R Tidyverse Methods for SoGE Geographers

Sami Rifai

10/15/2018

R for geographers

An exposition of the Tidyverse

Outline

1. **dplyr** & **tibble**: for manipulating data
2. **readr**: for reading and writing data
3. **lubridate**: for parsing dates
4. **tidyr**: for reshaping data
5. **ggplot2**: for visual exploration
6. **sf & stars**: The next gen of geospatial methods for R & Tidyverse methods

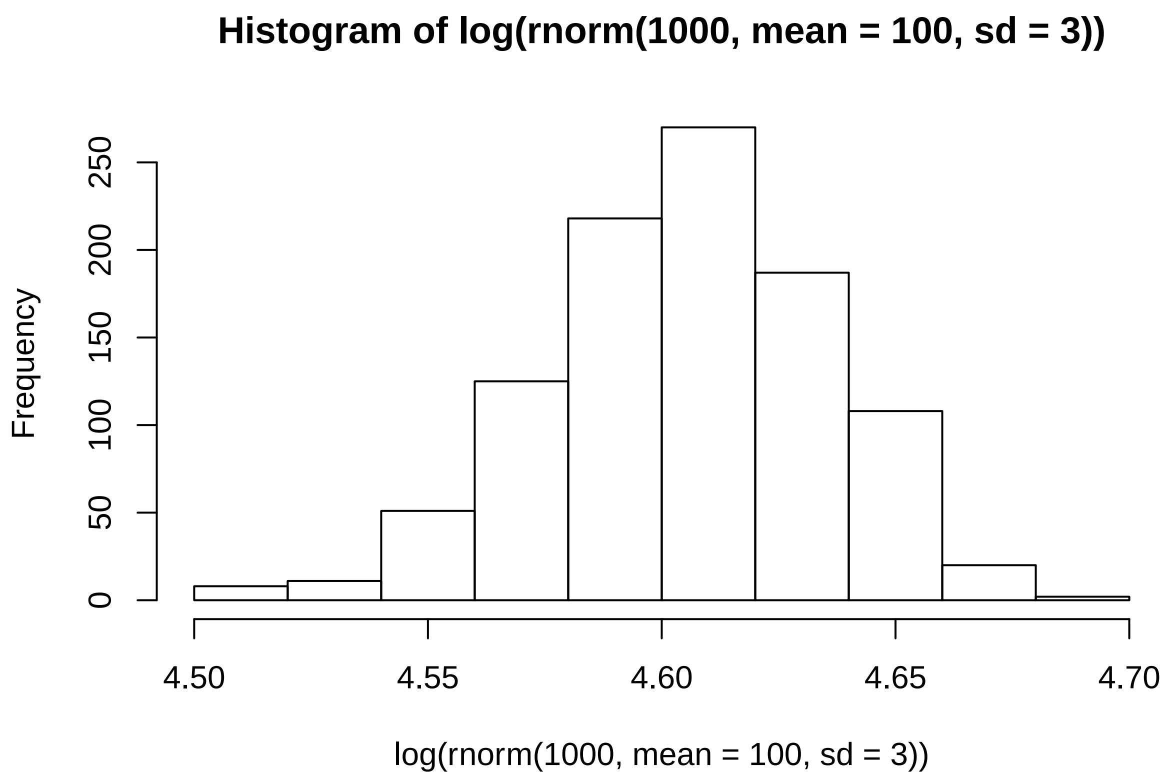
```
library(tidyverse); library(lubridate)
```

#(1) Basic plumbing with **dplyr** - This is a 'pipe' symbol: %>% - The keyboard shortcut in Rstudio is "ctrl+shift+m" - pipes allow chaining operations to an object

non pipe way

- this is the 'nested' way to do: generate a vector of random numbers, log it, and plot the distribution

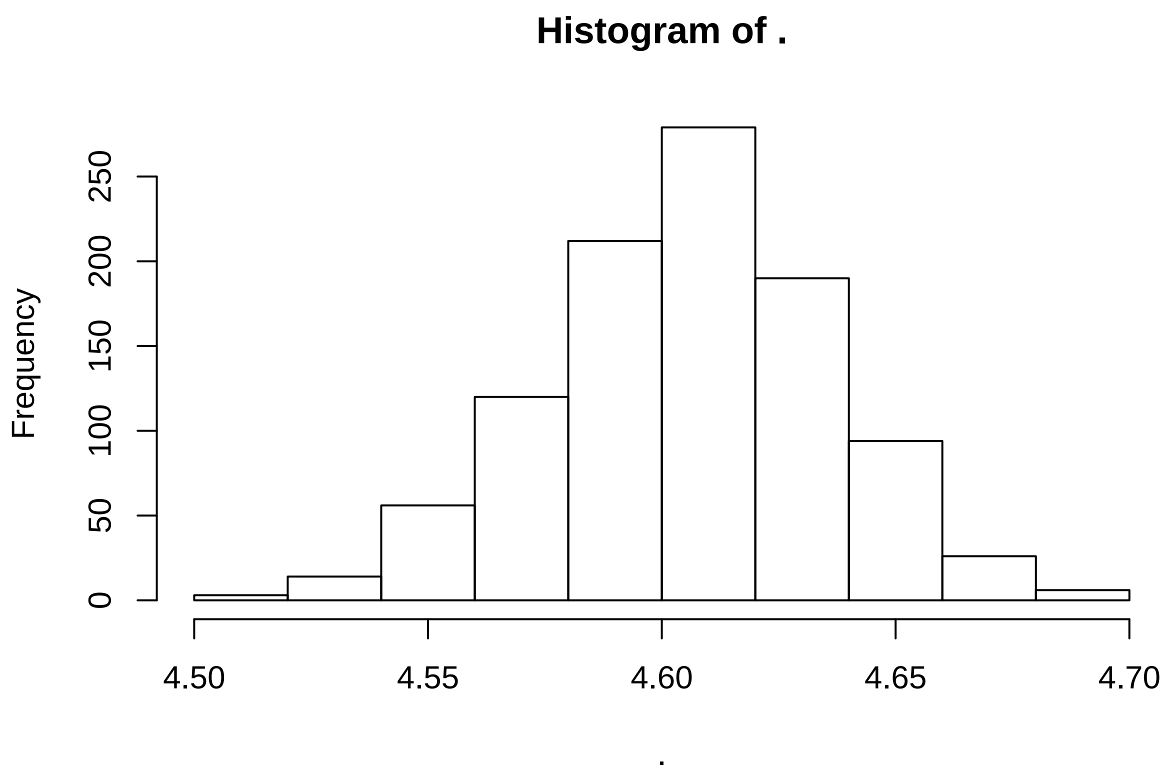
```
hist(log(rnorm(1000, mean = 100, sd=3)))
```



pipe way

- Here you can linearly read the chain of operations

```
#  vector object            -> function -> function
rnorm(1000, mean=100, sd=3) %>% log() %>% hist()
```



So pipes allow us to arbitrarily long things without nesting or creating copyies of dataframes

---

Life before and after dplyr

- excellent tutorial: [https://suzan.rbind.io/categories/tutorial/]

- Tibbles vs dataframes -

tibbles are a more clever version of data frames that offer some features (?tibble) but also, are compatible for tidyverse methods

```
junk_df <- data.frame(x=rnorm(50, mean = 0, sd=10),
                   y=rnorm(50, mean=10, sd=1))
print(junk_df) # prints the whole df!
```

```
##                x         y
## 1  -13.42013644  9.899080
## 2   -2.72624497  8.122140
## 3   14.33331489  9.377063
## 4   -2.00073700  8.786118
## 5   -8.95760280 10.146646
## 6    6.01841304 11.568485
## 7   -3.49476577  9.701002
## 8  -11.31254423 10.054335
## 9   -1.03993308 10.052783
```

```
## 10   2.37665169 10.395346
## 11  11.51411317  9.973592
## 12   9.02327334 10.671383
## 13  16.74033810 10.104073
## 14   4.04166159 11.701878
## 15  16.54674992  9.509778
## 16  -3.71409226  9.349494
## 17  -2.28297285 10.472558
## 18  -4.73320104 11.041515
## 19  25.07455501 10.812651
## 20 -10.87987460  8.889303
## 21   0.06751643  9.677153
## 22  -0.46693048  9.356081
## 23  -9.67103402  9.389366
## 24   4.77194121  8.883463
## 25  10.82688766 10.157656
## 26  -7.81454942  8.983932
## 27  -7.96608908 11.042147
## 28   1.31347288 11.039633
## 29  -1.64423169  8.295140
## 30 -11.56610671  9.070909
## 31  15.74055500 10.368546
## 32  -4.84091522  9.900596
## 33  16.94258051 10.019257
## 34   8.78923676 10.406685
## 35  10.85884840  9.443576
## 36  -6.00656340  8.879752
## 37 -13.47288831 11.205862
## 38  11.16952247 11.949249
## 39  -5.15879134  9.998317
## 40   2.72817255 11.523834
## 41  -7.16898800 10.601020
## 42   0.41469733 10.504434
## 43   5.28110522 11.680157
## 44   4.97441062  9.506238
## 45  13.91818093 11.155824
## 46   4.38467832  9.195625
## 47 -16.20258642 11.260998
## 48  -0.57010863  9.724688
## 49  -4.68880804 10.572386
## 50   4.77566221  8.402937
```

```r
junk_tb <- tibble(x=rnorm(50, mean = 0, sd=10),
                      y=rnorm(50, mean=10, sd=1))
print(junk_tb) # just prints the top 10 rows
```

```
## # A tibble: 50 x 2
##           x     y
##       <dbl> <dbl>
##  1  12.5    11.1
##  2   1.39   10.1
##  3   3.75   12.0
##  4  -0.0295  9.55
##  5 -10.4     9.61
##  6   8.50   10.6
```

```
## 7   0.988   9.50
## 8   0.411  10.5
## 9 -10.6    10.1
## 10  9.64    8.83
## # ... with 40 more rows
```

#**Filtering with dplyr** ## filtering data before dplyr:

```
iris[iris[,'Species']=="setosa" & iris[,"Sepal.Length"] > 5.0,]
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1           5.1         3.5          1.4         0.2  setosa
## 6           5.4         3.9          1.7         0.4  setosa
## 11          5.4         3.7          1.5         0.2  setosa
## 15          5.8         4.0          1.2         0.2  setosa
## 16          5.7         4.4          1.5         0.4  setosa
## 17          5.4         3.9          1.3         0.4  setosa
## 18          5.1         3.5          1.4         0.3  setosa
## 19          5.7         3.8          1.7         0.3  setosa
## 20          5.1         3.8          1.5         0.3  setosa
## 21          5.4         3.4          1.7         0.2  setosa
## 22          5.1         3.7          1.5         0.4  setosa
## 24          5.1         3.3          1.7         0.5  setosa
## 28          5.2         3.5          1.5         0.2  setosa
## 29          5.2         3.4          1.4         0.2  setosa
## 32          5.4         3.4          1.5         0.4  setosa
## 33          5.2         4.1          1.5         0.1  setosa
## 34          5.5         4.2          1.4         0.2  setosa
## 37          5.5         3.5          1.3         0.2  setosa
## 40          5.1         3.4          1.5         0.2  setosa
## 45          5.1         3.8          1.9         0.4  setosa
## 47          5.1         3.8          1.6         0.2  setosa
## 49          5.3         3.7          1.5         0.2  setosa
```

filtering with dplyr

```
iris %>% filter(Species=="setosa" & Sepal.Length > 5.0)
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1           5.1         3.5          1.4         0.2  setosa
## 2           5.4         3.9          1.7         0.4  setosa
## 3           5.4         3.7          1.5         0.2  setosa
## 4           5.8         4.0          1.2         0.2  setosa
## 5           5.7         4.4          1.5         0.4  setosa
## 6           5.4         3.9          1.3         0.4  setosa
## 7           5.1         3.5          1.4         0.3  setosa
## 8           5.7         3.8          1.7         0.3  setosa
## 9           5.1         3.8          1.5         0.3  setosa
## 10          5.4         3.4          1.7         0.2  setosa
## 11          5.1         3.7          1.5         0.4  setosa
## 12          5.1         3.3          1.7         0.5  setosa
## 13          5.2         3.5          1.5         0.2  setosa
## 14          5.2         3.4          1.4         0.2  setosa
```

```
## 15          5.4         3.4          1.5         0.4  setosa
## 16          5.2         4.1          1.5         0.1  setosa
## 17          5.5         4.2          1.4         0.2  setosa
## 18          5.5         3.5          1.3         0.2  setosa
## 19          5.1         3.4          1.5         0.2  setosa
## 20          5.1         3.8          1.9         0.4  setosa
## 21          5.1         3.8          1.6         0.2  setosa
## 22          5.3         3.7          1.5         0.2  setosa
```

- Creating or transforming variables -

```
iris <- iris %>%
  mutate(p_wl_ratio = Petal.Width/Petal.Length) %>%
  mutate(narrow = ifelse(p_wl_ratio < 0.25, TRUE, FALSE))
```

- Renaming variables -

```
iris %>% names()
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"
## [5] "Species"      "p_wl_ratio"   "narrow"
```

```
iris %>%
  rename_all(tolower) %>% # rename cols with lowercase
  head() # shows the top rows
```

```
##   sepal.length sepal.width petal.length petal.width species p_wl_ratio
## 1          5.1         3.5          1.4         0.2  setosa  0.1428571
## 2          4.9         3.0          1.4         0.2  setosa  0.1428571
## 3          4.7         3.2          1.3         0.2  setosa  0.1538462
## 4          4.6         3.1          1.5         0.2  setosa  0.1333333
## 5          5.0         3.6          1.4         0.2  setosa  0.1428571
## 6          5.4         3.9          1.7         0.4  setosa  0.2352941
##   narrow
## 1   TRUE
## 2   TRUE
## 3   TRUE
## 4   TRUE
## 5   TRUE
## 6   TRUE
```

- sample random rows -

```
iris %>%
  sample_n(10)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 59           6.6         2.9          4.6         1.3 versicolor
## 28           5.2         3.5          1.5         0.2     setosa
```

```
## 30           4.7         3.2          1.6         0.2     setosa
## 77           6.8         2.8          4.8         1.4 versicolor
## 76           6.6         3.0          4.4         1.4 versicolor
## 84           6.0         2.7          5.1         1.6 versicolor
## 141          6.7         3.1          5.6         2.4  virginica
## 125          6.7         3.3          5.7         2.1  virginica
## 94           5.0         2.3          3.3         1.0 versicolor
## 124          6.3         2.7          4.9         1.8  virginica
##     p_wl_ratio narrow
## 59   0.2826087  FALSE
## 28   0.1333333   TRUE
## 30   0.1250000   TRUE
## 77   0.2916667  FALSE
## 76   0.3181818  FALSE
## 84   0.3137255  FALSE
## 141  0.4285714  FALSE
## 125  0.3684211  FALSE
## 94   0.3030303  FALSE
## 124  0.3673469  FALSE
```

#**Summarize groups with a statistic**

```
library(tidyverse)
iris %>% # count number of observations per species
  group_by(Species) %>% # grouping variable
  summarize(nobs = n()) %>% # count the number of observations
  ungroup() # always ungroup, not strictly necessary, but it will save you much pain in time
```

```
## # A tibble: 3 x 2
##   Species     nobs
##   <fct>      <int>
## 1 setosa        50
## 2 versicolor    50
## 3 virginica     50
```

```
library(tidyverse);
iris %>% # calc mean of traits per species
  group_by(Species) %>%
  summarise_all(mean) %>% # quick way to generate statistic for many columns
  ungroup()
```

```
## # A tibble: 3 x 7
##   Species    Sepal.Length Sepal.Width Petal.Length Petal.Width p_wl_ratio
##   <fct>             <dbl>       <dbl>        <dbl>       <dbl>      <dbl>
## 1 setosa             5.01        3.43         1.46       0.246      0.168
## 2 versicolor         5.94        2.77         4.26       1.33       0.311
## 3 virginica          6.59        2.97         5.55       2.03       0.367
## # ... with 1 more variable: narrow <dbl>
```

```
iris %>% # calc mean of traits across three species
  group_by(Species) %>%
  summarise_all(mean) %>%
  ungroup() %>%
  select(-Species) %>% # de-select Species
  summarize_all(mean) %>%
  ungroup()
```

```
## # A tibble: 1 x 6
##   Sepal.Length Sepal.Width Petal.Length Petal.Width p_wl_ratio narrow
##          <dbl>       <dbl>        <dbl>       <dbl>      <dbl>  <dbl>
## 1         5.84        3.06         3.76        1.20      0.282  0.293
```

(3 & 4) readr & tidyr for getting data into a workable shape ———————————-

```r
library(tidyverse); library(lubridate);

read.table("data/mei_1950_2018.data",skip = 1, nrows = 68) # base R way
```

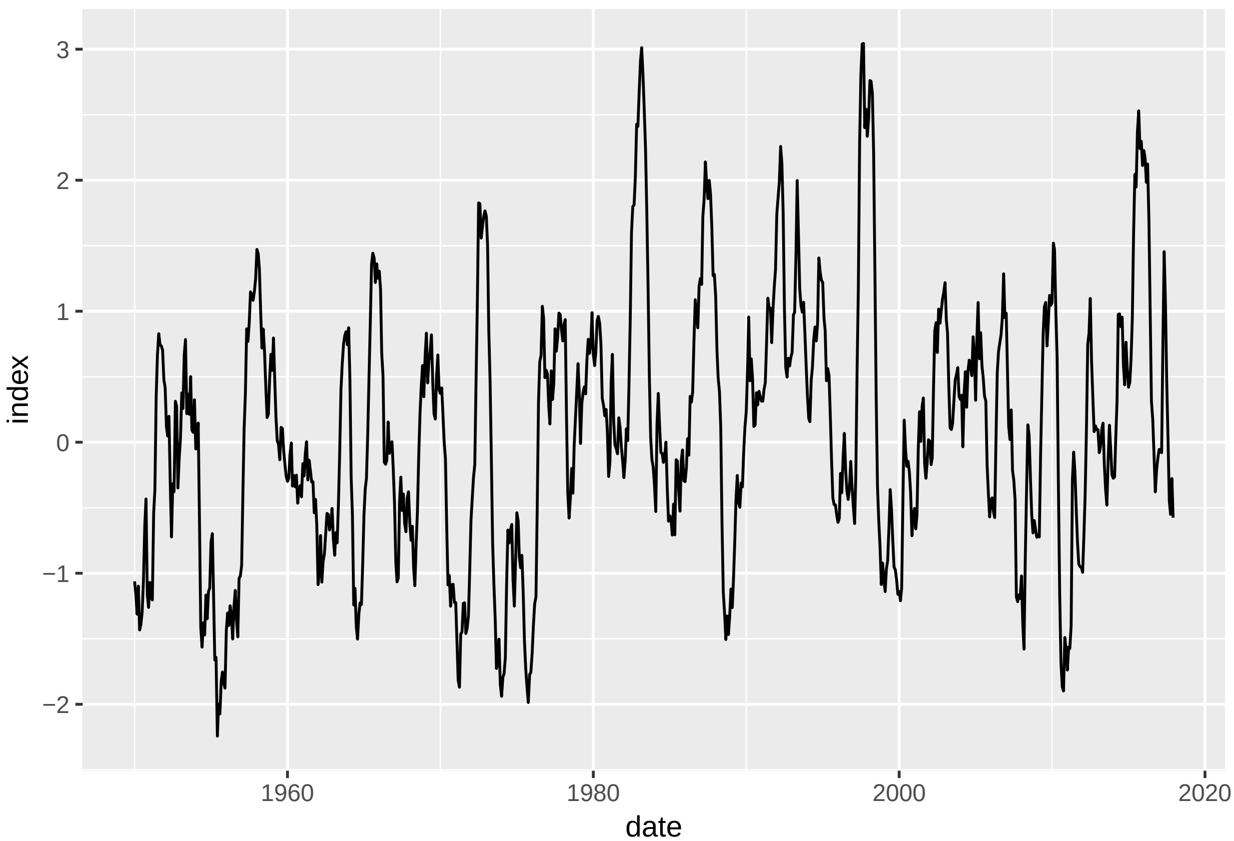
```
##      V1     V2     V3     V4     V5     V6     V7     V8     V9    V10
## 1  1950 -1.062 -1.163 -1.312 -1.098 -1.433 -1.391 -1.296 -1.053 -0.634
## 2  1951 -1.070 -1.183 -1.204 -0.544 -0.360  0.349  0.666  0.829  0.743
## 3  1952  0.419  0.117  0.047  0.198 -0.309 -0.723 -0.316 -0.378  0.313
## 4  1953  0.030  0.377  0.257  0.668  0.784  0.218  0.368  0.213  0.501
## 5  1954 -0.051 -0.048  0.147 -0.634 -1.416 -1.564 -1.378 -1.471 -1.166
## 6  1955 -0.762 -0.697 -1.147 -1.662 -1.642 -2.243 -1.998 -2.073 -1.823
## 7  1956 -1.437 -1.303 -1.399 -1.248 -1.317 -1.502 -1.259 -1.131 -1.359
## 8  1957 -0.941 -0.372  0.101  0.372  0.866  0.769  0.908  1.148  1.125
## 9  1958  1.472  1.439  1.320  0.987  0.719  0.864  0.693  0.427  0.188
## 10 1959  0.548  0.796  0.495  0.192  0.013 -0.018 -0.134  0.114  0.105
## 11 1960 -0.299 -0.274 -0.094 -0.005 -0.335 -0.254 -0.340 -0.251 -0.465
## 12 1961 -0.163 -0.257 -0.088  0.004 -0.288 -0.137 -0.216 -0.304 -0.301
## 13 1962 -1.087 -0.988 -0.712 -1.068 -0.910 -0.852 -0.701 -0.543 -0.551
## 14 1963 -0.739 -0.863 -0.690 -0.768 -0.477 -0.087  0.401  0.597  0.750
## 15 1964  0.874  0.468 -0.269 -0.562 -1.242 -1.115 -1.405 -1.503 -1.311
## 16 1965 -0.557 -0.353 -0.278  0.063  0.490  0.915  1.360  1.443  1.406
## 17 1966  1.306  1.170  0.681  0.506 -0.152 -0.168 -0.136  0.155 -0.085
## 18 1967 -0.473 -0.919 -1.066 -1.037 -0.455 -0.266 -0.521 -0.395 -0.621
## 19 1968 -0.619 -0.749 -0.641 -0.959 -1.095 -0.771 -0.527 -0.102  0.220
## 20 1969  0.664  0.833  0.453  0.616  0.696  0.820  0.467  0.218  0.177
## 21 1970  0.372  0.415  0.220  0.000 -0.126 -0.659 -1.089 -1.016 -1.252
## 22 1971 -1.223 -1.528 -1.817 -1.870 -1.464 -1.448 -1.230 -1.225 -1.460
## 23 1972 -0.596 -0.424 -0.269 -0.171  0.464  1.069  1.827  1.821  1.558
## 24 1973  1.726  1.500  0.860  0.473 -0.106 -0.769 -1.081 -1.347 -1.727
## 25 1974 -1.939 -1.793 -1.767 -1.643 -1.077 -0.670 -0.769 -0.671 -0.627
## 26 1975 -0.538 -0.600 -0.879 -0.959 -0.863 -1.150 -1.519 -1.730 -1.874
## 27 1976 -1.610 -1.392 -1.234 -1.180 -0.496  0.307  0.615  0.664  1.038
## 28 1977  0.521  0.273  0.139  0.545  0.326  0.451  0.866  0.695  0.800
## 29 1978  0.773  0.899  0.936  0.191 -0.388 -0.579 -0.433 -0.200 -0.389
## 30 1979  0.600  0.362 -0.010  0.292  0.380  0.423  0.369  0.625  0.786
## 31 1980  0.672  0.585  0.689  0.927  0.961  0.907  0.749  0.336  0.281
## 32 1981 -0.262 -0.151  0.456  0.671  0.161 -0.019 -0.048 -0.088  0.187
## 33 1982 -0.270 -0.137  0.103  0.013  0.429  0.944  1.604  1.799  1.811
## 34 1983  2.683  2.910  3.012  2.808  2.542  2.240  1.763  1.178  0.497
## 35 1984 -0.330 -0.529  0.139  0.373  0.131 -0.079 -0.084 -0.154 -0.106
## 36 1985 -0.561 -0.595 -0.709 -0.472 -0.707 -0.133 -0.143 -0.367 -0.526
## 37 1986 -0.301 -0.195  0.028 -0.099  0.350  0.306  0.383  0.775  1.088
## 38 1987  1.250  1.205  1.722  1.859  2.140  1.964  1.859  1.999  1.894
## 39 1988  1.119  0.706  0.491  0.387  0.119 -0.622 -1.145 -1.303 -1.506
```

```
## 40 1989 -1.120 -1.262 -1.054 -0.763 -0.435 -0.253 -0.459 -0.497 -0.311
## 41 1990  0.237  0.563  0.956  0.469  0.637  0.484  0.120  0.131  0.378
## 42 1991  0.313  0.314  0.402  0.454  0.759  1.100  1.023  1.024  0.760
## 43 1992  1.743  1.870  1.991  2.258  2.129  1.748  1.018  0.570  0.497
## 44 1993  0.687  0.974  0.990  1.417  1.998  1.591  1.170  1.042  0.992
## 45 1994  0.353  0.182  0.157  0.473  0.573  0.788  0.880  0.773  0.908
## 46 1995  1.220  0.946  0.853  0.469  0.563  0.508  0.207 -0.143 -0.426
## 47 1996 -0.612 -0.580 -0.238 -0.386 -0.127  0.068 -0.204 -0.374 -0.437
## 48 1997 -0.490 -0.621 -0.252  0.543  1.165  2.292  2.805  3.040  3.044
## 49 1998  2.466  2.761  2.755  2.661  2.212  1.292  0.347 -0.331 -0.600
## 50 1999 -1.053 -1.140 -0.971 -0.903 -0.660 -0.361 -0.507 -0.745 -0.953
## 51 2000 -1.139 -1.210 -1.113 -0.409  0.169 -0.053 -0.184 -0.145 -0.227
## 52 2001 -0.505 -0.661 -0.560 -0.055  0.233  0.006  0.270  0.338 -0.165
## 53 2002  0.009 -0.171 -0.121  0.414  0.851  0.913  0.685  1.017  0.908
## 54 2003  1.218  0.935  0.833  0.421  0.111  0.097  0.144  0.316  0.477
## 55 2004  0.327  0.359 -0.035  0.374  0.539  0.267  0.541  0.627  0.572
## 56 2005  0.320  0.810  1.067  0.637  0.836  0.585  0.490  0.352  0.315
## 57 2006 -0.438 -0.424 -0.527 -0.575  0.008  0.530  0.691  0.759  0.823
## 58 2007  0.985  0.528  0.120  0.020  0.247 -0.215 -0.288 -0.441 -1.181
## 59 2008 -1.020 -1.388 -1.579 -0.879 -0.368  0.133  0.054 -0.266 -0.551
## 60 2009 -0.726 -0.707 -0.723 -0.105  0.361  0.819  1.035  1.067  0.735
## 61 2010  1.067  1.520  1.469  0.990  0.643 -0.325 -1.156 -1.683 -1.868
## 62 2011 -1.739 -1.563 -1.575 -1.399 -0.288 -0.075 -0.228 -0.519 -0.769
## 63 2012 -0.993 -0.695 -0.398  0.112  0.747  0.835  1.098  0.619  0.339
## 64 2013  0.096 -0.080 -0.037  0.095  0.146 -0.168 -0.355 -0.480 -0.133
## 65 2014 -0.275 -0.266  0.027  0.312  0.976  0.980  0.882  0.954  0.585
## 66 2015  0.420  0.459  0.631  0.943  1.584  2.045  1.948  2.366  2.530
## 67 2016  2.227  2.169  1.984  2.124  1.699  1.001  0.312  0.175 -0.101
## 68 2017 -0.055 -0.056 -0.080  0.770  1.455  1.049  0.461  0.027 -0.449
##       V11    V12    V13
## 1  -0.433 -1.165 -1.261
## 2   0.736  0.703  0.478
## 3   0.275 -0.349 -0.124
## 4   0.093  0.075  0.324
## 5  -1.348 -1.140 -1.113
## 6  -1.753 -1.841 -1.877
## 7  -1.486 -1.038 -1.022
## 8   1.083  1.148  1.248
## 9   0.213  0.486  0.671
## 10 -0.060 -0.170 -0.261
## 11 -0.355 -0.331 -0.417
## 12 -0.539 -0.436 -0.634
## 13 -0.670 -0.623 -0.505
## 14  0.814  0.844  0.744
## 15 -1.225 -1.239 -0.936
## 16  1.219  1.362  1.252
## 17 -0.044  0.004 -0.199
## 18 -0.683 -0.426 -0.378
## 19  0.435  0.586  0.347
## 20  0.511  0.666  0.398
## 21 -1.088 -1.084 -1.223
## 22 -1.421 -1.329 -0.993
## 23  1.643  1.726  1.766
## 24 -1.667 -1.503 -1.848
```

```
## 25 -1.052 -1.251 -0.905
## 26 -1.987 -1.773 -1.757
## 27  0.946  0.493  0.550
## 28  0.986  0.975  0.860
## 29 -0.020  0.186  0.388
## 30  0.678  0.746  0.989
## 31  0.201  0.251  0.089
## 32  0.112 -0.038 -0.141
## 33  2.024  2.428  2.411
## 34  0.038 -0.132 -0.188
## 35  0.001 -0.352 -0.603
## 36 -0.139 -0.059 -0.293
## 37  0.979  0.873  1.190
## 38  1.647  1.271  1.282
## 39 -1.326 -1.468 -1.328
## 40 -0.341 -0.073  0.115
## 41  0.285  0.389  0.348
## 42  1.009  1.189  1.320
## 43  0.641  0.582  0.648
## 44  1.069  0.834  0.589
## 45  1.407  1.299  1.237
## 46 -0.477 -0.478 -0.554
## 47 -0.349 -0.146 -0.336
## 48  2.401  2.542  2.335
## 49 -0.798 -1.086 -0.922
## 50 -0.973 -1.050 -1.161
## 51 -0.387 -0.714 -0.566
## 52 -0.275 -0.153  0.019
## 53  1.000  1.090  1.145
## 54  0.516  0.570  0.351
## 55  0.508  0.805  0.667
## 56 -0.167 -0.392 -0.570
## 57  0.955  1.286  0.951
## 58 -1.217 -1.165 -1.193
## 59 -0.692 -0.597 -0.663
## 60  0.909  1.121  1.045
## 61 -1.899 -1.490 -1.577
## 62 -0.933 -0.949 -0.957
## 63  0.081  0.125  0.094
## 64  0.130 -0.053 -0.248
## 65  0.438  0.763  0.558
## 66  2.241  2.297  2.112
## 67 -0.379 -0.212 -0.121
## 68 -0.551 -0.277 -0.576
```

```r
mei_wide <- readr::read_table("data/mei_1950_2018.data", skip = 1, col_names = F, n_max = 68) # slightl

names(mei_wide) <- c("year",1:12) # add names to columns

mei_long <- mei_wide %>% # reshape the data frame from 'wide' to 'long'
  gather(key="month",value="index",-year) %>% # use gather() to assemble key-value pairs
  mutate(date=parse_date_time(paste(year,month,1),"ymd")) # mutate the date

mei_long %>%
```

```r
  ggplot(data=., aes(date, index))+geom_line()
```

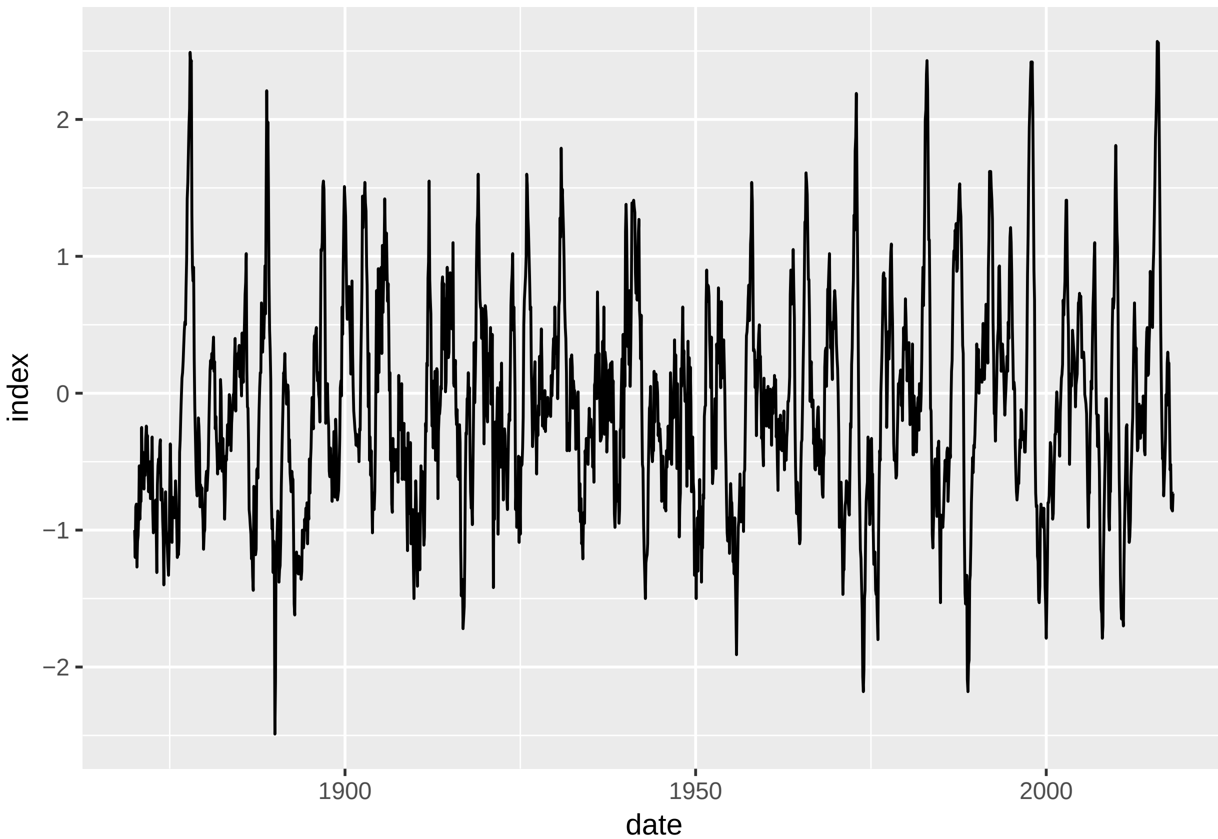


(5) ggplot2 for data visulization

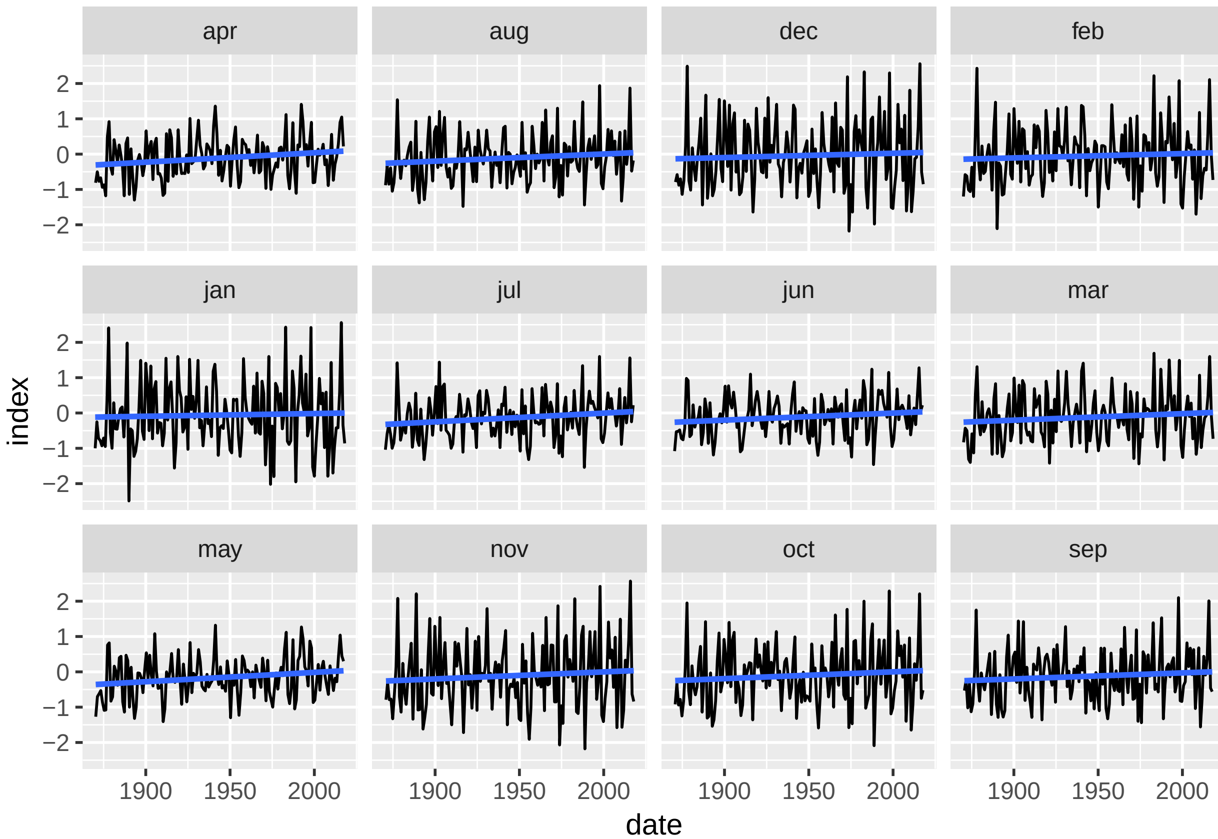
ggPlotting El Niño

"always plot your data"

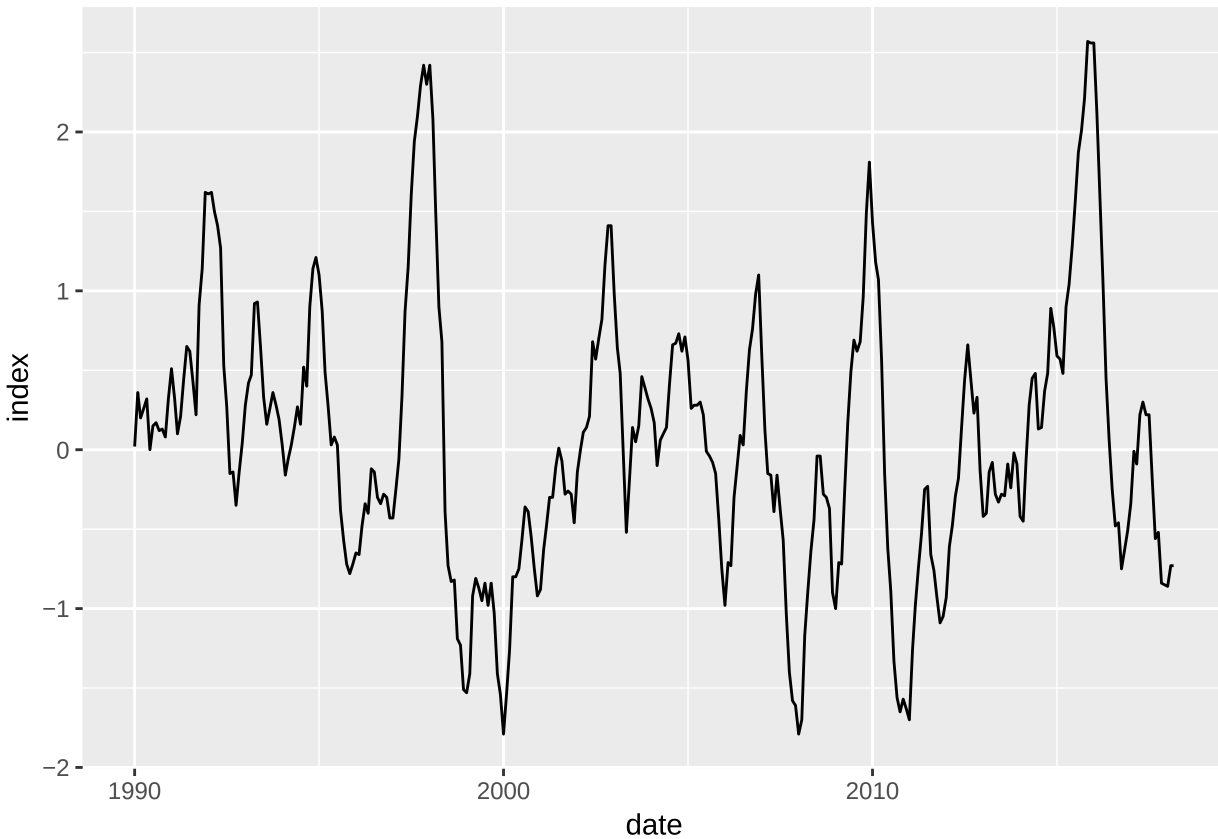
```r
library(tidyverse);

tmp <- read_csv("data/nino34_1870_2017.csv")

# plot the whole record
tmp %>%
  #_________________x_____y_______thing to add to plot
  ggplot(data=., aes(date, index))+geom_line()
```




```r
#plot record by month
tmp %>%
  ggplot(data=., aes(x=date, y=index))+ # what are the axis vars?
  geom_line()+                          # what will go on the plot?
  geom_smooth(method='lm',se=F)+        # add a linear trend line
  facet_wrap(~month)                    # generate the plot for every month
```

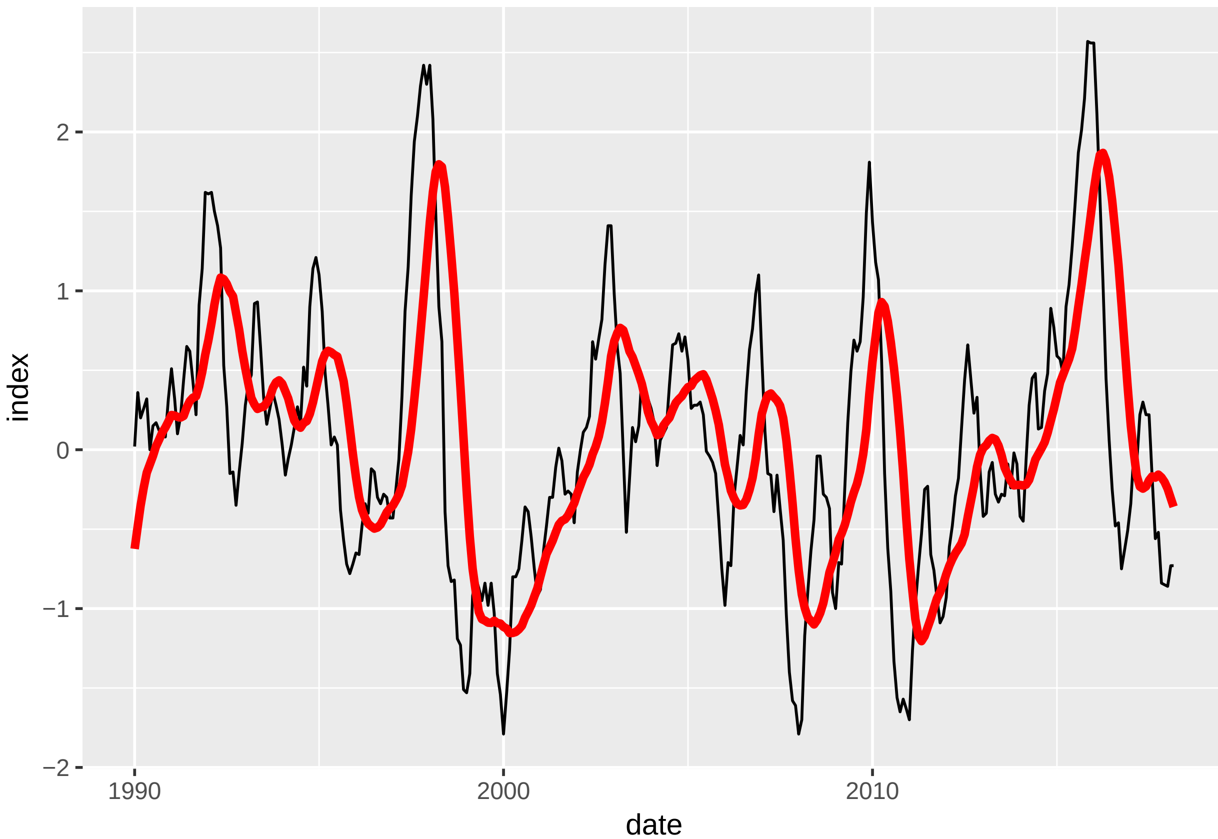



```
# plot recent ENSO record
tmp %>%
  filter(year>=1990) %>%                # filter for years >= 1990
  ggplot(data=., aes(date, index))+
  geom_line()
```



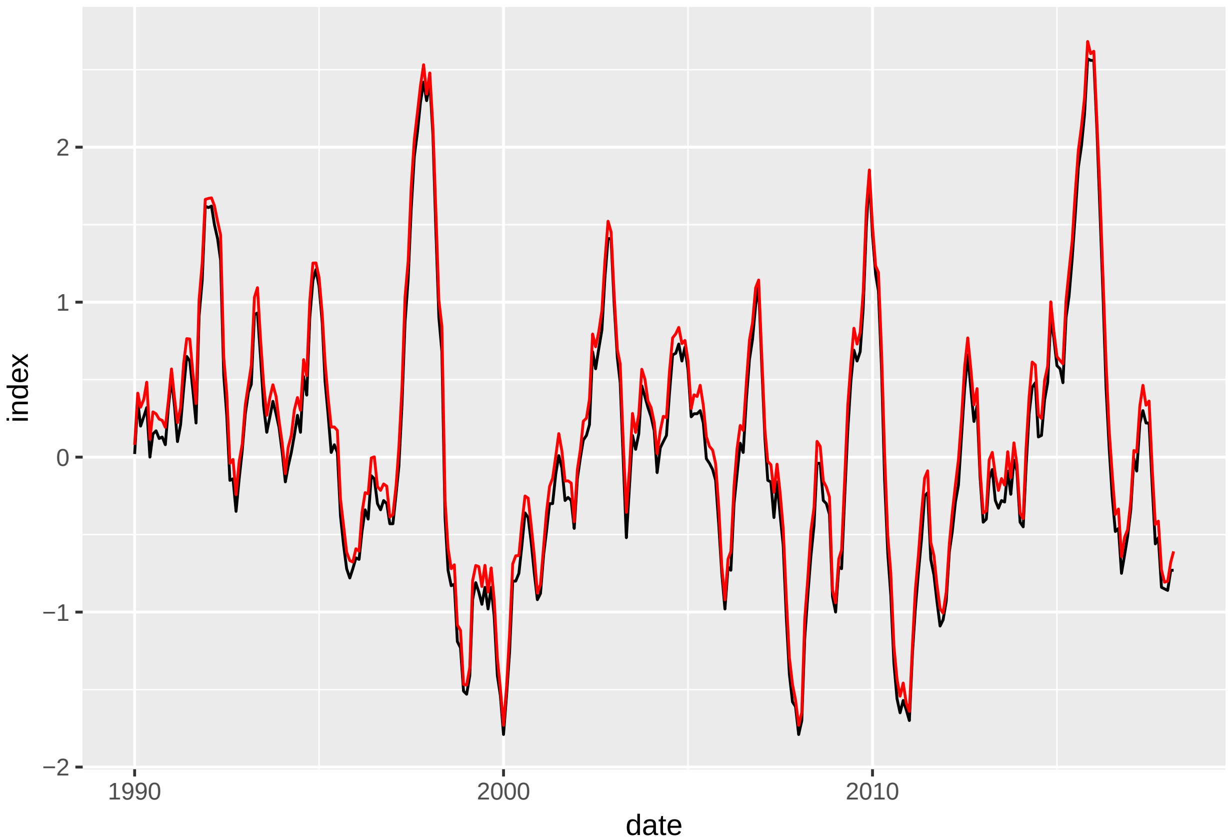
LETS 'smooth' the record with a moving average

```
library(RcppRoll)
tmp %>%
  arrange(date) %>%  # sort by the date
  mutate(index_12mo = roll_meanr(index, n=12, fill=NA)) %>%  # running 12 month mean
  filter(year>=1990) %>%
  ggplot(data=., aes(date, index))+
  geom_line()+
  geom_line(aes(date, index_12mo),col='red',lwd=1.5)
```



LET's 'deseasonlize' the record by subtracting the monthly mean

```
df_norms <- tmp %>%
  group_by(month) %>%
  summarize(index_u = mean(index, na.rm=T)) %>%
  ungroup()
tmp2 <- left_join(tmp, df_norms, by=c("month")) # now we join it back together

tmp2 %>%
  mutate(index_ds = index-index_u) %>%
  filter(year>=1990) %>%
  ggplot(data=., aes(date, index))+
  geom_line()+
  geom_line(aes(date, index_ds), col='red') # so that actually didn't make much of a difference
```



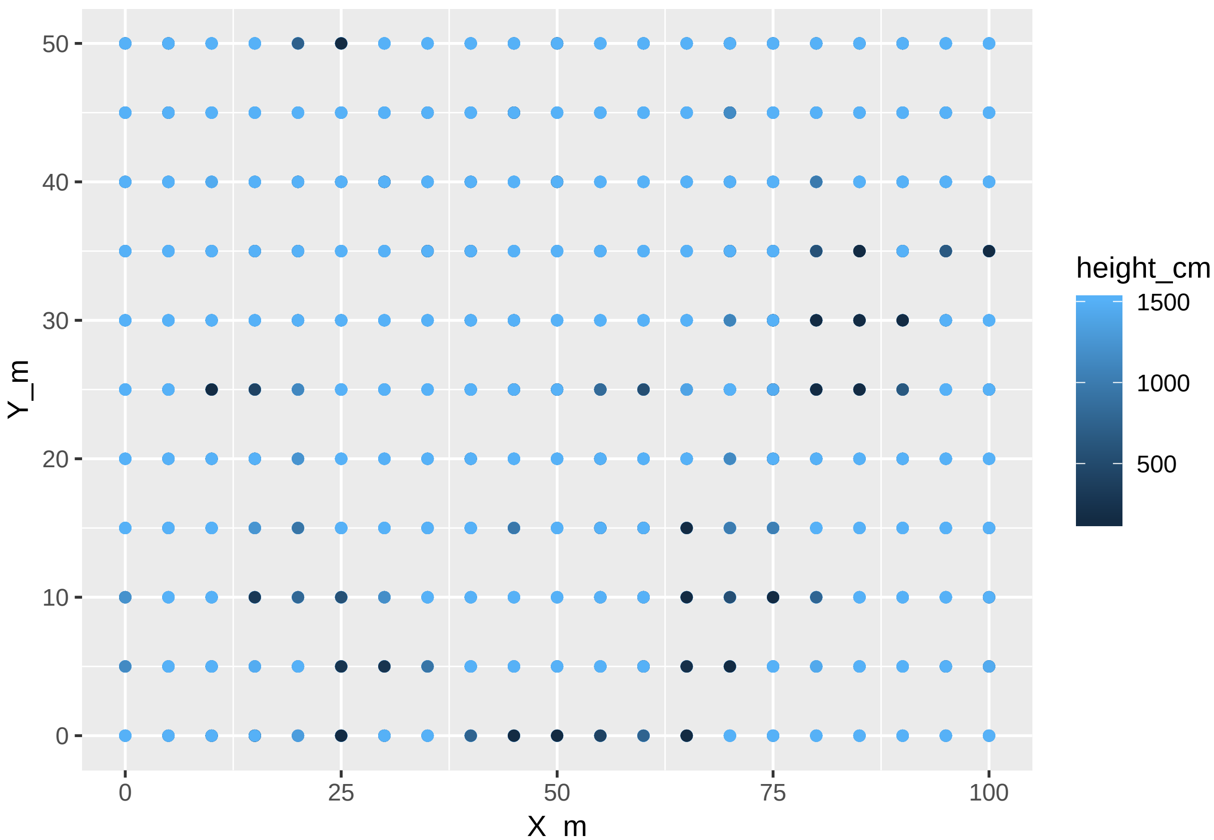
ggplot spatial data: La Selva CARBONO plots data

```
library(tidyverse); library(lubridate)
carb <- read_csv("data/claros1999_2012fulldataset.csv", skip = 5)
carb %>% glimpse()
```

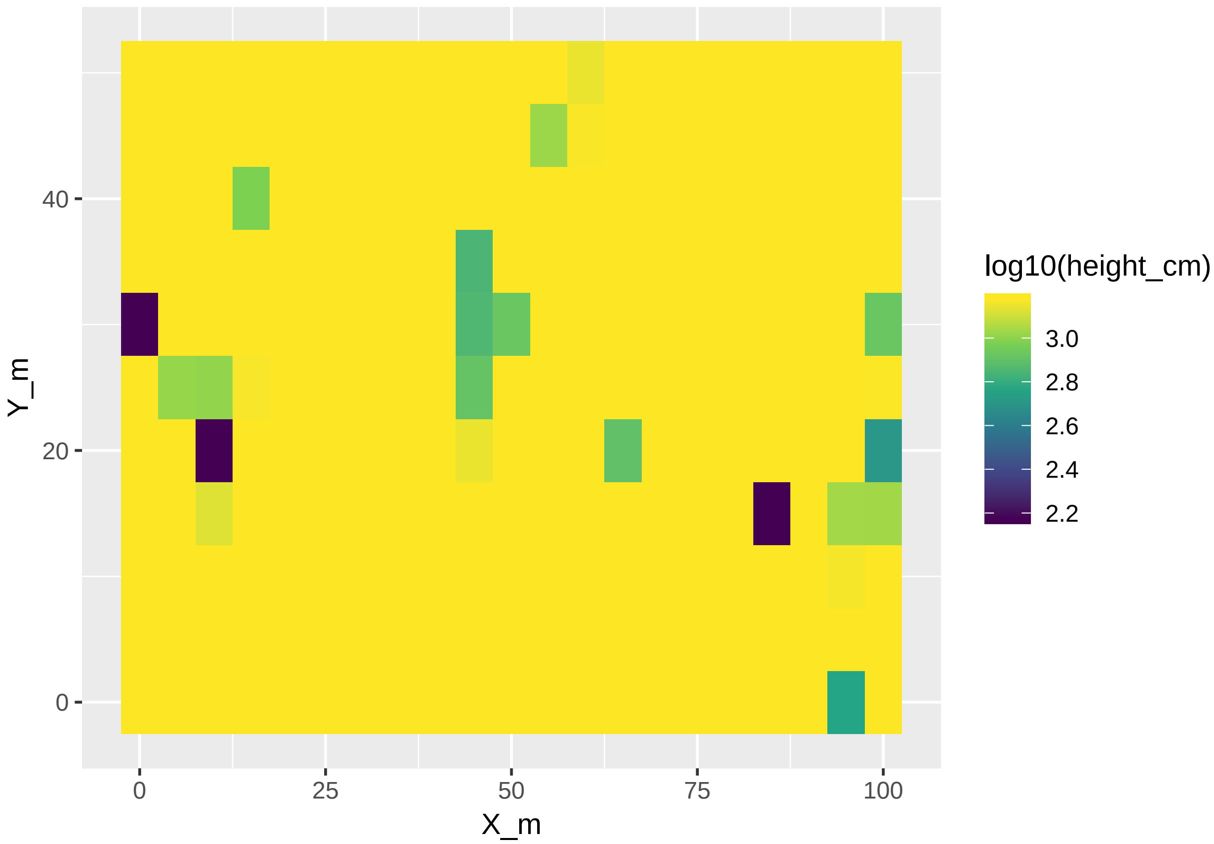
```
## Observations: 56,133
## Variables: 7
## $ `*Year`   <int> 1999, 1999, 1999, 1999, 1999, 1999, 1999, 1999, 1999...
## $ plot      <chr> "A1", "A1", "A1", "A1", "A1", "A1", "A1", "A1", "A1"...
## $ Y_m       <int> 15, 20, 30, 20, 0, 35, 30, 20, 25, 30, 30, 40, 25, 2...
## $ X_m       <int> 85, 10, 0, 100, 95, 45, 45, 65, 45, 100, 50, 15, 10,...
## $ height_cm <int> 150, 150, 150, 507, 579, 695, 712, 788, 811, 835, 83...
## $ Date      <chr> "12-Jul-99", "12-Jul-99", "12-Jul-99", "12-Jul-99", ...
## $ Comments  <chr> "Not done in 1999", "Not done in 1999", "Not done in...
```

```
carb <- carb %>%
  rename(year=`*Year`) %>%
  mutate(date = parse_date_time(Date, '%d-%m-%y')) %>%
  select(-Date)

carb %>% # bad way
  filter(year==1999) %>%
  ggplot(data=., aes(X_m, Y_m, color=height_cm))+
```

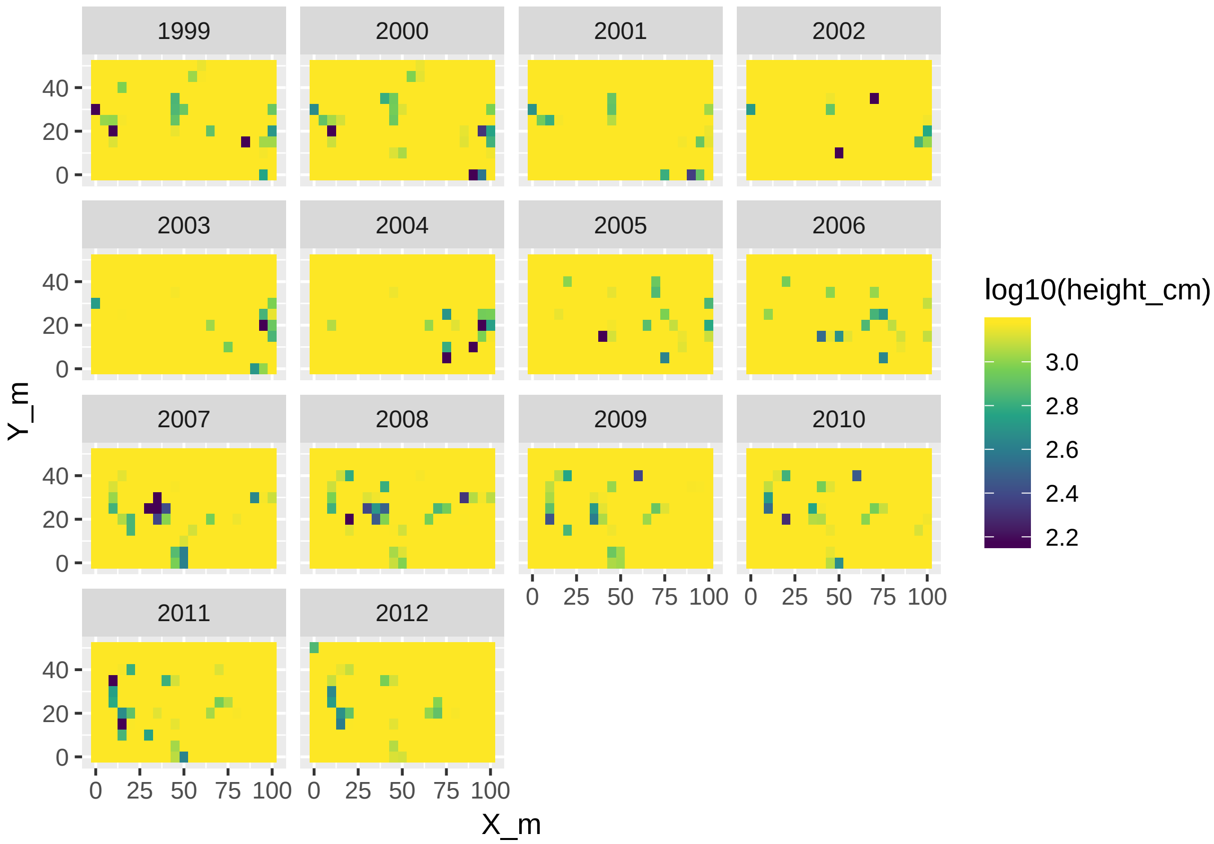
```
  geom_point()
```




```
carb %>% # better way
  filter(year==1999 & plot=="A1") %>%
  ggplot(data=., aes(X_m, Y_m, fill=log10(height_cm)))+
  geom_raster()+
  scale_fill_viridis_c()
```




```
library(tidyverse);
carb %>% # visualize through time
  filter(plot=='A1') %>%
  ggplot(data=., aes(X_m, Y_m, fill=log10(height_cm)))+
  geom_raster()+
  scale_fill_viridis_c()+
  facet_wrap(~year)
```




```
#REALLY VISUALIZE it
# library(gganimate)
# carb %>% # visualize through time
#   filter(year==1999) %>%
#   ggplot(data=., aes(X_m, Y_m, fill=log10(height_cm)))+
#   geom_raster()+
#   scale_fill_viridis_c()+
#   facet_wrap(~plot)
```

```
# library(gganimate)
# p <- carb %>% # visualize through time
#   ggplot(data=., aes(X_m, Y_m, fill=log10(height_cm), frame=year))+
#   geom_raster()+
#   coord_equal()+
#   scale_fill_viridis_c("Canopy Height [log cm]", option = 'B')+
#   facet_wrap(~plot)+
#   labs(title='Year: {frame_time}')
# gganimate(p, "outputs/carbono_plot_heights.gif")
```

Plot distributions

```
library(tidyverse)

hist(carb$height_cm) # old base-R way to plot histogram
plot(density(carb$height_cm)) # base-R way to plot kernel density
```

```
carb %>% glimpse
carb %>% ggplot(data=., aes(x=height_cm))+geom_histogram()
carb %>%
  filter(near(year,2000,tol = 0.1)) %>% # filtering for numbers can be tricky, use near to specify a fi
  ggplot(data=., aes(x=height_cm))+geom_histogram()+facet_wrap(~plot)
carb %>%
  filter(near(year,2000,tol = 0.1)) %>%
  ggplot(data=., aes(x= log1p(height_cm)))+
  geom_histogram(bins = 10)+
  scale_y_continuous(trans="log1p")+
  facet_wrap(~plot)
```

Spatiotemporal example

Plotting monthly ozone concentrations

```
library(tidyverse)
nasa                 # so it's not a tibble
```

```
## Source: local array [41,472 x 4]
## D: lat [dbl, 24]
## D: long [dbl, 24]
## D: month [int, 12]
## D: year [int, 6]
## M: cloudhigh [dbl]
## M: cloudlow [dbl]
## M: cloudmid [dbl]
## M: ozone [dbl]
## M: pressure [dbl]
## M: surftemp [dbl]
## M: temperature [dbl]
```

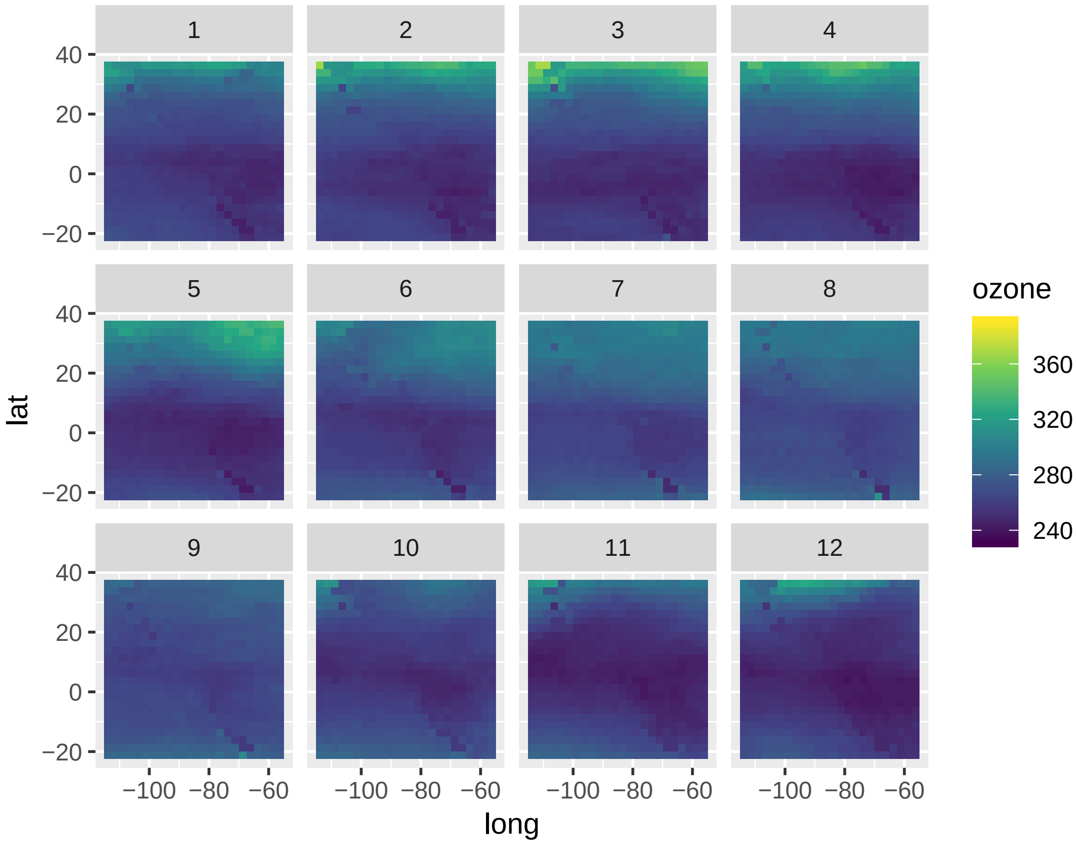
```
nasa %>% class      # what is the class of the data?
```

```
## [1] "tbl_cube"
```

```
nasa %>% glimpse() # examine the types of data in the object
```

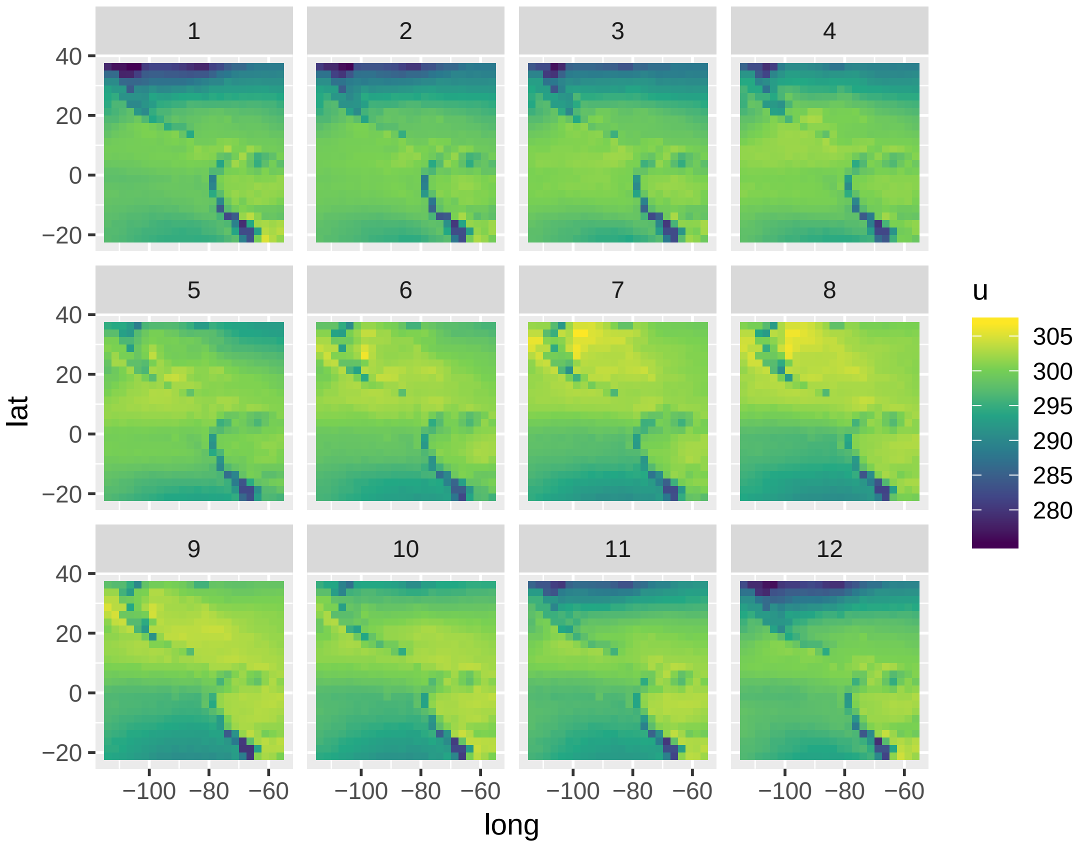
```
## List of 2
##  $ mets:List of 7
##   ..$ cloudhigh  : num [1:24, 1:24, 1:12, 1:6] 26 20 16 13 7.5 8 14.5 19.5 22.5 21 ...
##   ..$ cloudlow   : num [1:24, 1:24, 1:12, 1:6] 7.5 11.5 16.5 20.5 26 30 29.5 26.5 27.5 26 ...
##   ..$ cloudmid   : num [1:24, 1:24, 1:12, 1:6] 34.5 32.5 26 14.5 10.5 9.5 11 17.5 18.5 16.5 ...
##   ..$ ozone      : num [1:24, 1:24, 1:12, 1:6] 304 304 298 276 274 264 258 252 250 250 ...
##   ..$ pressure   : num [1:24, 1:24, 1:12, 1:6] 835 940 960 990 1000 1000 1000 1000 1000 1000 ...
##   ..$ surftemp   : num [1:24, 1:24, 1:12, 1:6] 273 280 285 289 292 ...
##   ..$ temperature: num [1:24, 1:24, 1:12, 1:6] 272 282 285 291 293 ...
##  $ dims:List of 4
##   ..$ lat  : num [1:24] 36.2 33.7 31.2 28.7 26.2 ...
##   ..$ long : num [1:24] -114 -111 -109 -106 -104 ...
##   ..$ month: int [1:12] 1 2 3 4 5 6 7 8 9 10 ...
##   ..$ year : int [1:6] 1995 1996 1997 1998 1999 2000
##  - attr(*, "class")= chr "tbl_cube"
```

```
nasa %>%                          # spatiotemporal tbl_cube
  as_tibble() %>%                 # convert to tibble for ggplot
  ggplot(data=., aes(long,lat))+ #
  geom_raster(aes(fill=ozone))+  # plot the ozone on a lat-long grid
  coord_equal()+                  #
  scale_fill_viridis_c() +
  facet_wrap(~month)
```

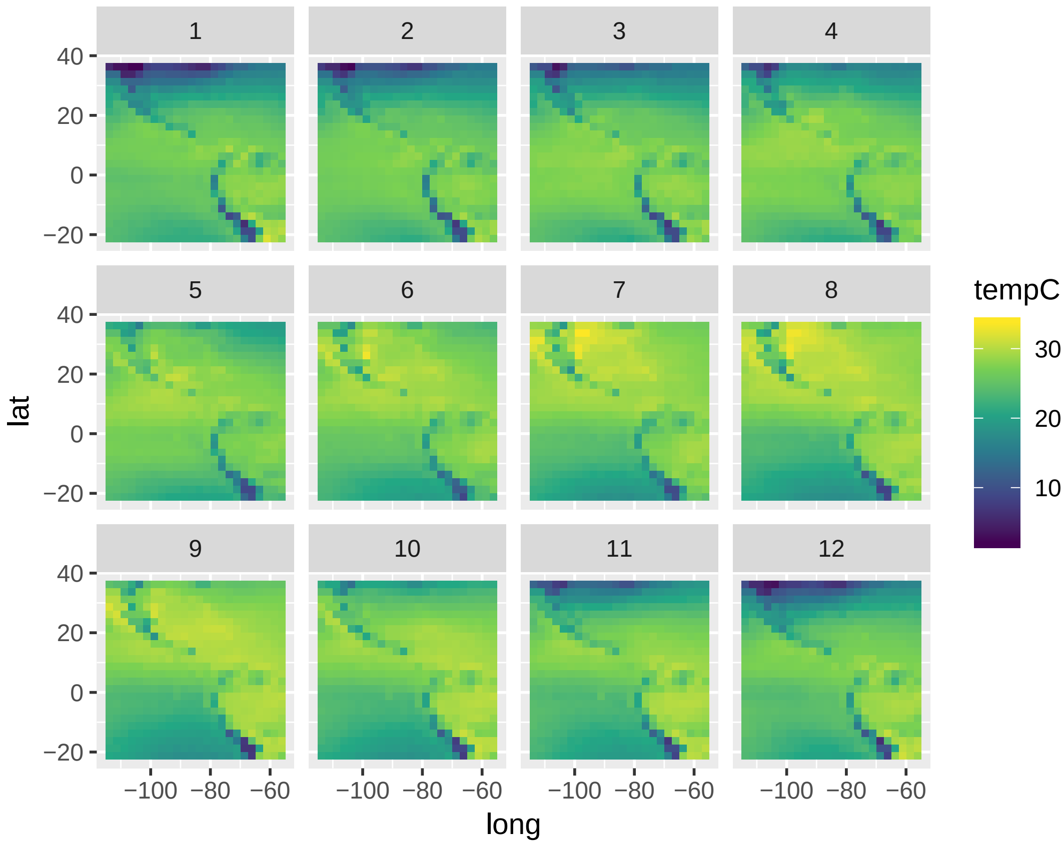


Plot the mean monthly temperature

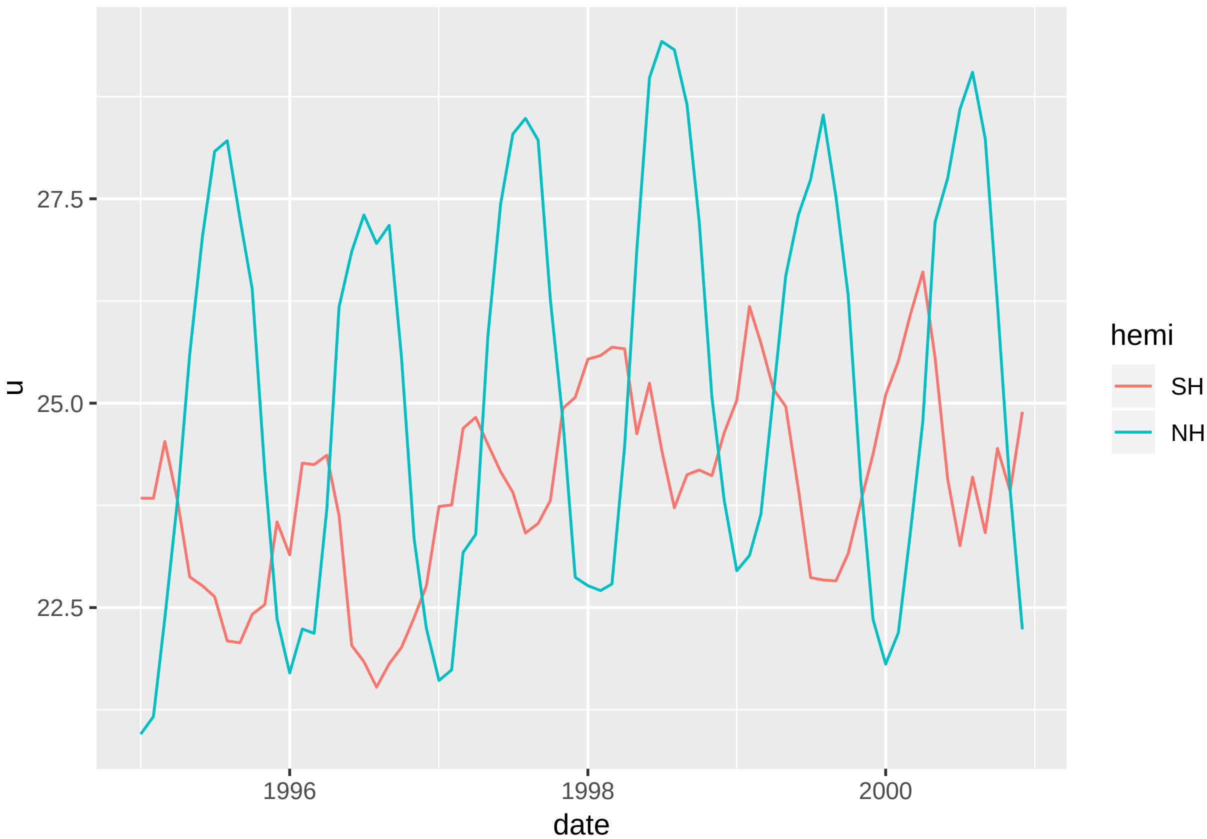
```
nasa %>%
  as_tibble() %>%
  group_by(lat,long, month) %>%
  summarize(u=mean(temperature,na.rm=T)) %>%
  ggplot(data=., aes(long,lat))+
  geom_raster(aes(fill=u))+
  coord_equal()+
  scale_fill_viridis_c() +
  facet_wrap(~month)
```




```r
nasa %>%
  as_tibble() %>%
  group_by(lat,long, month) %>%
  summarize(u=mean(temperature,na.rm=T)) %>%
  ungroup() %>%
  mutate(tempC = u - 273.15) %>%
  ggplot(data=., aes(long,lat))+
  geom_raster(aes(fill=tempC))+
  coord_equal()+
  scale_fill_viridis_c() +
  facet_wrap(~month)
```




```
nasa %>%
  as_tibble() %>%
  group_by(lat,long, year,month) %>%
  # summarize(u=mean(temperature,na.rm=T)) %>%
  # ungroup() %>%
  mutate(tempC = temperature - 273.15) %>%
  mutate(hemi = cut(lat,breaks = c(-Inf,0,Inf),labels = c("SH","NH"))) %>%
  group_by(hemi,year,month) %>%
  summarize(u=mean(tempC,na.rm=T)) %>%
  ungroup() %>%
  mutate(date=parse_date_time(paste(year,month,1),'ymd')) %>%
  ggplot(data=., aes(date,u,color=hemi))+
  geom_line()+
  scale_fill_viridis_c()
```

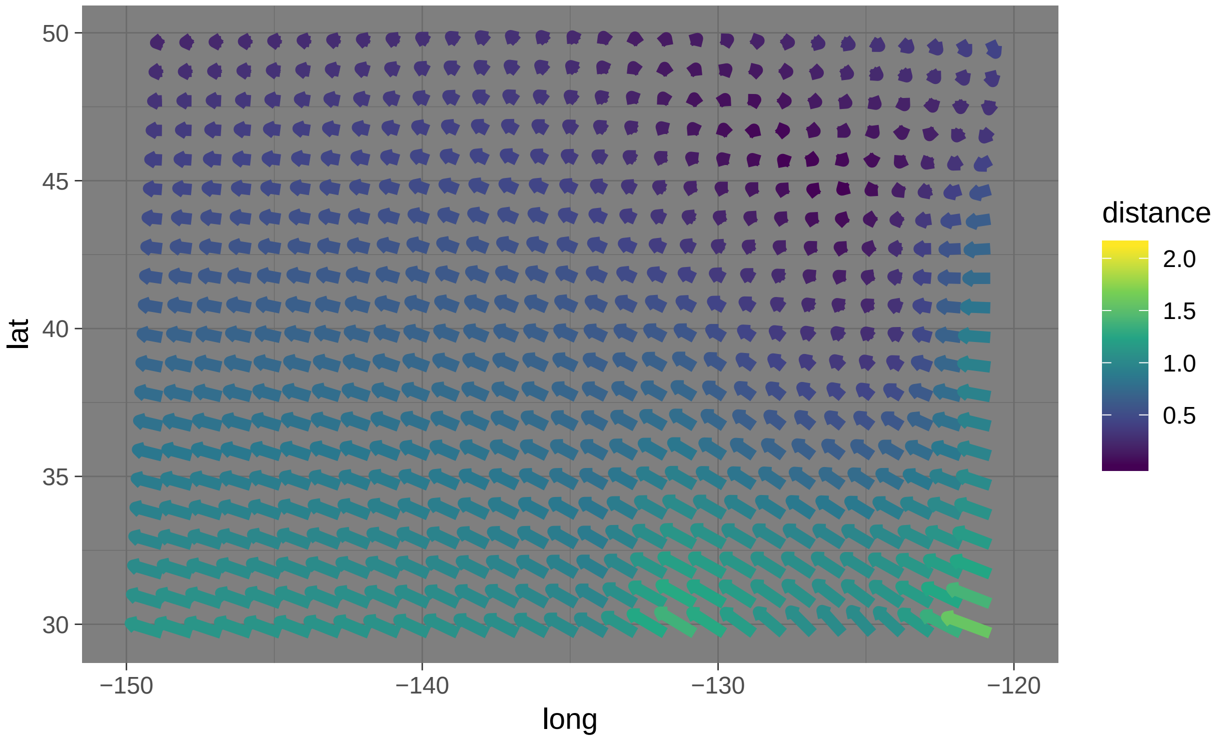


Seals

```
library(tidyverse)
data("seals")      # load example seals data
seals %>% glimpse # check the data types
```

```
## Observations: 1,155
## Variables: 4
## $ lat        <dbl> 29.7, 30.7, 31.7, 32.7, 33.7, 34.7, 35.7, 36.7, 37....
## $ long       <dbl> -172.8, -172.8, -172.8, -172.8, -172.8, -172.8, -17...
## $ delta_long <dbl> -0.91504624, -0.86701252, -0.81892489, -0.77077630,...
## $ delta_lat  <dbl> 0.143475254, 0.128388724, 0.113232481, 0.098020371,...
```

```
seals %>%
  mutate(distance=sqrt(delta_long**2 + delta_lat**2)) %>% # calc the distance travelled
  ggplot(., aes(long, lat, color=distance)) +
  geom_segment(aes(xend = long + delta_long, yend = lat + delta_lat), # add a vector plot
               arrow = arrow(length = unit(0.1,"cm")),lwd=2) +
  coord_equal()+ # fix the coords
  # borders("usa")+
  scale_x_continuous(limits = c(-150,-120))+
  scale_color_viridis_c()+
  theme_dark()
```



Plotting hurrican tracks

```r
library(tidyverse)
dplyr::storms
```

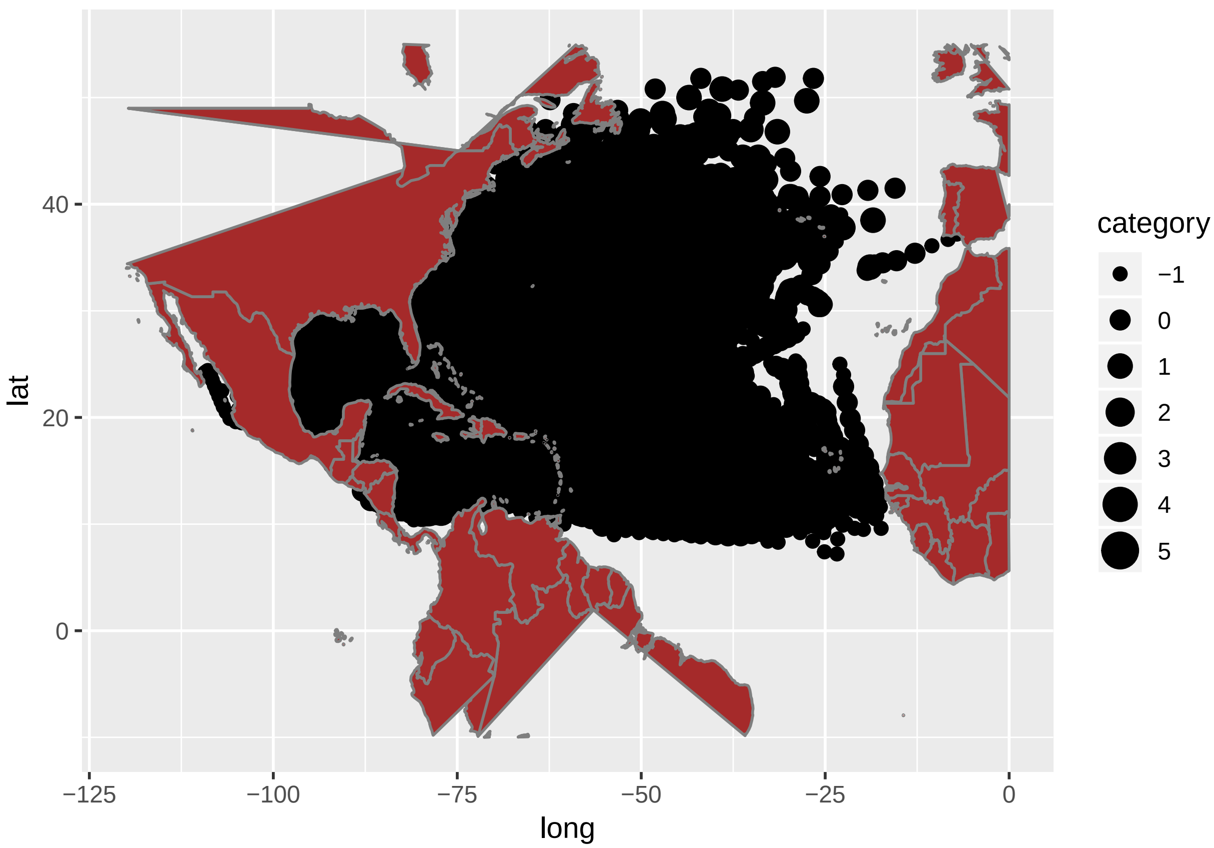
```
## # A tibble: 10,010 x 13
##    name   year month   day  hour   lat  long status          category  wind
##    <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>           <ord>    <int>
##  1 Amy    1975     6    27     0  27.5 -79   tropical depr~ -1          25
##  2 Amy    1975     6    27     6  28.5 -79   tropical depr~ -1          25
##  3 Amy    1975     6    27    12  29.5 -79   tropical depr~ -1          25
##  4 Amy    1975     6    27    18  30.5 -79   tropical depr~ -1          25
##  5 Amy    1975     6    28     0  31.5 -78.8 tropical depr~ -1          25
##  6 Amy    1975     6    28     6  32.4 -78.7 tropical depr~ -1          25
##  7 Amy    1975     6    28    12  33.3 -78   tropical depr~ -1          25
##  8 Amy    1975     6    28    18  34   -77   tropical depr~ -1          30
##  9 Amy    1975     6    29     0  34.4 -75.8 tropical storm 0           35
## 10 Amy    1975     6    29     6  34   -74.8 tropical storm 0           40
## # ... with 10,000 more rows, and 3 more variables: pressure <int>,
## #   ts_diameter <dbl>, hu_diameter <dbl>
```

```r
names(storms)
```

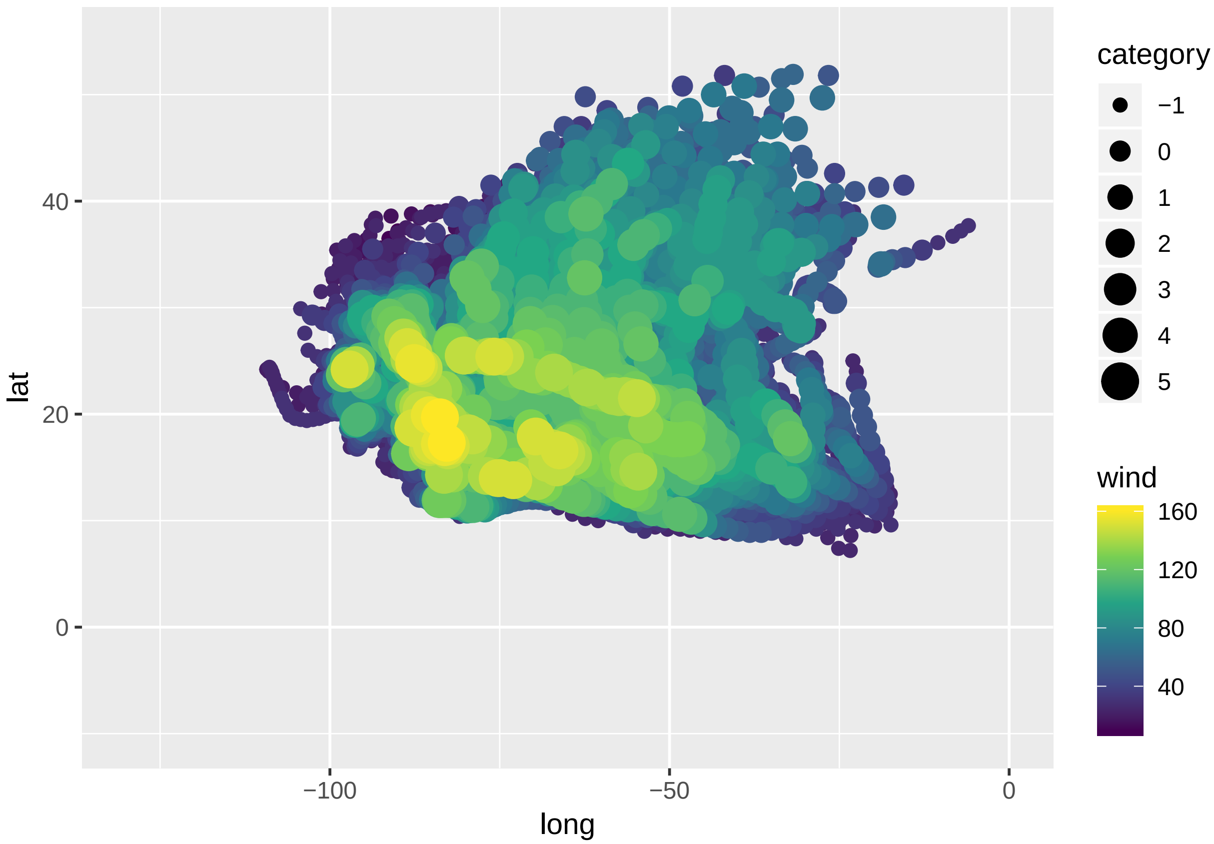
```
##  [1] "name"        "year"        "month"       "day"         "hour"
##  [6] "lat"         "long"        "status"      "category"    "wind"
## [11] "pressure"    "ts_diameter" "hu_diameter"
```

```r
# bad!
storms %>%
  ggplot(data=., aes(long,lat,size=category))+
```

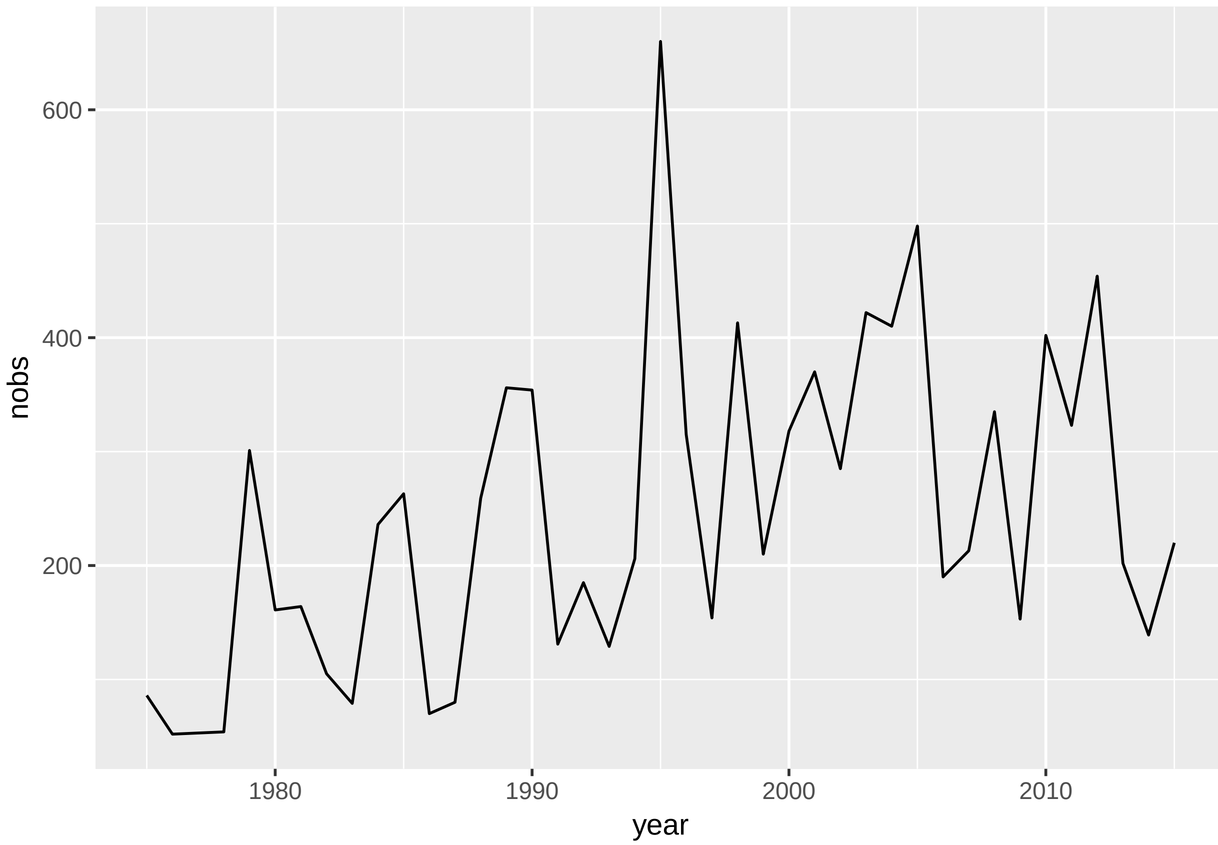
```r
  geom_point()+
  # borders("world")+
  scale_x_continuous(limits=c(-120,0))+
  scale_y_continuous(limits = c(-10,55))+
  borders('world', fill = "brown")
```



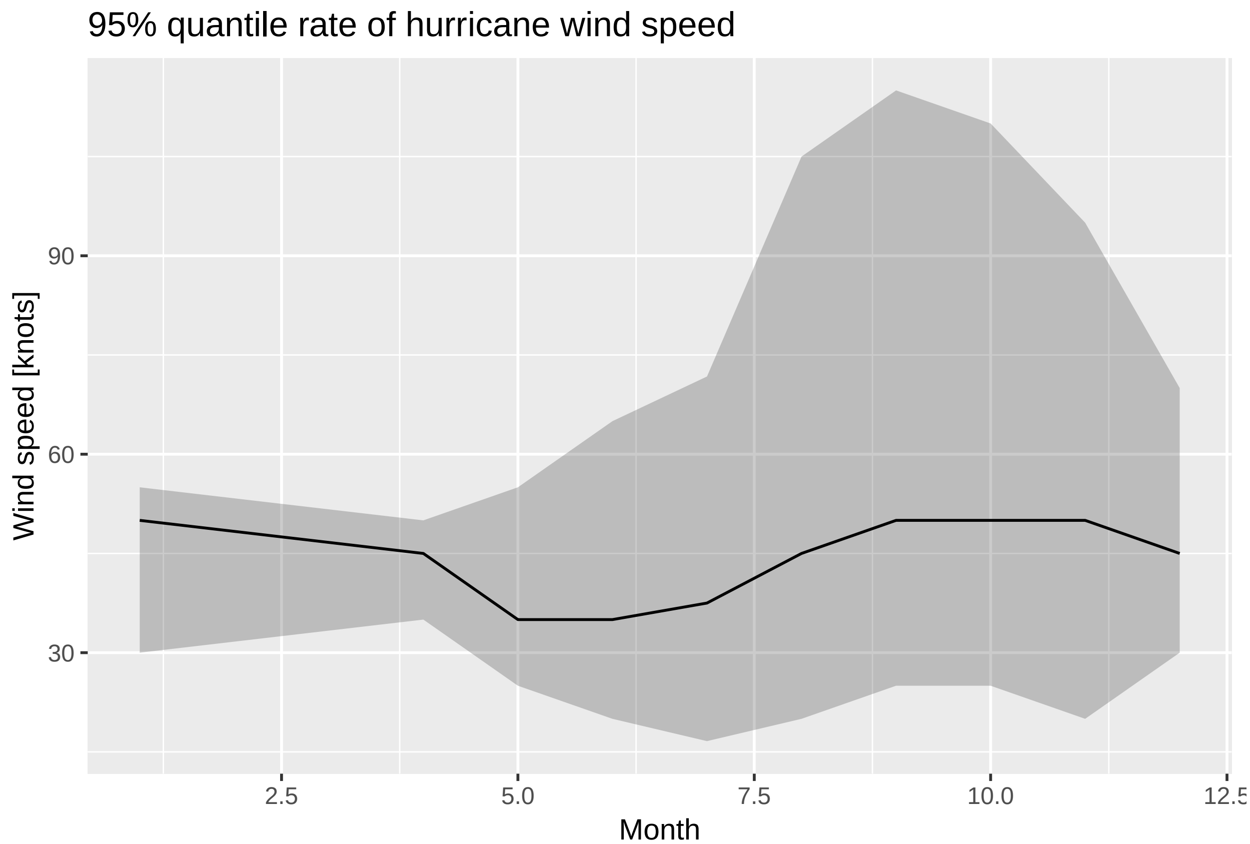

```r
# bad!
storms %>%
  arrange(wind) %>%
  ggplot(data=., aes(long,lat,size=category,color=wind))+
  # borders("coast")+
  geom_point()+
  scale_color_viridis_c()+
  scale_x_continuous(limits=c(-130,0))+
  scale_y_continuous(limits = c(-10,55))
```



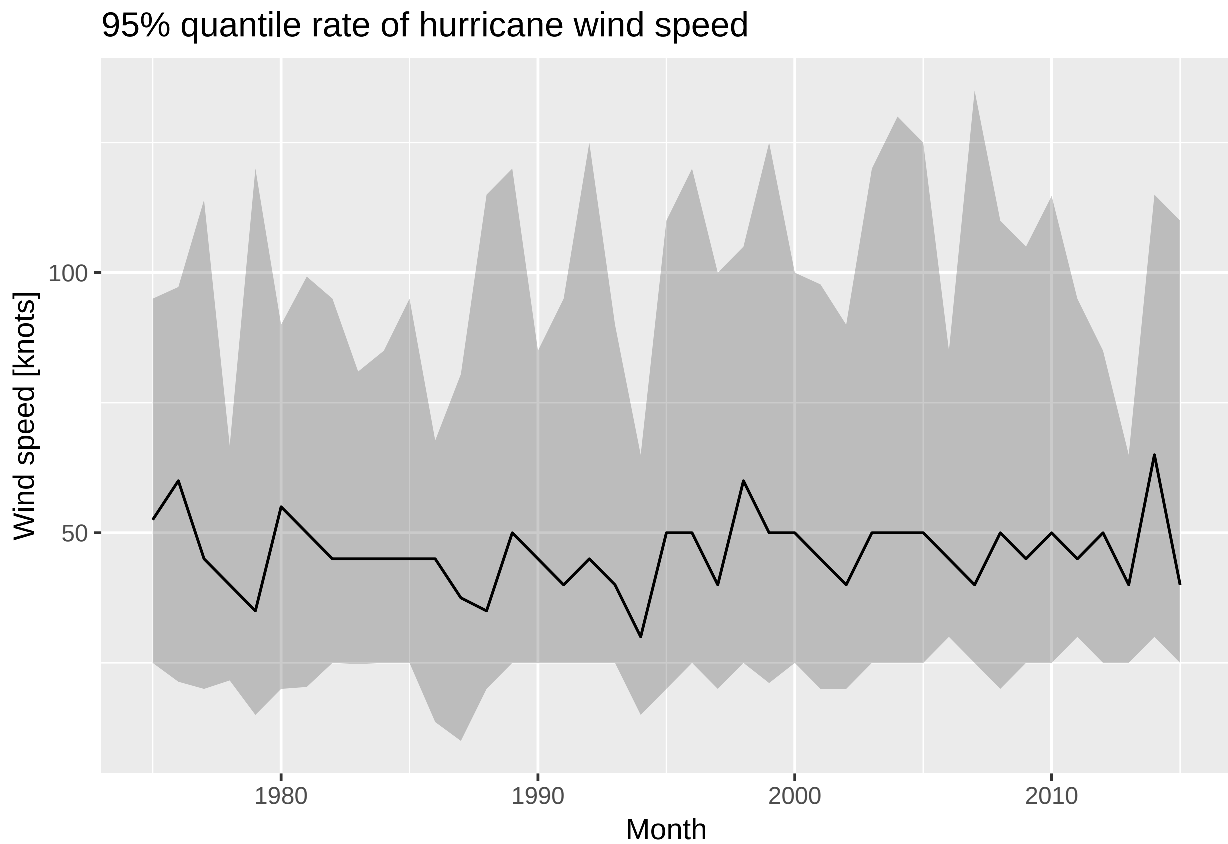

```
storms %>% group_by(year) %>% summarize(nobs=n()) %>% ggplot(data=., aes(year,nobs))+geom_line()
```



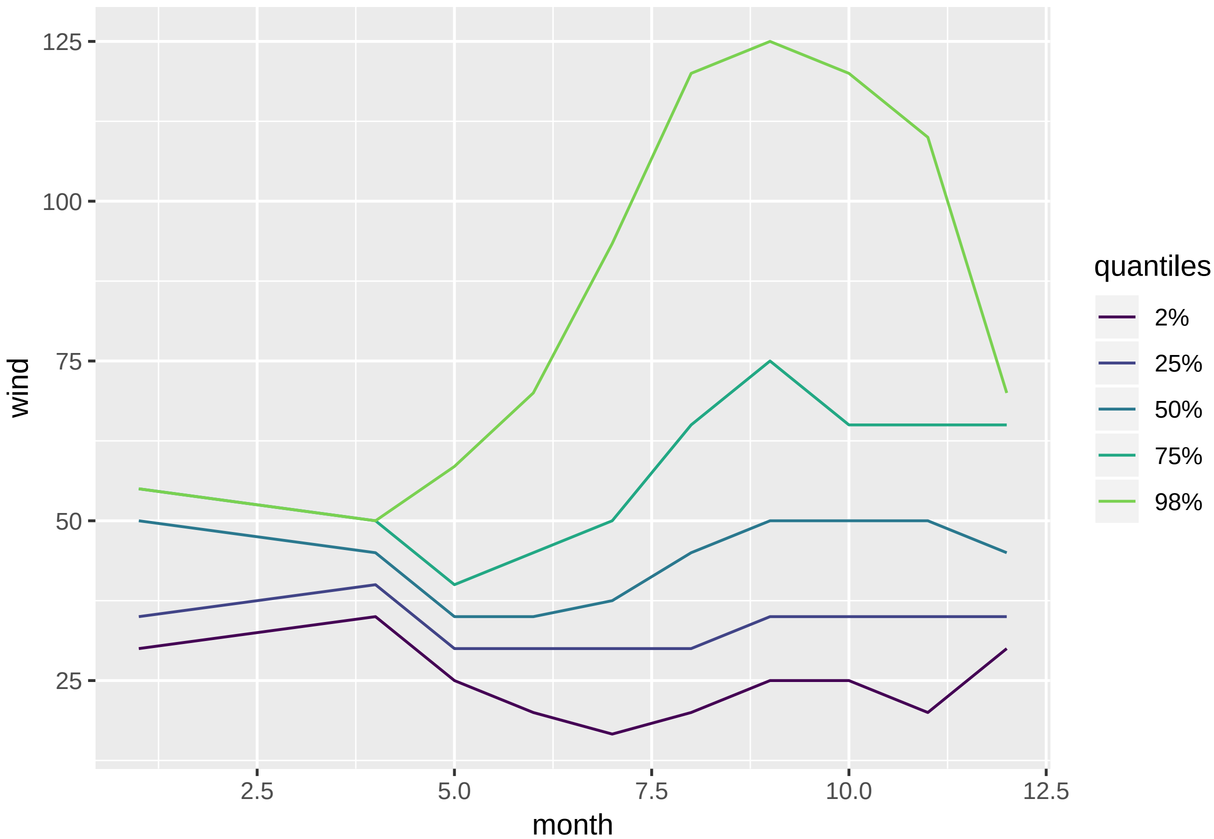

```
storms %>% group_by(month) %>% summarize(wind_25=quantile(wind,0.025),
                                          wind_50=median(wind),
                                          wind_75=quantile(wind, 0.95)) %>%
  ungroup() %>%
  ggplot(data=., aes(month, wind_50))+
  geom_ribbon(aes(x=month, ymax=wind_75, ymin=wind_25),lty=0,alpha=0.25)+
  geom_line()+
  labs(x="Month",y="Wind speed [knots]",title = "95% quantile rate of hurricane wind speed")
```




```
# swap 'month' for 'year'
storms %>% group_by(year) %>% summarize(wind_25=quantile(wind,0.025),
                                        wind_50=median(wind),
                                        wind_75=quantile(wind, 0.95)) %>%
  ungroup() %>%
  ggplot(data=., aes(year, wind_50))+
  geom_ribbon(aes(x=year, ymax=wind_75, ymin=wind_25),lty=0,alpha=0.25)+
  geom_line()+
  labs(x="Month",y="Wind speed [knots]",title = "95% quantile rate of hurricane wind speed")
```



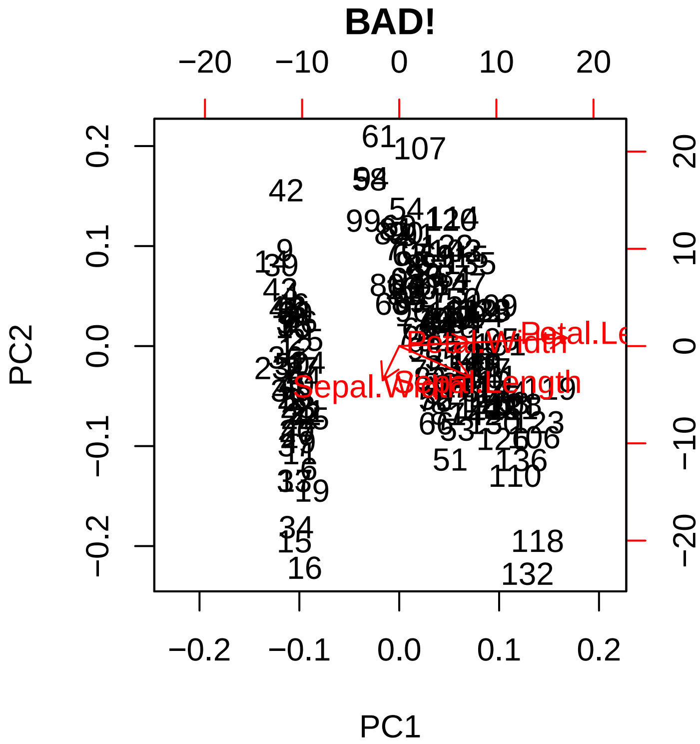

```r
# advanced!
p = c(0.025, 0.25,0.5,0.75,0.975)
storms %>%
  group_by(month) %>%
  summarise(quantiles = list(sprintf("%1.0f%%", p*100)),
            wind = list(quantile(wind, p))) %>%
  unnest %>%
  ggplot(data=., aes(month, wind, color=quantiles))+
  geom_line()+
  scale_color_viridis_d(end=0.8)
```



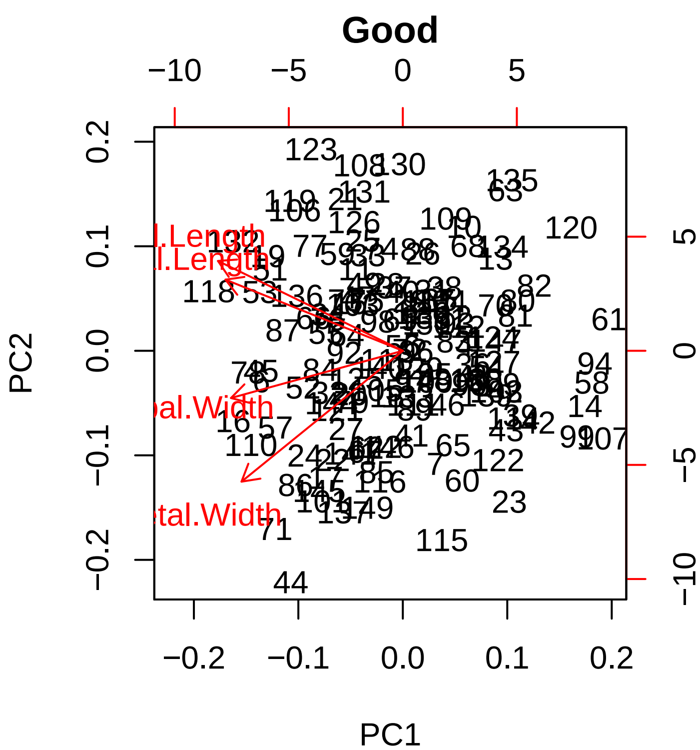
PCA example with columns scaling —————————————————

—-

```
library(tidyverse);
rm(iris); data('iris'); # restoring default version of iris dataset

iris %>%
  prcomp(~.-Species, data=.) %>% # run principle components on all vars, except Species
  biplot(main='BAD!')                            # plot biplot
```




```r
iris %>%
  group_by(Species) %>%           # grouping var
  mutate_all(scale) %>%           # center vars over zero, and divide by sd
  ungroup() %>%
  prcomp(~.-Species, data=.) %>% # run principle components on all vars, except Species
  biplot(main='Good')                        # plot biplot
```



Super advanced dplyr

inspired by: Suzan Baert and modified from her github repo tutorial on advanced dplyr

```
# using !! "bang"
vars <- c("lat","long","wind")
storms %>% select(!!vars)
```

```
## # A tibble: 10,010 x 3
##      lat  long  wind
##    <dbl> <dbl> <int>
##  1  27.5 -79      25
##  2  28.5 -79      25
##  3  29.5 -79      25
##  4  30.5 -79      25
##  5  31.5 -78.8    25
##  6  32.4 -78.7    25
##  7  33.3 -78      25
##  8  34   -77      30
##  9  34.4 -75.8    35
## 10  34   -74.8    40
## # ... with 10,000 more rows
```

```
# select columns by regex
who %>% names # lots of column names
```

```
##  [1] "country"      "iso2"         "iso3"         "year"
##  [5] "new_sp_m014"  "new_sp_m1524" "new_sp_m2534" "new_sp_m3544"
##  [9] "new_sp_m4554" "new_sp_m5564" "new_sp_m65"   "new_sp_f014"
## [13] "new_sp_f1524" "new_sp_f2534" "new_sp_f3544" "new_sp_f4554"
## [17] "new_sp_f5564" "new_sp_f65"   "new_sn_m014"  "new_sn_m1524"
## [21] "new_sn_m2534" "new_sn_m3544" "new_sn_m4554" "new_sn_m5564"
## [25] "new_sn_m65"   "new_sn_f014"  "new_sn_f1524" "new_sn_f2534"
## [29] "new_sn_f3544" "new_sn_f4554" "new_sn_f5564" "new_sn_f65"
## [33] "new_ep_m014"  "new_ep_m1524" "new_ep_m2534" "new_ep_m3544"
## [37] "new_ep_m4554" "new_ep_m5564" "new_ep_m65"   "new_ep_f014"
## [41] "new_ep_f1524" "new_ep_f2534" "new_ep_f3544" "new_ep_f4554"
## [45] "new_ep_f5564" "new_ep_f65"   "newrel_m014"  "newrel_m1524"
## [49] "newrel_m2534" "newrel_m3544" "newrel_m4554" "newrel_m5564"
## [53] "newrel_m65"   "newrel_f014"  "newrel_f1524" "newrel_f2534"
## [57] "newrel_f3544" "newrel_f4554" "newrel_f5564" "newrel_f65"
```

```
who %>% select(country, year, matches("*2534")) # select country, year, and columns with '2534' in the
```

```
## # A tibble: 7,240 x 10
##    country      year new_sp_m2534 new_sp_f2534 new_sn_m2534 new_sn_f2534
##    <chr>       <int>        <int>        <int>        <int>        <int>
##  1 Afghanistan  1980           NA           NA           NA           NA
##  2 Afghanistan  1981           NA           NA           NA           NA
##  3 Afghanistan  1982           NA           NA           NA           NA
##  4 Afghanistan  1983           NA           NA           NA           NA
##  5 Afghanistan  1984           NA           NA           NA           NA
##  6 Afghanistan  1985           NA           NA           NA           NA
##  7 Afghanistan  1986           NA           NA           NA           NA
```

```
##  8 Afghanistan  1987           NA           NA           NA           NA
##  9 Afghanistan  1988           NA           NA           NA           NA
## 10 Afghanistan  1989           NA           NA           NA           NA
## # ... with 7,230 more rows, and 4 more variables: new_ep_m2534 <int>,
## #   new_ep_f2534 <int>, newrel_m2534 <int>, newrel_f2534 <int>
```

```r
# rename columns with regex
library(stringr);
iris %>%
  as_tibble() %>%
  rename_all(tolower) %>%
  rename_all(~str_replace_all(., "\\.","_"))
```

```
## # A tibble: 150 x 5
##    sepal_length sepal_width petal_length petal_width species
##           <dbl>       <dbl>        <dbl>       <dbl> <fct>
##  1          5.1         3.5          1.4         0.2 setosa
##  2          4.9         3            1.4         0.2 setosa
##  3          4.7         3.2          1.3         0.2 setosa
##  4          4.6         3.1          1.5         0.2 setosa
##  5          5           3.6          1.4         0.2 setosa
##  6          5.4         3.9          1.7         0.4 setosa
##  7          4.6         3.4          1.4         0.3 setosa
##  8          5           3.4          1.5         0.2 setosa
##  9          4.4         2.9          1.4         0.2 setosa
## 10          4.9         3.1          1.5         0.1 setosa
## # ... with 140 more rows
```

```r
# mutate *observation* names
storms %>%
  select(name,year,status) %>%
  mutate_all(tolower) %>% # Amy -> amy
  mutate_all(~str_replace_all(., " ","_")) # 'tropical depression' -> 'tropical_depression'
```

```
## # A tibble: 10,010 x 3
##    name  year  status
##    <chr> <chr> <chr>
##  1 amy   1975  tropical_depression
##  2 amy   1975  tropical_depression
##  3 amy   1975  tropical_depression
##  4 amy   1975  tropical_depression
##  5 amy   1975  tropical_depression
##  6 amy   1975  tropical_depression
##  7 amy   1975  tropical_depression
##  8 amy   1975  tropical_depression
##  9 amy   1975  tropical_storm
## 10 amy   1975  tropical_storm
## # ... with 10,000 more rows
```

```r
# find highest values
storms %>%
  top_n(5, wind) # storms with 5 highest windspeeds
```

```
## # A tibble: 7 x 13
##   name   year month   day  hour   lat  long status category  wind pressure
##   <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>  <ord>    <int>    <int>
```

```
## 1 Gilb~  1988     9     14     0  19.7 -83.8 hurri~ 5          160      888
## 2 Gilb~  1988     9     14     6  19.9 -85.3 hurri~ 5          155      889
## 3 Mitch  1998    10     26    18  16.9 -83.1 hurri~ 5          155      905
## 4 Mitch  1998    10     27     0  17.2 -83.8 hurri~ 5          155      910
## 5 Rita   2005     9     22     3  24.7 -87.3 hurri~ 5          155      895
## 6 Rita   2005     9     22     6  24.8 -87.6 hurri~ 5          155      897
## 7 Wilma  2005    10     19    12  17.3 -82.8 hurri~ 5          160      882
## # ... with 2 more variables: ts_diameter <dbl>, hu_diameter <dbl>
```

```r
# making new vars from conditions
starwars %>%
  select(name, species, homeworld, birth_year, hair_color) %>%
  mutate(new_group = case_when(
    species == "Droid" ~ "Robot",
    homeworld == "Tatooine" & hair_color == "blond" ~ "Blond Tatooinian",
    homeworld == "Tatooine" ~ "Other Tatooinian",
    hair_color == "blond" ~ "Blond non-Tatooinian",
    TRUE ~ "Other Human"))
```

```
## # A tibble: 87 x 6
##    name               species homeworld birth_year hair_color    new_group
##    <chr>              <chr>   <chr>          <dbl> <chr>         <chr>
##  1 Luke Skywalker     Human   Tatooine        19   blond         Blond Ta~
##  2 C-3PO              Droid   Tatooine       112   <NA>          Robot
##  3 R2-D2              Droid   Naboo           33   <NA>          Robot
##  4 Darth Vader        Human   Tatooine        41.9 none          Other Ta~
##  5 Leia Organa        Human   Alderaan        19   brown         Other Hu~
##  6 Owen Lars          Human   Tatooine        52   brown, grey   Other Ta~
##  7 Beru Whitesun lars Human   Tatooine        47   brown         Other Ta~
##  8 R5-D4              Droid   Tatooine        NA   <NA>          Robot
##  9 Biggs Darklighter  Human   Tatooine        24   black         Other Ta~
## 10 Obi-Wan Kenobi     Human   Stewjon         57   auburn, white Other Hu~
## # ... with 77 more rows
```

(6) SPATIAL METHODS + TIDYVERSE

```r
# Required libraries:
library(sf);
library(tidyverse);
library(lubridate);

dir.create("data/SouthAmerica")
```

```
## Warning in dir.create("data/SouthAmerica"): 'data/SouthAmerica' already
## exists
```

```r
unzip(zipfile = "data/SouthAmerica.zip", exdir = "data/SouthAmerica")
```

```
## Warning in unzip(zipfile = "data/SouthAmerica.zip", exdir = "data/
## SouthAmerica"): error 1 in extracting from zip file
```

```r
list.files('data/SouthAmerica/')
```

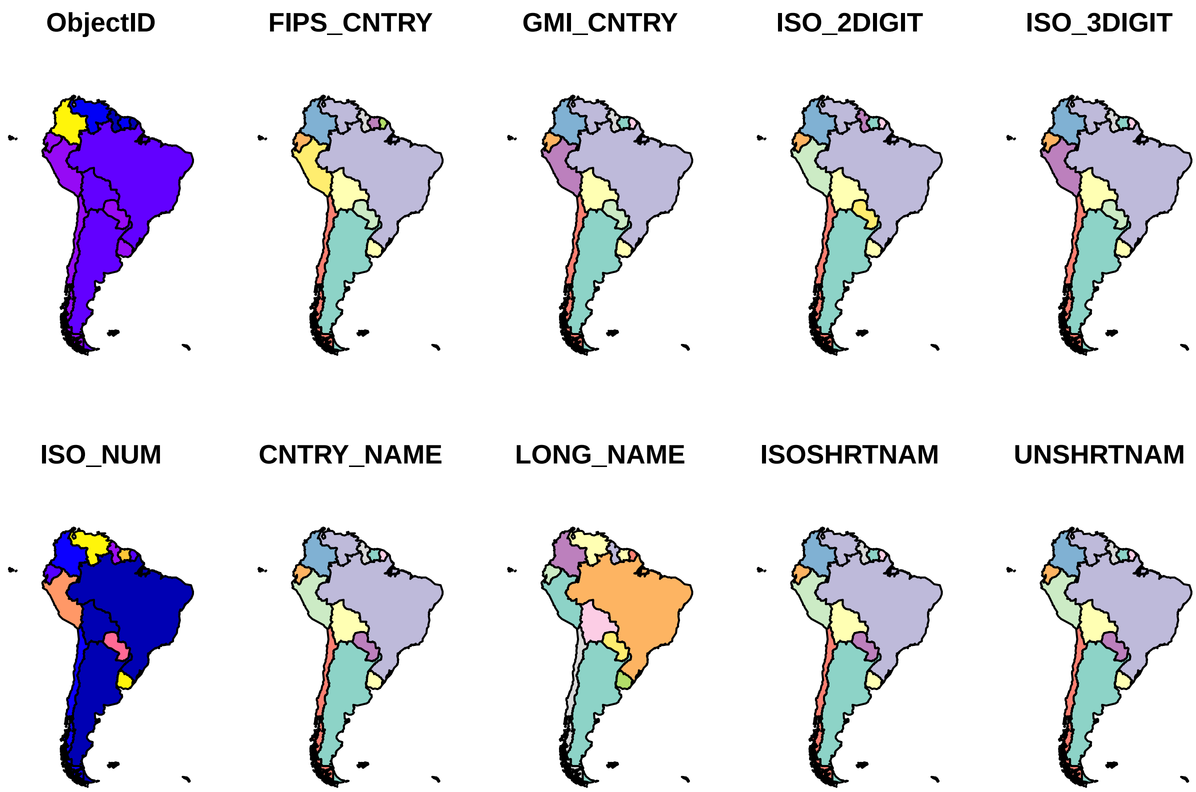
```
## [1] "SouthAmerica.dbf"     "SouthAmerica.prj"     "SouthAmerica.sbn"
## [4] "SouthAmerica.sbx"     "SouthAmerica.shp"     "SouthAmerica.shp.xml"
## [7] "SouthAmerica.shx"
```

```
SA <- sf::st_read("data/SouthAmerica/SouthAmerica.shp")
```

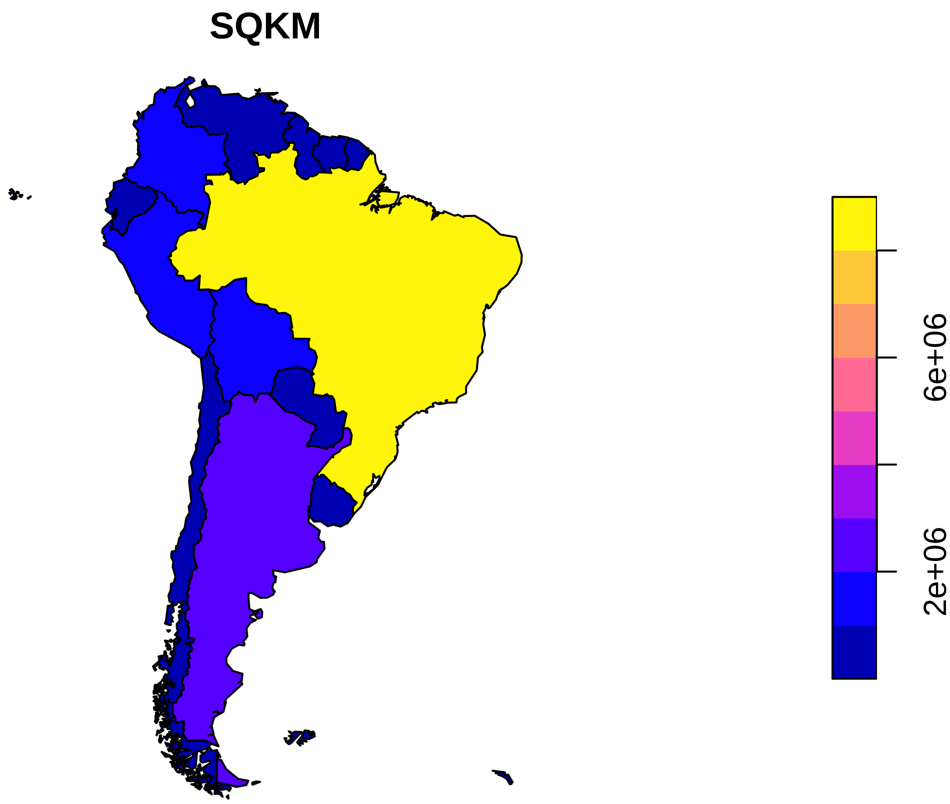
```
## Reading layer `SouthAmerica' from data source `/home/sami/srifai@gmail.com/work/Teaching/R_for_geograph
## Simple feature collection with 15 features and 18 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -10192560 ymin: -7508478 xmax: -3868796 ymax: 1396462
## epsg (SRID):    NA
## proj4string:    +proj=merc +lon_0=0 +lat_ts=0 +x_0=0 +y_0=0 +a=6371000 +b=6371000 +units=m +no_defs
```

```
plot(SA) # blah, not ideal
```

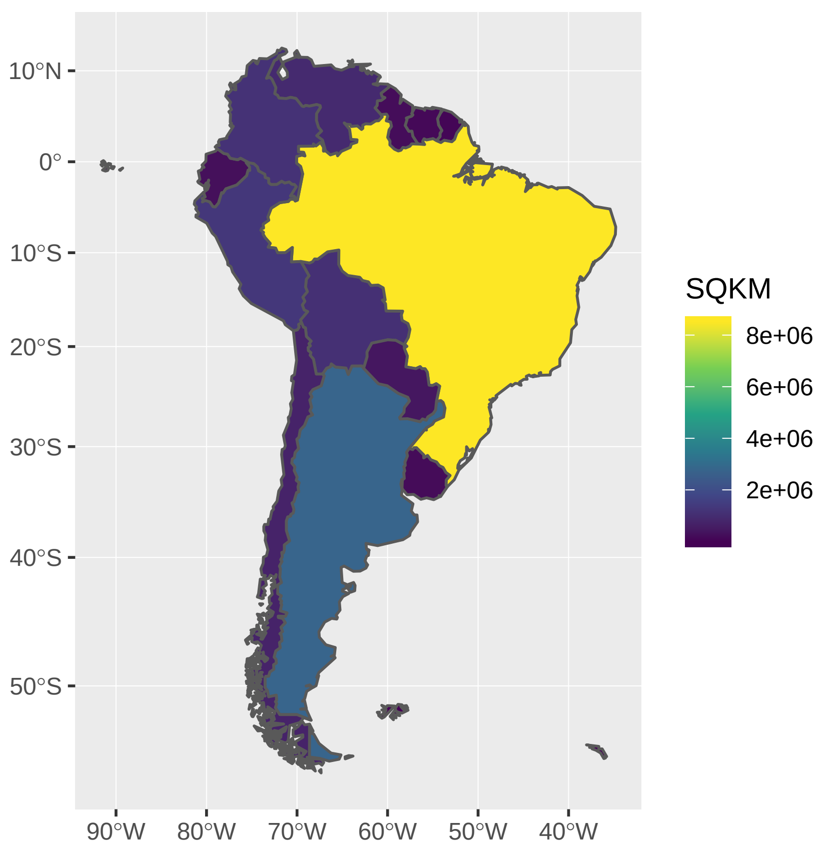
```
## Warning: plotting the first 10 out of 18 attributes; use max.plot = 18 to
## plot all
```



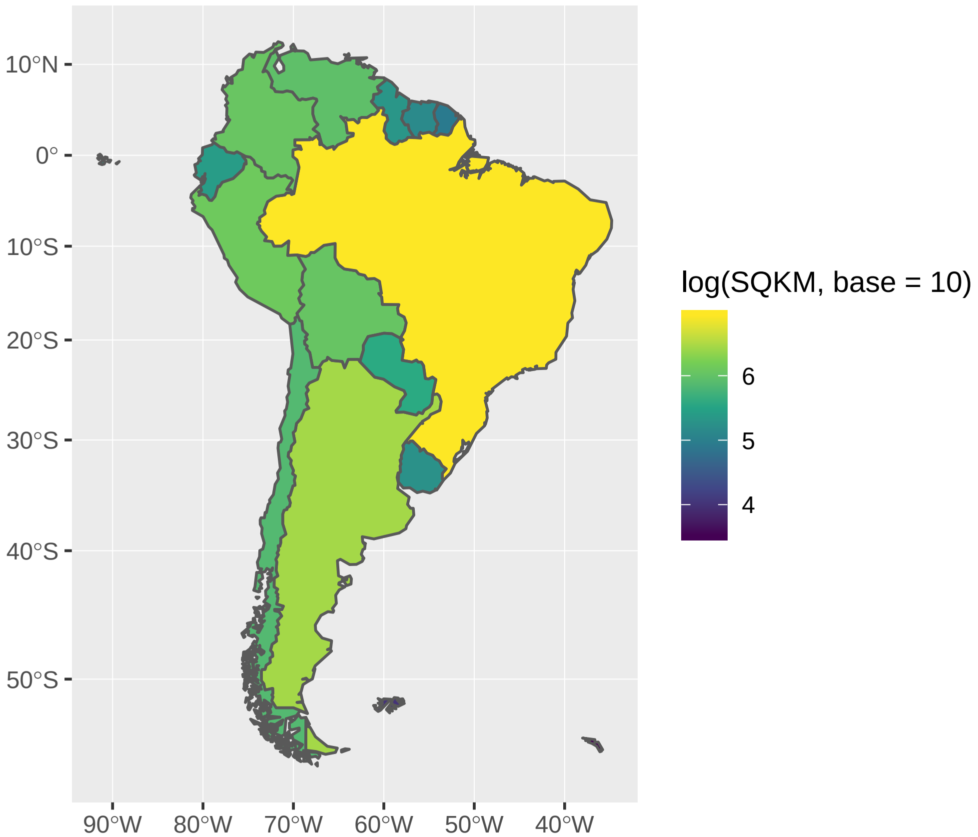

```
plot(SA["SQKM"]) # base R method - a little better, but not so easy to control
```




```
ggplot() + geom_sf(data=SA, aes(fill=SQKM))+
  scale_fill_viridis_c()
```



```
ggplot() + geom_sf(data=SA, aes(fill=log(SQKM, base = 10)))+
  scale_fill_viridis_c()
```




```
# SA %>% mutate(population=ifelse(POP2007>0, POP2007,1)) %>%
#   select(population) %>% pull(population)
#   ggplot()+
#   geom_sf(data=SA, aes(fill=population))+
#   scale_fill_viridis_c()



# SA %>%
#   ggplot(data=., aes())+geom_sf(fill="SQKM")+
#   geom_point(data=data.frame(lat=0,lon=-80),aes(lat,lon),col='red')
#
# ggplot(data=SA, aes())+geom_sf()+
#   geom_point(data=data.frame(lat=0,lon=-80),aes(lat,lon),col='red')
```

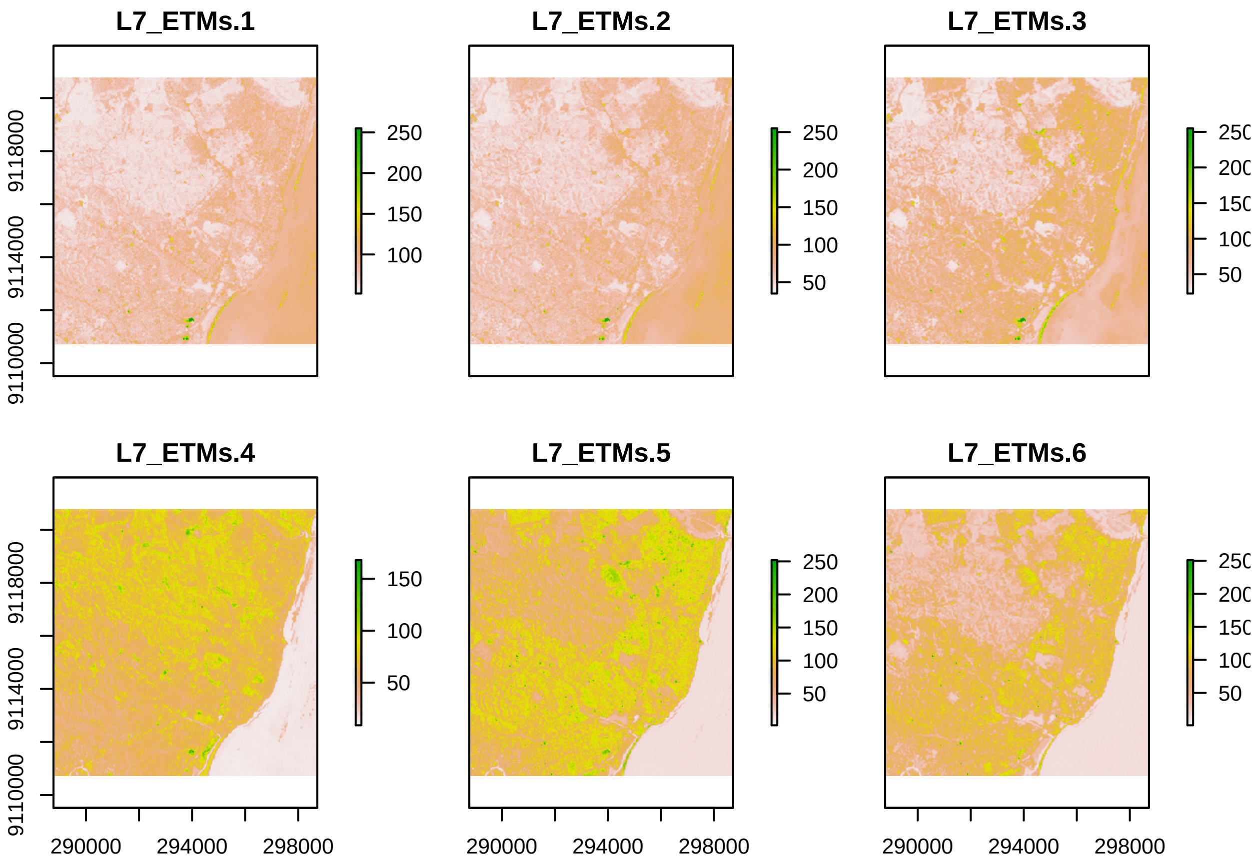
Plotting raster data with ggplot2

This works for visualizing single bands of smallish rasters (< 1000x1000)

```
library(tidyverse)
# Calculate NDVI from Landsat 7 -------------------------------------------
tif = system.file("tif/L7_ETMs.tif", package = "stars")
(r = raster::stack(tif))
```

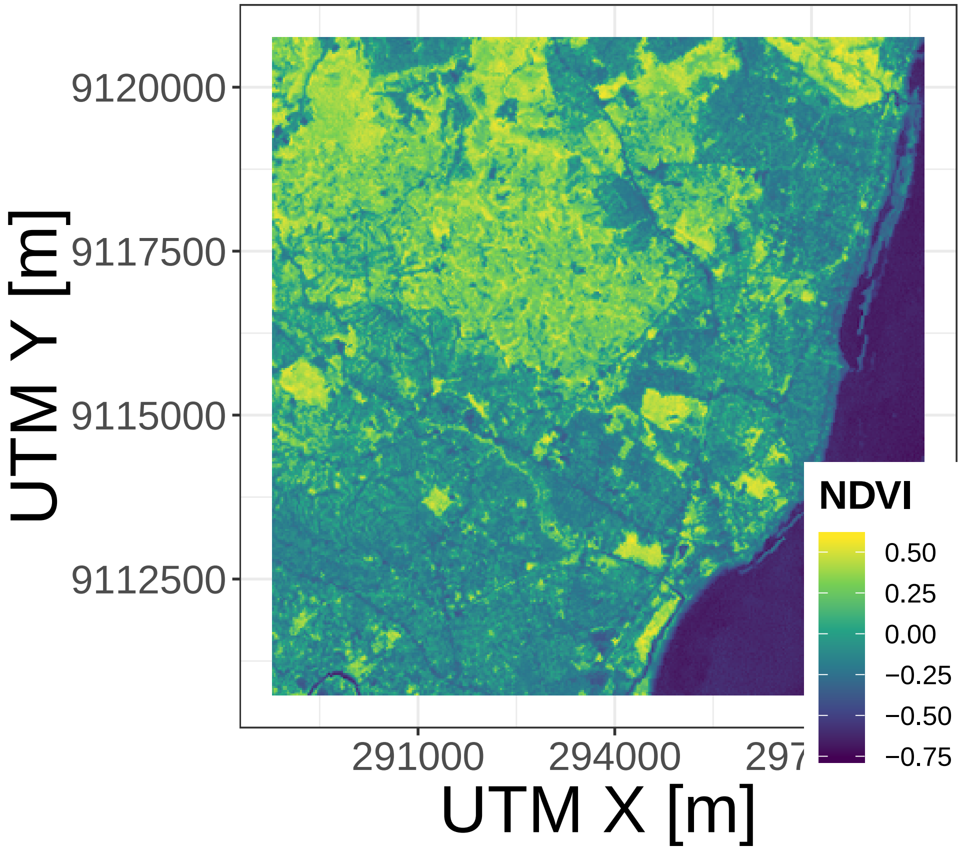
```
## class       : RasterStack
## dimensions  : 352, 349, 122848, 6  (nrow, ncol, ncell, nlayers)
## resolution  : 28.5, 28.5  (x, y)
## extent      : 288776.3, 298722.8, 9110729, 9120761  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=utm +zone=25 +south +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs
## names       : L7_ETMs.1, L7_ETMs.2, L7_ETMs.3, L7_ETMs.4, L7_ETMs.5, L7_ETMs.6
## min values  :         0,         0,         0,         0,         0,         0
## max values  :       255,       255,       255,       255,       255,       255
```

```
raster::plot(r) # Not ideal.
```




```
l7 <- raster::as.data.frame(r, xy=T) %>% as_tibble()
l7 <- l7 %>%
  mutate(ndvi=(L7_ETMs.4-L7_ETMs.3)/(L7_ETMs.4+L7_ETMs.3))
l7 %>%
  ggplot(data=., aes(x,y,fill=ndvi))+
  geom_raster()+
  coord_equal()+
  theme_bw()+
  scale_fill_viridis_c("NDVI")+
  labs(x="UTM X [m]",y="UTM Y [m]")+
  theme(legend.position = c(0.9,0.15),
        legend.title = element_text(size=15, face = 'bold'),
        legend.text = element_text(size=10),
        axis.title.x = element_text(size=25),
        axis.text.x = element_text(size=15),
        axis.title.y = element_text(size=25),
```

```
      axis.text.y = element_text(size=15))
```



A peak into stars!

stars is developing package for dealing with spatial raster and vector data

It's tidyverse compliant, and is/will be much better suited for processing large spatial data in R

#[https://www.r-spatial.org/r/2018/03/22/stars2.html]

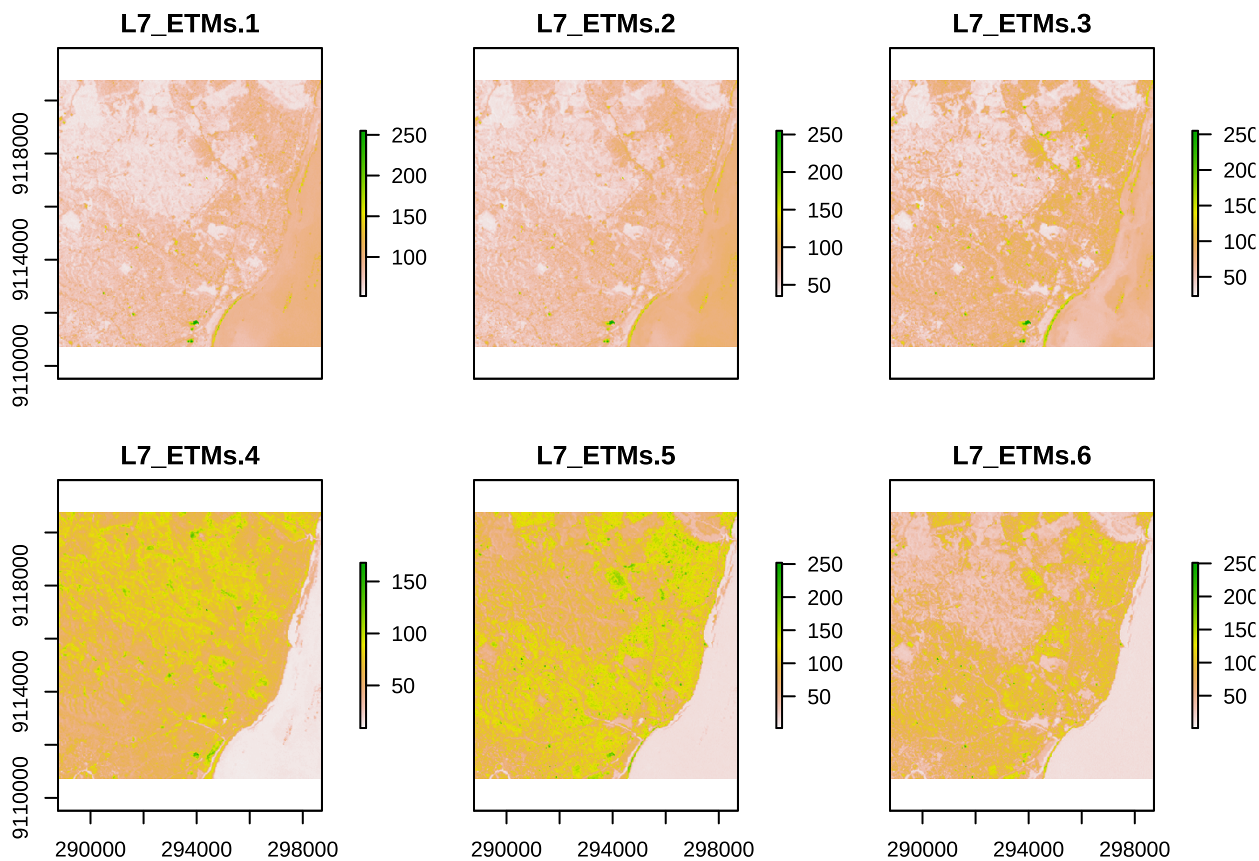
#! CAUTIONARY NOTE ! if you are processing Gbs worth of raster or other spatiotemporal data, consider doing it in Python

```
library(stars)
tif = system.file("tif/L7_ETMs.tif", package = "stars")
(r = raster::stack(tif))
```

```
## class       : RasterStack
## dimensions  : 352, 349, 122848, 6  (nrow, ncol, ncell, nlayers)
## resolution  : 28.5, 28.5  (x, y)
```

```
## extent      : 288776.3, 298722.8, 9110729, 9120761  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=utm +zone=25 +south +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs
## names       : L7_ETMs.1, L7_ETMs.2, L7_ETMs.3, L7_ETMs.4, L7_ETMs.5, L7_ETMs.6
## min values  :         0,         0,         0,         0,         0,         0
## max values  :       255,       255,       255,       255,       255,       255
```

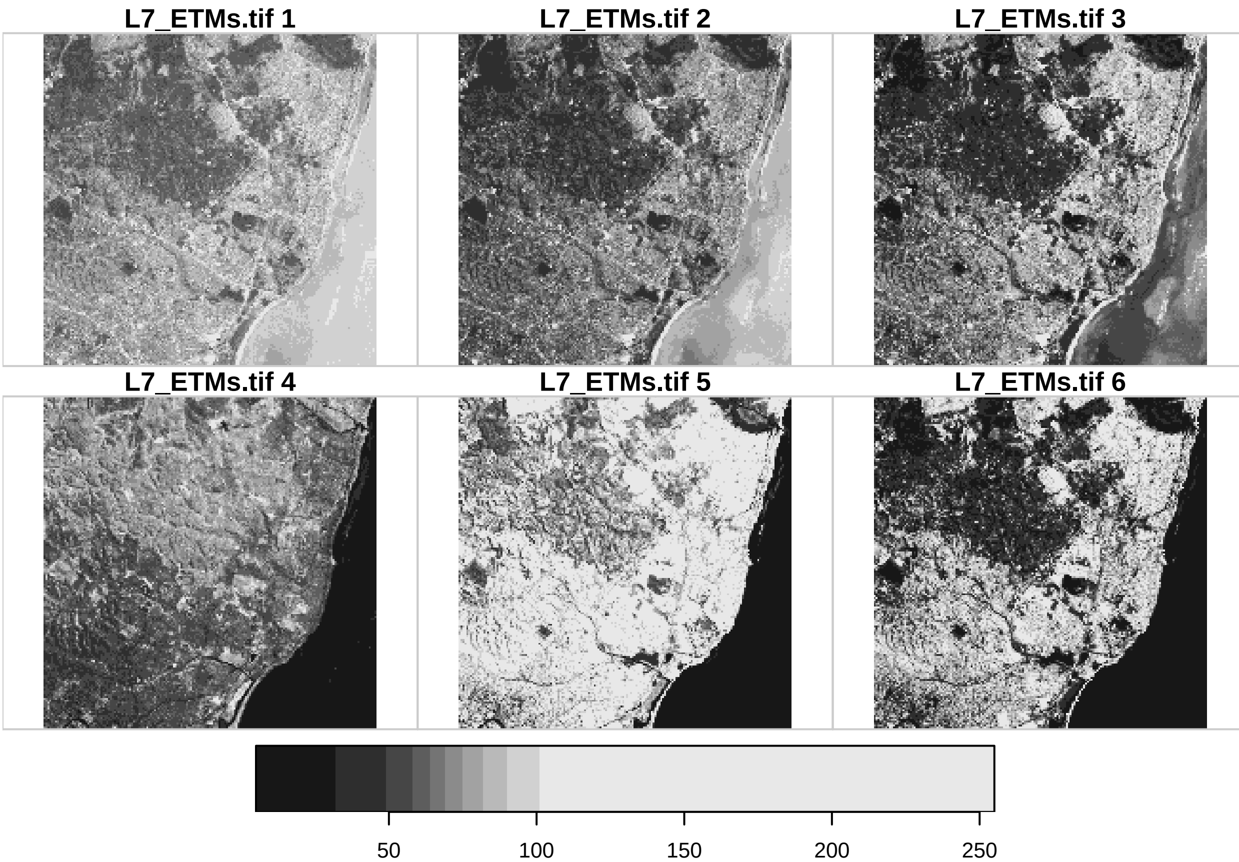
```r
raster::plot(r) # Not ideal.
```



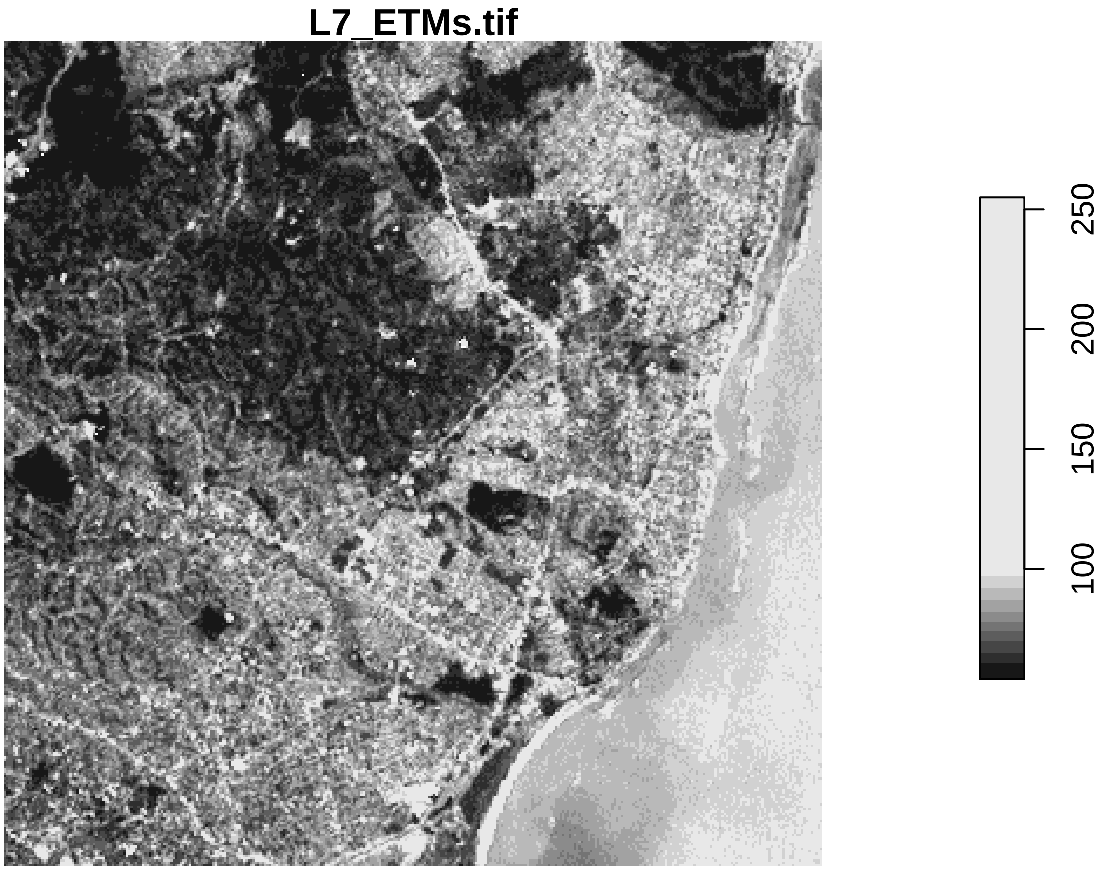

```r
(x = read_stars(tif))
```

```
## stars object with 3 dimensions and 1 attribute
## attribute(s):
##   L7_ETMs.tif
##  Min.   :  1.00
##  1st Qu.: 54.00
##  Median : 69.00
##  Mean   : 68.91
##  3rd Qu.: 86.00
##  Max.   :255.00
## dimension(s):
##      from  to  offset delta                       refsys point values
## x       1 349  288776  28.5 +proj=utm +zone=25 +south... FALSE   NULL
## y       1 352 9120761 -28.5 +proj=utm +zone=25 +south... FALSE   NULL
## band    1   6      NA    NA                           NA    NA   NULL
```

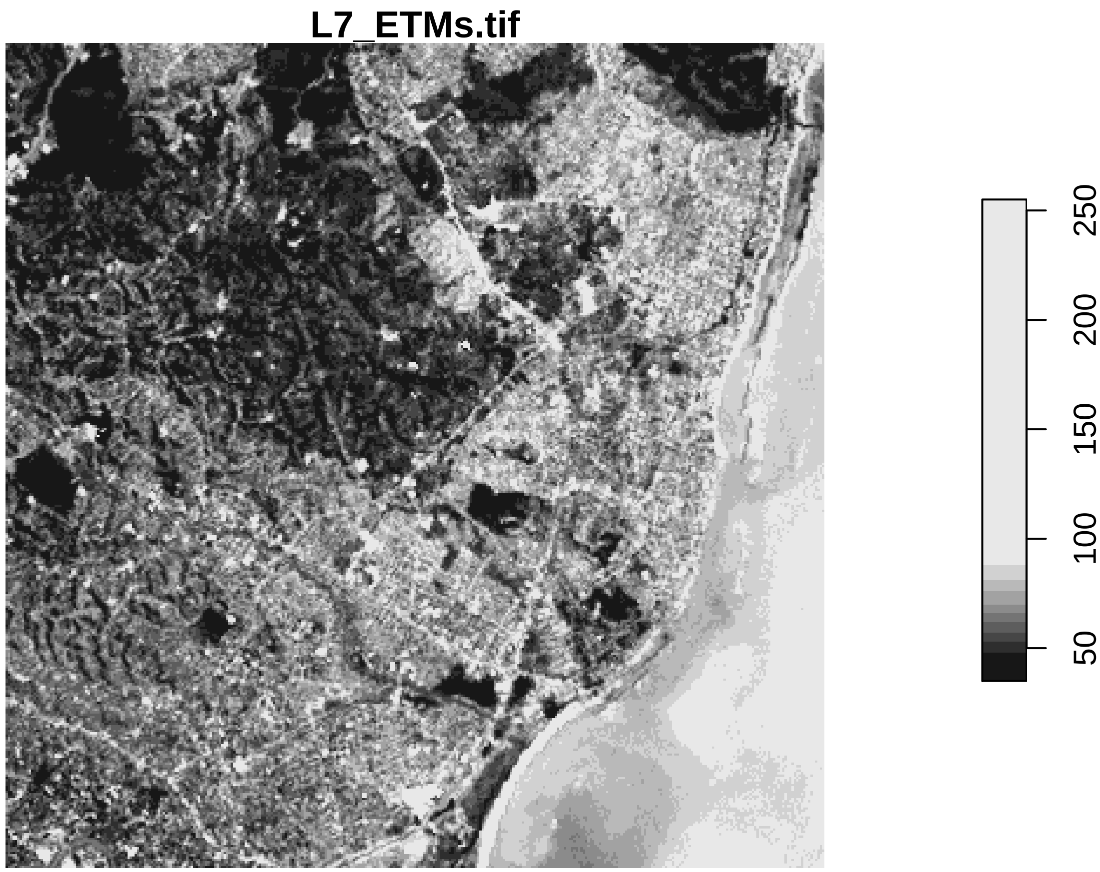
```r
plot(x) # much improved (see ?plot.stars)
```



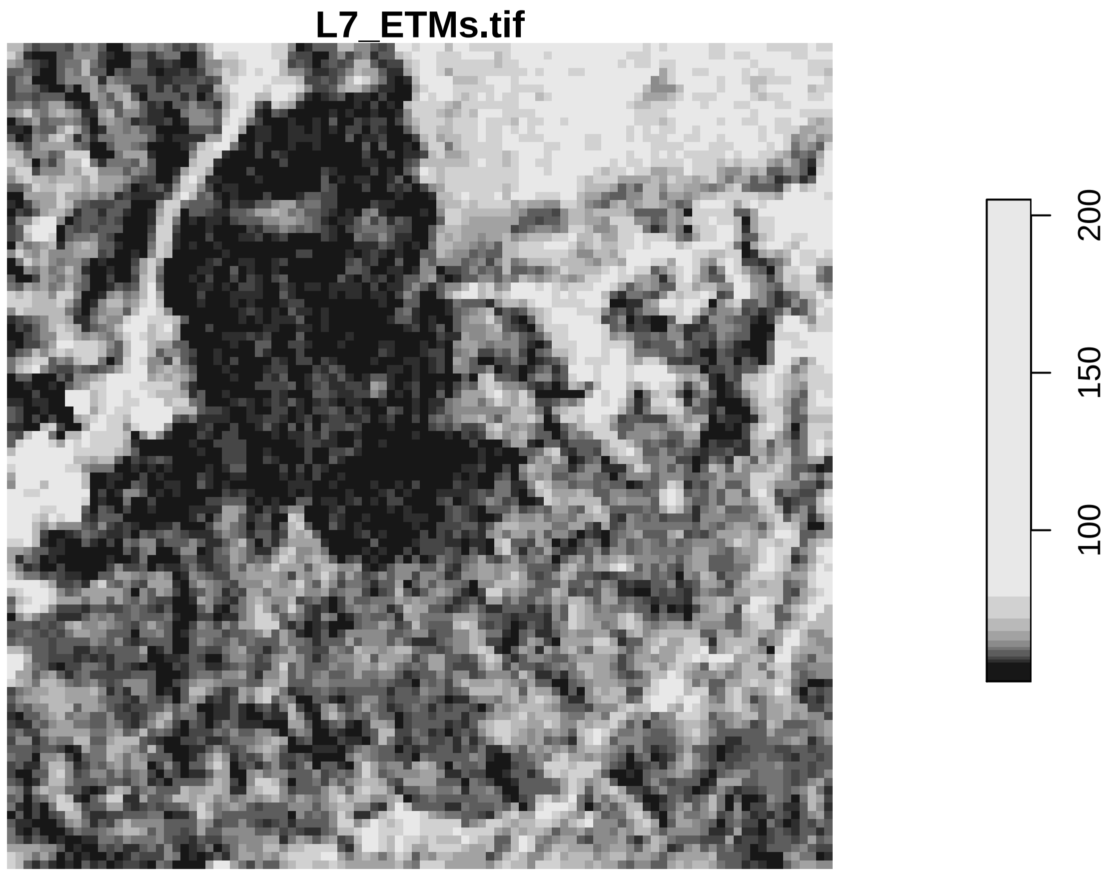

```
x[,,,1] %>% plot # plot band 1
```



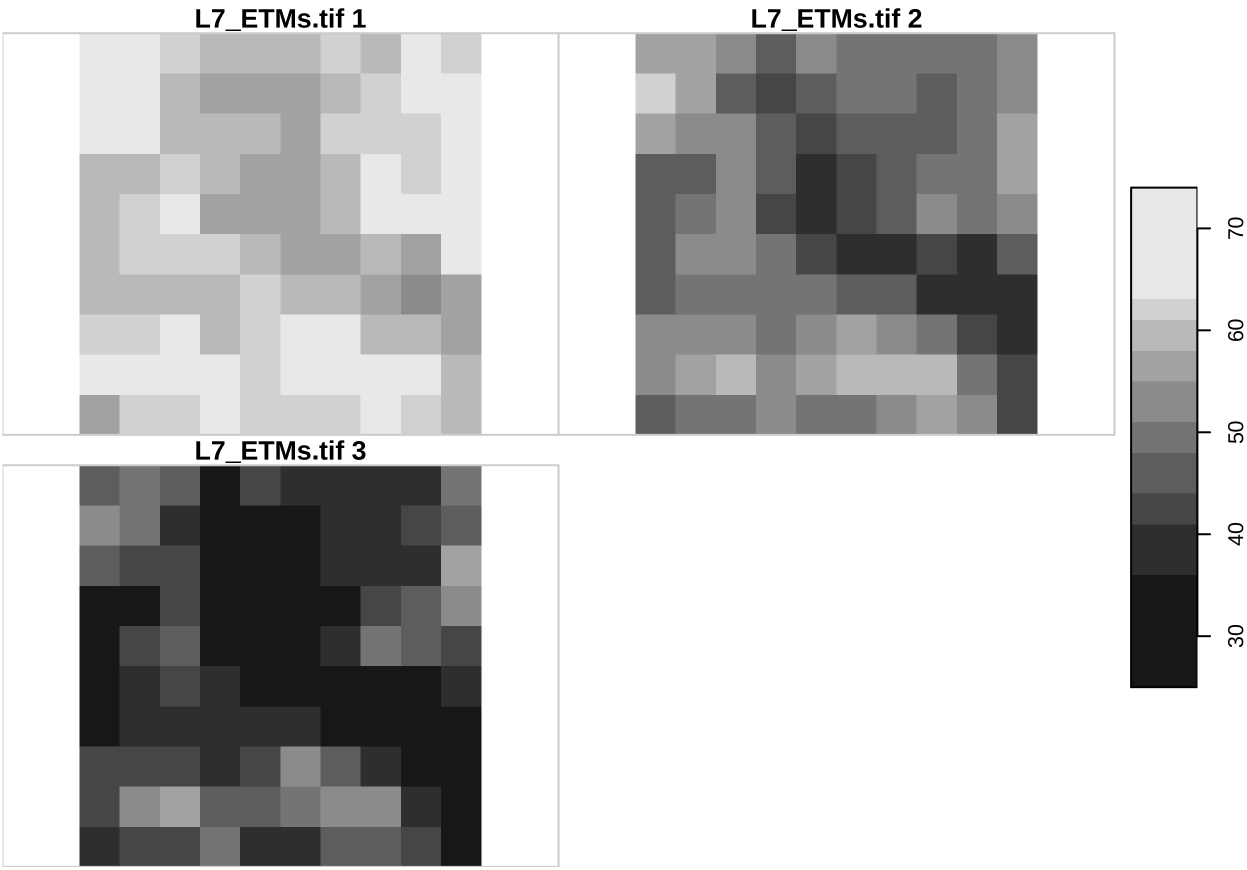

```
x[,,,2] %>% plot # plot band 2
```




```
x[,1:100,1:100,1] %>% plot # plot spatial subset
```




```
x[,1:10,1:10,c(1,2,3)] %>% plot
```

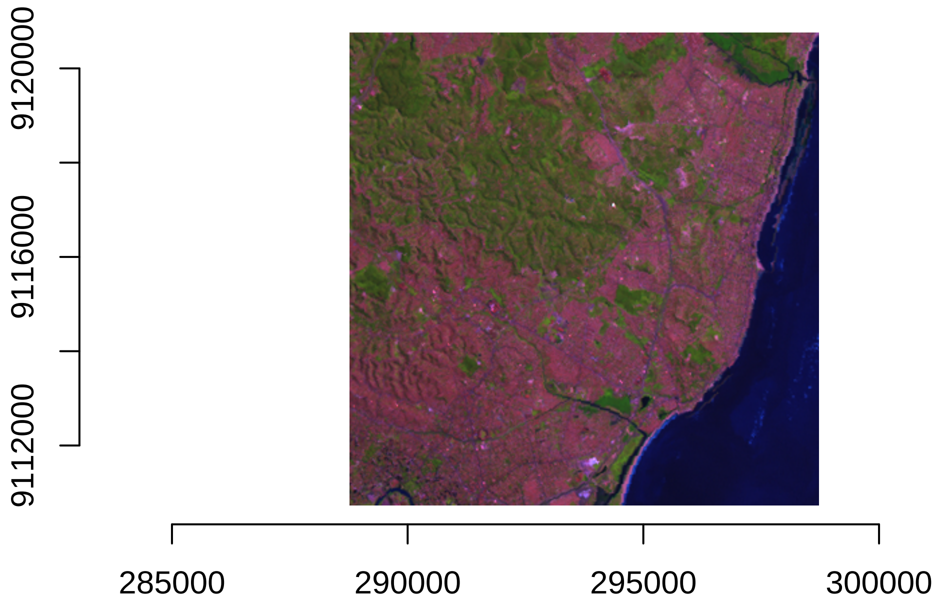


Plot an RGB with bands 5,4,3

```
library(stars)
tif = system.file("tif/L7_ETMs.tif", package = "stars")

(x = read_stars(tif))
```

```
## stars object with 3 dimensions and 1 attribute
## attribute(s):
##   L7_ETMs.tif
##  Min.   :  1.00
##  1st Qu.: 54.00
##  Median : 69.00
##  Mean   : 68.91
##  3rd Qu.: 86.00
##  Max.   :255.00
## dimension(s):
##      from  to  offset delta                       refsys point values
## x       1 349  288776  28.5 +proj=utm +zone=25 +south... FALSE   NULL
## y       1 352 9120761 -28.5 +proj=utm +zone=25 +south... FALSE   NULL
## band    1   6      NA    NA                           NA    NA   NULL
```

```
image(x, rgb=c(5,4,3), axes=T)
```


9120000
9116000
9112000
285000
290000
295000
300000