STA303/1002: Mini-mixed assessment (untimed component)

Starship crew analysis

Chief Science Officer Jiahao(Green) Bai; ID: 1005804097

Information	Note
Name	Mini-mixed assessment
Type (Main, Mini or Basket)	Mini
Value	5% (Path A)0% (Path B)
Due	Wednesday, March 9, 2022; assessment window from 8:00 a.m. ET to 8:00 p.m. ET
Submission instruction	Submission: Via Quercus quiz (50 minutes, 1 attempt, no pausing) and Markus (10 percentage point penalty for not submitting required files)
Accommodations and extension policy	In the case of a personal illness/emergency, a declaration can be made, but must be submitted no more than 3 days after the due date. Extensions may be requested through the same form up to 48 hours before the due date.

Mixed assessment 1 has two components:

- Untimed guided analysis (this)
- Timed assessment (50 minutes; 12-hour assessment window is 8:00 a.m. to 8:00 p.m. ET Wednesday, March 9)
- See the mixed assessments overview page for further information and revisions links.

How your grade is calculated

- The 98% of your mini-mixed grade is based on your performance on the **timed** component.

- 2% of your grade is based on the correctness of your Rmd. There will be a student facing autotest you can run on your submissions to check if the objects required are there and appear to be mostly sensible (note, this doesn't guarantee in all cases that they are fully RIGHT, just that they passes the checks set up for this component).

- If you do the timed component, but DON'T submit the appropriate Rmd and PDF to MarkUs by the end of the window, there is an **additional** 10 percentage point penalty.
  - Note the file name requirements: `sta303-w22-mini-mixed.Rmd` and `sta303-w22-mini-mixed.pdf`.
  - You can upload as many times as you like before the end of the window, so make sure you upload a 'safety' copy of your Rmd and PDF once you have started working on it.

Instructions

Before making any changes in this Rmd, you should Knit it to make sure it works.

1. Update the `yaml` at the top of this document to have your name and your student ID. There are TWO places you need to do this for each one, probably on lines 4 and 12. I.e., replace the square brackets and everything inside them with the appropriate details. Your student ID is all numbers (usually 10, sometimes 8 or 9), it is NOT your UTORid.
2. Complete the guided analysis below. You will want to complete this BEFORE attempting your timed assessment.
3. Complete your timed assessment. It will require your work in this document, as well general STA303 content knowledge.
4. Knit this .Rmd to .pdf and submit BOTH files to the submission link in the table above.

Setting up your libraries

If you are working on this on the Jupyter Hub, the `tidyverse`, `devtools`, `lme4`, `lattice` and `lmtest` packages will already be installed. `randomNames` and `myStarship` are also in the process of being added by the JupyterHub team.

If you're working locally, you'll have to install packages first if they are not already installed. On the JupyterHub, you may also need to install the `randomNames` package from CRAN and the `myStarship` package from GitHub. All the code you need to do this is in the `setup` chunk below.

Note: **Do not add any additional libraries/packages**. You do not need them to complete these tasks and they may interfere with the autochecking of your submission.

```
# Working locally? RUN THIS CHUNK FIRST!
  # You should only need to run it once on your local machine.
# On the JupyterHub, you may need to run it at the beginning of each new session.

# These are the packages you will need for this activity.
packages_needed <- c("tidyverse", "devtools", "lme4",
                     "lattice", "lmtest", "randomNames")

package.check <- lapply(
  packages_needed,
  FUN = function(x) {
    if (!require(x, character.only = TRUE)) {
      install.packages(x, dependencies = TRUE,
      repos = "https://cloud.r-project.org/") # you may need to change the mirror if
      # you're in China (and potentially other countries.)
      # Students in China have reported that
      # "https://mirrors.tuna.tsinghua.edu.cn/CRAN/" worked for them.
    }
  }
)

# Remove objects no longer needed
rm(packages_needed, package.check)

# You may be prompted to install or update additional packages
# If so, you'll see a message in the console
# Type a enter/return in the console to skip updating
devtools::install_github("elb0/myStarship", force = TRUE)
```

```
# Run libraries for easy access to the functions we'll be using
library(tidyverse)
library(lme4)
library(myStarship)
```

Get your data

IMPORTANT you MUST update your student ID in the function in the following chunk. You will be graded based on your *unique dataset* and so risk losing extensive marks if you use the wrong dataset.

```
# put your student ID in here
studentIDnum <- 1005804097
get_my_starship(studentIDnum)

# after you run this function, your unique dataset will appear in the environment
# it will be called crew_data
```

The goal

You are the Chief Science Officer of the SS Sloocrot. You have data about the productivity of the crew over a 12 week period after a shore leave (a holiday break for the crew). For each member of the crew you also have data on their `rank` within Starfleet, their role on the ship (`position`), which of the three main divisions (`division`) they are in (Command, Operations, Science), as well as their sub-division (`sub_division`, e.g. Engineering is a sub-division of Operations). You also know their `gender` (Feminine, Masculine, Non-binary), `name`, what their GPA upon graduating from Starfleet Academy was (`starfleet_gpa`, 0-10 scale, 10 being the best grade), their perseverance score (`perseverance_score`) from their most recent psych assessment (0-10 scale, 10 being high perseverance). `week` indicates the weeks since the shore leave (1 to 12) and their `productivity` score for each week is recorded.

Each crewmember is assigned to a duty shift (`duty_shift`). There are four 8-hour shifts covering each 24 hour period, Alpha, Beta, Delta and Gamma. Within each duty shift, each crewmember is assigned to a team (`shift_team`). Teams are numbered 1 to 6, or sometimes fewer, and these labels aren't meaningful, they are just for administrative purposes. E.g., being Team 1 in Alpha shift has nothing to do with being Team 1 in Beta shift.

The crewmembers in Team 2 on the Gamma shift are assigned to work together as a unit, but they are only considered to be 'working' with other members of Team 2 on Gamma shift, not the rest of the Gamma shift, nor the crew in Team 2 of other shifts.

Your goal is to better understand productivity aboard your ship.

```
glimpse(crew_data)
```

```
## Rows: 3,012
## Columns: 13
## $ crew_id           <dbl> 42196, 42196, 42196, 42196, 42196, 42196, 42196, 42~
## $ rank              <chr> "Captain", "Captain", "Captain", "Captain", "Captai~
## $ position          <chr> "Captain", "Captain", "Captain", "Captain", "Captai~
## $ division          <chr> "Command", "Command", "Command", "Command", "Comman~
## $ sub_division      <chr> "Command", "Command", "Command", "Command", "Comman~
## $ gender            <chr> "Masculine", "Masculine", "Masculine", "Masculine",~
## $ name              <chr> "Ross Sisk", "Ross Sisk", "Ross Sisk", "Ross Sisk",~
## $ duty_shift        <chr> "Alpha", "Alpha", "Alpha", "Alpha", "Alpha", "Alpha~
## $ shift_team        <chr> "Team 1", "Team 1", "Team 1", "Team 1", "Team 1", "~
## $ starfleet_gpa     <dbl> 7.32, 7.32, 7.32, 7.32, 7.32, 7.32, 7.32, 7.32, 7.3~
## $ perseverance_score <dbl> 8.22, 8.22, 8.22, 8.22, 8.22, 8.22, 8.22, 8.22, 8.2~
## $ week              <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, ~
## $ productivity      <dbl> 37.59940, 34.92518, 35.66736, 34.53231, 31.03972, 2~
```

Task set 1: familiarize yourself with the data

1. What is the name of your ship? Hint: check out the object `ship_name`.

```
ship_name
```

```
## [1] "SS Sloocrot"
```

2. What is the name of the Communications Officer? Save it in the object `comms_officer`.

This object should be a character string, (i.e. `is.character(comms_officer)` should equal `TRUE`). Double-check spelling and capitalization, these will need to be an exact match to be marked correct.

```
# Get the name of comms_officer
comms_officer = crew_data %>% filter(position == "Communications Officer") %>%
                distinct(name) %>% as.character()

# String output check
is.character(comms_officer)
```

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

```
comms_officer
```

```
## [1] "Joseph O'Neill"
```

3. How many crewmembers are in this dataset? Save it in the object `n_crew`.

Enter the number of crew members as a number, e.g. 4 (not four). This object should be numeric, (i.e. `is.numeric(n_crew)` should equal `TRUE`).

```
# Get the number of crew members
n_crew = crew_data %>% distinct(crew_id) %>% nrow()

# Numeric output check
is.numeric(n_crew)
```

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

```
n_crew
```

```
## [1] 251
```

Task set 2: create/alter variables

1. The Records Officer lets your know that there is a typo in the crew dataset. They think it is to do with on of the engineering roles, (maybe in one of the position titles?) but unfortunately they can't remember where or how. Find the mistake, fix it (and save that fix in the original `crew_data`) and then calculate what proportion of people in the Engineering subdivision have 'engineer' or 'engineering' in their position title. You must use the properly corrected dataset in order to get the appropriate value.

Save this numeric value, between 0 and 1, as `prop_eng`. Round to two decimal points, e.g., 0.24 or 0.99 etc.

```
# Alter the spelling of 'Enigneering Technician' in position
crew_data = crew_data %>%
            mutate(position = str_replace(position, "Enigneering",
                                                    "Engineering"))

# Calculate the proportion of people in the Engineering subdivision have
# 'engineer' or 'engineering'
prop_eng = crew_data %>% filter(sub_division == "Engineering") %>%
           summarise(mean(str_detect(position, "Engineer"))) %>%
           as.numeric() %>%
           round(., 2)

prop_eng
```

```
## [1] 0.62
```

2. Create a new variable in `crew_data` called `full_team` that indicates both the duty shift and the team each person is assigned to.

- You may find the `str_c()` function useful.
- You can specify how the values you're sticking together are separated with the `sep` parameter, e.g., `str_c(var1, var2, sep = " ")` would put a space between the values of var1 and var2 when sticking them together.
- Don't forget that `mutate()` helps you make new variables.

```
# Create full_team by combining duty_shift and shift_team
crew_data = crew_data %>% mutate(full_team = str_c(duty_shift, shift_team,
                                                   sep = " "))
```

Task set 3: exploring week 1 data

1. Create a new dataset called `week1` that filters to only the observations for week 1. You must also reverse the levels of the `duty_shift` factor in `week1` so that the order is: Gamma, Delta, Beta, Alpha. You can test if you've achieved this by running `table(week1$duty_shift)`. The table should be ordered with Gamma first.

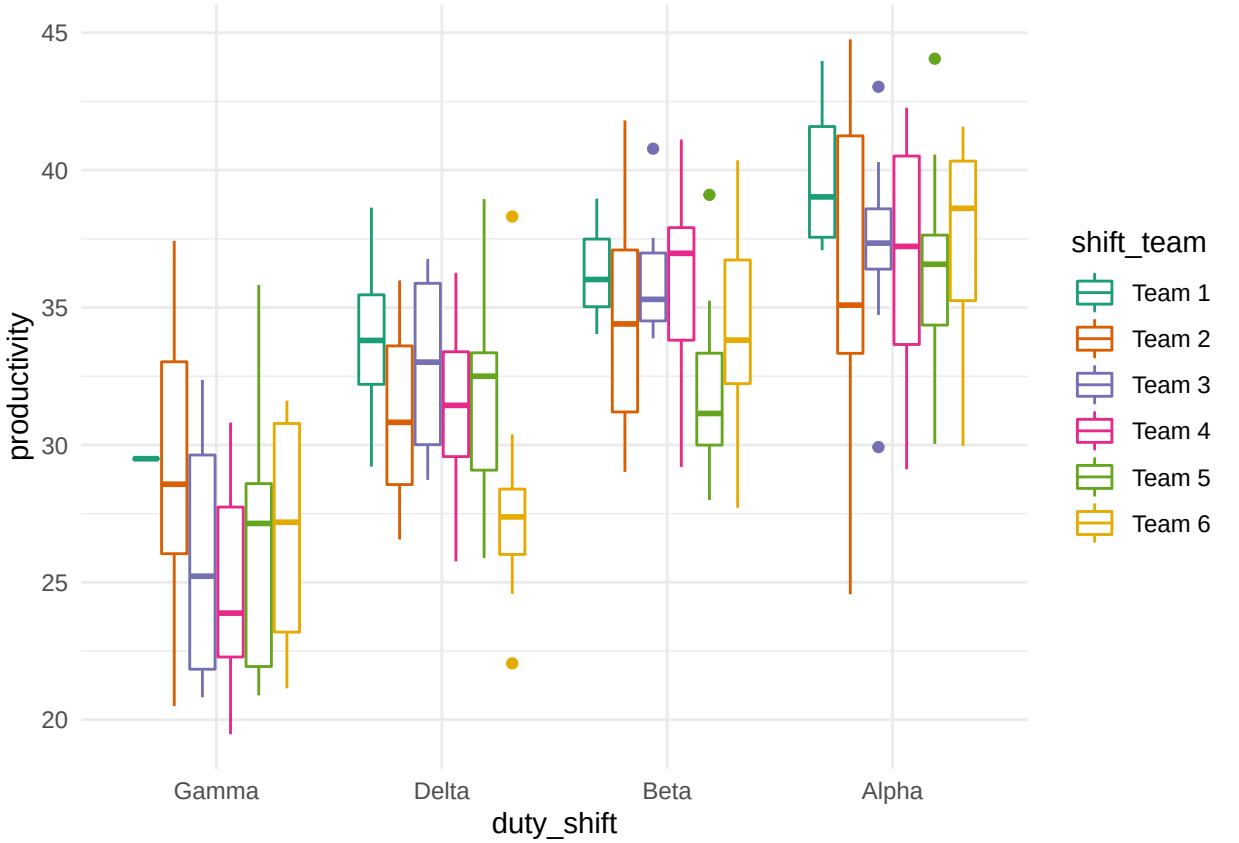
```
# Create week1
week1 = crew_data %>% filter(week == 1) %>%
        mutate(duty_shift = fct_rev(duty_shift))

table(week1$duty_shift)
```

```
##
## Gamma Delta  Beta Alpha
##    35    53    70    93
```

2. Using the `week1` dataset you created, create a plot with `productivity` on the y-axis, `duty_shift` on the x-axis and coloured boxplots for each `shift_team`. Use the "Dark2" colour palette from colour brewer.

- geom_boxplot() is the geometry that creates boxplots.
- use the colour aesthetic to get different boxplots for each `shift_team`
- `scale_colour_brewer()` will allow you to choose the Dark2 palette (when completed appropriately).

```
# Plot duty_shift vs productivity
week1 %>% ggplot(aes(x = duty_shift, y = productivity, colour = shift_team)) +
          geom_boxplot() +
          scale_color_brewer(palette = "Dark2") +
          theme_minimal()
```



3. Using the `week1` data, fit a linear model called `w1_shift` where `productivity` is the response and `duty_shift` is the only predictor. Run `summary` and `confint` on the model.

```
# Fit the linear model
w1_shift = lm(productivity ~ duty_shift, data = week1)

# Get summary and confidence interval
summary(w1_shift)
```

```
##
## Call:
## lm(formula = productivity ~ duty_shift, data = week1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.2509  -3.0296   0.2684   2.8438  10.5165
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      26.9120     0.6832  39.391  < 2e-16 ***
## duty_shiftDelta   3.8806     0.8803   4.408 1.56e-05 ***
## duty_shiftBeta    7.6496     0.8367   9.142  < 2e-16 ***
## duty_shiftAlpha   9.9047     0.8015  12.357  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.042 on 247 degrees of freedom
```

```
## Multiple R-squared:  0.4272, Adjusted R-squared:  0.4202
## F-statistic:  61.4 on 3 and 247 DF,  p-value: < 2.2e-16
```

```
confint(w1_shift)
```

```
##                      2.5 %    97.5 %
## (Intercept)     25.566349 28.257629
## duty_shiftDelta  2.146674  5.614540
## duty_shiftBeta   6.001560  9.297692
## duty_shiftAlpha  8.326025 11.483372
```

4. Fit three additional linear models and run summaries on them:

- Name the first model `w1_team`. It should have `productivity` as the response and then `shift_team` as the only predictor. `week1` is still the data to use.
- Name the first model `w1_int`. It should have `productivity` as the response and then the main effects and interaction of `duty_shift` and `shift_team` as the predictors. `week1` is still the data to use.
- Name the second model `w1_full`. It should have `productivity` as the response and `full_team` as the only predictor. `week1` is still the data to use.

```
# First model
w1_team = lm(productivity ~ shift_team, data = week1)

# Second model
w1_int = lm(productivity ~ shift_team * duty_shift, data = week1)

# Third model
w1_full = lm(productivity ~ full_team, data = week1)

# Run summaries
summary(w1_team)
```

```
##
## Call:
## lm(formula = productivity ~ shift_team, data = week1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.6407  -3.5887   0.3758   3.5341  11.3242
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         36.824      1.356  27.153  < 2e-16 ***
## shift_teamTeam 2    -3.387      1.521  -2.226  0.02690 *
## shift_teamTeam 3    -2.678      1.703  -1.573  0.11709
## shift_teamTeam 4    -2.710      1.527  -1.774  0.07724 .
## shift_teamTeam 5    -4.041      1.527  -2.646  0.00867 **
## shift_teamTeam 6    -4.537      1.590  -2.853  0.00470 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.252 on 245 degrees of freedom
## Multiple R-squared:  0.0405, Adjusted R-squared:  0.02092
## F-statistic: 2.068 on 5 and 245 DF,  p-value: 0.07
```

```
summary(w1_int)
```

```
##
## Call:
## lm(formula = productivity ~ shift_team * duty_shift, data = week1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.4138  -2.4252  -0.2423   2.6447  10.5397
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         29.4978     3.9508   7.466 1.75e-12 ***
## shift_teamTeam 2                    -0.6117     4.1437  -0.148   0.8828
## shift_teamTeam 3                    -3.5228     4.3279  -0.814   0.4165
## shift_teamTeam 4                    -4.6114     4.2236  -1.092   0.2761
## shift_teamTeam 5                    -2.9532     4.1905  -0.705   0.4817
## shift_teamTeam 6                    -2.7168     4.4171  -0.615   0.5391
## duty_shiftDelta                      4.3664     4.4171   0.989   0.3240
## duty_shiftBeta                       6.8433     4.5620   1.500   0.1350
## duty_shiftAlpha                     10.2708     4.2236   2.432   0.0158 *
## shift_teamTeam 2:duty_shiftDelta    -2.2437     4.7574  -0.472   0.6377
## shift_teamTeam 3:duty_shiftDelta     2.5392     5.1512   0.493   0.6225
## shift_teamTeam 4:duty_shiftDelta     1.9804     4.8272   0.410   0.6820
## shift_teamTeam 5:duty_shiftDelta     0.5777     4.7606   0.121   0.9035
## shift_teamTeam 6:duty_shiftDelta    -3.3763     4.9713  -0.679   0.4977
## shift_teamTeam 2:duty_shiftBeta     -1.3291     4.8261  -0.275   0.7833
## shift_teamTeam 3:duty_shiftBeta      3.3577     5.1512   0.652   0.5152
## shift_teamTeam 4:duty_shiftBeta      4.3363     4.8949   0.886   0.3766
## shift_teamTeam 5:duty_shiftBeta     -1.4563     4.8953  -0.297   0.7664
## shift_teamTeam 6:duty_shiftBeta      0.6756     5.0822   0.133   0.8944
## shift_teamTeam 2:duty_shiftAlpha    -3.1773     4.4881  -0.708   0.4797
## shift_teamTeam 3:duty_shiftAlpha     0.9664     4.7307   0.204   0.8383
## shift_teamTeam 4:duty_shiftAlpha     1.6949     4.5583   0.372   0.7104
## shift_teamTeam 5:duty_shiftAlpha    -0.4961     4.5276  -0.110   0.9129
## shift_teamTeam 6:duty_shiftAlpha     0.0374     4.8272   0.008   0.9938
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.951 on 227 degrees of freedom
## Multiple R-squared:  0.497,  Adjusted R-squared:  0.446
## F-statistic: 9.752 on 23 and 227 DF,  p-value: < 2.2e-16
```

```
summary(w1_full)
```

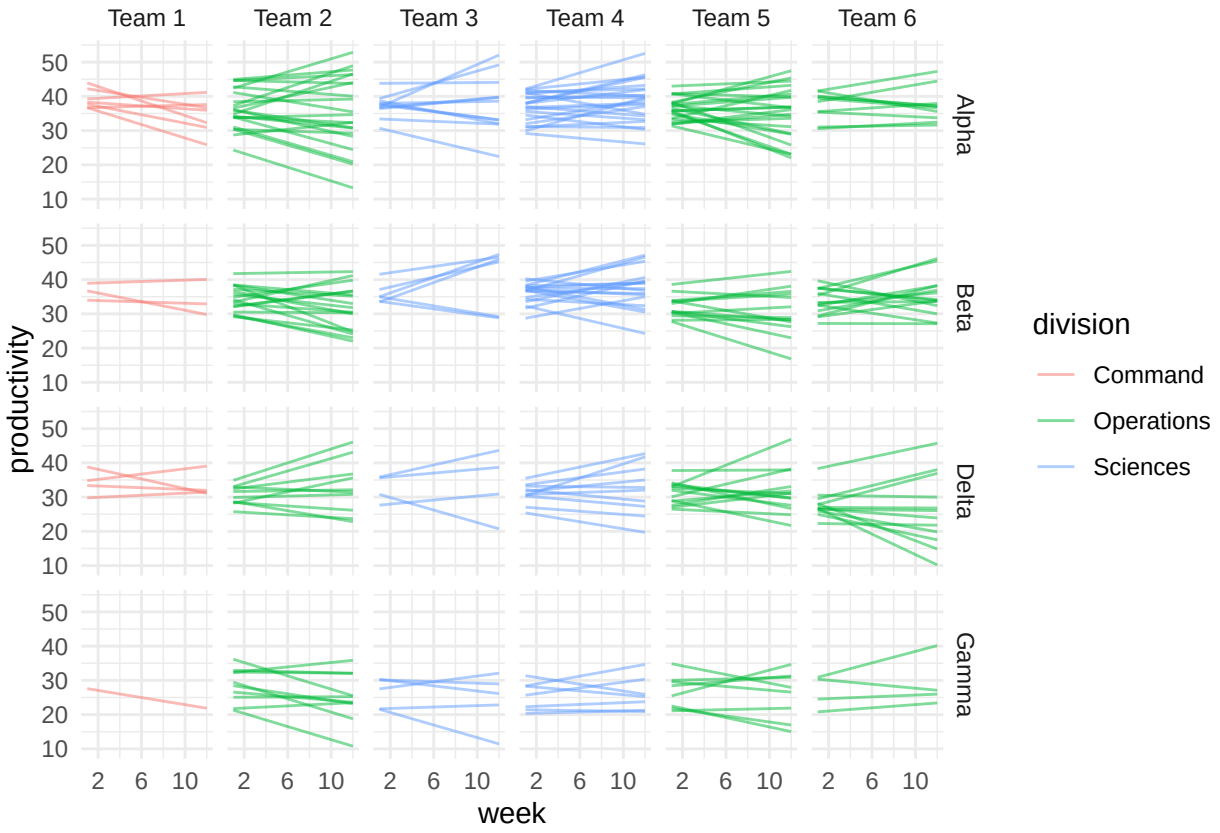
```
##
## Call:
## lm(formula = productivity ~ full_team, data = week1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.4138  -2.4252  -0.2423   2.6447  10.5397
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               39.769      1.493  26.632  < 2e-16 ***
## full_teamAlpha Team 2     -3.789      1.724  -2.197  0.02900 *
```

```
## full_teamAlpha Team 3   -2.556      1.910  -1.338  0.18213
## full_teamAlpha Team 4   -2.917      1.714  -1.701  0.09029 .
## full_teamAlpha Team 5   -3.449      1.714  -2.012  0.04542 *
## full_teamAlpha Team 6   -2.679      1.947  -1.376  0.17012
## full_teamBeta Team 1    -3.428      2.726  -1.257  0.20997
## full_teamBeta Team 2    -5.368      1.774  -3.026  0.00277 **
## full_teamBeta Team 3    -3.593      2.198  -1.635  0.10354
## full_teamBeta Team 4    -3.703      1.774  -2.087  0.03802 *
## full_teamBeta Team 5    -7.837      1.852  -4.231 3.37e-05 ***
## full_teamBeta Team 6    -5.469      1.829  -2.990  0.00309 **
## full_teamDelta Team 1   -5.904      2.476  -2.384  0.01793 *
## full_teamDelta Team 2   -8.760      1.947  -4.499 1.09e-05 ***
## full_teamDelta Team 3   -6.888      2.476  -2.782  0.00586 **
## full_teamDelta Team 4   -8.535      1.947  -4.384 1.78e-05 ***
## full_teamDelta Team 5   -8.280      1.852  -4.470 1.23e-05 ***
## full_teamDelta Team 6  -11.997      1.879  -6.385 9.56e-10 ***
## full_teamGamma Team 1  -10.271      4.224  -2.432  0.01580 *
## full_teamGamma Team 2  -10.883      1.947  -5.589 6.50e-08 ***
## full_teamGamma Team 3  -13.794      2.313  -5.963 9.42e-09 ***
## full_teamGamma Team 4  -14.882      2.112  -7.047 2.17e-11 ***
## full_teamGamma Team 5  -13.224      2.045  -6.467 6.05e-10 ***
## full_teamGamma Team 6  -12.988      2.476  -5.245 3.58e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.951 on 227 degrees of freedom
## Multiple R-squared:  0.497,  Adjusted R-squared:  0.446
## F-statistic: 9.752 on 23 and 227 DF,  p-value: < 2.2e-16
```

Task set 4: productivity post shore leave

1. Replace the 1s and add whatever other aesthetics are required in the aesthetic statement in the `ggplot()` function to recreate the graph below for your particular ship. Note that each line represents the productivity trend for one crewmember over the 12 week period.

```
crew_data %>%
  ggplot(aes(y = productivity, x = week, group = crew_id, colour = division)) +
  geom_line(stat="smooth", method = "lm", formula = 'y~x', alpha = 0.5) +
  facet_grid(duty_shift~shift_team) +
  scale_x_continuous(breaks = seq(2,12, by = 4)) +
  theme_minimal()
```



After discussing your investigation and the above graph with your Personnel Officer, they suggest you should *not* include rank, position, division, sub-division or gender in your analysis. They also tell you that ship-to-ship, how duty shifts are set up and how teams are allocated differs quite a lot. Some ships have more than the 4 shifts yours does, or have many more teams due to size, etc.

You're interested in presenting your work at the next Federation Science and Innovation Conference and want be able to provide information that might be relevant to the the Chief Science Officers on other ships, too.

Below are several models that you've fit and some tests on them.

```
model_1a <- lmer(productivity ~ week + starfleet_gpa + perseverance_score +
                   (1|name),
                 data = crew_data)

model_1b <- lmer(productivity ~ week + starfleet_gpa + perseverance_score +
                   (1 + week|name),
```

```
                  data = crew_data)

# Study prompt: How do we interpret the p-values here? What is relevant?
lmtest::lrtest(model_1a, model_1b)
```

```
## Likelihood ratio test
##
## Model 1: productivity ~ week + starfleet_gpa + perseverance_score + (1 |
##     name)
## Model 2: productivity ~ week + starfleet_gpa + perseverance_score + (1 +
##     week | name)
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   6 -7182.8
## 2   8 -5418.8  2 3528.1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Very important!
# Change eval=FALSE to eval=TRUE in the chunk options for this chunk.

model_2a <- lmer(productivity ~ week + starfleet_gpa + perseverance_score +
                   (1 + week|name) + (1|duty_shift:shift_team),
                 data = crew_data)

model_2b <- lmer(productivity ~ week + starfleet_gpa + perseverance_score +
                   (1 + week|name) + (1|full_team),
                 data = crew_data)
```

```
# Very important!
# Change eval=FALSE to eval=TRUE in the chunk options for this chunk

# Study prompt: How do we interpret the p-values here? What is relevant?
lmtest::lrtest(model_1b, model_2a)
```

```
## Likelihood ratio test
##
## Model 1: productivity ~ week + starfleet_gpa + perseverance_score + (1 +
##     week | name)
## Model 2: productivity ~ week + starfleet_gpa + perseverance_score + (1 +
##     week | name) + (1 | duty_shift:shift_team)
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   8 -5418.8
## 2   9 -5353.3  1 130.98  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lmtest::lrtest(model_2a, model_2b)
```

```
## Likelihood ratio test
##
## Model 1: productivity ~ week + starfleet_gpa + perseverance_score + (1 +
##     week | name) + (1 | duty_shift:shift_team)
## Model 2: productivity ~ week + starfleet_gpa + perseverance_score + (1 +
##     week | name) + (1 | full_team)
##   #Df  LogLik Df Chisq Pr(>Chisq)
## 1   9 -5353.3
```

```
## 2   9 -5353.3  0     0          1
```

Task set 5: competitive astrobiologists

While on shore leave, some of the astrobiologists had a little competition to see who could spot plants from the greatest number of **different planets or systems** in the hotel gardens. Note: The *number* of plants spotted doesn't actually matter as long as at least one was spotted.

They have asked for your impartial help to find out who the winner is.

You have three datasets:

- `astrobiologists` is a list of all the astrobiology crewmembers
- `competition_data` has the number of plants of each type that each participating astrobiologist recorded.
- `origin_data` contains information from the hotel about the plants in their collection and the the planets they are native to. They have warned you that is may be somewhat incomplete.

```
astrobiologists <- crew_data %>%
  filter(position == "Astrobiologist") %>%
  distinct(crew_id, name, .keep_all=TRUE) %>%
  transmute(crewmember = str_c(name, " (", crew_id, ")"))

competition_data <- tibble(crewmember =
                           c(astrobiologists$crewmember[1],
                           astrobiologists$crewmember[2],
                           astrobiologists$crewmember[3]),
          `Xupta tree` = c(3L, 7L, NA),
            `L'maki` = c(21L, NA, 21L),
          `Leola root` = c(40L, 45L, 26L),
            Klavaatu = c(2L, 3L, 2L),
           Waterplum = c(NA, 5L, 1L),
  `Folnar jewel plant` = c(17L, 12L, 10L),
        `Felaran rose` = c(28L, 7L, NA),
           Crystilia = c(12L, 3L, 9L),
             Parthas = c(4L, 3L, NA),
        `Borgia plant` = c(NA, 1L, 1L))


origin_data <- data.frame(plant = c("Xupta tree","L'maki","Leola root",
                                    "Waterplum","Vulcan orchid",
                                    "Lunar flower","Garlanic tree",
                                    "Folnar jewel plant",
                                    "Felaran rose","Crystilia","Parthas",
                                    "Borgia plant","Pod plant"),
                          native_to = c("Orellius system","Delta Quadrant",
                                    "Bajor","Mari","Vulcan",
                                    NA,"Elaysian homeworld","Folnar III",
                                    "Delta Quadrant","Telemarius IV",
                                    "Acamar III","M-113",NA))
```

Tip: I recommend running `View()` on `competition_data` and `origin_data` to explore them further so you are familiar with their structure and contents. (You can also do this by clicking on their titles in the Environment pane.)

1. Create a new dataset called `complete_comp` using the `competition_data`.
2. Assess whether `complete_comp`, at this current step, is currently tidy. (I.e., is `competition_data` tidy?) If yes, proceed. If no, alter it to be tidy. Specifically, it needs to be in the correct format to be useful for merging the `origin_data` on to it.

3. Continuing to manipulate the `complete_comp` object, merge on the `origin_data` such that any plants **not** present in the data provided by the hotel are **dropped**.
4. Restrict the `complete_comp` so it only contains rows where at least one plant was spotted.
5. Restrict the `complete_comp` to just observations from distinct planets or systems for each crewmember. (See hint code below.)
6. Calculate how many unique planets or systems each astrobiologist spotted at least one plant from. I suggest `count()` will be of help.

Use the following structure of code to complete these tasks.

```
complete_comp <- competition_data %>%
  ____________________ %>%
  ____________________ %>%
  ____________________ %>%
  distinct(crewmember, native_to) %>% # this line will achieve instruction 5
  ____________________  # achieve instruction 6
```

```
# Create the new dataset
complete_comp <- competition_data %>%
        pivot_longer(-crewmember, names_to = "plant",
                     values_to = "number") %>% # Task 2
        right_join(origin_data, by = "plant") %>% # Task 3
        filter(number > 0) %>% # Task 4
        distinct(crewmember, native_to) %>% # Task 5
        count(crewmember) # Task 6

complete_comp
```

```
## # A tibble: 3 x 2
##   crewmember                  n
##   <chr>                   <int>
## 1 Destiny Nigussu (42054)     6
## 2 Israa el-Rahmani (42094)    8
## 3 Jawhara el-Matar (42104)    6
```

Task set 5: Restoring native flora on Risa

Note: There is no code required for this task, just read and understand the structure of the study.

Tourism is a main part of the economy of the planet Risa. Extensive modification of climate and seismic activity have been undertaken to ensure optimal comfort for visitors. Researchers at a small astrobotany station were interested in understanding how different soil types influence the growth of two native species of dune grasses.

As part of their introductory training, junior astrobotanists collected weekly data on plants in pots containing sand samples. Data were examined to compare:

- **growth** (in mm) of two species of **plants**—— Amnophila Picardus and Amnophila Janewayus
- Sand from a busy tourist beach, sand from a private resort beach with minimal disturbance, and the area around a seismic stabilization unit.
- the effect of **sterilization**: half of the sampled sand was sterilized to determine if rhizosphere differences were responsible for the observed variation.

Additionally, it is worth noting that there were multiple plants in each pot and only one sand type and sterilization status per pot.

Each pot contained 4 plants, 2 of each species. There were 20 pots for each soil type, with half sterilized and the other half not sterilized. Data was collected over a **12 week period**.

Checklist

- Updated name and student ID number on lines 4 and 12 (i.e., in the YAML)
- Other than the `setup` chunk (where the packages are installed), **all chunks should have eval=TRUE in the chunk options** (you don't usually need to say this explicitly in each chunk option, but it makes the autograding simpler)
- **Values**
  - `studentIDnum`, your full student ID number
  - `comms_officer`
  - `n_crew`
  - `prop_eng`, rounded to 2 decimal places
- **Tibbles**
  - `crew_data`, with added column `full_team`
  - `week1`, with reordered factor variables
  - `complete_comp`
- PDF knit from RMD (directly, no interim HTML/Word step)
- Rmd and PDF files names correctly.
- Rmd and PDF files submitted to MarkUs BEFORE 8:00 p.m. ET