HW10

David Schultheiss

11/13/2020

Problem 1

a)

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## Warning: package 'ggplot2' was built under R version 3.6.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

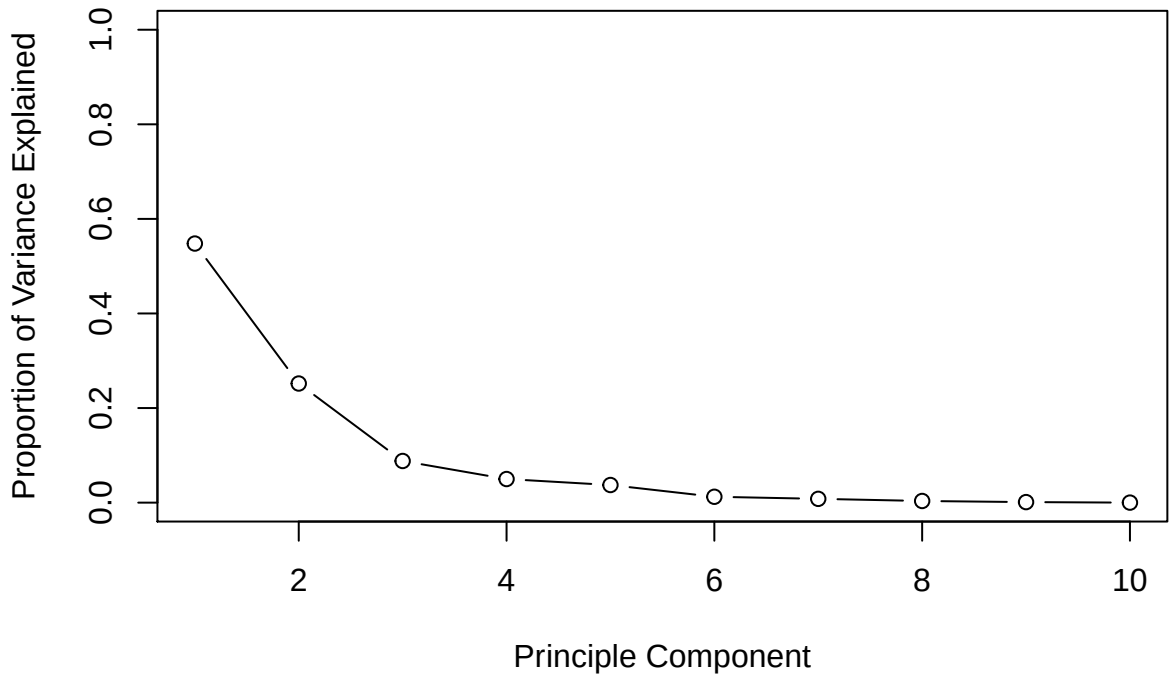
```
library(readr)
tumor = read_csv("/Users/davidschultheiss/Downloads/tumor.csv") %>%
  select(-1)
```

```
## Parsed with column specification:
## cols(
##   Diagnosis = col_character(),
##   Radius = col_double(),
##   Texture = col_double(),
##   Perimeter = col_double(),
##   Area = col_double(),
##   Smoothness = col_double(),
##   Compactness = col_double(),
##   Concavity = col_double(),
##   `Concave Points` = col_double(),
##   Symmetry = col_double(),
##   `Fractal Dimension` = col_double()
## )
```

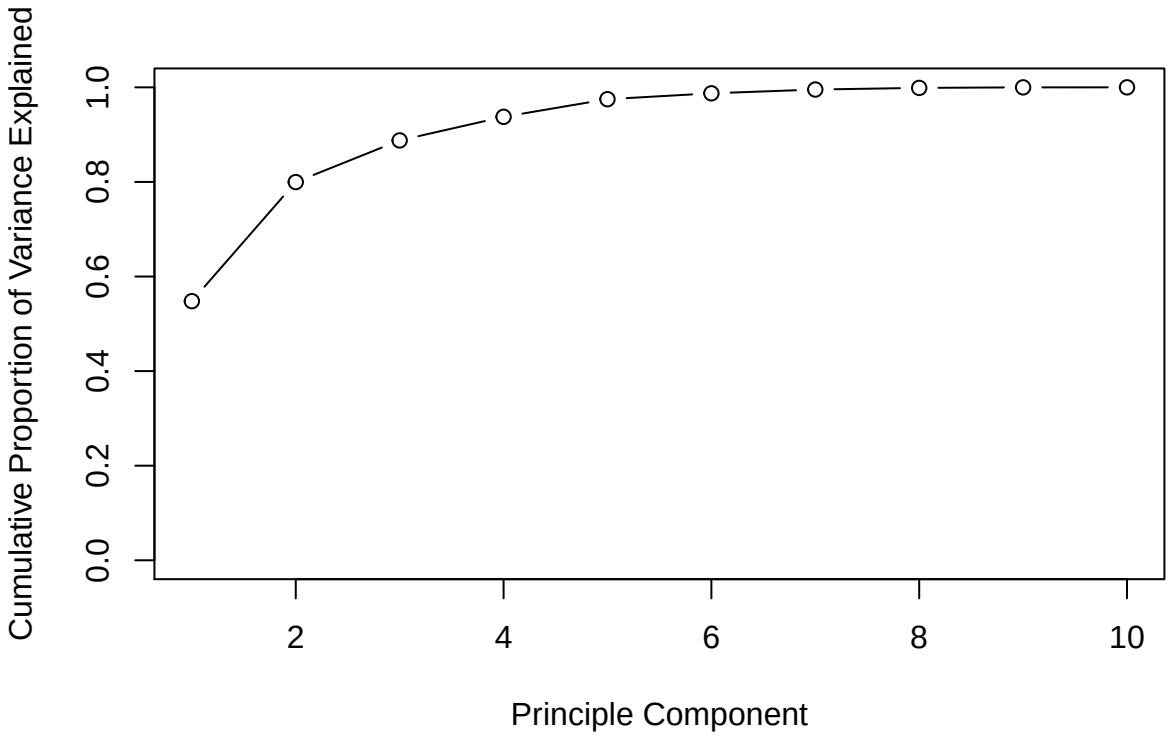
b)

```
pr.tumor = prcomp(tumor, scale=T)
pr.var = pr.tumor$sdev^2
pve = pr.var/sum(pr.var)

plot(pve, type= 'b', ylim= c(0,1),
     xlab= 'Principle Component',
     ylab= 'Proportion of Variance Explained')
```




```
plot(cumsum(pve), type= 'b', ylim= c(0,1),
     xlab= 'Principle Component',
     ylab= 'Cumulative Proportion of Variance Explained')
```



I would keep 5 principle components, beyond that very little additional variation explained.

c)

```
cumsum(pve)
```

```
##  [1] 0.5478588 0.7997302 0.8877917 0.9376926 0.9749465 0.9873607 0.9953692
##  [8] 0.9988582 0.9999718 1.0000000
```

Two Principal Components explain 80% of the variation. A third bumps this to 88.8%.

d)

```
pr.tumor$rotation
```

```
##                         PC1          PC2         PC3          PC4
## Radius            -0.36393793  0.313929073 -0.12442759  0.029558858
## Texture           -0.15445113  0.147180909  0.95105659  0.008916084
## Perimeter         -0.37604434  0.284657885 -0.11408360  0.013458069
## Area              -0.36408585  0.304841714 -0.12337786  0.013442682
## Smoothness        -0.23248053 -0.401962324 -0.16653247 -0.107802033
## Compactness       -0.36444206 -0.266013147  0.05827786 -0.185700413
## Concavity         -0.39574849 -0.104285968  0.04114649 -0.166653523
## Concave Points    -0.41803840 -0.007183605 -0.06855383 -0.072983951
## Symmetry          -0.21523797 -0.368300910  0.03672364  0.892998475
## Fractal Dimension -0.07183744 -0.571767700  0.11358395 -0.349331790
##                          PC5          PC6         PC7          PC8
## Radius            -0.031067022  0.264180150 -0.04418839  0.084834062
## Texture           -0.219922761  0.032206572  0.02055748 -0.007126797
```

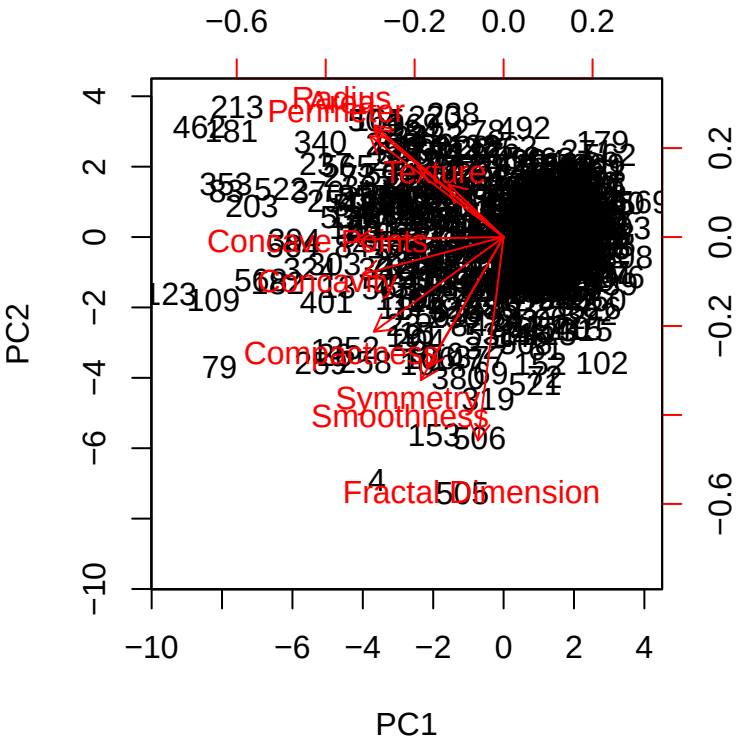
```
## Perimeter          -0.005945081  0.237819464 -0.08336923  0.089258879
## Area               -0.019341222  0.331707454  0.26118796  0.144609749
## Smoothness         -0.843745292 -0.062225368  0.01129197  0.170503128
## Compactness         0.240182967 -0.005271104 -0.80380484  0.063980134
## Concavity           0.312533244 -0.601467155  0.36713629  0.449573315
## Concave Points     -0.009180198 -0.265613395  0.14131308 -0.850918762
## Symmetry            0.112888068  0.061957003  0.04790201  0.016455606
## Fractal Dimension   0.264878077  0.567918997  0.34521359 -0.065259461
##                            PC9          PC10
## Radius              0.474425305 -0.6690714888
## Texture             0.004212629  0.0002497826
## Perimeter           0.380167210  0.7404905337
## Area               -0.747347357 -0.0323589585
## Smoothness          0.005847386  0.0036904058
## Compactness        -0.218732407 -0.0527527802
## Concavity           0.081170670 -0.0103668020
## Concave Points     -0.022024652 -0.0037475480
## Symmetry            0.009067850  0.0014669472
## Fractal Dimension   0.129667491  0.0070573477
```

We can see the highest loading for Z1 is from concave points. Concavity, compactness, radius, area, and perimeter all contribute most heavily to Z1.

The highest loading for Z2 is fractal dimension. Symmetry and smoothness contribute most heavily out of the rest of the features.

e)

```
biplot(pr.tumor, scale= 0)
```



The first principal component puts most weight on the features concave points, concavity, and compactness. We could conclude these variables are

related. The second principal component is primarily defined by the feature fractal dimension. Variables symmetry, smoothness, texture, perimeter, area, and radius all have moderate impacts on both principal components.

Symmetry and smoothness look to be closely related. Texture, area, perimeter, and radius are also closely related.

Problem 2

a) Two Clusters would probably be best because we know their are two diagnosis groups in our dataset - malignant and benign.

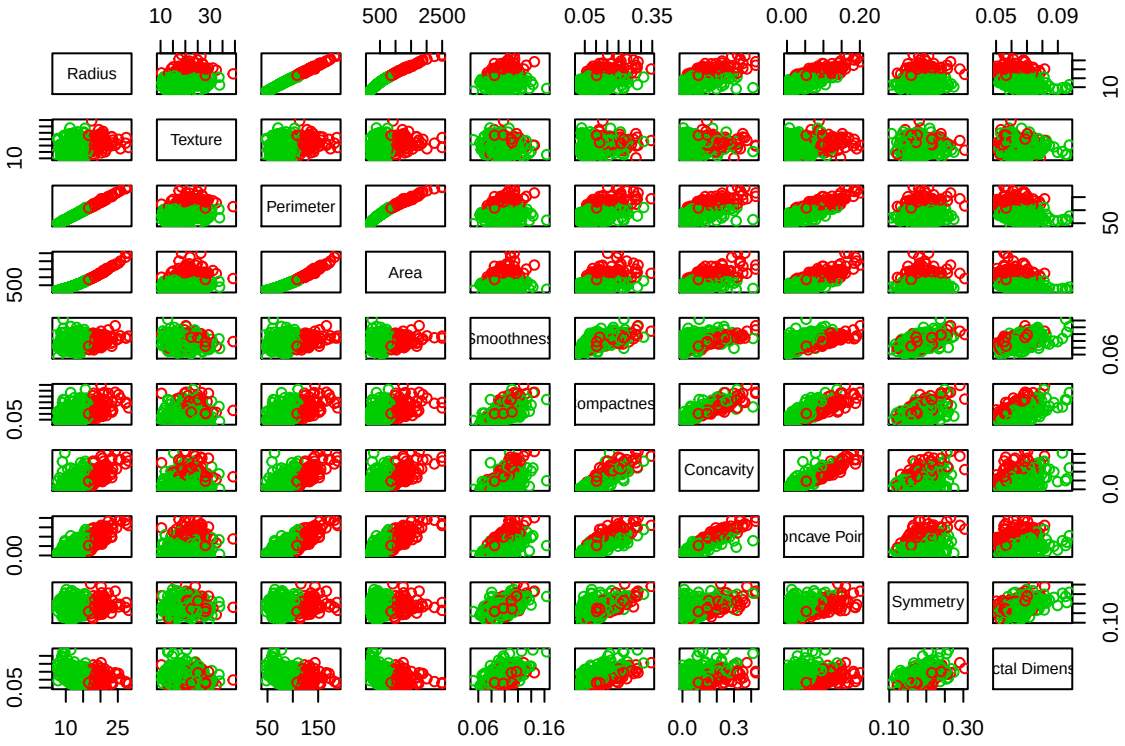
b)

```
km.tumor = kmeans(tumor, 2, nstart= 25)
km.tumor$centers
```

```
##     Radius  Texture Perimeter     Area Smoothness Compactness Concavity
## 1 19.61831 21.84194 129.70887 1211.936 0.10031129  0.14585258 0.1759842
## 2 12.59721 18.57845  81.45276  499.667 0.09525933  0.09277371 0.0645051
##   Concave Points  Symmetry Fractal Dimension
## 1     0.10033298 0.1900750        0.05994202
## 2     0.03459259 0.1786782        0.06359333
```

c)

```
plot(tumor, col= (km.tumor$cluster+1))
```



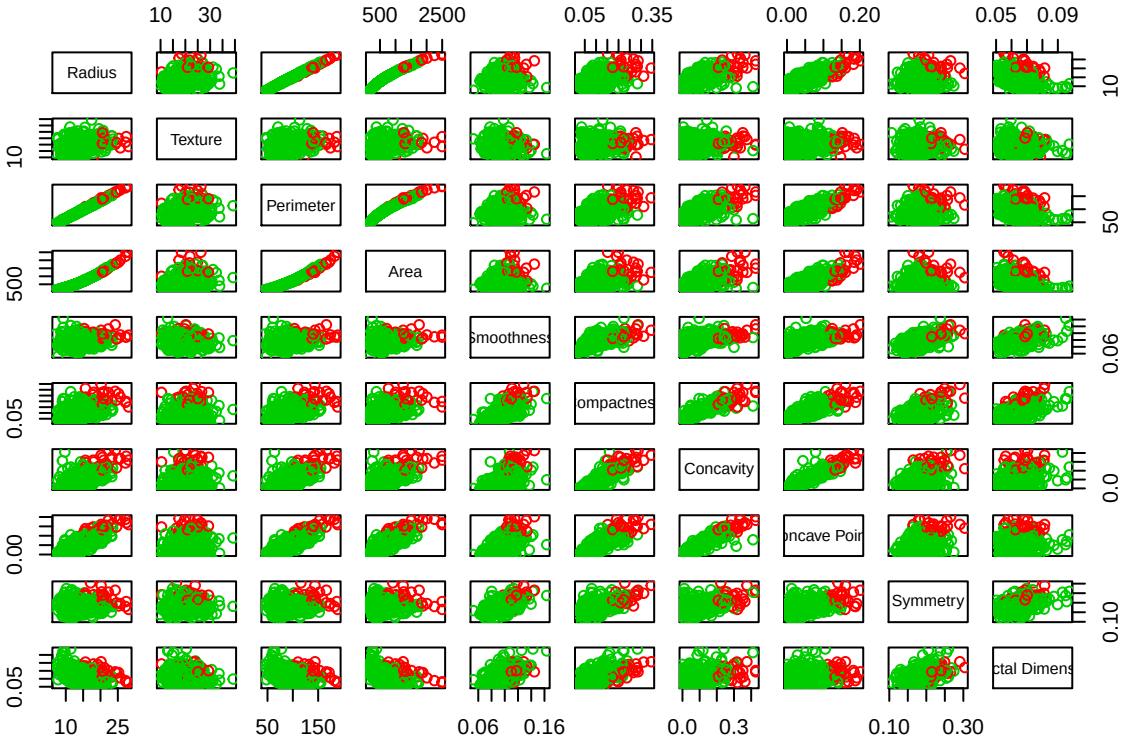
```
km.tumor
```

```
## K-means clustering with 2 clusters of sizes 124, 445
##
## Cluster means:
##     Radius  Texture Perimeter     Area Smoothness Compactness Concavity
## 1 19.61831 21.84194 129.70887 1211.936 0.10031129  0.14585258 0.1759842
## 2 12.59721 18.57845  81.45276  499.667 0.09525933  0.09277371 0.0645051
##   Concave Points  Symmetry Fractal Dimension
## 1     0.10033298 0.1900750        0.05994202
## 2     0.03459259 0.1786782        0.06359333
##
## Clustering vector:
##   [1] 1 1 1 2 1 2 1 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 1 1 1 2 1 2 1 1 2 1 1 2 1 2
##  [38] 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2
##  [75] 2 2 2 1 1 2 2 2 1 1 2 1 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
## [112] 2 2 2 2 2 2 2 2 1 2 1 1 2 2 2 2 1 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
## [149] 2 2 2 2 2 2 2 2 2 1 1 2 2 2 1 1 2 1 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2
## [186] 2 1 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 1 1 2 2 2 2 2 1 2 2 1 2 1 1 2 2 2 2 2 1 1 2 2
## [223] 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 1 1 2 1 2 2 2 2 2 1 2 2 2 2 2 1 2 1 1 1 2 1 2 2
## [260] 2 1 1 1 2 1 1 2 2 2 2 2 2 1 2 1 2 2 1 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [297] 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2
## [334] 2 2 1 2 1 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 1 1
## [371] 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2
## [408] 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 1 2 2
## [445] 1 2 1 2 2 1 2 1 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
## [482] 2 2 2 2 2 2 2 1 2 1 2 1 1 2 2 2 2 2 2 1 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1
## [519] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [556] 2 2 2 2 2 2 2 2 1 1 1 1 1 2
##
## Within cluster sum of squares by cluster:
## [1] 11209123 10056128
##  (between_SS / total_SS =  69.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Clustering is effective for scatterplots like smoothness x area, fractal dimension x area, concave points x area, etc. Area, perimeter, and radius are all interchangeable here. Clustering is ineffective for many other pairs like symmetry x smoothness, texture x compactness, etc.

d)

```
hc.tumor = hclust(dist(scale(tumor)), method= 'complete')
hc.1 = cutree(hc.tumor, 2)
plot(tumor, col= (hc.1+1))
```



The red group in the split is much smaller using heirarchical clustering. The groups were more even using k-means clusting. Hierarchical clustering makes more sense in the context of our dataset. Most of these data points should be in one cluster (benign).