## eLearning

# Logging

There are 3 levels of logging in this file TRACE (most words), INFO, and ERROR (least words) by default, script is logged at the INFO level All ERROR messages are also reported to Error folder, see log\_and\_quit().

# Load elearning data

Data is loaded from the 'R for MECSH/Load' folder. It arrives in this folder from the eLearnign website

#Load eLearning file  
courseData <- read\_excel("Load/Victoria-eLearning.xlsx", sheet = "Victoria-Current")  
  
#Table creation - these steps are unlikely to be needed in the future, as 3 raw tables will arive, not 1 merged table.  
  
ID\_status <- courseData[c("NurseID","Location",  
 "C1Completed On","C1Status","C2Status","C3Status","C4Status",  
 "C5Status","C6Status","C6Completed On")]  
  
# only because test load file has extra rows  
ID\_status <- ID\_status[1:29,1:10]  
  
ID\_px <- courseData[c("NurseID","Location","LICENSEE GROUP NAME","T3NOTES","FirstName","LastName",  
 "FPM Completed", "Position", "Position2" )]

## Completion statuses are calculated and checked for errors

# Training status by site

traing status has been grouped into 1 of 3 categories, Not Atempted, Started and Completed. This table shows the number of staff in each category for each site. All staff members in eLeanring system are included in this table.

Site\_status <- ID\_status %>% group\_by(Location, STATUS) %>%   
 select(Location, COMPLETED, STATUS) %>% summarise (n=n())  
  
Site\_status <- ungroup(spread(Site\_status, STATUS, n, fill = 0))  
Site\_status$Total\_Staff <- rowSums(Site\_status[c(2:4)])  
Site\_status <- rbind(Site\_status, c("Total", colSums(Site\_status[,2:5])))  
Site\_status

## # A tibble: 8 x 5  
## Location Completed `Not Attempted` Started Total\_Staff  
## <chr> <chr> <chr> <chr> <chr>  
## 1 Ballarat 4 0 0 4  
## 2 Casey 1 0 1 2  
## 3 Dandenong 5 0 2 7  
## 4 Frankston 4 0 1 5  
## 5 Hume 0 1 0 1  
## 6 Melton 0 3 0 3  
## 7 Whittlesea 3 2 2 7  
## 8 Total 17 6 6 29

# Join to Nurse details table

eLearning data joined with Staff data

trim\_px <- ID\_px[,c(1,7,9)]  
  
Staff\_type <- ID\_status %>% left\_join(trim\_px,by = "NurseID")

# Data cleaned and main reference table produced

This table is used for all further summaries/reports

Staff\_status

## # A tibble: 29 x 8  
## NurseID Location Strt\_date pre\_2016 COMPLETED STATUS  
## <dbl> <chr> <date> <chr> <int> <chr>  
## 1 1 Ballarat 2017-08-07 FALSE 1 Completed  
## 2 2 Ballarat 2013-06-25 TRUE 1 Completed  
## 3 3 Ballarat 2013-05-22 TRUE 1 Completed  
## 4 4 Ballarat 2013-05-22 TRUE 1 Completed  
## 5 5 Casey NA Unknown 0 Started  
## 6 6 Casey 2017-10-14 FALSE 1 Completed  
## 7 7 Dandenong 2017-05-09 FALSE 1 Completed  
## 8 8 Dandenong 2017-05-11 FALSE 1 Completed  
## 9 9 Dandenong 2015-07-12 TRUE 0 Started  
## 10 10 Dandenong 2013-07-30 TRUE 1 Completed  
## # ... with 19 more rows, and 2 more variables: `FPM Completed` <chr>,  
## # sees\_clients <chr>

# Main Outcomes - Full table

# Full table  
Full\_table <- Staff\_status %>% group\_by(pre\_2016, Location, sees\_clients) %>%   
 select(pre\_2016, Location, sees\_clients, COMPLETED) %>% summarise (n=n())  
Full\_table = ungroup(spread(Full\_table, sees\_clients, n, fill = 0))  
Full\_table$Total\_Staff <- rowSums(Full\_table[c("Nurse\_SW", "Other")])  
  
# check all rows account for  
if ((count(ID\_status) - colSums(Full\_table[,5])) != 0){  
 flog.error("CHECKING failed", name='logger.c')  
 err\_msg <- "CHECKING failed"  
 log\_and\_quit()  
 } else {  
 flog.info("CHECKING passed", name='logger.c')  
}  
  
#Show result  
Full\_table <- rbind(Full\_table, c("Total", "", colSums(Full\_table[,3:5])))  
Full\_table

## # A tibble: 15 x 5  
## pre\_2016 Location Nurse\_SW Other Total\_Staff  
## <chr> <chr> <chr> <chr> <chr>  
## 1 FALSE Ballarat 1 0 1  
## 2 FALSE Casey 1 0 1  
## 3 FALSE Dandenong 2 0 2  
## 4 FALSE Frankston 2 0 2  
## 5 FALSE Hume 1 0 1  
## 6 FALSE Melton 2 1 3  
## 7 FALSE Whittlesea 1 0 1  
## 8 TRUE Ballarat 3 0 3  
## 9 TRUE Dandenong 4 1 5  
## 10 TRUE Frankston 2 0 2  
## 11 TRUE Whittlesea 3 2 5  
## 12 Unknown Casey 0 1 1  
## 13 Unknown Frankston 1 0 1  
## 14 Unknown Whittlesea 1 0 1  
## 15 Total 24 5 29

Main Outcomes - nurses per category - position and training time

# nurses per category - position and training time  
  
sum\_count <- Staff\_status %>% group\_by(pre\_2016, sees\_clients) %>%   
 select(pre\_2016, sees\_clients, COMPLETED) %>% summarise (n=n())  
sum\_count <- ungroup(spread(sum\_count, sees\_clients, n, fill = 0))  
sum\_count$Total\_Staff <- rowSums(sum\_count[c("Nurse\_SW", "Other")])  
sum\_count <- rbind(sum\_count, c("Total", colSums(sum\_count[,2:4])))  
sum\_count

## # A tibble: 4 x 4  
## pre\_2016 Nurse\_SW Other Total\_Staff  
## <chr> <chr> <chr> <chr>  
## 1 FALSE 10 1 11  
## 2 TRUE 12 3 15  
## 3 Unknown 2 1 3  
## 4 Total 24 5 29

Main Outcomes - Summary table

# Summary table - filtered for position and training time  
  
Filtered\_table <- Staff\_status %>% filter(pre\_2016 == "TRUE" & sees\_clients == "Nurse\_SW") %>% group\_by(Location) %>% select(Location, COMPLETED) %>% summarise (n=n(), perc\_comp = mean(COMPLETED)\*100)  
Filtered\_table <- rbind(Filtered\_table, c("Total", colSums(Filtered\_table[,2]), colMeans(Filtered\_table[,3])))  
Filtered\_table

## # A tibble: 5 x 3  
## Location n perc\_comp  
## <chr> <chr> <chr>  
## 1 Ballarat 3 100  
## 2 Dandenong 4 75  
## 3 Frankston 2 100  
## 4 Whittlesea 3 100  
## 5 Total 12 93.75