Supplement of Earth Syst. Sci. Data, 10, 27–51, 2018
https://doi.org/10.5194/essd-10-27-2018-supplement

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)

~~~~
---
title:
author: "David Morris"
output:
  word_document:
  toc: true
  toc_depth: 2
  toc_float:
  collapsed: true
  smooth_scroll: false
  number_sections: true
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE, fig.keep = 'all', message=FALSE, warning=FALSE)
```

# BACKGROUND INFORMATION

Title is left blank because the Journal inserts it.

Code style based on https://google.github.io/styleguide/Rguide.xml

## Copyright Statement

This code is subject to Crown copyright protection. The material must be acknowledged as Crown copyright in this manner:

*Crown copyright, 2017*

## Data Availability

Cefas data are available from the Cefas Data Hub - https://www.cefas.co.uk/cefas-data-hub/

The data cited below are available from https://doi.org.10.14466/Cefasdatahub.4. This is a zipped single .csv file that contains all the data from all the Sources. Zipped file name is *CefasSWTdata20170707sourceALL*. This is an 84 MByte file that unzips to a 957 MByte .csv file of the same name.

## Software Used

R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/. R version 3.3.2

RStudio (2016). RStudio: Integrated development environment for R (Version 0.99.903) [Computer software]. Boston, MA. Retrieved September 27, 2016. Available from http://www.rstudio.org/RStudio version 1.0.44

# AUTHOR COMMENT (David Morris)

This file contains the R code used to analyse the Quality Controlled data assembled from 17 diverse and disparate Cefas Data Systems for this Data Paper.

The data sources and an overview of the extensive data assembly processes involved are described in the accompanying publication. In summary, data were obtained from Cefas Data Owners / Stewards as folder/files and extracts from various operational databases and systems.

Data were processed to extract temperature records as described in the manuscript, with over 15,000 source files totalling ~ 15 G Bytes, reducing to some 100+ 'summary' files resulting from the extracts of the many and diverse, bespoke formats and/or Excel/.csv/.txt/.dat, data source files; each designed for the particular requirements of the data system involved.

Data processing included standardising formats, extracting reference fields, a minimum/maximum sense check process to detect system and sensor errors, a geographical sense check plot & Temperature .v. Time plots to examine outliers and data at inappropriate levels for the time of year.

# FILE DESCRIPTION

~~~~

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

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.

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.

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.

SOURCE & LIBRARY STATEMENTS

Local parameters

WINDOWS 10 (16 Gigabyte RAM)

R 3.3.2

RStudio 1.0.44

ACQUIRE THE DATA

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.

Set local parameters and load .Rda file

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).

Local setup

```{r Localsetup}
projectPath="C:/Users/djm00/OneDrive - CEFAS/SWT-MASTER"
# INSERT APPROPRIATE LOCAL PATH
setwd(projectPath)
load("CefasSWTdata20170707.Rda") # n = 10,680,821
SWTfinal <- CefasSWTdata20170707 # minimises my code editing arising from changes in the data following systematic and detailed QC checks using tempreature levels and month to remove anomalies that appeared in the plots as they were developed.
rm(CefasSWTdata20170707) # reduces memory overheads and chances of changes to core file
```

Read from .csv files

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.

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

~~~~
***

This provides an R data frame called *CefasSWTdata20170707* with 10,680,440 observations of 11 variables.

### Create Time as DateTime

***
CefasSWTdata20170707$Time <- as.POSIXct(CefasSWTdata20170707$Time)
summary(CefasSWTdata20170707)
***

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.

# LIBRARIES

Libraries used in the assembly, checking & plotting processes

Maps have been created using the R package *marmap*.

*sessionInfo()* used for version numbers

**Load libraries*

```{r loadLibraries}
library(data.table) # 1.10.4
library(ggmap) # 2.6.1
library(ggplot2) # 2.2.1
library(graphics)
library(gridExtra) # 2.2.1
library(lubridate) # 1.6.0
library(mapproj) # 1.2-4
library(marmap)  # 0.9.6
library(raster) # 2.5-8
library(rgdal) # 1.2-5
library(rgeos) # 0.3-22
library(sp)  # 1.2-3
library(lattice) # 0.20-34
library(scatterplot3d) # 0.3-40
```

Other libraries have been added as the manuscript progressed. These are inserted where they are needed.

# FUNCTION DEFINITIONS

## Multiple Plots

Uses http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/

```{r Multiple Plots FUNCTION}
# Multiple plot function
#
# ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)
# - cols:   Number of columns in layout
# - layout: A matrix specifying the layout. If present, 'cols' is ignored.
#
# If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),
# then plot 1 will go in the upper left, 2 will go in the upper right, and
# 3 will go all the way across the bottom.
#
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
~~~~

````
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }

 if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
```

## Provide Scientific Notation

http://stackoverflow.com/questions/11610377/how-do-i-change-the-formatting-of-numbers-on-an-axis-with-ggplot

```{r SciNote}
fancy_scientific <- function(l) {
     # turn in to character string in scientific notation
     l <- format(l, scientific = TRUE)
     # quote the part before the exponent to keep all the digits
     l <- gsub("^(.*)e", "'\\1'e", l)
     # turn the 'e+' into plotmath format
     l <- gsub("e", "%*%10^", l)
     # return this as an expression
     parse(text=l)
}
```

# LOCATION PARAMETERS

These specify to coordinates for maps and subsets of data; provided as NESW bounding boxes.

## Bounding Boxes for maps

ukcs is a short form for an area representing the UK Continental Shelf

```{r BBoxes4Maps}
ukcs.north= 63
ukcs.east= 10
ukcs.south= 48
ukcs.west= -15
#
ukcsPlus.north= 82
ukcsPlus.east= 54
ukcsPlus.south= 37
ukcsPlus.west= -57
#
ices.north= 55
ices.east= 5
ices.south= 48
ices.west= -10
#
SNS.north = 54
SNS.east=6
SNS.south=51
SNS.west=0
````

````
```

## Bounding Boxes for ICES areas

```{r BBoxesICES}
lpoolBay.north=54
lpoolBay.east=-3
lpoolBay.south=53
lpoolBay.west=-5
#
celticSea.north=51
celticSea.east=-7
celticSea.south=50
celticSea.west=-9
#
brixham.north=50.5
brixham.east=-2
brixham.south=49.5
brixham.west=-4
#
thames.north=52.5
thames.east=3
thames.south=51.5
thames.west=1
```

## Bounding Boxes for the Southern North Sea area

```{r BBoxesSNS}
SNSbb.north=53.5
SNSbb.east=4
SNSbb.south=51.5
SNSbb.west=2
```

## Fix for Swiss GADM not plotting & Greenland/Turkey "tears"

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.

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.

**Define the rectangles to overlay tears in the maps**

```{r CHEfix}
box.CHE <- data.frame(maxlat = 52, minlat = 48, maxlong = 20, minlong = 6, id="a")
box.CHE <- transform(box.CHE, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
#
box.GRN <- data.frame(maxlat = 82, minlat = 72, maxlong = -25, minlong = -57, id="a")
box.GRN <- transform(box.GRN, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
#
box.TUR.1 <- data.frame(maxlat = 41, minlat = 37, maxlong = 45, minlong = 28, id="a")
box.TUR.1 <- transform(box.TUR.1, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
#
box.TUR.2 <- data.frame(maxlat = 41.6, minlat = 41, maxlong = 38, minlong = 32, id="a")
box.TUR.2 <- transform(box.TUR.2, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
#
```

# BASE MAPS (uses marmap)

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.

**Acquire the Base Maps**
````
````

````
```{r BaseMaps}
map.ukcs <- getNOAA.bathy(ukcs.west, ukcs.east, ukcs.south, ukcs.north, res = 1, keep=TRUE)
map.ukcsPlus <- getNOAA.bathy(ukcsPlus.west, ukcsPlus.east, ukcsPlus.south, ukcsPlus.north, res = 4, keep=TRUE)
map.ices <- getNOAA.bathy(ices.west, ices.east, ices.south, ices.north, res = 1, keep=TRUE)
map.SNS<-getNOAA.bathy(SNS.west, SNS.east, SNS.south, SNS.north, res = 1, keep=TRUE)
```
````

All maps have used **marmap** plus darkgreen country overlays; this is consistent for all figures.

NOTE the density plot does not allow a scaled fill for the data AND, e.g. the use of a simple fill for the topography, e.g. "etopo". Issues were also encountered in retaining the bathymetry & with over plots of land which were needed to mask "smear" from point plots and the (clearly impossible) land components of the density plots. The resulting combination of *marmap* and density plots resulted in compromises and quick fixes that exemplify the *it's ugly but it works* issues that arise between professional coders and scientists who code. The option of using the single world map from GADM requires dealing with shape files, another 'extension' in coding that was time constrained and hence not used.

## Download the Country polygons

As indicated above this is a less than elegant way of working around issues that arose from the initial selection of *marmap* and the subsequent requirement for density plots.

**Acquire and "fortify" the Country polygons**

GADM is the spatial database of Global Administrative Areas (http://www.gadm.org/) and is accessed using the *sp* package.

````
```{r CountryPolygons}
# Download relevant country polygons - see http://gadm.org/country UK & R SpatialPolygon - level 0 admin - Ver 2.8
#
#uses library(sp)
#
ALBcoast <- readRDS("ALB_adm0.rds")
ARMcoast <- readRDS("ARM_adm0.rds")
AUTcoast <- readRDS("AUT_adm0.rds")
AZEcoast <- readRDS("AZE_adm0.rds")
BELcoast <- readRDS("BEL_adm0.rds")
BGRcoast <- readRDS("BGR_adm0.rds")
BIHcoast <- readRDS("BIH_adm0.rds")
BLRcoast <- readRDS("BLR_adm0.rds")
CANcoast <- readRDS("CAN_adm0.rds")
CHEcoast <- readRDS("CHE_adm0.rds")
DEUcoast <- readRDS("DEU_adm0.rds")
DNKcoast <- readRDS("DNK_adm0.rds")
ESPcoast <- readRDS("ESP_adm0.rds")
ESTcoast <- readRDS("EST_adm0.rds")
FINcoast <- readRDS("FIN_adm0.rds")
FRAcoast <- readRDS("FRA_adm0.rds")
FROcoast <- readRDS("FRO_adm0.rds")
GBRcoast <- readRDS("GBR_adm0.rds")
GRCcoast <- readRDS("GRC_adm0.rds")
GEOcoast <- readRDS("GEO_adm0.rds")
GRLcoast <- readRDS("GRL_adm0.rds")
HRVcoast <- readRDS("HRV_adm0.rds")
HUNcoast <- readRDS("HUN_adm0.rds")
IMNcoast <- readRDS("IMN_adm0.rds")
IRLcoast <- readRDS("IRL_adm0.rds")
IRNcoast <- readRDS("IRN_adm0.rds")
ISLcoast <- readRDS("ISL_adm0.rds")
ITAcoast <- readRDS("ITA_adm0.rds")
KAZcoast <- readRDS("KAZ_adm0.rds")
LIEcoast <- readRDS("LIE_adm0.rds")
LTUcoast <- readRDS("LTU_adm0.rds")
LUXcoast <- readRDS("LUX_adm0.rds")
````
````

```
LVAcoast <- readRDS("LVA_adm0.rds")
MDAcoast <- readRDS("MDA_adm0.rds")
MKDcoast <- readRDS("MKD_adm0.rds")
MNEcoast <- readRDS("MNE_adm0.rds")
NLDcoast <- readRDS("NLD_adm0.rds")
NORcoast <- readRDS("NOR_adm0.rds")
POLcoast <- readRDS("POL_adm0.rds")
PRTcoast <- readRDS("PRT_adm0.rds")
ROUcoast <- readRDS("ROU_adm0.rds")
RUScoast <- readRDS("RUS_adm0.rds")
SJMcoast <- readRDS("SJM_adm0.rds")
SRBcoast <- readRDS("SRB_adm0.rds")
SVKcoast <- readRDS("SVK_adm0.rds")
SVNcoast <- readRDS("SVN_adm0.rds")
SWEcoast <- readRDS("SWE_adm0.rds")
TKMcoast <- readRDS("TKM_adm0.rds")
TURcoast <- readRDS("TUR_adm0.rds")
UKRcoast <- readRDS("UKR_adm0.rds")
XKOcoast <- readRDS("XKO_adm0.rds")
#
# proj4string(GBRcoast) # confirms WGS84
# GBRmerc <- spTransform(GBRcoast, CRS("+init=epsg:3395")) # not needed
#
ALB <- fortify(ALBcoast,id="id")
ARM <- fortify(ARMcoast,id="id")
AUT <- fortify(AUTcoast,id="id")
AZE <- fortify(AZEcoast,id="id")
BEL <- fortify(BELcoast,id="id")
BGR <- fortify(BGRcoast,id="id")
BIH <- fortify(BIHcoast,id="id")
BLR <- fortify(BLRcoast,id="id")
CAN <- fortify(CANcoast,id="id")
CHE <- fortify(CHEcoast,id="id")
DEU <- fortify(DEUcoast,id="id")
DNK <- fortify(DNKcoast,id="id")
ESP <- fortify(ESPcoast,id="id")
EST <- fortify(ESTcoast,id="id")
FIN <- fortify(FINcoast,id="id")
FRA <- fortify(FRAcoast,id="id")
FRO <- fortify(FROcoast,id="id")
GBR <- fortify(GBRcoast,id="id")
GRC <- fortify(GRCcoast,id="id")
GEO <- fortify(GEOcoast,id="id")
GRL <- fortify(GRLcoast,id="id")
HRV <- fortify(HRVcoast,id="id")
HUN <- fortify(HUNcoast,id="id")
IMN <- fortify(IMNcoast,id="id")
IRL <- fortify(IRLcoast,id="id")
IRN <- fortify(IRNcoast,id="id")
ISL <- fortify(ISLcoast,id="id")
ITA <- fortify(ITAcoast,id="id")
KAZ <- fortify(KAZcoast,id="id")
LIE <- fortify(LIEcoast,id="id")
LTU <- fortify(LTUcoast,id="id")
LVA <- fortify(LVAcoast,id="id")
LUX <- fortify(LUXcoast,id="id")
MDA <- fortify(MDAcoast,id="id")
MKD <- fortify(MKDcoast,id="id")
MNE <- fortify(MNEcoast,id="id")
NLD <- fortify(NLDcoast,id="id")
NOR <- fortify(NORcoast,id="id")
POL <- fortify(POLcoast,id="id")
PRT <- fortify(PRTcoast,id="id")
ROU <- fortify(ROUcoast,id="id")
RUS <- fortify(RUScoast,id="id")
SJM <- fortify(SJMcoast,id="id")
SRB <- fortify(SRBcoast,id="id")
SVK <- fortify(SVKcoast,id="id")
SVN <- fortify(SVNcoast,id="id")
SWE <- fortify(SWEcoast,id="id")
TKM <- fortify(TKMcoast,id="id")
TUR <- fortify(TURcoast,id="id")
UKR <- fortify(UKRcoast,id="id")
```

~~~
XKO <- fortify(XKOcoast,id="id")
```

# FIGURES
**Figures 1 & 2 are images of Research Vessels.**

The following Sections provide the code used to generate the Figures used in the Data Paper.

## Common /Map & / or Plot parameters

```{r CommonParams}
PointColour="red3" # all maps and figures
```

# FIGURE 3
**Overview of the location of "all" Cefas seawater temperature measurements**

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.

## F3 Create the required Labels for Lat & Long

Create the breaks- and label vectors

http://stackoverflow.com/questions/33302424/format-latitude-and-longitude-axis-labels-in-ggplot

```{r Fig3CreateLabels}
ewbrks <- seq(-50,50,20) # 50 West to 50 East in 10 degree units
nsbrks <- seq(40,70,10) # differ from ukcsPlus etc as rounded to 5 and 10
ewlbls <- unlist(lapply(ewbrks, function(x) ifelse(x < 0, paste(x*-1, "°W"), ifelse(x > 0, paste(x, "°E"),x))))
nslbls <- unlist(lapply(nsbrks, function(x) ifelse(x < 0, paste(x*-1, "°S"), ifelse(x > 0, paste(x, "°N"),x))))
```

## F3 Create the map

```{r Fig3PlotToFile}
ppi=300
Figure.width = 16/2.54
Figure.height = 16/2.54
#
autoplot(map.ukcsPlus,geom="contour",colour="gray",size=0.1) +
scale_x_continuous(limits = c(ukcsPlus.west, ukcsPlus.east), breaks=ewbrks, labels=ewlbls, expand = c(0, 0)) +
scale_y_continuous(limits = c(ukcsPlus.south, ukcsPlus.north), breaks=nsbrks, labels=nslbls, expand = c(0, 0)) +
coord_map(xlim=c(ukcsPlus.west,ukcsPlus.east),ylim=c(ukcsPlus.south,ukcsPlus.north)) +
theme(panel.background = element_rect(fill = "lightsteelblue1")) +
geom_point(aes(x=Long,y=Lat), data = SWTfinal, col=PointColour, size=0.5, alpha = 1/10) +
  labs(x="Longitude", y="Latitude") +
  theme(axis.title = element_text(size = 12)) +
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12)) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=RUS) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=ALB) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=ARM) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=AUT) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=AZE) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BEL) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BGR) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BIH) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BLR) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=CAN) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=CHE) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=DEU) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=DNK) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=ESP) +
~~~

````
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=EST) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=FIN) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=FRA) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=FRO) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=GBR) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=GEO) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=GRC) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=GRL) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=HRV) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=HUN) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=IMN) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=IRL) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=IRN) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=ISL) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=ITA) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=KAZ) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=LIE) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=LTU) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=LUX) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=LVA) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=MDA) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=MKD) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=MNE) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=NLD) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=NOR) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=POL) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=PRT) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=ROU) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=SJM) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=SRB) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=SVK) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=SVN) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=SWE) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=TKM) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=TUR) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=UKR) +
 geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=XKO) +
  geom_rect(data=box.CHE, aes(xmin=minlong, xmax=maxlong, ymin=minlat, ymax=maxlat),
  color="darkgreen", fill="darkgreen", inherit.aes=FALSE) +
  geom_rect(data=box.GRN, aes(xmin=minlong, xmax=maxlong, ymin=minlat, ymax=maxlat),
  color="darkgreen", fill="darkgreen", inherit.aes=FALSE) +
  geom_rect(data=box.TUR.1, aes(xmin=minlong, xmax=maxlong, ymin=minlat,
  ymax=maxlat), color="darkgreen", fill="darkgreen", inherit.aes=FALSE) +
  geom_rect(data=box.TUR.2, aes(xmin=minlong, xmax=maxlong, ymin=minlat,
  ymax=maxlat), color="darkgreen", fill="darkgreen", inherit.aes=FALSE) +
    theme(axis.text = element_text(size=12))
#
# https://gis.stackexchange.com/questions/165974/r-fortify-causing-polygons-to-tear
# Greenland and Turkey 'tear' - fixed the image with polygons for now
ggsave("fig03.png", width=Figure.width, height = Figure.height, dpi=ppi)
```
````

FIGURE 4

Data Density Plot - UKCS waters

F4 Subset the data

````
```{r Fig4CreateData}
ukcsAll <- subset(SWTfinal, Lat < ukcs.north)
ukcsAll <- subset(ukcsAll, Lat > ukcs.south)
ukcsAll <- subset(ukcsAll, Long < ukcs.east)
ukcsAll <- subset(ukcsAll, Long > ukcs.west)
ukcsLatLong <- ukcsAll[c("Lat","Long")]
LatLongTotal<-nrow(ukcsLatLong)
#
ReductionFactor = 5 # just fits into 16 Gbyte Cefas Windows 10 PC limit
#
ukcsSeq <-ukcsLatLong
ukcsSeq <- ukcsSeq[seq(1,LatLongTotal,ReductionFactor),]
```
````

F4 Plot the data
````

````
```{r Fig4CreateDensityData&Plot}
#DENSITY PARAMETERS
# provide optimum plot to illustrate data coverage
#
binNum = 2000
denSize=0.1
#
# Data for Density calculations
d <- ukcsSeq
#
# SET UP OUTPUT FILE
ppi=300
Figure.width = 16/2.54
Figure.height = 16/2.54
#
# CALCULATE DENSITIES AND PLOT
#
# 16 Gbyte Windows Memory limits approached - auto memory clear by R insufficiently reliable
gc()
# LABELS
ewbrks <- seq(-15,10,5)
nsbrks <- seq(45,60,5)
ewlbls <- unlist(lapply(ewbrks, function(x) ifelse(x < 0, paste(x*-1, "°W"), ifelse(x > 0, paste(x, "°E"),x))))
nslbls <- unlist(lapply(nsbrks, function(x) ifelse(x < 0, paste(x*-1, "°S"), ifelse(x > 0, paste(x, "°N"),x))))
#
# 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 ligtsteelblue and the grid lines AND the axis annotations, although the degrre fix works.
# moving theme(panel.background...) to the end of the code makes no difference.
# manually aligning the scales in the labels and the discrete scale works! duh!!!
# That will teach me to copy & paste too much.
autoplot(map.ukcs, geom="contour",colour="gray", size=0.1) +
  coord_map(xlim=c(ukcs.west,ukcs.east),ylim=c(ukcs.south,ukcs.north)) +
 scale_x_discrete(limits = c(-15,-10,-5,0,5,10), breaks=ewbrks, labels=ewlbls, expand = c(0, 0)) +
 scale_y_discrete(limits = c(45,50,55,60), breaks=nsbrks, labels=nslbls, expand = c(0, 0)) +
    labs(x="Longitude", y="Latitude") +
  stat_density2d(data = d, aes(x = Long, y = Lat, fill = ..level.., alpha = ..level..), size = 10, bins = binNum, geom = "polygon") +
  geom_density2d(data = d, aes(x = Long, y = Lat, size = denSize), colour=PointColour, size=0.01) +
  scale_fill_gradient2(low = "cyan", mid="green",high = "yellow",guide = "colourbar") +
  scale_alpha(range = c(0, 0.3), guide = FALSE) +
  theme(legend.position="none") +
  geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BEL) +
  #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=CHE) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DNK) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DEU) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=FRA) +
  geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=FRO) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=GBR) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IMN) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IRL) +
  #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LIE) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LUX) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NLD) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NOR)+
  theme(panel.background = element_rect(fill = "lightsteelblue1"))
  #
ggsave("fig04.png", width=Figure.width, height = Figure.height, dpi=ppi)
#
```
````

# FIGURE 5
**Overview of numbers of "all" Cefas seawater temperature measurements by year**

## F5 Subset the data

````
```{r Fig5CreateData}
#
SWT.early<-subset(SWTfinal,SWTfinal$Time<"1960-01-01 00:00:00")
SWT.late<-subset(SWTfinal,SWTfinal$Time>="1940-01-01 00:00:00")
#
```
````

## F5 Plot the data

````
```{r Fig5plotFile}
ppi=300
Figure.width = 16/2.54
Figure.height = 10/2.54
png("fig05.png", type = "cairo", width = Figure.width, height = Figure.height, units = "in", pointsize = 12, res = ppi)
p1 <- ggplot(data = SWT.early, aes(x=Time)) +
       geom_histogram() +
       scale_y_continuous(labels = fancy_scientific) +
       labs(x="Year", y="Count") +
       ggtitle("(a)") +
       theme_bw()
#
p2 <- ggplot(data = SWT.late, aes(x=Time)) +
  geom_histogram() +
  scale_y_continuous(labels = fancy_scientific) +
  labs(x="Year", y="Count") +
  ggtitle("(b)") +
  theme_bw()
#
multiplot(p1,p2, cols = 2)
dev.off()
```
````

# FIGURE 6
**Overview of "all" Cefas seawater temperature measurements by depth - 1 m bin**

## F6 Subset the data

a <=10, b >10 <=100 c >100 <=250 d >250 (surface, thermocline/productive, shelf, rest)

Surface means top 10 m here - to display data distribution only - this changes for other, comparison based graphs below.

````
```{r Fig6subsetData}
surface<-SWTfinal
surface<-subset(surface,Depth<=10)
#
productive<-SWTfinal
productive<-subset(productive,Depth>10)
productive<-subset(productive,Depth<=100)
#
shelf<-SWTfinal
shelf<-subset(shelf, Depth>100)
shelf<-subset(shelf, Depth<=250)
#
deepest<-SWTfinal
deepest<-subset(deepest, Depth>250)
#
# cross check
#
print(paste0("Surface           = ", NROW(surface)))
print(paste0("Productive        = ", NROW(productive)))
print(paste0("Shelf             = ", NROW(shelf)))
print(paste0("Deepest           = ", NROW(deepest)))
#
print(" ")
print(paste0("Sum of above      = ", NROW(surface) + NROW(productive) + NROW(shelf) + NROW(deepest)))
print(paste0("Total             = ", NROW(SWTfinal)))
#
````
````

````
```

## F6 Plot the data

```{r Fig6plotFile}
ppi=300
Figure.width = 16/2.54
Figure.height = 14/2.54
png("fig06.png", type = "cairo", width = Figure.width, height = Figure.height, units = "in", pointsize = 12, res = ppi)
#
p1 <- ggplot(data = surface, aes(x=Depth)) +
  geom_histogram() +
  scale_y_continuous(labels = fancy_scientific) +
  labs(x="Depth (m)", y="Count") +
  ggtitle("(a)") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_bw()
p2 <- ggplot(data = productive, aes(x=Depth)) +
  geom_histogram() +
  scale_y_continuous(labels = fancy_scientific) +
  labs(x="Depth (m)", y="Count") +
  ggtitle("(b)") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_bw()
p3 <- ggplot(data = shelf, aes(x=Depth)) +
  geom_histogram() +
  scale_y_continuous(labels = fancy_scientific) +
  labs(x="Depth (m)", y="Count") +
  ggtitle("(c)") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_bw()
p4 <- ggplot(data = deepest, aes(x=Depth)) +
  geom_histogram() +
  scale_y_continuous(labels = fancy_scientific) +
  labs(x="Depth (m)", y="Count") +
  ggtitle("(d)") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_bw()
#
multiplot(p1,p3,p2,p4, cols = 2)
dev.off()
```

FIGURE 7
Location of Cefas temperature data by ICES Rectangle group - SURFACE (0-5m)

F7 Set the selected depth thresholds.

```{r DepthThresholds}
#
# 5m used to include intake data from Ferries (1.5) & RVs (4, 4.5)
#
upperdepth.threshold=0
lowerdepth.threshold=5
interimdepth.threshold = 1 # includes SmartBuouy & WaveNet data - 1m and 0.4m nominal sensor depths respectively
```

F7 Subset the data

```{r Fig7subsetData}
#
SWTfinalSURFACE<-subset(SWTfinal, Depth>=upperdepth.threshold & Depth<=lowerdepth.threshold)
#
swt.lpoolBay<-SWTfinalSURFACE
swt.lpoolBay<-subset(swt.lpoolBay, Lat <= lpoolBay.north)
swt.lpoolBay<-subset(swt.lpoolBay, Lat >= lpoolBay.south)
swt.lpoolBay<-subset(swt.lpoolBay, Long >= lpoolBay.west)
swt.lpoolBay<-subset(swt.lpoolBay, Long <= lpoolBay.east)
#
swt.celticSea<-SWTfinalSURFACE
````

````
swt.celticSea<-subset(swt.celticSea, Lat <= celticSea.north)
swt.celticSea<-subset(swt.celticSea, Lat >= celticSea.south)
swt.celticSea<-subset(swt.celticSea, Long >= celticSea.west)
swt.celticSea<-subset(swt.celticSea, Long <= celticSea.east)
#
swt.brixham<-SWTfinalSURFACE
swt.brixham<-subset(swt.brixham, Lat <= brixham.north)
swt.brixham<-subset(swt.brixham, Lat >= brixham.south)
swt.brixham<-subset(swt.brixham, Long >= brixham.west)
swt.brixham<-subset(swt.brixham, Long <= brixham.east)
#
swt.thames<-SWTfinalSURFACE
swt.thames<-subset(swt.thames, Lat <= thames.north)
swt.thames<-subset(swt.thames, Lat >= thames.south)
swt.thames<-subset(swt.thames, Long >= thames.west)
swt.thames<-subset(swt.thames, Long <= thames.east)
#
```

## F7 Plot the data

```{r Fig7CreateLabels}
ewbrks <- seq(-10,4,2)
nsbrks <- seq(48, 54,2)
ewlbls <- unlist(lapply(ewbrks, function(x) ifelse(x < 0, paste(x*-1, "°W"), ifelse(x > 0, paste(x, "°E"),x))))
nslbls <- unlist(lapply(nsbrks, function(x) ifelse(x < 0, paste(x*-1, "°S"), ifelse(x > 0, paste(x, "°N"),x))))
```

```{r Fig7 createBounding Boxes}
box.Lpool <- data.frame(maxlat = 54, minlat = 53, maxlong = -3, minlong = -5, id="(a)")
box.Lpool <- transform(box.Lpool, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
box.Celtic <- data.frame(maxlat = 51, minlat = 50, maxlong = -7, minlong = -9, id="(b)")
box.Celtic <- transform(box.Celtic, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
box.Brixham <- data.frame(maxlat = 50.5, minlat = 49.5, maxlong = -2, minlong = -4, id="(c)")
box.Brixham <- transform(box.Brixham, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
box.Thames <- data.frame(maxlat = 52.5, minlat = 51.5, maxlong = 3, minlong = 1, id="(d)")
box.Thames <- transform(box.Thames, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
```

```{r Fig7PlotToFile}
ppi=300
Figure.width = 16/2.54
Figure.height = 16/2.54
#
autoplot(map.ices, geom="contour", colour="gray", size=0.1) +
scale_x_continuous(limits = c(ices.west, ices.east), breaks=ewbrks, labels=ewlbls, expand = c(0, 0)) +
scale_y_continuous(limits = c(ices.south, ices.north), breaks=nsbrks, labels=nslbls, expand = c(0, 0)) +
coord_map(xlim=c(ices.west,ices.east),ylim=c(ices.south,ices.north)) +
theme(panel.background = element_rect(fill = "lightsteelblue1")) +
  theme(legend.position = "none") +
  geom_point(aes(x=Long,y=Lat), data = swt.lpoolBay, col=PointColour, size=0.1, alpha = 1/10) +
  geom_point(aes(x=Long,y=Lat), data = swt.celticSea, col=PointColour, size=0.1, alpha = 1/10) +
  geom_point(aes(x=Long,y=Lat), data = swt.brixham, col=PointColour, size=0.1, alpha = 1/10) +
  geom_point(aes(x=Long,y=Lat), data = swt.thames, col=PointColour, size=0.1, alpha = 1/10) +
  labs(x="Longitude", y="Latitude") +
  theme(axis.title = element_text(size = 12)) +
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12)) +
  theme(axis.text = element_text(size=12)) +
````

````
  scale_x_continuous(breaks = ewbrks, labels = ewlbls, expand = c(0, 0)) +
  scale_y_continuous(breaks = nsbrks, labels = nslbls, expand = c(0, 0)) +
  geom_rect(data=box.Lpool, aes(xmin=minlong, xmax=maxlong, ymin=minlat, ymax=maxlat ), color="black", fill="transparent", inherit.aes=FALSE) +
  geom_text(data=box.Lpool, aes(x=labx, y=laby), color="black",label=" ") +
  geom_rect(data=box.Celtic, aes(xmin=minlong, xmax=maxlong, ymin=minlat, ymax=maxlat ), color="black", fill="transparent", inherit.aes=FALSE) +
  geom_text(data=box.Celtic, aes(x=labx, y=laby), color="black",label=" ") +
  geom_rect(data=box.Brixham, aes(xmin=minlong, xmax=maxlong, ymin=minlat, ymax=maxlat ), color="black", fill="transparent", inherit.aes=FALSE) +
  geom_text(data=box.Brixham, aes(x=labx, y=laby), color="black",label=" ") +
  geom_rect(data=box.Thames, aes(xmin=minlong, xmax=maxlong, ymin=minlat, ymax=maxlat ), color="black", fill="transparent", inherit.aes=FALSE) +
  geom_text(data=box.Thames, aes(x=labx, y=laby), color="black",label=" ")  +
  geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BEL) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DEU) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=FRA) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=GBR) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IMN) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IRL) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LIE) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LUX) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NLD)
  ggsave("fig07.png", width=Figure.width, height = Figure.height, dpi=ppi)
```

# FIGURE 8
**Southern North Sea - Data Availability - NEAR SURFACE**

## F8 Subset the Data

```{r Fig8subsetData}
# NEAR SURFACE DATA
#
SNS<-SWTfinal
SNSsurface=lowerdepth.threshold
SNS<-subset(SNS,Depth<=SNSsurface)
SNS<-subset(SNS,Lat>=SNSbb.south)
SNS<-subset(SNS,Lat<=SNSbb.north)
SNS<-subset(SNS,Long<=SNSbb.east)
SNS<-subset(SNS,Long>=SNSbb.west)
```

## F8 Plot the data

```{r Fig8CreateBB}
box.SNS <- data.frame(maxlat = SNSbb.north, minlat = SNSbb.south, maxlong = SNSbb.east, minlong = SNSbb.west, id=" ")
box.SNS <- transform(box.SNS, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
```

```{r Fig8CreateLabels}
ewbrks <- seq(-1,5,1)
nsbrks <- seq(51,54,1)
ewlbls <- unlist(lapply(ewbrks, function(x) ifelse(x < 0, paste(x*-1, "°W"), ifelse(x > 0, paste(x, "°E"),x))))
nslbls <- unlist(lapply(nsbrks, function(x) ifelse(x < 0, paste(x*-1, "°S"), ifelse(x > 0, paste(x, "°N"),x))))
```

```{r Fig8PlotToFile}
ppi=300
Figure.width = 16/2.54
Figure.height = 16/2.54
#
autoplot(map.SNS, coast=FALSE,geom="contour",colour="gray", size=0.1) +
scale_x_continuous(limits = c(SNS.west, SNS.east), breaks=ewbrks, labels=ewlbls, expand = c(0, 0)) +
scale_y_continuous(limits = c(SNS.south, SNS.north), breaks=nsbrks, labels=nslbls, expand = c(0, 0)) +
coord_map(xlim=c(SNS.west,SNS.east),ylim=c(SNS.south,SNS.north)) +
theme(panel.background = element_rect(fill = "lightsteelblue1")) +
  theme(legend.position = "none") +
````

````
    geom_point(aes(x=Long,y=Lat), data = SNS, col=PointColour, size=0.1, alpha = 1/10) +
    labs(x="Longitude", y="Latitude") +
    theme(axis.title = element_text(size = 12)) +
    theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12)) +
    theme(axis.text = element_text(size=12)) +
    scale_x_continuous(breaks = ewbrks, labels = ewlbls, expand = c(0, 0)) +
    scale_y_continuous(breaks = nsbrks, labels = nslbls, expand = c(0, 0)) +
     geom_rect(data=box.SNS, aes(xmin=minlong, xmax=maxlong, ymin=minlat, ymax=maxlat ), color="black", fill="transparent", inherit.aes=FALSE) +
    geom_text(data=box.SNS, aes(x=labx, y=laby, label=id), color="black") +
    geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BEL) +
    #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DEU) +
    geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=FRA) +
    geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=GBR) +
    #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IMN) +
    #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LIE) +
   # geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LUX) +
    geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NLD)
  #
  ggsave("fig08.png", width=Figure.width, height = Figure.height, dpi=ppi)
  ```

  # FIGURE 9
  **SNS Temperature .v. Time by key sources**

  ## F9 Subset the data
  **SEPARATE BY SOURCE - use top 9 by number**

  ```{r Fig9seperateSources}
  #
  SNS01<- subset(SNS, Source=="1") #exclude - low number <1000
  SNS02<- subset(SNS, Source=="2") #exclude - low number <1000
  SNS03<- subset(SNS, Source=="3")
  SNS04<- subset(SNS, Source=="4")
  SNS05<- subset(SNS, Source=="5") #exclude - low number <1000
  SNS06<- subset(SNS, Source=="6")
  SNS07<- subset(SNS, Source=="7")
  SNS08<- subset(SNS, Source=="8")
  SNS09<- subset(SNS, Source=="9") #exclude - low number <1000
  SNS10<- subset(SNS, Source=="10") #exclude - low number <1000
  SNS11<- subset(SNS, Source=="11")
  SNS12<- subset(SNS, Source=="12") ##exclude - low number <1000
  SNS13<- subset(SNS, Source=="13") #exclude - low number <1000
  SNS14<- subset(SNS, Source=="14") #exclude - low number <1000
  SNS15<- subset(SNS, Source=="15")
  SNS16<- subset(SNS, Source=="16")
  SNS17<- subset(SNS, Source=="17")
  #
  SNS03$year <- year(SNS03$Time)
  SNS04$year <- year(SNS04$Time)
  SNS06$year <- year(SNS06$Time)
  SNS07$year <- year(SNS07$Time)
  SNS08$year <- year(SNS08$Time)
  SNS11$year <- year(SNS11$Time)
  SNS15$year <- year(SNS15$Time)
  SNS16$year <- year(SNS16$Time)
  SNS17$year <- year(SNS17$Time)
  #
  ```

  ## F9 Plot the data

  ```{r Fig9plotFile}
  ppi=300
  Figure.width = 16/2.54
  Figure.height = 16/2.54
  png("fig09.png", type = "cairo", width = Figure.width, height = Figure.height, units = "in", pointsize = 12, res = ppi)
  #
  p1 <- ggplot(data = SNS03, aes(x=Time, y=tC)) +
    geom_point(color="blue",size=0.2) +
    labs(x="Year", y="Temperature °C") +
    ggtitle("Source 3") +
````

```
  # theme(plot.title = element_text(size=5, face = "bold")) +
  scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
  theme_bw() +
  theme(axis.title.x = element_text(face="plain", size=10),
           axis.text.x  = element_text(angle=90, vjust=0.5, size=10))
#
p2 <- ggplot(data = SNS04, aes(x=Time, y=tC)) +
  geom_point(color="gray64",size=0.2) +
  labs(x="Year", y="Temperature °C") +
  ggtitle("Source 4") +
  theme(plot.title = element_text(size=5, face = "bold")) +
  scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
  theme_bw() +
  theme(axis.title.x = element_text(face="plain", size=10),
           axis.text.x  = element_text(angle=90, vjust=0.5, size=10))
#
p3 <- ggplot(data = SNS06, aes(x=Time, y=tC)) +
  geom_point(color="orange",size=0.1) +
  labs(x="Year", y="Temperature °C") +
  ggtitle("Source 6") +
  theme(plot.title = element_text(size=5, face = "bold")) +
  scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
  theme_bw() +
  theme(axis.title.x = element_text(face="plain", size=10),
           axis.text.x  = element_text(angle=90, vjust=0.5, size=10))
#
p4 <- ggplot(data = SNS07, aes(x=Time, y=tC)) +
  geom_point(color="orange",size=0.1) +
  labs(x="Year", y="Temperature °C") +
  ggtitle("Source 7") +
  theme(plot.title = element_text(size=5, face = "bold")) +
  scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
  theme_bw() +
  theme(axis.title.x = element_text(face="plain", size=10),
           axis.text.x  = element_text(angle=90, vjust=0.5, size=10))
#
p5 <- ggplot(data = SNS08, aes(x=Time, y=tC)) +
  geom_point(color=PointColour,size=0.1) +
  labs(x="Year", y="Temperature °C") +
  ggtitle("Source 8") +
  theme(plot.title = element_text(size=5, face = "bold")) +
  scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
  theme_bw() +
  theme(axis.title.x = element_text(face="plain", size=10),
           axis.text.x  = element_text(angle=90, vjust=0.5, size=10))
#
p6 <- ggplot(data = SNS11, aes(x=Time, y=tC)) +
  geom_point(color="magenta",size=0.2) +
  labs(x="Year", y="Temperature °C") +
  ggtitle("Source 11") +
  theme(plot.title = element_text(size=5, face = "bold")) +
  scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
  theme_bw() +
  theme(axis.title.x = element_text(face="plain", size=10),
           axis.text.x  = element_text(angle=90, vjust=0.5, size=10))
#
p7 <- ggplot(data = SNS15, aes(x=Time, y=tC)) +
  geom_point(color="orange",size=0.1) +
  labs(x="Year", y="Temperature °C") +
  ggtitle("Source 15") +
  theme(plot.title = element_text(size=5, face = "bold")) +
  scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
  theme_bw() +
  theme(axis.title.x = element_text(face="plain", size=10),
           axis.text.x  = element_text(angle=90, vjust=0.5, size=10))
#
p8 <- ggplot(data = SNS16, aes(x=Time, y=tC)) +
  geom_point(color="green",size=0.2) +
  labs(x="Year", y="Temperature °C") +
  ggtitle("Source 16") +
  theme(plot.title = element_text(size=5, face = "bold")) +
  scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
 theme_bw() +
```

````
  theme(axis.title.x = element_text(face="plain", size=10),
           axis.text.x  = element_text(angle=90, vjust=0.5, size=10))
#
p9 <- ggplot(data = SNS17, aes(x=Time, y=tC)) +
  geom_point(color="turquoise3",size=0.2) +
  labs(x="Year", y="Temperature °C") +
  ggtitle("Source 17") +
  theme(plot.title = element_text(size=5, face = "bold")) +
  scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
 theme_bw() +
  theme(axis.title.x = element_text(face="plain", size=10),
           axis.text.x  = element_text(angle=90, vjust=0.5, size=10))
#
multiplot(p1,p4,p7,p2,p5,p8,p3,p6,p9, cols = 3)
#
dev.off()
```

# Figure 10
**SNS - Temperature .v. Date - All Surface Data - coloured by Source**

## F10 Subset the data

red > 100,000 points (Sources 6 & 7 - SmartBuoy & WaveNet)
blue 30,000 - 50,000 (8 & 15)
green 2,000 - 6,000 (3,4,11,16,17)

## F10 Plot the data

```{r Fig10plotFile}
ppi=300
Figure.width = 16/2.54
Figure.height = 12/2.54
png("fig10.png", type = "cairo", width = Figure.width, height = Figure.height, units = "in", pointsize = 12, res = ppi)

p1 <- ggplot(data = SNS, aes(x=Time)) +
       geom_histogram() +
       scale_y_continuous(labels = fancy_scientific) +
       labs(x="Year", y="Count") +
       ggtitle("(a)") +
       theme_bw()
#
p2 <- ggplot(data = SNS03, aes(x=Time, y=tC)) +
  geom_point(data=SNS03, color="blue",size=0.1) +
  geom_point(data=SNS04, color="gray64",size=0.1) +
  geom_point(data=SNS08, color=PointColour,size=0.1) +
  geom_point(data=SNS11, color="magenta",size=0.1) +
  geom_point(data=SNS15, color="orange",size=0.1) +
  geom_point(data=SNS16, color="green",size=0.1) +
  geom_point(data=SNS17, color="turquoise3",size=0.1) +
  geom_point(data=SNS06, color="orange",size=0.1, alpha = .5) +
  geom_point(data=SNS07, color="orange",size=0.1, alpha = .5) +
  labs(x="Year", y="Temperature °C") +
  ggtitle("(b)") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
#
multiplot(p1,p2, cols = 1)
dev.off()
```

The question arising from Fig 10 (b) is *"there looks to be a change in the minimum values after 2000, with no values < ~ 5C. Is this an artefact of data source?"*. The following few figures explore this.

# FIGURE 11
**SNS after 2000 - Distribution of data for Surface 0-5m and 0-1m & E/W - point/vessel**

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).

F11 Subset the data

```{r Fig11subsetData}
#
SNS2000 <- subset(SNS, year(Time) >= 2000)
#
SNS2000.1 <- subset(SNS2000, Depth <= interimdepth.threshold)
SNS2000.5 <- subset(SNS2000, Depth > interimdepth.threshold & Depth <= lowerdepth.threshold)
#
SBuoyWNet <- subset(SNS2000.1, Source == 6 | Source == 7) # OR
```

F11 Calculate Percentages

```{r SNSbias}
#
PercentPost2000 <- 100-(NROW(SNS)-NROW(SNS2000))*100/NROW(SNS)
#
PercentPost2000.1 <- 100-(NROW(SNS2000)-NROW(SNS2000.1))*100/NROW(SNS2000)
#
specify_decimal <- function(x, k) format(round(x, k), nsmall=k)
#
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 buouys but excluding vessel mounted pump systems] is ", specify_decimal(PercentPost2000.1,1))))
#
```

F11 Plot the data

```{r Fig11CreateBoxes}
box.SNSa <- data.frame(maxlat = SNSbb.north, minlat = SNSbb.south, maxlong = SNSbb.east, minlong = SNSbb.west, id="a")
box.SNSa <- transform(box.SNSa, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
#
box.SNSb <- data.frame(maxlat = SNSbb.north, minlat = SNSbb.south, maxlong = SNSbb.east, minlong = SNSbb.west, id="b")
box.SNSb <- transform(box.SNSb, laby=(maxlat +minlat )/2, labx=(maxlong+minlong)/2)
```

```{r Fig11CreateLabels}
ewbrks <- seq(0,5,1)
nsbrks <- seq(51,54,1)
ewlbls <- unlist(lapply(ewbrks, function(x) ifelse(x < 0, paste(x*-1, "°W"), ifelse(x > 0, paste(x, "°E"),x))))
nslbls <- unlist(lapply(nsbrks, function(x) ifelse(x < 0, paste(x*-1, "°S"), ifelse(x > 0, paste(x, "°N"),x))))
```

```{r Fig11PlotToFile}
ppi=300
Figure.width = 16/2.54
Figure.height = 8/2.54
#
png("fig11.png", type = "cairo", width = Figure.width, height = Figure.height, units = "in", pointsize = 12, res = ppi)
#
p1 <- autoplot(map.SNS, coast=FALSE,geom=("contour"),colour="gray", size=0.01) +
  theme(legend.position = "none") +
  geom_point(aes(x=Long,y=Lat), data = SNS2000.5, col=PointColour, size=0.1, alpha = 1/10) +
  coord_map(xlim=c(SNS.west,SNS.east),ylim=c(SNS.south,SNS.north)) +
  labs(x="Longitude", y="Latitude") +
  theme(axis.title = element_text(size = 12)) +
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12)) +
  theme(axis.text = element_text(size=12)) +
  scale_x_continuous(breaks = ewbrks, labels = ewlbls, expand = c(0, 0)) +
````
````

```
  scale_y_continuous(breaks = nsbrks, labels = nslbls, expand = c(0, 0)) +
   geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BEL) +
  #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DEU) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=FRA) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=GBR) +
  #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IMN) +
  #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LIE) +
 # geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LUX) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NLD) +
   geom_rect(data=box.SNSa, aes(xmin=minlong, xmax=maxlong, ymin=minlat, ymax=maxlat ), color="black", fill="transparent", inherit.aes=FALSE) +
  geom_text(data=box.SNSa, aes(x=labx, y=laby, label=id), color="black")+
  theme(panel.background = element_rect(fill = "lightsteelblue1"))
p2 <- autoplot(map.SNS, coast=FALSE, geom=("contour"),colour="gray", size=0.01) +
  #scale_fill_etopo() +
  theme(legend.position = "none") +
    coord_map(xlim=c(SNS.west,SNS.east),ylim=c(SNS.south,SNS.north)) +
  geom_point(aes(x=Long, y=Lat), data = SNS2000.1, col=PointColour, size=0.1, alpha = 1/10) +
  geom_point(aes(x=Long, y=Lat), data = SBuoyWNet, shape=18, col = "orange", size = 2) +
  labs(x="Longitude", y="Latitude") +
  theme(axis.title = element_text(size = 12)) +
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12)) +
  theme(axis.text = element_text(size=12)) +
  scale_x_continuous(breaks = ewbrks, labels = ewlbls, expand = c(0, 0)) +
  scale_y_continuous(breaks = nsbrks, labels = nslbls, expand = c(0, 0)) +
   geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BEL) +
  #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DEU) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=FRA) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=GBR) +
  #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IMN) +
  #geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LIE) +
 # geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LUX) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NLD) +
   geom_rect(data=box.SNSb, aes(xmin=minlong, xmax=maxlong, ymin=minlat, ymax=maxlat ), color="black", fill="transparent", inherit.aes=FALSE) +
  geom_text(data=box.SNSb, aes(x=labx, y=laby, label=id), color="black") +
  theme(panel.background = element_rect(fill = "lightsteelblue1"))
# ggsave and multiplot issues so
#
multiplot(p1,p2, cols = 2)
dev.off()
#
```

This illustrates a numerical and a geographic bias for the point sources, south and west.

```{r CalcPercents}
specify_decimal <- function(x, k) format(round(x, k), nsmall=k)
Percent.Moored <- 100-(NROW(SNS2000.1)-NROW(SBuoyWNet))*100/NROW(SNS2000.1)
print(paste0("Percentage of point source data (Sources 6 & 7) in SNS Surface (0-1m), post 2000 is ", specify_decimal(Percent.Moored,1)))
Percent.Moored <- 100-(NROW(SNS2000.1)-NROW(SBuoyWNet))*100/NROW(SNS2000.5)
print(paste0("Percentage of point source data (Sources 6 & 7) in SNS Surface (0-5m), post 2000 is ", specify_decimal(Percent.Moored,1)))
```

# FIGURE 12
**SNS post 2000 - Surface (0-1m) Temperature .v. Time for non-point and point data**

## F12 Subset the data

```{r Fig12subsetData}
NOTSBuoyWNet <- subset(SNS2000.1, Source != 6 & Source != 7) # NOT

print(NROW(NOTSBuoyWNet))
#
print(NROW(SBuoyWNet))
```

## F12 Plot the data

````
```{r Fig12plotFile}
ppi=300
Figure.width = 16/2.54
Figure.height = 8/2.54
png("fig12.png", type = "cairo", width = Figure.width, height = Figure.height, units = "in", pointsize = 12, res = ppi)
#
p1 <- ggplot(data = NOTSBuoyWNet, aes(x=Time, y=tC)) +
  geom_point(color=PointColour,size=0.05) +
  labs(x="Year", y="Temperature °C") +
  ggtitle("(a)") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
p2 <- ggplot(data = SBuoyWNet, aes(x=Time, y=tC)) +
  geom_point(color="orange",size=0.05) +
  labs(x="Year", y="Temperature °C") +
  ggtitle("(b)") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(-2, 25),breaks = seq(0,25,5)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
#
multiplot(p1,p2, cols=2)
#
dev.off()
```

FIGURE 13
Plot of min, mean & max for SNS surface data (0-5m)

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).

F13 Create the statistics data

```{r SNSstats}
# stats by year
# create a new Variable called Year
SNS$Year <- year(SNS$Time)
#
SNS1925on <- subset(SNS, Year >= 1925 & Year <= 2015)
#
# use Data Table for easy extraction of descriptive statistics
# recreate a simple data frame of year and tC
Year.tC <- data.frame(SNS1925on$Year, SNS1925on$tC)
setnames(Year.tC, old=c("SNS1925on.Year", "SNS1925on.tC"), new=c("Year","Temperature"))
#
# now convert to data.table for quick summary stats
#
Simple.Data <- data.table(Year.tC)
tC.stats <- Simple.Data[,list(mean=mean(Temperature),
                              min=min(Temperature),
                              max=max(Temperature)),
                        by='Year']

# revert to data frames to keep plotting consistenmt and reduce data.table plot learning curve to 0
#
#e.g. tC.stats[, matplot(SNS.sub.Year, mean, type="p", ylab="Temperature stats", xlab="Year"), .SDcols = !"SNS.sub.Year"]
#
class(as.data.frame(tC.stats))
#
# generate a subset for lines as overplot
#
line.data <- subset(tC.stats, Year >= 1955)
line.data <- line.data[order(line.data$Year),]
#
````
````

````
```

## F13 Plot the data

```{r Fig13plotFile}
#
ppi=300
Figure.width = 16/2.54
Figure.height = 16/2.54 # height increased to show more tC detail
png("fig13.png", type = "cairo", width = Figure.width, height = Figure.height, units = "in", pointsize = 12, res = ppi)
#
ggplot(data = tC.stats, aes(x=Year, y=mean)) +
  geom_point(color="green") +
  geom_line(data=line.data, aes(x=Year, y=mean), colour = "green") +
  geom_point(data = tC.stats, aes(x=Year, y=max),color=PointColour) +
  geom_line(data= line.data, aes(x=Year, y=max), color=PointColour) +
  geom_point(data = tC.stats, aes(x=Year, y=min),color="blue") +
  geom_line(data= line.data, aes(x=Year, y=min), color="blue") +
  labs(x="Year", y="Annual Average Temperature °C") +
  scale_y_continuous(breaks = seq(0,20,5)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
dev.off()
#
```

FIGURE 14
3d Plot of Data Sources in SNS

A selected smaller 'belt' to illustrate multiple sources and how they are represented in a depth profile. SNS sub-area & small to make plot look clearer - ALL depths.

F14 Subset the data

```{r CreateData3dPlotSNSbelt}
belt.north = 54.8
belt.south = 54.3
belt.east = 3.28
belt.west = 3.24
#
##Subset Belt data
NSdata <- subset(SWTfinal, Lat < belt.north)
NSdata <- subset(NSdata, Lat > belt.south)
NSdata <- subset(NSdata, Long < belt.east)
NSdata <- subset(NSdata, Long > belt.west)
summary(NSdata)
```

F14 Create columns indicating point color, shape and size

```{r 3dPlotCreateColourColumns}
NSdata$pcolour[NSdata$Source==3] <- "blue"
NSdata$pcolour[NSdata$Source==4] <- "gray64"
NSdata$pcolour[NSdata$Source==8] <- PointColour
NSdata$pcolour[NSdata$Source==9] <- "black"
NSdata$pcolour[NSdata$Source==11] <- "magenta"
NSdata$pcolour[NSdata$Source==15] <- "orange"
NSdata$pcolour[NSdata$Source==16] <- "green"
NSdata$pcolour[NSdata$Source==17] <- "turquoise3"
#
NSdata$ppch[NSdata$Source==3] <- 20
NSdata$ppch[NSdata$Source==4] <- 20
NSdata$ppch[NSdata$Source==8] <- 20
NSdata$ppch[NSdata$Source==9] <- 8
NSdata$ppch[NSdata$Source==11] <- 17
NSdata$ppch[NSdata$Source==15] <- 20
NSdata$ppch[NSdata$Source==16] <- 20
NSdata$ppch[NSdata$Source==17] <- 20
#
NSdata$pcex[NSdata$Source==3] <- 0.1
NSdata$pcex[NSdata$Source==4] <- 0.1
````

````
NSdata$pcex[NSdata$Source==8] <- 0.1
NSdata$pcex[NSdata$Source==9] <- 1
NSdata$pcex[NSdata$Source==11] <- 0.5
NSdata$pcex[NSdata$Source==15] <- 0.1
NSdata$pcex[NSdata$Source==16] <- 0.1
NSdata$pcex[NSdata$Source==17] <- 0.1
```

## F14 Plot the data

```{r Fig143dScatterplotSNSbelt}
pts <-
  data.frame(x = NSdata$Long,
             y = NSdata$Lat,
             z = NSdata$Depth* -1)
# pts$ppch <- as.numeric(pts$ppch) # needed as was 'chr' in development stages
pts$pcolour <- NSdata$pcolour
pts$ppch <- NSdata$ppch
pts$pcex <- NSdata$pcex
#
ppi=300
Figure.width = 16/2.54
Figure.height = 10/2.54 # height increased to show more tC detail
png("fig14.png", type = "cairo", width = Figure.width, height = Figure.height, units = "in", pointsize = 12, res = ppi)
#
with(pts, {
     s3d <- scatterplot3d(pts$x,pts$y,pts$z,
              color=pts$pcolour, pch=pts$ppch, cex.symbols=pts$pcex, type="p",
              scale.y=1,
              scale.x=1,
              cex.lab=0.75,
              cex.axis=0.5,
              x.ticklabs = c("3.24°E","3.25°E","3.26°E","3.27°E","3.28°E","3.29°E"),
              xlab="Longitude",
              y.ticklabs = c("54.3°N","54.4°N","54.5°N","54.6°N","54.7°N"," "),
              ylab="Latitude",
              z.ticklabs = c(50,40,30,20,10,0),
              zlab="Depth (m)")
              #add legend
              legend("topright", bty="y",cex=0.5,"Data by Source", c("Source = 3","Source = 4","Source = 8","Source = 9","Source = 11","Source = 15","Source = 16","Source = 17"), fill = c("blue","gray64", PointColour, "black", "magenta", "orange", "green", "turquoise3"))
              })
dev.off()
```

# FIGURE 15
**UKCS map and Data Availablity - Surface and Mid water**

## F15 Subset the data

```{r Fig15data}
#restrict to UKCS
ukcs <- subset(SWTfinal, Lat < ukcs.north)
ukcs <- subset(ukcs, Lat > ukcs.south)
ukcs <- subset(ukcs, Long < ukcs.east)
ukcs <- subset(ukcs, Long > ukcs.west)
#
# restrict by depth
Dupperdepth.threshold=20
Dlowerdepth.threshold=25
#
ukcs.surface <- subset(ukcs,Depth <=lowerdepth.threshold)
ukcs.mid <- subset(ukcs,Depth >=Dupperdepth.threshold & Depth <=Dlowerdepth.threshold)
```

## F15 Plot the data

```{r Fig15CreateLabels}
ewbrks <- seq(-15,10,5)
nsbrks <- seq(48,63,2)
````

````
ewlbls <- unlist(lapply(ewbrks, function(x) ifelse(x < 0, paste(x*-1, "°W"), ifelse(x > 0, paste(x, "°E"),x))))
nslbls <- unlist(lapply(nsbrks, function(x) ifelse(x < 0, paste(x*-1, "°S"), ifelse(x > 0, paste(x, "°N"),x))))
```

```{r Fig15plotFile}
ppi=300
Figure.width = 16/2.54
Figure.height = 16/2.54
#
autoplot(map.ukcs, geom="contour",colour="gray", size=0.1) +
  scale_x_continuous(limits = c(ukcs.west, ukcs.east), breaks=ewbrks, labels=ewlbls, expand = c(0, 0)) +
scale_y_continuous(limits = c(ukcs.south, ukcs.north), breaks=nsbrks, labels=nslbls, expand = c(0, 0)) +
coord_map(xlim=c(ukcs.west,ukcs.east),ylim=c(ukcs.south,ukcs.north)) +
  theme(legend.position = "none") +
  theme(panel.background = element_rect(fill = "lightsteelblue1")) +
  geom_point(aes(x=Long,y=Lat), data = ukcs.surface, col=PointColour, size=0.1, alpha = 1/10) +
  geom_point(aes(x=Long,y=Lat), data = ukcs.mid, col="blue", size=0.1, alpha = 1/10) +
  labs(x="Longitude", y="Latitude") +
  theme(axis.title = element_text(size = 12)) +
  theme(axis.text.x = element_text(size=12), axis.text.y = element_text(size=12)) +
  theme(axis.text = element_text(size=12)) +
  scale_x_continuous(breaks = ewbrks, labels = ewlbls, expand = c(0, 0)) +
  scale_y_continuous(breaks = nsbrks, labels = nslbls, expand = c(0, 0)) +
  geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=BEL) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=CHE) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DNK) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=DEU) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=FRA) +
  geom_polygon(aes(x=long,y=lat, group=group),fill="darkgreen",data=FRO) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=GBR) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IMN) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=IRL) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LIE) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=LUX) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NLD) +
  geom_polygon(aes(x=long,y=lat,group=group),fill="darkgreen",data=NOR)
  #
  ggsave("fig15.png", width=Figure.width, height = Figure.height, dpi=ppi)
```

# FIGURE 16
**UKCS Surface and Mid Water Temperature by 14 YEAR periods**

Mid Water data extends from 1970 so that is the time range selected for comparison of surface and mid water data

## F16 Subset the data

```{r Fig16data}
ukcs.surface.7084 <- subset(ukcs.surface, Time >= "1970-01-01 00:00:00" & Time <= "1984-12-31 23:59:59")
ukcs.surface.8599 <- subset(ukcs.surface, Time >= "1985-01-01 00:00:00" & Time <= "1999-12-31 23:59:59")
ukcs.surface.0015 <- subset(ukcs.surface, Time >= "2000-01-01 00:00:00" & Time <= "2015-12-31 23:59:59")
#
ukcs.mid.7084 <- subset(ukcs.mid, Time >= "1970-01-01 00:00:00" & Time <= "1984-12-31 23:59:59")
ukcs.mid.8599 <- subset(ukcs.mid, Time >= "1985-01-01 00:00:00" & Time <= "1999-12-31 23:59:59")
ukcs.mid.0015 <- subset(ukcs.mid, Time >= "2000-01-01 00:00:00" & Time <= "2015-12-31 23:59:59")
#
```

## F16 Plot the data

```{r Fig16plotFile}
````

````
ppi=300
Figure.width = 16/2.54
Figure.height = 16/2.54
png("fig16.png", type = "cairo", width = Figure.width, height = Figure.height, units = "in", pointsize = 12, res = ppi)
#
p1 <- ggplot(data = ukcs.surface.7084, aes(x=Time, y=tC)) +
  geom_point(color=PointColour,size=0.1) +
  geom_point(data=ukcs.mid.7084, aes(x=Time, y=tC), colour = "blue",size=0.5)+
  labs(x="Year", y="Temperature °C") +
  ggtitle("(a)") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(-2, 32),breaks = seq(0,25,5)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
#
p2 <- ggplot(data = ukcs.surface.8599, aes(x=Time, y=tC)) +
  geom_point(color=PointColour,size=0.1) +
  geom_point(data=ukcs.mid.8599, aes(x=Time, y=tC), colour = "blue",size=0.5)+
  labs(x="Year", y="Temperature °C") +
  ggtitle("(b)") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(-2, 32),breaks = seq(0,25,5)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
#
p3 <- ggplot(data = ukcs.surface.0015, aes(x=Time, y=tC)) +
  geom_point(color=PointColour,size=0.1) +
  geom_point(data=ukcs.mid.0015, aes(x=Time, y=tC), colour = "blue",size=0.5)+
  labs(x="Year", y="Temperature °C") +
  ggtitle("(c)") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(-2, 32),breaks = seq(0,25,5)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
#
multiplot(p1,p2,p3, cols=1)
dev.off()
```

# FIGURE 17
**UKCS Monthly Average Temperature .v. Year - SURFACE**

## F17 Calculate Monthly averages

```{r Fig17data}
#
#http://stackoverflow.com/questions/6052631/aggregate-daily-data-to-month-year-in-r
#
# SURFACE
#
Mon.avg <- setDT(ukcs.surface)[, .(MonthlyMeans = mean(tC)), by = .(year(Time), month(Time))]
#
# to see counts for means
# Mon.count <- setDT(ukcs.surface)[, .(COUNT = .N), by = .(year(Time), month(Time))]
#
#convert to dates
Mon.avg$year <- as.character(Mon.avg$year)
Mon.avg$month <- as.character(Mon.avg$month)
#combine chr
Mon.avg$YearMon <- paste(Mon.avg$year, Mon.avg$month, sep = "-0")
# replace 010 011 and 012 with 10 11 12
Mon.avg$YearMon <- gsub('-010', '-10',Mon.avg$YearMon)
Mon.avg$YearMon <- gsub('-011', '-11',Mon.avg$YearMon)
Mon.avg$YearMon <- gsub('-012', '-12',Mon.avg$YearMon)
#convert to date - add day 01
Mon.avg$YearMon <- as.Date(paste(Mon.avg$YearMon,"-01",sep=""))
#
Mon.avg.SURFACE <- Mon.avg
#
# MID WATER
#
````

```
Mon.avg <- setDT(ukcs.mid)[, .(MonthlyMeans = mean(tC)), by = .(year(Time), month(Time))]
#
#Mon.count <- setDT(ukcs.mid)[, .(COUNT = .N), by = .(year(Time), month(Time))]
#
#convert to dates
Mon.avg$year <- as.character(Mon.avg$year)
Mon.avg$month <- as.character(Mon.avg$month)
#combine chr
Mon.avg$YearMon <- paste(Mon.avg$year, Mon.avg$month, sep = "-0")
# replace 010 011 and 012 with 10 11 12
Mon.avg$YearMon <- gsub('-010', '-10',Mon.avg$YearMon)
Mon.avg$YearMon <- gsub('-011', '-11',Mon.avg$YearMon)
Mon.avg$YearMon <- gsub('-012', '-12',Mon.avg$YearMon)
#convert to date - add day 01
Mon.avg$YearMon <- as.Date(paste(Mon.avg$YearMon,"-01",sep=""))
#
Mon.avg.MID <- Mon.avg
```

F17 Subset the data

```{r Fig17subsetData}
JanS.avg <- subset(Mon.avg.SURFACE,month == "1")
JanM.avg <- subset(Mon.avg.MID,month == "1")
JanS.avg$year <- as.numeric(JanS.avg$year)
JanM.avg$year <- as.numeric(JanM.avg$year)
#
FebS.avg <- subset(Mon.avg.SURFACE,month == "2")
FebM.avg <- subset(Mon.avg.MID,month == "2")
FebS.avg$year <- as.numeric(FebS.avg$year)
FebM.avg$year <- as.numeric(FebM.avg$year)
#
MarS.avg <- subset(Mon.avg.SURFACE,month == "3")
MarM.avg <- subset(Mon.avg.MID,month == "3")
MarS.avg$year <- as.numeric(MarS.avg$year)
MarM.avg$year <- as.numeric(MarM.avg$year)
#
AprS.avg <- subset(Mon.avg.SURFACE,month == "4")
AprM.avg <- subset(Mon.avg.MID,month == "4")
AprS.avg$year <- as.numeric(AprS.avg$year)
AprM.avg$year <- as.numeric(AprM.avg$year)
#
MayS.avg <- subset(Mon.avg.SURFACE,month == "5")
MayM.avg <- subset(Mon.avg.MID,month == "5")
MayS.avg$year <- as.numeric(MayS.avg$year)
MayM.avg$year <- as.numeric(MayM.avg$year)
#
JunS.avg <- subset(Mon.avg.SURFACE,month == "6")
JunM.avg <- subset(Mon.avg.MID,month == "6")
JunS.avg$year <- as.numeric(JunS.avg$year)
JunM.avg$year <- as.numeric(JunM.avg$year)
#
JulS.avg <- subset(Mon.avg.SURFACE,month == "7")
JulM.avg <- subset(Mon.avg.MID,month == "7")
JulS.avg$year <- as.numeric(JulS.avg$year)
JulM.avg$year <- as.numeric(JulM.avg$year)
#
AugS.avg <- subset(Mon.avg.SURFACE,month == "8")
AugM.avg <- subset(Mon.avg.MID,month == "8")
AugS.avg$year <- as.numeric(AugS.avg$year)
AugM.avg$year <- as.numeric(AugM.avg$year)
#
SepS.avg <- subset(Mon.avg.SURFACE,month == "9")
SepM.avg <- subset(Mon.avg.MID,month == "9")
SepS.avg$year <- as.numeric(SepS.avg$year)
SepM.avg$year <- as.numeric(SepM.avg$year)
#
OctS.avg <- subset(Mon.avg.SURFACE,month == "10")
OctM.avg <- subset(Mon.avg.MID,month == "10")
OctS.avg$year <- as.numeric(OctS.avg$year)
OctM.avg$year <- as.numeric(OctM.avg$year)
#
```

````
NovS.avg <- subset(Mon.avg.SURFACE,month == "11")
NovM.avg <- subset(Mon.avg.MID,month == "11")
NovS.avg$year <- as.numeric(NovS.avg$year)
NovM.avg$year <- as.numeric(NovM.avg$year)
#
DecS.avg <- subset(Mon.avg.SURFACE,month == "12")
DecM.avg <- subset(Mon.avg.MID,month == "12")
DecS.avg$year <- as.numeric(DecS.avg$year)
DecM.avg$year <- as.numeric(DecM.avg$year)
```

## F17 Plot the data

```{r Fig17plotFile}
# lowess
#https://ww2.coastal.edu/kingw/statistics/R-tutorials/simplelinear.html
#
PointSize = 0.5
ppi=300
Figure.width = 16/2.54
Figure.height = 24/2.54
# size increased to allow for margins - 16/10 gave plot error margins too small
png("fig17.png", type = "cairo", width = Figure.width, height = Figure.height, units = "in", pointsize = 12, res = ppi)
#
p1 <- ggplot(data = JanS.avg, aes(x=year, y=MonthlyMeans)) +
  geom_point(color=PointColour, size = PointSize) +
  geom_point(data=JanM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size = PointSize) +
    labs(x="Year", y="Temperature °C") +
  ggtitle("Jan") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(0, 20)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
p2 <- ggplot(data = FebS.avg, aes(x=year, y=MonthlyMeans)) +
  geom_point(color=PointColour, size = PointSize) +
  geom_point(data=FebM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size = PointSize) +
    labs(x="Year", y="Temperature °C") +
  ggtitle("Feb") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(0, 20)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
p3 <- ggplot(data = MarS.avg, aes(x=year, y=MonthlyMeans)) +
  geom_point(color=PointColour, size = PointSize) +
  geom_point(data=MarM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size = PointSize) +
    labs(x="Year", y="Temperature °C") +
  ggtitle("Mar") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(0, 20)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
p4 <- ggplot(data = AprS.avg, aes(x=year, y=MonthlyMeans)) +
  geom_point(color=PointColour, size = PointSize) +
  geom_point(data=AprM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size = PointSize) +
    labs(x="Year", y="Temperature °C") +
  ggtitle("Apr") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(0, 20)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
p5 <- ggplot(data = MayS.avg, aes(x=year, y=MonthlyMeans)) +
  geom_point(color=PointColour, size = PointSize) +
  geom_point(data=MayM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size = PointSize) +
    labs(x="Year", y="Temperature °C") +
  ggtitle("May") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(0, 20)) +
````

```
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
p6 <- ggplot(data = JunS.avg, aes(x=year, y=MonthlyMeans)) +
  geom_point(color=PointColour, size = PointSize) +
  geom_point(data=JunM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size = PointSize) +
    labs(x="Year", y="Temperature °C") +
  ggtitle("Jun") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(0, 20)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
p7 <- ggplot(data = JulS.avg, aes(x=year, y=MonthlyMeans)) +
  geom_point(color=PointColour, size = PointSize) +
  geom_point(data=JulM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size = PointSize) +
    labs(x="Year", y="Temperature °C") +
  ggtitle("Jul") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(0, 20)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
p8 <- ggplot(data = AugS.avg, aes(x=year, y=MonthlyMeans)) +
  geom_point(color=PointColour, size = PointSize) +
  geom_point(data=AugM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size = PointSize) +
    labs(x="Year", y="Temperature °C") +
  ggtitle("Aug") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(0, 20)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
p9 <- ggplot(data = SepS.avg, aes(x=year, y=MonthlyMeans)) +
  geom_point(color=PointColour, size = PointSize) +
  geom_point(data=SepM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size = PointSize) +
    labs(x="Year", y="Temperature °C") +
  ggtitle("Sep") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(0, 20)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
p10 <- ggplot(data = OctS.avg, aes(x=year, y=MonthlyMeans)) +
  geom_point(color=PointColour, size = PointSize) +
  geom_point(data=OctM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size = PointSize) +
    labs(x="Year", y="Temperature °C") +
  ggtitle("Oct") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(0, 20)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
p11 <- ggplot(data = NovS.avg, aes(x=year, y=MonthlyMeans)) +
  geom_point(color=PointColour, size = PointSize) +
  geom_point(data=NovM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size = PointSize) +
    labs(x="Year", y="Temperature °C") +
  ggtitle("Nov") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(0, 20)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
p12 <- ggplot(data = DecS.avg, aes(x=year, y=MonthlyMeans)) +
  geom_point(color=PointColour, size = PointSize) +
  geom_point(data=DecM.avg, aes(x=year, y=MonthlyMeans), colour = "blue", size = PointSize) +
    labs(x="Year", y="Temperature °C") +
  ggtitle("Dec") +
  theme(plot.title = element_text(size=10, face = "bold")) +
  scale_y_continuous(limits = c(0, 20)) +
  theme(axis.text.x=element_text(angle = 90, hjust = 0)) +
  theme_bw()
#
```

````
multiplot(p1,p3,p5,p7,p9,p11,p2,p4,p6,p8,p10,p12, cols=2)
dev.off()
```

# **END**
````
````