Standardizing Data

Here are a few different ways we can scale the data, the first is using a loop, which are incredibly versatile BUT kinda tricky to wrap your head around. (Here is more info)[https://intro2r.com/prog_r.html] on loops that is a good reference

Loop

First, lets read in the data:

```r
dat<- read.csv(
  here("Data", "Yellowtail", "2024Env-annual-yellowtail_GLORYS_UNSTANDARDIZED.csv")
)%>%
  filter(year>=1998&year<=2021) # lets filter the data so we dont have any missing values
```

Now, lets create a dataframe. We will "fill" this dataframe using a loop

```r
stand_dat <-dat #using the same shape as dat will also let us retain the column names
```

Now we can create the actual loop:

```r
#i is what we want the loop to iterate over, in this case i is columns. If we want the
# loop to loop over every column, we can start it at 1 and go to the total number
# of columns. ncol() is a function that counts the number of columns in a dataframe
# so in this case we will loop from column 1 to the last column

for(i in 1:ncol(dat)){
 stand_dat[,i]<- (dat[,i]-mean(dat[,i]))/sd(dat[,i])
}
```

Our stand_dat dataframe now contains each column with a mean of 0 and standard deviation of 1. Lets check it...

```
colMeans(stand_dat)
```

```
         year         DDpre         DDegg         DDlarv        DDpjuv
-9.251859e-18  8.367150e-16 -2.151057e-16  2.035409e-16  9.390636e-16
        DDben          Tcop         Tpart        MLDpart       MLDlarv
-1.292947e-15  1.327642e-15  8.835525e-16  3.238150e-16 -3.469447e-16
      MLDpjuv        CSTlarv       CSTpjuv        LSTlarv       LSTpjuv
 8.974303e-16  3.700743e-17  1.850372e-17  8.789266e-17 -2.220446e-16
    hci1_larv     hci1_pjuv     hci2_larv      hci2_pjuv       oni_pre
 0.000000e+00 -1.249001e-16  0.000000e+00 -1.249001e-16 -1.850372e-17
      oni_larv      oni_pjuv      pdo_larv       pdo_pjuv   lusi_annual
-1.850372e-17  2.312965e-17  7.401487e-17 -2.775558e-17 -3.654484e-16
     ZOOpjuvN       ZOObenN      ZOOpjuvS        ZOObenS             X
 0.000000e+00 -6.013708e-17  3.238150e-17 -6.245005e-17 -9.251859e-18
  CutiSTIpjuv   CutiTUMIpjuv          Year  BeutiTUMIpjuv        Year.1
 2.312965e-17  2.914335e-16 -9.251859e-18 -7.401487e-17 -9.251859e-18
  BeutiSTIpjuv       CHLpjuv        PPpjuv     bakun_sti
-1.989150e-16  6.476301e-17  8.396062e-16 -1.850372e-17
```

```
apply(stand_dat, 2, sd)
```

```
        year          DDpre          DDegg          DDlarv         DDpjuv
           1              1              1               1              1
       DDben           Tcop          Tpart         MLDpart        MLDlarv
           1              1              1               1              1
     MLDpjuv        CSTlarv        CSTpjuv         LSTlarv        LSTpjuv
           1              1              1               1              1
   hci1_larv      hci1_pjuv      hci2_larv       hci2_pjuv        oni_pre
           1              1              1               1              1
    oni_larv       oni_pjuv       pdo_larv        pdo_pjuv    lusi_annual
           1              1              1               1              1
    ZOOpjuvN        ZOObenN       ZOOpjuvS         ZOObenS              X
           1              1              1               1              1
 CutiSTIpjuv   CutiTUMIpjuv           Year  BeutiTUMIpjuv         Year.1
           1              1              1               1              1
 BeutiSTIpjuv       CHLpjuv         PPpjuv      bakun_sti
           1              1              1               1
```

The means of each column is pretty close to 0! And the sd is as well!

Apply

Now you may have noticed that when calculating the sd of each function I used apply. Rather than writing a loop we could have just used this! Lets use it here to standardize the data...

```
stand_data_apply <- apply(dat, 2, scale)
head(stand_data_apply )
```

```
            year        DDpre       DDegg       DDlarv      DDpjuv       DDben
[1,] -1.6263456  2.79432823  2.0652425  2.2468318  1.97182592  0.3776698
[2,] -1.4849242  0.03494097  0.2375537 -0.7622902 -0.91217826 -0.4744541
[3,] -1.3435029 -0.31874437 -0.1290300  0.1215045 -0.03340699 -0.9624530
[4,] -1.2020815 -0.67892077 -0.9378641 -0.4239759 -0.73198157 -0.6951369
[5,] -1.0606602 -0.79438257 -0.4608233 -0.9887974 -1.35184416 -0.6292660
[6,] -0.9192388 -0.23015000  0.2453647  0.8054927  0.23617465 -0.2484564
            Tcop        Tpart       MLDpart      MLDlarv      MLDpjuv      CSTlarv
[1,]  2.6666699  2.19889421  3.82323623  3.39269147  0.1485115 -0.57884030
[2,]  0.4379791 -0.85057733  1.06985511  2.11898054  1.0200999  0.90561290
[3,] -0.5602335 -0.05980499 -0.51832950 -0.18220965 -1.0067481  0.15083688
[4,] -0.5305484 -0.42450685 -0.23769989 -0.53042904  0.2297167 -0.44020836
[5,] -0.6697892 -0.85819721 -0.38622627 -0.80924151  0.1426017  0.07036301
[6,] -1.9395112  0.97069017 -0.01705528 -0.04951702 -0.3553550  0.60915913
          CSTpjuv     LSTlarv      LSTpjuv     hci1_larv     hci1_pjuv     hci2_larv
[1,] -0.03802977  1.7264370  0.4889551 -1.56798699 -1.27115913 -1.56798699
[2,]  0.94350146  1.4760473 -1.2856393  1.25559625  1.19972209  1.25559625
[3,]  1.19176038  0.2180359 -0.7130612  0.09881201  0.06562815  0.09881201
[4,]  0.38474393 -0.6262714 -0.5019289  0.39967244  1.13446472  0.39967244
[5,]  0.53120956 -0.9255748 -1.3013491  1.07861953  1.10480228  1.07861953
[6,] -0.10811819 -0.6993861 -0.1147803 -1.38296587 -0.50883443 -1.38296587
          hci2_pjuv     oni_pre      oni_larv      oni_pjuv     pdo_larv     pdo_pjuv
[1,] -1.27115913  2.1429247  2.0090279 -0.05089855  1.9366648  0.3970298
[2,]  1.19972209 -1.2653885 -1.2340431 -1.98712104 -0.5796399 -1.4867611
[3,]  0.06562815 -1.3052852 -1.2961926 -1.14365931 -0.5104292 -0.5346554
[4,]  1.13446472 -0.5632076 -0.4374003 -0.18801302  0.2904379 -0.8282001
[5,]  1.10480228 -0.1083859  0.1727943  1.41165577 -0.5697527 -0.2309005
[6,] -0.50883443  0.9893417  0.6756398  0.12361077  1.5461185  0.9994344
        lusi_annual      ZOOpjuvN      ZOObenN      ZOOpjuvS      ZOObenS           X
[1,]    1.0055400 -1.952427025 -0.3585604  1.7126823  0.7954300 -1.6263456
[2,]    1.5213590 -0.169297841  0.7000897 -0.7176185 -1.6095081 -1.4849242
[3,]   -0.4682286 -0.004760062  1.2082420 -0.9834969 -0.9465854 -1.3435029
[4,]    0.3791884  0.235604522  1.3160679 -0.9022688 -1.0887306 -1.2020815
[5,]    1.2266053  0.556019342  1.3121875 -0.8130939 -0.5867960 -1.0606602
```

```
[6,]  -0.3945401 -0.067985973 -0.4133460   0.5036207 -0.3517562 -0.9192388
     CutiSTIpjuv CutiTUMIpjuv       Year BeutiTUMIpjuv     Year.1 BeutiSTIpjuv
[1,]  -0.1918014   -0.4939859 -1.6263456  -1.568685313 -1.6263456   -0.1790974
[2,]   0.3504340    0.1430873 -1.4849242   0.001992683 -1.4849242    0.4387064
[3,]  -0.7109629   -0.6343623 -1.3435029  -1.099932577 -1.3435029   -0.7706116
[4,]  -1.2993460    0.9778535 -1.2020815   0.712209704 -1.2020815   -1.4409943
[5,]  -0.9878491    1.5685664 -1.0606602   1.339731782 -1.0606602   -0.6785982
[6,]   0.9157433    0.3323588 -0.9192388  -1.220105642 -0.9192388    1.0827996
        CHLpjuv     PPpjuv  bakun_sti
[1,] -1.9364182 -2.0454212 -1.5268571
[2,] -1.4005172 -1.0273683 -1.2167479
[3,] -0.7564655 -0.6926709  1.1090711
[4,] -0.8155079 -0.6337877  0.3854830
[5,]  2.2314761  2.3045496 -0.3381052
[6,]  0.6224340  0.9496238  0.2304284
```

Let's check, does the data from apply look the same as the data from the loop? You can use apply to also run a function across rows.

```
diff<-stand_data_apply - stand_dat
head(diff)
```

```
  year DDpre        DDegg        DDlarv DDpjuv        DDben        Tcop Tpart
1    0     0 1.332268e-15 -1.332268e-15      0 2.609024e-15 -2.664535e-15     0
2    0     0 1.498801e-15 -6.661338e-16      0 2.498002e-15 -2.664535e-15     0
3    0     0 1.498801e-15 -8.465451e-16      0 2.331468e-15 -2.664535e-15     0
4    0     0 1.554312e-15 -7.216450e-16      0 2.442491e-15 -2.664535e-15     0
5    0     0 1.498801e-15 -6.661338e-16      0 2.442491e-15 -2.664535e-15     0
6    0     0 1.498801e-15 -9.992007e-16      0 2.525757e-15 -2.664535e-15     0
        MLDpart      MLDlarv      MLDpjuv      CSTlarv      CSTpjuv
1 1.776357e-15 2.220446e-15 7.133183e-15 -5.551115e-16 -1.318390e-16
2 1.110223e-15 2.220446e-15 7.105427e-15 -4.440892e-16 -1.110223e-16
3 7.771561e-16 2.775558e-15 7.105427e-15 -5.273559e-16 -2.220446e-16
4 8.049117e-16 2.886580e-15 7.105427e-15 -5.551115e-16 -1.110223e-16
5 7.216450e-16 2.886580e-15 7.105427e-15 -5.134781e-16 -1.110223e-16
6 8.326673e-16 2.754741e-15 7.160939e-15 -5.551115e-16 -1.387779e-16
        LSTlarv      LSTpjuv hci1_larv hci1_pjuv hci2_larv hci2_pjuv
1 -4.440892e-16 7.216450e-16         0         0         0         0
2 -4.440892e-16 4.440892e-16         0         0         0         0
3 -2.498002e-16 5.551115e-16         0         0         0         0
4 -2.220446e-16 5.551115e-16         0         0         0         0
5 -1.110223e-16 4.440892e-16         0         0         0         0
```

```
6 -2.220446e-16 6.245005e-16            0            0            0            0
        oni_pre      oni_larv      oni_pjuv      pdo_larv      pdo_pjuv
1  0.000000e+00 0.000000e+00  9.714451e-17  2.220446e-16 5.551115e-17
2  0.000000e+00 0.000000e+00  2.220446e-16 -3.330669e-16 2.220446e-16
3  0.000000e+00 0.000000e+00  0.000000e+00 -2.220446e-16 0.000000e+00
4  0.000000e+00 5.551115e-17  1.110223e-16 -5.551115e-17 0.000000e+00
5 -1.387779e-17 8.326673e-17 -2.220446e-16 -2.220446e-16 8.326673e-17
6  0.000000e+00 0.000000e+00  6.938894e-17  2.220446e-16 0.000000e+00
  lusi_annual       ZOOpjuvN ZOObenN ZOOpjuvS       ZOObenS X  CutiSTIpjuv
1           0 0.000000e+00       0        0 1.110223e-16 0 4.718448e-16
2           0 2.775558e-17       0        0 2.220446e-16 0 4.996004e-16
3           0 1.734723e-17       0        0 1.110223e-16 0 3.330669e-16
4           0 5.551115e-17       0        0 0.000000e+00 0 4.440892e-16
5           0 0.000000e+00       0        0 1.110223e-16 0 3.330669e-16
6           0 2.775558e-17       0        0 5.551115e-17 0 5.551115e-16
  CutiTUMIpjuv Year BeutiTUMIpjuv Year.1 BeutiSTIpjuv       CHLpjuv PPpjuv
1 6.106227e-16    0 -1.110223e-15      0 5.273559e-16 -1.332268e-15      0
2 6.106227e-16    0 -1.004839e-15      0 6.661338e-16 -1.110223e-15      0
3 6.661338e-16    0 -8.881784e-16      0 4.440892e-16 -1.110223e-15      0
4 5.551115e-16    0 -9.992007e-16      0 2.220446e-16 -1.221245e-15      0
5 6.661338e-16    0 -8.881784e-16      0 4.440892e-16 -1.332268e-15      0
6 5.551115e-16    0 -8.881784e-16      0 6.661338e-16 -1.221245e-15      0
      bakun_sti
1  0.000000e+00
2  0.000000e+00
3  0.000000e+00
4  5.551115e-17
5 -5.551115e-17
6  8.326673e-17
```

It sure does! The values are identical, so we know both work

Scale

Now a useful thing about scale is that you actually don't need to loop or apply because that functionality is built into it! We can scale columns using the function directly

```
scale_dat_scale <- scale(dat)
head(scale_dat_scale)
```

```
         year        DDpre        DDegg        DDlarv        DDpjuv        DDben
```

```
[1,] -1.6263456  2.79432823  2.0652425  2.2468318  1.97182592  0.3776698
[2,] -1.4849242  0.03494097  0.2375537 -0.7622902 -0.91217826 -0.4744541
[3,] -1.3435029 -0.31874437 -0.1290300  0.1215045 -0.03340699 -0.9624530
[4,] -1.2020815 -0.67892077 -0.9378641 -0.4239759 -0.73198157 -0.6951369
[5,] -1.0606602 -0.79438257 -0.4608233 -0.9887974 -1.35184416 -0.6292660
[6,] -0.9192388 -0.23015000  0.2453647  0.8054927  0.23617465 -0.2484564
            Tcop       Tpart     MLDpart     MLDlarv     MLDpjuv      CSTlarv
[1,]  2.6666699  2.19889421  3.82323623  3.39269147  0.1485115 -0.57884030
[2,]  0.4379791 -0.85057733  1.06985511  2.11898054  1.0200999  0.90561290
[3,] -0.5602335 -0.05980499 -0.51832950 -0.18220965 -1.0067481  0.15083688
[4,] -0.5305484 -0.42450685 -0.23769989 -0.53042904  0.2297167 -0.44020836
[5,] -0.6697892 -0.85819721 -0.38622627 -0.80924151  0.1426017  0.07036301
[6,] -1.9395112  0.97069017 -0.01705528 -0.04951702 -0.3553550  0.60915913
         CSTpjuv      LSTlarv     LSTpjuv    hci1_larv    hci1_pjuv    hci2_larv
[1,] -0.03802977  1.7264370  0.4889551 -1.56798699 -1.27115913 -1.56798699
[2,]  0.94350146  1.4760473 -1.2856393  1.25559625  1.19972209  1.25559625
[3,]  1.19176038  0.2180359 -0.7130612  0.09881201  0.06562815  0.09881201
[4,]  0.38474393 -0.6262714 -0.5019289  0.39967244  1.13446472  0.39967244
[5,]  0.53120956 -0.9255748 -1.3013491  1.07861953  1.10480228  1.07861953
[6,] -0.10811819 -0.6993861 -0.1147803 -1.38296587 -0.50883443 -1.38296587
       hci2_pjuv    oni_pre    oni_larv     oni_pjuv    pdo_larv    pdo_pjuv
[1,] -1.27115913  2.1429247  2.0090279 -0.05089855  1.9366648  0.3970298
[2,]  1.19972209 -1.2653885 -1.2340431 -1.98712104 -0.5796399 -1.4867611
[3,]  0.06562815 -1.3052852 -1.2961926 -1.14365931 -0.5104292 -0.5346554
[4,]  1.13446472 -0.5632076 -0.4374003 -0.18801302  0.2904379 -0.8282001
[5,]  1.10480228 -0.1083859  0.1727943  1.41165577 -0.5697527 -0.2309005
[6,] -0.50883443  0.9893417  0.6756398  0.12361077  1.5461185  0.9994344
     lusi_annual     ZOOpjuvN     ZOObenN    ZOOpjuvS    ZOObenS           X
[1,]   1.0055400 -1.952427025 -0.3585604  1.7126823  0.7954300 -1.6263456
[2,]   1.5213590 -0.169297841  0.7000897 -0.7176185 -1.6095081 -1.4849242
[3,]  -0.4682286 -0.004760062  1.2082420 -0.9834969 -0.9465854 -1.3435029
[4,]   0.3791884  0.235604522  1.3160679 -0.9022688 -1.0887306 -1.2020815
[5,]   1.2266053  0.556019342  1.3121875 -0.8130939 -0.5867960 -1.0606602
[6,]  -0.3945401 -0.067985973 -0.4133460  0.5036207 -0.3517562 -0.9192388
     CutiSTIpjuv CutiTUMIpjuv        Year BeutiTUMIpjuv      Year.1 BeutiSTIpjuv
[1,]  -0.1918014   -0.4939859 -1.6263456  -1.568685313 -1.6263456   -0.1790974
[2,]   0.3504340    0.1430873 -1.4849242   0.001992683 -1.4849242    0.4387064
[3,]  -0.7109629   -0.6343623 -1.3435029  -1.099932577 -1.3435029   -0.7706116
[4,]  -1.2993460    0.9778535 -1.2020815   0.712209704 -1.2020815   -1.4409943
[5,]  -0.9878491    1.5685664 -1.0606602   1.339731782 -1.0606602   -0.6785982
[6,]   0.9157433    0.3323588 -0.9192388  -1.220105642 -0.9192388    1.0827996
         CHLpjuv     PPpjuv  bakun_sti
[1,] -1.9364182 -2.0454212 -1.5268571
```

```
[2,] -1.4005172 -1.0273683 -1.2167479
[3,] -0.7564655 -0.6926709  1.1090711
[4,] -0.8155079 -0.6337877  0.3854830
[5,]  2.2314761  2.3045496 -0.3381052
[6,]  0.6224340  0.9496238  0.2304284
```

Is it right???

```
apply(scale_dat_scale, 2, mean)
```

```
          year          DDpre          DDegg          DDlarv          DDpjuv
-1.850372e-17   7.968163e-16   1.292947e-15  -6.059967e-16   9.529414e-16
         DDben           Tcop          Tpart         MLDpart         MLDlarv
 1.309138e-15  -1.327642e-15   8.789266e-16   1.119475e-15   2.377728e-15
       MLDpjuv         CSTlarv        CSTpjuv         LSTlarv         LSTpjuv
 8.049117e-15  -4.440892e-16  -1.110223e-16  -2.220446e-16   4.255855e-16
     hci1_larv       hci1_pjuv      hci2_larv       hci2_pjuv         oni_pre
 0.000000e+00  -5.088522e-17   0.000000e+00  -5.088522e-17  -3.700743e-17
      oni_larv        oni_pjuv       pdo_larv        pdo_pjuv     lusi_annual
 4.163336e-17   8.789266e-17  -7.401487e-17   1.110223e-16  -3.469447e-16
      ZOOpjuvN         ZOObenN       ZOOpjuvS         ZOObenS               X
 0.000000e+00  -3.700743e-17   6.476301e-17   1.225871e-16  -1.850372e-17
    CutiSTIpjuv    CutiTUMIpjuv           Year    BeutiTUMIpjuv          Year.1
 5.458597e-16   8.974303e-16  -1.850372e-17  -1.110223e-15  -1.850372e-17
   BeutiSTIpjuv        CHLpjuv         PPpjuv      bakun_sti
 3.590878e-16  -1.110223e-15   8.141636e-16   5.551115e-17
```

```
apply(scale_dat_scale, 2, sd)
```

```
          year          DDpre          DDegg          DDlarv          DDpjuv
             1              1              1              1              1
         DDben           Tcop          Tpart         MLDpart         MLDlarv
             1              1              1              1              1
       MLDpjuv         CSTlarv        CSTpjuv         LSTlarv         LSTpjuv
             1              1              1              1              1
     hci1_larv       hci1_pjuv      hci2_larv       hci2_pjuv         oni_pre
             1              1              1              1              1
      oni_larv        oni_pjuv       pdo_larv        pdo_pjuv     lusi_annual
             1              1              1              1              1
      ZOOpjuvN         ZOObenN       ZOOpjuvS         ZOObenS               X
             1              1              1              1              1
```

```
   CutiSTIpjuv  CutiTUMIpjuv          Year  BeutiTUMIpjuv          Year.1
             1             1             1              1               1
   BeutiSTIpjuv       CHLpjuv        PPpjuv     bakun_sti
             1             1             1              1
```

There are a few things I did not execute flawlessly in this...you will notice I did not remove all variables. I also left year in the dataframe so it standardized that as well!! Which we don't want...I will leave it up to you to troubleshoot that :)

One thing to note is that if we want to use your long table, the easiest way to create standardized data in that format is looping over your variable name column. Alternatively, you could use the methods here and then merge or join the datasets together after you create a table of the standardized values.

Hope this helps! Happy standardizing!