1. Why Is My Evil Lecturer Forcing Me to Learn Statistics?
   1. Research Process
      1. Data 🡨🡪 Initial Observation
      2. Generate Theory
      3. Generate Hypothesis 🡪 Identify Variables
      4. Collect Data to Test Theory 🡪 Measure Variables
      5. Analyze Data 🡪 Graph Data; Fit Model; Generate Theory
   2. Types of Variables
      1. **Categorical**
      2. **Binary**
      3. **Ordinal**
      4. **Continuous**
      5. **Interval**
      6. **Ratio**
   3. **Measurement error**
   4. Validity and Reliability
      1. **Validity**: whether an instrument measures what it says it does.
         1. **Criterion validity**
         2. **Content validity**: whether it covers the full range.
         3. **Ecological validity**: uninfluenced by environment or researcher.
      2. **Reliability**: whether an instrument can be interpreted consistently.
         1. **Test-retest reliability**: test the same group twice and likely to produce the same result.
   5. Correlational Research Methods vs. Experimental Research Methods
      1. **Correlational/Cross-Sectional**: observe what goes on without interfering.
         1. ***Tertium quid***: a third variable may be involved.
            1. **Confounding variable**
      2. **Experimental**: manipulate one variable.
   6. Two Methods of Data Collection
      1. **Between-groups/Between**-**subjects/Independent design**: different groups take part in each experimental condition
      2. **Within-subject/Repeated measures design**: manipulate a variable within the same participants
   7. Two Types of Variation
      1. **Unsystematic**: small differences created by unknown factors.
      2. **Systematic**: differences created by experimenter doing something to all participants in one condition but not the other.
         1. Sources
            1. **Practice effects**: participant performance differing in 2nd condition due to familiarity.
            2. **Boredom effects**: participant performance differing in 2nd condition due to boredom.
         2. **Counterbalancing**: ensuring no systematic variation by counterbalancing the order in which a participant participates in a condition.
      3. **Randomization**: allows reduction of systematic errors.
   8. Frequency Distributions
      1. **Kurtosis**: pointiness of distribution.
         1. **Leptokurtic**: positive kurtosis; pointy.
         2. **Platykurtic**: negative kurtosis; flat.
   9. Fitting Statistical Models to the Data
      1. **Alternative/Experimental hypothesis (H1)**: prediction from theory saying an effect will be present.
      2. **Null hypothesis (H0)**: no effect will occur.
2. Everything You Ever Wanted To Know About Statistics (Well, Sort Of)
   1. **Linear models**: most models used to describe data.
   2. Assessing the Fit of the Mean: Sum of Squares, Variance, and Standard Deviations (Measures of how well the model fits the data)
      1. **Deviance**: difference between each value and the mean.
      2. **Total error**: sum of deviances.
         1. Problem: Some deviances are cancelled out.
      3. **Sum of Squared errors (SS)**: sum of the squares of each deviance.
         1. Problem: dependent upon the number of data points.
      4. **Sample average error**: SS divided by the number of data points, *n*.
      5. **Variance (s2)**: SS divided by the degrees of freedom.
         1. **Degrees of freedom**: number of observations that are free to vary to estimate the population mean, keeping one parameter constant.
         2. ’
         3. Problem: gives answers in units squared.
      6. **Standard deviation (s or SD)**: square root of variance.
   3. Expressing the Mean as a Model
      1. Fundamental equation of statistics
   4. The Standard Error
      1. **Sampling variation**: samples vary because they contain different members of the population.
      2. **Sampling distribution**: frequency distribution of sample means (or any other statistics).
      3. **Standard error of the mean (SE)**: standard deviation of sample means; measure of how representative a sample is likely to be of the population.
      4. **Central limit theorem**
         1. As samples get large (*n* > 30), sampling distributions have a normal distribution with mean equal to population mean.
         2. If they are small, they follow a *t*-distribution.
   5. Calculating Confidence Intervals
      1. **Confidence intervals**: boundaries within which we believe the true value of the mean will fall.
         * 1. + for upper boundary
           2. – for lower boundary
         1. Types
            1. 95%: z-score of 1.96

Most common

* + - * 1. 99%: z-score of 2.58
  1. Calculating Confidence Intervals in Small Samples
     + 1. + for upper boundary
       2. – for lower boundary
  2. Showing Confidence Intervals Visually
     1. Shown as error bars
        1. Should overlap
        2. If not, probably from different populations
           1. Experimental manipulation has induced a difference.
           2. Thus, experimental manipulation is successful.
  3. **Test statistics**: ratio of systematic to unsystematic variance (effect to error).
     1. **Importance of an effect**
        1. Sometimes significance of unimportant events may be magnified by *n*.
     2. **Non-significant results**
        1. No difference between means; no relationship between variables.
        2. The effect is not big enough to be anything other than a chance finding; not necessarily that null hypothesis is true.
           1. Null hypothesis can never be true because sample means will always be different.
     3. **Significant results**
        1. Say very little about the null hypothesis.
  4. One-and Two-tailed Tests
     1. **One-tailed test**: statistical model which tests a directional hypothesis.
     2. **Two-tailed test**: tests a non-directional hypothesis.
  5. Type I and Type II Errors
     1. **Type I error**: believe there is a genuine effect on population, when there isn’t.
        1. **α-level**: probability of a Type I error.
           1. Acceptable: 5% or less.
     2. **Type II error**: believe there is no effect on population, when there is.
        1. **β-level**: probability of Type II error.
           1. Acceptable: 20% or less.
  6. **Effect size**: standardized measure of the magnitude of an observed effect.
     1. Many measures
        1. Cohen’s *d*
        2. Pearson’s correlation coefficient *r* (variance)
           1. r = 0; no effect
           2. r = 1; perfect effect
           3. r = -1 perfect opposite effect
           4. Nonlinear

r = 6 isn’t twice as big as r = 3

* + - 1. Odds ratio
    1. **Meta-analysis**: combining effect sizes from different studies researching the same question to estimate population effect sizes.
  1. **Statistical power**: ability to detect an effect of a certain size.
  2. What We Can Do With *n*, effect size, α-level, and statistical power.
     1. **Calculate the power of a test**:
     2. **Calculate *n* needed to achieve a given level of power**

1. The R Environment
   1. The R-chitechture
      1. **Packages**: add specific functionality to the program.
      2. **Comprehensive R Archive Network (CRAN)**: universal storage for packages as well as R.
   2. The Main Windows
      1. **Console**: type commands and see results of executions.
      2. **Editor**: separate window in which you can write commands
         1. Collections of commands can be saved.
      3. **Graphics/Quartz**: produce graphics or graphs.
   3. Using R
      1. Commands, Objects, and Functions
         1. R **is** case-sensitive in all its parts.
         2. **Commands**
            1. object<-function

“Object is created from function.”

* + - * 1. Multiple commands can be run on a single line by simply adding “; ” between commands.
      1. **Objects**: anything created in R (variable, multiple variable, statistical model, single objects, collection of objects).
      2. **Functions**: things that you do in R to create objects.
         1. **Concatenate function (c())**: groups things together with double quotation marks, comma, and a space.

metallica<-c(“Lars”, “James”, “Jason”, “Kirk”)

Contents can be viewed by typing

metallica

print(metallica)

Contents can be redefined

Removal by !=

metallica<-metallica[metallica != “Jason”]

“Redefine metallica using metallica, but get rid of Jason.”

Addition by using c()

metallica<-c(metallica, “Rob”)

“Redefine metallica using metallica, but group Rob.”

* + 1. Using scripts
       1. **Script**: document of commands written in the R editor.
          1. Savable and reloadable
  1. Packages
     1. Can be downloaded
     2. Disambiguating functions
        1. See R’s Soul’s Tip 3.4
  2. Getting data into R
     1. **String variables**: always need quotation marks.
        1. metallicaNames<-c(“Lars”, “James”, “Jason”, “Kirk”)
     2. **Numeric variables**: no need for quotation marks.
        1. metallicaAges<-(47,47,46,48)
     3. **Dataframe**: a spreadsheet containing variables.
        1. metallica<-data.frame(Name = metallicaNames, Age = metallicaAges)
        2. Print a variable within the dataframe.
           1. metallica$Age
        3. Define a variable within the dataframe.
           1. metallica$childAge<-c(12,12,4,6)
        4. Print variable names within the dataframe.
           1. names(metallica)
        5. Alternatives
           1. list()

Puts variables in a handbag

* + - * 1. cbind()

Puts variables in a matrix

Useful for variables of the same type

* 1. Calculating new variables from existing ones
     1. +: add
     2. -:subtract
     3. \*:multiply
     4. /:divide
     5. ^ or \*\*:exponent
     6. <: less than
     7. <=: less than or equal to
     8. >: greater than
     9. >=: greater than or equal to
     10. ==: exactly equal to
     11. !=: not equal to
     12. !x: not x
     13. x | y: x OR y
     14. x & y: x AND y
     15. isTRUE(x): test if x is true
  2. Organizing your data
     1. **wide format**
        1. Rows: data from one entity
        2. Columns: variables
           1. Independent/Dependent
           2. **Factors**: grouping (between-group) variables

*Nominal numeric variable*: uses codes (numbers) to represent names.

* + 1. Creating a *string variable*
       1. name<-c("Ben", "Martin", "Andy", "Paul", "Graham", "Carina", "Karina", "Doug", "Mark", "Zoe")
    2. Creating a **date variable**
       1. as.Date()
          1. Takes strings and converts them to date variables.
          2. variable<-as.Date(c("yyyy-mm-dd"))
    3. Creating coding variables/factors
       1. job<-c(1,1,1,1,1,2,2,2,2,2)
          1. Quick way using rep()function

job<-c(rep (1,5),rep(2,5))

* + - 1. Use the factor()function
         1. Factor(variable, levels = c(1, 2, … 3), labels = c(“label1, “label2”, “label3”))

factor(variable)creates the factor

levels = c(1, 2, … 3)lets R know which values we have used to denote different groups.

levels = c(1:3), allows R to count all values in between.

labels = c(“label1”, “label2”, “label3”)allows R to denote those labels to their corresponding codes.

* + - 1. Use the gl()general (factor) levels function
         1. newFactor<-gl(number of levels, cases in each level/group, total number of cases, labels = c(“label1”, “label2”…))

Total number of cases is optional

Multiply number of groups by number of groups per case

job<-gl(2, 5, 10, labels = c("Lecturer", "Student"))

* + - 1. Use levels(factor) to see level names, or change them.
         1. levels(job)<-c(“Medical Lecturer”, “Medical Student”)
    1. Creating a numeric variable
       1. Put all values in a dataframe for ease.
    2. Missing values
       1. Type NA
          1. na.rm = TRUE in a function to tell R to ignore missing values before a computation.