My name goes here

The names of the people I have worked with go here

2022-03-16 10:20:04

Instructions

```
df.data = read_csv("responses.csv") %>%
  rename(average_sleep = 'average sleep',
         average_quality = 'average quality',
         average_mood = 'average mood',
         midterm_sleep = 'midterm sleep',
         midterm_quality = 'midterm quality',
         midterm_mood = 'midterm mood')
```

```
## Rows: 31 Columns: 8
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (8): participants, midterms, average sleep, average quality, average moo...
```

```
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
mean(df.data$average_sleep)
```

```
## [1] 7.645161
```

```
mean(df.data$average_quality)
```

```
## [1] 7.516129
```

```
mean(df.data$average_mood)
```

```
## [1] 7.967742
```

```
sd(df.data$average_sleep)
```

```
## [1] 0.9146361
```

```
sd(df.data$average_quality)
```

```
## [1] 1.028623
```

```
sd(df.data$average_mood)
```

```
## [1] 0.7520581
```

```
mean(df.data$midterm_sleep)
```

```
## [1] 6.741935
```

```
mean(df.data$midterm_quality)
```

```
## [1] 7.258065
```

```
mean(df.data$midterm_mood)
```

```
## [1] 7
```

```
sd(df.data$midterm_sleep)
```

```
## [1] 1.264061
```

```
sd(df.data$midterm_quality)
```

```
## [1] 0.9989242
```

```
sd(df.data$midterm_mood)
```

```
## [1] 1.095445
```

```
t.test(df.data$midterm_sleep, mu = 7.718 )
```

```
##
##  One Sample t-test
##
## data:  df.data$midterm_sleep
## t = -4.2992, df = 30, p-value = 0.0001668
## alternative hypothesis: true mean is not equal to 7.718
## 95 percent confidence interval:
##  6.278274 7.205597
## sample estimates:
## mean of x
##  6.741935
```

```r
t.test(df.data$midterm_quality, mu = 7.281 )
```

```
## 
##  One Sample t-test
## 
## data:  df.data$midterm_quality
## t = -0.12784, df = 30, p-value = 0.8991
## alternative hypothesis: true mean is not equal to 7.281
## 95 percent confidence interval:
##  6.891656 7.624473
## sample estimates:
## mean of x 
##  7.258065
```

```r
t.test(df.data$midterm_mood, mu = 7.968 )
```

```
## 
##  One Sample t-test
## 
## data:  df.data$midterm_mood
## t = -4.92, df = 30, p-value = 2.92e-05
## alternative hypothesis: true mean is not equal to 7.968
## 95 percent confidence interval:
##  6.598187 7.401813
## sample estimates:
## mean of x 
##         7
```

```r
fms = lm(df.data$midterms~df.data$midterm_sleep)
summary(fms)
```

```
## 
## Call:
## lm(formula = df.data$midterms ~ df.data$midterm_sleep)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -1.4388 -0.4388 -0.3890  0.5612  1.5612 
## 
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)   
## (Intercept)              2.7873     0.8132   3.428  0.00184 **
## df.data$midterm_sleep   -0.0498     0.1186  -0.420  0.67770   
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 0.8212 on 29 degrees of freedom
## Multiple R-squared:  0.006041,   Adjusted R-squared:  -0.02823 
## F-statistic: 0.1763 on 1 and 29 DF,  p-value: 0.6777
```

```
fmq = lm(df.data$midterms~df.data$midterm_quality)
summary(fmq)
```

```
##
## Call:
## lm(formula = df.data$midterms ~ df.data$midterm_quality)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4397 -0.4397 -0.3933  0.5372  1.6067
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              2.11530    1.10090   1.921   0.0646 .
## df.data$midterm_quality  0.04634    0.15031   0.308   0.7601
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.8224 on 29 degrees of freedom
## Multiple R-squared:  0.003266,   Adjusted R-squared:  -0.0311
## F-statistic: 0.09503 on 1 and 29 DF,  p-value: 0.7601
```

```
fmm = lm(df.data$midterms~df.data$midterm_mood)
summary(fmm)
```

```
##
## Call:
## lm(formula = df.data$midterms ~ df.data$midterm_mood)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1461 -0.4516 -0.1461  0.5067  1.5484
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)            4.5905     0.8854   5.185 1.52e-05 ***
## df.data$midterm_mood  -0.3056     0.1250  -2.444   0.0208 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.7501 on 29 degrees of freedom
## Multiple R-squared:  0.1708, Adjusted R-squared:  0.1422
## F-statistic: 5.974 on 1 and 29 DF,  p-value: 0.02084
```

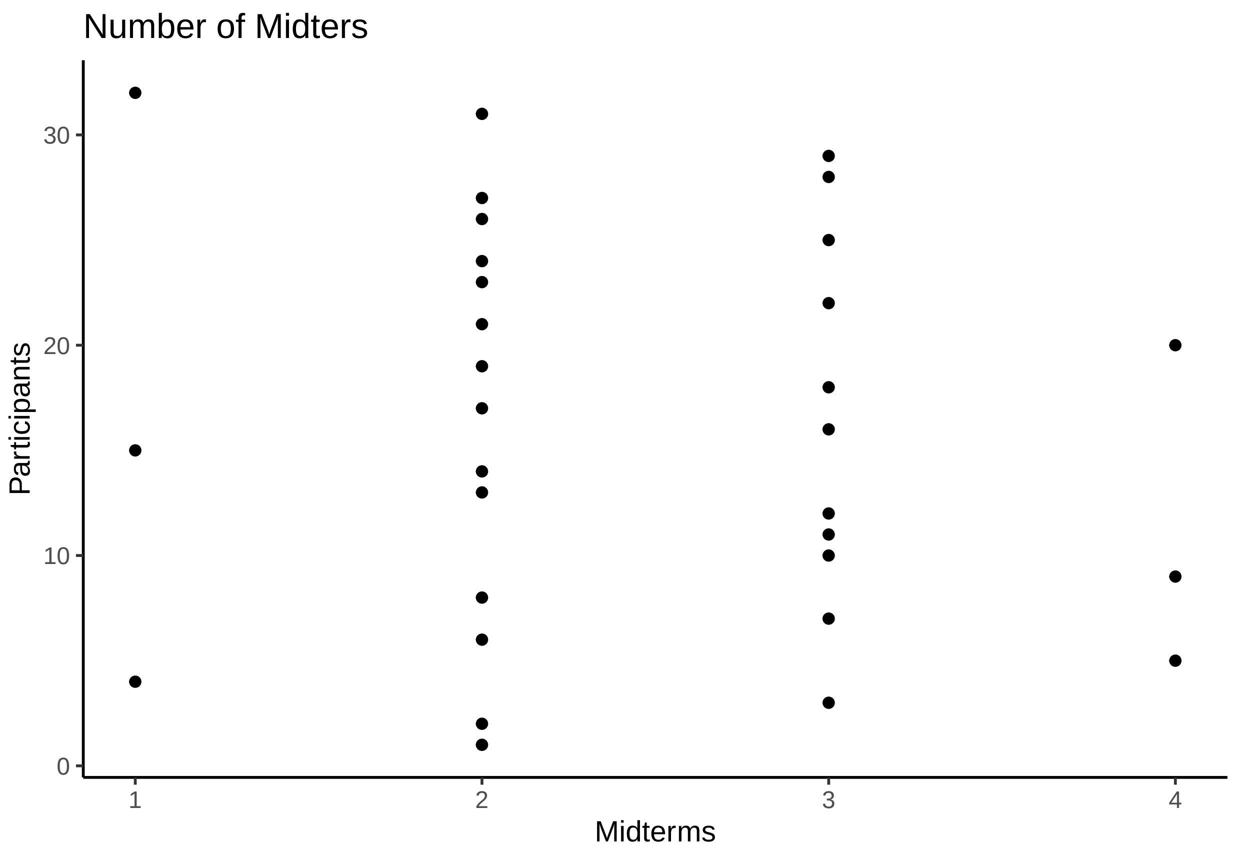
```
fit.midterm = lm(formula = df.data$midterms ~ (df.data$midterm_sleep + df.data$midterm_quality + df.data
             data = df.data)
fit.midterm %>%
  anova()
```

```
## Analysis of Variance Table
##
## Response: df.data$midterms
##                         Df  Sum Sq Mean Sq F value  Pr(>F)  
## df.data$midterm_sleep    1  0.1189  0.1189  0.2099 0.65053  
## df.data$midterm_quality  1  0.0057  0.0057  0.0101 0.92077  
## df.data$midterm_mood     1  4.2601  4.2601  7.5214 0.01069 *
## Residuals               27 15.2927  0.5664                  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

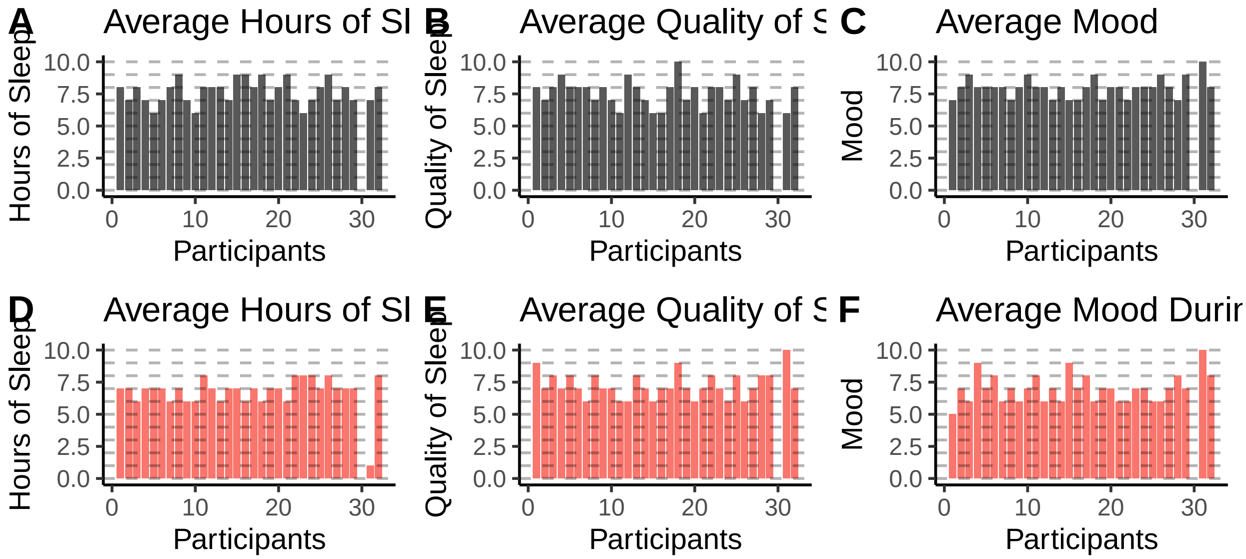
```
pwr.anova.test(k = 2,
               n = NULL,
               f = 0.699,
               sig.level = 0.05,
               power = 0.8)
```

```
## 
##      Balanced one-way analysis of variance power calculation 
## 
##               k = 2
##               n = 9.100321
##               f = 0.699
##       sig.level = 0.05
##           power = 0.8
## 
## NOTE: n is number in each group
```

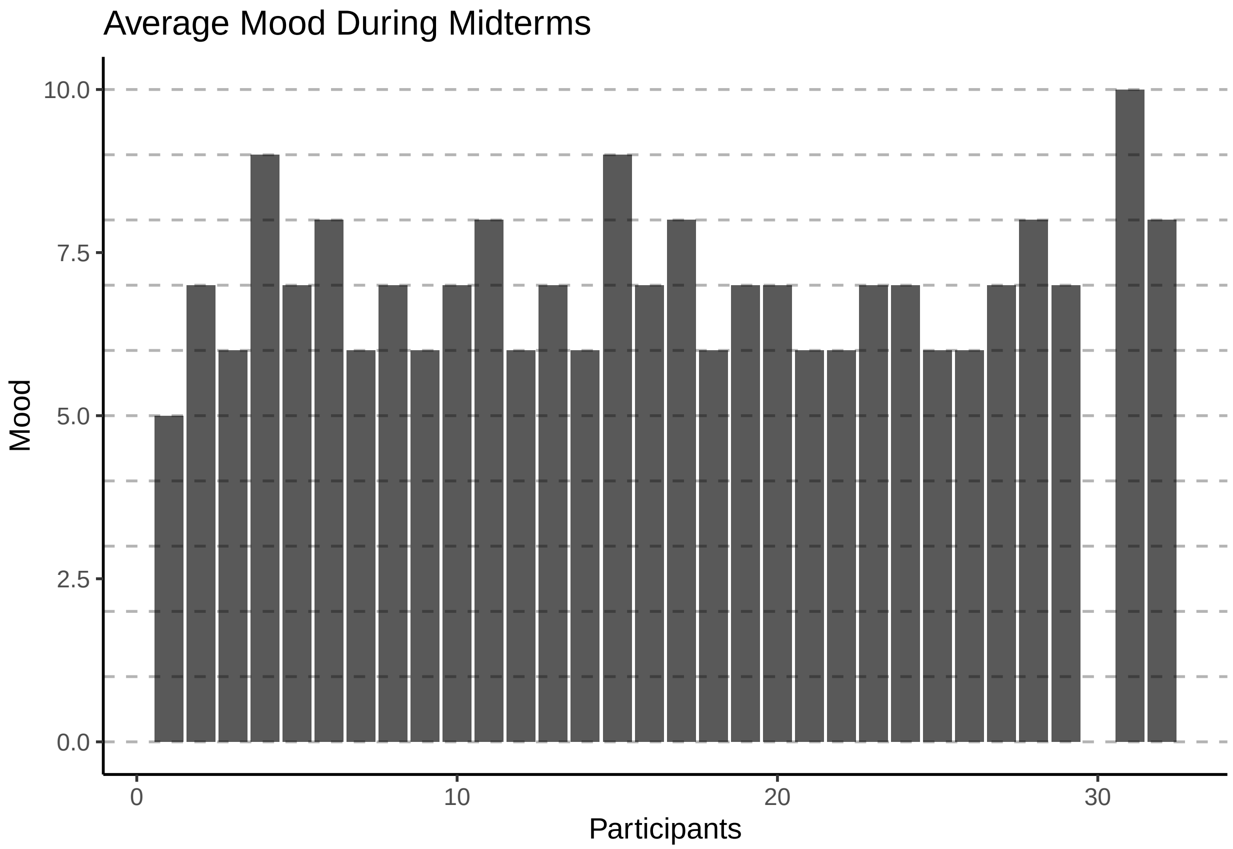
```
ggplot(data = df.data,
       mapping = aes(x = df.data$midterms,
                     y = df.data$participants,
                     )) +
  geom_point() +
  labs(title = "Number of Midters",
       x = "Midterms",
       y = "Participants")
```



```
AS <- ggplot(data = df.data,
                 aes(x = participants,
                     y = average_sleep)) +
  geom_col(position = position_dodge()) +
  labs(title = "Average Hours of Sleep",
       x = "Participants",
       y = "Hours of Sleep") +
  theme(legend.position="none") +
  geom_hline(yintercept = seq(from = 0,
                              to = 10,
                              by = 1),
             linetype = 2,
             alpha = 0.3)


MS <- ggplot(data = df.data,
       aes(x = participants,
           y = midterm_sleep,
           fill = "red")) +
  geom_col(position = position_dodge()) +
  labs(title = "Average Hours of Sleep During Midterms",
       x = "Participants",
       y = "Hours of Sleep") +
  theme(legend.position="none") +
  geom_hline(yintercept = seq(from = 0,
                              to = 10,
                              by = 1),
             linetype = 2,
             alpha = 0.3)

AQ <- ggplot(data = df.data,
       aes(x = participants,
           y = average_quality)) +
  geom_col(position = position_dodge()) +
  labs(title = "Average Quality of Sleep During",
       x = "Participants",
       y = "Quality of Sleep") +
  theme(legend.position="none") +
  geom_hline(yintercept = seq(from = 0,
                              to = 10,
                              by = 1),
             linetype = 2,
             alpha = 0.3)

MQ <- ggplot(data = df.data,
       aes(x = participants,
           y = midterm_quality,
           fill = "red")) +
  geom_col(position = position_dodge()) +
  labs(title = "Average Quality of Sleep During Midterms",
       x = "Participants",
       y = "Quality of Sleep") +
  theme(legend.position="none") +
  geom_hline(yintercept = seq(from = 0,
                              to = 10,
                              by = 1),
             linetype = 2,
             alpha = 0.3)

AM <- ggplot(data = df.data,
       aes(x = participants,
           y = average_mood)) +
  geom_col(position = position_dodge()) +
  labs(title = "Average Mood",
       x = "Participants",
       y = "Mood") +
  theme(legend.position="none") +
  geom_hline(yintercept = seq(from = 0,
                              to = 10,
                              by = 1),
             linetype = 2,
             alpha = 0.3)

MM <- ggplot(data = df.data,
       aes(x = participants,
           y = midterm_mood,
           fill = "red")) +
  geom_col(position = position_dodge()) +
  labs(title = "Average Mood During Midterms",
       x = "Participants",
       y = "Mood") +
  theme(legend.position="none") +
  geom_hline(yintercept = seq(from = 0,
                              to = 10,
                              by = 1),
             linetype = 2,
             alpha = 0.3)

figure <- ggarrange(AS, AQ, AM, MS, MQ, MM,
                    labels = c("A", "B", "C", "D", "E", "F"),
                    ncol = 3, nrow = 3)
figure
```



```
ggplot(data = df.data,
       aes(x = participants,
           y = midterm_mood)) +
  geom_col(position = position_dodge()) +
  labs(title = "Average Mood During Midterms",
       x = "Participants",
       y = "Mood") +
  theme(legend.position="none") +
  geom_hline(yintercept = seq(from = 0,
                              to = 10,
                              by = 1),
             linetype = 2,
             alpha = 0.3)
```



# Part 1 (5 points)

Load and visualize data

1.1 (1 points)