Project Data

Jed Green

2024-01-18

This my attempt at an R Markdown script. I hope this will make the data more reproducible.

Below is a list of all the packages I used:

library(haven)  
library(gmodels)  
library(rcompanion)

## Registered S3 method overwritten by 'DescTools':  
## method from   
## reorder.factor gdata

library(chisq.posthoc.test)  
library(stargazer)

##   
## Please cite as:

## Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.

## R package version 5.2.3. https://CRAN.R-project.org/package=stargazer

library(ggplot2)  
library(tidyverse)

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --  
## v dplyr 1.1.2 v readr 2.1.4  
## v forcats 1.0.0 v stringr 1.5.1  
## v lubridate 1.9.2 v tibble 3.2.1  
## v purrr 1.0.1 v tidyr 1.3.0

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## i Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors

library(xtable)  
library(AICcmodavg)  
library(rmarkdown)  
library(tinytex)

As all of our data comes from the longitudinal C19PRC\_Uk survey our first step is to merge all of the waves together. The first step is to read in all the different .sav files containing all of the data from the different waves. We did this using “read\_sav” from the package Haven as R cannot naturally read in .sav files. The second step was to extract only the variables that are necessary as the data set would be too large if we merged everything. The two variables from each subsequent data set that We selected was “PID” and the “Wave type”. This is because the “PID” variable allows us to merge the data based on individuals as each respondent has a unique “PID”. In addition we needed the “Wave type” variable from each wave in order to identify who was a recontact and who was a respondent. Finally we merged the data by the “PID” variable in order to get all of our Data into one data set.

### Step 1: Reading in all databases:  
  
df<-read\_sav("C19PRC\_UKW1W2\_archive\_final.sav")  
df1<-read\_sav("C19PRC\_UK\_W3\_archive\_final.sav")  
df2<-read\_sav("C19PRC\_UK\_W4\_archive\_final.sav")  
df3<-read\_sav("C19PRC\_UKW5\_archive\_final.sav")  
df4<-read\_sav("C19PRC\_UK\_W6\_archive\_final.sav")  
  
### Step 2: Making the databases contain only variables needed:  
  
ldf1<-as.data.frame(cbind(df1$pid,df1$W3\_Type))  
colnames(ldf1)<-c("pid","W3\_Type")  
  
ldf2<-as.data.frame(cbind(df2$pid,df2$W4\_Type))  
colnames(ldf2)<-c("pid","W4\_Type")  
  
ldf3<-as.data.frame(cbind(df3$pid,df3$W5\_Type))  
colnames(ldf3)<-c("pid","W5\_Type")  
  
ldf4<-as.data.frame(cbind(df4$pid,df4$W6\_Type))  
colnames(ldf4)<-c("pid","W6\_Type")  
  
  
### Step 3: Merging data sets:  
  
mdf1<-merge.data.frame(df,ldf1, by ="pid", all = T)  
mdf2<-merge.data.frame(mdf1,ldf2, by ="pid", all = T)  
mdf3<-merge.data.frame(mdf2,ldf3, by ="pid", all = T)  
mdf4<-merge.data.frame(mdf3,ldf4, by ="pid", all = T)

The next step now that we have merged all of our data together is insert values for the missing data. As both the “Present” and “Wave Type” variable are unique to each wave to creates a lot NA’s when we merge them together. Therefore, we replaced all the NA’s with -99 values. This allows for them to be represented in our analysis.

mdf4$W1\_Present[is.na(mdf4$W1\_Present)]<--99  
mdf4$W2\_Present[is.na(mdf4$W2\_Present)]<--99  
mdf4$W3\_Type[is.na(mdf4$W3\_Type)]<--99  
mdf4$W4\_Type[is.na(mdf4$W4\_Type)]<--99  
mdf4$W5\_Type[is.na(mdf4$W5\_Type)]<--99  
mdf4$W6\_Type[is.na(mdf4$W6\_Type)]<--99

Now we have edited our data we need to tidy and clean it. First lets create a new variable that separates the PHQ-9 variable into the different depression severity levels. This will go on to form our dependent variable.

### creating Depression severity variable  
  
mdf4$W1\_Dep\_Severity <- NA  
mdf4$W1\_Dep\_Severity[mdf4$W1\_Depression\_Total >= 0 &mdf4$W1\_Depression\_Total <= 4 ] <- "None minimal"  
mdf4$W1\_Dep\_Severity[mdf4$W1\_Depression\_Total >= 5 & mdf4$W1\_Depression\_Total <= 9 ] <- "Mild"  
mdf4$W1\_Dep\_Severity[mdf4$W1\_Depression\_Total >= 10 & mdf4$W1\_Depression\_Total <= 14] <- "Moderate"  
mdf4$W1\_Dep\_Severity[mdf4$W1\_Depression\_Total >= 15 & mdf4$W1\_Depression\_Total <= 19] <- "Moderately Severe"  
mdf4$W1\_Dep\_Severity[mdf4$W1\_Depression\_Total >= 20 & mdf4$W1\_Depression\_Total <= 27] <- "Severe"

Next lets create some variables to identify how many people attended each wave consecutively. In order to do this we used the wave type and present variable for each wave. For the first two waves there were no top-up respondents and therefore we can use the W1\_present and W2\_present variables in order to calculate the number of people who were present in wave 1 and then attend wave 2. For subsequent waves it gets more complex as top ups are introduced. Therefore from wave 3 onwards we use each individual wave type variable to identify if they are a recontact or a top up respondent. For all waves (except wave 3) if the Wave type variable is = 1 then it means that they were a recontact (for wave 3 if they were a recontact wave3 type = 0).

### Re-coding variable: answering waves one after another in order:  
mdf4$presentwave1<- ifelse(mdf4$W1\_Present == 1,1,0)  
mdf4$presentwave1\_2 <- ifelse(mdf4$W1\_Present == 1 & mdf4$W2\_Present == 1,1,0)  
mdf4$presentwave1\_2\_3 <-ifelse(mdf4$W1\_Present == 1 & mdf4$W2\_Present == 1 & mdf4$W3\_Type == 0,1,0)  
mdf4$presentwave1\_2\_3\_4<-ifelse(mdf4$W1\_Present == 1 & mdf4$W2\_Present == 1 & mdf4$W3\_Type == 0 & mdf4$W4\_Type == 1,1,0)  
mdf4$presentwave1\_2\_3\_4\_5<-ifelse(mdf4$W1\_Present == 1 & mdf4$W2\_Present == 1 & mdf4$W3\_Type == 0 & mdf4$W4\_Type == 1 & mdf4$W5\_Type == 1,1,0)  
mdf4$presentwave1\_2\_3\_4\_5\_6<-ifelse(mdf4$W1\_Present == 1 & mdf4$W2\_Present == 1 & mdf4$W3\_Type == 0 & mdf4$W4\_Type   
 == 1 & mdf4$W5\_Type == 1 & mdf4$W6\_Type == 1,1,0)

Now we have created these new variables we can now create tables out them. The sum of each table is equal to the total number of contacts across all six waves (both Top-Ups and Recontacts) which is 5364. The results under 0 in each table equal to the total number of Top-Ups and those who drop out at each stage of the survey process. The results under 1 in each table is the total number of people who completed each wave up until and including the last one listed. Therefore, if we were to subtract the total number of Top-Up contacts (3339) from the 0 value in each table we would be left with the number of drop outs between each wave - and this is what we find.

### Printing each variable in a table  
table(mdf4$presentwave1)

##   
## 0 1   
## 3339 2025

table(mdf4$presentwave1\_2)

##   
## 0 1   
## 3958 1406

table(mdf4$presentwave1\_2\_3)

##   
## 0 1   
## 4414 950

table(mdf4$presentwave1\_2\_3\_4)

##   
## 0 1   
## 4593 771

table(mdf4$presentwave1\_2\_3\_4\_5)

##   
## 0 1   
## 4687 677

table(mdf4$presentwave1\_2\_3\_4\_5\_6)

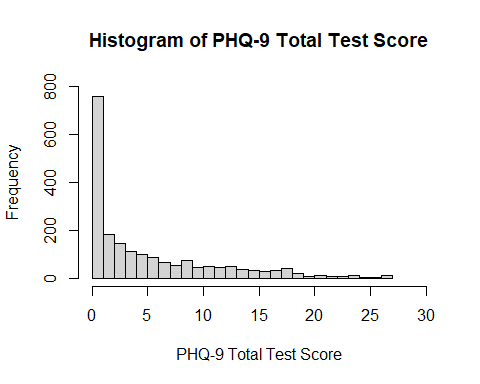
##   
## 0 1   
## 4782 582

Now we can input the values generated from this filtering into its own matrix. We did this by running each variable and inputting each of the successful test cases (ie. output = 1) into a matrix. This matrix was then converted into a data frame. This data frame represents the number of people who attended each wave without missing one.

Waves\_attended\_consecutively <- matrix(c(2025,1406, 950, 771, 677, 582), ncol=1, byrow=TRUE)  
rownames(Waves\_attended\_consecutively) <- c('Wave 1','Wave 2','Wave 3','Wave 4','Wave 5','Wave 6')  
colnames(Waves\_attended\_consecutively) <- c('Present')  
Waves\_attended\_consecutivelydf<-as.data.frame(Waves\_attended\_consecutively)  
Waves\_attended\_consecutivelydf$Waves <- rownames(Waves\_attended\_consecutivelydf)

Now that we have made collated and cleaned our data we can now create some initial descriptive statistics. Lets first explore the distribution of depression amount wave 1 respondents. Below we have two graphs, both use the PHQ-9 variable. The first is a scale 1-27 the second is the distribution split into the different severity levels.

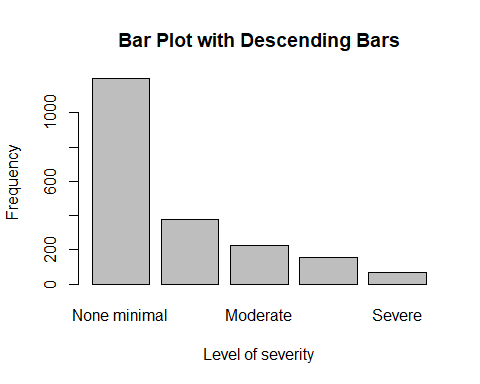
# Graph 1: Depression PHQ-9 scale  
  
hist(df$W1\_Depression\_Total,  
 breaks = 30,  
 xlim = c(0,30),  
 ylim = c(0,800),  
 xlab = "PHQ-9 Total Test Score",  
 main = "Histogram of PHQ-9 Total Test Score")



#Graph 2: Depression severity   
  
table(mdf4$W1\_Dep\_Severity)

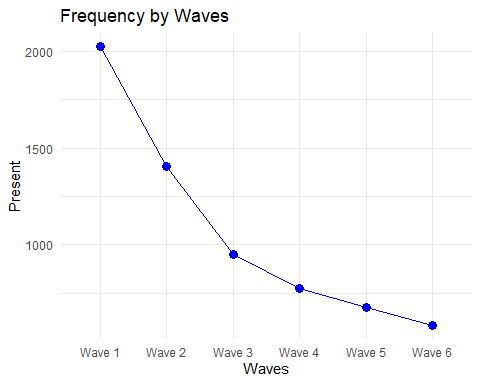
##   
## Mild Moderate Moderately Severe None minimal   
## 378 227 154 1199   
## Severe   
## 67

data <- c(378, 227, 154, 1199, 67)  
categories <- c("Mild", "Moderate", "Moderately Severe", "None minimal", "Severe")  
  
# Create a data frame  
BP1 <- data.frame(Category = categories, Count = data)  
  
# Sort the data frame by Count in descending order  
BP1 <- BP1[order(-BP1$Count), ]  
  
# Create the bar plot  
barplot(BP1$Count, names.arg = BP1$Category, main = "Bar Plot with Descending Bars",  
 xlab = "Level of severity", ylab = "Frequency")



Graph 3 below shows the attrition rate of between waves. Reminder that this represents those who did not complete the waves in order

### Graph 3: Attrition between waves:  
  
ggplot(Waves\_attended\_consecutivelydf, aes(x = Waves, y = Present, group = 1)) +  
 geom\_line(color = "blue") +  
 geom\_point(color = "blue", size = 3) +  
 labs(title = "Frequency by Waves", x = "Waves", y = "Present") +  
 theme\_minimal()



Our next step now that we have conducted some initial descriptive statistics is to compare the attrition rate between the different levels of depression severity. In order to do that we need to calculate the figures. We did this by creating tables showing the respondents who were were present at each wave (and present at all subsequent waves) and their wave 1 depression severity level. For each of the tables below values under the “0” Row represent all those who had dropped out/missed a survey. All values under the “1” row represent all respondents who have completed every survey up to and including the last number listed in the variable. This allows us to see all those who completed all surveys and their given depression severity. Finally the tables also have total columns so that we ensure that “0” and “1” rows add up to the total number contacts at wave one (2025) as this is the baseline from which we are measuring attrition.

### Depression and wave attrition tables  
addmargins(table(mdf4$presentwave1, mdf4$W1\_Dep\_Severity),2)

##   
## Mild Moderate Moderately Severe None minimal Severe Sum  
## 0 0 0 0 0 0 0  
## 1 378 227 154 1199 67 2025

addmargins(table(mdf4$presentwave1\_2, mdf4$W1\_Dep\_Severity),2)

##   
## Mild Moderate Moderately Severe None minimal Severe Sum  
## 0 135 89 59 303 33 619  
## 1 243 138 95 896 34 1406

addmargins(table(mdf4$presentwave1\_2\_3, mdf4$W1\_Dep\_Severity),2)

##   
## Mild Moderate Moderately Severe None minimal Severe Sum  
## 0 215 149 95 562 54 1075  
## 1 163 78 59 637 13 950

addmargins(table(mdf4$presentwave1\_2\_3\_4, mdf4$W1\_Dep\_Severity),2)

##   
## Mild Moderate Moderately Severe None minimal Severe Sum  
## 0 254 172 107 663 58 1254  
## 1 124 55 47 536 9 771

addmargins(table(mdf4$presentwave1\_2\_3\_4\_5,mdf4$W1\_Dep\_Severity),2)

##   
## Mild Moderate Moderately Severe None minimal Severe Sum  
## 0 271 176 117 725 59 1348  
## 1 107 51 37 474 8 677

addmargins(table(mdf4$presentwave1\_2\_3\_4\_5\_6,mdf4$W1\_Dep\_Severity),2)

##   
## Mild Moderate Moderately Severe None minimal Severe Sum  
## 0 287 186 126 785 59 1443  
## 1 91 41 28 414 8 582

As we have calculated all those present at each wave and their depression severity level we can put this information into a data frame so that we can plot it. The figures for each row (as discussed above) are just the figures in the “1” Row i.e have completed every survey up to and including the last number listed in the variable.

### Creating the data frame from the table  
  
wave\_data <- data.frame(  
 Severity = c("None minimal (0-4)", "Mild (5-9)", "Moderate (10-14)", "Moderately Severe (15-19)", "Severe (20-27)"),  
 Wave\_1 = c(1199, 378, 227, 154, 67),  
 Wave\_2 = c(896 , 243, 138, 95, 34),  
 Wave\_3 = c(637, 163, 78, 59, 13),  
 Wave\_4 = c(536, 124, 55, 47, 9),  
 Wave\_5 = c(474, 107, 51, 37, 8),  
 Wave\_6 = c(414, 91, 41, 28, 8))

With the Data frame created above we can now plot this information. In order to do this we used the ggplot2 and the tidyr package. Our First step was to us the pivot\_longer() command which takes our original wide-format data frame wave\_data and converts it into a long-format data frame called wave\_data\_long, where the wave numbers are stacked in a single column named Wave\_Number, and the corresponding frequencies are stacked in another column named Frequency. This transformation allows us to then plot this new data frame.

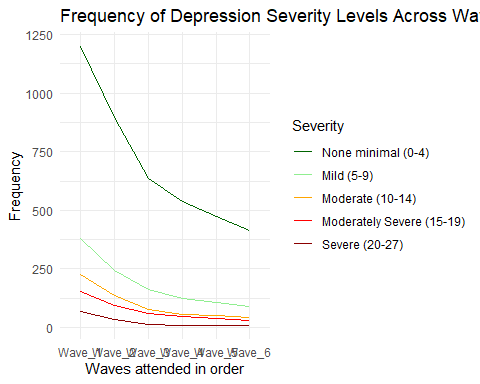
### Converting into a long-format data frame:  
  
wave\_data\_long <- pivot\_longer(wave\_data, cols = -Severity, names\_to = "Wave\_Number", values\_to = "Frequency")

Next we created some values in order to help organise our visualisation code. These where just the colors of each level of depression and the order of the legend.

custom\_colors <- c("None minimal (0-4)" = "darkgreen", "Mild (5-9)"= "lightgreen","Moderate (10-14)" = "orange",  
 "Moderately Severe (15-19)" = "red", "Severe (20-27)" = "darkred")  
legend\_order <- c("None minimal (0-4)", "Mild (5-9)", "Moderate (10-14)", "Moderately Severe (15-19)", "Severe (20-27)")

Below is the code used to create our first results graph. It has the frequency plotted on the Y axis and the number of waves attended in order on the X axis. It is then grouped by Wave 1 depression severity.

### Plotting the line graph for Frequencies:  
  
ggplot(wave\_data\_long, aes(x = Wave\_Number, y = Frequency, color = Severity, group = Severity)) +  
 geom\_line() +  
 scale\_color\_manual(values = custom\_colors, breaks = legend\_order) + # Assigning custom colors  
 labs(title = "Frequency of Depression Severity Levels Across Waves",  
 x = "Waves attended in order", y = "Frequency") +  
 theme\_minimal()

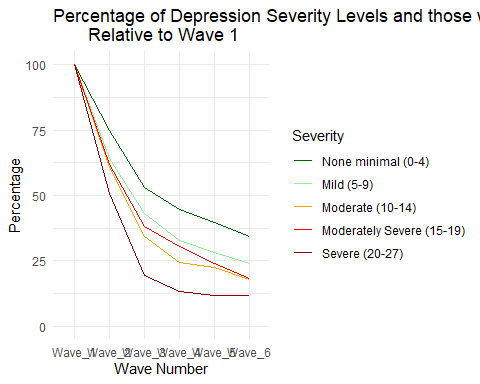


Whilst the graph above is great it might be better to visualise our data reactivate to wave 1. Therefore we need to mutate the frequency column in order to turn it into a percentage.

### Calculating percentages relative to Wave 1 for each severity level:  
  
wave\_data\_long <- wave\_data\_long %>%  
 group\_by(Severity) %>%  
 mutate(Percentage = Frequency / Frequency[Wave\_Number == "Wave\_1"] \* 100)

Now we can re-plot the data but this time have percentage instead of frequency represented on the Y axis.

###Plotting the line graph with percentages relative to Wave 1 on the y-axis:  
  
ggplot(wave\_data\_long, aes(x = Wave\_Number, y = Percentage, color = Severity, group = Severity)) +  
 geom\_line() +  
 scale\_color\_manual(values = custom\_colors, breaks = legend\_order) + # Assigning custom colors and order  
 labs(title = "Percentage of Depression Severity Levels and those who answered consecutive waves   
 Relative to Wave 1",  
 x = "Wave Number", y = "Percentage") +  
 theme\_minimal() +  
 ylim(0,100)



Finally below is the code for an anvoa test using the data from our long-format data frame in order to test for significance. After this we run a Tukey Post-Hoc test in order to compare the means of every treatment to the means of every other treatment.

### ANOVA of above data  
  
two.way<-aov(wave\_data\_long$Percentage~wave\_data\_long$Severity+ wave\_data\_long$Wave\_Number)  
summary(two.way)

## Df Sum Sq Mean Sq F value Pr(>F)   
## wave\_data\_long$Severity 4 1705 426 20.21 8.5e-07 \*\*\*  
## wave\_data\_long$Wave\_Number 5 22890 4578 217.05 < 2e-16 \*\*\*  
## Residuals 20 422 21   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(two.way)

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = wave\_data\_long$Percentage ~ wave\_data\_long$Severity + wave\_data\_long$Wave\_Number)  
##   
## $`wave\_data\_long$Severity`  
## diff lwr upr  
## Moderate (10-14)-Mild (5-9) -5.446783 -13.381194 2.4876279  
## Moderately Severe (15-19)-Mild (5-9) -3.310887 -11.245298 4.6235243  
## None minimal (0-4)-Mild (5-9) 9.004932 1.070521 16.9393430  
## Severe (20-27)-Mild (5-9) -14.188318 -22.122729 -6.2539067  
## Moderately Severe (15-19)-Moderate (10-14) 2.135896 -5.798515 10.0703073  
## None minimal (0-4)-Moderate (10-14) 14.451715 6.517304 22.3861261  
## Severe (20-27)-Moderate (10-14) -8.741535 -16.675946 -0.8071237  
## None minimal (0-4)-Moderately Severe (15-19) 12.315819 4.381408 20.2502297  
## Severe (20-27)-Moderately Severe (15-19) -10.877431 -18.811842 -2.9430201  
## Severe (20-27)-None minimal (0-4) -23.193250 -31.127661 -15.2588388  
## p adj  
## Moderate (10-14)-Mild (5-9) 0.2777075  
## Moderately Severe (15-19)-Mild (5-9) 0.7239564  
## None minimal (0-4)-Mild (5-9) 0.0213671  
## Severe (20-27)-Mild (5-9) 0.0002686  
## Moderately Severe (15-19)-Moderate (10-14) 0.9258618  
## None minimal (0-4)-Moderate (10-14) 0.0002154  
## Severe (20-27)-Moderate (10-14) 0.0264442  
## None minimal (0-4)-Moderately Severe (15-19) 0.0013143  
## Severe (20-27)-Moderately Severe (15-19) 0.0044774  
## Severe (20-27)-None minimal (0-4) 0.0000003  
##   
## $`wave\_data\_long$Wave\_Number`  
## diff lwr upr p adj  
## Wave\_2-Wave\_1 -37.551563 -46.68153 -28.4215948 0.0000000  
## Wave\_3-Wave\_1 -62.334959 -71.46493 -53.2049910 0.0000000  
## Wave\_4-Wave\_1 -70.862091 -79.99206 -61.7321234 0.0000000  
## Wave\_5-Wave\_1 -74.745389 -83.87536 -65.6154212 0.0000000  
## Wave\_6-Wave\_1 -78.642672 -87.77264 -69.5127045 0.0000000  
## Wave\_3-Wave\_2 -24.783396 -33.91336 -15.6534284 0.0000006  
## Wave\_4-Wave\_2 -33.310529 -42.44050 -24.1805608 0.0000000  
## Wave\_5-Wave\_2 -37.193826 -46.32379 -28.0638586 0.0000000  
## Wave\_6-Wave\_2 -41.091110 -50.22108 -31.9611419 0.0000000  
## Wave\_4-Wave\_3 -8.527132 -17.65710 0.6028353 0.0758466  
## Wave\_5-Wave\_3 -12.410430 -21.54040 -3.2804624 0.0043255  
## Wave\_6-Wave\_3 -16.307714 -25.43768 -7.1777457 0.0002166  
## Wave\_5-Wave\_4 -3.883298 -13.01327 5.2466700 0.7618104  
## Wave\_6-Wave\_4 -7.780581 -16.91055 1.3493867 0.1239630  
## Wave\_6-Wave\_5 -3.897283 -13.02725 5.2326845 0.7591805