SOC-GA 2332 Intro to Stats Lab 7

Risa Gelles-Watnick

10/14/2025

```
# load packages
pacman::p_load(
  tidyverse,
  stargazer,
  coefplot,
  sandwich, # robust standard errors
  lmtest,
  ggpubr,
  psych,
  naniar, # missing data
  Amelia # multiple imputation
)
```

Part 0: Housekeeping

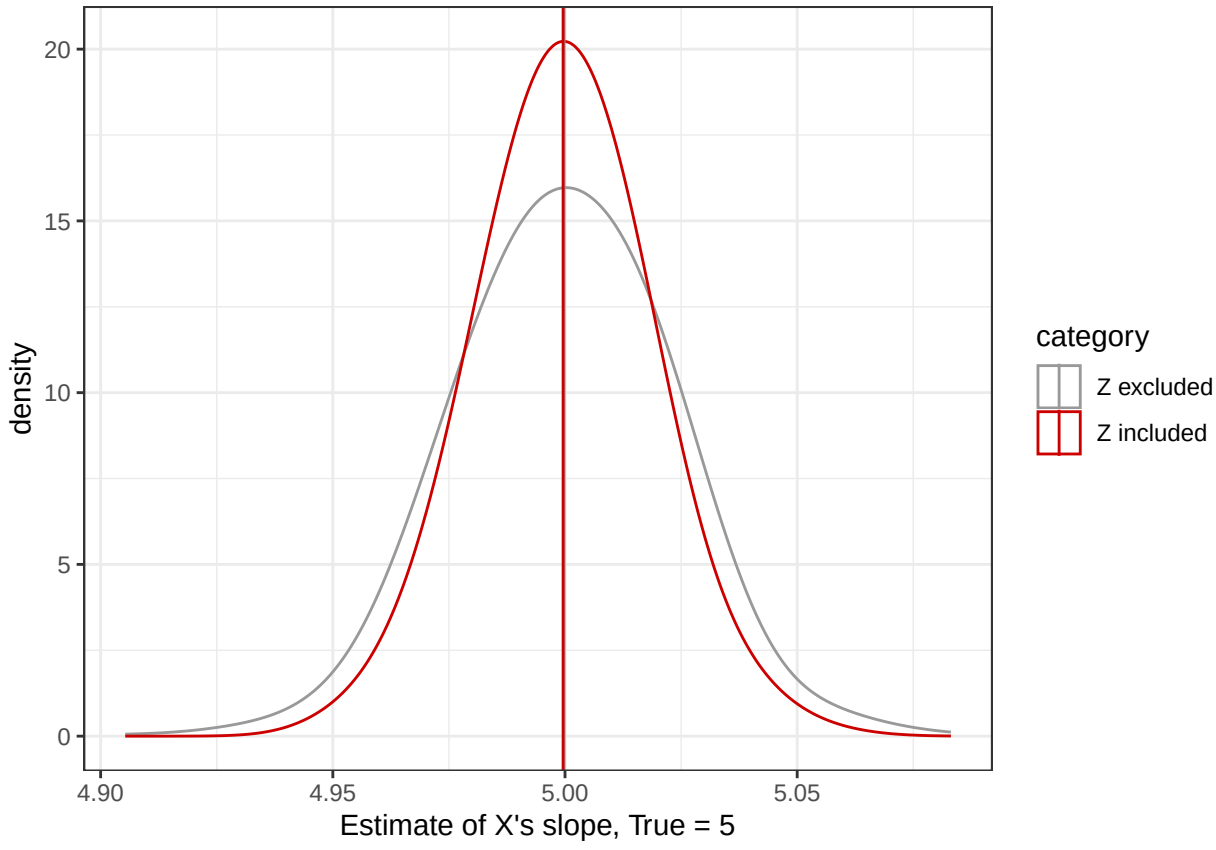
- **Problem Set 2** is due on Friday Oct 17th, 11:59 pm (tonight!)
- Lab next week **(10/24)** will be in the NYU Academic Resource Center, ARC_LL01
- Lab will end at 12:15pm on **11/14** for Prosem TA practice

Part 1: How Does Multivariate Relationships Affect Regression Estimates

- Multiple Causes

```
set.seed(3636)

## empty results
woz <- c()
wz <- c()

for (i in 1:1000){

 ## create hypothetical variables
  X <- runif(500, min=1, max=10)
  Z <- runif(500, min=2, max=5)
  Y <- 10 + 5*X + Z + rnorm(500,0,1)

  ## data
  data <- data.frame(X=X,Y=Y,Z=Z)
```

```r
  ## regression
  lm1 <- lm(Y ~ X, data)
  lm2 <- lm(Y ~ X + Z, data)

  ## extract results
  woz <- c(woz, summary(lm1)$coef[2,1])
  wz <- c(wz, summary(lm2)$coef[2,1])
}

## combine results
results <-
  data.frame(estimate = c(woz,wz),
             category = c(rep("Z excluded",1000),rep("Z included",1000)))

mean <- results %>%
  group_by(category) %>%
  summarize(mean = mean(estimate))

## plot
results %>%
  ggplot(aes(x=estimate,group=category)) +
  geom_density(aes(color=category),bw=0.01) +
  geom_vline(data = mean, aes(xintercept = mean, color = category)) +
  scale_color_manual(values=c("#999999", "red3")) +
  xlab("Estimate of X's slope, True = 5") +
  theme_bw()
```

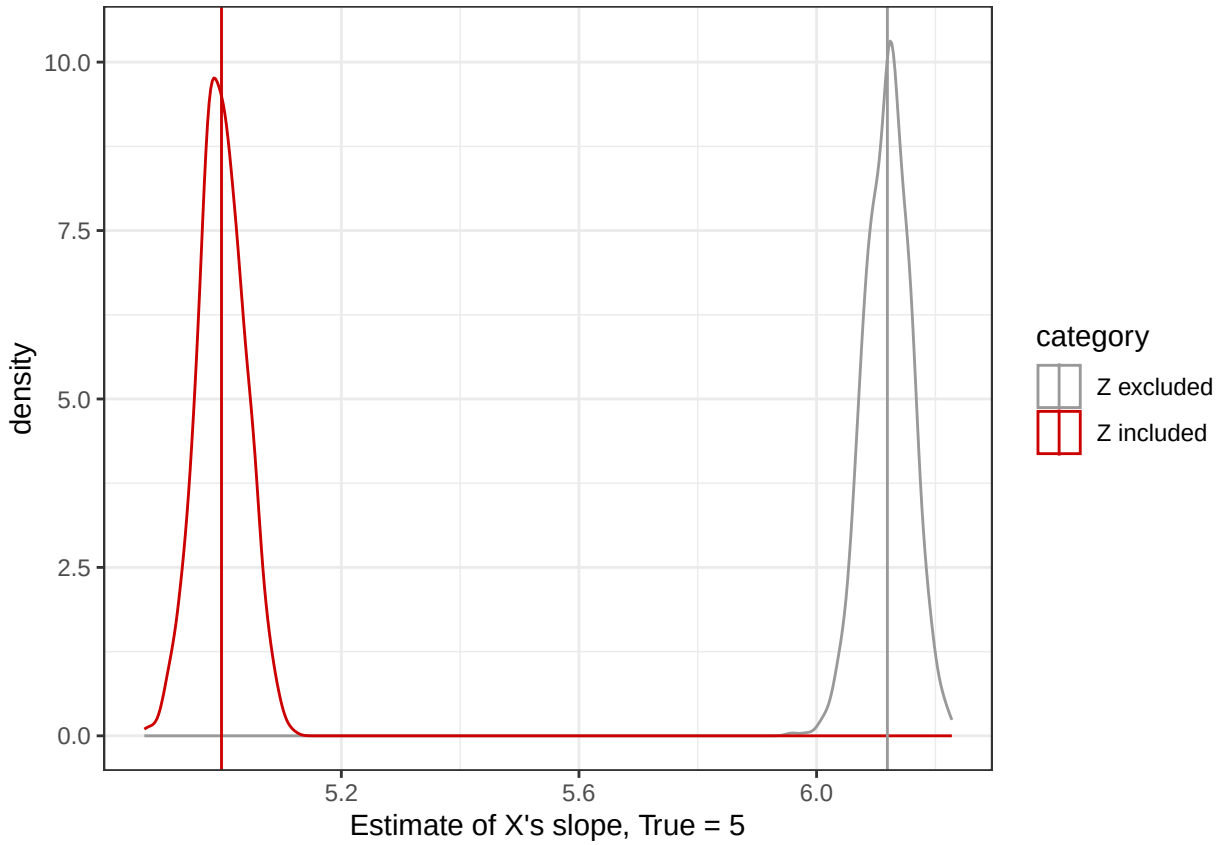


- Confounding

Question: What is the difference between multiple causes and confounding?

In multiple causes, X and Z both only affect Y. In confounding, X affects Y and Z affects both X and Y.

```
set.seed(2023)

## empty results
woz <- c()
wz <- c()

for (i in 1:1000){

 ## create hypothetical variables
  Z <- runif(500, min=1, max=10)
  X <- runif(500, min=1, max=5) + 0.5*Z
  Y <- 10 + 5*X + Z + rnorm(500,0,1)

  ## data
  data <- data.frame(X=X,Y=Y,Z=Z)

  ## regression
  lm1 <- lm(Y ~ X, data)
  lm2 <- lm(Y ~ X + Z, data)

  ## extract results
  woz <- c(woz, summary(lm1)$coef[2,1])
  wz <- c(wz, summary(lm2)$coef[2,1])
}

## combine results
results <-
  data.frame(estimate = c(woz,wz),
             category = c(rep("Z excluded",1000),rep("Z included",1000)))

mean <- results %>%
  group_by(category) %>%
  summarize(mean = mean(estimate))

## plot
results %>%
  ggplot(aes(x=estimate,group=category)) +
  geom_density(aes(color=category),bw=0.01) +
  geom_vline(data = mean, aes(xintercept = mean, color = category)) +
  scale_color_manual(values=c("#999999", "red3")) +
  xlab("Estimate of X's slope, True = 5") +
  theme_bw()
```



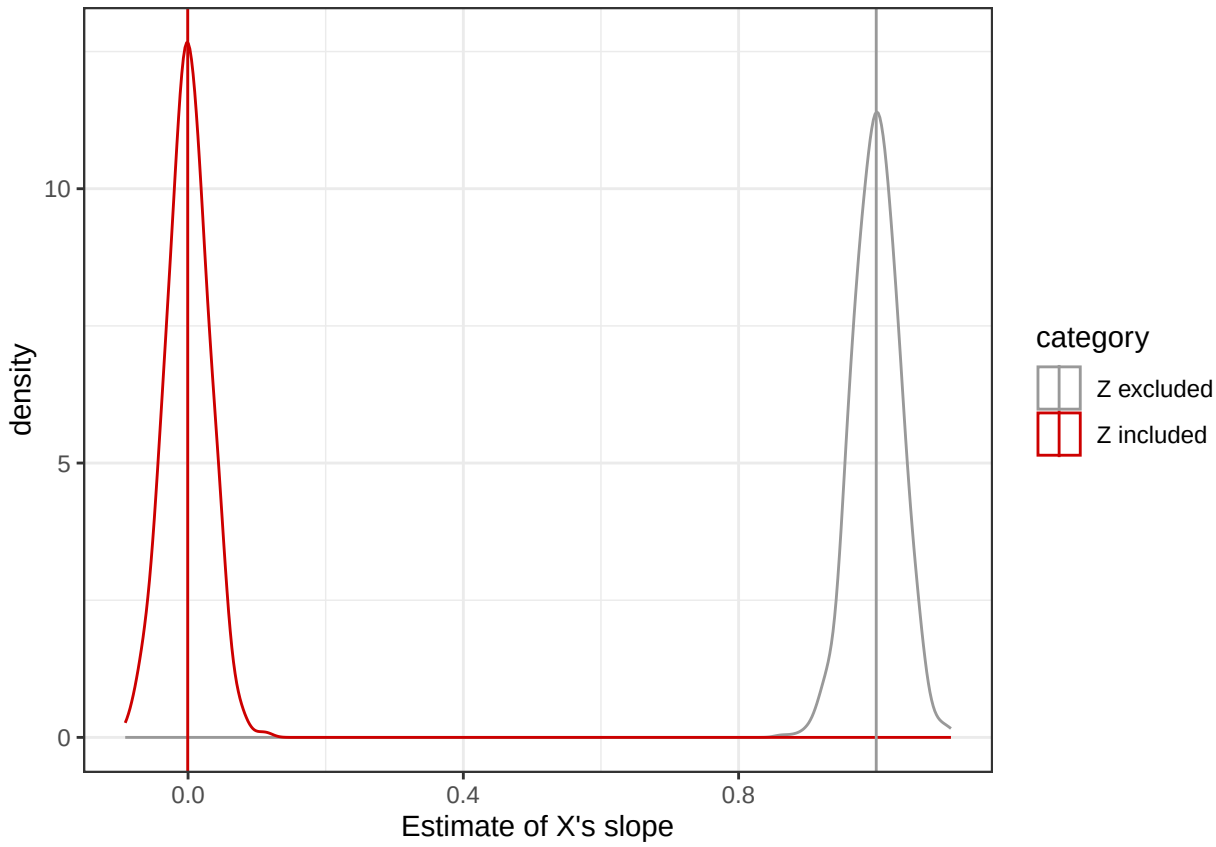
- Mediation

Question: What is mediation?

Mediation is when a variable (X) affects the outcome (Y) through its effect on a third variable (Z).

```
set.seed(3636)

## empty results
woz <- c()
wz <- c()

for (i in 1:1000){

 ## create hypothetical variables
  X <- runif(500, min=1, max=10)
  Z <- runif(500, min=2, max=5) + 0.5*X
  Y <- 2*Z + rnorm(500,0,1)

  ## data
  data <- data.frame(X=X,Y=Y,Z=Z)

  ## regression
  lm1 <- lm(Y ~ X, data)
  lm2 <- lm(Y ~ X + Z, data)

  ## extract results
  woz <- c(woz, summary(lm1)$coef[2,1])
```

```
  wz <- c(wz, summary(lm2)$coef[2,1])
}

## combine results
results <-
  data.frame(estimate = c(woz,wz),
             category = c(rep("Z excluded",1000),rep("Z included",1000)))

mean <- results %>%
  group_by(category) %>%
  summarize(mean = mean(estimate))

## plot
results %>%
  ggplot(aes(x=estimate,group=category)) +
  geom_density(aes(color=category),bw=0.01) +
  geom_vline(data = mean, aes(xintercept = mean, color = category)) +
  scale_color_manual(values=c("#999999", "red3")) +
  xlab("Estimate of X's slope") +
  theme_bw()
```



Question: What is the "true" effect of X on Y in this model, given our known data generating process? Does the regression calculate this "true" slope for X when controlling for Z? Why or why not?

The "true" effect of X on Y can be found by plugging X into the regression formula.

$$Y = 2 * Z + \epsilon$$

$$Y = 2 * (U_z + 0.5 * X) + \epsilon$$
$$Y = 2U_z + 1 * X + \epsilon$$

So the "true" effect of X on Y is 1. We don't get a slope of 1 for X in our regression because we're controlling for Z. Since X is entirely moderated through Z, that is, X only affects Y through its effect on Z, when we control for Z the effect of X on Y is 0.

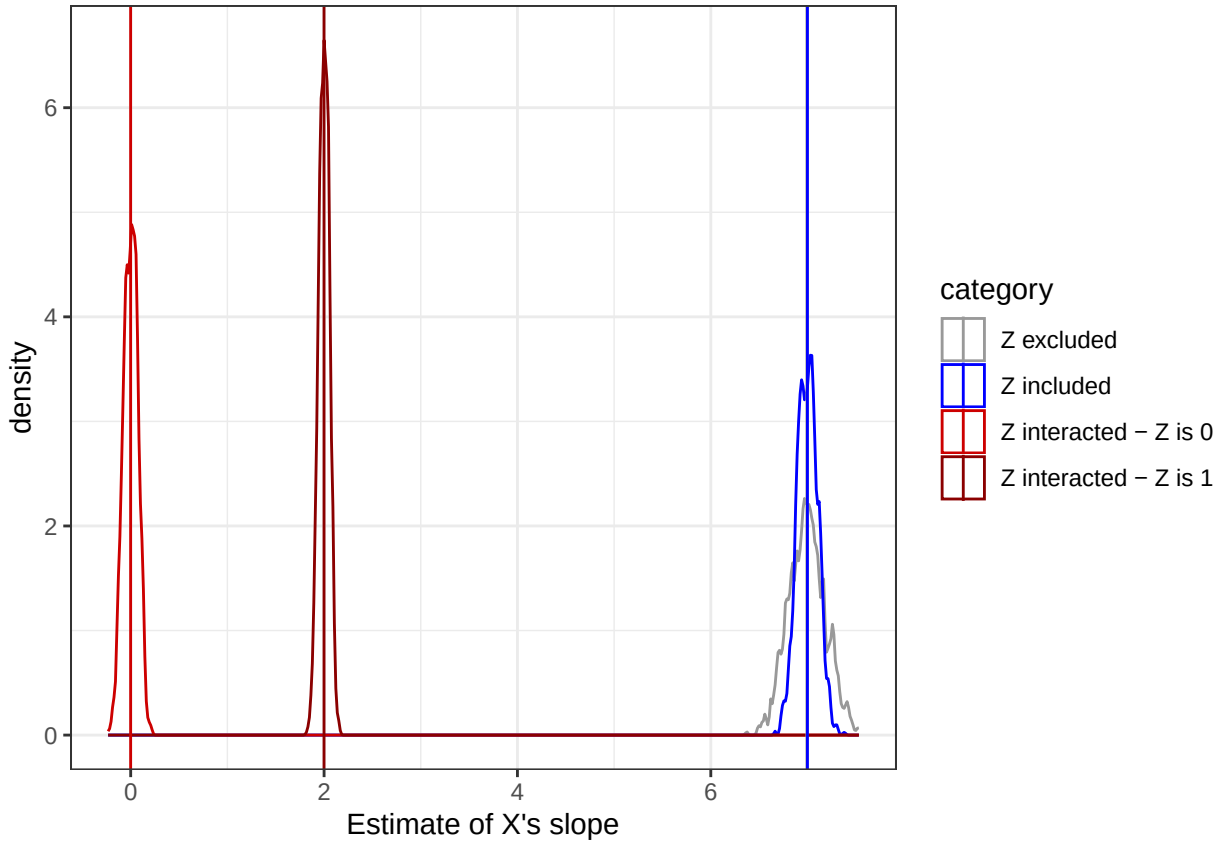
- Moderation

Question: What is moderation? How do we control for it in regressions?

Moderation is when the effect of X on Y is different depending on a third variable (Z). We can control for this in a regression by interacting X and Y.

```
set.seed(2023)

## empty results
woz <- c()
wz <- c()
wzintz0 <- c()
wzintz1 <- c()

for (i in 1:1000){

 ## create hypothetical variables
  X <- runif(500, min=1, max=10)
  Z <- runif(500, min=2, max=5)
  Y <- 2*X*Z + rnorm(500,0,1)

  ## data
  data <- data.frame(X=X,Y=Y,Z=Z)

  ## regression
  lm1 <- lm(Y ~ X, data)
  lm2 <- lm(Y ~ X + Z, data)
  lm3 <- lm(Y ~ X*Z, data) # this is equivalent to lm(Y ~ X + Z + X*Z, data)

  ## extract results
  woz <- c(woz, summary(lm1)$coef[2,1])
  wz <- c(wz, summary(lm2)$coef[2,1])
  wzintz0 <- c(wzintz0, summary(lm3)$coef[2,1])
  wzintz1 <- c(wzintz1, summary(lm3)$coef[2,1] + summary(lm3)$coef[4,1])
}

## combine results
results <-
  data.frame(estimate = c(woz, wz, wzintz0, wzintz1),
             category = c(
               rep("Z excluded", 1000),
               rep("Z included", 1000),
               rep("Z interacted - Z is 0", 1000),
               rep("Z interacted - Z is 1", 1000))
  )
```

```r
mean <- results %>%
  group_by(category) %>%
  summarize(mean = mean(estimate))

## plot
results %>%
  ggplot(aes(x=estimate,group=category)) +
  geom_density(aes(color=category),bw=0.01) +
  geom_vline(data = mean, aes(xintercept = mean, color = category)) +
  scale_color_manual(values=c("#999999", "blue", "red3", "red4")) +
  xlab("Estimate of X's slope") +
  theme_bw()
```



Question: What's a good way to check whether you should be including an interaction term in your regression model?

Theory and prior research can help you predict where there may be interaction. You can also always make a color-coded scatter plot of your three variables and see whether the relationship between X and Y depends on the value of Z.

Part 2: Additional Topics in Regression

1. Heteroskedasticity and Robust Standard Errors

- Heteroskedasticity occurs when the **variance of the error term changes across different values of the explanatory variables**; $Var(\epsilon_i|X) \neq Var(\epsilon_i)$, or, as we see in lecture, we assume that $Var(\epsilon_i|X) = \sigma^2 h(X)$

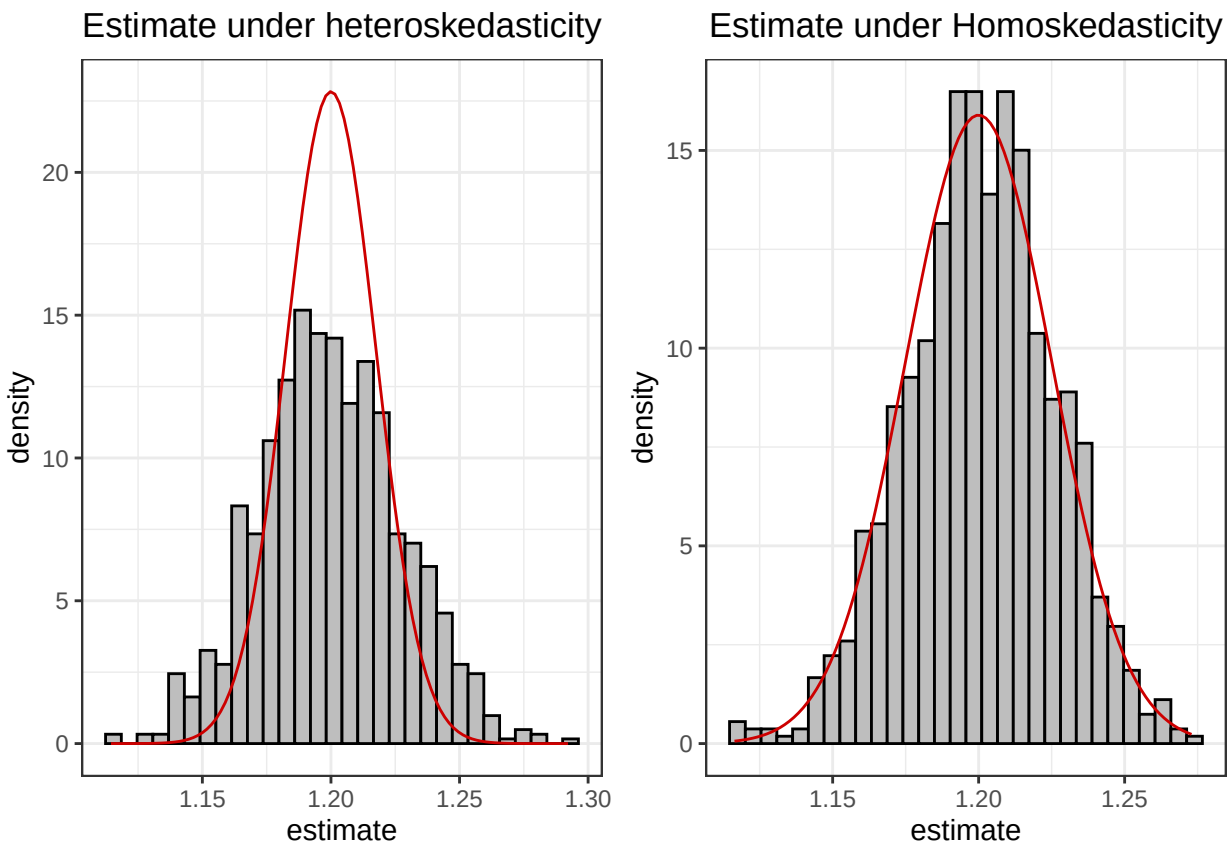
- Heteroskedasticity violates the basic assumption of OLS, in which the variance of the error term should be constant across different values of the explanatory variables.
- *Question*: Will heteroskedasticity make estimates biased and inconsistent?

  `+ It will not make the coefficient estimates biased or inconsistent, but standard errors will be bi`
- In OLS estimation, the standard error of $\hat{\beta}_1$, $se_{\hat{\beta}_1}$ is derived by assuming homoskedasticity. Specifically, given known X and the uncertainty coming from sampling the same X but with different ϵ_i from the population, we assume $Var(y|X) = Var(\epsilon|X) = Var(\epsilon) = \sigma^2$

$$
\begin{aligned}
\hat{\beta} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \\
\text{Var}(\hat{\beta}|\mathbf{X}) &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\ \text{Var}(\mathbf{y}|\mathbf{X})\ \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} \\
&= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\ \sigma^2\mathbf{I}\ \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} \\
&= \sigma^2(\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{X})(\mathbf{X}'\mathbf{X})^{-1} \\
&= \sigma^2(\mathbf{X}'\mathbf{X})^{-1}
\end{aligned}
$$

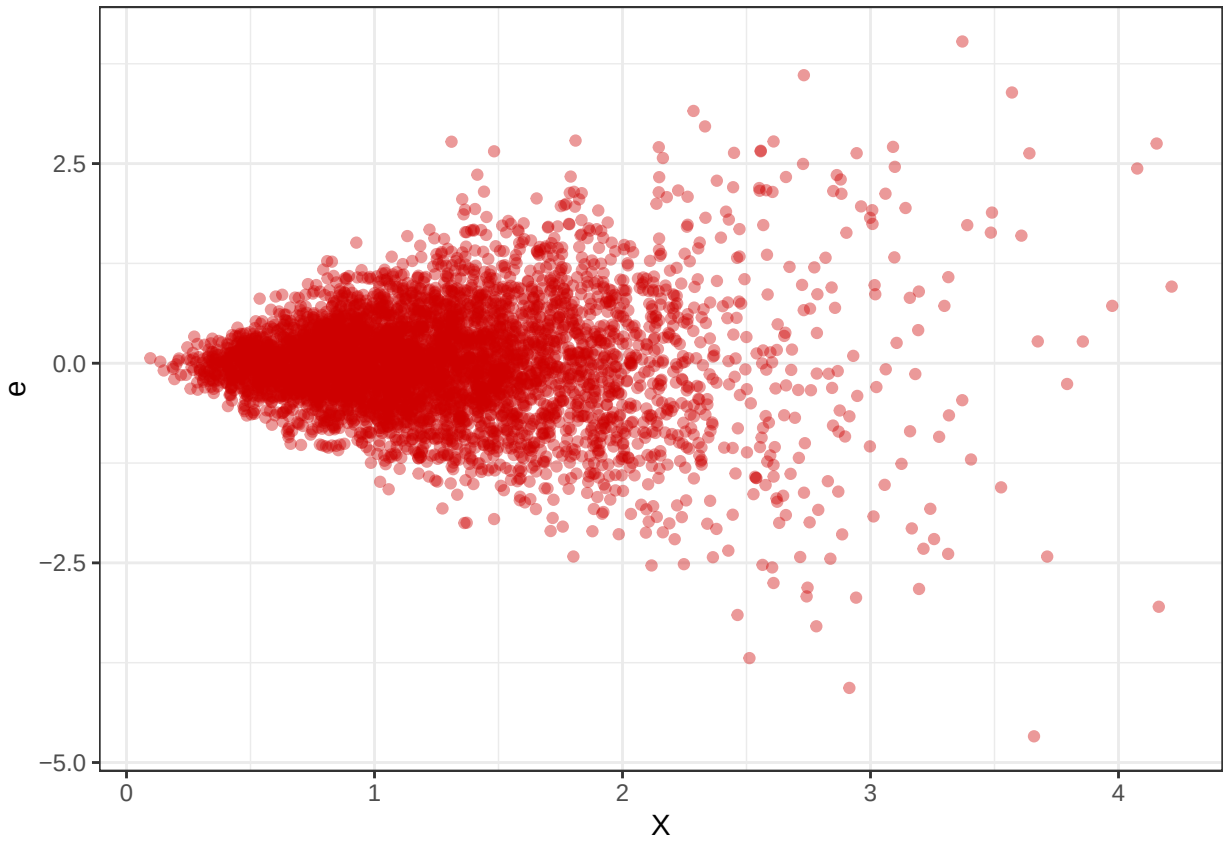
```r
set.seed(2023)

## empty results
homo <- c()
hetero <- c()

## create x
X <- rgamma(5000, 5, 4)

for (i in 1:1000){

 ## create hypothetical data
  homo_Y <- -0.25 + 1.2*X + rnorm(5000,0,1)
  hetero_Y <- -0.25 + 1.2*X + rnorm(5000,0,0.5*X)

  ## data
  homo_data <- data.frame(X=X,homo_Y)
  hetero_data <- data.frame(X=X,hetero_Y)

  ## regression
  lm1 <- lm(homo_Y ~ X, homo_data)
  lm2 <- lm(hetero_Y ~ X, hetero_data)

  ## extract results
  homo <- c(homo, summary(lm1)$coef[2,1])
  hetero <- c(hetero, summary(lm2)$coef[2,1])
}

## combine results
results <-
  data.frame(estimate = c(homo,hetero),
             category = c(rep("homoskedasticity",1000),rep("heteroskedasticity",1000)))

## plot
hetero <-
  results %>%
  filter(category=="heteroskedasticity") %>%
```

```r
  ggplot(aes(x=estimate)) +
  geom_histogram(aes(y=..density..),fill="grey",color="black") +
  stat_function(fun = dnorm,
                args = list(mean = 1.2,
                            sd = summary(lm2)$coef[2,2]),
                color = "red3") +
  ggtitle("Estimate under heteroskedasticity") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

homo <-
  results %>%
  filter(category=="homoskedasticity") %>%
  ggplot(aes(x=estimate)) +
  geom_histogram(aes(y=..density..),fill="grey",color="black") +
  stat_function(fun = dnorm,
                args = list(mean = 1.2,
                            sd = summary(lm1)$coef[2,2]),
                color = "red3") +
  ggtitle("Estimate under Homoskedasticity") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

ggarrange(hetero, homo)
```



- Detecting heteroskedasticity by plotting $Y - \hat{Y}$ over X

```r
data.frame(X = X, e = hetero_Y - predict.lm(lm2,hetero_data)) %>%
  ggplot(aes(x=X,y=e)) +
  geom_point(color="red3",alpha=0.4) +
  theme_bw()
```



- Robust standard error
- If you have a reason to believe that your dataset violates the assumption of homoskedasticity, you can use the packages `sandwich` and `lmtest` to get robust standard errors.

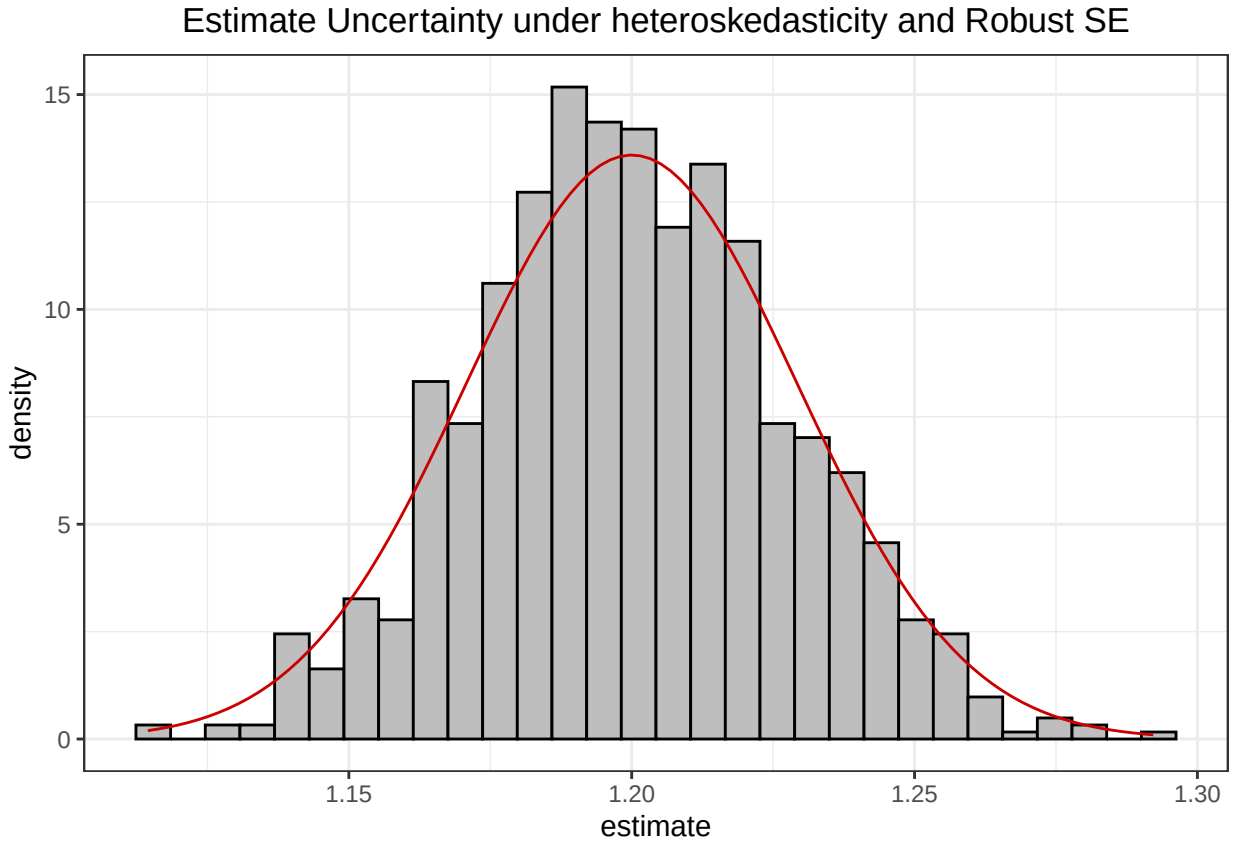
```r
## original SE
summary(lm2)$coef
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -0.2255046 0.02382858 -9.463619 4.461315e-21
## X            1.1793730 0.01747531 67.487975 0.000000e+00
```

```r
## robust SE
coeftest(lm2, vcov = vcovHC(lm2, type="HC1")) # heteroskedasticity-consistent variance-covariance
```

```
##
## t test of coefficients:
##
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept) -0.225505   0.031452 -7.1698 8.611e-13 ***
## X            1.179373   0.029350 40.1837 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
results %>%
  filter(category=="heteroskedasticity") %>%
  ggplot(aes(x=estimate)) +
  geom_histogram(aes(y=..density..),fill="grey",color="black") +
  stat_function(fun = dnorm,
                args = list(mean = 1.2,
                            sd = coeftest(lm2, vcov = vcovHC(lm2, type="HC1"))[2,2]),
                color = "red3") +
  ggtitle("Estimate Uncertainty under heteroskedasticity and Robust SE") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



Question: What is an example of a research question where we might expect heteroskedasticity of errors?

The relationship between household income and among put into savings. Low-income households have a low ceiling on how much they can put into savings, so there is less variation. However as household incomes increase, there can be a much wider range of savings behaviors.

2. Clustering of Errors

- One of the basic OLS assumptions is that the error term is independently distributed across observations. i.e.:

$$\text{Corr}(\epsilon_i, \epsilon_j | X) = 0 \ \forall \ i \neq j$$

- This assumption could be violated when your data have a "nested" structure, or your data is ordered by time and the trend is highly correlated between time unit. In such cases, you should employ other modeling techniques to address correlated errors. For example, you can use multilevel modeling for nested data, and longitudinal data analysis techniques for time-series data.

Part 3: Missing Data

Often, not every observation in your data set will have a value for every variable. There are three different types of missingness in data:

1. *Missing completely at random (MCAR):* data missing at random, that is, whether data is missing does not depend on any observed or unobserved variable.
2. *Missing at random (MAR):* the missingness depends on a known variable in the observed data
3. *Missing not at random (MNAR):* the missingness depends on an unknown or unmeasured variable that does not appear in the observed data

How can you tell what type of missingness you have in your data? Let's create two sample datasets to practice with. We'll use the same earnings dataset from last class, but we'll create one version with MCAR missingness and one with MAR missingness.

```
# set seed
set.seed(3636)

# importing datasets to use in example
earnings_df <- read.csv('data/earnings_df.csv') %>%

  # recoding missing age from 9999 to NA
  mutate(
    age = ifelse(age > 200, NA_integer_, age)
  )

# creating earnings dataset with MCAR missingness
mcar_earnings <- earnings_df %>%

  # randomly choosing 40 observations and deleting their education data
  sample_n(size = 100, replace = FALSE) %>%
  mutate(
    edu = NA_integer_,
    deleted_edu = 1
  ) %>%

  # re-merging deleted data observations with the rest of the dataset
  right_join(., earnings_df,
             by = c("unique_id", "earn", "sex", "race", "age")) %>%

  # creating a edu column with random missingness
  mutate(
    edu = ifelse(deleted_edu %>% is.na() == FALSE, edu.x, edu.y)
  ) %>%
  select(everything(), -edu.x, -edu.y, -deleted_edu)

# set seed
set.seed(66)

# creating earnings dataset with MAR missingness
mar_earnings <- earnings_df %>%

  # filtering to women
  filter(sex == "female") %>%
```

```
  # randomly choosing 40 observations and deleting their education data
  sample_n(size = 100, replace = FALSE) %>%
  mutate(
    edu = NA_integer_,
    deleted_edu = 1
  ) %>%

  # re-merging deleted data observations with the rest of the dataset
  right_join(., earnings_df,
             by = c("unique_id", "earn", "sex", "race", "age")) %>%

  # creating a edu column with random missingness
  mutate(
    edu = ifelse(deleted_edu %>% is.na() == FALSE, edu.x, edu.y)
  ) %>%
  select(everything(), -edu.x, -edu.y, -deleted_edu)
```

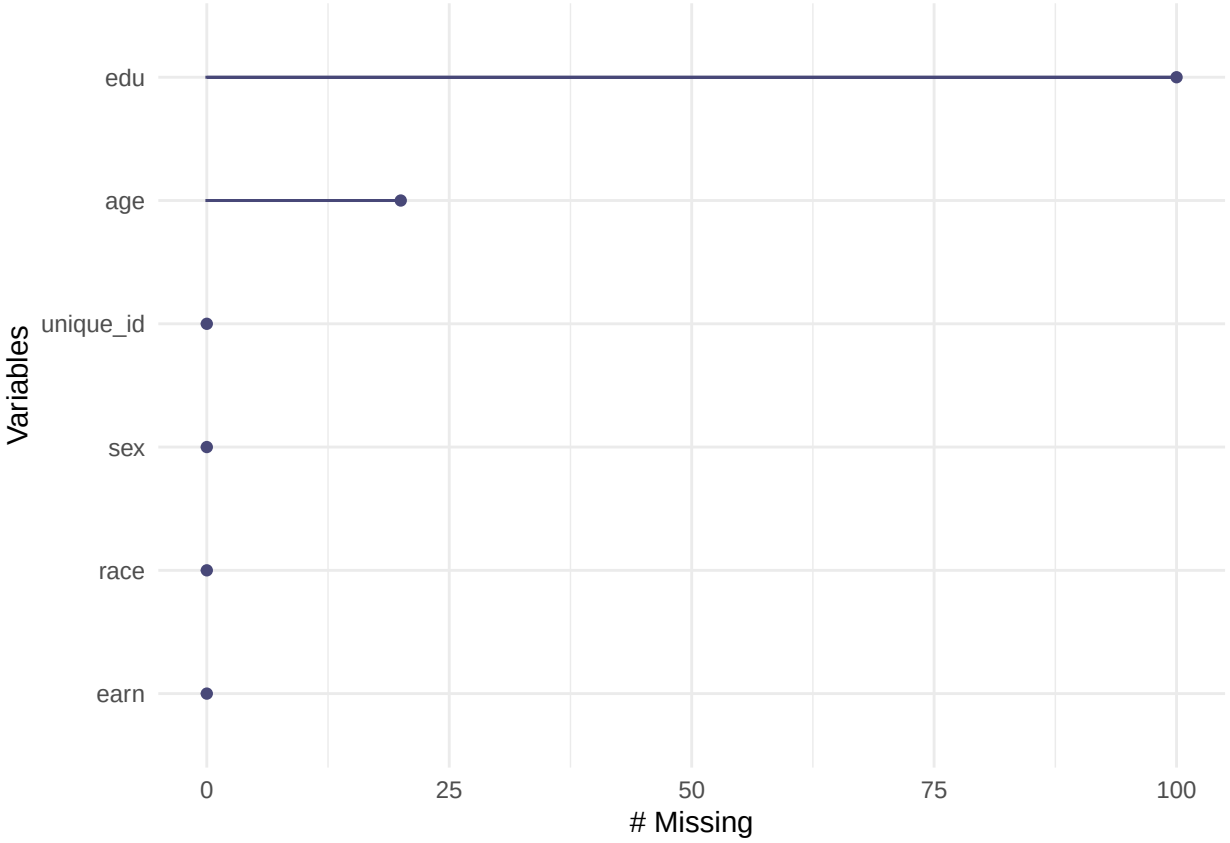
Using the `summary()` function, we can see whether there are NA values in any of our variables. Remember that data sources often code missingness differently, so you also want to be sure to check the codebook for your data to determine what value missing data has.

```
summary(mar_earnings)
```

```
##    unique_id          earn             sex                race
##  Min.   :10001   Min.   :  8.831   Length:1000        Length:1000
##  1st Qu.:10251   1st Qu.: 36.966   Class :character   Class :character
##  Median :10500   Median : 46.915   Mode  :character   Mode  :character
##  Mean   :10500   Mean   : 49.717
##  3rd Qu.:10750   3rd Qu.: 60.812
##  Max.   :11000   Max.   :123.554
##
##       age              edu
##  Min.   :21.00   Min.   : 0.000
##  1st Qu.:32.00   1st Qu.: 4.000
##  Median :44.00   Median : 6.000
##  Mean   :43.26   Mean   : 6.062
##  3rd Qu.:54.25   3rd Qu.: 8.000
##  Max.   :65.00   Max.   :15.000
##  NA's   :20      NA's   :100
```

We can also look at missingness visually with the `naniar` package in R. This package has a bunch of helpful commands for working with missing data. Feel free to explore more here.

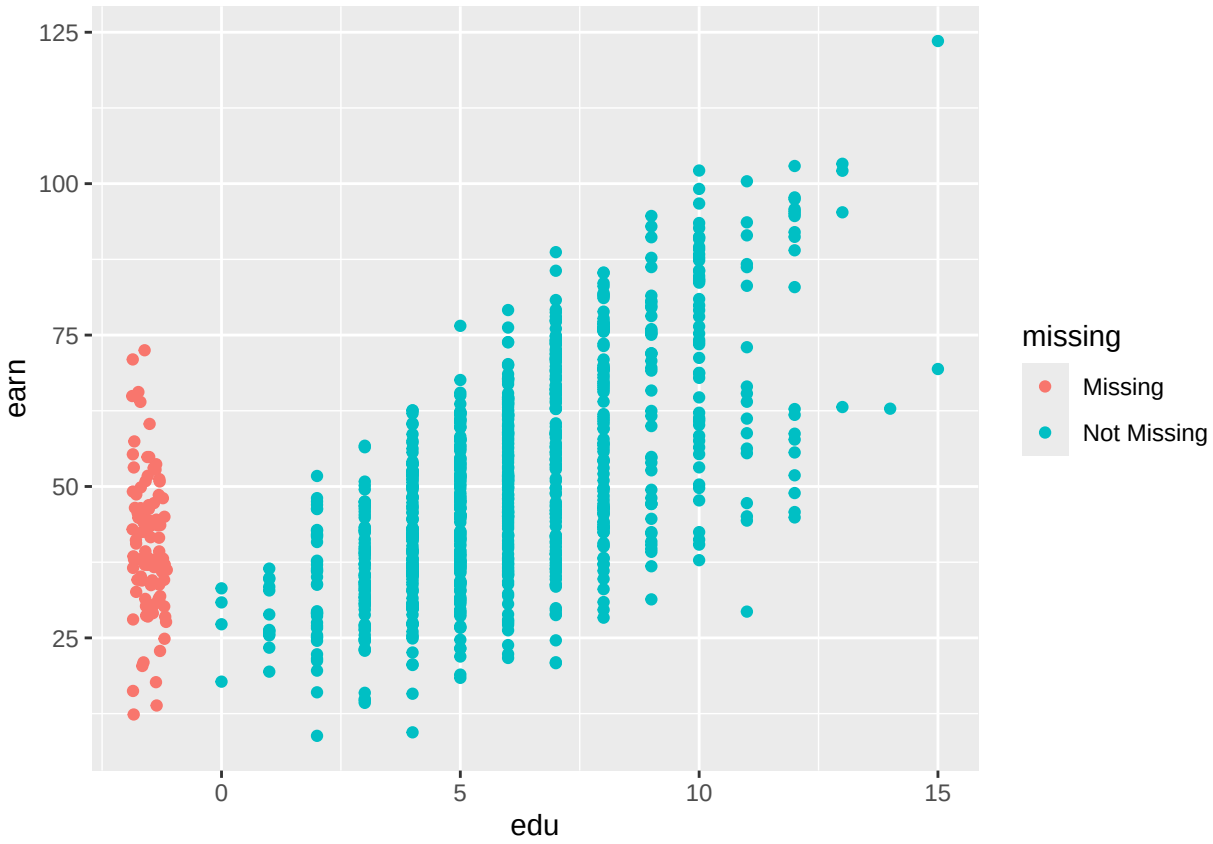
```
# plot missing data
gg_miss_var(mar_earnings)
```



It seems like we have missingness in two variables, age and education. Let's focus on education for now.

`ggplot()` automatically excludes missing values and gives you a warning message. We can create a visualization of the missing data points using `geom_miss_point()` from `naniar`.

```
# plotting missing variables in MAR dataset
mar_earnings %>%
ggplot(
       aes(x = edu,
           y = earn)) +
  geom_miss_point()
```



In the MAR dataset, it looks like the missingness in education might be clustered around lower income levels. Note that this doesn't necessarily tell us what is causing the missingness, but it does suggest that it is not MCAR. To help you determine the source of missingness, you can compare missingness rates within levels of different variables in your dataset. Prior research and theory is also very helpful here. Researchers have done a lot of work on who is more likely to skip various questions on surveys.

The `naniar()` package also has a formal test for MCAR missingness based on Little's test called `mcar_test()`. Note that the null hypothesis for this test is that the data is MCAR, so a p-value $< .05$ means that we cannot reject the hypothesis that the data is MCAR.

```
# running Little's test on MCAR data
mcar_test(mcar_earnings)
```

```
## # A tibble: 1 x 4
##   statistic    df p.value missing.patterns
##       <dbl> <dbl>   <dbl>            <int>
## 1      8.54    14   0.859                4
```

```
# running Little's test on MAR data
mcar_test(mar_earnings)
```

```
## # A tibble: 1 x 4
##   statistic    df p.value missing.patterns
##       <dbl> <dbl>   <dbl>            <int>
## 1      125.    14       0                4
```

Depending on what kind of missingness you have in your data, there are different ways of dealing with missingness. Let's go through some strategies and how you implement each one in R.

Option 1: Complete case estimator

The simplest option is just to exclude all observations with missing data in key variables. This is often called "listwise deletion" or a "complete case estimator." While this is the simplest method for dealing with missingness, it will result in a biased estimate unless you have MCAR.

To demonstrate this, we can compare the results of listwise deletion in our MCAR dataset and MAR dataset. Unless you tell it otherwise, the `lm()` function will automatically remove any observations with missingness in any of the included variables.

```
# calculating the relationship between education and earnings in the full dataset
complete <- earnings_df %>%
  lm(earn ~ edu,
     data = .)

# education vs. earnings in MCAR dataset
mcar <- mcar_earnings %>%
  lm(earn ~ edu,
     data = .)

# education vs. earnings in MAR dataset
mar <- mar_earnings %>%
  lm(earn ~ edu,
     data = .)
```

```
# results
stargazer(
  complete,
  mcar,
  mar,
  type = "latex",
  object.names = TRUE,
  model.numbers = FALSE
)
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Thu, Oct 16, 2025 - 16:21:48

As you can see, when we do listwise deletion with MCAR data, the coefficient on education is similar to the "true" value we see in the complete data set. However when we do this on MAR data, the coefficient is farther from the "true" coefficient".

Option 2: Imputation by guessing

If you do not have MCAR data, another option is to impute missing values based on similar observations. The simplest way to do this is by imputing the mean value, or the mean value of the group the variable is in.

```
# imputing education values based on mean education in that age group
earnings_imp_guess <- mar_earnings %>%
  group_by(age) %>%
  summarize(
    mean_edu = mean(edu, na.rm = TRUE)
  ) %>% right_join(., mar_earnings,
                   by = "age") %>%
  mutate(
    edu = ifelse(is.na(edu), mean_edu, edu)
  )
```

Table 1:

	Dependent variable:		
	earn		
	complete	mcar	mar
edu	4.358***	4.436***	4.622***
	(0.171)	(0.181)	(0.180)
Constant	23.204***	22.675***	22.732***
	(1.126)	(1.190)	(1.181)
Observations	1,000	900	900
R^2	0.395	0.401	0.424
Adjusted R^2	0.394	0.401	0.423
Residual Std. Error	13.710 (df = 998)	13.732 (df = 898)	13.575 (df = 898)
F Statistic	650.957*** (df = 1; 998)	602.360*** (df = 1; 898)	660.274*** (df = 1; 898)
Note:			*p<0.1; **p<0.05; ***p<0.01

```
# running regression with this imputed dataset
marimp <- earnings_imp_guess %>%
  lm(earn ~ edu,
     data = .)
```

```
# results
stargazer(
  complete,
  mar,
  marimp,
  type = "latex",
  object.names = TRUE,
  model.numbers = FALSE
)
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Thu, Oct 16, 2025 - 16:21:48

You can do this based on as many variables as you want. For example, you might replace each missing education observation with the mean education of people who have the same age, gender, and race as the missing observation.

Option 3: Probabilistic imputation

Probabilistic imputation or multiple imputation is when missing data points are imputed multiple times from a distribution of possible values, creating multiple different data sets, and then the results of these data sets are pooled together. This allows us to account for the uncertainty around the true value of the missing data and calculate standard errors.

The general process is:

1. Impute m values for each missing value creating m completed datasets
    - m is usually between 5 and 10

Table 2:

	Dependent variable:		
		earn	
	complete	mar	marimp
edu	4.358***	4.622***	4.609***
	(0.171)	(0.180)	(0.182)
Constant	23.204***	22.732***	21.773***
	(1.126)	(1.181)	(1.185)
Observations	1,000	900	1,000
R^2	0.395	0.424	0.392
Adjusted R^2	0.394	0.423	0.391
Residual Std. Error	13.710 (df = 998)	13.575 (df = 898)	13.746 (df = 998)
F Statistic	650.957*** (df = 1; 998)	660.274*** (df = 1; 898)	642.393*** (df = 1; 998)
Note:			*p<0.1; **p<0.05; ***p<0.01

2. Analyze each of these m completed datasets separately
3. Combine the m results
    - Usually by taking the average and adjusting the SE

This may sound like a lot of work, but luckily R has packages that make it very easy! There are many different packages which use slightly different methods, such as `Amelia`, `mice`, `missForest`, and `mi`. We'll use `Amelia` for our demo.

Importantly, `Amelia` assumes that the data follows a multivariate normal distribution. If your data doesn't follow a multivariate normal distribution, there are other types of multiple imputation you can use (the `mice` package for example uses conditional multiple imputation).

Any variables that are included in your analysis model should be included in your imputation model. In fact, it's often helpful to include even factors that won't be included in your analysis model. We can use the `amelia()` command to create multiple imputations:

```
# running multiple imputation
a.out <- mar_earnings %>%
  amelia(., # telling amelia to use every variable in our dataset for imputation
         m = 5, # number of times to impute/number of datasets to create
         noms = c("sex", "race") # we have to specifically tell Amelia which variables are nominal or o
         )
```

```
## -- Imputation 1 --
##
##   1  2  3
##
## -- Imputation 2 --
##
##   1  2  3
##
## -- Imputation 3 --
##
##   1  2  3
```

```
##
## -- Imputation 4 --
##
##   1  2  3
##
## -- Imputation 5 --
##
##   1  2  3
```

```
# we can save and view each individual imputed dataset with this command
#write.amelia(obj = a.out, file.stem = "amelia_outdata")
```

We then can run regressions with the imputed data sets in one step using the `with()` command. Our result will be a list of the output of the `lm()` command applied to each dataset.

```
# running our regression model with each data set
mult_imp <- with(
  a.out,
  lm(earn ~ edu)
)

# viewing two of the regression results
mult_imp[1:2]
```

```
## [[1]]
##
## Call:
## lm(formula = earn ~ edu)
##
## Coefficients:
## (Intercept)          edu
##      22.158        4.515
##
##
## [[2]]
##
## Call:
## lm(formula = earn ~ edu)
##
## Coefficients:
## (Intercept)          edu
##      22.162        4.539
```

We can combine these with the `mi.combine()` function. To learn more about the rules this function uses to combine results, see King et al. 2001.

```
# combining results
mar_mi <- mi.combine(mult_imp, conf.int = TRUE)

# showing results
mar_mi
```

```
## # A tibble: 2 x 10
##   term  estimate std.error statistic   p.value     df       r miss.info conf.low
##   <chr>    <dbl>     <dbl>     <dbl>     <dbl>  <dbl>   <dbl>     <dbl>    <dbl>
## 1 (Int~    22.2      1.14       19.4 1.10e- 83 5.11e5 0.00280   0.00280    24.4
## 2 edu       4.52     0.175      25.9 7.33e-146 3.20e4 0.0113    0.0112      4.86
```

```
## # i 1 more variable: conf.high <dbl>
```

```
# comparing multiple imputation results to other methods
stargazer(
  complete,
  mar,
  marimp,
  type = "latex",
  object.names = TRUE,
  model.numbers = FALSE
)
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Thu, Oct 16, 2025 - 16:21:48

Table 3:

	Dependent variable:		
	earn		
	complete	mar	marimp
edu	4.358***	4.622***	4.609***
	(0.171)	(0.180)	(0.182)
Constant	23.204***	22.732***	21.773***
	(1.126)	(1.181)	(1.185)
Observations	1,000	900	1,000
R^2	0.395	0.424	0.392
Adjusted R^2	0.394	0.423	0.391
Residual Std. Error	13.710 (df = 998)	13.575 (df = 898)	13.746 (df = 998)
F Statistic	650.957*** (df = 1; 998)	660.274*** (df = 1; 898)	642.393*** (df = 1; 998)
Note:			*p<0.1; **p<0.05; ***p<0.01

Exercise (10 minutes)

Let's say we want to investigate the relationship between ozone levels and temperature. How should we handle the missingness in the `airquality()` data set? Load in this dataset, try to figure out what kind of missingness it has, and decide what analysis strategy is best for handling this missingness.

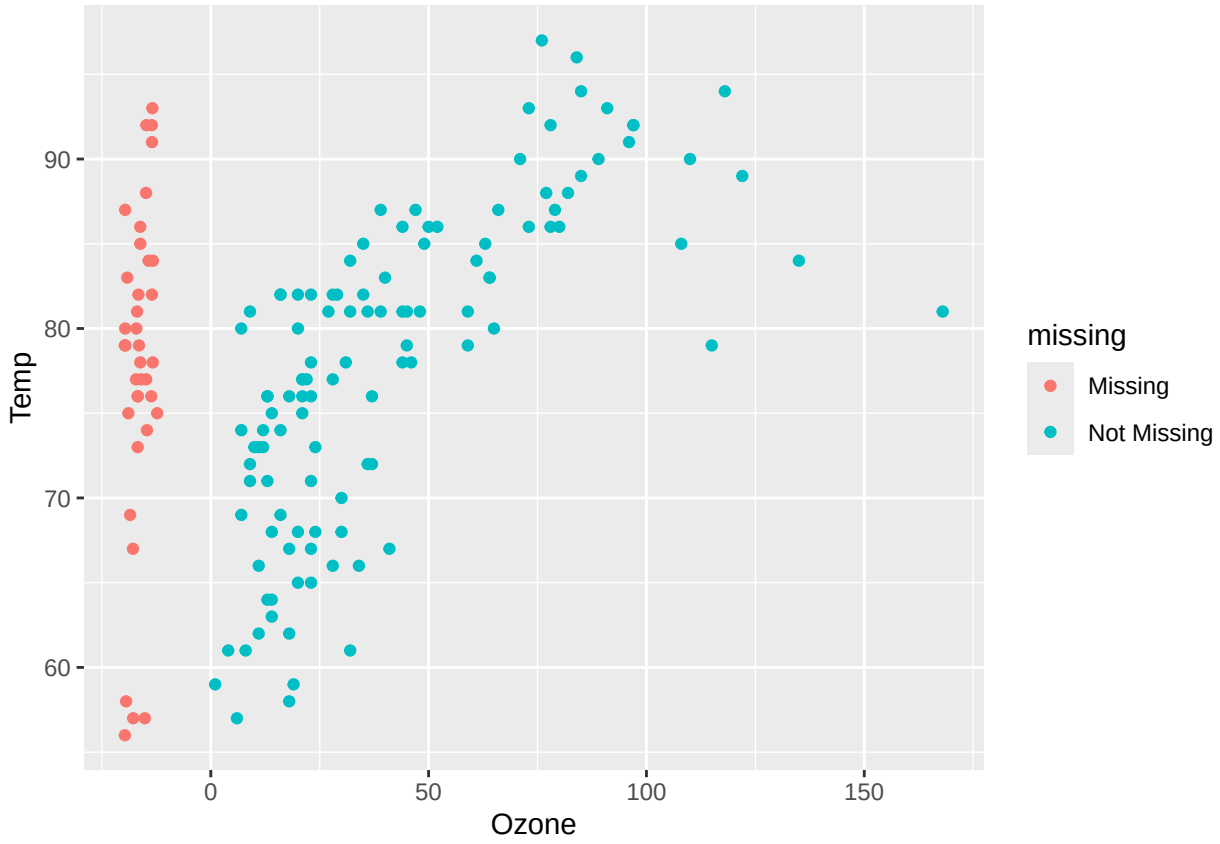
The `airquality` dataset comes preloaded in R, so you can call it by name without loading it. For more information on the dataset, remember you can use the `?airquality` command.

```
summary(airquality)
```

```
##      Ozone           Solar.R           Wind             Temp
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00
##  Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88
##  3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00
##  NA's   :37       NA's   :7
##      Month            Day
```

```
##  Min.   :5.000   Min.   : 1.0
##  1st Qu.:6.000   1st Qu.: 8.0
##  Median :7.000   Median :16.0
##  Mean   :6.993   Mean   :15.8
##  3rd Qu.:8.000   3rd Qu.:23.0
##  Max.   :9.000   Max.   :31.0
##
```

```
# plotting missing variables
airquality %>%
ggplot(
       aes(x = Ozone,
           y = Temp)) +
  geom_miss_point()
```



Higher temperatures seem more likely to be missing. So we likely have MAR data.

```
# running Little's test
mcar_test(airquality)
```

```
## # A tibble: 1 x 4
##   statistic    df p.value missing.patterns
##       <dbl> <dbl>   <dbl>            <int>
## 1      35.1    14 0.00142                4
```

Again, suggests we don't have MCAR data. I'll try multiple imputation!

```
# running multiple imputation
a.out.aq <- airquality %>%
  amelia(., # telling amelia to use every variable in our dataset for imputation
```

```
        m = 5, # number of times to impute/number of datasets to create
        )
```

```
## -- Imputation 1 --
##
##   1  2  3  4  5  6  7  8
##
## -- Imputation 2 --
##
##   1  2  3  4  5
##
## -- Imputation 3 --
##
##   1  2  3  4  5  6  7
##
## -- Imputation 4 --
##
##   1  2  3  4  5  6  7
##
## -- Imputation 5 --
##
##   1  2  3  4  5  6
```

```
# we can save and view each individual imputed dataset with this command
#write.amelia(obj = a.out, file.stem = "amelia_outdata")
```

```
# running a regression model with each data set
aq_imp_mult <- with(
  a.out.aq,
  lm(Temp ~ Ozone)
)

# combining results
aq_imp_results <- mi.combine(aq_imp_mult, conf.int = TRUE)

# showing results
aq_imp_results
```

```
## # A tibble: 2 x 10
##   term    estimate std.error statistic   p.value    df      r miss.info conf.low
##   <chr>      <dbl>     <dbl>     <dbl>     <dbl> <dbl>  <dbl>     <dbl>    <dbl>
## 1 (Inter~   69.4      0.955       72.7 1.58e-279  535. 0.0946    0.0898   71.3
## 2 Ozone      0.198    0.0174      11.3 8.82e- 29 1618. 0.0523    0.0509    0.232
## # i 1 more variable: conf.high <dbl>
```

Part 4: Replication Project Tips

1 Samples

- For the 1970 sample, use the 1% state sample
- For the 1990 sample, use the 1% metro sample
  - The reason is that it is the only 1970 sample that provides non-missing `METRO` information
- For the 2010 sample, use the single-year ACS sample, not 3- or 5-year pooled sample
- Some of you may encounter memory issues.
  - A. Try restarting your PC. This will release some used memories.

  - B. Try restricting the sample first before doing any operations. Specifically, you should drop Rs who are younger than 25 and older than 59 (confirm if this is the case in the paper); keep only non-Hispanic White and Black Rs; Rs who are in the workforce (variable `LABFORCE`) and have valid occupation (variable `OCC1990`); and who are economically active (`INCWAGE>0`; pp.1046)
- If you read pp.1046 carefully, you will notice that Rs with the top and bottom earning percentile are excluded
  - You can create percentiles using `quantile(ma$WEEKEARN, seq(0.01,1,0.01))`, suppose your dataframe is `ma`, and the weekly earning variable is `WEEKEARN`
- You should also drop Rs who have missing values for any of the used variables
  - You may consider using the function `complete.cases()` to enable this feature. Reference

2 Variables

- Use `BPL` rather than `NATIVITY`
- The latter has no valid values for most samples
- Use `HISPAN` to exclude Hispanic Whites and Hispanic Blacks
- Use `CLASSWKR` to determine whether R is in a public sector or not
  - You should look at CLASSWKRD, which gives detailed classification of `CLASSWKR`
- Use CPI99 to adjust inflation for `INCWAGE`
- The main dependent variable is the logged form of **weekly earnings**
  - You will need `WKSWORK1` and `WKSWORK2` to measure the number of weeks worked last year. `WKSWORK1` always gives the best continuous estimate, but when `WKSWORK1` is not available, you should turn to `WKSWORK2`
  - `WKSWORK2` is coded in intervals. For example, `WKSWORK2` = 1 means R worked for 1-13 weeks. Use the middle number as a proxy, that is, 7 weeks.
- Recoding weekly working hours has a similar process. You will need `UHRSWORK` and `HRSWORK2` to construct the measure. `UHRSWORK` always gives the best continuous estimate, but when `UHRSWORK` is not available, you should turn to `HRSWORK2` using the middle number as a proxy for the interval estimate.
- To estimate potential years of experience, the formula is given by `LMEXP = AGE - EDUYEAR - 6`
  - `EDUYEAR` needs to be estimated
  - Codes for this process are available in the `code` folder

3 Duncan's Dissimilarity Index

- In Table A1a and A1b, you will notice that there is a dissimilarity index. This is a very commonly used measure of occupational segregation.
  - Check Martin-Caughey (2022) on within-occupation variation and gender segregation using job titles and verbatim texts in GSS that describe jobs
  - The standard Duncan's Dissimilarity/Segregation Index is given by:

$$D = \frac{1}{2}\sum_{i=1}^{n}\left|\frac{a_i}{A} - \frac{b_i}{B}\right|$$

where a_i and b_i is the number of White and Black workers in occupation group i. A and B represents the total number of White and Black workers. * Instead of using hundreds of `OCC1990` categories, you will use 2-digit aggregated categories of `OCC1990` + Use the `merge()` function

4 General Instructions

- It is totally okay if you cannot get exactly the same numbers! I also couldn't.
- But they should be close enough. If they deviate a lot, you need to explain your speculations why the numbers differ this much.
- The total number of observation N may give you some hints (e.g., you did not restrict your sample as much as the original paper).

Part 4: Kitagawa-Oaxaca-Blinder (KOB) Decomposition

- KOB Decomposition is a common practice to decompose group-wise difference into 1. endowment difference, 2. coefficient (slope) difference, and 3. unexplained portion.
  - The group-wise difference is measured at the mean level, e.g. *mean* weakly income
  - For example, gender difference in mean weekly income can be decomposed into:
    - 1. the gender difference in *mean* education, years of experience, etc. Women once lagged behind men in education, but now has surpassed men in mean education
    - 2. the gender difference in *returns* to education, years of experience, etc. Women typically have lower returns to education, but the trend is recently reversed, especially at the lower end (e.g. women with HS or some college are less penalized than men).
    - 3. the portion that cannot be explained by above observed characteristics
- KOB starts from OLS regression. Now we focus on racial (White-Black) gap in mean weakly income
  - OLS by race, White: $Y_{iw} = \alpha_w + \sum_{k=1}^{\ell} \beta_{kw} X_{ikw} + \epsilon_{iw}$
  - OLS by race, Black: $Y_{ib} = \alpha_b + \sum_{k=1}^{\ell} \beta_{kb} X_{ikb} + \epsilon_{ib}$
- The mean value of Y_i, $\bar{Y}_w$ or $\bar{Y}_b$, according to the properties of OLS, is:

$$\bar{Y}_w = \alpha_w + \sum_{k=1}^{\ell} \beta_{kw} \bar{X}_{kw} \bar{Y}_b = \alpha_b + \sum_{k=1}^{\ell} \beta_{kb} \bar{X}_{kb}$$

- The mean racial pay gap:

$$\begin{aligned} \bar{Y}_w - \bar{Y}_b &= \alpha_w + \sum_{k=1}^{\ell} \beta_{kw} \bar{X}_{kw} - \alpha_b - \sum_{k=1}^{\ell} \beta_{kb} \bar{X}_{kb} \\ &= \underbrace{\sum_{k=1}^{\ell} (\bar{X}_{kw} - \bar{X}_{kb}) \beta_{kw}}_{\text{endowment difference}} + \left[\underbrace{\sum_{k=1}^{\ell} \bar{X}_{kb} (\beta_{kw} - \beta_{kb})}_{\text{coefficient difference}} + \underbrace{(\alpha_w - \alpha_b)}_{\text{intercept difference}} \right] \end{aligned}$$

- When we focus on one of the explanatory features, such as education, we may visualize the decomposition as follows:

```
knitr::include_graphics("graph/KOB.png")
```

- Although KOB Decomposition is clean and intuitive, it also has some clear shortcomings. One of the key problems of KOB is that the decomposition results are sensitive to the reference group. E.g., when analyzing the education component in the racial pay gap, whether HS or graduate education is used as the reference group will change the results.
  - For a detailed demonstration of this problem, see Jones and Kelley (1984)
  - Recent work used normalized coefficients to address the issue. Check this well-cited SMR paper: Kim (2010)
  - It also suffers from the typical problem of OLS. Without a causal inference design, the "discrimination" part of the model may be due to unmeasured characteristics (e.g., years of education vs field of studies).

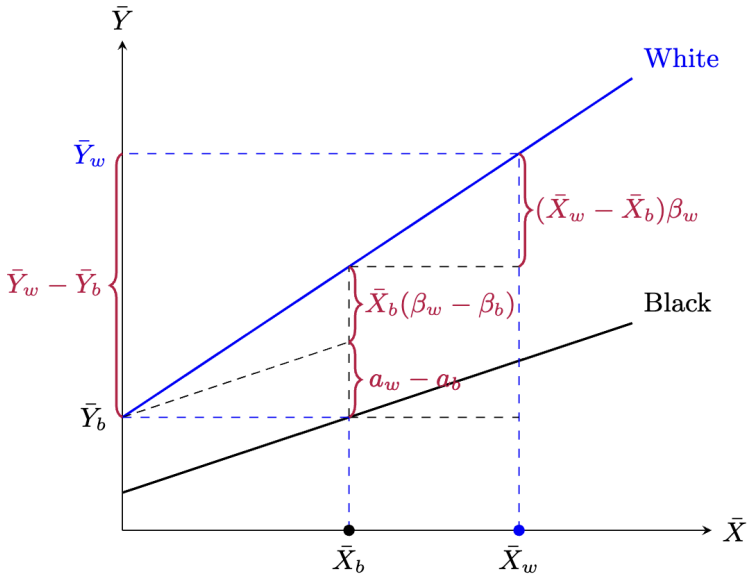


Figure 1: Graphical Demonstration of KOB Decomposition