STOR 565 Homework 6

Name:Ziyang Qin

Collaborated with:

This homework is due on Feb. 27st at 11:55 pm.

Instruction: fill your answers in the `.Rmd` , compile it to HTML/PDF and submit the complied file. Uncompiled `.Rmd` file will not be graded.

Remark. This homework aims to help you understand PCA and its applications.

1 NBA Dataset

Import the dataset from "nba-teams-2017.csv". Create a new `data.frame` that contains the following columns:

- `team`
- `wins`
- `points`
- `points3`
- `free_throws`
- `off_rebounds`
- `def_rebounds`
- `assists`
- `steals`
- `personal_fouls`

1.0.1 (a) (*5 pt*) Create box plots of the quantitative features (i.e. all but) teams to see if you should scale the data when performing PCA. Describe your findings in words.

```
library(readr)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following objects are masked from 'package:Matrix':
##
##     expand, pack, unpack
```

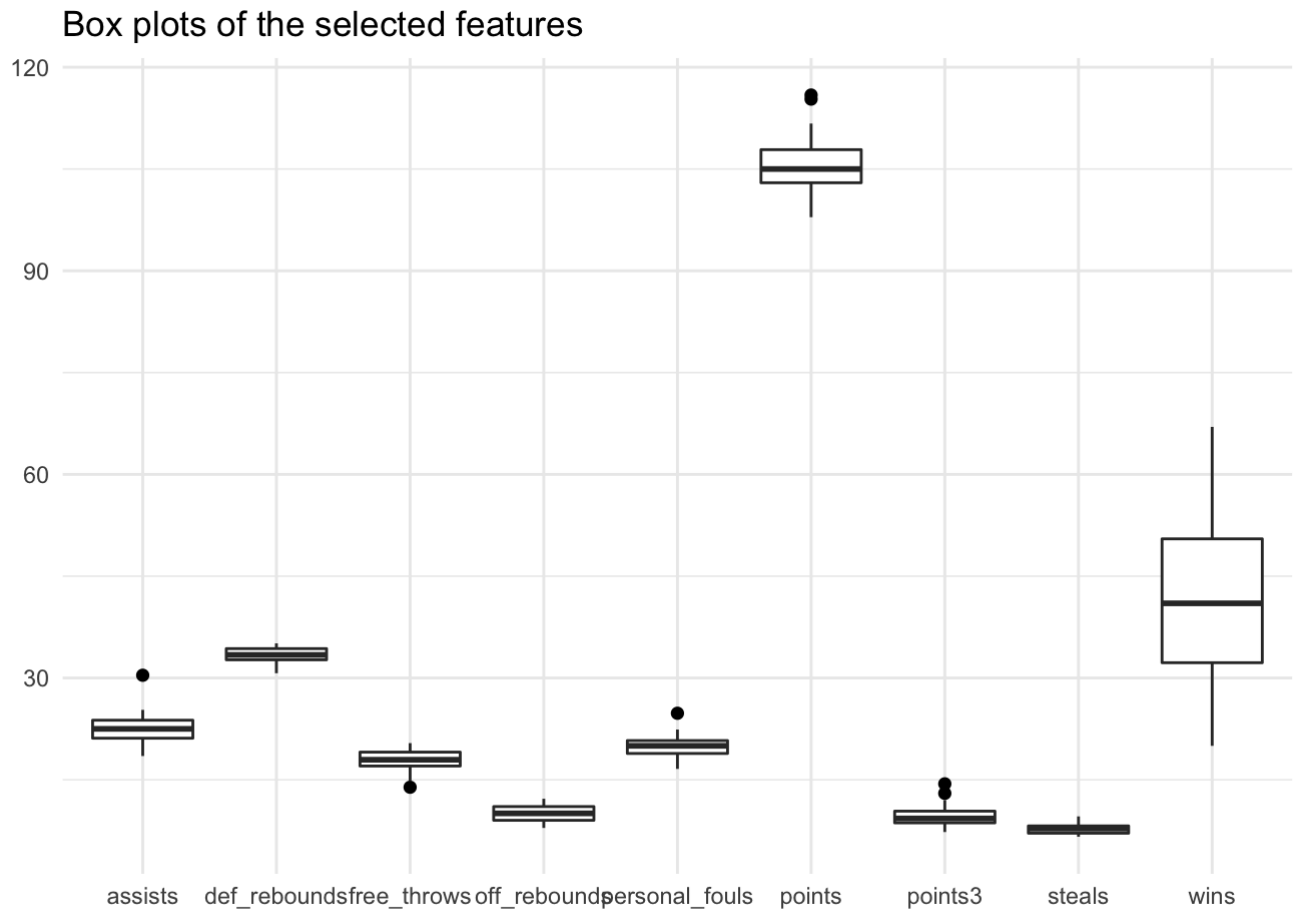
```
library(ggplot2)
library(dplyr)
```

```
##
## 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
```

```
nba <- read.csv('nba-teams-2017.csv')



#first select predictors
nba_predictor <- select(nba, team, wins, points, points3, free_throws, off_rebounds, def
_rebounds, assists, steals, personal_fouls)
nba_predictor_long <- nba_predictor %>%
  pivot_longer(names_to = "metric", values_to="value", cols = -team)

#generate boxplot
ggplot(nba_predictor_long, aes(x = metric, y = value)) +
  geom_boxplot(outlier.colour="black", outlier.shape=16,
               outlier.size=2, notch=FALSE) +
  labs(title = "Box plots of the selected features",
       x = "",
       y = "") +
  theme_minimal()
```



###The box plot shows that the mean and variance are not in the accepctale range, and there are some outliers, so we have to scale the data.

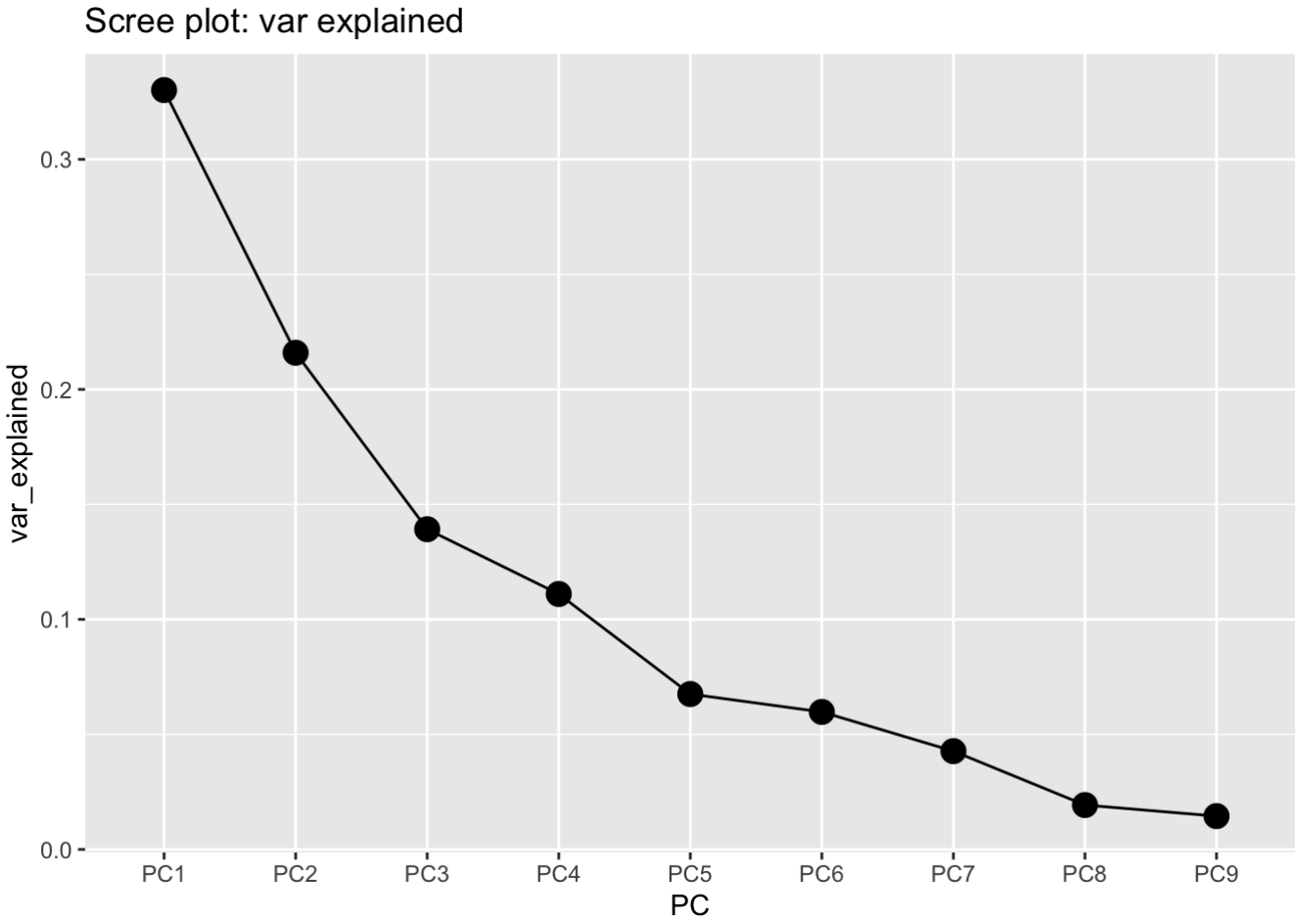
1.0.2 (b) (*5 pt*) Obtain PC loadings of the first four princple components (PCs). Only display the first few elements of each loading in your report.

```
pca_nba_predictor <- prcomp(nba_predictor %>% select(-team), scale = TRUE) # here we sca
le the data.

pca_nba_predictor$rotation # matrix of loadings
```

```
##                       PC1         PC2         PC3         PC4         PC5
## wins           -0.42366308  0.07555818 -0.11681772  0.21657745 -0.61595064
## points         -0.50246947 -0.21708761  0.19677723 -0.07946391 -0.05044092
## points3        -0.41664778  0.17268215 -0.08316881 -0.51204012 -0.25692307
## free_throws    -0.24452950 -0.41519726  0.30946852 -0.33272209  0.25982453
## off_rebounds    0.08111297 -0.39160091  0.47926467  0.46463787 -0.39815320
## def_rebounds   -0.26053718  0.26461371  0.57662166  0.15459073  0.39642492
## assists        -0.45236958  0.05322237 -0.26482231  0.27220000  0.28037225
## steals         -0.20525546 -0.41895791 -0.45168498  0.37698249  0.29428871
## personal_fouls  0.11583180 -0.58585494 -0.09277492 -0.34335891 -0.06040805
##                       PC6         PC7          PC8          PC9
## wins            0.48542178 -0.22633056  0.271643228 -0.119177660
## points         -0.20056805  0.04186789  0.045916018  0.780210602
## points3        -0.32914693 -0.04503329 -0.498095849 -0.320949727
## free_throws     0.52512090  0.39181975 -0.001769406 -0.254077175
## off_rebounds   -0.31732735  0.18751727 -0.211619157 -0.235198714
## def_rebounds   -0.01248518 -0.56416648  0.020304034 -0.172349829
## assists        -0.39106430  0.35391157  0.442443029 -0.309445202
## steals          0.13742263 -0.27523048 -0.504936613  0.002393494
## personal_fouls -0.25957599 -0.48684607  0.424366978 -0.169467411
```

1.0.3 (c) (*5 pt*) Generate a scree plot describing the amount explained by the various PCs.

```
var_explained_nba_predictor = data.frame(PC= paste0("PC",1:9),
                                 var_explained=(pca_nba_predictor$sdev)^2/sum((pca_nba_p
redictor$sdev)^2),
                                 cum_explained=cumsum((pca_nba_predictor$sdev)^2)/sum((p
ca_nba_predictor$sdev)^2))

var_explained_nba_predictor %>%
  ggplot(aes(x=PC,y=var_explained, group=1))+
  geom_point(size=4)+
  geom_line()+
  labs(title="Scree plot: var explained")
```

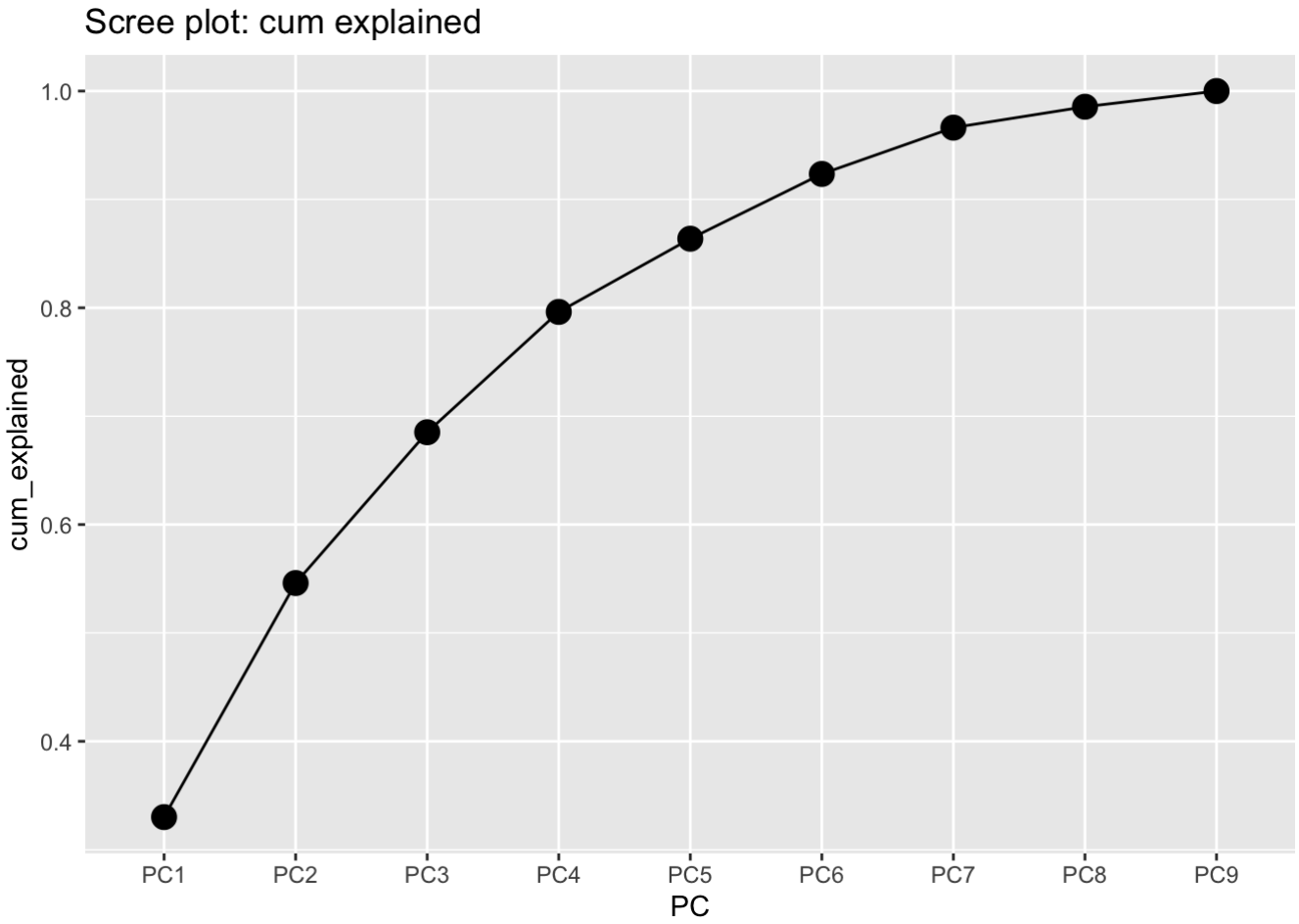


1.0.4 (d) (*5 pt*) Make another plot showing the cumulative percent of the variance explained.

Precisely: for each $1 \leq k \leq 10$ you are plotting:

$$\frac{\sum_{j=1}^{k} d_j^2}{\sum_{j=1}^{10} d_j^2}.$$

```
var_explained_nba_predictor %>%
  ggplot(aes(x=PC,y=cum_explained, group=1))+
  geom_point(size=4)+
  geom_line()+
  labs(title="Scree plot: cum explained")
```



1.0.5 (e) (*5 pt*) If you were to retain all PCs which explain at least 90% of the variance, how many PCs would you retain?

```
library(gt)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(textshape)
```

```
##
## Attaching package: 'textshape'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

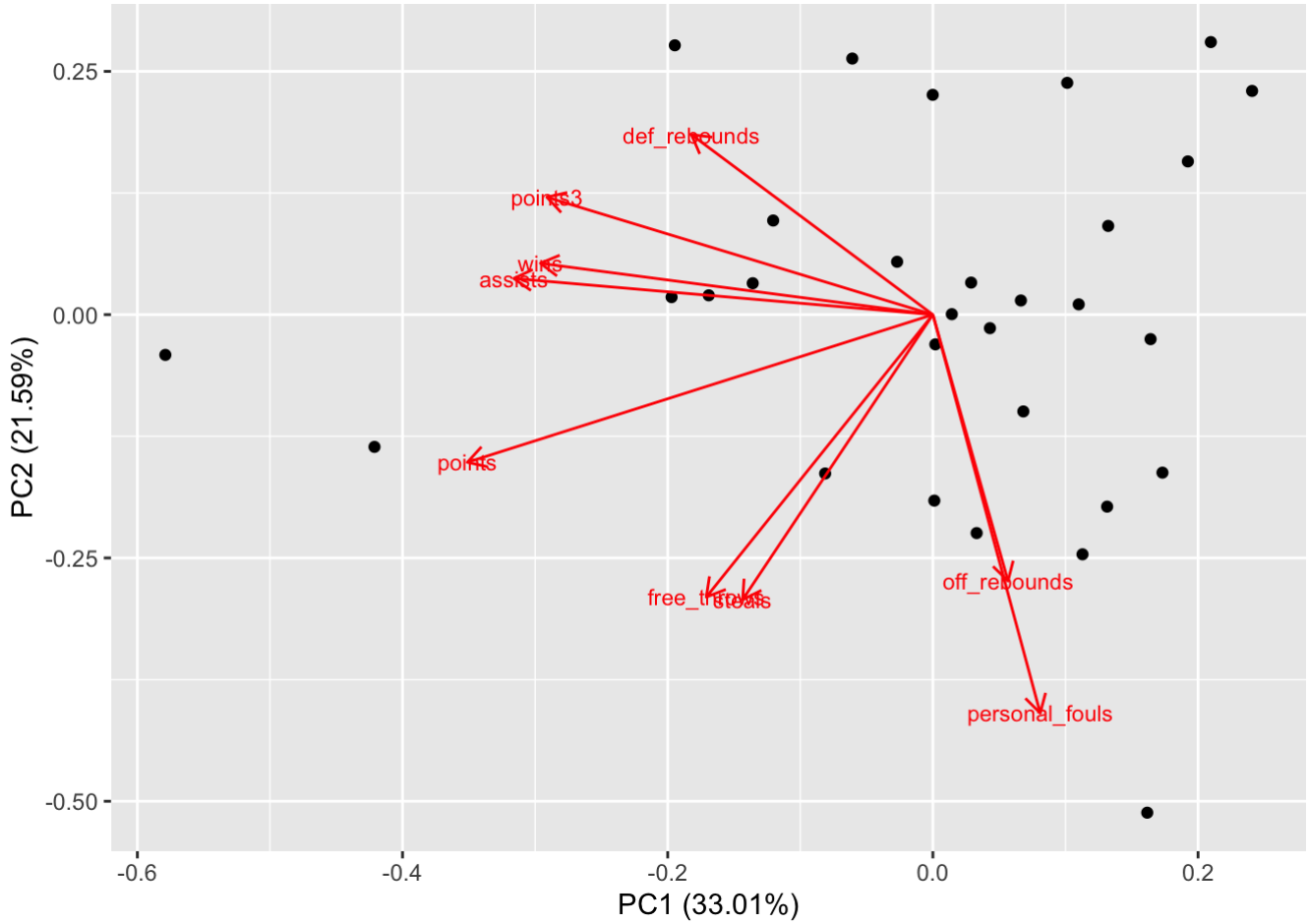
```
varnames <- c("wins", "points", "points3", "free_throws", "off_rebounds", "def_rebound
s", "assists", "steals", "personal_fouls")
nba_e <- read.csv("nba-teams-2017.csv", stringsAsFactors = FALSE) %>% select(all_of(c("t
eam", varnames))) %>% column_to_rownames("team")

nba_scale <- scale(nba_e)
nba_svd <- svd(nba_scale)
variances_e <- nba_svd$d^2

a <- data.frame(var = cumsum(variances_e)) %>% mutate(n_pcs = row_number(),
var_prop = var/max(var),
var_prop_prec = scales::percent(var_prop, accuracy = 0.01))

a %>% filter(var_prop >= 0.9) %>%
select(- var_prop) %>%
top_n(1, - n_pcs) %>%
kable(booktabs = TRUE, linesep = "", align = "r",
col.names = c("Variance Explained", "PCs", "Proportion of Variance Explained"),
        caption = "PCs",
escape = TRUE) %>% kable_styling(latex_options = "HOLD_position") %>% row_spec(0)
```

PCs

Variance Explained	PCs	Proportion of Variance Explained
241.0413	6	92.35%

1.0.6 (f) (*10 pt*) Plot PC1 vs PC2 with the team names and try to interpret your findings.

```
library(ggfortify)

library(dplyr)
pca_res <- prcomp(nba_predictor %>% select(-team), scale. = TRUE)
autoplot(pca_res, data = nba_predictor, label.size = 3,  loadings = TRUE, loadings.label
= TRUE, loadings.label.size = 3)
```



2 RedfinHouse Image

Import the image from “Redfin_house.png”. Let X be the pixel intensity associated with the **red color** in the image.

```
library(png, quietly = TRUE)
library(grid, quietly = TRUE)

 X <- readPNG("Redfin_house.png")[,,1]
 grid.raster(X)
```

Hints.

- Review tutorial in “ `Week7_Feb22_Feb24/HM6/More PCA Examples` ” in class dropbox folder. **Example 2** can be useful for this problem, e.g., how to load the png image data to R use R function `readPNG` .
- See the **Value** section of `?png::readPNG` to remind yourself of the organization of the raster array output.

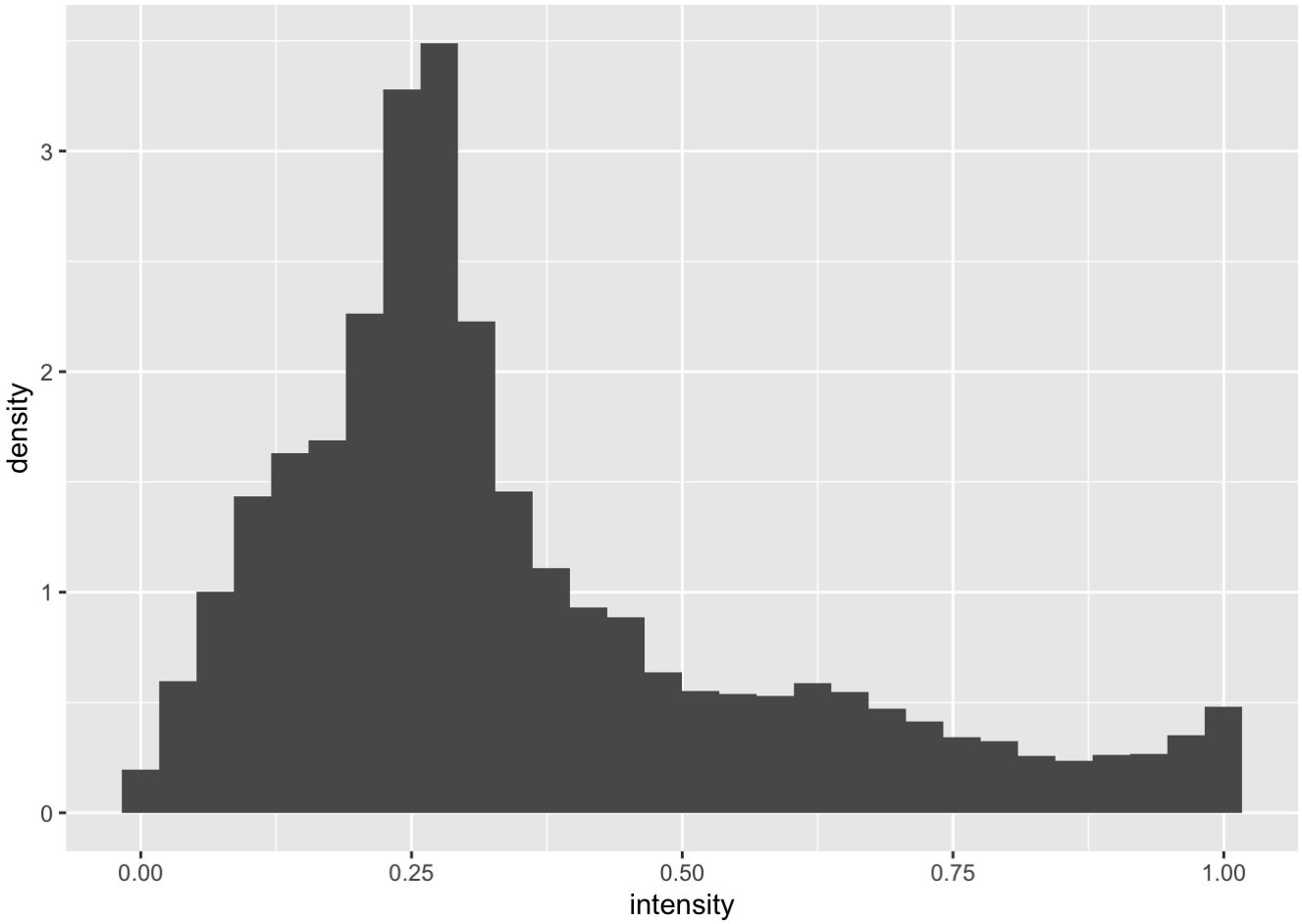
2.0.1 (a) (*5 pt*) What are the dimensions of X? Plot a histogram of the pixel intensities within the image.

```
dim(X)
```

```
## [1] 505 798
```

```
data.frame(intensity = as.vector(X)) %>% ggplot(aes(x = intensity)) + geom_histogram(aes
(, y = ..density..))
```

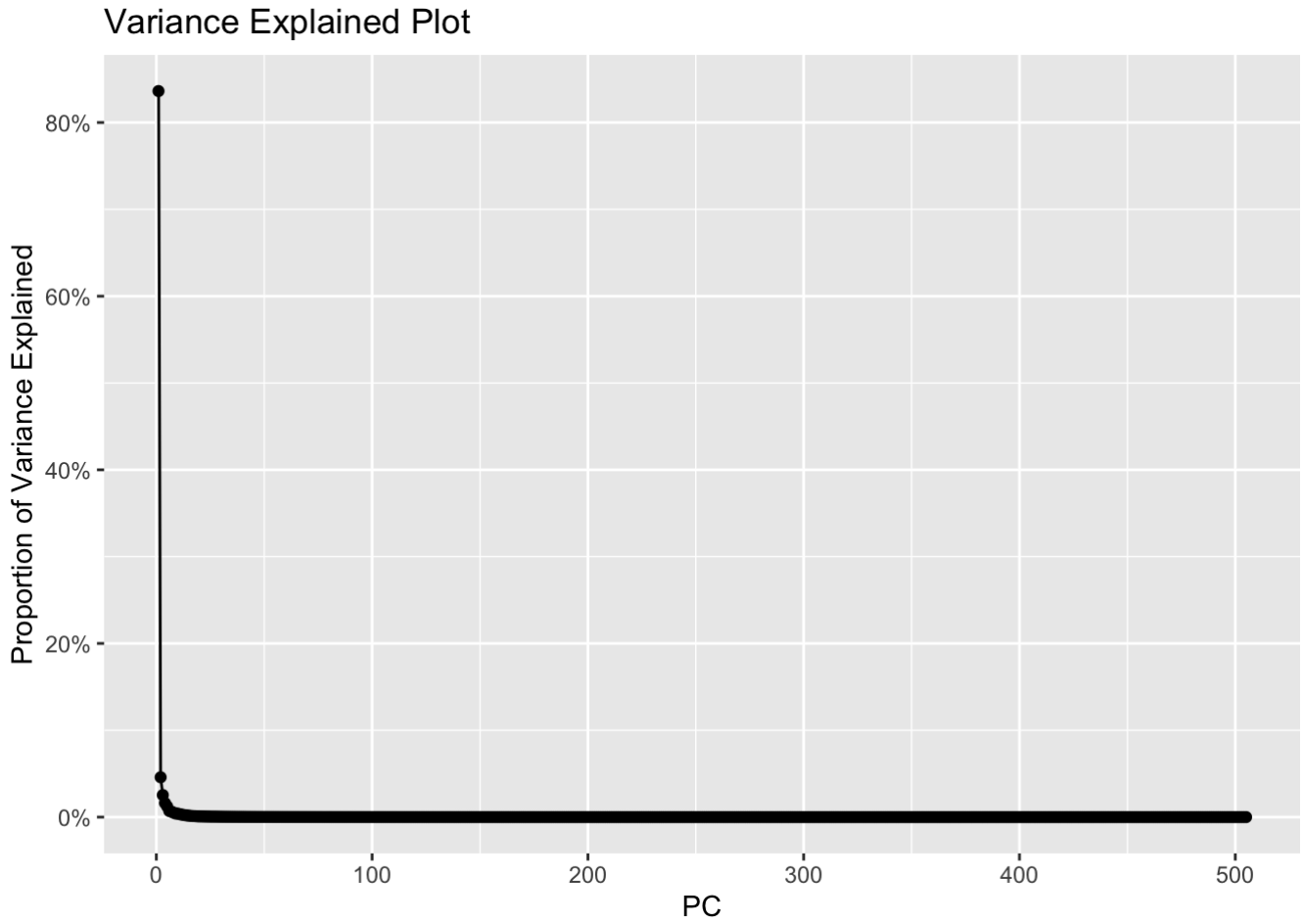
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



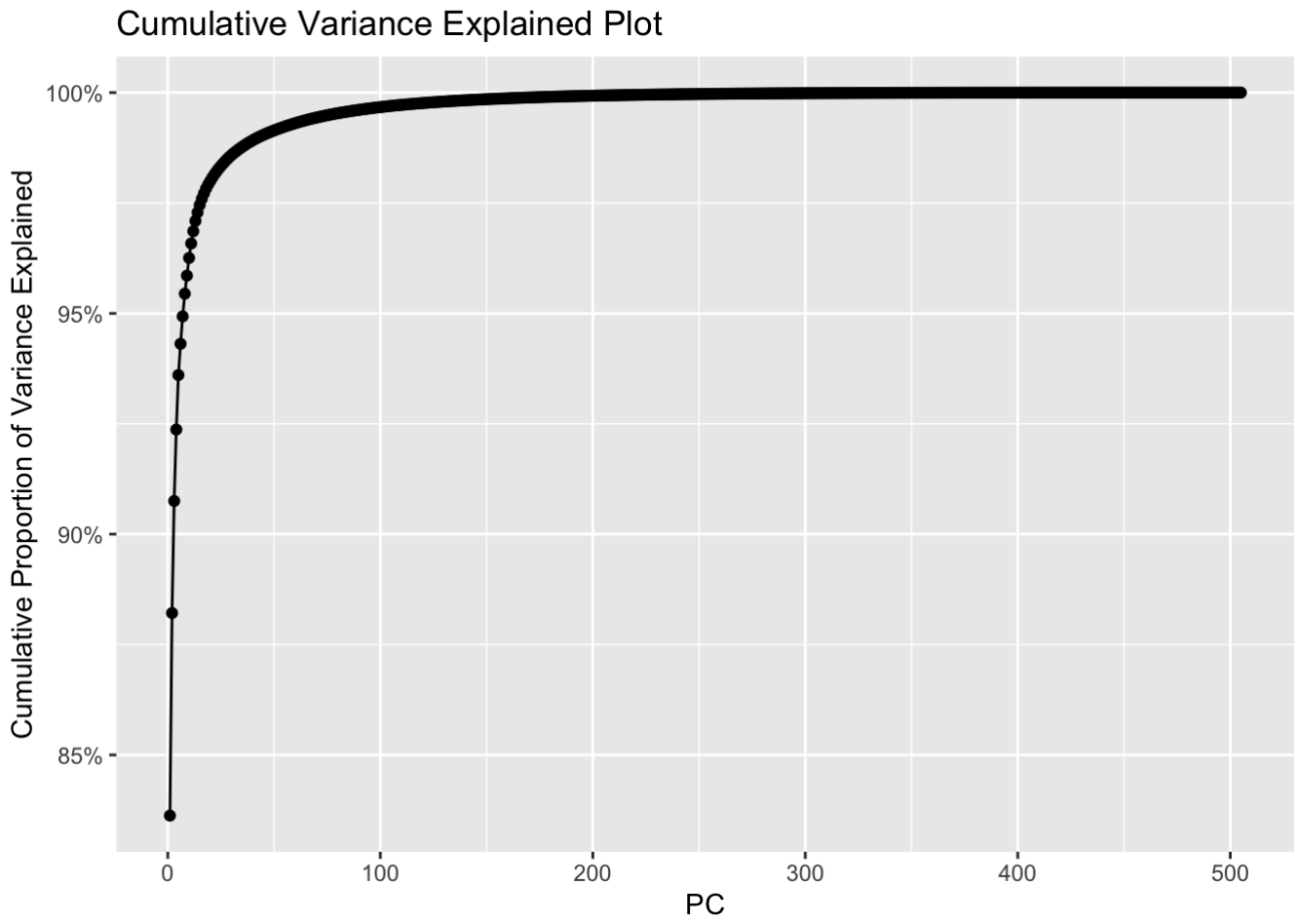
2.0.2 (b) (*10 pt*) Now let's do PCA for the row vectors in X. Plot the scree plots for this data, which illustrate the percentage variation explained against the number of principal components and the cumulative percentage variation explained against the number of principal components. How many PCs are needed to explain 90% of the total variation of X?

```
library(kableExtra)             # Load the kableExtra package
```

```
library(gt)
library(ggplot2)

X_svd <- svd(X)
variances <- X_svd$d^2

data.frame(var = variances) %>%
mutate(id = row_number(), var_prop = var/sum(var)) %>% ggplot(aes(x = id, y = var_prop))
+
geom_line() +
geom_point() +
scale_y_continuous(labels = scales::percent) + labs(title = "Variance Explained Plot",
       x = "PC",
       y = "Proportion of Variance Explained")
```




```
b <- data.frame(var = cumsum(variances)) %>% mutate(n_pcs = row_number(),
var_prop = var/max(var),
var_prop_perc = scales::percent(var_prop, accuracy = 0.01))

b %>% ggplot(aes(x = n_pcs, y = var_prop)) + geom_line() +
geom_point() +
scale_y_continuous(labels = scales::percent) + labs(title = "Cumulative Variance Explain
ed Plot",
     x = "PC",
     y = "Cumulative Proportion of Variance Explained")
```



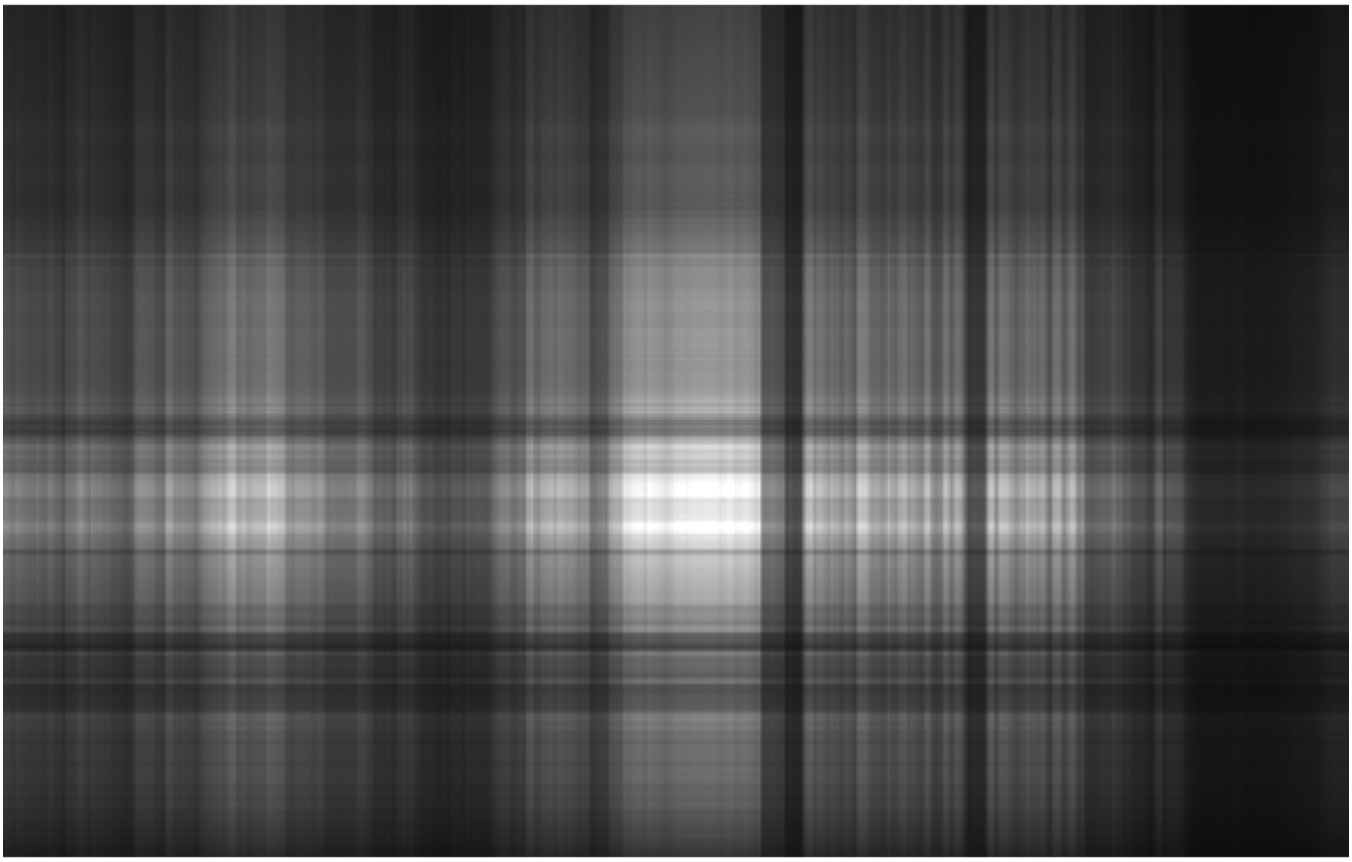

```
b %>% filter(var_prop >= 0.9) %>%
select(- var_prop) %>%
top_n(1, - n_pcs) %>%
kable(booktabs = TRUE, linesep = "", align = "r",
col.names = c("Variance Explained", "Number of PCs", "Proportion of Variance Explaine
d"),
        caption = "PC",
escape = TRUE) %>% kable_styling(latex_options = "HOLD_position")
```

PC

Variance Explained	Number of PCs	Proportion of Variance Explained
64122.39	3	90.75%

2.0.3 (c) (*10 pt*) For $d = 1, 5, 10, 15, 20, 30, 50, 100, 200$ project the image onto the first d principal components and plot the resulting compressed image for each d. For each of the nine plots, include the cumulative percentage variation explained by the projection in the title of your plots.

```
pcs <- c(1, 5, 10, 15, 20, 30, 50, 100, 200)

X_pro <- matrix(0, nrow = nrow(X), ncol = ncol(X))
for(pc in pcs) {
  if(pc == 1) {
    X_pro <- with(X_svd, d[1] * u[,1] %*% t(v[,1]))
  }
  else {
    X_pro <- with(X_svd, u[,1:pc] %*% diag(d[1:pc]) %*% t(v[,1:pc]))
  }
  grid.newpage(recording = FALSE)
  grid.raster(pmax(pmin(X_pro, 1), 0))
}
```




REDFIN