STAT2430: Assignment 4

Liam Cook

2024-03-16

Introduction

Each question is worth 5 marks, for a total of 15. The value will be scaled in Brightspace so that all assignments are worth the same amount.

Goals

The goal of this assignment is to practice PCA and K-means methods for multivariate data visualization.

You will have to read data available from the Tidy Tuesday project. For more on Tidy Tuesday see our lesson on data sources.

You will:

1. first read and explore the data (EDA), inspect/graph/tabulate properties of the dataset (e.g. the number of missing values per column)
2. then apply the various types of analyses (unsupervised) seen in week 9 to a subset of the dataset that does not have any missing value.

Required libraries

Exploratory Data Analysis [5 pts]

(Exploratory Data Analysis is also called EDA, or simply Data Exploration).

For this assignment, you will use crop yield data from Tidy Tuesday 2020-09-01. This dataset lists for many years and countries the yield (tonnes per hectare) for 11 different crops.

Note that there is a column named 'entity'. The value of 'entity' is either a country (e.g. Canada) or a region name (e.g. Europe). This creates messy redundancy. Therefore, I have listed the names of non-country values (regions) in a vector called strangeent.

Let us read the dataset directly from the URL on github (you only need to have an internet connection for this)

```
d=read.csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/refs/heads/main/data/2020/202
names(d)
```

```
##  [1] "entity"                       "code"
##  [3] "year"                         "wheat_tonnes_per_hectare"
##  [5] "rice_tonnes_per_hectare"      "maize_tonnes_per_hectare"
##  [7] "soybeans_tonnes_per_hectare"  "potatoes_tonnes_per_hectare"
##  [9] "beans_tonnes_per_hectare"     "peas_tonnes_per_hectare"
## [11] "cassava_tonnes_per_hectare"   "barley_tonnes_per_hectare"
## [13] "cocoa_beans_tonnes_per_hectare" "bananas_tonnes_per_hectare"
```

You could also read the data like this,but this is overkill. Do NOT use this code!

Now run the following code. It will create a list, named strangeent, of 'unwanted' values of entity.

```r
write.csv(unique(d$entity),file='une.csv')
strangeent=unique(d$entity)[
c(2,6,11,12,14,23,44,64,65,66,74,75,76,77,117,120,126,141,142,152,161,162,163,165,198,202,203,206,207,2
```

You can run this code for yourself (do not suppress eval=FALSE) to check the values of strangeent:

```r
strangeent
```

[Assignment QUESTION]

Apply the function *subset* of *dplyr* to datframe *d* so as to keep only the records of *d* for which the value of column *entity* is NOT in the vector *strangeent*. Save the result to a dataframe called *key_crop_yields*.

[code snippet-1: 1 pt]

```r
# replace NA by your own R code.
key_crop_yields = d %>% filter(!entity %in% strangeent)
```

Run the next code snippet to check for yourself that no entity value in dataframe key_crop_yields is in strangeent. Do not remove eval=FALSE)

```r
sum(key_crop_yields$entity %in%strangeent)
```

Finally run the next piece of code to rename the dataframe as d, for simplicity.

```r
# make a copy of the dataframe under the name d.
d=key_crop_yields
```

From now on, you can use the dataframe *d*. (in later questions we will also build subsets of d).

[Assignment QUESTION]

First, take a look at the data. Make a table that counts the number of observation per geo-political entity (these are stored in column entity).

[code snippet-2: 1 pt]

```r
d %>% count(entity)
```

```
##                               entity  n
## 1                        Afghanistan 58
## 2                            Albania 58
```

```
## 3                              Algeria 58
## 4                       American Samoa 58
## 5                               Angola 58
## 6                  Antigua and Barbuda 58
## 7                            Argentina 58
## 8                              Armenia 27
## 9                            Australia 58
## 10                             Austria 58
## 11                          Azerbaijan 27
## 12                             Bahamas 58
## 13                             Bahrain 58
## 14                          Bangladesh 58
## 15                            Barbados 58
## 16                             Belarus 27
## 17                             Belgium 19
## 18                              Belize 58
## 19                               Benin 58
## 20                             Bermuda 58
## 21                              Bhutan 58
## 22                             Bolivia 58
## 23              Bosnia and Herzegovina 27
## 24                            Botswana 58
## 25                              Brazil 58
## 26              British Virgin Islands 53
## 27                              Brunei 58
## 28                            Bulgaria 58
## 29                        Burkina Faso 58
## 30                             Burundi 58
## 31                            Cambodia 58
## 32                            Cameroon 58
## 33                              Canada 58
## 34                          Cape Verde 58
## 35                           Caribbean 58
## 36                      Cayman Islands 53
## 37            Central African Republic 58
## 38                                Chad 58
## 39                               Chile 58
## 40                               China 58
## 41                            Colombia 58
## 42                             Comoros 58
## 43                               Congo 58
## 44                        Cook Islands 58
## 45                          Costa Rica 58
## 46                       Cote d'Ivoire 58
## 47                             Croatia 27
## 48                                Cuba 58
## 49                              Cyprus 58
## 50                      Czech Republic 26
## 51                      Czechoslovakia 32
## 52        Democratic Republic of Congo 58
## 53                             Denmark 58
## 54                            Djibouti 38
## 55                            Dominica 58
## 56                  Dominican Republic 58
```

```
## 57                      Ecuador 58
## 58                        Egypt 58
## 59                  El Salvador 58
## 60            Equatorial Guinea 58
## 61                      Eritrea 26
## 62                      Estonia 27
## 63                     Ethiopia 26
## 64               Faeroe Islands 58
## 65                         Fiji 58
## 66                      Finland 58
## 67                       France 58
## 68                French Guiana 58
## 69             French Polynesia 58
## 70                        Gabon 58
## 71                       Gambia 58
## 72                      Georgia 27
## 73                      Germany 58
## 74                        Ghana 58
## 75                       Greece 58
## 76                      Grenada 58
## 77                   Guadeloupe 58
## 78                         Guam 58
## 79                    Guatemala 58
## 80                       Guinea 58
## 81                Guinea-Bissau 58
## 82                       Guyana 58
## 83                        Haiti 58
## 84                     Honduras 58
## 85                    Hong Kong 51
## 86                      Hungary 58
## 87                      Iceland 58
## 88                        India 58
## 89                    Indonesia 58
## 90                         Iran 58
## 91                         Iraq 58
## 92                      Ireland 58
## 93                       Israel 58
## 94                        Italy 58
## 95                      Jamaica 58
## 96                        Japan 58
## 97                       Jordan 58
## 98                   Kazakhstan 27
## 99                        Kenya 58
## 100                    Kiribati 58
## 101                      Kuwait 51
## 102                  Kyrgyzstan 27
## 103                        Laos 58
## 104                      Latvia 27
## 105                     Lebanon 58
## 106                     Lesotho 58
## 107                     Liberia 58
## 108                       Libya 58
## 109                   Lithuania 27
## 110                  Luxembourg 19
```

```
## 111                      Macedonia 27
## 112                     Madagascar 58
## 113                         Malawi 58
## 114                       Malaysia 58
## 115                       Maldives 58
## 116                           Mali 58
## 117                          Malta 58
## 118                     Martinique 58
## 119                     Mauritania 58
## 120                      Mauritius 58
## 121                      Melanesia 58
## 122                         Mexico 58
## 123           Micronesia (country) 28
## 124                        Moldova 27
## 125                       Mongolia 58
## 126                     Montenegro 13
## 127                     Montserrat 58
## 128                        Morocco 58
## 129                     Mozambique 58
## 130                        Myanmar 58
## 131                        Namibia 58
## 132                          Nepal 58
## 133                    Netherlands 58
## 134                  New Caledonia 58
## 135                    New Zealand 58
## 136                      Nicaragua 58
## 137                          Niger 58
## 138                        Nigeria 58
## 139                           Niue 58
## 140                    North Korea 58
## 141                         Norway 58
## 142                           Oman 58
## 143  Pacific Islands Trust Territory 30
## 144                       Pakistan 58
## 145                      Palestine 25
## 146                         Panama 58
## 147               Papua New Guinea 58
## 148                       Paraguay 58
## 149                           Peru 58
## 150                    Philippines 58
## 151                         Poland 58
## 152                      Polynesia 58
## 153                       Portugal 58
## 154                    Puerto Rico 58
## 155                          Qatar 48
## 156                        Reunion 58
## 157                        Romania 58
## 158                         Russia 27
## 159                         Rwanda 58
## 160            Saint Kitts and Nevis 54
## 161                    Saint Lucia 58
## 162 Saint Vincent and the Grenadines 58
## 163                          Samoa 58
## 164          Sao Tome and Principe 58
```

```
## 165                 Saudi Arabia 58
## 166                      Senegal 58
## 167                       Serbia 13
## 168        Serbia and Montenegro 14
## 169                   Seychelles 58
## 170                 Sierra Leone 58
## 171                    Singapore 55
## 172                     Slovakia 26
## 173                     Slovenia 27
## 174              Solomon Islands 58
## 175                      Somalia 58
## 176                 South Africa 58
## 177                  South Korea 58
## 178                  South Sudan  7
## 179                        Spain 58
## 180                    Sri Lanka 58
## 181                        Sudan  7
## 182                     Suriname 58
## 183                    Swaziland 58
## 184                       Sweden 58
## 185                  Switzerland 58
## 186                        Syria 58
## 187                       Taiwan 58
## 188                   Tajikistan 27
## 189                     Tanzania 58
## 190                     Thailand 58
## 191                        Timor 58
## 192                         Togo 58
## 193                      Tokelau 58
## 194                        Tonga 58
## 195          Trinidad and Tobago 58
## 196                      Tunisia 58
## 197                       Turkey 58
## 198                 Turkmenistan 27
## 199                         USSR 31
## 200                       Uganda 58
## 201                      Ukraine 27
## 202         United Arab Emirates 44
## 203               United Kingdom 58
## 204                United States 58
## 205                      Uruguay 58
## 206                   Uzbekistan 27
## 207                      Vanuatu 58
## 208                    Venezuela 58
## 209                      Vietnam 58
## 210            Wallis and Futuna 58
## 211                        Yemen 58
## 212                   Yugoslavia 31
## 213                       Zambia 58
## 214                     Zimbabwe 58
```

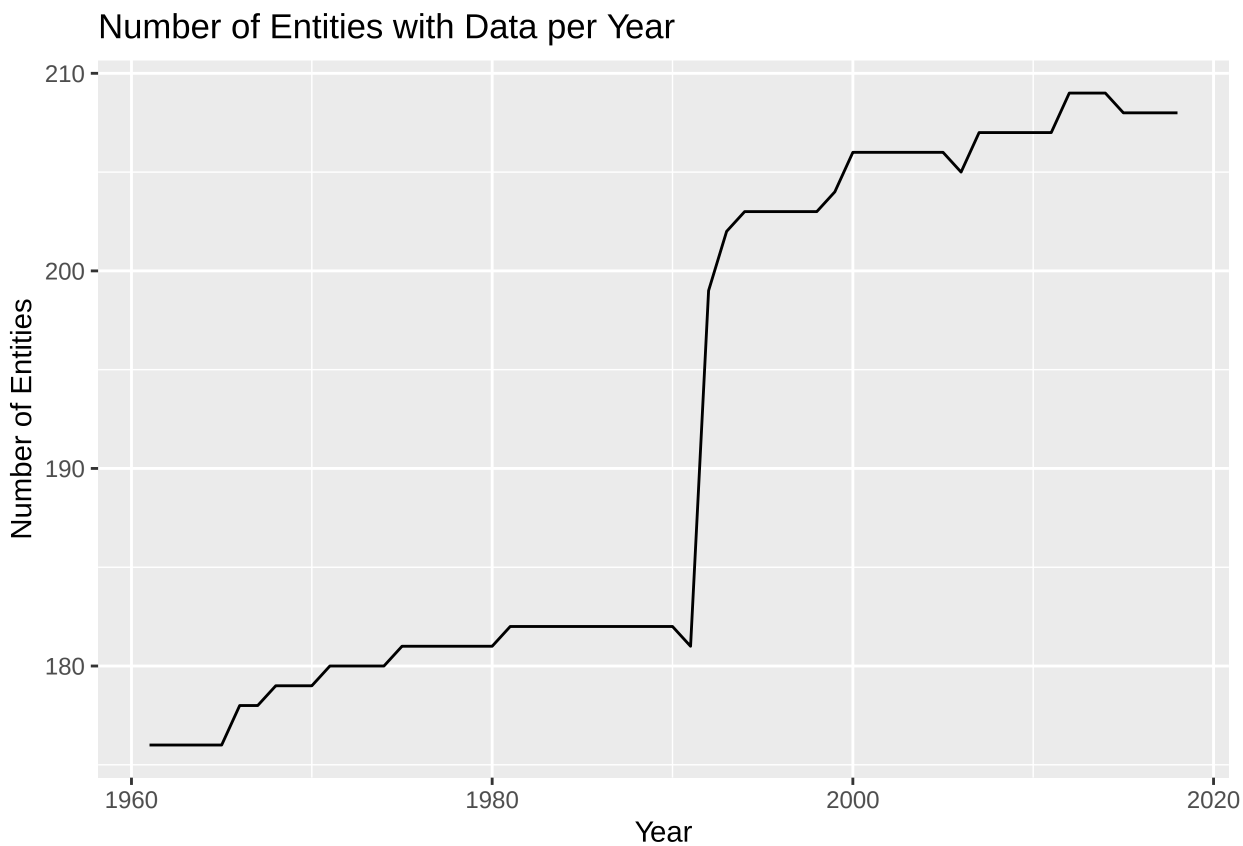
[Assignment QUESTION]

Graph the number of entities with data for each year. (So I want to see the graph of the function n=n(t) where n is the number of entities=countries measured in year t. Note that you will first have to count how many entities are measured each year.).

[code snippet-3: 1 pt]

```r
# Count number of unique entities per year
yearly_counts = d %>% group_by(year) %>% summarise(n = n_distinct(entity))

# Plot the number of entities per year
ggplot(yearly_counts, aes(x = year, y = n)) +
  geom_line() +
  labs(title = "Number of Entities with Data per Year",
       x = "Year",
       y = "Number of Entities")
```



[Assignment QUESTION]

We now want to create a subset of the dataframe with no missing data. Make a table that counts the number of missing data for each of the crop yields and the total number of observations in the table.

[code snippet-4: 0.5 pt]

Hint:

You can create a summary table that sums the NA values in each column. Check how to use the function summarize_all (use the R documentation or look for this online).

```r
# Count missing values per column using summarize_all()
missing_counts = d %>%
```

```
  summarize_all(~ sum(is.na(.)))

# Print the table in an easy-to-read format
print(missing_counts)
```

```
##   entity code year wheat_tonnes_per_hectare rice_tonnes_per_hectare
## 1      0    0    0                     4861                    4420
##   maize_tonnes_per_hectare soybeans_tonnes_per_hectare
## 1                     2230                        6834
##   potatoes_tonnes_per_hectare beans_tonnes_per_hectare peas_tonnes_per_hectare
## 1                        2950                     4957                    6680
##   cassava_tonnes_per_hectare barley_tonnes_per_hectare
## 1                       5147                      6152
##   cocoa_beans_tonnes_per_hectare bananas_tonnes_per_hectare
## 1                           7566                       3873
```

[Assignment QUESTION] **Which 4 crops have the most non-missing observations?**

[text answer: 0.5 pt]

```
Write your answer here.
The 4 crops that have the most NON-MISSING observatons are:
1. Maize
2. Potatoes
3. Bananas
4. Rice
```

Let us now build a new subset of d, named *crop_subset*.

This is done by the next cell of code (run it!) which uses the filter and select functions to pick out a subset of the observations and variables that have no missing data.

I have also simplified the variable names to remove "(tonnes per hectare)" from each name.

```
crop_subset <- d |>
  rename_with(function(x) str_remove(x, "_tonnes_per_hectare")) |>
  filter(year > 1995) |>
  filter(year == 1996 | year == 2015) |>
  select(entity, year, maize, potatoes, bananas, rice, beans) |>
  rowwise() |>
  mutate(na_count = sum(is.na(c_across(-entity))),
         Year_f = factor(year)) |>
  filter(na_count == 0) |>
  select(-na_count)
```

[Keep eval=FALSE. runm this for yourself. No need to run this in the knitted file.]

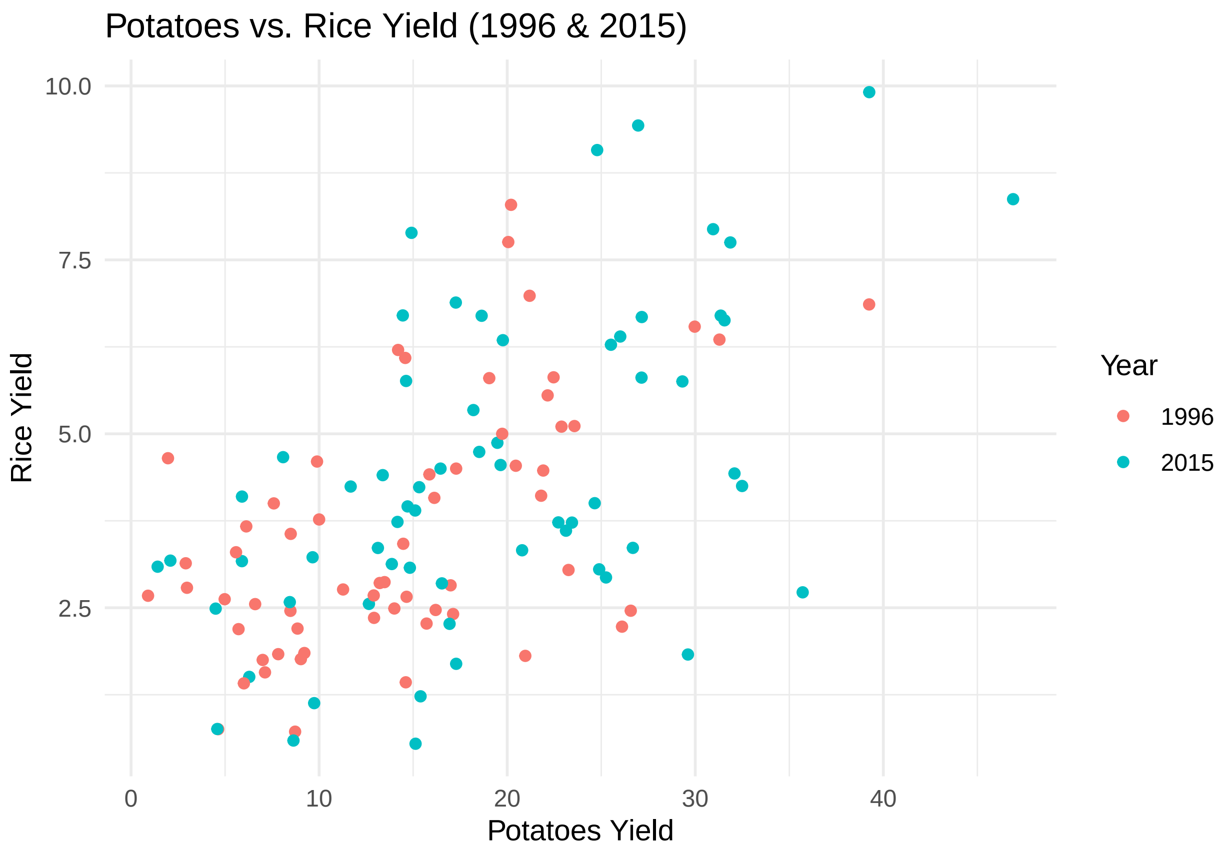
```
str(crop_subset)
head(crop_subset)
```

[Assignment QUESTION]

Make a scatter plot of potatoes (x) and rice (y), and color points by values of Year_f.

[code snippet-5: 1 pt]

```r
ggplot(crop_subset, aes(x = potatoes, y = rice, color = Year_f)) +
  geom_point() +
  labs(title = "Potatoes vs. Rice Yield (1996 & 2015)",
       x = "Potatoes Yield",
       y = "Rice Yield",
       color = "Year") +
  theme_minimal()
```



PCA [5 pts]

[Assignment QUESTION]

Write a R code that performs a PCA on the crop yield data and saves the result to object crop_subset_pca.

Hint:

Before applying PCA, create a new dataframe *crop_subset_q* that only keeps the quantitative columns of *crop_subset*.

Be certain to scale your data (`scale=TRUE`) so they all have the same variance, since the yields are very different for each crop and we don't want the crop with the largest yield (tonnes per hectare) to automatically dominate the analysis and visualization.

When crop_subset_pca is calculated, you can create a summary of this object to display the results.

[code snippet-6: 1 pt]

```r
crop_subset_q = crop_subset %>%
  select(where(is.numeric))

crop_subset_pca = prcomp(crop_subset_q, scale = TRUE)

summary(crop_subset_pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     1.7324 0.9643 0.9488 0.72076 0.63183 0.50008
## Proportion of Variance 0.5002 0.1550 0.1500 0.08658 0.06654 0.04168
## Cumulative Proportion  0.5002 0.6552 0.8052 0.89178 0.95832 1.00000
```

[Assignment QUESTION]

Make a graph that shows a point for each observation, uses colour or shape to distinguish the years of the observations, and includes the arrows for the yields projected on the first two principal components.

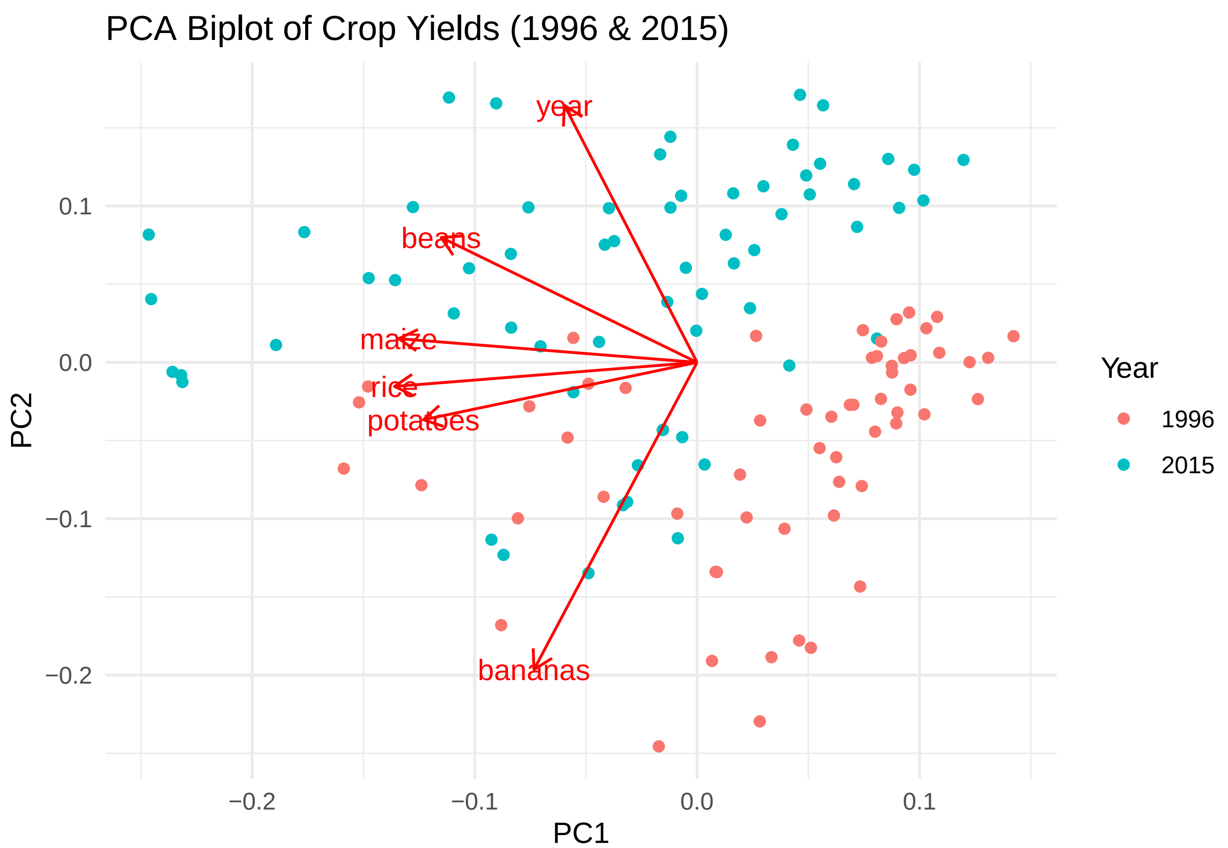
In other words: do a biplot from object crop_subset_pca

Hint:

you can use the autoplot function. Make sure that the loadings and loadings.label arguments are set to TRUE.

[code snippet-7: 1 pt]

```r
autoplot(crop_subset_pca, data = crop_subset, colour = "Year_f",
         loadings = TRUE, loadings.label = TRUE) +
  labs(title = "PCA Biplot of Crop Yields (1996 & 2015)",
       x = "PC1",
       y = "PC2",
       colour = "Year") +
  theme_minimal()
```



[Assignment QUESTION]

Write a short paragraph that summarizes the trend along the first and second principal components.

Hint: explain which variables are more aligned with the PC1 (resp. PC2) axis.

[text answer: 1pt]

```
The first PC1 mainly captures variation in total crop yield, with maize, rice, and potatoes aligning st
That suggests that PC1 is a measure of overall productivity. The second PC2 highlights differences in s
This indicates that PC2 may represent variations in crop specialization or growing conditions in its re
Additionally, the distribution of points for 1996 and 2015 suggests a shift in crop production trends o
```

[Assignment QUESTION]

Make a new PCA using only observations from ONE of the two years.

[code snippet-8: 1 pt]

```
crop_subset_one_year = crop_subset %>%
  filter(year == 2015)

crop_subset_q_one_year = crop_subset_one_year %>%
  select(-entity, -year)


str(crop_subset_q_one_year)
```

```
## rowws_df [63 x 6] (S3: rowwise_df/tbl_df/tbl/data.frame)
##  $ maize   : num [1:63] 3.92 1.12 7.31 8.3 6.98 ...
##  $ potatoes: num [1:63] 29.61 6.29 31.35 39.25 19.65 ...
##  $ bananas : num [1:63] 19.7 28.3 20.9 21.4 16.6 ...
##  $ rice    : num [1:63] 1.83 1.51 6.7 9.91 4.55 ...
##  $ beans   : num [1:63] 0.795 0.513 1.385 0.914 1.516 ...
##  $ Year_f  : Factor w/ 2 levels "1996","2015": 2 2 2 2 2 2 2 2 2 2 ...
##  - attr(*, "groups")= tibble [63 x 1] (S3: tbl_df/tbl/data.frame)
##   ..$ .rows: list<int> [1:63]
##   .. ..$ : int 1
##   .. ..$ : int 2
##   .. ..$ : int 3
##   .. ..$ : int 4
##   .. ..$ : int 5
##   .. ..$ : int 6
##   .. ..$ : int 7
##   .. ..$ : int 8
##   .. ..$ : int 9
##   .. ..$ : int 10
##   .. ..$ : int 11
##   .. ..$ : int 12
##   .. ..$ : int 13
##   .. ..$ : int 14
##   .. ..$ : int 15
##   .. ..$ : int 16
##   .. ..$ : int 17
##   .. ..$ : int 18
##   .. ..$ : int 19
##   .. ..$ : int 20
##   .. ..$ : int 21
##   .. ..$ : int 22
##   .. ..$ : int 23
##   .. ..$ : int 24
##   .. ..$ : int 25
##   .. ..$ : int 26
##   .. ..$ : int 27
##   .. ..$ : int 28
##   .. ..$ : int 29
##   .. ..$ : int 30
##   .. ..$ : int 31
##   .. ..$ : int 32
##   .. ..$ : int 33
##   .. ..$ : int 34
##   .. ..$ : int 35
##   .. ..$ : int 36
##   .. ..$ : int 37
##   .. ..$ : int 38
##   .. ..$ : int 39
##   .. ..$ : int 40
##   .. ..$ : int 41
##   .. ..$ : int 42
##   .. ..$ : int 43
##   .. ..$ : int 44
##   .. ..$ : int 45
```

```
##   .. ..$ : int 46
##   .. ..$ : int 47
##   .. ..$ : int 48
##   .. ..$ : int 49
##   .. ..$ : int 50
##   .. ..$ : int 51
##   .. ..$ : int 52
##   .. ..$ : int 53
##   .. ..$ : int 54
##   .. ..$ : int 55
##   .. ..$ : int 56
##   .. ..$ : int 57
##   .. ..$ : int 58
##   .. ..$ : int 59
##   .. ..$ : int 60
##   .. ..$ : int 61
##   .. ..$ : int 62
##   .. ..$ : int 63
##   .. ..@ ptype: int(0)
```

```r
crop_subset_q_one_year = crop_subset_q_one_year %>%
  mutate(across(everything(), as.numeric))

crop_subset_q_one_year = crop_subset_q_one_year %>%
  select(where(~ var(.) > 0))

crop_subset_pca_one_year = prcomp(crop_subset_q_one_year, scale = TRUE)

summary(crop_subset_pca_one_year)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5
## Standard deviation     1.6779 0.9655 0.7540 0.66076 0.49744
## Proportion of Variance 0.5631 0.1864 0.1137 0.08732 0.04949
## Cumulative Proportion  0.5631 0.7495 0.8632 0.95051 1.00000
```

```r
highest_banana = crop_subset %>%
  group_by(entity) %>%
  summarise(max_banana = max(bananas, na.rm = TRUE)) %>%
  arrange(desc(max_banana)) %>%
  slice(1) %>%
  pull(entity)

highest_beans = crop_subset %>%
  group_by(entity) %>%
  summarise(max_beans = max(beans, na.rm = TRUE)) %>%
  arrange(desc(max_beans)) %>%
  slice(1) %>%
  pull(entity)

low_yield_country = crop_subset %>%
  group_by(entity) %>%
  summarise(mean_yield = mean(c(maize, potatoes, bananas, rice, beans), na.rm = TRUE)) %>%
```

```r
  arrange(mean_yield) %>%
  slice(1) %>%
  pull(entity)

  highest_banana
```

```
## [1] "Indonesia"
```

```r
  highest_beans
```

```
## [1] "Sudan"
```

```r
  low_yield_country
```

```
## [1] "Democratic Republic of Congo"
```

[Assignment QUESTION]

Identify a country in each of the following three categories:

[text answer: 1 pt]

Hint:

You can use group_by and arrange(desc(VAR)) to retrieve the top values of a variable VAR of your choice. VAR can either be one of the original columns, or a new column created with statistical functions (such as mean()). You will need to engineer this for the third XXX value.

```r
highest_banana = crop_subset %>%
  group_by(entity) %>%
  summarise(max_banana = max(bananas, na.rm = TRUE)) %>%
  arrange(desc(max_banana)) %>%
  slice(1) %>%
  pull(entity)

highest_beans = crop_subset %>%
  group_by(entity) %>%
  summarise(max_beans = max(beans, na.rm = TRUE)) %>%
  arrange(desc(max_beans)) %>%
  slice(1) %>%
  pull(entity)

low_yield_country = crop_subset %>%
  group_by(entity) %>%
  summarise(mean_yield = mean(c(maize, potatoes, bananas, rice, beans), na.rm = TRUE)) %>%
  arrange(mean_yield) %>%
  slice(1) %>%
  pull(entity)

  highest_banana
```

```
## [1] "Indonesia"
```

```r
highest_beans
```

```
## [1] "Sudan"
```

```r
low_yield_country
```

```
## [1] "Democratic Republic of Congo"
```

```
Replace XXX by the name of an entity=country
* country with Highest Banana yield: Indonesia
* country with Highest Bean yield: Sudan
* Country with 'low' yields for many of the selected crops: Democratic Republic of Congo
```

K-means clustering [5 pts]

[Assignment QUESTION]

Perform a k-means clustering on dataframe crop_subset to divide records into 2 groups Use all quantitative yields columns in crop_subset. Save your result in object kclust1.

[code snippet-9: 1 pt]

```r
set.seed(123) # do not remove this code. write your own code below.
crop_subset_kmeans = crop_subset %>%
  select(maize, potatoes, bananas, rice, beans)

kclust1 = kmeans(crop_subset_kmeans, centers = 2, nstart = 25)
print(kclust1)
```

```
## K-means clustering with 2 clusters of sizes 81, 42
##
## Cluster means:
##      maize potatoes  bananas     rice     beans
## 1 2.675678 13.92779 11.78739 3.413830 0.9345926
## 2 4.504569 22.58285 38.01793 5.179638 1.1453238
##
## Clustering vector:
##   [1] 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 2 2 1 1 1 1 2 2
##  [38] 1 2 2 2 1 1 1 1 2 2 2 1 1 2 2 1 2 2 2 1 2 2 2 1 1 1 1 1 1 1 2 1 1 1 2 2 2
##  [75] 2 1 1 1 2 2 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 2 2 1
## [112] 2 2 1 1 1 2 1 1 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 9287.426 9087.399
##  (between_SS / total_SS =  53.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
crop_subset$cluster = as.factor(kclust1$cluster)
head(crop_subset)
```

```
## # A tibble: 6 x 9
## # Rowwise:
##   entity     year maize potatoes bananas  rice beans Year_f cluster
##   <chr>     <int> <dbl>    <dbl>   <dbl> <dbl> <dbl> <fct>  <fct>
## 1 Algeria    2015 3.92     29.6     19.7   1.83 0.795 2015   1
## 2 Angola     1996 0.699     7        9.42  1.75 0.354 1996   1
## 3 Angola     2015 1.12      6.29    28.3   1.51 0.513 2015   1
## 4 Argentina  1996 4.04     22.9     14.2   5.10 0.911 1996   1
## 5 Argentina  2015 7.31     31.4     20.9   6.70 1.39  2015   2
## 6 Australia  1996 5.77     31.3     24.7   6.35 0.945 1996   2
```

```r
table(crop_subset$cluster)
```

```
##
##  1  2
## 81 42
```

[Assignment QUESTION]

Show a table of the centers of each of the clusters you formed.

[Meaning show the coordinates of the cluster means vectors, for each cluster]

[code snippet-10: 1 pt]

Hint

Apply tidy and pipe to kable. Keep 2 digits

```r
cluster_centers = as.data.frame(kclust1$centers) %>%
  mutate(Cluster = row_number()) %>%
  relocate(Cluster)

cluster_centers %>%
  mutate(across(where(is.numeric), ~ round(.x, 2))) %>%
  kable()
```

Cluster	maize	potatoes	bananas	rice	beans
1	2.68	13.93	11.79	3.41	0.93
2	4.50	22.58	38.02	5.18	1.15

[Assignment QUESTION]

Make a scatter plot of your data in the (x=potatoes, y=maize) space, colouring each point according to the cluster number.

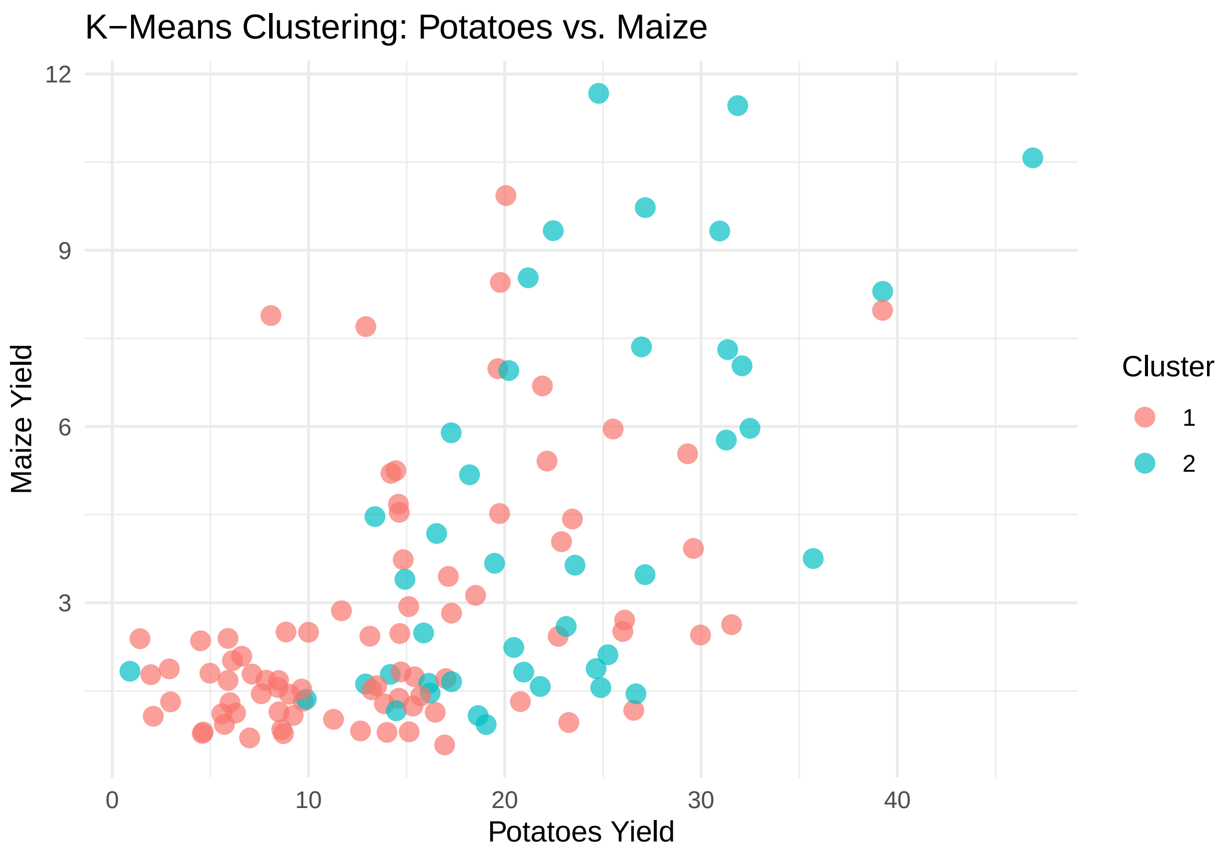
[code snippet-11: 1 pt]

Hint

Use the augment function and mimic the code in the lecture of Week 9.

```r
clustered_data = augment(kclust1, crop_subset)

ggplot(clustered_data, aes(x = potatoes, y = maize, color = .cluster)) +
  geom_point(size = 3, alpha = 0.7) +
  labs(title = "K-Means Clustering: Potatoes vs. Maize",
       x = "Potatoes Yield",
       y = "Maize Yield",
       color = "Cluster") +
  theme_minimal()
```



[Assignment QUESTION]

Make a scatter plot in potatoes,maize space where you show only, in blue, points that have been clustered to cluster number 2. Each point should be shown with its label (=country name=value of the entity column).

Note that the scatter plot should ONLY show points that belong in cluster 2 (in blue)

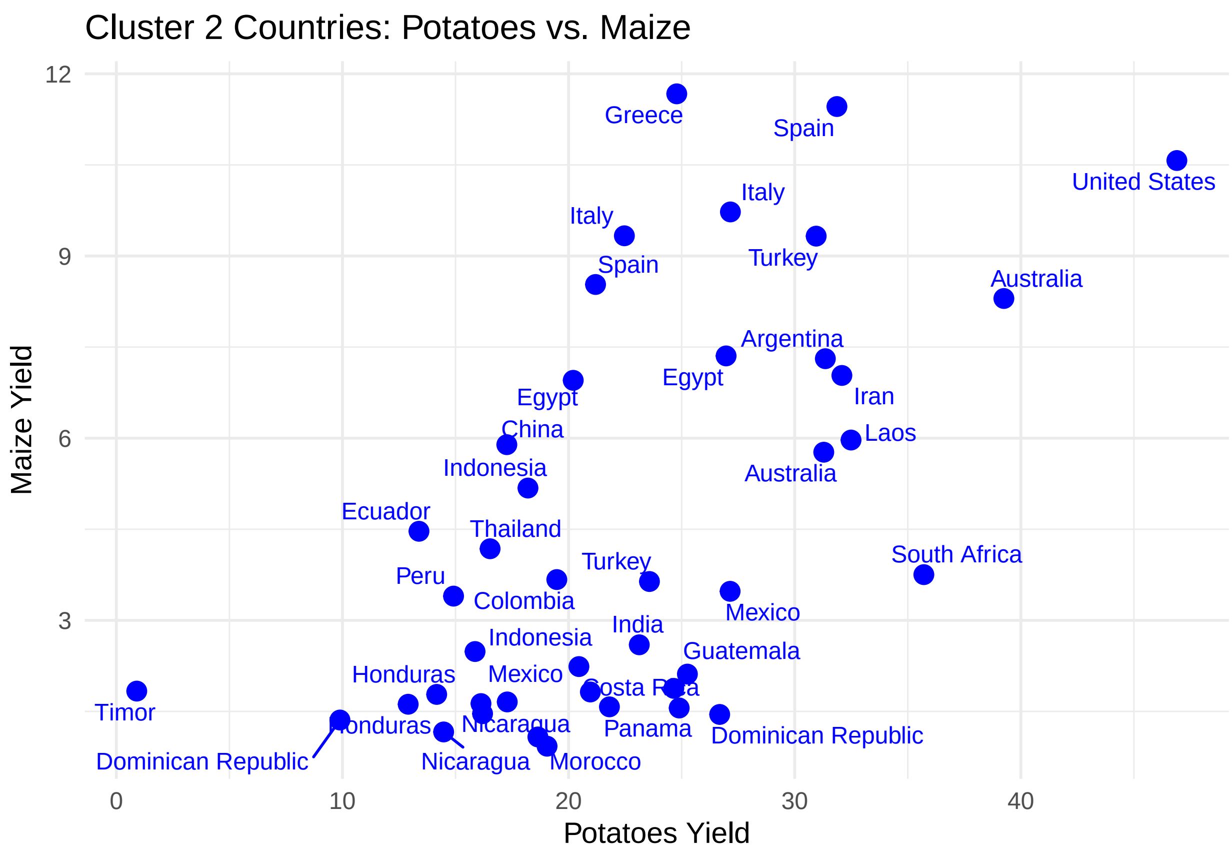
[code snippet-12: 1 pt]

```r
cluster_2_data = crop_subset %>%
  filter(cluster == "2")

ggplot(cluster_2_data, aes(x = potatoes, y = maize, label = entity)) +
  geom_point(color = "blue", size = 3) +
  geom_text_repel(size = 3, color = "blue") +
  labs(title = "Cluster 2 Countries: Potatoes vs. Maize",
       x = "Potatoes Yield",
```

```
      y = "Maize Yield") +
  theme_minimal()
```

```
## Warning: ggrepel: 5 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



[Assignment QUESTION]

Discuss qualitatively the interpretation of each cluster. Explain which kind of country is found in cluster 1 and which kind is found in cluster 2.

[text answer: 1 pt]

Hint

You may have to print the list of countries in each cluster to identify 'a pattern'.

```
cluster_1_countries = crop_subset %>%
  filter(cluster == "1") %>%
  select(entity)

cluster_2_countries = crop_subset %>%
  filter(cluster == "2") %>%
  select(entity)

print("Cluster 1 Countries:")
```

```
## [1] "Cluster 1 Countries:"
```

```r
print(cluster_1_countries)
```

```
## # A tibble: 81 x 1
## # Rowwise:
##    entity
##    <chr>
##  1 Algeria
##  2 Angola
##  3 Angola
##  4 Argentina
##  5 Bangladesh
##  6 Bangladesh
##  7 Belize
##  8 Belize
##  9 Benin
## 10 Benin
## # i 71 more rows
```

```r
print("Cluster 2 Countries:")
```

```
## [1] "Cluster 2 Countries:"
```

```r
print(cluster_2_countries)
```

```
## # A tibble: 42 x 1
## # Rowwise:
##    entity
##    <chr>
##  1 Argentina
##  2 Australia
##  3 Australia
##  4 China
##  5 Colombia
##  6 Colombia
##  7 Costa Rica
##  8 Costa Rica
##  9 Dominican Republic
## 10 Dominican Republic
## # i 32 more rows
```

```
Cluster 1 primarily consists of countries with lower agricultural productivity for bananas and potatoes
These countries tend to have lower overall crop yields across all crops.
This cluster may include regions with less developed agricultural infrastructure.

Cluster 2 includes countries with significantly higher crop yields, particularly for bananas and potatoe
These are likely to be countries with advanced systems, good irrigation and fertilizers, and nicer clima
```