Ideas:

- Fit a MLR model and look at diagnostic plots.
  - Maybe include transformation that fixes the problem, and they need to comment on it.
- Make them simulate two different datasets, one that meets the assumption and one that doesn't, so they can see the difference.
  - Normality vs Non-normality
  - Homoskedasticity vs nonconstant variance
  - See STAT4010_Unit4_Code Problem 2

In [ ]:

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.

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.

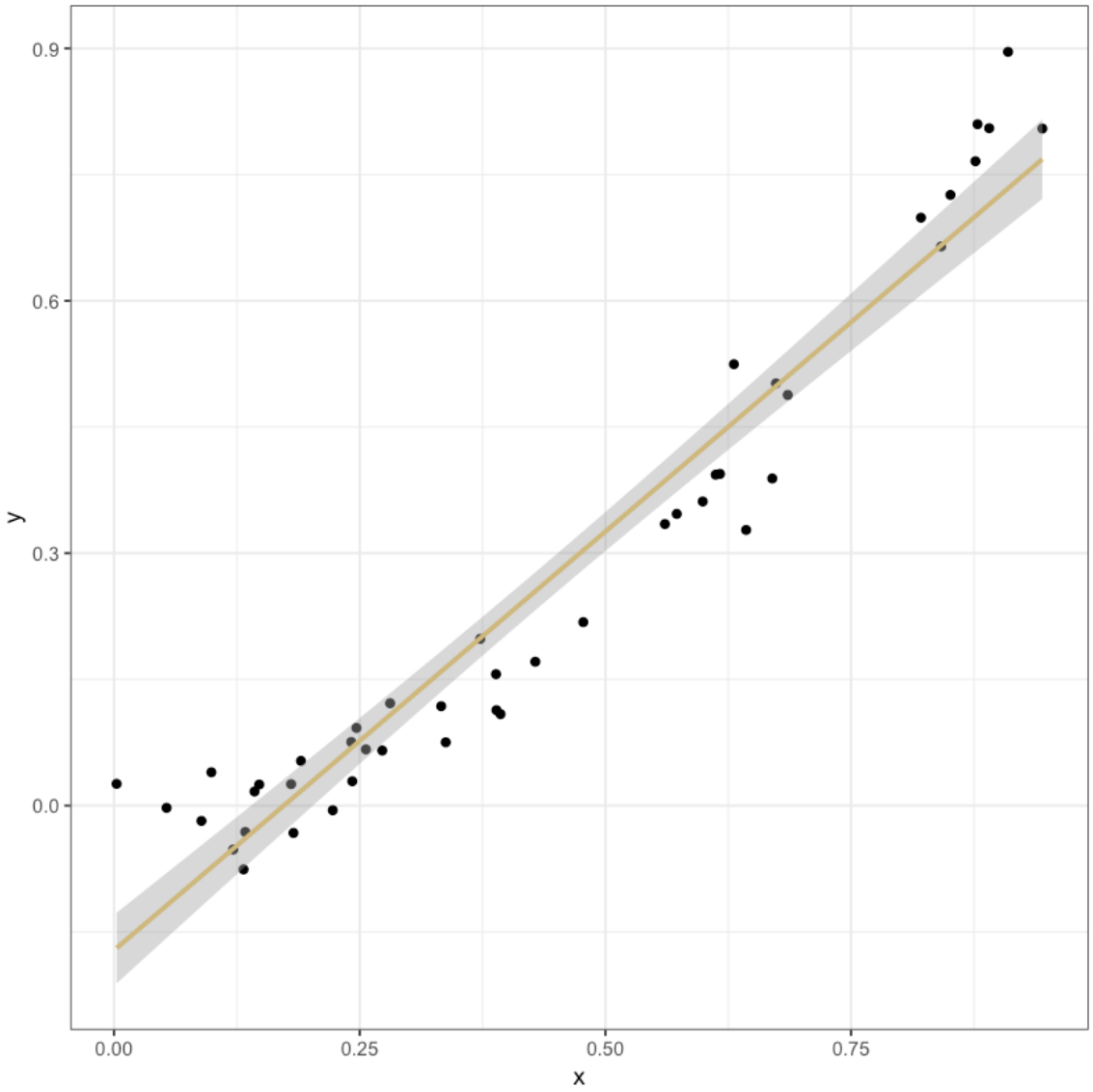
1. (a) Linearity

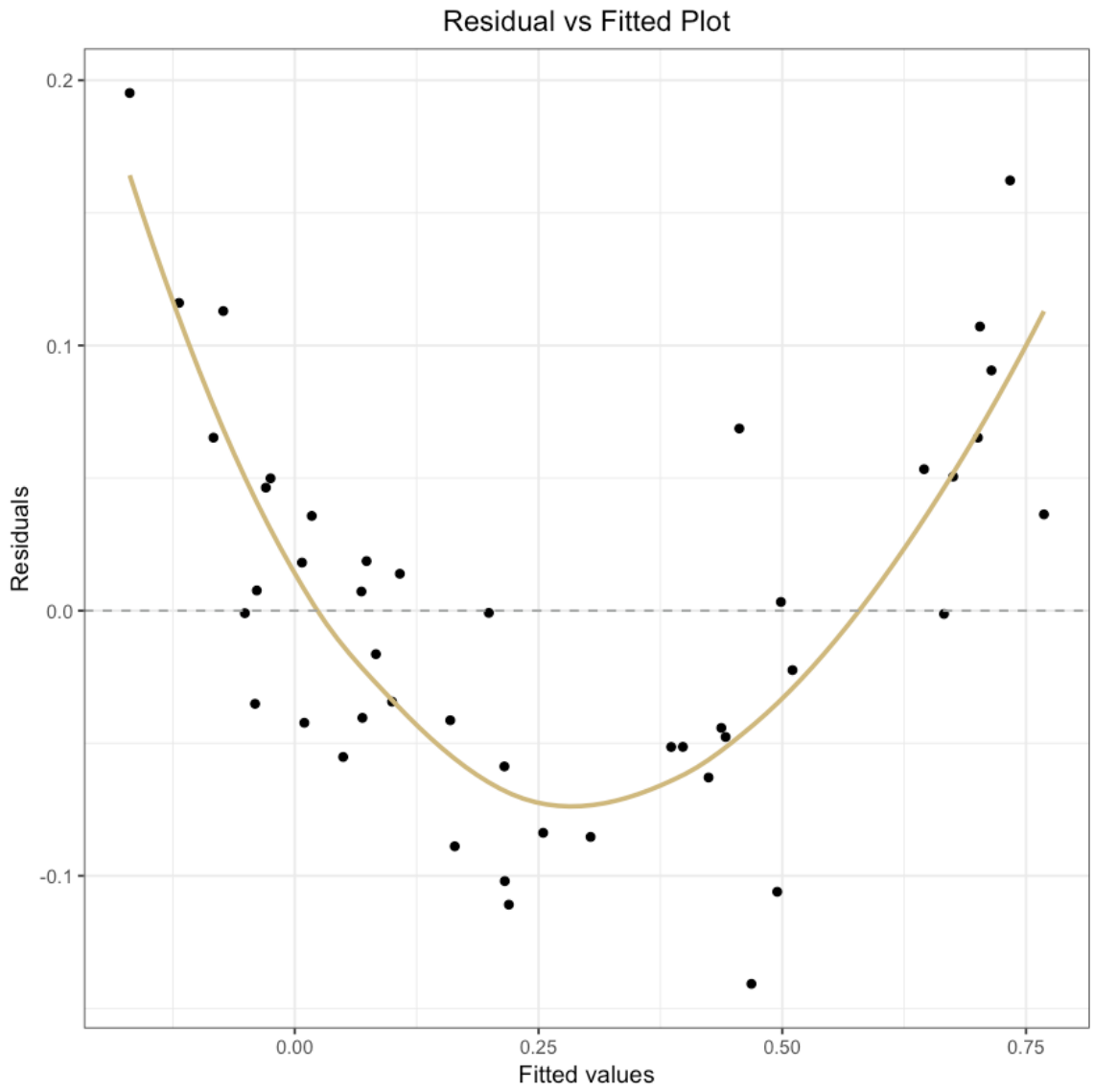
Generate SLR data that violates the linearity assumption, but maintains the other assumptions. Create a scatterplot for these data using ggplot.

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

In [15]:

```
set.seed(2016)
library(ggplot2)
# Simulation for violating linearity
n = 45; x = runif(n, 0, 1); y = 1 + x + rnorm(n,0,abs(x))
y = x^2 + rnorm(n, sd = 0.05)
lmod = lm(y ~ x);

ggplot(mapping = aes(x = x, y = y)) +
    geom_point() +
    geom_smooth(method = "lm", col = "#CFB87C")+theme_bw()

#diagnostic plot
p1=ggplot(lmod, aes(.fitted, .resid))+geom_point()
p1= p1+stat_smooth(method="loess", col = "#CFB87C", se = FALSE)
p1=p1 + geom_hline(yintercept=0, col="#A2A4A3", linetype="dashed")
p1=p1+xlab("Fitted values")+ylab("Residuals")
    p1<-p1+ggtitle("Residual vs Fitted Plot")+theme_bw() +
    theme(plot.title = element_text(hjust=0.5))


p1
```


0.9
0.6
0.3
0.0
y
0.00
0.25
0.50
0.75
x

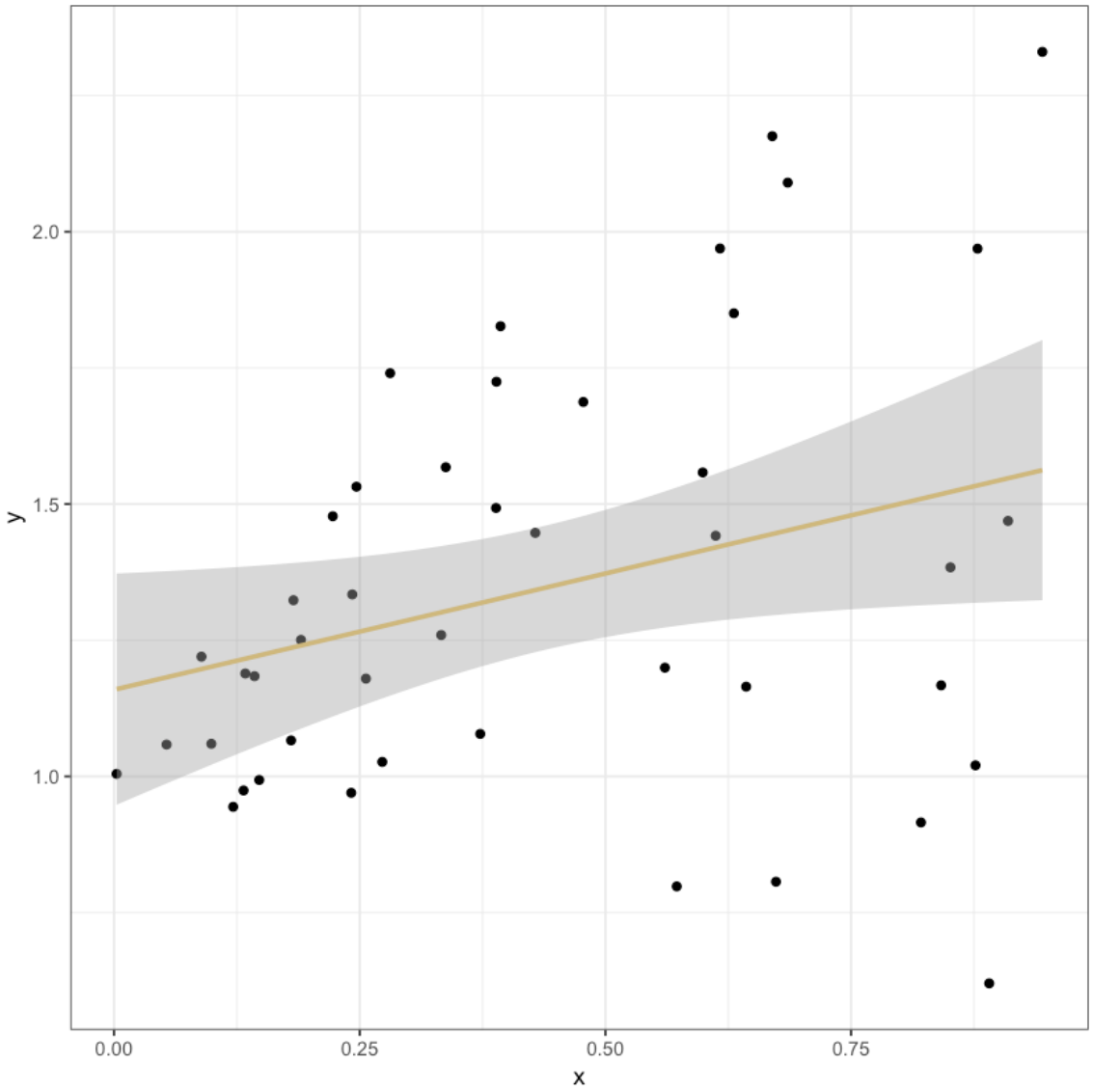


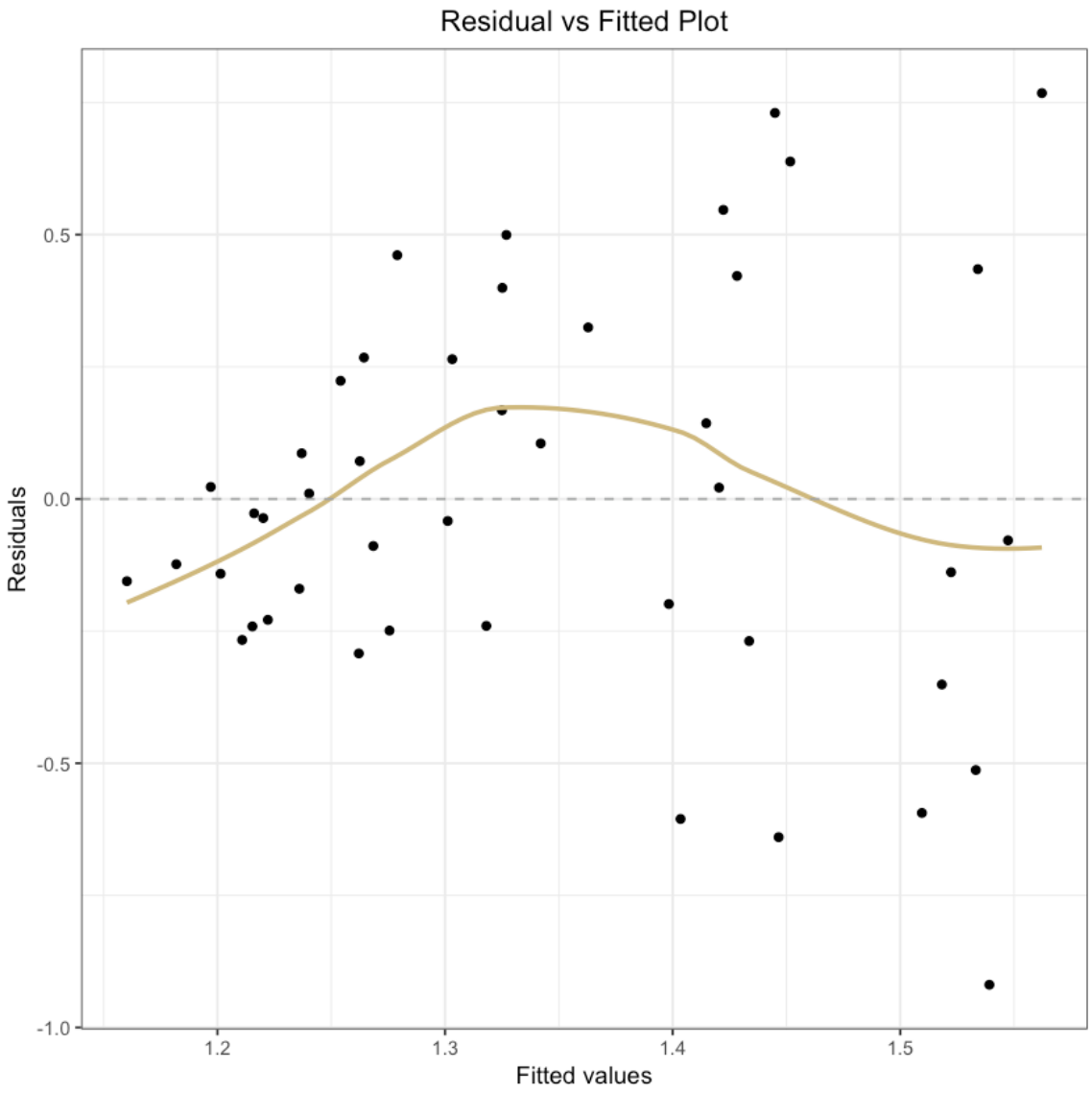
1. (b) Homoskedasticity

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 [16]:

```
#Simulation violating constant variance
y = 1 + x + rnorm(n,0,abs(x))
lmod = lm(y ~ x)

ggplot(mapping = aes(x = x, y = y)) +
    geom_point() +
    geom_smooth(method = "lm", col = "#CFB87C")+theme_bw()

#diagnostic plot
p1=ggplot(lmod, aes(.fitted, .resid))+geom_point()
p1= p1+stat_smooth(method="loess", col = "#CFB87C", se = FALSE)
p1=p1 + geom_hline(yintercept=0, col="#A2A4A3", linetype="dashed")
p1=p1+xlab("Fitted values")+ylab("Residuals")
    p1<-p1+ggtitle("Residual vs Fitted Plot")+theme_bw() +
    theme(plot.title = element_text(hjust=0.5))


p1
```


2.0
1.5
1.0
y
0.00
0.25
0.50
0.75
x

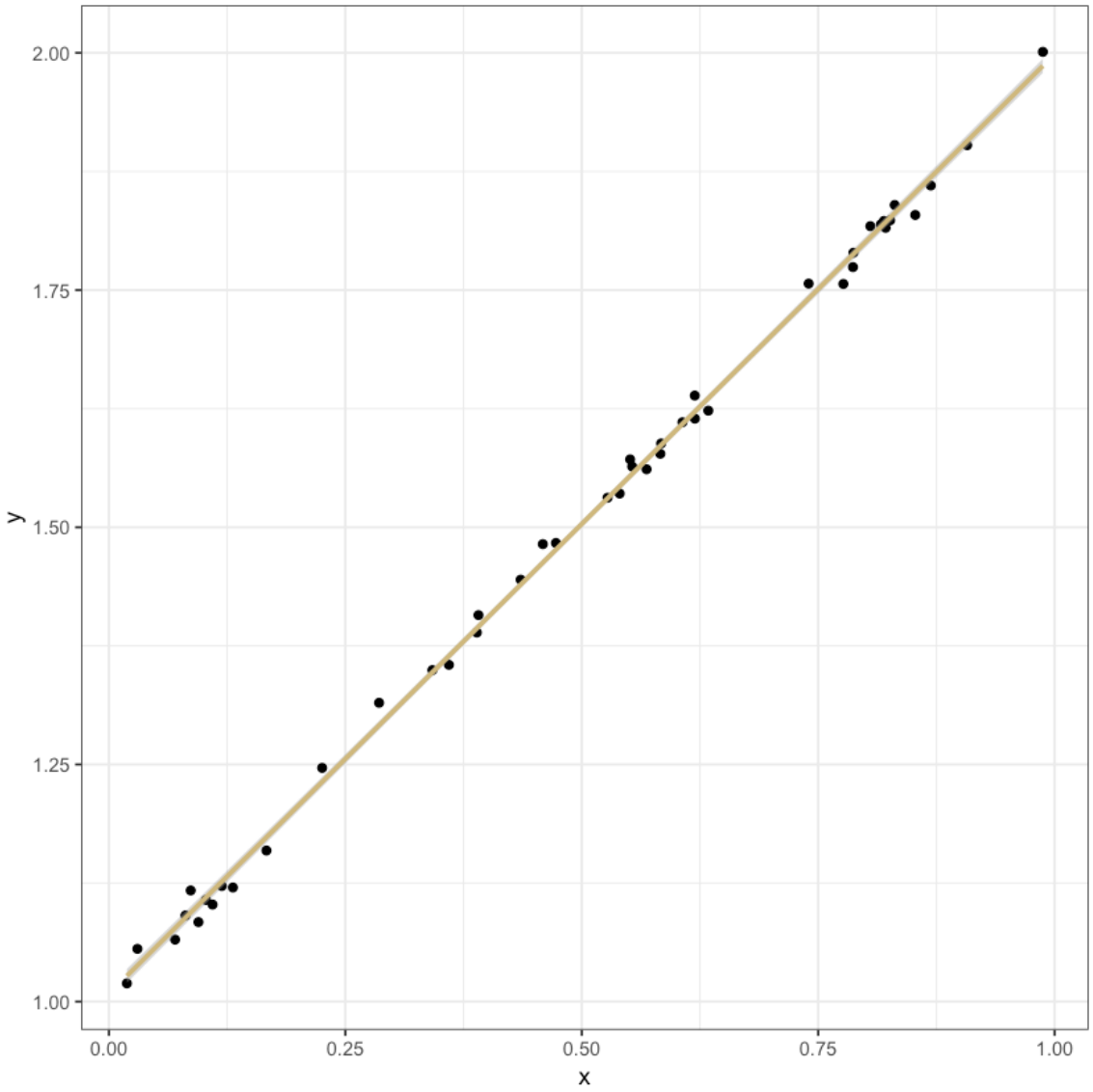


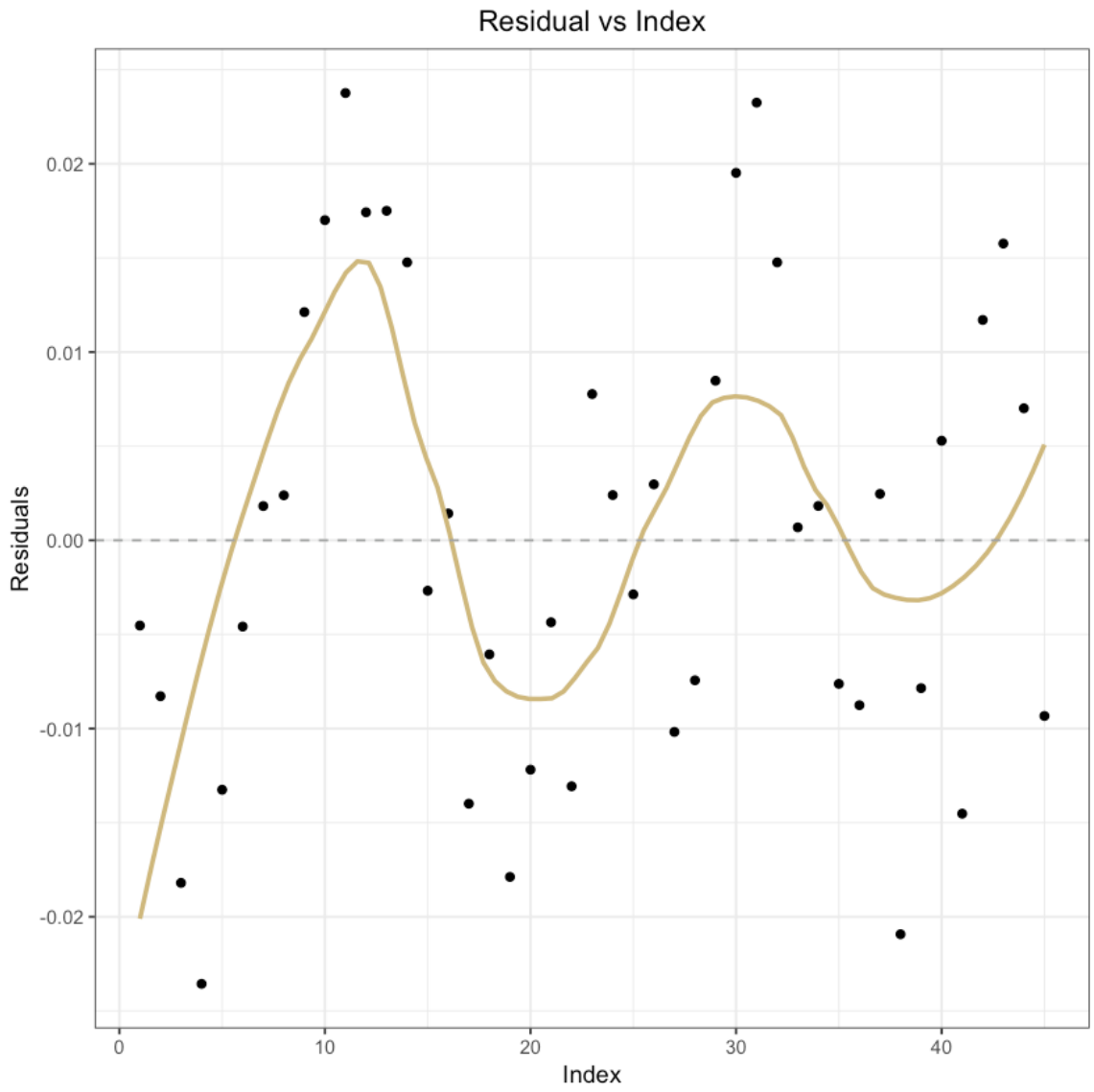
1. (c) Independent Errors

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

In [43]:

```
#Simulation violating independent errors
set.seed(999)
n = 45; x = runif(n, 0, 1);

e = matrix(NA, nrow = n)

rho = 0.7
e[1] = 0
for (i in 2:n){
    e[i] = rho*e[i-1] + rnorm(1,0,0.01)
}

y = 1 + x + e
lmod = lm(y ~ x);


ggplot(mapping = aes(x = x, y = y)) +
    geom_point() +
    geom_smooth(method = "lm", col = "#CFB87C")+theme_bw()

#diagnostic plot
p1=ggplot(lmod, aes(1:n, .resid))+geom_point()
p1= p1+stat_smooth(method="loess", col = "#CFB87C", se = FALSE, spa
n = 0.5)
p1=p1 + geom_hline(yintercept=0, col="#A2A4A3", linetype="dashed")
p1=p1+xlab("Index")+ylab("Residuals")
    p1<-p1+ggtitle("Residual vs Index")+theme_bw() +
    theme(plot.title = element_text(hjust=0.5))


p1
```


2.00
1.75
1.50
1.25
1.00
y
0.00
0.25
0.50
0.75
1.00
x



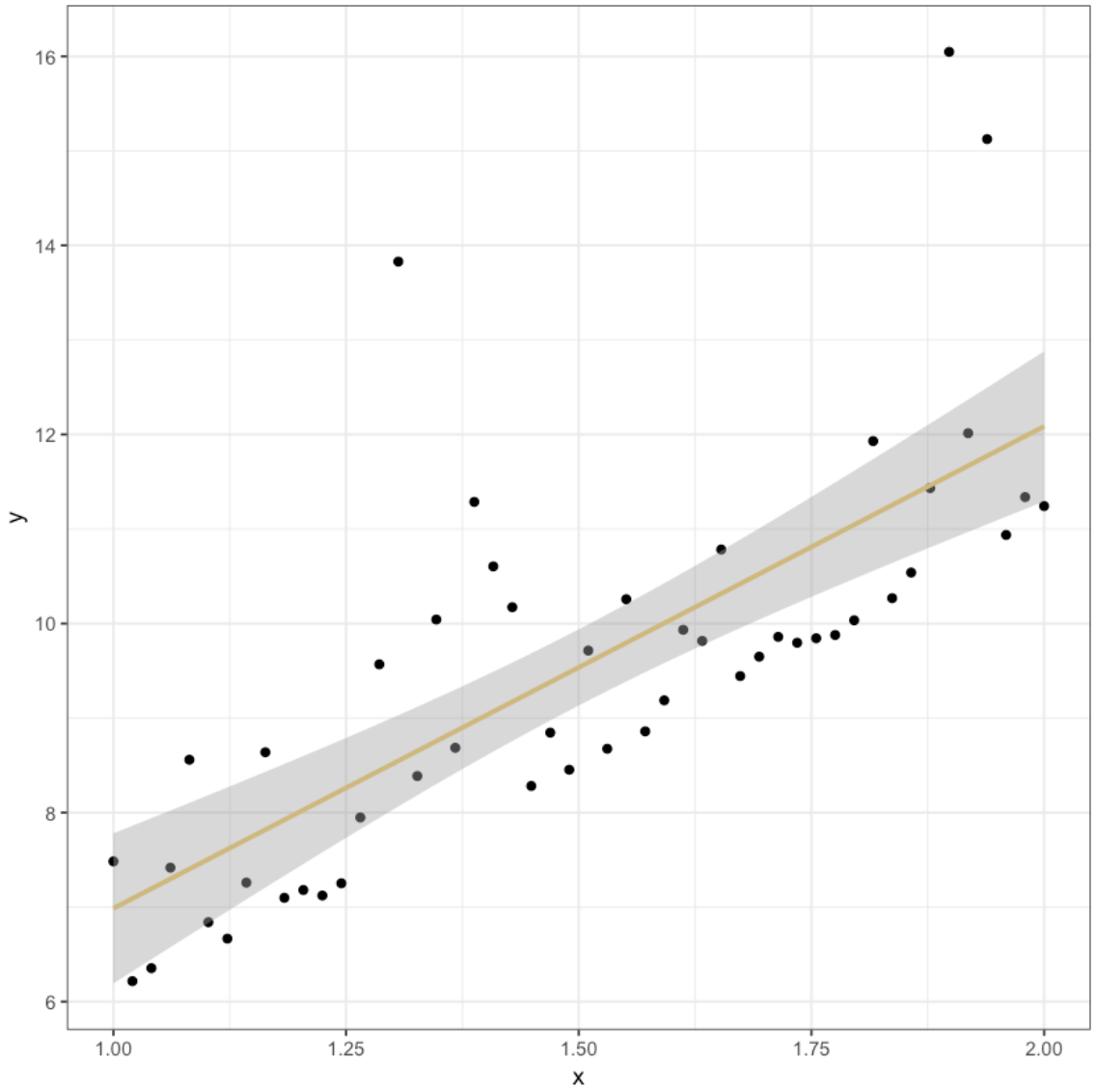
We generated data so that successive errors are highly correlated. The residuals vs index plot reflects this: we see an oscilating pattern in the residuals, suggesting that the errors are correlated.

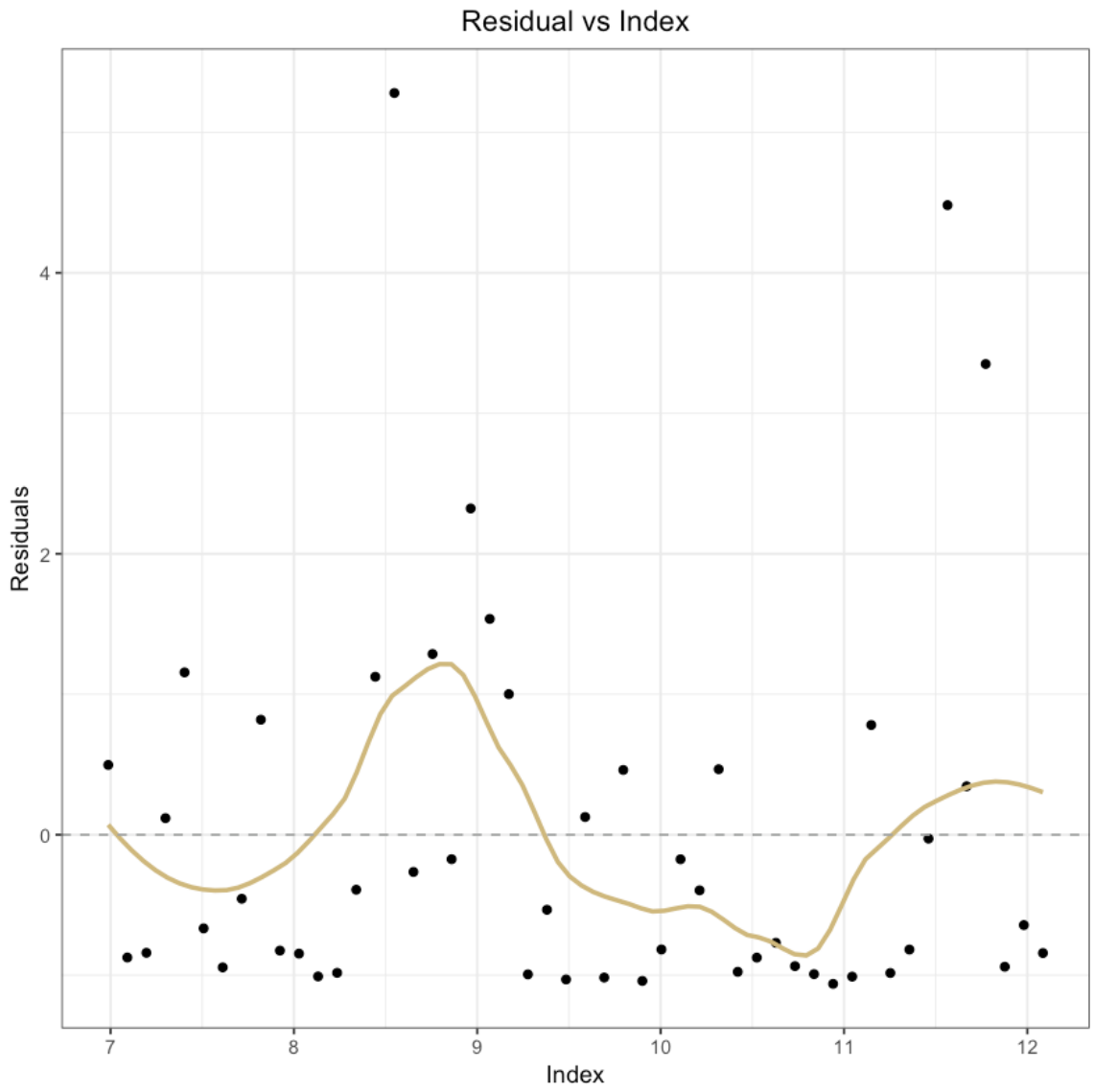
1. (d) Normally Distributed Errors

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

In [47]:

```
# Errors from a gamma(1/2, 1/2) distribution
n = 50; x = seq(1,2, length.out = n); b0 = 1; b1 = 5; e = rgamma(n,
0.5, 0.5)
y = b0 + b1*x + e

lmod = lm(y ~ x)

ggplot(mapping = aes(x = x, y = y)) +
    geom_point() +
    geom_smooth(method = "lm", col = "#CFB87C")+theme_bw()

#diagnostic plot
p1=ggplot(lmod, aes(.fitted, .resid))+geom_point()
p1= p1+stat_smooth(method="loess", col = "#CFB87C", se = FALSE, spa
n = 0.5)
p1=p1 + geom_hline(yintercept=0, col="#A2A4A3", linetype="dashed")
p1=p1+xlab("Index")+ylab("Residuals")
    p1<-p1+ggtitle("Residual vs Index")+theme_bw() +
    theme(plot.title = element_text(hjust=0.5))


p1
```


16
14
12
10
8
6
y
1.00
1.25
1.50
1.75
2.00
x



Here we generated data from a gamma distribution. We see residual values spread around zero in a way that would be very rare if the errors/residuals were normal. In particular, many values fall below zero, with a long tail above zero.

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, $\widehat{\boldsymbol{\varepsilon}}$, 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.

$$\widehat{Y} = X\widehat{\boldsymbol{\beta}} = X(X^TX)^{-1}X^T\mathbf{Y} = HY$$

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

$$H^T = \left(X(X^TX)^{-1}X^T\right)^T = (X^T)^T\left[(X^TX)^{-1}\right]^T X^T = X\left[(X^TX)^T\right]^{-1}X^T$$
$$= X(X^TX)^{-1}X^T = H$$

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

Note that

$$HH = \left(X(X^TX)^{-1}X^T\right)\left(X(X^TX)^{-1}X^T\right) = X(X^TX)^{-1}X^TX(X^TX)^{-1}X^T$$
$$= X(X^TX)^{-1}X^T = H,$$

because $(X^TX)^{-1}X^TX = I_{p+1}$.

Thus, $H(I_n - H) = HI_n - HH = H - H = 0_n$

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.

$$\widehat{\mathbf{Y}}^T\widehat{\boldsymbol{\varepsilon}} = (H\mathbf{Y})^T(I - H)\mathbf{Y} = \mathbf{Y}^TH^T(I_n - H)Y = \mathbf{Y}^TH(I_n - H)Y = \mathbf{Y}^T0_nY = 0.$$

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

This result shows that, if the linear regression assumptions are met, then there should be no correlation between the fitted values, $\widehat{\mathbf{Y}}$, and the residuals, $\widehat{\boldsymbol{\varepsilon}}$. Thus, it gives us a way to check some of our model assumptions.

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.

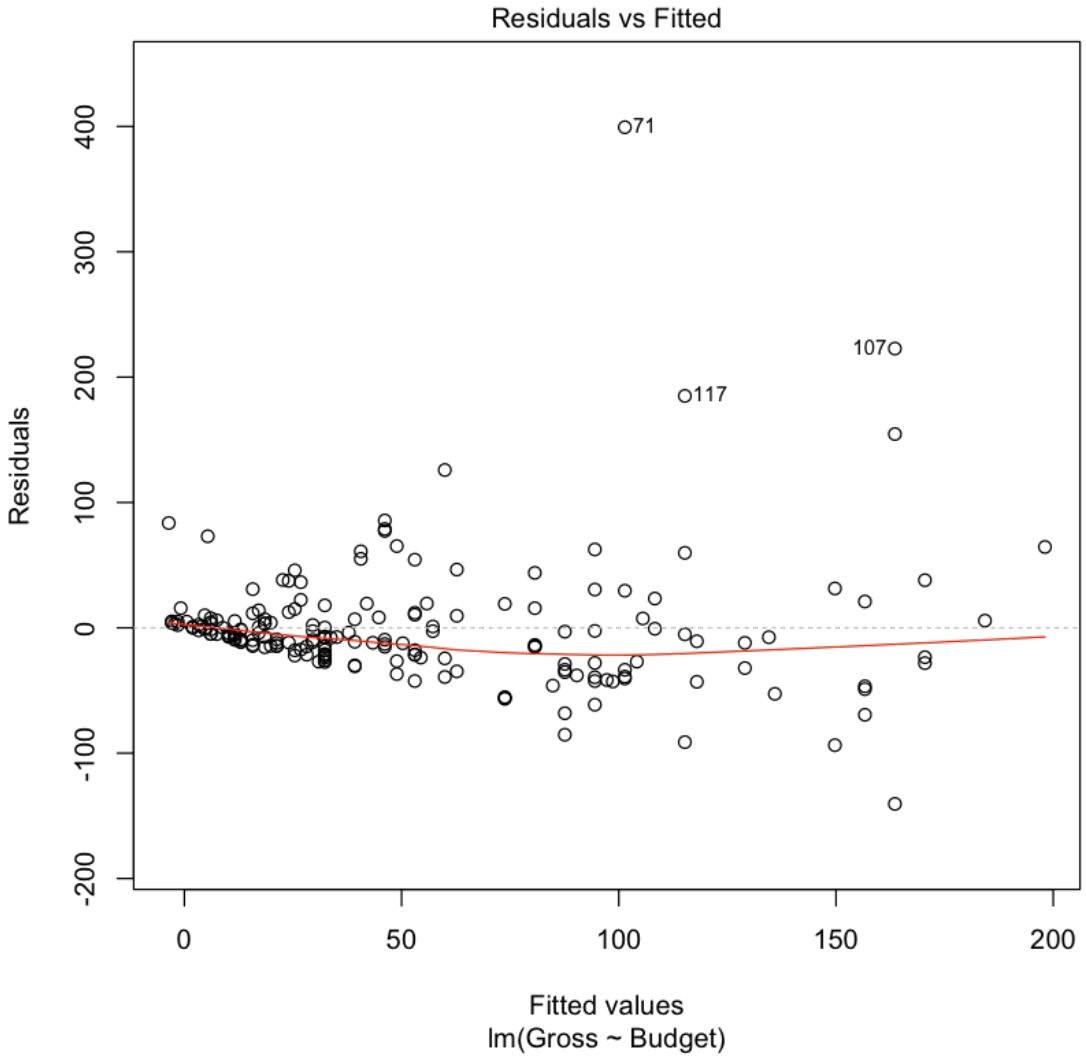
Specify whether each of the four regression model assumptions are being violated.

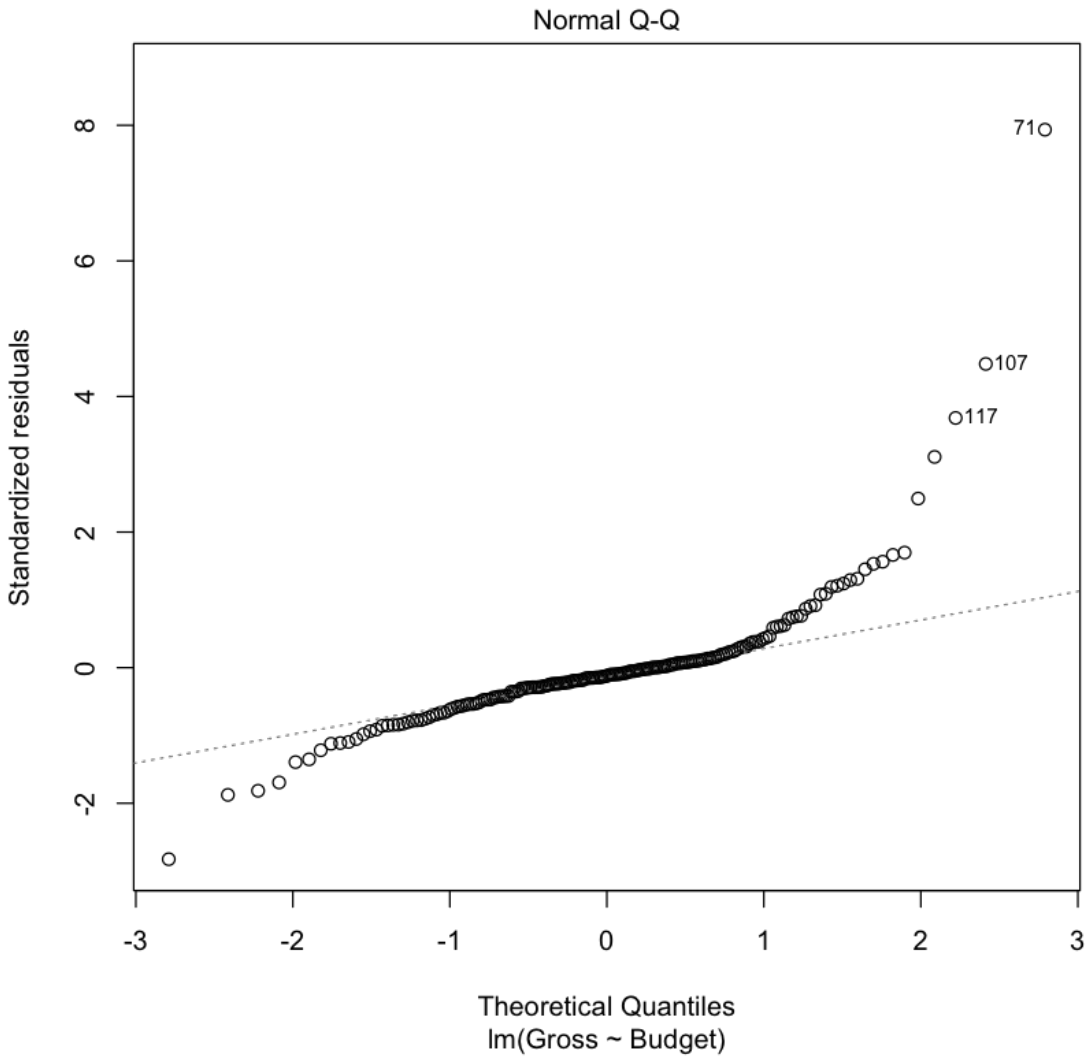
Data Source: http://www.bollymoviereviewz.com (http://www.bollymoviereviewz.com)

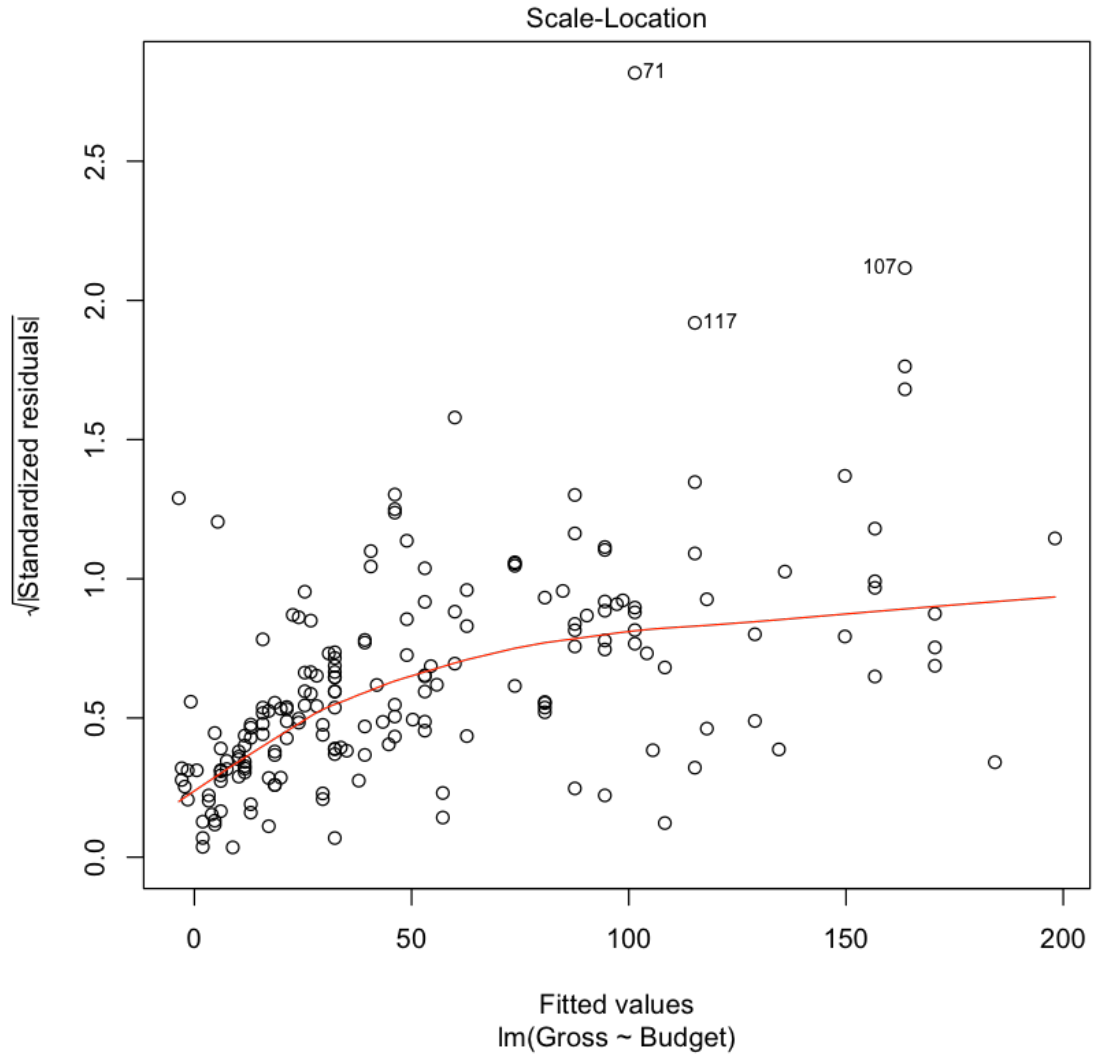
In [59]:

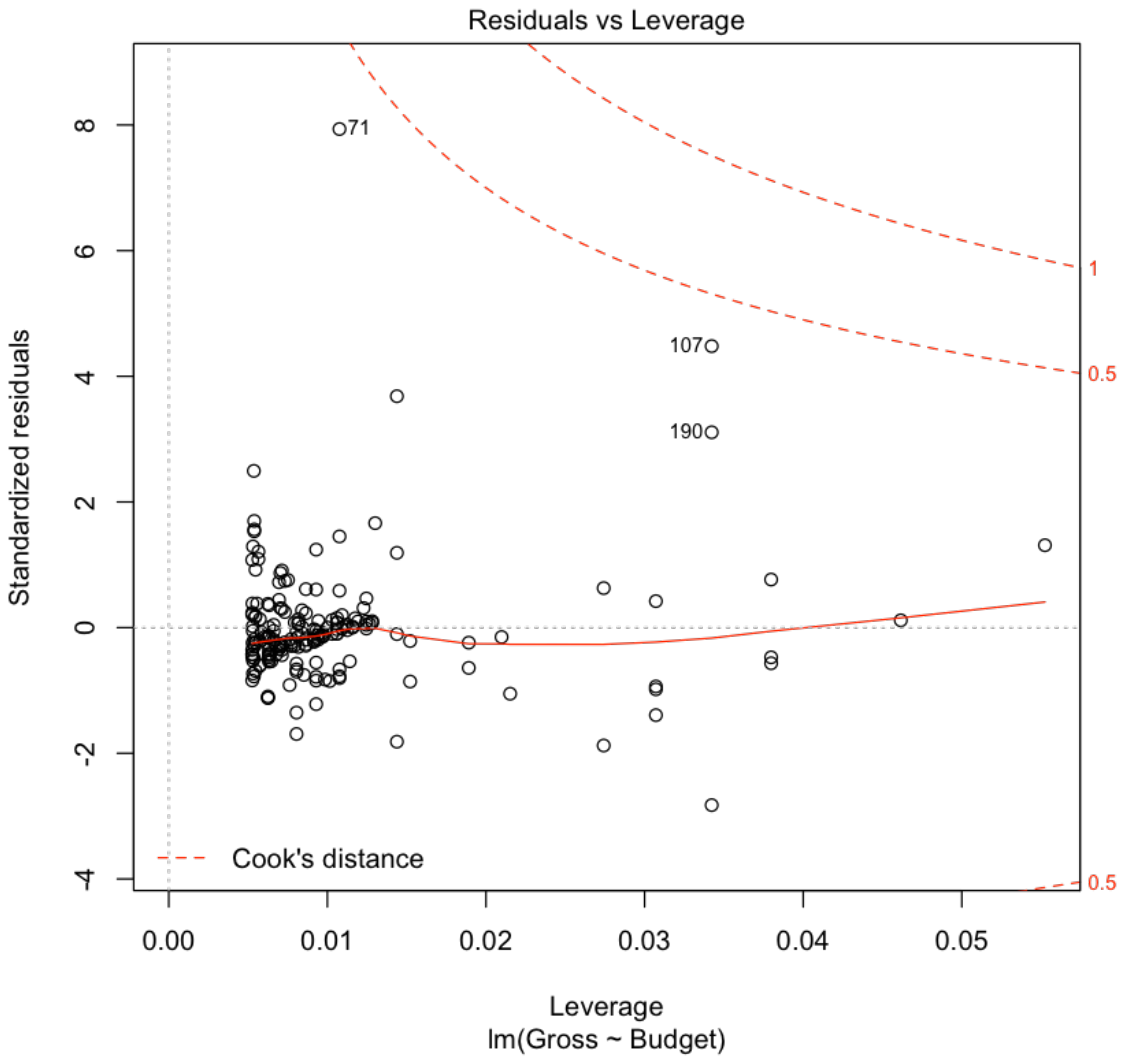
```
#filepath = "bollywood_boxoffice.csv"
library(RCurl) #a package that includes the function getURL(), whic
h allows for reading data from github.
library(ggplot2)
url = getURL(paste0("https://raw.githubusercontent.com/bzaharatos
/",
                    "-Statistical-Modeling-for-Data-Science-Applica
tions/",
                    "master/Modern%20Regression%20Analysis%20/Datas
ets/bollywood_boxoffice.txt"))
bollywood = read.csv(text = url, sep = "\t")
summary(bollywood)

lm_bollywood = lm(Gross ~ Budget, data = bollywood)
#par(mfrow = c(2,2))
plot(lm_bollywood)

#independence
p1=ggplot(lm_bollywood, aes(1:dim(bollywood)[1], .resid))+geom_poin
t()
p1= p1+stat_smooth(method="loess", col = "#CFB87C", se = FALSE, spa
n = 0.3)
p1=p1 + geom_hline(yintercept=0, col="#A2A4A3", linetype="dashed")
p1=p1+xlab("Index")+ylab("Residuals")
    p1<-p1+ggtitle("Residual vs Index")+theme_bw() +
    theme(plot.title = element_text(hjust=0.5))

p1

n = dim(bollywood)[1];
x = head(resid(lm_bollywood), n-1)
y = tail(resid(lm_bollywood), n-1)
cor(x,y)
srp = data.frame(x,y)
ggplot(srp, 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])) +
    ggtitle("Successive Residual Plot") +
    theme_bw() +
    theme(plot.title = element_text(hjust = 0.5))
```

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

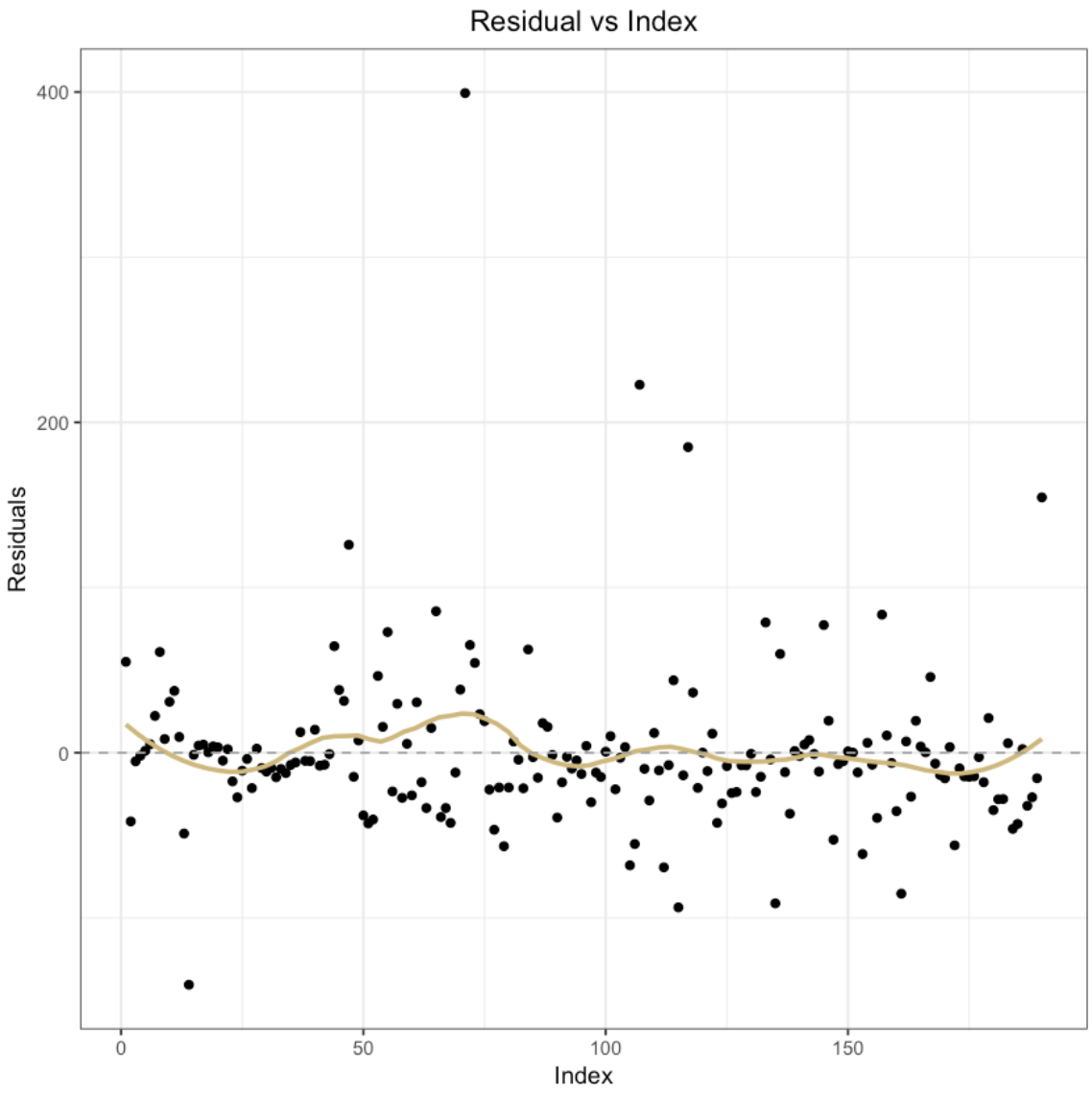


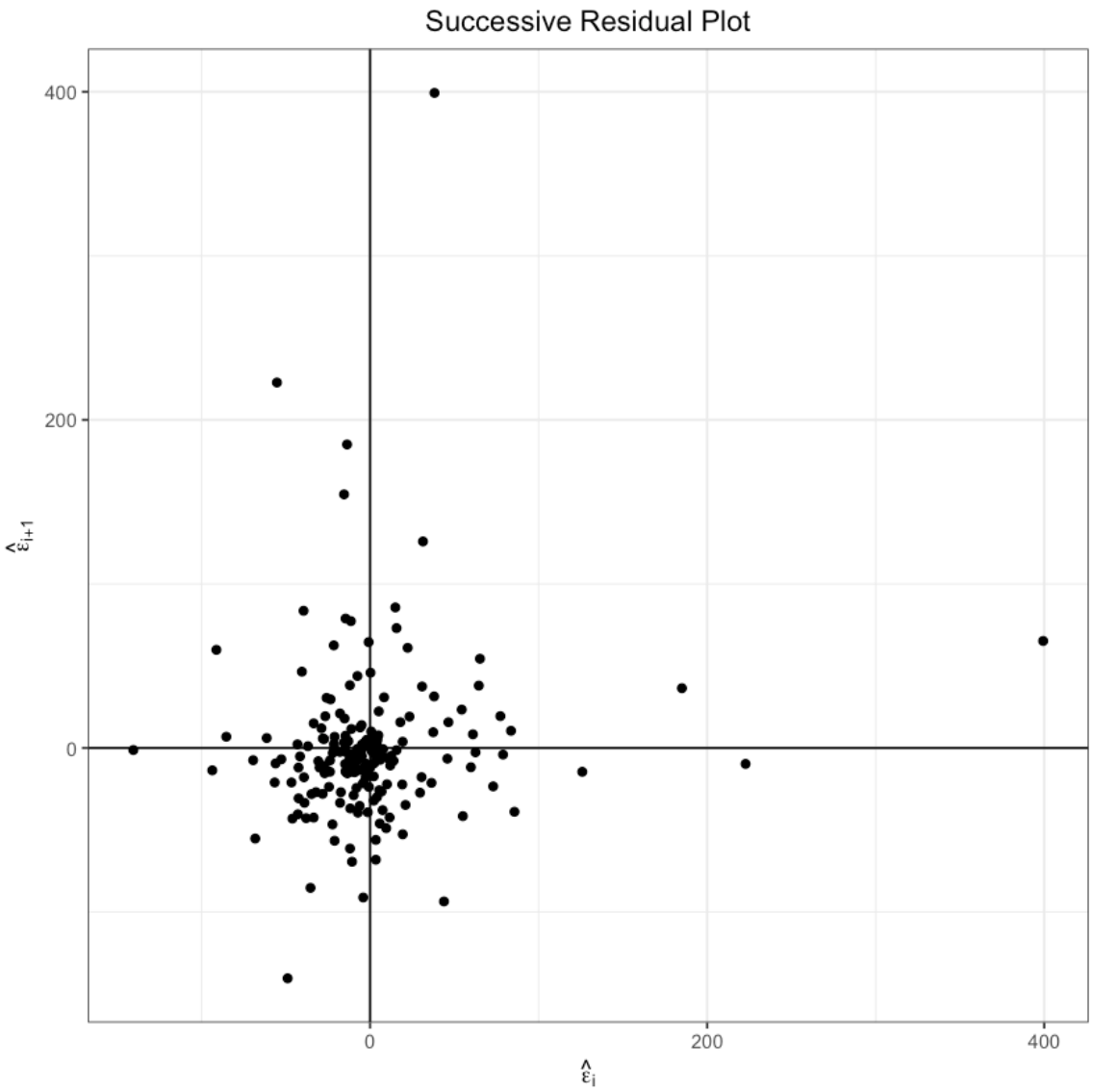

Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Gross ~ Budget)
71
107
117
-3
-2
-1
0
1
2
3
-2
0
2
4
6
8


Scale-Location
71
107
117
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
0
50
100
150
200
Fitted values
lm(Gross ~ Budget)



0.111594190315052


Residual vs Index
400
200
0
Residuals
0
50
100
150
Index



1. Linearity: the residual vs fitted plot shows a (perhaps weak) downward trend, suggesting some violation of the linearity assumption.
2. Independence: The residual vs index plot shows little structure, and the successive residuals plot shows a weak/no correlation. From this, we can conclude that we have no evidence of successive error terms being dependent. However, there may be other orderings of the data that show a dependence structure.
3. Constant variance: there is some evidence (perhaps weak) of non-constant variance. We see this in the residual vs fitted plot: there is more variability for large fitted values than for small fitted values.
4. Normality: The QQ-plot shows clear evidence of a deviation from normality. But this is likely due to the violation of linearity and constant variance. We should try to fix those issues and return to the normality assumption.

3. (b) Transformations

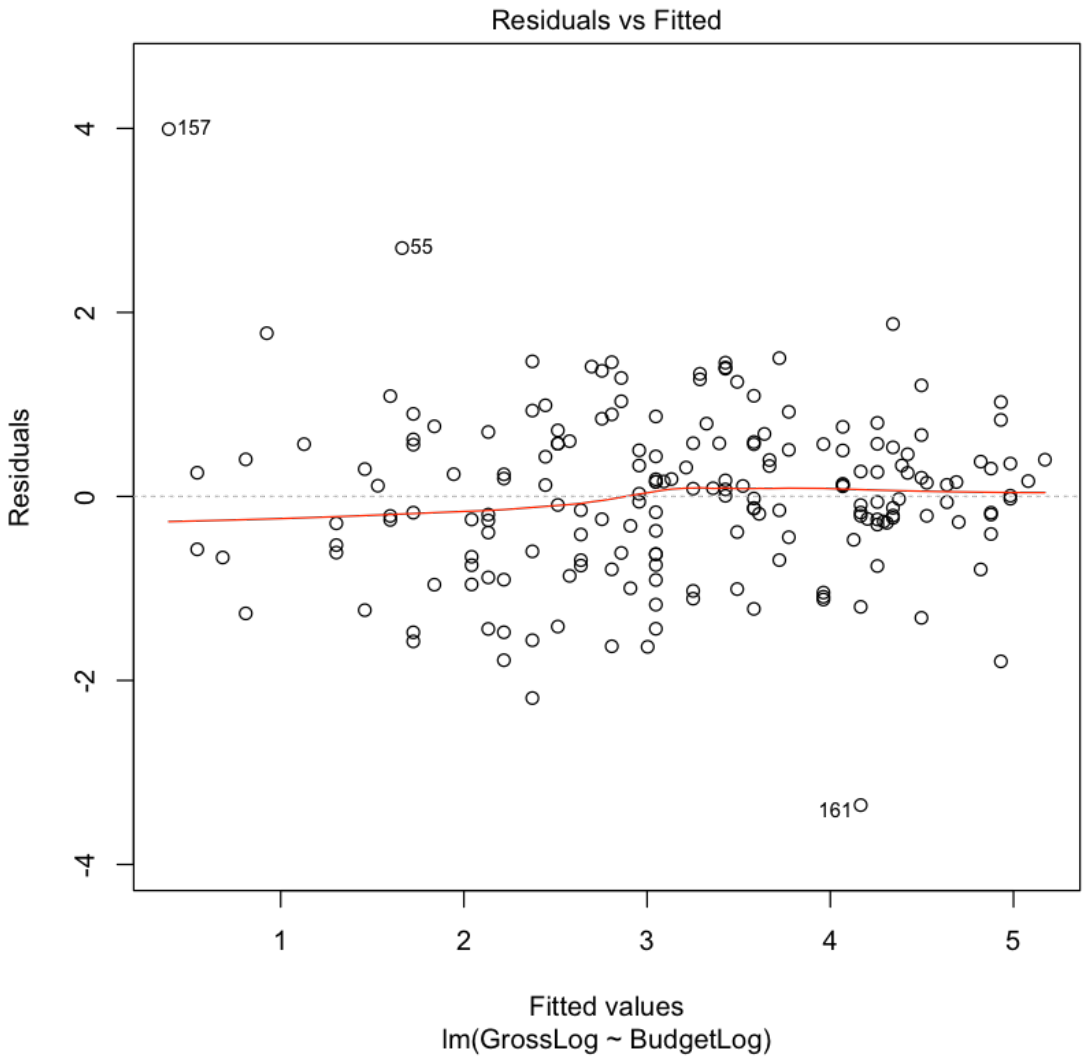
Notice that the Residuals vs. Fitted Values plot has a 'trumpet" shape to it, the points have a greater spread as the Fitted value increases. This means that there is not a constant variance, which violates the homoskedasticity assumption.

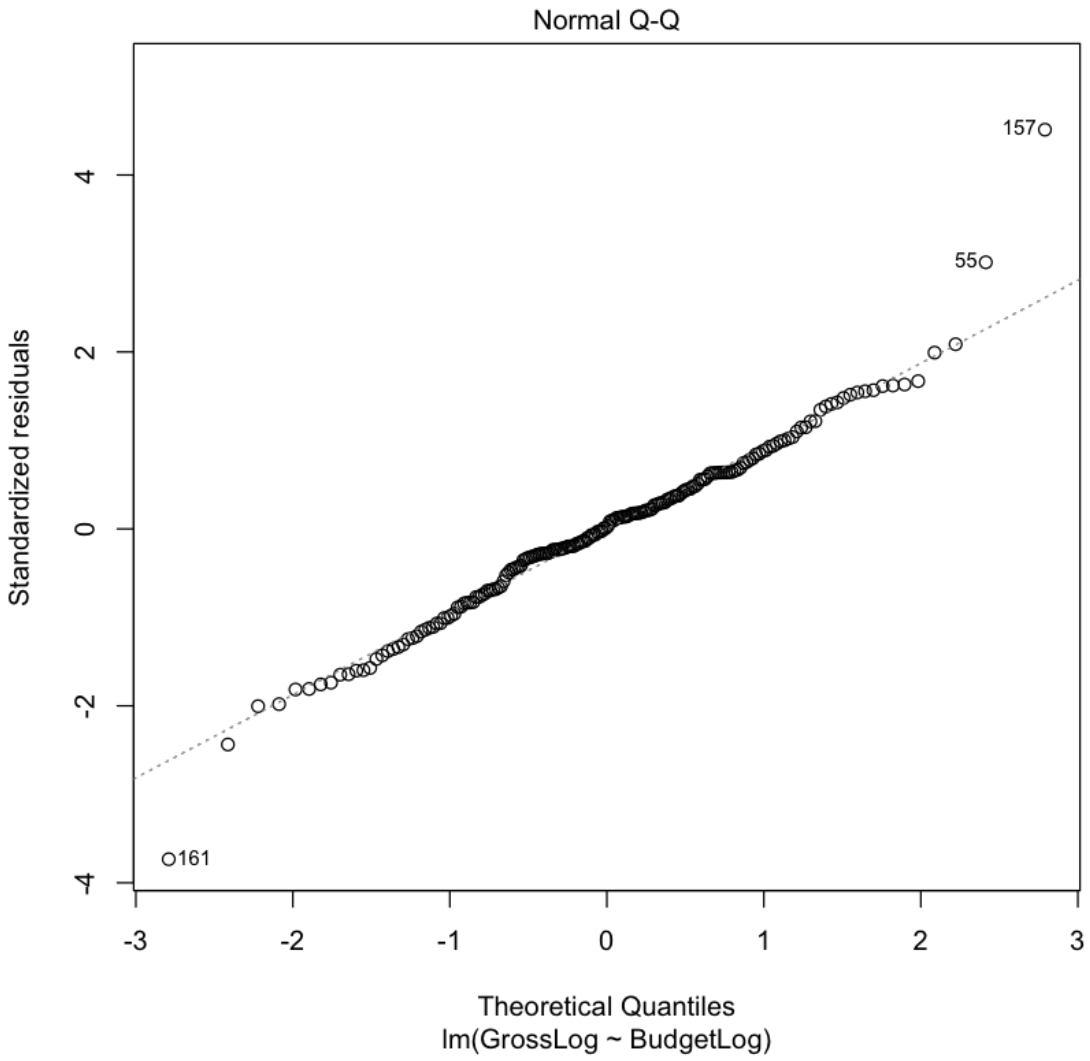
So how do we address this? Sometimes transforming the predictors or response can help stabilize the variance. Experiment with transfomrations on `Budget` and/or `Gross` so that, in the transformed scale, the relationship is approximately linear with a constant variance. Limit your transformations to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

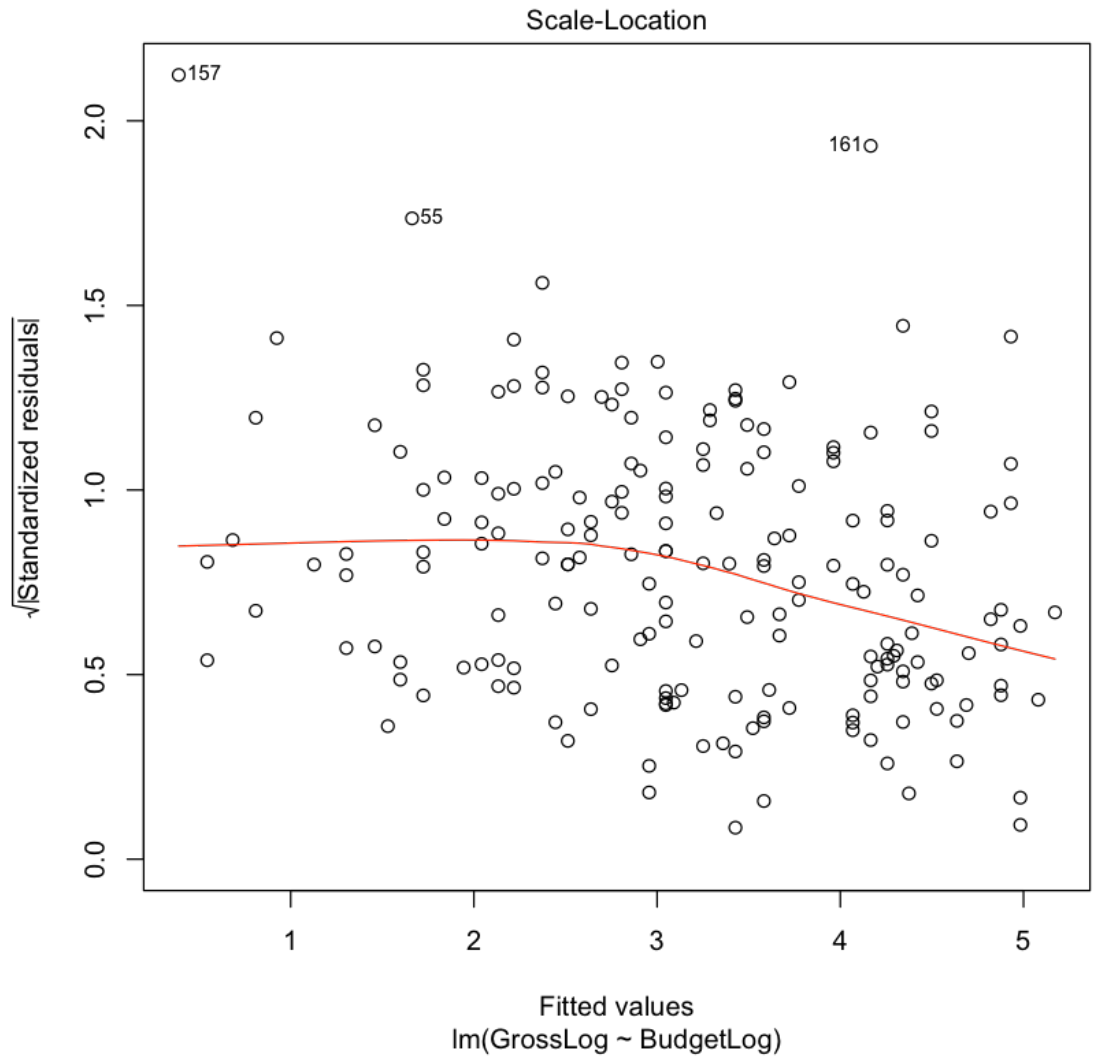
Note: There may be multiple transformations that fix this violation and give similar results. For the purposes of this problem, the transformed model doesn't have the be the "best" model, so long as it maintains both the linearity and homoskedasticity assumptions.

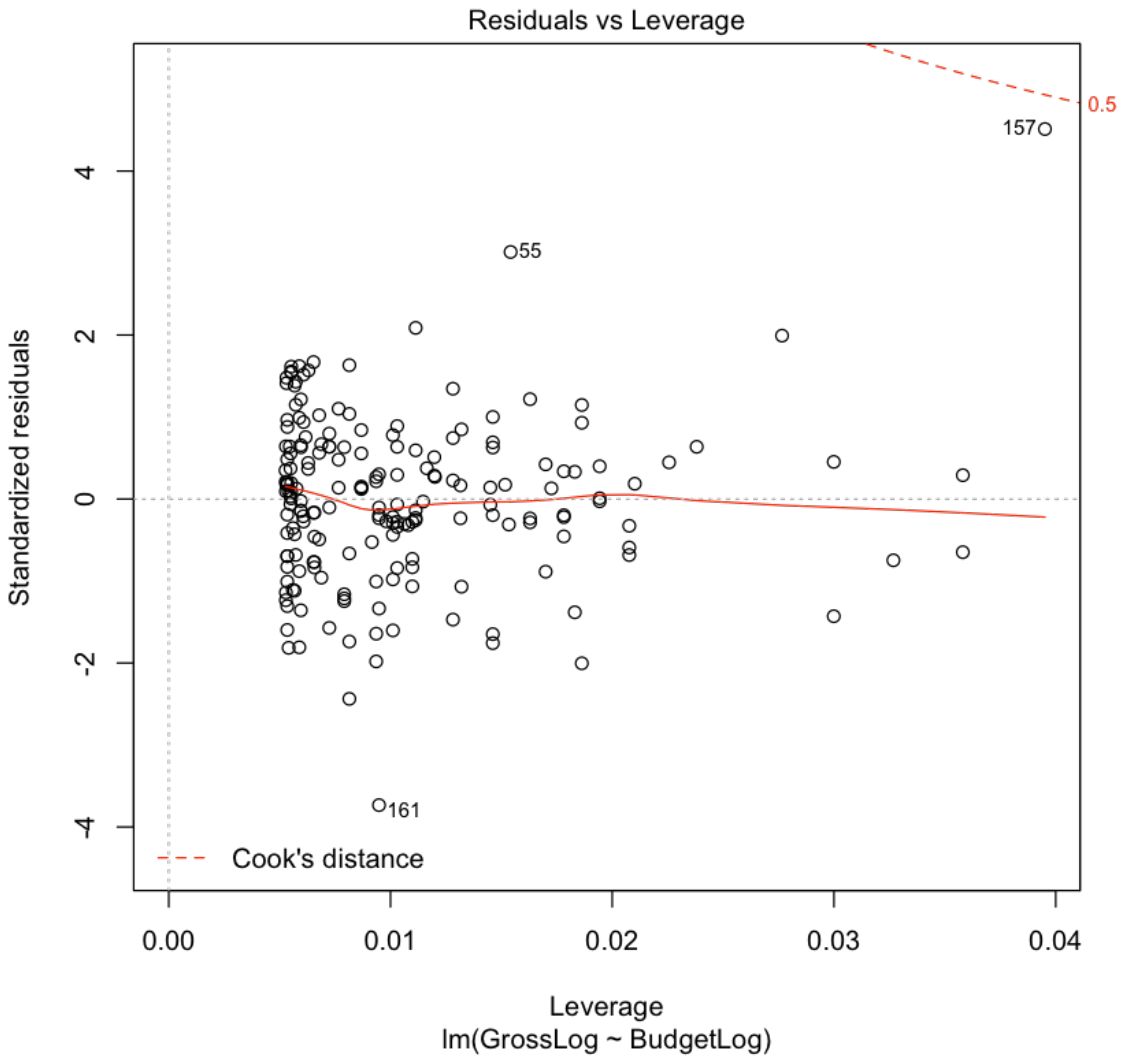
In [76]:

```
bollywood$BudgetLog = log(bollywood$Budget)
bollywood$GrossLog = log(bollywood$Gross)
bollywood$BudgetSqrt = sqrt(bollywood$Budget)
bollywood$GrossSqrt = sqrt(bollywood$Gross)

lm_log_log = lm(GrossLog ~ BudgetLog, bollywood)
plot(lm_log_log)
```


Residuals vs Fitted
157
55
161
Residuals
4
2
0
-2
-4
1
2
3
4
5
Fitted values
lm(GrossLog ~ BudgetLog)


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(GrossLog ~ BudgetLog)
157
55
161
-4
-2
0
2
4
-3
-2
-1
0
1
2
3


Scale-Location
157
161
55
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
1
2
3
4
5
Fitted values
lm(GrossLog ~ BudgetLog)



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?

In [77]:
```
summary(lm_log_log)
exp(1.44023)
```

```
Call:
lm(formula = GrossLog ~ BudgetLog, data = bollywood)

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 ***
BudgetLog    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
```

4.22166668868043

$$
\begin{aligned}
&log(Gross) = \widehat{\beta}_0 + \widehat{\beta}_1 \log(Budget) \\
\implies &Gross = exp\left(\widehat{\beta}_0 + \widehat{\beta}_1 \log(Budget)\right) \\
\implies &Gross = exp\left(\widehat{\beta}_0\right) Budget^{\widehat{\beta}_1} \\
\implies &Gross = exp\left(-1.44023\right) Budget^{1.32} \\
\implies &Gross = 4.22 Budget^{1.32}
\end{aligned}
$$

If the **log** budget is increased by one unit, then the **log** gross increases by 1.32, on average. However, the transformed model does not have a similar interpretation. The change in `Gross` for a change in `Budget` from 80 to 81 is different than for a change from 81 to 82.

In [87]:
```
b = coef(lm_log_log); b
x = c(80, 81, 82)
g1 = exp(b[1])*x[1]^b[2]
g2 = exp(b[1])*x[2]^b[2]
g3 = exp(b[1])*x[3]^b[2]

g2 - g1

g3 - g2
```

(Intercept)	-1.44023488495719
BudgetLog	1.31954649224769

(Intercept): 1.27036469213314

(Intercept): 1.2753863101067