Example 2: General Linear Models with One Quantitative or Binary Predictor
(complete data, syntax, and output in R available online)

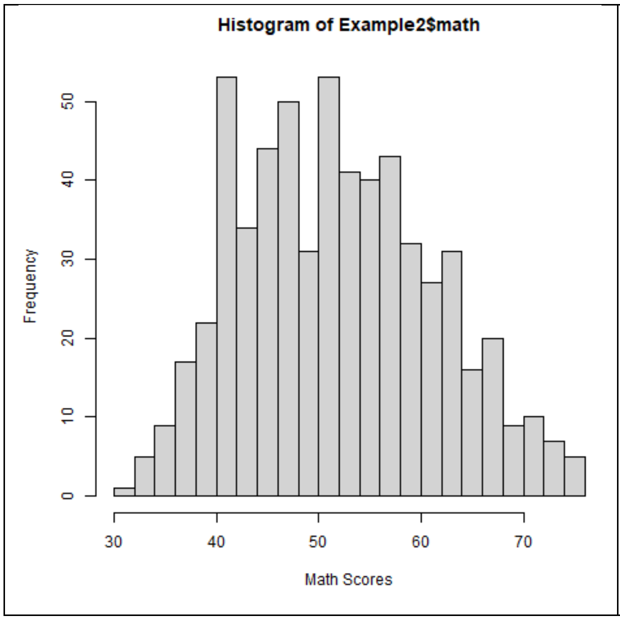
These example data were selected from the High School and Beyond 1980 dataset (in the `candisc` R package). The current example will use general linear models (estimated within the base R `lm` function) to examine associations of math with writing and binary gender. It will also introduce how to obtain linear combinations of fixed effects to create predicted outcomes using the `glht` function from the `multcomp` R package.

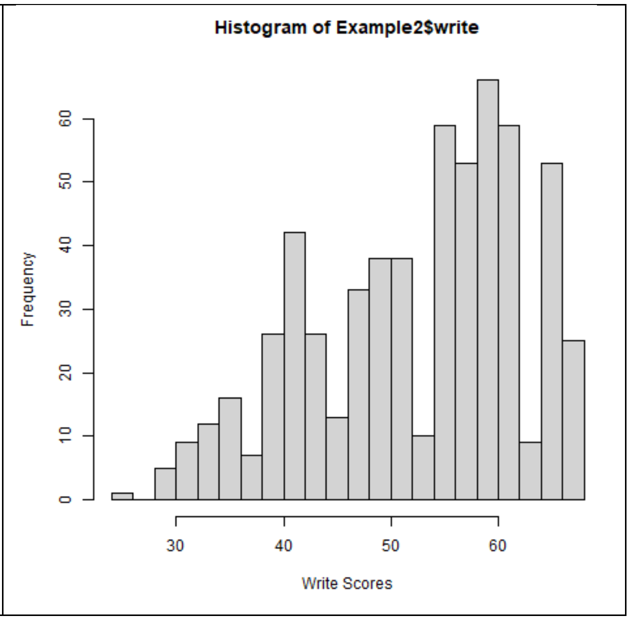
R Syntax for Importing and Preparing Data for Analysis (after loading packages **`TeachingDemos, readxl, psych, multcomp, lm.beta, Hmisc, and supernova`**):

```
# Set working directory (to import and export files to)
# Paste in the folder address where your data file is saved in quotes
# Note the slashes are backwards relative to Windows file paths
setwd("C:/Dropbox/26_EDF9770/Example2/")

# Import "HSB_Example.xlsx" from sheet "Sheet1" with first row as variable names
Example2 = read_excel(path="HSB_Example.xlsx", sheet="Sheet1", col_names=TRUE)
# Convert to data frame to use for analysis
Example2 = as.data.frame(Example2)

# Load R functions for this class from R file in working directory
source("EDF9770_Functions.R")
```

R Descriptive Statistics:

```
# to save a plot: open a file, create the plot, then close the file
png(file="Math Histogram Plot.png")  # open file
hist(x=Example2$math, freq=TRUE, breaks=20, ylab="Frequency", xlab="Math Scores")
dev.off()  # close file

# to save a plot: open a file, create the plot, then close the file
png(file="Write Histogram Plot.png")  # open file
hist(x=Example2$write, freq=TRUE, breaks=20, ylab="Frequency", xlab="Write Scores")
dev.off()  # close file
```





```
# describe prints sample descriptive statistics for quantitative variables
# List variables to be included in separate quotes within c concatenate function
# Wrapped a print command around to get more than two significant digits

print("Descriptive Statistics for Quantitative or Binary Variables")
print(psych::describe(x=Example2[ , c("math","write","gender")], fast=TRUE), digits=3)

       vars   n   mean    sd median  min  max range   skew kurtosis    se
math      1 600 51.849 9.415   51.3 31.8 75.5  43.7  0.263   -0.653 0.384
write     2 600 52.385 9.726   54.1 25.5 67.1  41.6 -0.470   -0.714 0.397
gender    3 600  1.545 0.498    2.0  1.0  2.0   1.0 -0.180   -1.971 0.020

print("Get variances too (on diagonal of output covariance matrix)")
var(x=Example2[ , c("math","write","gender")])

          math  write  gender
math   88.6373 57.935 -0.2262
write  57.9345 94.604  1.1844
gender -0.2262  1.184  0.2484
```

In this “variance–covariance” matrix:

Diagonal has **variances**: $var(y_i) = \frac{\sum_{i=1}^{N}(y_i-\bar{y})^2}{N-1}$

Off-diagonals have **covariances**: $cov(x_i, y_i) = \frac{\sum_{i=1}^{N}[(x_i-\bar{x})(y_i-\bar{y})]}{N-1}$

Because **covariances are pretty much impossible to interpret** (other than their sign), we usually summarize bivariate associations using **correlations** instead (such as the Pearson correlations given below).

```
print("Pearson Correlation Matrix with p-values using rcorr from Hmisc package")
print("Must convert data frame to matrix to use rcorr")
Hmisc::rcorr(x=as.matrix(Example2[,c("math","write","gender")]), type="pearson")

        math write gender
math    1.00  0.63  -0.05
write   0.63  1.00   0.24
gender -0.05  0.24   1.00
n= 600


P
       math   write  gender
math          0.0000 0.2383
write  0.0000        0.0000
gender 0.2383 0.0000
```

In this “correlation” matrix:
Diagonal has **variances of standardized variables = 1**
Off-diagonals have **correlations** (the variables’ covariance divided by their standard deviations) : $r = \frac{cov(x_i,y_i)}{s_x s_y}$

These are the corresponding *p*-values for significance tests of each correlation against $H_0 = 0$, in which .0000 would be reported as < .0001 instead (i.e., *p*-values are never 0).

Empty General Linear Model (no predictors):

$$\boldsymbol{math_i} = \boldsymbol{\beta_0} + \boldsymbol{e_i}$$

The empty model is our starting point—the most naïve prediction of $\boldsymbol{y_i}$ = math in which everyone is predicted to have the mean math: $\boldsymbol{\hat{y}_i} = \boldsymbol{\beta_0}$. Thus, the variance of the $\boldsymbol{e_i}$ residuals will be ALL the $\boldsymbol{y_i}$ variance, as given above by 88.637 and in as given below in the first “Sums of Squares Table”.

```
print("Empty GLM Predicting Math -- save as Math_Empty")
print("1 represents fixed intercept")
Math_Empty = lm(data=Example2, formula=math~1)
obj=LMsummary(Math_Empty, explain=TRUE)  # Full custom output with explanations

Sums of Squares Table
             SS  DF     MS F p R2
Model
Error
Total 53093.718 599 88.637

Explanation:
SS = Sum of Squares, MS = Mean Square, DF = Degrees of Freedom,
F = F test-statistic, p = two-sided p-value, R2 = R-square
```

The “Fixed Effects Table” below provides full results for each fixed effect—just the β_0 intercept so far.

```
Fixed Effects Table
             Est    SE       t      p    LCI    UCI
Intercept 51.849 0.384 134.899 <0.001 51.094 52.604

Explanation:
Est = Estimate, SE = Standard Error, t = t test-statistic,
p = p-value, LCI = Lower Confidence Interval,
UCI = Upper Confidence Interval
```

Now let’s see if writing can predict math by giving writing a fixed linear slope!

Because writing has a lower bound of 25.5 (with a mean = 52.385), we first need to center it so that 0 will be a meaningful value for the β_0 fixed intercept. I picked 50 because it is a round number on the middle (but any value within the range of the write predictor would be fine).

$$math_i = \beta_0 + \beta_1(write_i - 50) + e_i$$

```
# Center quantitative write variable to be used as a predictor
Example2$write50 = Example2$write-50  # write50: Writing Score (0=50)

print("GLM Predicting Math from Centered Write 0=50 -- save as MathWrite50")
Math_Write50 = lm(data=Example2, formula=math~1+write50)

summary(Math_Write50)  # Default output

Residuals:
    Min      1Q  Median      3Q     Max 
-20.968  -4.861  -0.288   5.007  21.701 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  50.3886     0.3067     164   <2e-16
write50       0.6124     0.0307      20   <2e-16

Residual standard error: 7.3 on 598 degrees of freedom
Multiple R-squared:   0.4,       Adjusted R-squared:  0.399 
F-statistic:  399 on 1 and 598 DF,  p-value: <2e-16
```

Here is what is given by the default summary function for lm. The residual variance is not given, but the residual SD is (mislabeled as residual SE). In practice, the residual variance should be reported in results sections for each model.

And here is what is returned with my LMsummary function instead:

```
obj=LMsummary(Math_Write50, explain=TRUE)  # Full custom output with explanations

Sums of Squares Table
             SS  DF        MS       F      p    R2
Model 21251.652   1 21251.652 399.110 <0.001 0.400
Error 31842.066 598    53.248                     
Total 53093.718 599    88.637                     

Explanation:
SS = Sum of Squares, MS = Mean Square, DF = Degrees of Freedom,
F = F test-statistic, p = two-sided p-value, R2 = R-square

Fixed Effects Table
             Est    SE       t      p    LCI    UCI
Intercept 50.389 0.307 164.271 <0.001 49.786 50.991
write50    0.612 0.031  19.978 <0.001  0.552  0.673

Explanation:
Est = Estimate, SE = Standard Error, t = t test-statistic,
p = p-value, LCI = Lower Confidence Interval,
UCI = Upper Confidence Interval
```

Interpret β_0 = intercept:

Interpret β_1 = slope of write50:

How much math variance is leftover after considering writing (i.e., what is σ_e^2)?

Generating predicted outcomes for illustrative purposes:

$$math_i = \beta_0 + \beta_1(write_i - 50) + e_i$$

Pred math for write=40: $\widehat{math} = 50.389(1) + 0.612(40 - 50) = 44.265$
Pred math for write=50: $\widehat{math} = 50.389(1) + 0.612(50 - 50) = 50.389$
Pred math for write=60: $\widehat{math} = 50.389(1) + 0.612(60 - 50) = 56.512$

```
print("Get predicted outcomes -- save as Pred_Math_Write50")
print("In number lists below, the values are multipliers for each fixed effect in order")
Pred_Math_Write50 = multcomp::glht(model=Math_Write50, linfct=rbind(
  "Pred math for write=40 (write50=-10)" = c(1,-10),  # beta0(1) + beta1*(-10)
  "Pred math for write=50 (write50=  0)" = c(1,  0),  # beta0(1) + beta1*(  0)
  "Pred math for write=60 (write50= 10)" = c(1, 10))) # beta0(1) + beta1*( 10)
obj=glhtSummary(glhtObject=Pred_Math_Write50, explain=TRUE) # Custom output + explanations
```

```
Linear Combinations Table
                                        Est    SE       t      p    LCI    UCI
Pred Math for write=40 (write50=-10) 44.265 0.483  91.727 <0.001 43.317 45.212
Pred Math for write=50 (write50=  0) 50.389 0.307 164.271 <0.001 49.786 50.991
Pred Math for write=60 (write50= 10) 56.512 0.378 149.320 <0.001 55.769 57.256

Explanation:
Est = Estimate, SE = Standard Error, t = t test-statistic,
p = p-value, LCI = Lower Confidence Interval,
UCI = Upper Confidence Interval
```

Generating one measure of effect size—a standardized slope:

```
print("Standardized fixed effect solution using lm.beta package")
lm.beta(Math_Write50)
```

```
Standardized Coefficients::
(Intercept)      write50
         NA     0.632666
```

Because there is only one predictor, the standardized slope (i.e., between z-scored versions of the variables, in which mean = 0 and SD = 1) is the same as the Pearson correlation between math and write. This provides an effect size to describe the size of the relation between math and writing in absolute terms.

Now let's see if there are gender differences in math by giving gender a fixed linear slope!

Because gender was coded as 1 = male and 2 = female, we first need to recode it (i.e., center it) so that 0 will be a meaningful value for the β_0 fixed intercept as one of the groups. I tend to name binary variables with the letters in order of the 0 and 1 groups, as shown below. Alternatively, it is common to name them after the group coded 1 (i.e., a name of "female" instead here).

$$math_i = \beta_0 + \beta_1(genderMF_i) + e_i$$

```
# Recode gender predictor so that 0 is meaningful (as male)
Example2$genderMF=NA  # Create new empty variable
Example2$genderMF[which(Example2$gender==1)]=0  # genderMF=0 if gender=1 (male)
Example2$genderMF[which(Example2$gender==2)]=1  # genderMF=1 if gender=2 (female)
# genderMF: 0=male, 1=female                    # Label as a comment only

print("GLM Predicting Math from Gender (0=Male, 1=Female) -- save as Math_GenderMF")
Math_GenderMF = lm(data=Example2, formula=math~1+genderMF)
obj=LMsummary(Math_GenderMF, effectsizes=TRUE) # Custom output with effect sizes
```

```
Sums of Squares Table
             SS  DF      MS     F     p    R2
Model   123.443   1 123.443 1.394 0.238 0.002
Error 52970.275 598  88.579
Total 53093.718 599  88.637

Fixed Effects Table
             Est    SE      t      p    LCI    UCI
Intercept 52.345 0.570 91.896 <0.001 51.227 53.464
genderMF  -0.911 0.772 -1.181  0.238 -2.426  0.604
```

The residual variance is now σ_e^2 **= 88.579**. **What do you think that means about how well gender predicts math?**

Interpret β_0 = intercept:

Interpret β_1 = slope of genderMF:

How much math variance is leftover after considering writing (i.e., what is σ_e^2)?

What would the slope be if gender was coded 0 = females and 1 = males instead?

To get a Cohen's d effect size for the mean math difference between males and females, we can calculate d from the t test-statistic: $d = \frac{2t}{\sqrt{DF_{den}}} = \frac{2*-1.181}{\sqrt{598}} = -0.097$ → means this is about 0.097 standard deviations lower for females than males.

`effectsizes=TRUE` creates this table, which repeats Est and *p* for convenience and adds three effect sizes.

In this model with only one predictor (i.e., a bivariate model), the partial correlation given here is the same as Pearson correlation given earlier.

```
Effect Sizes for Fixed Effects Table
            Est     p      d     pr   sR2
genderMF -0.911 0.238 -0.097 -0.048 0.002

Explanation:
Est = Estimate, p = two-sided p-value, d = Cohen's d,
pr = Partial r, sR2 = Semi-Partial R-square
```

Generating predicted outcomes for illustrative purposes:

$$math_i = \beta_0 + \beta_1(genderMF_i) + e_i$$

Pred math for males=0: $\widehat{math} = \mathbf{52.345}(\mathbf{1}) - \mathbf{0.911}(\mathbf{40} - \mathbf{50}) = \mathbf{52.345}$

Pred math for females=1: $\widehat{math} = \mathbf{52.345}(\mathbf{1}) - \mathbf{0.911}(\mathbf{50} - \mathbf{50}) = \mathbf{51.435}$

```
print("Get predicted outcomes -- save as Pred_Math_GenderMF")
print("In number lists below, values are multiplier for each fixed effect in order")
Pred_Math_GenderMF = multcomp:glht(model=Math_GenderMF, linfct=rbind(
  "Pred math for genderMF=0" = c(1,0),    # beta0(1) + beta1*(0)
  "Pred math for genderMF=1" = c(1,1)))  # beta0(1) + beta1*(1)
obj=glhtSummary(glhtObject=Pred_Math_GenderMF)  # Brief custom output
```

```
Linear Combinations Table
                            Est     SE      t      p    LCI    UCI
Pred Math for GenderMF=0 52.345 0.570 91.896 <0.001 51.227 53.464
Pred Math for GenderMF=1 51.435 0.520 98.824 <0.001 50.412 52.457
```

Example Results Section
(although it's more verbose than would be typical for the sake of completeness here):

The extent to which math could be predicted from writing and binary gender was examined in separate general linear models (i.e., simple linear regressions). All analyses were conducted using the lm function in R v. 4.5.2. Predicted outcomes were generated using the glht function within the multcomp package v. 1.4-29.

To create a meaningful model intercept, writing was centered such that 0 = 50 (near its mean). Writing was found to be a significant predictor of math: Relative to the reference expected math for a person with writing = 50 provided by the model intercept of 50.39 (SE = 0.31), for every additional unit of writing, math was expected to be higher by 0.61 (SE = 0.03, *p* < .001), resulting in a standardized slope = 0.63 (i.e., the Pearson correlation between math and writing). For example, persons with writing = 40 had predicted math = 44.27 (SE = 0.48), whereas persons with writing = 60 had predicted math = 56.51 (SE = 0.38).

We then examined prediction of annual income by binary gender. To create a meaningful model intercept, gender was recoded so that 0 = males and 1 = females. Gender was not a significant predictor of math: Relative to the reference expected math for male students provided by the model intercept of 52.35 (SE = 0.57), female students were expected to have nonsignificantly lower math by 0.91 (SE = 0.77, *p* = .238), resulting in a predicted math for female students of 51.44 (SE = 0.52) and a standardized mean difference of Cohen's *d* = −0.10.

Note: because a GLM with a single binary predictor is also known as a "two-sample t-test" here is what the results would look like written from that angle… A two-sample *t*-test (i.e., assuming homogeneous variance across groups) was used to examine mean differences between male and female students in math scores. A nonsignificant mean difference was found, *t*(598) = −1.18, *p* = .238, such that math for female students (*M* = 51.44, SE = 0.52) was nonsignificantly lower on average than math for male students (*M* = 52.35, SE = 0.57).

Challenges:

1. Write syntax to predict science from locus of control, in which 0.5 is the reference.
   What do you conclude?
2. Write syntax to predict science from school type (1=public, 2= private), in which private is the reference.
   What do you conclude?