Lecture 10: Gelman Hill Ch 8

Fake-data simulation

Fake data can be used to validate statistical algorithms (important when you are writing your own..) and to compare properties of estimation procedures.

Confidence Interval??: A 95% confidence interval contains the true value with probability 95.

How can we verify that the coverage properties of these intervals are reasonable?

```
set.seed(03032020)
n <- 100
x <- runif(n)
beta <- 1
sigma <- .1
lm_data <- tibble(y = rnorm(n, x*beta, sigma), x=x)

lm_data <- lm(y ~ x, data = lm_data)
display(lm_data)
```

```
## lm(formula = y ~ x, data = lm_data)
##             coef.est coef.se
## (Intercept) 0.02     0.02
## x           0.98     0.04
## ---
## n = 100, k = 2
## residual sd = 0.10, R-Squared = 0.89
```

```
confint(lm_data)
```

```
##                   2.5 %     97.5 %
## (Intercept) -0.01799708 0.05796958
## x            0.91402043 1.05581767
```

Describe a procedure to do this at a larger scale... perhaps with pseudocode.

```
num_replicates <- 10000
n <- 100
beta <- 1
sigma <- .1
in_interval <- rep(FALSE, num_replicates)
conf_width <- rep(0, num_replicates)
for (i in 1:num_replicates){
  x <- runif(n)
  lm_data <- tibble(y = rnorm(n, x*beta, sigma), x=x)
  lm_data <- lm(y ~ x, data = lm_data)
  in_interval[i] <- (beta > confint(lm_data)[2,1] ) & (beta < confint(lm_data)[2,2])
  conf_width[i] <- as.numeric(diff(confint(lm_data)[2,]))
}
```

Based on 10,000 samples, the coverage of the interval in this procedure is 0.946. Furthermore, the average confidence interval width is 0.138

Now what happens if the model is not actually what we assumed... One option is to consider a t distribution.

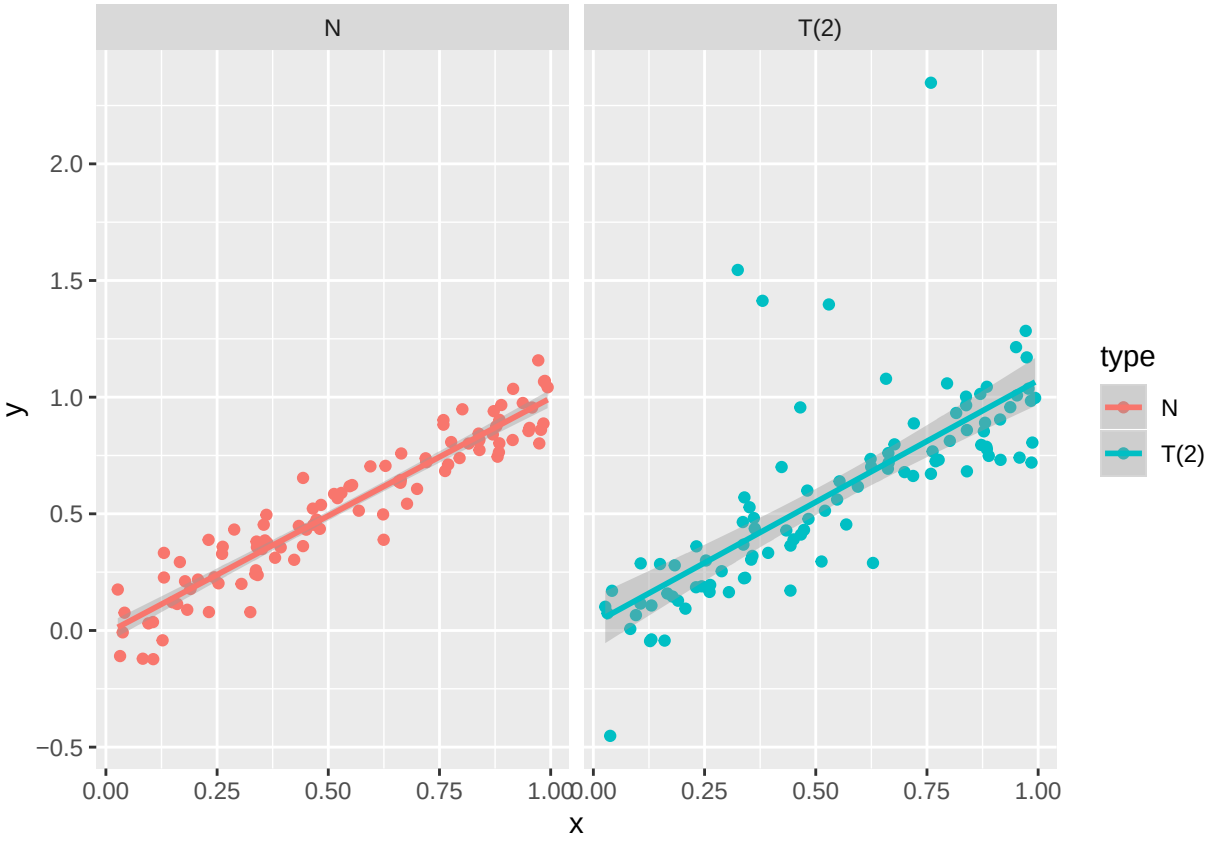
```
num_replicates <- 10000
n <- 100
beta <- 1
sigma <- .1
in_interval <- rep(FALSE, num_replicates)
conf_width <- rep(0, num_replicates)
deg_free <- 2

for (i in 1:num_replicates){
  x <- runif(n)
  lm_data <- tibble(y = rt(n,deg_free) * sigma + x*beta, x=x)
  lm_data <- lm(y ~ x, data = lm_data)
  in_interval[i] <- (beta > confint(lm_data)[2,1] ) & (beta < confint(lm_data)[2,2])
  conf_width[i] <- as.numeric(diff(confint(lm_data)[2,]))
}
```

Based on 10,000 samples, the coverage of the interval in this procedure is 0.95. Furthermore, the average confidence interval width is 0.382. So for this setting with symmetric residuals, the coverage is reasonably good, but notice the confidence intervals are considerably wider due to the larger estimated variance. We will come back to this and look at residual plots.

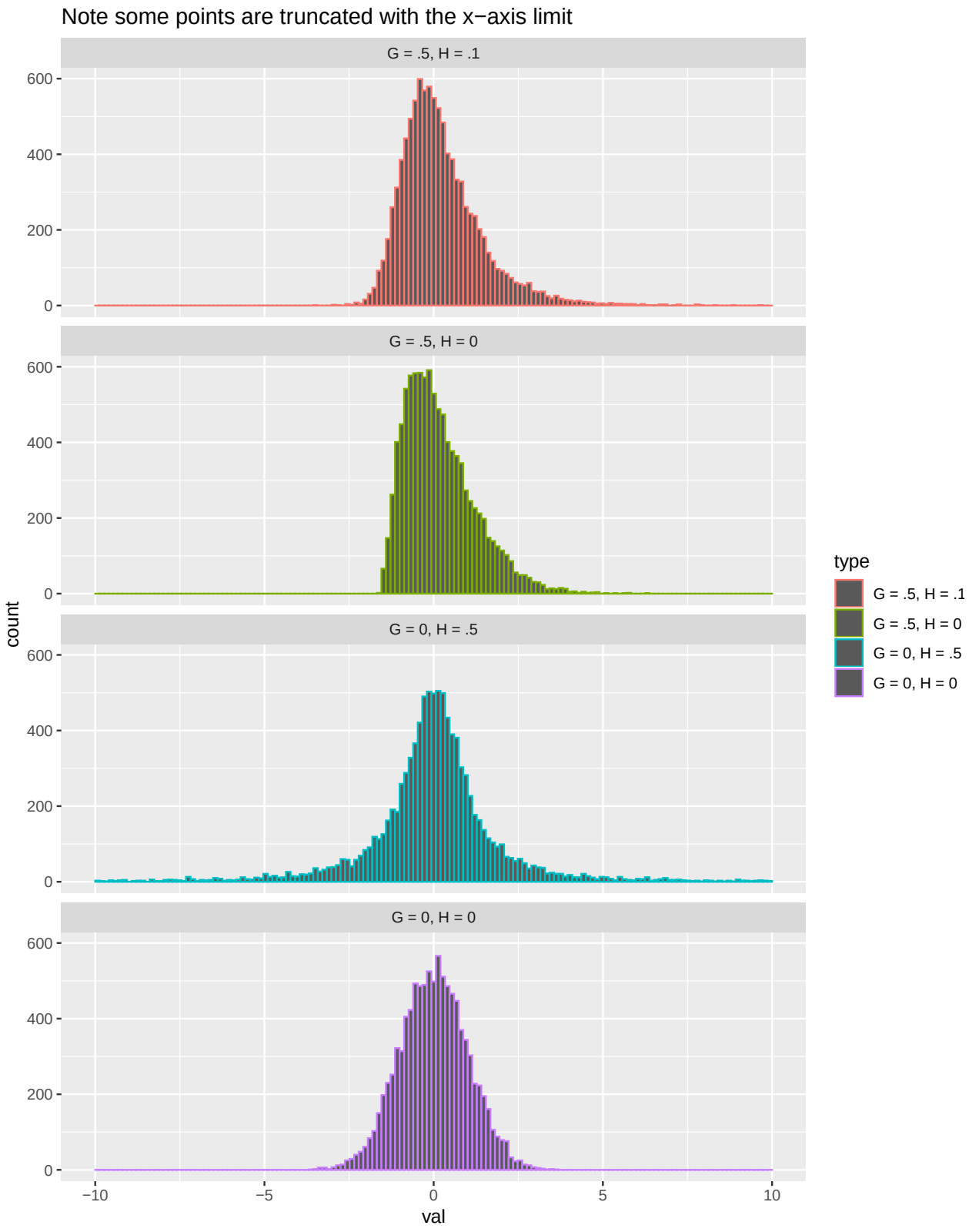
```
x <- runif(n)
lm_data_comb <- tibble(y = c(rnorm(n, x*beta, sigma),rt(n,deg_free) * sigma + x*beta),
                       x=c(x,x), type = rep(c('N','T(2)'), each = n))

lm_data_comb %>% ggplot(aes(y=y, x=x, color = type)) +
  geom_point() + geom_smooth(method='lm') + facet_wrap(.~type)
```



What about a non-symmetic distribution? Consider the Tukey g-and-h distribution https://rss.onlinelibrary.wiley.com/doi/10.1111/j.1740-9713.2019.01273.x. The Gaussian distribution is a special case of the g-and-h distribution (when g=h=0); otherwise, the distribution can capture skewed behavior.

```
num_sims = 10000
sims <- tibble(val = c(rgh(num_sims,A = 0, B = 1, g = 0, h = 0),
                rgh(num_sims,A = 0, B = 1, g = 0, h = .5),
                rgh(num_sims,A = 0, B = 1, g = .5, h = 0),
                rgh(num_sims,A = 0, B = 1, g = .5, h = .1)),
       type = rep(c('G = 0, H = 0', 'G = 0, H = .5','G = .5, H = 0', 'G = .5, H = .1' ), each = num_sims
```


Note some points are truncated with the x−axis limit
G = .5, H = .1
G = .5, H = 0
G = 0, H = .5
G = 0, H = 0
600
400
200
0
count
−10
−5
0
5
10
val
type
G = .5, H = .1
G = .5, H = 0
G = 0, H = .5
G = 0, H = 0

Model Checking

Recall: GH list the assumption in descreasing order of importance.

1. **Validity:** "Most importantly, the data you are analyzing should map to the research question you are trying to answer... Optimally, this means that the outcome measure should accurately reflect the phenomenon of interest, the model should include all relevant predictors, and the model should generalize to the cases to which it will be applied to. Data used in empirical research rarely meet all (if any) of these criteria precisely. However, keeping these goals in mind can help you be precise about the types of questions you can *and cannot* answer.

2. **Additivity and linearity:** "the most important mathematical assumption of the regression model is that its deterministic component is a linear function of the separate predictors: $y = \beta_1 x_1 + \beta_2 x_2 + ...$ If additivity is violated, it might make sense to transform the data or to add interactions. If linearity is violated, perhaps a predictor should be transformed or included in as a basis function."

3. **Independence of errors:** The simple regression model assumes the errors from the prediction line are independent... more later with multilevel models.

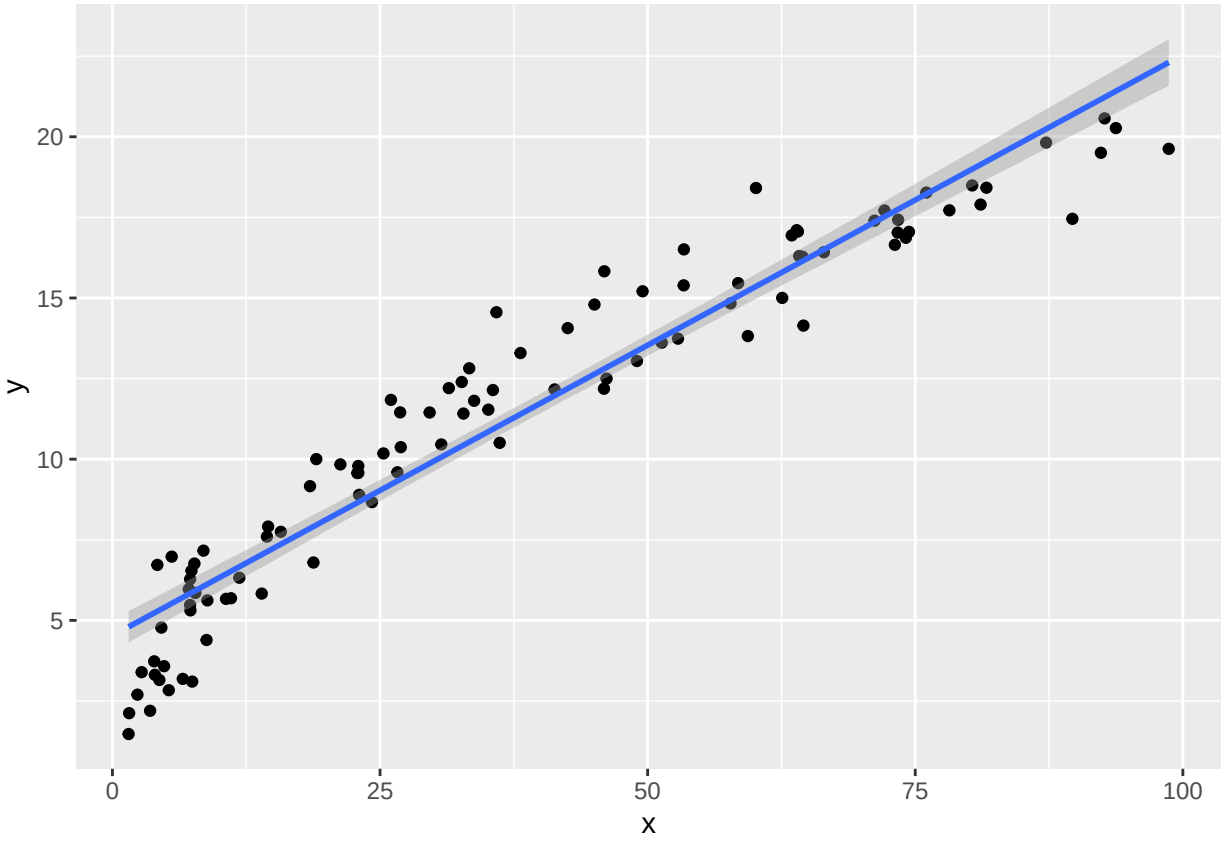
4. **Equal variance of errors:** If the variance of the regression errors are unequal, estimation is more efficiently performed using weighted least squares. However, unequal variance does not affect the predictor $X\underline{\beta}$.

5. **Normality of errors:** "The regression assumption that is generally *least* important is that the errors are normally distributed. GH do *not* recommend diagnostics of the normality of the regression residuals.

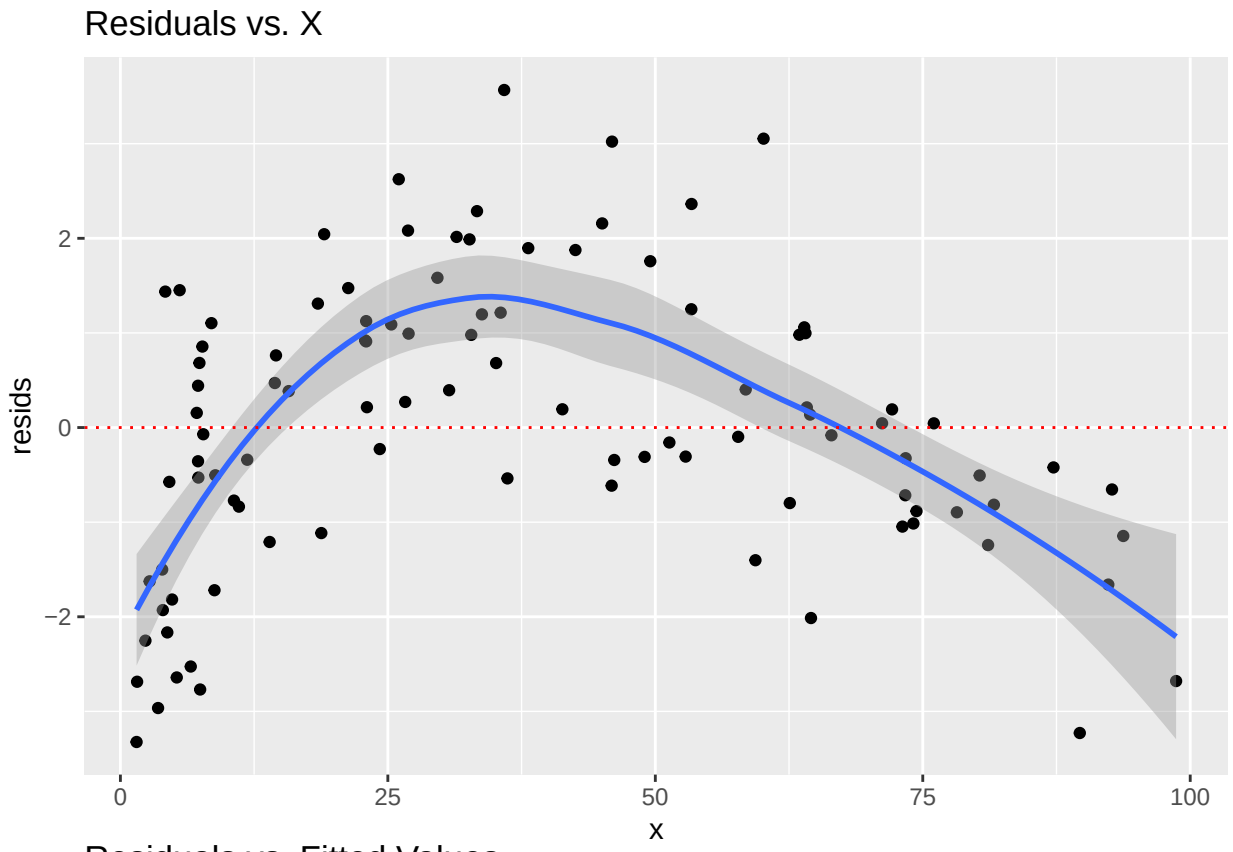
We will continue using simulation to evaluate these assumptions.

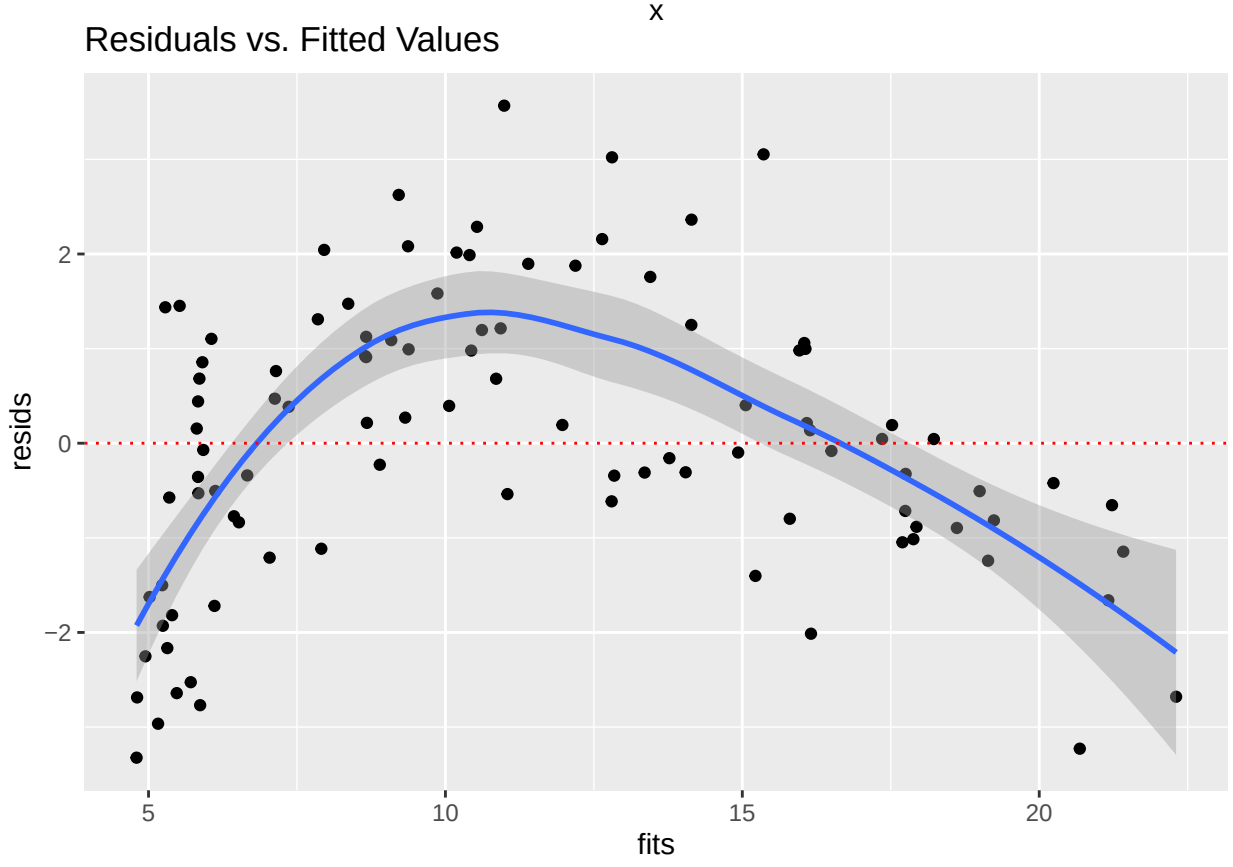
Additivity and Linearity

Consider the following relationship between x and y...




```
## lm(formula = y ~ x, data = lm_data)
##             coef.est coef.se
## (Intercept) 4.53     0.25
## x           0.18     0.01
## ---
## n = 100, k = 2
## residual sd = 1.51, R-Squared = 0.92
```

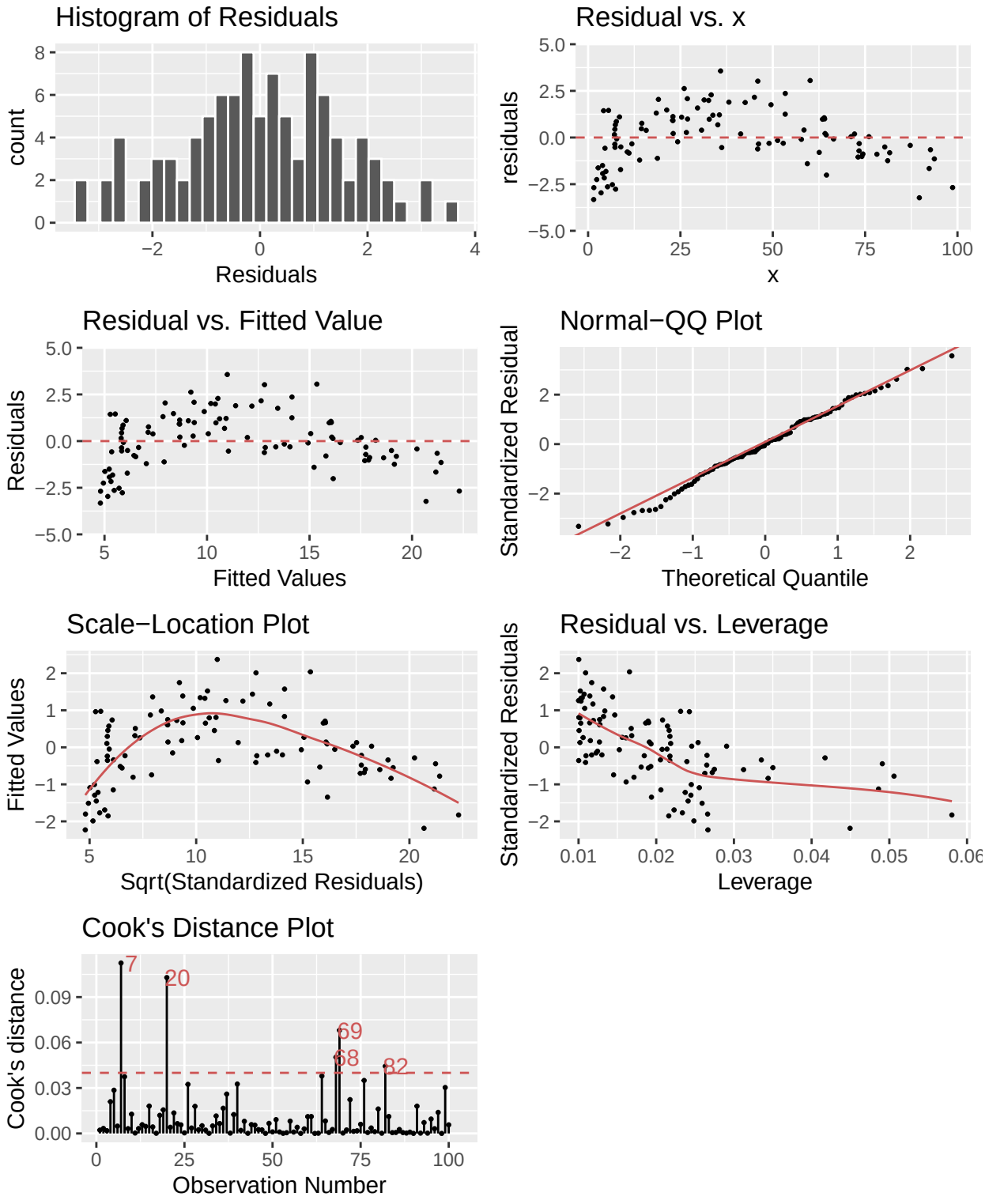

Residuals vs. X
resids
2
0
−2
0
25
50
75
100
x


Residuals vs. Fitted Values
resids
2
0
−2
5
10
15
20
fits

Some of these figures are built into regression diagnostics, such as `gg_diagnose` in `lindia`.

```
gg_diagnose(lm_fit)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

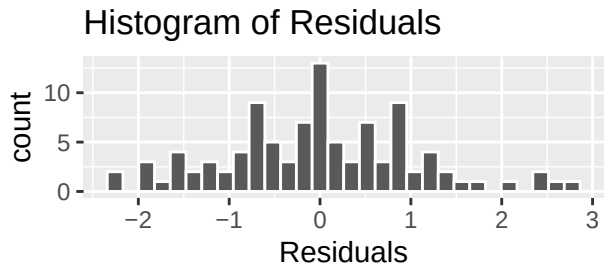


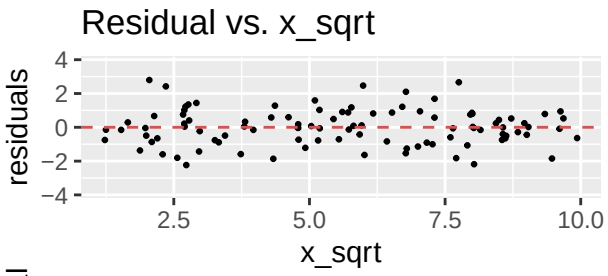
Now consider fitting the appropriate model..

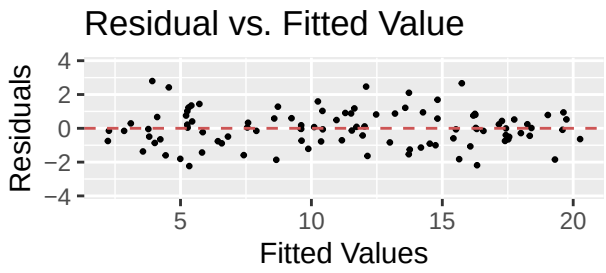
```
lm_fit2 <- lm(y ~ x_sqrt, data = lm_data)
display(lm_fit2)
```

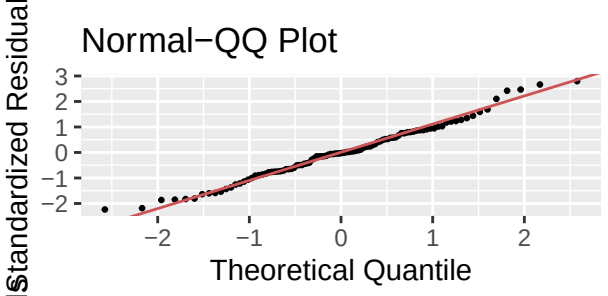
```
## lm(formula = y ~ x_sqrt, data = lm_data)
##             coef.est coef.se
## (Intercept) -0.32     0.27
## x_sqrt       2.07     0.04
## ---
## n = 100, k = 2
## residual sd = 1.07, R-Squared = 0.96
```

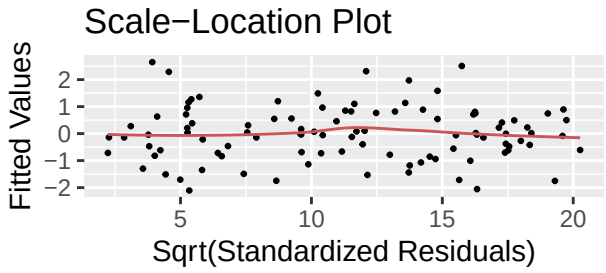
```
gg_diagnose(lm_fit2)
```

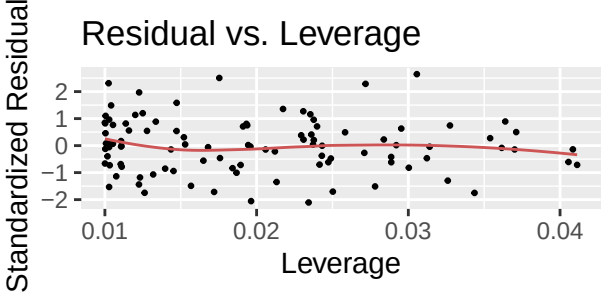


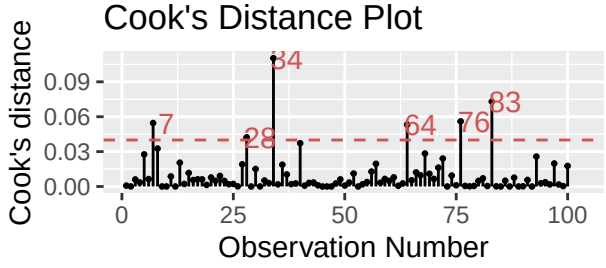






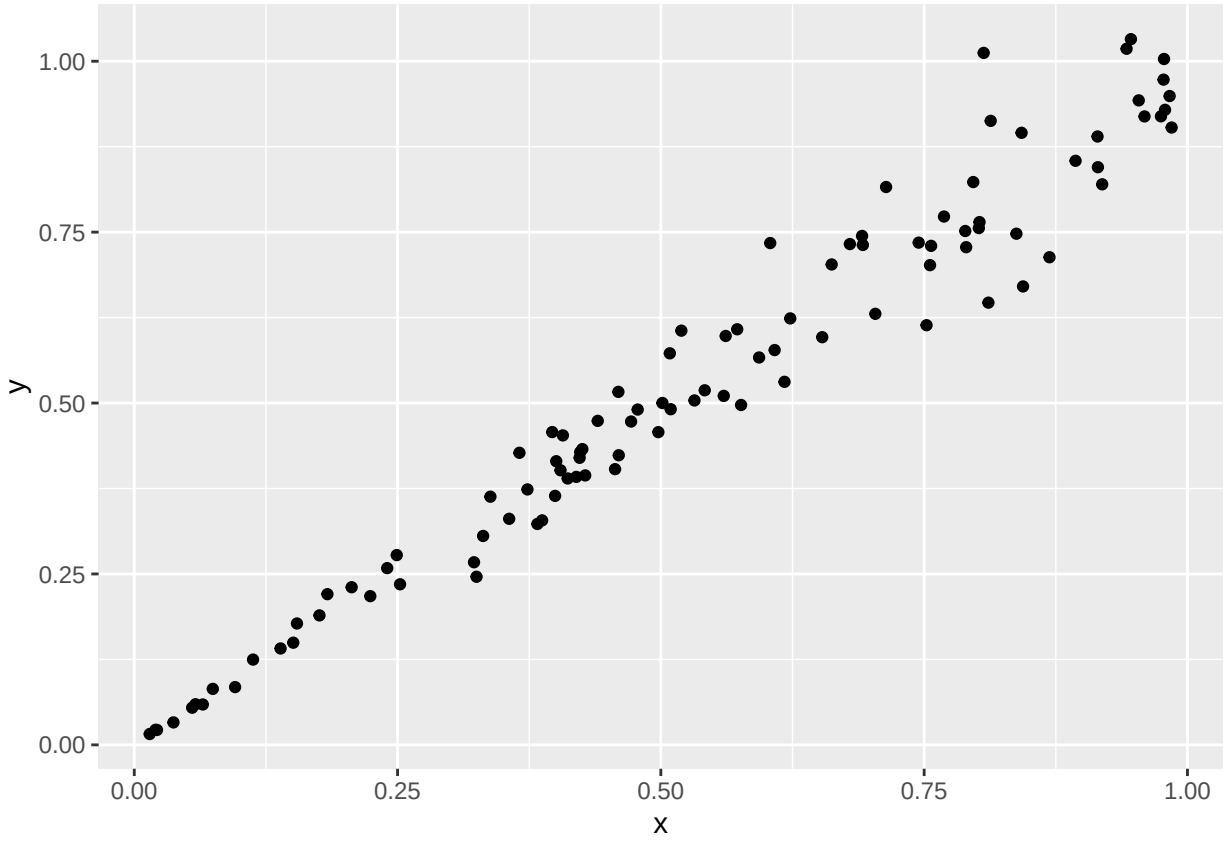


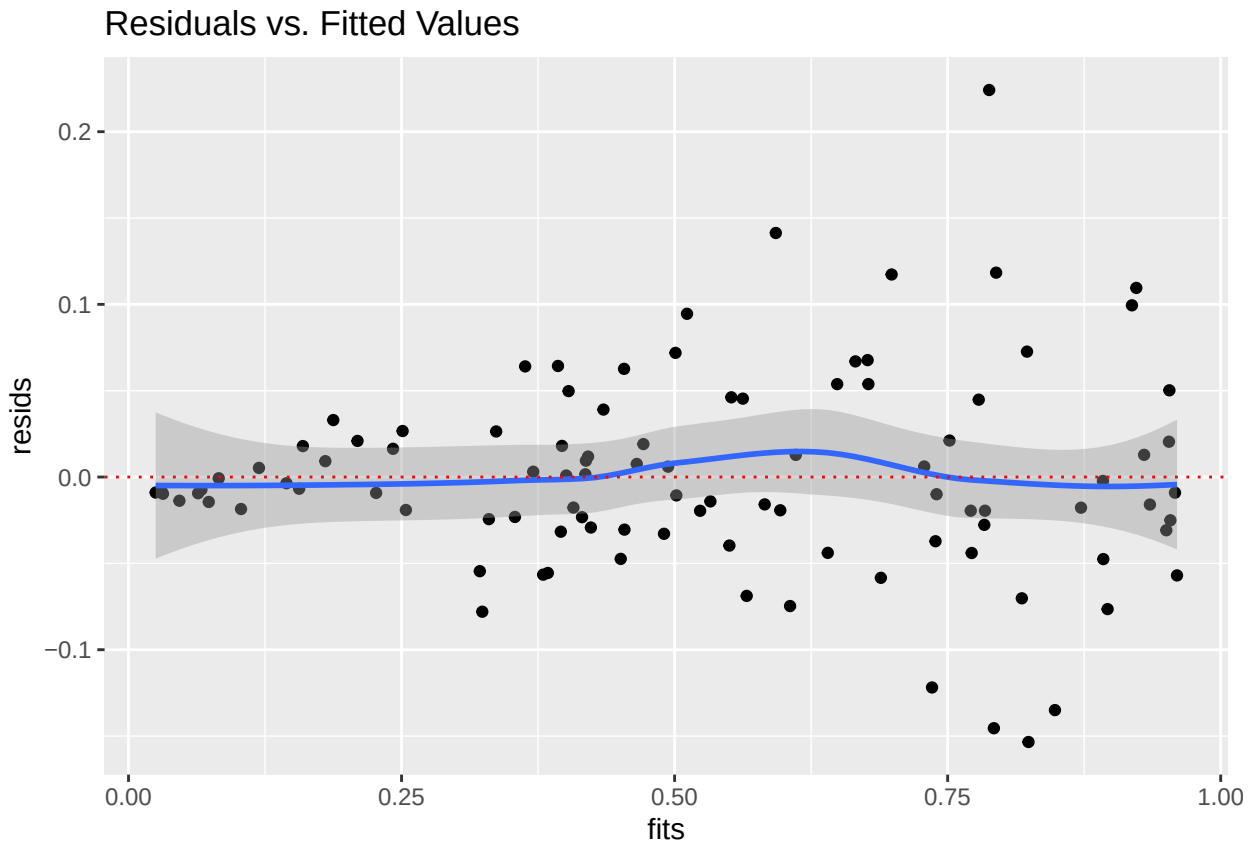




Equal Variance

```
n <- 100
x <- runif(n)
beta <- 1
sigma <- .1
w <- 1/x
lm_var <- tibble(y = rnorm(n, x*beta, sigma / w), x=x)
lm_var %>% ggplot(aes(y = y, x=x)) + geom_point()
```





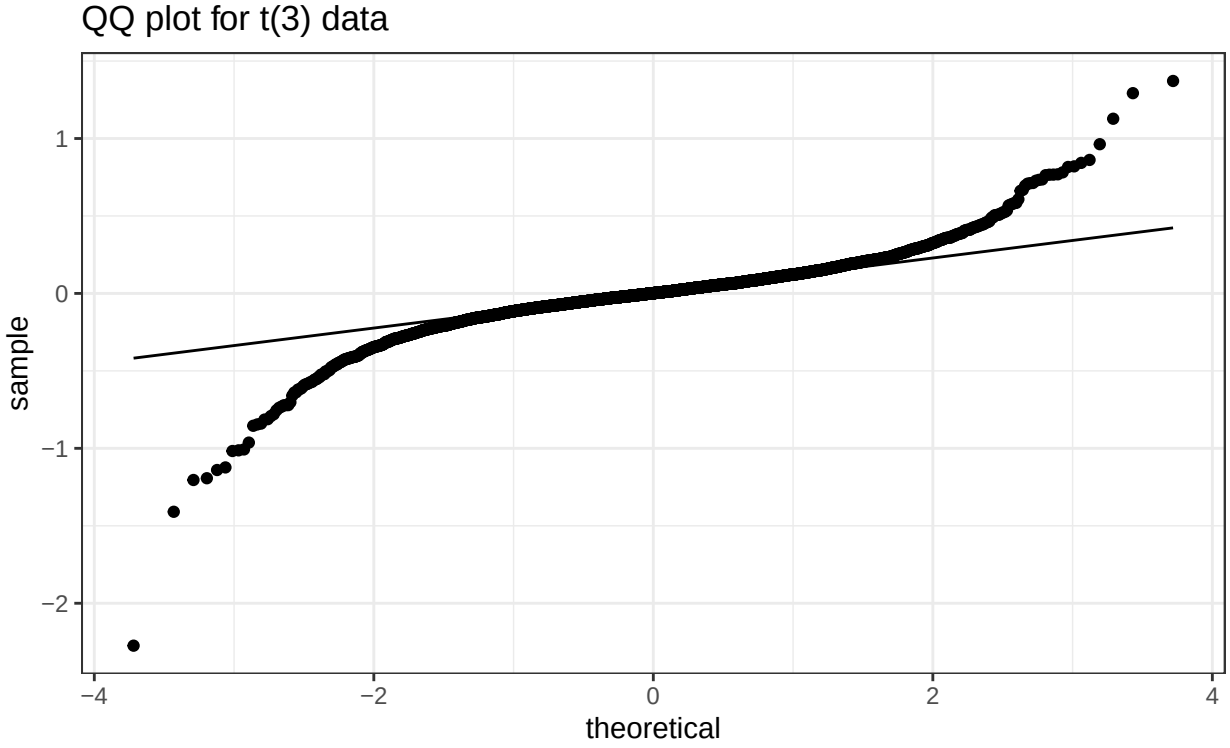

```
lm_wls <- lm(y ~ x, data = lm_var, weights = w)
display(lm_wls)
```

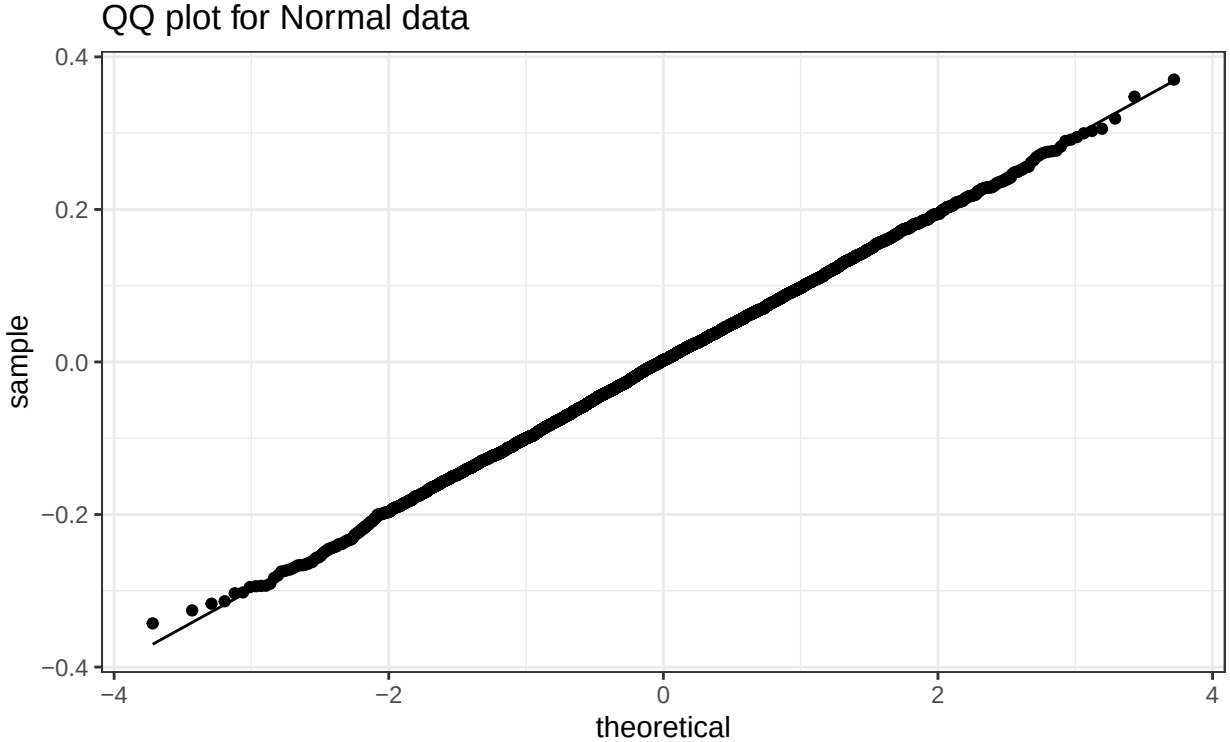
```
## lm(formula = y ~ x, data = lm_var, weights = w)
##             coef.est coef.se
## (Intercept) 0.00     0.00
## x           0.98     0.01
## ---
## n = 100, k = 2
## residual sd = 0.07, R-Squared = 0.98
```

```
display(lm_varmod)
```

```
## lm(formula = y ~ x, data = lm_var)
##             coef.est coef.se
## (Intercept) 0.01     0.01
## x           0.96     0.02
## ---
## n = 100, k = 2
## residual sd = 0.06, R-Squared = 0.96
```

Normality of Errors





We previously saw that the impact on the coverage of the confidence interval was fairly minor. I think this is why GH put normality as the lowest priority for checking assumptions, HOWEVER...

GH also present model simulation, we will return to this after studying GLMs.