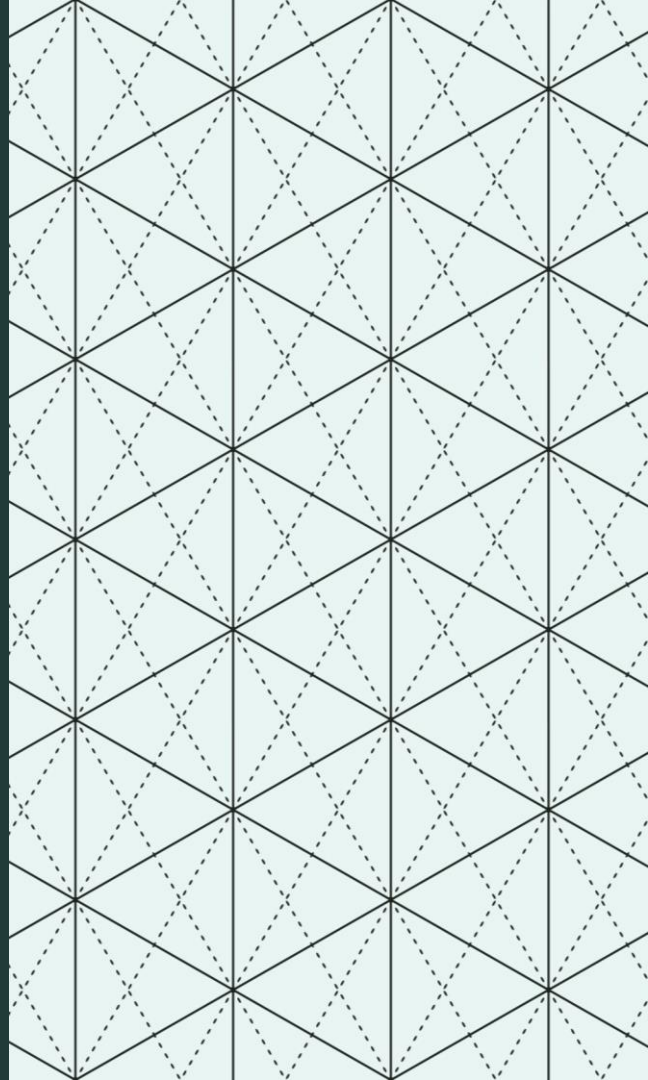
# WELCOME TO PSY 653 LAB!

MODULE 01:

NESTED AND INCOMPLETE DESIGNS IN ANOVAS



# Objectives

- Reviewing ANOVAs
- What is a nested design?
- Viewing Kevin’s nested designs
- Dataset description
- The “aggregate()” function (Because you saw it in lecture)
- Coding tutorial

# A quick review of ANOVAs

An example from last semester:

```
# Conduct an ANOVA
```{r}
lm1 <- lm(StressLevel ~ ProgramCode.f, data = stress)
anova(lm1)
```


 Analysis of Variance Table

 Response: StressLevel
                Df Sum Sq Mean Sq F value  Pr(>F)
 ProgramCode.f   3  54.83 18.2750  3.5623 0.01643 *
 Residuals     116 595.10  5.1302
 ---
 Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

*The p-value for the model F-test is significant at p<0.05, indicating that there was a significant effect of ProgramCode on stress level.*

ANOVAs evaluate how a continuous outcome varies across levels of a categorical predictor(s)

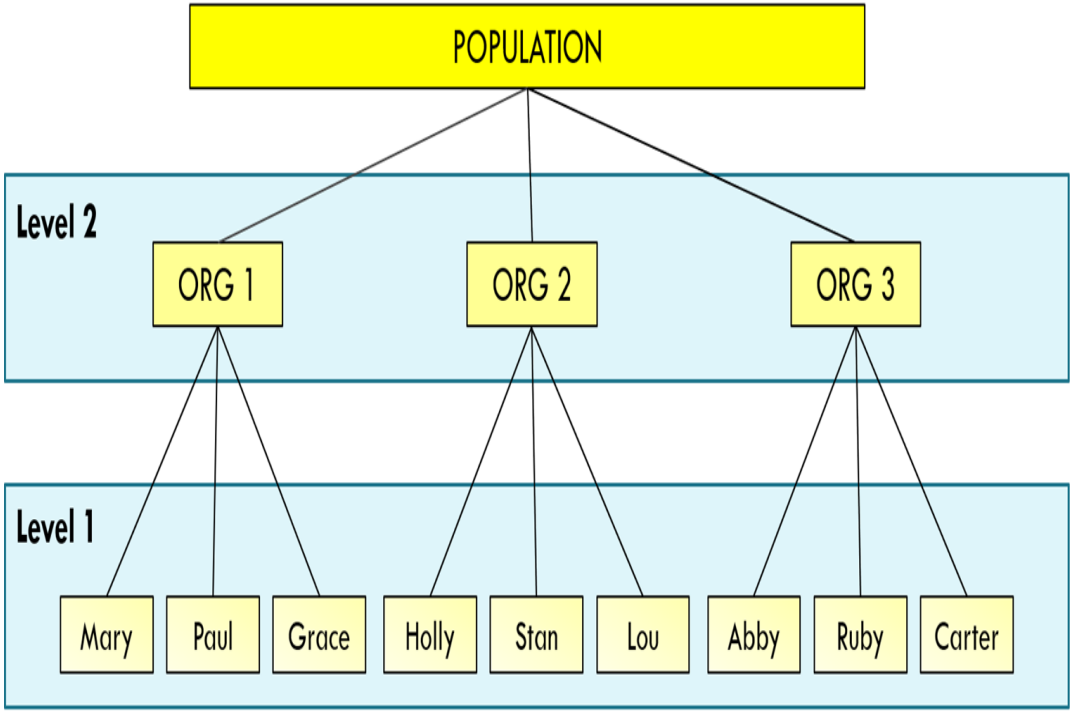
**Sum of squares (SS)**: The sum of squared deviations from each individual value and the mean

**Mean square (MS)**: The average distance values deviate from the mean (SS/df)

**F-statistic**: The ratio of model mean square to the residual mean square ($MS_{explained}/MS_{residual}$)

# What is a nested design?

- A **nested design** (aka hierarchical design) is when a factor(s) is *nested* within another factor.
- A **nested ANOVA** (aka hierarchical ANOVA) is used to evaluate the differences within variables while taking into account the nested structure of the data






https://www.statisticshowto.datasciencecentral.com/nested-model-anova-factors/

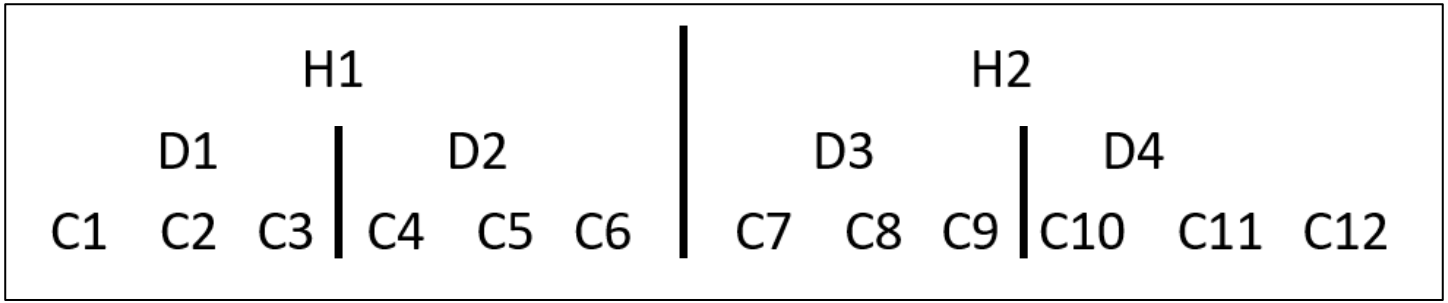
# Looking at Kevin's Nested Designs: ***Design 1***


**Drug *is* nested within hospital** in this data, because the drugs differ between the two hospitals.

**Condition *is* nested within hospital** in this data, because the conditions differ between the two hospitals.

**Condition *is* nested within drug & hospital**, because the conditions differ between the two drugs within hospital.

# Looking at Kevin's Nested Designs: ***Design 1***

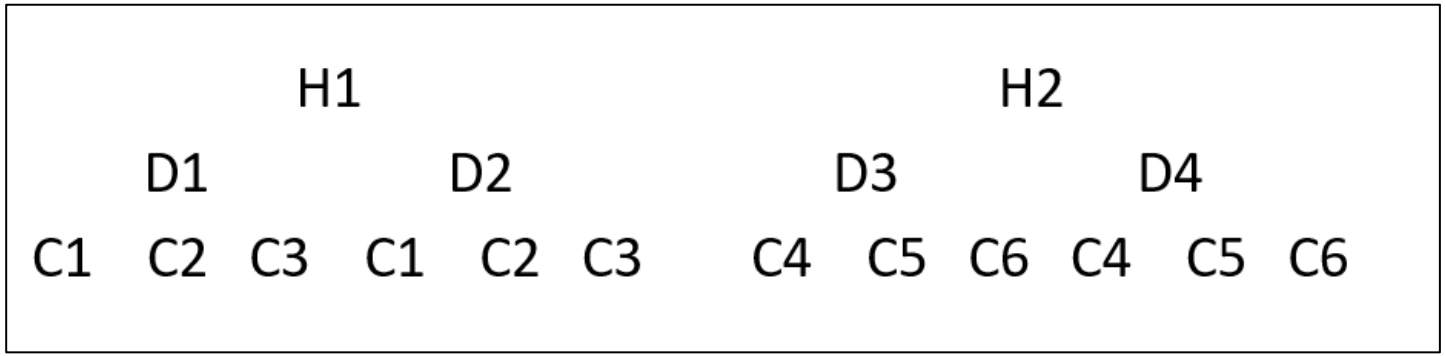



**You Can <u>NOT</u> Run any interactions with this design (It is a fully nested model).**

Y ~ Hospital + Hospital/Drug + Hospital/Drug/Condition

* A " / " indicates a nested structure

# Looking at Kevin's Nested Designs: *Design 2*

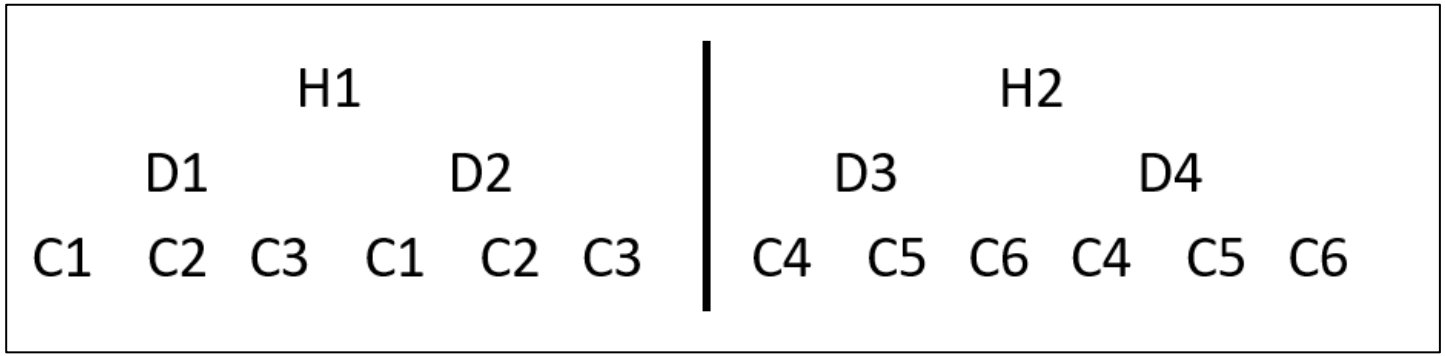



**Drug *is* nested within hospital** in this data, because the drugs different between the two hospitals

**Condition *is* nested within hospital** in this data, because the conditions different between the two hospitals.

**Condition is *NOT* nested within drug**, because the conditions are the same between the two drugs *within* hospital.

# Looking at Kevin's Nested Designs: *Design 2*

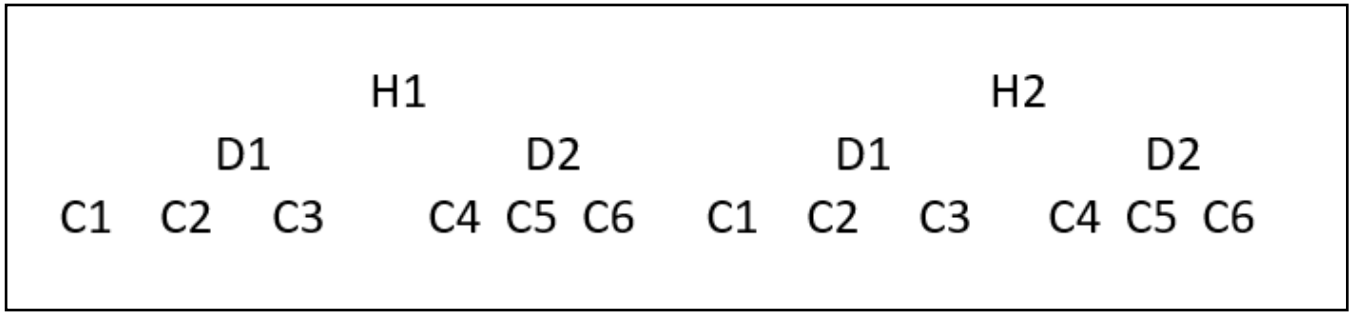



**You can run <u>ONE</u> interaction within this design (Hospital/Drug * Hospital/Condition) because the conditions are the same within hospital.**

Y ~ Hospital + Hospital/Drug + Hospital/Condition + Hospital/Drug*Hospital/Condition

* A " / " indicates a nested structure

# Looking at Kevin's Nested Designs: *Design 3*

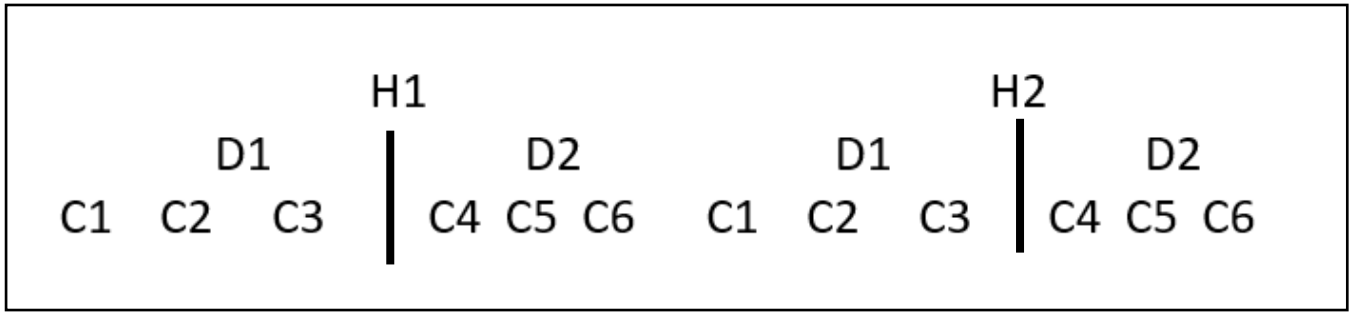



**Drug is <u>not</u> nested within hospital** in this data, because the drugs are the same between the two hospitals (i.e., the effects of a particular drug could be different from one hospital to the next).

**Condition is <u>not</u> nested within hospital** in this data, because the conditions are the same between the two hospitals (i.e., the effects of a particular condition could be different from one hospital to the next).

However, **condition *<u>is</u>* nested within drug**, because the conditions are different between the two drugs. (i.e., since each condition was not tested for each drug, we cannot assess the possibility that the different conditions are different between the drugs).

# Looking at Kevin's Nested Designs: *Design 3*




**You can run <u>TWO</u> interactions within this design (Hospital * Drug <u>AND</u> Hospital * Drug/Condition).**

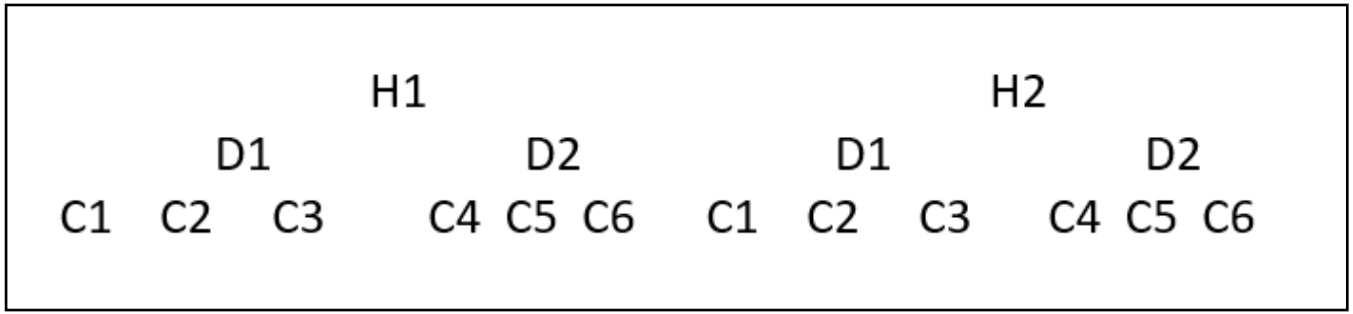
Y ~ Hospital + Drug + Drug/Condition + Hospital*Drug + Hospital*Drug/Condition

* A " / " indicates a nested structure

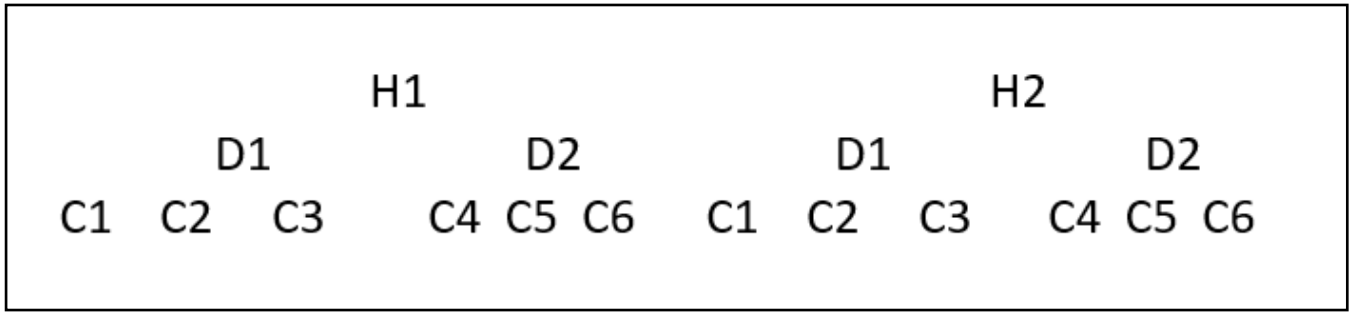
# Dataset Description

- This demo uses data from 114 patients who participated in study to evaluate the effects of different drugs and treatment conditions on health.
- Our overall research question:
  - *Does patient health significantly vary across hospitals, drugs, and treatment conditions?*
  - Outcome = Y (a proxy for health; Values can range = 1 - 15; Higher scores = better health)
  - Predictors = Hospital, Drug, Condition

# Structure of the data



# Structure of the data




**Drug is <u>not</u> nested within hospital** in this data, because the drugs are the same between the two hospitals (i.e., the effects of a particular drug could be different from one hospital to the next).

**Condition is <u>not</u> nested within hospital** in this data, because the conditions are the same between the two hospitals (i.e., the effects of a particular condition could be different from one hospital to the next).

However, **condition ***<u>is</u>* **nested within drug**, because the conditions are different between the two drugs. (i.e., since each condition was not tested for each drug, we cannot assess the possibility that the different conditions are different between the drugs).

# The aggregate function, broken down (Because you saw it in lecture)

```
aggregate(nested_demo$Y, by=list(Hospital=nested_demo$Hospital), mean)
```

- aggregate() = function
- nested_demo$Y = Continuous outcome variable
- Hospital = New name given for output (must be 1 word)
- nested_demo$Hospital = Categorical predictor variable
- mean = Desired descriptive function (i.e. mean, sd, median, etc.)

# The aggregate function, broken down (Because you saw it in lecture)

```
aggregate(nested_demo$Y, by=list(Hospital=nested_demo$Hospital), mean)
```

- aggregate() = function
- nested_demo$Y = Continuous outcome variable
- Hospital = New name given for output (must be 1 word)
- nested_demo$Hospital = Categorical predictor variable
- mean = Desired descriptive function (i.e. mean, sd, median, etc.)

# Let’s Code!