

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

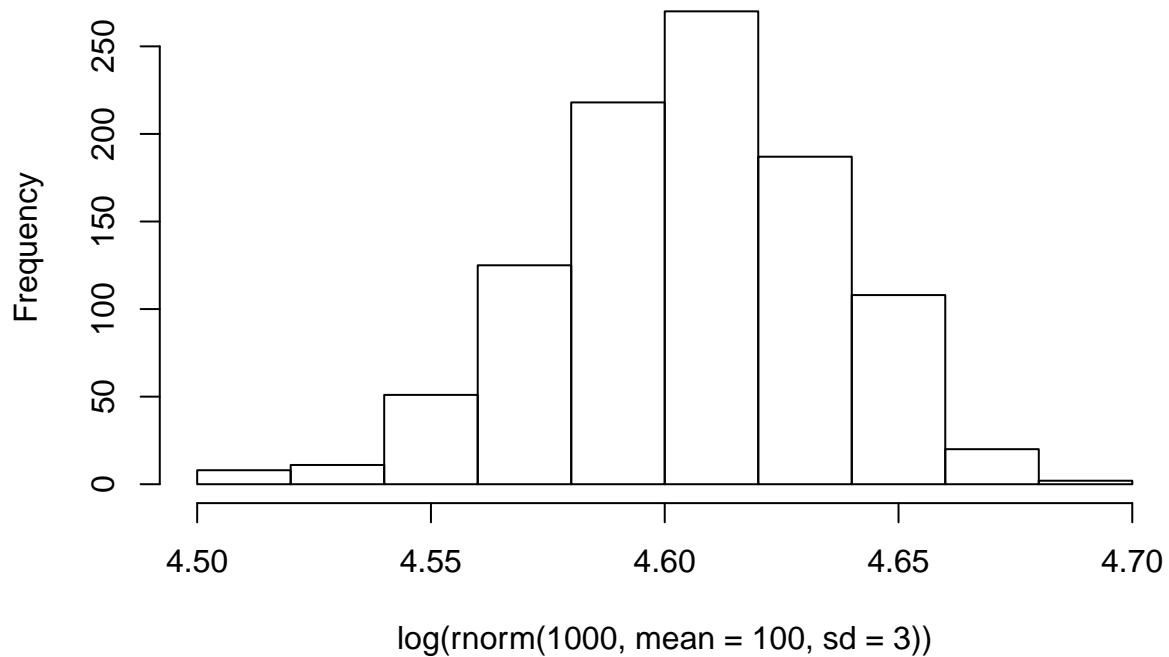
- This is a ‘pipe’ symbol: `%>%`
- The keyboard shortcut in Rstudio is “ctrl+shift+m”
- pipes allow chaining operations to an object

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)))
```

Histogram of $\log(rnorm(1000, \text{mean} = 100, \text{sd} = 3))$

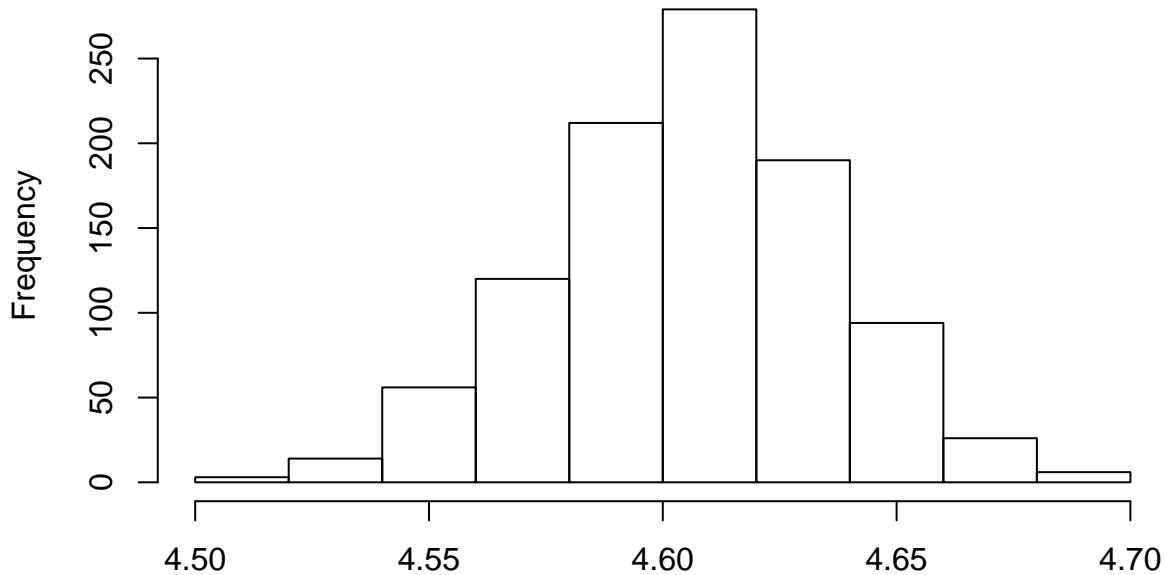


pipe way

- Here you can linearly read the chain of operations

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

Histogram of .



So pipes allow us to arbitrarily long things without nesting or creating copies 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(20, mean = 0, sd=10),
                      y=rnorm(20, mean=10, sd=1))
print(junk_df) # prints the whole df!
```

```
##          x         y
## 1 -13.420136 10.006752
## 2  -2.726245  9.953307
## 3   14.333315  9.032897
## 4  -2.000737 10.477194
## 5  -8.957603 11.082689
## 6   6.018413  9.218545
## 7  -3.494766  9.203391
## 8  -11.312544 10.131347
## 9  -1.039933  9.835577
```

```

## 10  2.376652 8.843389
## 11 11.514113 11.574055
## 12  9.023273 9.515908
## 13 16.740338 11.694258
## 14  4.041662 10.878924
## 15 16.546750 11.085885
## 16 -3.714092 9.399344
## 17 -2.282973 8.652711
## 18 -4.733201 11.116952
## 19 25.074555 9.484121
## 20 -10.879875 10.272817

**** NOW LET'S MAKE A BIG TIBBLE****
junk_tb <- tibble(x=rnorm(100, mean = 0, sd=10),
                  y=rnorm(100, mean=10, sd=1))
print(junk_tb) # just prints the top 10 rows

## # A tibble: 100 x 2
##       x     y
##   <dbl> <dbl>
## 1 -7.17  9.09
## 2  0.415  8.84
## 3  5.28  10.2
## 4  4.97  8.62
## 5 13.9   8.13
## 6  4.38  9.16
## 7 -16.2   9.13
## 8 -0.570  9.89
## 9 -4.69  10.8
## 10 4.78   9.70
## # ... with 90 more rows

```

Filtering with dplyr

Base R approach to filtering data:

```

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 -

```

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

print(tmp)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1      5.1      3.5      1.4      0.2  setosa
## 2      4.9      3.0      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      5.0      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	5.0	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
## 11	5.4	3.7	1.5	0.2	setosa
## 12	4.8	3.4	1.6	0.2	setosa
## 13	4.8	3.0	1.4	0.1	setosa
## 14	4.3	3.0	1.1	0.1	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
## 23	4.6	3.6	1.0	0.2	setosa
## 24	5.1	3.3	1.7	0.5	setosa
## 25	4.8	3.4	1.9	0.2	setosa
## 26	5.0	3.0	1.6	0.2	setosa
## 27	5.0	3.4	1.6	0.4	setosa
## 28	5.2	3.5	1.5	0.2	setosa
## 29	5.2	3.4	1.4	0.2	setosa
## 30	4.7	3.2	1.6	0.2	setosa
## 31	4.8	3.1	1.6	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
## 35	4.9	3.1	1.5	0.2	setosa
## 36	5.0	3.2	1.2	0.2	setosa
## 37	5.5	3.5	1.3	0.2	setosa
## 38	4.9	3.6	1.4	0.1	setosa
## 39	4.4	3.0	1.3	0.2	setosa
## 40	5.1	3.4	1.5	0.2	setosa
## 41	5.0	3.5	1.3	0.3	setosa
## 42	4.5	2.3	1.3	0.3	setosa
## 43	4.4	3.2	1.3	0.2	setosa
## 44	5.0	3.5	1.6	0.6	setosa
## 45	5.1	3.8	1.9	0.4	setosa
## 46	4.8	3.0	1.4	0.3	setosa
## 47	5.1	3.8	1.6	0.2	setosa
## 48	4.6	3.2	1.4	0.2	setosa
## 49	5.3	3.7	1.5	0.2	setosa
## 50	5.0	3.3	1.4	0.2	setosa
## 51	7.0	3.2	4.7	1.4	versicolor
## 52	6.4	3.2	4.5	1.5	versicolor
## 53	6.9	3.1	4.9	1.5	versicolor
## 54	5.5	2.3	4.0	1.3	versicolor
## 55	6.5	2.8	4.6	1.5	versicolor
## 56	5.7	2.8	4.5	1.3	versicolor
## 57	6.3	3.3	4.7	1.6	versicolor
## 58	4.9	2.4	3.3	1.0	versicolor
## 59	6.6	2.9	4.6	1.3	versicolor

## 60	5.2	2.7	3.9	1.4	versicolor
## 61	5.0	2.0	3.5	1.0	versicolor
## 62	5.9	3.0	4.2	1.5	versicolor
## 63	6.0	2.2	4.0	1.0	versicolor
## 64	6.1	2.9	4.7	1.4	versicolor
## 65	5.6	2.9	3.6	1.3	versicolor
## 66	6.7	3.1	4.4	1.4	versicolor
## 67	5.6	3.0	4.5	1.5	versicolor
## 68	5.8	2.7	4.1	1.0	versicolor
## 69	6.2	2.2	4.5	1.5	versicolor
## 70	5.6	2.5	3.9	1.1	versicolor
## 71	5.9	3.2	4.8	1.8	versicolor
## 72	6.1	2.8	4.0	1.3	versicolor
## 73	6.3	2.5	4.9	1.5	versicolor
## 74	6.1	2.8	4.7	1.2	versicolor
## 75	6.4	2.9	4.3	1.3	versicolor
## 76	6.6	3.0	4.4	1.4	versicolor
## 77	6.8	2.8	4.8	1.4	versicolor
## 78	6.7	3.0	5.0	1.7	versicolor
## 79	6.0	2.9	4.5	1.5	versicolor
## 80	5.7	2.6	3.5	1.0	versicolor
## 81	5.5	2.4	3.8	1.1	versicolor
## 82	5.5	2.4	3.7	1.0	versicolor
## 83	5.8	2.7	3.9	1.2	versicolor
## 84	6.0	2.7	5.1	1.6	versicolor
## 85	5.4	3.0	4.5	1.5	versicolor
## 86	6.0	3.4	4.5	1.6	versicolor
## 87	6.7	3.1	4.7	1.5	versicolor
## 88	6.3	2.3	4.4	1.3	versicolor
## 89	5.6	3.0	4.1	1.3	versicolor
## 90	5.5	2.5	4.0	1.3	versicolor
## 91	5.5	2.6	4.4	1.2	versicolor
## 92	6.1	3.0	4.6	1.4	versicolor
## 93	5.8	2.6	4.0	1.2	versicolor
## 94	5.0	2.3	3.3	1.0	versicolor
## 95	5.6	2.7	4.2	1.3	versicolor
## 96	5.7	3.0	4.2	1.2	versicolor
## 97	5.7	2.9	4.2	1.3	versicolor
## 98	6.2	2.9	4.3	1.3	versicolor
## 99	5.1	2.5	3.0	1.1	versicolor
## 100	5.7	2.8	4.1	1.3	versicolor
## 101	6.3	3.3	6.0	2.5	virginica
## 102	5.8	2.7	5.1	1.9	virginica
## 103	7.1	3.0	5.9	2.1	virginica
## 104	6.3	2.9	5.6	1.8	virginica
## 105	6.5	3.0	5.8	2.2	virginica
## 106	7.6	3.0	6.6	2.1	virginica
## 107	4.9	2.5	4.5	1.7	virginica
## 108	7.3	2.9	6.3	1.8	virginica
## 109	6.7	2.5	5.8	1.8	virginica
## 110	7.2	3.6	6.1	2.5	virginica
## 111	6.5	3.2	5.1	2.0	virginica
## 112	6.4	2.7	5.3	1.9	virginica
## 113	6.8	3.0	5.5	2.1	virginica

```

## 114      5.7      2.5      5.0      2.0  virginica
## 115      5.8      2.8      5.1      2.4  virginica
## 116      6.4      3.2      5.3      2.3  virginica
## 117      6.5      3.0      5.5      1.8  virginica
## 118      7.7      3.8      6.7      2.2  virginica
## 119      7.7      2.6      6.9      2.3  virginica
## 120      6.0      2.2      5.0      1.5  virginica
## 121      6.9      3.2      5.7      2.3  virginica
## 122      5.6      2.8      4.9      2.0  virginica
## 123      7.7      2.8      6.7      2.0  virginica
## 124      6.3      2.7      4.9      1.8  virginica
## 125      6.7      3.3      5.7      2.1  virginica
## 126      7.2      3.2      6.0      1.8  virginica
## 127      6.2      2.8      4.8      1.8  virginica
## 128      6.1      3.0      4.9      1.8  virginica
## 129      6.4      2.8      5.6      2.1  virginica
## 130      7.2      3.0      5.8      1.6  virginica
## 131      7.4      2.8      6.1      1.9  virginica
## 132      7.9      3.8      6.4      2.0  virginica
## 133      6.4      2.8      5.6      2.2  virginica
## 134      6.3      2.8      5.1      1.5  virginica
## 135      6.1      2.6      5.6      1.4  virginica
## 136      7.7      3.0      6.1      2.3  virginica
## 137      6.3      3.4      5.6      2.4  virginica
## 138      6.4      3.1      5.5      1.8  virginica
## 139      6.0      3.0      4.8      1.8  virginica
## 140      6.9      3.1      5.4      2.1  virginica
## 141      6.7      3.1      5.6      2.4  virginica
## 142      6.9      3.1      5.1      2.3  virginica
## 143      5.8      2.7      5.1      1.9  virginica
## 144      6.8      3.2      5.9      2.3  virginica
## 145      6.7      3.3      5.7      2.5  virginica
## 146      6.7      3.0      5.2      2.3  virginica
## 147      6.3      2.5      5.0      1.9  virginica
## 148      6.5      3.0      5.2      2.0  virginica
## 149      6.2      3.4      5.4      2.3  virginica
## 150      5.9      3.0      5.1      1.8  virginica
##          p_wl_ratio narrow
## 1  0.14285714  TRUE
## 2  0.14285714  TRUE
## 3  0.15384615  TRUE
## 4  0.13333333  TRUE
## 5  0.14285714  TRUE
## 6  0.23529412  TRUE
## 7  0.21428571  TRUE
## 8  0.13333333  TRUE
## 9  0.14285714  TRUE
## 10 0.06666667  TRUE
## 11 0.13333333  TRUE
## 12 0.12500000  TRUE
## 13 0.07142857  TRUE
## 14 0.09090909  TRUE
## 15 0.16666667  TRUE
## 16 0.26666667  FALSE

```

```
## 17 0.30769231 FALSE
## 18 0.21428571 TRUE
## 19 0.17647059 TRUE
## 20 0.20000000 TRUE
## 21 0.11764706 TRUE
## 22 0.26666667 FALSE
## 23 0.20000000 TRUE
## 24 0.29411765 FALSE
## 25 0.10526316 TRUE
## 26 0.12500000 TRUE
## 27 0.25000000 FALSE
## 28 0.13333333 TRUE
## 29 0.14285714 TRUE
## 30 0.12500000 TRUE
## 31 0.12500000 TRUE
## 32 0.26666667 FALSE
## 33 0.06666667 TRUE
## 34 0.14285714 TRUE
## 35 0.13333333 TRUE
## 36 0.16666667 TRUE
## 37 0.15384615 TRUE
## 38 0.07142857 TRUE
## 39 0.15384615 TRUE
## 40 0.13333333 TRUE
## 41 0.23076923 TRUE
## 42 0.23076923 TRUE
## 43 0.15384615 TRUE
## 44 0.37500000 FALSE
## 45 0.21052632 TRUE
## 46 0.21428571 TRUE
## 47 0.12500000 TRUE
## 48 0.14285714 TRUE
## 49 0.13333333 TRUE
## 50 0.14285714 TRUE
## 51 0.29787234 FALSE
## 52 0.33333333 FALSE
## 53 0.30612245 FALSE
## 54 0.32500000 FALSE
## 55 0.32608696 FALSE
## 56 0.28888889 FALSE
## 57 0.34042553 FALSE
## 58 0.30303030 FALSE
## 59 0.28260870 FALSE
## 60 0.35897436 FALSE
## 61 0.28571429 FALSE
## 62 0.35714286 FALSE
## 63 0.25000000 FALSE
## 64 0.29787234 FALSE
## 65 0.36111111 FALSE
## 66 0.31818182 FALSE
## 67 0.33333333 FALSE
## 68 0.24390244 TRUE
## 69 0.33333333 FALSE
## 70 0.28205128 FALSE
```

```
## 71 0.37500000 FALSE
## 72 0.32500000 FALSE
## 73 0.30612245 FALSE
## 74 0.25531915 FALSE
## 75 0.30232558 FALSE
## 76 0.31818182 FALSE
## 77 0.29166667 FALSE
## 78 0.34000000 FALSE
## 79 0.33333333 FALSE
## 80 0.28571429 FALSE
## 81 0.28947368 FALSE
## 82 0.27027027 FALSE
## 83 0.30769231 FALSE
## 84 0.31372549 FALSE
## 85 0.33333333 FALSE
## 86 0.35555556 FALSE
## 87 0.31914894 FALSE
## 88 0.29545455 FALSE
## 89 0.31707317 FALSE
## 90 0.32500000 FALSE
## 91 0.27272727 FALSE
## 92 0.30434783 FALSE
## 93 0.30000000 FALSE
## 94 0.30303030 FALSE
## 95 0.30952381 FALSE
## 96 0.28571429 FALSE
## 97 0.30952381 FALSE
## 98 0.30232558 FALSE
## 99 0.36666667 FALSE
## 100 0.31707317 FALSE
## 101 0.41666667 FALSE
## 102 0.37254902 FALSE
## 103 0.35593220 FALSE
## 104 0.32142857 FALSE
## 105 0.37931034 FALSE
## 106 0.31818182 FALSE
## 107 0.37777778 FALSE
## 108 0.28571429 FALSE
## 109 0.31034483 FALSE
## 110 0.40983607 FALSE
## 111 0.39215686 FALSE
## 112 0.35849057 FALSE
## 113 0.38181818 FALSE
## 114 0.40000000 FALSE
## 115 0.47058824 FALSE
## 116 0.43396226 FALSE
## 117 0.32727273 FALSE
## 118 0.32835821 FALSE
## 119 0.33333333 FALSE
## 120 0.30000000 FALSE
## 121 0.40350877 FALSE
## 122 0.40816327 FALSE
## 123 0.29850746 FALSE
## 124 0.36734694 FALSE
```

```

## 125 0.36842105 FALSE
## 126 0.30000000 FALSE
## 127 0.37500000 FALSE
## 128 0.36734694 FALSE
## 129 0.37500000 FALSE
## 130 0.27586207 FALSE
## 131 0.31147541 FALSE
## 132 0.31250000 FALSE
## 133 0.39285714 FALSE
## 134 0.29411765 FALSE
## 135 0.25000000 FALSE
## 136 0.37704918 FALSE
## 137 0.42857143 FALSE
## 138 0.32727273 FALSE
## 139 0.37500000 FALSE
## 140 0.38888889 FALSE
## 141 0.42857143 FALSE
## 142 0.45098039 FALSE
## 143 0.37254902 FALSE
## 144 0.38983051 FALSE
## 145 0.43859649 FALSE
## 146 0.44230769 FALSE
## 147 0.38000000 FALSE
## 148 0.38461538 FALSE
## 149 0.42592593 FALSE
## 150 0.35294118 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
```

Summarize many variables for a corresponding group

```
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 & tidyverse for getting data into a workable shape —

```

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

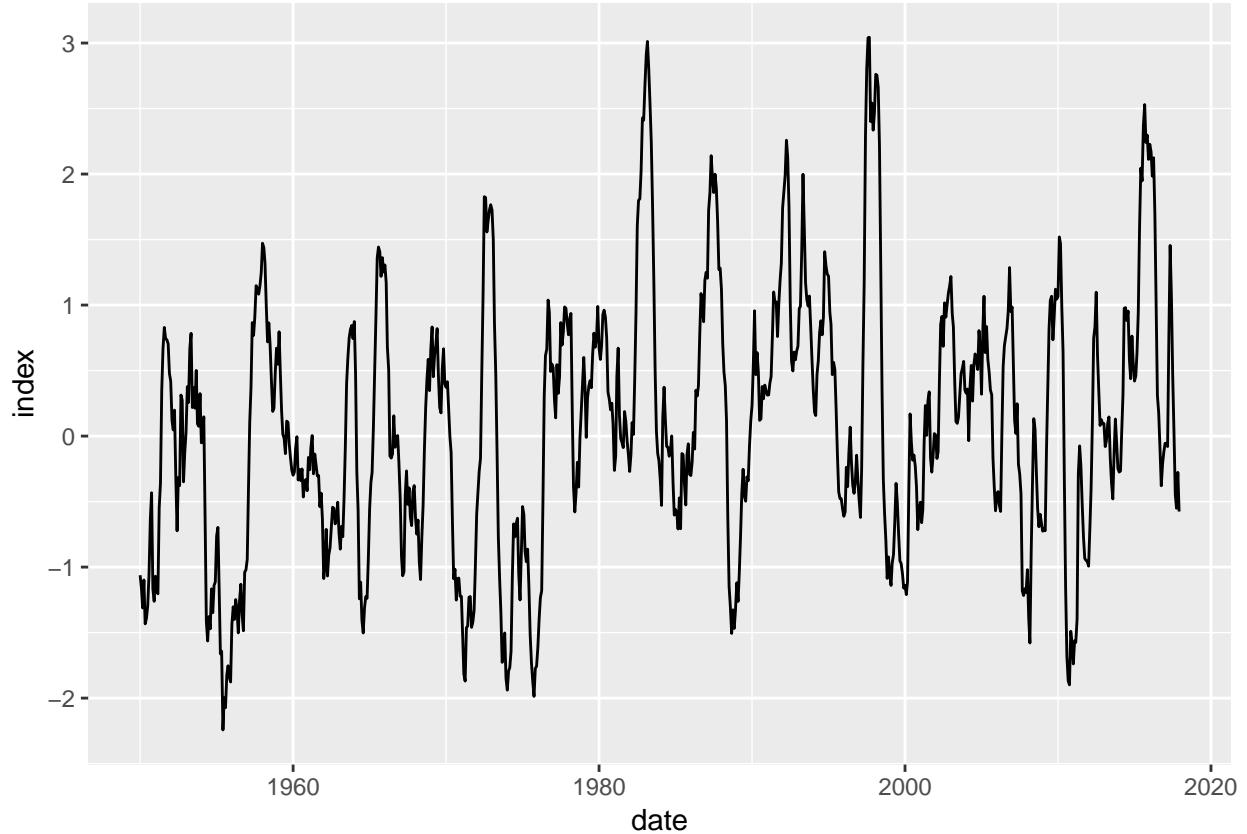
mei_wide <- readr::read_table("data/mei_1950_2018.data", skip = 1, col_names = F, n_max = 68) # slightly off

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 %>%
  ggplot(data=., aes(date, index))+geom_line()

```



Import and summarize data from the EE format

Zonal statistics from Earth Engine are typically exported from Earth Engine with a system:index column. This can be used to extract the date. Here we will import a time series from four Colombian cities from the “Climate Hazards Infrared with Station data” product. The observations are recorded on the pentad

(5-days). We will aggregate to the month and plot the time series.

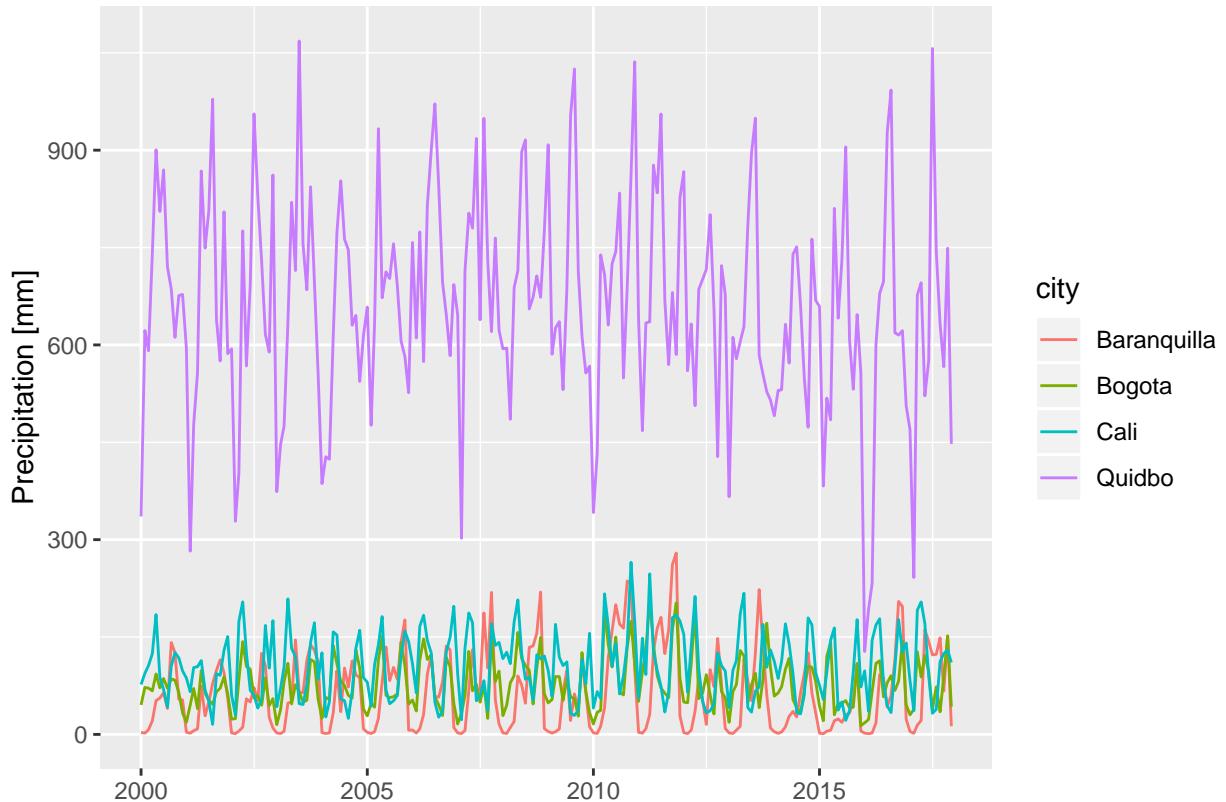
```
library(tidyverse);
library(lubridate);

# rainfall data exported from Google Earth Engine
tmp <- read_csv("data/CHIRPS_ColombiaCiudades_20000101_20180101.csv")

tmp %>% glimpse()

## Observations: 5,184
## Variables: 4
## $ `system:index` <chr> "20000101_0", "20000101_1", "20000101_2", "2000...
## $ mean              <dbl> 0.7941765, 5.7620168, 10.1810789, 35.3571243, 0...
## $ name               <chr> "Baranquilla", "Bogota", "Cali", "Quidbo", "Bar...
## $ .geo                <chr> NA, ...

tmp %>%
  rename(precip=mean, city=name) %>% # rename column
  mutate(date=parse_date_time(substr(`system:index`,1,8), 'ymd')) %>%
  mutate(year=year(date),      # lubridate functions to extract year and month from POSIXct date object
        month=month(date)) %>%
  group_by(year,month,city) %>% # what groups will we use to summarize the data?
  summarize(precip_tot = sum(precip, na.rm=T)) %>%
  ungroup() %>%
  mutate(date = parse_date_time(paste(year,month,1), 'ymd')) %>% # reassemble date
  ggplot(data=., aes(date, precip_tot, color=city))+geom_line()+
  labs(x="", y="Precipitation [mm]")
```

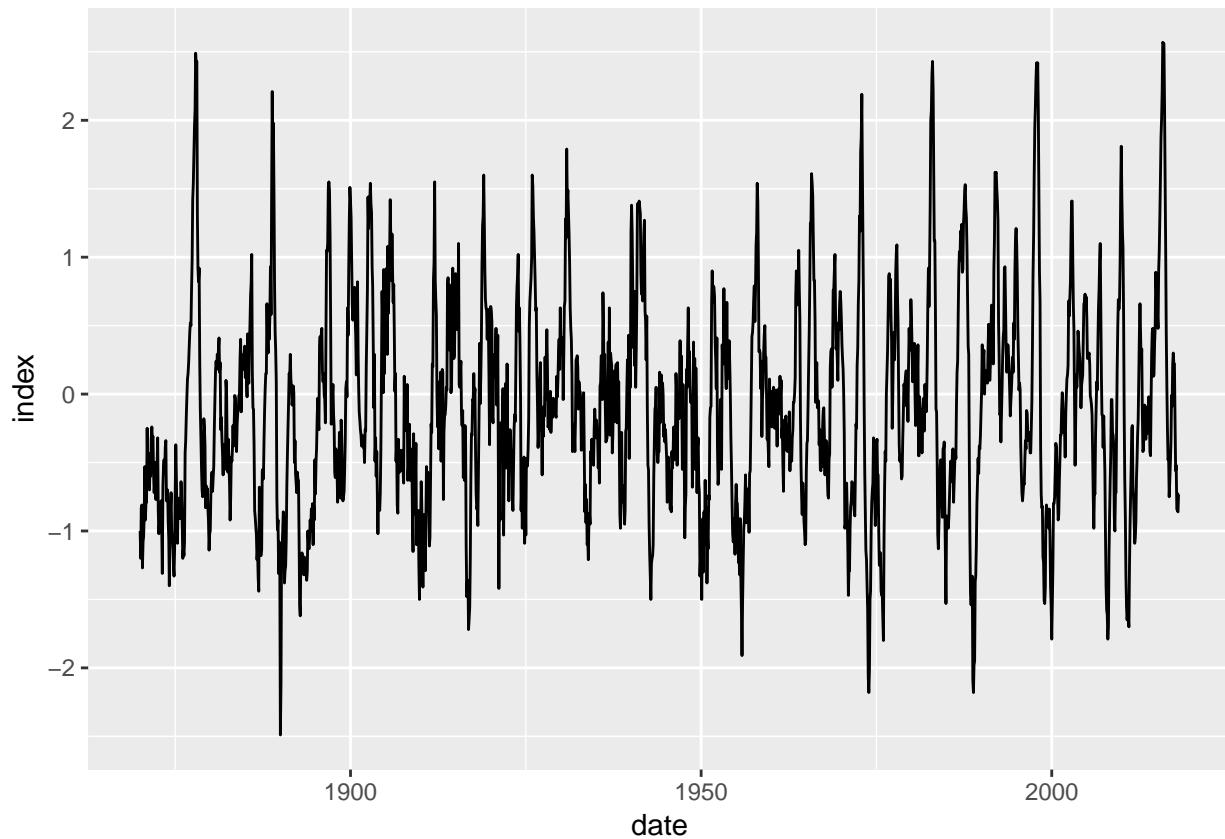


(5) ggplot2 for data visualization

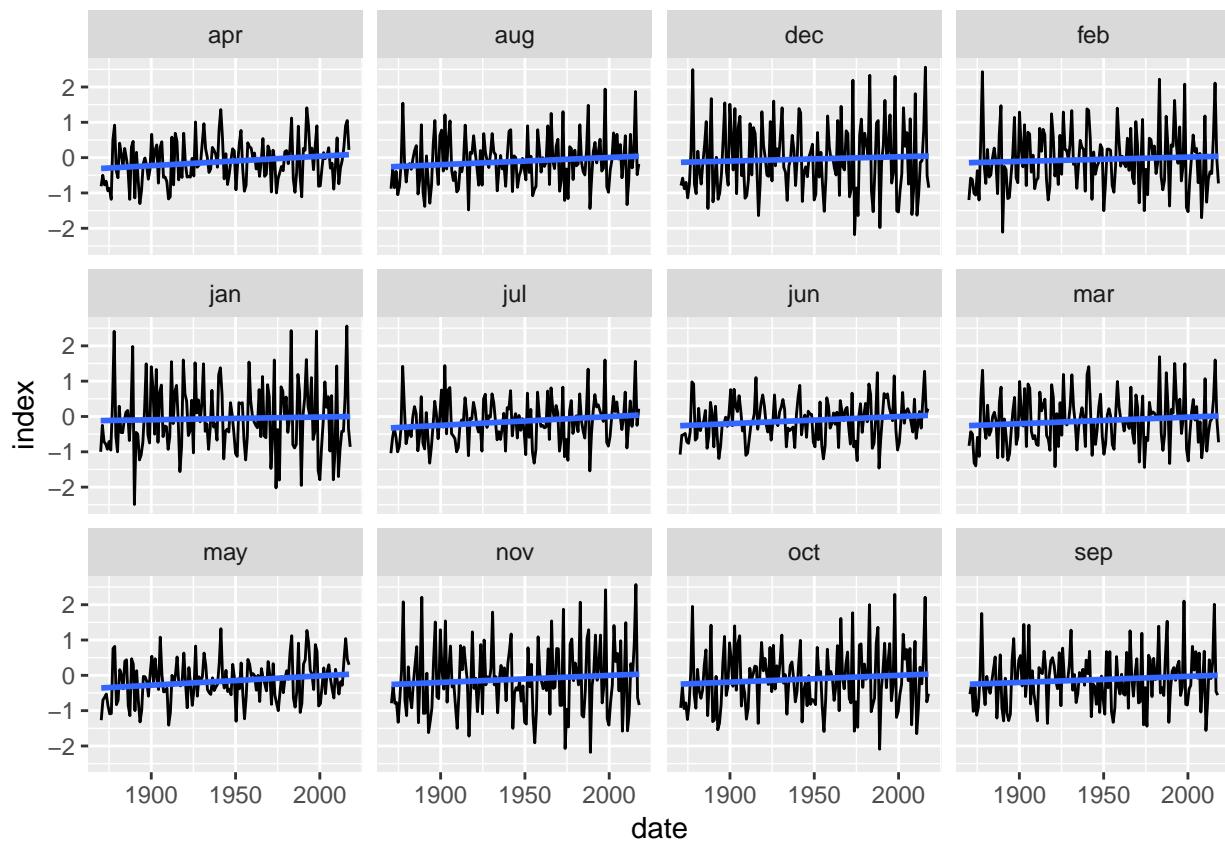
ggPlotting El Niño

“always plot your data”

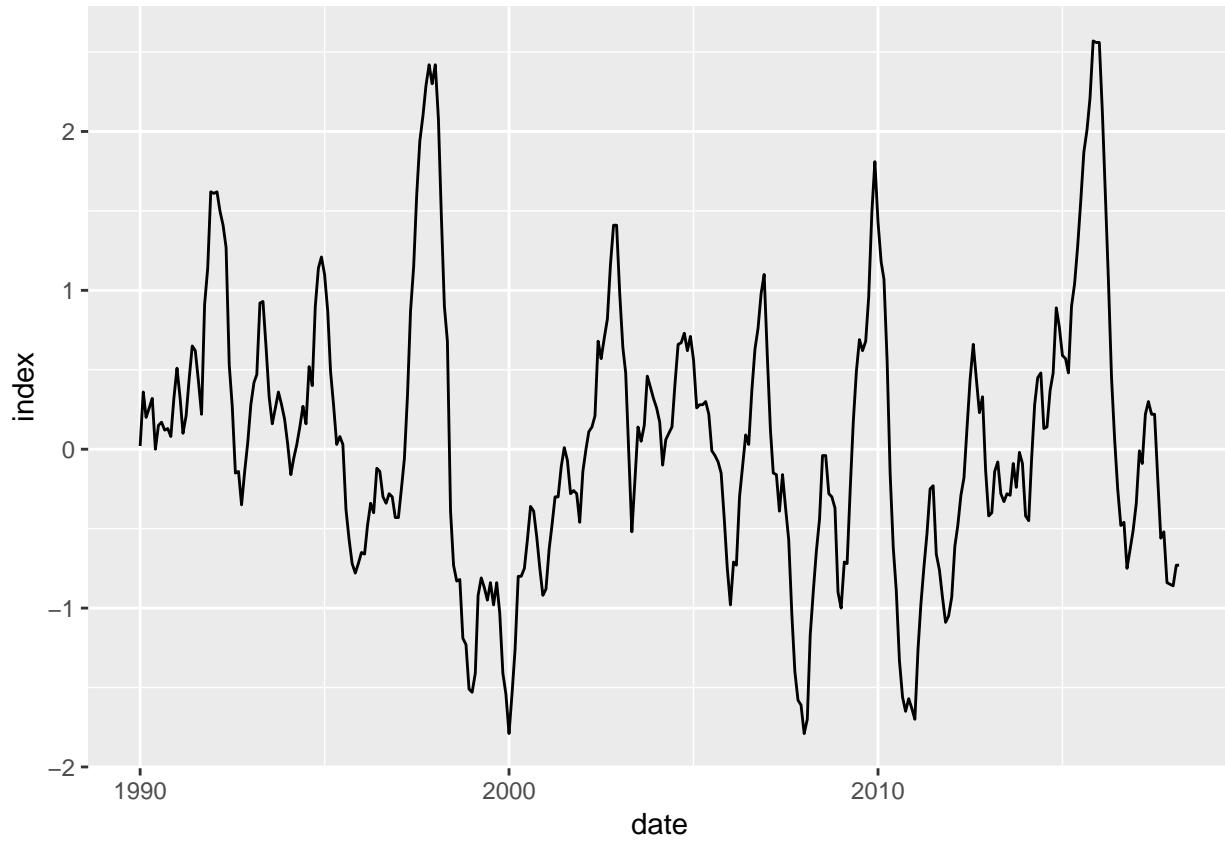
```
library(tidyverse);  
  
tmp <- read_csv("data/nino34_1870_2017.csv")  
  
# plot the whole record  
tmp %>%  
  #-----x-----y-----thing to add to plot  
  ggplot(data=., aes(date, index))+geom_line()
```



```
#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
```

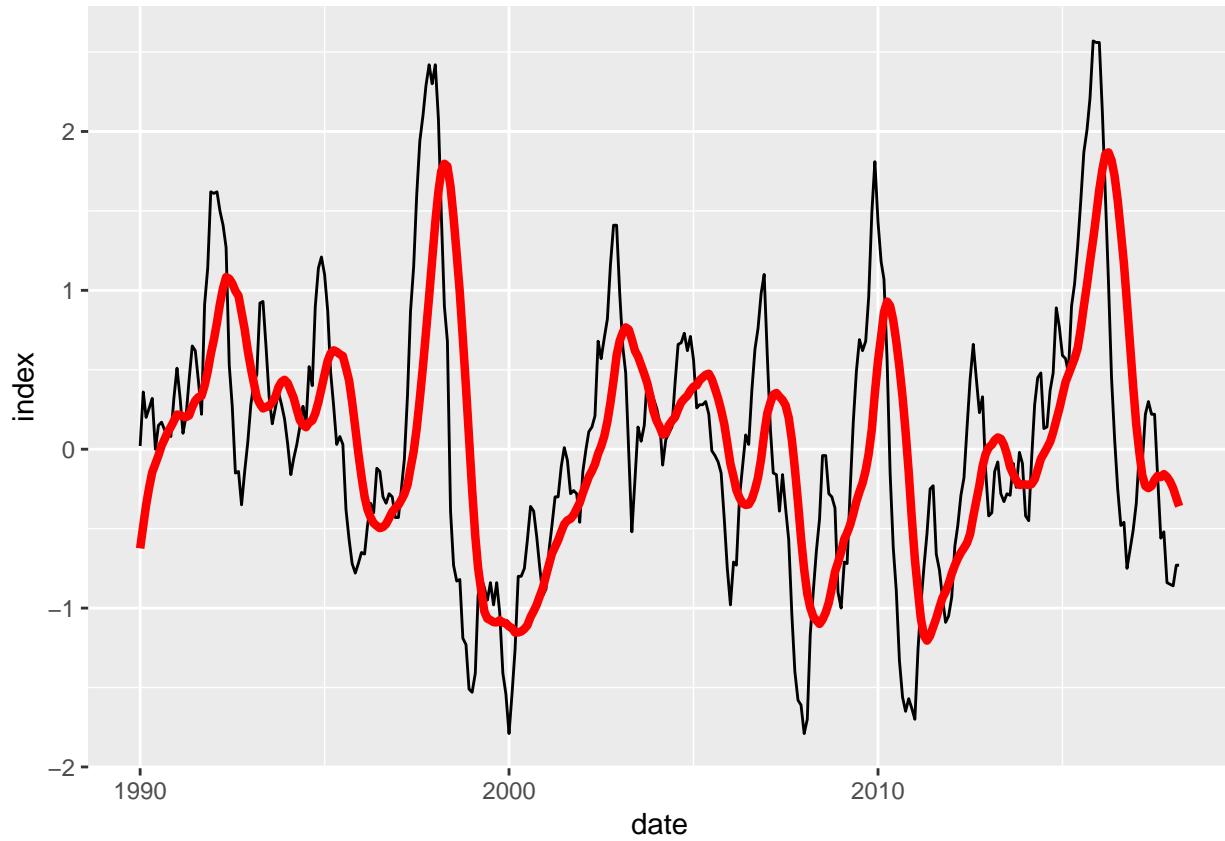


```
# plot recent ENSO record
tmp %>%
  filter(year>=1990) %>%          # filter for years >= 1990
  ggplot(data=., aes(date, index))+
  geom_line()
```



LETS ‘smooth’ the record with a moving average

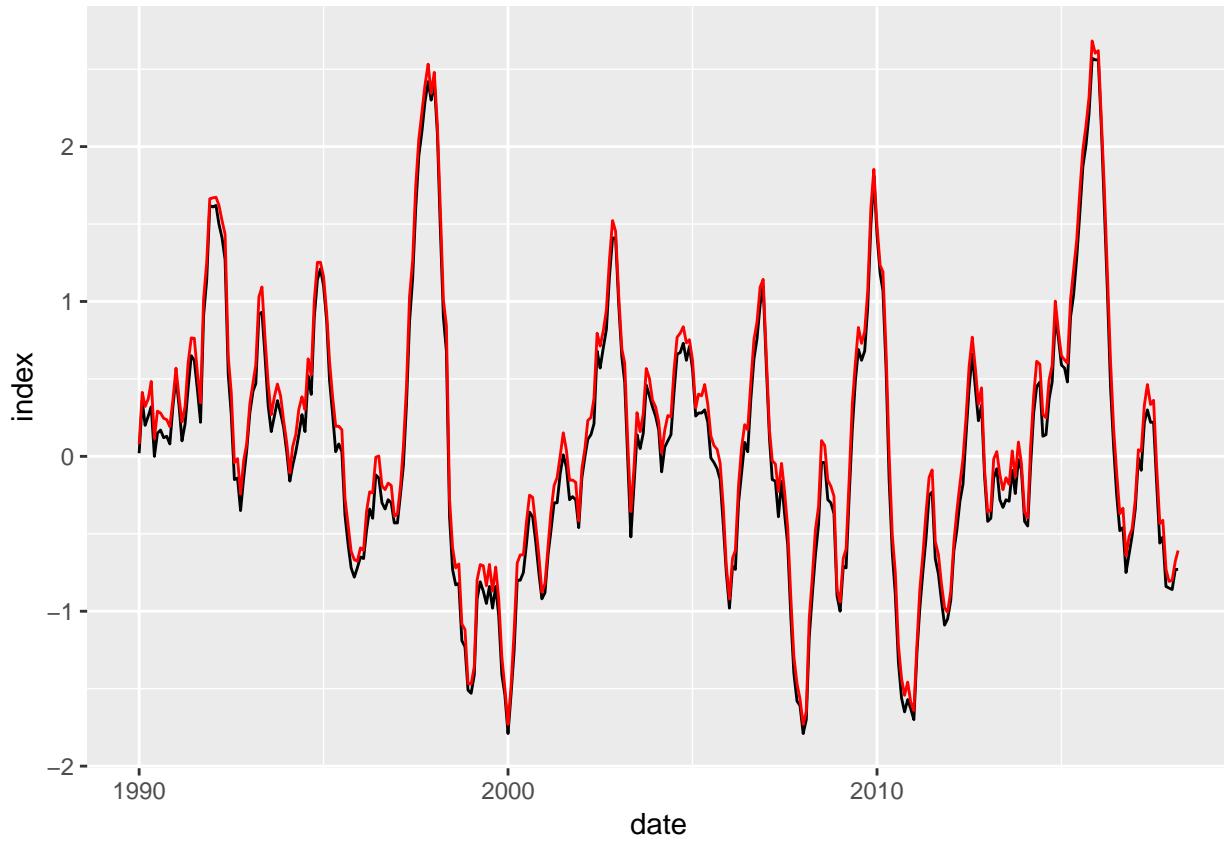
```
library(RcppRoll)
tmp %>%
  arrange(date) %>% # sort by the date
  mutate(index_12mo = roll_meanr(index, n=12, fill=NA)) %>% # running 12 month mean
  filter(year>=1990) %>%
  ggplot(data=., aes(date, index))+
  geom_line()+
  geom_line(aes(date, index_12mo), col='red', lwd=1.5)
```



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

```

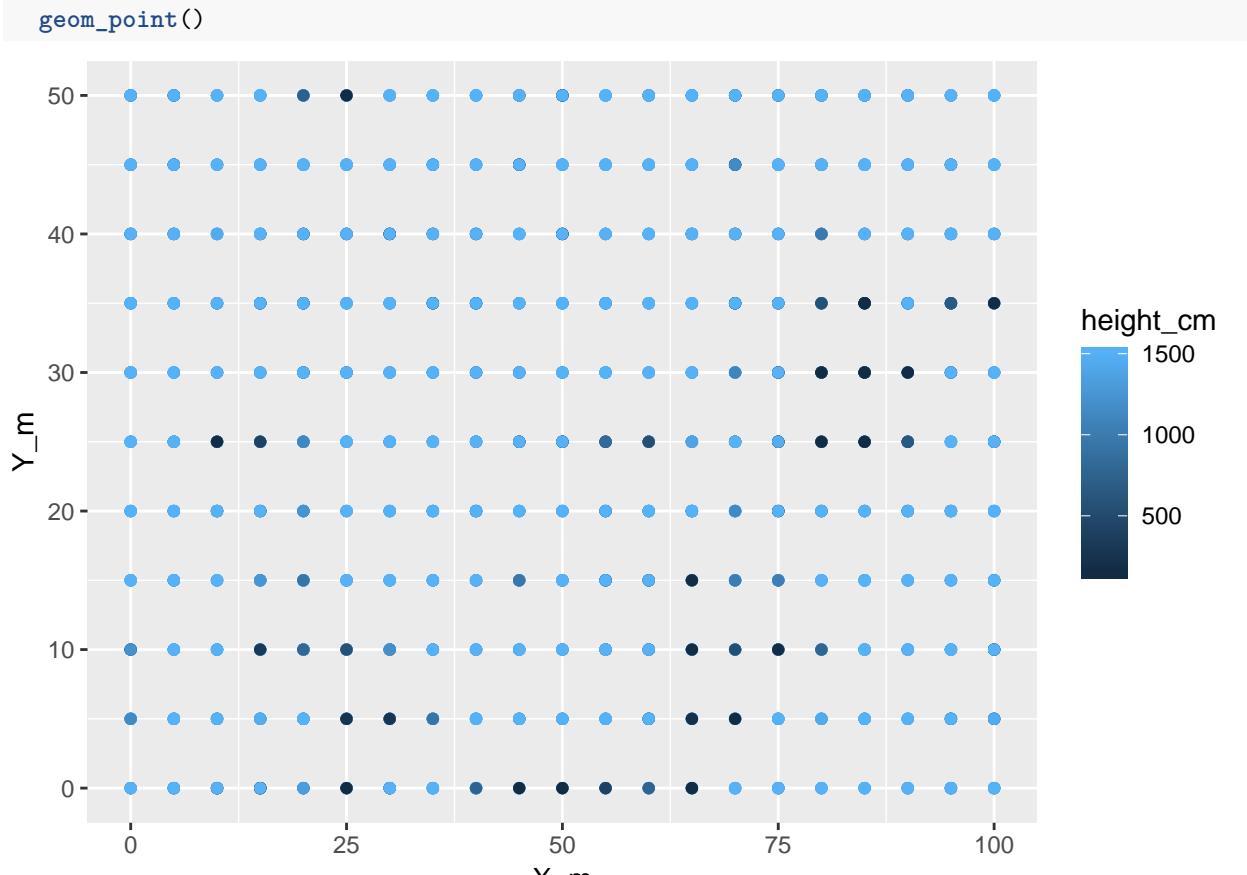
library(tidyverse); library(lubridate)
carb <- read_csv("data/claros1999_2012fulldataset.csv", skip = 5)
carb %>% glimpse()

## # 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"...
## $ 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...

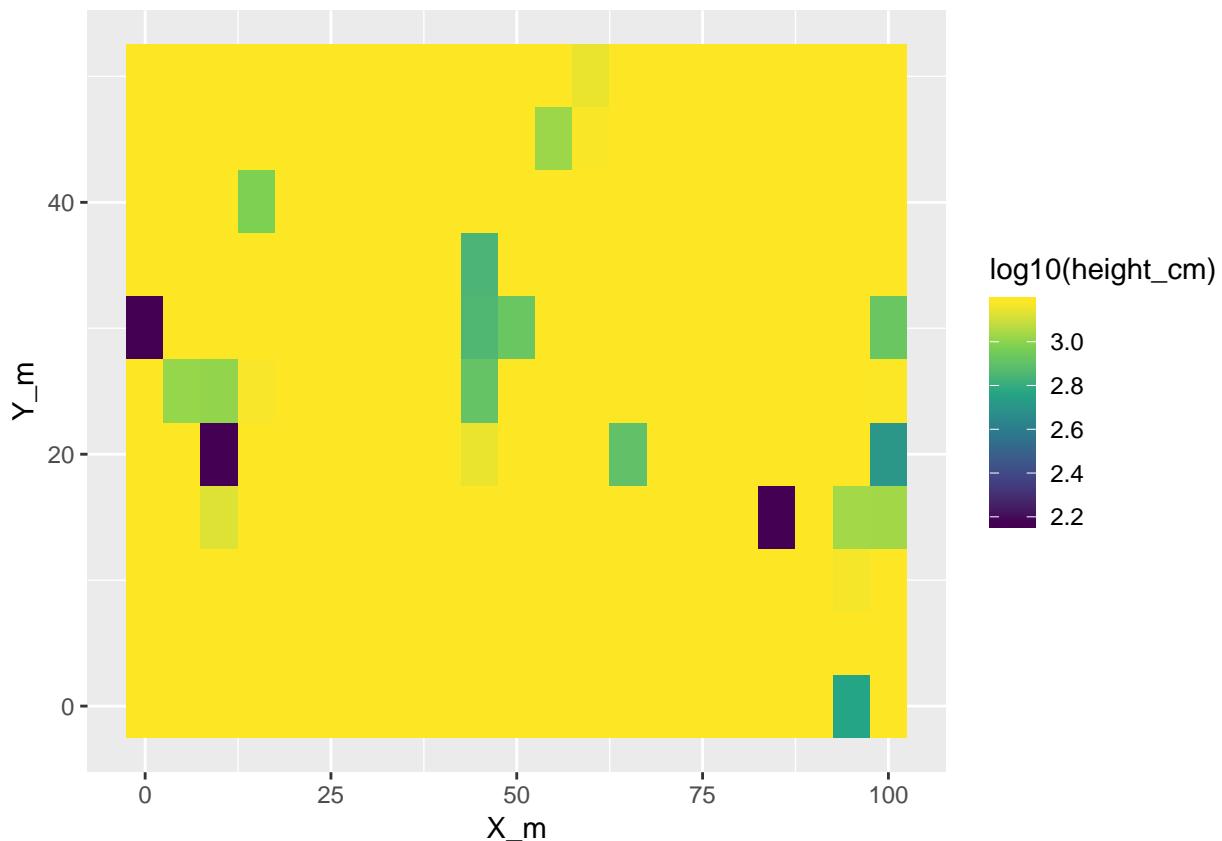
carb <- carb %>%
  rename(year= `*Year` ) %>%
  mutate(date = parse_date_time(Date, '%d-%m-%y')) %>%
  select(-Date)

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

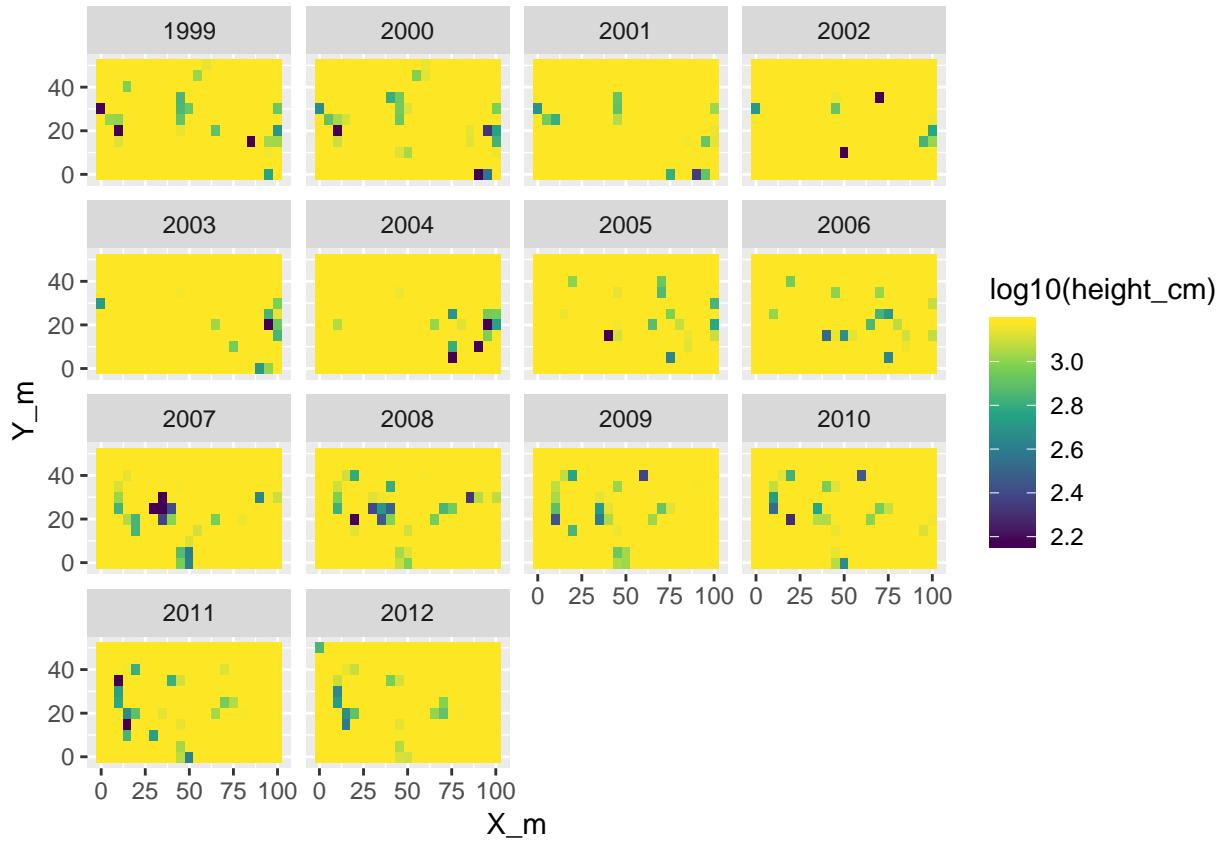
```



```
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

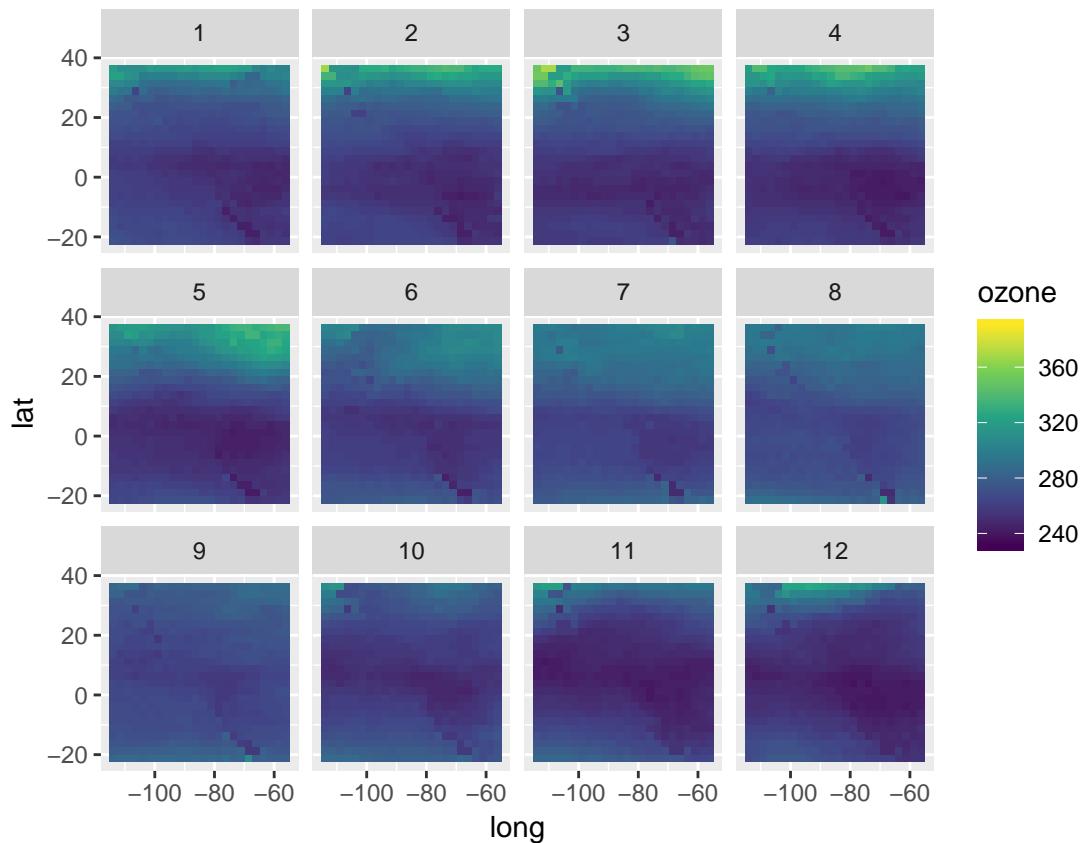
## List of 2
## $ mets:List of 7
##   ..$ cloudhigh : num [1:24, 1:24, 1:12, 1:6] 26 20 16 13 7.5 8 14.5 19.5 22.5 21 ...
##   ..$ cloudlow  : num [1:24, 1:24, 1:12, 1:6] 7.5 11.5 16.5 20.5 26 30 29.5 26.5 27.5 26 ...
##   ..$ cloudmid : num [1:24, 1:24, 1:12, 1:6] 34.5 32.5 26 14.5 10.5 9.5 11 17.5 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 %>%
  as_tibble() %>%
  ggplot(data=., aes(long,lat))+
  geom_raster(aes(fill=ozone))+
  coord_equal()+
  scale_fill_viridis_c() +
  facet_wrap(~month)

```

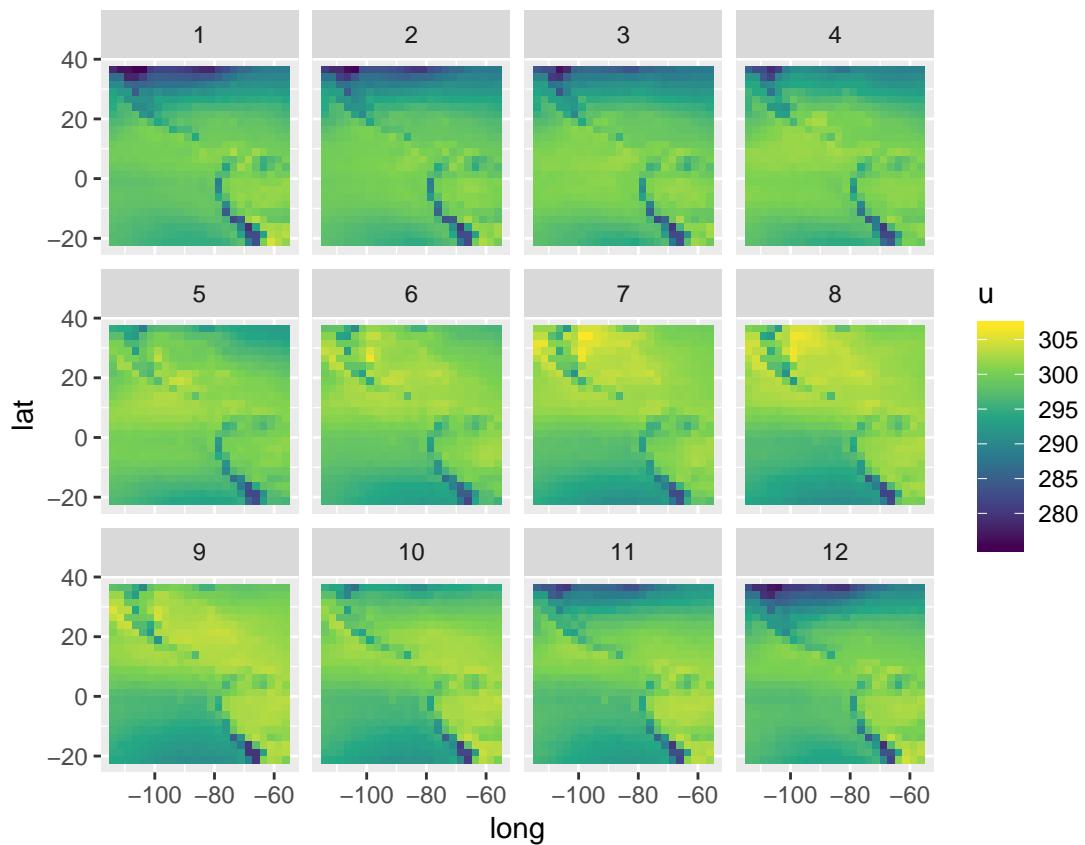


Plot the mean monthly temperature

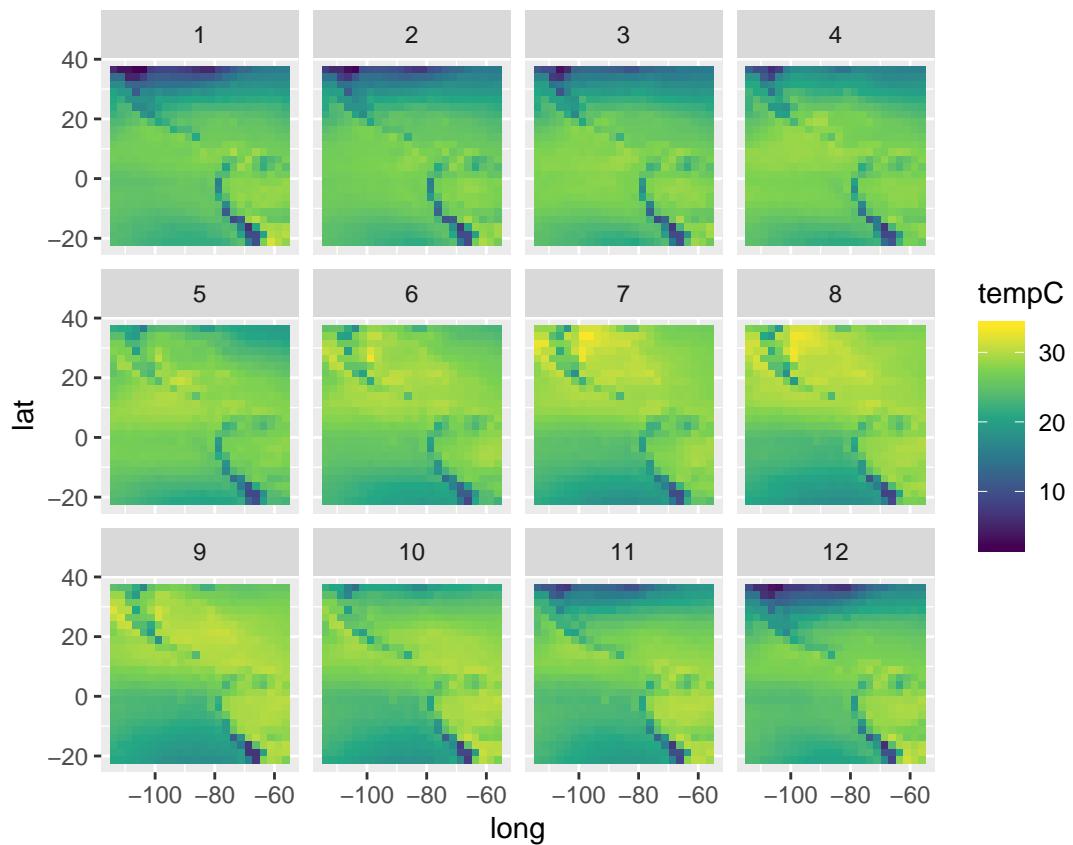
```

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)

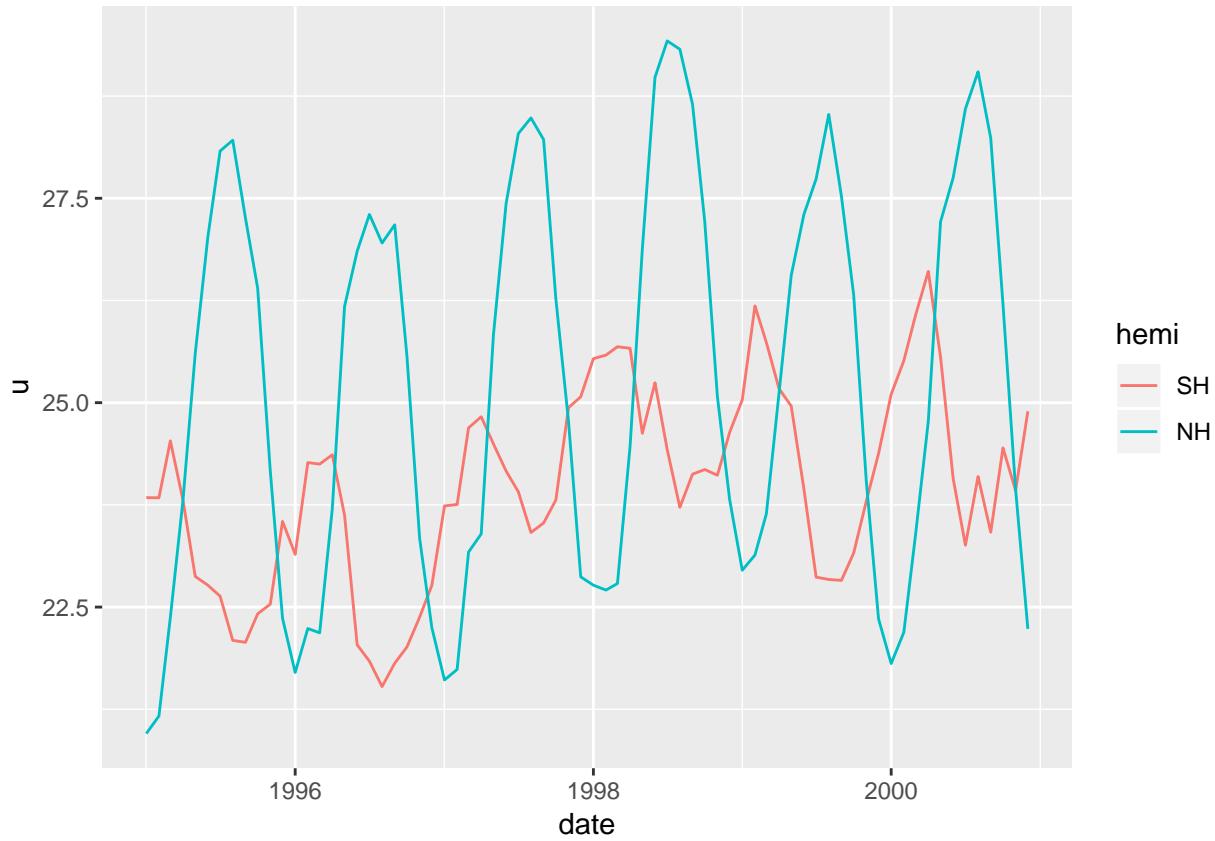
```



```
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) %>%
  # summarise(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) %>%
  summarise(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()
```

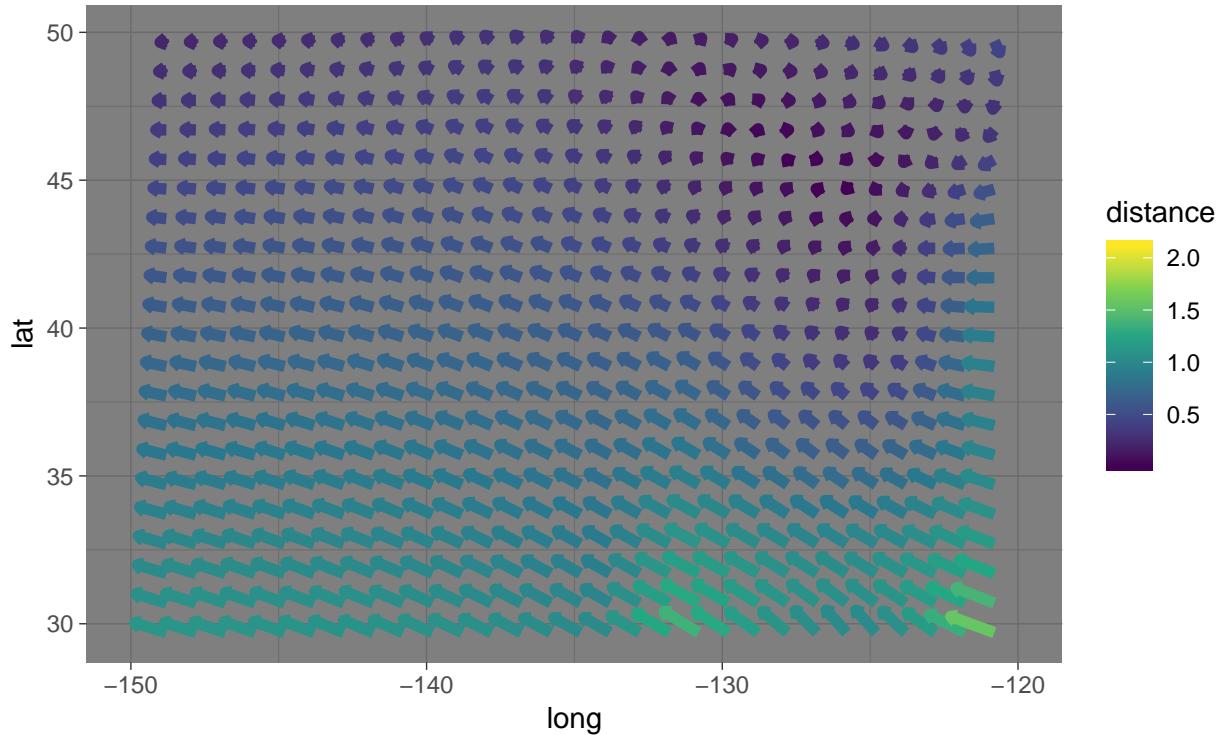


Seals

```
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))+
```



Plotting hurricane tracks

```

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>
names(storms)

## [1] "name"          "year"          "month"         "day"           "hour"
## [6] "lat"           "long"          "status"        "category"      "wind"
## [11] "pressure"      "ts_diameter"   "hu_diameter"
# bad!
storms %>%
  ggplot(data=., aes(long,lat,size=category))+

```

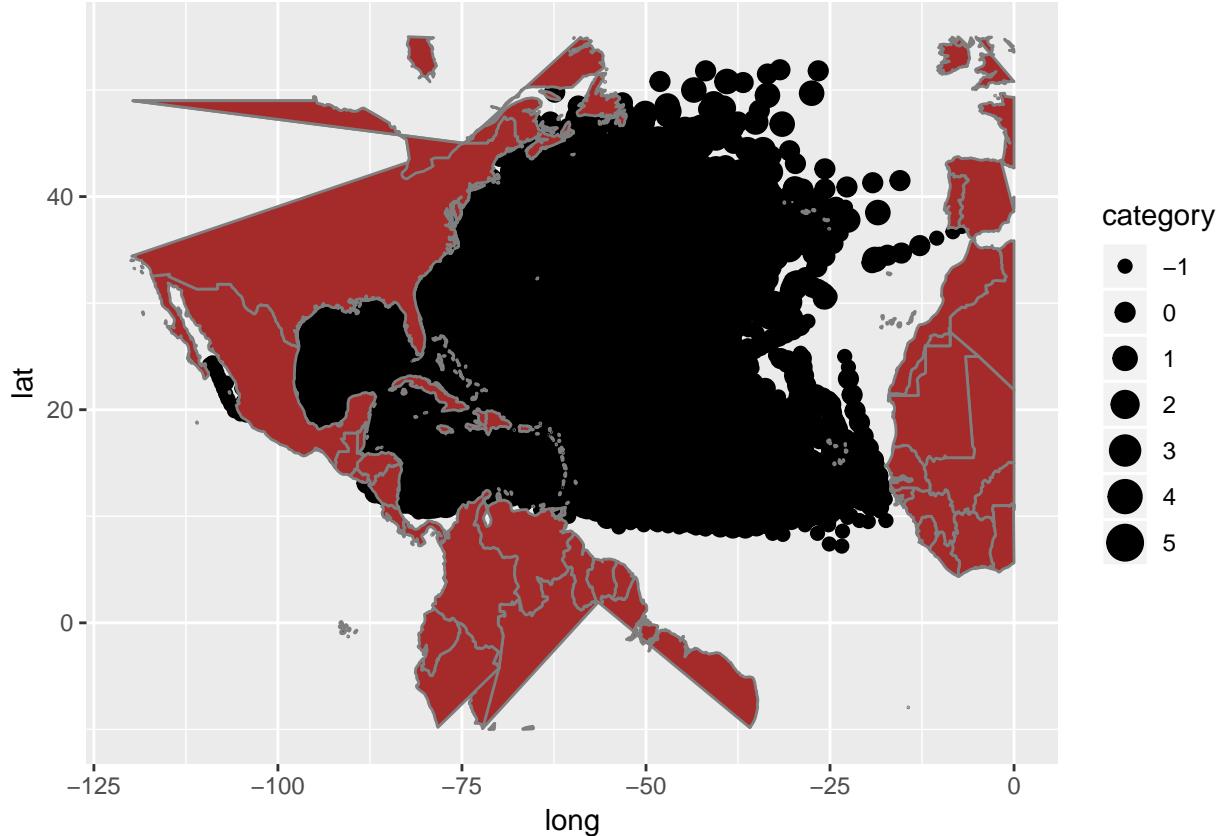
```

geom_point()+
# borders("world")+
scale_x_continuous(limits=c(-120,0))+  

scale_y_continuous(limits = c(-10,55))+  

borders('world', fill = "brown")

```



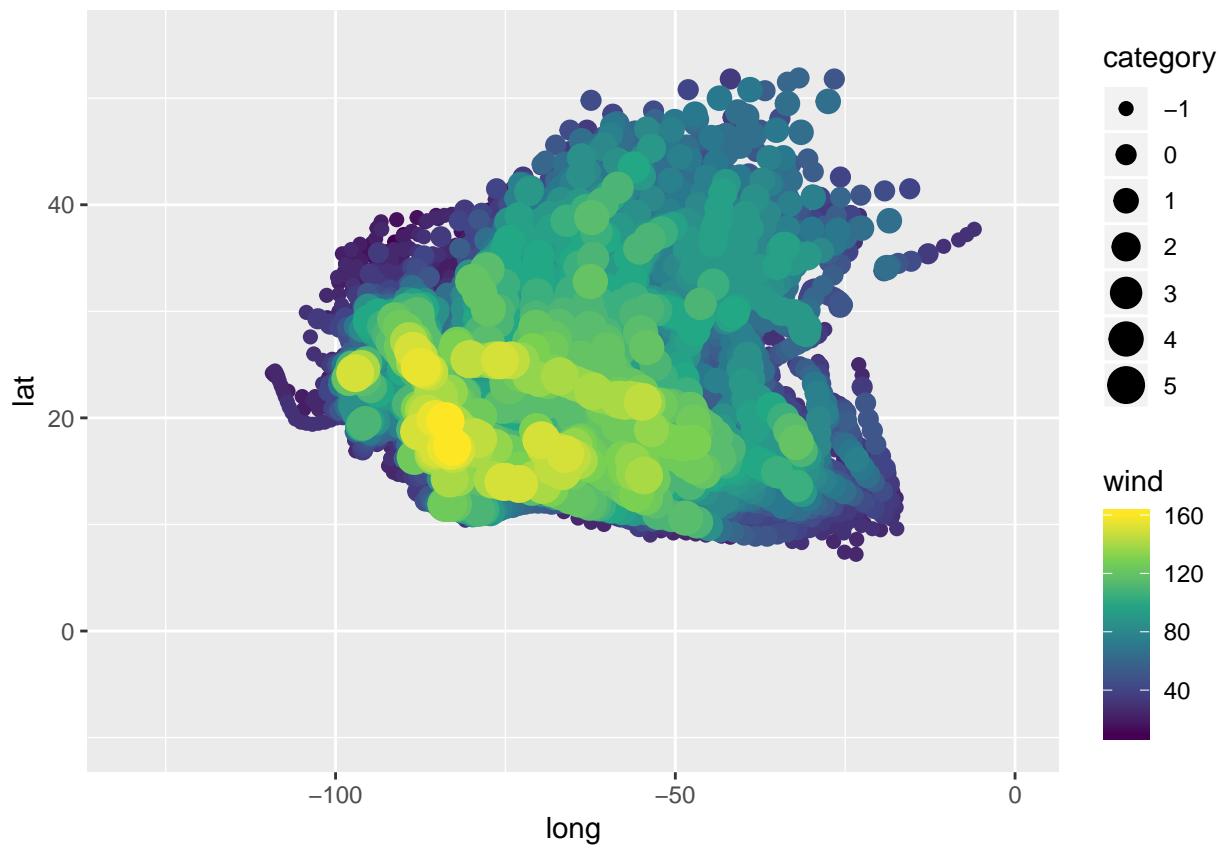
```

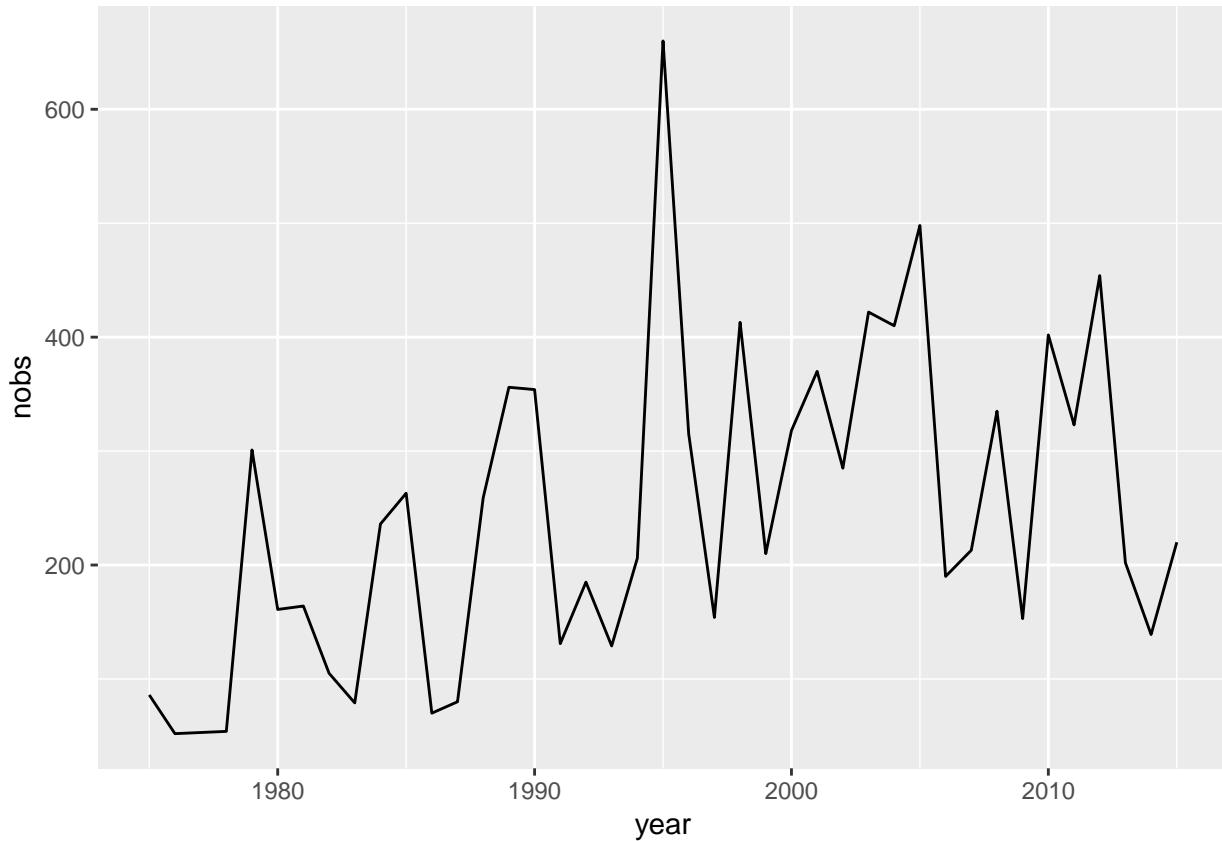
# bad!
storms %>%
  arrange(wind) %>%
  ggplot(data=., aes(long,lat,size=category,color=wind))+  

# borders("coast")+
  geom_point()+
  scale_color_viridis_c()+
  scale_x_continuous(limits=c(-130,0))+  

  scale_y_continuous(limits = c(-10,55))

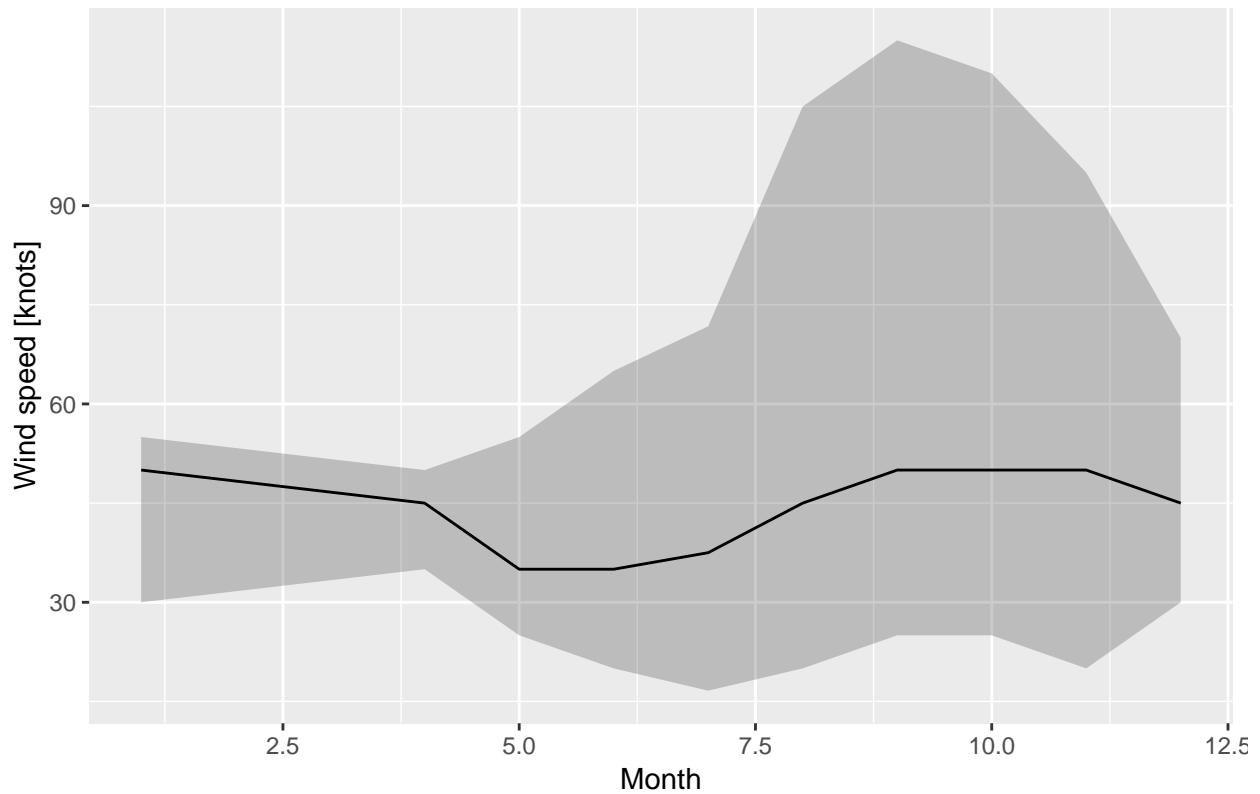
```





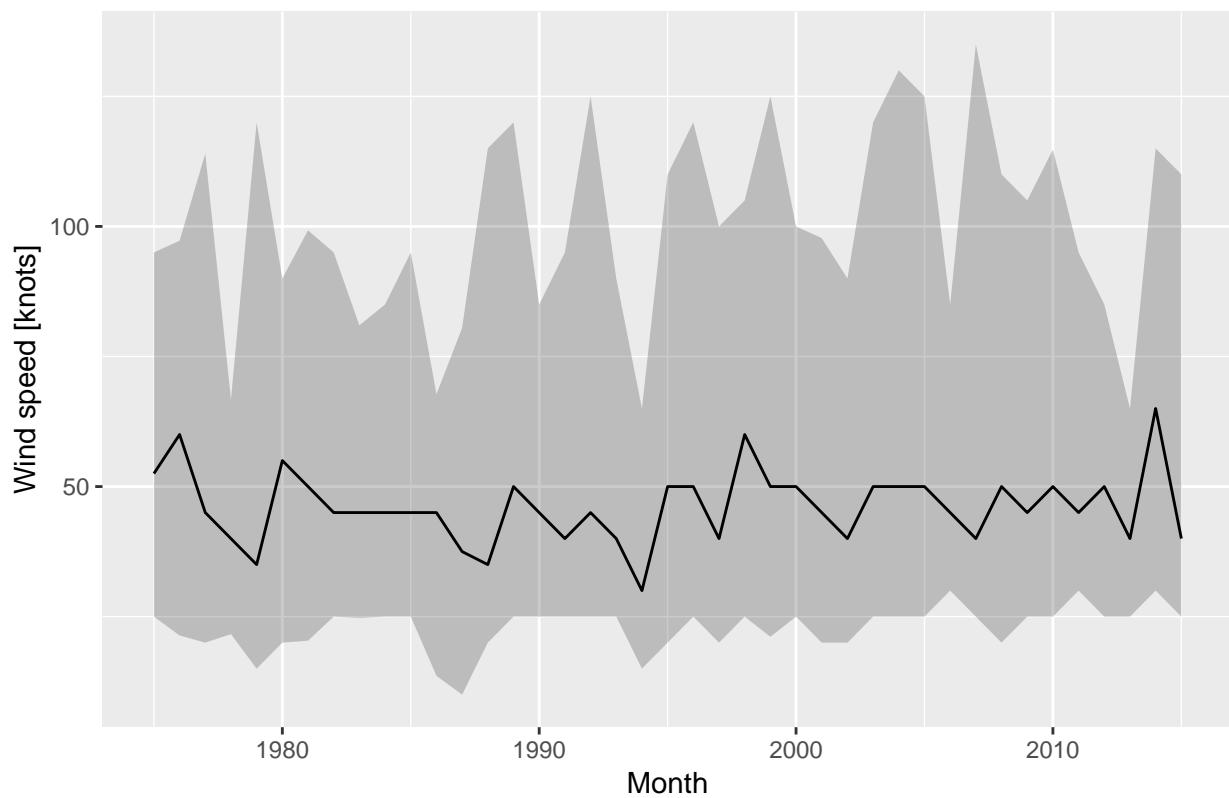
```
storms %>% group_by(month) %>% summarize(wind_25=quantile(wind,0.025),
                                              wind_50=median(wind),
                                              wind_75=quantile(wind, 0.95)) %>%
  ungroup() %>%
  ggplot(data=., aes(month, wind_50))+ 
  geom_ribbon(aes(x=month, 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")
```

95% quantile rate of hurricane wind speed

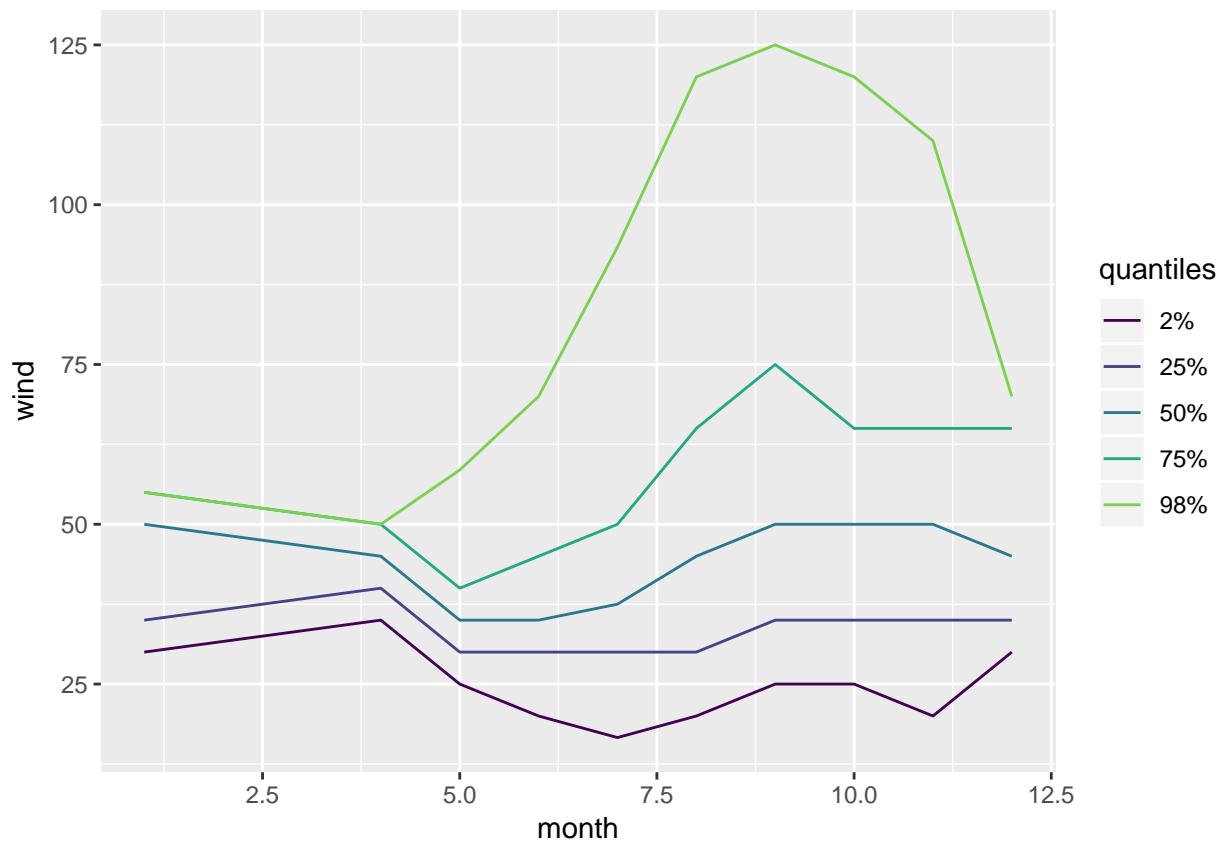


```
# swap 'month' for 'year'
storms %>% group_by(year) %>% summarize(wind_25=quantile(wind,0.025),
                                              wind_50=median(wind),
                                              wind_75=quantile(wind, 0.95)) %>%
  ungroup() %>%
  ggplot(data=., aes(year, wind_50))+ 
  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")
```

95% quantile rate of hurricane wind speed



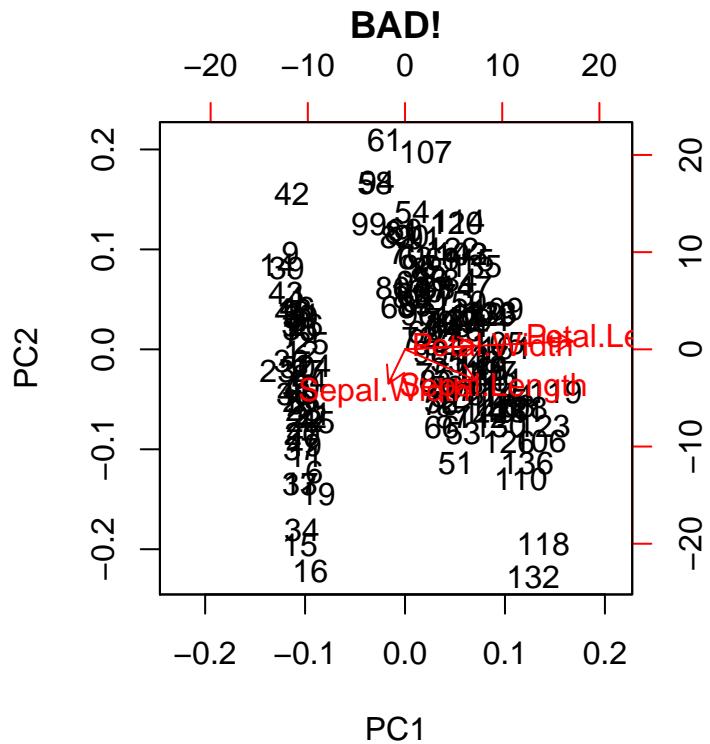
```
# advanced!
p = c(0.025, 0.25, 0.5, 0.75, 0.975)
storms %>%
  group_by(month) %>%
  summarise(quantiles = list(sprintf("%1.0f%%", p*100)),
            wind = list(quantile(wind, p))) %>%
  unnest %>%
  ggplot(data=., aes(month, wind, color=quantiles))+
  geom_line()+
  scale_color_viridis_d(end=0.8)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



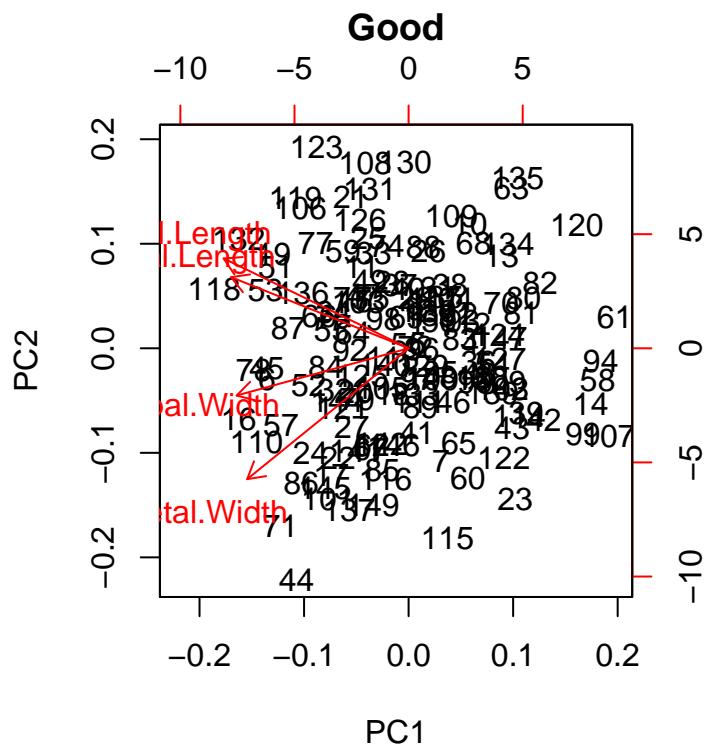
PCA example with columns scaling

```
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
```



```
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 name

## # 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>
# 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

# 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

# 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> <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>
# 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

```

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

dir.create("data/SouthAmerica")

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

  ObjectID  FIPS_CNTRY  GMI_CNTRY  ISO_2DIGIT  ISO_3DIGIT





  ISO_NUM  CNTRY_NAME  LONG_NAME  ISOSHRTNAM  UNSHRTNAM

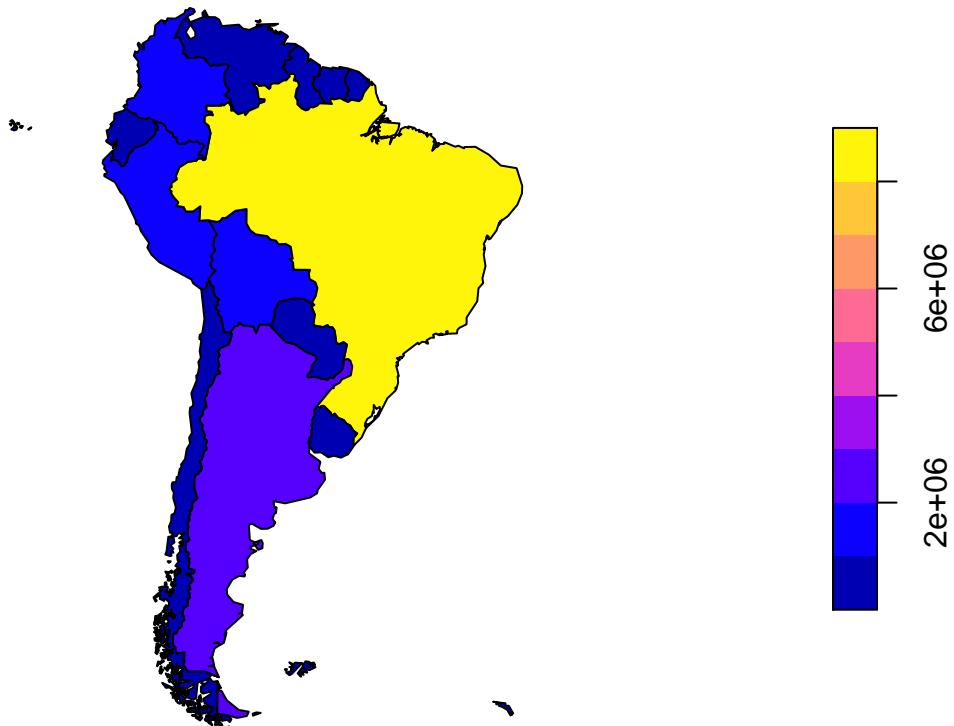




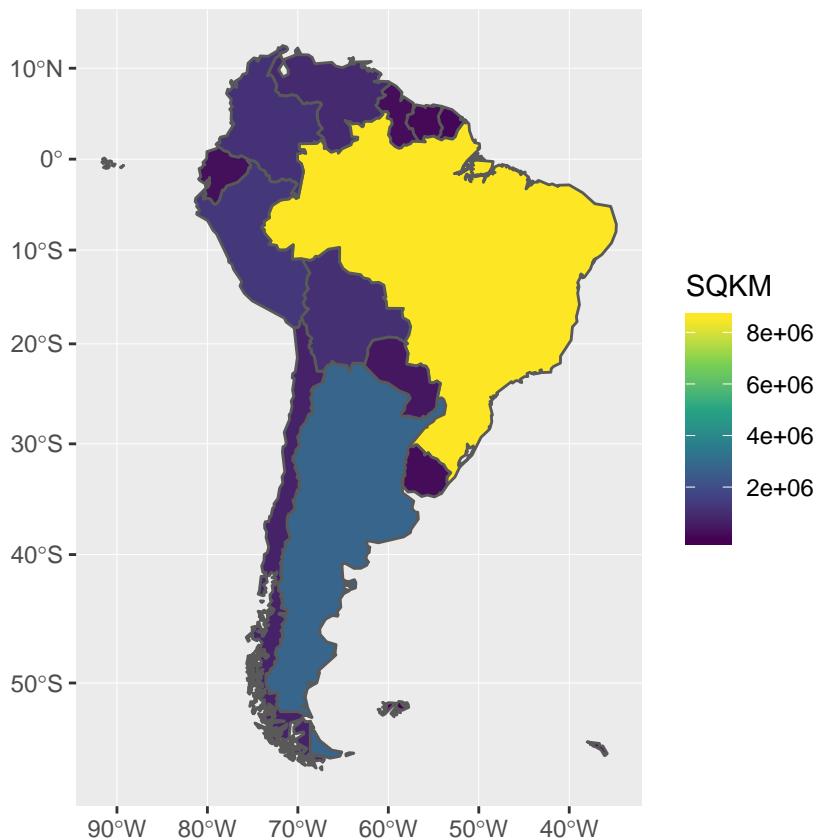

```

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

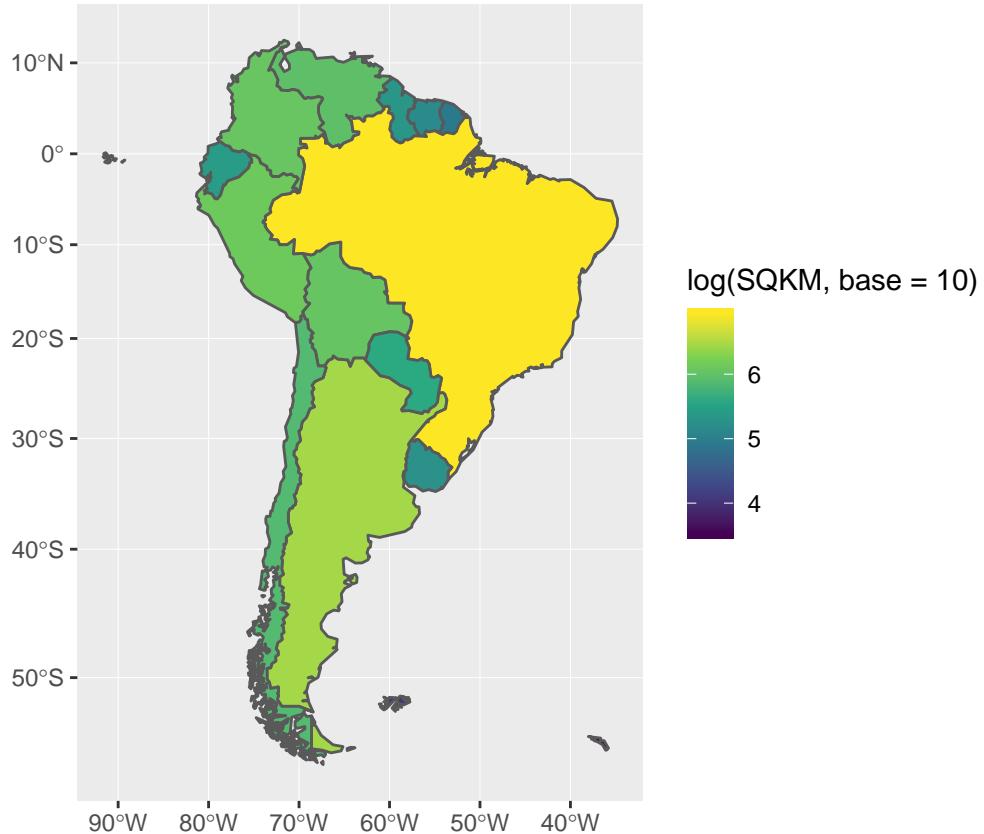
SQKM



```
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

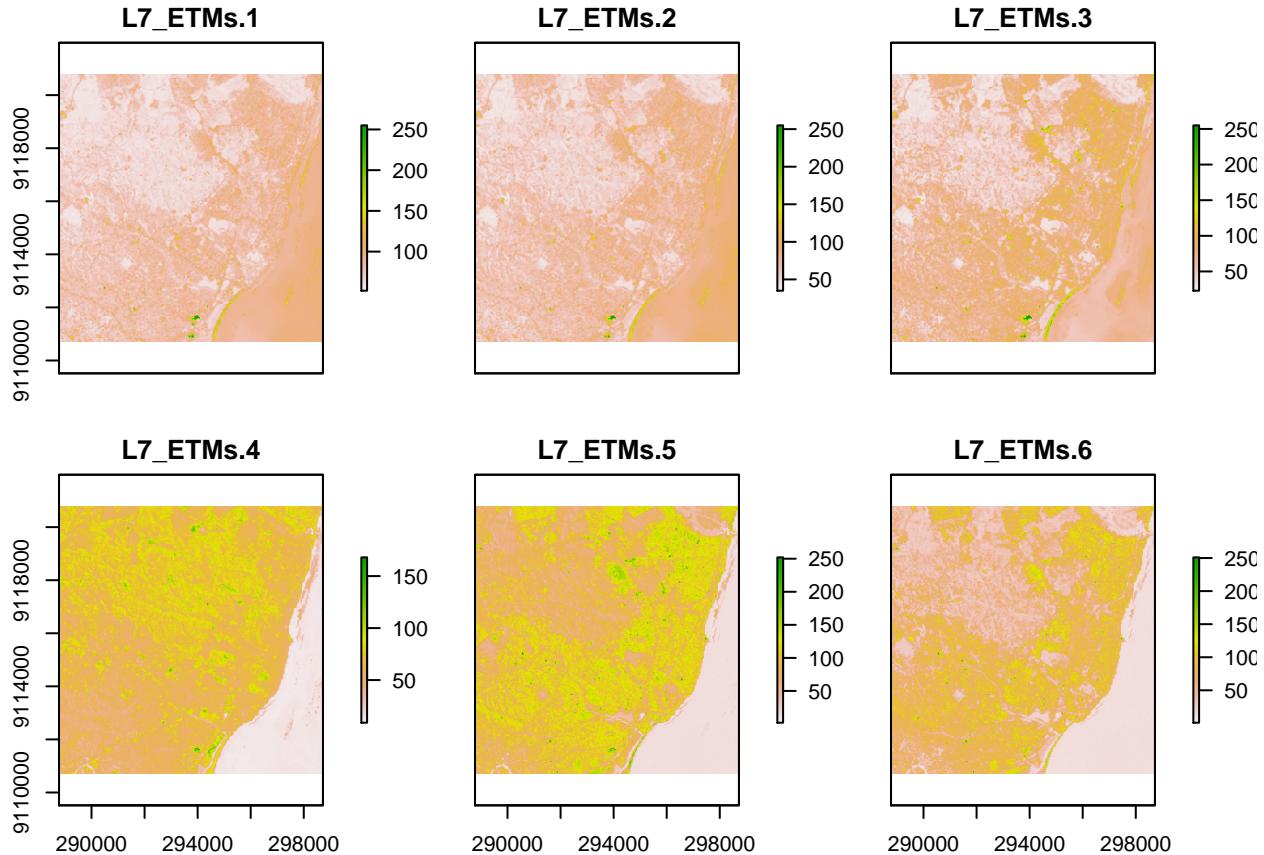
This works for visualizing single bands of smallish rasters (< 1000x1000)

```
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.

```

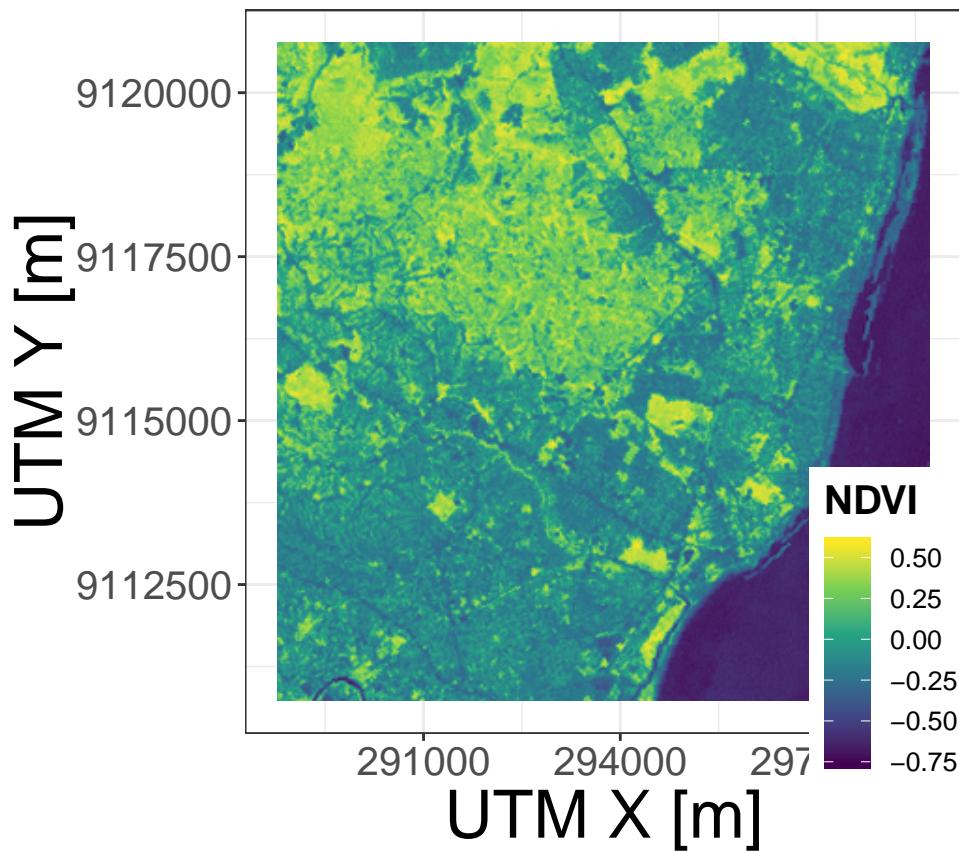


```

17 <- raster::as.data.frame(r, xy=T) %>% as_tibble()
17 <- 17 %>%
  mutate(ndvi=(L7_ETMs.4-L7_ETMs.3)/(L7_ETMs.4+L7_ETMs.3))
17 %>%
  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

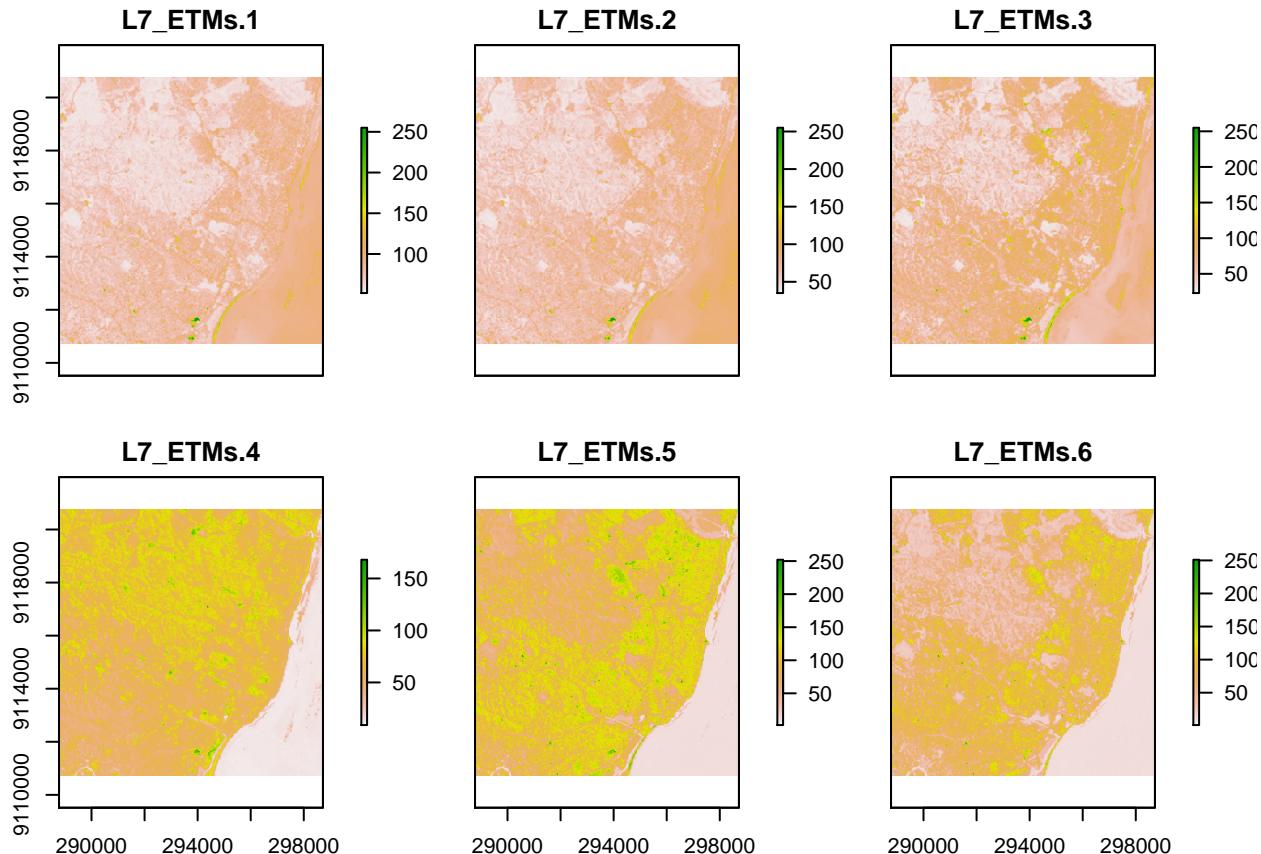
```
# [https://www.r-spatial.org/r/2018/03/22/stars2.html]
#! CAUTIONARY NOTE ! if you are processing Gbs worth of raster or other spatiotemporal data, consider
# doing it in Python
library(stars)
tif = system.file("tif/L7_ETMs.tif", package = "stars")
(r = raster::stack(tif))

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

```

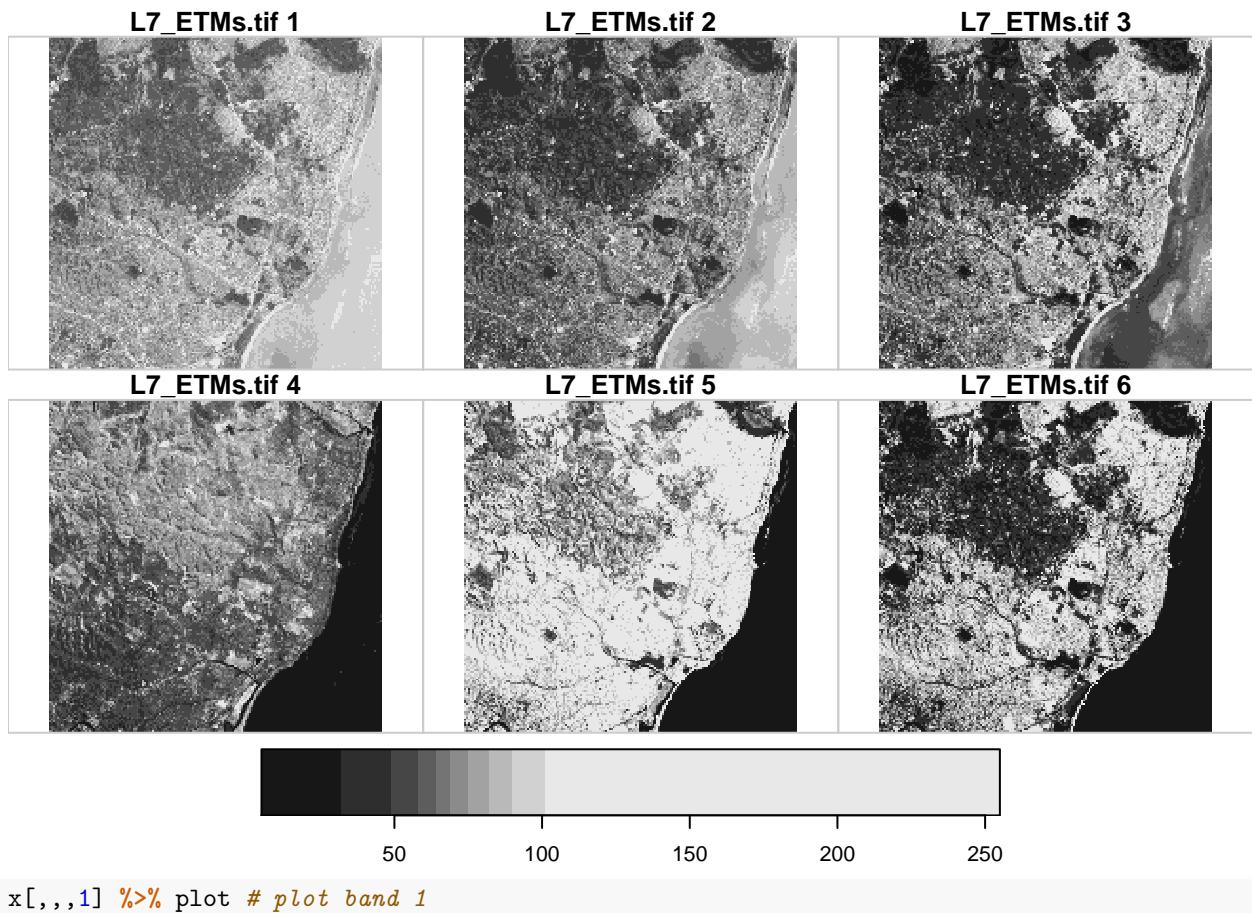


```
(x = read_stars(tif))
```

```

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

```



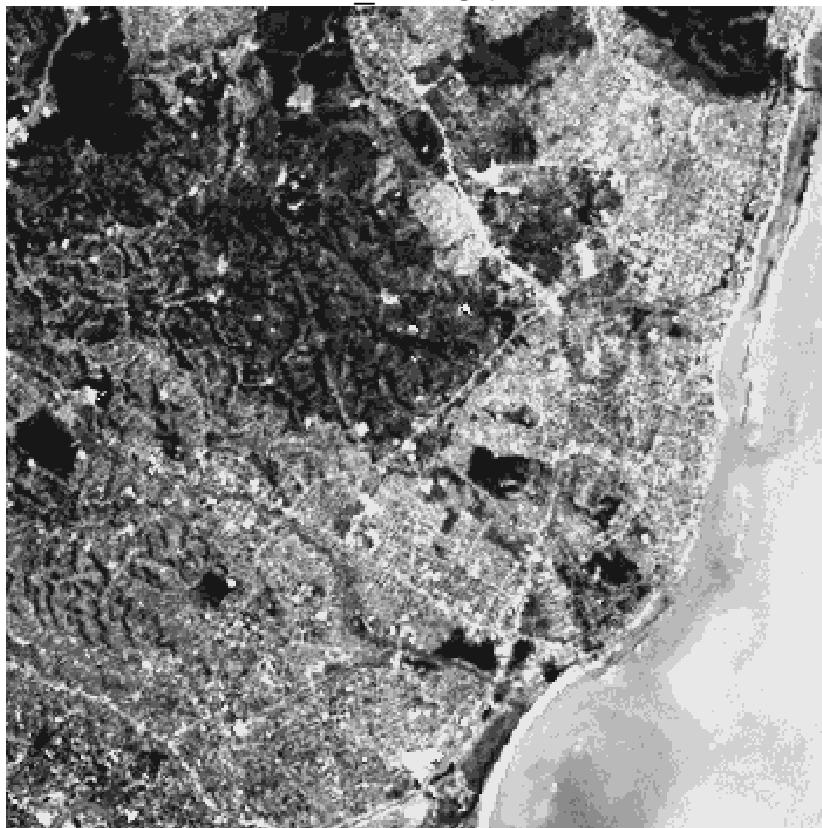
```
x[,,1] %>% plot # plot band 1
```

L7_ETMs.tif



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

L7_ETMs.tif

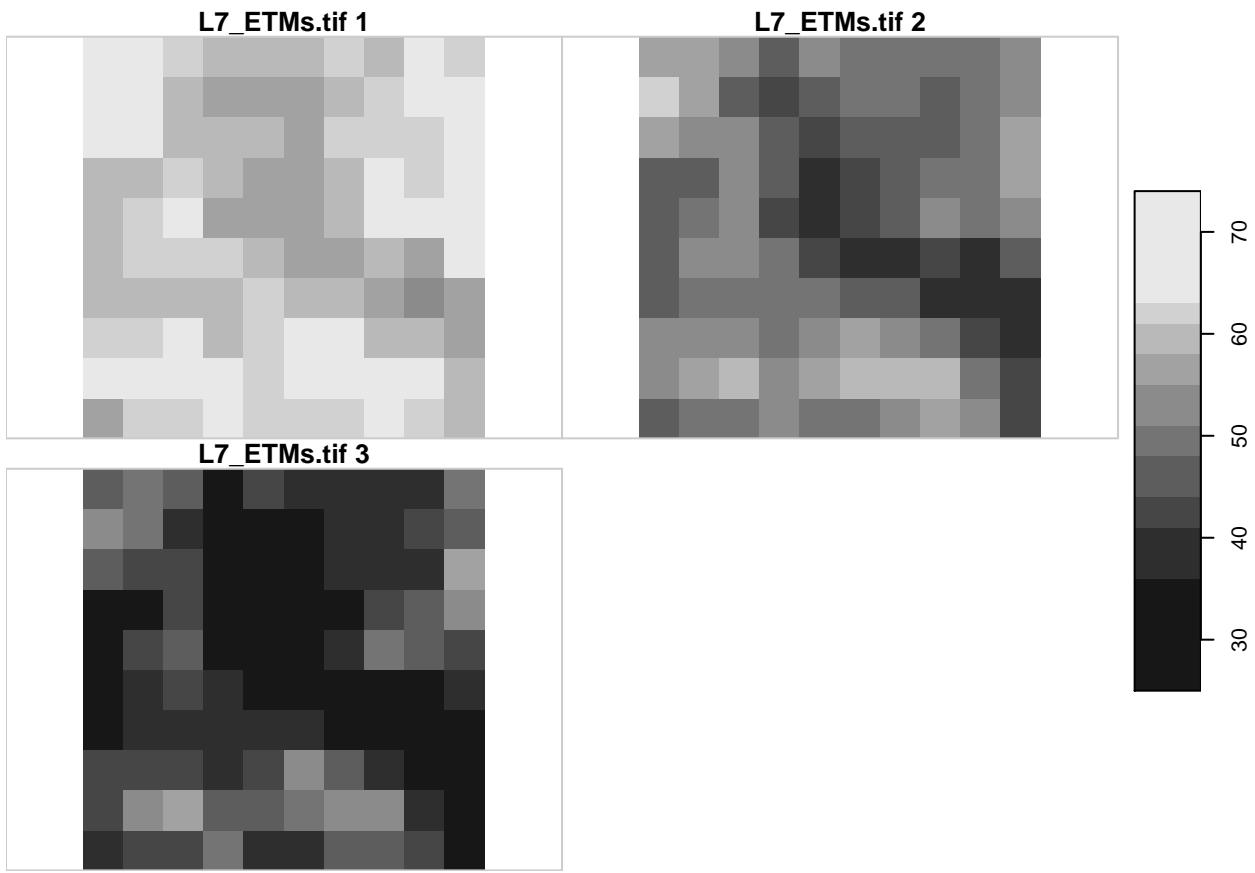


```
x[,1:100,1:100,1] %>% plot # plot spatial subset
```

L7_ETMs.tif



```
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)

```

