Lab 5

PSTAT131-231

Objectives

- Implement PCA in R
  - via SVD
  - via spectral decomposition
- Explore interpretation of PCs
- Profile U.S. colleges using PCA

The lab will make use of the U.S. News and World Report's 1995 college statistics. This dataset is in the ISLR package, and contains measurements for 777 U.S. colleges on admissions, expenditures, faculty, and student-faculty ratio.

For the lab, the variables are modified slightly to include the following:

- Private/public
- Percentage of new students from the top 10% and top 25% of their high school class
- Percentage of full time undergraduates
- Out of state tuition
- Percentage of faculty with PhD degrees
- Percentage of faculty with terminal degrees
- Student-faculty ratio
- Graduation rate
- Acceptance rate
- Expenditure per student
- Student expenses (room, board, books, and spending)

```
load('data/college.RData')
```

PCA will be conducted on the data after removing the `college` and `private` variables.

Computing principal components

The following illustrates computing principal components via both the SVD and the spectral (eigen) decomposition. As an initial step, the features should be stored separately from the college name and type (public/private), and centered and scaled.

```
# extract features and center and scale
x_mx <- college %>%
  select(-c('college', 'private')) %>%
  scale(center = T, scale = T)
```

To compute PCs via the SVD, pass the features directly to `svd()`. This computes the decomposition $\mathbf{X} = \mathbf{UDV}'$, where $\mathbf{V}$ are the feature loadings.

```
# compute SVD
x_svd <- svd(x_mx)
```

```r
# get loadings
v_svd <- x_svd$v
```

Alternately, the loadings can be found by the spectral decomposition of the covariance matrix of the features. To implement, pass `cov(x_mx)` to `eigen()`. This computes the decomposition $\text{cov}(\mathbf{X}) = \mathbf{V}\mathbf{\Lambda}\mathbf{V}'$ where $\mathbf{\Lambda}$ is a diagonal matrix of the eigenvalues.

```r
# compute eigendecomposition of covariance
x_eigen <- cov(x_mx) %>% eigen()

# get pc loadings
v_eigen <- x_eigen$vectors

# compare
bind_cols(svd = v_svd[, 1], eigen = v_eigen[, 1])
```

Notice that the result matches the SVD but with opposite sign – the eigendecomposition is unique up to signs.

Check also that $\mathbf{\Lambda}$ matches $\mathbf{D}'\mathbf{D}$:

```r
# compare eigenvalues and diagonal matrix from svd
bind_cols(svd = x_svd$d^2/(nrow(x_mx) - 1),
          eigen = x_eigen$values)
```

So either approach can be used to find the PC loadings and variances.

The principal components are the product of the loadings with the feature matrix: $\mathbf{Z} = \mathbf{X}\mathbf{V}$. Computing them is a simple algebraic operation in R using matrix multiplication (the `%*%` operator):

```r
# compute PCs
z_mx <- x_mx %*% x_svd$v
```

The principal component variances are usually obtained from the decomposition. However, we can check whether those match the PC variances from a direct calculation of the variances of the columns in $\mathbf{Z}$:

```r
# compute PC variances
pc_vars <- x_svd$d^2/(nrow(x_mx) - 1)

# compare with direct calculation
z_vars <- cov(z_mx) %>% diag()
cbind(z_vars, pc_vars)
```

In lecture it was claimed that the sum of these variances (the 'total variance') captures in some sense the total variation in $\mathbf{X}$. We can verify this by comparing the generalized variance of $\mathbf{X}$ with that of $\mathbf{Z}$.

Your turn The generalized variance of a matrix $\mathbf{Y}$ is defined as $\det(\text{cov}(\mathbf{Y}))$. Compute this quantity for both the original data matrix and the principal components. Are they equal?

```r
# compare generalized variance of Z and X
det(cov(z_mx))
```

```
## [1] 0.00152986
```

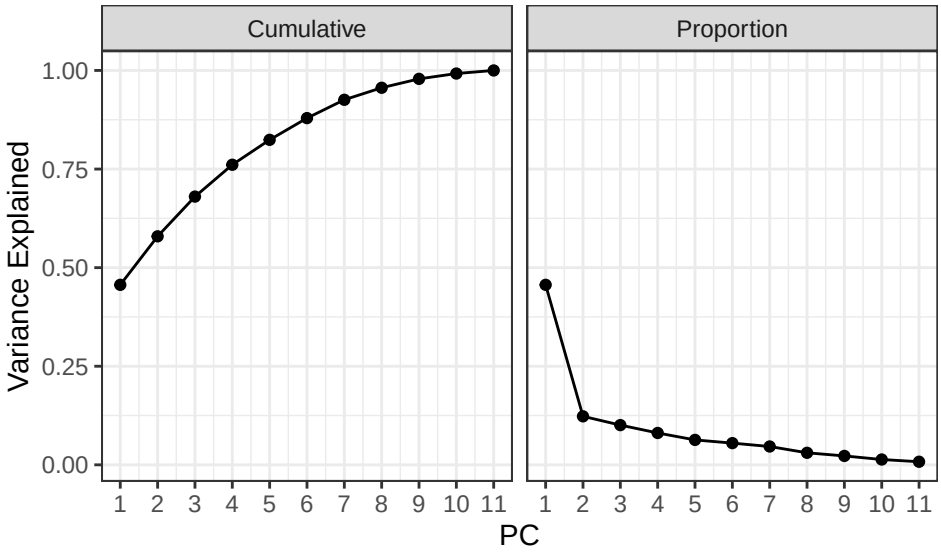
```r
det(cov(x_mx))
```

```
## [1] 0.00152986
```

From this we can see that the generalized variance of Z and X are equal.

Scree and cumulative variance plots

The number of PC's is typically chosen by inspecting plots of the proportion of variance explained by each PC (the 'scree' plot) and the cumulative variance explained by the first q PC's. These two plots are shown below.

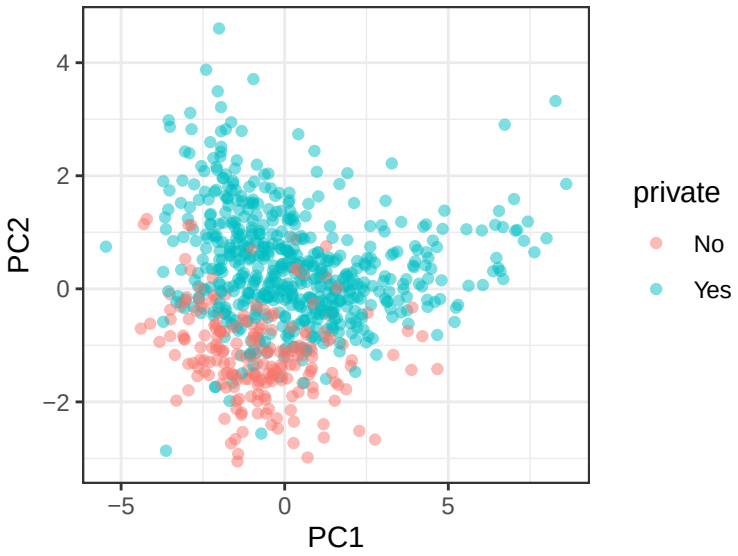


Your turn How many principal components are needed to capture at least 75% of the total variance?

About 4 principal components are needed to capture at least 75% of the total variance.

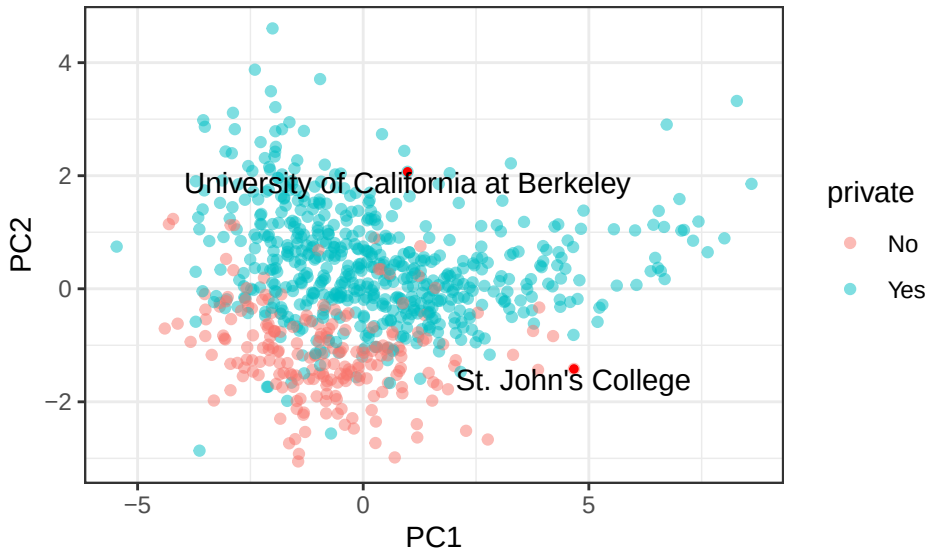
Visualizations based on PCA

When PCA is used as a visualization tool, usually the first two PCs are used to generate a 2D scatterplot of the data. The plot below shows each college plotted according to the value of the first two PC's, with the color aesthetic distinguishing private from public colleges.

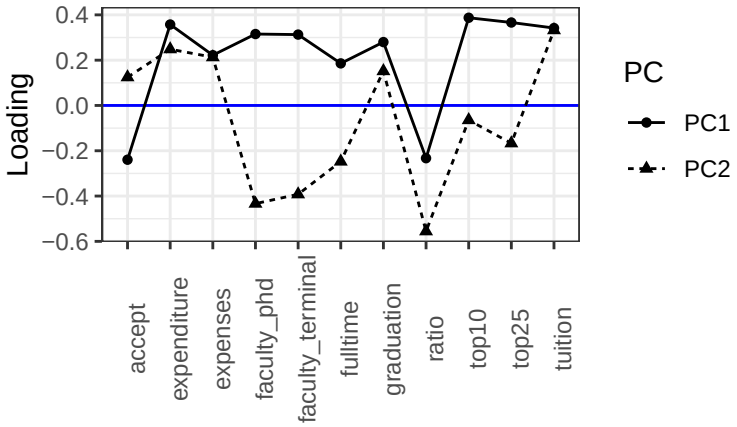


Your turn Add points to the plot above highlighting UC Berkeley and St. John's College with text labels (hint: create a separate tibble with just those two colleges, and add a `geom_point` and `geom_text` layer using that separate tibble for the data argument with appropriate aesthetics.)

```
colleges <- c("University of California at Berkeley", "St. John's College")

colleges2 <- z_mx[which(college$college %in% colleges), 1:2] %>%
  as.data.frame(row.names = colleges) %>%
  rename(PC1 = V1, PC2 = V2)
z_mx[, 1:2] %>%
  as.data.frame() %>%
  rename(PC1 = V1, PC2 = V2) %>%
  bind_cols(select(college, college, private)) %>%
  ggplot(aes(x = PC1, y = PC2)) +
  geom_point(aes(color = private), alpha = 0.5) +
  geom_point(data = colleges2, size = 1, col = "red") +
  geom_text(data = colleges2, label = row.names(colleges2), vjust = 1) +
  theme_bw()
```



Interpreting the plot requires examining and attempting to interpret the PC loadings. A loading plot is shown below for the first two PCs.



The central question in interpreting the loadings is: 'what combinations of high/low variable values produce a large PC value?' For example, look at PC1: PC1 will be large and positive whenever the acceptance rate and student-faculty ratio are low and the remaining variables – expenses, expenditure, faculty degrees, percentage of fulltime students, graduation rate, tuition, and students from the top 10% and 25% of their classes – are high. This seems to describe selective and expensive traditional schools with strong faculty and small class

sizes.

Your turn Interpret PC2 as best you can by considering which variable combinations produce a large *negative* PC value.

PC2 will be large and negative whenever the faculty degrees and student-faculty ratio are low and the remaining variables – acceptance rate, expenses, expenditure, graduation rate, tuition, percentage of full time students,terminal faculty, and students from the top 10% and 25% of their classes – are high.

Classification using PCs (all your turn)

One potential application of PCA is creating derived variables of lower dimension to use as inputs to supervised methods. Carry out the following steps:

1. Decide on a number of principal components to use based on the foregoing.

I will use 5 principal components going forward because the decrease in variance becomes more stable after this point.

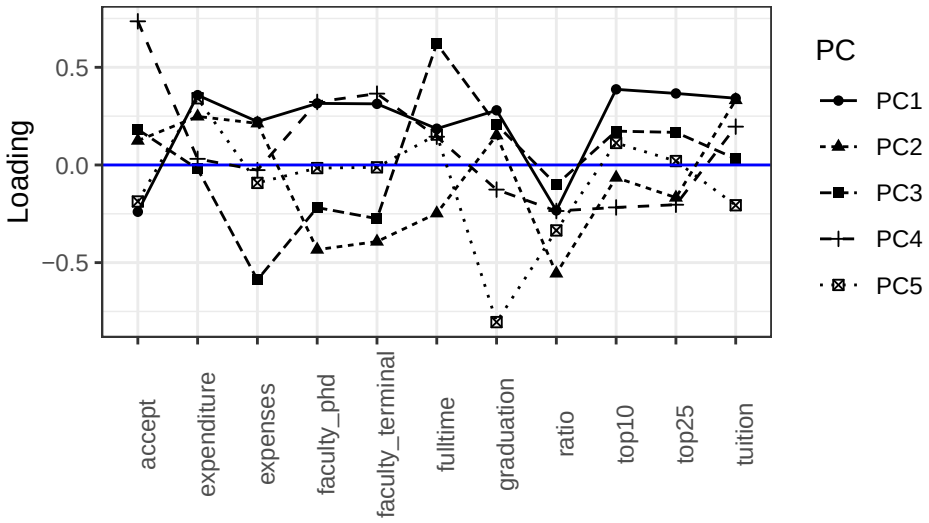
2. Fit a logistic regression model to predict college type (public/private) based on your chosen number of PCs, and check the classification accuracy.

```
new_pcs <- z_mx[,1:5] %>%
  as.data.frame() %>%
  rename(PC1 = V1, PC2 = V2, PC3 = V3, PC4 = V4, PC5 = V5) %>%
  bind_cols(select(college,college,private))

pcs_model <- glm(private ~ PC1+PC2+PC3+PC4+PC5, family = 'binomial', data = new_pcs)

predicted <- predict(pcs_model, new_pcs, type = 'response')
predicted_vals <- factor(predicted > 0.50, labels = c('No', 'Yes'))

errors <- table(class=college$private, pred = predicted_vals)
errors/rowSums(errors)
```

```
##      pred
## class         No        Yes
##   No  0.75000000 0.25000000
##   Yes 0.06017699 0.93982301
```

3. Plot the loadings for each principal component included in your classifier. Do any of the PC's seem to have clear interpretations?

```
v_svd[, 1:5] %>%
  as.data.frame() %>%
  rename(PC1 = V1, PC2 = V2, PC3 = V3, PC4 = V4, PC5 = V5) %>%
  mutate(variable = colnames(x_mx)) %>%
  gather(key = 'PC', value = 'Loading', 1:5) %>%
  arrange(variable) %>%
  ggplot(aes(x = variable, y = Loading)) +
  geom_point(aes(shape = PC)) +
  theme_bw() +
  geom_hline(yintercept = 0, color = 'blue') +
  geom_path(aes(linetype = PC, group = PC)) +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(x = '')
```



4. (Optional) Based on the PC interpretations and model coefficients, what seems to most distinguish private from public colleges?