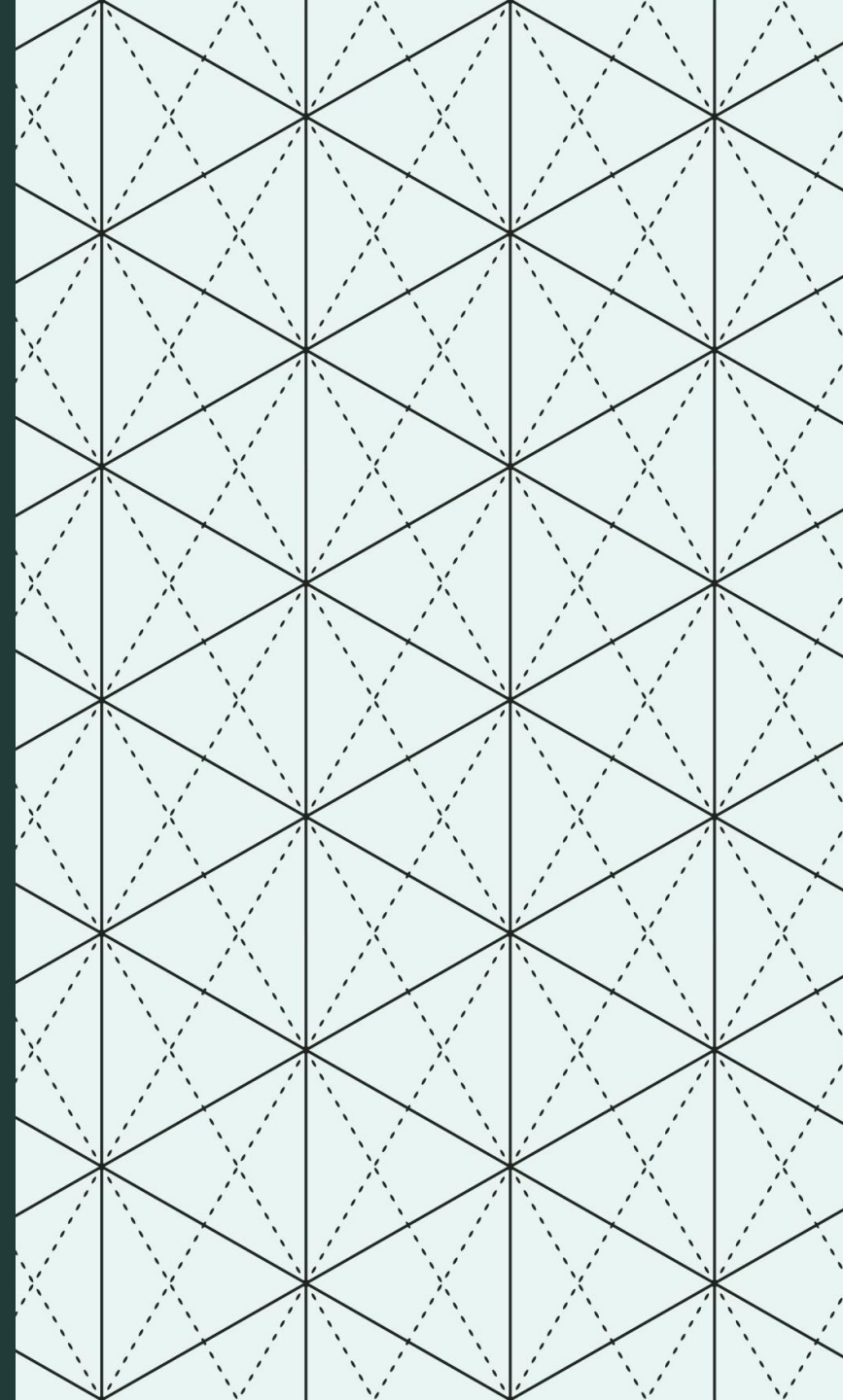


---

# WELCOME TO PSY 653 LAB!

MODULE 10: META-ANALYSES & MISSING DATA

\*Thanks to Gemma Wallace for her help with these slides





# OBJECTIVES

- Part 1: Meta-Analyses
- Part 2: Handling Missing data

# PART 1: META- ANALYSES



# Nine Steps to Meta Analyses

We recommend in general the following nine steps of meta analysis. These nine steps are in general applicable to all meta-analyses.

1. Frame a question (based on a theory)
2. Run a search (on Pubmed/Medline, Google Scholar, other sources)
3. Read the abstract and title of the individual papers.
4. Abstract information from the selected set of final articles.
5. Determine the quality of the information in these articles. This is done using a judgment of their internal validity but also using the GRADE criteria
6. Determine the extent to which these articles are heterogeneous
7. Estimate the summary effect size in the form of Odds Ratio and using both fixed and random effects models and construct a forest plot
8. Determine the extent to which these articles have publication bias and run a funnel plot
9. Conduct subgroup analyses and meta regression to test if there are subsets of research that capture the summary effects

# Nine Steps to Meta Analyses

We recommend in general the following nine steps of meta analysis. These nine steps are in general applicable to all meta-analyses.

1. Frame a question (based on a theory)
2. Run a search (on Pubmed/Medline, Google Scholar, other sources)
3. Read the abstract and title of the individual papers.
4. Abstract information from the selected set of final articles.
5. Determine the quality of the information in these articles. This is done using a judgment of their internal validity but also using the GRADE criteria
6. Determine the extent to which these articles are heterogeneous
7. Estimate the summary effect size in the form of Odds Ratio and using both fixed and random effects models and construct a forest plot
8. Determine the extent to which these articles have publication bias and run a funnel plot
9. Conduct subgroup analyses and meta regression to test if there are subsets of research that capture the summary effects

We are here





---

# CREATE A NEW R-PROJECT AND R-NOTEBOOK!

Download the "studies.csv" file  
from Canvas and save it into your  
R-project file

```
1 --- Load Libraries
2 title: "Lab 1: Meta-analyses"
3 output: html_notebook
4 ---
5
6
7 # Load libraries
8 ```{r}
9 library(tidyverse)
10 install.packages("meta")
11 library(meta)
12 ```
13
```

```
14 # Read in data
15 {r}
16 studies <- read_csv(file="studies.csv")
17
```

Parsed with column specification:

```
cols(
  r = col_double(),
  n = col_double(),
  Study = col_character()
)
```



## A LOOK AT OUR DATASET

	<b>r</b>	<b>n</b>	<b>Study</b>
<b>1</b>	0.32	210	S1
<b>2</b>	0.21	160	S2
<b>3</b>	0.35	90	S3
<b>4</b>	0.27	410	S4
<b>5</b>	0.29	100	S5
<b>6</b>	0.44	60	S6
<b>-</b>	0.47	70	S7

- **r** = Study effect size (correlations)
- **n** = Study sample size
- **Study** = Name of study (i.e. Yetz et al., 2003)

---

Run the meta analysis with metacor()

```
# Compare correlations across studies
```

```
```{r}
```

```
MA <- metacor(cor = r, n = n, studlab= Study, data=studies)
```

```
MA|
```

```
```
```

## Run the meta analysis with metacor()

```
# Compare correlations across studies
```

```
{r}  
MA <- metacor(cor = r, n = n, studlab= Study, data=studies)  
MA
```

- **MA** = object
- **metacor()** = function to compare correlations across studies (From meta package)
- "**cor =**" = column that correlations are placed (r)
- "**n =**" = column where sample sizes are placed (n)
- "**studlab =**" = column where study names are placed (Study)
- "**data =**" = dataset where information is stored (studies)

# OUTPUT!

|     | COR    | 95%-CI            | %W(fixed) | %W(random) |
|-----|--------|-------------------|-----------|------------|
| S1  | 0.3200 | [ 0.1930; 0.4365] | 6.5       | 6.5        |
| S2  | 0.2100 | [ 0.0567; 0.3536] | 4.9       | 4.9        |
| S3  | 0.3500 | [ 0.1541; 0.5194] | 2.7       | 2.7        |
| S4  | 0.2700 | [ 0.1778; 0.3575] | 12.7      | 12.7       |
| S5  | 0.2900 | [ 0.0992; 0.4602] | 3.0       | 3.0        |
| S6  | 0.4400 | [ 0.2095; 0.6242] | 1.8       | 1.8        |
| S7  | 0.1700 | [-0.0642; 0.3864] | 2.2       | 2.2        |
| S8  | 0.2900 | [ 0.1104; 0.4513] | 3.4       | 3.4        |
| S9  | 0.1800 | [ 0.0169; 0.3337] | 4.4       | 4.4        |
| S10 | 0.2900 | [ 0.1846; 0.3888] | 9.6       | 9.6        |
| S11 | 0.2000 | [ 0.0696; 0.3237] | 6.8       | 6.8        |
| S12 | 0.2500 | [ 0.0918; 0.3959] | 4.5       | 4.5        |
| S13 | 0.3100 | [ 0.1887; 0.4220] | 7.1       | 7.1        |
| S14 | 0.2200 | [ 0.0111; 0.4105] | 2.7       | 2.7        |
| S15 | 0.4000 | [ 0.2148; 0.5574] | 2.8       | 2.8        |
| S16 | 0.3100 | [ 0.1800; 0.4294] | 6.2       | 6.2        |
| S17 | 0.4000 | [ 0.2507; 0.5307] | 4.3       | 4.3        |
| S18 | 0.3300 | [ 0.2070; 0.4427] | 6.8       | 6.8        |
| S19 | 0.1600 | [-0.0125; 0.3233] | 4.0       | 4.0        |
| S20 | 0.2600 | [ 0.0855; 0.4190] | 3.7       | 3.7        |

Number of studies combined: k = 20

|                      | COR    | 95%-CI           | z     | p-value  |
|----------------------|--------|------------------|-------|----------|
| Fixed effect model   | 0.2809 | [0.2487; 0.3124] | 16.34 | < 0.0001 |
| Random effects model | 0.2809 | [0.2487; 0.3124] | 16.34 | < 0.0001 |

Quantifying heterogeneity:

$\tau^2 = 0$  [0.0000; 0.0064];  $\tau = 0$  [0.0000; 0.0800];  
 $I^2 = 0.0\%$  [0.0%; 37.5%];  $H = 1.00$  [1.00; 1.27]

Test of heterogeneity:

| Q     | d.f. | p-value |
|-------|------|---------|
| 15.84 | 19   | 0.6682  |

Details on meta-analytical method:

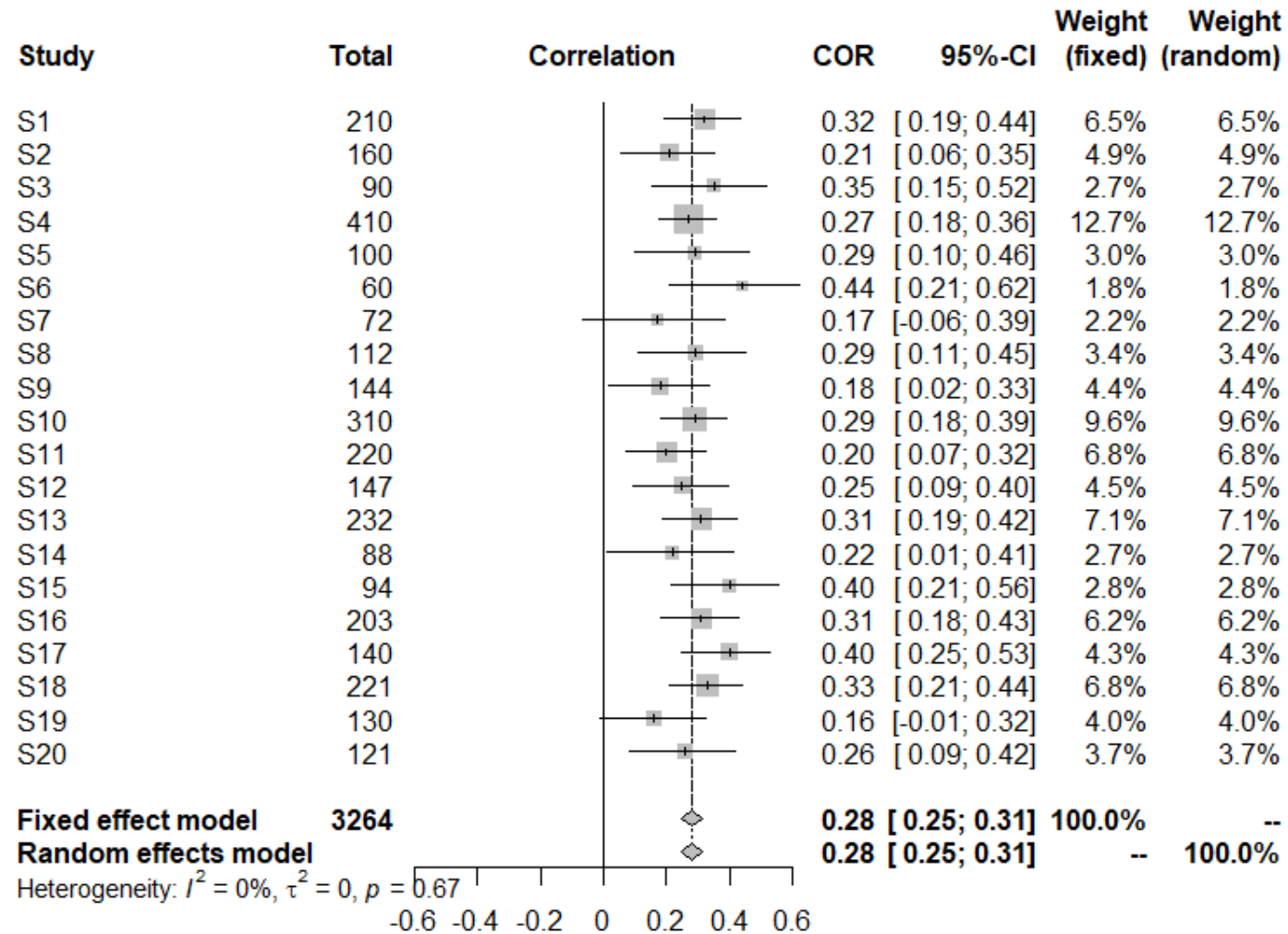
- Inverse variance method
- DerSimonian-Laird estimator for  $\tau^2$
- Jackson method for confidence interval of  $\tau^2$  and  $\tau$
- Fisher's z transformation of correlations

- Weight each study encompasses for both fixed & random effect models.
- Weighted average correlation for both fixed & random effects models.
- **I<sup>2</sup>**: Percentage of total variability due to *true heterogeneity across studies*.  
[Hopefully lower than 30%]
- **Cochran's Q**: The sum of squared deviations of ES estimates from the overall mean value. A significant value is an indicator of heterogeneity [Not good].

## FOREST PLOT WITH forest()

```
28  {r, fig.height = 8}  
29  forest(MA)  
30  
```

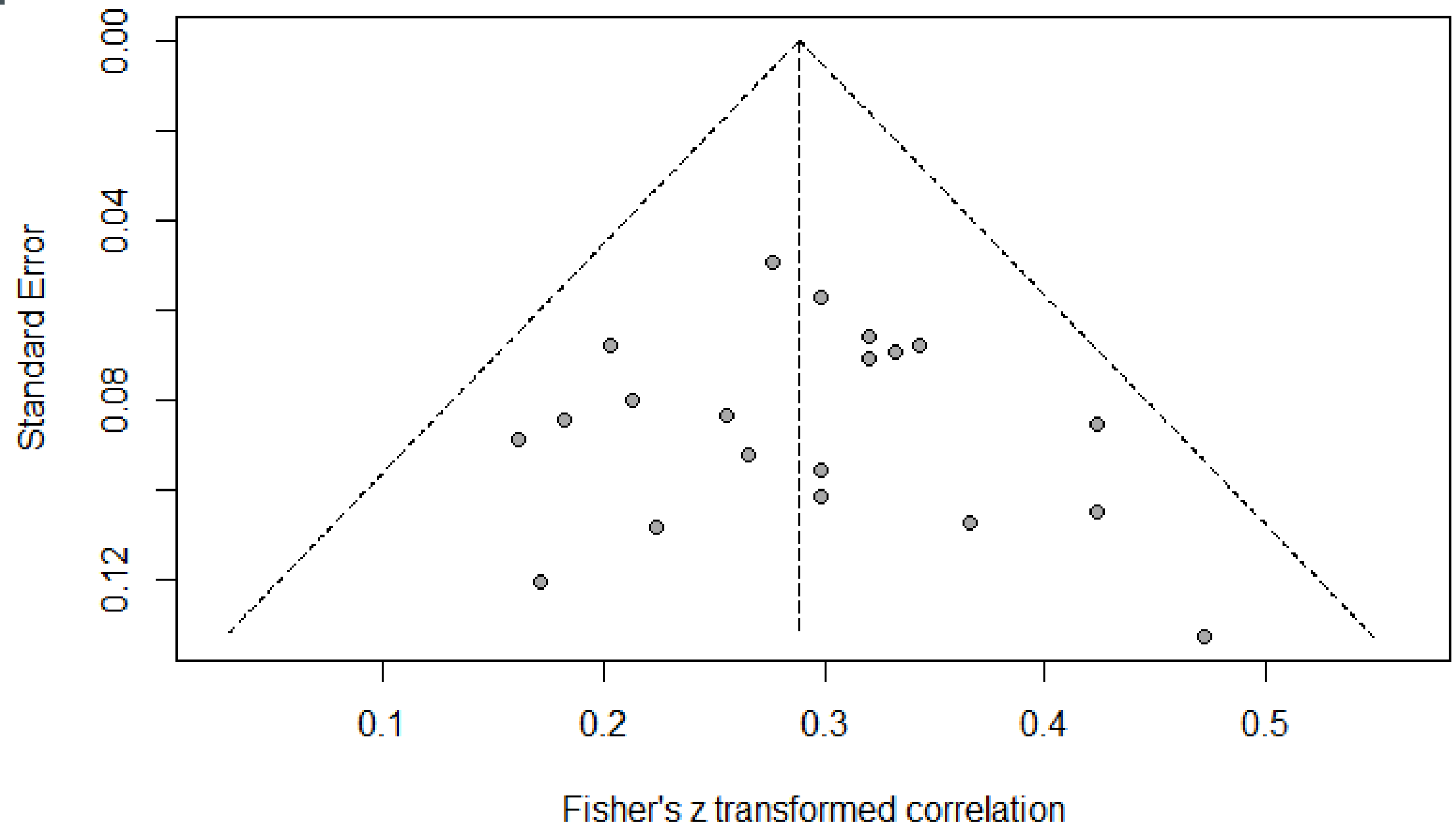
# FOREST PLOT WITH forest()



## FUNNEL PLOT WITH `funnel()`

```
31  
32 {r}  
33 funnel(MA)  
34
```

FUNNEL PLOT WITH funnel()





---

## A NOTE ON THE META PACKAGE

- The meta package allows you to run many more meta analyses rather than comparing correlations across studies.
- You can compare different effect sizes such as Odds ratios (metabin) or Cohen's d values (metacont) and more!
- To read more on the meta package, view the vignette here:
  - <https://cran.r-project.org/web/packages/meta/meta.pdf>

# PART 2: HANDLING MISSING DATA



# MISSING DATA

Missing data are observations that should be part of your data but aren't

| ID | Y  | X1 | X2 | X3 |
|----|----|----|----|----|
| 1  | 32 | 0  | 6  | 5  |
| 2  | 25 | 1  | 5  | 3  |
| 3  | 40 | 1  | 7  | 6  |
| 4  | ?  | ?  | ?  | ?  |
| 5  | 5  | 1  | 4  | 2  |
| 6  | 27 | 0  | ?  | 7  |

---

# METHODS FOR HANDLING MISSING DATA

- There are several!
- R uses listwise deletion by default
  - Can lose power and/or bias results
- Multiple Imputation by Chained Equations (MICE)
  - Imputation = substituting missing data with estimated values

---

# MICE LAB DEMO

- 1) Run a simple linear regression using pairwise deletion, the default in R
- 2) Impute dataset's missing values using the mice package
- 3) Run a simple linear regression in the imputed data
- 4) Compare model estimates across missing data techniques



---

# CREATE A NEW R-PROJECT AND R-NOTEBOOK!

Download the "mice\_data1.csv"  
file from Canvas and save it into  
your R-project file

# Load Libraries

```
1 ---
2 title: "PSY 653 Module 1: Missing Data"
3 subtitle: "Jan 29, 2020"
4 output:
5   html_document:
6     df_print: paged
7 ---
8
9
10
11 ## Load Libraries
12 ```{r,message=FALSE}
13 library(tidyverse)
14 library(mice)
15 library(olsrr)
16 ```
17
```

## READ IN DATA

```
## Read in data
```

```
`{r,message=FALSE}
```

```
mice_data1 <- read_csv("mice_data1.csv")
```

```
`
```

This dataset has 2 simulated variables: X1 and X2

X1 has some missing values

| X1 | X2 |
|----|----|
| 1  | 3  |
| 2  | 0  |
| 3  | 3  |
| 1  | 0  |
| 2  | 0  |
| 3  | 4  |
| NA | 0  |
| 1  | 0  |
| 2  | 1  |
| 3  | 0  |
| 1  | 2  |
| 2  | 0  |
| 3  | 2  |
| NA | 0  |
| 1  | 2  |
| 2  | 2  |
| 3  | 0  |
| 1  | 2  |
| 2  | 0  |
| 3  | 0  |
| NA | 1  |
| 1  | 0  |



Use a Simple Linear Regression to regress X1 on X2

```
## Simple Linear regression model X1 ~ X2
```

```
Using pairwise deletion for missing data by default
```

```
```{r}
```

```
mod1 <- lm(X1 ~ X2, data = mice_data1)
```

```
ols_regress(mod1)
```

```
```
```

# SIMPLE LINEAR REGRESSION OUTPUT WITH PAIRWISE DELETION

## Model Summary

|                |        |           |        |
|----------------|--------|-----------|--------|
| R              | 0.026  | RMSE      | 0.714  |
| R-Squared      | 0.001  | Coef. Var | 39.353 |
| Adj. R-Squared | -0.002 | MSE       | 0.510  |
| Pred R-Squared | -0.009 | MAE       | 0.592  |

RMSE: Root Mean Square Error

MSE: Mean Square Error

MAE: Mean Absolute Error

## ANOVA

|            | Sum of Squares | DF  | Mean Square | F    | Sig.   |
|------------|----------------|-----|-------------|------|--------|
| Regression | 0.137          | 1   | 0.137       | 0.27 | 0.6039 |
| Residual   | 204.939        | 402 | 0.510       |      |        |
| Total      | 205.077        | 403 |             |      |        |

## Parameter Estimates

| model       | Beta   | Std. Error | Std. Beta | t      | Sig   | lower  | upper |
|-------------|--------|------------|-----------|--------|-------|--------|-------|
| (Intercept) | 1.828  | 0.044      |           | 41.152 | 0.000 | 1.741  | 1.916 |
| X2          | -0.011 | 0.022      | -0.026    | -0.519 | 0.604 | -0.055 | 0.032 |

We interpret this output as usual

In write-up, would specify "missing data were handled using pairwise deletion"

# IMPUTE THE DATA WITH MICE

```
## Impute the dataset 5 times (using mice)
```{r}
imputed_data <- mice(mice_data1, m=5, maxit = 50, method = 'pmm', seed = 500)
```
```

- × **mice\_data1** = name of the dataset you are imputing
- × **m** = # of imputations (# of imputed *versions* of the dataset you will create)
- × **maxit** = number of iterations for each imputation (default is 5, generally do more)
- × **method = pmm** = "Predictive Mean Matching"
- × **seed** = specifying a # will allow you to get the same results each time

# WHAT THE IMPUTATION PROCESS LOOK LIKE

MICE uses all of the other variables to predict each missing value

| iter | imp | variable |
|------|-----|----------|
| 1    | 1   | X1       |
| 1    | 2   | X1       |
| 1    | 3   | X1       |
| 1    | 4   | X1       |
| 1    | 5   | X1       |
| 2    | 1   | X1       |
| 2    | 2   | X1       |
| 2    | 3   | X1       |
| 2    | 4   | X1       |
| 2    | 5   | X1       |
| 3    | 1   | X1       |
| 3    | 2   | X1       |
| 3    | 3   | X1       |

## RUN THE SAME SIMPLE LINEAR REGRESSION OF X1 ON X2, BUT THIS TIME USE THE IMPUTED DATASET

```
## Regress X1 on X2 on imputed dataset using the "with" function
```{r, results= hide}
mod.imp <- with(imputed_data, exp= lm(X1 ~ X2))
summary(mod.imp)
```
```

- × **mod.imp** = model name
- × **with()** = tells R to run the analysis in all imputations of the data
- × **exp** = an expression with a formula object
- × **lm(X1 ~ X2)** = the model you want to run. In this case, a simple linear regression
- × **summary()** = use to view model output

# SIMPLE LINEAR REGRESSION OUTPUT WITH MICE FOR ALL IMPUTATIONS

```
mod.imp <- with(imputed_data, exp= lm(X1 ~ X2))
summary(mod.imp)
...
```

| term<br><chr> | estimate<br><dbl> | std.error<br><dbl> | statistic<br><dbl> | p.value<br><dbl> |
|---------------|-------------------|--------------------|--------------------|------------------|
| (Intercept)   | 1.843998789       | 0.04306806         | 42.81592551        | 1.230265e-158    |
| X2            | -0.019184199      | 0.02092467         | -0.91682191        | 3.597405e-01     |
| (Intercept)   | 1.829037385       | 0.04236086         | 43.17752920        | 6.251121e-160    |
| X2            | -0.001437869      | 0.02058108         | -0.06986363        | 9.443341e-01     |
| (Intercept)   | 1.835000757       | 0.04178940         | 43.91067814        | 1.559009e-162    |
| X2            | -0.019259876      | 0.02030343         | -0.94860191        | 3.433462e-01     |
| (Intercept)   | 1.832034206       | 0.04266410         | 42.94088903        | 4.385727e-159    |
| X2            | -0.020546390      | 0.02072841         | -0.99121896        | 3.221259e-01     |
| (Intercept)   | 1.791168458       | 0.04269351         | 41.95411954        | 1.589717e-155    |
| X2            | 0.012751627       | 0.02074270         | 0.61475263         | 5.390373e-01     |

1-10 of 10 rows

|                 |                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|-----------------|
| Imputation<br>1 | Imputation<br>2 | Imputation<br>3 | Imputation<br>4 | Imputation<br>5 |
|-----------------|-----------------|-----------------|-----------------|-----------------|

# POOL MODEL ESTIMATES FROM ALL THE IMPUTATIONS

```
## Pool model estimates across imputed versions of the dataset
```

```
```{r}  
combined_imp <- pool(mod.imp)  
summary(combined_imp)  
```
```

|             | estimate     | std.error  | statistic  | df       | p.value |
|-------------|--------------|------------|------------|----------|---------|
| (Intercept) | 1.822457999  | 0.04641326 | 39.2658866 | 100.5742 | 0.00000 |
| X2          | -0.006076888 | 0.02123498 | -0.2861734 | 289.4252 | 0.77495 |

- × **mod.imp** = model name for SLR
- × **pool()** = tells R to combine model estimates across each imputation
- × **summary()** = use to view model output

# COMPARE MODEL RESULTS BETWEEN THE MISSING DATA TECHNIQUES

Pairwise Deletion:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.82821    0.04443   41.152  <2e-16 ***
X2           -0.01142    0.02200   -0.519    0.604
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Pooled across MICE datasets:

```
              estimate  std.error  statistic      df p.value
(Intercept)  1.822457999  0.04641326  39.2658866 100.5742 0.000000
X2           -0.006076888  0.02123498  -0.2861734 289.4252 0.77495
```



# A FEW NOTES ON THE MICE PACKAGE

- ▣ The with() and pool() functions allow you to pool model estimates for many common analyses
- ▣ In general, you should examine missing data patterns before using mice
- ▣ Can take a lot of computational power and time to run in larger datasets
- ▣ Not currently compatible with machine learning and some multivariate analyses
  - ▣ Mplus has its own code for multiple imputation
- ▣ To read more on the mice package, view the vignette here:
  - ▣ <https://cran.r-project.org/web/packages/mice/mice.pdf>