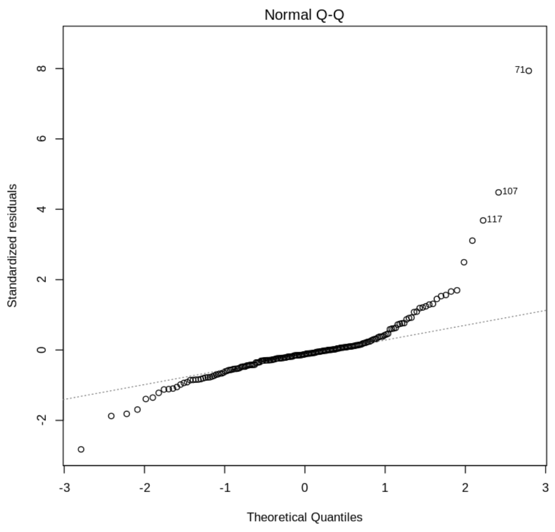
```
`geom_smooth()` using formula 'y ~ x'


Call:
lm(formula = y ~ x, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-140.48  -21.39   -6.06    7.27  399.36 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -9.120      6.112  -1.492    0.137    
x              1.381      0.108  12.792   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 50.6 on 188 degrees of freedom
Multiple R-squared:  0.4653,	Adjusted R-squared:  0.4625 
F-statistic: 163.6 on 1 and 188 DF,  p-value: < 2.2e-16
```

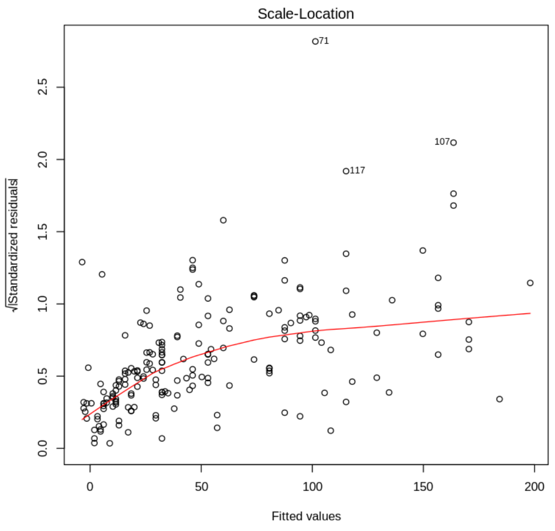


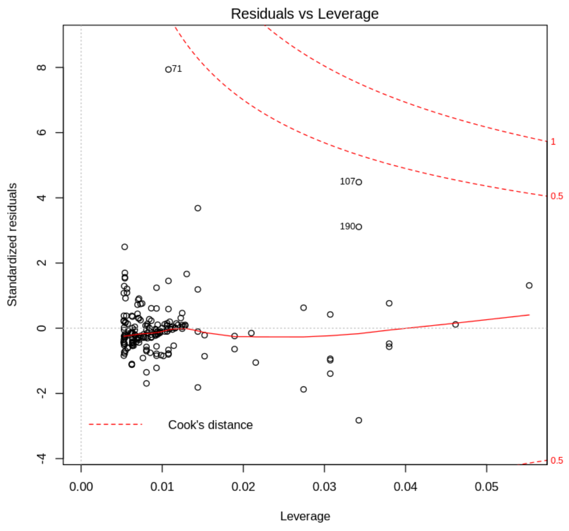


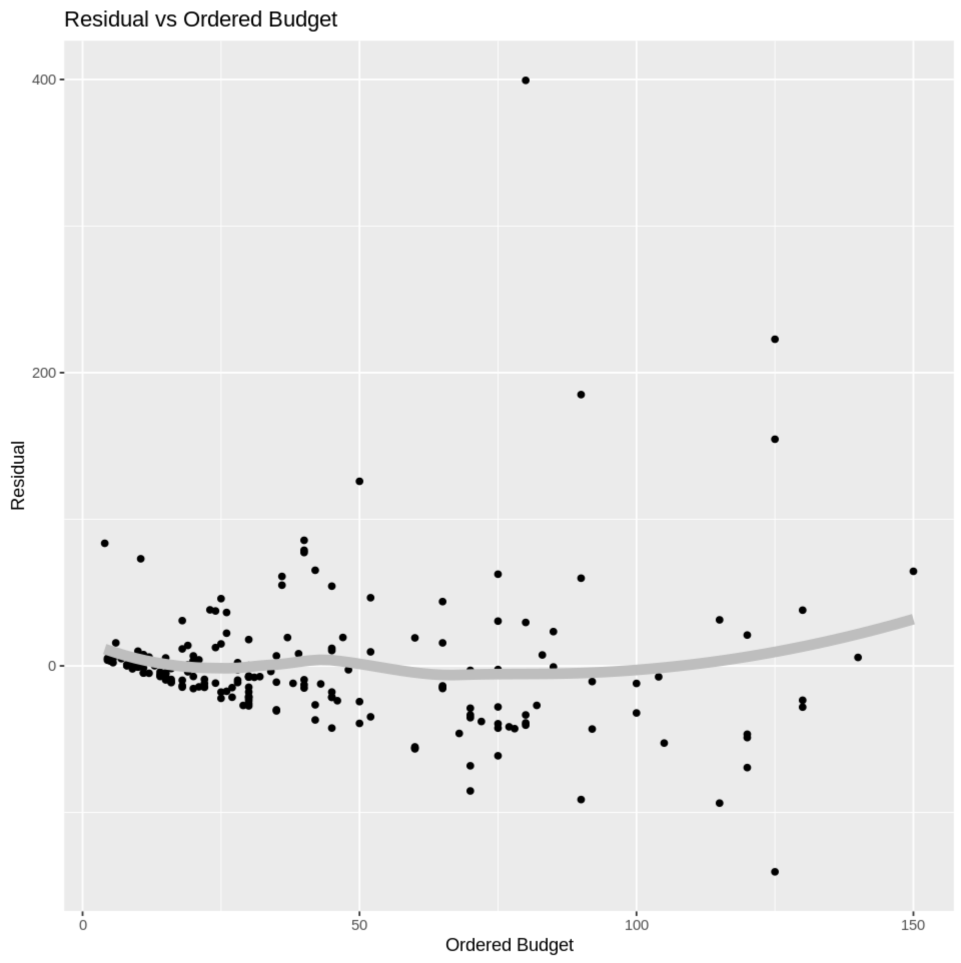
```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```


Residuals vs Fitted
Residuals
-200
-100
0
100
200
300
400
0
50
100
150
200
Fitted values
71
107
117

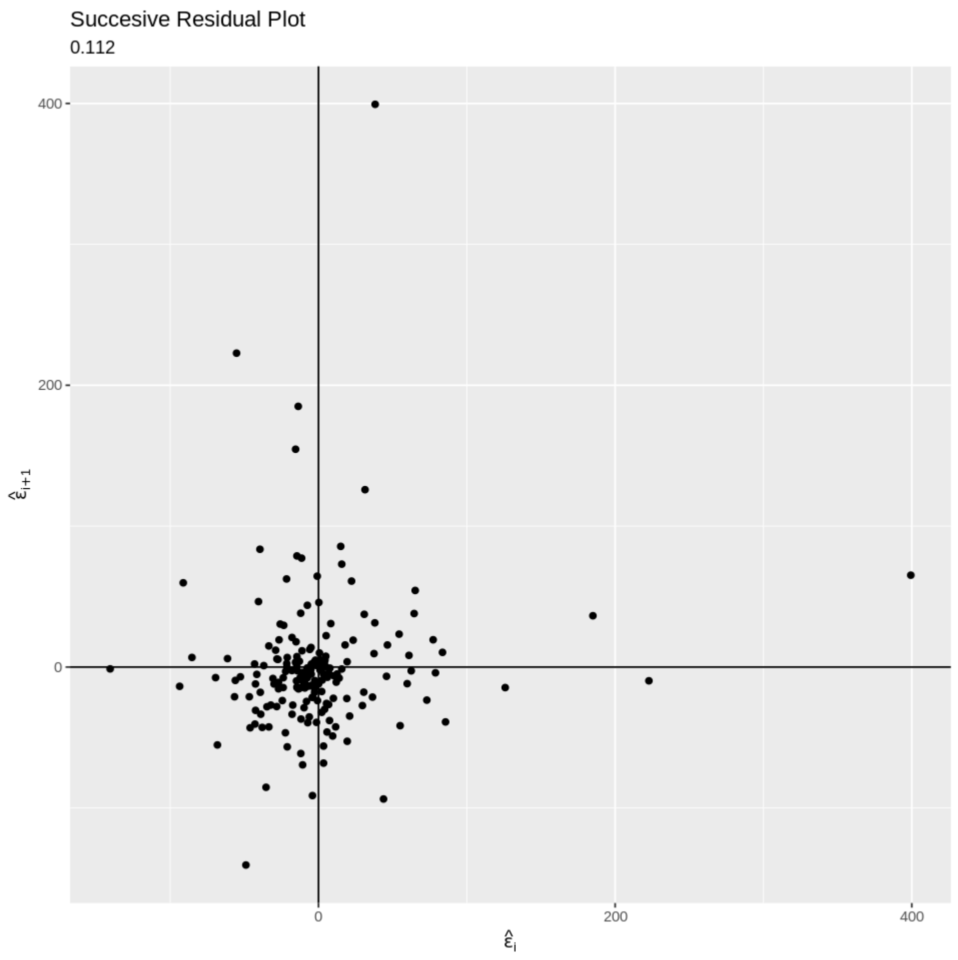

Normal Q-Q
Standardized residuals
-2
0
2
4
6
8
-3
-2
-1
0
1
2
3
Theoretical Quantiles
71
107
117


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
0
50
100
150
200
Fitted values
71
107
117


Residuals vs Leverage
Standardized residuals
-4
-2
0
2
4
6
8
0.00
0.01
0.02
0.03
0.04
0.05
Leverage
71
107
190
1
0.5
0.5
Cook's distance



```
        Durbin-Watson test

data:  model
DW = 1.727, p-value = 0.02739
alternative hypothesis: true autocorrelation is greater than 0


        Shapiro-Wilk normality test

data:  resid(model)
W = 0.72434, p-value < 2.2e-16
```



1. Looking at Observed vs Fitted and Residuals vs Fitted plots, it looks like there might be some deviations from linearity
2. Looking at Residuals vs Fitted plot we clearly see that the magnitude of residuals grow significantly as we move along Fitted value axis. Hence we reject the homoscedasticity assumption.
3. Ordering by Budget and looking at residuals doesn't indicate any particular pattern other than heteroscedasticity. Successive Residual plot also does not show any particular structure. The Durbin-Watson test rejects null ($\rho = 0$) at 5% but it looks like it might be due to a few outliers.
4. QQ-plot shows definite signs of non-normal behavior at the tails. Residuals vs Fitted plot also shows structure inconsistent with normality. Shapiro-Wilk test rejects normality at practically zero p-value.

Summarizing, it looks like the data violates following assumptions

- homoscedasticity of residuals
- residuals' normality

that to some degree interfere with two other assumptions.

3. (b) Transformations

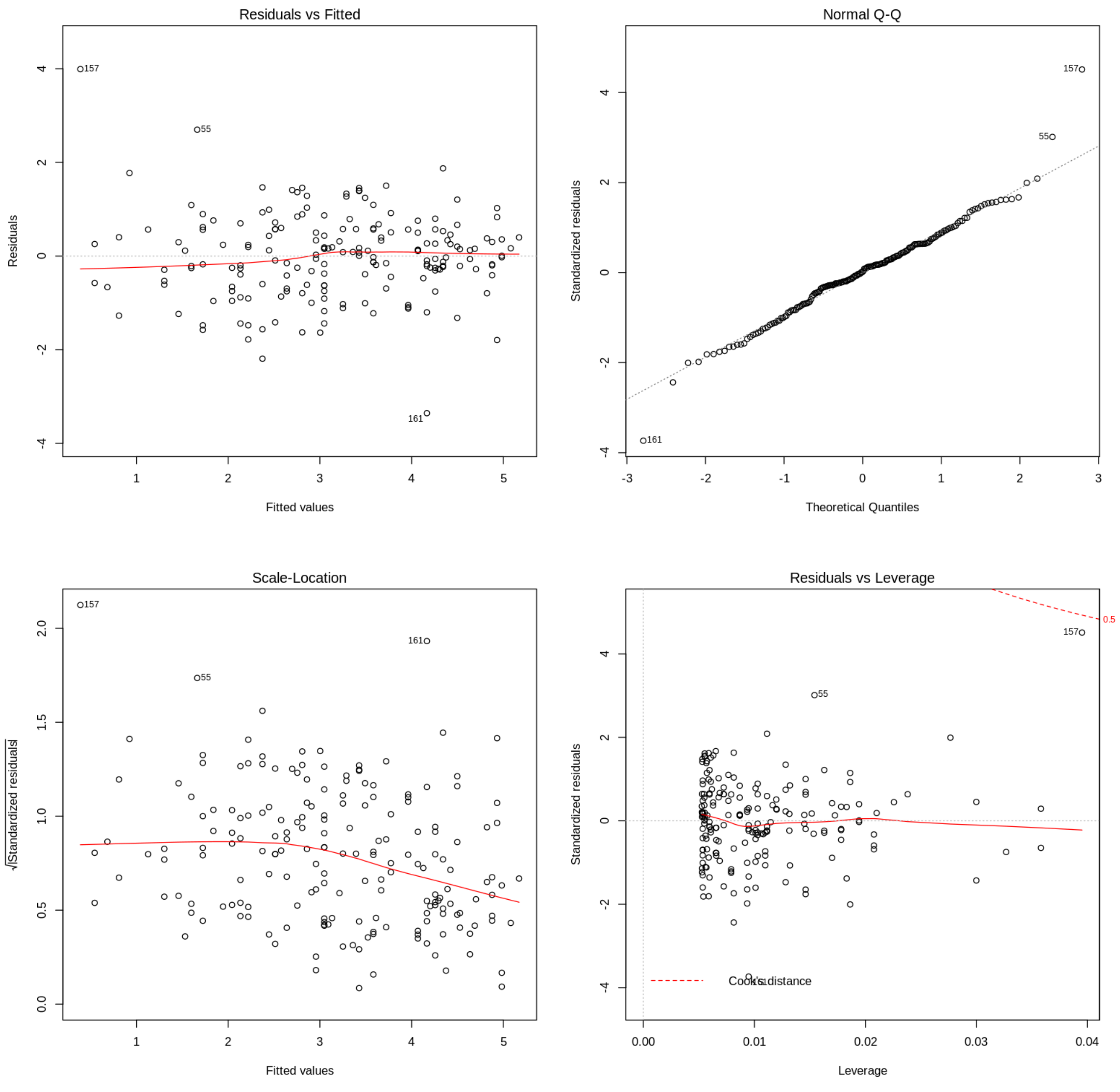
Notice that the Residuals vs. Fitted Values plot has a 'trumpet" shape to it, the points have a greater spread as the Fitted value increases. This means that there is not a constant variance, which violates the homoskedasticity assumption.

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 [7]:
```
# Your Code Here

model = lm(log(y) ~ log(x), data)
summary(model)

options(repr.plot.width = 13, repr.plot.height = 13)
par(mfrow = c(2, 2))
plot(model)
```

```
Call:
lm(formula = log(y) ~ log(x), data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.3549 -0.5634  0.0186  0.5664  3.9930 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.44023    0.28410  -5.069 9.51e-07 ***
log(x)       1.31955    0.07887  16.730  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9029 on 188 degrees of freedom
Multiple R-squared:  0.5982,    Adjusted R-squared:  0.5961 
F-statistic: 279.9 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?

It looks like

$$\log y = \beta_0 + \beta_1 \log x$$

where x is `Budget` and y is `Gross` rectifies both homoscedasticity and normality assumptions.

The fitted values are $\beta_0 = -1.44$ and $\beta_1 = 1.32$.

Going back to the original variables

$$y = e^{\beta_0} x^{\beta_1}.$$

The fitted coefficients have different interpretation:

1. If one were to invest 1 unit of capital (it not clear what are the units in this data so I'll stick to this lingo) into the movie (i.e. `Budget` $x = 1$) then on average the `Gross` would be $y = e^{\beta_0}$ which in our case ≈ 0.24.
2. If one were it increases the movie `Budget` n-fold: $x \to nx$ then on average movie's `Gross` will increase by a factor of n^{β_1} : $y \to n^{\beta_1} y$ which in out case $\approx n^{1.32}$.

Two observations for our case:

- movies with small budget on average don't break even
- n-fold increase in movie budget gets amplified and translates into $n^{1.32}$-fold on average increase in gross, implying that making high budget movies is a good investment.