UNIT1_Assignment

Maggie Sha

2020/10/17

Assignment Goal

This assignment aims to let us practice what we have learnt in Unit 1. Using part of the data, exploring the data and visualizing. There is no strict requirement or exact question waiting me to solve, I will try to do some exploration and show the results that I found.

Acquire and read the data

These data were collected from the USDA database selector: https://quickstats.nass.usda.gov

The data were stored online and then downloaded as a CSV file.

```
## read the data

ag_data <- read_csv("berries.csv", col_names = TRUE)
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   Year = col_double(),
##   `Week Ending` = col_logical(),
##   `Ag District` = col_logical(),
##   `Ag District Code` = col_logical(),
##   County = col_logical(),
##   `County ANSI` = col_logical(),
##   `Zip Code` = col_logical(),
##   Region = col_logical(),
##   Watershed = col_logical(),
##   `CV (%)` = col_logical()
## )
```

```
## See spec(...) for full column specifications.
```

```
## look at number of unique values in each column
ag_data %>% summarize_all(n_distinct) -> aa


## make a list of the columns with only one unique value
bb <- which(aa[1,]==1)
```

```
## list the 1-unique value column names
cn <- colnames(ag_data)[bb]
```

Data selected from the NASS database often has columns without any data or with a single repeated Values. The berries data had only 8 out of 21 columns containing meaningful data.

```
## remove the 1-unique columns from the dataset
ag_data %<>% select(-all_of(bb))

aa %<>% select(-all_of(bb))


## State name and the State ANSI code are (sort of) redundant
## Just keep the name
ag_data %<>% select(-4)
aa %<>% select(-4)


kable(head(ag_data)) %>%
  kable_styling(font_size=12)
```

Year	Period	State	Commodity	Data Item
2019	MARKETING YEAR	CALIFORNIA	BLUEBERRIES	BLUEBERRIES, TAME - PRICE R
2019	MARKETING YEAR	CALIFORNIA	BLUEBERRIES	BLUEBERRIES, TAME, FRESH M
2019	MARKETING YEAR	CALIFORNIA	BLUEBERRIES	BLUEBERRIES, TAME, PROCESS
2019	MARKETING YEAR	CALIFORNIA	RASPBERRIES	RASPBERRIES - PRICE RECEIVE
2019	MARKETING YEAR	CALIFORNIA	RASPBERRIES	RASPBERRIES, FRESH MARKET
2019	MARKETING YEAR	CALIFORNIA	RASPBERRIES	RASPBERRIES, PROCESSING - P

```
berry <- unique(ag_data$Commodity)
nberry <- length(berry)
```

This table contains informaton about 3 berries: blueberries, raspberries, and strawberries.

Here I only use Strawberries for the data cleaning and organization. Only the "YEAR" time periond will be considered.

Data cleaning and organization - Strawberries

```
straw <- ag_data %>% filter((Commodity=="STRAWBERRIES") & (Period=="YEAR"))
straw %<>% select(-c(Period, Commodity))

#### Does every Data Item begin with "
sum(str_detect(straw$`Data Item`, "^STRAWBERRIES, ")) == length(straw$`Data Item`)

for (i in 1:5){
  straw[i,]['Data Item'] <- str_replace(straw[i,]['Data Item'], "STRAWBERRIES", "STRAWBERRIES, TAME")
}
```

```
straw %<>% separate(`Data Item`, c("S","type", "meas", "what"), sep = ",")
straw %<>% select(-S)

 head(straw$type, n=20)
 ty <- str_split(straw$type, " ", simplify=TRUE)
head(ty, n=20)

straw %<>% separate(type,c("b1", "type", "b2", "lab1", "lab2","lab3"), " ")

straw %<>% select(-c(b1,b2,lab3))

straw[is.na(straw)] <- " "  ## now Data Item has been split into parts
```

```
# straw$Domain %>% unique()

straw %<>% separate(Domain, c("D_left", "D_right"), sep = ", ")

# straw$D_left %>% unique()
# straw$D_right %>% unique()

straw[is.na(straw)] <- " "

## And now Domain Category

## straw$`Domain Category` %>% unique()

straw %<>% separate(`Domain Category`, c("DC_left", "DC_right"), sep = ", ")

## looks like DC_left combines labels

head(straw$DC_left %>% unique(),n=20)
head(straw$DC_right %>% unique(), n=20)


## work on DC_left first

straw %<>% separate(DC_left, c("DC_left_l", "DC_left_r"), sep = ": ")

## straw$DC_left_l %>% unique()
## straw$DC_left_r %>% unique()

## now work on DC_right

head(straw$DC_right %>% unique(), n=20)

straw %<>% separate(DC_right, c("DC_right_l", "DC_right_r"), sep = ": ")


straw[is.na(straw)] <- " "

## now we need to eliminate the redundancy
```

```r
## remove redundant columns

## paste(straw$D_left,straw$DC_left_l) %>% unique
## returns -- "TOTAL NOT SPECIFIED"   "CHEMICAL CHEMICAL"     "FERTILIZER FERTILIZER"
##            "CHEMICAL CHEMICAL, INSECTICIDE: (CYFLUMETOFEN = 138831)"

## remove column straw$DC_left_l
straw %<>% select(-DC_left_l)

## test

# sum(straw$D_right == straw$DC_right_l)
# [1] 3200
# > straw$DC_left_r %>% unique()
# [1] " "           "(NITROGEN)"  "(PHOSPHATE)" "(POTASH)"    "(SULFUR)"

## remove column DC_right_l
straw %<>% select(-DC_right_l)

## Test for lab1, lab2

# paste(straw$lab1, straw$lab2) %>% unique()
# [1] "ACRES HARVESTED" "ACRES PLANTED"   "PRODUCTION  "    "YIELD  "         "APPLICATIONS  "  "TREATE
# [9] "$  "             "CWT  "           "CWT /"

straw %<>% mutate(label = paste(lab1,lab2))

## test for necisity of "chemical" in col D_left

# paste(straw$D_left, straw$D_right) %>% unique()
# [1] "TOTAL  "             "CHEMICAL FUNGICIDE"   "CHEMICAL HERBICIDE"   "CHEMICAL INSECTICIDE"
# [5] "CHEMICAL OTHER"      "FERTILIZER  "


## remove "Chemical" and joint the columns
straw %<>% mutate(D_left = "CHEMICAL", D_left = "")

straw %<>% mutate(Chemical=paste(D_left, D_right))

straw %<>% select(-c(D_left, D_right))

straw %<>% select(Year, State, type, what, meas, label, DC_left_r, DC_right_r, Chemical, Value)
```

```r
### Now the problem is that we have entries in both the "what" and "meas" columns
##  that begin  "MEASURED IN"
##  how many are there

## in the column "what"
cnt_1 <- str_detect(straw$what, "MEASURED IN")
sum(cnt_1)

## in the column "meas"
cnt_2 <- str_detect(straw$meas, "MEASURED IN")
```

```r
sum(cnt_2)

## We want to put them all in the same column
## So, we will separate them from their current column and put them into
## two columns -- then we will test to make sure there aren't any overlaps
## and then merge the two columns

## we're going to use PURRR.  We need a simple function that takes a logical
## variable and a second variable.  It returns the second variable if the logical
## variable is true and returns a blank if it is false


f1 <- function(a,b){
  if(a){
    return(b)
  }else{
      return("")
  }
}


## now let's separate the "MEASURED IN" entries in the meas column
## form an index of the entries to be separated out

index_meas <- str_detect(straw$meas, "MEASURED IN")

## verify the first six values against the dats straw
head(index_meas)
new <- map2(index_meas, straw$meas, f1)
new <- unlist(new)
head(new, n=20)


straw %<>% mutate(m_in_1 = unlist(map2(index_meas, straw$meas, f1)))


straw %<>% mutate(meas = str_replace(straw$meas, "MEASURED IN.*$", ""))

## Check
cnt_3 <- str_detect(straw$meas, "MEASURED IN")
sum(cnt_3)


##########################
## Now we will do the same thing with the
## "what" column

### index of cells to be isolated
index_what <- str_detect(straw$what, "MEASURED IN")
sum(index_what)

### create a column of the isolated cells
straw %<>% mutate(m_in_2 = unlist(map2(index_what, straw$what, f1)))
```

```
###  eliminate the isolated cells from the original column
straw %<>% mutate(what = str_replace(straw$what, "MEASURED IN.*$", ""))

### test that there are no more "MEASURED IN" cells in the original column
cnt_what <- str_detect(straw$what, "MEASURED IN")
sum(cnt_what)

### Check for overlaps
straw %<>% mutate(units = str_trim(paste(m_in_1, m_in_2)))
straw$units %>% unique()
```

```
## now let's clean it up

straw$what %>% unique()  ## rename Avg
```

```
## [1] " "    " AVG"
```

```
straw$meas %>% unique()  ## rename marketing
```

```
## [1] " "                      " UTILIZED - PRODUCTION"
```

```
straw$label %>% unique() ## rename harvest
```

```
##  [1] "ACRES HARVESTED" "ACRES PLANTED"   "PRODUCTION  "    "YIELD  "
##  [5] "APPLICATIONS  "  "TREATED  "       "- PRODUCTION"    "   "
##  [9] "$  "             "CWT  "           "CWT /"
```

```
straw$DC_left_r %>% unique() # rename chemical_family
```

```
## [1] " "           "(NITROGEN)"  "(PHOSPHATE)" "(POTASH)"    "(SULFUR)"
```

```
tmp <- straw$DC_right_r %>% unique() # rename materials

tmp <- straw$Value %>% unique() # values

tmp <- straw$units %>% unique() # Measures


straw %<>% rename(Avg = what)
straw %<>% rename(Marketing = meas, Harvest = label, Chem_family = DC_left_r, Materials = DC_right_r, M

colnames(straw)
```

```
##  [1] "Year"        "State"       "type"        "Avg"         "Marketing"
##  [6] "Harvest"     "Chem_family" "Materials"   "Chemical"    "Value"
## [11] "m_in_1"      "m_in_2"      "Measures"
```

```
straw %<>% select(Year, State, type, Marketing,
                  Measures, Avg, Harvest, Chem_family,
                  Materials, Chemical, Value )

str_trim(paste(straw$Marketing, straw$Harvest)) %>% unique
```

```
##  [1] "ACRES HARVESTED"       "ACRES PLANTED"         "PRODUCTION"
##  [4] "YIELD"                 "APPLICATIONS"          "TREATED"
##  [7] "- PRODUCTION"          "UTILIZED - PRODUCTION" ""
## [10] "$"                     "CWT"                   "CWT /"
```

```
###  these belong in one column

straw %<>% mutate(production = str_trim(paste(Marketing, Harvest)))

straw %<>% select(Year, State, type, production, Measures,
                  Avg, Chem_family, Materials, Chemical, Value)


straw %<>% mutate(Chemical = str_trim(paste(Chem_family, Chemical)))

straw %<>% select(Year, State, type, production, Avg, Measures, Materials, Chemical, Value)
# export the cleaned data as csv
write.csv(straw,"D:\\MA615\\straw.csv", row.names = FALSE)

straw
```

```
## # A tibble: 3,220 x 9
##     Year State  type   production  Avg   Measures   Materials     Chemical Value
##    <dbl> <chr>  <chr>  <chr>       <chr> <chr>      <chr>         <chr>    <chr>
##  1  2019 CALIF~ TAME   ACRES HARV~ " "   ""         " "           ""       35,4~
##  2  2019 CALIF~ TAME   ACRES PLAN~ " "   ""         " "           ""       36,0~
##  3  2019 CALIF~ TAME   PRODUCTION  " "   "MEASURED~ " "           ""       2,22~
##  4  2019 CALIF~ TAME   PRODUCTION  " "   "MEASURED~ " "           ""       20,5~
##  5  2019 CALIF~ TAME   YIELD       " "   "MEASURED~ " "           ""       580
##  6  2019 CALIF~ BEARI~ APPLICATIO~ " "   "MEASURED~ "(AZOXYSTROB~ "FUNGIC~ 5,500
##  7  2019 CALIF~ BEARI~ APPLICATIO~ " "   "MEASURED~ "(BACILLUS A~ "FUNGIC~ (NA)
##  8  2019 CALIF~ BEARI~ APPLICATIO~ " "   "MEASURED~ "(BACILLUS A~ "FUNGIC~ (NA)
##  9  2019 CALIF~ BEARI~ APPLICATIO~ " "   "MEASURED~ "(BACILLUS P~ "FUNGIC~ (NA)
## 10  2019 CALIF~ BEARI~ APPLICATIO~ " "   "MEASURED~ "(BACILLUS S~ "FUNGIC~ (NA)
## # ... with 3,210 more rows
```

```
### let's look at chemicals being applied to food

unfood <- straw %<>% filter(production=="APPLICATIONS")


unfood %<>% filter(Value != "(D)")

unfood %<>% filter(Value !=  "(NA)")

unfood %<>% filter(Measures == "MEASURED IN LB / ACRE / APPLICATION")
```

```
unfood$type %>% unique()                    ## BEARING
unfood$production %>% unique()              ## APPLICATIONS
unfood$Avg %>% unique()                     ## AVG
unfood$Measures %>% unique()                ## MEASURED IN LB / ACRE / APPLICATION
unfood$Materials %>% unique()       ## long list!
unfood$Chemical %>% unique()
write.csv(unfood,"D:\\MA615\\unfood.csv", row.names = FALSE)
```

Visualization

I am only interested in the chemicals being applied to food.

This is how the cleaned dataset looks like:

```
unfood
```

```
## # A tibble: 225 x 9
##     Year State  type   production  Avg   Measures     Materials   Chemical Value
##    <dbl> <chr>  <chr>  <chr>       <chr> <chr>        <chr>       <chr>    <chr>
##  1  2019 CALIF~ BEARI~ APPLICATIO~ " AV~ MEASURED IN~ (AZOXYSTRO~ FUNGICI~ 0.205
##  2  2019 CALIF~ BEARI~ APPLICATIO~ " AV~ MEASURED IN~ (BLAD = 30~ FUNGICI~ 0.482
##  3  2019 CALIF~ BEARI~ APPLICATIO~ " AV~ MEASURED IN~ (BORAX DEC~ FUNGICI~ 0.023
##  4  2019 CALIF~ BEARI~ APPLICATIO~ " AV~ MEASURED IN~ (BOSCALID ~ FUNGICI~ 0.337
##  5  2019 CALIF~ BEARI~ APPLICATIO~ " AV~ MEASURED IN~ (CAPTAN = ~ FUNGICI~ 1.739
##  6  2019 CALIF~ BEARI~ APPLICATIO~ " AV~ MEASURED IN~ (CYFLUFENA~ FUNGICI~ 0.023
##  7  2019 CALIF~ BEARI~ APPLICATIO~ " AV~ MEASURED IN~ (CYPRODINI~ FUNGICI~ 0.32
##  8  2019 CALIF~ BEARI~ APPLICATIO~ " AV~ MEASURED IN~ (DIFENOCON~ FUNGICI~ 0.114
##  9  2019 CALIF~ BEARI~ APPLICATIO~ " AV~ MEASURED IN~ (FENHEXAMI~ FUNGICI~ 0.623
## 10  2019 CALIF~ BEARI~ APPLICATIO~ " AV~ MEASURED IN~ (FLUDIOXON~ FUNGICI~ 0.213
## # ... with 215 more rows
```

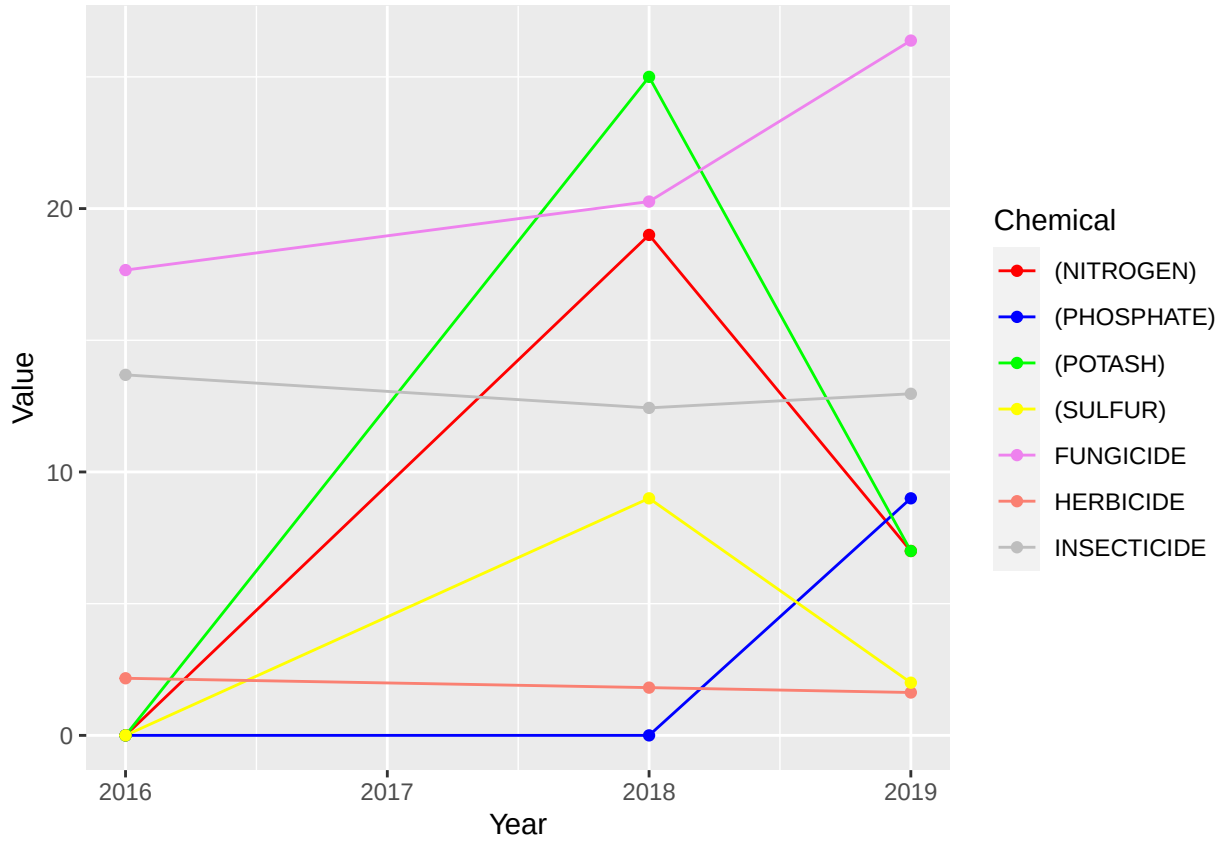
First I want to see how the usage trends of each kind of chemicals look like from 2016 to 2019 regardless of the locations. * I found that there is no useful data in year 2017, so the plot is made only based on year 2016, 2018 and 2019. As you can see, there is no dot in year 2017. * I also found that the value of chemical in 'Other' category is much larger than other chemicals, but we have no idea on what exactly is inside this category, so it is meaningless here and I eliminate this category. * There are many missing data, which are the '0's in the plot. But they basically correspond to year 2016, so the trends from 2018 to 2019 is still meaningful.

```
# define a function to find the values of a specific chemical in a specific year

  f <- function(t,y) {
    funn <- unfood %>% filter(Chemical == t)%>%filter(Year == y)
    sum1 <- sum(as.numeric(funn$Value))
    return (sum1)
  }

# create a dataframe containing seven chemicals' values in 2016,2018 and 2019

for (i in c("FUNGICIDE","HERBICIDE","INSECTICIDE","(NITROGEN)","(PHOSPHATE)","(POTASH)","(SULFUR)")) {
   if (i == "FUNGICIDE"){
     dd <- cbind.data.frame('Chemical' = i, 'Year' = c('2016','2018','2019'),'Value' = c(f(i,2016),f(i,
```

```
   } else{
   dd <- rbind.data.frame(dd,cbind.data.frame('Chemical' = i, 'Year' = c('2016','2018','2019'),'Value' =
   }
}
```

```
## Warning in f(i, 2018):       NA
```

```
dd[is.na(dd)] = 0 # turn NAs to 0

# make a line graph to show the trend
dd$Year <- as.numeric(dd$Year)
ggplot(dd, aes(x = Year, y = Value, color = Chemical)) +
  geom_line() +
  geom_point() +
  scale_color_manual(values = c("red", "blue","green","yellow","violet","salmon","grey"))
```

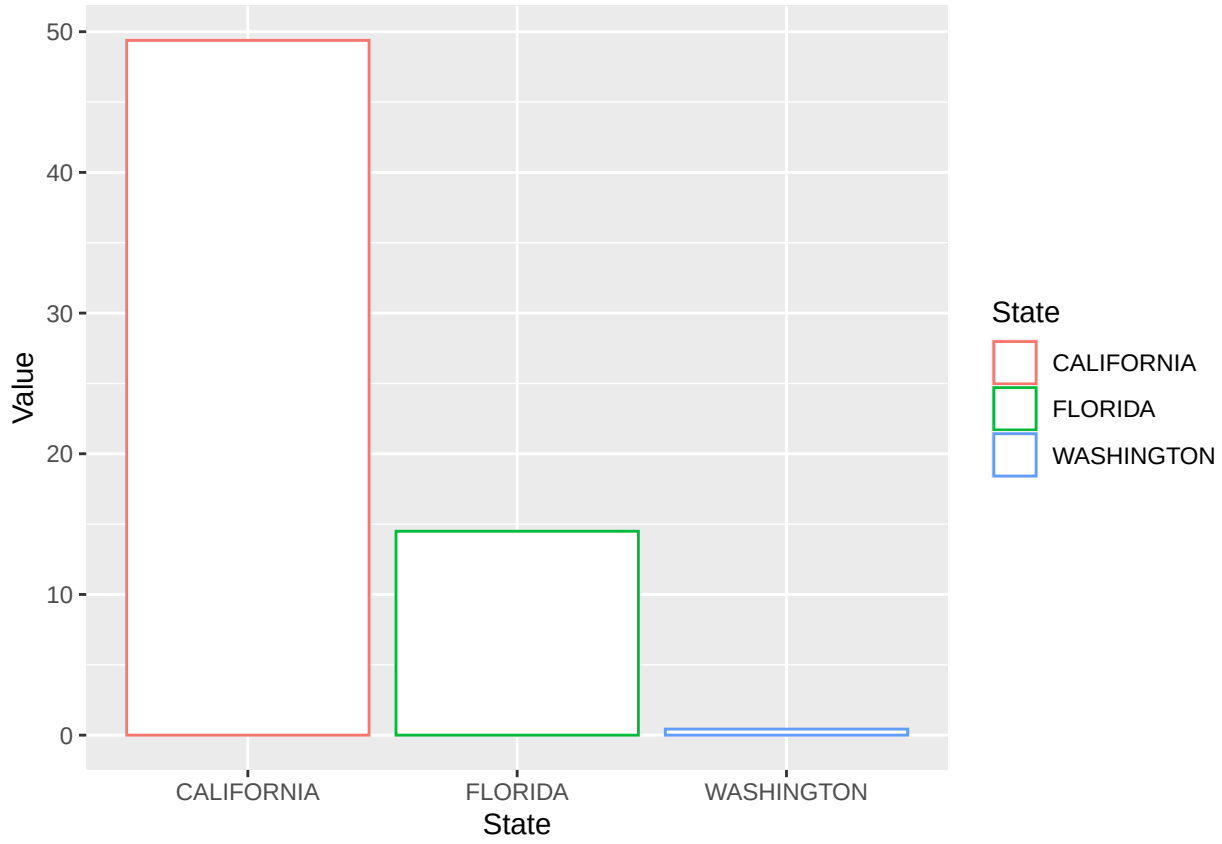


I notice that FUNGICIDE was been used much more than other chemicals in general, and its slope is continuously positive from 2016 to 2019. Now I want to explore in detail that which State used it the most in the 3 years.

As we can see in the barplot, California State used this chemical the most. Washington State used it the least.

```
# FUNGICIDE is been used much more than the other two kinds
# Let's see which State used it the most in 3 years
```

```
Cali <- unfood%>%filter(State == 'CALIFORNIA')%>%filter(Chemical == 'FUNGICIDE')
cal_sum <- sum(as.numeric(Cali$Value))

Flor <- unfood%>%filter(State == 'FLORIDA')%>%filter(Chemical == 'FUNGICIDE')
flo_sum <- sum(as.numeric(Flor$Value))

Washt <- unfood%>%filter(State == 'WASHINGTON')%>%filter(Chemical == 'FUNGICIDE')
wa_sum <- sum(as.numeric(Washt$Value))
df <- data.frame('State' = c('CALIFORNIA','FLORIDA','WASHINGTON'),'Value' = c(cal_sum,flo_sum,wa_sum))
ggplot(df, aes(x=State, y=Value, color=State)) +
  geom_bar(stat="identity", fill="white")
```



Next I want to find the correlations of the chemicals by Principal component analysis(PCA). The correlation is indicated by the angles, the larger the angle is, the smaller the correlation is.

It turns out that FUNGICIDE has little correlation with other chemicals, NiTROGEN has large correlation with POTASH, etc.

```
# define functions to find the value of a specific chemical in each State
ca <- function(i){
  dt <- unfood%>%filter(State == 'CALIFORNIA') %>% filter(Chemical == i)
  sum <- sum(as.numeric(gsub(",", "", dt$Value)))
  return (sum)
}
fl <- function(i){
  dt <- unfood%>%filter(State == 'FLORIDA') %>% filter(Chemical == i)
```

```
  sum <- sum(as.numeric(gsub(",", "", dt$Value)))
  return (sum)
}
wa <-wa <- function(i){
  dt <- unfood%>%filter(State == 'WASHINGTON') %>% filter(Chemical == i)
  sum <- sum(as.numeric(gsub(",", "", dt$Value)))
  return (sum)
}

# create dataframe containing the values of chemicals in different States

for (i in c("FUNGICIDE","HERBICIDE","INSECTICIDE","OTHER","(NITROGEN)","(PHOSPHATE)","(POTASH)","(SULFU
   if (i == "FUNGICIDE"){
     cali <- rbind.data.frame('Chemical' = i, 'Value' = ca(i),fl(i),wa(i))
   } else{
   cali <- cbind.data.frame(cali,rbind.data.frame('Chemical' = i, 'Value' = ca(i),fl(i),wa(i)))
   }
}
cali[is.na(cali)] = 0 # make NAs to be 0

# turn the first row to be titles
names(cali) <- as.matrix(cali[1,])
cali <- cali[-1,]
cali[] <- lapply(cali,function(x) type.convert(as.character(x)))

pca_result <- prcomp(cali, scale = T)
names(pca_result)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

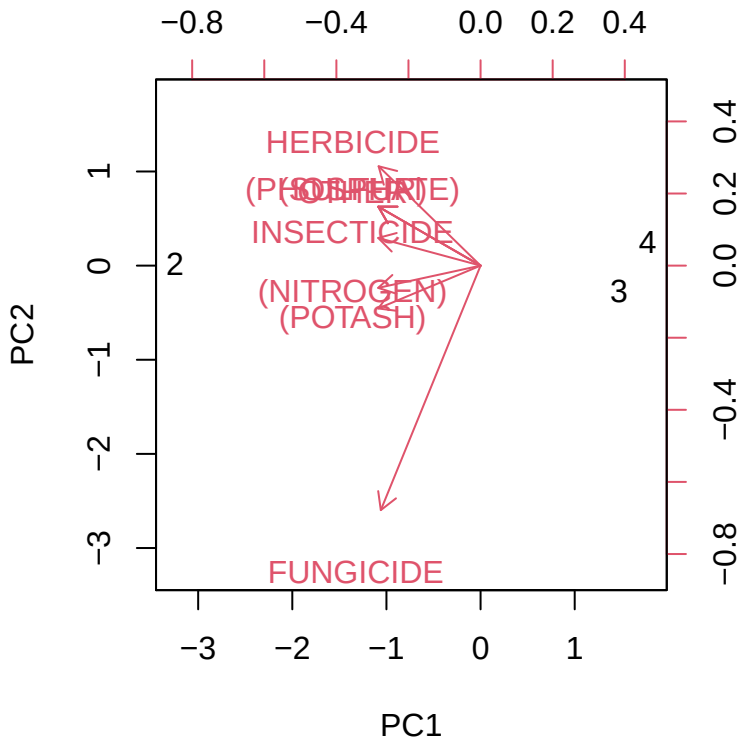
```
# means
kable(t(pca_result$center))
```

FUNGICIDE	HERBICIDE	INSECTICIDE	OTHER	(NITROGEN)	(PHOSPHATE)	(POTASH)	(SULFUR)
21.43633	1.869667	13.02967	482.2683	8.666667	7.666667	10.66667	3.66666

```
# standard deviations
kable(t(pca_result$scale))
```

FUNGICIDE	HERBICIDE	INSECTICIDE	OTHER	(NITROGEN)	(PHOSPHATE)	(POTASH)	(SULFUR)
25.20547	3.04304	21.49886	835.3133	13.31666	13.27906	15.94783	6.35085

```
biplot(pca_result, scale = 0)
```



References

```
citation("knitr")
```

```
##
## To cite the 'knitr' package in publications use:
##
##   Yihui Xie (2020). knitr: A General-Purpose Package for Dynamic Report
##   Generation in R. R package version 1.29.
##
##   Yihui Xie (2015) Dynamic Documents with R and knitr. 2nd edition.
##   Chapman and Hall/CRC. ISBN 978-1498716963
##
##   Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible
##   Research in R. In Victoria Stodden, Friedrich Leisch and Roger D.
##   Peng, editors, Implementing Reproducible Computational Research.
##   Chapman and Hall/CRC. ISBN 978-1466561595
##
## To see these entries in BibTeX format, use 'print(<citation>,
## bibtex=TRUE)', 'toBibtex(.)', or set
## 'options(citation.bibtex.max=999)'.
```

```r
citation("magrittr")
```

```
##
## To cite package 'magrittr' in publications use:
##
##   Stefan Milton Bache and Hadley Wickham (2014). magrittr: A
##   Forward-Pipe Operator for R. R package version 1.5.
##   https://CRAN.R-project.org/package=magrittr
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {magrittr: A Forward-Pipe Operator for R},
##     author = {Stefan Milton Bache and Hadley Wickham},
##     year = {2014},
##     note = {R package version 1.5},
##     url = {https://CRAN.R-project.org/package=magrittr},
##   }
##
## ATTENTION: This citation information has been auto-generated from the
## package DESCRIPTION file and may need manual editing, see
## 'help("citation")'.
```

```r
citation("kableExtra")
```

```
##
## To cite package 'kableExtra' in publications use:
##
##   Hao Zhu (2020). kableExtra: Construct Complex Table with 'kable' and
##   Pipe Syntax. R package version 1.2.1.
##   https://CRAN.R-project.org/package=kableExtra
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {kableExtra: Construct Complex Table with 'kable' and Pipe Syntax},
##     author = {Hao Zhu},
##     year = {2020},
##     note = {R package version 1.2.1},
##     url = {https://CRAN.R-project.org/package=kableExtra},
##   }
```

```r
citation("citation")
```

```
##
## Dietrich J (2020). _citation: Software Citation Tools_. R package
## version 0.4.1.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {citation: Software Citation Tools},
```

```
##     author = {Jan Philipp Dietrich},
##     year = {2020},
##     note = {R package version 0.4.1},
##   }
```

```
citation("ggplot2")
```

```
##
## To cite ggplot2 in publications, please use:
##
##   H. Wickham. ggplot2: Elegant Graphics for Data Analysis.
##   Springer-Verlag New York, 2016.
##
## A BibTeX entry for LaTeX users is
##
##   @Book{,
##     author = {Hadley Wickham},
##     title = {ggplot2: Elegant Graphics for Data Analysis},
##     publisher = {Springer-Verlag New York},
##     year = {2016},
##     isbn = {978-3-319-24277-4},
##     url = {https://ggplot2.tidyverse.org},
##   }
```

citation(package="tidyverse")

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686