22-1-2

Nikhil Gopal

4/8/2022

21.1

Measurement error in y: Simulate data (x, y)i, i = 1,. . . , n from a linear regression model, y = a + bx + error, but suppose that the outcome y is not observed directly, but instead we observe v = y + error, with independent measurement errors with mean zero. Use simulations to understand the statistical properties of the observed-data regression of v on x, compared to the desired regression of y on x.

```
df <- data.frame(
  "x" <- rnorm(10000, mean = 0, sd = 1),
  "y" <- rnorm(10000, mean = 400, sd = 20)
  )
colnames(df) <- c("x", "y")

df$v <- df$y + rnorm(1, mean = 0, sd = 40)
```

Above I generated the data with 10000 observations. I made x a normal random variable with mean 0 and sd 1, and y a normal random variable with mean 400 and sd 20. Finally, I made v equal to y plus a random variable with mean 0 and sd 40. Below we will fit and compare the regressions:

```
set.seed(123)

fit_y <- stan_glm(y~x, data = df, refresh = 0)
fit_error <- stan_glm(v~x, data = df, refresh = 0)

print(fit_y)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      y ~ x
##  observations: 10000
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 399.8    0.2
## x             0.0    0.2
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 20.2    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

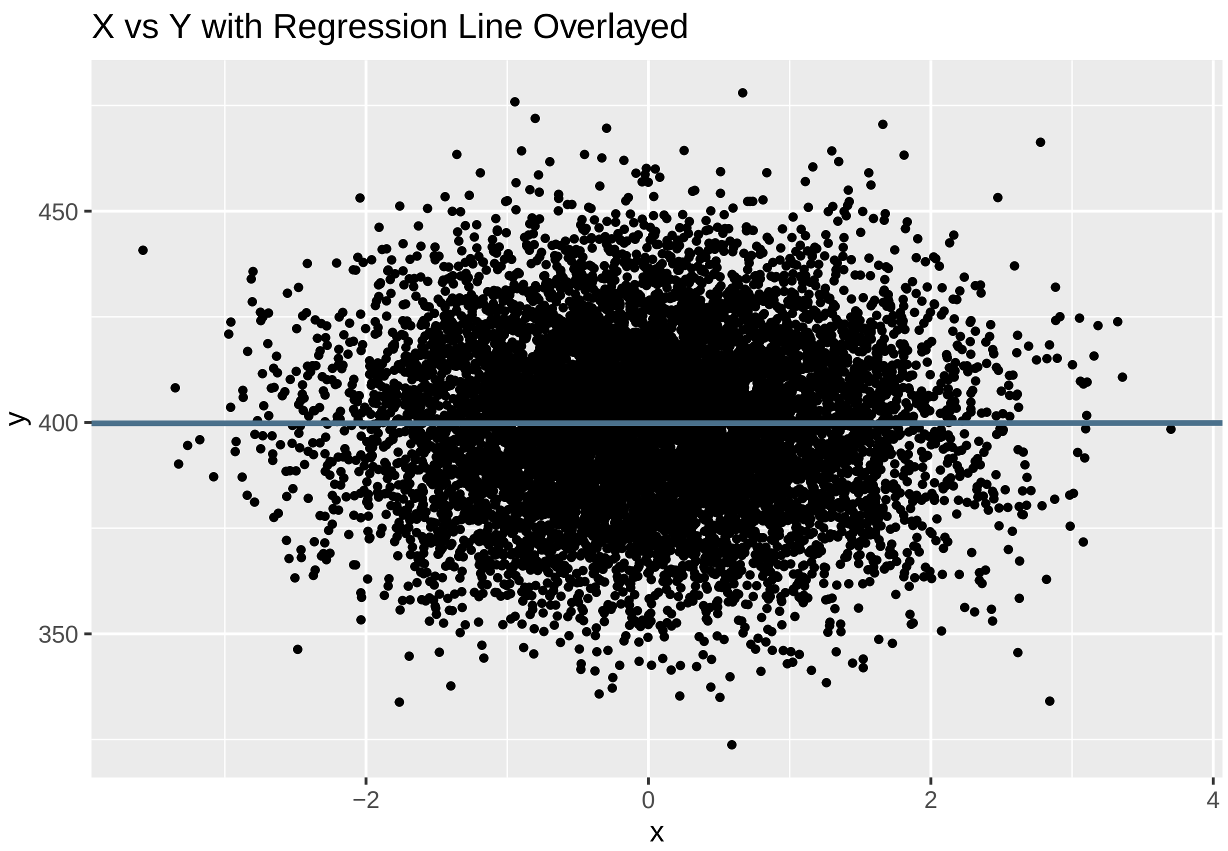
```
print(fit_error)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      v ~ x
##  observations: 10000
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 360.6    0.2
## x             0.0    0.2
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 20.2    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

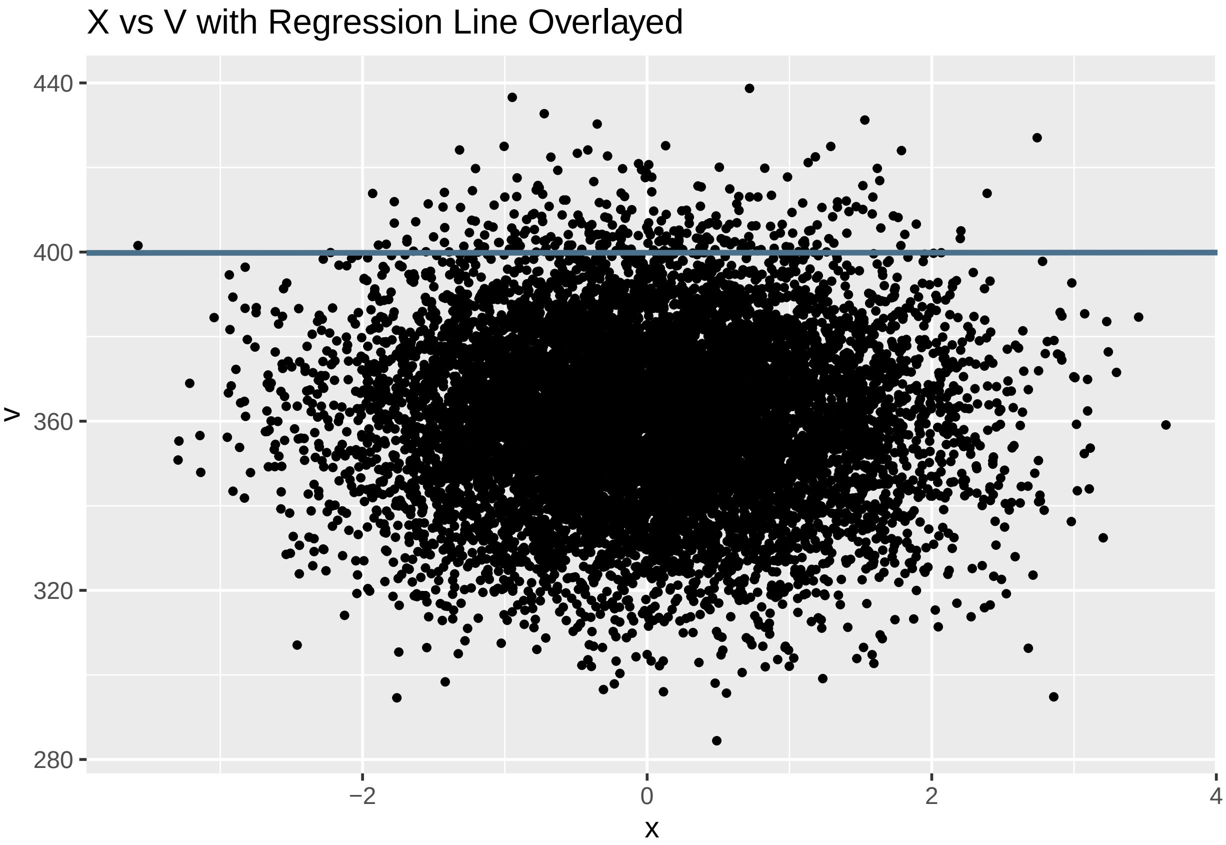
For the first regression without error, the intercept term was 400.3 and the coefficient of x was 0.1. The SD for the coefficients was 0.2. For the regression with error, the standard deviations and x coefficient was the same, but the intercept term changed to 427.7. It makes sense that it would be a little higher as V has mean 400 + error, whereas Y has mean 400. In fact, the mean of all the observations in the V column was 427.

Below we will compare regression fit:

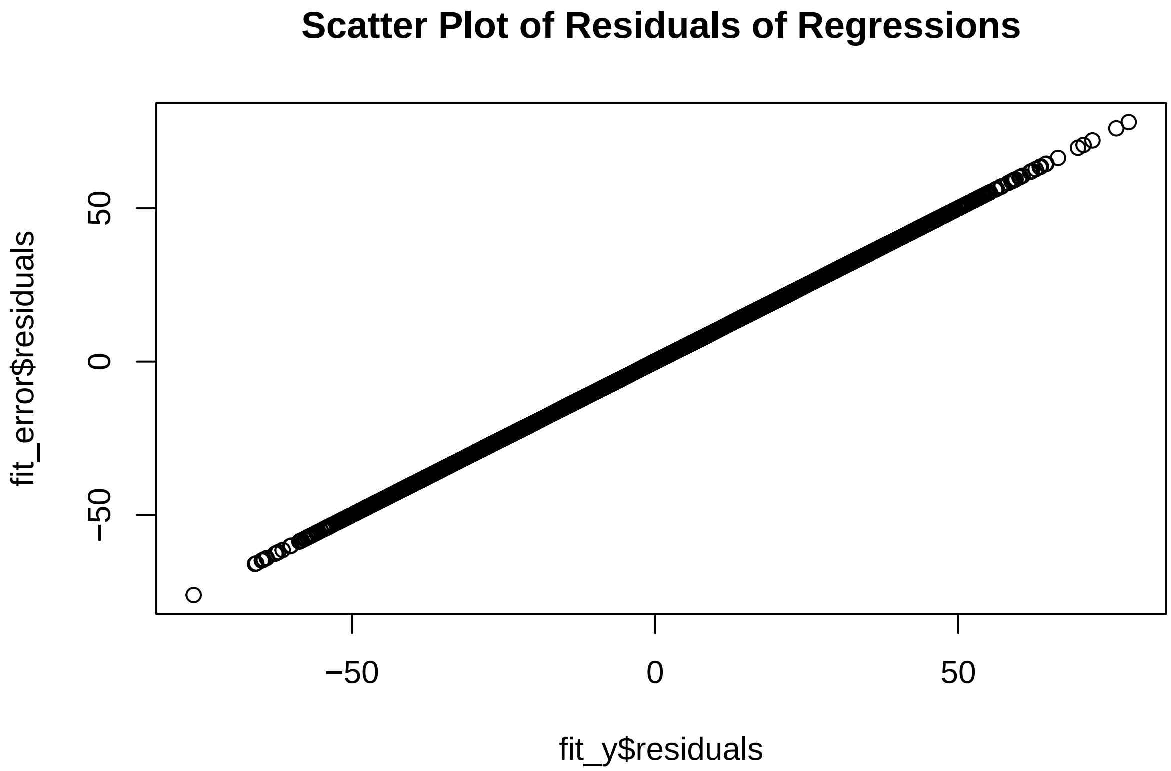
```
ggplot(df, aes(x = x, y = y)) +
  geom_point(size = 1, position = position_jitter(height = 0.05, width = 0.1)) +
  geom_abline(intercept = coef(fit_y)[1], slope = coef(fit_y)[2], color = "skyblue4", size = 1) +
  ggtitle ("X vs Y with Regression Line Overlayed")
```



```
ggplot(df, aes(x = x, y = v)) +
  geom_point(size = 1, position = position_jitter(height = 0.05, width = 0.1)) +
  geom_abline(intercept = coef(fit_y)[1], slope = coef(fit_y)[2], color = "skyblue4", size = 1) +
  ggtitle ("X vs V with Regression Line Overlayed")
```



```
plot(fit_y$residuals, fit_error$residuals, main = "Scatter Plot of Residuals of Regressions")
```



We don't observe a significant difference in model fit. Both models seem to have the same fit when the regression line is overlayed.

```
sd(fit_y$residuals)
```

```
## [1] 20.20524
```

```
sd(fit_error$residuals)
```

```
## [1] 20.20524
```

Both regressions have the same standard deviation of their residuals.

22.2

Measurement error in x: Simulate data (x, y)i, i = 1,. . . , n from a linear regression model, y = a + bx + error, but suppose that the predictor x is not observed directly, but instead we observe u = x + error, with independent measurement errors with mean zero. Use simulations to understand the statistical properties of the observed-data regression of y on u, compared to the desired regression of y on x.

```
df <- data.frame(
  "x" <- rnorm(10000, mean = 0, sd = 1),
  "y" <- rnorm(10000, mean = 400, sd = 20)
  )
colnames(df) <- c("x", "y")

df$u <- df$x + rnorm(1, mean = 0, sd = 40)
```

I generated data with the same qualities as above, except I added error to the x term instead of the y term this time and called that variable u.

```
set.seed(123)

fit_x <- stan_glm(y~x, data = df, refresh = 0)
fit_x_error <- stan_glm(y~u, data = df, refresh = 0)

print(fit_x)
```

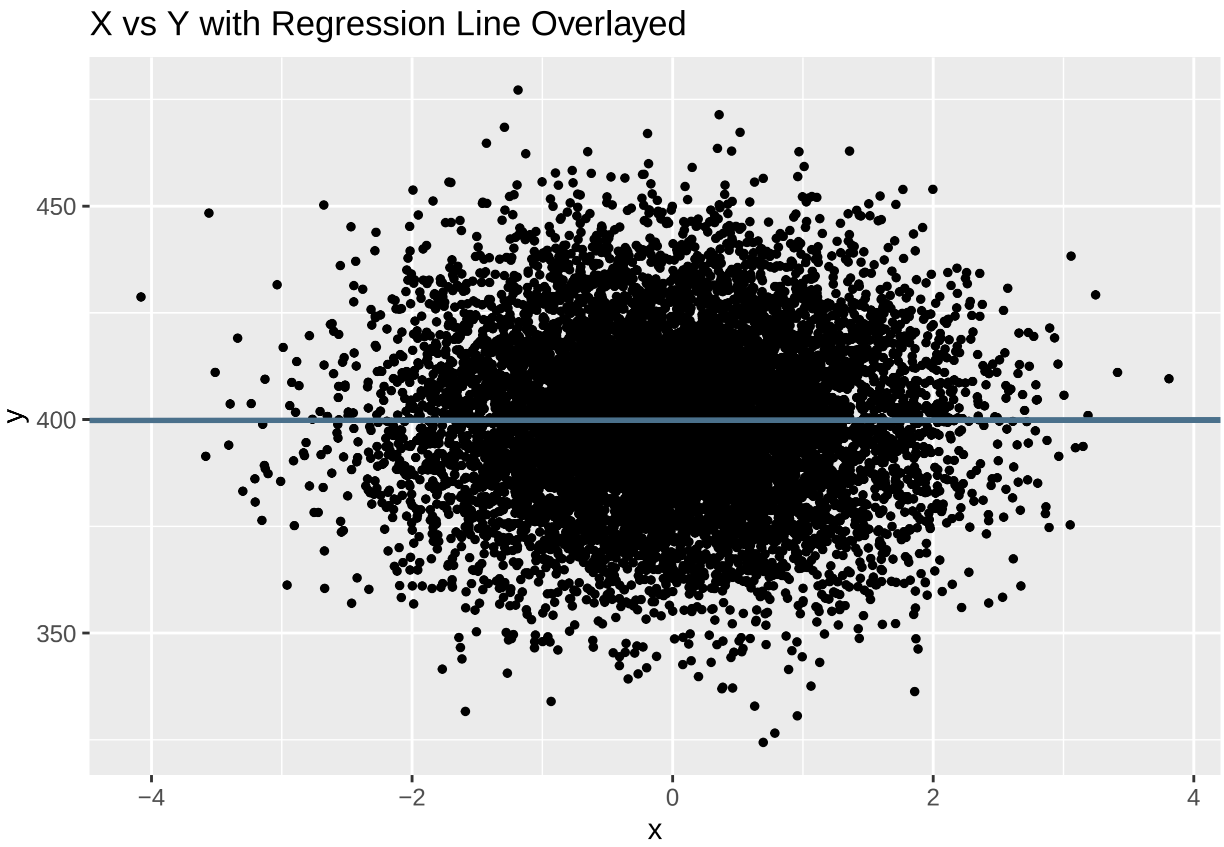
```
## stan_glm
##  family:       gaussian [identity]
##  formula:      y ~ x
##  observations: 10000
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 400.0    0.2
## x             0.0    0.2
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 19.9    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
print(fit_x_error)
```

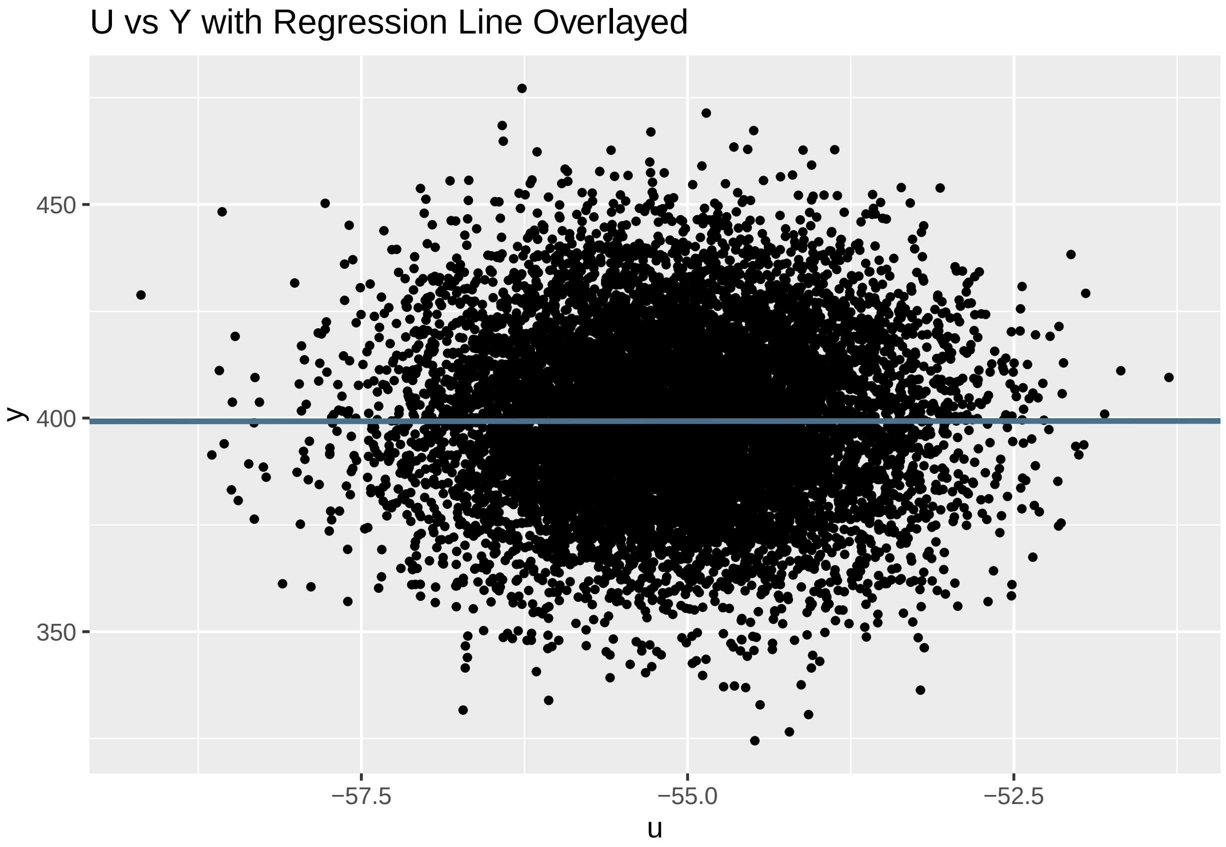
```
## stan_glm
##  family:       gaussian [identity]
##  formula:      y ~ u
##  observations: 10000
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 398.9   11.4
## u             0.0    0.2
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 19.9    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Both of these regressions had essentially the same coefficients and intercept values.

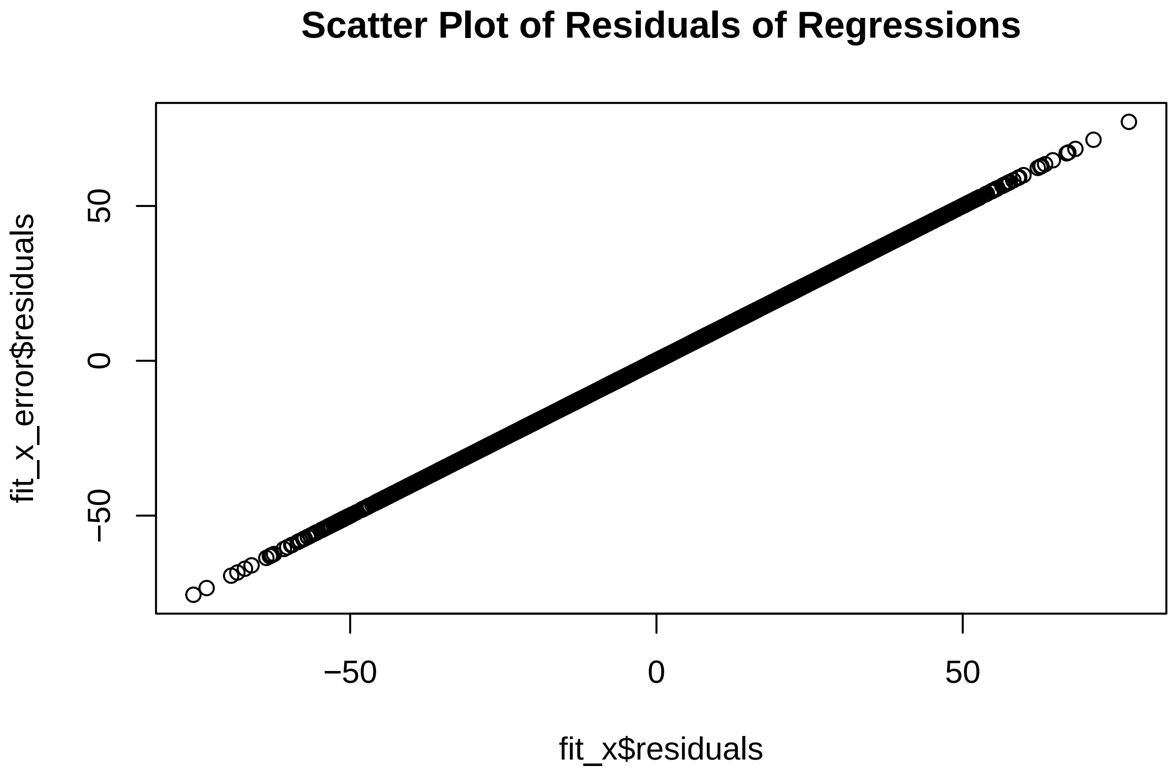
```
ggplot(df, aes(x = x, y = y)) +
  geom_point(size = 1, position = position_jitter(height = 0.05, width = 0.1)) +
  geom_abline(intercept = coef(fit_y)[1], slope = coef(fit_y)[2], color = "skyblue4", size = 1) +
  ggtitle ("X vs Y with Regression Line Overlayed")
```



```
ggplot(df, aes(x = u, y = y)) +
  geom_point(size = 1, position = position_jitter(height = 0.05, width = 0.1)) +
  geom_abline(intercept = coef(fit_y)[1], slope = coef(fit_y)[2], color = "skyblue4", size = 1) +
  ggtitle ("U vs Y with Regression Line Overlayed")
```



```r
plot(fit_x$residuals, fit_x_error$residuals, main = "Scatter Plot of Residuals of Regressions")
```



```
sd(fit_x$residuals)
```

```
## [1] 19.90111
```

```
sd(fit_x_error$residuals)
```

```
## [1] 19.90111
```

There did not appear to be a big difference in model fit between both cases. I suspect this to be because the mean error was 0, and there was a large sample size (10000). I think the effects of error on y or x would definitely be more pronounced in a situation where the mean of the error distribution was not equal to zero, or in studies with small sample sizes.