My calender management

Anthony Davidson

2020-07-22

Contents

Overview

I find a challenging aspect of academic research is assoicated with organising both my work and personal time management. Now that I am attempting to work with the skills obtained during my PhD I have so many different projects running at/in different levels of development and collaboration. This repository is a draft approach to my time management using `tidyPipes` workflow.

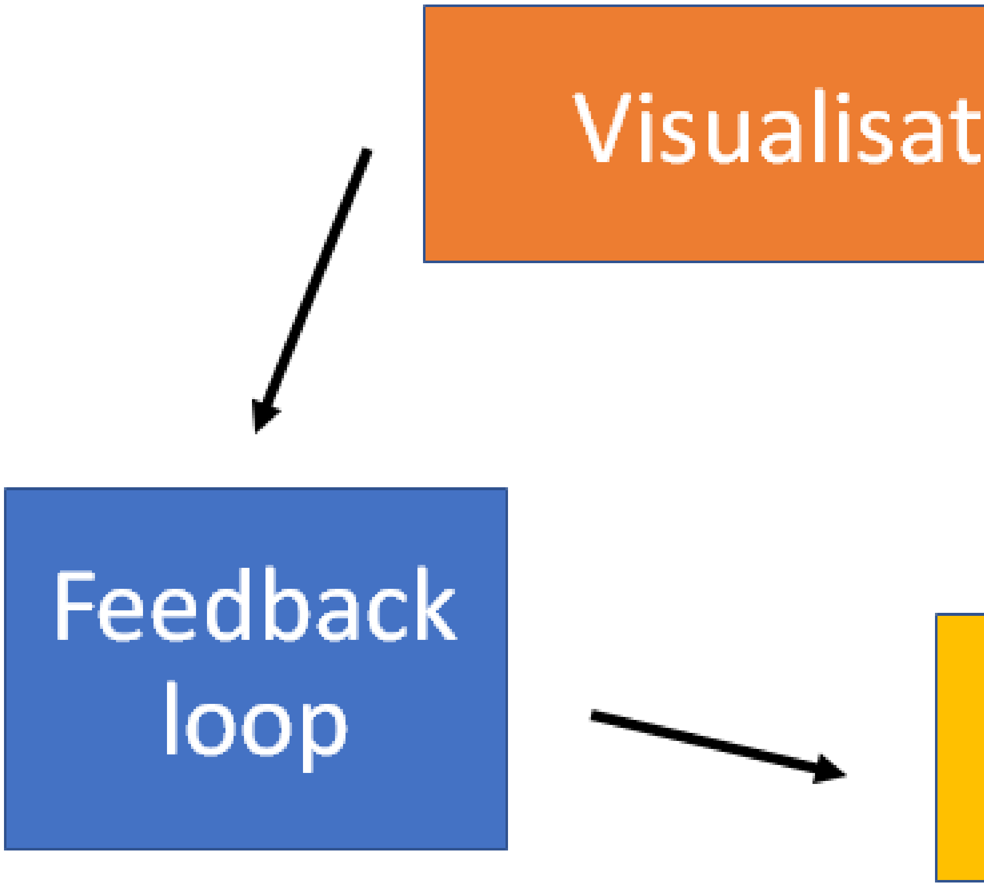
> NOTE: This is my first attempt to apply my `tidyPipes` workflow (Figure above) to my project management and integrate this into my normal workflow.

Generally, this work is focused on writing the scripts to automate the intergration between emails, PhD timeline and other projects. To do this we need to import datasets and modify the structure of these imputs to match the information needed to contruct a timeline of tasks and objectives.

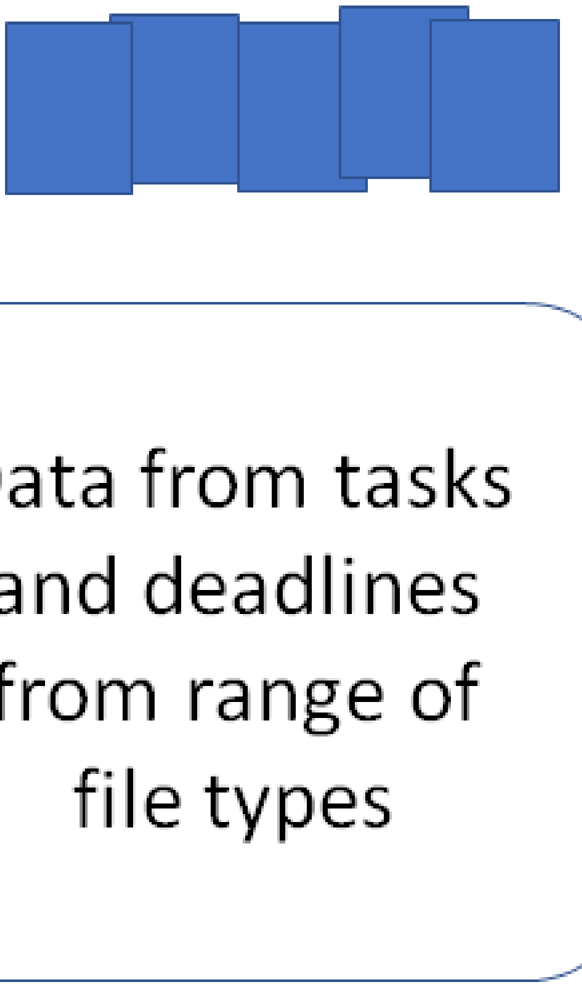
Chapter 1

Introduction

Navigating the path between graduate studies and an academic career is a difficult task at the best of times. One of the key steps in becoming a establish researcher in the current academic environment.



To do this efficiently, time management is one key aspect, however when there are so many little projects running it can be hard to know what to work on.



This document is to record the method to proposing my 6month timeline for my PhD completion, as well as, showing the key aspects of the `tidyPipes` approach to research and the draft project plan for the invertebrate work I am proposing to do with Ben Kefford's lab.

Chapter 2

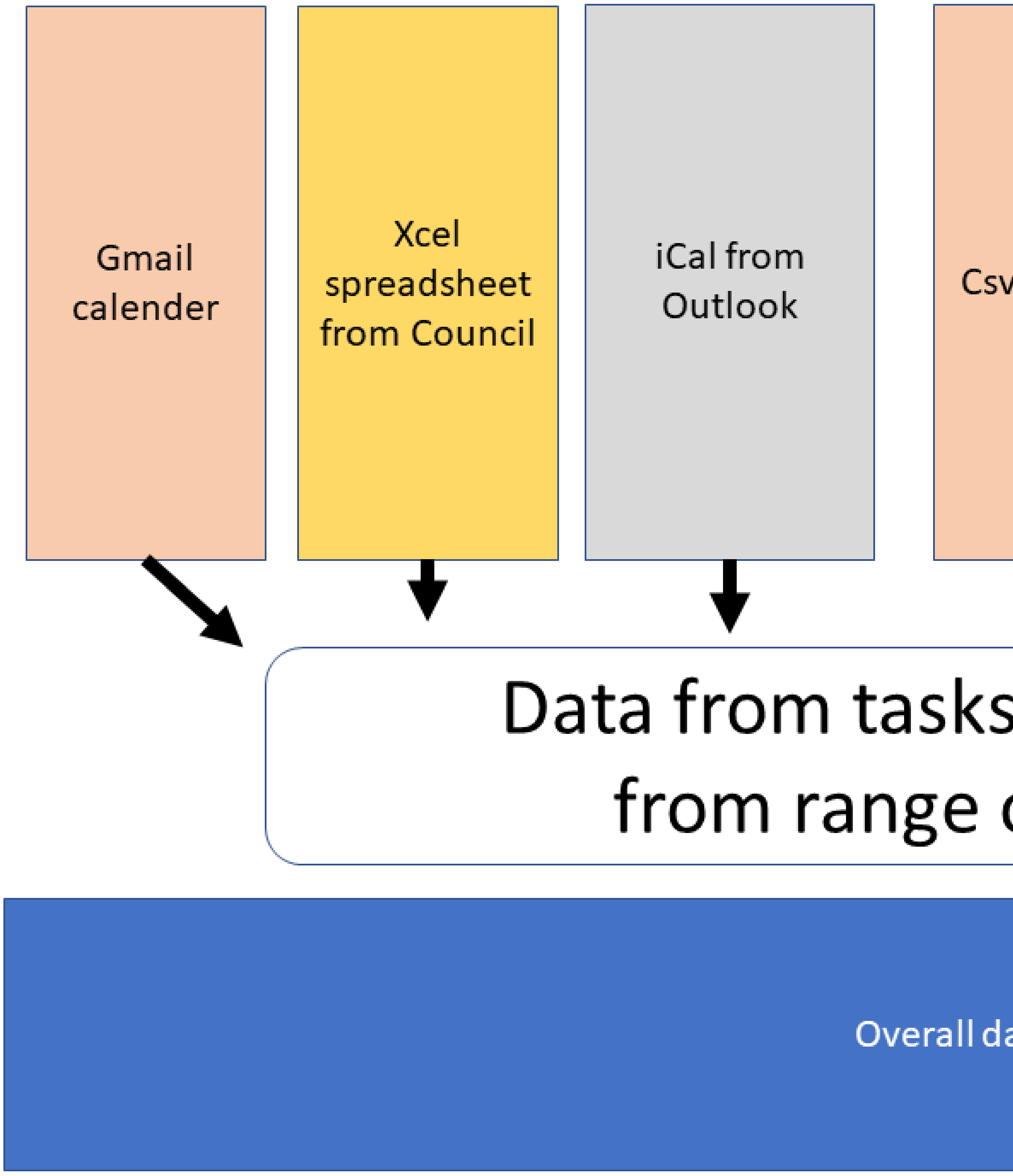
Methods

Generally the concept is to create a baseline dataset of information and then extend this using `dataspice` to create a tidy format of data that can then be modelled and visualised using the `tidyverse` suite of tools.

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.0.2
```

```
knitr::include_graphics(path = "./img/TidyPipes-calenderJUL2020v2.png")
```


Gmail calender
Xcel spreadsheet from Council
iCal from Outlook
Csv
Data from tasks
from range
Overall da

There are multiple difference sources of information for this calendar. To be able to keep this upto date and current I need to write scripts for each data-source to my database of events (here). These are the following importing scripts:

- Timeline figure
  - Past
  - Future

2.1.1 Data/information

The purpose of this vignette is to show how, with reference to a real-world application: creating a timetable for a new module. It assumes you've installed the package following instructions in the README and have attached it as follows:

Overall this is time series data. A good general tutorial for this sort of data is here on youtube. There are several ways to visualise this data, below are two selected bits of code that do this. Overall there are two generalised datasets that may be helpful to other individuals for each project or combination of projects (for the APR for example). The data for this collection of tasks associated with timelines and targets. The baseline dataset is found in the `.xlsx` file named "baseline-dataset-calender.xlsx". This is the base file I have been adding information to when I change the overall structure of the calendar projects.

2.1.2 Manual data (`.csv` files)

To begin with I have collated and restructured the avaliable data from downloaded `.ics` data as a csv and the UCSRC council calendar.

This sorted data was orginally saved as "baseline-dataset-calender.xlsx" but as I couldnt get the xcel package to work nicely I converted each project dataset into a csv file stored in the `./data/` folder.

2.1.3 `.xlsx`

These are excel workbooks. For now this is very simple and works with the current version of excel files (2020). Each "sheet" of the excel file contains a single projects information. This is then converted to a csv file when needed. In the future each project will have its own file that can be added to or modified in a shiny interactive web app.

```
#excel read

#number of sheets in project currently


## Saved as csv's and imported as so below...
```

2.1.4 .csv

Generally the data can be imported as a csv, or other form.

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.0.2
```

```
emailsCalender1 <- read_csv("data/raw_data/anuemails.CSV")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   `Start Time` = col_time(format = ""),
##   `End Time` = col_time(format = ""),
##   `All day event` = col_logical(),
##   `Reminder on/off` = col_logical(),
##   `Reminder Time` = col_time(format = ""),
##   `Billing Information` = col_logical(),
##   Mileage = col_logical(),
##   Private = col_logical(),
##   `Show time as` = col_double()
## )

## See spec(...) for full column specifications.
```

```
#str(emailsCalender1)
```

2.1.5 .iCal data

```
#this currently online
```

2.1.6 Other data

Chapter 3

Visualisation

Creating timeline charts in R (Generating Timeline charts)

Creating timeline charts in R (Generating Timeline charts)

To do this I have created a calendar for each key project/impact/aspect of short-term timeline, objectives, as well as, my career and life projection. To begin with I need to create timelines and other project goals under covid19. I have put this into a single dataset called `dat` here.

These figures can be generated using `ggplot` and other `tidyverse` approaches due to the implantation of the `dataspice` packages above. We will use *ggplot* function from *ggplot2* package to generate timeline charts. The following plots can be created using layers to detail charts.

3.1 Timeline

Timeline charts can be used in a lot of applications like tracking equipment or a process status changes, resource availability & scheduling, project timelines, documenting start and end times of events. The beauty of ggplot2 package is that the code can be easily customized, and more details can be added to the plots.

3.2 Activity

Workout timeline with a heat-map of calories burnt with activity type.

3.3 Calendars

Calender information….

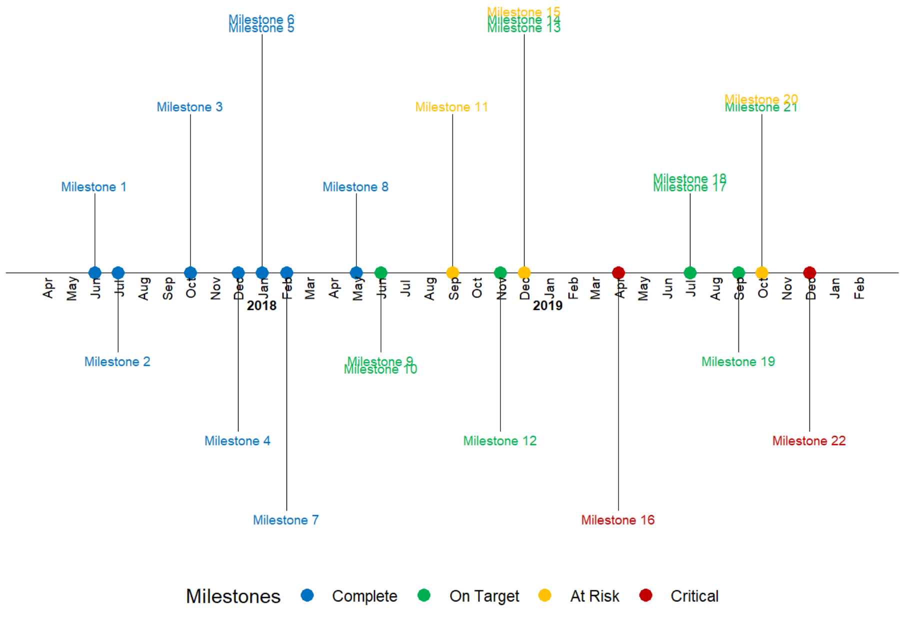


Figure 3.1: image-20200722105325021

3.4 Interactivity

Shiny....

Chapter 4

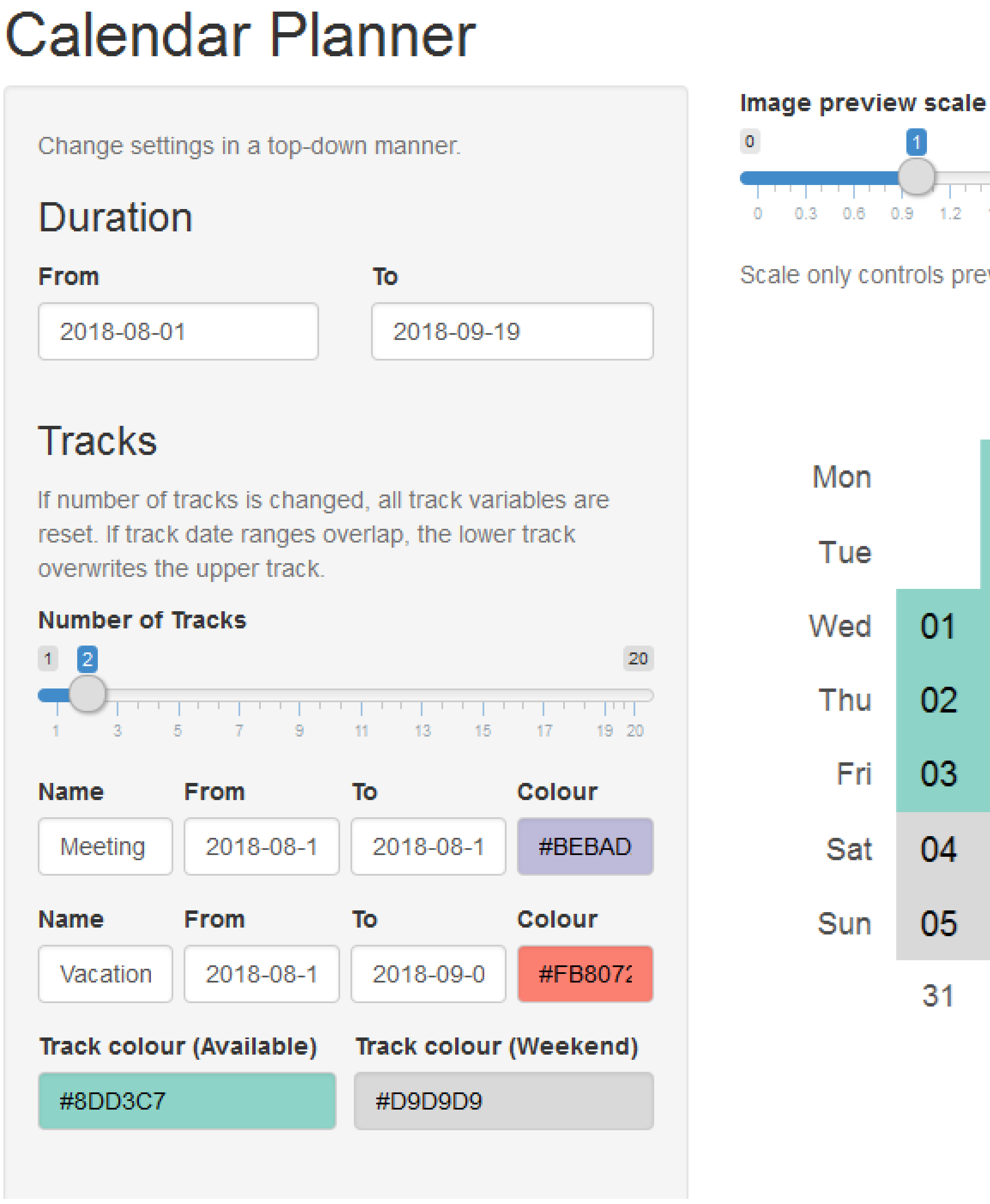
Feedback loop

To create the feedback loop (to get information back from supervisors) I have began to develop a interactive shiny app within the same structure as the baseline dataset so that there is limited coding needed to create the tidypipes "cycle" of community engagement.

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.0.2
```

```
knitr::include_graphics(path = "./img/preview.png")
```



Calendar Planner

Change settings in a top-down manner.

Duration

From: 2018-08-01 To: 2018-09-19

Tracks

If number of tracks is changed, all track variables are reset. If track date ranges overlap, the lower track overwrites the upper track.

Number of Tracks: 2

Name	From	To	Colour
Meeting	2018-08-1	2018-08-1	#BEBAD
Vacation	2018-08-1	2018-09-0	#FB807

Track colour (Available): #8DD3C7

Track colour (Weekend): #D9D9D9

Image preview scale: 1

Scale only controls pre

As computational work takes over our regular management of time over the tradional hard copy “diary”. I like this because important information can not be left in the “local cafe” however as I have used “gmail”, “outlook” and there suites of applications and tools for calenders I have muddled everything up and missed appointments etc.

There is alot of my development work in this section because I have attempted to combine these two packages in a way to document all the council emails and other work that I have undertaken as part of the `COVID19` pandemic in Australia.

To try and counter this I have developed a `tidypipes` workflow for my tasks, projects and other collarorations. See presentation here.

Chapter 5

Overall outputs

5.1 Timeline

One of the tricky bits about timelines is the format of the date and time of each event. This is different for each event so therefore we have four columns to incorporate this information in the following from for this repository and database. This will also mean that some of the information in other formats will have to be converted into these dimension and any other elements defined in each dataset I am combining to make a overall timeline.

By integrating these stage with some `dataspice` code/approach's allows for me to generate metadata from each sheet of the excel file quickly using R. Here are the steps to do this.

```r
#dataspice from github
library(dataspice)

#each project needs to be imported and then saved as csv in raw_data file to document variable na
# raw_data <-
project2 <- readxl::read_excel("./data/raw_data/Sem two planning.xlsx", sheet = 4)
project2
```

```
## # A tibble: 22 x 11
##    eventCode shortName month startDate finishDate startTime
##    <chr>     <chr>     <chr> <chr>     <chr>      <dttm>
##  1 scr20200~ WorldRef~ june  20/06/20~ 20/06/2020 1899-12-31 09:00:00
##  2 scr20200~ Oweek     july  44039     44043      1899-12-31 09:00:00
##  3 scr20200~ ReasonSt~ augu~ 44044     <NA>       1899-12-31 09:00:00
##  4 scr20200~ SSAFsurv~ augu~ 44044     <NA>       1899-12-31 09:00:00
##  5 scr20200~ TownHall2 augu~ 44044     <NA>       1899-12-31 09:00:00
##  6 scr20200~ openDay   augu~ 44044     <NA>       1899-12-31 09:00:00
##  7 scr20200~ nic21st   augu~ 44044     <NA>       NA
##  8 scr20200~ SSAFbids  sept~ 44075     <NA>       1899-12-31 09:00:00
##  9 scr20200~ Grad      sept~ 44075     <NA>       1899-12-31 09:00:00
## 10 scr20200~ SSAFFunds sept~ 44075     <NA>       NA
## # ... with 12 more rows, and 5 more variables: endTime <dttm>,
## #   description <chr>, src <lgl>, ucX <lgl>, week <lgl>
```

```
#date
data_actions <- readxl::read_excel("./data/raw_data/Sem two planning.xlsx", sheet = 1)
```

```
## New names:
## * `` -> ...7
## * `` -> ...8
## * `` -> ...9
## * `` -> ...10
## * `` -> ...11
## * ...
```

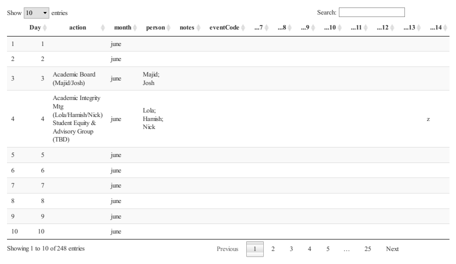
```
data_names <- readxl::read_excel("./data/raw_data/Sem two planning.xlsx", sheet = 3)
#time

#location
```

5.1.0.1 Past Outcomes

5.1.1 Action timeline

These sources of data are combined for my general timeline below. These summarised actions are also the flagged tasks from outlook calendar to catch up with this information but this can be automated in the future.

```
# DT::datatable(emailsCalender1)
## handmade data
DT::datatable(data_actions)
```



Show 10 entries Search:

	Day	action	month	person	notes	eventCode	...7	...8	...9	...10	...11	...12	...13	...14
1	1		june											
2	2		june											
3	3	Academic Board (Majid/Josh)	june	Majid; Josh										
4	4	Academic Integrity Mtg (Lola/Hamish/Nick) Student Equity & Advisory Group (TBD)	june	Lola; Hamish; Nick										z
5	5		june											
6	6		june											
7	7		june											
8	8		june											
9	9		june											
10	10		june											

Showing 1 to 10 of 248 entries Previous 1 2 3 4 5 … 25 Next

```r
# there should be an actions...
data_actions
```

```
## # A tibble: 248 x 14
##      Day action month person notes eventCode ...7  ...8  ...9  ...10 ...11 ...12
##    <dbl> <chr>  <chr> <chr>  <lgl> <lgl>     <lgl> <lgl> <lgl> <lgl> <lgl> <lgl>
##  1     1 <NA>   june  <NA>   NA    NA        NA    NA    NA    NA    NA    NA
##  2     2 <NA>   june  <NA>   NA    NA        NA    NA    NA    NA    NA    NA
##  3     3 Acade~ june  Majid~ NA    NA        NA    NA    NA    NA    NA    NA
##  4     4 Acade~ june  Lola;~ NA    NA        NA    NA    NA    NA    NA    NA
##  5     5 <NA>   june  <NA>   NA    NA        NA    NA    NA    NA    NA    NA
##  6     6 <NA>   june  <NA>   NA    NA        NA    NA    NA    NA    NA    NA
##  7     7 <NA>   june  <NA>   NA    NA        NA    NA    NA    NA    NA    NA
##  8     8 <NA>   june  <NA>   NA    NA        NA    NA    NA    NA    NA    NA
##  9     9 <NA>   june  <NA>   NA    NA        NA    NA    NA    NA    NA    NA
## 10    10 <NA>   june  <NA>   NA    NA        NA    NA    NA    NA    NA    NA
## # ... with 238 more rows, and 2 more variables: ...13 <lgl>, ...14 <chr>
```

5.1.2 Future targets

5.1.2.1 Plot current data

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.2
```

```
library(scales)
```

```
## Warning: package 'scales' was built under R version 4.0.2
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.0.2

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
source("./R/importDATAscript.R")
```

```
## Warning: package 'readxl' was built under R version 4.0.2

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Parsed with column specification:
## cols(
##   eventCode = col_character(),
##   shortName = col_character(),
##   month = col_character(),
##   startDate = col_character(),
##   finishDate = col_character(),
##   startTime = col_time(format = ""),
##   endTime = col_time(format = ""),
##   description = col_character(),
##   project = col_character(),
##   ucX = col_logical(),
##   week = col_logical()
## )
## Parsed with column specification:
## cols(
##   eventCode = col_character(),
##   shortName = col_character(),
##   month = col_character(),
##   startDate = col_character(),
##   finishDate = col_character(),
##   startTime = col_time(format = ""),
##   endTime = col_time(format = ""),
##   description = col_character(),
##   project = col_character(),
##   ucX = col_logical(),
##   week = col_logical()
## )
## Parsed with column specification:
## cols(
##   eventCode = col_character(),
##   shortName = col_character(),
##   month = col_character(),
##   startDate = col_character(),
##   finishDate = col_character(),
##   startTime = col_time(format = ""),
##   endTime = col_time(format = ""),
##   description = col_character(),
##   project = col_character(),
##   ucX = col_logical(),
##   week = col_logical()
## )
```

```
# Now let's load in some data
#milestones is the demo data
#actions is my data

df <- readr::read_csv('./data/milestones.csv')
```

```
## Parsed with column specification:
## cols(
##   month = col_double(),
##   year = col_double(),
##   milestone = col_character(),
##   status = col_character()
## )
```

```
df
```

```
## # A tibble: 22 x 4
##    month  year milestone    status
##    <dbl> <dbl> <chr>        <chr>
##  1     6  2017 Milestone 1  Complete
##  2     7  2017 Milestone 2  Complete
##  3    10  2017 Milestone 3  Complete
##  4    12  2017 Milestone 4  Complete
##  5     1  2018 Milestone 5  Complete
##  6     1  2018 Milestone 6  Complete
##  7     2  2018 Milestone 7  Complete
##  8     5  2018 Milestone 8  Complete
##  9     6  2018 Milestone 9  On Target
## 10     6  2018 Milestone 10 On Target
## # ... with 12 more rows
```

5.1.2.2 Subset dataframe

to correct ggplot subset

```
df$date <- with(df, ymd(sprintf('%04d%02d%02d', year, month, 1)))
df <- df[with(df, order(date)), ]
head(df)
```

```
## # A tibble: 6 x 5
##   month  year milestone   status   date
##   <dbl> <dbl> <chr>       <chr>    <date>
## 1     6  2017 Milestone 1 Complete 2017-06-01
## 2     7  2017 Milestone 2 Complete 2017-07-01
## 3    10  2017 Milestone 3 Complete 2017-10-01
## 4    12  2017 Milestone 4 Complete 2017-12-01
## 5     1  2018 Milestone 5 Complete 2018-01-01
## 6     1  2018 Milestone 6 Complete 2018-01-01
```

```
text_offset <- 0.05

#factoring
status_levels <- c("Complete", "On Target", "At Risk", "Critical")

status_colors <- c("#0070C0", "#00B050", "#FFC000", "#C00000")

df$status <- factor(df$status, levels=status_levels, ordered=TRUE)

#direction
positions <- c(0.5, -0.5, 1.0, -1.0, 1.5, -1.5)
directions <- c(1, -1)

line_pos <- data.frame(
    "date"=unique(df$date),
    "position"=rep(positions, length.out=length(unique(df$date))),
    "direction"=rep(directions, length.out=length(unique(df$date)))
)

df <- merge(x=df, y=line_pos, by="date", all = TRUE)
df <- df[with(df, order(date, status)), ]
```

```
df$month_count <- ave(df$date==df$date, df$date, FUN=cumsum)
df$text_position <- (df$month_count * text_offset * df$direction) + df$position
head(df)
```

```
##         date month year   milestone   status position direction month_count
## 1 2017-06-01     6 2017 Milestone 1 Complete      0.5         1           1
## 2 2017-07-01     7 2017 Milestone 2 Complete     -0.5        -1           1
## 3 2017-10-01    10 2017 Milestone 3 Complete      1.0         1           1
## 4 2017-12-01    12 2017 Milestone 4 Complete     -1.0        -1           1
## 5 2018-01-01     1 2018 Milestone 5 Complete      1.5         1           1
## 6 2018-01-01     1 2018 Milestone 6 Complete      1.5         1           2
##   text_position
## 1          0.55
## 2         -0.55
## 3          1.05
## 4         -1.05
## 5          1.55
## 6          1.60
```

5.1.2.3 Counts

```
text_offset <- 0.05

df$month_count <- ave(df$date==df$date, df$date, FUN=cumsum)
df$text_position <- (df$month_count * text_offset * df$direction) + df$position
head(df)
```

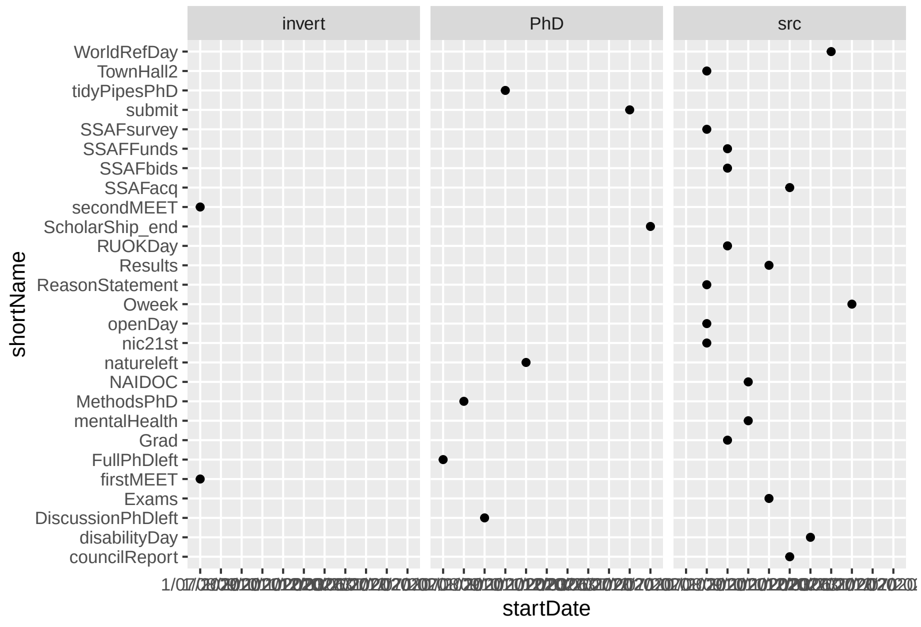
```
##         date month year   milestone   status position direction month_count
## 1 2017-06-01     6 2017 Milestone 1 Complete      0.5         1           1
## 2 2017-07-01     7 2017 Milestone 2 Complete     -0.5        -1           1
## 3 2017-10-01    10 2017 Milestone 3 Complete      1.0         1           1
## 4 2017-12-01    12 2017 Milestone 4 Complete     -1.0        -1           1
## 5 2018-01-01     1 2018 Milestone 5 Complete      1.5         1           1
## 6 2018-01-01     1 2018 Milestone 6 Complete      1.5         1           2
##   text_position
## 1          0.55
## 2         -0.55
## 3          1.05
## 4         -1.05
## 5          1.55
## 6          1.60
```

5.1.2.4 Buffering times

```
month_buffer <- 2

month_date_range <- seq(min(df$date) - months(month_buffer), max(df$date) + months(mon
month_format <- format(month_date_range, '%b')
month_df <- data.frame(month_date_range, month_format)
```

```
year_date_range <- seq(min(df$date) - months(month_buffer), max(df$date) + months(mont
year_date_range <- as.Date(
    intersect(
        ceiling_date(year_date_range, unit="year"),
        floor_date(year_date_range, unit="year")
    ),  origin = "1970-01-01"
)
year_format <- format(year_date_range, '%Y')
year_df <- data.frame(year_date_range, year_format)
```

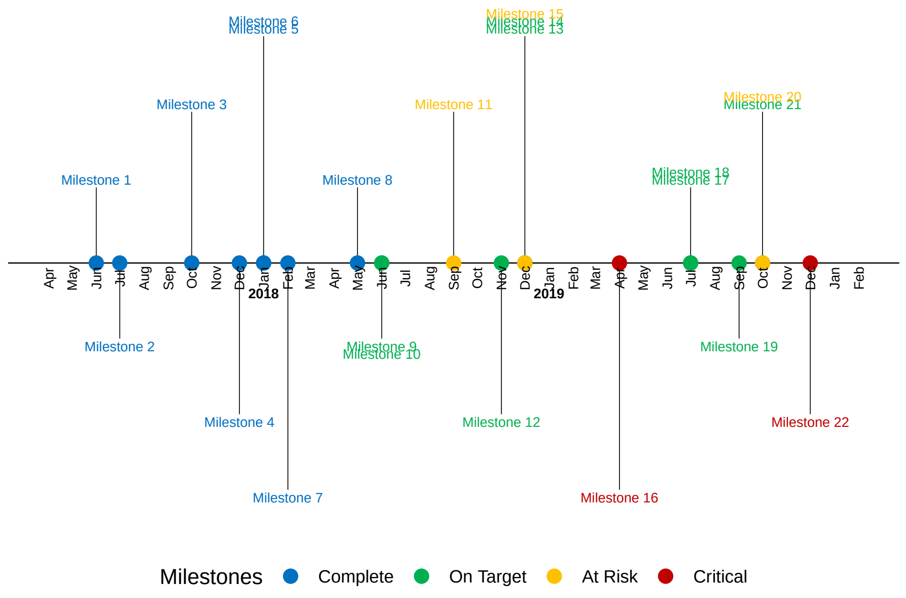
```
# names(datBASE)
# datBASE$project

ggplot(datBASE, aes(x = startDate, y = shortName)) +
  geom_point() +
  facet_wrap(~project)
```



5.1.3.2 Final plot option 1

See Ben Alex Keen's blog with the following output:

```
timeline_plot<-ggplot(df,aes(x=date,y=0, col=status, label=milestone))
timeline_plot<-timeline_plot+labs(col="Milestones")
timeline_plot<-timeline_plot+scale_color_manual(values=status_colors, labels=status_levels, drop
timeline_plot<-timeline_plot+theme_classic()

# Plot horizontal black line for timeline
timeline_plot<-timeline_plot+geom_hline(yintercept=0,
                color = "black", size=0.3)

# Plot vertical segment lines for milestones
timeline_plot<-timeline_plot+geom_segment(data=df[df$month_count == 1,], aes(y=position,yend=0,xe

# Plot scatter points at zero and date
timeline_plot<-timeline_plot+geom_point(aes(y=0), size=3)

# Don't show axes, appropriately position legend
timeline_plot<-timeline_plot+theme(axis.line.y=element_blank(),
                 axis.text.y=element_blank(),
                 axis.title.x=element_blank(),
                 axis.title.y=element_blank(),
                 axis.ticks.y=element_blank(),
                 axis.text.x =element_blank(),
                 axis.ticks.x =element_blank(),
                 axis.line.x =element_blank(),
                 legend.position = "bottom"
                )

# Show text for each month
timeline_plot<-timeline_plot+geom_text(data=month_df, aes(x=month_date_range,y=-0.1,la
# Show year text
timeline_plot<-timeline_plot+geom_text(data=year_df, aes(x=year_date_range,y=-0.2,label
# Show text for each milestone
timeline_plot<-timeline_plot+geom_text(aes(y=text_position,label=milestone),size=2.5)
print(timeline_plot)
```



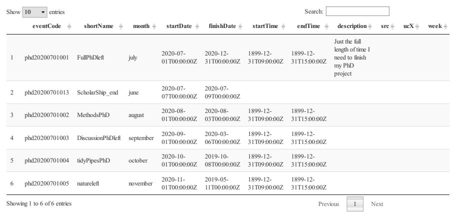
5.1.4 Individual projects

Each of my tasks come from a collection of overall projects I collaborate on and develop code with on timeframes that range from monthly to undefined. The current projects I have integrated into my timeline are:

5.1.5 PhD

Over the duration of my PhD I have currently developed my thesis and publications to align with a 6 month hand-in date from the 1st July 2020.

```r
project1 <- readxl::read_excel("./data/raw_data/Sem two planning.xlsx", sheet = 3)

DT::datatable(head(project1))
```



Show 10 entries   Search:

	eventCode	shortName	month	startDate	finishDate	startTime	endTime	description	src	ucX	week
1	phd20200701001	FullPhDleft	july	2020-07-01T00:00:00Z	2020-12-31T00:00:00Z	1899-12-31T09:00:00Z	1899-12-31T15:00:00Z	Just the full length of time I need to finish my PhD project			
2	phd20200701013	ScholarShip_end	june	2020-07-07T00:00:00Z	2020-07-09T00:00:00Z						
3	phd20200701002	MethodsPhD	august	2020-08-01T00:00:00Z	2020-08-03T00:00:00Z	1899-12-31T09:00:00Z	1899-12-31T15:00:00Z				
4	phd20200701003	DiscussionPhDleft	september	2020-09-01T00:00:00Z	2020-03-06T00:00:00Z	1899-12-31T09:00:00Z	1899-12-31T15:00:00Z				
5	phd20200701004	tidyPipesPhD	october	2020-10-01T00:00:00Z	2019-10-08T00:00:00Z	1899-12-31T09:00:00Z	1899-12-31T15:00:00Z				
6	phd20200701005	natureleft	november	2020-11-01T00:00:00Z	2019-05-11T00:00:00Z	1899-12-31T09:00:00Z	1899-12-31T15:00:00Z				

Showing 1 to 6 of 6 entries   Previous 1 Next

5.1.5.1 Introduction

5.1.5.2 Methods

5.1.5.3 Conclusion

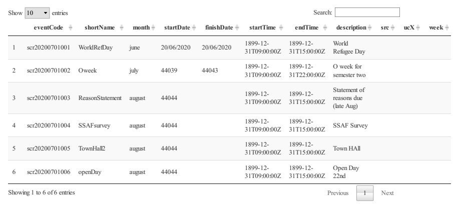
5.1.5.4 Discussion

5.1.6 Previous achievements and tasks

5.1.7 Council tasks

Being nominated to represent the Graduate community on the University of Canberra Council in November 2019 was a great honour. At the time I did understand the impact of

```r
project2 <- readxl::read_excel("./data/raw_data/Sem two planning.xlsx", sheet = 4)

DT::datatable(head(project2))
```


Show 10 entries Search:

	eventCode	shortName	month	startDate	finishDate	startTime	endTime	description	src	ucX	week
1	scr20200701001	WorldRefDay	june	20/06/2020	20/06/2020	1899-12-31T09:00:00Z	1899-12-31T15:00:00Z	World Refugee Day			
2	scr20200701002	Oweek	july	44039	44043	1899-12-31T09:00:00Z	1899-12-31T22:00:00Z	O week for semester two			
3	scr20200701003	ReasonStatement	august	44044		1899-12-31T09:00:00Z	1899-12-31T15:00:00Z	Statement of reasons due (late Aug)			
4	scr20200701004	SSAFsurvey	august	44044		1899-12-31T09:00:00Z	1899-12-31T15:00:00Z	SSAF Survey			
5	scr20200701005	TownHall2	august	44044		1899-12-31T09:00:00Z	1899-12-31T15:00:00Z	Town HAll			
6	scr20200701006	openDay	august	44044		1899-12-31T09:00:00Z	1899-12-31T15:00:00Z	Open Day 22nd			

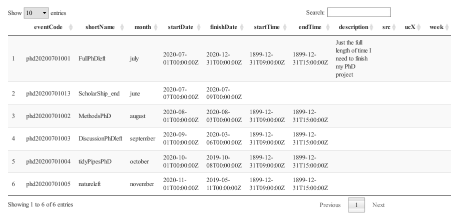
Showing 1 to 6 of 6 entries Previous 1 Next

5.1.7.1 UC-Council

Generally it is regarded that there will be about a week (40hrs) of background reading and investigation before each council meeting. Under covid19 conditions I think this may be much greater.

Here are the general tasks and overall timetable of the Council obligations in 2020:

```r
dataCouncil <- readxl::read_excel("./data/raw_data/Sem two planning.xlsx", sheet = 3)

DT::datatable(head(dataCouncil))
```


Show 10 entries Search:

	eventCode	shortName	month	startDate	finishDate	startTime	endTime	description	src	ucX	week
1	phd20200701001	FullPhDleft	july	2020-07-01T00:00:00Z	2020-12-31T00:00:00Z	1899-12-31T09:00:00Z	1899-12-31T15:00:00Z	Just the full length of time I need to finish my PhD project			
2	phd20200701013	ScholarShip_end	june	2020-07-07T00:00:00Z	2020-07-09T00:00:00Z						
3	phd20200701002	MethodsPhD	august	2020-08-01T00:00:00Z	2020-08-03T00:00:00Z	1899-12-31T09:00:00Z	1899-12-31T15:00:00Z				
4	phd20200701003	DiscussionPhDleft	september	2020-09-01T00:00:00Z	2020-03-06T00:00:00Z	1899-12-31T09:00:00Z	1899-12-31T15:00:00Z				
5	phd20200701004	tidyPipesPhD	october	2020-10-01T00:00:00Z	2019-10-08T00:00:00Z	1899-12-31T09:00:00Z	1899-12-31T15:00:00Z				
6	phd20200701005	natureleft	november	2020-11-01T00:00:00Z	2019-05-11T00:00:00Z	1899-12-31T09:00:00Z	1899-12-31T15:00:00Z				

Showing 1 to 6 of 6 entries Previous 1 Next

5.1.7.2 UC-SRC

This is a short demo site to help with planning for the SRC for semester 2 2020.

5.1.7.3 Supporting Reproducibility at UC

My Phd studies put me in a unique situation where I can apply the tools and computational development I have done with my PhD and conceptually test the framework for the application in the education sector.

- `UCdown`
- `councilCOMOS`
- `UCSRC covid support`

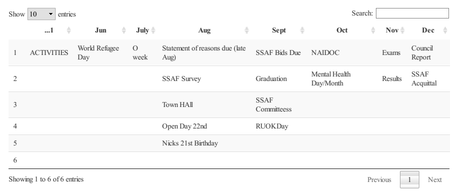
5.1.8 UC-Invertebrates

This work has its own repository so far.

```r
project3 <- readxl::read_excel("./data/raw_data/Sem two planning.xlsx", sheet = 5)
```

```
## New names:
## * `` -> ...1
```

```r
DT::datatable(head(project3))
```



Show 10 entries Search:

	...1	Jun	July	Aug	Sept	Oct	Nov	Dec
1	ACTIVITIES	World Refugee Day	O week	Statement of reasons due (late Aug)	SSAF Bids Due	NAIDOC	Exams	Council Report
2				SSAF Survey	Graduation	Mental Health Day/Month	Results	SSAF Acquittal
3				Town HAll	SSAF Committeess			
4				Open Day 22nd	RUOKDay			
5				Nicks 21st Birthday				
6								

Showing 1 to 6 of 6 entries Previous 1 Next

Chapter 6

Extra projects

During Covid19 I have undertaken other additional learning and development tasks to keep me informed for my positions on Council and for my PhD work.

6.1 Courses

During Covid19 as there are less academic activities being undertaken in reality I have upskilled with other online courses and personal development courses below:

6.1.1 Coursa

Course Name	Enrolment Date	Paid	Y/N	Grade Achieved
Using clinical health data for better healthcare	2019-07-01	N	0.00	
A Life of Happiness and Fulfillment	2020-04-04	N	0.00	
Science Matters: Let's Talk About COVID-19	2020-04-03	N	0.00	
Reproducible Research	2020-04-21	N	0.00	
Understanding Clinical Research: Behind the Statistics	2020-04-04	N	0.00	
The Science of Well-Being	2020-04-21	N	0.00	
Learning How to Learn: Powerful mental tools to help you master tough subjects	2020-04-04	N	0.00	

Course Name	Enrolment Date	Paid	Y/N	Grade Achieved
Mindshift: Break Through Obstacles to Learning and Discover Your	Hidden Potential	2020-04-04	N	0.00
Mountains 101	2020-04-04	N	0.00	
Machine Learning	2020-04-04	N	0.00	

6.1.2 Seminars

During 2020 I will undertake several small seminars with regard to my PhD work. These include progress seminars and other departmental talks.

PAST: - ANU Masters Class - GEM 2020

FUTURE: - GEM - IAE final seminar - ANU lectures?

6.1.3 Teaching

6.1.3.1 ANU: Climate Change

6.1.3.2 ANU: Masters Projects

6.1.4 Community projects

6.1.4.1 Hackett seedlings

6.1.4.2 Ants Surf&Skate

6.1.4.3 Gov Hack...