An Application of Linear Algebra in Least-Squares Solutions and Statistical Inference

Linear Algebra, Spring 2017

Mark Lavelle

Science and Least-Squares Solutions

In scientific investigations, multiple, related types of data are often recorded on different variables involved in a phenomenon for the purposes of understanding, predicting, and controlling the relationships between those variables and the phenomenon itself. Relationships between quantitative data may be modeled with equations. Commonly, the question is posed as to how data covary as a function of one another. Often, the dependent variable is expressed as a function of a linear combination of a-priori hypothesized functions (e.g. polynomials, sinusoids) of the independent variables. The purpose of the data analyses, in these cases, is to estimate the coefficients on those functions.

A typical analysis of this form would involve equations such as

$$y_i = B_0 + B_1 f(X1_i) + \ldots + B_k g(Xn_i) + e_i \quad \textbf{(1)}$$

where *i* is the observation or data sample, *y* represents the dependent variable, B_0 represents the intercept, *f(), g(),* etc., represent functions of the independent variables X1-Xn, B_1-B_k represent the coefficients for the *k* functions for which we need to solve, and e_i represents the error term, or the difference between the value of the DV predicted from the model and the value actually recorded for a given observation *i*. Commonly, multiple observations on all the variables are recorded, and these can be stored in vectors in R^i. The vector of the observations of the dependent variable is commonly referred to in linear algebra as the **observation vector**, ***y***. The vectors of the function-values of the independent variables can be augmented into a matrix whose dimensions are *i*-by-*k*+1, where *k* is the number of functions of the independent variables included in the equation; B_0 is usually represented by a vector of 1s (e.g.<1,1,…,1>) in R^i and is also augmented with those vectors, giving the matrix it's +1 column. This matrix is often called the **design matrix**, ***X***. The unknown parameters B_0-B_k are stored in a vector in R^{K+1} titled the **parameter vector, *B***.

Linear algebra is useful for solving the equation

$$\boldsymbol{XB=y} \quad \textbf{(2)}$$

for ***B***. In typical applications, this equation usually forms many inconsistent linear equations because of unaccounted variance in ***y***. Additionally, *i* is (necessarily, for the purposes of finding a unique solution to equation 2) much larger than *k*. However, it is often desirable to obtain parameter estimates to *approximate* ***y*** with as little prediction error as possible. Though prediction error can be measured many ways, it is ubiquitously expressed as the sum of the squared e_i terms. A unique solution for equation 2 that minimizes the sums of squares of the residuals (SS_{resid}) is usually obtainable. This is commonly referred to as the least-squares solution. This solution, known as $\hat{\boldsymbol{B}}$, contains the coefficients on the columns of ***X*** which minimize the distance between ***y*** and the hyperplane spanned by the columns of ***X***. Supposing the columns of ***X*** are independent,

$$\hat{\boldsymbol{B}} = (\boldsymbol{X'X})^{-1}\boldsymbol{X'y} \textbf{ (3)}$$

where ' represents the transpose operator and $^{-1}$ represents the inverse operator. The method of solving this equation to obtain $\hat{\boldsymbol{B}}$ is called ordinary least squares (OLS) regression.

Statistical Assumptions and Inference

It is often desirable to infer from one's sample of data the behavior of the phenomenon in general, or in cases that have not yet been observed. Under certain assumptions about the **residuals** (the e_i terms), it can be shown that the values for B_0-B_k obtained for any particular sample are unbiased estimates of those same parameters in the population. Therefore, $\hat{\boldsymbol{B}}$ from one's sample is the best guess of how the functions of the independent variables relate to the dependent variable. Exactly how accurate the parameter estimates $\hat{\boldsymbol{B}}$ are, or how confident we can be that a given margin of error around $\hat{\boldsymbol{B}}$ contains the true population values, is a function of the variability in the independent variables X1-Xn and their covariance with the y. An index of this variance within an IV and covariance between that IV and the DV is the **standard error** (**SE**) of the estimate. For regression coefficients, the standard errors can be obtained by

$$\sigma^2(\boldsymbol{X'X})^{-1} \textbf{ (4)}$$

where $\sigma^2 = SS_{resid}$. σ^2 can be calculated as

$$(I - \boldsymbol{X}(\boldsymbol{X'X})^{-1}\boldsymbol{X})\boldsymbol{y} \textbf{ (5)}$$

where I is the i-by-i identity matrix.

Taking for granted assumptions about the model in equation 1, and the assumptions about the residuals, Student's t distributions can be used to: test the probability that a parameter is non-zero (also known as null-hypothesis significance testing, NHST), or; provide a multiplier to apply to the SE, yielding a **margin of error** (**ME**) with a given confidence level. This margin of error is then subtracted and added to the parameter estimate to yield a **confidence interval** (CI) with a given probability of confidence. For example, a t-test against the null hypothesis can be conducted by dividing the value of a particular parameter estimate by its SE, yielding a t-score with a given amount of **degrees of freedom** (df). The t-score can be compared to the t-distribution with mean=0, SE=the standard error of the parameter being tested, and given df, to determine the percentage of the distribution which is more extreme than that score. This percentage is commonly referred to in statistics for the social sciences as the significance- or p-value. Similarly, a t-score with a particular p-value can be chosen from a t-distribution with given df. Commonly, t-scores with p-value = .025 are chosen and are used to provide a two-tailed 95% CI around the parameter estimate. The 95% CI around the parameter estimate is interpreted to contain the true population parameter 19 out of 20 times. A connection between NHST and confidence intervals is that if a p% CI does not include zero, the significance value of the test that the parameter equals zero is less than p/100.

In addition to testing the significance of particular parameters, it is often of interest to determine how well the entire model in equation 1 can predict the dependent variable. This is often quantified as a question about how much variance in the DV that the model accounts for, relative to the total variance in the dependent variable. Analysis of variance (ANOVA) tests exactly that question by taking the ratio of explained variance ($SS_{regression}/df_{regression}$) to the unexplained variance ($SS_{residual}/df_{residual}$), known as the F ratio. The null hypothesis is that the

explained variance will be proportional to the unexplained variance, that is F=1. The significance of the F statistic can be found by comparison to an F distribution with $df_{\text{regression}}$, df_{residual}. The significance value reflects the likelihood that the model does not explain more variance than would be expected by chance due to sampling error.

Application to Experimental Psychology Data

Data. The above theory was applied to a dataset available through a statistics class I previously took. The data are purported to have been collected in a psychological experiment assessing the influence of induced fear and human participants' biological sex on their perceptions of affordances judged from auditory stimuli. In particular, participants were randomly assigned to either: 1) write with pen and paper about an experience that caused them to be fearful, or; 2) write with pen and paper about a neutral topic. Participants' biological sexes were also noted. After writing about either type of experience, participants were asked to repeatedly listen to auditory stimuli that were simulated to give the perception of sound-sources at different distances. For each stimulus, participants were asked to indicate whether they thought they could reach the sound source. Finally, the average distance of the sound sources that they judged reachable was calculated. These distances were recorded in centimeters. 67 participants completed the experiment, 40 of which were female. 34 were assigned to the fearful writing condition. The mean of participants' average distances judged reachable was 77.8cm, SD=14.9.

Coding Categorical Variables. The researchers were interested in whether induced fear would change peoples' perceptions of their average distances judged reachable, and whether this relationship would change according to gender. To test these hypotheses, average distances judged reachable were regressed onto gender, writing condition, and their interaction, as categorical variables. Gender was dummy coded with 0 meaning male, 1 meaning female. This dummy code variable is referred to hereafter as "female". Writing condition was dummy coded with 0 meaning the control condition (writing about a neutral topic) and 1 meaning the fear condition. This dummy code variable is referred to hereafter as "fear". The interaction between fear and female was calculated by assigning a 1 to participants who were both female AND assigned to write about a fearful experience. All other participants were assigned a zero.

The least-squares equation. The design matrix and parameter vector used in the normal equation to find a least-squares solution for the observation vector were constructed as follows. A design matrix was constructed via augmentation of the constant vector (containing all ones), the fear vector, the female vector, and the interaction vector, in that order. The parameter vector consisted of 4 rows, with the first assigned for B_0, or the intercept. As an artifact of the coding scheme, the intercept represents the mean distance judged reachable for males in the control condition (i.e., participants with 0s for female and fear). The second row in the parameter vector designated B_1, or the coefficient for the fear variable. The value of B_1 represents the average difference between males assigned to different writing conditions in terms of distances judged reachable, with positive values indicating longer distances judged reachable in the fear relative to the control condition. The third row designated B_2, or the coefficient for the female vector. The value of B_2 represents the average difference between males and females in the control condition in terms of distances judged reachable, with positive values indicating that females judged longer distances as reachable than males. The third row designated B_3, or the coefficient for the interaction between fear and female. The value of B_3 represents the difference between males

and females in the relationship between writing condition and distance judged reachable, with positive values indicating that the effect of writing condition on distances judged reachable is greater for females than males. Here is the equation in scalar form

$$\text{Distance judged reachable}=B_0 + B_1*(\text{Condition: Control-0; Fear-1}) + B_2*(\text{Sex: Male-0; Female-1}) + B_3*(\text{Condition*Sex}) \quad \textbf{(6)}$$

The parameters were solved for using a custom script in *R*. Standard errors, significance values, and confidence intervals were also obtained. Total model fit was calculated. Figure 1 shows distances judged reachable plotted against writing condition, with least-squares lines added separately for gender. Figure 2 shows distances judged reachable plotted against gender, with least-squares lines added separately for writing condition. Figure 3 illustrates the improvement in prediction of distances judged reachable by using the model, relative to the mean of the data alone, i.e. the residuals of the regression relative to the variance of the raw data.

Results. Typically, statistical effects with significance value greater than .05 are considered "insignificant," that is they are considered as spurious findings due to sampling error. In this spirit, only the coefficient for the female dummy code variable was significant. The parameter estimate obtained from this sample indicates that, on average, females tend to perceive their capacity for reaching as 11.4cm shorter than the distance over which males perceive themselves as capable of reaching. Because the interaction variable was insignificant at p=.05, this effect for gender on distances judged reachable is regarded as generalizing across both writing conditions. The main hypothesis was not supported, in that induced fear did not significantly change distances judged as reachable relative to neutral emotional state.

The model as a whole accounted for 16.7% of the variance in distances judged reachable. This reduction of variance was significant, $F(3,63)=4.28$, p=.008

Results of OLS Multiple Regression					
Coefficient	**Parameter Estimate**	**Standard Error**	***t*-score (df=63)**	**95% CI**	**Significance**
B_0	79.3	14.9			
B_1	2.12	5.39	.394	[-8.64, 12.89]	.695
B_2	-11.4	4.87	-2.35	[-21.1, -1.7]	.022
B_3	12.7	6.97	1.82	[-1.23, 26.61]	.074

Conclusion

Linear equations involving empirical data are almost never consistent because of measurement error and unaccounted variance. Therefore, least-squares solutions are sought for the unknown parameters involved in hypothesized models that attempt to predict dependent variables from independent variables. When these models and the residuals meet certain assumptions, statistical inference from the sample estimates of parameters to the population parameters is possible. Linear algebra aids in the conceptualization and computation of solving for parameter estimates and in calculating the forms of variance involved in statistical inference. Therefore, linear algebra is a ubiquitous, invaluable tool in scientific inquiry.

Figure 1.

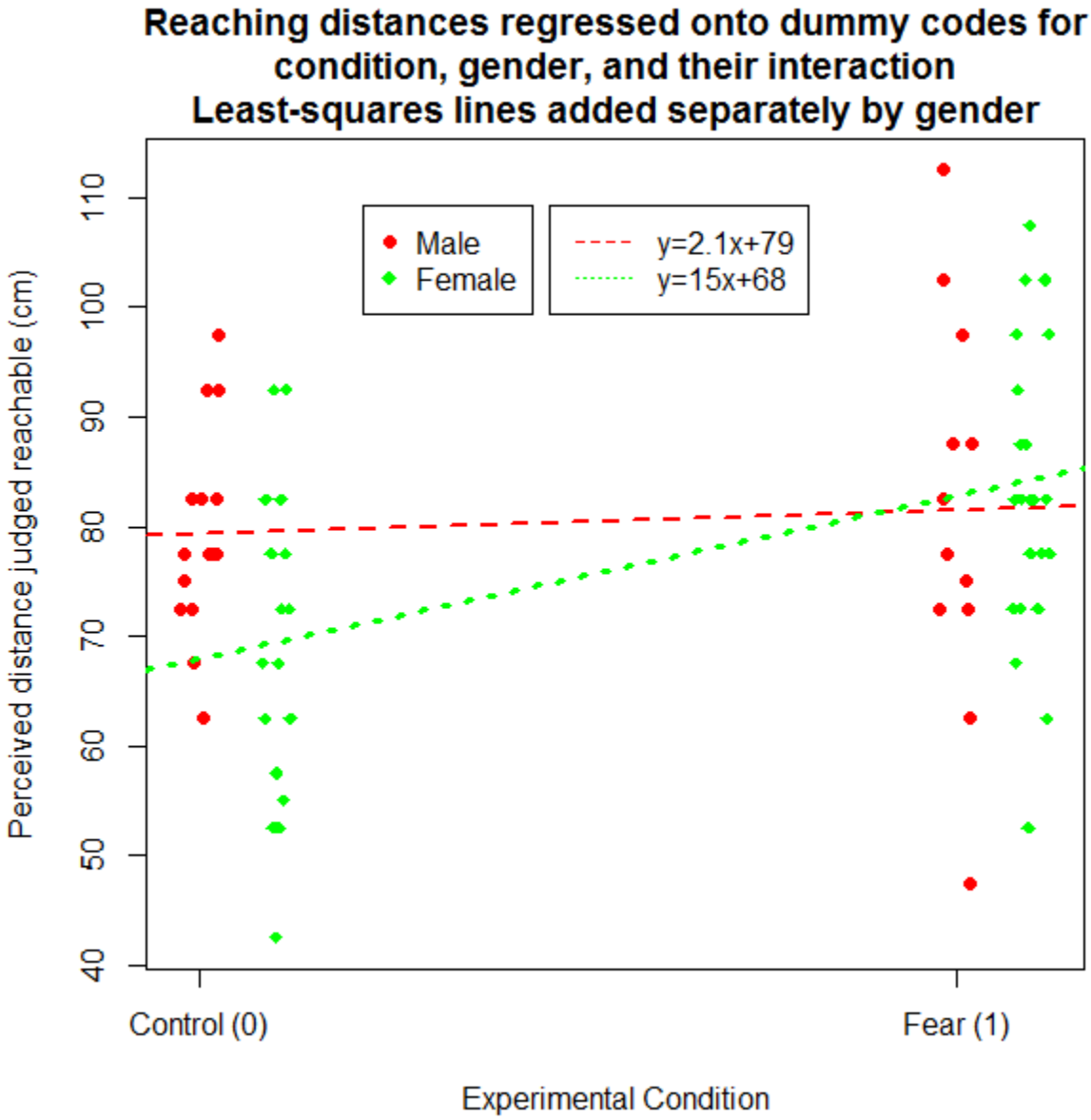

Reaching distances regressed onto dummy codes for condition, gender, and their interaction
Least-squares lines added separately by gender
Male
Female
y=2.1x+79
y=15x+68
Perceived distance judged reachable (cm)
40
50
60
70
80
90
100
110
Control (0)
Fear (1)
Experimental Condition

Figure 2.

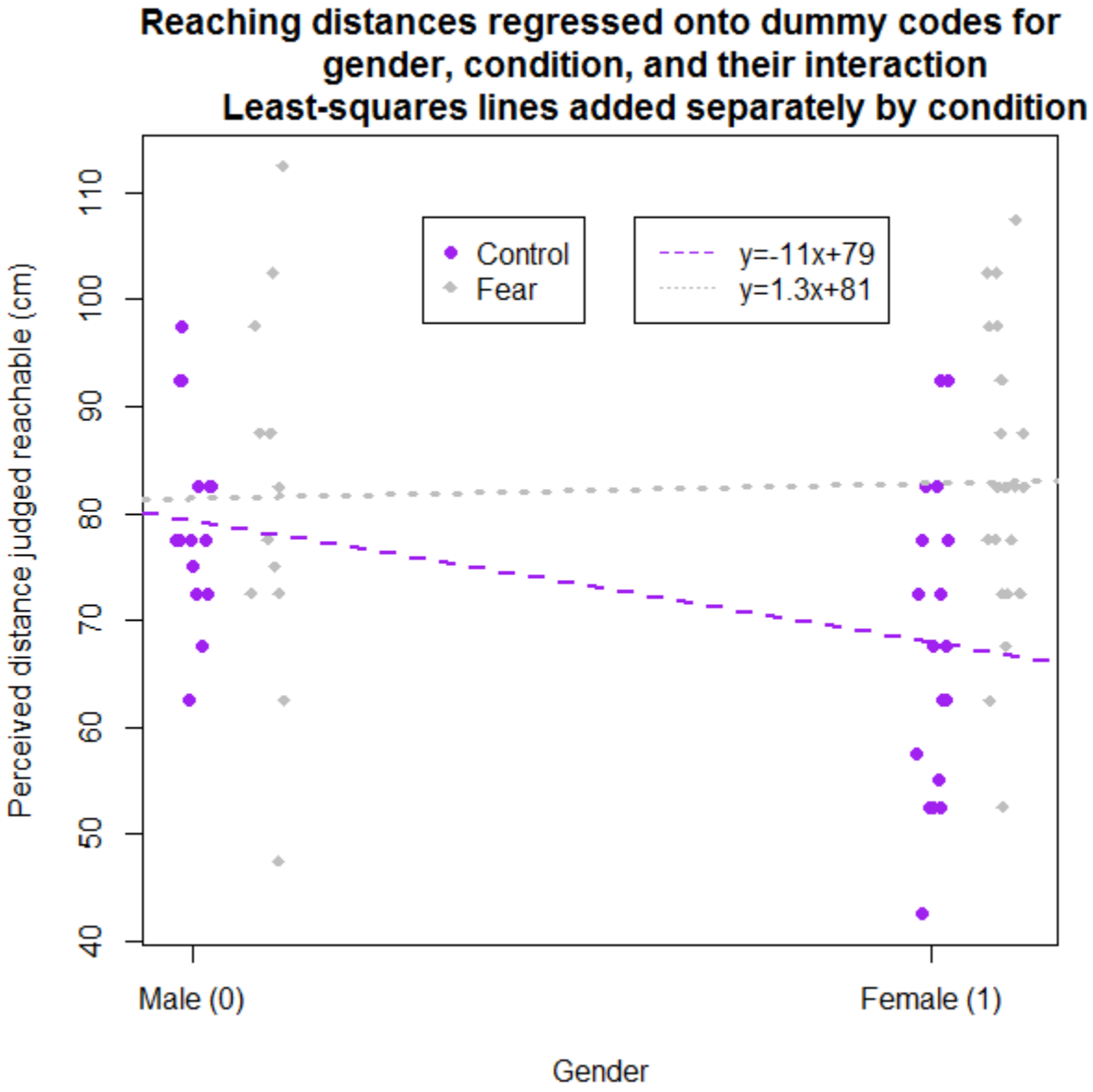
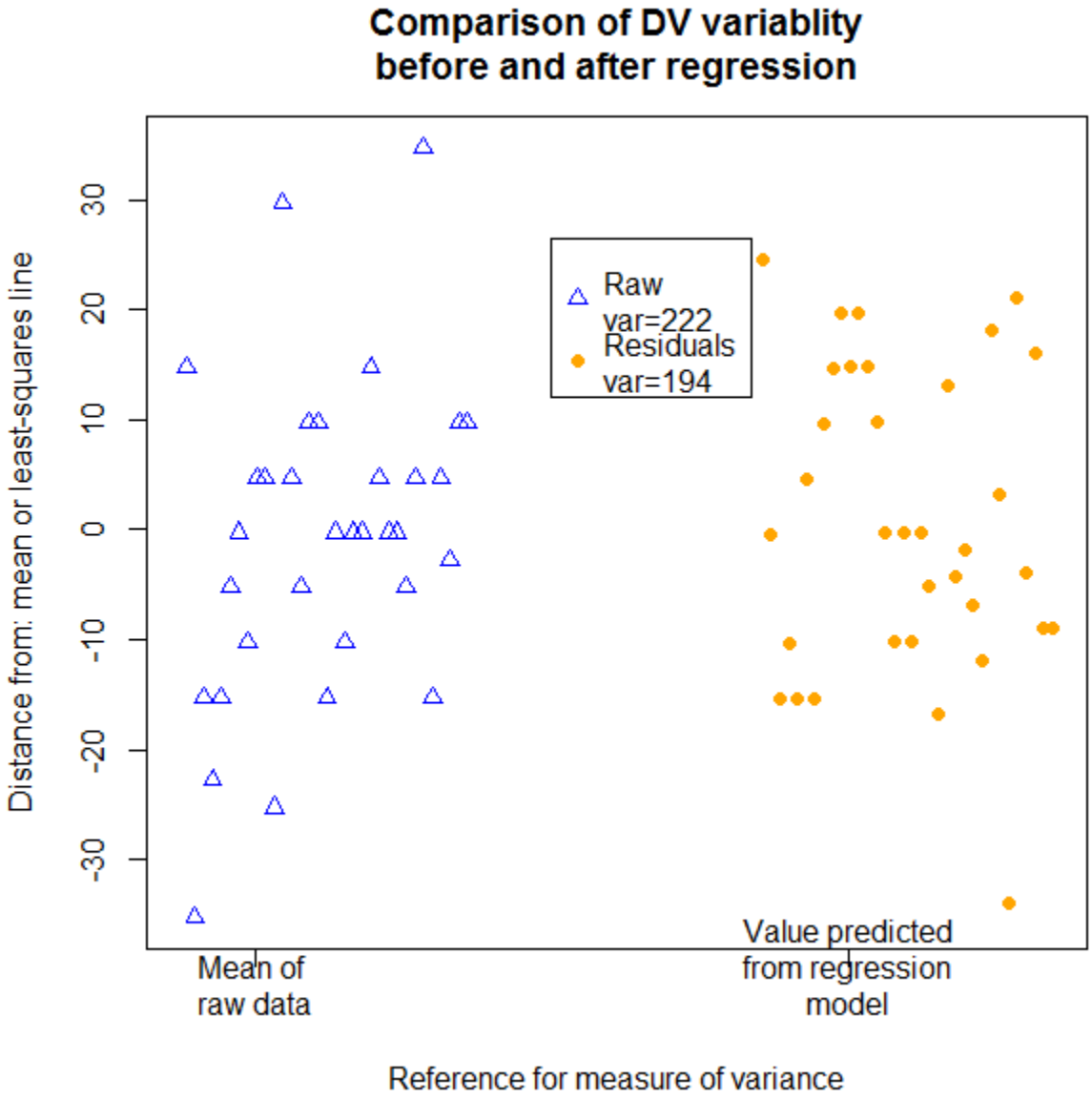

Reaching distances regressed onto dummy codes for gender, condition, and their interaction
Least-squares lines added separately by condition
Perceived distance judged reachable (cm)
40
50
60
70
80
90
100
110
Control
Fear
y=-11x+79
y=1.3x+81
Male (0)
Female (1)
Gender

Figure 3.


Comparison of DV variablity
before and after regression
Distance from: mean or least-squares line
30
20
10
0
-10
-20
-30
Raw
var=222
Residuals
var=194
Mean of
raw data
Value predicted
from regression
model
Reference for measure of variance

Source Code:

```
################################################################################
###################OLS Regression Using Matrix Operations#####################
################################################################################
#The project will involve:
# reading the file;
# coding the categorical variables;
# arranging data in objects amenable to matrix operations;
# solving the normal equations to obtain the regression parameter estimates;
# calculating standard errors of parameter estimates and conducting hypothesis
#  testing on those parameters;
# visualizing the data and the mathematical operations involved in generating;
#  the output


library("Matrix", lib.loc="C:/Program Files/R/R-3.3.2/library")
library("foreign", lib.loc="C:/Program Files/R/R-3.3.2/library")
library("graphics", lib.loc="C:/Program Files/R/R-3.3.2/library")
library("lattice", lib.loc="C:/Program Files/R/R-3.3.2/library")
library("stats", lib.loc="C:/Program Files/R/R-3.3.2/library")
library("quantreg", lib.loc="~/R/win-library/3.3")
library("plyr", lib.loc="~/R/win-library/3.3")
library("matlib", lib.loc="~/R/win-library/3.3")
library("plyr", lib.loc="~/R/win-library/3.3")
library("knitr", lib.loc="~/R/win-library/3.3")
################################Loading the data#################################


pathString<-paste("C:/Users/Tofu/Documents/College/Spring 2017/Linear Algebra/",
                  "OLS Project/HW5 - Categorical Interactions (1).csv",
                  sep="")

rawReachingData=as.data.frame(read.csv(pathString))



##############################Organize as Matrix################################
distancesVector=as.vector(rawReachingData[,3])

# Recode categorical variables into dummy codes.
#################  Gender: 0=male, 1=female ##################
#################  Condition: 0=control, 1=fear ##############

sampleN=length(distancesVector)

femaleVector = vector(, sampleN)
```

```
#female coding
for (i in 1:sampleN)
{
 if (rawReachingData[i,2] == "Female")
   {
    femaleVector[i]=1
   }
 else
 {
  femaleVector[i]=0
 }
}

fearVector = vector(, sampleN)

#Condition coding
for (i in 1:sampleN)
{
 if (rawReachingData[i,1] == "Fear")
 {
  fearVector[i]=1
 }
 else
 {
  fearVector[i]=0
 }
}


constantVector = as.vector(matrix(1,nrow=sampleN,ncol=1))

interactionVector = as.vector(matrix(0,nrow=sampleN, ncol=1))

for (i in 1:sampleN)
{
 if (fearVector[i] == 1 && femaleVector[i] == 1)
 {
  interactionVector[i]=1
 }
}

#combine the IV vectors and a constant into the Design Matrix
# Design Matrix X: X'XB=X'y. Note the apostrophe denotes the transpose operator

designMatrix=cbind(constantVector,fearVector,femaleVector, interactionVector)


############################Solve the normal equations#############################
```

```
# B=inv(X'X)X'y. The elements in B are the parameter estimates

parameterVector = as.data.frame(c(0,0,0,0), row.names = c("Intercept",
                                        "Fear", "Female",
                                        "Interaction"))

for (i in 1:4)
{
  parameterVector[i,1] =
    (inv(t(designMatrix) %*% designMatrix) %*%
      (t(designMatrix)%*%distancesVector))[i,1]
}

###################Find standard errors of paremeter estimates###################
I67X67 = diag(67)

modelResidualMaker = I67X67 - (designMatrix %*% inv(t(designMatrix) %*%
                                     designMatrix) %*%
                         t(designMatrix))

distancesOnModelResiduals = modelResidualMaker %*% distancesVector

# SS residuals
SSresid = t(distancesOnModelResiduals)%*%distancesOnModelResiduals

# Variance of residuals
VarResid=SSresid/(sampleN-4)

#Method for finding standard errors of the coefficients
fearMean = mean(designMatrix[,2])
femaleMean = mean(designMatrix[,3])
interactionMean=mean(designMatrix[,4])

fearMeanVect = as.vector(matrix(fearMean, nrow=sampleN, ncol=1))
femaleMeanVect = as.vector(matrix(femaleMean, nrow=sampleN, ncol=1))
interactionMeanVect = as.vector(matrix(interactionMean, nrow=sampleN, ncol=1))

meanCenteredDesignMaker = cbind(fearMeanVect, femaleMeanVect,
                          interactionMeanVect)

centeredDesignMatrix=cbind(fearVector, femaleVector, interactionVector) -
                       meanCenteredDesignMaker

invDesignDotProd = inv(t(centeredDesignMatrix) %*% centeredDesignMatrix)

coefficientVarCov = VarResid[1,1] * invDesignDotProd

coefficientSEs = sqrt(diag(coefficientVarCov))
```

```
#Calculate significance values for t-scores of the parameters
tScoreFear=parameterVector[2,1]/coefficientSEs[1]
tScoreFemale=parameterVector[3,1]/coefficientSEs[2]
tScoreIntrx=parameterVector[4,1]/coefficientSEs[3]

# Two-tailed probabilities
significanceFear=pt(abs(tScoreFear), df=sampleN-4, lower.tail = FALSE)*2
significanceFemale=pt(abs(tScoreFemale), df=sampleN-4, lower.tail = FALSE)*2
significanceIntrx=pt(abs(tScoreIntrx), df=sampleN-4, lower.tail = FALSE)*2

#Confidence intervals
CImultiplier = qt(.975, df=63)
fearCI=c(parameterVector[2,1]-(CImultiplier*coefficientSEs[1]),
      parameterVector[2,1]+(CImultiplier*coefficientSEs[1]))

femaleCI=c(parameterVector[3,1]-(CImultiplier*coefficientSEs[2]),
      parameterVector[3,1]+(CImultiplier*coefficientSEs[2]))

intrxCI=c(parameterVector[4,1]-(CImultiplier*coefficientSEs[3]),
      parameterVector[4,1]+(CImultiplier*coefficientSEs[3]))

# Total Model Fit
centeredDistancesVector=distancesVector-mean(distancesVector)
SStotal=t(centeredDistancesVector)%*%centeredDistancesVector
SSregression=SStotal-SSresid

RSquare=SSregression/SStotal

modelRSqr=2488.128/14869.179
MSbetween=2488.128/3
MSwithin=SSresid/(sampleN-4)
fRatio=MSbetween/MSwithin
significanceModel=pf(fRatio, df1=3, df2=(sampleN-4), lower.tail = FALSE)

##########################Graph the least squares lines##########################

#SET UP for Distances vs Condition

# Jitter the x-variable

jitterers = sample(seq(from = -.025, to = .025, by = (.05/99) ), sampleN)
fearJittered = fearVector+jitterers
for (i in 1:sampleN)
{
 if (femaleVector[i] == 1)
 {
  fearJittered[i]=fearJittered[i]+.1
```

```
  }
}

# Intercept and slope for female=0
slopeFem0=2.124993
interceptFem0=79.337

# Intercept and slope for female=1
slopeFem1=slopeFem0+12.685628
interceptFem1=interceptFem0-11.416677


# Make different point characters for males and females

genderPtChars = as.vector(matrix(0,nrow=sampleN, ncol=1))
for (i in 1:sampleN)
{
  if (femaleVector[i]==0)
  {
    genderPtChars[i]=16
  }
  else
  {
    genderPtChars[i]=18
  }
}

genderPtCols = as.vector(matrix(0,nrow=sampleN, ncol=1))
for (i in 1:sampleN)
{
  if (femaleVector[i]==0)
  {
    genderPtCols[i]="red"
  }
  else
  {
    genderPtCols[i]="green"
  }
}

# GRAPHING

dev.new(width=5,height=5)
plot (x=fearJittered, y=distancesVector,
      pch=t(genderPtChars),
      col=t(genderPtCols),
      title(main="Reaching distances regressed onto dummy codes for
condition, gender, and their interaction
```

```
Least-squares lines added separately by gender"),
     ylab="Perceived distance judged reachable (cm)",
     xlab="Experimental Condition",
     xaxt="n")
axis(1, at=c(0,1), labels=c("Control (0)", "Fear (1)"))

abline(b=slopeFem0, a=interceptFem0, lty=2, lwd=2, col="red")
abline(b=slopeFem1, a=interceptFem1, lty=3, lwd=3, col="green")

legend(locator(1), legend=c("Male", "Female"), col=c("red", "green"),
      pch=c(16,18))

legend(locator(1), legend=c("y=2.1x+79", "y=15x+68"), col=c("red", "green"),
      lty=c(2,3))

######LOOK HERE! YES HERE, AT THE EDITOR
######CLICK THE PLOT TO ADD THE LEGENDS


#SET UP for Distances vs Gender

# Jitter the x-variable

femaleJittered = femaleVector+jitterers

#Distinguish the control from the fear participants
for (i in 1:sampleN)
{
  if (fearVector[i] == 1)
  {
    femaleJittered[i]=femaleJittered[i]+.1
  }
}

# Intercept and slope for fear=0
slopeFear0=-11.4
interceptFear0=79.337

# Intercept and slope for fear=1
slopeFear1=slopeFear0+12.685628
interceptFear1=interceptFear0+2.12


# Make different point characters for control and fear

conditionPtChars = as.vector(matrix(0,nrow=sampleN, ncol=1))
for (i in 1:sampleN)
{
```

```
 if (fearVector[i]==0)
 {
  conditionPtChars[i]=16
 }
 else
 {
  conditionPtChars[i]=18
 }
}

conditionPtCols = as.vector(matrix(0,nrow=sampleN, ncol=1))
for (i in 1:sampleN)
{
 if (fearVector[i]==0)
 {
  conditionPtCols[i]="purple"
 }
 else
 {
  conditionPtCols[i]="gray"
 }
}

# GRAPHING

dev.new(width=5,height=5)
plot (x=femaleJittered, y=distancesVector,
    pch=t(conditionPtChars),
    col=t(conditionPtCols),
    title(main="Reaching distances regressed onto dummy codes for
        gender, condition, and their interaction
        Least-squares lines added separately by condition"),
    ylab="Perceived distance judged reachable (cm)",
    xlab="Gender",
    xaxt="n")
axis(1, at=c(0,1), labels=c("Male (0)", "Female (1)"))

abline(b=slopeFear0, a=interceptFear0, lty=2, lwd=2, col="purple")
abline(b=slopeFear1, a=interceptFear1, lty=3, lwd=3, col="gray")

legend(locator(1), legend=c("Control", "Fear"), col=c("purple", "gray"),
     pch=c(16,18))

legend(locator(1), legend=c("y=-11x+79", "y=1.3x+81"), col=c("purple", "gray"),
     lty=c(2,3))

######LOOK HERE! YES HERE, AT THE EDITOR
######CLICK THE PLOT TO ADD THE LEGENDS
```

```
########################Graph the raw data and residuals#######################
prePostX = c(as.vector(matrix(data=seq(from=0, to=1, by=1/66), nrow=sampleN, ncol=1)),
        as.vector(matrix(data=seq(from=2, to=3, by=1/66), nrow=sampleN,
                ncol=1)))

prePostY=c((distancesVector-mean(distancesVector)), distancesOnModelResiduals)

prePostPtChars = as.vector(matrix(0,nrow=sampleN*2, ncol=1))
for (i in 1:sampleN*2)
{
 if (i <= sampleN)
 {
  prePostPtChars[i]=2
 }
 else
 {
  prePostPtChars[i]=16
 }
}

prePostPtCols = as.vector(matrix(0,nrow=sampleN*2, ncol=1))
for (i in 1:sampleN*2)
{
 if (i <= sampleN)
 {
  prePostPtCols[i]="blue"
 }
 else
 {
  prePostPtCols[i]="orange"
 }
}

dev.new(width=5, height=5)
plot(prePostX, prePostY,
    pch=t(prePostPtChars),
    col=t(prePostPtCols),
    title(main="Comparison of DV variablity
before and after regression"),
    ylab="Distance from: mean or least-squares line",
    xlab="Reference for measure of variance",
    xaxt="n")
axis(1, at=c(.25,2.3), labels=c("Mean of
raw data", "Value predicted
from regression
model"))
```

```
legend(locator(1), legend=c("Raw
var=222", "Residuals
var=194"), col=c("blue", "orange"),
    pch=c(2,16))
```