AdvTutorial

Nicholas Gawron & Livia Popa

1/28/2022

Advanced Tutorial

Importing Data From NCEI

Text on importing **with an API**.

- This explains how to extract data without a package for certain API's

```
##library(httr)
#library(jsonlite)
#base_url <- "https://www.ncdc.noaa.gov/cdo-web/api/v2/"
#endpoint <- "datasets"
#token <- "gykUMKPJzpcpQonBrUWjbFqYevOPkhwc"
#full_url <- paste0(base_url, "/",endpoint,"/")
#Raw  <- GET(full_url)

#curl -H "token:gykUMKPJzpcpQonBrUWjbFqYevOPkhwc" #"https://www.ncdc.noaa.gov/cdo-web/api/v2/datasets"
```

Attempting to pull data from RNOAA - not super successful rn

```
options(noaakey = "--key goes here --")
# a comment goes here

#list of all stations
ghcnd_stations()
```

```
## using cached file: C:\Users\nickg\AppData\Local/Cache/R/noaa_ghcnd/ghcnd-stations.rds
```

```
## date created (size, mb): 2022-02-09 11:44:30 (2.159)
```

```
## using cached file: C:\Users\nickg\AppData\Local/Cache/R/noaa_ghcnd/ghcnd-inventory.rds
```

```
## date created (size, mb): 2022-02-09 11:49:01 (2.669)
```

```
## # A tibble: 710,581 x 11
##    id          latitude longitude elevation state name   gsn_flag wmo_id element
##    <chr>          <dbl>     <dbl>     <dbl> <chr> <chr>  <chr>    <chr>  <chr>
##  1 ACW00011604     17.1     -61.8      10.1 ""    ST JO~ ""       ""     TMAX
##  2 ACW00011604     17.1     -61.8      10.1 ""    ST JO~ ""       ""     TMIN
##  3 ACW00011604     17.1     -61.8      10.1 ""    ST JO~ ""       ""     PRCP
##  4 ACW00011604     17.1     -61.8      10.1 ""    ST JO~ ""       ""     SNOW
##  5 ACW00011604     17.1     -61.8      10.1 ""    ST JO~ ""       ""     SNWD
##  6 ACW00011604     17.1     -61.8      10.1 ""    ST JO~ ""       ""     PGTM
##  7 ACW00011604     17.1     -61.8      10.1 ""    ST JO~ ""       ""     WDFG
##  8 ACW00011604     17.1     -61.8      10.1 ""    ST JO~ ""       ""     WSFG
##  9 ACW00011604     17.1     -61.8      10.1 ""    ST JO~ ""       ""     WT03
## 10 ACW00011604     17.1     -61.8      10.1 ""    ST JO~ ""       ""     WT08
## # ... with 710,571 more rows, and 2 more variables: first_year <int>,
## #   last_year <int>
```

```r
#tibble  of all stations given certain lattitude and logitudes
raliegh_stations<-ghcnd_stations()%>%dplyr::filter(latitude>34 & latitude<36 & longitude>-80 & longitude
```

```
## using cached file: C:\Users\nickg\AppData\Local/Cache/R/noaa_ghcnd/ghcnd-stations.rds

## date created (size, mb): 2022-02-09 11:44:30 (2.159)

## using cached file: C:\Users\nickg\AppData\Local/Cache/R/noaa_ghcnd/ghcnd-inventory.rds

## date created (size, mb): 2022-02-09 11:49:01 (2.669)
```

```r
raliegh_stations
```

```
## # A tibble: 7,128 x 11
##    id          latitude longitude elevation state name   gsn_flag wmo_id element
##    <chr>          <dbl>     <dbl>     <dbl> <chr> <chr>  <chr>    <chr>  <chr>
##  1 US1NCAL0010     36.0     -79.3      172. NC    GRAHA~ ""       ""     PRCP
##  2 US1NCAL0010     36.0     -79.3      172. NC    GRAHA~ ""       ""     SNOW
##  3 US1NCAL0013     36.0     -79.3      172. NC    GRAHA~ ""       ""     PRCP
##  4 US1NCAL0013     36.0     -79.3      172. NC    GRAHA~ ""       ""     SNOW
##  5 US1NCAL0014     36.0     -79.3      176. NC    HAW R~ ""       ""     PRCP
##  6 US1NCAL0014     36.0     -79.3      176. NC    HAW R~ ""       ""     SNOW
##  7 US1NCAL0014     36.0     -79.3      176. NC    HAW R~ ""       ""     SNWD
##  8 US1NCAL0014     36.0     -79.3      176. NC    HAW R~ ""       ""     DAPR
##  9 US1NCAL0014     36.0     -79.3      176. NC    HAW R~ ""       ""     MDPR
## 10 US1NCAL0014     36.0     -79.3      176. NC    HAW R~ ""       ""     WESD
## # ... with 7,118 more rows, and 2 more variables: first_year <int>,
## #   last_year <int>
```

```r
real_ral<-ghcnd(stationid='GHCND:US1NCAL0013')
real_ral
```

```
## # A tibble: 0 x 0
```

```r
Ralz_dat <- ncdc(datasetid='GHCND', stationid='GHCND:US1NCAL0013', datatypeid=c('TAVG','PRCP'), startda
```

```
## Warning: Error: (400) - The token parameter provided is not valid.
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```r
Ralz_dat
```

```
## $meta
## [1] NA
##
## $data
## # A tibble: 0 x 0
##
## attr(,"class")
## [1] "ncdc_data"
```

```r
#medeo_tidy
```

Machine Learning

Content from beg. lectures for time being

```r
cardinal <- read_csv("cardinal_data.csv",
     col_types = list(`Average Air Temperature (F)` = col_number(),
         `Maximum Air Temperature (F)` = col_number(),
         `Minimum Air Temperature (F)` = col_number(),
         `Average Experimental Leaf Wetness (mV)` = col_number(),
         `Total Precipitation (in)` = col_number(),
         `Average Relative Humidity (%)` = col_number(),
         `Average Soil Moisture (m3/m3)` = col_number(),
         `Average Soil Temperature (F)` = col_number(),
         `Average Solar Radiation (W/m2)` = col_number(),
         `Average Station Pressure (mb)` = col_number()))
```

```
## Warning: One or more parsing issues, see `problems()` for details
```

```r
cardinal<-drop_na(cardinal)
str(cardinal)
```

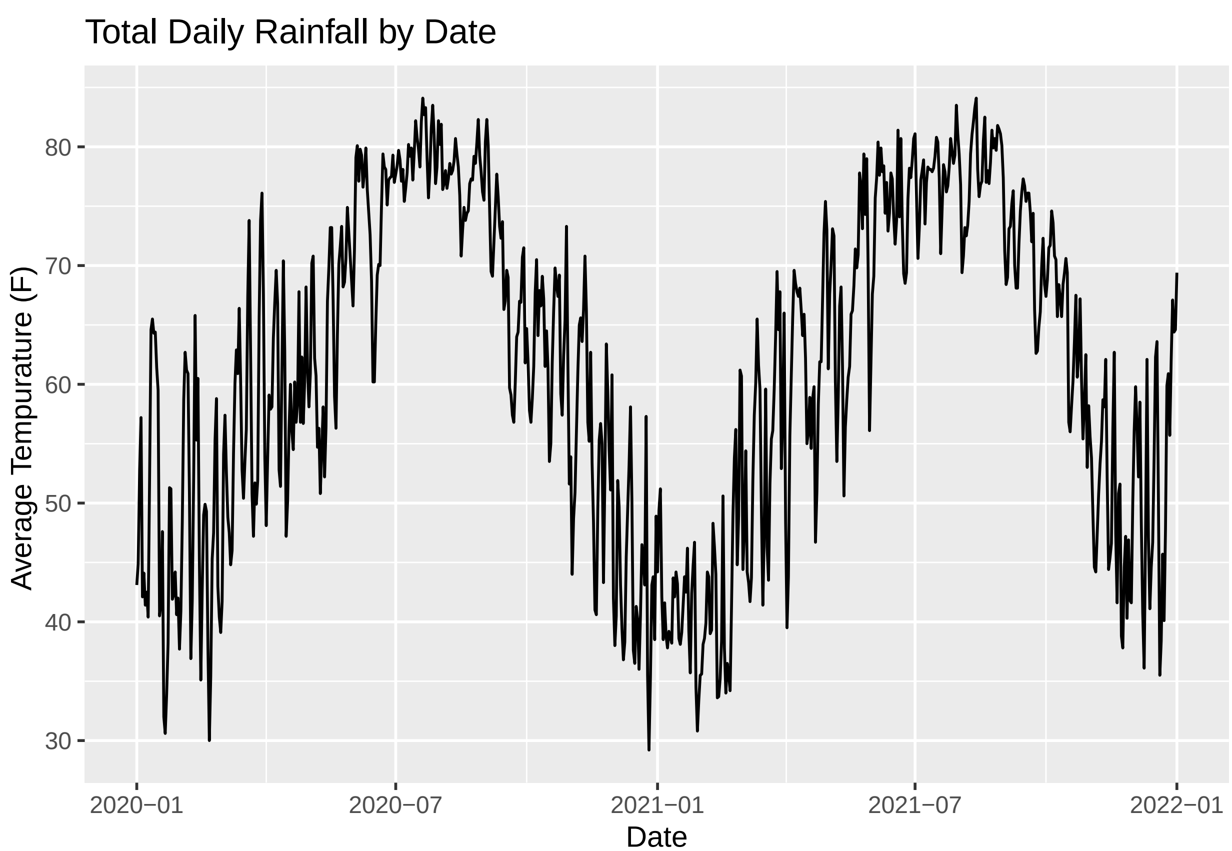
```
## tibble [729 x 11] (S3: tbl_df/tbl/data.frame)
##  $ Date                                   : chr [1:729] "1/1/20" "1/2/20" "1/3/20" "1/4/20" ...
##  $ Average Air Temperature (F)            : num [1:729] 43.1 44.9 52.8 57.2 42.1 44.1 41.4 42.5 40.4 5
##  $ Maximum Air Temperature (F)            : num [1:729] 53.6 55.4 64.9 65.1 50.5 58.5 52 57.6 50.5 65
##  $ Minimum Air Temperature (F)            : num [1:729] 35.1 35.2 45.7 42.6 34.9 32 31.3 29.7 31.3 38
```

```
##  $ Average Experimental Leaf Wetness (mV): num [1:729] 266 274 362 373 265 ...
##  $ Total Precipitation (in)              : num [1:729] 0 0.05 0.95 0.52 0 0 0.07 0 0 0 ...
##  $ Average Relative Humidity (%)         : num [1:729] 63.8 72 92.1 83.5 57 ...
##  $ Average Soil Moisture (m3/m3)         : num [1:729] 0.28 0.28 0.29 0.35 0.33 0.31 0.3 0.3 0.3 0.29
##  $ Average Soil Temperature (F)          : num [1:729] 48.6 47.6 51 54.6 48.3 46.1 44.6 43.3 43.3 46
##  $ Average Solar Radiation (W/m2)        : num [1:729] 134.8 66 31.1 44.9 135.4 ...
##  $ Average Station Pressure (mb)         : num [1:729] 999 1003 998 993 1005 ...
```

```
cardinal$Date<-as.Date(cardinal$Date, tryFormats= c("%m/%d/%y"))
view(cardinal)

#changes col names
colnames(cardinal)=c("date","AvgT","MaxT","MinT","AvgLw","Tprep","AvgHum","AvgSm","AvgSt","AvgSr","AvgS



cardinal$IfRain<- (cardinal$Tprep>0)
cardinal$IfRain<-as.factor(as.integer(cardinal$IfRain))
```

Basic Plotting with Ggplot

```
ggplot(cardinal,aes(x=date,y=AvgT))+geom_line()+labs(title="Total Daily Rainfall by Date",y="Average Te
```



- EDA is how we can motivate future ML models!

- We can use forecasting to extend this trend!

Testing and training data

- concept of seeing how well a model works
- *cut to nice images of cross-validation?*

TIme Series forecasting

- We were thinking of using logistic regression but may not?

```
#logistic regression
fit1 = glm(IfRain~date+AvgT+AvgLw+AvgSt+AvgSr, data=cardinal, family="binomial")
summary(fit1)
```

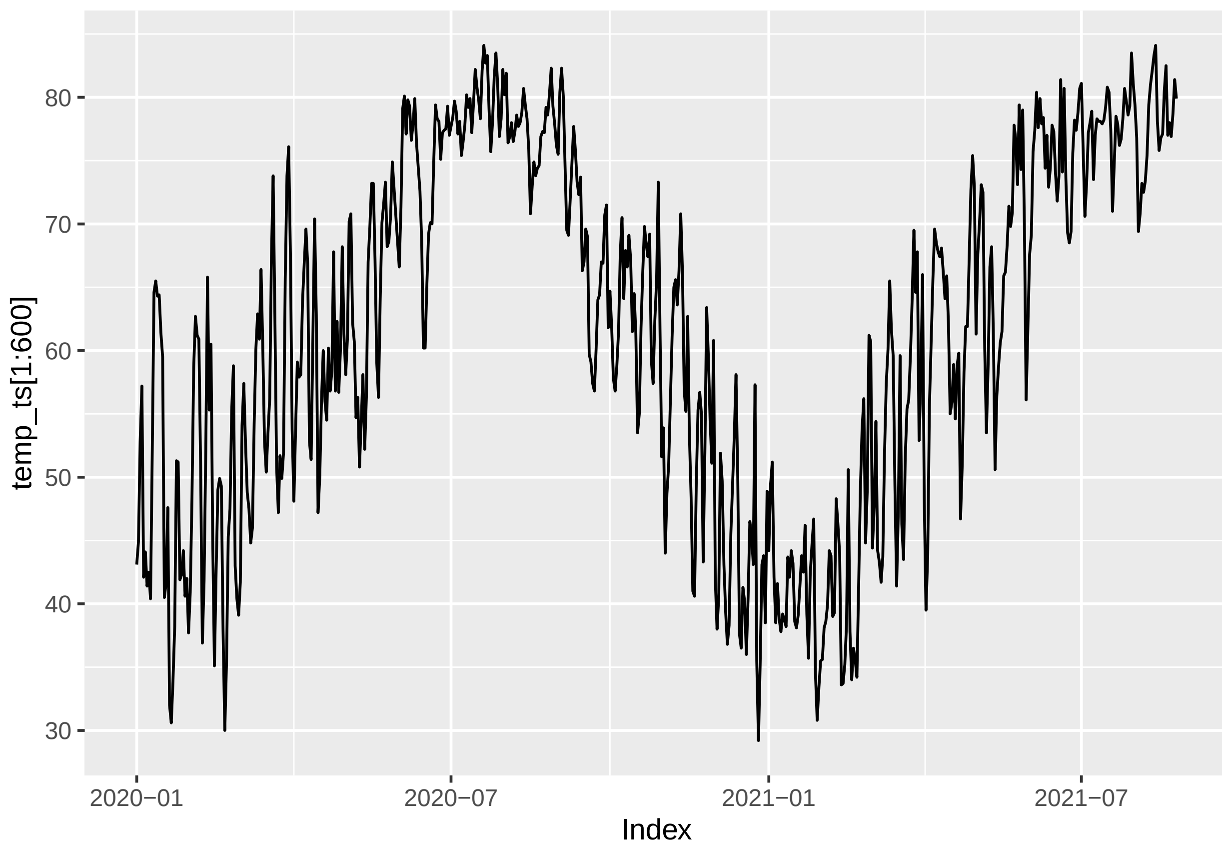
```
##
## Call:
## glm(formula = IfRain ~ date + AvgT + AvgLw + AvgSt + AvgSr, family = "binomial",
##     data = cardinal)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2141  -0.6948  -0.3678   0.6401   2.5534
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) 28.6554531  8.3881623   3.416 0.000635 ***
## date        -0.0019994  0.0004614  -4.333 1.47e-05 ***
## AvgT         0.0138684  0.0228607   0.607 0.544084
## AvgLw        0.0194774  0.0027333   7.126 1.03e-12 ***
## AvgSt        0.0630555  0.0242543   2.600 0.009329 **
## AvgSr       -0.0162324  0.0017147  -9.466  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 971.07  on 728  degrees of freedom
## Residual deviance: 662.62  on 723  degrees of freedom
## AIC: 674.62
##
## Number of Fisher Scoring iterations: 5
```

```
#predict something with logistic regression
```

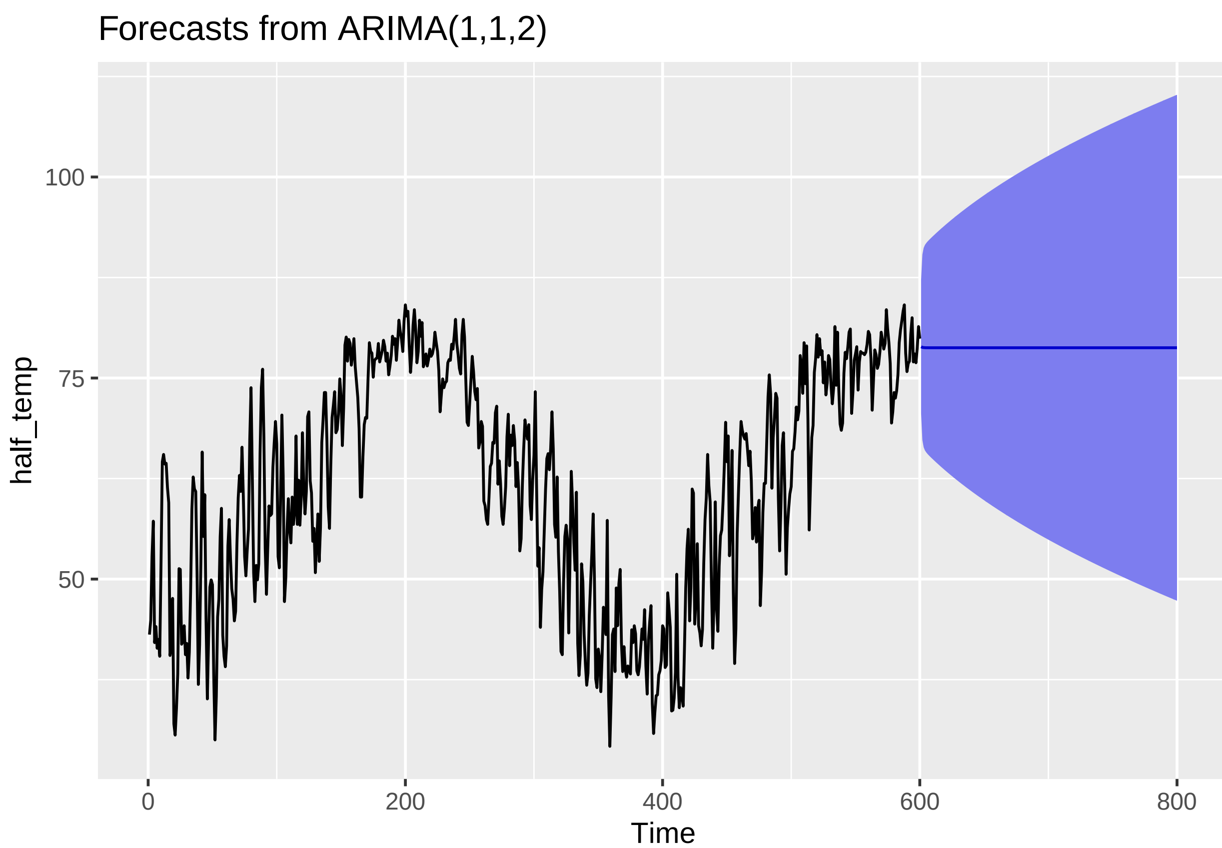
```
# n climate  grid data

temp_ts <- xts(cardinal$AvgT,cardinal$date)
head(temp_ts)
```

```
##              [,1]
## 2020-01-01 43.1
## 2020-01-02 44.9
## 2020-01-03 52.8
## 2020-01-04 57.2
## 2020-01-05 42.1
## 2020-01-06 44.1
```

```r
autoplot(temp_ts[1:600])
```




```r
half_temp <-temp_ts[1:600]

library(forecast)
d.arima <- auto.arima(half_temp)
d.forecast <- forecast(d.arima, level = c(90), h = 200)
autoplot(d.forecast)
```



PCA to cluster rain variable

- using cardinal data to obsevre *if* there is clustering
- used for future models
- helps us describe higher dimensional data with **less**

Three general steps:

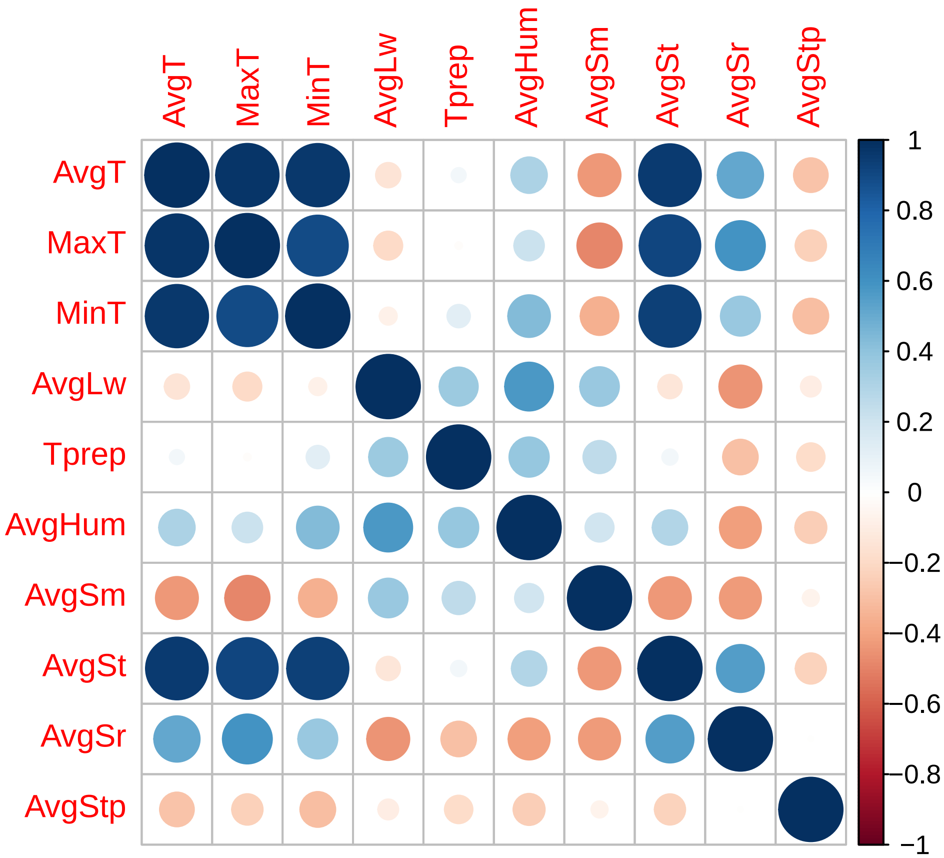
1. Remove heavily correlated columns! - Min Temp and Max Temp for a certain day will correlate with one another!
2. Center Data

Observe:

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrplot(cor(cardinal[,-c(1,12)]))
```



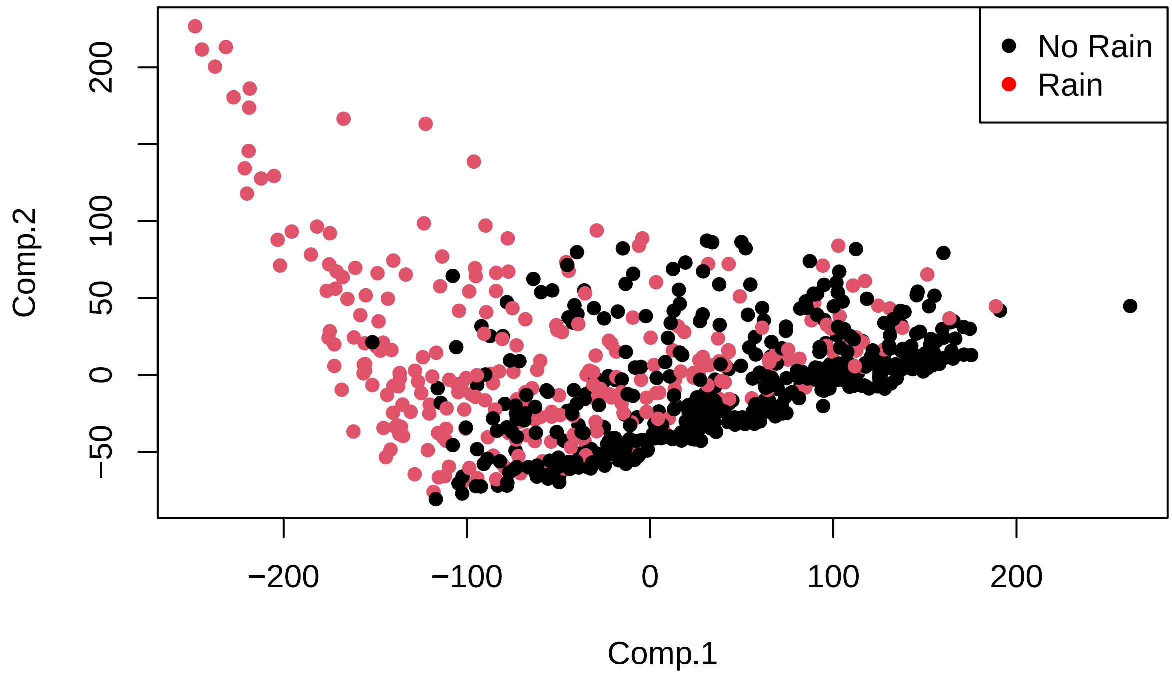
- Tells us to remove all but one temperature variable

```
IfRainVar<- cardinal$IfRain
cardshort <- cardinal%>%select(-c(date,IfRain,Tprep,MinT,MaxT))
cardshort
```

```
## # A tibble: 729 x 7
##     AvgT AvgLw AvgHum AvgSm AvgSt AvgSr AvgStp
##    <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>  <dbl>
##  1  43.1  266.   63.8  0.28  48.6 135.    999.
##  2  44.9  274.   72.0  0.28  47.6  66.0  1003.
##  3  52.8  362.   92.1  0.29  51    31.1   998.
##  4  57.2  373    83.5  0.35  54.6  44.9   993.
##  5  42.1  265.   57.0  0.33  48.3 135.   1005.
##  6  44.1  265.   57.6  0.31  46.1 138.   1005.
##  7  41.4  274.   75.2  0.3   44.6  40.9  1002.
##  8  42.5  314.   58.9  0.3   43.3 136.   1010.
##  9  40.4  265.   60.2  0.3   43.3 122.   1022.
## 10  52    266.   73.5  0.29  46.1  74.6  1019.
## # ... with 719 more rows
```

```
pca_card<- princomp(scale(cardshort,scale=FALSE),cor = FALSE)
```

```
plot(pca_card$scores, pch = 16, col =IfRainVar)
legend("topright",c("No Rain","Rain"),pch=16,col=c("black","red"))
```

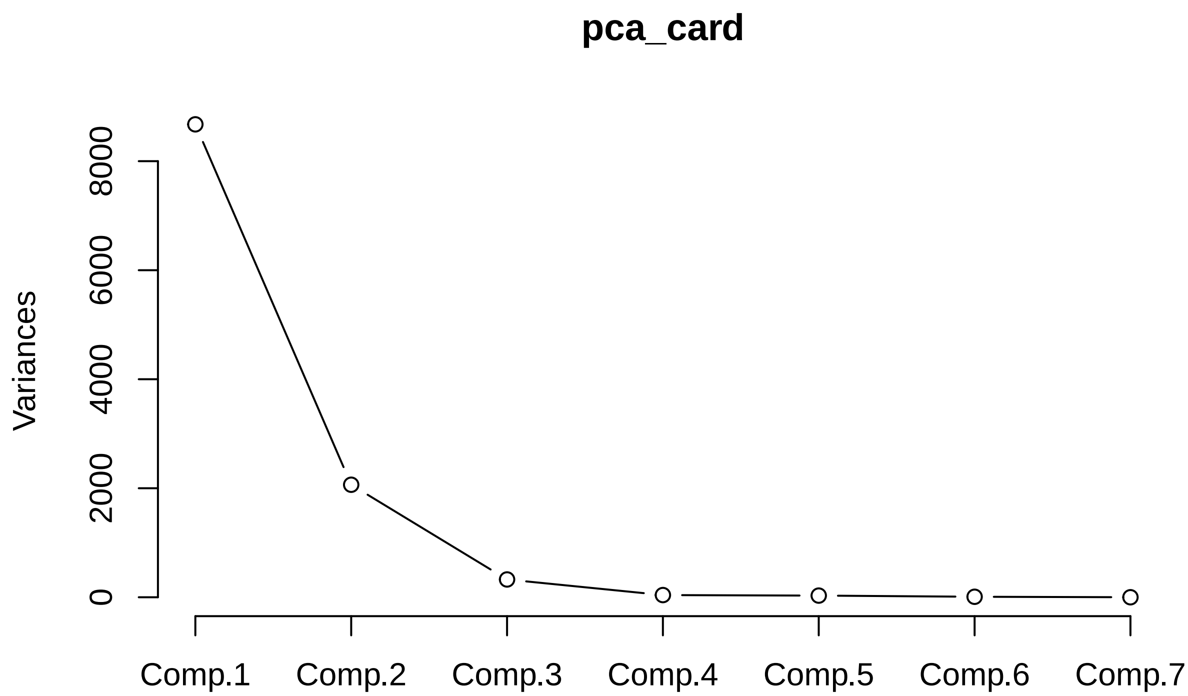


- Here we can look at how good PCA does at describing changes in data
- We see 2 components describes 96% of the data's variation ! (This is very good)

```
summary(pca_card)
```

```
## Importance of components:
##                            Comp.1     Comp.2      Comp.3     Comp.4      Comp.5
## Standard deviation     93.1434884 45.4290622 18.05966455 6.31468558 5.494486034
## Proportion of Variance  0.7784786  0.1851865  0.02926584 0.00357804 0.002708918
## Cumulative Proportion   0.7784786  0.9636651  0.99293094 0.99650898 0.999217895
##                              Comp.6       Comp.7
## Standard deviation     2.9520630681 3.804718e-02
## Proportion of Variance 0.0007819752 1.298933e-07
## Cumulative Proportion  0.9999998701 1.000000e+00
```

```
screeplot(pca_card, type = "lines")
```



How are the original variables related to the principal components?

- Does not print small values, less impactful to correlation

```
loadings(pca_card)
```

```
##
## Loadings:
##        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## AvgT                  0.622  0.369         0.684
## AvgLw  -0.329  0.933 -0.117
## AvgHum         0.115  0.489 -0.861
## AvgSm                                             1.000
## AvgSt                 0.582  0.318  0.182 -0.717
## AvgSr   0.936  0.325 -0.103
## AvgStp               -0.102         0.983  0.131
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var  0.143  0.143  0.143  0.143  0.143  0.143  0.143
## Cumulative Var  0.143  0.286  0.429  0.571  0.714  0.857  1.000
```

- The loading are simple correlations between the principal components and the original variables (Pearson's r).

- Values closest to 1 (positive) or -1 (negative) will represent the strongest relationships, with zero being uncorrelated.

We see in PC 1 that there is a high positive correlation between AvgSr. We see the correlation between solar radiation and the component direction is quite high. So by looking at the second component or the y-axis of our previous plot: we see for the most part, Leaf wetness correlated well with the occurance of rain.

- Another visual to observe the impact of each variable on the principal component!
- Not super pretty here

```
biplot(pca_card)
```

```
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
```