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)
library(reshape2)

##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

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)
```

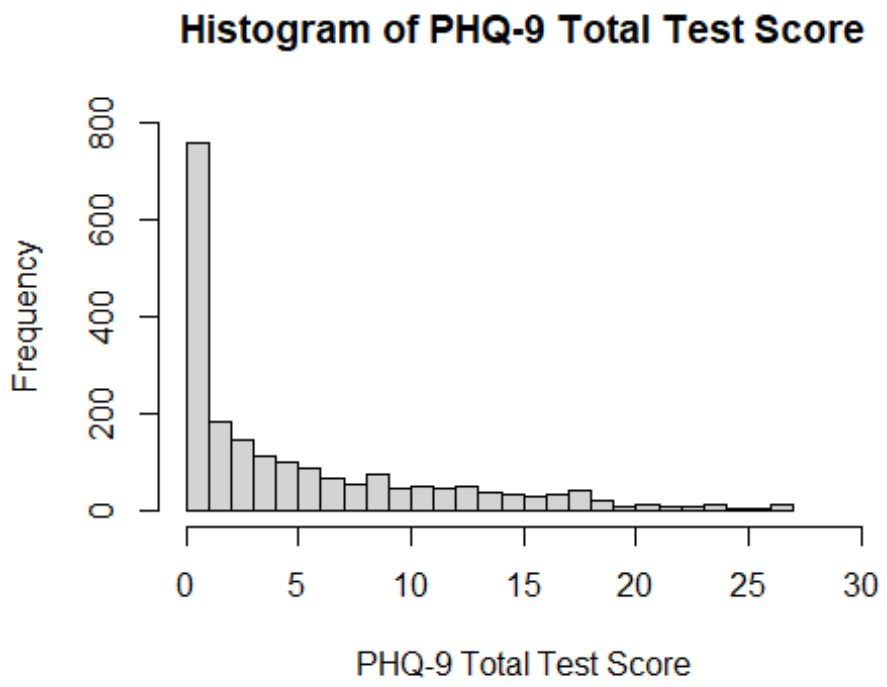
```
##
##    0    1
## 4782  582
```

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.

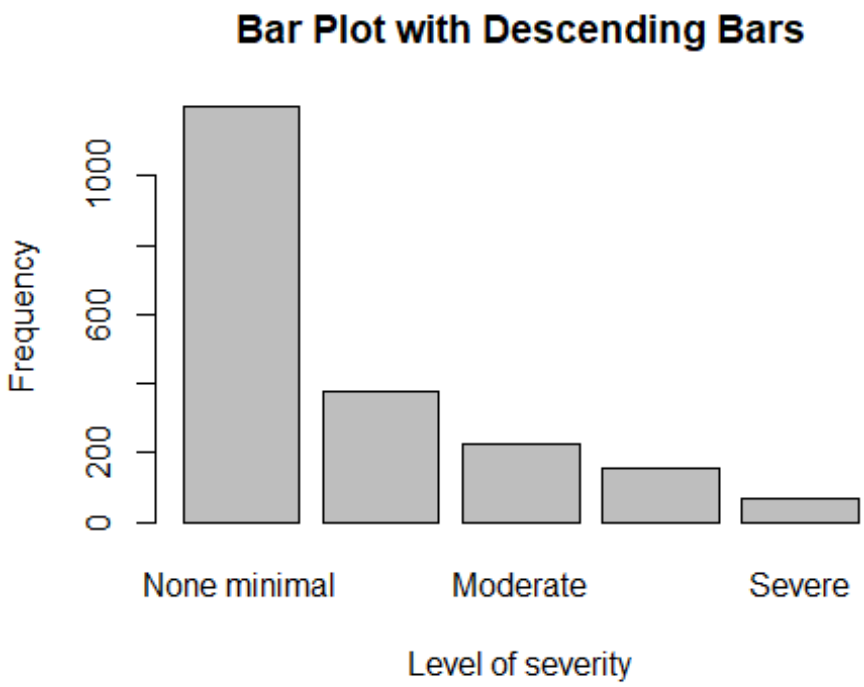
```
Waves_attended_consecutively <- matrix(c(2025,1406, 950, 771, 677, 582),
ncol=1, byrow=TRUE)
rownames(Waves_attended_consecutively) <- c('Wave 1','Wave 2','Wave 3','Wave
4','Wave 5','Wave 6')
colnames(Waves_attended_consecutively) <- c('Present')
Waves_attended_consecutivelydf<-as.data.frame(Waves_attended_consecutively)
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")
```

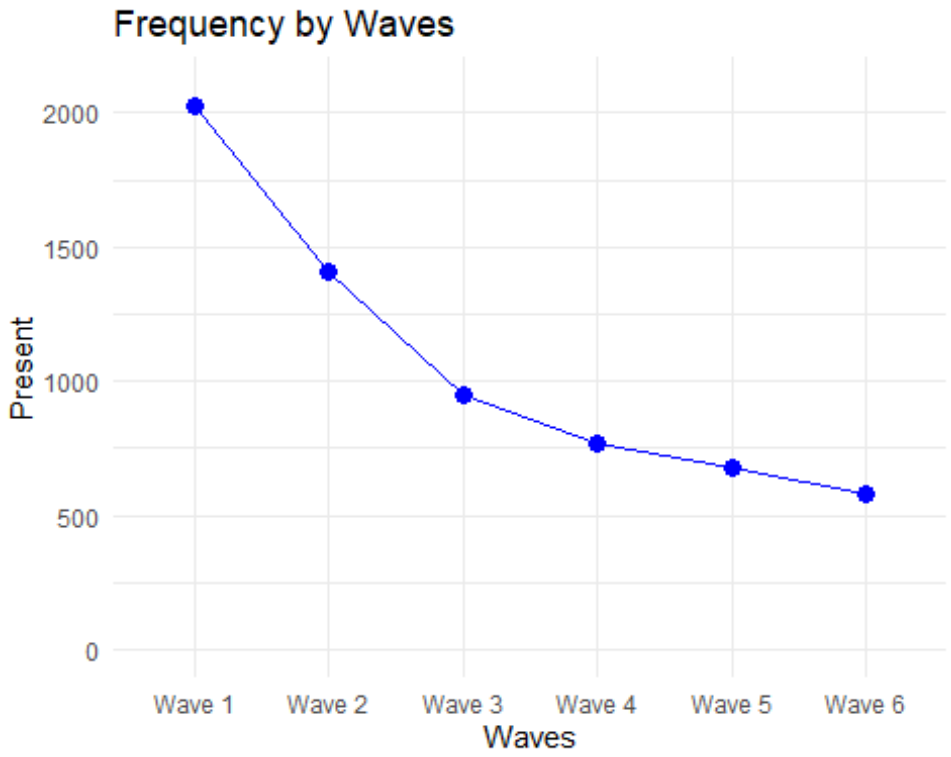



```r
#Graph 2: Depression severity

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() +
  ylim(c(0,2100))
```



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.

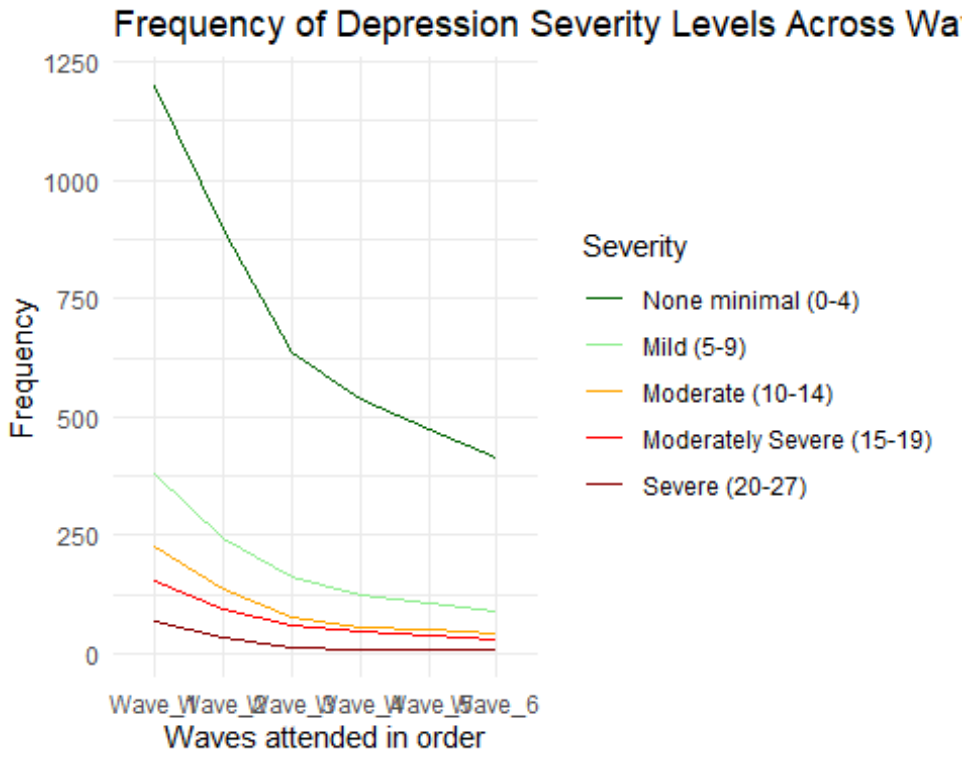
```
### Converting into a long-format data frame:

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()
```

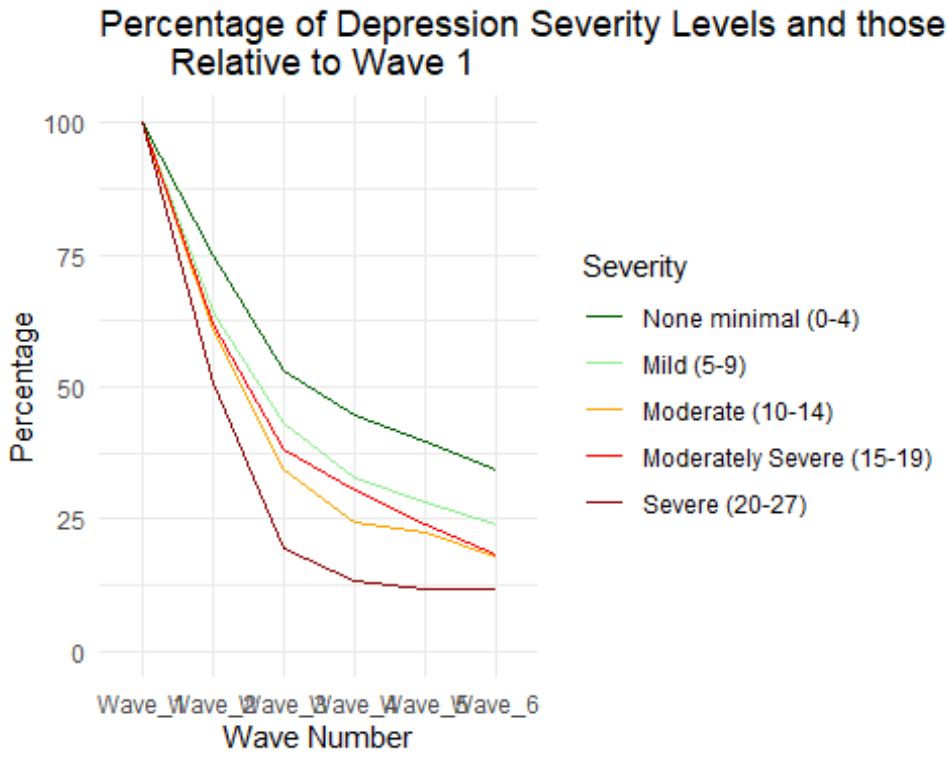


Whilst the graph above is great it might be better to visualise our data reactivate to wave 1. Therefore we need to mutate the frequency column in order to turn it into a percentage.

```
### 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 Lets run a few tests for significance: 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
```

We can also run a simple OlS regression. In order to do this we first need to convert our data frame from a wide format to a long format so that it is conducive to our regression. The wide format of our data is not conducive to regression analysis because each observation (or row) should represent a unique combination of the predictor variables. Once we have done that we also need to convert our depression severity measure from a character variable to a numeric one so that it too is conducive to regression analysis. Finally we run our regression with the counts of respondents at each wave as our dependent and an interaction term between the wave number level of depression severity.

```
# Reshape the data to long format

data_long <- melt(wave_data, id.vars = "Severity", variable.name = "Wave",
value.name = "Counts")

# Convert Severity to numeric
severity_mapping <- c('None minimal (0-4)' = 1, 'Mild (5-9)' = 2, 'Moderate
(10-14)' = 3, 'Moderately Severe (15-19)' = 4, 'Severe (20-27)' = 5)
data_long$Severity_Num <- as.numeric(factor(data_long$Severity, levels =
names(severity_mapping)))



# Perform OLS regression
model <- lm(Counts ~ Wave * Severity_Num, data = data_long)

# Print the summary of the regression model
summary(model)

##
## Call:
## lm(formula = Counts ~ Wave * Severity_Num, data = data_long)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -275.8 -112.7    1.4  115.3  296.4
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1151.40     190.26   6.052 1.01e-05 ***
## WaveWave_2                -308.60     269.06  -1.147 0.266422
## WaveWave_3                -555.80     269.06  -2.066 0.053562 .
## WaveWave_4                -657.90     269.06  -2.445 0.024992 *
## WaveWave_5                -715.40     269.06  -2.659 0.015987 *
## WaveWave_6                -772.50     269.06  -2.871 0.010160 *
## Severity_Num              -248.80      57.36  -4.337 0.000397 ***
## WaveWave_2:Severity_Num    61.60      81.13   0.759 0.457492
```

```
## WaveWave_3:Severity_Num   113.60      81.13   1.400 0.178425
## WaveWave_4:Severity_Num   135.70      81.13   1.673 0.111674
## WaveWave_5:Severity_Num   148.60      81.13   1.832 0.083593 .
## WaveWave_6:Severity_Num   161.30      81.13   1.988 0.062206 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 181.4 on 18 degrees of freedom
## Multiple R-squared:  0.7483, Adjusted R-squared:  0.5946
## F-statistic: 4.866 on 11 and 18 DF,  p-value: 0.00156
```

Interpretation of Coefficients:

Intercept: The intercept of 1151.40 represents the estimated count of attendees when the severity level is at its minimum (0-4) and the wave number is 1. Severity_Num: The coefficient of -248.80 for Severity_Num suggests that, on average, for each unit increase in the severity level (measured in the numeric scale), the count of attendees decreases by 248.80, holding the wave constant. Wave Coefficients: The coefficients for each wave indicate the change in the count of attendees compared to the reference wave (Wave 1). Interaction Terms: The coefficients for the interaction terms represent how the effect of severity level changes across different waves. None of the interaction terms appear to be statistically significant at conventional levels.

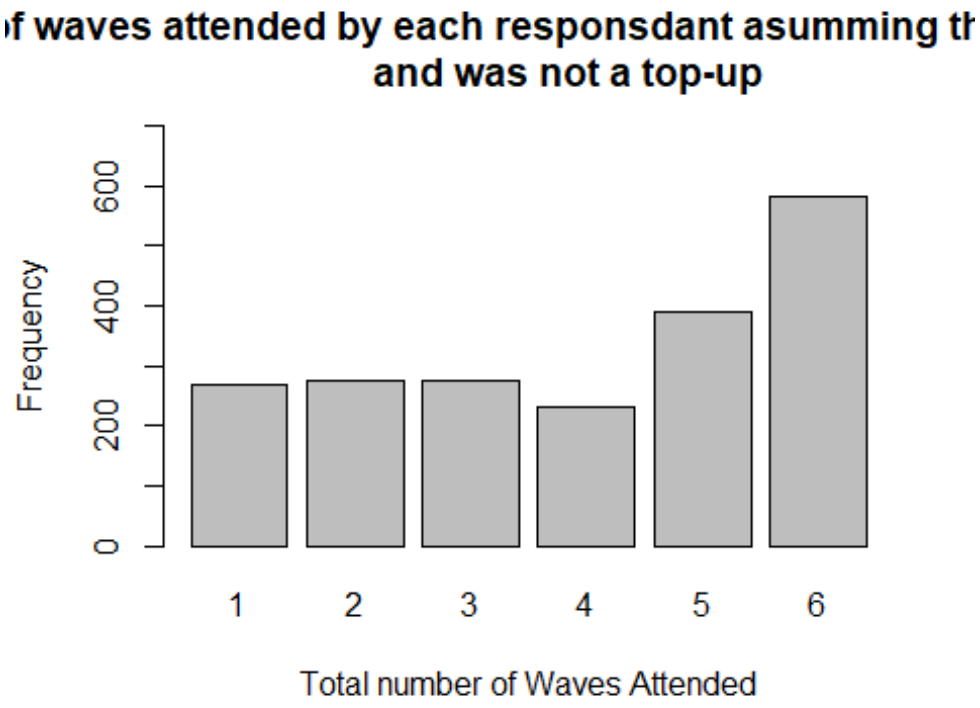
Now we have explored the attrition rate in terms of Number of waves attended in order it may also be beneficial to explore the same relationship but instead this time using the total number of waves attended by each respondent. This is because respondents may skip a given wave but then rejoin (I.e a respondent may complete wave 1 skip wave 2 and 3 but renter and complete wave 4). Below is how we create the variable for this:

```
mdf4$Waves_attended<-0
mdf4$Waves_attended <- ifelse(mdf4$W1_Present == 1, mdf4$Waves_attended + 1,
mdf4$Waves_attended)
mdf4$Waves_attended <- ifelse(mdf4$W2_Present == 1 & mdf4$W1_Present == 1,
mdf4$Waves_attended + 1, mdf4$Waves_attended)
mdf4$Waves_attended <- ifelse(mdf4$W3_Type == 0 & mdf4$W1_Present == 1,
mdf4$Waves_attended + 1, mdf4$Waves_attended)
mdf4$Waves_attended <- ifelse(mdf4$W4_Type == 1 & mdf4$W1_Present == 1,
mdf4$Waves_attended + 1, mdf4$Waves_attended)
mdf4$Waves_attended <- ifelse(mdf4$W5_Type == 1 & mdf4$W1_Present == 1,
mdf4$Waves_attended + 1, mdf4$Waves_attended)
mdf4$Waves_attended <- ifelse(mdf4$W6_Type == 1 & mdf4$W1_Present == 1,
mdf4$Waves_attended + 1, mdf4$Waves_attended)
table(mdf4$Waves_attended)

##
##    0    1    2    3    4    5    6
## 3339  268  275  277  233  390  582
```

We used the ifelse function to add 1 to the waves_attended variable if a given respondent was present for a given wave regardless if they had completed all of the previous waves or

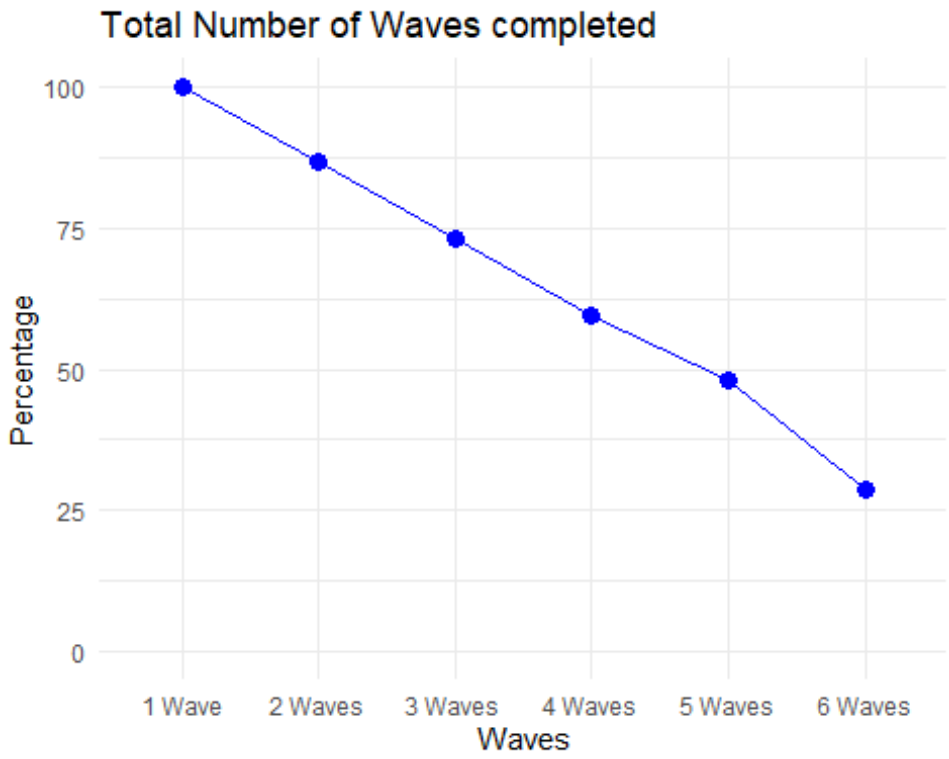
not. Similar to the "Present_XXX" variable created above we used the wave type and present variables to identify whether or not they completed the survey at a given wave. As we can see the total number of people who didn't attended any waves (Waves_attended = 0) is 3339 - this represents all of the Top-ups across all waves. Below is a bar chart of the above data minus the Top-ups

```
wave_table_total <- table(mdf4$Waves_attended)

# Remove the first count
wave_table_total  <- wave_table_total[-1]

# Plot the bar chart for the frequencies without the first bar
barplot(wave_table_total,
        xlab = "Total number of Waves Attended",
        ylab = "Frequency",
        main = "Total number of waves attended by each responsdant asumming
they attened wave 1
        and was not a top-up",
        ylim = c(0,700)
)
```



The above graph shows how man how many waves each of the respondents attended. It might also be useful to plot this on a line graph to show cumulatively how many attended number of waves. Below is this graph:

```r
### Putting previous variable into a matrix

Waves_attended_Total <- matrix(c(2025,1757, 1482, 1205, 972, 582), ncol=1,
byrow=TRUE)
rownames(Waves_attended_Total) <- c('1 Wave','2 Waves','3 Waves ','4
Waves','5 Waves','6 Waves')
colnames(Waves_attended_Total) <- c('Present')
Waves_attended_Totaldf<-as.data.frame(Waves_attended_Total)
Waves_attended_Totaldf$Waves <- rownames(Waves_attended_Totaldf)

Waves_attended_Totaldf$Percentage <- (Waves_attended_Totaldf$Present /
Waves_attended_Totaldf$Present[1]) * 100
ggplot(Waves_attended_Totaldf, aes(x = Waves, y = Percentage, group = 1)) +
  geom_line(color = "blue") +
  geom_point(color = "blue", size = 3) +
  labs(title = "Total Number of Waves completed", x = "Waves", y =
"Percentage") +
  theme_minimal() +
  ylim(c(0,100))
```



This graph shoes how many attended at least each level of waves (I.e 100% completed one wave and around 30% completed all 6 waves) Now that we have created some descriptive statistics we can now go on to explore the impact of depression severity on the number of waves completed. First we need to create a matrix of our results. We can do this by running a table of total number of waves attended and depression severity. Then we can subtract each of these results from wave 1 totals in order to make our matrix:

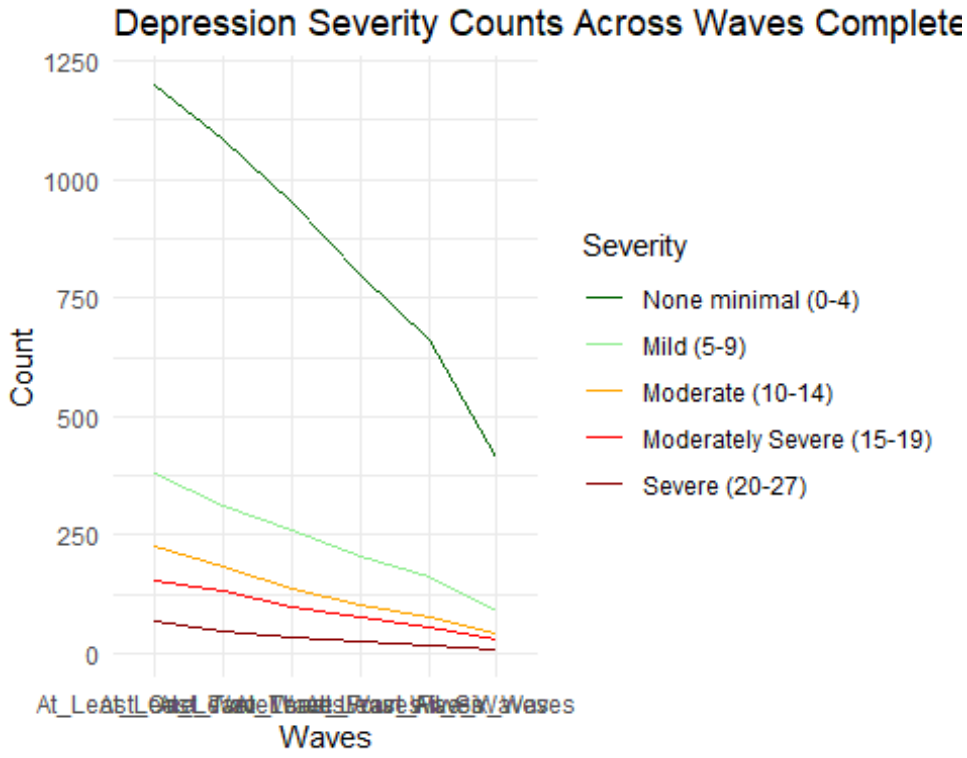
```
table(mdf4$Waves_attended,mdf4$W1_Dep_Severity)

##
##     Mild Moderate Moderately Severe None minimal Severe
##   0    0        0                 0            0      0
##   1   65       42                23          116     22
##   2   53       48                32          131     11
##   3   56       33                23          155     10
##   4   42       26                22          137      6
##   5   71       37                26          246     10
##   6   91       41                28          414      8
```

```
wave_data1 <- data.frame(
  Severity = c("None minimal (0-4)", "Mild (5-9)", "Moderate (10-14)",
"Moderately Severe (15-19)", "Severe (20-27)"),
  At_Least_One_Wave = c(1199, 378, 227, 154, 67),
  At_Least_Two_Waves = c(1083, 313, 185, 131, 45),
  At_Least_Three_Waves = c(952, 260, 137, 99, 34),
  At_Least_Four_Waves = c(797, 204, 104, 76, 24),
  At_Least_Five_Waves = c(660, 162, 78, 54, 18),
  All_Six_Waves =       c(414, 91, 41, 28, 8)
)
```

We can then plot this. As with the other results graph above we used the ggplot2 and the tidyr package. We also again had to use 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.:

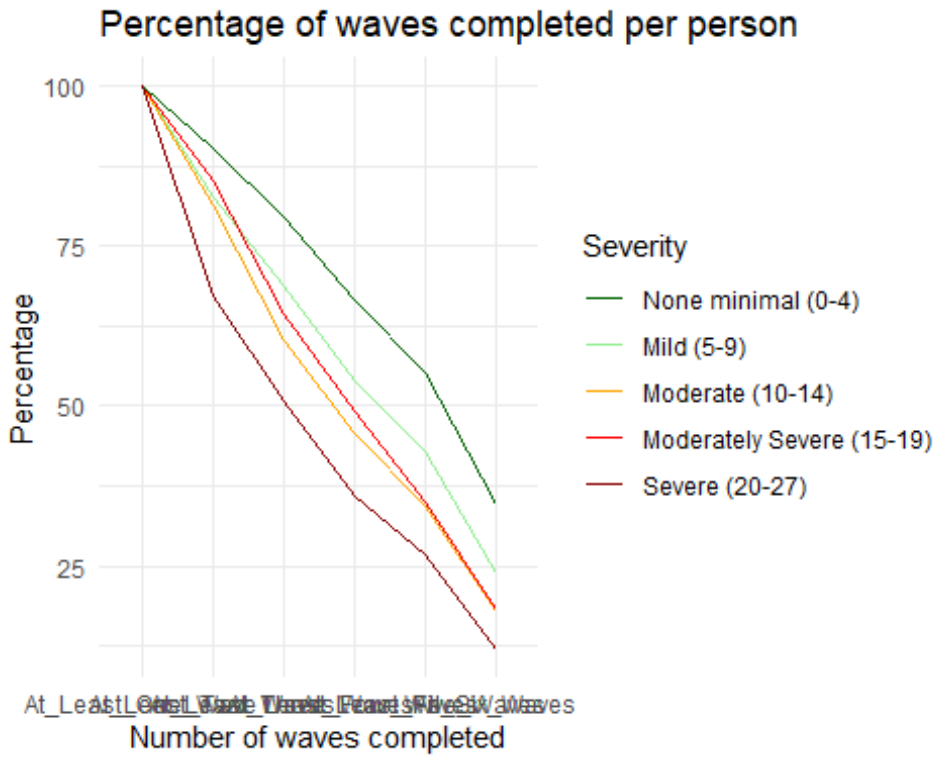
```
custom_colors <- c("None minimal (0-4)" = "darkgreen", "Mild (5-9)"=
"lightgreen","Moderate (10-14)" = "orange",
                   "Moderately Severe (15-19)" = "red", "Severe (20-27)" =
"darkred")
legend_order <- c("None minimal (0-4)", "Mild (5-9)", "Moderate (10-14)",
"Moderately Severe (15-19)", "Severe (20-27)")


wave_data1_long <- wave_data1 %>%
  pivot_longer(cols = -Severity, names_to = "Waves", values_to = "Counts")
%>%
  mutate(Waves = factor(Waves, levels = c("At_Least_One_Wave",
"At_Least_Two_Waves", "At_Least_Three_Waves", "At_Least_Four_Waves",
"At_Least_Five_Waves", "All_Six_Waves")))

# Plotting
ggplot(wave_data1_long, aes(x = Waves, y = Counts, group = Severity, color =
Severity)) +
  geom_line() +
  labs(title = "Depression Severity Counts Across Waves Completed",
```

```
       x = "Waves",
       y = "Count") +
  scale_color_manual(values = custom_colors, breaks = legend_order) +  # You
can change the palette if needed
  theme_minimal()
```



We can also show this as a percentage of Wave 1:

```
wave_data1_long <- wave_data1_long %>%
  group_by(Severity) %>%
  mutate(Percentage = Counts / Counts[Waves == "At_Least_One_Wave"] * 100)

# Plotting the line graph with percentages relative to Wave 1 on the y-axis
ggplot(wave_data1_long, aes(x = Waves, 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 waves completed per person",
       x = "Number of waves completed", y = "Percentage") +
  theme_minimal()
```



We can then run a regression again:

```
### Regression Tests

# Reshape the data to long format

data_long1 <- melt(wave_data1, id.vars = "Severity", variable.name = "Wave",
value.name = "Counts")

# Convert Severity to numeric
severity_mapping <- c('None minimal (0-4)' = 1, 'Mild (5-9)' = 2, 'Moderate
(10-14)' = 3, 'Moderately Severe (15-19)' = 4, 'Severe (20-27)' = 5)
data_long1$Severity_Num <- as.numeric(factor(data_long1$Severity, levels =
names(severity_mapping)))



# Perform OLS regression
model1 <- lm(Counts ~ Wave * Severity_Num, data = data_long1)
summary(model1)

##
## Call:
## lm(formula = Counts ~ Wave * Severity_Num, data = data_long1)
##
## Residuals:
```

```
##    Min     1Q Median     3Q    Max
## -275.8 -153.8    0.7  133.4  296.4
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          1151.40     225.67   5.102 7.45e-05
***
## WaveAt_Least_Two_Waves               -122.60     319.15  -0.384  0.70537
## WaveAt_Least_Three_Waves             -255.90     319.15  -0.802  0.43311
## WaveAt_Least_Four_Waves              -408.20     319.15  -1.279  0.21713
## WaveAt_Least_Five_Waves              -539.40     319.15  -1.690  0.10825
## WaveAll_Six_Waves                    -772.50     319.15  -2.421  0.02629
*
## Severity_Num                         -248.80      68.04  -3.657  0.00181
**
## WaveAt_Least_Two_Waves:Severity_Num    23.00      96.23   0.239  0.81379
## WaveAt_Least_Three_Waves:Severity_Num  49.10      96.23   0.510  0.61607
## WaveAt_Least_Four_Waves:Severity_Num   81.40      96.23   0.846  0.40870
## WaveAt_Least_Five_Waves:Severity_Num  109.60      96.23   1.139  0.26964
## WaveAll_Six_Waves:Severity_Num        161.30      96.23   1.676  0.11097
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 215.2 on 18 degrees of freedom
## Multiple R-squared:  0.7388, Adjusted R-squared:  0.5791
## F-statistic: 4.627 on 11 and 18 DF,  p-value: 0.002079
```

while there are differences in how waves are represented between the two models, the significance of predictors and the overall model fit remain similar. The choice between the two representations may depend on the specific research question and the interpretation of the waves.