Ve406 Lecture 6

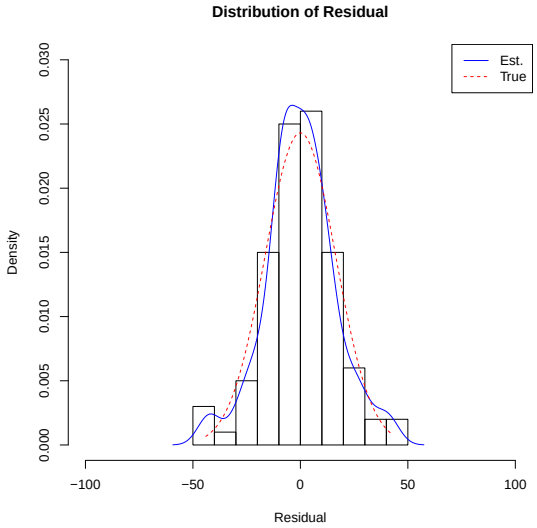
Jing Liu

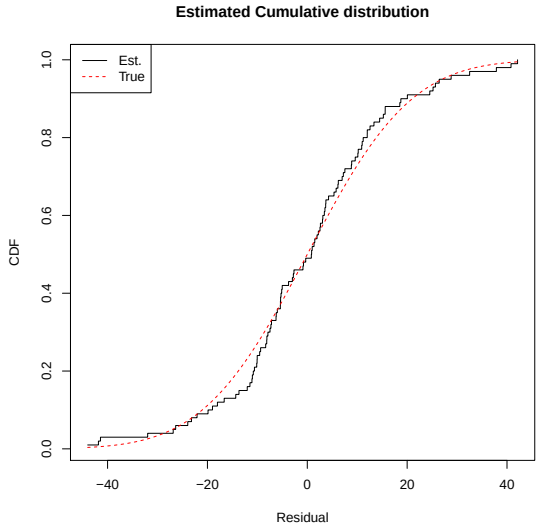
UM-SJTU Joint Institute

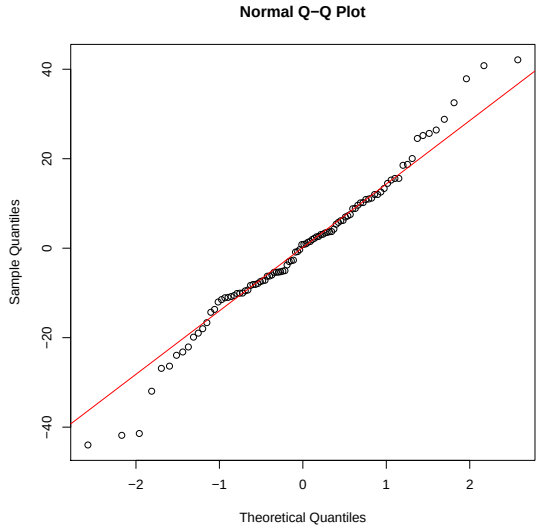
May 30, 2018

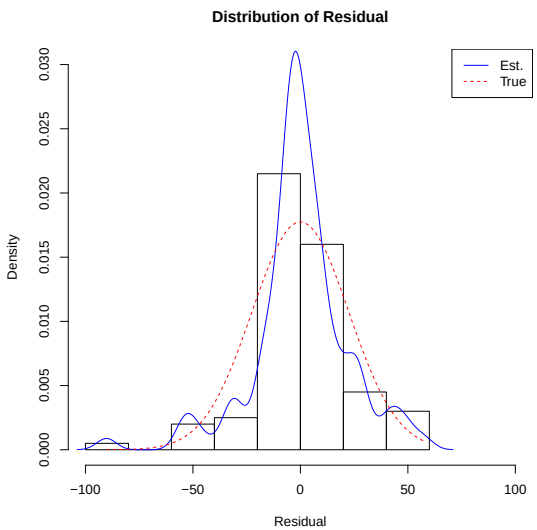
- Normality can be checked by plotting estimated /distribution

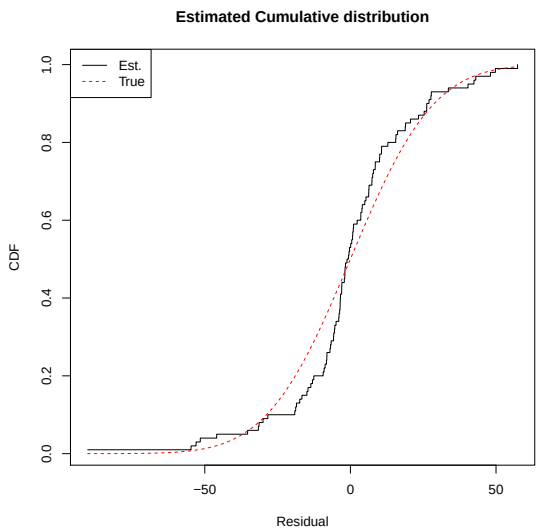
4. The errors follow the normal distribution of $\mathrm{N}\left(0, \sigma^2\right)$.

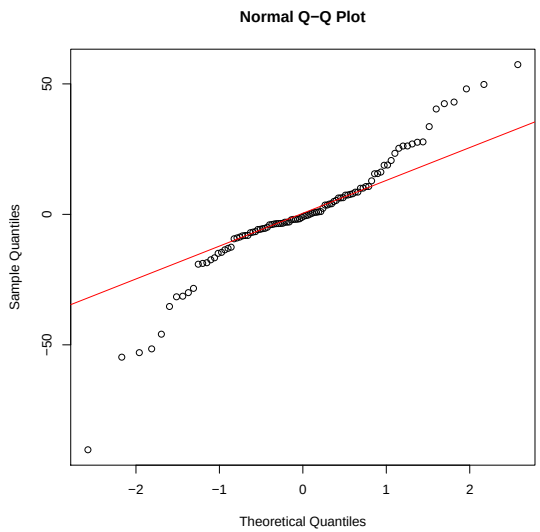












```
> tmp = seq(-100, 100, length.out = 100)
> res = residuals(lmlist[[1]])
> # res = residuals(msl.lm)
>
> sigma = sqrt(sum(res^2) / (n-2))
> # Density function
> hist(res, probability = TRUE,
+      xlim = c(-100, 100), ylim = c(0, 0.03),
+      xlab = "Residual",
+      main = "Distribution of Residual")
>
> lines(density(res), col = "blue")
>
> lines(tmp, dnorm(tmp, sd = sigma),
+       col ="red", lty = 2)
>
> legend("topright", legend = c("Est.", "True"),
+        lty = c(1,2), col = c(4, 2))
```

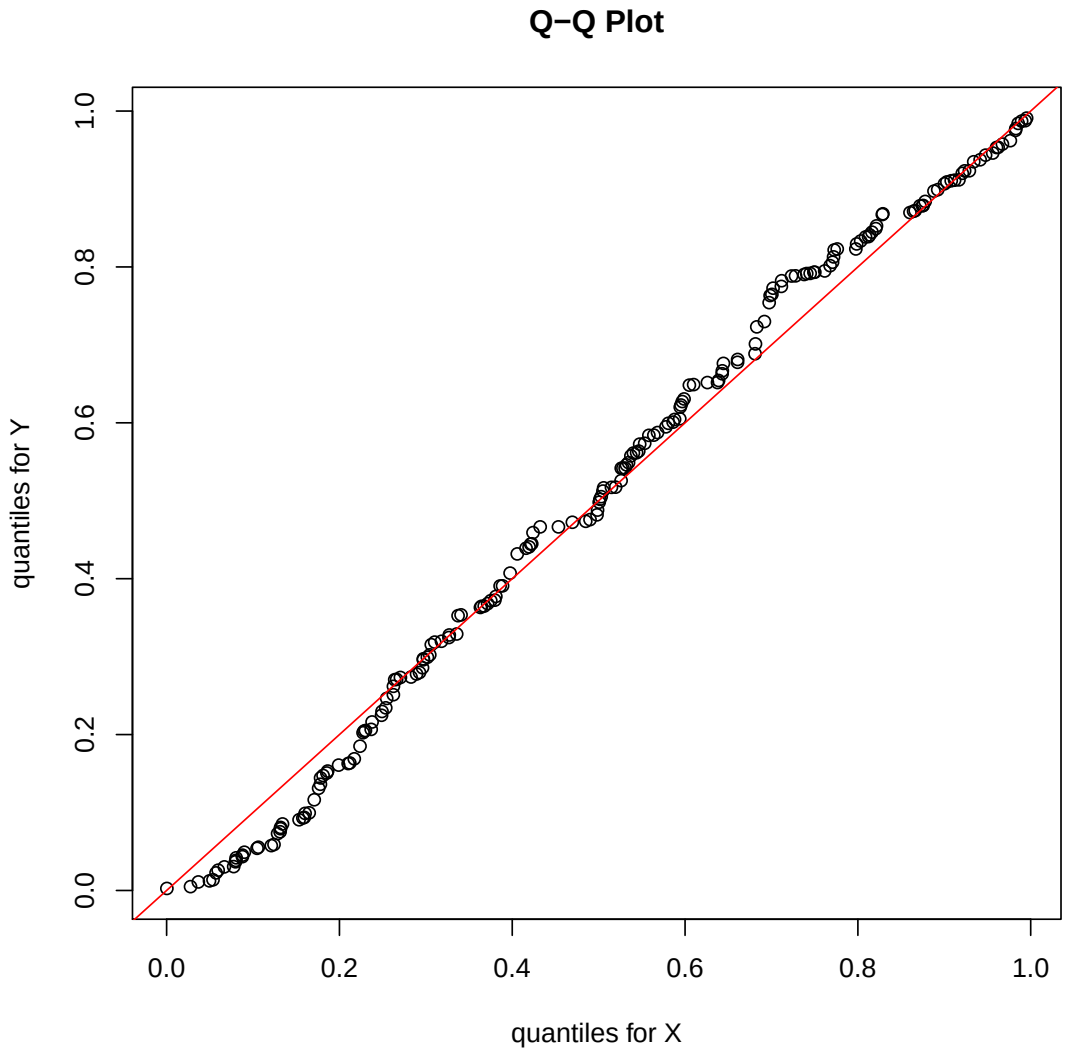
```
> # Distribution function
> sample_quantile = sort(res); sample_cdf = (1:n)/n
>
> tmp = seq(min(sample_quantile),
+           max(sample_quantile), length.out = 100)
>
> plot(sample_quantile, sample_cdf,
+      xlab = "Residual", ylab = "CDF",
+      main = "Estimated Cumulative distribution",
+      type = "s")
>
> lines(tmp, pnorm(tmp, sd = sigma),
+       col = 2, lty = 2)
>
> legend("topleft", legend = c("Est.", "True"),
+        lty = c(1, 2), col = c(1, 2))
```

```
> # Quantile-Quantile Normal plot
> qqnorm(res)
> qqline(res, col = "red")
```

- In general, we can compare the quantiles of two arbitrary distributions

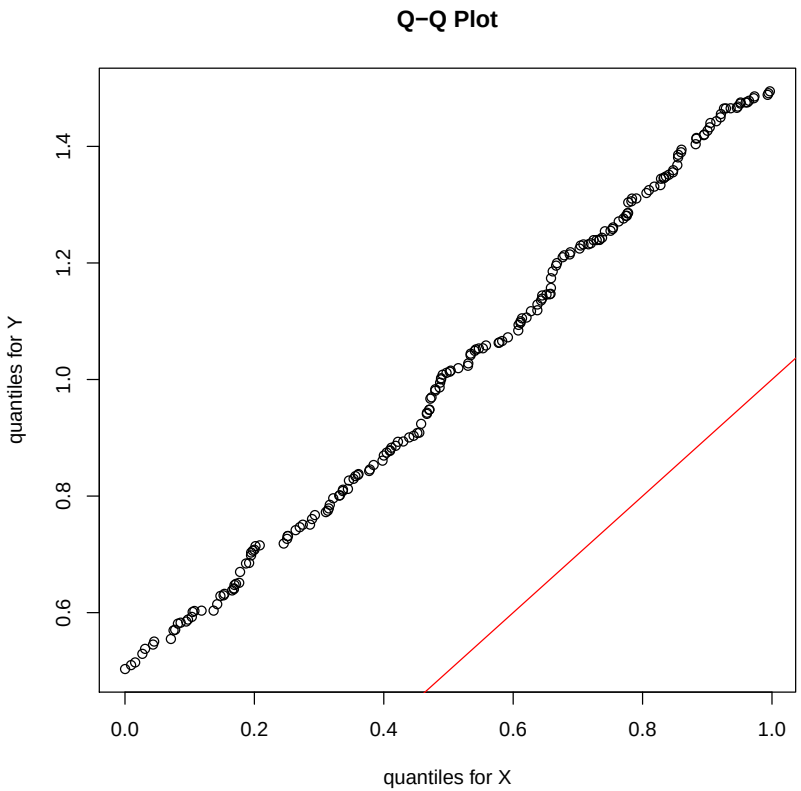
```
> num = 200 # number of observation

> # Compare samples from the same distribution
> x = runif(num, min = 0, max = 1)
> y = runif(num, min = 0, max = 1)

> qqplot(x, y, main = "Q-Q Plot",
+        xlab = "quantiles for x",
+        ylab = "quantiles for y")
>
> abline(a = 0, b = 1, col = 2)
```

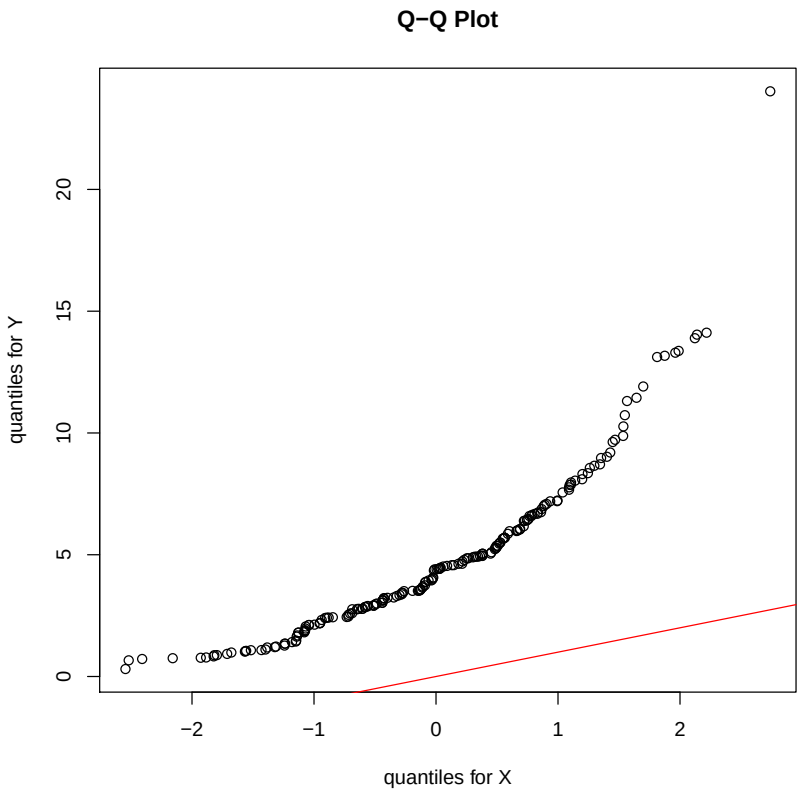
Q: What do you expect to see?



Q: What do you expect to see in the Q-Q plot if we have the following instead?

```
> # The same shape, but different center
> x = runif(num, min = 0, max = 1)
> y = runif(num, min = 0.5, max = 1.5)
> qqplot(x, y, main = "Q-Q Plot",
+        xlab = "quantiles for X",
+        ylab = "quantiles for Y")
> abline(a = 0, b = 1, col = 2)

> # One is Skewed to the right, one symmetric
> x = rnorm(num)
> y = rchisq(num, df = 5)
> qqplot(x, y, main = "Q-Q Plot",
+        xlab = "quantiles for X",
+        ylab = "quantiles for Y")
> abline(a = 0, b = 1, col = 2)
```

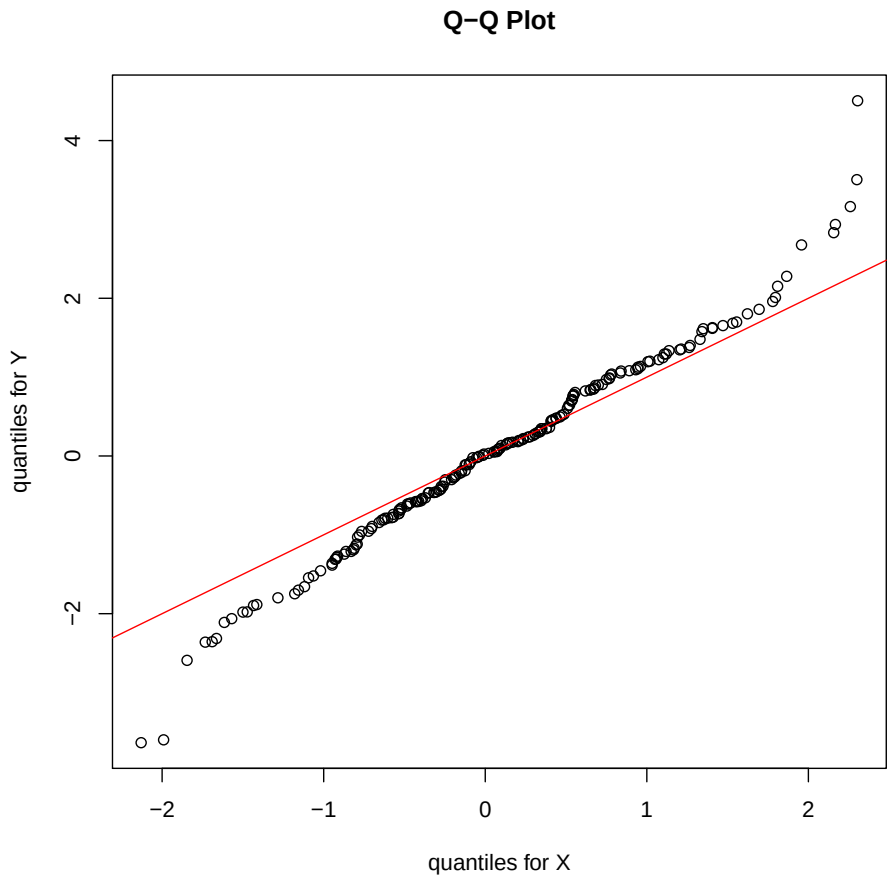


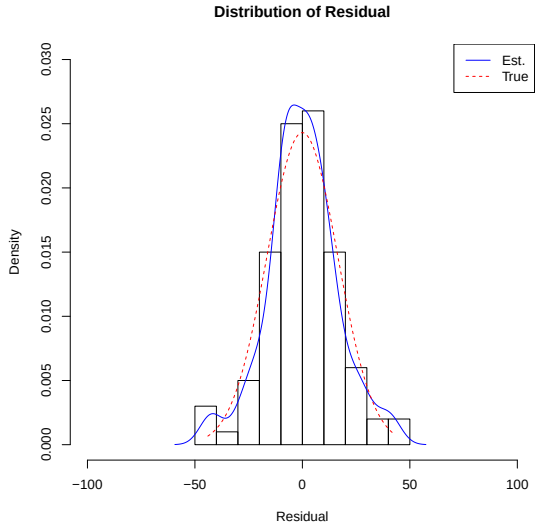


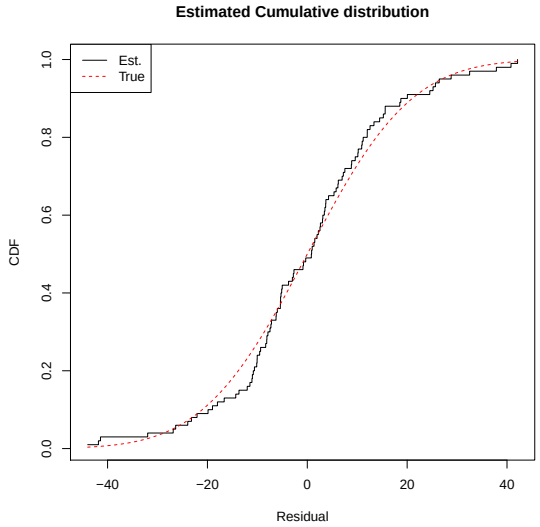
Q: How about the following?

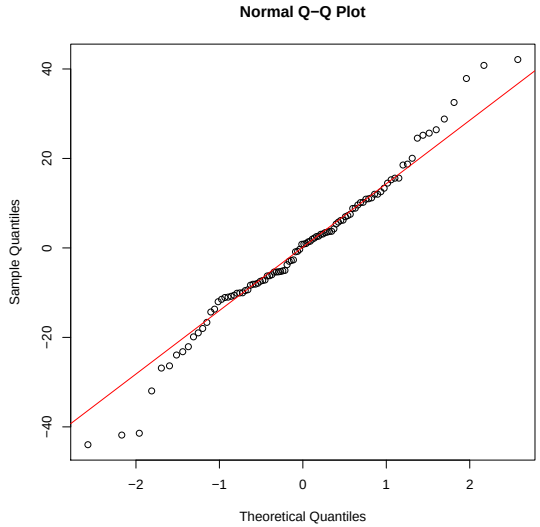
```
> # One has longer tail than the other
> x = rnorm(num)
> y = rt(num, df = 5)
```

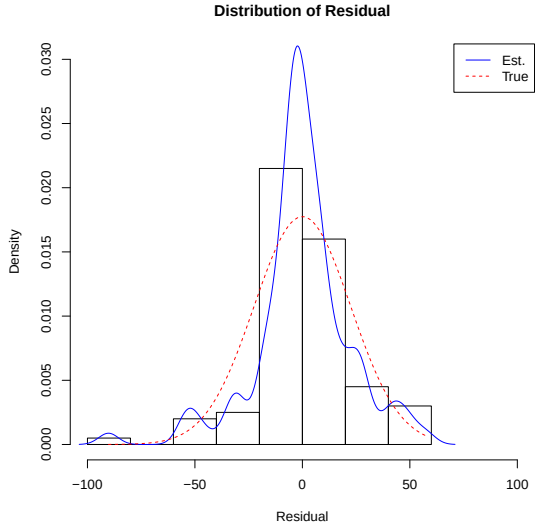
```
> qqplot(x, y, main = "Q-Q Plot",
+          xlab = "quantiles for X",
+          ylab = "quantiles for Y")
> abline(a = 0, b = 1, col = 2)
```

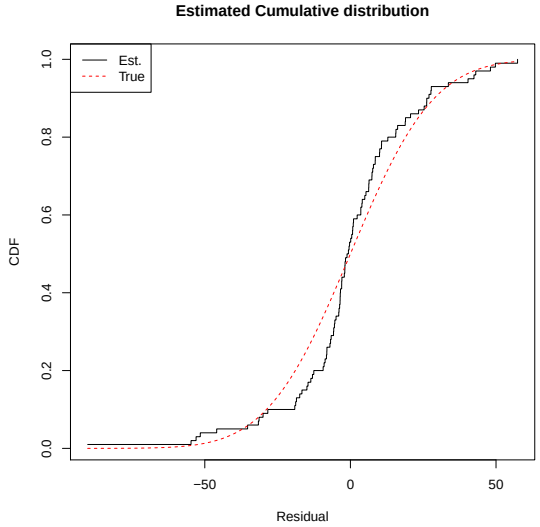


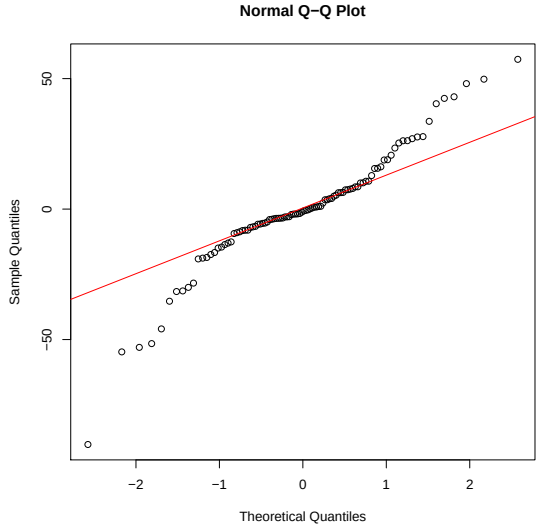

Distribution of Residual
Est.
True
Density
Residual


Estimated Cumulative distribution
Est.
True
CDF
Residual


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles

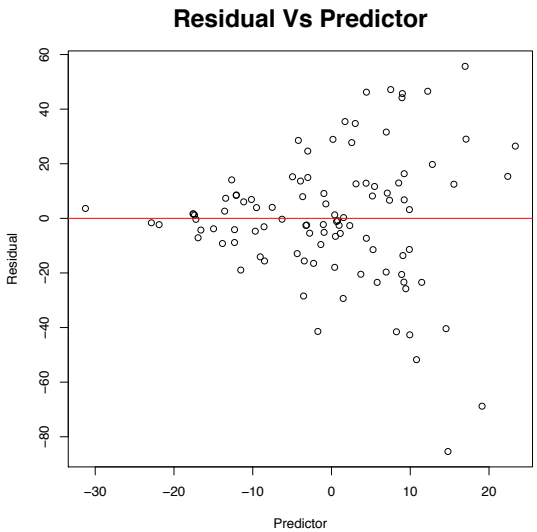

Distribution of Residual
Est.
True
Density
Residual

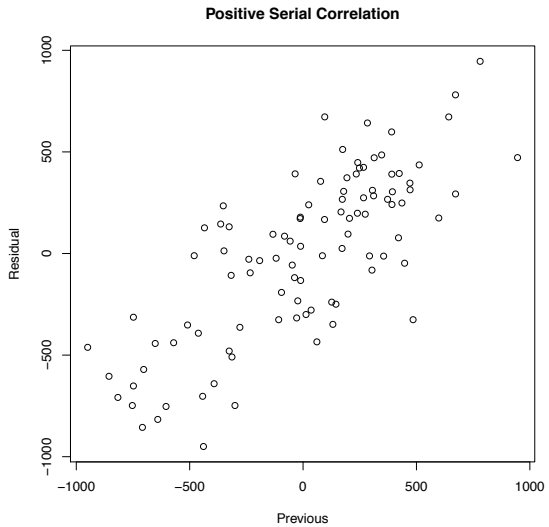

Estimated Cumulative distribution
Est.
True
CDF
Residual

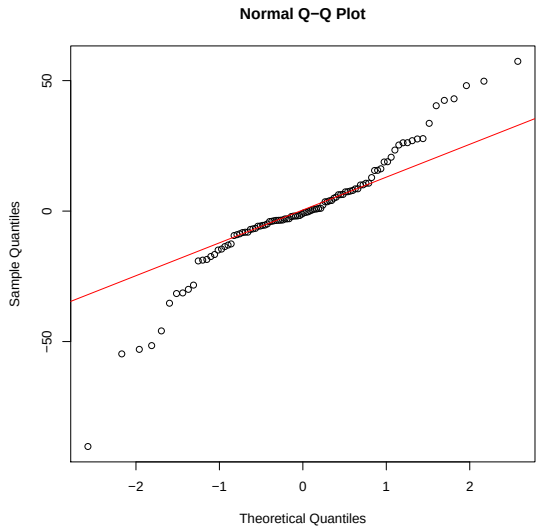

Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles

Q: What happens if one of assumptions is violated?

```
> mq = beta0 + beta1 * x.vec^2
> y.mq.vec = rlogis(n, location = mq, scale = s)
>
> mc = beta0 + beta1 * x.vec^3
> y.mc.vec = rlogis(n, location = mc, scale = s)
>
> me = beta0 + beta1 * exp(x.vec)
> y.me.vec = rlogis(n, location = me, scale = s)
```







- There are three approaches:

1. Transformation on the variables
2. Switch to advanced Models
3. Do nothing!

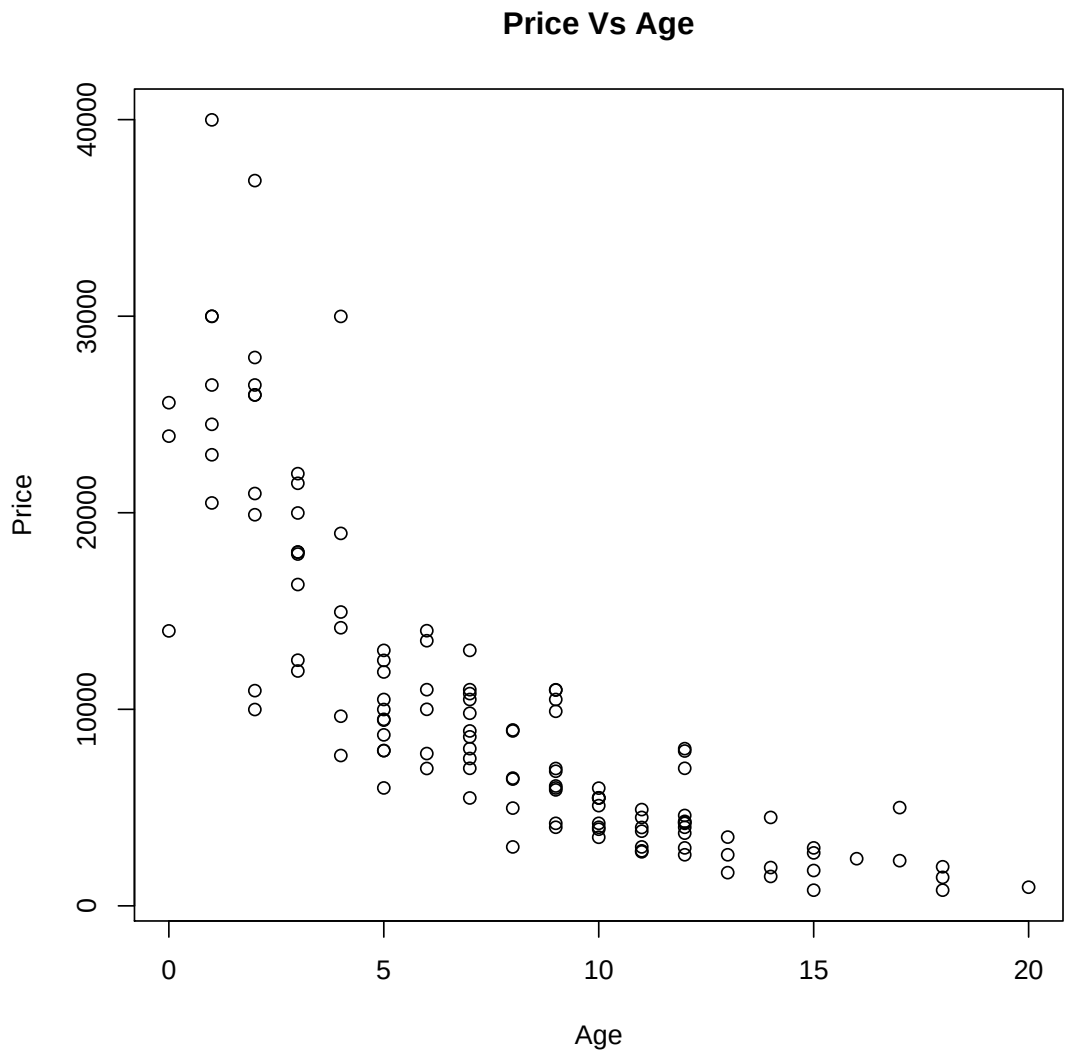
- To illustrate how transformation can help, consider the following dataset

```
> oc.df =
+   read.table("~/Desktop/old_car.txt",
+              header = TRUE)
> str(oc.df)
```

```
'data.frame':   123 obs. of  2 variables:
 $ year : int  79 82 83 ...
 $ price: int  2950 5900 2999 ...
```

```
> # Data was collect in 1991
> age = 91 - oc.df$year; n = nrow(oc.df)
```

```
> plot(age, oc.df$price, main = "Price Vs Age",
+       xlab = "Age", ylab = "Price")
```



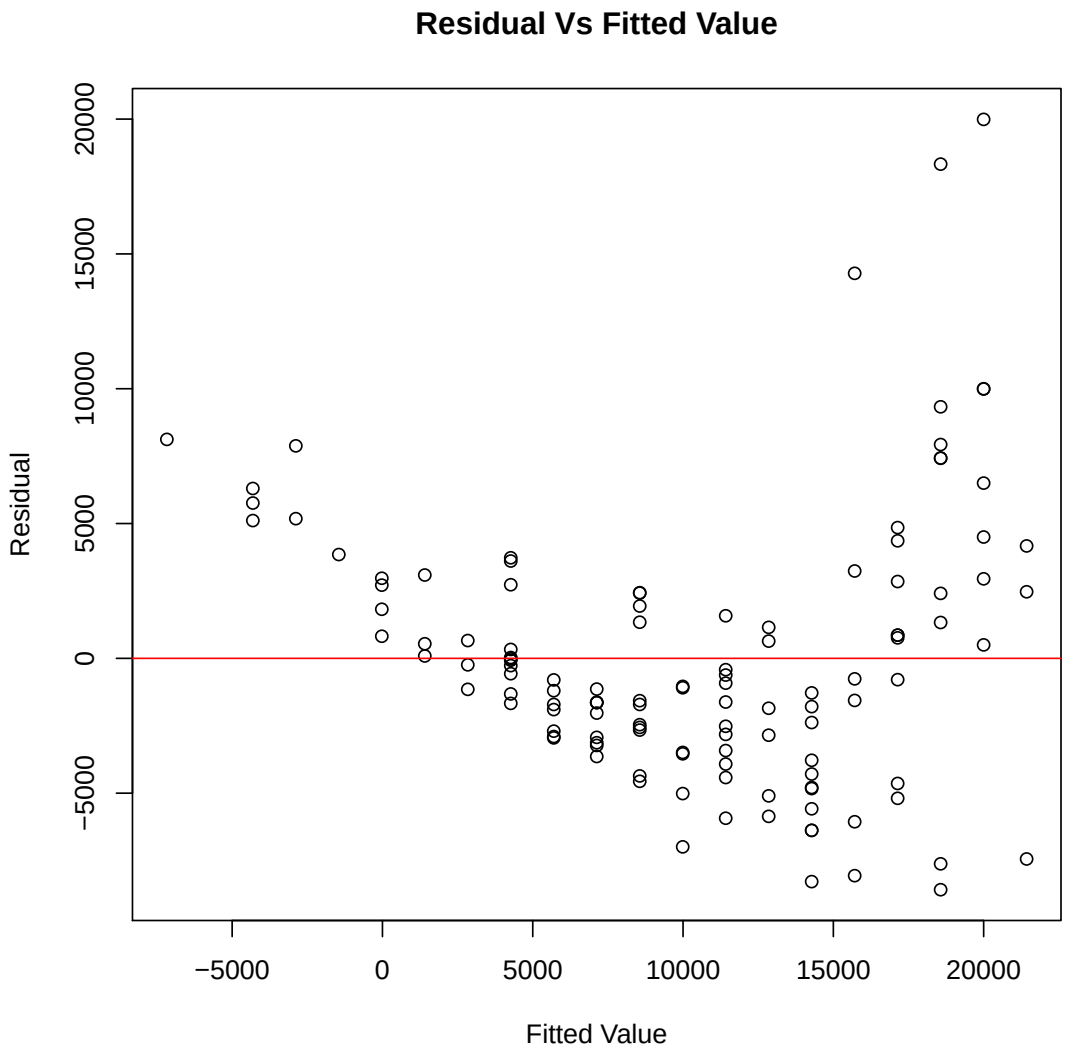
```
> oc.LM = lm(price~age, data = oc.df)
> summary(oc.LM)
```

```
Call:
lm(formula = price ~ age, data = oc.df)

Residuals:
    Min      1Q  Median      3Q     Max
-8579.5 -2938.9  -919.1  2453.2 19990.4

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept)  21429.6      870.7   24.61   <2e-16 ***
age          -1430.1       96.3  -14.85   <2e-16 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 4950 on 121 degrees of freedom
Multiple R-squared:  0.6457,    Adjusted R-squared:  0.6428
F-statistic: 220.5 on 1 and 121 DF,  p-value: < 2.2e-16
```

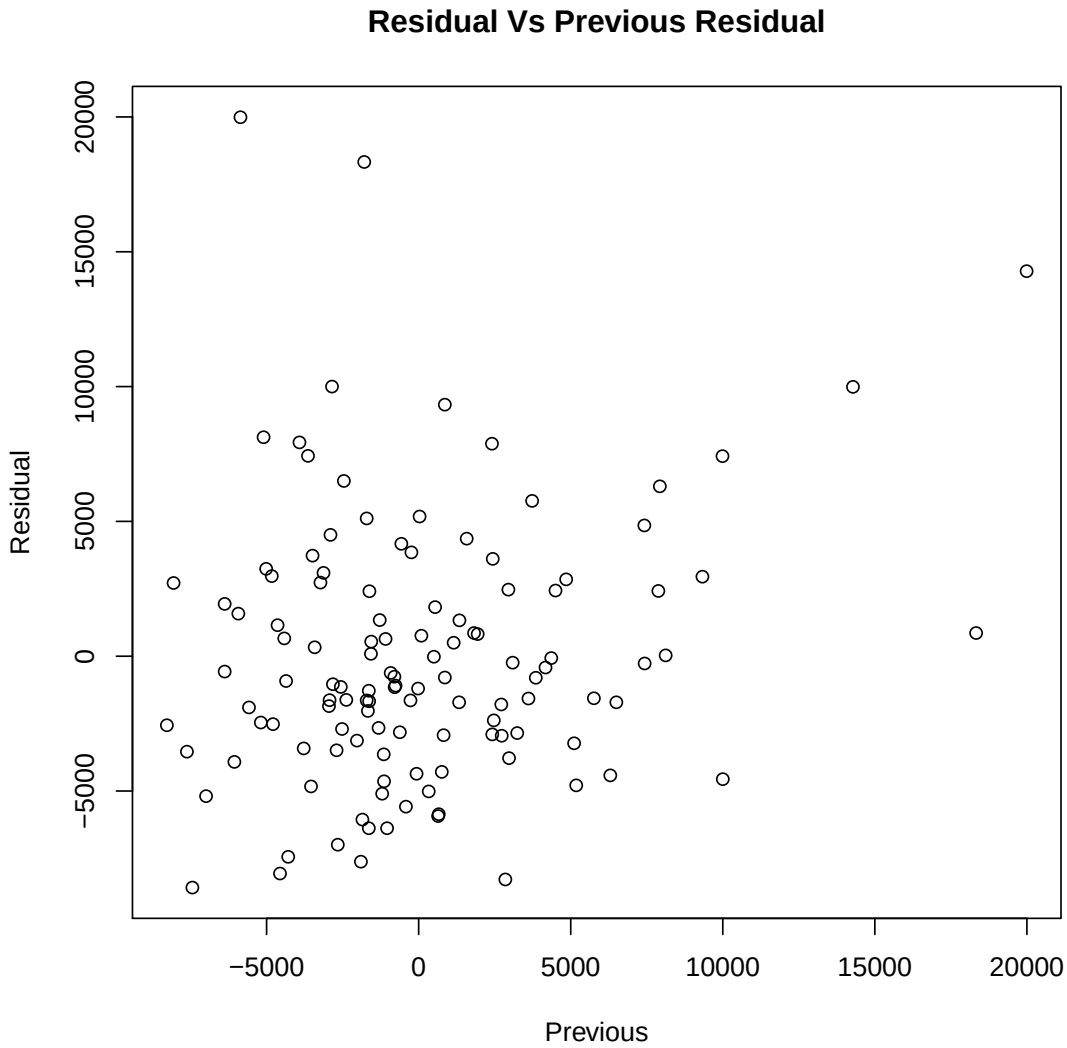
- But from the last plot, you probably can guess the above is mostly rubbish!

```
> plot(oc.LM$fitted.values, oc.LM$residuals,
+      main = "Residual Vs Fitted Value",
+      xlab = "Fitted Value", ylab = "Residual")
```

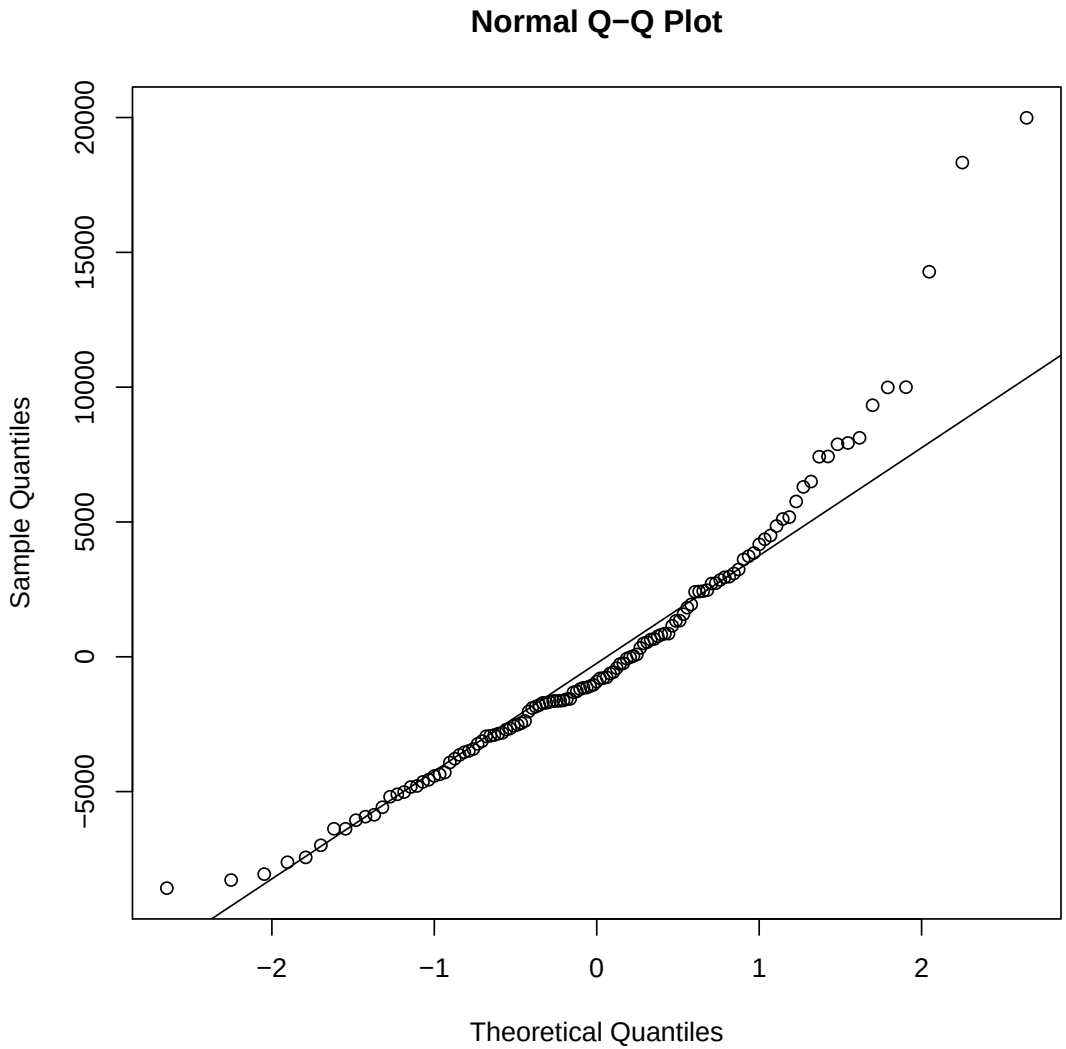
```
> abline(h = 0, col = "red")
```



```
> plot(oc.LM$residuals[-n], oc.LM$residuals[-1],
+       xlab = "Previous", ylab = "Residual",
+       main = "Residual Vs Previous Residual")
```



```
> qqnorm(oc.LM$residuals); qqline(oc.LM$residuals)
```

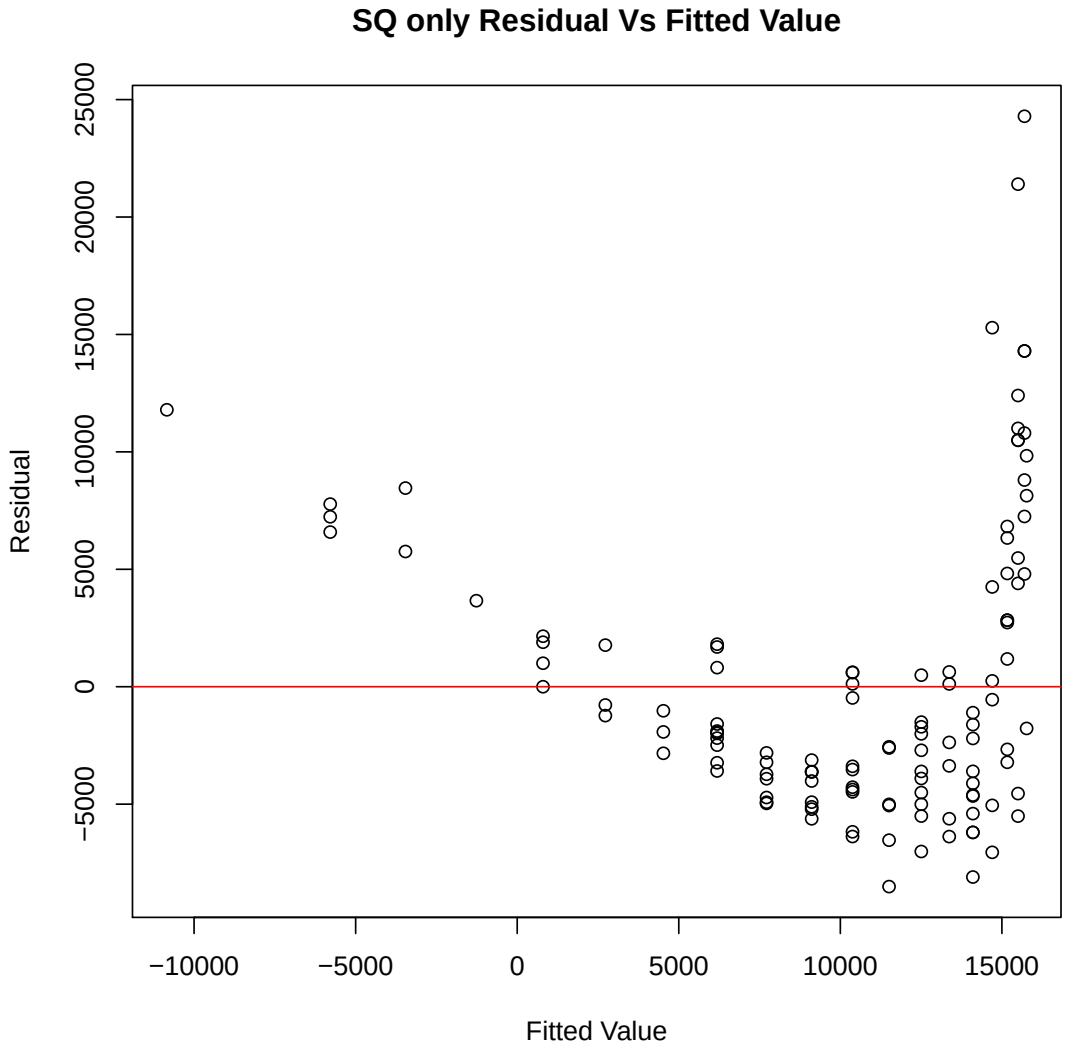


- It seems everything can go wrong goes wrong!
- You might suggest the followings, the second of which is the correct move

```
> ocq.LM = lm(price~I(age^2), data = oc.df)
>
> plot(ocq.LM$fitted.values, ocq.LM$residuals,
+      main = "SQ only Residual Vs Fitted Value",
+      xlab = "Fitted Value", ylab = "Residual")
> abline(h = 0, col = "red")
>
> oclq.LM = lm(price~age+I(age^2), data = oc.df)
>
> plot(oclq.LM$fitted.values, oclq.LM$residuals,
+      main = "Quad Residual Vs Fitted Value",
+      xlab = "Fitted Value", ylab = "Residual")
> abline(h = 0, col = "red")
```

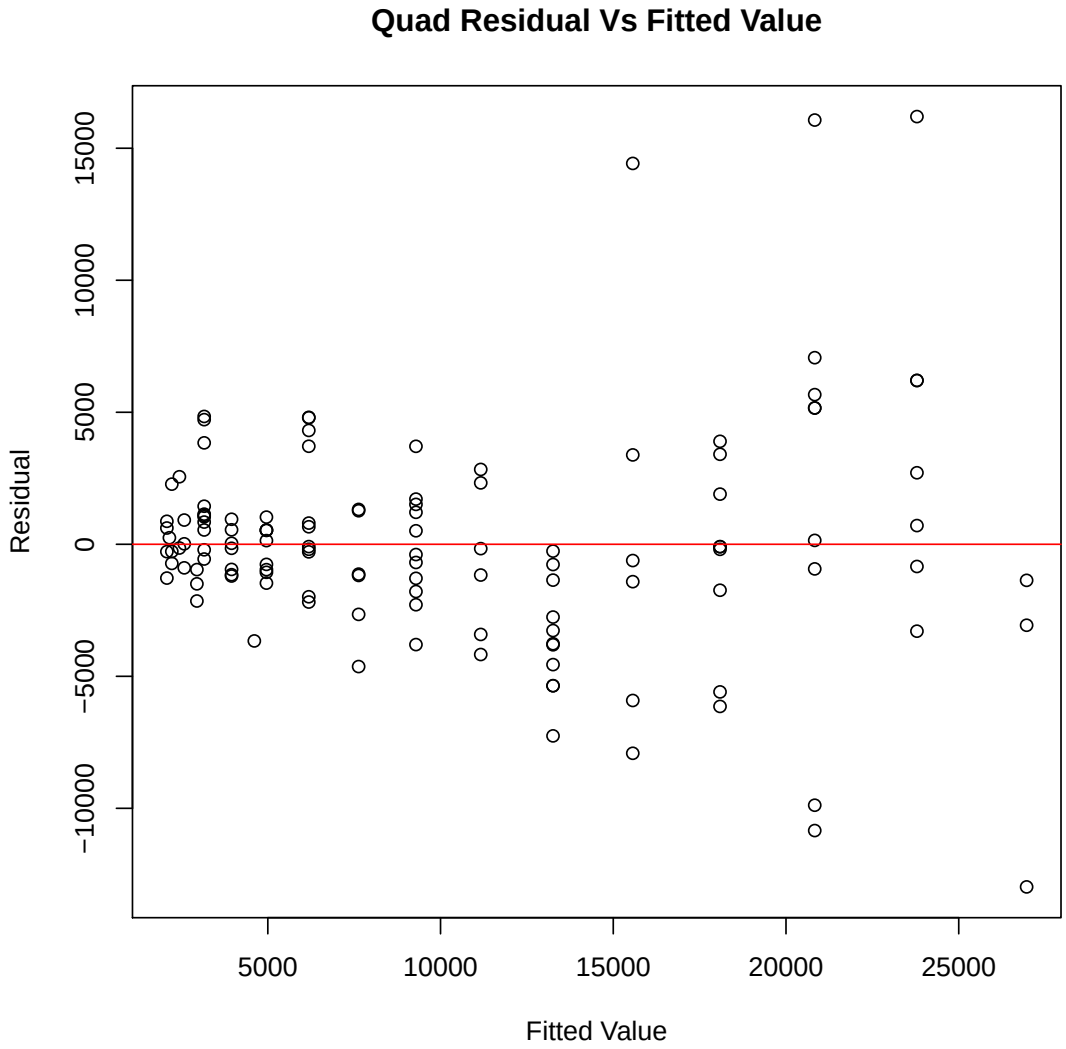
- However, neither leads to a good end in this case.

- We don't usually do this, underfitting is much more dangerous.

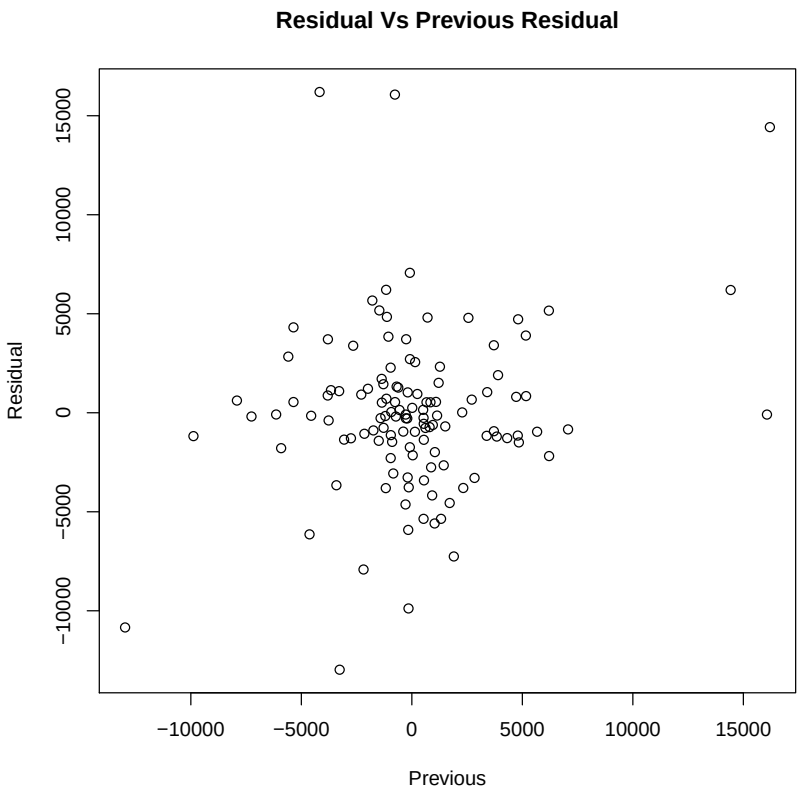


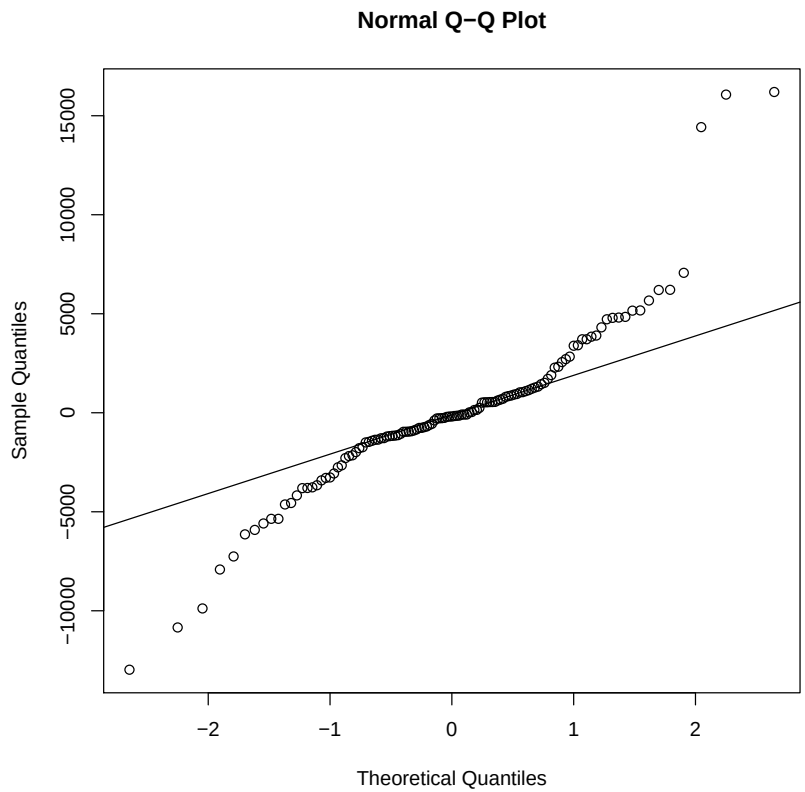
- and it is not helpful at all in this case.

- It seems the polynomial term helps to a certain degree



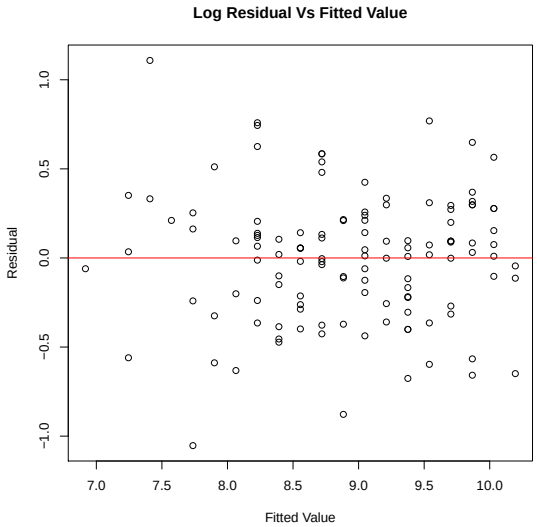
```
> plot(oclq.LM$residuals[-n], oclq.LM$residuals[-1],
+       xlab = "Previous", ylab = "Residual",
+       main = "Residual Vs Previous Residual")

> qqnorm(oclq.LM$residuals); qqline(oclq.LM$residuals)
```

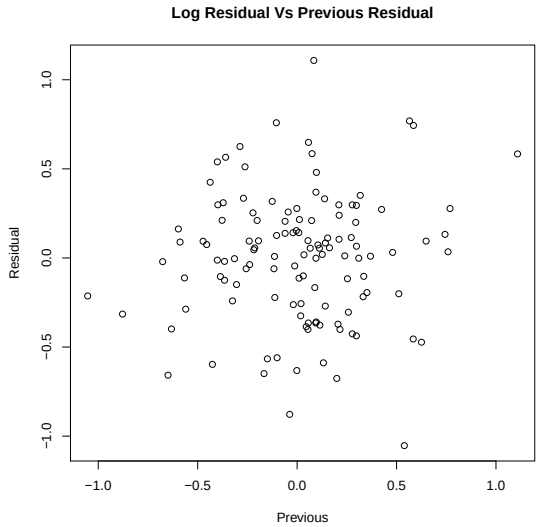


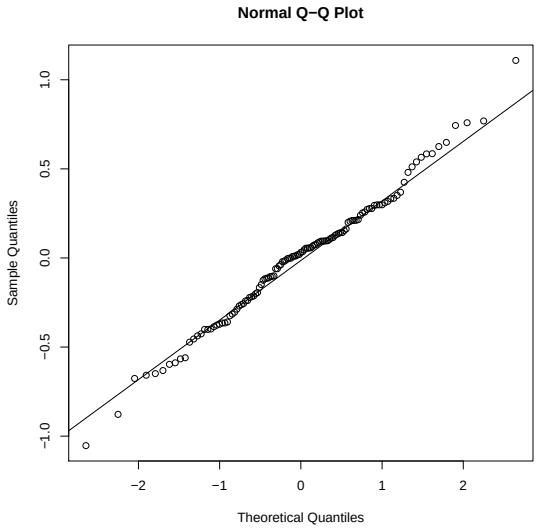


- Assumption 1. and 3. must be fixed first.
- Normality only affects the "precision" of our inference, and we can rely on CLT when the data size is big and the distribution is reasonably symmetric.
- If constant variance is the only issue, often we can fixed it by transform Y.

```
> oclog.LM = lm(log(price)~age, data = oc.df)
```







```
> plot(oclog.LM$fitted.values, oclog.LM$residuals,
+      main = "Log Residual Vs Fitted Value",
+      xlab = "Fitted Value", ylab = "Residual")
>
> abline(h = 0, col = "red")
>
> plot(oclog.LM$residuals[-n], oclog.LM$residuals[-1],
+      xlab = "Previous", ylab = "Residual",
+      main = "Log Residual Vs Previous Residual")
>
> qqnorm(oclog.LM$residuals)
> qqline(oclog.LM$residuals)
```