STA303/1002: Mixed assessment 1

Starship crew analysis

Chief Science Officer Elric Francis Lazaro; ID: 1003457644

Information	Note
Name	Mixed assignment 1
Type	Type 2
Value	14%
Due	This untimed submission must be submitted by 4:30 p.m. ET Wednesday, Feb 24
Submission link	PDF & RMD: https://q.utoronto.ca/courses/204826/assignments/415116
Accommodations and extension policy	If you miss a type 2 assessment due to illness or a serious personal emergency, please complete **this form** within ONE week of the due date of the assignment (i.e. the end of the timed assessment window).

Mixed assessment 1 has two components:

- Untimed guided analysis (this)
- Timed assessment (50 minutes; 24-hour assessment window is 4:30 p.m. ET Tuesday, Feb 23 to 4:30 p.m. ET Wednesday, Feb 24)
- See the mixed assessments overview page for further information and revisions links.

Instructions

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. Once you've written your code and are ready to knit change `knitr::opts_chunk$set(eval = FALSE)` to read `eval = TRUE` in the libraries chunk.
5. Knit this .Rmd to .pdf and submit BOTH files to the submission link in the table above.

Note: This component is ungraded BUT there is a 10-percentage point penalty for not submitting your .Rmd and .pdf to the Quercus dropbox by the end of the assessment window, i.e., by Wednesday, February 24 at 4:30 p.m. ET. The intention is to allow confirmation of your work for academic integrity purposes and/or if as a way to confirm your personalized data if there are issues.

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

```
# 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)

# Once you've updated the code and are ready to knit
# change this to eval = TRUE
knitr::opts_chunk$set(eval = TRUE)
```

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
get_my_starship(1003457644)

# 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 Strow. 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> 42235, 42235, 42235, 42235, 42235, 42235, 42235, 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> "Feminine", "Feminine", "Feminine", "Feminine", "Fe~
## $ name               <chr> "Trai Hussein", "Trai Hussein", "Trai Hussein", "Tr~
## $ 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> 8.30, 8.30, 8.30, 8.30, 8.30, 8.30, 8.30, 8.30, 8.3~
## $ perseverance_score <dbl> 8.11, 8.11, 8.11, 8.11, 8.11, 8.11, 8.11, 8.11, 8.1~
## $ week               <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, ~
## $ productivity       <dbl> 39.35186, 37.43893, 38.46483, 37.52857, 36.05468, 3~
```

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 Strow"
```

2. What is the name of the Communications Officer?

```
crew_data$name[crew_data$position == "Communications Officer"][1]
```

```
## [1] "Kayla Rogers"
```

3. How many crewmembers are in this dataset?

```
length(unique(crew_data$crew_id))
```

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

Task set 2: create/alter variables

1. The Records Officer lets your know that there is a typo in the crew dataset, where 'Engineering' has been misspelled somewhere, (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.

```
## Check all possible positions and check for mispelling
# unique(crew_data$position)
## Found 'Enigeering'

## Correct mispelling
crew_data <- crew_data %>%
  mutate(position = str_replace(position, "Enigneering", "Engineering"))
## Check if typo corrected
# unique(crew_data$position)

complete_engineers <- crew_data %>%
  filter(sub_division == "Engineering")
engineers_pos <- complete_engineers %>%
  filter(str_detect(str_to_lower(position), "engineer") | str_detect(str_to_lower(position), "engineerin

length(unique(engineers_pos$crew_id))/length(unique(complete_engineers$crew_id))
```

```
## [1] 0.6206897
```

About 62.07% of people in Engineering subdivision have 'engineer' or 'engineering' in their position title

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.

```
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.

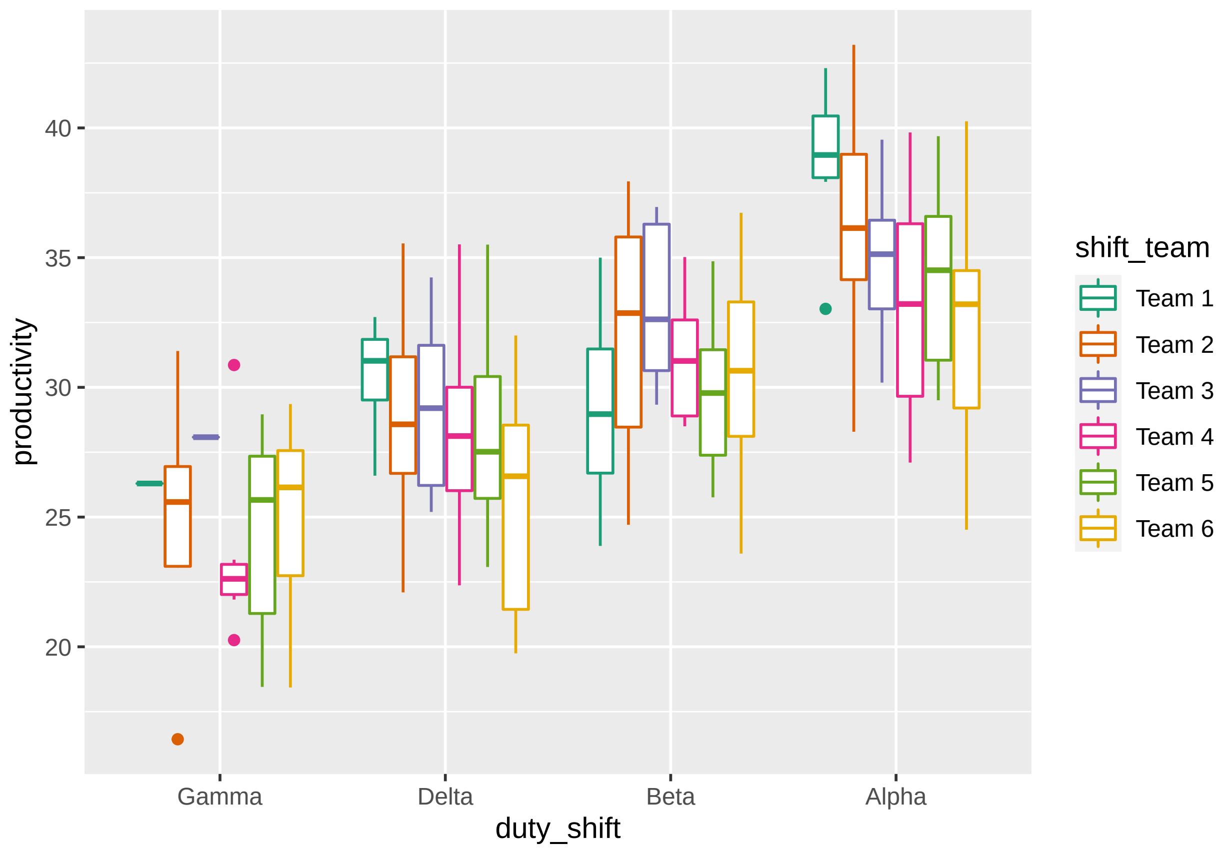
```
week1 <- crew_data %>%
  filter(week == 1) %>%
  mutate(duty_shift = factor(duty_shift,
                             levels = c("Gamma", "Delta", "Beta", "Alpha")))

table(week1$duty_shift)
```

```
##
## Gamma Delta  Beta Alpha
##    26    66    63    96
```

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).

```
week1 %>%
  ggplot(aes(x = duty_shift, y = productivity, color = shift_team)) +
  geom_boxplot() +
  scale_color_brewer(palette = "Dark2")
```



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.

```
w1_shift <- lm(productivity ~ duty_shift, data = week1)

summary(w1_shift)
```

```
##
## Call:
## lm(formula = productivity ~ duty_shift, data = week1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8760 -2.4421  0.0123  2.8444  8.8161
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      24.6303     0.7570  32.535  < 2e-16 ***
## duty_shiftDelta   3.6703     0.8938   4.106 5.47e-05 ***
## duty_shiftBeta    6.5361     0.8998   7.264 4.90e-12 ***
## duty_shiftAlpha   9.7574     0.8534  11.433  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.86 on 247 degrees of freedom
```

```
## Multiple R-squared:  0.4176, Adjusted R-squared:  0.4105
## F-statistic: 59.03 on 3 and 247 DF,  p-value: < 2.2e-16
```

```r
confint(w1_shift)
```

```
##                      2.5 %    97.5 %
## (Intercept)     23.139167 26.121343
## duty_shiftDelta  1.909848  5.430760
## duty_shiftBeta   4.763791  8.308315
## duty_shiftAlpha  8.076465 11.438309
```

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.

```r
w1_team <- lm(productivity ~ shift_team, data = week1)

summary(w1_team)
```

```
##
## Call:
## lm(formula = productivity ~ shift_team, data = week1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.9718  -3.0778   0.0482   3.9301  11.0854
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        33.0980     1.2670  26.123  < 2e-16 ***
## shift_teamTeam 2   -0.6894     1.4214  -0.485  0.62812
## shift_teamTeam 3   -1.0479     1.5910  -0.659  0.51076
## shift_teamTeam 4   -2.2875     1.4266  -1.603  0.11013
## shift_teamTeam 5   -3.2548     1.4266  -2.281  0.02338 *
## shift_teamTeam 6   -3.9273     1.4857  -2.643  0.00874 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.907 on 245 degrees of freedom
## Multiple R-squared:  0.06651,    Adjusted R-squared:  0.04746
## F-statistic: 3.491 on 5 and 245 DF,  p-value: 0.004567
```

```r
confint(w1_team)
```

```
##                       2.5 %     97.5 %
## (Intercept)       30.602439 35.5935799
```

```
## shift_teamTeam 2 -3.489099  2.1103753
## shift_teamTeam 3 -4.181696  2.0859592
## shift_teamTeam 4 -5.097478  0.5225050
## shift_teamTeam 5 -6.064778 -0.4447951
## shift_teamTeam 6 -6.853589 -1.0009578
```

```
w1_int <- lm(productivity ~ duty_shift + shift_team + duty_shift*shift_team, data = week1)

summary(w1_int)
```

```
##
## Call:
## lm(formula = productivity ~ duty_shift + shift_team + duty_shift *
##     shift_team, data = week1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.257 -2.681  0.000  2.633  8.163
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       26.2938     3.7452   7.021 2.53e-11 ***
## duty_shiftDelta                    4.0444     4.1873   0.966  0.33513
## duty_shiftBeta                     2.9125     4.1873   0.696  0.48742
## duty_shiftAlpha                   12.3727     4.0453   3.059  0.00249 **
## shift_teamTeam 2                  -1.6001     4.1027  -0.390  0.69690
## shift_teamTeam 3                   1.7855     5.2965   0.337  0.73634
## shift_teamTeam 4                  -2.7047     4.0453  -0.669  0.50443
## shift_teamTeam 5                  -1.8825     4.0453  -0.465  0.64212
## shift_teamTeam 6                  -1.3590     4.0038  -0.339  0.73460
## duty_shiftDelta:shift_teamTeam 2   0.2749     4.6279   0.059  0.95269
## duty_shiftBeta:shift_teamTeam 2    4.4191     4.6060   0.959  0.33837
## duty_shiftAlpha:shift_teamTeam 2  -0.9555     4.4446  -0.215  0.82998
## duty_shiftDelta:shift_teamTeam 3  -2.8054     5.7413  -0.489  0.62557
## duty_shiftBeta:shift_teamTeam 3    2.2620     5.7934   0.390  0.69658
## duty_shiftAlpha:shift_teamTeam 3  -5.5438     5.6696  -0.978  0.32921
## duty_shiftDelta:shift_teamTeam 4   0.8419     4.6124   0.183  0.85532
## duty_shiftBeta:shift_teamTeam 4    4.6029     4.5771   1.006  0.31567
## duty_shiftAlpha:shift_teamTeam 4  -2.8232     4.3843  -0.644  0.52027
## duty_shiftDelta:shift_teamTeam 5  -0.5288     4.5357  -0.117  0.90729
## duty_shiftBeta:shift_teamTeam 5    2.6099     4.5687   0.571  0.56839
## duty_shiftAlpha:shift_teamTeam 5  -2.5874     4.4248  -0.585  0.55930
## duty_shiftDelta:shift_teamTeam 6  -3.1080     4.5930  -0.677  0.49930
## duty_shiftBeta:shift_teamTeam 6    3.0465     4.5930   0.663  0.50782
## duty_shiftAlpha:shift_teamTeam 6  -5.2140     4.3936  -1.187  0.23658
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.745 on 227 degrees of freedom
## Multiple R-squared:  0.4962, Adjusted R-squared:  0.4451
## F-statistic: 9.719 on 23 and 227 DF,  p-value: < 2.2e-16
```

```r
confint(w1_int)
```

```
##                                         2.5 %    97.5 %
## (Intercept)                         18.913913 33.673599
## duty_shiftDelta                     -4.206479 12.295351
## duty_shiftBeta                      -5.338430 11.163400
## duty_shiftAlpha                      4.401545 20.343829
## shift_teamTeam 2                    -9.684297  6.484129
## shift_teamTeam 3                    -8.651152 12.222196
## shift_teamTeam 4                   -10.675809  5.266474
## shift_teamTeam 5                    -9.853640  6.088643
## shift_teamTeam 6                    -9.248402  6.530366
## duty_shiftDelta:shift_teamTeam 2  -8.844314  9.394043
## duty_shiftBeta:shift_teamTeam 2   -4.656925 13.495089
## duty_shiftAlpha:shift_teamTeam 2  -9.713393  7.802402
## duty_shiftDelta:shift_teamTeam 3 -14.118534  8.507637
## duty_shiftBeta:shift_teamTeam 3   -9.153774 13.677782
## duty_shiftAlpha:shift_teamTeam 3 -16.715585  5.628000
## duty_shiftDelta:shift_teamTeam 4  -8.246572  9.930427
## duty_shiftBeta:shift_teamTeam 4   -4.416219 13.621964
## duty_shiftAlpha:shift_teamTeam 4 -11.462219  5.815886
## duty_shiftDelta:shift_teamTeam 5  -9.466257  8.408610
## duty_shiftBeta:shift_teamTeam 5   -6.392584 11.612389
## duty_shiftAlpha:shift_teamTeam 5 -11.306355  6.131534
## duty_shiftDelta:shift_teamTeam 6 -12.158330  5.942415
## duty_shiftBeta:shift_teamTeam 6   -6.003882 12.096863
## duty_shiftAlpha:shift_teamTeam 6 -13.871342  3.443414
```

```r
w1_team <- lm(productivity ~ full_team, data = week1)

summary(w1_team)
```

```
##
## Call:
## lm(formula = productivity ~ full_team, data = week1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.257 -2.681  0.000  2.633  8.163
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)              38.666      1.529  25.289  < 2e-16 ***
## full_teamAlpha Team 2   -2.556      1.709  -1.495 0.136311
## full_teamAlpha Team 3   -3.758      2.023  -1.858 0.064451 .
## full_teamAlpha Team 4   -5.528      1.690  -3.270 0.001242 **
## full_teamAlpha Team 5   -4.470      1.793  -2.493 0.013376 *
## full_teamAlpha Team 6   -6.573      1.809  -3.633 0.000346 ***
## full_teamBeta Team 1    -9.460      2.418  -3.913 0.000120 ***
## full_teamBeta Team 2    -6.641      1.793  -3.704 0.000266 ***
## full_teamBeta Team 3    -5.413      2.084  -2.598 0.009999 **
## full_teamBeta Team 4    -7.562      1.848  -4.091 5.97e-05 ***
```

```
## full_teamBeta Team 5    -8.733      1.827  -4.779 3.17e-06 ***
## full_teamBeta Team 6    -7.773      1.974  -3.938 0.000109 ***
## full_teamDelta Team 1   -8.328      2.418  -3.445 0.000680 ***
## full_teamDelta Team 2   -9.653      1.848  -5.222 3.99e-07 ***
## full_teamDelta Team 3   -9.348      1.934  -4.834 2.47e-06 ***
## full_teamDelta Team 4  -10.191      1.934  -5.269 3.18e-07 ***
## full_teamDelta Team 5  -10.740      1.743  -6.160 3.27e-09 ***
## full_teamDelta Team 6  -12.795      1.974  -6.482 5.57e-10 ***
## full_teamGamma Team 1  -12.373      4.045  -3.059 0.002491 **
## full_teamGamma Team 2  -13.973      2.268  -6.161 3.25e-09 ***
## full_teamGamma Team 3  -10.587      4.045  -2.617 0.009463 **
## full_teamGamma Team 4  -15.077      2.162  -6.973 3.35e-11 ***
## full_teamGamma Team 5  -14.255      2.162  -6.593 2.99e-10 ***
## full_teamGamma Team 6  -13.732      2.084  -6.590 3.03e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.745 on 227 degrees of freedom
## Multiple R-squared:  0.4962, Adjusted R-squared:  0.4451
## F-statistic: 9.719 on 23 and 227 DF,  p-value: < 2.2e-16
```

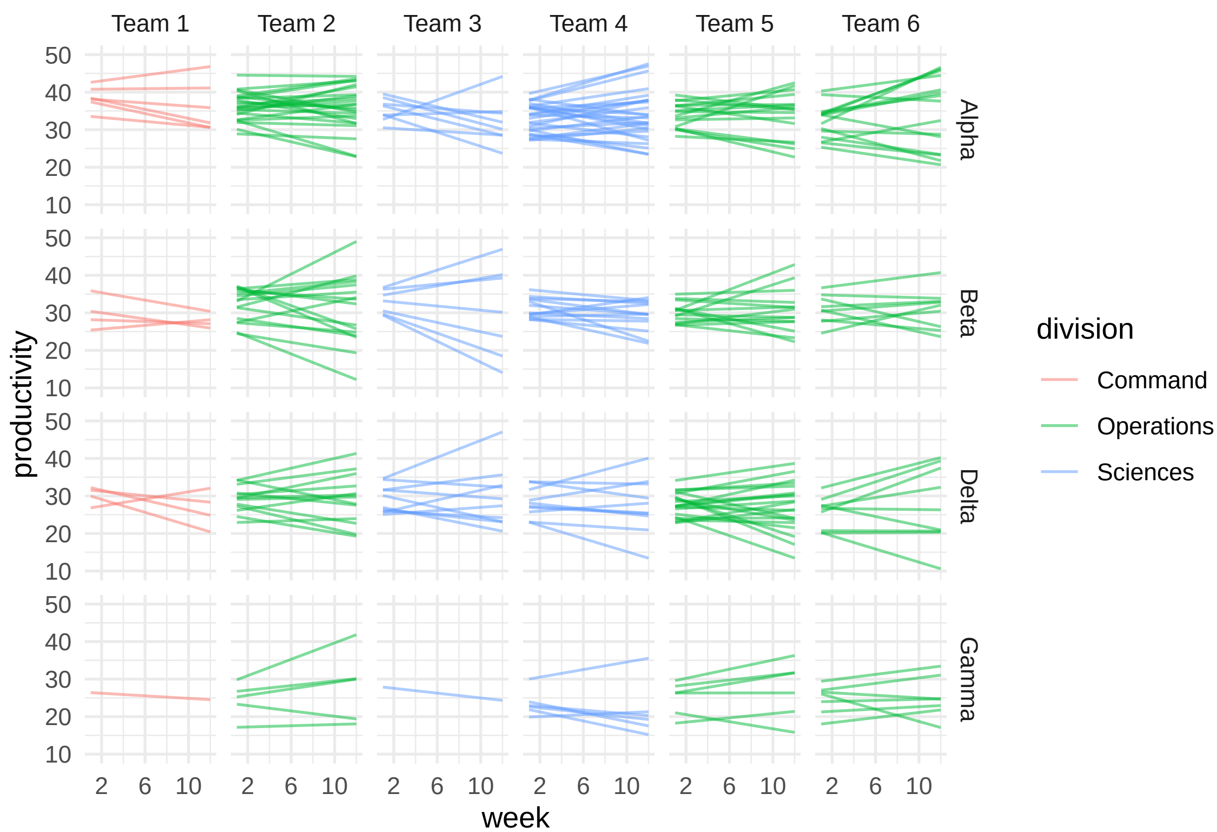
```r
confint(w1_team)
```

```
##                             2.5 %      97.5 %
## (Intercept)            35.653634  41.6792510
## full_teamAlpha Team 2  -5.924001   0.8128429
## full_teamAlpha Team 3  -7.743841   0.2273001
## full_teamAlpha Team 4  -8.858619  -2.1970489
## full_teamAlpha Team 5  -8.002740  -0.9370782
## full_teamAlpha Team 6 -10.137785  -3.0081793
## full_teamBeta Team 1  -14.223870  -4.6965338
## full_teamBeta Team 2  -10.174034  -3.1083725
## full_teamBeta Team 3   -9.518442  -1.3069106
## full_teamBeta Team 4  -11.204304  -3.9196890
## full_teamBeta Team 5  -12.333793  -5.1318039
## full_teamBeta Team 6  -11.662248  -3.8832108
## full_teamDelta Team 1 -13.091919  -3.5645828
## full_teamDelta Team 2 -13.295778  -6.0111626
## full_teamDelta Team 3 -13.159112  -5.5372431
## full_teamDelta Team 4 -14.001926  -6.3800564
## full_teamDelta Team 5 -14.174704  -7.3044435
## full_teamDelta Team 6 -16.684746  -8.9057079
## full_teamGamma Team 1 -20.343829  -4.4015454
## full_teamGamma Team 2 -18.441487  -9.5040536
## full_teamGamma Team 3 -18.558307  -2.6160236
## full_teamGamma Team 4 -19.338109 -10.8165998
## full_teamGamma Team 5 -18.515940  -9.9944315
## full_teamGamma Team 6 -17.837471  -9.6259394
```

Task set 4: productivity post shore leave

1. Replace the 1s and add whatver 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, color = division, group = crew_id)) +
  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.

```
# You can ignore this line, it is just to set things up so
# you hopefully don't get a convergence error.
```

```r
# If you do, don't worry about it if the model parameters are estimated.
lmerControl(optCtrl=list(xtol_abs=1e-8,
                         ftol_abs=1e-8,
                         optimizer = "Nelder_Mead"))
```

```
## $optimizer
## [1] "nloptwrap"
##
## $restart_edge
## [1] TRUE
##
## $boundary.tol
## [1] 1e-05
##
## $calc.derivs
## [1] TRUE
##
## $use.last.params
## [1] FALSE
##
## $checkControl
## $checkControl$check.nobs.vs.rankZ
## [1] "ignore"
##
## $checkControl$check.nobs.vs.nlev
## [1] "stop"
##
## $checkControl$check.nlev.gtreq.5
## [1] "ignore"
##
## $checkControl$check.nlev.gtr.1
## [1] "stop"
##
## $checkControl$check.nobs.vs.nRE
## [1] "stop"
##
## $checkControl$check.rankX
## [1] "message+drop.cols"
##
## $checkControl$check.scaleX
## [1] "warning"
##
## $checkControl$check.formula.LHS
## [1] "stop"
##
##
## $checkConv
## $checkConv$check.conv.grad
## $checkConv$check.conv.grad$action
## [1] "warning"
##
## $checkConv$check.conv.grad$tol
## [1] 0.002
```

```
##
## $checkConv$check.conv.grad$relTol
## NULL
##
##
## $checkConv$check.conv.singular
## $checkConv$check.conv.singular$action
## [1] "message"
##
## $checkConv$check.conv.singular$tol
## [1] 1e-04
##
##
## $checkConv$check.conv.hess
## $checkConv$check.conv.hess$action
## [1] "warning"
##
## $checkConv$check.conv.hess$tol
## [1] 1e-06
##
##
##
## $optCtrl
## $optCtrl$xtol_abs
## [1] 1e-08
##
## $optCtrl$ftol_abs
## [1] 1e-08
##
## $optCtrl$optimizer
## [1] "Nelder_Mead"
##
##
## attr(,"class")
## [1] "lmerControl" "merControl"
```

```r
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 -7051.4
## 2   8 -5404.1  2 3294.6  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We have very strong evidence against the hypothesis that the simpler model (1a) fits the data just as well - choose 1b

```
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)

# 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 -5404.1
## 2   9 -5330.8  1 146.47  < 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 -5330.8
## 2   9 -5330.8  0     0          1
```

i) We have very strong evidence against the hypothesis that the simpler model (1b) fits the data just as well - choose 2a
ii) The two models are the same as interaction between duty shift and shift team is the same as full team given it is a concatenation of the two - choose 2b

2. Determine which model from the above is the most appropriate out of those shown. Make appropriate alterations to `model_3` so that it will be the same as your chosen model with the addition of the term shown below, and uses the appropriate likelihood method to allow you to compare the models.

```r
model_3 <- lmer(productivity ~ week + starfleet_gpa + perseverance_score +
                  (1 + week|name) + (1|full_team) +
                 (1 + week|full_team),
                data = crew_data)

lmtest::lrtest(model_2b, model_3)
```

We don't have strong evidence against the hypothesis that the simpler model (2b) fits the data just as well - choose 2b over 3

3. Run `summary()` and `confint()` on whichever model you think is the most appropriate

```r
summary(model_2b)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: productivity ~ week + starfleet_gpa + perseverance_score + (1 +
##     week | name) + (1 | full_team)
##    Data: crew_data
##
## REML criterion at convergence: 10661.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.3950 -0.6046  0.0086  0.6145  2.8065
##
## Random effects:
##  Groups    Name        Variance Std.Dev. Corr
##  name      (Intercept)  8.7634  2.960
##            week         0.2621  0.512    -0.04
##  full_team (Intercept) 13.9340  3.733
##  Residual               0.9820  0.991
## Number of obs: 3012, groups:  name, 251; full_team, 24
##
## Fixed effects:
##                    Estimate Std. Error t value
## (Intercept)         9.00241    2.32016   3.880
## week               -0.01990    0.03274  -0.608
## starfleet_gpa       2.01963    0.20666   9.773
## perseverance_score  0.82868    0.18943   4.375
##
## Correlation of Fixed Effects:
##             (Intr) week   strfl_
## week        -0.005
## starflet_gp -0.738  0.000
## prsvrnc_scr -0.663  0.000  0.119
```

```r
confint(model_2b)
```

```
## Computing profile confidence intervals ...
```

```
##                           2.5 %      97.5 %
```

```
## .sig01              2.68403425  3.25239264
## .sig02             -0.16687114  0.09675417
## .sig03              0.46815273  0.56032181
## .sig04              2.72022756  5.09878998
## .sigma              0.96417285  1.01902429
## (Intercept)         4.46521873 13.54875066
## week               -0.08417683  0.04438371
## starfleet_gpa       1.61455762  2.42455015
## perseverance_score  0.45649203  1.19968055
```

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 run `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.

You DO NOT have to use the exact same code I do to get the associated questions in the timed component correct, as long as it fulfills these instructions, in the correct order. As a hint, here is the structure of my code to complete these tasks.

```
complete_comp <- competition_data %>%
  ___________________ %>%
  ___________________ %>%
  ___________________ %>%
  distinct(crewmember, native_to) %>% # this line will achieve instruction 5
  ___________________ %>% # these last two lines achieve instruction 6,
  ___________________ # but could be done in only one line also
```

```r
incomplete_comp <- competition_data %>%
  pivot_longer(-c(crewmember),
               names_to = "plant", values_to = "num_recorded") %>%
  right_join(origin_data, by = "plant")

complete_comp <- competition_data %>%
  pivot_longer(-c(crewmember),
               names_to = "plant", values_to = "num_recorded") %>%
  right_join(origin_data, by = "plant") %>%
  filter(!(is.na(num_recorded))) %>%
  distinct(crewmember, native_to) %>% # this line will achieve instruction 5
  group_by(crewmember) %>% # these last two lines achieve instruction 6
  summarise(num_unique = n())
```

Task set 6: make your Statistician's life easier

Suppose you were trying to run the following code. It throws an error. (Note: DON'T fix the error, that isn't the point of this activity.) Create a reprex (a reproducible example, see week 1) with everything required for your statistician to reproduce this error. The only 'error' in the output should be the one produced by *this* code. (Hint: there is a library you should include, and you'll also need to provide the data. Once you've copied the complete code for the reprex to your clipboard, you can then run `reprex()` and the content for your reprex will then be added to you clipboard, (i.e., with Ctrl+V or Cmd+V you can paste it.))

```
library(tidyverse)
  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))

origin_data %>%
  filter(nativeto == "Delta Quadrant")
```

```
library(reprex)
reprex()
```

```
## x Install the styler package in order to use `style = TRUE`.

## i Non-interactive session, setting `html_preview = FALSE`.

## ! No input provided and clipboard is not available.

## i Rendering reprex...
```

END