# R Notebook

## Author: OA Lab, NWFSC

### Title: Aquarium Temperature Investigation: Temperature SubSelectScript.R

### Date: April-May 2020

# Aquarium Temperature Investigation

R script below will subselect and plot temperature data for MOATs Overall goal is to determine if MOATs (per treatment) are true replicates

```
#*********************************
##Libraries
#*********************************
library(shiny)
```

```
## Warning: package 'shiny' was built under R version 3.6.2
```

```
library(tidyverse)
```

```
## ── Attaching packages ──────────────────────────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.3.0     ✓ purrr   0.3.4
## ✓ tibble  3.0.1     ✓ dplyr   0.8.5
## ✓ tidyr   1.1.0     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.5.0
```

```
## Warning: package 'tibble' was built under R version 3.6.2
```

```
## Warning: package 'tidyr' was built under R version 3.6.2
```

```
## Warning: package 'purrr' was built under R version 3.6.2
```

```
## ── Conflicts ─────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(stringr)
library(readxl)
library(readr)
library(tidyr)
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.6.2
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
## The following objects are masked from 'package:dplyr':
##
##     intersect, setdiff, union
```

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

```
library(violinmplot)
```

```
## Loading required package: lattice
```

```
## Warning: package 'lattice' was built under R version 3.6.2
```

```
library(vioplot)
```

```
## Warning: package 'vioplot' was built under R version 3.6.2
```

```
## Loading required package: sm
```

```
## Package 'sm', version 2.2-5.6: type help(sm) for summary information
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 3.6.2
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(yarrr)
```

```
## Loading required package: jpeg
```

```
## Loading required package: BayesFactor
```

```
## Loading required package: coda
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## ************
## Welcome to BayesFactor 0.9.12-4.2. If you have questions, please contact Richard Morey (richarddmorey@gmail.co
m).
##
## Type BFManual() to open the manual.
## ************
```

```
## Loading required package: circlize
```

```
## Warning: package 'circlize' was built under R version 3.6.2
## ========================================
## circlize version 0.4.9
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: https://jokergoo.github.io/circlize_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
##   in R. Bioinformatics 2014.
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(circlize))
## ========================================
```

```
## yarrr v0.1.5. Citation info at citation('yarrr'). Package guide at yarrr.guide()
```

```
## Email me at Nathaniel.D.Phillips.is@gmail.com
```

```
##
## Attaching package: 'yarrr'
```

```
## The following object is masked from 'package:ggplot2':
##
##     diamonds
```

```
library(datapasta)
library(reprex)
library(miniUI)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

# Outline

(Current as of 2020.06.25)

- 1.) Working Directory
- 2.) Spolling Data into one CSV
- 3.) Creating the Dataframe “dml”
- 4.) Setting a new working directory
- 4.) Creating dateTime objects
- 5.) Creating Treatment Variables
- 6.) Creating Night and Day Periods
- 7.) Insitu data
- 8.) Cleaning up dml “Cdml”
- 9.) Framing filters for Cdml
- 10.) Temperature “Jumps”
- 11.) Calculating averages by treatment & day/night
- 12.) Summary & Group by Cdml
- 13.) Plots Boxplots & timeseries
- 14.) Duration between conditions- Histogram & Cummulative Distribtuion Plot section
- 15.) aTemp bands investig.

#1.) Working Directory

Inside the working directory (folder) are CSV files generated after using the moats graph app Link to moats.Graph.app (http://github.com/pmcelhany/moatsGraphs.git) Moving average window (nObs) set to 4. This results in the observations being averaged into a 24second timeframe. CSVs were created for each moats: M01, M02, M03, M04, M05, M06, M07, M08, M09, M10, M11, M12, M13. Files are also available on the OA Google drive. OA Drive for MOATS data (https://drive.google.com/open?id=13a2hk1a9I9gRlgf2Xyl1dSYDf5vnyYvs)

# 2.) Spooling Data into one CSV

Combining multiple CSV files into 1 document. Original input files from individual LVM (logical volumne management) files off each MOATs. 2.1 Create a list of files

```
## 2.1 Create a list of files
# All files to be joined have ext. "csv" can use that pattern to join
files <- list.files(path= "/Users/katherinerovinski/GIT/NWFSC.MUK_MOATs_SMR2019/LabViewLogs(CSV)Rbind", pattern =
".*\\.csv")
print(files)
```

```
##  [1] "M1_df_MoatsData.csv"  "M10_df_MoatsData.csv" "M11_df_MoatsData.csv"
##  [4] "M12_df_MoatsData.csv" "M13_df_MoatsData.csv" "M2_df_MoatsData.csv"
##  [7] "M3_df_MoatsData.csv"  "M4_df_MoatsData.csv"  "M5_df_MoatsData.csv"
## [10] "M6_df_MoatsData.csv"  "M7_df_MoatsData.csv"  "M8_df_MoatsData.csv"
## [13] "M9_df_MoatsData.csv"
```

2.2 Create a temporary place for files

```
## 2.2 Create a temporary place for files

setwd("/Users/katherinerovinski/GIT/NWFSC.MUK_MOATs_SMR2019/LabViewLogs(CSV)Rbind")

temp <- lapply(files, fread, sep= ",")
print(temp)
```

```
## [[1]]
##        moats            dateTime aTemperature sTemperature         pH       DO
##     1:   M01 2019-09-05 17:04:00     12.16840    11.324116 0.02244800 9.269025
##     2:   M01 2019-09-05 17:04:22     12.17155    11.346000 0.03559150 9.268738
##     3:   M01 2019-09-05 17:04:46     12.17656    11.367861 0.02263175 9.268492
##     4:   M01 2019-09-05 17:05:10     12.17969    11.396164 0.04348125 9.267310
##     5:   M01 2019-09-05 17:05:34     12.18132    11.439413 0.02432750 9.266493
##    ---                                                                        
## 96693:   M01 2019-09-17 16:50:23     11.19017     9.521056 0.03718850 9.259634
## 96694:   M01 2019-09-17 16:50:48     11.18994     9.509823 0.05128550 9.261294
## 96695:   M01 2019-09-19 16:52:00     11.91160    13.162302 7.66178975 8.847867
## 96696:   M01 2019-09-19 16:52:21     11.91988    13.164492 7.66264175 8.847612
## 96697:   M01 2019-09-19 16:52:45     11.92752    13.168515 7.66253800 8.846189
##        salinity
##     1:     29.5
##     2:     29.5
##     3:     29.5
##     4:     29.5
##     5:     29.5
##    ---         
## 96693:     29.5
## 96694:     29.5
## 96695:     28.8
## 96696:     28.8
## 96697:     28.8
## 
## [[2]]
##        moats            dateTime aTemperature sTemperature       pH       DO
##     1:   M10 2019-09-24 16:05:00     12.08942     10.90406 7.727869 8.892254
##     2:   M10 2019-09-24 16:05:23     12.09632     10.88388 7.728531 8.903144
##     3:   M10 2019-09-24 16:05:47     12.10228     10.86738 7.729645 8.913513
##     4:   M10 2019-09-24 16:06:12     12.10285     10.85012 7.730739 8.923338
##     5:   M10 2019-09-24 16:06:35     12.10142     10.84065 7.731811 8.933912
##    ---                                                                      
## 80639:   M10 2019-10-30 10:50:59     11.44499     11.70883 7.681737 6.159690
## 80640:   M10 2019-10-30 10:53:00     11.50010     11.69634 7.674728 6.176614
## 80641:   M10 2019-10-30 10:55:00     11.52432     11.73733 7.673905 6.208007
## 80642:   M10 2019-10-30 10:57:00     11.54326     11.78689 7.679314 6.267933
## 80643:   M10 2019-10-30 10:58:59     11.52632     11.78225 7.683825 6.351308
##        salinity
##     1:     28.8
##     2:     28.8
##     3:     28.8
##     4:     28.8
##     5:     28.8
##    ---         
## 80639:     28.8
## 80640:     28.8
## 80641:     28.8
## 80642:     28.8
## 80643:     28.8
## 
## [[3]]
##         moats            dateTime aTemperature sTemperature       pH         DO
##      1:   M11 2019-10-28 11:27:00     13.24384     13.27378 7.655841  2.3327030
##      2:   M11 2019-10-28 11:27:24     13.24669     13.28229 7.652207  2.4434278
##      3:   M11 2019-10-28 11:27:47     13.23905     13.27764 7.654789  2.4257235
##      4:   M11 2019-10-28 11:28:12     13.23424     13.26904 7.654091  2.5788760
##      5:   M11 2019-10-28 11:28:36     13.22975     13.26124 7.651458  2.6075580
##     ---                                                                        
## 110339:   M11 2019-10-30 11:24:59     13.10857     13.14075 7.666257 -0.8970227
## 110340:   M11 2019-10-30 11:25:24     13.10800     13.12946 7.669591 -0.8970350
## 110341:   M11 2019-10-30 11:25:48     13.10633     13.13023 7.667792 -0.8970733
## 110342:   M11 2019-10-30 11:26:12     13.11069     13.13927 7.662918 -0.8969603
## 110343:   M11 2019-10-30 11:26:36     13.11461     13.15588 7.659757 -0.8968665
##         salinity
##      1:     28.8
##      2:     28.8
##      3:     28.8
##      4:     28.8
##      5:     28.8
##     ---         
## 110339:     28.8
## 110340:     28.8
## 110341:     28.8
## 110342:     28.8
## 110343:     28.8
## 
## [[4]]
##        moats            dateTime aTemperature sTemperature       pH       DO
##     1:   M12 2019-09-24 10:48:00     11.54457     12.05596 7.635548 6.666244
##     2:   M12 2019-09-24 10:48:23     11.55216     12.04850 7.636722 6.681026
##     3:   M12 2019-09-24 10:48:47     11.56142     12.05127 7.638049 6.697410
##     4:   M12 2019-09-24 10:49:11     11.56440     12.06437 7.639172 6.715560
##     5:   M12 2019-09-24 10:49:35     11.57110     12.08176 7.640389 6.734496
##    ---                                                                      
## 54784:   M12 2019-11-01 12:43:59     11.16364     11.52935 7.750611 9.125633
## 54785:   M12 2019-11-01 12:45:59     11.16199     11.54860 7.751082 9.121086
## 54786:   M12 2019-11-01 12:47:59     11.16497     11.57361 7.751337 9.126159
## 54787:   M12 2019-11-01 12:49:59     11.17101     11.58790 7.751266 9.143706
## 54788:   M12 2019-11-01 12:51:59     11.17299     11.59811 7.750636 9.164638
##        salinity
##     1:     28.8
##     2:     28.8
##     3:     28.8
##     4:     28.8
##     5:     28.8
##    ---         
## 54784:     28.8
## 54785:     28.8
## 54786:     28.8
## 54787:     28.8
## 54788:     28.8
## 
## [[5]]
##          moats            dateTime aTemperature sTemperature       pH       DO
##      1:   M13 2019-10-05 09:42:00     12.88292     12.90622 7.525024 4.209248
##      2:   M13 2019-10-05 09:42:24     12.87668     12.91175 7.526525 4.201476
##      3:   M13 2019-10-05 09:42:47     12.87611     12.91363 7.528626 4.196523
##      4:   M13 2019-10-05 09:43:12     12.88322     12.91140 7.529908 4.195774
##      5:   M13 2019-10-05 09:43:36     12.88943     12.90763 7.530890 4.198372
##     ---                                                                      
## 107420:   M13 2019-10-31 08:38:00     13.86670     12.98007 7.719215 8.378414
## 107421:   M13 2019-10-31 08:39:59     13.85943     12.97847 7.718890 8.376869
## 107422:   M13 2019-10-31 08:42:00     13.85518     12.98079 7.719110 8.380475
## 107423:   M13 2019-10-31 08:43:59     13.85470     12.97842 7.720154 8.386980
## 107424:   M13 2019-10-31 08:46:00     13.85757     12.97633 7.721720 8.395183
##         salinity
##      1:     28.8
##      2:     28.8
##      3:     28.8
##      4:     28.8
##      5:     28.8
##     ---         
## 107420:     28.8
## 107421:     28.8
## 107422:     28.8
## 107423:     28.8
## 107424:     28.8
## 
## [[6]]
##        moats            dateTime aTemperature sTemperature       pH       DO
##     1:   M02 2019-09-24 16:56:00     14.00188     11.88287 7.544728 8.688206
##     2:   M02 2019-09-24 16:56:24     13.99150     11.91185 7.544594 8.700327
##     3:   M02 2019-09-24 16:56:48     14.00338     11.93261 7.544907 8.711961
##     4:   M02 2019-09-24 16:57:12     13.99937     11.95824 7.545630 8.722980
##     5:   M02 2019-09-24 16:57:36     13.99001     11.98529 7.546176 8.733106
##    ---                                                                      
## 96770:   M02 2019-10-25 13:23:00     13.67577     14.50265 7.553553 7.499465
## 96771:   M02 2019-10-25 13:25:00     13.67210     14.50724 7.557383 7.427122
## 96772:   M02 2019-10-25 13:27:00     13.67808     14.51468 7.561196 7.381927
## 96773:   M02 2019-10-25 13:29:00     13.68171     14.53864 7.565191 7.383728
## 96774:   M02 2019-10-25 13:30:59     13.69302     14.58706 7.569311 7.437285
##        salinity
##     1:     28.8
##     2:     28.8
##     3:     28.8
##     4:     28.8
##     5:     28.8
##    ---         
## 96770:     28.8
## 96771:     28.8
## 96772:     28.8
## 96773:     28.8
## 96774:     28.8
## 
## [[7]]
##         moats            dateTime aTemperature sTemperature       pH       DO
##      1:   M03 2019-09-24 16:23:00     12.20874     11.12296 7.716566 8.857545
##      2:   M03 2019-09-24 16:23:23     12.21485     11.09998 7.717669 8.858215
##      3:   M03 2019-09-24 16:23:47     12.21552     11.07615 7.718555 8.859583
##      4:   M03 2019-09-24 16:24:11     12.21425     11.05154 7.719685 8.861280
##      5:   M03 2019-09-24 16:24:35     12.21551     11.02739 7.720459 8.862019
##     ---                                                                      
## 103962:   M03 2019-10-27 12:35:59     11.18896     11.20713 7.723452 9.457098
## 103963:   M03 2019-10-27 12:37:59     11.18676     11.22743 7.723114 9.461530
## 103964:   M03 2019-10-27 12:39:59     11.19001     11.23277 7.722659 9.466332
## 103965:   M03 2019-10-27 12:42:00     11.19109     11.25243 7.721918 9.470595
## 103966:   M03 2019-10-27 12:43:59     11.19839     11.26209 7.723172 9.473182
##         salinity
##      1:     28.8
##      2:     28.8
##      3:     28.8
##      4:     28.8
##      5:     28.8
##     ---
## 103962:     28.8
## 103963:     28.8
## 103964:     28.8
## 103965:     28.8
## 103966:     28.8
##
## [[8]]
##         moats            dateTime aTemperature sTemperature          pH
##      1:   M04 2019-10-05 14:03:00     11.91434     12.44716    8.825669
##      2:   M04 2019-10-05 14:03:23     11.91526     12.44246    8.826660
##      3:   M04 2019-10-05 14:03:47     11.91462     12.44424    8.830653
##      4:   M04 2019-10-05 14:04:12     11.91659     12.44531    8.830290
##      5:   M04 2019-10-05 14:04:35     11.91764     12.44556    8.835112
##     ---
## 100737:   M04 2019-10-18 15:07:00   -273.04429   -273.22433 8123.273242
## 100738:   M04 2019-10-18 15:08:59   -273.04429   -273.22433 8123.273242
## 100739:   M04 2019-10-18 15:11:00   -273.04429   -273.22433 8123.273242
## 100740:   M04 2019-10-18 15:13:00   -273.04429   -273.22433 8123.273242
## 100741:   M04 2019-10-18 15:15:00   -273.04429   -273.22433 8123.273242
##                DO salinity
##      1:  9.024436     28.8
##      2:  9.024051     28.8
##      3:  9.023525     28.8
##      4:  9.023639     28.8
##      5:  9.023378     28.8
##     ---
## 100737: -2.070290     28.8
## 100738: -2.070290     28.8
## 100739: -2.070290     28.8
## 100740: -2.070290     28.8
## 100741: -2.070290     28.8
##
## [[9]]
##        moats            dateTime aTemperature sTemperature       pH       DO
##     1:   M05 2019-09-24 10:04:00     13.45163     13.38484 7.604477 4.430357
##     2:   M05 2019-09-24 10:04:23     13.45253     13.38669 7.604743 4.428692
##     3:   M05 2019-09-24 10:04:47     13.46313     13.38174 7.604795 4.424452
##     4:   M05 2019-09-24 10:05:12     13.46482     13.38264 7.605025 4.418166
##     5:   M05 2019-09-24 10:05:35     13.47658     13.38145 7.604209 4.411646
##    ---
## 69881:   M05 2019-10-19 09:54:00    -12.13223     13.69912 8.720219 7.927978
## 69882:   M05 2019-10-19 09:56:00    -12.13115     13.67656 8.721747 7.929497
## 69883:   M05 2019-10-19 09:58:00    -12.12418     13.65593 8.724045 7.927378
## 69884:   M05 2019-10-19 10:00:00    -12.19022     13.65425 8.726931 7.933592
## 69885:   M05 2019-10-19 10:01:59    -12.19581     13.63223 8.726680 7.937080
##        salinity
##     1:     28.8
##     2:     28.8
##     3:     28.8
##     4:     28.8
##     5:     28.8
##    ---
## 69881:     28.8
## 69882:     28.8
## 69883:     28.8
## 69884:     28.8
## 69885:     28.8
##
## [[10]]
##        moats            dateTime aTemperature sTemperature       pH       DO
##     1:   M06 2019-09-24 16:42:00     13.97340     13.59691 7.733539 8.457947
##     2:   M06 2019-09-24 16:42:23     13.97440     13.59848 7.732976 8.467310
##     3:   M06 2019-09-24 16:42:47     13.97289     13.59922 7.732138 8.476004
##     4:   M06 2019-09-24 16:43:11     13.97367     13.59896 7.731699 8.485997
##     5:   M06 2019-09-24 16:43:35     13.97104     13.59855 7.731255 8.495649
##    ---
## 82681:   M06 2019-10-23 16:32:59     14.09326     14.18709 7.713069 8.819638
## 82682:   M06 2019-10-23 16:35:00     14.06091     14.20211 7.712702 8.793442
## 82683:   M06 2019-10-23 16:37:00     14.06299     14.27481 7.711946 8.767664
## 82684:   M06 2019-10-23 16:39:00     14.07089     14.31402 7.711172 8.739593
## 82685:   M06 2019-10-23 16:40:59     14.07484     14.32679 7.710819 8.709578
##        salinity
##     1:     28.8
##     2:     28.8
##     3:     28.8
##     4:     28.8
##     5:     28.8
##    ---         
## 82681:     28.8
## 82682:     28.8
## 82683:     28.8
## 82684:     28.8
## 82685:     28.8
## 
## [[11]]
##        moats            dateTime aTemperature sTemperature       pH       DO
##     1:   M07 2019-09-24 16:07:00     12.13279     10.80555 7.714717 8.967500
##     2:   M07 2019-09-24 16:07:24     12.14024     10.77122 7.714582 8.973155
##     3:   M07 2019-09-24 16:07:48     12.12668     10.74974 7.713819 8.978636
##     4:   M07 2019-09-24 16:08:12     12.12292     10.73651 7.714620 8.983874
##     5:   M07 2019-09-24 16:08:36     12.12922     10.71033 7.714409 8.989965
##    ---                                                                      
## 51418:   M07 2019-10-23 11:18:22     11.92310     12.44842 7.700380 7.728110
## 51419:   M07 2019-10-23 11:20:22     11.93719     12.52979 7.698907 7.824580
## 51420:   M07 2019-10-23 11:22:22     11.94029     12.61646 7.694582 7.900057
## 51421:   M07 2019-10-23 11:24:22     11.95217     12.68966 7.697589 7.958912
## 51422:   M07 2019-10-23 11:26:22     12.02391     12.75254 7.696608 8.004696
##        salinity
##     1:     28.8
##     2:     28.8
##     3:     28.8
##     4:     28.8
##     5:     28.8
##    ---         
## 51418:     28.8
## 51419:     28.8
## 51420:     28.8
## 51421:     28.8
## 51422:     28.8
## 
## [[12]]
##        moats            dateTime aTemperature sTemperature       pH       DO
##     1:   M08 2019-09-24 16:19:00     13.97487     13.65684 7.532832 8.589048
##     2:   M08 2019-09-24 16:19:24     13.97642     13.65925 7.534100 8.589045
##     3:   M08 2019-09-24 16:19:47     13.98313     13.65838 7.534952 8.589651
##     4:   M08 2019-09-24 16:20:12     13.98327     13.66222 7.536382 8.590224
##     5:   M08 2019-09-24 16:20:36     13.97546     13.67105 7.538302 8.590668
##    ---                                                                      
## 82819:   M08 2019-10-20 16:10:00     13.78250     13.90584 7.566084 8.552211
## 82820:   M08 2019-10-20 16:12:00     13.78034     13.97506 7.572273 8.547564
## 82821:   M08 2019-10-20 16:14:00     13.78263     14.07372 7.577503 8.541789
## 82822:   M08 2019-10-20 16:16:00     13.77039     14.19781 7.579950 8.534088
## 82823:   M08 2019-10-20 16:17:59     13.77569     14.30966 7.577583 8.526458
##        salinity
##     1:     28.8
##     2:     28.8
##     3:     28.8
##     4:     28.8
##     5:     28.8
##    ---         
## 82819:     28.8
## 82820:     28.8
## 82821:     28.8
## 82822:     28.8
## 82823:     28.8
## 
## [[13]]
##        moats            dateTime aTemperature sTemperature       pH       DO
##     1:   M09 2019-09-26 14:51:00     11.08368     10.68647 1.011406 8.890546
##     2:   M09 2019-09-26 14:51:24     11.07862     10.67480 1.011406 8.889532
##     3:   M09 2019-09-26 14:51:47     11.07612     10.66337 1.011406 8.890120
##     4:   M09 2019-09-26 14:52:12     11.07909     10.63342 1.011406 8.891280
##     5:   M09 2019-09-26 14:52:35     11.08047     10.60913 1.011406 8.893174
##    ---                                                                      
## 99685:   M09 2019-09-28 14:48:59     10.99281     10.91662 1.011406 8.747725
## 99686:   M09 2019-09-28 14:49:24     10.97035     10.88477 1.011406 8.743497
## 99687:   M09 2019-09-28 14:49:47     10.95868     10.85422 1.011406 8.740896
## 99688:   M09 2019-09-28 14:50:12     10.95436     10.81981 1.011406 8.739749
## 99689:   M09 2019-09-28 14:50:36     10.98239     10.78134 1.011406 8.741034
##        salinity
##     1:     29.5
##     2:     29.5
##     3:     29.5
##     4:     29.5
##     5:     29.5
##    ---         
## 99685:     29.5
## 99686:     29.5
## 99687:     29.5
## 99688:     29.5
## 99689:     29.5
```

2.3 Create a new vector for Moats data logs

```
## 2.3 Create a new vector for Moats data logs
# "M01thruM13Moatslog_data" via rbind
M01thruM13moatslog_data <- rbindlist(temp)
print(M01thruM13moatslog_data)
```

```
##          moats            dateTime aTemperature sTemperature         pH
##       1:   M01 2019-09-05 17:04:00     12.16840     11.32412 0.02244800
##       2:   M01 2019-09-05 17:04:22     12.17155     11.34600 0.03559150
##       3:   M01 2019-09-05 17:04:46     12.17656     11.36786 0.02263175
##       4:   M01 2019-09-05 17:05:10     12.17969     11.39616 0.04348125
##       5:   M01 2019-09-05 17:05:34     12.18132     11.43941 0.02432750
##      ---
## 1137876:   M09 2019-09-28 14:48:59     10.99281     10.91662 1.01140600
## 1137877:   M09 2019-09-28 14:49:24     10.97035     10.88477 1.01140600
## 1137878:   M09 2019-09-28 14:49:47     10.95868     10.85422 1.01140600
## 1137879:   M09 2019-09-28 14:50:12     10.95436     10.81981 1.01140600
## 1137880:   M09 2019-09-28 14:50:36     10.98239     10.78134 1.01140600
##                DO salinity
##       1: 9.269025     29.5
##       2: 9.268738     29.5
##       3: 9.268492     29.5
##       4: 9.267310     29.5
##       5: 9.266493     29.5
##      ---
## 1137876: 8.747725     29.5
## 1137877: 8.743497     29.5
## 1137878: 8.740896     29.5
## 1137879: 8.739749     29.5
## 1137880: 8.741034     29.5
```

2.4 Write the new csv document

```
## 2.5 Write the new csv document |
# "M01thruM13moatslog" can uncomment this command line as needed
#write.csv(M01thruM13moatslog_data, file = "M01thruM13moatslog.csv", row.names = FALSE)
```

Saved on the OA Google Drive M01thruM13moatslog.csv (https://drive.google.com/open?id=15iBXct9b4EjKDq75vKnm5NobowBwK3G-)

# 3.) Creating the Dataframe “dml”

3.1 Reading the CSV

```
## 3.1 Reading the CSV |
## ensuring column names and types
## Data Moats Log = dml
dml <- read.csv( file = "M01thruM13moatslog.csv", stringsAsFactors = FALSE)
dim(dml)
```

```
## [1] 1137880       7
```

3.1a Duplication Check

```
## 3.1a Duplication Check
## duplicates observed in "dml" on 2020.05.07
dup2 <- dml[duplicated(dml),]
#if no dups, Cdml2 has same number of rows as Cdml
dml2 <- dml %>% distinct()
```

3.1b Sub sampling dataframe “dml”

```
## 3.1b Sub sampling dataframe "dml"
## creating a sub sample of the data moats log dml dataframe to allow for quick graphs


#subsample every 17th row (because prime numbers are indeed cool)
dml <- dml %>% arrange(moats, dateTime) %>% filter(row_number() %% 17 == 0)

#write.csv(dml, file = "M01thruM13moatslog_n17.csv", row.names = FALSE)
```

3.2 Checking variables

```
## 3.2 Checking variables
## Looking to ensure the different variables are treated as the correct variable type
## Checking the names in the dataframe
names(dml)
```

```
## [1] "moats"        "dateTime"     "aTemperature" "sTemperature" "pH"
## [6] "DO"           "salinity"
```

```
## Checking variable type/class
class(dml$moats)
```

```
## [1] "character"
```

```
dml$moats <- as.factor(dml$moats)
```

3.3 Changing variables

```
## 3.3 Changing variables |
## Changing MOATs to Factors for the 13 different MOATs- these will be the discrete units for follow analysis
dml$moats <- factor(dml$moats)
# Checking the names of the different levels
levels(dml$moats)
```

```
##  [1] "M01" "M02" "M03" "M04" "M05" "M06" "M07" "M08" "M09" "M10" "M11" "M12"
## [13] "M13"
```

```
##checking the dataset, dimensions
dim(dml)
```

```
## [1] 66934     7
```

#4.) Creating dateTime objects

4.0 Establish the dateTime objects

```
# 4.0 establish the date time object of the CSV |
dml$dateTime <- as.POSIXct(dml$dateTime, format="%Y-%m-%d %H:%M:%OS")
ReferenceTime <- as.POSIXct("2019-09-20 23:59:00")
class(ReferenceTime)
```

```
## [1] "POSIXct" "POSIXt"
```

```
# QA check
dim(dml)
```

```
## [1] 66934     7
```

# 5.) Creating Treatment Variables

5.1 Identifying Treatments by MOATS

```
## 5.1 Identifying treatments by moats
## establishing treatments
dml$treatment <- ""
dml$treatment[dml$moats == "M07" | dml$moats== "M10" | dml$moats== "M12"] <- "current"
dml$treatment[dml$moats == "M01"| dml$moats== "M06"] <- "hightemperature"
dml$treatment[dml$moats == "M02"| dml$moats== "M08" | dml$moats== "M13"] <- "allchange"
dml$treatment[dml$moats == "M03"| dml$moats == "M04" | dml$moats == "M05" | dml$moats== "M11"] <- "broken_and_ambientbroken"
#verify that this new column has been created
names(dml)
```

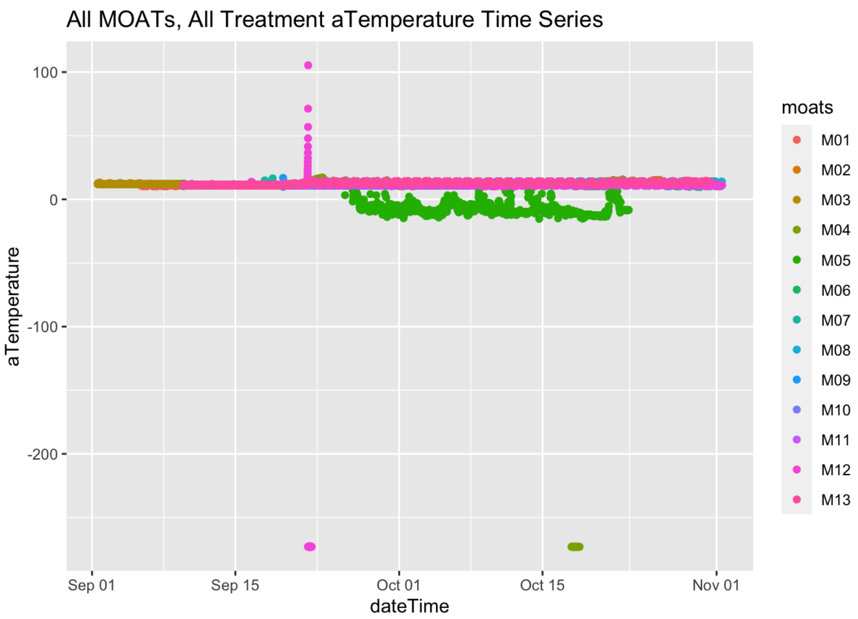
```
## [1] "moats"        "dateTime"     "aTemperature" "sTemperature" "pH"
## [6] "DO"           "salinity"     "treatment"
```

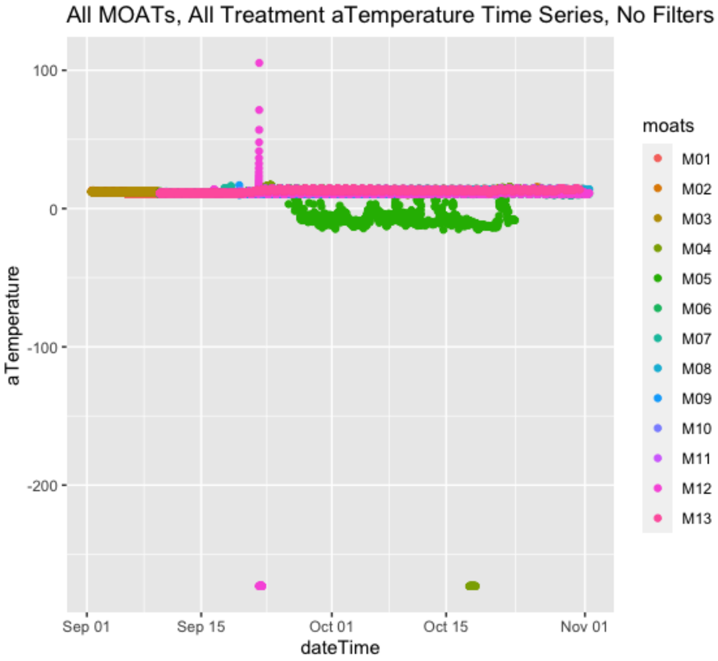
```
#results should include:
#[1] "moats"        "dateTime"     "aTemperature" "sTemperature" "pH"
#[6] "DO"           "salinity"     "treatment"

# QA check
dim(dml)
```

```
## [1] 66934     8
```

5.2 Simple ggplot to show the different MOATs without polish

```
plot5.2 <- ggplot(dml, aes(x=dateTime, y=aTemperature)) +
  geom_point(aes(colour=moats, point=)) +
  ggtitle("All MOATs, All Treatment aTemperature Time Series")

plot5.2
```





Plot 5.2 Figure1

# 6.) Creating Night and Day Periods

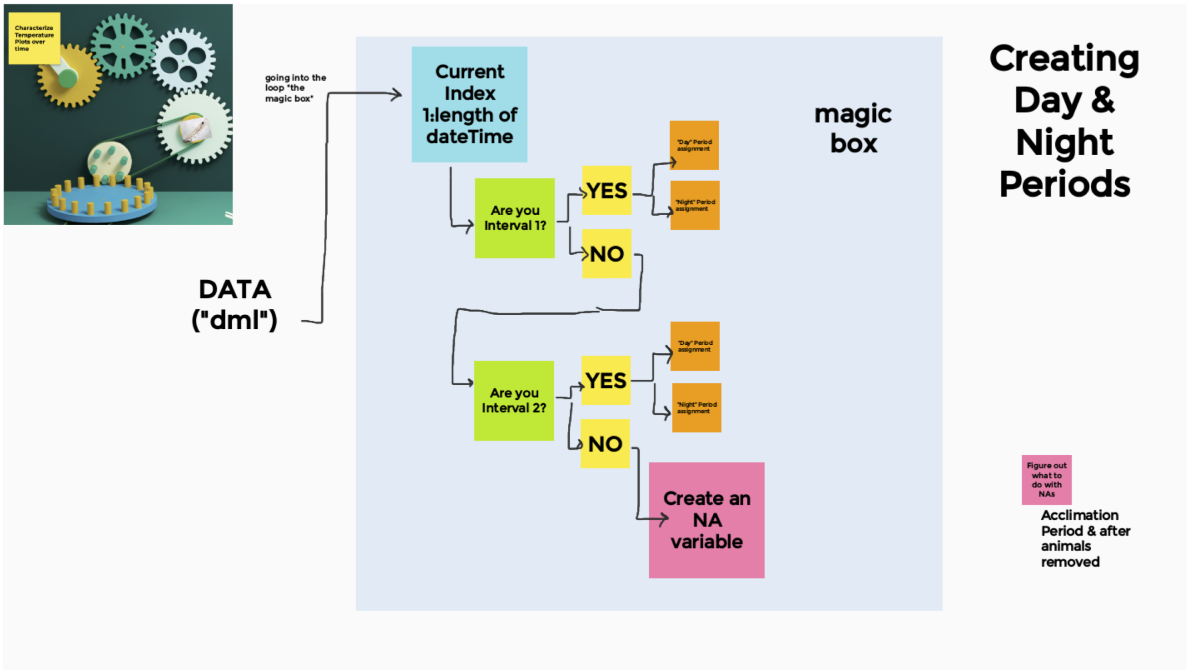
6.1 Narrative (Overall) Creating a day and night variables Day and night periods exclude the acclimation period by their definition. Day and night changed at about ~ 1230 on 05OCT19 Treatment start date considered to begin Monday 23SEP19 at 1200pm

Krill Night Started 1200 (~1230*) and ends 2100 Krill Days Started 2101 and ends 1159 (~1229*)

### Interval 1 start 1200 23SEP19, end 1229 05OCT19

### Interval 2 start 1230 05OCT19, end 2100 30OCT19

# Concept Diagram graphic


Characterize Temperature Plots over time
going into the loop "the magic box"
DATA ("dml")
Current Index 1:length of dateTime
Are you Interval 1?
YES
"Day" Period assignment
"Night" Period assignment
NO
Are you Interval 2?
YES
"Day" Period assignment
"Night" Period assignment
NO
Create an NA variable
magic box
Creating Day & Night Periods
Figure out what to do with NAs
Acclimation Period & after animals removed

Day/Night Definitions Concept Diagram Figure2

6.2 New Column, New Variable

```
## 6.2 New Column, New Variable in dml
#creating a new column, new variable "period"
dml$period <- ""
```

6.3 Disassembling dateTime to create 2 new variables

```
## 6.3 Disassembling dateTime to create 2 new variables
# Create new split date and time columns
dml$ObservationDate <- as.Date(dml$dateTime)
dml$ObservationTime <- format(as.POSIXct(dml$dateTime) ,format = "%H:%M:%S")
```

## 6.4 Narrative about Intervals

## Interval 1

- Interval Date Start “2019-09-23”
- Interval Date End “2019-10-05”
- Day Start Time “21:01:00”
- Day End Time “12:01:00”
- Night Start Time “12:00:00”
- Night End Time “21:00:00”
- Other Time

## Interval 2

- Interval Date Start “2019-10-05”
- Interval Date End “2019-10-30”
- Day Start Time “21:01:00”
- Day End Time “12:29:00”
- Night Start Time “12:30:00”
- Night End Time “21:00:00”
- Other Time

6.5 Day / Night Assignments

```
## 6.5 Day / Night Assignments
# Using the "case_when" function in the tidyverse in the place of a loop

dml <- dml %>% mutate(period=case_when(
                          (ObservationDate >= "2019-09-23")
                          & (ObservationDate <="2019-10-05")
                          & (ObservationTime >= "12:00:00")
                          & (ObservationTime <="21:00:00") ~"night",

                          (ObservationDate >= "2019-10-05")
                          & (ObservationDate <= "2019-10-30")
                          & (ObservationTime >= "12:30:00")
                          & (ObservationTime <="21:00:00") ~"night",

                          (ObservationDate >= "2019-09-23")
                          & (ObservationDate <="2019-10-05")
                          & ((ObservationTime >= "21:01:00")
                          | (ObservationTime <="11:59:00")) ~"day",

                          (ObservationDate >= "2019-10-05")
                          & (ObservationDate <= "2019-10-30")
                          & ((ObservationTime >= "21:01:00")
                          | (ObservationTime <= "12:29:00")) ~"day",
                          TRUE ~"other"
                        )

                    )
```

# 7.) Insitu data

7.0 In-situ data files Two files represent manual checks of aquarium salinity. Extracted from this are point measurements of aquarim temperature

```
# ## 7.1 Read in files
# # carefule about navigating to the second working directory
#
#
# knitr::opts_chunk$set(echo = TRUE)
# knitr::opts_knit$set(root.dir = "/Users/katherinerovinski/GIT/NWFSC.MUK_KRL_SMR2019/06. MOATS replication verif
ication/01. Raw Data/InSituSampling")
#
#
# d.insitu <- read.csv(file = "KRL19_insitu_sample_Day.csv", stringsAsFactors = FALSE)
# n.insitu <- read.csv(file = "KRL19_insitu_sample_Night.csv", stringsAsFactors = FALSE)
```

# 8.) Cleaning up dml

Creating a new dataframe cleaned with the various filters below checking on the variables inside dml Cleaned-Up dml = Cdml

```
## 8.1 Noting names of the different variables in dml
names(dml)
```

```
##  [1] "moats"           "dateTime"        "aTemperature"    "sTemperature"
##  [5] "pH"              "DO"              "salinity"        "treatment"
##  [9] "period"          "ObservationDate" "ObservationTime"
```

# 8.2 Narrative on desired changes to create Cdml

8.2 Narrative on Cdml Temperatures below 5C and above 30C are thought to be less than probable

```
# Changes to be made the dataframe by variable
#[1] "moats" "M03", "M04", "M05", "M09", "M11" to be filtered out- all these
#                                           MOATs were dropped from the study
#[2] "dateTime" - no changes
#[3] "aTemperature" - no changes
#[4] "sTemperature" - no changes
#[5] "pH" - no changes
#[6] "DO"- no changes
#[7] "salinity" - no changes
#[8] "treatment" - dropping the listed MOATs will eliminate the
#                           "broken_and_ambientbroken" treatment
#[9] "period" - filtering out "other"
#[10] "ObservationDate" - no changes
#[11] "ObservationTime" - no changes, note that each observation could be
#                                                 spaced 6.8minutes apart

Cdml <- dml %>% filter(!moats %in% c("M03", "M04", "M05", "M11")) %>%
  filter(aTemperature>= 5 & aTemperature<=30) %>%
  filter(treatment %in% c("current", "allchange", "hightemperature")) %>%
  filter(period != "other")
```

### 8.3 Option to write a CSV of Cdml parameters

droping the levels of the moats

```
levels(Cdml$moats)
```

```
##  [1] "M01" "M02" "M03" "M04" "M05" "M06" "M07" "M08" "M09" "M10" "M11" "M12"
## [13] "M13"
```

```
Cdml$moats <- droplevels(Cdml$moats)
```

# 9.) Framing filters for Cdml

9.0 cleaning names of levels, factors

```
## 9.0 Removing the names of moats & treatments removed from Cdml
# Removal should allow for cleaner graphs
# Determined moats lab never got under 5C and was never over 30C

filteredFrame = filter(Cdml,
  !moats %in% c('M03', "M04", "M05", "M11") &
  (aTemperature>= 5 & aTemperature<=30) &
  treatment %in% c("current", "allchange", "hightemperature") &
  period != "other")
```

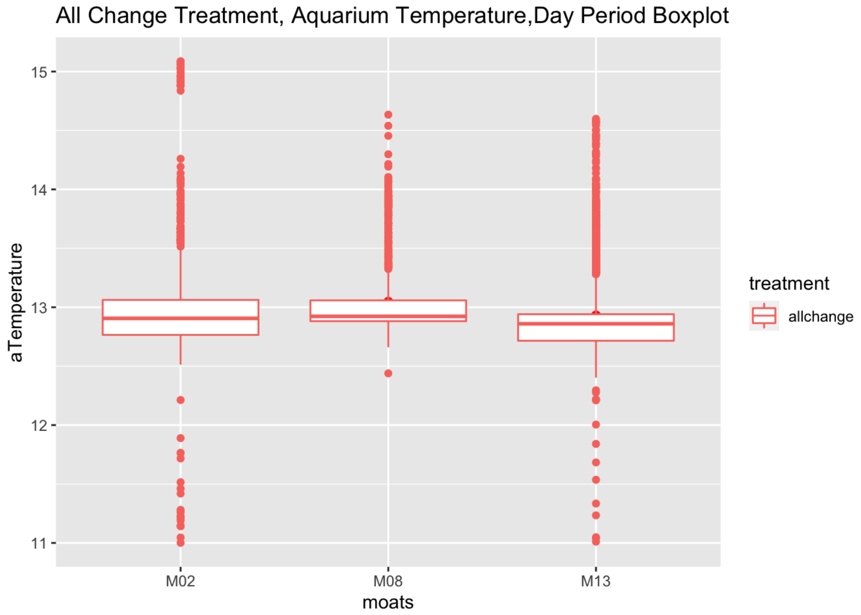
Without dropping levels and factors not named label may still crowd plots #### 9.1 Dropping names and factors

```
## 9.1 Dropping levels and factors
filteredFrame$moats <- droplevels(filteredFrame$moats)
filteredFrame$treatment <- factor(filteredFrame$treatment)
```

9.2 Broad-gague test of plots by treatment

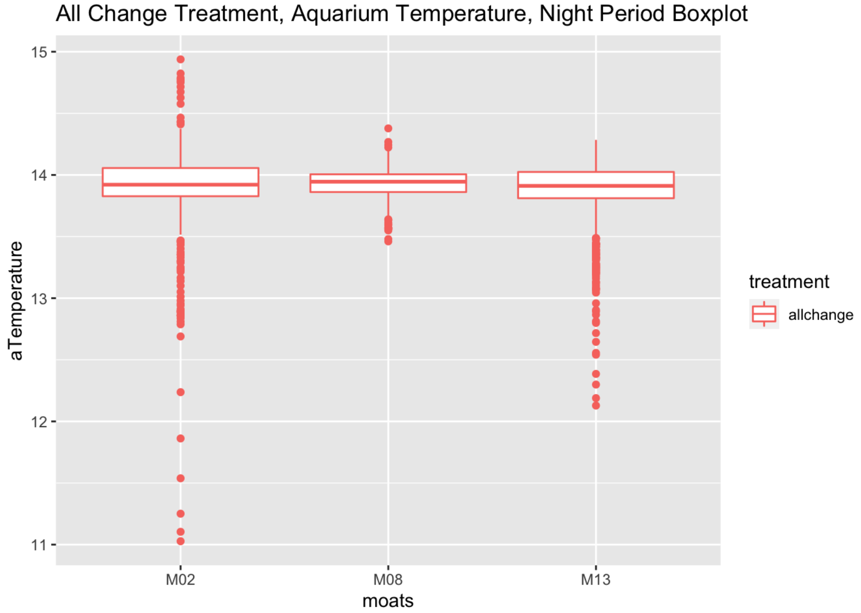
### 9.2.1 All Change Day Period Boxplot

```
simpleplot_allchg_day <- ggplot(subset(Cdml[Cdml$treatment == "allchange", ],
                                          period %in% ("day")),
                                          aes(x=moats, y=aTemperature,
                                                        colour=treatment)) +
                                          stat_summary(fun=mean, geom="point",
                                                        size=2, color="red")  +
                                          geom_boxplot() +
                                          ggtitle("All Change Treatment, Aquarium Temperature,Day Period Boxplot"
)


simpleplot_allchg_day
```


All Change Treatment, Aquarium Temperature,Day Period Boxplot

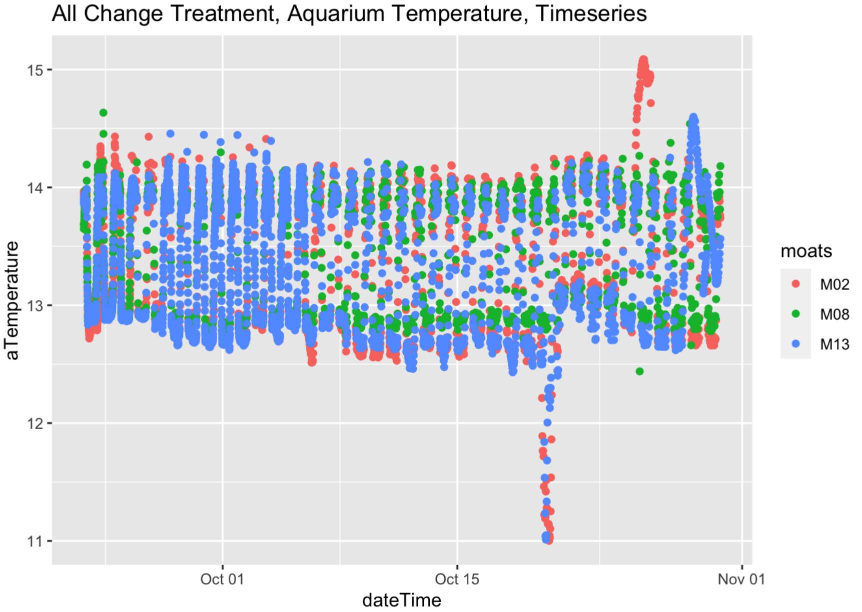
### 9.2.2 All Change Night Period Boxplot

```
simpleplot_allchg_night <- ggplot(subset(Cdml[Cdml$treatment == "allchange", ],
                                         period %in% ("night")),
                                         aes(x=moats, y=aTemperature,
                                                       colour=treatment)) +
                                         stat_summary(fun=mean, geom="point",
                                                       size=2, color="red")  +
                                         geom_boxplot() +
                                         ggtitle("All Change Treatment, Aquarium Temperature, Night Period Boxpl
ot")

simpleplot_allchg_night
```


All Change Treatment, Aquarium Temperature, Night Period Boxplot

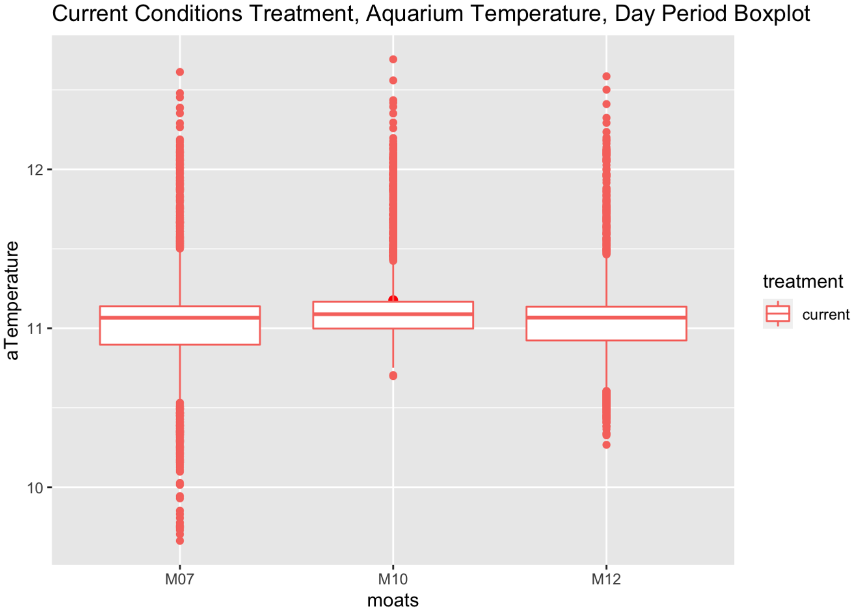
### 9.2.3 All Change Timeseries

```
simpleplot_allchg_timeseries <- ggplot(subset(Cdml[Cdml$treatment == "allchange", ])+
                                 aes(x=dateTime, y=aTemperature) +
                                 geom_point(aes(colour=moats, point=)) +
                                 ggtitle("All Change Treatment, Aquarium Temperature, Timeseries")

simpleplot_allchg_timeseries
```


All Change Treatment, Aquarium Temperature, Timeseries

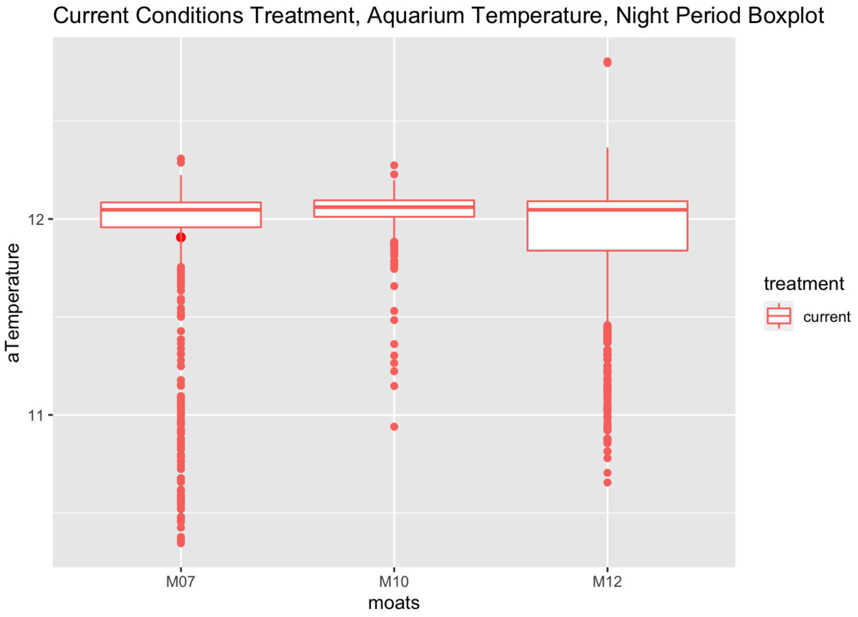
### 9.2.4 Current Day Period Boxplot

```
simpleplot_cur_day <- ggplot(subset(Cdml[Cdml$treatment == "current", ],
                                          period %in% ("day")),
                                          aes(x=moats, y=aTemperature,
                                                         colour=treatment)) +
                                          stat_summary(fun=mean, geom="point",
                                                         size=2, color="red")  +
                                          geom_boxplot() +
                                          ggtitle("Current Conditions Treatment, Aquarium Temperature, Day Period
Boxplot")


simpleplot_cur_day
```


Current Conditions Treatment, Aquarium Temperature, Day Period Boxplot

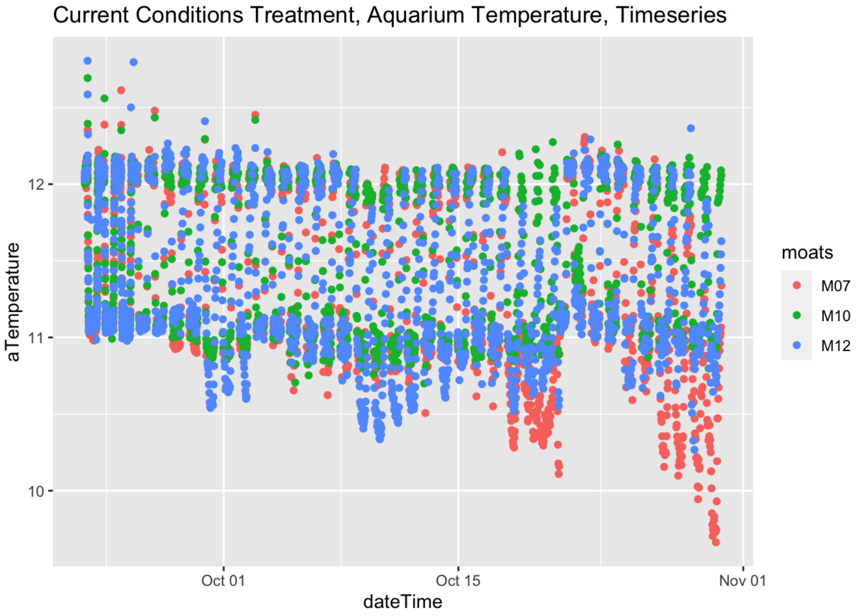
### 9.2.5 Current Night Period Boxplot

```
simpleplot_cur_night <- ggplot(subset(Cdml[Cdml$treatment == "current", ],
                                          period %in% ("night")),
                                          aes(x=moats, y=aTemperature,
                                                        colour=treatment)) +
                                          stat_summary(fun=mean, geom="point",
                                                        size=2, color="red")  +
                                          geom_boxplot() +
                                          ggtitle("Current Conditions Treatment, Aquarium Temperature, Night Peri
od Boxplot")


simpleplot_cur_night
```


Current Conditions Treatment, Aquarium Temperature, Night Period Boxplot

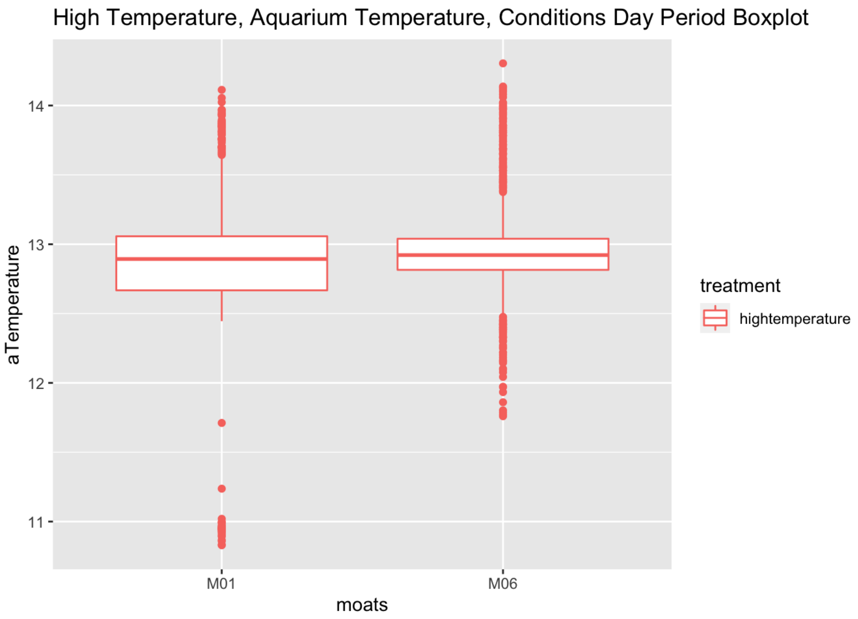
### 9.2.6 Current Conditions Timeseries

```
simpleplot_cur_timeseries <- ggplot(subset(Cdml[Cdml$treatment == "current", ]))+
                                  aes(x=dateTime, y=aTemperature) +
                                  geom_point(aes(colour=moats, point=)) +
                                  ggtitle("Current Conditions Treatment, Aquarium Temperature, Timeseries")

simpleplot_cur_timeseries
```


Current Conditions Treatment, Aquarium Temperature, Timeseries

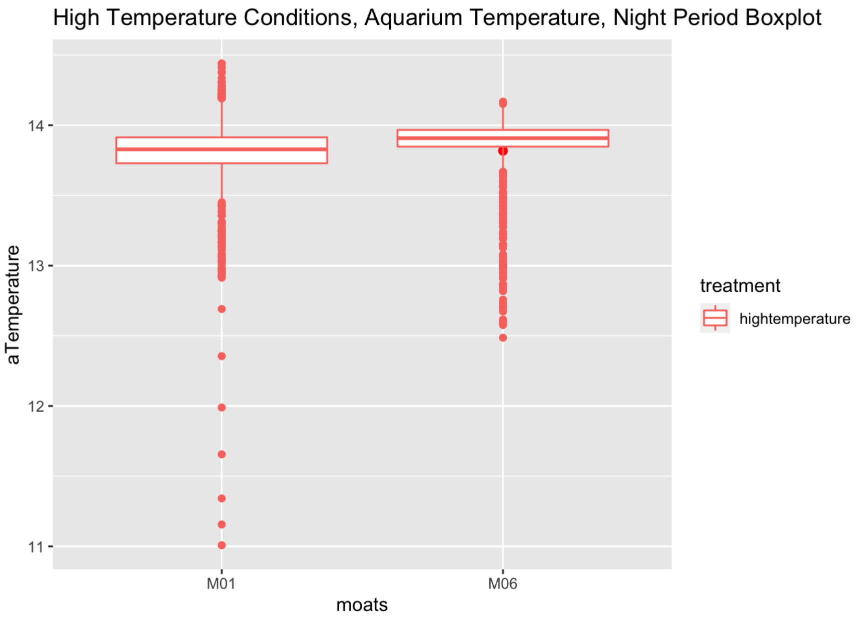
### 9.2.7 High Temperature Day Period Boxplot

```
simpleplot_hitemp_day <- ggplot(subset(Cdml[Cdml$treatment == "hightemperature", ],
                                        period %in% ("day")),
                                        aes(x=moats, y=aTemperature,
                                                      colour=treatment)) +
                                        stat_summary(fun=mean, geom="point",
                                                      size=2, color="red")  +
                                        geom_boxplot() +
                                        ggtitle("High Temperature, Aquarium Temperature, Conditions Day Period
 Boxplot")


simpleplot_hitemp_day
```



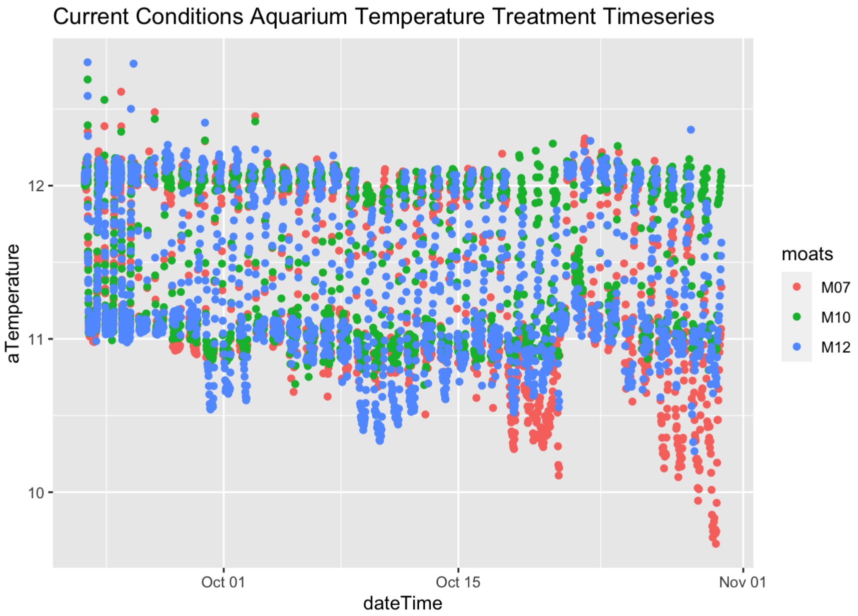
### 9.2.8 High Temperature Night Period Boxplot

```
simpleplot_hitemp_night <- ggplot(subset(Cdml[Cdml$treatment == "hightemperature", ],
                                        period %in% ("night")),
                                        aes(x=moats, y=aTemperature,
                                                      colour=treatment)) +
                                        stat_summary(fun=mean, geom="point",
                                                      size=2, color="red")  +
                                        geom_boxplot() +
                                        ggtitle("High Temperature Conditions, Aquarium Temperature, Night Perio
d Boxplot")


simpleplot_hitemp_night
```



### 9.2.9 High Temperature Timeseries

```
simpleplot_cur_timeseries <- ggplot(subset(Cdml[Cdml$treatment == "current", ]))+
                                  aes(x=dateTime, y=aTemperature) +
                                  geom_point(aes(colour=moats, point=)) +
                                  ggtitle("Current Conditions Aquarium Temperature Treatment Timeseries")

simpleplot_cur_timeseries
```



# 10.) Temperature Jumps

### 10.0 Creating new column

```
## 10.0 Creating new column - showing the different between 2 adjacent times
# first sort the data so you are comparing adjacent times
# then add new column
# the diff() function computes x[i+lag] - x[i], the default value is lag = 1
```

### 10.1 diff()function & deltaTempLag1 Narrative

```
## 10.1 diff()function  & deltaTempLag1 Narrative
#the result of the diff() function has a length that is the length of
# the original vector - lag

#therefore need to fill the new variable, deltaTempLag1,
# with c(0,abs(diff(aTemperature))) it is right length and first value
# (with no valid diff) is zero

# creating a new object to be a validation flag under name deltaTempLag1

# will create a numerical value that will represent those observations
# to be filtered out
```

### 10.2 Creating deltaTempLag1

```
## 10.2 Creating deltaTempLag1
Cdml <- filteredFrame %>% arrange(moats, dateTime)%>%
  mutate(deltaTempLag1 = c(0,abs(diff(aTemperature))))
#the diff value comparing the last time in M01 to the first time in M02 (etc.)
#                                                                  is not valid,
# Set those to zero
Cdml$deltaTempLag1[lag(Cdml$moats) != Cdml$moats] <- 0

#this shows all the rows that jump more that 1 degree from previous row
#                      (ingoring rows that are transition from one moats to another)
Cdml %>% filter(deltaTempLag1 >1)
```

```
##    moats            dateTime aTemperature sTemperature       pH       DO
## 1    M01 2019-10-19 21:59:00     11.71136     11.67883 7.722845 5.179468
## 2    M07 2019-09-26 21:43:00     11.39582     10.83490 7.689979 5.656842
## 3    M08 2019-09-30 21:43:59     13.23410     12.25973 7.548849 5.518189
## 4    M08 2019-10-19 21:19:59     12.95044     12.85344 7.556915 5.804039
## 5    M08 2019-10-24 21:27:59     13.19683     13.42282 7.559878 6.215845
## 6    M08 2019-10-25 21:15:59     12.43890     11.90826 7.586635 6.300650
## 7    M08 2019-10-28 21:47:59     13.33322     13.46518 7.534707 4.086717
## 8    M10 2019-09-26 21:44:59     11.41436     10.86563 7.691018 7.202747
## 9    M12 2019-09-29 21:41:59     11.38074     10.89259 7.690470 5.511001
## 10   M12 2019-10-28 21:33:59     11.28371     11.55635 7.713491 7.449388
## 11   M13 2019-10-09 21:32:59     12.70052     12.62047 7.558021 5.813056
##    salinity       treatment period ObservationDate ObservationTime
## 1      28.8 hightemperature    day      2019-10-20        21:59:00
## 2      28.8         current    day      2019-09-27        21:43:00
## 3      28.8       allchange    day      2019-10-01        21:43:59
## 4      28.8       allchange    day      2019-10-20        21:19:59
## 5      28.8       allchange    day      2019-10-25        21:27:59
## 6      28.8       allchange    day      2019-10-26        21:15:59
## 7      28.8       allchange    day      2019-10-29        21:47:59
## 8      28.8         current    day      2019-09-27        21:44:59
## 9      28.8         current    day      2019-09-30        21:41:59
## 10     28.8         current    day      2019-10-29        21:33:59
## 11     28.8       allchange    day      2019-10-10        21:32:59
##    deltaTempLag1
## 1       1.386639
## 2       1.083771
## 3       1.063837
## 4       1.090132
## 5       1.046439
## 6       1.828905
## 7       1.206709
## 8       1.020578
## 9       1.029939
## 10      1.081021
## 11      1.122322
```

### 10.3 Creating tDeltaThreshold

```{r10.3 Creating tDeltaThreshold}

## 10.3 Creating tDeltaThreshold

#Creating another column of variables to be able to graph the temperature jumps Cdml <- Cdml %>% mutate(tDeltaThreshold = if_else(deltaTempLag1 > 0.5, TRUE, FALSE)) ```

## 11.) Calculating Averages by Treatment & Day/Night

```
## 11.0 Intercept Narrative
# These averages will be the yintercepts in plots
# example of what to put into with ggplot "
#    geom_hline(yintercept = dtemperatur$`mean(aTemperature)`)
```

### 11.1 All Treatments (Day & Night)

```
## 11.1 All Treatments (Day & Night)
# Night Period
subsetNightaTemp <- subset(filteredFrame, period == "night" & aTemperature >0,
                                  select = c( moats, dateTime, treatment, aTemperature ))

# Day Period
subsetDayaTemp <- subset(filteredFrame, period == "day" & aTemperature >0,
                           select = c( moats, dateTime, treatment, aTemperature ))
```

### 11.2 Current Treatment Day/Night Averages

```
## 11.2 Current Treatment
# Current Night
curNightaTemp <- subset(Cdml, period == "night" & treatment == "current",
                           select = c(dateTime, aTemperature ))
avg_curNightaTemp <- mean(curNightaTemp$aTemperature)
print(avg_curNightaTemp)
```

```
## [1] 11.95039
```
```

```
# [1] 11.95122


# Current Day
curDayaTemp <- subset(Cdml, period == "day" & treatment == "current",
                        select = c(dateTime, aTemperature ))
avg_curDayaTemp <- mean(curDayaTemp$aTemperature)
print(avg_curDayaTemp)
```

```
## [1] 11.09783
```

```
# [1] 11.09983
```

## 11.3 All Change Treatment Day/Night Averages

```
## 11.3 All Change Treatment
# All Change Night
allchgNightaTemp <- subset(Cdml, period == "night" & treatment == "allchange",
                        select = c(dateTime, aTemperature ))
avg_allchgNightaTemp <- mean(allchgNightaTemp$aTemperature)
print(avg_allchgNightaTemp)
```

```
## [1] 13.89858
```

```
# [1] 13.89847

# All Change Day
allchgDayaTemp <- subset(Cdml, period == "day" & treatment == "allchange",
                      select = c(dateTime, aTemperature ))
avg_allchgDayaTemp <- mean(allchgDayaTemp$aTemperature)
print(avg_allchgDayaTemp)
```

```
## [1] 12.97331
```

```
# [1] 12.97466
```

## 11.4 High Temperature Treatment

```
# High Temperature Conditions
# High Temperature Night
hitempNightaTemp <- subset(Cdml, period == "night" & treatment == "hightemperature",
                           select = c(dateTime, aTemperature ))
avg_hitempNightaTemp <- mean(hitempNightaTemp$aTemperature)
print(avg_hitempNightaTemp)
```

```
## [1] 13.80435
```

```
#[1] 13.80397

# High Temperature Day
hitempDayaTemp <- subset(Cdml, period == "day" & treatment == "hightemperature",
                           select = c(dateTime, aTemperature ))
avg_hitempDayaTemp <- mean(hitempDayaTemp$aTemperature)
print(avg_hitempDayaTemp)
```

```
## [1] 12.92259
```

```
#[1] 12.92148
```

## 11.5 Review of all Treatments

```
## 11.5 Review of all Treatments
# Review of 6 new variables
avg_allchgDayaTemp
```

```
## [1] 12.97331
```

```
avg_allchgNightaTemp
```

```
## [1] 13.89858
```

```
avg_curDayaTemp
```

```
## [1] 11.09783
```

```
avg_curNightaTemp
```

```
## [1] 11.95039
```

```
avg_hitempDayaTemp
```

```
## [1] 12.92259
```

```
avg_hitempNightaTemp
```

```
## [1] 13.80435
```

### 11.6 Table of Averages

```
## 11.6 Table of Averages
meanCdmldatemp <- filteredFrame %>% group_by(treatment, period) %>%
                                                      summarise(mean(aTemperature))
# Groups:   treatment [3]
#treatment       period `mean(aTemperature)`
#<chr>           <chr>                 <dbl>
# 1 allchange       day                   13.0
# 2 allchange       night                 13.9
# 3 current         day                   11.1
# 4 current         night                 12.0
# 5 hightemperature day                   12.9
# 6 hightemperature night                 13.8
print(meanCdmldatemp)
```

```
## # A tibble: 6 x 3
## # Groups:   treatment [3]
##   treatment       period `mean(aTemperature)`
##   <fct>           <chr>                  <dbl>
## 1 allchange       day                     13.0
## 2 allchange       night                   13.9
## 3 current         day                     11.1
## 4 current         night                   12.0
## 5 hightemperature day                     12.9
## 6 hightemperature night                   13.8
```

# 12.) Summary & Group by Cdml

```
# 12.0 Cdml Day Summary, group by, mutate

Cdml.daynight.summary <- Cdml %>% group_by(treatment, period) %>%
  summarize(sd = sd(aTemperature, na.rm = TRUE),
            mean = mean(aTemperature, na.rm = TRUE),
            median = median(aTemperature, na.rm = TRUE),
            IQR = IQR(aTemperature, na.rm = TRUE),
            n = n()) %>%
  mutate(se = sd/sqrt(n)) %>%
  mutate(ci = se*1.96)


write.csv(Cdml.daynight.summary, "2020.07.08_Cdml_daynight_summary.csv")
```

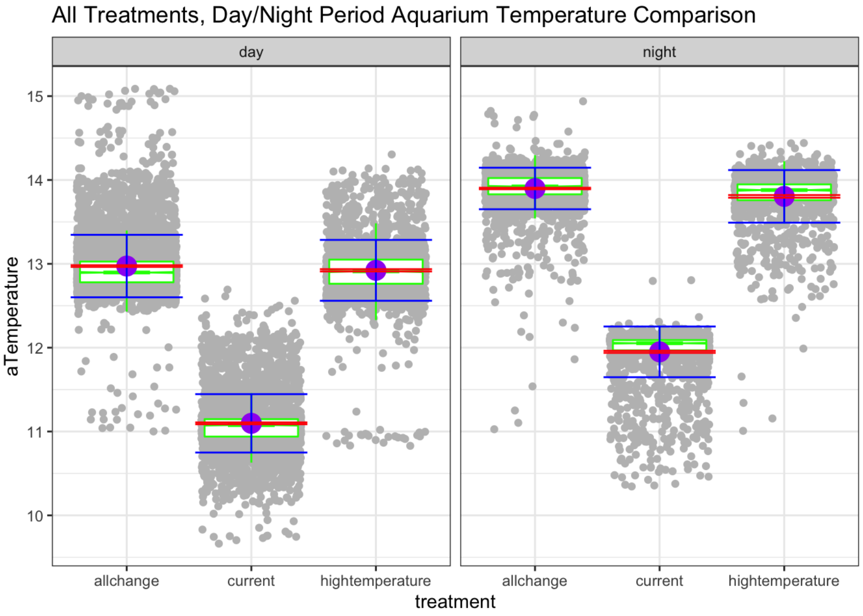
### 12.1 Tribble of Day/Night Summary

```
tibble::tribble(
         ~treatment, ~period,         ~sd,       ~mean,     ~median,        ~IQR,    ~n,         ~se,          ~c
i,
        "allchange",   "day", 0.372506855, 12.97330841,   12.896611,  0.24780225, 4956L, 0.005291376, 0.010371097
,
        "allchange", "night", 0.247588899, 13.89858376, 13.92320525, 0.193622875, 2843L, 0.004643471, 0.009101203
,
          "current",   "day", 0.348206031, 11.09782575, 11.07660075, 0.207690563, 3974L, 0.005523602, 0.010826259
,
          "current", "night", 0.302526369, 11.95038837,  12.0529935, 0.122263063, 2290L, 0.006321869, 0.012390863
,
  "hightemperature",   "day", 0.362913386, 12.92259015,  12.9101975, 0.288474875, 2687L, 0.007001146, 0.013722247
,
  "hightemperature", "night", 0.313433995, 13.80435016, 13.87954738, 0.188507875, 1504L, 0.008082062, 0.015840842
  )
```

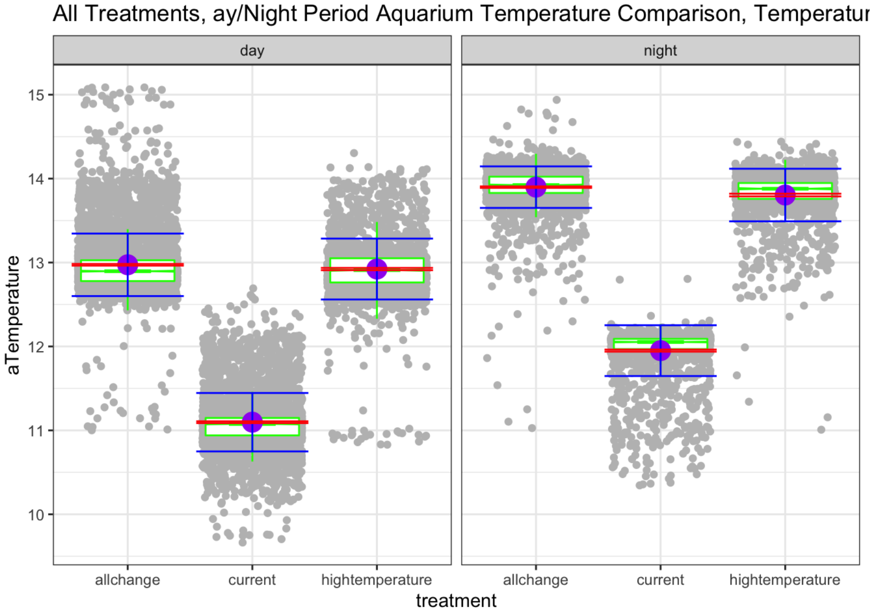
```
## # A tibble: 6 x 9
##   treatment       period    sd  mean median   IQR     n      se      ci
##   <chr>           <chr>  <dbl> <dbl>  <dbl> <dbl> <int>   <dbl>   <dbl>
## 1 allchange       day    0.373  13.0   12.9 0.248  4956 0.00529 0.0104
## 2 allchange       night  0.248  13.9   13.9 0.194  2843 0.00464 0.00910
## 3 current         day    0.348  11.1   11.1 0.208  3974 0.00552 0.0108
## 4 current         night  0.303  12.0   12.1 0.122  2290 0.00632 0.0124
## 5 hightemperature day    0.363  12.9   12.9 0.288  2687 0.00700 0.0137
## 6 hightemperature night  0.313  13.8   13.9 0.189  1504 0.00808 0.0158
```

# 13.) Plots

```
## 13.1 Boxplot aTemp by moats
#boxplot(aTemperature~moats, Cdml)

ggplot(Cdml, aes(treatment, aTemperature)) +
            geom_jitter(color = "grey") +
            geom_boxplot(notch = TRUE, outlier.shape = NA, colour = "green") +
            geom_point(data = Cdml.daynight.summary, aes(x=treatment, y=mean), size=5, color = "purple") +
            geom_errorbar(data = Cdml.daynight.summary,
                          aes(x=treatment, y=mean, ymin = mean-sd, ymax = mean+sd),
                          color = "blue") +
            geom_errorbar(data = Cdml.daynight.summary,
                          aes(x=treatment, y=mean, ymin = mean-ci, ymax = mean+ci),
                          colour = "red") +
            facet_wrap(~period) +
            ggtitle("All Treatments, Day/Night Period Aquarium Temperature Comparison") +
            theme_bw()
```

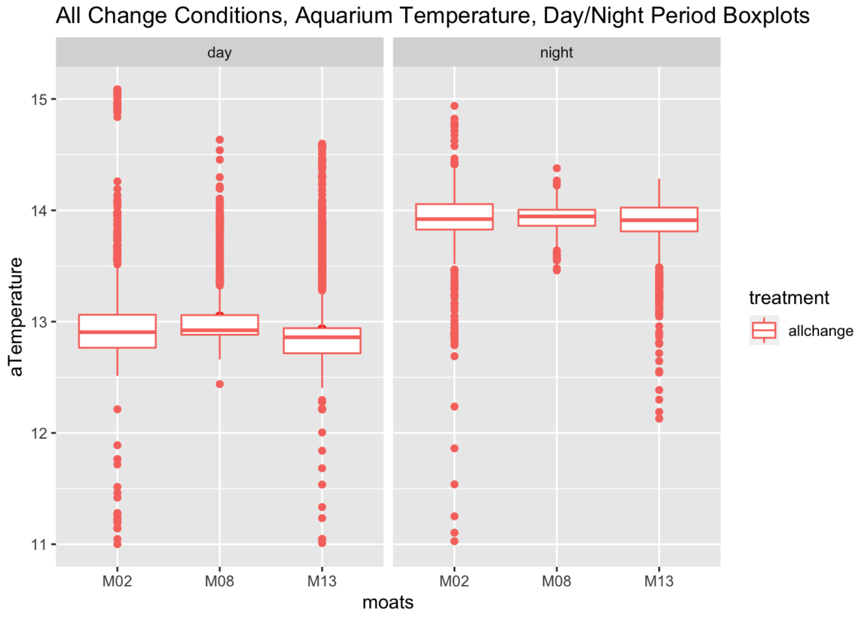


```
## 13.1a Boxplot aTemp by moats
#boxplot(aTemperature~moats, Cdml)

ggplot(Cdml, aes(treatment, aTemperature)) +
            geom_jitter(color = "grey") +
            geom_boxplot(notch = TRUE, outlier.shape = NA, colour = "green") +
            geom_point(data = Cdml.daynight.summary, aes(x=treatment, y=mean), size=5, color = "purple") +
            geom_errorbar(data = Cdml.daynight.summary,
                          aes(x=treatment, y=mean, ymin = mean-sd, ymax = mean+sd),
                          color = "blue") +
            geom_errorbar(data = Cdml.daynight.summary,
                          aes(x=treatment, y=mean, ymin = mean-ci, ymax = mean+ci),
                          colour = "red") +
            facet_wrap(~period) +
            ggtitle("All Treatments, ay/Night Period Aquarium Temperature Comparison, Temperature Jumps at .5C")
 +
            theme_bw()
```

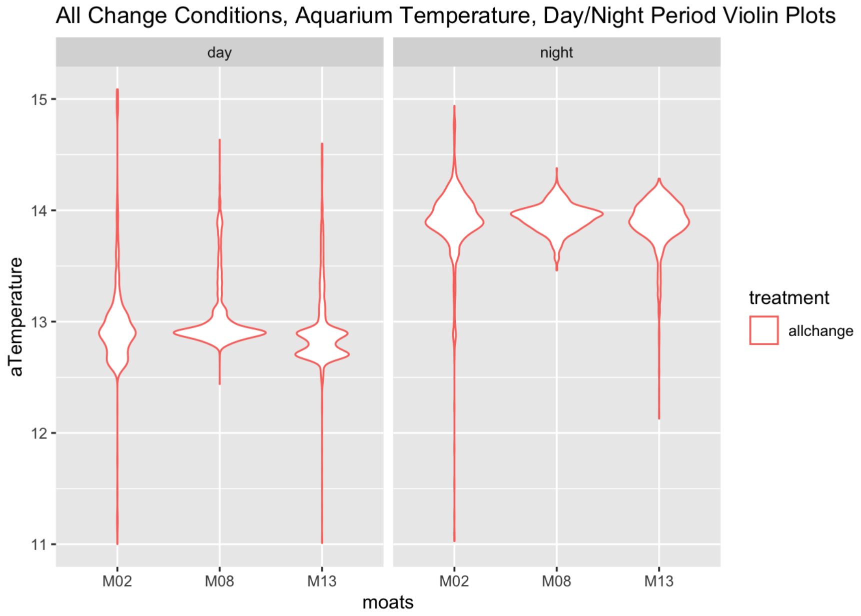


### 13.1.1 Aquarium Temperature, All Change Conditions, Day & Night

```
p_allchg_facetwrap <- ggplot(subset(Cdml[Cdml$treatment == "allchange", ])) +
                                          aes(x=moats, y=aTemperature,
                                                        colour=treatment) +
                                          stat_summary(fun=mean, geom="point",
                                                        size=2, color="red")  +
                                          geom_boxplot() +
                                          facet_wrap(~period) +
                                          ggtitle("All Change Conditions, Aquarium Temperature, Day/Night Period
 Boxplots")

p_allchg_facetwrap
```

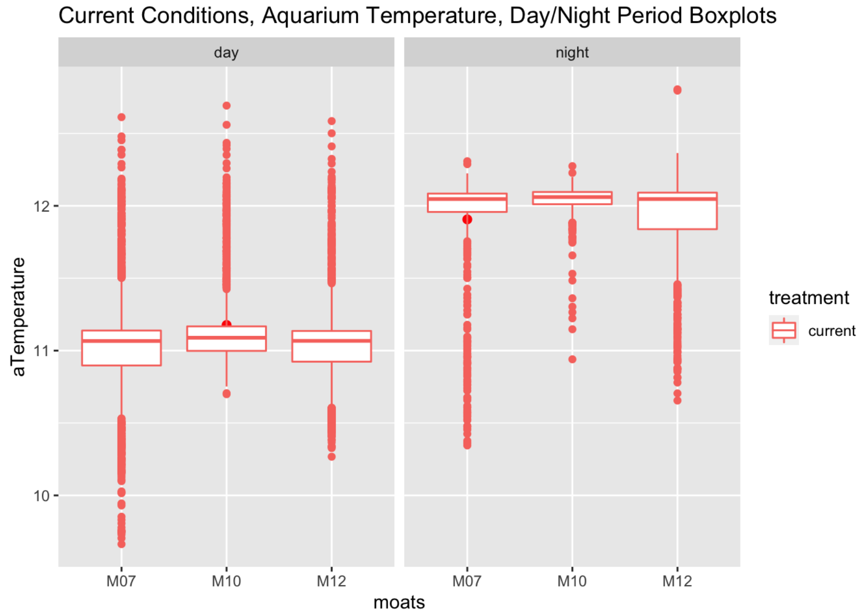


```
p_allchg_facetwrap_vio <- ggplot(subset(Cdml[Cdml$treatment == "allchange", ])) +
                                          aes(x=moats, y=aTemperature,
                                                        colour=treatment) +
                                          stat_summary(fun=mean, geom="point",
                                                        size=2, color="red")  +
                                          geom_violin() +
                                          facet_wrap(~period) +
                                          ggtitle("All Change Conditions, Aquarium Temperature, Day/Night Period
 Violin Plots")

p_allchg_facetwrap_vio
```

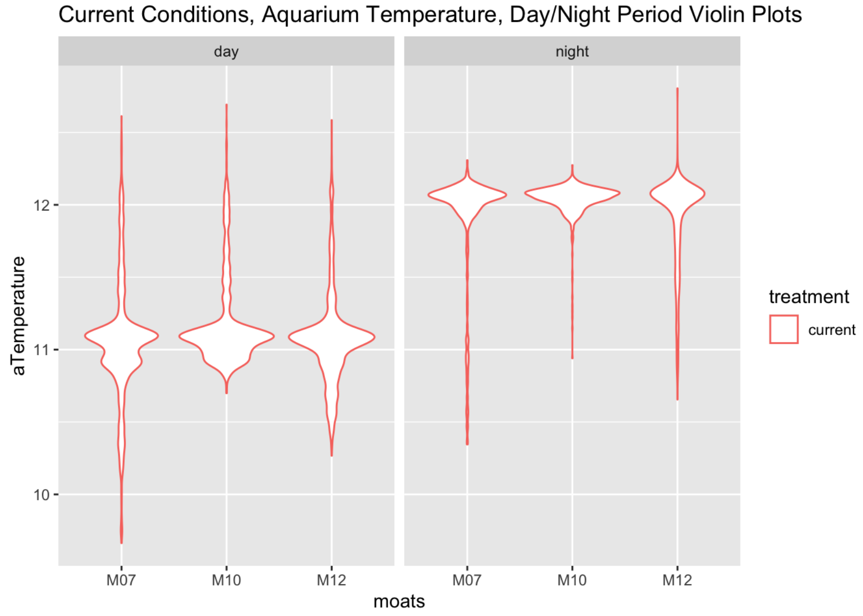


### 13.1.3 Aquarium Temperature, Current Conditions, Day & Night

```
p_cur_facetwrap_box <- ggplot(subset(Cdml[Cdml$treatment == "current", ])) +
                                          aes(x=moats, y=aTemperature,
                                                        colour=treatment) +
                                          stat_summary(fun=mean, geom="point",
                                                        size=2, color="red")  +
                                          geom_boxplot() +
                                          facet_wrap(~period) +
                                          ggtitle("Current Conditions, Aquarium Temperature, Day/Night Period Box
plots")

p_cur_facetwrap_box
```

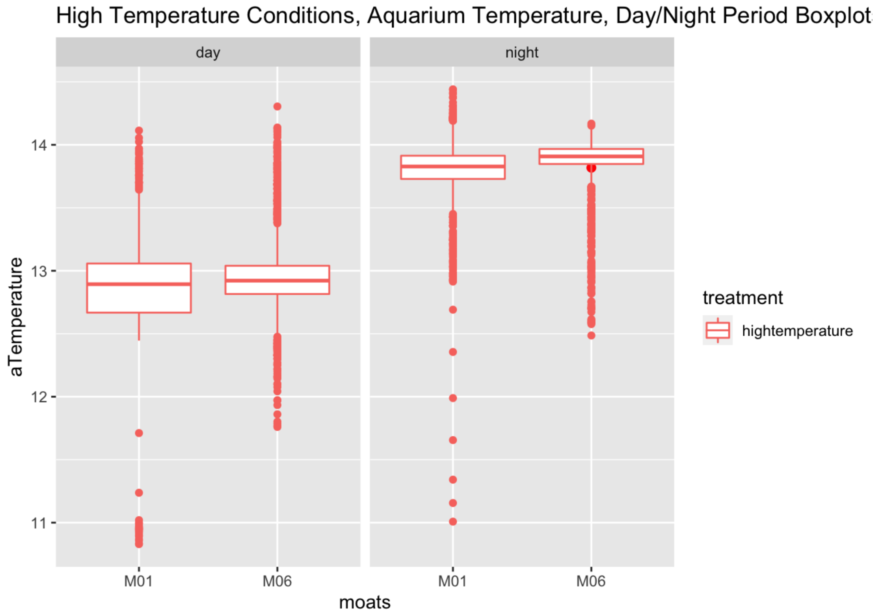


```
p_allchg_facetwrap_vio <- ggplot(subset(Cdml[Cdml$treatment == "current", ])) +
                                          aes(x=moats, y=aTemperature,
                                                        colour=treatment) +
                                          stat_summary(fun=mean, geom="point",
                                                        size=2, color="red")  +
                                          geom_violin() +
                                          facet_wrap(~period) +
                                          ggtitle("Current Conditions, Aquarium Temperature, Day/Night Period Vio
lin Plots")

p_allchg_facetwrap_vio
```



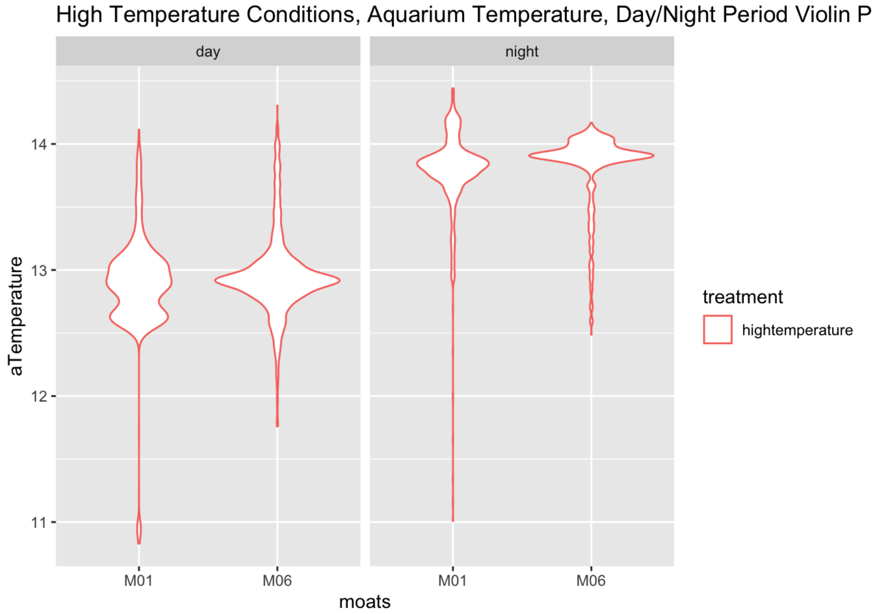
### 13.1.4 Aquarium Temperature, High Temperature Conditions, Day & Night Boxplots

```
p_hitemp_facetwrap_box <- ggplot(subset(Cdml[Cdml$treatment == "hightemperature", ])) +
                                          aes(x=moats, y=aTemperature,
                                                        colour=treatment) +
                                          stat_summary(fun=mean, geom="point",
                                                        size=2, color="red")  +
                                          geom_boxplot() +
                                          facet_wrap(~period) +
                                          ggtitle("High Temperature Conditions, Aquarium Temperature, Day/Night P
eriod Boxplots")

p_hitemp_facetwrap_box
```



### 13.1.5 Aquarium Temperature, High Temperature Conditions, Day & Night Violin plots

```
p_hitemp_facetwrap_vio <- ggplot(subset(Cdml[Cdml$treatment == "hightemperature", ])) +
                                          aes(x=moats, y=aTemperature,
                                                        colour=treatment) +
                                          stat_summary(fun=mean, geom="point",
                                                        size=2, color="red")  +
                                          geom_violin() +
                                          facet_wrap(~period) +
                                          ggtitle("High Temperature Conditions, Aquarium Temperature, Day/Night P
eriod Violin Plots")

p_hitemp_facetwrap_vio
```

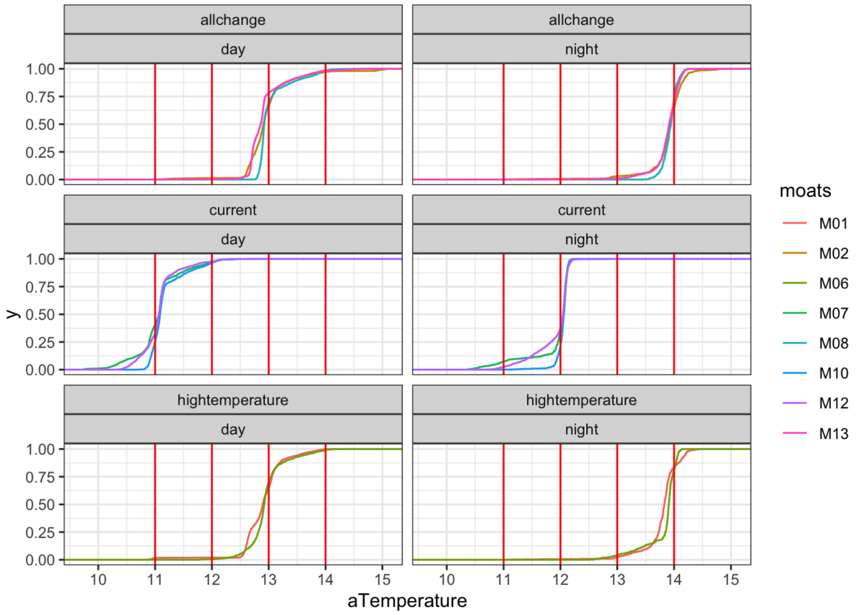


# 14. Cumulative Frequency Distribution

A Cumulative Frequency Distribution plot was used to illustration the time spent around the mean per period per treatment.

### 14.1a CFD plot 1

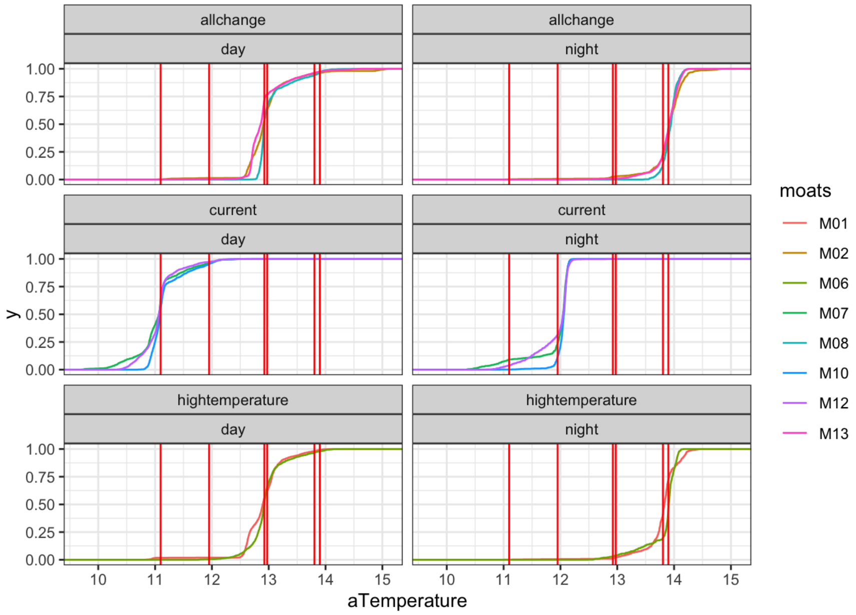
```
CFD1 <- ggplot(Cdml, aes(aTemperature)) +
            stat_ecdf(aes(colour = moats)) +
            geom_vline(xintercept = c(11,12,13,14), colour = "red") +
            facet_wrap(vars(treatment, period), ncol = 2, scales = "fixed")  +
            theme_bw()
CFD1
```



## 14.2 Empirical Cumulative Distribution Function Generated Plots

Using a ECDF to better visualize distribution Compute empirical cumulative distribution : The empirical cumulative distribution function (ECDF) provides an alternative visualisation of distribution. Compared to other visualisations that rely on density (like geom_histogram()), the ECDF doesn’t require any tuning parameters and handles both continuous and categorical variables.

```
## 14.2a ECDF Plot

ECDFplot <-      ggplot(Cdml, aes(aTemperature)) +
                stat_ecdf(aes(colour = moats)) +
                geom_vline(xintercept = Cdml.daynight.summary$mean, colour = "red") +
                facet_wrap(vars(treatment, period), ncol = 2, scales = "fixed")  +
                theme_bw()
ECDFplot
```



## 14.3 Creating a function to incorporate our statisitics

```
## 14.3 Creating a function to incorporate our statisitics- show our averages

 coolECDF <- function(d, dsum, treat, per){
   dsum <- dsum %>% filter(treatment == treat & period == per )
   p <- d %>% filter(treatment == treat & period == per ) %>% ggplot(aes(aTemperature)) +
     stat_ecdf(aes(colour = moats)) +
     geom_vline(xintercept = dsum$mean, colour = "red") +
     ggtitle(paste(treat, per)) +
     theme_bw()
   return(p)
 }
```

### 14.4 Levels ECDF

```
  levels(Cdml$treatment)
```

```
## [1] "allchange"       "current"         "hightemperature"
```

```
  levels(factor(Cdml$period))
```

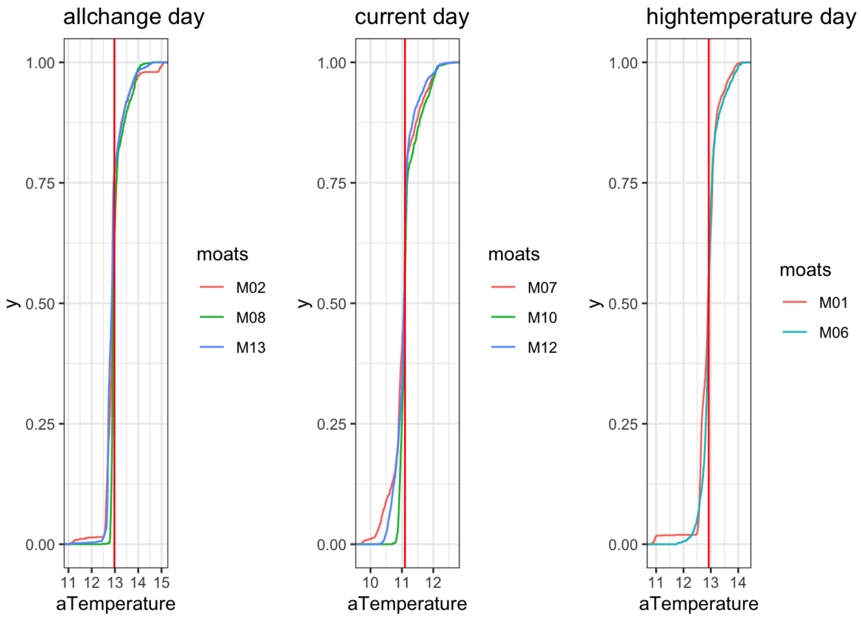
```
## [1] "day"   "night"
```

### 14.5 Creating the different averages from the Summary Table (ECDF)

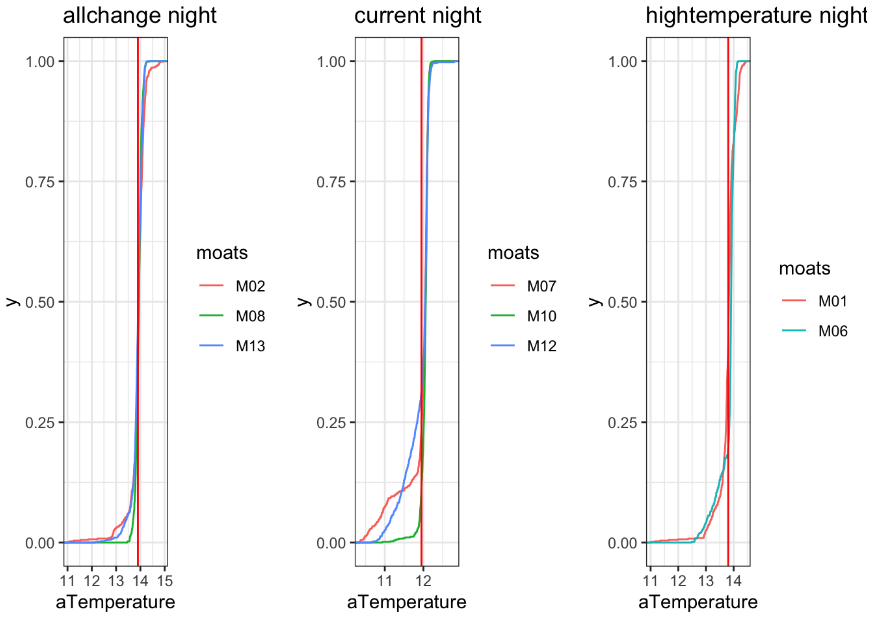
```
        cfd.a.d <- coolECDF(Cdml, Cdml.daynight.summary, "allchange", "day")
        cfd.a.n <- coolECDF(Cdml, Cdml.daynight.summary, "allchange", "night")
        cfd.c.d <- coolECDF(Cdml, Cdml.daynight.summary, "current", "day")
        cfd.c.n <- coolECDF(Cdml, Cdml.daynight.summary, "current", "night")
        cfd.h.d <- coolECDF(Cdml, Cdml.daynight.summary, "hightemperature", "day")
        cfd.h.n <- coolECDF(Cdml, Cdml.daynight.summary, "hightemperature", "night")
```

### 14.6 Incorporating Grid Arrange to display the ECDF plots

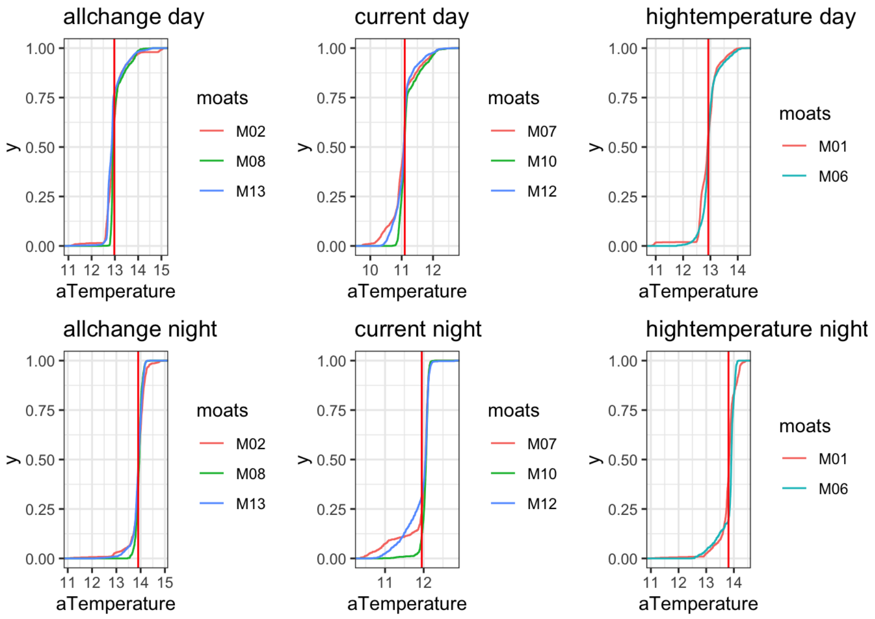
```
## Incorporating Grid Arrange to display the ECDF plots
#  Basic Grid Arrange

      grid.arrange(cfd.a.d, cfd.c.d, cfd.h.d, nrow = 1)
```



```
      grid.arrange(cfd.a.n, cfd.c.n, cfd.h.n, nrow = 1)
```



```
#make a list of all the plots then pass the list to grid.arrange()
ecdfList <- list(cfd.a.d, cfd.c.d, cfd.h.d, cfd.a.n, cfd.c.n, cfd.h.n)
grid.arrange(grobs = ecdfList, ncol=3)
```



# 15.0 Temperature Investigation

## 15.1 Temperature Investigation High Temperature

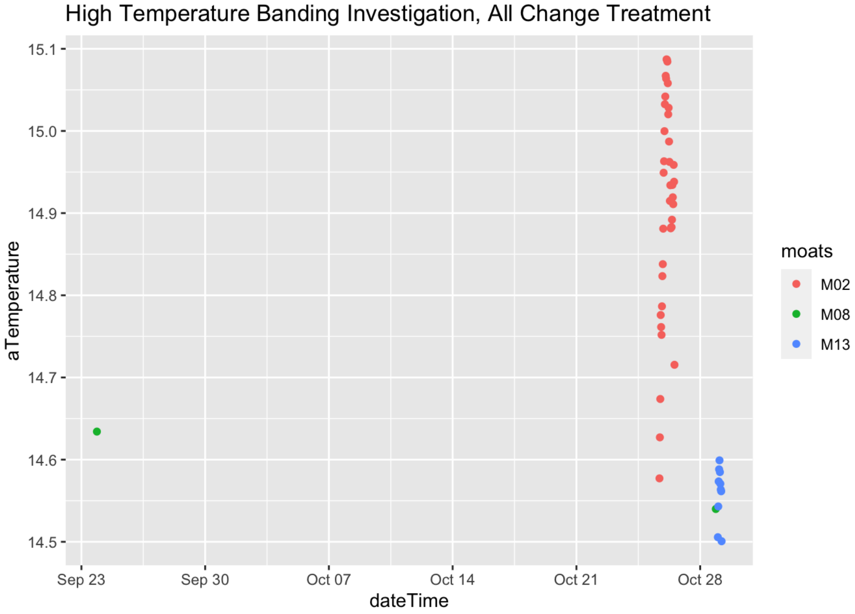
```
# Selecting for the high aTemperature Band
 investig.HighaTemp <- Cdml %>%
   filter(aTemperature>= 14.50 & aTemperature<=15.50) %>%
   filter(treatment %in% c("current",
                            "allchange",
                            "hightemperature")) %>%
   filter(period != "other")
```

## 15.1a High Temperature Banding Investigation, All Change Treatment

```
hitemp_plot1 <- ggplot(subset(investig.HighaTemp[investig.HighaTemp$treatment == "allchange", ])) +
        aes(x=dateTime, y=aTemperature) +
   geom_point(aes(colour=moats, point=)) +
   ggtitle("High Temperature Banding Investigation, All Change Treatment")
   ylim (10, 20)
```

```
## <ScaleContinuousPosition>
##  Range:
##  Limits:   10 --   20
```

```
hitemp_plot1
```

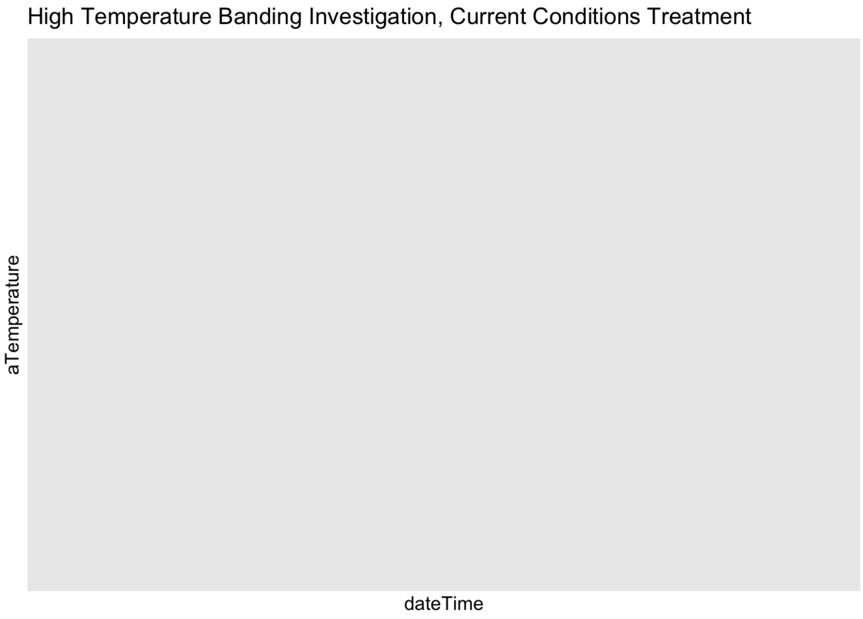


### 15.1b High Temperature Banding Investigation, Current Conditions Treatment

```
hitemp_plot2 <- ggplot(subset(investig.HighaTemp[investig.HighaTemp$treatment == "current", ])) +
        aes(x=dateTime, y=aTemperature) +
   geom_point(aes(colour=moats, point=)) +
   ggtitle("High Temperature Banding Investigation, Current Conditions Treatment")
   ylim (10, 20)
```

```
## <ScaleContinuousPosition>
##  Range:
##  Limits:   10 --   20
```

```
hitemp_plot2
```

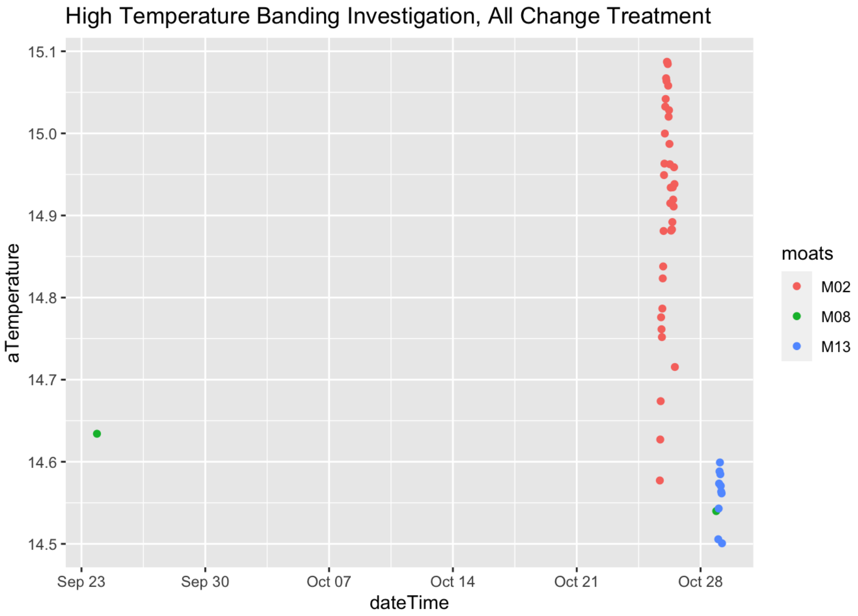


### 15.1c High Temperature Banding Investigation, High Temperature Treatment Conditions Treatment

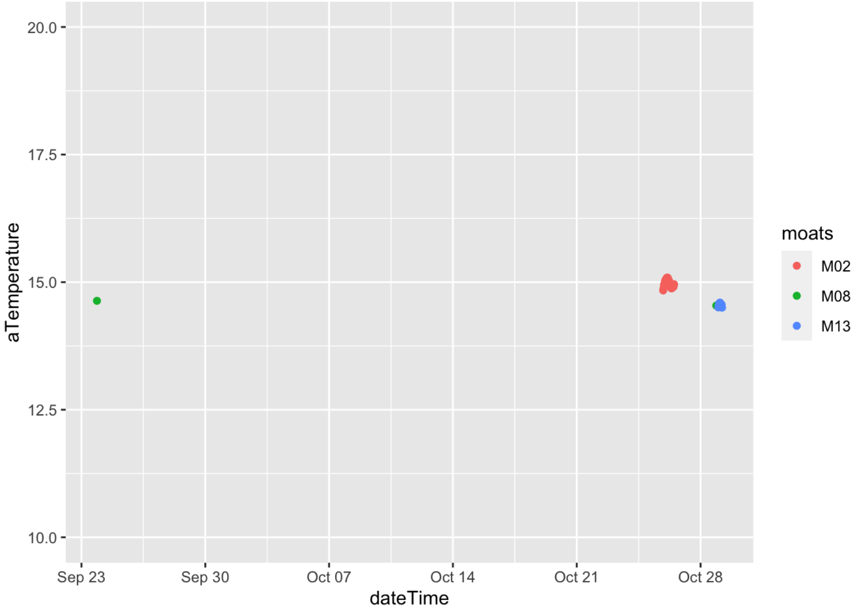
```{r15.1b High Temperature Banding Investigation, High Temperature Treatment Treatment}

hitemp_plot3 <- ggplot(subset(investig.HighaTemp[investig.HighaTemp$treatment == “hightemperature”, ])) + aes(x=dateTime, y=aTemperature) + geom_point(aes(colour=moats, point=)) + ggtitle(“High Temperature Banding Investigation, High Temperature Treatment”) ylim (10, 20)
```

hitemp_plot3 ```

```
 # Time series plot
hitemp_plot4 <- ggplot(subset(investig.HighaTemp[investig.HighaTemp$treatment == "allchange", ],
               period %in% ("night")),
        aes(x=dateTime, y=aTemperature)) +
   geom_point(aes(colour=moats, point=)) +
   ylim (10, 20)

hitemp_plot1
```



```
 # Time series plot
hitemp_plot7 <- ggplot(subset(investig.HighaTemp[investig.HighaTemp$treatment == "allchange", ],
               period %in% ("day")),
        aes(x=dateTime, y=aTemperature)) +
   geom_point(aes(colour=moats, point=)) +
   ylim (10, 20)

hitemp_plot7
```



current to line 879

#*************E*N*D*************# #********************************* ## END OF SCRIPT | END OF DOCUMENT #*********************************

#________$

$..
#______

$
# ______

$$

$_$
# _____

$

$
# ______

$

$
# _____

$

$_

$
# ____

$$

$_____

$ # ____

$

$____$
# ____

$

$
# _____

$

$
# _____

$$

$
# ______

$$

$
# _

___

$
# __

$$

$
# _

$$

```
$
# __
```

```
$
#
```

```
$
# __$__
```

$

```
$
# ____
```

$

```
$
# ____
```

```
$
# ___
```

$

```
$____$
# ___
```

$

```
$___
```

```
# ____
```

```
$___$__
```

# _**$$**

```
$_____
```

**# _$$**

```
$____
```

**$ #** _____

$$

```
$
# _________
```

$$