Generating ASMR estimates for North Area BGC units

Hardy Griesbauer

17/09/2019

## Description of choosing climate stations for BGC units in the North Area

Below is R code and description used to select climate station data to generate ASMR estimates for various BGC units in the North Area.

# Load libraries  
library(forestDroughtTool)

## Warning: replacing previous import 'magrittr::extract' by 'tidyr::extract'  
## when loading 'forestDroughtTool'

library(tidyverse)  
library(weathercan)  
library(ggplot2)  
library(here)  
library(magrittr)

## What are the BGC units of interest in the North Area?

What units do we want to generate drought hazard estimates for?

# North Area BGC units are saved in .rda file  
load(here("dat","northBGC.rda"))  
northBGC

## [1] "ICHmc1" "MHmm2" "SBSmc2" "CWHws2" "ICHmc2" "ESSFmv3" "ESSFmv1"  
## [8] "SBSwk3" "SBSdw3" "SBSwk2" "ESSFmv2" "ESSFmv4" "ESSFwc3" "ESSFwk2"  
## [15] "BWBSwk2" "SBSmk1" "SBSmk2" "SBSwk1" "SBSmc3" "ESSFxv1" "SBSdw1"   
## [22] "SBSdw2" "ICHvk2" "ICHwk3" "ICHwk4" "ESSFwk1" "ESSFmm1" "IDFdk3"   
## [29] "SBSmc1"

## What ECCC climate stations exist for these units?

Using a bunch of different packages, we can query the ECCC climate station data and see which ones are located in the BGC units of interest.

# DON'T RUN - Takes a long time, instead use load()  
# # Step 1 - Select stations in BC with daily data and assign to 'stn'  
# stn<-  
# weathercan::stations %>%   
# dplyr::filter(prov=="BC" & interval=="day") %>%   
#   
# # Step 2 - Convert to a spatial file and merge with BGC units (this will take awhile!)  
# sf::st\_as\_sf(coords=c("lon","lat")) %>% # convert to spatial file  
# sf::st\_set\_crs(4326) %>% # set to WGS1984 datum  
# bcmaps::transform\_bc\_albers() %>% # set to BC albers projection  
# sf::st\_join(bcmaps::bec()[,"MAP\_LABEL"]) %>% # merge with BEC  
# sf::st\_drop\_geometry() %>% # drop geometry  
#   
# # Step 3 - Merge with BGC units of interest (above)  
# dplyr::filter(MAP\_LABEL%in%northBGC) %>% # filter for BGC units of interest  
# dplyr::mutate(length=end-start+1) %>% # create a column to show record length  
# dplyr::arrange(MAP\_LABEL,desc(length)) %>%   
# dplyr::select(station\_name,station\_id,bgc=MAP\_LABEL,start,end,length)  
#   
# # step 4 save stn so you don't have to run above lines of code  
# save(stn,file=here("dat","stationList.rda"))  
  
# Load station data compiled from above lines of code  
 load(here("dat","stationList.rda"))  
  
# Print to screen  
 stn

## # A tibble: 154 x 6  
## station\_name station\_id bgc start end length  
## <chr> <fct> <chr> <int> <int> <dbl>  
## 1 PINK MOUNTAIN 1432 BWBSwk2 1973 1982 10  
## 2 PINK MOUNTAIN 2 1430 BWBSwk2 1982 1990 9  
## 3 BUCKINGHORSE RIVER 1450 BWBSwk2 1973 1977 5  
## 4 TAHTSA LAKE WEST 547 CWHws2 1951 2000 50  
## 5 ALICE ARM 375 CWHws2 1948 1964 17  
## 6 KITSAULT MINESITE 474 CWHws2 1969 1972 4  
## 7 KITSAULT MINESITE 405 CWHws2 1968 1969 2  
## 8 CRANBERRY RIVER 460 CWHws2 1973 1974 2  
## 9 NASS RIVER 483 CWHws2 1956 1956 1  
## 10 FIRVALE 6941 CWHws2 1991 1991 1  
## # ... with 144 more rows

## Download climate data

Need to download climate data for stations. This will take a long time, so I saved the downloaded data into an .rda file.

* Note that in the future we will want to update some of the station data with cleaned data from Vanessa. This will take me some time!\*

I printed out the station list, and manually selected stations (not shown) that have long records that span the 1961-1990 climate normal period, as much as possible. In some cases, we may want to join records.

# Row numbers of stations of interest  
stnRow=c(1, 2, 4, 11, 12, 15, 17, 21, 22, 36, 37, 39, 66, 67,  
 72, 73, 85, 86, 87, 93, 108, 109, 123, 137, 143,  
 144, 149, 150, 151, 153)  
  
# Don't run - takes a long time. Use load() instead  
# Download data using the stnRow (this can take a long time!)  
# climData<-  
# weather\_dl(station\_ids=stn$station\_id[stnRow],interval="day") %>%  
#   
# # format climate data (see below for more information)  
# select(stn=station\_name,date,tmn=min\_temp,tmx=max\_temp,ppt=total\_precip,year,month,day)  
  
# save(climData,file=here("dat","climData.rda"))  
  
load(here("dat","climData.rda"))

## Clean climate station data

We need to process missing values in the daily climate data. For this project, we: - omitted any years with >10 consecutive missing data in any climate variable;and - imputed missing data using adjacent values (closest data before and after missing value) - Leap years have 366 days, so we removed February 29 from those years to keep number of days to 365.

# DON'T RUN - this takes a long time, use load() instead  
  
# cleanECData() for each station  
# x1<-by(INDICES=climData$stn,function(x) cleanECData(x),data=climData)   
#   
# # Formatting  
# climData\_cleaned<-dplyr::bind\_rows(x1,.id="id")   
# climData\_cleaned$stn<-names(x1)[as.numeric(climData\_cleaned$id)]  
# rm(x1)  
  
# # Format date columns  
# climData\_cleaned%<>%  
# mutate(month=as.integer(month)) %>%   
# mutate(day=as.integer(day)) %>%   
# mutate(year=as.integer(year)) %>%   
# mutate(date=paste(year,month,day,sep="-"))  
  
# Rename columns to make it easier to pass to asmrCalc()  
# climData\_cleaned%<>%  
# rename(!!'tmn':=tmn\_filled,!!'ppt':=ppt\_filled,!!'tmx':=tmx\_filled)  
  
# save(climData\_cleaned,file=here("dat","climData\_cleaned.rda"))  
  
# Load cleaned EC climate data from above lines of code  
load(here("dat","climData\_cleaned.rda"))  
  
# Summarize station data   
climData\_cleaned %>%  
 group\_by(stn,year) %>%  
 summarise(n()) %>%  
 ungroup() %>%  
 group\_by(stn) %>%  
 summarise(Num.years=n(),start=min(year),end=max(year))

## Warning: Grouping rowwise data frame strips rowwise nature

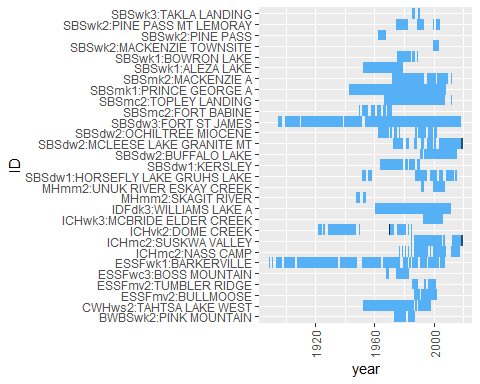
## # A tibble: 30 x 4  
## stn Num.years start end  
## <chr> <int> <int> <int>  
## 1 ALEZA LAKE 27 1953 1979  
## 2 BARKERVILLE 102 1889 2007  
## 3 BLUE RIVER 39 1947 1985  
## 4 BOSS MOUNTAIN 11 1968 1983  
## 5 BOWRON LAKE 12 1976 1989  
## 6 BUFFALO LAKE 24 1991 2015  
## 7 BULLMOOSE 15 1987 2002  
## 8 DOME CREEK 36 1922 1985  
## 9 FORT BABINE 16 1950 1971  
## 10 FORT ST JAMES 118 1895 2018  
## # ... with 20 more rows

## Figure showing data coverage for each station

It might be useful to produce a figure showing climate data coverage for each station and BGC unit.

climData\_cleaned %>%   
 group\_by(stn,year) %>%   
 summarise(days=n()) %>%   
 inner\_join(stn,by=c("stn"="station\_name")) %>%   
 select(stn,year,days,bgc) %>%   
 ungroup() %>%   
 mutate(ID=paste(bgc,stn,sep=":")) %>%   
 ggplot(aes(year, ID)) +  
 geom\_tile(aes(fill = days))+  
 theme(legend.position = "none")+  
 theme(axis.text.x = element\_text(angle = 90,vjust = 0.5))

## Warning: Grouping rowwise data frame strips rowwise nature



## Select stations and climate data to run model

Generally, we have been summarising site ASMR using 10 years of daily climate data. The rationale for this is that we want to keep data consistent between BGC units, and we need to be able to use stations with relatively sparse data (ie. 10 years).

Based on a conversation with Vanessa, we’ve decided on the following:

1. Restrict data to 1940 and 1990. Data pre-1940 may have measurement errors, and post-1990 is starting to get into the “climate change” realm.
2. Ideally, restrict years to 1961-1990 normal period. If number of years available is less than 10, then use years from 1940-1990.
3. Select 10 years randomly from within the dataset. If there are less than 10 years of data within 1961-1990 dataset, then use all the years from 1961-1990, and add random years’ data from 1940-1960 until number of years = 10.

This is all taken care of in the asmrSelect() function in the forestDroughtTool package.

# Need to change Pink Mountain2 to Pink Mountain, to concatenate the datasets  
climData\_cleaned %<>%   
 mutate(stn=replace(stn,stn=="PINK MOUNTAIN 2","PINK MOUNTAIN"))

### Use data for ICHwk1 and ICHvk2

Dome creek and Crescent Spur stations are located near the valley bottoms in the Robson Valley, near the boundary between ICH and SBSvk. Given their proximity to the BGC unit boundary, and based on expert ecologist opinion (ie., Craig and Bruce), we decided to use these stations to represent adjacent ICH variants. Same for Blue River station.

stations %>%   
 filter(str\_detect(station\_name,'DOME CREEK|CRESCENT SPUR|BLUE RIVER')) %>%   
 filter(interval=="day")

## # A tibble: 8 x 14  
## prov station\_name station\_id climate\_id WMO\_id TC\_id lat lon elev  
## <chr> <chr> <int> <fct> <int> <fct> <dbl> <dbl> <dbl>  
## 1 BC DOME CREEK 580 1092519 NA <NA> 53.8 -121. 671.  
## 2 BC DOME CREEK 581 1092520 NA <NA> 53.7 -121. 648.  
## 3 BC BLUE RIVER A 1237 1160899 NA YCP 52.1 -119. 690.  
## 4 BC BLUE RIVER ~ 1238 1160900 NA <NA> 52.2 -119. 689.  
## 5 BC BLUE RIVER ~ 8214 1160H99 71883 WSV 52.1 -119. 683.  
## 6 BC CRESCENT SP~ 26989 1092120 NA <NA> 53.6 -121. 686   
## 7 BC BLUE RIVER A 53423 1160898 NA YCP 52.1 -119. 690.  
## 8 BC BLUE RIVER A 53424 1160902 NA YCP 52.1 -119. 690.  
## # ... with 5 more variables: tz <chr>, interval <chr>, start <int>,  
## # end <int>, normals <lgl>

Blue River North (station id = 1238) and Dome Creek (station id = 581) look to have good data. Crescent Spur only has data from 1992 - so we won’t use this station for now.

Blue River A does not have adequate data!

Next step is to extract clean/data for Dome Creek and Blue River North and compile it with the rest of the climate dataset.

# Don't run this -takes a long time. Use load() instead   
# climData2<-  
# weather\_dl(station\_ids=c(581,1238), interval="day") %>%   
# select(stn=station\_name,date,tmn=min\_temp,tmx=max\_temp,ppt=total\_precip,year,month,day)  
  
# cleanECData() for each station  
  
# dome<-data.frame(stn="DOME CREEK",cleanECData(filter(climData2,stn=="DOME CREEK"))) %>%   
# mutate(bgc="ICHvk2") %>%   
# mutate(stn\_id=581) %>%   
# mutate(id=31)  
#   
# blue<-data.frame(stn="BLUE RIVER",cleanECData(filter(climData2,stn=="BLUE RIVER NORTH"))) %>%   
# mutate(bgc="ICHwk1") %>%   
# mutate(stn\_id=1238) %>%   
# mutate(id=32)  
#   
# x1<-rbind(blue,dome)  
#   
# # Format date columns  
# x1%<>%  
# mutate(month=as.integer(month)) %>%   
# mutate(day=as.integer(day)) %>%   
# mutate(year=as.integer(year)) %>%   
# mutate(date=paste(year,month,day,sep="-")) %>%   
#   
# # Rename columns to make it easier to pass to asmrCalc()  
# rename(!!'tmn':=tmn\_filled,!!'ppt':=ppt\_filled,!!'tmx':=tmx\_filled) %>%   
#   
# # select columns  
# dplyr::select(names(climData\_cleaned))  
#   
#   
# # Merge with climData\_cleaned  
# climData\_cleaned=rbind(climData\_cleaned,x1)  
# save(climData\_cleaned,file=here::here("dat","climData\_cleaned.rda"))  
   
# Designate stations to use:  
stnList=c(1432,547,1397,569,568,482,664,601,623,588,496,631,1423,564,1429,1238,581)  
  
# Bring in Dome Creek and Blue River station data as well  
stnID<-  
 stations %>%   
 filter(station\_id==581|station\_id==1238) %>%   
 filter(interval=="day") %>%   
 mutate(bgc="ICHwk1") %>%   
 mutate(bgc=replace(bgc,station\_id==581,"ICHvk2")) %>%  
 mutate(station\_name=replace(station\_name,station\_id==1238,"BLUE RIVER")) %>%   
 mutate(length=end-start+1) %>%   
 select(station\_name,station\_id,bgc,start,end,length) %>%   
 rbind(stn[stn$station\_id%in%stnList,]) # rbind with the rest of the stations  
   
  
# Print out stations used in ASMR  
stnID

## # A tibble: 17 x 6  
## station\_name station\_id bgc start end length  
## <chr> <chr> <chr> <int> <int> <dbl>  
## 1 DOME CREEK 581 ICHvk2 1970 1995 26  
## 2 BLUE RIVER 1238 ICHwk1 1929 1985 57  
## 3 PINK MOUNTAIN 1432 BWBSwk2 1973 1982 10  
## 4 TAHTSA LAKE WEST 547 CWHws2 1951 2000 50  
## 5 BULLMOOSE 1397 ESSFmv2 1982 2003 22  
## 6 BOSS MOUNTAIN 569 ESSFwc3 1967 1984 18  
## 7 BARKERVILLE 568 ESSFwk1 1888 2015 128  
## 8 NASS CAMP 482 ICHmc2 1973 2017 45  
## 9 WILLIAMS LAKE A 664 IDFdk3 1960 2012 53  
## 10 KERSLEY 601 SBSdw1 1962 1992 31  
## 11 OCHILTREE MIOCENE 623 SBSdw2 1963 2003 41  
## 12 FORT ST JAMES 588 SBSdw3 1895 2018 124  
## 13 TOPLEY LANDING 496 SBSmc2 1962 2017 56  
## 14 PRINCE GEORGE A 631 SBSmk1 1942 2009 68  
## 15 MACKENZIE A 1423 SBSmk2 1971 2013 43  
## 16 BOWRON LAKE 564 SBSwk1 1974 1990 17  
## 17 PINE PASS MT LEMORAY 1429 SBSwk2 1974 2004 31

## Calculate ASMR for the stations

Now let’s run a loop to generate the dataset and export it.

# Don't run this takes a long time, use load () instead  
# for (j in 1:nrow(stnID)) {  
#   
# # Get ASMR summaries for the station  
# x<-  
# climData\_cleaned %>%  
# filter(stn==stnID$station\_name[j]) %>%  
# asmrCalc() %>%  
# asmrSelect()  
#   
# # Extract ASMR mean and SD  
# asmr<-  
# x$asmr %>%  
# mutate(bgc=stnID$bgc[j]) %>%  
# dplyr::select(bgc,month,everything())  
#   
# # Extract years used to calculate ASMR  
# yearASMR<-  
# c(stnID$bgc[j],x$years)  
#   
# # Compile asmr into data frame  
# if(j==1) {asmrNorth2=asmr} else {asmrNorth2=rbind(asmrNorth2,asmr)}  
#   
# # Compile years into data frame  
# if(j==1) {years=yearASMR} else {years=rbind(years,yearASMR)}  
#   
# if (j==nrow(stnID)) {  
#   
# years=as.data.frame(years)  
# names(years)=c("bgc",paste("year",1:10,sep=""))  
# years<-as\_tibble(years)  
#   
# }  
#   
# }  
#   
# save(asmrNorth2,file=here::here("dat","asmrNorth2.rda"))  
# save(years,file=here::here("dat","years.rda"))  
  
load(here::here("dat","asmrNorth2.rda"))  
load(here::here("dat","years.rda"))

# ANALYSIS

## Generate annual ASMR estimates by BGC unit

# Generate annual AET/PET for five sites by BGC unit  
asmrANNUAL<-  
 asmrNorth2 %>%   
 dplyr::select(bgc,month,contains(".mean")) %>%   
 group\_by(bgc) %>%   
 summarise\_at(vars(contains(".mean")),mean)   
  
# Print asmrANNUAL  
asmrANNUAL

## # A tibble: 17 x 6  
## bgc S1.mean S2.mean S3.mean S4.mean S5.mean  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 BWBSwk2 0.829 0.912 0.953 0.996 1  
## 2 CWHws2 1 1 1 1 1  
## 3 ESSFmv2 0.900 0.947 0.970 0.993 1  
## 4 ESSFwc3 1 1 1 1 1  
## 5 ESSFwk1 0.968 0.990 0.996 1 1  
## 6 ICHmc2 0.870 0.914 0.939 0.969 1  
## 7 ICHvk2 0.912 0.960 0.978 1.000 1  
## 8 ICHwk1 0.919 0.964 0.979 0.994 1  
## 9 IDFdk3 0.665 0.734 0.768 0.822 1  
## 10 SBSdw1 0.742 0.809 0.838 0.888 1  
## 11 SBSdw2 0.805 0.863 0.888 0.934 1  
## 12 SBSdw3 0.783 0.830 0.854 0.904 1  
## 13 SBSmc2 0.818 0.869 0.887 0.936 1  
## 14 SBSmk1 0.795 0.844 0.874 0.927 1  
## 15 SBSmk2 0.856 0.911 0.938 0.974 1  
## 16 SBSwk1 0.890 0.948 0.970 0.992 1  
## 17 SBSwk2 0.861 0.923 0.957 0.972 1

## BGC units without climate data

For the following BGC units that don’t have climate data, can we lump any BGC units together (i.e., create ASMR estimates for one BGC unit using ASMR from another):

northBGC[!northBGC%in%asmrANNUAL$bgc]

## [1] "ICHmc1" "MHmm2" "ESSFmv3" "ESSFmv1" "SBSwk3" "ESSFmv4" "ESSFwk2"  
## [8] "SBSmc3" "ESSFxv1" "ICHwk3" "ICHwk4" "ESSFmm1" "SBSmc1"

### Amalgamate drought estimates by subzone (not variant) for variants missing climate data

Create a list of BGC units that have ASMR data that we can use to provide estimates for climate-missing BGC units:

# Create a function to assign subzones to their respective variant with climate data  
  
# bgcMissing is the BGC unit that is missing climate data  
# bgcComplete is the BGC unit that you want to use the ASMR values for bgcMissing  
  
bgc.subF<-function(bgcMissing,bgcComplete) {   
   
 return(data.frame(bgc=bgcMissing,  
 asmrANNUAL[asmrANNUAL$bgc%in%bgcComplete,2:ncol(asmrANNUAL)]  
 ))  
 }  
  
# Now run the function for BGC units missing data  
  
asmrANNUAL2<-  
 bgc.subF("SBSwk3","SBSwk1") %>%   
 rbind(bgc.subF("ESSFmv4","ESSFmv2")) %>%   
 rbind(bgc.subF("ESSFmv3","ESSFmv2")) %>%   
 rbind(bgc.subF("ESSFmv1","ESSFmv2")) %>%   
 rbind(bgc.subF("SBSmc1","SBSmc2")) %>%   
 rbind(bgc.subF("SBSmc3","SBSmc2")) %>%   
 rbind(bgc.subF("ICHmc1","ICHmc2")) %>%  
 rbind(bgc.subF("ESSFwk2","ESSFwk1")) %>%   
 rbind(bgc.subF("ICHwk3","ICHwk1")) %>%   
 rbind(bgc.subF("ICHwk4","ICHwk1")) %>%   
 rbind(asmrANNUAL)  
   
# Print results to document  
asmrANNUAL2

## bgc S1.mean S2.mean S3.mean S4.mean S5.mean  
## 1 SBSwk3 0.8895833 0.9485000 0.9699167 0.9922500 1  
## 2 ESSFmv4 0.8999167 0.9473333 0.9695833 0.9930833 1  
## 3 ESSFmv3 0.8999167 0.9473333 0.9695833 0.9930833 1  
## 4 ESSFmv1 0.8999167 0.9473333 0.9695833 0.9930833 1  
## 5 SBSmc1 0.8177500 0.8686667 0.8871667 0.9362500 1  
## 6 SBSmc3 0.8177500 0.8686667 0.8871667 0.9362500 1  
## 7 ICHmc1 0.8695833 0.9141667 0.9389167 0.9690833 1  
## 8 ESSFwk2 0.9679167 0.9902500 0.9958333 1.0000000 1  
## 9 ICHwk3 0.9186667 0.9640000 0.9791667 0.9938333 1  
## 10 ICHwk4 0.9186667 0.9640000 0.9791667 0.9938333 1  
## 11 BWBSwk2 0.8290833 0.9121667 0.9532500 0.9960000 1  
## 12 CWHws2 1.0000000 1.0000000 1.0000000 1.0000000 1  
## 13 ESSFmv2 0.8999167 0.9473333 0.9695833 0.9930833 1  
## 14 ESSFwc3 1.0000000 1.0000000 1.0000000 1.0000000 1  
## 15 ESSFwk1 0.9679167 0.9902500 0.9958333 1.0000000 1  
## 16 ICHmc2 0.8695833 0.9141667 0.9389167 0.9690833 1  
## 17 ICHvk2 0.9116667 0.9597500 0.9780833 0.9995000 1  
## 18 ICHwk1 0.9186667 0.9640000 0.9791667 0.9938333 1  
## 19 IDFdk3 0.6645833 0.7335833 0.7675833 0.8218333 1  
## 20 SBSdw1 0.7423333 0.8088333 0.8380833 0.8878333 1  
## 21 SBSdw2 0.8050000 0.8628333 0.8875000 0.9343333 1  
## 22 SBSdw3 0.7834167 0.8295000 0.8538333 0.9036667 1  
## 23 SBSmc2 0.8177500 0.8686667 0.8871667 0.9362500 1  
## 24 SBSmk1 0.7950833 0.8441667 0.8745000 0.9270833 1  
## 25 SBSmk2 0.8556667 0.9113333 0.9383333 0.9742500 1  
## 26 SBSwk1 0.8895833 0.9485000 0.9699167 0.9922500 1  
## 27 SBSwk2 0.8605833 0.9230000 0.9565833 0.9722500 1

Now let’s see which bgc units are missing data:

northBGC[!northBGC%in%asmrANNUAL2$bgc]

## [1] "MHmm2" "ESSFxv1" "ESSFmm1"

knitr::kable(asmrANNUAL2,digits=2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| bgc | S1.mean | S2.mean | S3.mean | S4.mean | S5.mean |
| SBSwk3 | 0.89 | 0.95 | 0.97 | 0.99 | 1 |
| ESSFmv4 | 0.90 | 0.95 | 0.97 | 0.99 | 1 |
| ESSFmv3 | 0.90 | 0.95 | 0.97 | 0.99 | 1 |
| ESSFmv1 | 0.90 | 0.95 | 0.97 | 0.99 | 1 |
| SBSmc1 | 0.82 | 0.87 | 0.89 | 0.94 | 1 |
| SBSmc3 | 0.82 | 0.87 | 0.89 | 0.94 | 1 |
| ICHmc1 | 0.87 | 0.91 | 0.94 | 0.97 | 1 |
| ESSFwk2 | 0.97 | 0.99 | 1.00 | 1.00 | 1 |
| ICHwk3 | 0.92 | 0.96 | 0.98 | 0.99 | 1 |
| ICHwk4 | 0.92 | 0.96 | 0.98 | 0.99 | 1 |
| BWBSwk2 | 0.83 | 0.91 | 0.95 | 1.00 | 1 |
| CWHws2 | 1.00 | 1.00 | 1.00 | 1.00 | 1 |
| ESSFmv2 | 0.90 | 0.95 | 0.97 | 0.99 | 1 |
| ESSFwc3 | 1.00 | 1.00 | 1.00 | 1.00 | 1 |
| ESSFwk1 | 0.97 | 0.99 | 1.00 | 1.00 | 1 |
| ICHmc2 | 0.87 | 0.91 | 0.94 | 0.97 | 1 |
| ICHvk2 | 0.91 | 0.96 | 0.98 | 1.00 | 1 |
| ICHwk1 | 0.92 | 0.96 | 0.98 | 0.99 | 1 |
| IDFdk3 | 0.66 | 0.73 | 0.77 | 0.82 | 1 |
| SBSdw1 | 0.74 | 0.81 | 0.84 | 0.89 | 1 |
| SBSdw2 | 0.80 | 0.86 | 0.89 | 0.93 | 1 |
| SBSdw3 | 0.78 | 0.83 | 0.85 | 0.90 | 1 |
| SBSmc2 | 0.82 | 0.87 | 0.89 | 0.94 | 1 |
| SBSmk1 | 0.80 | 0.84 | 0.87 | 0.93 | 1 |
| SBSmk2 | 0.86 | 0.91 | 0.94 | 0.97 | 1 |
| SBSwk1 | 0.89 | 0.95 | 0.97 | 0.99 | 1 |
| SBSwk2 | 0.86 | 0.92 | 0.96 | 0.97 | 1 |

# write.csv(asmrANNUAL,file="asmrANNUAL.csv")  
  
# Classify  
asmrCut<-function(x) cut(x,breaks=c(0,asmrClass$asmrUL),labels=asmrClass$class)  
  
asmrNorthClass<-  
 asmrANNUAL2 %>%   
 mutate\_if(is.numeric,asmrCut) %>%   
 mutate(bgc=as.character(bgc)) %>%   
 arrange(bgc)   
   
 knitr::kable(asmrNorthClass)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| bgc | S1.mean | S2.mean | S3.mean | S4.mean | S5.mean |
| BWBSwk2 | MD | SD | F | F | F |
| CWHws2 | F | F | F | F | F |
| ESSFmv1 | SD | SD | F | F | F |
| ESSFmv2 | SD | SD | F | F | F |
| ESSFmv3 | SD | SD | F | F | F |
| ESSFmv4 | SD | SD | F | F | F |
| ESSFwc3 | F | F | F | F | F |
| ESSFwk1 | F | F | F | F | F |
| ESSFwk2 | F | F | F | F | F |
| ICHmc1 | SD | SD | SD | F | F |
| ICHmc2 | SD | SD | SD | F | F |
| ICHvk2 | SD | F | F | F | F |
| ICHwk1 | SD | F | F | F | F |
| ICHwk3 | SD | F | F | F | F |
| ICHwk4 | SD | F | F | F | F |
| IDFdk3 | VD2 | VD2 | MD | MD | F |
| SBSdw1 | VD2 | MD | MD | SD | F |
| SBSdw2 | MD | SD | SD | SD | F |
| SBSdw3 | MD | MD | SD | SD | F |
| SBSmc1 | MD | SD | SD | SD | F |
| SBSmc2 | MD | SD | SD | SD | F |
| SBSmc3 | MD | SD | SD | SD | F |
| SBSmk1 | MD | MD | SD | SD | F |
| SBSmk2 | SD | SD | SD | F | F |
| SBSwk1 | SD | SD | F | F | F |
| SBSwk2 | SD | SD | F | F | F |
| SBSwk3 | SD | SD | F | F | F |

## Final task: Export data and save it to

# Create data to export as a list  
stnYears<-  
 stnID %>%   
 dplyr::select(stn=station\_name,stnID=station\_id,bgc) %>%   
 inner\_join(years,by="bgc")

## Warning: Column `bgc` joining character vector and factor, coercing into  
## character vector

asmrNorth<-list(current=list(class=asmrNorthClass,asmr=asmrANNUAL2,stnYears),future=list())