Group Project 1

Lee Ann & Jonathan

2025-11-04

```
##############################################################################
#####################Data import and wrangling - LAS##########################
##############################################################################



#Libraries
library(readr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v purrr     1.1.0
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(reshape2)
```

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

```
library(pastecs)
```

```
##
## Attaching package: 'pastecs'
##
## The following objects are masked from 'package:dplyr':
##
##     first, last
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(ggplot2)
library(dplyr)
library(tidyr)
library(reshape2)
```

```r
#First I will import the dataset we will be working from and call it raw
#raw <- read_csv("C:/Users/Owner/Downloads/TextMessages.csv")

#Then I will recreate that data here so that my partner and I can work on it
#without having to worry about it being located in different spots on each
#of our computers. This function will output the code needed to recreate the raw
#data frame
#dput(as.data.frame(raw))

#Then I can take that code and turn it into the df here
df <- structure(list(Group = c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
                              1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,
                              2,2), Baseline = c(52,68,85,47,73,57,63,50,66,
                             60,51,72,77,57,79,75,53,72,62,71,53,64,79,75
                             ,60,65,57,66,71,75,61,80,66,53,62,61,77,66,52
                             ,60,58,54,72,71,87,75,57,59,46,89), Six_months
                    = c(32,48,62,16,63,53,59,58,59,57,60,56,61,52,9,76,38,63,53
                        ,61,50,78,33,68,59,62,50,62,61,70,64,64,55,47,61,56,64,
                        62,47,56,78,74,61,61,78,62,71,55,46,79), Participant =
                      c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,
                        22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,
                        40,41,42,43,44,45,46,47,48,49,50)),
               row.names = c(NA,-50L), class = "data.frame")

#Now that we both have the df available, I will make sure all of our
#variables are named appropriately and are of the correct classes
names(df)
```

```
## [1] "Group"       "Baseline"    "Six_months"  "Participant"
```

```r
#These names are of reasonable lengths and descriptions so we can keep them as
#is
class (df$Group)
```

```
## [1] "numeric"
```

```r
class (df$Baseline)
```

```
## [1] "numeric"
```

```r
class (df$Six_months)
```

```
## [1] "numeric"
```

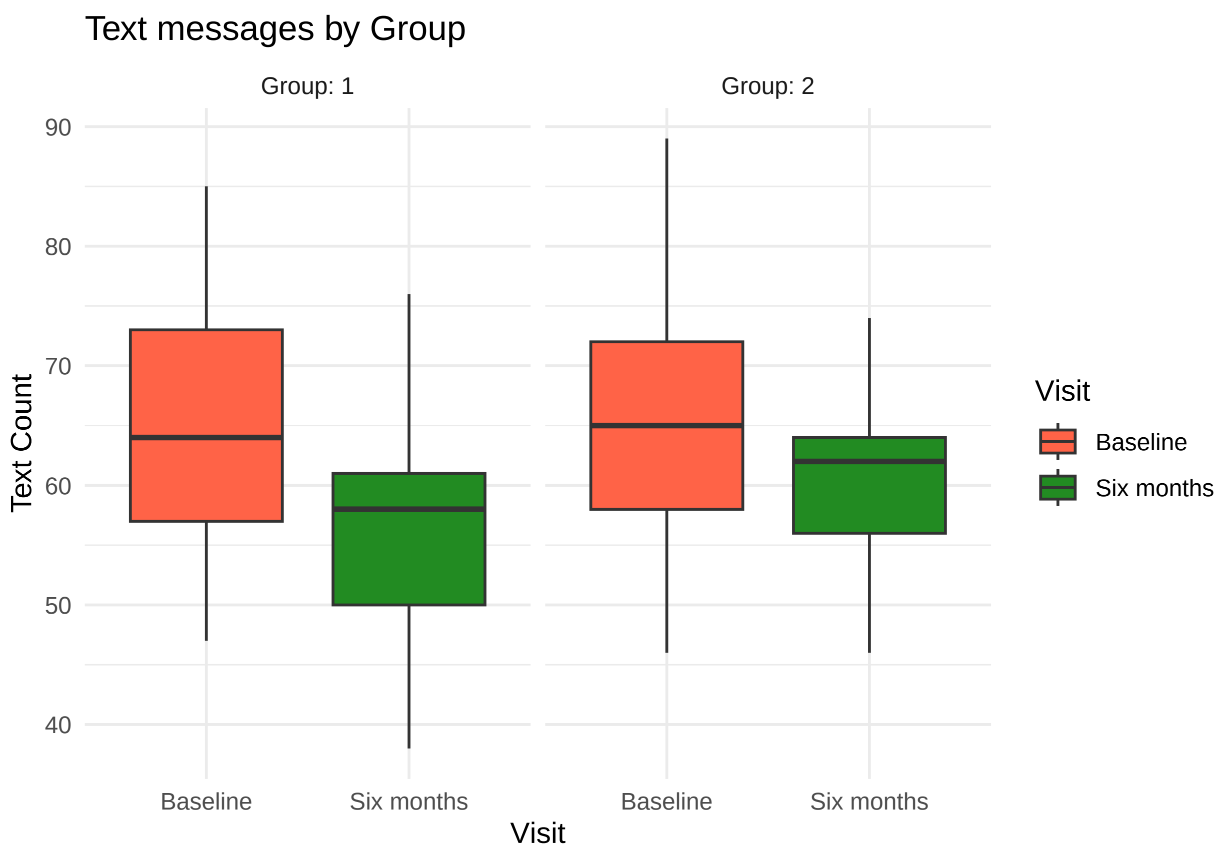
```r
class (df$Participant)
```

```
## [1] "numeric"
```

```r
#Currently all of our variables are numeric. This is appropriate for Baseline,
#Six_months, and Participants variables. Group however should be a factor.
#We will do that here
df$Group <- as.factor(df$Group)
#And check that it worked here
is.factor(df$Group)
```

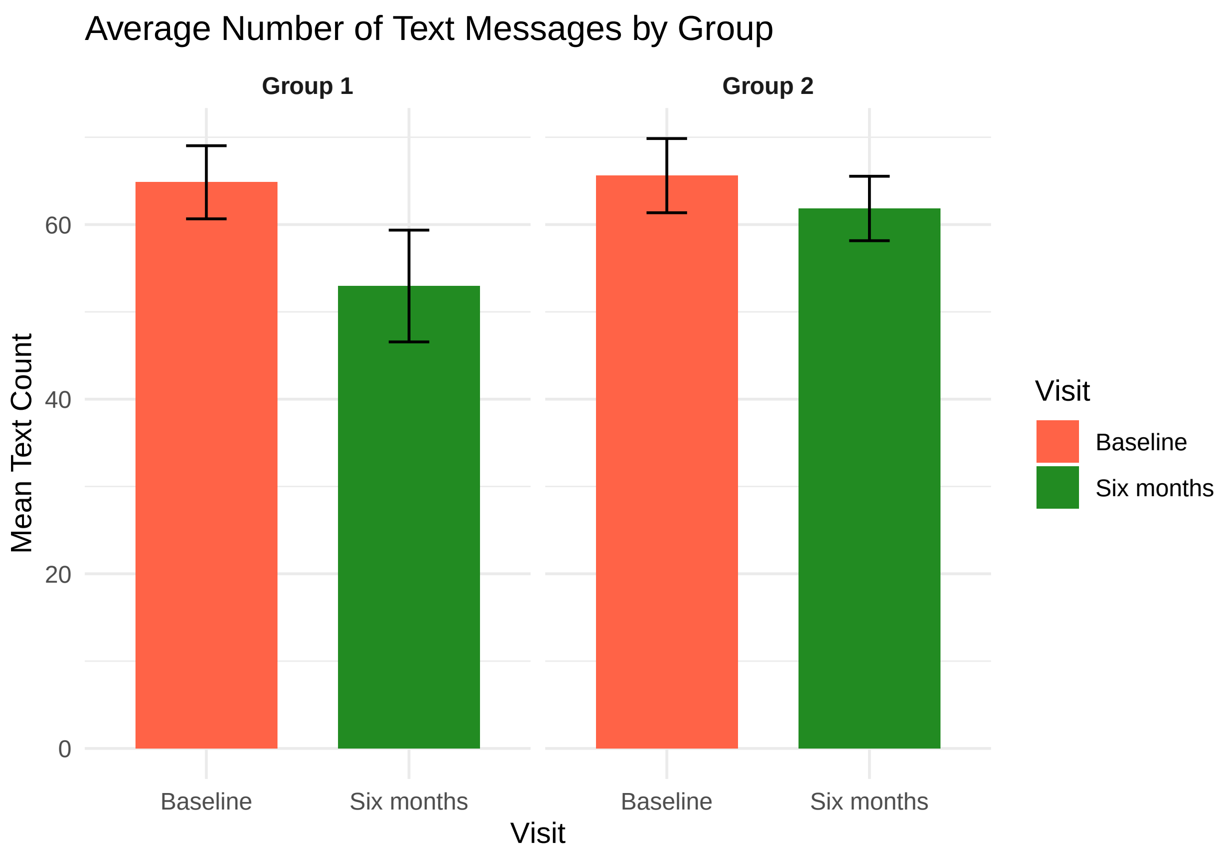
```
## [1] TRUE
```

```r
#It is TRUE confirming we successfully converted it to a factor
#Now that our data is imported and has the correct names and classes, we can
#begin to create our figures and start our analysis.

###############################################################################
################Box Plot - JR##################################################
###############################################################################

#I'll start by reordering the columns and then shifting to a long "tidy"
#data set by "melting" the Baseline and Six_months columns. I create an object,
#pipe df into select to reorder my columns (just a little OCD), pipe into
#rename(this is so the six months variable appears without the underscore in
#the graph),
#and pipe into melt to create a tidy data frame.

df_long <- df %>% select(Participant, Group, Baseline, Six_months) %>%
  rename("Six months" = Six_months) %>%
    melt(id.vars = c("Participant", "Group"), variable.name = "Visit",
          value.name = "Text_Count")

#Next, I'll create the stratified boxplot by group. I assign an object, pipe the
#new data frame into ggplot, assign aesthetics, add a boxplot layer,
#removed the outliers, used "free_y" in facet_grid; this
#allows R to handle how to assign the y-axis.facet by group with a labeller
#argument for the two groups, add a color scheme by visit, and in the themes
#layer, I remove the legend, and (just for fun) change the background color.

text_count_boxplot2 <- df_long %>%
  ggplot(aes(x = Visit, y = Text_Count, fill = Visit)) +
  geom_boxplot(outliers = FALSE)  +
  facet_grid(~Group, labeller = label_both, scales = "free_y") +
  scale_fill_manual(values = c("tomato", "forestgreen")) +
  theme(legend.position = "none"
      ) + theme_minimal() +
  labs(title = "Text messages by Group", y = "Text Count")

text_count_boxplot2
```




```
##############################################################################
################bar charts - LAS#############################################
###############################################################################

#To make our bar plot we will first convert our data from wide to long format.
#Jonathan has already done this above

#Next, since our bar plot will compare the means between groups, we will
#calculate the mean text messages and 95% CI for each group and timepoint.
# Compute summary statistics for mean and 95% CI per group/time. We will
#put these results in their own data frame called df_summary

df_summary <- df_long %>%
  group_by(Group, Visit) %>%
  summarize(
    mean_count = mean(Text_Count, na.rm = TRUE),
    sd = sd(Text_Count, na.rm = TRUE),
    n = n(),
    se = sd / sqrt(n),
    ci_lower = mean_count - 1.96 * se,
    ci_upper = mean_count + 1.96 * se,
    .groups = "drop"
  )

#Now that we have our values, I will  rename them so that they look more
#aesthetically pleasing in our plot.
```

```r
df_summary <- df_summary %>%
  mutate(
    Visit = recode(Visit,
                  "Baseline" = "Baseline",
                  "Six_months" = "Six months"),
    Group = recode_factor(as.factor(Group),
                          "1" = "Group 1",
                          "2" = "Group 2")
  )

#we can now plot our bar plot. We will use ggplot to
#make a bar plot that compares the mean text messages at each time point and
#we will facet by groups so that we can visualize the differences in time
#points between groups as well

ggplot(df_summary, aes(x = Visit, y = mean_count, fill = Visit)) +
  geom_col(position = position_dodge(), width = 0.7) +
  geom_errorbar(aes(ymin = ci_lower, ymax = ci_upper),
               width = 0.2, position = position_dodge(0.7)) +
  facet_wrap(~ Group) +
  labs(
    title = "Average Number of Text Messages by Group",
    x = "Visit",
    y = "Mean Text Count",
    fill = "Visit"
  ) +
  theme_minimal() +
  scale_fill_manual(values = c("tomato", "forestgreen")) +
  theme(
    legend.position = "right",
    strip.text = element_text(face = "bold")
  )
```




```
#Here we can see the mean and 95% CI plotted at each time point for each group
#Baseline values are pink and six month follow up values are blue. The color
#designations are describe din the legend on the right. The panel
#on the left show the results of group 1 while the one on the right shows the
#results of group 2.The error bars on the top of each bar represent the 95%
#CI. We can clearly see that both group one and group 2 experience a decrease
#In the mean number of messages at follow up compared to baseline. The 95% CI
#do not appear to overlap in group one, but they do appear to overlap in group
#2 suggesting that the decrease in messages was statistically significant for
#group 1 but not for group 2.

###############################################################################
#######################Summary statistics  - JR################################
###############################################################################

#Summary statistics by group and time:

by(df$Baseline, df$Group, function(x) round(stat.desc(x, norm = TRUE), 3))
```

```
## df$Group: 1
##       nbr.val      nbr.null        nbr.na           min           max         range
##        25.000         0.000         0.000        47.000        85.000        38.000
##           sum        median          mean       SE.mean  CI.mean.0.95           var
##      1621.000        64.000        64.840         2.136         4.408       114.057
##       std.dev      coef.var      skewness      skew.2SE      kurtosis      kurt.2SE
##        10.680         0.165         0.035         0.037        -1.273        -0.706
```

```
##   normtest.W   normtest.p
##        0.962        0.448
## ------------------------------------------------------------
## df$Group: 2
##      nbr.val     nbr.null       nbr.na          min          max        range
##       25.000        0.000        0.000       46.000       89.000       43.000
##          sum       median         mean      SE.mean CI.mean.0.95          var
##     1640.000       65.000       65.600        2.167        4.473      117.417
##      std.dev     coef.var     skewness     skew.2SE     kurtosis     kurt.2SE
##       10.836        0.165        0.418        0.451       -0.587       -0.325
##   normtest.W   normtest.p
##        0.971        0.669
```

```r
by(df$Baseline, df$Group, summary)
```

```
## df$Group: 1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   47.00   57.00   64.00   64.84   73.00   85.00
## ------------------------------------------------------------
## df$Group: 2
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    46.0    58.0    65.0    65.6    72.0    89.0
```

```r
#At Baseline, the two groups are quite similar with similar means, medians
#and interquartile ranges. The variances are also quite close, thus so is the
#coefficients of variation and 95% CIs. For both groups, a Shapiro-Wilks normality
#test shows both groups with a high likelihood of being normally distributed.

by(df$Six_months, df$Group, function(x) round(stat.desc(x, norm = TRUE), 3))
```

```
## df$Group: 1
##      nbr.val     nbr.null       nbr.na          min          max        range
##       25.000        0.000        0.000        9.000       78.000       69.000
##          sum       median         mean      SE.mean CI.mean.0.95          var
##     1324.000       58.000       52.960        3.266        6.741      266.707
##      std.dev     coef.var     skewness     skew.2SE     kurtosis     kurt.2SE
##       16.331        0.308       -1.100       -1.186        0.808        0.448
##   normtest.W   normtest.p
##        0.877        0.006
## ------------------------------------------------------------
## df$Group: 2
##      nbr.val     nbr.null       nbr.na          min          max        range
##       25.000        0.000        0.000       46.000       79.000       33.000
##          sum       median         mean      SE.mean CI.mean.0.95          var
##     1546.000       62.000       61.840        1.882        3.884       88.557
##      std.dev     coef.var     skewness     skew.2SE     kurtosis     kurt.2SE
##        9.410        0.152        0.164        0.177       -0.741       -0.411
##   normtest.W   normtest.p
##        0.942        0.161
```

```r
by(df$Six_months, df$Group, summary)
```

```
## df$Group: 1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9.00   50.00   58.00   52.96   61.00   78.00
## ------------------------------------------------------------
```

```
## df$Group: 2
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   46.00   56.00   62.00   61.84   64.00   79.00
```

```
#At six months, Group 1 is lower in almost every category: mean, median,
#interquartile range. However, group 1 shows significantly more deviation, with
#a coefficient of varation more than twice that of group 2. Group 1 also may
#not be normally distributed, as the output for the Shapiro-Wilks test would
#reject the null at an alpha level = .05 with a p-value of .006, suggesting that
#the data is likely not normally distributed. Group two, on the other hand,
#tests as likely normal.
```