PSYCH308D - Data Analysis (DA03)

Brady C. Jackson

2025/04/19

Table of Contents

# 1 Libraries

Load all requisite libraries here.

# Load packages. Set messages and warnings to FALSE so I don't have to see the  
# masking messages in the output.  
library(jmv) # for descriptive  
library(ggplot2)  
library(dplyr)  
library(corrplot) # For fancy covariance matrix plots  
library(apaTables) # For Word formatted tables  
library(car) # for ncvTest (Breusch Pagan)  
library(tidyverse)  
library(jmv) # for descriptives  
library(ggplot2)  
library(dplyr)  
library(psych)  
library(corrplot) # For fancy covariance matrix plots  
library(car) # for ncvTest (Breusch Pagan)  
library(stringr) # for sub\_str operations  
library(Hmisc) # for fun.dat substitution  
library(see) # for outliers analysis   
library(magrittr)  
library(foreign)  
library(broom)  
library(robmed)  
library(mediation) # For mediation analysis  
library(multilevel)  
library(GGally)  
library(lsr)  
library(car)  
library(mvnTest) # Multivariate Normality  
library(lm.beta)  
library(lavaan) # Structural Equation Modeling  
library(haven)  
library(foreign)  
library(parallel)  
# library(AER)  
library(janitor) # Data cleaning  
library(naniar) # Data cleaning  
library(performance) # Data cleaning  
library(mice) # Data cleaning

# 2 Metadata

This section of code is to setup some general variables that we’ll use throughout the code (e.g. figure colors, etc)

# First we'll defines some meta-data to use in all of our plots so they're nice and clean  
font\_color = "#4F81BD"  
grid\_color\_major = "#9BB7D9"  
grid\_color\_minor = "#C8D7EA"  
back\_color = "gray95"  
rb\_colmap = colorRampPalette( c("firebrick", "grey86", "dodgerblue3") )(200)  
  
# I'm going to try to save off my preferred ggplot theme combinations as a unqiue theme object that I can just reference  
# later in the code....totally unclear if ggplot works this way....  
my\_gg\_theme = theme\_minimal() +  
 theme( plot.title = element\_text(size = 12, face = "italic", color = font\_color),  
 axis.title.x = element\_text(color = font\_color),  
 axis.title.y = element\_text(color = font\_color),  
 axis.text.x = element\_text(color = font\_color),  
 axis.text.y = element\_text(color = font\_color),  
 legend.title = element\_text(color = font\_color),  
 legend.text = element\_text(color = font\_color),  
 panel.grid.minor = element\_line(color = grid\_color\_minor),  
 panel.grid.major = element\_line(color = grid\_color\_major),  
 panel.background = element\_rect(fill = back\_color, color = font\_color)  
 )

# 3 Part 1: Data Cleaning

Download the dataset posted on canvas called “308A.DA3.Data.csv” and create an RMarkdown file. This DA consists of three categories of tasks for you to complete – data cleaning (complete in RStudio), data querying (complete in RStudio and respond to questions below), and a code investigation (respond below). Upload both a word document with your completed questions and your knitted RMarkdown file in either word or pdf format.

The dataset contains data regarding average grades for Exam 1 and 2 for various classes, each case is classified by school level (elem, midd, high), subject, year, and location.

## 3.1 Load the Data

# Load the assignment data from CSV  
raw\_dat = read.csv("./308D.DA3.Data.csv")  
  
# Rename columns to lower because why not  
colnames(raw\_dat) <- tolower( colnames(raw\_dat) )  
  
# Ensure that the numbers of each subject in the study are unique to prevent any duplicate data  
# if the size of the unique-entries only is the same as the whole vector then there are no duplicate subjects  
# NOTE: This fails if the colname of the subject ID is input wrong. So make sure you UPDATE the "test\_col"   
# entry below  
test\_colname = "x"  
  
test\_unique = ( length( unique( raw\_dat[test\_colname] ) ) == length(raw\_dat[test\_colname]))  
if(!test\_unique){  
 print("WARNING: There are duplicate data entries in the raw data")  
}else{  
 print("No duplicate entries detected in raw data")  
}

## [1] "No duplicate entries detected in raw data"

## 3.2 Handle Missing Data for all Variables

## 3.3 Detect Outliers and Handle Accordingly.

## 3.4 Convert Categorical Variables

Convert School Level, Subject, Year, and Location to categorical variables

## 3.5 Rename the Variables “exam1” and “exam2”

## 3.6 Check the Alpha for “exam1” and “exam2”

Check the Alpha for “exam1” and “exam2” to see if we can make a composite score.

## 3.7 Combine Exam Grades for Each Classes

Create 1 variable for exam grade for each class (average of the two)

## 3.8 Reorder the Columns

Reorder the Columns so all categories (level, subject, year, location) are listed first,  
followed by Interpersonal, Exam 1, Exam 2, and average Exam

## 3.9 Construct Reverse Codes

There was an error in qualtrics and the scores for Interpersonal skills were not set up with  
reverse coding. Reverse code the Interpersonal scores using R.

## 3.10 Standardize the Exam and Interpersonal Scores

Standardize the Exam and Interpersonal Scores for ease of comparison.

## 3.11 Dummy Code Location

Dummy Code the location variable with CA as the reference group.

# 4 Part 2: Queries

## 4.1 What is the average overall grade for each level of school?

## 4.2 What is the average exam 2 grade for math classes?

## 4.3 Calculate the overall average exam grade for all classes.

## 4.4 Create a new data frame with only classes from CA.

What is the average exam 1 score?