

Supplement of Earth Syst. Sci. Data, 10, 27–51, 2018
<https://doi.org/10.5194/essd-10-27-2018-supplement>
© Author(s) 2018. This work is distributed under
the Creative Commons Attribution 4.0 License.



Supplement of

Over 10 million seawater temperature records for the United Kingdom Continental Shelf between 1880 and 2014 from 17 Cefas (United Kingdom government) marine data systems

D. J. Morris et al.

Correspondence to: David J. Morris (david.morris@cefas.co.uk)

The copyright of individual parts of the supplement might differ from the CC BY 4.0 License.

```
1 ---
2 title:
3 author: "David Morris"
4 output:
5   word_document:
6     toc: true
7     toc_depth: 2
8     toc_float:
9     collapsed: true
10    smooth_scroll: false
11    number_sections: true
12 ---
13
14 ```{r setup, include=FALSE}
15 knitr::opts_chunk$set(echo = TRUE, fig.keep = 'all', message=FALSE, warning=FALSE)
16 ```
17
18 # BACKGROUND INFORMATION
19
20 Title is left blank because the Journal inserts it.
21
22 Code style based on https://google.github.io/styleguide/Rguide.xml
23
24 ## Copyright Statement
25
26 This code is subject to Crown copyright protection. The material must be
27 acknowledged as Crown copyright in this manner:
28
29 *Crown copyright, 2017*
30
31 ## Data Availability
32
33 Cefas data are available from the Cefas Data Hub -
34 https://www.cefas.co.uk/cefas-data-hub/
35
36 The data cited below are available from https://doi.org.10.14466/Cefasdatahub.4.
37 This is a zipped single .csv file that contains all the data from all the Sources.
38 Zipped file name is *CefasSWTdata20170707sourceALL*. This is an 84 MByte file that
39 unzips to a 957 MByte .csv file of the same name.
40
41 ## Software Used
42
43 R Core Team (2016). R: A language and environment for statistical computing. R
44 Foundation for Statistical Computing, Vienna, Austria. URL
45 https://www.R-project.org/. R version 3.3.2
46
47 RStudio (2016). RStudio: Integrated development environment for R (Version 0.99.903)
48 [Computer software]. Boston, MA. Retrieved September 27, 2016. Available from
49 http://www.rstudio.org/RStudio version 1.0.44
50
51 # AUTHOR COMMENT (David Morris)
52
53 This file contains the R code used to analyse the Quality Controlled data assembled
54 from 17 diverse and disparate Cefas Data Systems for this Data Paper.
55
56 The data sources and an overview of the extensive data assembly processes involved
57 are described in the accompanying publication. In summary, data were obtained from
58 Cefas Data Owners / Stewards as folder/files and extracts from various operational
59 databases and systems.
60
61 Data were processed to extract temperature records as described in the manuscript,
62 with over 15,000 source files totalling ~ 15 G Bytes, reducing to some 100+
63 'summary' files resulting from the extracts of the many and diverse, bespoke formats
64 and/or Excel/.csv/.txt/.dat, data source files; each designed for the particular
65 requirements of the data system involved.
66
67 Data processing included standardising formats, extracting reference fields, a
68 minimum/maximum sense check process to detect system and sensor errors, a
69 geographical sense check plot & Temperature .v. Time plots to examine outliers and
70 data at inappropriate levels for the time of year.
71
72 # FILE DESCRIPTION
73
```

54 The total data set from all 17 sources was saved as an R file (.Rda) and this is used for the following code.

55

56 Code is written within my limited R coding experience and for accessibility rather than computational elegance or efficiency. This approach inevitably evolved as the coding requirements emerged so there are different ways of doing things, especially around controlling plot outputs (I started with `*plot*` ended with `*ggplot*` and had to 'solve' issues around discrete & continuous scales and over plotting). Please accept my apologies for any inelegance & inconsistency.

57

58 Similarly, the code is verbose and repetitive, e.g. when dealing with monthly plots. I recognise the use of loops would reduce this but they came late to the party and I initially, as recommended on numerous websites, avoided them as something else new. The use of lists would have shortened the code but these were deferred as part of the focus on the ends rather than the means. Any potential for errors in the copying and editing is mitigated by the clarity of each line being clear as to what it is doing.

59

60 The code provides sections on what was done to select data for plotting (maps and plots) with a separate section (chunk) for plots to the files used in the manuscript. I encountered ggplot output issues with multiple plots; my solution means that they don't appear in the compiled Markdown document and my 'quick fix' (reimporting the files) seems unnecessary as the aim of this document is to present and explain the code and data workings. The file plots were set to the specified publication standards.

61

62 # SOURCE & LIBRARY STATEMENTS

63

64 ## Local parameters

65

66 WINDOWS 10 (16 Gigabyte RAM)

67

68 R 3.3.2

69

70 RStudio 1.0.44

71

72 # ACQUIRE THE DATA

73

74 The main .csv file is provided as a .zip (~84 MBytes) which extracts to a ~957 MByte .csv file which takes a long time to "load" (faster R .csv read methods are available). The POSIXct conversion also takes time.

75

76 ## Set local parameters and load .Rda file

77

78 The final code development used a .Rda file generated from the `*CefasSWTdata20170707sourceALL*` .csv file. An import process for the .csv file is described below. My understanding is that the provision of .Rda files as a means of sharing data is not necessarily recommended on technical grounds (see e.g. `*Useful References*` at <https://www.loc.gov/preservation/digital/formats/fdd/fdd000470.shtml>).

79

80 ****Local setup****

81

82 ```{r Localsetup}

83 projectPath="C:/Users/djm00/OneDrive - CEFAS/SWT-MASTER"

84 # INSERT APPROPRIATE LOCAL PATH

85 setwd(projectPath)

86 load("CefasSWTdata20170707.Rda") # n = 10,680,821

87 SWTfinal <- CefasSWTdata20170707 # minimises my code editing arising from changes in the data following systematic and detailed QC checks using temperature levels and month to remove anomalies that appeared in the plots as they were developed.

88 rm(CefasSWTdata20170707) # reduces memory overheads and chances of changes to core file

89 ```

90

91 ## Read from .csv files

92

93 The following section is text rather than an R code `*chunk*` so it doesn't execute under the `*knitr*` compilation of the document. It is included to allow re-creation of the .Rda file above from the downloaded .csv "all data" file.

94

95 *******

96 CefasSWTdata20170707 <- read.csv("CefasSWTdata20170707sourceAll.csv", header=TRUE, sep="," , stringsAsFactors = FALSE)

```

97  ***
98
99  This provides an R data frame called *CefasSWTdata20170707* with 10,680,440
    observations of 11 variables.
100
101  ### Create Time as DateTime
102
103  ***
104  CefasSWTdata20170707$Time <- as.POSIXct(CefasSWTdata20170707$Time)
105  summary(CefasSWTdata20170707)
106  ***
107
108  A check using the *summary* command showed the .Rda used and data frame created from
    the .csv file are identical following the as.POSIXct conversion above.
109
110  # LIBRARIES
111
112  Libraries used in the assembly, checking & plotting processes
113
114  Maps have been created using the R package *marmap*.
115
116  *sessionInfo()* used for version numbers
117
118  **Load libraries*
119
120  ```{r loadLibraries}
121  library(data.table) # 1.10.4
122  library(ggmap) # 2.6.1
123  library(ggplot2) # 2.2.1
124  library(graphics)
125  library(gridExtra) # 2.2.1
126  library(lubridate) # 1.6.0
127  library(mapproj) # 1.2-4
128  library(marmap) # 0.9.6
129  library(raster) # 2.5-8
130  library(rgdal) # 1.2-5
131  library(rgeos) # 0.3-22
132  library(sp) # 1.2-3
133  library(lattice) # 0.20-34
134  library(scatterplot3d) # 0.3-40
135  ```
136
137  Other libraries have been added as the manuscript progressed. These are inserted
    where they are needed.
138
139  # FUNCTION DEFINITIONS
140
141  ## Multiple Plots
142
143  Uses http://www.cookbook-r.com/Graphs/Multiple\_graphs\_on\_one\_page\_\(ggplot2\)/
144
145  ```{r Multiple Plots FUNCTION}
146  # Multiple plot function
147  #
148  # ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)
149  # - cols: Number of columns in layout
150  # - layout: A matrix specifying the layout. If present, 'cols' is ignored.
151  #
152  # If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),
153  # then plot 1 will go in the upper left, 2 will go in the upper right, and
154  # 3 will go all the way across the bottom.
155  #
156  multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
157    library(grid)
158
159    # Make a list from the ... arguments and plotlist
160    plots <- c(list(...), plotlist)
161
162    numPlots = length(plots)
163
164    # If layout is NULL, then use 'cols' to determine layout
165    if (is.null(layout)) {
166      # Make the panel

```

```

167     # ncol: Number of columns of plots
168     # nrow: Number of rows needed, calculated from # of cols
169     layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
170                      ncol = cols, nrow = ceiling(numPlots/cols))
171   }
172
173   if (numPlots==1) {
174     print(plots[[1]])
175
176   } else {
177     # Set up the page
178     grid.newpage()
179     pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))
180
181     # Make each plot, in the correct location
182     for (i in 1:numPlots) {
183       # Get the i,j matrix positions of the regions that contain this subplot
184       matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))
185
186       print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
187                                     layout.pos.col = matchidx$col))
188     }
189   }
190 }
191 ...
192
193 ## Provide Scientific Notation
194
195 http://stackoverflow.com/questions/11610377/how-do-i-change-the-formatting-of-numbers-on-an-axis-with-ggplot
196
197 ```{r SciNote}
198 fancy_scientific <- function(l) {
199   # turn in to character string in scientific notation
200   l <- format(l, scientific = TRUE)
201   # quote the part before the exponent to keep all the digits
202   l <- gsub("^(.*)e", "'\\1'e", l)
203   # turn the 'e+' into plotmath format
204   l <- gsub("e", "%*%10^", l)
205   # return this as an expression
206   parse(text=l)
207 }
208 ...
209
210 # LOCATION PARAMETERS
211
212 These specify to coordinates for maps and subsets of data; provided as NESW bounding
213 boxes.
214
215 ## Bounding Boxes for maps
216
217 ukcs is a short form for an area representing the UK Continental Shelf
218
219 ```{r BBoxes4Maps}
220 ukcs.north= 63
221 ukcs.east= 10
222 ukcs.south= 48
223 ukcs.west= -15
224 #
225 ukcsPlus.north= 82
226 ukcsPlus.east= 54
227 ukcsPlus.south= 37
228 ukcsPlus.west= -57
229 #
230 ices.north= 55
231 ices.east= 5
232 ices.south= 48
233 ices.west= -10
234 #
235 SNS.north = 54
236 SNS.east=6
237 SNS.south=51
238 SNS.west=0

```

```

238   ```
239
240   ## Bounding Boxes for ICES areas
241
242   ```{r BBoxesICES}
243   lpoolBay.north=54
244   lpoolBay.east=-3
245   lpoolBay.south=53
246   lpoolBay.west=-5
247   #
248   celticSea.north=51
249   celticSea.east=-7
250   celticSea.south=50
251   celticSea.west=-9
252   #
253   brixham.north=50.5
254   brixham.east=-2
255   brixham.south=49.5
256   brixham.west=-4
257   #
258   thames.north=52.5
259   thames.east=3
260   thames.south=51.5
261   thames.west=1
262   ```
263
264   ## Bounding Boxes for the Southern North Sea area
265
266   ```{r BBoxesSNS}
267   SNSbb.north=53.5
268   SNSbb.east=4
269   SNSbb.south=51.5
270   SNSbb.west=2
271   ```
272
273   ## Fix for Swiss GADM not plotting & Greenland/Turkey "tears"
274
275   Following the initial selection of *marmap* as the source for basemaps, internal
   requests for changes to formatting resulted in the use of country overlays which
   created issues that were resolved by the overlaying of suitably coloured rectangles.
   These are defined here.
276
277   Some "tears" in the country overlays, e.g. the file for Russia generating a
   horizontal band (because, I believe, of an unclosed polygon from the Aleutian's)
   were dealt with by constraining the map size. The following were not, hence the
   quick fix.
278
279   **Define the rectangles to overlay tears in the maps**
280
281   ```{r CHEfix}
282   box.CHE <- data.frame(maxlat = 52, minlat = 48, maxlong = 20, minlong = 6, id="a")
283   box.CHE <- transform(box.CHE, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
284   #
285   box.GRN <- data.frame(maxlat = 82, minlat = 72, maxlong = -25, minlong = -57, id="a")
286   box.GRN <- transform(box.GRN, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
287   #
288   box.TUR.1 <- data.frame(maxlat = 41, minlat = 37, maxlong = 45, minlong = 28, id="a")
289   box.TUR.1 <- transform(box.TUR.1, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
290   #
291   box.TUR.2 <- data.frame(maxlat = 41.6, minlat = 41, maxlong = 38, minlong = 32,
   id="a")
292   box.TUR.2 <- transform(box.TUR.2, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
293   #
294   ```
295
296   # BASE MAPS (uses marmap)
297
298   keep = TRUE retains the relevant .csv files downloaded from the NOAA database. This
   explains the *File already exists; loading 'marmap_coord_XXXX'* message that appears
   in the compiled version of this code but which will not appear the first time you
   run the *getNOAA.bathy* function.
299
300   **Acquire the Base Maps**

```

```

301
302 ```{r BaseMaps}
303 map.ukcs <- getNOAA.bathy(ukcs.west, ukcs.east, ukcs.south, ukcs.north, res = 1,
304 keep=TRUE)
305 map.ukcsPlus <- getNOAA.bathy(ukcsPlus.west, ukcsPlus.east, ukcsPlus.south,
306 ukcsPlus.north, res = 4, keep=TRUE)
307 map.ices <- getNOAA.bathy(ices.west, ices.east, ices.south, ices.north, res = 1,
308 keep=TRUE)
309 map.SNS<-getNOAA.bathy(SNS.west, SNS.east, SNS.south, SNS.north, res = 1, keep=TRUE)
310
311 All maps have used marmap plus darkgreen country overlays; this is consistent
312 for all figures.
313
314 NOTE the density plot does not allow a scaled fill for the data AND, e.g. the use of
315 a simple fill for the topography, e.g. "etopo". Issues were also encountered in
316 retaining the bathymetry & with over plots of land which were needed to mask "smear"
317 from point plots and the (clearly impossible) land components of the density plots.
318 The resulting combination of marmap and density plots resulted in compromises and
319 quick fixes that exemplify the it's ugly but it works issues that arise between
320 professional coders and scientists who code. The option of using the single world
321 map from GADM requires dealing with shape files, another 'extension' in coding that
322 was time constrained and hence not used.
323
324 ## Download the Country polygons
325
326 As indicated above this is a less than elegant way of working around issues that
327 arose from the initial selection of marmap and the subsequent requirement for
328 density plots.
329
330 Acquire and "fortify" the Country polygons
331
332 GADM is the spatial database of Global Administrative Areas (http://www.gadm.org/)
333 and is accessed using the sp package.
334
335 ```{r CountryPolygons}
336 # Download relevant country polygons - see http://gadm.org/country UK & R
337 SpatialPolygon - level 0 admin - Ver 2.8
338 #
339 #uses library(sp)
340 #
341 ALBcoast <- readRDS("ALB_adm0.rds")
342 ARMcoast <- readRDS("ARM_adm0.rds")
343 AUTcoast <- readRDS("AUT_adm0.rds")
344 AZEcoast <- readRDS("AZE_adm0.rds")
345 BELcoast <- readRDS("BEL_adm0.rds")
346 BGRcoast <- readRDS("BGR_adm0.rds")
347 BIHcoast <- readRDS("BIH_adm0.rds")
348 BLRcoast <- readRDS("BLR_adm0.rds")
349 CANcoast <- readRDS("CAN_adm0.rds")
350 CHEcoast <- readRDS("CHE_adm0.rds")
351 DEUcoast <- readRDS("DEU_adm0.rds")
352 DNKcoast <- readRDS("DNK_adm0.rds")
353 ESPcoast <- readRDS("ESP_adm0.rds")
354 ESTcoast <- readRDS("EST_adm0.rds")
355 FINcoast <- readRDS("FIN_adm0.rds")
356 FRAcoast <- readRDS("FRA_adm0.rds")
357 FROcoast <- readRDS("FRO_adm0.rds")
358 GBRcoast <- readRDS("GBR_adm0.rds")
359 GRCcoast <- readRDS("GRC_adm0.rds")
360 GEOcoast <- readRDS("GEO_adm0.rds")
361 GRLcoast <- readRDS("GRL_adm0.rds")
362 HRVcoast <- readRDS("HRV_adm0.rds")
363 HUNcoast <- readRDS("HUN_adm0.rds")
364 IMNcoast <- readRDS("IMN_adm0.rds")
365 IRLcoast <- readRDS("IRL_adm0.rds")
366 IRNcoast <- readRDS("IRN_adm0.rds")
367 ISLcoast <- readRDS("ISL_adm0.rds")
368 ITAcoast <- readRDS("ITA_adm0.rds")
369 KAZcoast <- readRDS("KAZ_adm0.rds")
370 LIEcoast <- readRDS("LIE_adm0.rds")
371 LTUcoast <- readRDS("LTU_adm0.rds")
372 LUXcoast <- readRDS("LUX_adm0.rds")

```

```
358 LVAcoast <- readRDS("LVA_adm0.rds")
359 MDAcoast <- readRDS("MDA_adm0.rds")
360 MKDcoast <- readRDS("MKD_adm0.rds")
361 MNEcoast <- readRDS("MNE_adm0.rds")
362 NLDcoast <- readRDS("NLD_adm0.rds")
363 NORcoast <- readRDS("NOR_adm0.rds")
364 POLcoast <- readRDS("POL_adm0.rds")
365 PRTcoast <- readRDS("PRT_adm0.rds")
366 ROUcoast <- readRDS("ROU_adm0.rds")
367 RUScoast <- readRDS("RUS_adm0.rds")
368 SJMcoast <- readRDS("SJM_adm0.rds")
369 SRBcoast <- readRDS("SRB_adm0.rds")
370 SVKcoast <- readRDS("SVK_adm0.rds")
371 SVNcoast <- readRDS("SVN_adm0.rds")
372 SWEcoast <- readRDS("SWE_adm0.rds")
373 TKMcoast <- readRDS("TKM_adm0.rds")
374 TURcoast <- readRDS("TUR_adm0.rds")
375 UKRcoast <- readRDS("UKR_adm0.rds")
376 XKOcoast <- readRDS("XKO_adm0.rds")
377 #
378 # proj4string(GBRcoast) # confirms WGS84
379 # GBRmerc <- spTransform(GBRcoast, CRS("+init=epsg:3395")) # not needed
380 #
381 ALB <- fortify(ALBcoast, id="id")
382 ARM <- fortify(ARMcoast, id="id")
383 AUT <- fortify(AUTcoast, id="id")
384 AZE <- fortify(AZEcoast, id="id")
385 BEL <- fortify(BELcoast, id="id")
386 BGR <- fortify(BGRcoast, id="id")
387 BIH <- fortify(BIHcoast, id="id")
388 BLR <- fortify(BLRcoast, id="id")
389 CAN <- fortify(CANcoast, id="id")
390 CHE <- fortify(CHEcoast, id="id")
391 DEU <- fortify(DEUcoast, id="id")
392 DNK <- fortify(DNKcoast, id="id")
393 ESP <- fortify(ESPcoast, id="id")
394 EST <- fortify(ESTcoast, id="id")
395 FIN <- fortify(FINcoast, id="id")
396 FRA <- fortify(FRAcoast, id="id")
397 FRO <- fortify(FROcoast, id="id")
398 GBR <- fortify(GBRcoast, id="id")
399 GRC <- fortify(GRCcoast, id="id")
400 GEO <- fortify(GEOcoast, id="id")
401 GRL <- fortify(GRLcoast, id="id")
402 HRV <- fortify(HRVcoast, id="id")
403 HUN <- fortify(HUNcoast, id="id")
404 IMN <- fortify(IMNcoast, id="id")
405 IRL <- fortify(IRLcoast, id="id")
406 IRN <- fortify(IRNcoast, id="id")
407 ISL <- fortify(ISLcoast, id="id")
408 ITA <- fortify(ITAcoast, id="id")
409 KAZ <- fortify(KAZcoast, id="id")
410 LIE <- fortify(LIEcoast, id="id")
411 LTU <- fortify(LTUcoast, id="id")
412 LVA <- fortify(LVAcoast, id="id")
413 LUX <- fortify(LUXcoast, id="id")
414 MDA <- fortify(MDAcoast, id="id")
415 MKD <- fortify(MKDcoast, id="id")
416 MNE <- fortify(MNEcoast, id="id")
417 NLD <- fortify(NLDcoast, id="id")
418 NOR <- fortify(NORcoast, id="id")
419 POL <- fortify(POLcoast, id="id")
420 PRT <- fortify(PRTcoast, id="id")
421 ROU <- fortify(ROUcoast, id="id")
422 RUS <- fortify(RUScoast, id="id")
423 SJM <- fortify(SJMcoast, id="id")
424 SRB <- fortify(SRBcoast, id="id")
425 SVK <- fortify(SVKcoast, id="id")
426 SVN <- fortify(SVNcoast, id="id")
427 SWE <- fortify(SWEcoast, id="id")
428 TKM <- fortify(TKMcoast, id="id")
429 TUR <- fortify(TURcoast, id="id")
430 UKR <- fortify(UKRcoast, id="id")
```



```

431 XKO <- fortify(XKOcoast,id="id")
432 ```
433
434 # FIGURES
435 **Figures 1 & 2 are images of Research Vessels.**
436
437 The following Sections provide the code used to generate the Figures used in the
Data Paper.
438
439 ## Common /Map & / or Plot parameters
440
441 ```{r CommonParams}
442 PointColour="red3" # all maps and figures
443 ```
444
445 # FIGURE 3
446 **Overview of the location of "all" Cefas seawater temperature measurements**
447
448 The ESSD target journal requires images to be no less than 8cm wide and at 300 dpi,
with vector formats (e.g. PDF) preferred. PNG are used for images (e.g. Research
vessels) and maps. Plot file size is also a consideration. Point and text sizes are
adjusted to suit the file output so may not look as good on screen/in the printed
Notebook. Installation issues for PDF output unresolved so PNG is default format.
449
450 ## F3 Create the required Labels for Lat & Long
451
452 Create the breaks- and label vectors
453
454 http://stackoverflow.com/questions/33302424/format-latitude-and-longitude-axis-labels-
in-ggplot
455
456 ```{r Fig3CreateLabels}
457 ewbrks <- seq(-50,50,20) # 50 West to 50 East in 10 degree units
458 nsbrks <- seq(40,70,10) # differ from ukcsPlus etc as rounded to 5 and 10
459 ewlbls <- unlist(lapply(ewbrks, function(x) ifelse(x < 0, paste(x*-1, "°W"),
ifelse(x > 0, paste(x, "°E"),x)))
460 nslbls <- unlist(lapply(nsbrks, function(x) ifelse(x < 0, paste(x*-1, "°S"),
ifelse(x > 0, paste(x, "°N"),x)))
461 ```
462
463 ## F3 Create the map
464
465 ```{r Fig3PlotToFile}
466 ppi=300
467 Figure.width = 16/2.54
468 Figure.height = 16/2.54
469 #
470 autoplot(map.ukcsPlus,geom="contour",colour="gray",size=0.1) +
471 scale_x_continuous(limits = c(ukcsPlus.west, ukcsPlus.east), breaks=ewbrks,
labels=ewlbls, expand = c(0, 0)) +
472 scale_y_continuous(limits = c(ukcsPlus.south, ukcsPlus.north), breaks=nsbrks,
labels=nslbls, expand = c(0, 0)) +
473 coord_map(xlim=c(ukcsPlus.west,ukcsPlus.east),ylim=c(ukcsPlus.south,ukcsPlus.north)) +
474 theme(panel.background = element_rect(fill = "lightsteelblue1")) +
475 geom_point(aes(x=Long,y=Lat), data = SWTfinal, col=PointColour, size=0.5, alpha =
1/10) +
476 labs(x="Longitude", y="Latitude") +
477 theme(axis.title = element_text(size = 12)) +
478 theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12)) +
479 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=RUS) +
480 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=ALB) +
481 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=ARM) +
482 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=AUT) +
483 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=AZE) +
484 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BEL) +
485 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BGR) +
486 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BIH) +
487 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BLR) +
488 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=CAN) +
489 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=CHE) +
490 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=DEU) +
491 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=DNK) +
492 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=ESP) +

```

```

493 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=EST) +
494 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=FIN) +
495 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=FRA) +
496 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=FRO) +
497 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=GBR) +
498 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=GEO) +
499 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=GRC) +
500 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=GRL) +
501 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=HRV) +
502 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=HUN) +
503 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=IMN) +
504 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=IRL) +
505 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=IRN) +
506 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=ISL) +
507 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=ITA) +
508 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=KAZ) +
509 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=LIE) +
510 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=LTU) +
511 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=LUX) +
512 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=LVA) +
513 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=MDA) +
514 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=MKD) +
515 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=MNE) +
516 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=NLD) +
517 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=NOR) +
518 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=POL) +
519 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=PRT) +
520 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=ROU) +
521 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=SJM) +
522 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=SRB) +
523 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=SVK) +
524 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=SVN) +
525 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=SWE) +
526 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=TKM) +
527 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=TUR) +
528 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=UKR) +
529 geom_polygon(aes(x=long,y=lat, group=group), fill="darkgreen", data=XKO) +
530 geom_rect(data=box.CHE, aes(xmin=minlong, xmax=maxlong, ymin=minlat, ymax=maxlat),
  color="darkgreen", fill="darkgreen", inherit.aes=FALSE) +
531 geom_rect(data=box.GRN, aes(xmin=minlong, xmax=maxlong, ymin=minlat, ymax=maxlat),
  color="darkgreen", fill="darkgreen", inherit.aes=FALSE) +
532 geom_rect(data=box.TUR.1, aes(xmin=minlong, xmax=maxlong, ymin=minlat,
  ymax=maxlat), color="darkgreen", fill="darkgreen", inherit.aes=FALSE) +
533 geom_rect(data=box.TUR.2, aes(xmin=minlong, xmax=maxlong, ymin=minlat,
  ymax=maxlat), color="darkgreen", fill="darkgreen", inherit.aes=FALSE) +
534 theme(axis.text = element_text(size=12))
535 #
536 # https://gis.stackexchange.com/questions/165974/r-fortify-causing-polygons-to-tear
537 # Greenland and Turkey 'tear' - fixed the image with polygons for now
538 ggsave("fig03.png", width=Figure.width, height = Figure.height, dpi=ppi)
539 ```
540
541 # FIGURE 4
542 **Data Density Plot - UKCS waters**
543
544 ## F4 Subset the data
545
546 ```{r Fig4CreateData}
547 ukcsAll <- subset(SWTfinal, Lat < ukcs.north)
548 ukcsAll <- subset(ukcsAll, Lat > ukcs.south)
549 ukcsAll <- subset(ukcsAll, Long < ukcs.east)
550 ukcsAll <- subset(ukcsAll, Long > ukcs.west)
551 ukcsLatLong <- ukcsAll[c("Lat", "Long")]
552 LatLongTotal<-nrow(ukcsLatLong)
553 #
554 ReductionFactor = 5 # just fits into 16 Gbyte Cefas Windows 10 PC limit
555 #
556 ukcsSeq <-ukcsLatLong
557 ukcsSeq <- ukcsSeq[seq(1,LatLongTotal,ReductionFactor),]
558 ```
559
560 ## F4 Plot the data
561

```

```

562 ```{r Fig4CreateDensityData&Plot}
563 #DENSITY PARAMETERS
564 # provide optimum plot to illustrate data coverage
565 #
566 binNum = 2000
567 denSize=0.1
568 #
569 # Data for Density calculations
570 d <- ukcsSeq
571 #
572 # SET UP OUTPUT FILE
573 ppi=300
574 Figure.width = 16/2.54
575 Figure.height = 16/2.54
576 #
577 # CALCULATE DENSITIES AND PLOT
578 #
579 # 16 Gbyte Windows Memory limits approached - auto memory clear by R insufficiently
reliable
580 gc()
581 # LABELS
582 ewbrks <- seq(-15,10,5)
583 nsbrks <- seq(45,60,5)
584 ewlbls <- unlist(lapply(ewbrks, function(x) ifelse(x < 0, paste(x*-1, "°W"),
ifelse(x > 0, paste(x, "°E"),x)))
585 nslbls <- unlist(lapply(nsbrks, function(x) ifelse(x < 0, paste(x*-1, "°S"),
ifelse(x > 0, paste(x, "°N"),x)))
586 #
587 # Using the previous code does not work well. The density & contours result in
inland Norway & Europe not plotting andthe coastline of France plotting along the
coast in a straight line from the border to the edge of the plot wth darkgreen fills
acriss bats. Using scale_x_discrete (and for y) fixes this but loses the background
lightsteelblue and the grid lines AND the axis annotations, although the degrre fix
works.
588 # moving theme(panel.background...) to the end of the code makes no difference.
589 # manually aligning the scales in the labels and the discrete scale works! duh!!!
590 # That will teach me to copy & paste too much.
591 autoplot(map.ukcs, geom="contour",colour="gray", size=0.1) +
592   coord_map(xlim=c(ukcs.west,ukcs.east),ylim=c(ukcs.south,ukcs.north)) +
593   scale_x_discrete(limits = c(-15,-10,-5,0,5,10), breaks=ewbrks, labels=ewlbls,
expand = c(0, 0)) +
594   scale_y_discrete(limits = c(45,50,55,60), breaks=nsbrks, labels=nslbls, expand =
c(0, 0)) +
595   labs(x="Longitude", y="Latitude") +
596   stat_density2d(data = d, aes(x = Long, y = Lat, fill = ..level.., alpha =
..level..), size = 10, bins = binNum, geom = "polygon") +
597   geom_density2d(data = d, aes(x = Long, y = Lat, size = denSize),
colour=PointColour, size=0.01) +
598   scale_fill_gradient2(low = "cyan", mid="green",high = "yellow",guide =
"colourbar") +
599   scale_alpha(range = c(0, 0.3), guide = FALSE) +
600   theme(legend.position="none") +
601   geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BEL) +
602   #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=CHE) +
603   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DNK) +
604   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DEU) +
605   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=FRA) +
606   geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=FRO) +
607   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=GBR) +
608   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IMN) +
609   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IRL) +
610   #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LIE) +
611   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LUX) +
612   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NLD) +
613   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NOR) +
614   theme(panel.background = element_rect(fill = "lightsteelblue1"))
615 #
616 ggsave("fig04.png", width=Figure.width, height = Figure.height, dpi=ppi)
617 #
618 ```
619
620 # FIGURE 5
621 **Overview of numbers of "all" Cefas seawater temperature measurements by year**

```

```

622
623 ## F5 Subset the data
624
625 ```{r Fig5CreateData}
626 #
627 SWT.early<-subset(SWTfinal,SWTfinal$Time<"1960-01-01 00:00:00")
628 SWT.late<-subset(SWTfinal,SWTfinal$Time>="1940-01-01 00:00:00")
629 #
630 ```
631
632 ## F5 Plot the data
633
634 ```{r Fig5plotFile}
635 ppi=300
636 Figure.width = 16/2.54
637 Figure.height = 10/2.54
638 png("fig05.png", type = "cairo", width = Figure.width, height = Figure.height, units
639 = "in", pointsize = 12, res = ppi)
640 p1 <- ggplot(data = SWT.early, aes(x=Time)) +
641   geom_histogram() +
642   scale_y_continuous(labels = fancy_scientific) +
643   labs(x="Year", y="Count") +
644   ggtitle("(a)") +
645   theme_bw()
646 #
647 p2 <- ggplot(data = SWT.late, aes(x=Time)) +
648   geom_histogram() +
649   scale_y_continuous(labels = fancy_scientific) +
650   labs(x="Year", y="Count") +
651   ggtitle("(b)") +
652   theme_bw()
653 #
654 multiplot(p1,p2, cols = 2)
655 dev.off()
656 ```
657
658 # FIGURE 6
659 **Overview of "all" Cefas seawater temperature measurements by depth - 1 m bin**
660
661 ## F6 Subset the data
662
663 a <=10, b >10 <=100 c >100 <=250 d >250 (surface, thermocline/productive, shelf, rest)
664
665 Surface means top 10 m here - to display data distribution only - this changes for
666 other, comparison based graphs below.
667
668 ```{r Fig6subsetData}
669 surface<-SWTfinal
670 surface<-subset(surface,Depth<=10)
671 #
672 productive<-SWTfinal
673 productive<-subset(productive,Depth>10)
674 productive<-subset(productive,Depth<=100)
675 #
676 shelf<-SWTfinal
677 shelf<-subset(shelf, Depth>100)
678 shelf<-subset(shelf, Depth<=250)
679 #
680 deepest<-SWTfinal
681 deepest<-subset(deepest, Depth>250)
682 #
683 # cross check
684 #
685 print(paste0("Surface", NROW(surface)))
686 print(paste0("Productive", NROW(productive)))
687 print(paste0("Shelf", NROW(shelf)))
688 print(paste0("Deepest", NROW(deepest)))
689 #
690 print(" ")
691 print(paste0("Sum of above", NROW(surface) + NROW(productive) + NROW(shelf)
692 + NROW(deepest)))
693 print(paste0("Total", NROW(SWTfinal)))
694 #

```

```

692   ```
693
694   ## F6 Plot the data
695
696   ```{r Fig6plotFile}
697   ppi=300
698   Figure.width = 16/2.54
699   Figure.height = 14/2.54
700   png("fig06.png", type = "cairo", width = Figure.width, height = Figure.height, units
701       = "in", pointsize = 12, res = ppi)
702   #
703   p1 <- ggplot(data = surface, aes(x=Depth)) +
704     geom_histogram() +
705     scale_y_continuous(labels = fancy_scientific) +
706     labs(x="Depth (m)", y="Count") +
707     ggtitle("(a)") +
708     theme(plot.title = element_text(hjust = 0.5)) +
709     theme_bw()
710   p2 <- ggplot(data = productive, aes(x=Depth)) +
711     geom_histogram() +
712     scale_y_continuous(labels = fancy_scientific) +
713     labs(x="Depth (m)", y="Count") +
714     ggtitle("(b)") +
715     theme(plot.title = element_text(hjust = 0.5)) +
716     theme_bw()
717   p3 <- ggplot(data = shelf, aes(x=Depth)) +
718     geom_histogram() +
719     scale_y_continuous(labels = fancy_scientific) +
720     labs(x="Depth (m)", y="Count") +
721     ggtitle("(c)") +
722     theme(plot.title = element_text(hjust = 0.5)) +
723     theme_bw()
724   p4 <- ggplot(data = deepest, aes(x=Depth)) +
725     geom_histogram() +
726     scale_y_continuous(labels = fancy_scientific) +
727     labs(x="Depth (m)", y="Count") +
728     ggtitle("(d)") +
729     theme(plot.title = element_text(hjust = 0.5)) +
730     theme_bw()
731   #
732   multiplot(p1,p3,p2,p4, cols = 2)
733   dev.off()
734   ```
735
736   # FIGURE 7
737   **Location of Cefas temperature data by ICES Rectangle group - SURFACE (0-5m)**
738
739   ## F7 Set the selected depth thresholds.
740
741   ```{r DepthThresholds}
742   #
743   # 5m used to include intake data from Ferries (1.5) & RVs (4, 4.5)
744   #
745   upperdepth.threshold=0
746   lowerdepth.threshold=5
747   interimdepth.threshold = 1 # includes SmartBuouy & WaveNet data - 1m and 0.4m
748   nominal sensor depths respectively
749   ```
750
751   ## F7 Subset the data
752
753   ```{r Fig7subsetData}
754   #
755   SWTfinalSURFACE<-subset(SWTfinal, Depth>=upperdepth.threshold &
756     Depth<=lowerdepth.threshold)
757   #
758   swt.lpoolBay<-SWTfinalSURFACE
759   swt.lpoolBay<-subset(swt.lpoolBay, Lat <= lpoolBay.north)
760   swt.lpoolBay<-subset(swt.lpoolBay, Lat >= lpoolBay.south)
761   swt.lpoolBay<-subset(swt.lpoolBay, Long >= lpoolBay.west)
762   swt.lpoolBay<-subset(swt.lpoolBay, Long <= lpoolBay.east)
763   #
764   swt.celticSea<-SWTfinalSURFACE

```

```

762 swt.celticSea<-subset(swt.celticSea, Lat <= celticSea.north)
763 swt.celticSea<-subset(swt.celticSea, Lat >= celticSea.south)
764 swt.celticSea<-subset(swt.celticSea, Long >= celticSea.west)
765 swt.celticSea<-subset(swt.celticSea, Long <= celticSea.east)
766 #
767 swt.brixham<-SWTfinalSURFACE
768 swt.brixham<-subset(swt.brixham, Lat <= brixham.north)
769 swt.brixham<-subset(swt.brixham, Lat >= brixham.south)
770 swt.brixham<-subset(swt.brixham, Long >= brixham.west)
771 swt.brixham<-subset(swt.brixham, Long <= brixham.east)
772 #
773 swt.thames<-SWTfinalSURFACE
774 swt.thames<-subset(swt.thames, Lat <= thames.north)
775 swt.thames<-subset(swt.thames, Lat >= thames.south)
776 swt.thames<-subset(swt.thames, Long >= thames.west)
777 swt.thames<-subset(swt.thames, Long <= thames.east)
778 #
779 ```
780
781 ## F7 Plot the data
782
783 ```{r Fig7CreateLabels}
784 ewbrks <- seq(-10,4,2)
785 nsbrks <- seq(48, 54,2)
786 ewlbls <- unlist(lapply(ewbrks, function(x) ifelse(x < 0, paste(x*-1, "°W"),
787 ifelse(x > 0, paste(x, "°E"),x))))
787 nslbls <- unlist(lapply(nsbrks, function(x) ifelse(x < 0, paste(x*-1, "°S"),
788 ifelse(x > 0, paste(x, "°N"),x))))
788 ```
789
790 ```{r Fig7 createBounding Boxes}
791 box.Lpool <- data.frame(maxlat = 54, minlat = 53, maxlong = -3, minlong = -5,
792 id="(a)")
792 box.Lpool <- transform(box.Lpool, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
793 box.Celtic <- data.frame(maxlat = 51, minlat = 50, maxlong = -7, minlong = -9,
794 id="(b)")
794 box.Celtic <- transform(box.Celtic, laby=(maxlat +minlat )/2,
795 labx=(maxlong+minlong)/2)
795 box.Brixham <- data.frame(maxlat = 50.5, minlat = 49.5, maxlong = -2, minlong = -4,
796 id="(c)")
796 box.Brixham <- transform(box.Brixham, laby=(maxlat +minlat )/2,
797 labx=(maxlong+minlong)/2)
797 box.Thames <- data.frame(maxlat = 52.5, minlat = 51.5, maxlong = 3, minlong = 1,
798 id="(d)")
798 box.Thames <- transform(box.Thames, laby=(maxlat +minlat )/2,
799 labx=(maxlong+minlong)/2)
800 ```
801
802 ```{r Fig7PlotToFile}
802 ppi=300
803 Figure.width = 16/2.54
804 Figure.height = 16/2.54
805 #
806 autoplot(map.ices, geom="contour", colour="gray", size=0.1) +
807 scale_x_continuous(limits = c(ices.west, ices.east), breaks=ewbrks, labels=ewlbls,
808 expand = c(0, 0)) +
809 scale_y_continuous(limits = c(ices.south, ices.north), breaks=nsbrks, labels=nslbls,
810 expand = c(0, 0)) +
811 coord_map(xlim=c(ices.west,ices.east),ylim=c(ices.south,ices.north)) +
812 theme(panel.background = element_rect(fill = "lightsteelblue1")) +
813 theme(legend.position = "none") +
814 geom_point(aes(x=Long,y=Lat), data = swt.lpoolBay, col=PointColour, size=0.1,
815 alpha = 1/10) +
816 geom_point(aes(x=Long,y=Lat), data = swt.celticSea, col=PointColour, size=0.1,
817 alpha = 1/10) +
818 geom_point(aes(x=Long,y=Lat), data = swt.brixham, col=PointColour, size=0.1, alpha
819 = 1/10) +
820 geom_point(aes(x=Long,y=Lat), data = swt.thames, col=PointColour, size=0.1, alpha
821 = 1/10) +
822 labs(x="Longitude", y="Latitude") +
823 theme(axis.title = element_text(size = 12)) +
824 theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12)) +
825 theme(axis.text = element_text(size=12)) +

```

```

820 scale_x_continuous(breaks = ewbrks, labels = ewlbls, expand = c(0, 0)) +
821 scale_y_continuous(breaks = nsbrks, labels = nslbls, expand = c(0, 0)) +
822 geom_rect(data=box.Lpool, aes(xmin=minlong, xmax=maxlong, ymin=minlat, ymax=maxlat
), color="black", fill="transparent", inherit.aes=FALSE) +
823 geom_text(data=box.Lpool, aes(x=labx, y=laby), color="black",label=" ") +
824 geom_rect(data=box.Celtic, aes(xmin=minlong, xmax=maxlong, ymin=minlat,
ymax=maxlat ), color="black", fill="transparent", inherit.aes=FALSE) +
825 geom_text(data=box.Celtic, aes(x=labx, y=laby), color="black",label=" ") +
826 geom_rect(data=box.Brixham, aes(xmin=minlong, xmax=maxlong, ymin=minlat,
ymax=maxlat ), color="black", fill="transparent", inherit.aes=FALSE) +
827 geom_text(data=box.Brixham, aes(x=labx, y=laby), color="black",label=" ") +
828 geom_rect(data=box.Thames, aes(xmin=minlong, xmax=maxlong, ymin=minlat,
ymax=maxlat ), color="black", fill="transparent", inherit.aes=FALSE) +
829 geom_text(data=box.Thames, aes(x=labx, y=laby), color="black",label=" ") +
830 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BEL) +
831 geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DEU) +
832 geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=FRA) +
833 geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=GBR) +
834 geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IMN) +
835 geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IRL) +
836 geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LIE) +
837 geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LUX) +
838 geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NLD)
839 ggsave("fig07.png", width=Figure.width, height = Figure.height, dpi=ppi)
840 ````
841
842 # FIGURE 8
843 **Southern North Sea - Data Availability - NEAR SURFACE**
844
845 ## F8 Subset the Data
846
847 ```{r Fig8subsetData}
848 # NEAR SURFACE DATA
849 #
850 SNS<-SWTfinal
851 SNSsurface=lowerdepth.threshold
852 SNS<-subset(SNS,Depth<=SNSsurface)
853 SNS<-subset(SNS,Lat>=SNSbb.south)
854 SNS<-subset(SNS,Lat<=SNSbb.north)
855 SNS<-subset(SNS,Long<=SNSbb.east)
856 SNS<-subset(SNS,Long>=SNSbb.west)
857 ````
858
859 ## F8 Plot the data
860
861 ```{r Fig8CreateBB}
862 box.SNS <- data.frame(maxlat = SNSbb.north, minlat = SNSbb.south, maxlong =
SNSbb.east, minlong = SNSbb.west, id=" ")
863 box.SNS <- transform(box.SNS, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
864 ````
865
866 ```{r Fig8CreateLabels}
867 ewbrks <- seq(-1,5,1)
868 nsbrks <- seq(51,54,1)
869 ewlbls <- unlist(lapply(ewbrks, function(x) ifelse(x < 0, paste(x*-1, "°W"),
ifelse(x > 0, paste(x, "°E"),x))))
870 nslbls <- unlist(lapply(nsbrks, function(x) ifelse(x < 0, paste(x*-1, "°S"),
ifelse(x > 0, paste(x, "°N"),x))))
871 ````
872
873 ```{r Fig8PlotToFile}
874 ppi=300
875 Figure.width = 16/2.54
876 Figure.height = 16/2.54
877 #
878 autoplot(map.SNS, coast=FALSE,geom="contour",colour="gray", size=0.1) +
879 scale_x_continuous(limits = c(SNS.west, SNS.east), breaks=ewbrks, labels=ewlbls,
expand = c(0, 0)) +
880 scale_y_continuous(limits = c(SNS.south, SNS.north), breaks=nsbrks, labels=nslbls,
expand = c(0, 0)) +
881 coord_map(xlim=c(SNS.west,SNS.east),ylim=c(SNS.south,SNS.north)) +
882 theme(panel.background = element_rect(fill = "lightsteelblue1")) +
883 theme(legend.position = "none") +

```

```

884 geom_point(aes(x=Long,y=Lat), data = SNS, col=PointColour, size=0.1, alpha = 1/10) +
885 labs(x="Longitude", y="Latitude") +
886 theme(axis.title = element_text(size = 12)) +
887 theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12)) +
888 theme(axis.text = element_text(size=12)) +
889 scale_x_continuous(breaks = ewbrks, labels = ewlbls, expand = c(0, 0)) +
890 scale_y_continuous(breaks = nsbrks, labels = nslbls, expand = c(0, 0)) +
891 geom_rect(data=box.SNS, aes(xmin=minlong, xmax=maxlong, ymin=minlat,
892 ymax=maxlat), color="black", fill="transparent", inherit.aes=FALSE) +
893 geom_text(data=box.SNS, aes(x=labx, y=laby, label=id), color="black") +
894 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BEL) +
895 #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DEU) +
896 geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=FRA) +
897 geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=GBR) +
898 #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IMN) +
899 # geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LUX) +
900 geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NLD)
901 #
902 ggsave("fig08.png", width=Figure.width, height = Figure.height, dpi=ppi)
903 ```
904
905 # FIGURE 9
906 **SNS Temperature .v. Time by key sources**
907
908 ## F9 Subset the data
909 **SEPARATE BY SOURCE - use top 9 by number**
910
911 ```{r Fig9seperateSources}
912 #
913 SNS01<- subset(SNS, Source=="1") #exclude - low number <1000
914 SNS02<- subset(SNS, Source=="2") #exclude - low number <1000
915 SNS03<- subset(SNS, Source=="3")
916 SNS04<- subset(SNS, Source=="4")
917 SNS05<- subset(SNS, Source=="5") #exclude - low number <1000
918 SNS06<- subset(SNS, Source=="6")
919 SNS07<- subset(SNS, Source=="7")
920 SNS08<- subset(SNS, Source=="8")
921 SNS09<- subset(SNS, Source=="9") #exclude - low number <1000
922 SNS10<- subset(SNS, Source=="10") #exclude - low number <1000
923 SNS11<- subset(SNS, Source=="11")
924 SNS12<- subset(SNS, Source=="12") ##exclude - low number <1000
925 SNS13<- subset(SNS, Source=="13") #exclude - low number <1000
926 SNS14<- subset(SNS, Source=="14") #exclude - low number <1000
927 SNS15<- subset(SNS, Source=="15")
928 SNS16<- subset(SNS, Source=="16")
929 SNS17<- subset(SNS, Source=="17")
930 #
931 SNS03$year <- year(SNS03$Time)
932 SNS04$year <- year(SNS04$Time)
933 SNS06$year <- year(SNS06$Time)
934 SNS07$year <- year(SNS07$Time)
935 SNS08$year <- year(SNS08$Time)
936 SNS11$year <- year(SNS11$Time)
937 SNS15$year <- year(SNS15$Time)
938 SNS16$year <- year(SNS16$Time)
939 SNS17$year <- year(SNS17$Time)
940 #
941 ```
942
943 ## F9 Plot the data
944
945 ```{r Fig9plotFile}
946 ppi=300
947 Figure.width = 16/2.54
948 Figure.height = 16/2.54
949 png("fig09.png", type = "cairo", width = Figure.width, height = Figure.height, units
950 = "in", pointsize = 12, res = ppi)
951 #
952 p1 <- ggplot(data = SNS03, aes(x=Time, y=tC)) +
953 geom_point(color="blue",size=0.2) +
954 labs(x="Year", y="Temperature °C") +
955 ggtitle("Source 3") +

```



```

955 # theme(plot.title = element_text(size=5, face = "bold")) +
956 scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
957 theme_bw() +
958 theme(axis.title.x = element_text(face="plain", size=10),
959       axis.text.x = element_text(angle=90, vjust=0.5, size=10))
960 #
961 p2 <- ggplot(data = SNS04, aes(x=Time, y=tC)) +
962 geom_point(color="gray64",size=0.2) +
963 labs(x="Year", y="Temperature °C") +
964 ggtitle("Source 4") +
965 theme(plot.title = element_text(size=5, face = "bold")) +
966 scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
967 theme_bw() +
968 theme(axis.title.x = element_text(face="plain", size=10),
969       axis.text.x = element_text(angle=90, vjust=0.5, size=10))
970 #
971 p3 <- ggplot(data = SNS06, aes(x=Time, y=tC)) +
972 geom_point(color="orange",size=0.1) +
973 labs(x="Year", y="Temperature °C") +
974 ggtitle("Source 6") +
975 theme(plot.title = element_text(size=5, face = "bold")) +
976 scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
977 theme_bw() +
978 theme(axis.title.x = element_text(face="plain", size=10),
979       axis.text.x = element_text(angle=90, vjust=0.5, size=10))
980 #
981 p4 <- ggplot(data = SNS07, aes(x=Time, y=tC)) +
982 geom_point(color="orange",size=0.1) +
983 labs(x="Year", y="Temperature °C") +
984 ggtitle("Source 7") +
985 theme(plot.title = element_text(size=5, face = "bold")) +
986 scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
987 theme_bw() +
988 theme(axis.title.x = element_text(face="plain", size=10),
989       axis.text.x = element_text(angle=90, vjust=0.5, size=10))
990 #
991 p5 <- ggplot(data = SNS08, aes(x=Time, y=tC)) +
992 geom_point(color=PointColour,size=0.1) +
993 labs(x="Year", y="Temperature °C") +
994 ggtitle("Source 8") +
995 theme(plot.title = element_text(size=5, face = "bold")) +
996 scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
997 theme_bw() +
998 theme(axis.title.x = element_text(face="plain", size=10),
999       axis.text.x = element_text(angle=90, vjust=0.5, size=10))
1000 #
1001 p6 <- ggplot(data = SNS11, aes(x=Time, y=tC)) +
1002 geom_point(color="magenta",size=0.2) +
1003 labs(x="Year", y="Temperature °C") +
1004 ggtitle("Source 11") +
1005 theme(plot.title = element_text(size=5, face = "bold")) +
1006 scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
1007 theme_bw() +
1008 theme(axis.title.x = element_text(face="plain", size=10),
1009       axis.text.x = element_text(angle=90, vjust=0.5, size=10))
1010 #
1011 p7 <- ggplot(data = SNS15, aes(x=Time, y=tC)) +
1012 geom_point(color="orange",size=0.1) +
1013 labs(x="Year", y="Temperature °C") +
1014 ggtitle("Source 15") +
1015 theme(plot.title = element_text(size=5, face = "bold")) +
1016 scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
1017 theme_bw() +
1018 theme(axis.title.x = element_text(face="plain", size=10),
1019       axis.text.x = element_text(angle=90, vjust=0.5, size=10))
1020 #
1021 p8 <- ggplot(data = SNS16, aes(x=Time, y=tC)) +
1022 geom_point(color="green",size=0.2) +
1023 labs(x="Year", y="Temperature °C") +
1024 ggtitle("Source 16") +
1025 theme(plot.title = element_text(size=5, face = "bold")) +
1026 scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
1027 theme_bw() +

```

```

1028     theme(axis.title.x = element_text(face="plain", size=10),
1029           axis.text.x = element_text(angle=90, vjust=0.5, size=10))
1030 #
1031 p9 <- ggplot(data = SNS17, aes(x=Time, y=tC)) +
1032   geom_point(color="turquoise3",size=0.2) +
1033   labs(x="Year", y="Temperature °C") +
1034   ggtitle("Source 17") +
1035   theme(plot.title = element_text(size=5, face = "bold")) +
1036   scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
1037   theme_bw() +
1038   theme(axis.title.x = element_text(face="plain", size=10),
1039         axis.text.x = element_text(angle=90, vjust=0.5, size=10))
1040 #
1041 multiplot(p1,p4,p7,p2,p5,p8,p3,p6,p9, cols = 3)
1042 #
1043 dev.off()
1044 ```
1045
1046 # Figure 10
1047 **SNS - Temperature .v. Date - All Surface Data - coloured by Source**
1048
1049 ## F10 Subset the data
1050
1051 red > 100,000 points (Sources 6 & 7 - SmartBuoy & WaveNet)
1052 blue 30,000 - 50,000 (8 & 15)
1053 green 2,000 - 6,000 (3,4,11,16,17)
1054
1055 ## F10 Plot the data
1056
1057 ```{r Fig10plotFile}
1058 ppi=300
1059 Figure.width = 16/2.54
1060 Figure.height = 12/2.54
1061 png("fig10.png", type = "cairo", width = Figure.width, height = Figure.height, units
1062     = "in", pointsize = 12, res = ppi)
1063
1064 p1 <- ggplot(data = SNS, aes(x=Time)) +
1065   geom_histogram() +
1066   scale_y_continuous(labels = fancy_scientific) +
1067   labs(x="Year", y="Count") +
1068   ggtitle("(a)") +
1069   theme_bw()
1070 #
1071 p2 <- ggplot(data = SNS03, aes(x=Time, y=tC)) +
1072   geom_point(data=SNS03, color="blue",size=0.1) +
1073   geom_point(data=SNS04, color="gray64",size=0.1) +
1074   geom_point(data=SNS08, color=PointColour,size=0.1) +
1075   geom_point(data=SNS11, color="magenta",size=0.1) +
1076   geom_point(data=SNS15, color="orange",size=0.1) +
1077   geom_point(data=SNS16, color="green",size=0.1) +
1078   geom_point(data=SNS17, color="turquoise3",size=0.1) +
1079   geom_point(data=SNS06, color="orange",size=0.1, alpha = .5) +
1080   geom_point(data=SNS07, color="orange",size=0.1, alpha = .5) +
1081   labs(x="Year", y="Temperature °C") +
1082   ggtitle("(b)") +
1083   theme(plot.title = element_text(size=10, face = "bold")) +
1084   scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
1085   theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1086   theme_bw()
1087 #
1088 multiplot(p1,p2, cols = 1)
1089 dev.off()
1090 ```
1091
1092 The question arising from Fig 10 (b) is *"there looks to be a change in the minimum
1093 values after 2000, with no values < ~ 5C. Is this an artefact of data source?"*. The
1094 following few figures explore this.
1095
1096 # FIGURE 11
1097 **SNS after 2000 - Distribution of data for Surface 0-5m and 0-1m & E/W -
1098 point/vessel**
1099
1100 This explores potential bias in the data related to numbers, depth and location.

```

0-1m is chosen to "isolate" the fixed mooring point, high data number, sources 6 & 7 (SmartBuoy & WaveNet).

```
1097
1098 ## F11 Subset the data
1099
1100 ```{r Fig11subsetData}
1101 #
1102 SNS2000 <- subset(SNS, year(Time) >= 2000)
1103 #
1104 SNS2000.1 <- subset(SNS2000, Depth <= interimdepth.threshold)
1105 SNS2000.5 <- subset(SNS2000, Depth > interimdepth.threshold & Depth <=
lowerdepth.threshold)
1106 #
1107 SBuoyWNet <- subset(SNS2000.1, Source == 6 | Source == 7) # OR
1108 ```
1109
1110 ### F11 Calculate Percentages
1111
1112 ```{r SNSbias}
1113 #
1114 PercentPost2000 <- 100-(NROW(SNS)-NROW(SNS2000))*100/NROW(SNS)
1115 #
1116 PercentPost2000.1 <- 100-(NROW(SNS2000)-NROW(SNS2000.1))*100/NROW(SNS2000)
1117 #
1118 specify_decimal <- function(x, k) format(round(x, k), nsmall=k)
1119 #
1120 cat((paste0("Percentage of all SNS data in SNS Surface (0-5m), post 2000, is ",
specify_decimal(PercentPost2000,1))))); cat("\n"); cat((paste0("Percentage of the
0-5m, post 2000 SNS data that is in the 0-1m range [encompassing autonomous buoys
but excluding vessel mounted pump systems] is ",
specify_decimal(PercentPost2000.1,1))))
1121 #
1122 ```
1123
1124 ## F11 Plot the data
1125
1126 ```{r Fig11CreateBoxes}
1127 box.SNSa <- data.frame(maxlat = SNSbb.north, minlat = SNSbb.south, maxlong =
SNSbb.east, minlong = SNSbb.west, id="a")
1128 box.SNSa <- transform(box.SNSa, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
1129 #
1130 box.SNSb <- data.frame(maxlat = SNSbb.north, minlat = SNSbb.south, maxlong =
SNSbb.east, minlong = SNSbb.west, id="b")
1131 box.SNSb <- transform(box.SNSb, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
1132 ```
1133
1134 ```{r Fig11CreateLabels}
1135 ewbrks <- seq(0,5,1)
1136 nsbrks <- seq(51,54,1)
1137 ewlbls <- unlist(lapply(ewbrks, function(x) ifelse(x < 0, paste(x*-1, "°W"),
ifelse(x > 0, paste(x, "°E"),x))))
1138 nslbls <- unlist(lapply(nsbrks, function(x) ifelse(x < 0, paste(x*-1, "°S"),
ifelse(x > 0, paste(x, "°N"),x))))
1139 ```
1140
1141 ```{r Fig11PlotToFile}
1142 ppi=300
1143 Figure.width = 16/2.54
1144 Figure.height = 8/2.54
1145 #
1146 png("fig11.png", type = "cairo", width = Figure.width, height = Figure.height, units
= "in", pointsize = 12, res = ppi)
1147 #
1148 p1 <- autoplot(map.SNS, coast=FALSE,geom=("contour"),colour="gray", size=0.01) +
1149   theme(legend.position = "none") +
1150   geom_point(aes(x=Long,y=Lat), data = SNS2000.5, col=PointColour, size=0.1, alpha =
1/10) +
1151   coord_map(xlim=c(SNS.west,SNS.east),ylim=c(SNS.south,SNS.north)) +
1152   labs(x="Longitude", y="Latitude") +
1153   theme(axis.title = element_text(size = 12)) +
1154   theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12)) +
1155   theme(axis.text = element_text(size=12)) +
1156   scale_x_continuous(breaks = ewbrks, labels = ewlbls, expand = c(0, 0)) +
```

```

1157 scale_y_continuous(breaks = nsbrks, labels = nslbls, expand = c(0, 0)) +
1158   geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BEL) +
1159   #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DEU) +
1160   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=FRA) +
1161   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=GBR) +
1162   #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IMN) +
1163   #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LIE) +
1164   # geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LUX) +
1165   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NLD) +
1166   geom_rect(data=box.SNSa, aes(xmin=minlong, xmax=maxlong, ymin=minlat,
1167     ymax=maxlat ), color="black", fill="transparent", inherit.aes=FALSE) +
1167   geom_text(data=box.SNSa, aes(x=labx, y=laby, label=id), color="black")+
1168   theme(panel.background = element_rect(fill = "lightsteelblue1"))
1169 p2 <- autoplot(map.SNS, coast=FALSE, geom=("contour"),colour="gray", size=0.01) +
1170   #scale_fill_etopo() +
1171   theme(legend.position = "none") +
1172   coord_map(xlim=c(SNS.west,SNS.east),ylim=c(SNS.south,SNS.north)) +
1173   geom_point(aes(x=Long, y=Lat), data = SNS2000.1, col=PointColour, size=0.1, alpha
1174     = 1/10) +
1174   geom_point(aes(x=Long, y=Lat), data = SBuoyWNet, shape=18, col = "orange", size =
1175     2) +
1175   labs(x="Longitude", y="Latitude") +
1176   theme(axis.title = element_text(size = 12)) +
1177   theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12)) +
1178   theme(axis.text = element_text(size=12)) +
1179   scale_x_continuous(breaks = ewbrks, labels = ewlbls, expand = c(0, 0)) +
1180   scale_y_continuous(breaks = nsbrks, labels = nslbls, expand = c(0, 0)) +
1181   geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BEL) +
1182   #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DEU) +
1183   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=FRA) +
1184   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=GBR) +
1185   #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IMN) +
1186   #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LIE) +
1187   # geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LUX) +
1188   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NLD) +
1189   geom_rect(data=box.SNSb, aes(xmin=minlong, xmax=maxlong, ymin=minlat,
1190     ymax=maxlat ), color="black", fill="transparent", inherit.aes=FALSE) +
1190   geom_text(data=box.SNSb, aes(x=labx, y=laby, label=id), color="black") +
1191   theme(panel.background = element_rect(fill = "lightsteelblue1"))
1192 # ggsave and multiplot issues so
1193 #
1194 multiplot(p1,p2, cols = 2)
1195 dev.off()
1196 #
1197 ```
1198
1199 This illustrates a numerical and a geographic bias for the point sources, south and
1200 west.
1201 ```{r CalcPercents}
1202 specify_decimal <- function(x, k) format(round(x, k), nsmall=k)
1203 Percent.Moored <- 100-(NROW(SNS2000.1)-NROW(SBuoyWNet))*100/NROW(SNS2000.1)
1204 print(paste0("Percentage of point source data (Sources 6 & 7) in SNS Surface (0-1m),
1205   post 2000 is ", specify_decimal(Percent.Moored,1)))
1206 Percent.Moored <- 100-(NROW(SNS2000.1)-NROW(SBuoyWNet))*100/NROW(SNS2000.5)
1207 print(paste0("Percentage of point source data (Sources 6 & 7) in SNS Surface (0-5m),
1208   post 2000 is ", specify_decimal(Percent.Moored,1)))
1209 ```
1210
1211 # FIGURE 12
1212 **SNS post 2000 - Surface (0-1m) Temperature .v. Time for non-point and point data**
1213
1214 ## F12 Subset the data
1215
1216 ```{r Fig12subsetData}
1217 NOTSBuoyWNet <- subset(SNS2000.1, Source != 6 & Source != 7) # NOT
1218
1219 print(NROW(NOTSBuoyWNet))
1220 #
1221 print(NROW(SBuoyWNet))
1222 ```
1223
1224 ## F12 Plot the data

```

```

1223
1224 ```{r Fig12plotFile}
1225 ppi=300
1226 Figure.width = 16/2.54
1227 Figure.height = 8/2.54
1228 png("fig12.png", type = "cairo", width = Figure.width, height = Figure.height, units
= "in", pointsize = 12, res = ppi)
1229 #
1230 p1 <- ggplot(data = NOTSBuoynet, aes(x=Time, y=tC)) +
1231   geom_point(color=PointColour,size=0.05) +
1232   labs(x="Year", y="Temperature °C") +
1233   ggtitle("(a)") +
1234   theme(plot.title = element_text(size=10, face = "bold")) +
1235   scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
1236   theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1237   theme_bw()
1238 p2 <- ggplot(data = SBuoynet, aes(x=Time, y=tC)) +
1239   geom_point(color="orange",size=0.05) +
1240   labs(x="Year", y="Temperature °C") +
1241   ggtitle("(b)") +
1242   theme(plot.title = element_text(size=10, face = "bold")) +
1243   scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
1244   theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1245   theme_bw()
1246 #
1247 multiplot(p1,p2, cols=2)
1248 #
1249 dev.off()
1250 ```
1251
1252 # FIGURE 13
1253 **Plot of min, mean & max for SNS surface data (0-5m)**
1254
1255 Having established that there don't seem to be any location, number or depth effects
in the SNS surface data subset, this section looks at a plot of the Min, Mean and
Max for the data to see if this sheds light on the plot in Fig 9(b).
1256
1257 ## F13 Create the statistics data
1258
1259 ```{r SNSstats}
1260 # stats by year
1261 # create a new Variable called Year
1262 SNS$Year <- year(SNS$Time)
1263 #
1264 SNS1925on <- subset(SNS, Year >= 1925 & Year <= 2015)
1265 #
1266 # use Data Table for easy extraction of descriptive statistics
1267 # recreate a simple data frame of year and tC
1268 Year.tC <- data.frame(SNS1925on$Year, SNS1925on$tC)
1269 setnames(Year.tC, old=c("SNS1925on.Year", "SNS1925on.tC"),
new=c("Year", "Temperature"))
1270 #
1271 # now convert to data.table for quick summary stats
1272 #
1273 Simple.Data <- data.table(Year.tC)
1274 tC.stats <- Simple.Data[,list(mean=mean(Temperature),
1275                               min=min(Temperature),
1276                               max=max(Temperature)),
1277                          by='Year']
1278
1279 # revert to data frames to keep plotting consistent and reduce data.table plot
learning curve to 0
1280 #
1281 #e.g. tC.stats[, matplot(SNS.sub.Year, mean, type="p", ylab="Temperature stats",
xlab="Year"), .SDcols = !"SNS.sub.Year"]
1282 #
1283 class(as.data.frame(tC.stats))
1284 #
1285 # generate a subset for lines as overplot
1286 #
1287 line.data <- subset(tC.stats, Year >= 1955)
1288 line.data <- line.data[order(line.data$Year),]
1289 #

```

```

1290   ```
1291
1292   ## F13 Plot the data
1293
1294   ```{r Fig13plotFile}
1295   #
1296   ppi=300
1297   Figure.width = 16/2.54
1298   Figure.height = 16/2.54 # height increased to show more tC detail
1299   png("fig13.png", type = "cairo", width = Figure.width, height = Figure.height, units
1300   = "in", pointsize = 12, res = ppi)
1301   #
1302   ggplot(data = tC.stats, aes(x=Year, y=mean)) +
1303     geom_point(color="green") +
1304     geom_line(data=line.data, aes(x=Year, y=mean), colour = "green") +
1305     geom_point(data = tC.stats, aes(x=Year, y=max),color=PointColour) +
1306     geom_line(data= line.data, aes(x=Year, y=max), color=PointColour) +
1307     geom_point(data = tC.stats, aes(x=Year, y=min),color="blue") +
1308     geom_line(data= line.data, aes(x=Year, y=min), color="blue") +
1309     labs(x="Year", y="Annual Average Temperature °C") +
1310     scale_y_continuous(breaks = seq(0,20,5)) +
1311     theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1312     theme_bw()
1313   #
1314   ```
1315
1316   # FIGURE 14
1317   **3d Plot of Data Sources in SNS**
1318
1319   A selected smaller 'belt' to illustrate multiple sources and how they are
1320   represented in a depth profile. SNS sub-area & small to make plot look clearer - ALL
1321   depths.
1322
1323   ## F14 Subset the data
1324
1325   ```{r CreateData3dPlotSNSbelt}
1326   belt.north = 54.8
1327   belt.south = 54.3
1328   belt.east = 3.28
1329   belt.west = 3.24
1330   #
1331   ##Subset Belt data
1332   NSdata <- subset(SWTfinal, Lat < belt.north)
1333   NSdata <- subset(NSdata, Lat > belt.south)
1334   NSdata <- subset(NSdata, Long < belt.east)
1335   NSdata <- subset(NSdata, Long > belt.west)
1336   summary(NSdata)
1337   ```
1338
1339   ### F14 Create columns indicating point color, shape and size
1340
1341   ```{r 3dPlotCreateColourColumns}
1342   NSdata$pcolour[NSdata$Source==3] <- "blue"
1343   NSdata$pcolour[NSdata$Source==4] <- "gray64"
1344   NSdata$pcolour[NSdata$Source==8] <- PointColour
1345   NSdata$pcolour[NSdata$Source==9] <- "black"
1346   NSdata$pcolour[NSdata$Source==11] <- "magenta"
1347   NSdata$pcolour[NSdata$Source==15] <- "orange"
1348   NSdata$pcolour[NSdata$Source==16] <- "green"
1349   NSdata$pcolour[NSdata$Source==17] <- "turquoise3"
1350   #
1351   NSdata$ppch[NSdata$Source==3] <- 20
1352   NSdata$ppch[NSdata$Source==4] <- 20
1353   NSdata$ppch[NSdata$Source==8] <- 20
1354   NSdata$ppch[NSdata$Source==9] <- 8
1355   NSdata$ppch[NSdata$Source==11] <- 17
1356   NSdata$ppch[NSdata$Source==15] <- 20
1357   NSdata$ppch[NSdata$Source==16] <- 20
1358   NSdata$ppch[NSdata$Source==17] <- 20
1359   #
1360   NSdata$pcex[NSdata$Source==3] <- 0.1
1361   NSdata$pcex[NSdata$Source==4] <- 0.1

```

```

1360 NSdata$pcex[NSdata$Source==8] <- 0.1
1361 NSdata$pcex[NSdata$Source==9] <- 1
1362 NSdata$pcex[NSdata$Source==11] <- 0.5
1363 NSdata$pcex[NSdata$Source==15] <- 0.1
1364 NSdata$pcex[NSdata$Source==16] <- 0.1
1365 NSdata$pcex[NSdata$Source==17] <- 0.1
1366 ```
1367
1368 ## F14 Plot the data
1369
1370 ```{r Fig143dScatterplotSNSbelt}
1371 pts <-
1372   data.frame(x = NSdata$Long,
1373             y = NSdata$Lat,
1374             z = NSdata$Depth* -1)
1375 # pts$ppch <- as.numeric(pts$ppch) # needed as was 'chr' in development stages
1376 pts$pcolour <- NSdata$pcolour
1377 pts$ppch <- NSdata$ppch
1378 pts$pcex <- NSdata$pcex
1379 #
1380 ppi=300
1381 Figure.width = 16/2.54
1382 Figure.height = 10/2.54 # height increased to show more tC detail
1383 png("fig14.png", type = "cairo", width = Figure.width, height = Figure.height, units
1384     = "in", pointsize = 12, res = ppi)
1385 #
1386 with(pts, {
1387   s3d <- scatterplot3d(pts$x,pts$y,pts$z,
1388                       color=pts$pcolour, pch=pts$ppch, cex.symbols=pts$pcex, type="p",
1389                       scale.y=1,
1390                       scale.x=1,
1391                       cex.lab=0.75,
1392                       cex.axis=0.5,
1393                       x.ticklabs = c("3.24°E","3.25°E","3.26°E","3.27°E","3.28°E","3.29°E"),
1394                       xlab="Longitude",
1395                       y.ticklabs = c("54.3°N","54.4°N","54.5°N","54.6°N","54.7°N"," "),
1396                       ylab="Latitude",
1397                       z.ticklabs = c(50,40,30,20,10,0),
1398                       zlab="Depth (m)")
1399   #add legend
1400   legend("topright", bty="y",cex=0.5,"Data by Source", c("Source =
1401     3","Source = 4","Source = 8","Source = 9","Source = 11","Source =
1402     15","Source = 16","Source = 17"), fill = c("blue","gray64",
1403     PointColour, "black", "magenta", "orange", "green", "turquoise3"))
1404 })
1405 dev.off()
1406 ```
1407
1408 # FIGURE 15
1409 **UKCS map and Data Availability - Surface and Mid water**
1410
1411 ## F15 Subset the data
1412
1413 ```{r Fig15data}
1414 #restrict to UKCS
1415 ukcs <- subset(SWTfinal, Lat < ukcs.north)
1416 ukcs <- subset(ukcs, Lat > ukcs.south)
1417 ukcs <- subset(ukcs, Long < ukcs.east)
1418 ukcs <- subset(ukcs, Long > ukcs.west)
1419 #
1420 # restrict by depth
1421 Dupperdepth.threshold=20
1422 Dlowerdepth.threshold=25
1423 #
1424 ukcs.surface <- subset(ukcs,Depth <=lowerdepth.threshold)
1425 ukcs.mid <- subset(ukcs,Depth >=Dupperdepth.threshold & Depth <=Dlowerdepth.threshold)
1426 ```
1427
1428 ## F15 Plot the data
1429
1430 ```{r Fig15CreateLabels}
1431 ewbrks <- seq(-15,10,5)
1432 nsbrks <- seq(48,63,2)

```

```

1429 ewlbls <- unlist(lapply(ewbrks, function(x) ifelse(x < 0, paste(x*-1, "°W"),
1430 ifelse(x > 0, paste(x, "°E"),x)))
1431
1432
1433 ```{r Fig15plotFile}
1434 ppi=300
1435 Figure.width = 16/2.54
1436 Figure.height = 16/2.54
1437 #
1438 autoplot(map.ukcs, geom="contour",colour="gray", size=0.1) +
1439   scale_x_continuous(limits = c(ukcs.west, ukcs.east), breaks=ewbrks, labels=ewlbls,
1440     expand = c(0, 0)) +
1441   scale_y_continuous(limits = c(ukcs.south, ukcs.north), breaks=nsbrks, labels=nslbls,
1442     expand = c(0, 0)) +
1443   coord_map(xlim=c(ukcs.west,ukcs.east),ylim=c(ukcs.south,ukcs.north)) +
1444   theme(legend.position = "none") +
1445   theme(panel.background = element_rect(fill = "lightsteelblue1")) +
1446   geom_point(aes(x=Long,y=Lat), data = ukcs.surface, col=PointColour, size=0.1,
1447     alpha = 1/10) +
1448   geom_point(aes(x=Long,y=Lat), data = ukcs.mid, col="blue", size=0.1, alpha = 1/10) +
1449   labs(x="Longitude", y="Latitude") +
1450   theme(axis.title = element_text(size = 12)) +
1451   theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12)) +
1452   theme(axis.text = element_text(size=12)) +
1453   scale_x_continuous(breaks = ewbrks, labels = ewlbls, expand = c(0, 0)) +
1454   scale_y_continuous(breaks = nsbrks, labels = nslbls, expand = c(0, 0)) +
1455   geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BEL) +
1456   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=CHE) +
1457   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DNK) +
1458   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DEU) +
1459   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=FRA) +
1460   geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=FRO) +
1461   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=GBR) +
1462   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IMN) +
1463   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IRL) +
1464   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LIE) +
1465   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LUX) +
1466   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NLD) +
1467   geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NOR)
1468 #
1469 ggsave("fig15.png", width=Figure.width, height = Figure.height, dpi=ppi)
1470 ```
1471
1472 # FIGURE 16
1473 **UKCS Surface and Mid Water Temperature by 14 YEAR periods**
1474
1475 Mid Water data extends from 1970 so that is the time range selected for comparison
1476 of surface and mid water data
1477
1478 ## F16 Subset the data
1479
1480 ```{r Fig16data}
1481 ukcs.surface.7084 <- subset(ukcs.surface, Time >= "1970-01-01 00:00:00" & Time <=
1482 "1984-12-31 23:59:59")
1483 ukcs.surface.8599 <- subset(ukcs.surface, Time >= "1985-01-01 00:00:00" & Time <=
1484 "1999-12-31 23:59:59")
1485 ukcs.surface.0015 <- subset(ukcs.surface, Time >= "2000-01-01 00:00:00" & Time <=
1486 "2015-12-31 23:59:59")
1487 #
1488 ukcs.mid.7084 <- subset(ukcs.mid, Time >= "1970-01-01 00:00:00" & Time <=
1489 "1984-12-31 23:59:59")
1490 ukcs.mid.8599 <- subset(ukcs.mid, Time >= "1985-01-01 00:00:00" & Time <=
1491 "1999-12-31 23:59:59")
1492 ukcs.mid.0015 <- subset(ukcs.mid, Time >= "2000-01-01 00:00:00" & Time <=
1493 "2015-12-31 23:59:59")
1494 #
1495 ```
1496
1497 ## F16 Plot the data
1498
1499 ```{r Fig16plotFile}

```



```

1490 ppi=300
1491 Figure.width = 16/2.54
1492 Figure.height = 16/2.54
1493 png("fig16.png", type = "cairo", width = Figure.width, height = Figure.height, units
= "in", pointsize = 12, res = ppi)
1494 #
1495 p1 <- ggplot(data = ukcs.surface.7084, aes(x=Time, y=tC)) +
1496   geom_point(color=PointColour,size=0.1) +
1497   geom_point(data=ukcs.mid.7084, aes(x=Time, y=tC), colour = "blue",size=0.5)+
1498   labs(x="Year", y="Temperature °C") +
1499   ggtitle("(a)") +
1500   theme(plot.title = element_text(size=10, face = "bold")) +
1501   scale_y_continuous(limits = c(-2, 32),breaks = seq(0,25,5)) +
1502   theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1503   theme_bw()
1504 #
1505 p2 <- ggplot(data = ukcs.surface.8599, aes(x=Time, y=tC)) +
1506   geom_point(color=PointColour,size=0.1) +
1507   geom_point(data=ukcs.mid.8599, aes(x=Time, y=tC), colour = "blue",size=0.5)+
1508   labs(x="Year", y="Temperature °C") +
1509   ggtitle("(b)") +
1510   theme(plot.title = element_text(size=10, face = "bold")) +
1511   scale_y_continuous(limits = c(-2, 32),breaks = seq(0,25,5)) +
1512   theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1513   theme_bw()
1514 #
1515 p3 <- ggplot(data = ukcs.surface.0015, aes(x=Time, y=tC)) +
1516   geom_point(color=PointColour,size=0.1) +
1517   geom_point(data=ukcs.mid.0015, aes(x=Time, y=tC), colour = "blue",size=0.5)+
1518   labs(x="Year", y="Temperature °C") +
1519   ggtitle("(c)") +
1520   theme(plot.title = element_text(size=10, face = "bold")) +
1521   scale_y_continuous(limits = c(-2, 32),breaks = seq(0,25,5)) +
1522   theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1523   theme_bw()
1524 #
1525 multiplot(p1,p2,p3, cols=1)
1526 dev.off()
1527 ```
1528
1529 # FIGURE 17
1530 **UKCS Monthly Average Temperature .v. Year - SURFACE**
1531
1532 ## F17 Calculate Monthly averages
1533
1534 ```{r Fig17data}
1535 #
1536 #http://stackoverflow.com/questions/6052631/aggregate-daily-data-to-month-year-in-r
1537 #
1538 # SURFACE
1539 #
1540 Mon.avg <- setDT(ukcs.surface)[, .(MonthlyMeans = mean(tC)), by = .(year(Time),
month(Time))]
1541 #
1542 # to see counts for means
1543 # Mon.count <- setDT(ukcs.surface)[, .(COUNT = .N), by = .(year(Time), month(Time))]
1544 #
1545 #convert to dates
1546 Mon.avg$year <- as.character(Mon.avg$year)
1547 Mon.avg$month <- as.character(Mon.avg$month)
1548 #combine chr
1549 Mon.avg$YearMon <- paste(Mon.avg$year, Mon.avg$month, sep = "-0")
1550 # replace 010 011 and 012 with 10 11 12
1551 Mon.avg$YearMon <- gsub('-010', '-10',Mon.avg$YearMon)
1552 Mon.avg$YearMon <- gsub('-011', '-11',Mon.avg$YearMon)
1553 Mon.avg$YearMon <- gsub('-012', '-12',Mon.avg$YearMon)
1554 #convert to date - add day 01
1555 Mon.avg$YearMon <- as.Date(paste(Mon.avg$YearMon,"-01", sep=""))
1556 #
1557 Mon.avg.SURFACE <- Mon.avg
1558 #
1559 # MID WATER
1560 #

```

```

1561 Mon.avg <- setDT(ukcs.mid)[, .(MonthlyMeans = mean(tC)), by = .(year(Time),
1562 # month(Time))]
1563 #
1564 #Mon.count <- setDT(ukcs.mid)[, .(COUNT = .N), by = .(year(Time), month(Time))]
1565 #
1566 #convert to dates
1567 Mon.avg$year <- as.character(Mon.avg$year)
1568 Mon.avg$month <- as.character(Mon.avg$month)
1569 #combine chr
1570 Mon.avg$YearMon <- paste(Mon.avg$year, Mon.avg$month, sep = "-0")
1571 # replace 010 011 and 012 with 10 11 12
1572 Mon.avg$YearMon <- gsub('-010', '-10', Mon.avg$YearMon)
1573 Mon.avg$YearMon <- gsub('-011', '-11', Mon.avg$YearMon)
1574 Mon.avg$YearMon <- gsub('-012', '-12', Mon.avg$YearMon)
1575 #convert to date - add day 01
1576 Mon.avg$YearMon <- as.Date(paste(Mon.avg$YearMon, "-01", sep=""))
1577 #
1578 Mon.avg.MID <- Mon.avg
1579 ``
1580 ## F17 Subset the data
1581
1582 ```{r Fig17subsetData}
1583 JanS.avg <- subset(Mon.avg.SURFACE, month == "1")
1584 JanM.avg <- subset(Mon.avg.MID, month == "1")
1585 JanS.avg$year <- as.numeric(JanS.avg$year)
1586 JanM.avg$year <- as.numeric(JanM.avg$year)
1587 #
1588 FebS.avg <- subset(Mon.avg.SURFACE, month == "2")
1589 FebM.avg <- subset(Mon.avg.MID, month == "2")
1590 FebS.avg$year <- as.numeric(FebS.avg$year)
1591 FebM.avg$year <- as.numeric(FebM.avg$year)
1592 #
1593 MarS.avg <- subset(Mon.avg.SURFACE, month == "3")
1594 MarM.avg <- subset(Mon.avg.MID, month == "3")
1595 MarS.avg$year <- as.numeric(MarS.avg$year)
1596 MarM.avg$year <- as.numeric(MarM.avg$year)
1597 #
1598 AprS.avg <- subset(Mon.avg.SURFACE, month == "4")
1599 AprM.avg <- subset(Mon.avg.MID, month == "4")
1600 AprS.avg$year <- as.numeric(AprS.avg$year)
1601 AprM.avg$year <- as.numeric(AprM.avg$year)
1602 #
1603 MayS.avg <- subset(Mon.avg.SURFACE, month == "5")
1604 MayM.avg <- subset(Mon.avg.MID, month == "5")
1605 MayS.avg$year <- as.numeric(MayS.avg$year)
1606 MayM.avg$year <- as.numeric(MayM.avg$year)
1607 #
1608 JunS.avg <- subset(Mon.avg.SURFACE, month == "6")
1609 JunM.avg <- subset(Mon.avg.MID, month == "6")
1610 JunS.avg$year <- as.numeric(JunS.avg$year)
1611 JunM.avg$year <- as.numeric(JunM.avg$year)
1612 #
1613 JulS.avg <- subset(Mon.avg.SURFACE, month == "7")
1614 JulM.avg <- subset(Mon.avg.MID, month == "7")
1615 JulS.avg$year <- as.numeric(JulS.avg$year)
1616 JulM.avg$year <- as.numeric(JulM.avg$year)
1617 #
1618 AugS.avg <- subset(Mon.avg.SURFACE, month == "8")
1619 AugM.avg <- subset(Mon.avg.MID, month == "8")
1620 AugS.avg$year <- as.numeric(AugS.avg$year)
1621 AugM.avg$year <- as.numeric(AugM.avg$year)
1622 #
1623 SepS.avg <- subset(Mon.avg.SURFACE, month == "9")
1624 SepM.avg <- subset(Mon.avg.MID, month == "9")
1625 SepS.avg$year <- as.numeric(SepS.avg$year)
1626 SepM.avg$year <- as.numeric(SepM.avg$year)
1627 #
1628 OctS.avg <- subset(Mon.avg.SURFACE, month == "10")
1629 OctM.avg <- subset(Mon.avg.MID, month == "10")
1630 OctS.avg$year <- as.numeric(OctS.avg$year)
1631 OctM.avg$year <- as.numeric(OctM.avg$year)
1632 #

```

```

1633 NovS.avg <- subset(Mon.avg.SURFACE,month == "11")
1634 NovM.avg <- subset(Mon.avg.MID,month == "11")
1635 NovS.avg$year <- as.numeric(NovS.avg$year)
1636 NovM.avg$year <- as.numeric(NovM.avg$year)
1637 #
1638 DecS.avg <- subset(Mon.avg.SURFACE,month == "12")
1639 DecM.avg <- subset(Mon.avg.MID,month == "12")
1640 DecS.avg$year <- as.numeric(DecS.avg$year)
1641 DecM.avg$year <- as.numeric(DecM.avg$year)
1642 ```
1643
1644 ## F17 Plot the data
1645
1646 ```{r Fig17plotFile}
1647 # lowess
1648 #https://ww2.coastal.edu/kingw/statistics/R-tutorials/simplelinear.html
1649 #
1650 PointSize = 0.5
1651 ppi=300
1652 Figure.width = 16/2.54
1653 Figure.height = 24/2.54
1654 # size increased to allow for margins - 16/10 gave plot error margins too small
1655 png("fig17.png", type = "cairo", width = Figure.width, height = Figure.height, units
= "in", pointsize = 12, res = ppi)
1656 #
1657 p1 <- ggplot(data = JanS.avg, aes(x=year, y=MonthlyMeans)) +
1658   geom_point(color=PointColour, size = PointSize) +
1659   geom_point(data=JanM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size =
PointSize) +
1660   labs(x="Year", y="Temperature °C") +
1661   ggtitle("Jan") +
1662   theme(plot.title = element_text(size=10, face = "bold")) +
1663   scale_y_continuous(limits = c(0, 20)) +
1664   theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1665   theme_bw()
1666 p2 <- ggplot(data = FebS.avg, aes(x=year, y=MonthlyMeans)) +
1667   geom_point(color=PointColour, size = PointSize) +
1668   geom_point(data=FebM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size =
PointSize) +
1669   labs(x="Year", y="Temperature °C") +
1670   ggtitle("Feb") +
1671   theme(plot.title = element_text(size=10, face = "bold")) +
1672   scale_y_continuous(limits = c(0, 20)) +
1673   theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1674   theme_bw()
1675 p3 <- ggplot(data = MarS.avg, aes(x=year, y=MonthlyMeans)) +
1676   geom_point(color=PointColour, size = PointSize) +
1677   geom_point(data=MarM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size =
PointSize) +
1678   labs(x="Year", y="Temperature °C") +
1679   ggtitle("Mar") +
1680   theme(plot.title = element_text(size=10, face = "bold")) +
1681   scale_y_continuous(limits = c(0, 20)) +
1682   theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1683   theme_bw()
1684 p4 <- ggplot(data = AprS.avg, aes(x=year, y=MonthlyMeans)) +
1685   geom_point(color=PointColour, size = PointSize) +
1686   geom_point(data=AprM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size =
PointSize) +
1687   labs(x="Year", y="Temperature °C") +
1688   ggtitle("Apr") +
1689   theme(plot.title = element_text(size=10, face = "bold")) +
1690   scale_y_continuous(limits = c(0, 20)) +
1691   theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1692   theme_bw()
1693 p5 <- ggplot(data = MayS.avg, aes(x=year, y=MonthlyMeans)) +
1694   geom_point(color=PointColour, size = PointSize) +
1695   geom_point(data=MayM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size =
PointSize) +
1696   labs(x="Year", y="Temperature °C") +
1697   ggtitle("May") +
1698   theme(plot.title = element_text(size=10, face = "bold")) +
1699   scale_y_continuous(limits = c(0, 20)) +

```

```

1700     theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1701     theme_bw()
1702 p6 <- ggplot(data = JunS.avg, aes(x=year, y=MonthlyMeans)) +
1703     geom_point(color=PointColour, size = PointSize) +
1704     geom_point(data=JunM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size =
1705         PointSize) +
1706     labs(x="Year", y="Temperature °C") +
1707     ggtitle("Jun") +
1708     theme(plot.title = element_text(size=10, face = "bold")) +
1709     scale_y_continuous(limits = c(0, 20)) +
1710     theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1711     theme_bw()
1712 p7 <- ggplot(data = JulS.avg, aes(x=year, y=MonthlyMeans)) +
1713     geom_point(color=PointColour, size = PointSize) +
1714     geom_point(data=JulM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size =
1715         PointSize) +
1716     labs(x="Year", y="Temperature °C") +
1717     ggtitle("Jul") +
1718     theme(plot.title = element_text(size=10, face = "bold")) +
1719     scale_y_continuous(limits = c(0, 20)) +
1720     theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1721     theme_bw()
1722 p8 <- ggplot(data = AugS.avg, aes(x=year, y=MonthlyMeans)) +
1723     geom_point(color=PointColour, size = PointSize) +
1724     geom_point(data=AugM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size =
1725         PointSize) +
1726     labs(x="Year", y="Temperature °C") +
1727     ggtitle("Aug") +
1728     theme(plot.title = element_text(size=10, face = "bold")) +
1729     scale_y_continuous(limits = c(0, 20)) +
1730     theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1731     theme_bw()
1732 p9 <- ggplot(data = SepS.avg, aes(x=year, y=MonthlyMeans)) +
1733     geom_point(color=PointColour, size = PointSize) +
1734     geom_point(data=SepM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size =
1735         PointSize) +
1736     labs(x="Year", y="Temperature °C") +
1737     ggtitle("Sep") +
1738     theme(plot.title = element_text(size=10, face = "bold")) +
1739     scale_y_continuous(limits = c(0, 20)) +
1740     theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1741     theme_bw()
1742 p10 <- ggplot(data = OctS.avg, aes(x=year, y=MonthlyMeans)) +
1743     geom_point(color=PointColour, size = PointSize) +
1744     geom_point(data=OctM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size =
1745         PointSize) +
1746     labs(x="Year", y="Temperature °C") +
1747     ggtitle("Oct") +
1748     theme(plot.title = element_text(size=10, face = "bold")) +
1749     scale_y_continuous(limits = c(0, 20)) +
1750     theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1751     theme_bw()
1752 p11 <- ggplot(data = NovS.avg, aes(x=year, y=MonthlyMeans)) +
1753     geom_point(color=PointColour, size = PointSize) +
1754     geom_point(data=NovM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size =
1755         PointSize) +
1756     labs(x="Year", y="Temperature °C") +
1757     ggtitle("Nov") +
1758     theme(plot.title = element_text(size=10, face = "bold")) +
1759     scale_y_continuous(limits = c(0, 20)) +
1760     theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1761     theme_bw()
1762 p12 <- ggplot(data = DecS.avg, aes(x=year, y=MonthlyMeans)) +
1763     geom_point(color=PointColour, size = PointSize) +
1764     geom_point(data=DecM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size =
1765         PointSize) +
1766     labs(x="Year", y="Temperature °C") +
1767     ggtitle("Dec") +
1768     theme(plot.title = element_text(size=10, face = "bold")) +
1769     scale_y_continuous(limits = c(0, 20)) +
1770     theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
1771     theme_bw()
1772 #

```

```
1766 multiplot(p1,p3,p5,p7,p9,p11,p2,p4,p6,p8,p10,p12, cols=2)
1767 dev.off()
1768 ```
1769
1770 # **END**
```