Hypothesis Testing and Confidence Intervals with R

Econ 440 - Introduction to Econometrics

Patrick Toche, ptoche@fullerton.edu

26 April 2022

Load dataset

```
library(readxl)
df <- read_xlsx("CASchools_EE141_InSample.xlsx", trim_ws=TRUE)
head(df)
```

```
## # A tibble: 6 x 25
##   countyname districtname schoolname zipcode frpm_frac_s enrollment_s ell_frac_s
##   <chr>      <chr>        <chr>        <dbl>       <dbl>        <dbl>      <dbl>
## 1 San Diego  La Mesa-Spr~ La Presa ~   91941       0.774          411      0.281
## 2 Orange     Centralia E~ Temple (R~   90620       0.587          509      0.230
## 3 Butte      Palermo Uni~ Golden Hi~   95966       0.820          284      0.133
## 4 Butte      Oroville Ci~ Oakdale H~   95966       0.948          442      0.177
## 5 Tulare     Rockford El~ Rockford ~   93257       0.557          406      0.123
## 6 Santa Cla~ Campbell Un~ Castlemon~   95008       0.513          727      0.374
## # ... with 18 more variables: edi_s <dbl>, te_fte_s <dbl>, te_avgyr_s <dbl>,
## #   te_salary_low_d <dbl>, te_salary_avg_d <dbl>, te_days_d <dbl>,
## #   te_serdays_d <dbl>, age_frac_5_17_z <dbl>, pop_1_older_z <dbl>,
## #   ed_frac_hs_z <dbl>, ed_frac_sc_z <dbl>, ed_frac_ba_z <dbl>,
## #   ed_frac_grd_z <dbl>, med_income_z <dbl>, testscore <dbl>, str_s <dbl>,
## #   ada_enrollment_ratio_d <dbl>, charter_s <dbl>
```

Regression of Test Score on Student/Teacher ratio

```
ols <- lm(testscore ~ str_s, data = df)
```

Add confidence intervals:

```
library(broom)
ols.augment <- augment(ols)
ols.augment
```

```
## # A tibble: 500 x 8
##    testscore str_s .fitted .resid    .hat .sigma    .cooksd .std.resid
##        <dbl> <dbl>   <dbl>  <dbl>   <dbl>  <dbl>      <dbl>      <dbl>
##  1      728.  27.0    751. -23.4  0.00344   60.3 0.000261      -0.389
##  2      756   25.5    752.   3.50 0.00232   60.3 0.00000394     0.0582
##  3      708.  23.1    755. -47.2  0.00214   60.3 0.000660      -0.785
##  4      686   25.1    753. -66.8  0.00219   60.3 0.00135       -1.11
##  5      734.  24.5    753. -18.9  0.00203   60.3 0.000101      -0.315
##  6      808.  21.7    756.  51.7  0.00285   60.3 0.00106        0.860
##  7      734.  24.6    753. -19.0  0.00204   60.3 0.000103      -0.316
```

```
## 8        685.  31.4    747. -62.1  0.0106    60.3 0.00578       -1.04
## 9        676.  27.6    750. -74.2  0.00401   60.2 0.00306       -1.23
## 10       862.  24.6    753. 109.   0.00206   60.1 0.00337        1.81
## # ... with 490 more rows
```

Model Fitness

- How well does a line fit data? How tightly clustered around the line are the data points?
- How much variation in Y_i is explained by the model?

$$\underbrace{Y_i}_{Obs} = \underbrace{\widehat{Y}_i}_{Pred} + \underbrace{\hat{u}}_{Error}$$

- OLS estimators minimize the Sum of Squared Errors (SSE):

$$\sum_{i=1}^{n} \hat{u_i}^2 \rightarrow \min$$

Goodness of Fit (coefficient of determination): R^2

- The fraction of variation in Y explained by variation in predicted values $\hat{Y}$

$$R^2 = \frac{var(\widehat{Y}_i)}{var(Y_i)} = \frac{ESS}{TSS} = 1 - \frac{SSE}{TSS}$$

- Explained Sum of Squares (ESS): sum of squared deviations of *predicted* values from their mean

$$ESS = \sum_{i=1}^{n} (\hat{Y}_i - \bar{Y})^2$$

- Total Sum of Squares (TSS): sum of squared deviations of *observed* values from their mean

$$TSS = \sum_{i=1}^{n} (Y_i - \bar{Y})^2$$

- The OLS estimator satisfies:

$$\bar{\hat{Y}}_i = \bar{Y}$$

Goodness of Fit

- As the square of the correlation coefficient between X and Y:

$$R^2 = (r_{X,Y})^2$$

R^2 as the correlation coefficient squared:

```
# Base R
cor(df$testscore, df$str_s)^2
```

```
## [1] 0.0031453
```

```r
# dplyr
df %>%
  summarize(r_sq = cor(testscore,str_s)^2)
```

```
## # A tibble: 1 x 1
##      r_sq
##     <dbl>
## 1 0.00315
```

R^2 in R

- Explore the `broom` package.
- The `broom augment()` command produces:
  - `.fitted`: predicted values $\hat{Y_i}$
  - `.resid`: estimated residuals $\hat{u}_i$

```r
library(broom)
ols %>%
  augment() %>%
  head(., n=5)
```

```
## # A tibble: 5 x 8
##   testscore str_s .fitted .resid    .hat .sigma    .cooksd .std.resid
##       <dbl> <dbl>   <dbl>  <dbl>   <dbl>  <dbl>      <dbl>      <dbl>
## 1      728.  27.0    751. -23.4  0.00344   60.3 0.000261      -0.389
## 2      756   25.5    752.   3.50 0.00232   60.3 0.00000394     0.0582
## 3      708.  23.1    755. -47.2  0.00214   60.3 0.000660      -0.785
## 4      686   25.1    753. -66.8  0.00219   60.3 0.00135       -1.11
## 5      734.  24.5    753. -18.9  0.00203   60.3 0.000101      -0.315
```

R^2 as a ratio of the variances

- R^2 calculated from $\frac{ESS}{TSS}$

```r
ols %>%
  augment() %>%
  summarize(r_sq = var(.fitted)/var(testscore))
```

```
## # A tibble: 1 x 1
##      r_sq
##     <dbl>
## 1 0.00315
```

Standard Error of the Regression

- The standard Error of the Regression $hat\sigma_u$ is an estimator of the standard deviation of u_i:

$$\hat{\sigma_u} = \sqrt{\frac{SSE}{n-2}}$$

- Measures the mean distance between data points and the regression line - A mean prediction error of the regression line - Degrees of Freedom correction: $n-2$

Calculate SER

```r
ols %>%
  augment() %>%
  summarize(SSE = sum(.resid^2),
            df = n()-2,
            SER = sqrt(SSE/df))
```

```
## # A tibble: 1 x 3
##         SSE    df   SER
##       <dbl> <dbl> <dbl>
## 1 1808912.   498  60.3
```

- In large samples, $n-2 \approx n$,

```r
ols %>%
  augment() %>%
  summarize(sd_resid = sd(.resid))
```

```
## # A tibble: 1 x 1
##   sd_resid
##      <dbl>
## 1     60.2
```

Sampling Distributions of the OLS Estimators

Inferential Statistics and Sampling Distributions

- Inferential statistics: Analyze a **sample** to make inferences about an unobservable **population**
- Each sample is **independent** of each other sample (due to replacement)
- Each sample comes from the **identical** underlying population distribution

1. Identification: Exogeneity Vs Endogeneity
   - X is **exogenous** if its variation is *independent* of the other factors that affect Y
   - X is **endogenous** if its variation is *related* to some factor that affects Y
2. Inference: Causality vs Randomness
   - Data is random due to **natural sampling variation**
   - Each sample from the same population yields (slightly) different information

Distributions of the OLS Estimators

- OLS estimators ($\hat{\beta}_0$ and $\hat{\beta}_1$) are computed from a finite (specific) sample of data

- The OLS model contains **2 sources of randomness**:

  - *Modeled* randomness: u includes all factors affecting Y *other* than X

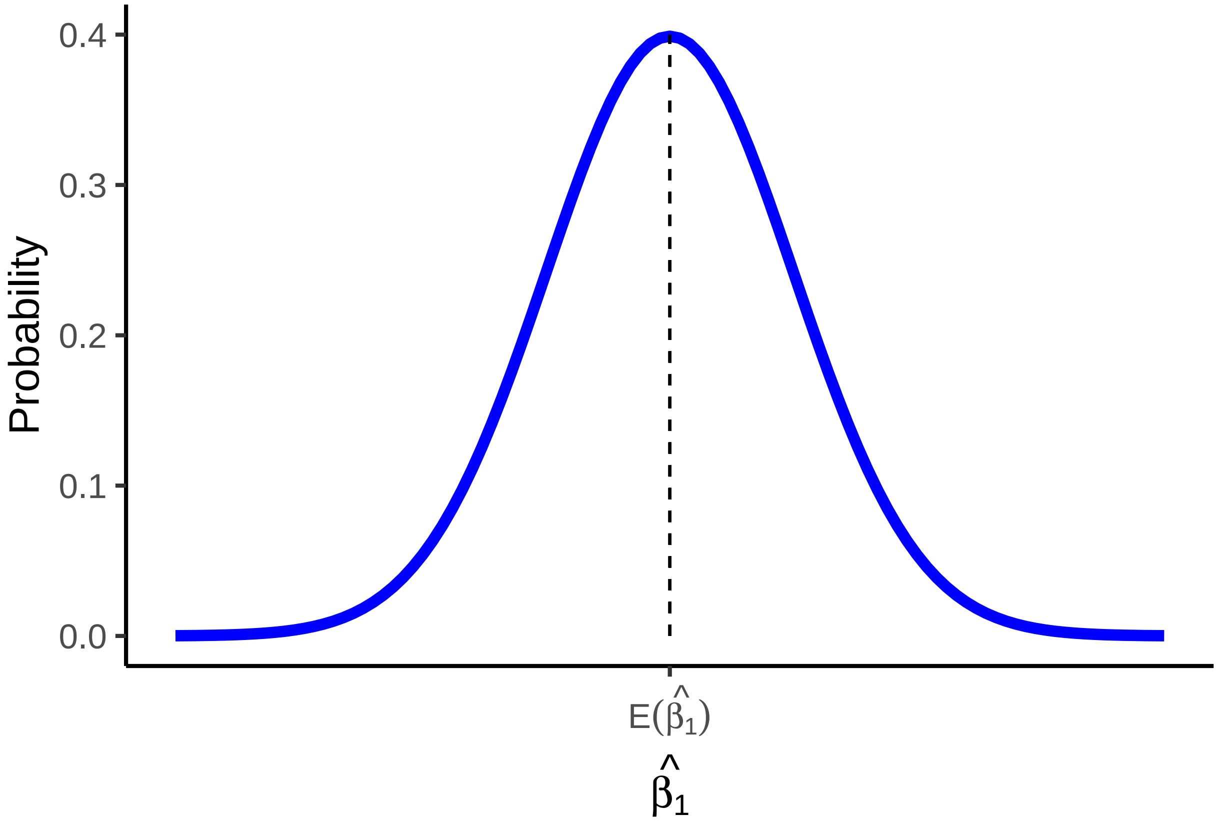
  - different samples will have different values of those other factors (u_i)

  - *Sampling* randomness: different samples will generate different OLS estimators

  - $\hat{\beta}_0, \hat{\beta}_1$ are **random variables**

```r
beta.dist <- ggplot(data=tibble(x=-4:4), aes(x=x)) +
  stat_function(fun=dnorm, size=2, color="blue") +
  geom_segment(aes(x=0, xend=0, y=0, yend=0.4), linetype="dashed") +
  scale_x_continuous(breaks=0,
                     labels=expression(E(hat(beta[1])))) +
  labs(x=expression(hat(beta[1])),
```

```
       y="Probability")
beta.dist
```



The Sampling Distribution of $\hat{\beta}_1$

$$\hat{\beta}_1 \sim N(E[\hat{\beta}_1], \sigma_{\hat{\beta}_1})$$

1. $E[\hat{\beta}_1]$: The reference estimate for the distribution
2. $\sigma_{\hat{\beta}_1}$: The precision of our estimate

Assumptions about Errors

1. The expected value of the residuals is 0
$$E[u] = 0$$
2. The variance of the residuals, conditioning on X is constant:
$$var(u|X) = \sigma_u^2$$
3. Errors are not correlated across observations:
$$cor(u_i, u_j) = 0 \quad \forall i \neq j$$
4. Errors are not correlated with X:
$$cor(X, u) = 0 \text{ or } E[u|X] = 0$$

- Assumptions 1 and 2 imply errors are **i.i.d.**, drawn from the same distribution with mean 0 and variance σ_u^2
- Assumption 2 means that errors are **homoskedastic**.
- Assumption 3: No Serial Correlation
  - Time-series & panel data nearly always contain serial correlation between errors. Also known as autocorrelation.
- Assumption 4: The Zero Conditional Mean Assumption
  - If X contain useful information about u, the model is **endogenous**, **biased** and **not-causal**!

Exogeneity and Unbiasedness

- $\hat{\beta}_1$ is *unbiased* iff there is no systematic difference, on average, between sample values of $\hat{\beta}_1$ and the true population parameter β_1:
$$E[\hat{\beta}_1] = \beta_1$$
- Expect random errors above and below the true value to cancel out, so that on average $E[\hat{u}|X] = 0$)

Endogeneity and Bias

- Nearly all independent variables are *endogenous*, they are related to the error term u:

$$cor(X, u) \neq 0$$

Example: Suppose we estimate the following relationship:

$$\text{Violent crimes}_t = \beta_0 + \beta_1 \text{Ice cream sales}_t + u_t$$

- We find $\hat{\beta}_1 > 0$
- It does not mean that Ice cream sales cause Violent crimes!
- The true expected value of $\hat{\beta}_1$ is actually:

$$E[\hat{\beta}_1] = \beta_1 + cor(X, u)\frac{\sigma_u}{\sigma_X}$$

- If X is exogenous: $cor(X, u) = 0$, this reduces to β_1
- The larger $cor(X, u)$, the larger the bias: $\left(E[\hat{\beta}_1] - \beta_1\right)$
- The direction of the bias depends on $cor(X, u)$:
  - Positive $cor(X, u)$ overestimates the true β_1 ($\hat{\beta}_1$ is too high
  - Negative $cor(X, u)$ underestimates the true β_1 ($\hat{\beta}_1$ is too low

Example: Suppose we estimate the following relationship:

$$wages_i = \beta_0 + \beta_1 education_i + u$$

- Is this an accurate reflection of the effect of *education* on *wages*?
- Is $E[u|education] = 0$?
- What would $E[u|education] > 0$ mean?

Regression of test scores on student ratios

```r
library(broom)
ols.tidy <- tidy(ols, conf.int = TRUE)
ols.tidy
```

```
## # A tibble: 2 x 7
##   term        estimate std.error statistic   p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept)  777.       18.4       42.2  1.45e-166   741.      813.
## 2 str_s         -0.950     0.758     -1.25 2.11e-  1    -2.44      0.539
```

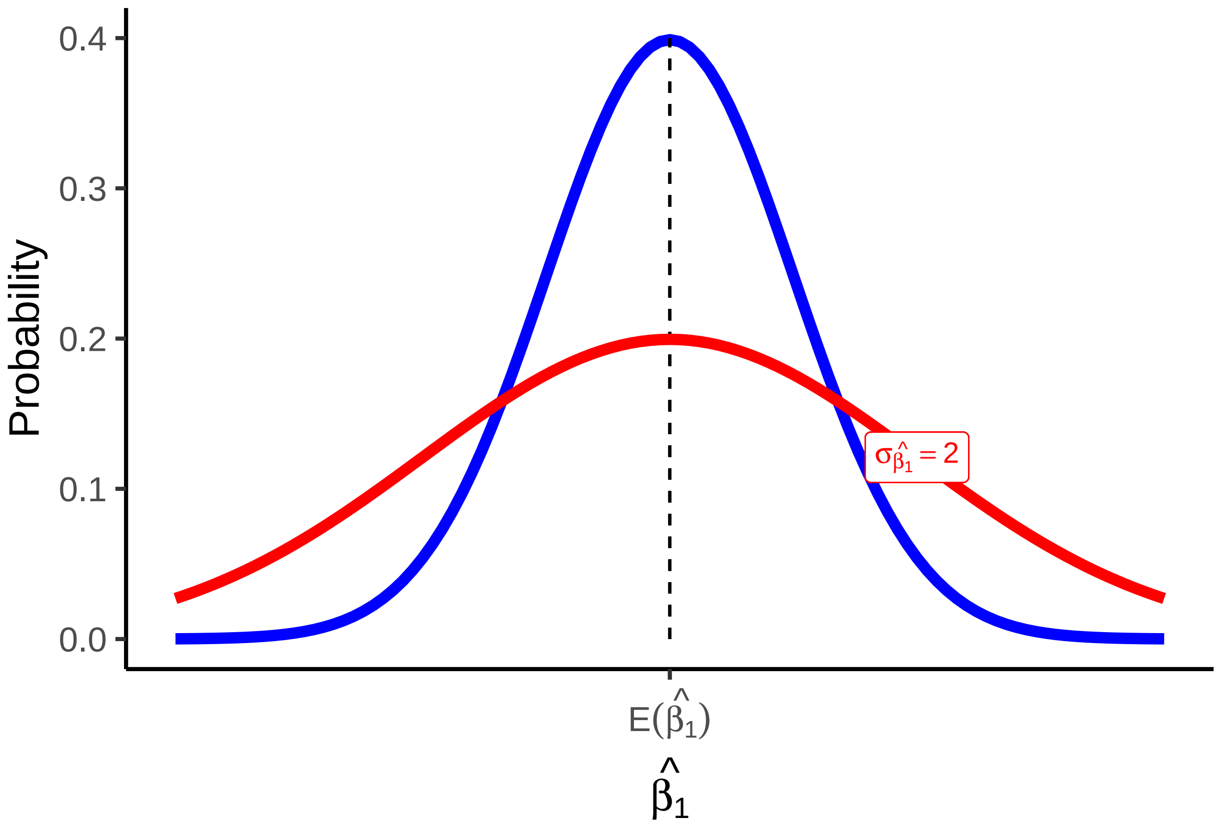
The Sampling Distribution of $\hat{\beta}_1$

$$\hat{\beta}_1 \sim N(E[\hat{\beta}_1], \sigma_{\hat{\beta}_1})$$

- Standard **"error"** is the analog of standard *deviation* when talking about the *sampling distribution* of a sample statistic (such as $\bar{X}$ or $\hat{\beta}_1$).

```r
beta.dist +
  stat_function(fun=dnorm, args=list(mean=0, sd=2), size=2, color="red") +
  geom_label(x=2, y=dnorm(2,0,2), label=expression(sigma[hat(beta[1])]==2), color="red")
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'
```



What Affects Variation in $\hat{\beta}_1$

$$var(\hat{\beta}_1) = \frac{(SER)^2}{n \times var(X)}$$

$$se(\hat{\beta}_1) = \sqrt{var(\hat{\beta}_1)} = \frac{SER}{\sqrt{n} \times sd(X)}$$

- Variation in $\hat{\beta}_1$ is affected by 3 things:

1. Goodness of fit of the model SER:

- Larger $SER \implies$ larger $var(\hat{\beta}_1)$

2. Sample size n

- Larger $n \implies$ smaller $var(\hat{\beta}_1)$

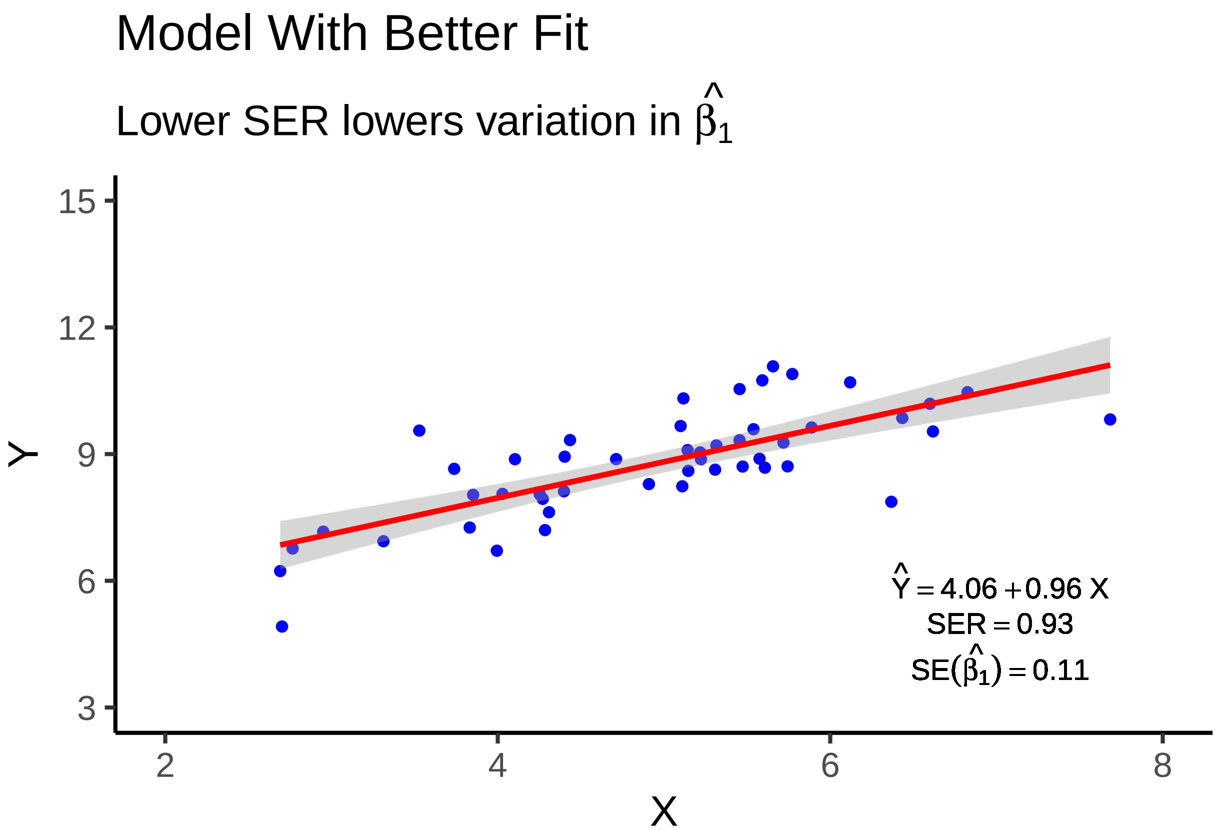
3. Variance of X:

- Larger $var(X) \implies$ smaller $var(\hat{\beta}_1)$

```
df1 <- tibble(x=rnorm(50,5,1),
              u=rnorm(50,1,1),
              y=3+x+u)

sd_x_1 <- lm(y~x, data=df1) %>%
  tidy() %>%
  slice(2) %>% # get second row (which is x coefficient, beta 1)
  pull(std.error) %>%
  round(.,2) %>%
  as.character()

beta0_1 <- lm(y~x, data=df1) %>%
  tidy() %>%
  slice(1) %>% # get first row (which is intercept, beta 0)
  pull(estimate) %>%
  round(.,2) %>%
  as.character()

beta1_1 <- lm(y~x, data=df1) %>%
  tidy() %>%
  slice(2) %>% # get second row (which is x coefficient, beta 1)
  pull(estimate)%>%
  round(.,2) %>%
  as.character()

ser_1 <- lm(y~x, data=df1) %>%
  glance() %>%
  pull(sigma) %>%
  round(.,2) %>%
  as.character()

ggplot(data=df1, aes(x=x, y=y)) +
  geom_point(color="blue") +
  geom_smooth(method="lm", color="red") +
  geom_text(aes(x=7,y=6),
            label=list(paste('~hat(Y)==', beta0_1, '~+', beta1_1, '~X')), parse=TRUE) +
  geom_text(aes(x=7,y=5),
            label=list(paste('~SER==', ser_1)), parse=TRUE) +
  geom_text(aes(x=7,y=4),
            label=list(paste('~SE(hat(beta[1])) ==', sd_x_1)), parse=TRUE) +
```

```
  scale_x_continuous(breaks=seq(2,8,2),
                     limits=c(2,8)) +
  scale_y_continuous(breaks=seq(3,15,3),
                     limits=c(3,15)) +
  labs(x = "X",
       y = "Y",
       title = "Model With Better Fit",
       subtitle = expression(paste("Lower SER lowers variation in ", hat(beta[1]))))
```

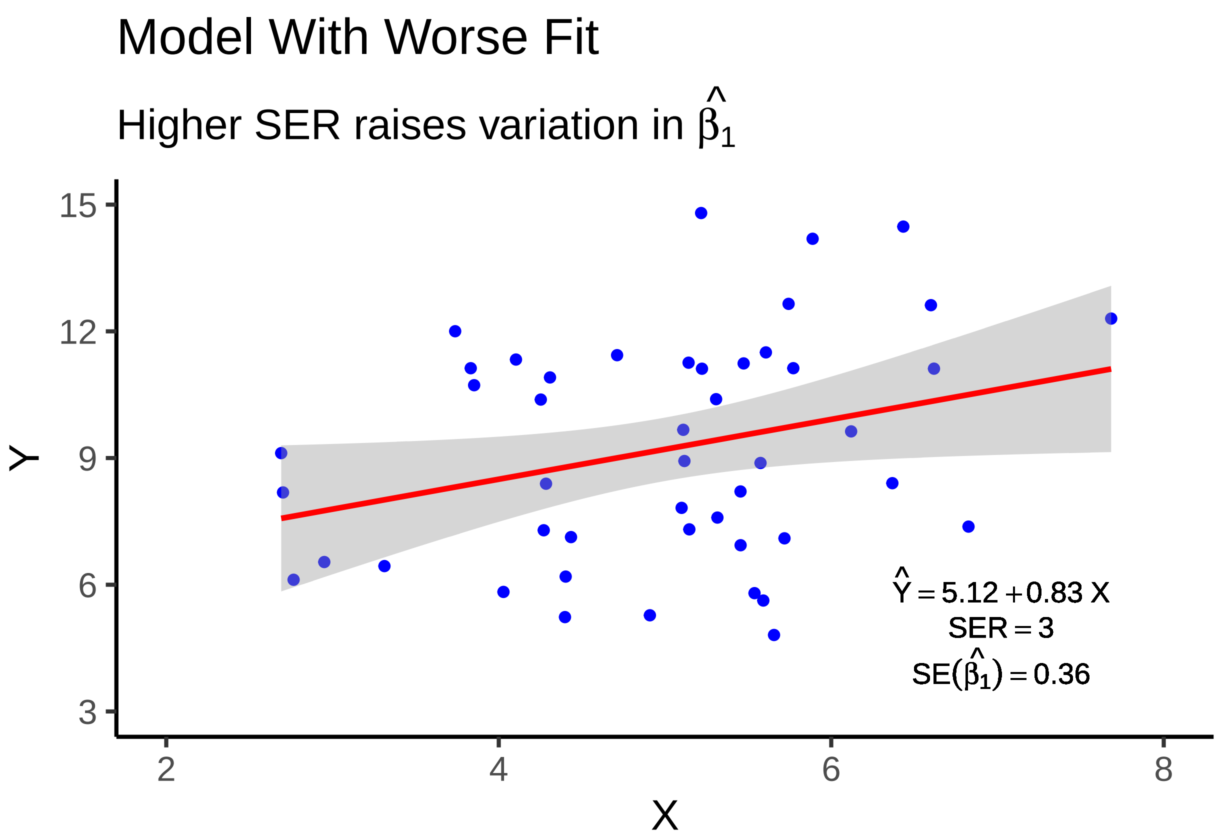
```
## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 1 rows containing non-finite values (stat_smooth).

## Warning: Removed 1 rows containing missing values (geom_point).
```




```
df2 <- tibble(x=df1$x,
              u=rnorm(50,1,3),
              y=3+x+u)

sd_x_2 <- lm(y~x, data=df2) %>%
  tidy() %>%
  slice(2) %>% # get second row (which is x coefficient, beta 1)
  pull(std.error) %>%
  round(.,2) %>%
  as.character()

beta0_2 <- lm(y~x, data=df2) %>%
  tidy() %>%
  slice(1) %>% # get first row (which is intercept, beta 0)
```

```r
  pull(estimate) %>%
  round(.,2) %>%
  as.character()

beta1_2 <- lm(y~x, data=df2) %>%
  tidy() %>%
  slice(2) %>% # get second row (which is x coefficient, beta 1)
  pull(estimate) %>%
  round(.,2) %>%
  as.character()

ser_2 <- lm(y~x, data=df2) %>%
  glance() %>%
  pull(sigma) %>%
  round(.,2) %>%
  as.character()

ggplot(data=df2, aes(x=x, y=y)) +
  geom_point(color="blue") +
  geom_smooth(method="lm", color="red") +
  geom_text(aes(x=7,y=6), label=list(paste('~hat(Y)==', beta0_2, '~+', beta1_2, '~X')), parse=TRUE) +
  geom_text(aes(x=7,y=5), label=list(paste('~SER==', ser_2)), parse=TRUE) +
  geom_text(aes(x=7,y=4), label=list(paste('~SE(hat(beta[1])) ==', sd_x_2)), parse=TRUE) +
  scale_x_continuous(breaks=seq(2,8,2),
                     limits=c(2,8)) +
  scale_y_continuous(breaks=seq(3,15,3),
                     limits=c(3,15)) +
  labs(x = "X",
       y = "Y",
       title = "Model With Worse Fit",
       subtitle = expression(paste("Higher SER raises variation in ", hat(beta[1]))))
```

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

## Warning: Removed 3 rows containing non-finite values (stat_smooth).

## Warning: Removed 3 rows containing missing values (geom_point).
```



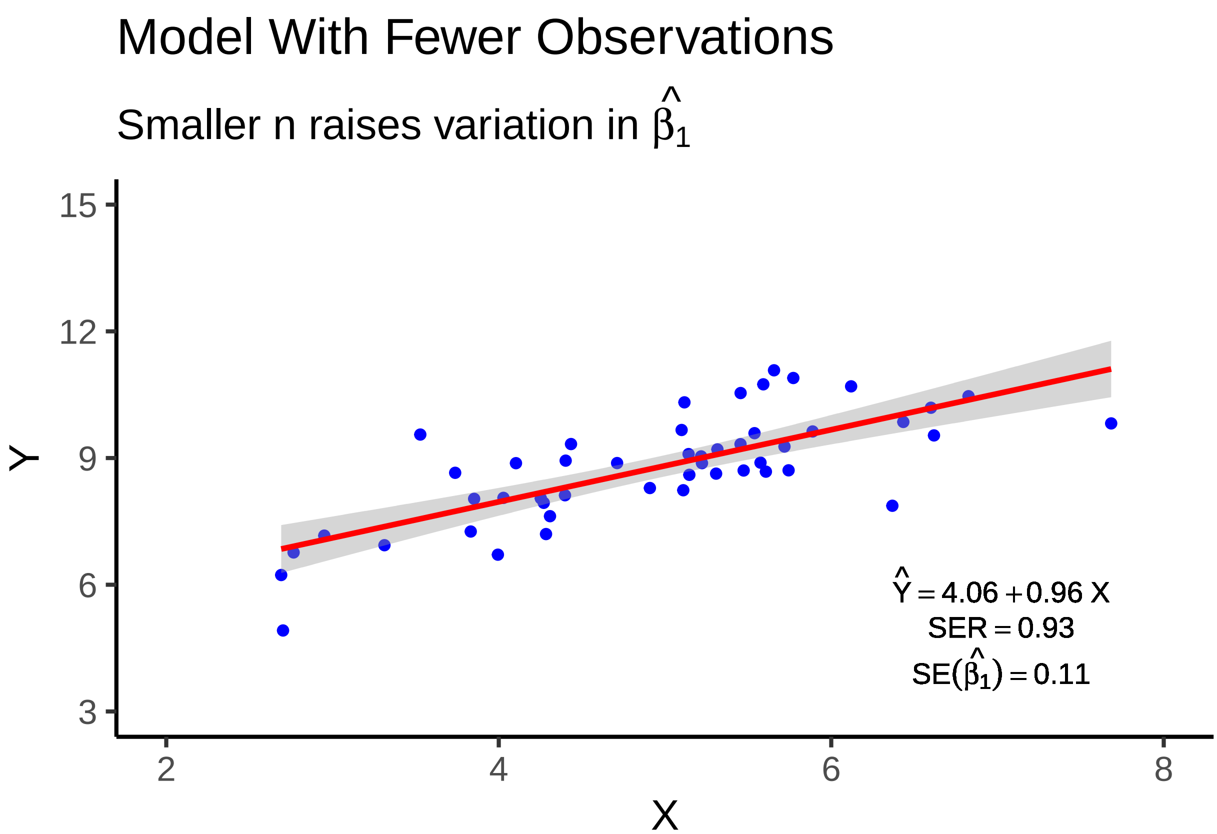
Variation in $\hat{\beta}_1$: Sample Size

```
ggplot(data=df1, aes(x=x, y=y)) +
  geom_point(color="blue") +
  geom_smooth(method="lm", color="red") +
  geom_text(aes(x=7,y=6), label=list(paste('~hat(Y)==', beta0_1, '~+', beta1_1, '~X')), parse=TRUE) +
  geom_text(aes(x=7,y=5), label=list(paste('~SER==', ser_1)), parse=TRUE) +
  geom_text(aes(x=7,y=4), label=list(paste('~SE(hat(beta[1])) ==', sd_x_1)), parse=TRUE) +
  scale_x_continuous(breaks=seq(2,8,2),
                     limits=c(2,8)) +
  scale_y_continuous(breaks=seq(3,15,3),
                     limits=c(3,15)) +
  labs(x = "X",
       y = "Y",
       title = "Model With Fewer Observations",
       subtitle = expression(paste("Smaller n raises variation in ", hat(beta[1]))))
```

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

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

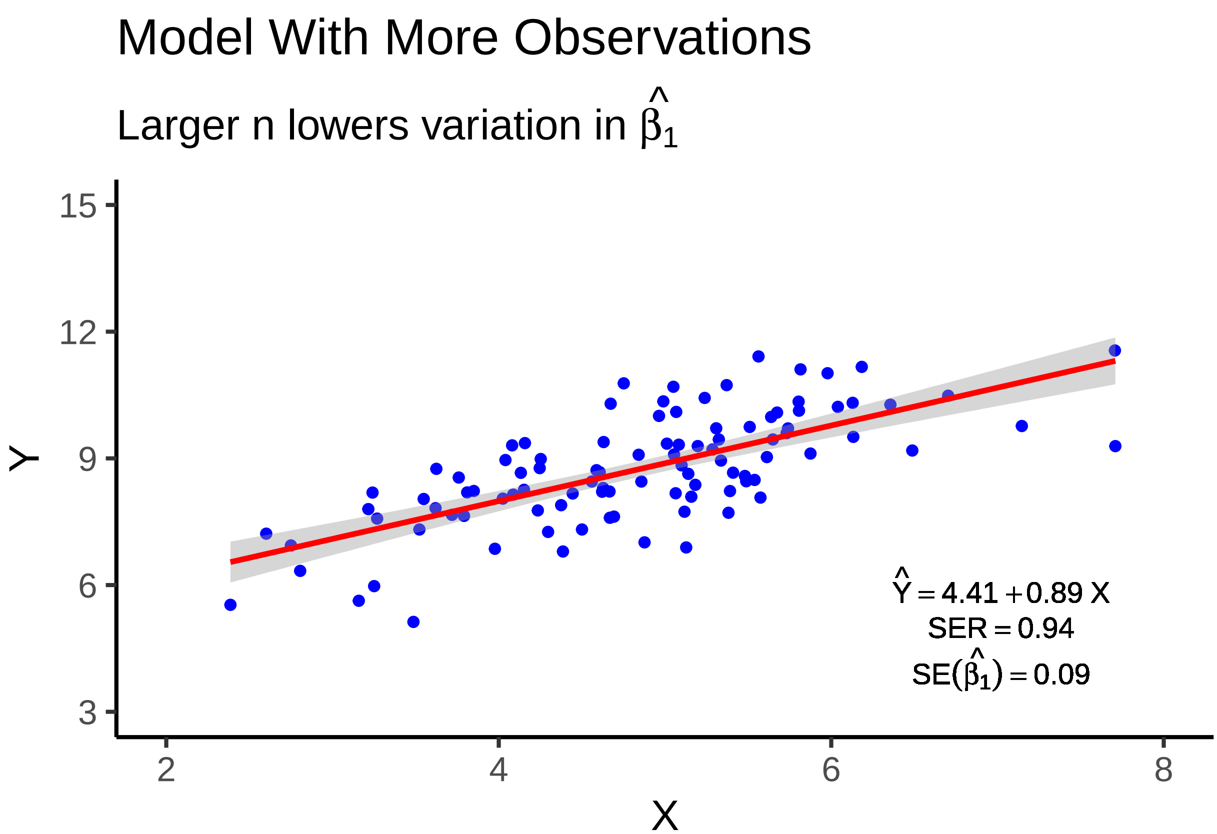
```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
df3 <- tibble(x=rnorm(100,5,1),
              u=rnorm(100,1,1),
              y=3+x+u)

sd_x_3 <- lm(y~x, data=df3) %>%
  tidy() %>%
  slice(2) %>% # get second row (which is x coefficient, beta 1)
  pull(std.error) %>%
  round(.,2) %>%
  as.character()

beta0_3 <- lm(y~x, data=df3) %>%
  tidy() %>%
  slice(1) %>% # get first row (which is intercept, beta 0)
  pull(estimate)  %>%
  round(.,2) %>%
  as.character()

beta1_3 <- lm(y~x, data=df3) %>%
  tidy() %>%
  slice(2) %>% # get second row (which is x coefficient, beta 1)
  pull(estimate) %>%
  round(.,2) %>%
  as.character()

ser_3 <- lm(y~x, data=df3) %>%
  glance() %>%
  pull(sigma) %>%
```

```
  round(.,2) %>%
  as.character()

ggplot(data=df3, aes(x=x, y=y)) +
  geom_point(color="blue") +
  geom_smooth(method="lm", color="red") +
  geom_text(aes(x=7,y=6), label=list(paste('~hat(Y)==', beta0_3, '~+', beta1_3, '~X')), parse=TRUE) +
  geom_text(aes(x=7,y=5), label=list(paste('~SER==', ser_3)), parse=TRUE) +
  geom_text(aes(x=7,y=4), label=list(paste('~SE(hat(beta[1])) ==', sd_x_3)), parse=TRUE) +
  scale_x_continuous(breaks=seq(2,8,2),
                     limits=c(2,8)) +
  scale_y_continuous(breaks=seq(3,15,3),
                     limits=c(3,15)) +
  labs(x = "X",
       y = "Y",
       title = "Model With More Observations",
       subtitle = expression(paste("Larger n lowers variation in ", hat(beta[1]))))
```

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

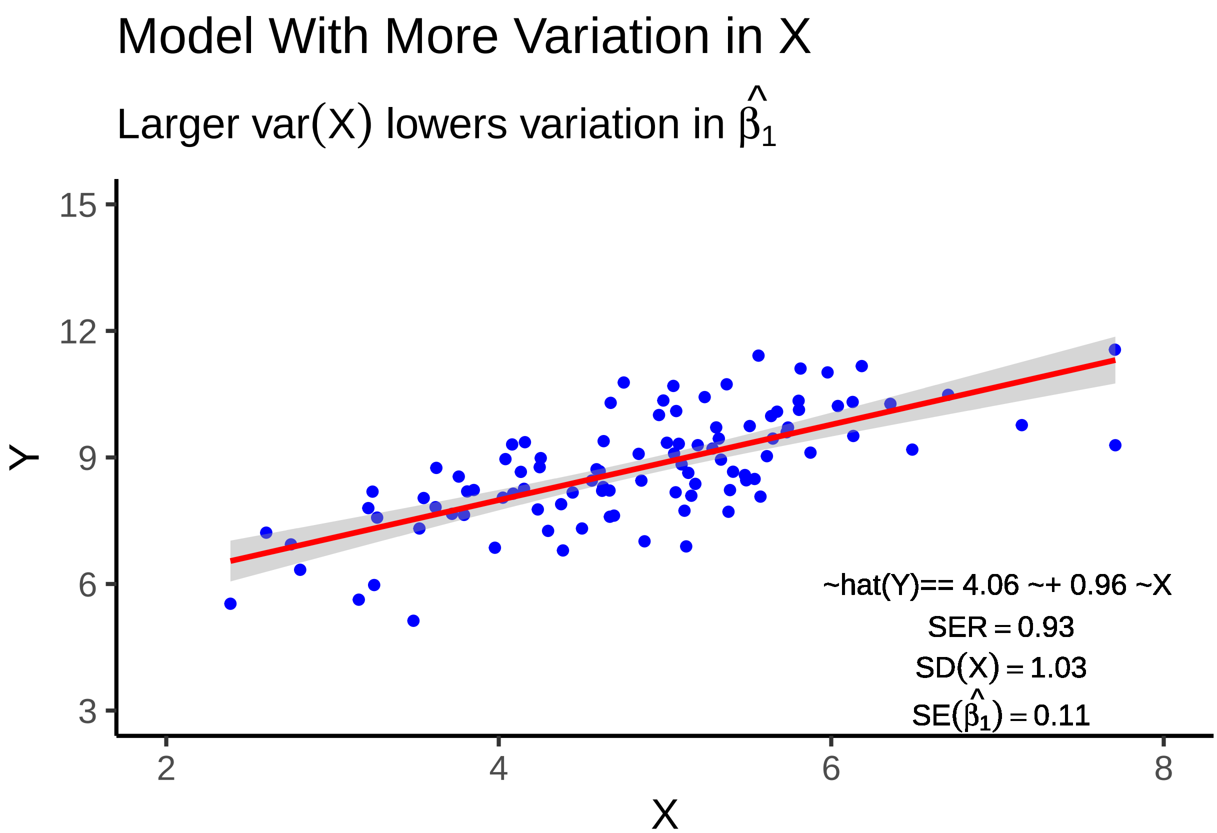


Variation in $\hat{\beta}_1$: Variation in X

```
sd_X_3 <- df3 %>%
  summarize(sd_x_3=sd(x)) %>%
  round(.,2) %>%
  as.character()
```

```r
ggplot(data=df3, aes(x=x, y=y)) +
  geom_point(color="blue") +
  geom_smooth(method="lm", color="red") +
  geom_text(aes(x=7,y=6), label=list(paste('~hat(Y)==', beta0_1, '~+', beta1_1, '~X'))) +
  geom_text(aes(x=7,y=5), label=list(paste('~SER==', ser_1)), parse=TRUE) +
  geom_text(aes(x=7,y=4), label=list(paste('~SD(X) ==', sd_X_3)), parse=TRUE) +
  geom_text(aes(x=7,y=3), label=list(paste('~SE(hat(beta[1])) ==', sd_x_1)), parse=TRUE) +
  scale_x_continuous(breaks=seq(2,8,2),
                     limits=c(2,8)) +
  scale_y_continuous(breaks=seq(3,15,3),
                     limits=c(3,15)) +
  labs(x = "X",
       y = "Y",
       title = "Model With More Variation in X",
       subtitle = expression(paste("Larger ", var(X), " lowers variation in ", hat(beta[1]))))
```

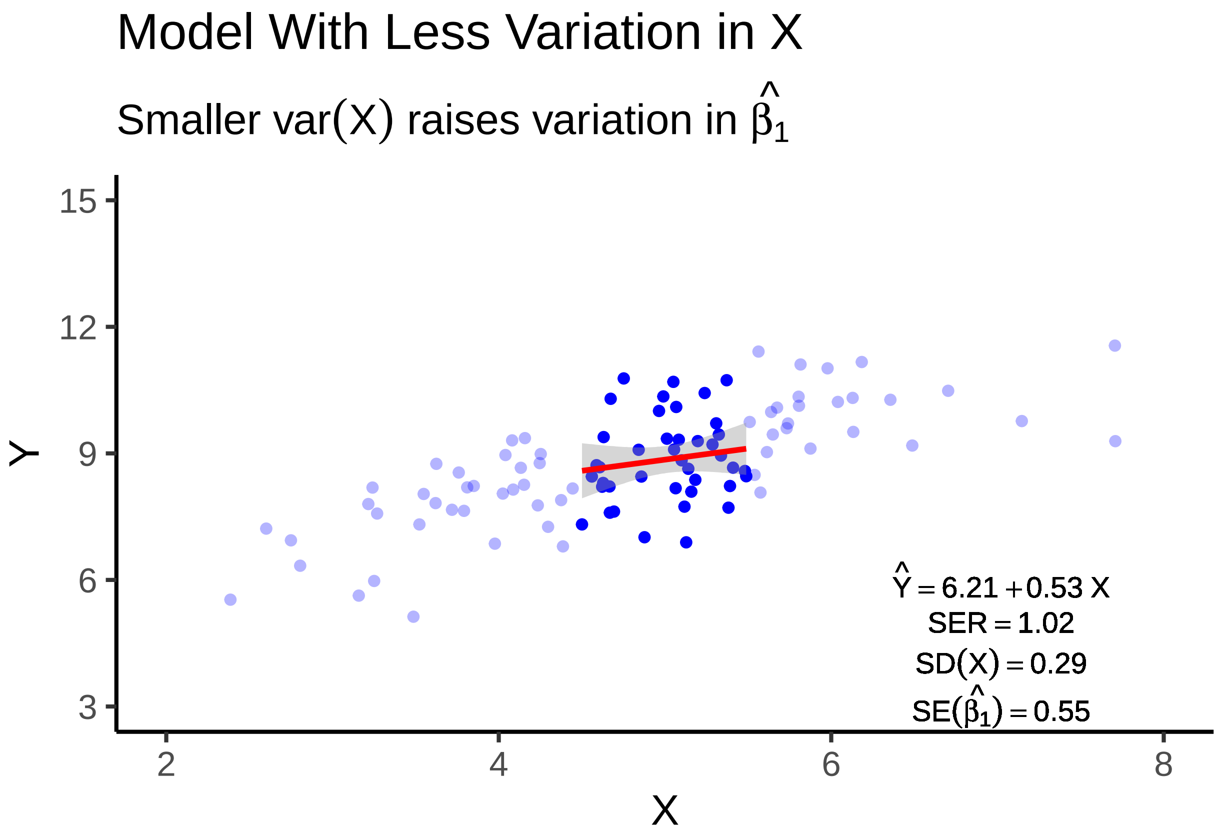
```
## `geom_smooth()` using formula 'y ~ x'
```




```r
df4 <- df3 %>%
  filter(x>4.5, x<5.5)

sd_X_4 <- df4 %>%
  summarize(sd_x_4=sd(x)) %>%
  round(.,2) %>%
  as.character()

sd_x_4 <- lm(y~x, data=df4) %>%
  tidy() %>%
```

```r
  slice(2) %>% # get second row (which is x coefficient, beta 1)
  pull(std.error) %>%
  round(.,2) %>%
  as.character()

beta0_4 <- lm(y~x, data=df4) %>%
  tidy() %>%
  slice(1) %>% # get first row (which is intercept, beta 0)
  pull(estimate) %>%
  round(.,2) %>%
  as.character()

beta1_4 <- lm(y~x, data=df4) %>%
  tidy() %>%
  slice(2) %>% # get second row (which is x coefficient, beta 1)
  pull(estimate) %>%
  round(.,2) %>%
  as.character()

ser_4 <- lm(y~x, data=df4) %>%
  glance() %>%
  pull(sigma) %>%
  round(.,2) %>%
  as.character()

ggplot(data=df4, aes(x=x, y=y)) +
  geom_point(color="blue") +
  geom_point(data=df3, aes(x=x, y=y), color="blue", alpha=0.3) +
  geom_smooth(method = "lm", color = "red") +
  geom_text(aes(x=7,y=6), label=list(paste('~hat(Y)==', beta0_4, '~+', beta1_4, '~X')), parse=TRUE) +
  geom_text(aes(x=7,y=5), label=list(paste('~SER==', ser_4)), parse=TRUE) +
  geom_text(aes(x=7,y=4), label=list(paste('~SD(X) ==', sd_X_4)), parse=TRUE) +
  geom_text(aes(x=7,y=3), label=list(paste('~SE(hat(beta[1])) ==', sd_x_4)), parse=TRUE) +
  scale_x_continuous(breaks=seq(2,8,2),
                     limits=c(2,8)) +
  scale_y_continuous(breaks=seq(3,15,3),
                     limits=c(3,15)) +
  labs(x = "X",
       y = "Y",
       title = "Model With Less Variation in X",
       subtitle = expression(paste("Smaller ", var(X), " raises variation in ", hat(beta[1]))))
```

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



Presenting Regression Results

```
library(huxtable)
```

```
##
## Attaching package: 'huxtable'

## The following object is masked from 'package:dplyr':
##
##     add_rownames

## The following object is masked from 'package:ggplot2':
##
##     theme_grey
```

```
huxreg(ols)
```

Presenting Regression Results

```
library(huxtable)
huxreg("Test Score" = ols,
       coefs=c("Intercept" = "(Intercept)",
                      "STR" = "str_s"),
       statistics=c("N" = "nobs",
            "R-Squared" = "r.squared",
                  "SER" = "sigma"),
       number_format=2)
```

- Can give title to each column

	(1)
(Intercept)	776.678 ***
	(18.406)
str_s	-0.950
	(0.758)
N	500
R2	0.003
logLik	-2757.876
AIC	5521.753

*** p < 0.001; ** p < 0.01; * p < 0.05.

	Test Score
Intercept	776.68 ***
	(18.41)
STR	-0.95
	(0.76)
N	500
R-Squared	0.00
SER	60.27

*** p < 0.001; ** p < 0.01; * p < 0.05.

- Can rename coefficients
- Can choose what statistics to include
- Can choose how many decimal places to round to

Regression Diagnostics

- Examine the Residuals:

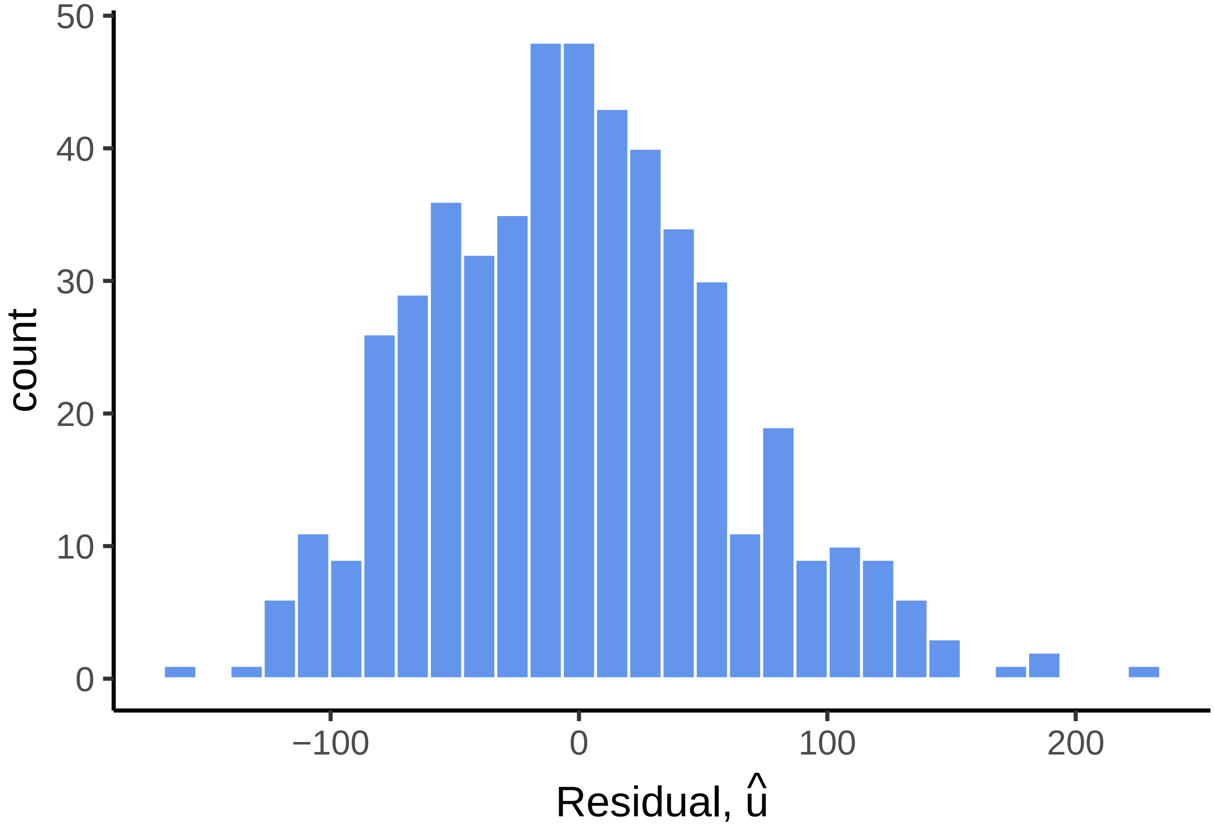
```
ols.augment %>%
  summarize(E_u = mean(.resid),
            sd_u = sd(.resid))
```

E_u	sd_u
-8.12e-14	60.2

- Histogram of the Residuals:

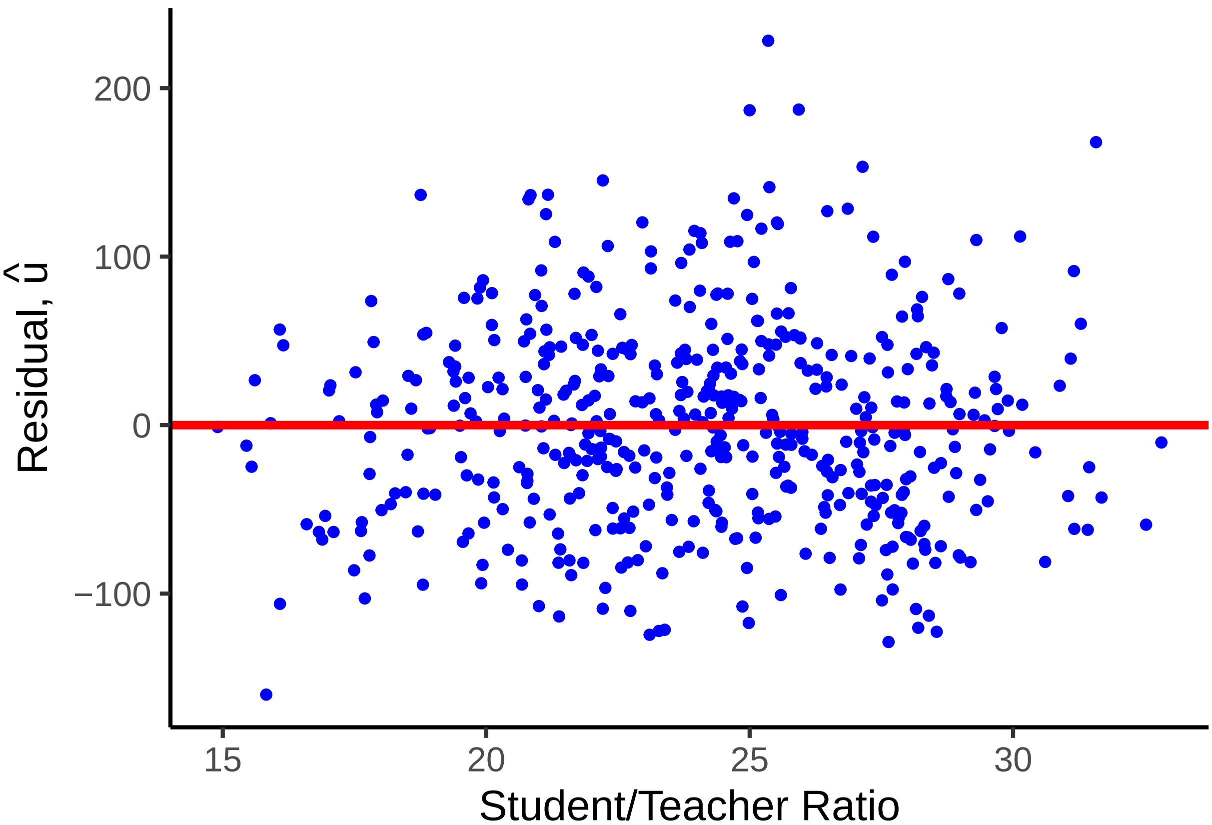
```
ggplot(data=ols.augment, aes(x=.resid)) +
  geom_histogram(color="white", fill="cornflowerblue") +
  labs(x = expression(paste("Residual, ", hat(u))))
```

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



- Scatter of the Resdiuals:
- Look for systematic patterns about residuals
  - x-axis are X values (`str`)
  - y-axis are u values (`.resid`)

```
ggplot(data=ols.augment, aes(x=str_s, y=.resid)) +
  geom_point(color="blue") +
  geom_hline(aes(yintercept = 0), color="red", size=1.5) +
  labs(x = "Student/Teacher Ratio",
       y = expression(paste("Residual, ", hat(u))))
```



Heteroskedasticity

- Heteroskedasticity: variance of the residuals conditional on X is NOT constant:

$$var(u|X) \neq \sigma_u^2$$

- The estimate $\hat{\beta}_1$ is not biased.
- The usual standard error of $\hat{\beta}_1$ is incorrect!
- Inference is invalid.
- Under heteroskedasticity, the standard error of $\hat{\beta}_1$ is:

$$se(\hat{\beta}_1) = \sqrt{\frac{\sum\limits_{i=1}^{n}(X_i - \bar{X})^2 \hat{u}^2}{\left[\sum\limits_{i=1}^{n}(X_i - \bar{X})^2\right]^2}}$$

- Under homoskedasticity, the standard error of $\hat{\beta}_1$ simplifies to:

$$se(\hat{\beta}_1) = \sqrt{var(\hat{\beta}_1)} = \frac{SER}{\sqrt{n} \times sd(X)}$$

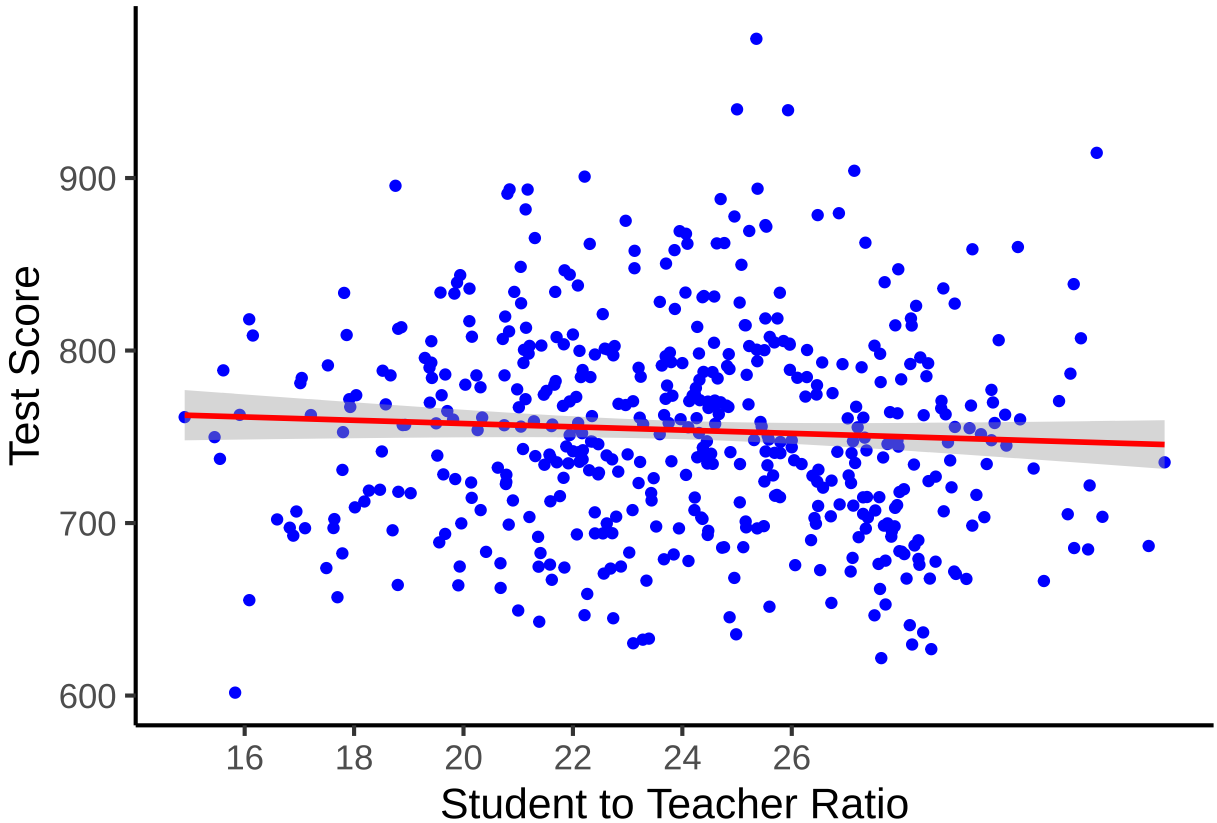
Heteroskedasticity

- Does the spread of the errors change over different values of str?
  - No: homoskedastic

– Yes: heteroskedastic

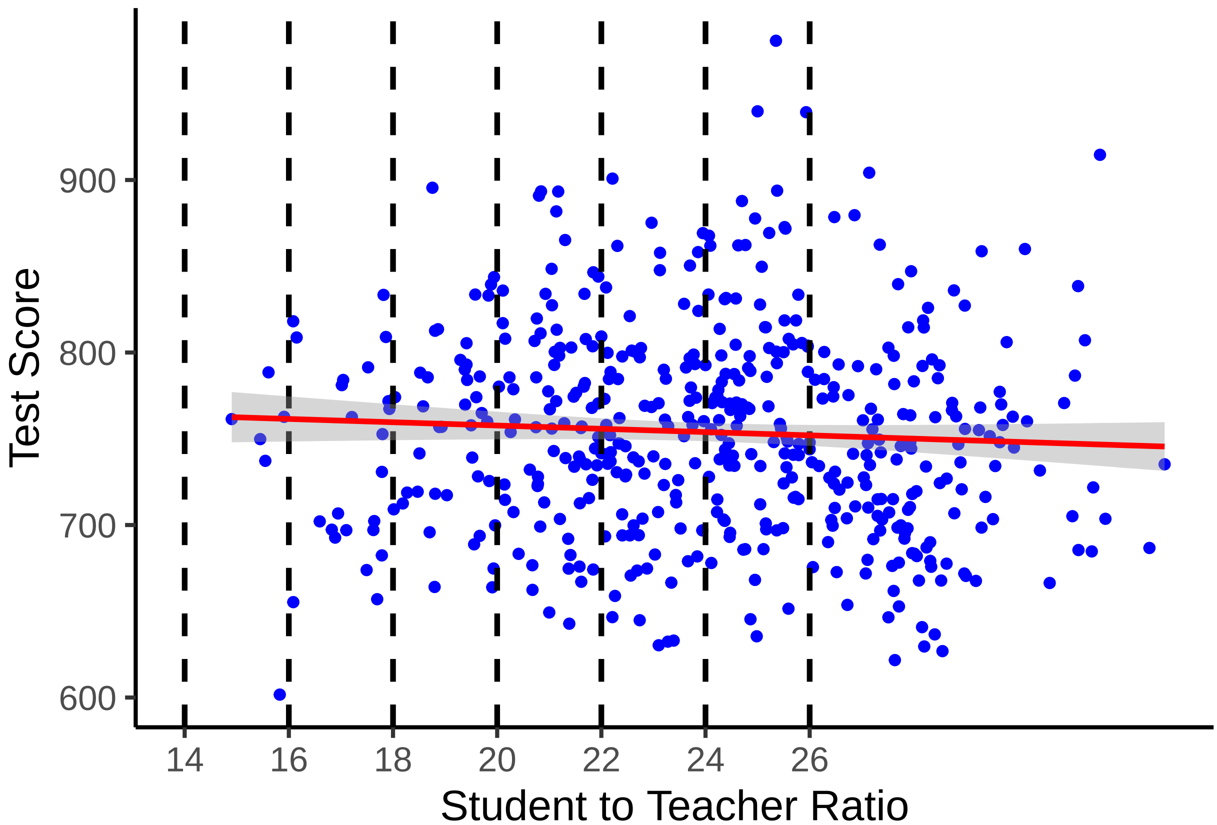
```
ggplot(data=df, aes(x=str_s, y=testscore)) +
  geom_point(color="blue") +
  geom_smooth(method="lm", color="red") +
  scale_x_continuous(breaks = seq(14,26,2)) +
  labs(x = "Student to Teacher Ratio",
       y = "Test Score")
```

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

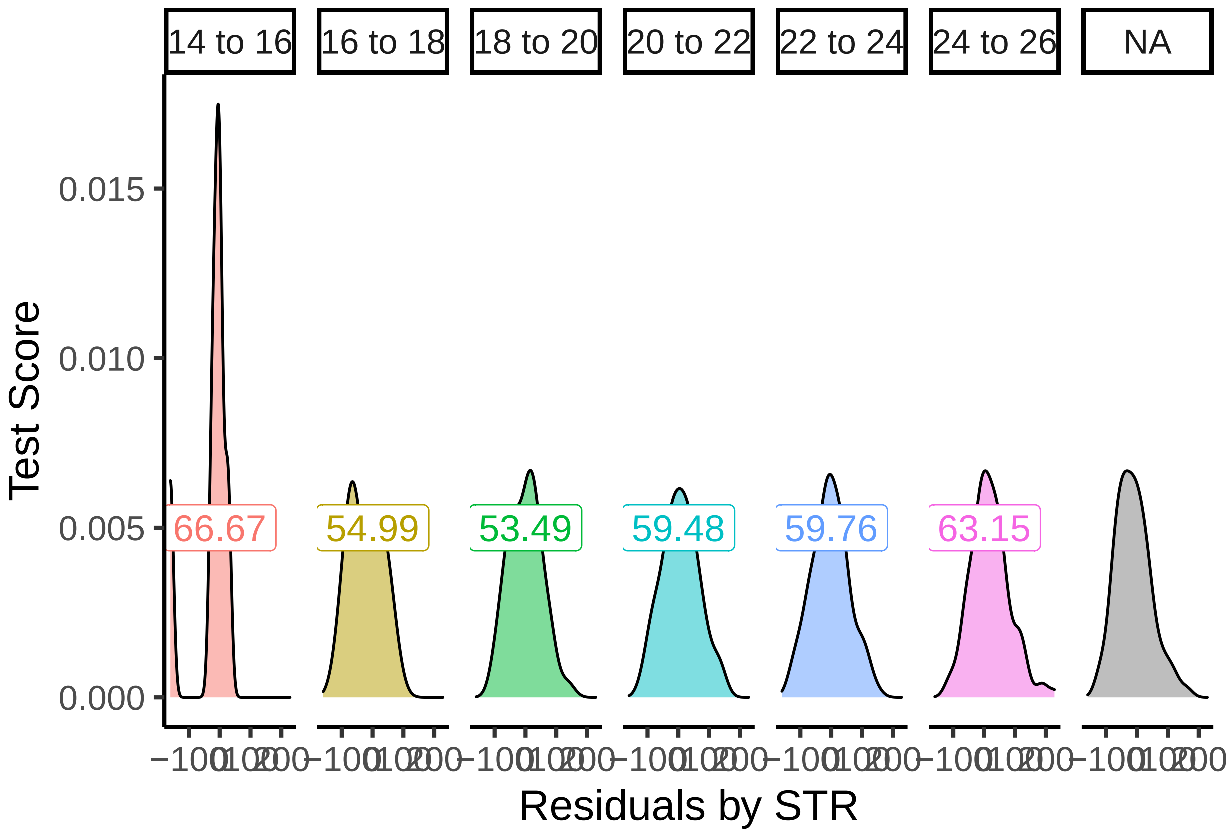



```
ggplot(data=df, aes(x=str_s, y=testscore)) +
  geom_point(color="blue") +
  geom_vline(xintercept=seq(14,26,2), linetype="dashed", size=1) +
  geom_smooth(method="lm", color="red") +
  scale_x_continuous(breaks = seq(14,26,2)) +
  labs(x = "Student to Teacher Ratio",
       y = "Test Score")
```

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




```
ols.augment.het <- ols.augment %>%
  mutate(range = case_when(str_s>=14 & str_s<16 ~ "14 to 16",
                           str_s>=16 & str_s<18 ~ "16 to 18",
                           str_s>=18 & str_s<20 ~ "18 to 20",
                           str_s>=20 & str_s<22 ~ "20 to 22",
                           str_s>=22 & str_s<24 ~ "22 to 24",
                           str_s>=24 & str_s<26 ~ "24 to 26"),
         range = factor(range, levels = c("14 to 16", "16 to 18", "18 to 20", "20 to 22", "22 to 24", "2

ols.augment.het.sigmas <- ols.augment.het %>%
  group_by(range) %>%
  summarize(sigmas = as.character(round(sd(.resid),2))) %>%
  slice(1:6) # remove NA row 7

ggplot(data=ols.augment.het, aes(x=.resid)) +
  geom_density(aes(fill=range), alpha=0.5) +
  geom_label(data=ols.augment.het.sigmas,
             aes(x=0,y=0.005,
                 label=sigmas,
                 color=range), size=5) +
  facet_grid(~range) +
  guides(fill="none", color="none") +
  labs(x = "Residuals by STR",
       y = "Test Score")
```



Visualize Heteroskedasticity

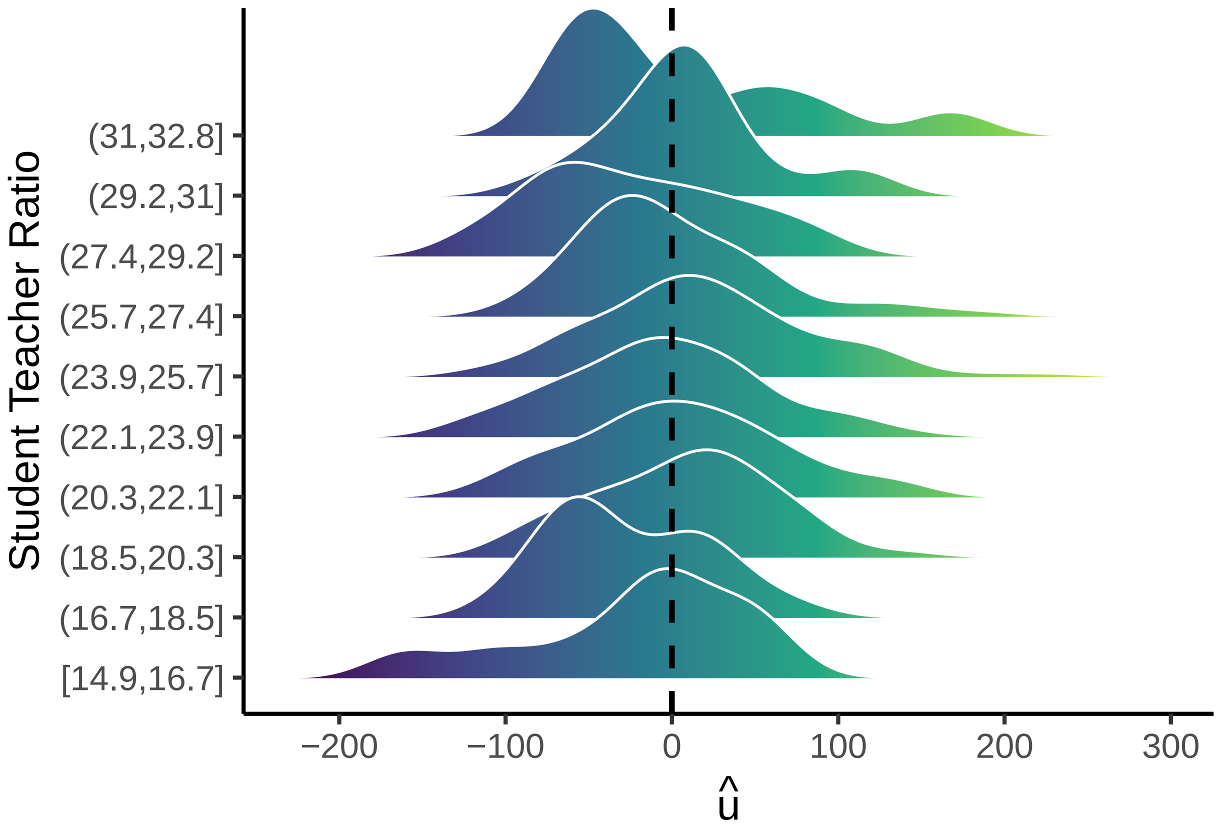
- Using the `ggridges` package
- Plot the (conditional) distribution of errors by STR
- See that the variation in errors ($\hat{u}$) changes across class sizes!

```
library(ggridges)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
ols.augment %>%
  mutate(bins=cut_interval(str_s, n=10)) %>%
ggplot(data=., aes(x=.resid, y=bins)) +
  geom_density_ridges_gradient(
    aes(fill = ..x..),
    color = "white",
    scale = 2.5,
    size = 0.5
  ) +
  geom_vline(xintercept=0, size=1 , linetype="dashed") +
  scale_fill_viridis_c() +
  labs(x = expression(hat(u)),
       y = "Student Teacher Ratio") +
  theme(legend.position="none")
```

```
## Picking joint bandwidth of 23.7
```

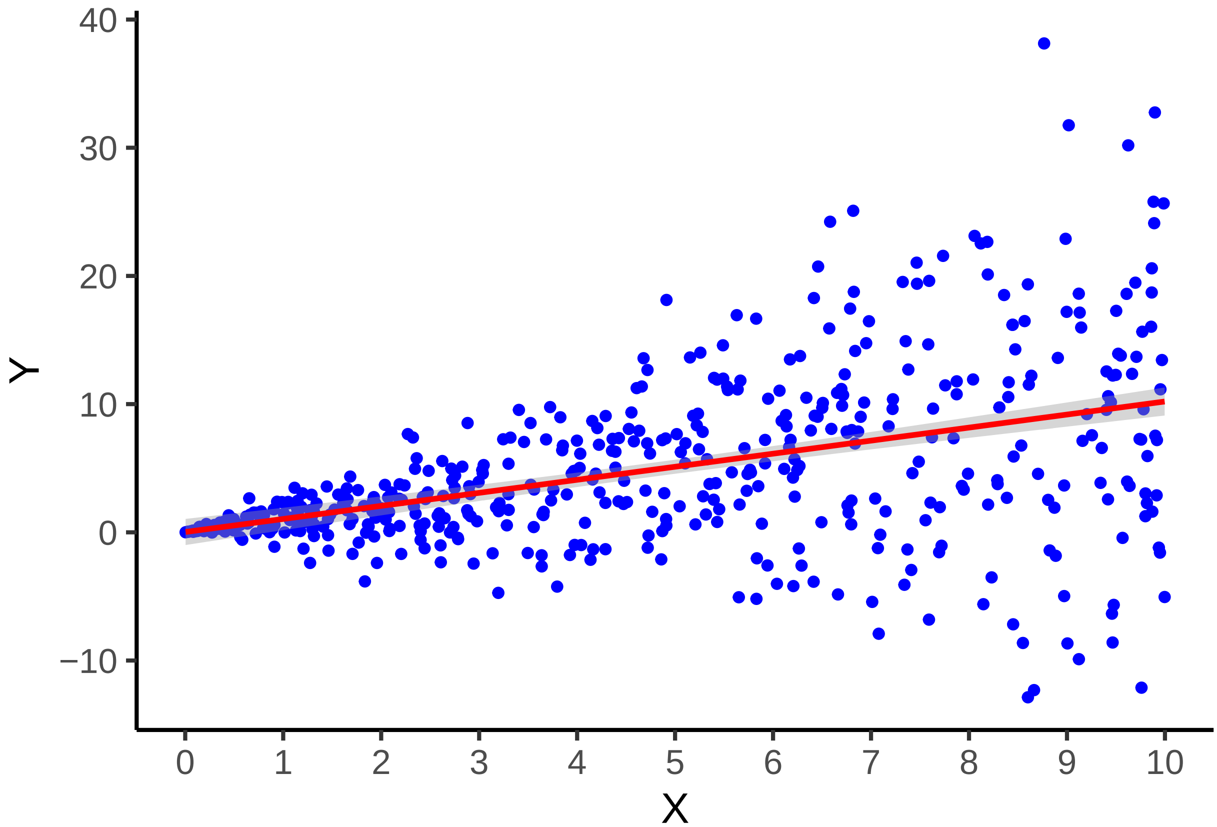


Visualize Heteroskedasticity

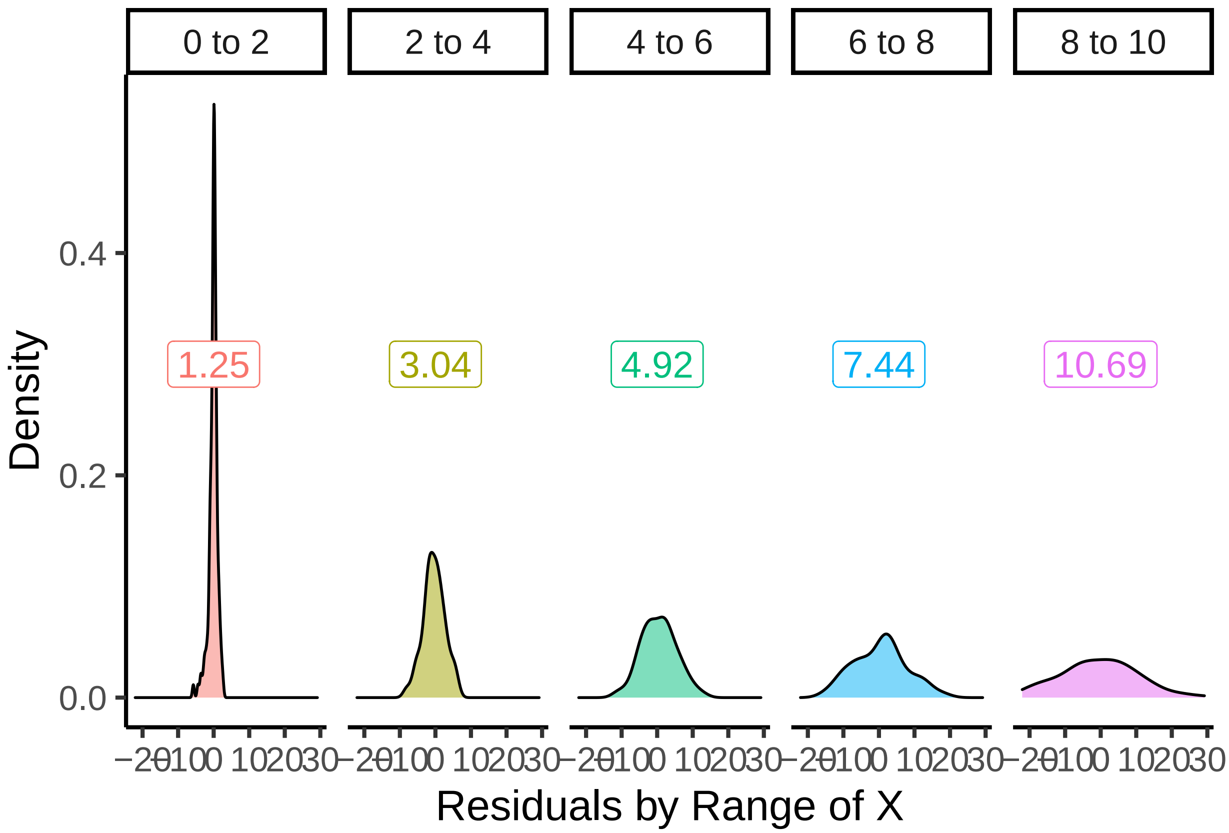
- Visual cue: data is "fan-shaped"
  - Data points are closer to line in some areas
  - Data points are more spread from line in other areas

```
df.het <- tibble(x = runif(500,0,10),
                 y = rnorm(500,x,x))
ggplot(data=df.het, aes(x=x, y=y))  +
  geom_point(color="blue") +
  geom_smooth(method="lm", color="red") +
  scale_x_continuous(breaks=seq(0,10,1)) +
    labs(x="X", y="Y")
```

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




```
het.reg <- lm(y~x, data = df.het)
aug.het.reg <- het.reg %>% augment()
aug.het.reg <- aug.het.reg %>%
  mutate(range = case_when(x>=0 & x<2 ~ "0 to 2",
                           x>=2 & x<4 ~ "2 to 4",
                           x>=4 & x<6 ~ "4 to 6",
                           x>=6 & x<8 ~ "6 to 8",
                           x>=8 & x<=10 ~ "8 to 10"),
         range = factor(range, levels = c("0 to 2", "2 to 4", "4 to 6", "6 to 8", "8 to 10"))) # needs

aug.het.reg_sigmas <- aug.het.reg %>%
  group_by(range) %>%
  summarize(sigmas = as.character(round(sd(.resid),2))) %>%
  slice(1:6) # remove NA row 7

ggplot(data=aug.het.reg, aes(x = .resid)) +
  geom_density(aes(fill=range), alpha=0.5) +
  geom_label(data=aug.het.reg_sigmas,
             aes(x=0, y=0.3,
                 label=sigmas,
                 color=range),size=5) +
  facet_grid(~range) +
  guides(fill="none", color="none") +
  labs(x = "Residuals by Range of X",
       y = "Density")
```

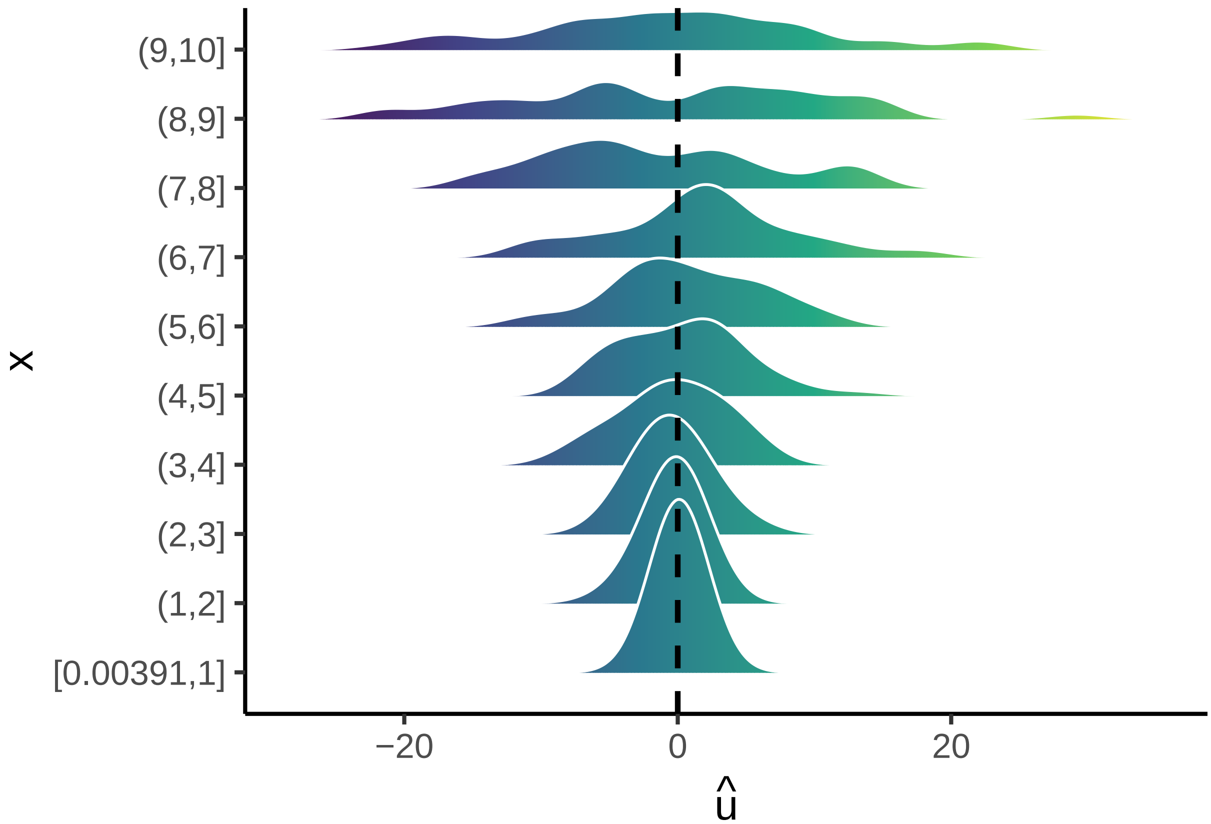


Heteroskedasticity: Another View

- Using the `ggridges` package
- Plotting the (conditional) distribution of errors by x

```
aug.het.reg %>%
  mutate(bins=cut_interval(x, n=10)) %>%
ggplot(data=., aes(x=.resid, y=bins)) +
  geom_density_ridges_gradient(
    aes(fill = ..x..),
    color = "white",
    scale = 2.5,
    size = 0.5
  ) +
  geom_vline(xintercept=0, size=1 , linetype="dashed") +
  scale_fill_viridis_c()+
  labs(x = expression(hat(u)),
       y = "x") +
  theme(legend.position="none")
```

```
## Picking joint bandwidth of 2.12
```



What Might Cause Heteroskedastic Errors?

$$\widehat{wage_i} = \hat{\beta}_0 + \hat{\beta}_1 educ_i$$

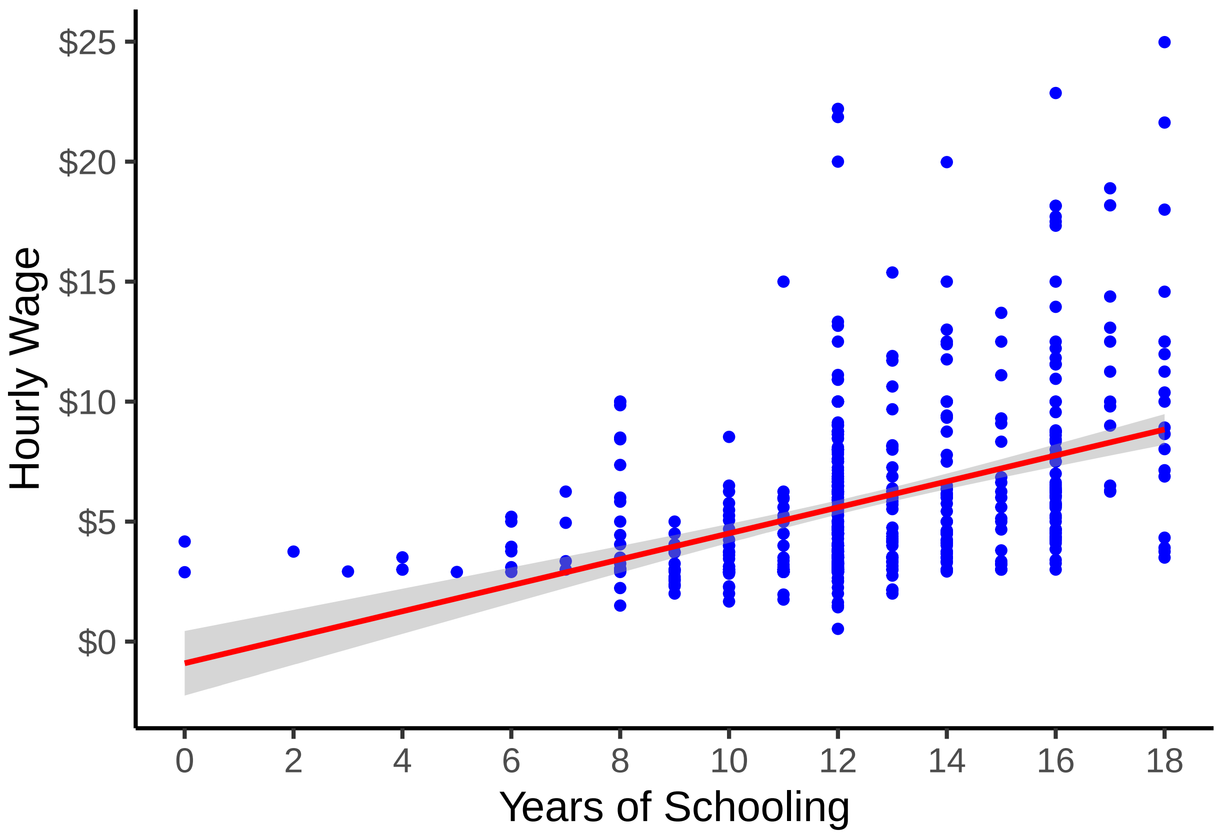
```
library(wooldridge)
wage.reg <- lm(wage~educ, data=wage1)
huxreg("Wage" = wage.reg,
       coefs = c("Intercept" = "(Intercept)",
        "Years of Schooling" = "educ"),
       statistics = c("N" = "nobs",
              "R-Squared" = "r.squared",
                    "SER" = "sigma"),
       number_format=2)
```

```
plot_wage <- ggplot(data=wage1, aes(x=educ, y=wage))+
  geom_point(color="blue") +
  geom_smooth(method="lm", color="red") +
  scale_x_continuous(breaks=seq(0,20,2)) +
  scale_y_continuous(labels=scales::dollar) +
    labs(x = "Years of Schooling",
         y = "Hourly Wage")
plot_wage
```

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

	Wage
Intercept	-0.90
	(0.68)
Years of Schooling	0.54 ***
	(0.05)
N	526
R-Squared	0.16
SER	3.38

*** p < 0.001; ** p < 0.01; * p < 0.05.



What Might Cause Heteroskedastic Errors?

$$\widehat{wage_i} = \hat{\beta}_0 + \hat{\beta}_1 educ_i$$

```
library(wooldridge)
wage.reg <- lm(wage~educ, data=wage1)
huxreg("Wage" = wage.reg,
       coefs = c("Intercept" = "(Intercept)",
```

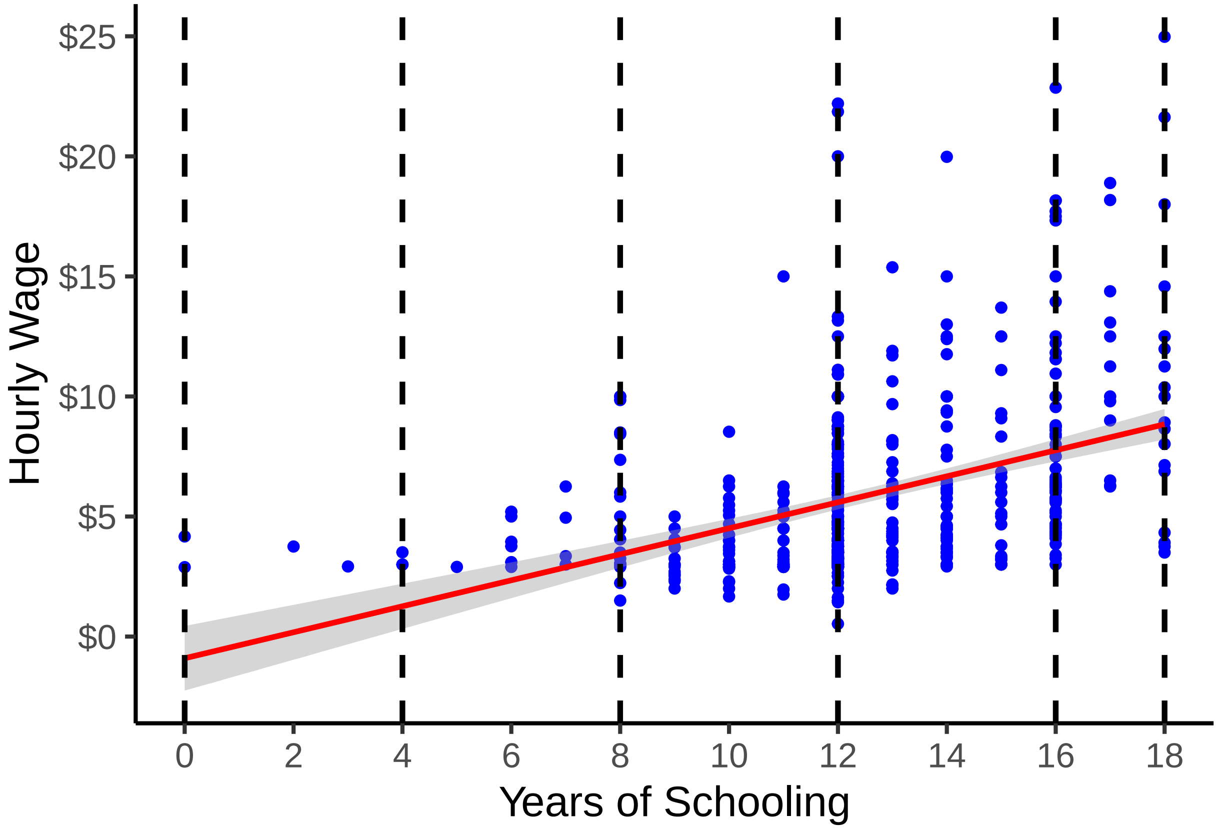
```r
        "Years of Schooling" = "educ"),
       statistics = c("N" = "nobs",
              "R-Squared" = "r.squared",
                    "SER" = "sigma"),
       number_format=2)
```

	Wage
Intercept	-0.90
	(0.68)
Years of Schooling	0.54 ***
	(0.05)
N	526
R-Squared	0.16
SER	3.38

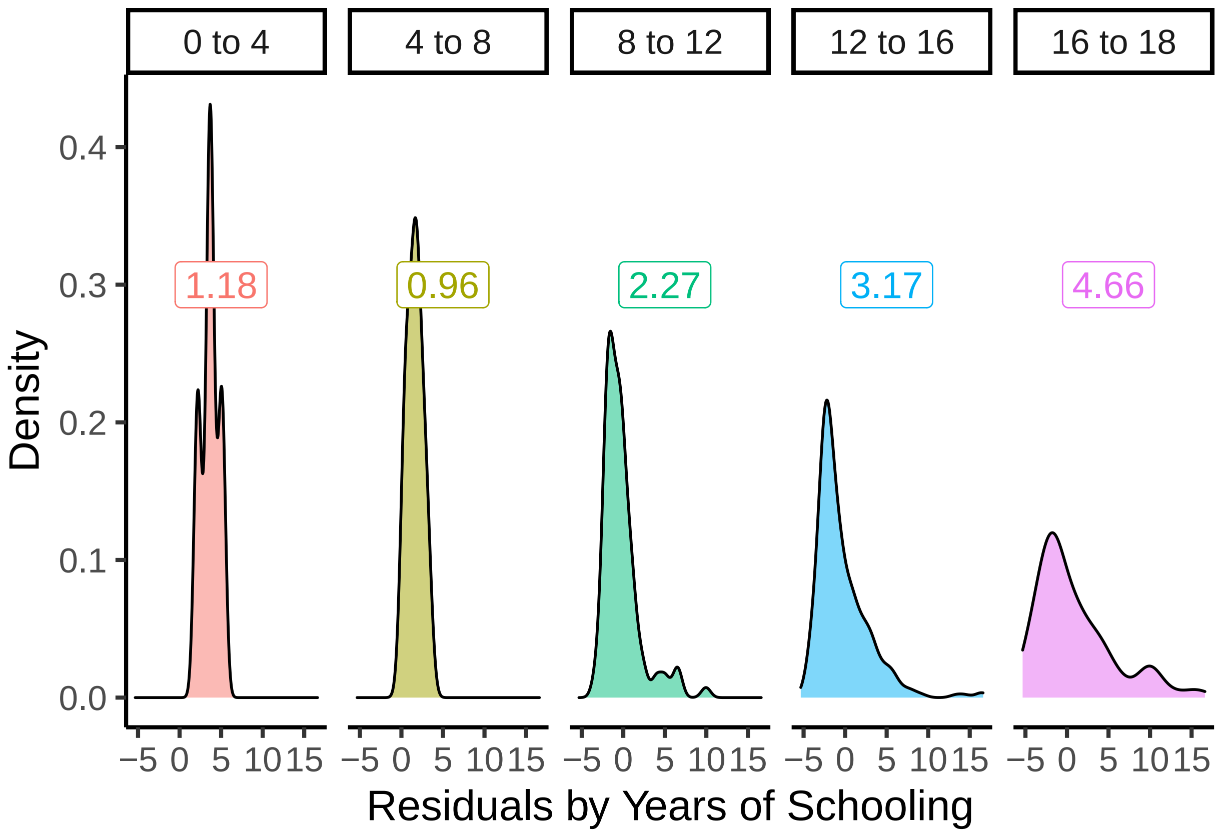
*** p < 0.001; ** p < 0.01; * p < 0.05.

```r
wage.reg <- lm(wage ~ educ, data=wage1)
aug.wage.reg <- wage.reg %>%
  augment()

plot_wage +
  geom_vline(xintercept=c(0,4,8,12,16,18),
             linetype="dashed",
             size=1)
```

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



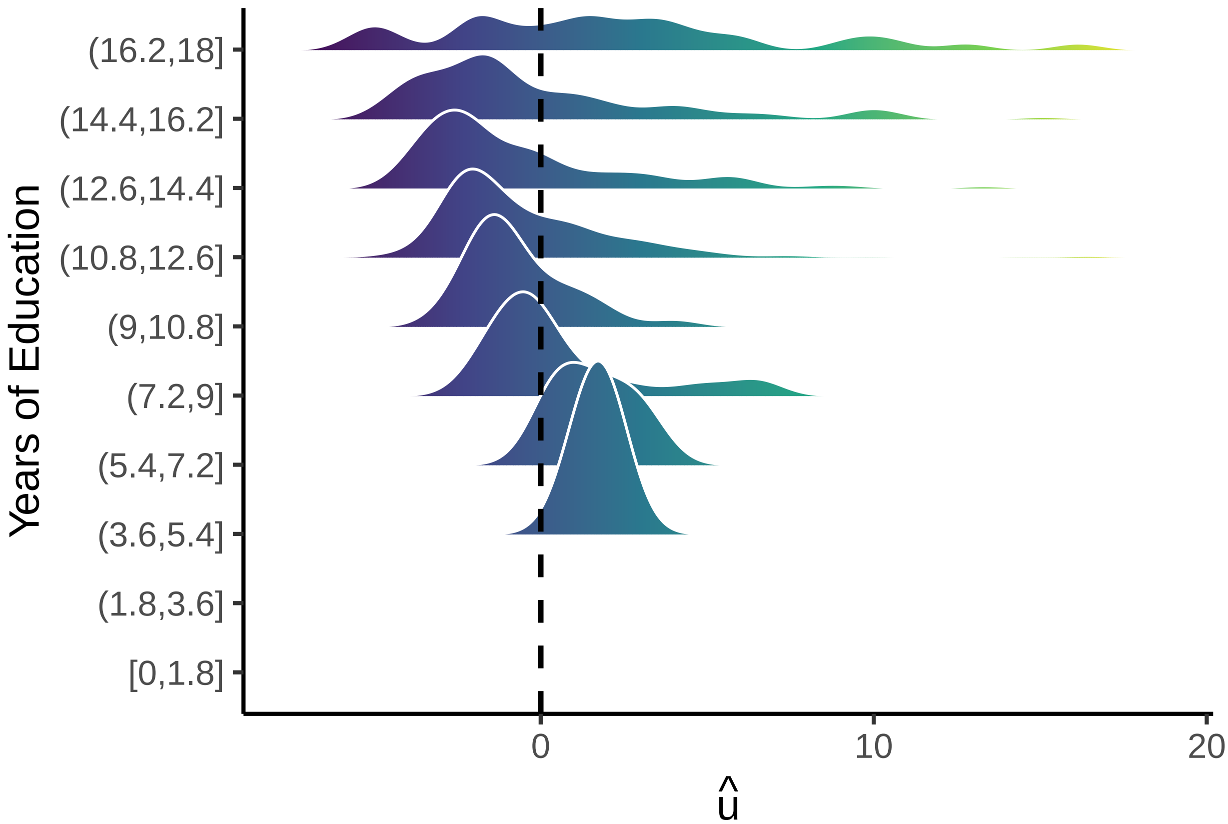

```
aug.wage.reg <- aug.wage.reg %>%
  mutate(range = case_when(educ>=0 & educ<4 ~ "0 to 4",
                           educ>=4 & educ<8 ~ "4 to 8",
                           educ>=8 & educ<12 ~ "8 to 12",
                           educ>=12 & educ<16 ~ "12 to 16",
                           educ>=16 & educ<20 ~ "16 to 18"),
         range = factor(range, levels = c("0 to 4", "4 to 8", "8 to 12", "12 to 16", "16 to 18"))) # ne

aug.wage.reg.het.sigmas <- aug.wage.reg %>%
  group_by(range) %>%
  summarize(sigmas = as.character(round(sd(.resid),2))) %>%
  slice(1:6) # remove NA row 7

ggplot(data=aug.wage.reg, aes(x=.resid)) +
  geom_density(aes(fill=range), alpha=0.5) +
  geom_label(data=aug.wage.reg.het.sigmas,
             aes(x=5,y=0.3,
                 label=sigmas,
                 color=range),size=5) +
  facet_grid(~range) +
  guides(fill="none", color="none") +
  labs(x="Residuals by Years of Schooling",
       y="Density")
```



Heteroskedasticity

```
aug.wage.reg %>%
  mutate(bins = cut_interval(educ, n=10)) %>%
ggplot(data=., aes(x=.resid, y=bins)) +
  geom_density_ridges_gradient(
    aes(fill = ..x..),
    color = "white",
    scale = 2.5,
    size = 0.5
  ) +
  geom_vline(xintercept = 0, size =1 , linetype="dashed") +
  scale_fill_viridis_c() +
  labs(x = expression(hat(u)),
       y = "Years of Education") +
  theme(legend.position="none")
```

```
## Picking joint bandwidth of 0.754
```



Detecting Heteroskedasticity

- Several tests to check if data is heteroskedastic
- One common test is **Breusch-Pagan test**
- Can use `bptest()` with `lmtest` package in `R`
  - H_0: homoskedastic
  - If p-value < 0.05, reject $H_0 \implies$ heteroskedastic

```
# install.packages("lmtest")
library("lmtest")
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
bptest(ols)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  ols
## BP = 0.446, df = 1, p-value = 0.5
```

Fixing Heteroskedasticity

- Heteroskedasticity is easy to fix with software that can calculate robust standard errors
- One approach is to use `estimatr` package
  - `lm_robust()` command (instead of `lm`) to run regression
  - set `se_type="stata"` to calculate robust SEs using the formula above

```
#install.packages("estimatr")
library(estimatr)
ols.robust <-lm_robust(testscore ~ str_s, se_type="stata", data=df)
ols.robust
```

```
##              Estimate Std. Error t value     Pr(>|t|) CI Lower   CI Upper  DF
## (Intercept) 776.67816   18.02677 43.0847 4.1817e-170 741.2603 812.09605 498
## str_s        -0.95009    0.74704 -1.2718  2.0403e-01  -2.4178   0.51764 498
```

Fixing Heteroskedasticity

```
library(huxtable)
huxreg("Normal" = ols,
       "Robust" = ols.robust,
       coefs = c("Intercept" = "(Intercept)",
                 "STR" = "str_s"),
       statistics = c("N" = "nobs",
              "R-Squared" = "r.squared",
                    "SER" = "sigma"),
       number_format=2)
```

	Normal	Robust
Intercept	776.68 ***	776.68 ***
	(18.41)	(18.03)
STR	-0.95	-0.95
	(0.76)	(0.75)
N	500	500
R-Squared	0.00	0.00
SER	60.27	

*** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$.