Unsupervised Learning and PCA

Khanin Sisaengsuwanchai

2022-08-18

1.

```
data = read.csv("nci.csv")
dim(data)
```

```
## [1]   64 6831
```

```
# Data Pre-processing
#data$labs = as.factor(data$labs)
table(data[, "labs"])
```

```
##
##      BREAST         CNS       COLON K562A-repro K562B-repro    LEUKEMIA
##           7           5           7           1           1           6
## MCF7A-repro MCF7D-repro    MELANOMA       NSCLC     OVARIAN    PROSTATE
##           1           1           8           9           6           2
##       RENAL     UNKNOWN
##           9           1
```

```
# By default, this function will scale the data to have mean zero and unit variance.
# scaled_data = as.data.frame(scale(data[,-1])) # Remove the fist column
scaled_data = scale(data[,-1])
head(scaled_data[, 1:10]) # Recheck the data with top 10 columns
```

```
##             X1           X2         X3         X4         X5          X6
## [1,] 0.7229554  1.594614647  1.3152906  1.3450554 -0.6001006 -0.21892339
## [2,] 1.5838967  1.739790603  0.4382214  0.6489885  0.9047460  1.63581692
## [3,] 2.1731106 -0.016089747 -0.3463542  0.2643754 -1.3010255 -0.01917014
## [4,] 0.6776381 -0.372557113  1.6153098 -0.4408142  1.2346734 -0.01917014
## [5,] 1.1421409 -0.577195786  0.9575754  1.1298352  0.3585172 -0.03343823
## [6,] 0.7456141 -0.002887252 -0.1848054 -0.1202735 -0.4764080 -1.56012378
##              X7         X8          X9        X10
## [1,]  0.8910931 -0.8619276 -1.05030928 -1.0508663
## [2,]  1.8351898  2.2091606 -0.09510822 -0.4742284
## [3,]  0.1896859  1.9730294  1.00007333  0.7104027
## [4,]  0.4055035  0.7917973  0.04478144  0.1181175
## [5,]  0.1761973  0.3931315  0.54572719  2.2924275
## [6,] -1.7796496 -1.6592592 -1.79590295 -0.4118218
```

```r
#a) Perform hierarchical clustering of the cancer cell samples using complete linkage.
#Plot the resulting dendrogram.

# Find distance
distance = dist(scaled_data)
head(distance)
```

```
## [1]  77.04594  87.30561 103.18322 113.72295 108.29369 110.38386
```

```r
length(distance)
```

```
## [1] 2016
```

```r
distmat = as.matrix(distance)
dim(distmat)
```

```
## [1] 64 64
```

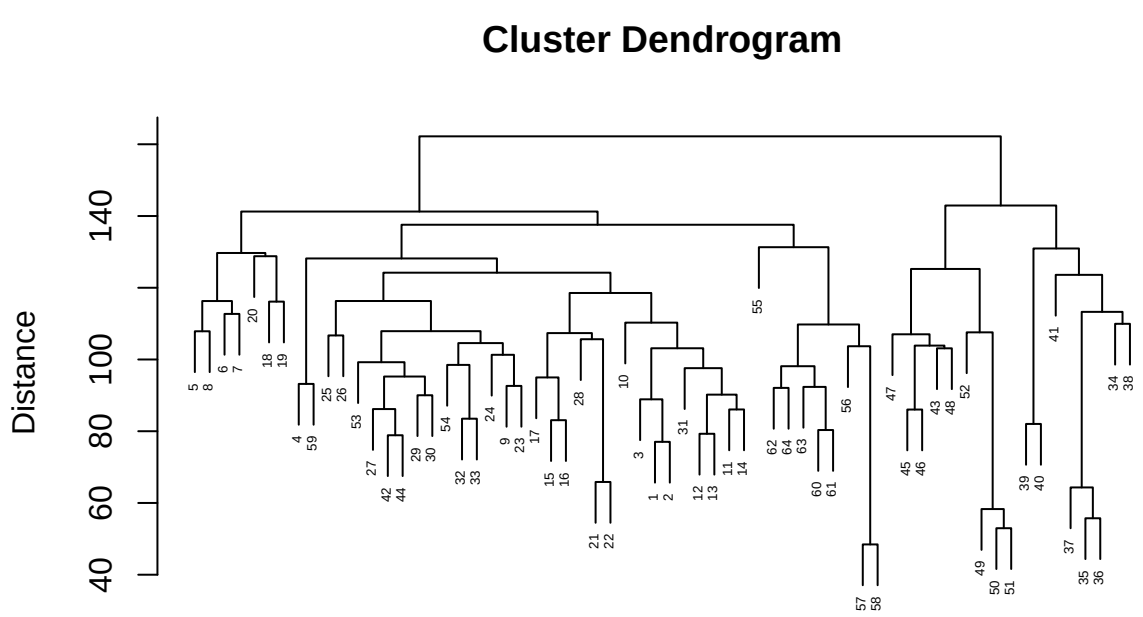
```r
distmat[1:7, 1:7]
```

```
##           1         2         3         4        5        6        7
## 1   0.00000  77.04594  87.30561 103.18322 113.7230 108.2937 110.3839
## 2  77.04594   0.00000  88.89531 106.64318 116.1610 111.7084 108.0612
## 3  87.30561  88.89531   0.00000  95.79984 101.0443 108.0419 113.6874
## 4 103.18322 106.64318  95.79984   0.00000 107.0625 120.0285 125.5473
## 5 113.72295 116.16097 101.04429 107.06253   0.0000 114.3239 113.7342
## 6 108.29369 111.70837 108.04193 120.02845 114.3239   0.0000 112.7059
## 7 110.38386 108.06121 113.68741 125.54730 113.7342 112.7059   0.0000
```

```r
set.seed(1)
# Plot dendrogram
h1 = hclust(distance, method ='complete') # complete linkage
str(h1)
```

```
## List of 7
##  $ merge      : int [1:63, 1:2] -57 -50 -35 -49 -37 -21 -1 -42 -12 -60 ...
##  $ height     : num [1:63] 48.4 53 55.7 58.3 64.3 ...
##  $ order      : int [1:64] 5 8 6 7 20 18 19 4 59 25 ...
##  $ labels     : NULL
##  $ method     : chr "complete"
##  $ call       : language hclust(d = distance, method = "complete")
##  $ dist.method: chr "euclidean"
##  - attr(*, "class")= chr "hclust"
```

```r
plot(h1,cex=0.4,xlab="Rows (leaves)",sub="",ylab="Distance")
```




```
# Note: clusters have not been created yet.
```

```
# b) Group the samples into 4 clusters. Plot the clusters in PC1, PC2 space.

library(factoextra)
```

```
## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
# Cut the dendrograms to 4 clusters
set.seed(1)
cut4clusters = cutree(h1, k=4)
head(cut4clusters)
```

```
## [1] 1 1 1 1 2 2
```

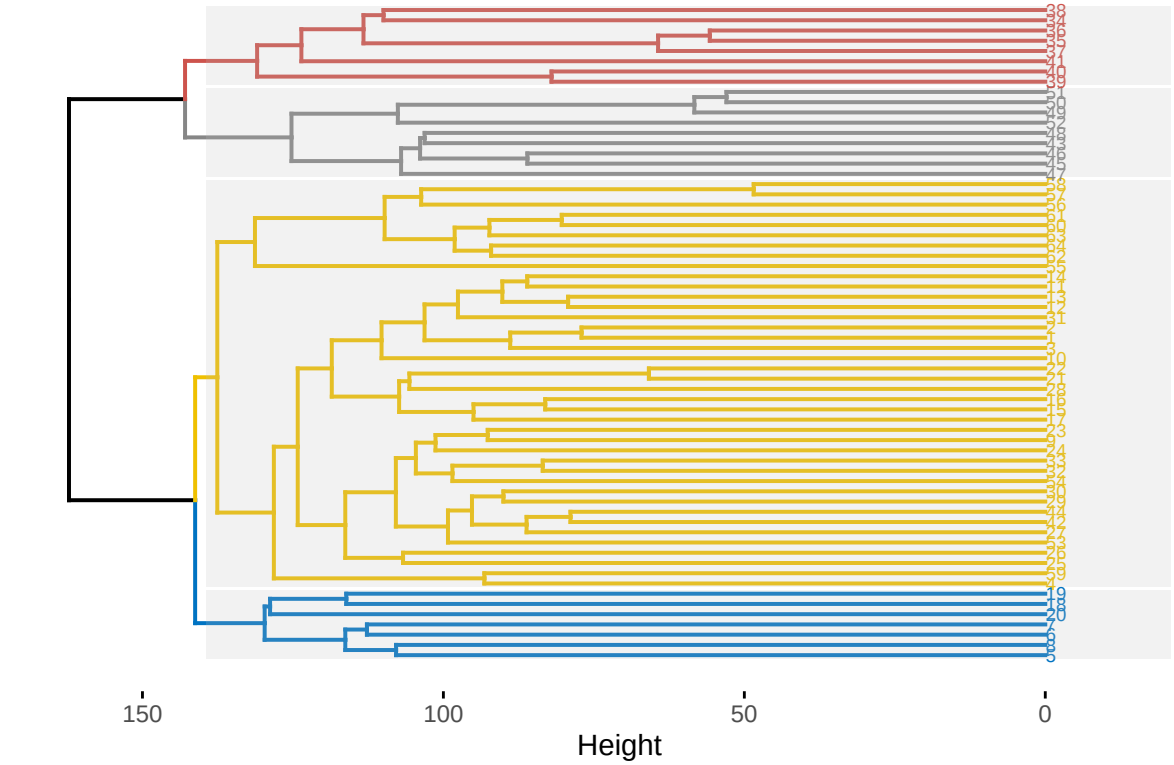
```
data_with_clusters = data.frame(scaled_data,cluster = cut4clusters)

# number of members per cluster
table(cut4clusters)
```

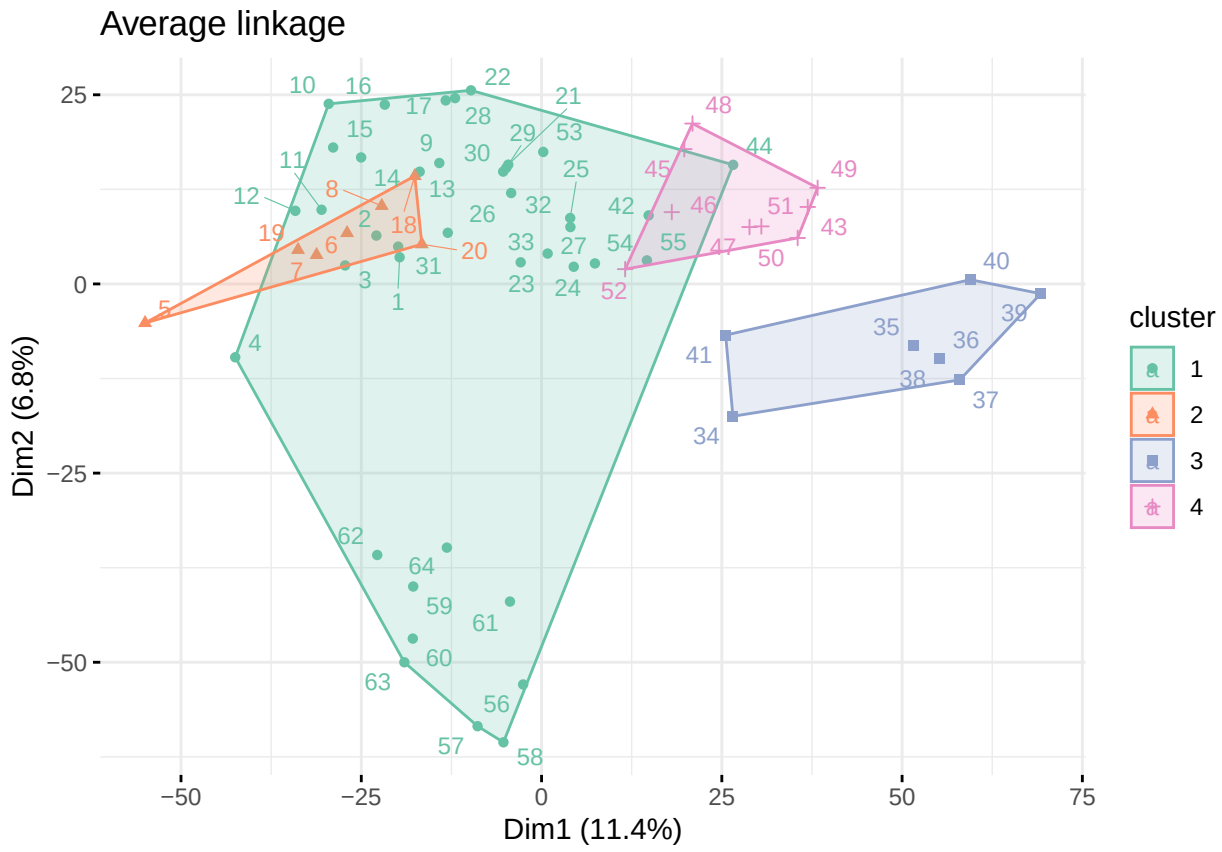
```
## cut4clusters
##  1  2  3  4
## 40  7  8  9
```

```
# plot dendrogram again with k=4
fviz_dend(h1,k=4,cex = 0.5, main="Complete linkage",k_colors = "jco", rect = T,
          rect_fill = T, horiz = T)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



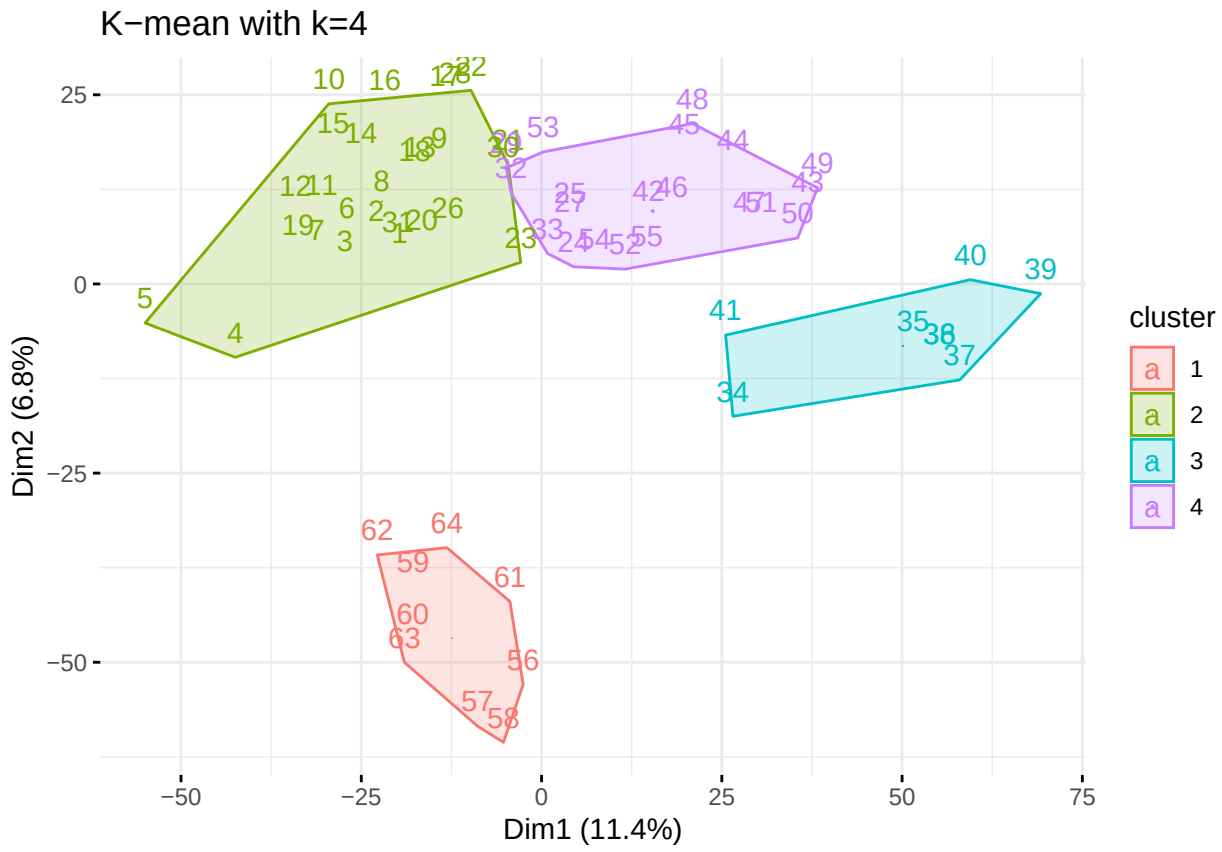

```
# Plot the clusters in PC1 and PC2 space
fviz_cluster(list(data = scaled_data, cluster = cut4clusters),
             main="Average linkage",
             palette = "Set2",show.clust.cent = F,
             labelsize = 10,
             repel = T, # Avoid label overlap (slow)
             ggtheme = theme_minimal())
```



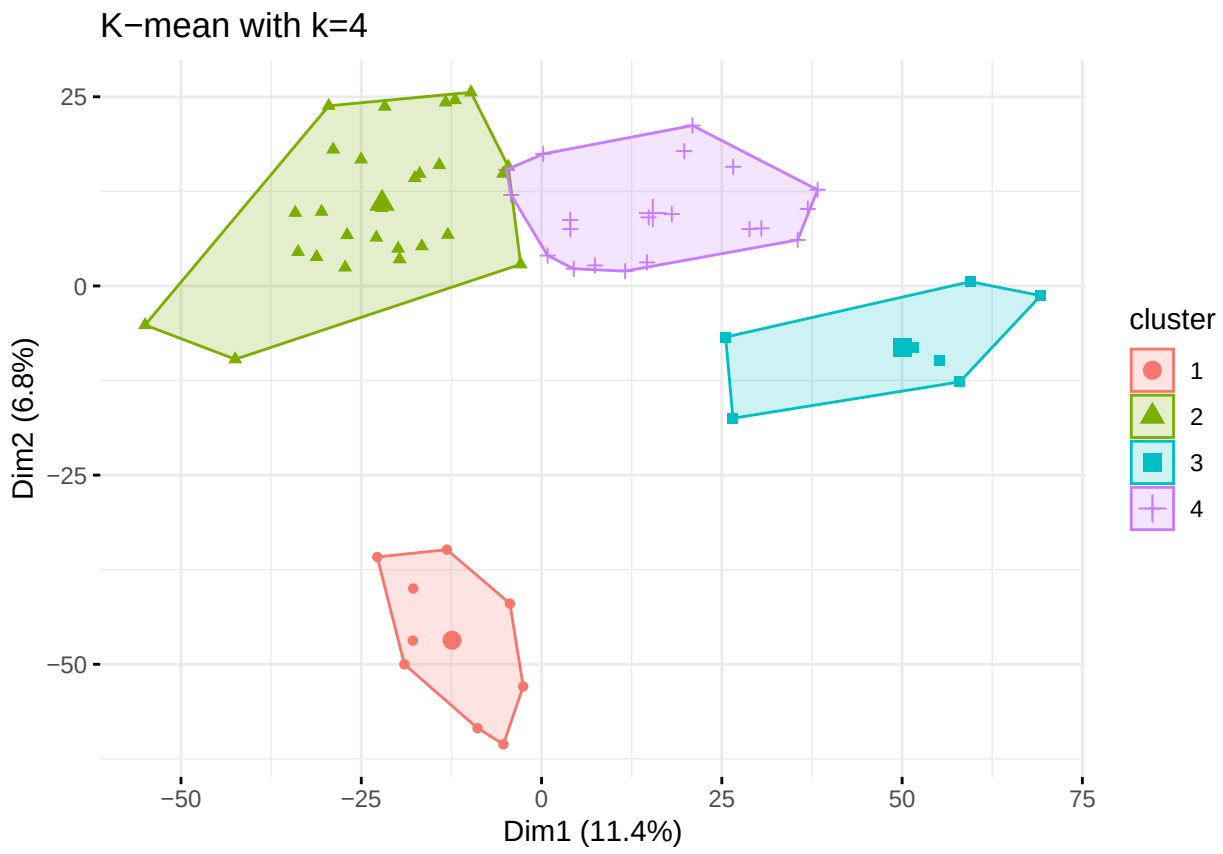

```
# c) Perform K-means clustering (k = 4 and ntart=20) of the cancer cell samples.
# Plot the clusters in PC1, PC2 space.
set.seed(1)
km_clus = kmeans(scaled_data, centers = 4, nstart = 20)
str(km_clus)
```

```
## List of 9
##  $ cluster     : int [1:64] 2 2 2 2 2 2 2 2 2 2 ...
##  $ centers     : num [1:4, 1:6830] 0.0205 0.2232 -0.4695 -0.1228 -0.0821 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "1" "2" "3" "4"
##   .. ..$ : chr [1:6830] "X1" "X2" "X3" "X4" ...
##  $ totss       : num 430290
##  $ withinss    : num [1:4] 37150 154545 44071 108801
##  $ tot.withinss: num 344567
##  $ betweenss   : num 85723
##  $ size        : int [1:4] 9 27 8 20
##  $ iter        : int 2
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
# Plot with row numbers
fviz_cluster(km_clus, data=scaled_data, main = "K-mean with k=4",
             ggtheme = theme_minimal(), geom = "text")
```




```
# Plot with points
fviz_cluster(km_clus, data=scaled_data, main = "K-mean with k=4",
             ggtheme = theme_minimal(), geom = "point")
```



2.

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)

df = read.csv("fires.csv")

df = select(df, STATE,YEAR = YEAR_,ACRES = TOTALACRES,CAUSE)
head(df)
```

```
##     STATE YEAR ACRES CAUSE
## 1 Arizona 1988  1500 Human
## 2 Arizona 1986 10390 Human
## 3 Montana 1986  1400 Human
## 4 Arizona 2002  1035 Human
## 5 Arizona 2000  5700 Human
## 6 Arizona 2000  2750 Human
```

```
dim(df)
```

```
## [1] 7179    4
```

```
# Theme set
theme_set(theme_bw())
```

```
# a) Group by YEAR, then sum ACRES from each year (call it TOTALACRES).
# Print the resulting table. What year had the largest number of fires?
# What year had the largest fire? Draw a scatterplot of TOTALACRES by YEAR,
# with a straight fitted line.

df2 = group_by(df, YEAR)
total_acres_df = summarize(df2, TOTALACRES = sum(ACRES))
# Printing resulting table
total_acres_df
```

```
## # A tibble: 37 x 2
##     YEAR TOTALACRES
##    <int>      <dbl>
##  1  1980    562457
##  2  1981    824805.
##  3  1982    261296
##  4  1983    893999
##  5  1984   1064861
##  6  1985   1937114.
##  7  1986   1101747
##  8  1987   1679488.
##  9  1988   4392164.
## 10  1989    663591.
## # ... with 27 more rows
```

```
# Printing the year that had largest number of fires
num_fire_df = summarize(df2, NumOfFire = n())
arrange(num_fire_df, desc(NumOfFire))
```

```
## # A tibble: 37 x 2
##     YEAR NumOfFire
##    <int>     <int>
##  1  2006       415
##  2  2000       369
##  3  1996       363
##  4  2012       349
##  5  2007       334
```

```
##  6  1994       331
##  7  1999       272
##  8  2003       267
##  9  1988       265
## 10  2005       256
## # ... with 27 more rows
```

```
# 2006 is the year the had the maximum number of fires (415).

# What year had the largest fire?
df[which.max(df$ACRES), ]
```

```
##      STATE YEAR  ACRES   CAUSE
## 4510 Idaho 2007 590620 Natural
```

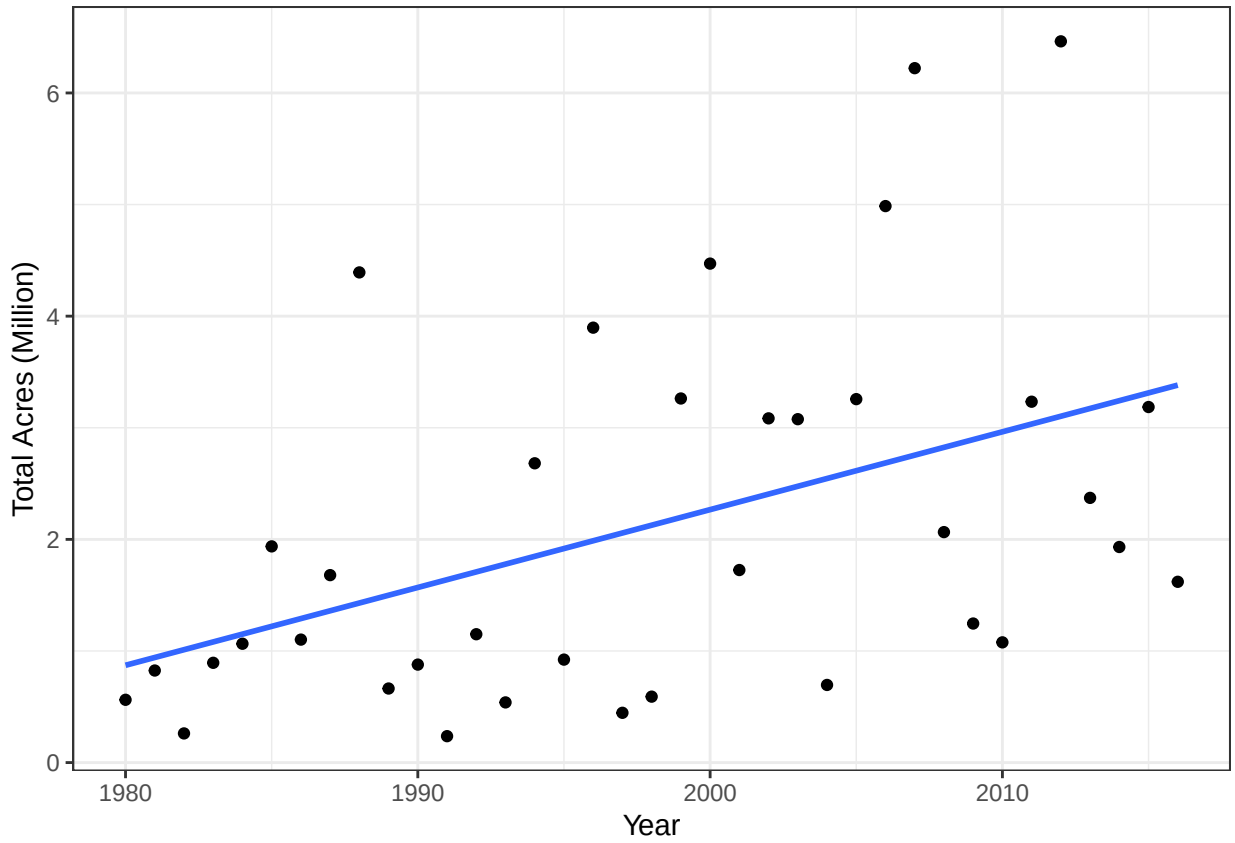
```
# 2007 had the largest fire with 590,620 Acres at Idaho.

# For better visualization, I will divide TOTALACRES by 10^6
total_acres_df$TOTALACRES = total_acres_df$TOTALACRES/1000000
total_acres_df
```

```
## # A tibble: 37 x 2
##     YEAR TOTALACRES
##    <int>      <dbl>
##  1  1980      0.562
##  2  1981      0.825
##  3  1982      0.261
##  4  1983      0.894
##  5  1984      1.06
##  6  1985      1.94
##  7  1986      1.10
##  8  1987      1.68
##  9  1988      4.39
## 10  1989      0.664
## # ... with 27 more rows
```

```
# Draw a scatterplot of TOTALACRES by YEAR with a straight fitted line.
ggplot(data=total_acres_df, mapping = aes(x=YEAR, y = TOTALACRES)) +
  geom_point() +
  geom_smooth(method='lm', se=FALSE) +
  labs(y="Total Acres (Million)", x="Year")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Note that TOTALACRES scatter around the fitted line with increased dispersion
# as the year in- creases. Thus in the next question it is suggested to use
# y=log10(TOTALACRES) in place of TOTALACRES.
```

```
# b) For Arizona, California, and Washington, group by STATE and YEAR,
# then sum ACRES from each year. Draw subplots showing the scatterplots
# (with a fitted straight line) of y=log10(TOTALACRES) by YEAR using
# facet_wrap(~STATE).

# For Arizona, California, and Washington, group by STATE and YEAR then sum ACRES

# Select only Arizona, California, and Washington states
filtered_df = filter(df,STATE=="Arizona"|STATE=="California"|STATE=="Washington")

# Group by STATE and YEAR
df3 = group_by(filtered_df, STATE, YEAR)
total_acres_state_year_df = summarize(df3, TOTALACRES = sum(ACRES))
```

```
## `summarise()` has grouped output by 'STATE'. You can override using the `.groups` argument.
```

```
total_acres_state_year_df
```

```
## # A tibble: 109 x 3
## # Groups:   STATE [3]
```

```
##    STATE    YEAR TOTALACRES
##    <chr>   <int>      <dbl>
##  1 Arizona  1980     111906
##  2 Arizona  1981       1225
##  3 Arizona  1983      12070
##  4 Arizona  1984      19372
##  5 Arizona  1985      13880
##  6 Arizona  1986      35635
##  7 Arizona  1987      46380
##  8 Arizona  1988      46402
##  9 Arizona  1989      64382
## 10 Arizona  1990      38744
## # ... with 99 more rows
```

```
# Draw subplots showing the scatterplots with a fitted straight line
ggplot(data=total_acres_state_year_df,
       mapping = aes(x=YEAR, y = log10(TOTALACRES))) +
  geom_point(size=0.5) +
  geom_smooth(method='lm', se=FALSE) +
  labs(y="Log10 of Total Acres", x="Year") +
  facet_wrap(~STATE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```