Principal Component Analysis

Nate Wells

Math 243: Stat Learning

November 18th, 2020

Outline

In today's class, we will. . .

- Discuss Principal Component Analysis as an example of unsupervised learning
- Implement PCA in R and interpret PCA in context

Section 1

Principal Component Analysis

Unsupervised Learning

Thus far, we have concerned ourselves with **supervised learning** methods, where we predict the value of a response Y based on the values of the predictors $X_1, \ldots, X_n$.

Unsupervised Learning

Thus far, we have concerned ourselves with **supervised learning** methods, where we predict the value of a response Y based on the values of the predictors $X_1, \ldots, X_n$.

- Ex: Classify whether a particular tumor is malignant or benign based on size and shape.

Unsupervised Learning

Thus far, we have concerned ourselves with **supervised learning** methods, where we predict the value of a response Y based on the values of the predictors $X_1, \dots, X_n$.

- Ex: Classify whether a particular tumor is malignant or benign based on size and shape.

In **unsupervised learning**, we use statistical tools to analyze relationships among several features $X_1, \dots, X_n$ without an associated response variable Y.

Unsupervised Learning

Thus far, we have concerned ourselves with **supervised learning** methods, where we predict the value of a response Y based on the values of the predictors $X_1, \ldots, X_n$.

- Ex: Classify whether a particular tumor is malignant or benign based on size and shape.

In **unsupervised learning**, we use statistical tools to analyze relationships among several features $X_1, \ldots, X_n$ without an associated response variable Y.

- Ex: Investigate patterns in online purchases based on demographic information.

Unsupervised Learning

Thus far, we have concerned ourselves with **supervised learning** methods, where we predict the value of a response Y based on the values of the predictors $X_1, \dots, X_n$.

- Ex: Classify whether a particular tumor is malignant or benign based on size and shape.

In **unsupervised learning**, we use statistical tools to analyze relationships among several features $X_1, \dots, X_n$ without an associated response variable Y.

- Ex: Investigate patterns in online purchases based on demographic information.
- Compared to supervised learning, analysis of unsupervised learning methods tend to be more subjective (since we can't assess accuracy using a response variable)

Unsupervised Learning

Thus far, we have concerned ourselves with **supervised learning** methods, where we predict the value of a response Y based on the values of the predictors $X_1, \ldots, X_n$.

- Ex: Classify whether a particular tumor is malignant or benign based on size and shape.

In **unsupervised learning**, we use statistical tools to analyze relationships among several features $X_1, \ldots, X_n$ without an associated response variable Y.

- Ex: Investigate patterns in online purchases based on demographic information.
- Compared to supervised learning, analysis of unsupervised learning methods tend to be more subjective (since we can't assess accuracy using a response variable)
- But unsupervised learning represents an instrumental part of exploratory data analysis (and of pattern recognition, more generally)

PCA

To compute the principal components $Z_1, Z_2, \ldots, Z_p$ on a data set with variables $X_1, \ldots, X_p$, we do **not** ever use the values of a response variable Y.

PCA

To compute the principal components $Z_1, Z_2, \ldots, Z_p$ on a data set with variables $X_1, \ldots, X_p$, we do **not** ever use the values of a response variable Y.

- Although for Principal Component Regression, we did later use those principal components to make predictions about Y.

PCA

To compute the principal components $Z_1, Z_2, \ldots, Z_p$ on a data set with variables $X_1, \ldots, X_p$, we do **not** ever use the values of a response variable Y.

- Although for Principal Component Regression, we did later use those principal components to make predictions about Y.

PCA can be used as a means of unsupervised learning and exploratory data analysis.

PCA

To compute the principal components $Z_1, Z_2, \ldots, Z_p$ on a data set with variables $X_1, \ldots, X_p$, we do **not** ever use the values of a response variable Y.

- Although for Principal Component Regression, we did later use those principal components to make predictions about Y.

PCA can be used as a means of unsupervised learning and exploratory data analysis.

- PCA finds the consecutive linear combinations of predictors (or features) that have the most variance, once prior linear combinations are taken into account.

PCA

The first principal component of $X_1, \ldots, X_p$ is the normalized linear combination

$$Z_1 = \phi_{11} X_1 + \cdots + \phi_{p1} X_p \qquad \text{with } \sum \phi_{i1}^2 = 1$$

PCA

The first principal component of $X_1, \dots, X_p$ is the normalized linear combination

$$Z_1 = \phi_{11} X_1 + \cdots + \phi_{p1} X_p \qquad \text{with} \sum \phi_{i1}^2 = 1$$

- The vector $\phi_1 = \begin{pmatrix} \phi_{11} \\ \vdots \\ \phi_{p1} \end{pmatrix}$ is called the loading (and entries ϕ_{i1} the loading of X_i)

PCA

The first principal component of $X_1, \dots, X_p$ is the normalized linear combination

$$Z_1 = \phi_{11}X_1 + \cdots + \phi_{p1}X_p \qquad \text{with } \sum \phi_{i1}^2 = 1$$

- The vector $\phi_1 = \begin{pmatrix} \phi_{11} \\ \vdots \\ \phi_{p1} \end{pmatrix}$ is called the loading (and entries ϕ_{i1} the loading of X_i)
- The first principal component loading vector solves the optimization problem:

$$\phi_1 = \text{argmax}_{\phi_{11},\dots,\phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \text{ given } \sum_{j=1}^{p} \phi_{ji}^2 = 1$$

PCA

The first principal component of $X_1, \ldots, X_p$ is the normalized linear combination

$$Z_1 = \phi_{11} X_1 + \cdots + \phi_{p1} X_p \qquad \text{with} \sum \phi_{i1}^2 = 1$$

- The vector $\phi_1 = \begin{pmatrix} \phi_{11} \\ \vdots \\ \phi_{p1} \end{pmatrix}$ is called the loading (and entries ϕ_{i1} the loading of X_i)
- The first principal component loading vector solves the optimization problem:

$$\phi_1 = \text{argmax}_{\phi_{11}, \ldots, \phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \text{ given } \sum_{j=1}^{p} \phi_{ji}^2 = 1$$

- The vector of loadings $\phi_1 \in \mathbb{R}$ points in the direction in feature space along which the data varies the most.

PCA

The second principal component Z_2 is the linear combination of $X_1, \ldots, X_p$ that has maximal variance among all lin. combos. that are uncorrelated with Z_1, and takes the form

$$Z_2 = \phi_{12}X_1 + \cdots + \phi_{p2}X_p \qquad \text{with } \sum \phi_{i1}^2 = 1 \text{ and } \mathrm{Corr}(Z_1, Z_2) = 0$$

PCA

The second principal component Z_2 is the linear combination of $X_1, \dots, X_p$ that has maximal variance among all lin. combos. that are uncorrelated with Z_1, and takes the form

$$Z_2 = \phi_{12}X_1 + \cdots + \phi_{p2}X_p \qquad \text{with } \sum \phi_{i1}^2 = 1 \text{ and } \mathrm{Corr}(Z_1, Z_2) = 0$$

- Z_2 can also be obtained by projecting all observations onto the hyperplane perpendicular to ϕ_1 and finding the 1st principal component of the resulting data set.

PCA

The second principal component Z_2 is the linear combination of $X_1, \ldots, X_p$ that has maximal variance among all lin. combos. that are uncorrelated with Z_1, and takes the form

$$Z_2 = \phi_{12}X_1 + \cdots + \phi_{p2}X_p \qquad \text{with } \sum \phi_{i1}^2 = 1 \text{ and } \mathrm{Corr}(Z_1, Z_2) = 0$$

- Z_2 can also be obtained by projecting all observations onto the hyperplane perpendicular to ϕ_1 and finding the 1st principal component of the resulting data set.

In general, the kth principal component is a linear combination that has maximal variance among all combos that are uncorrelated with $Z_1, \ldots, Z_{k-1}$

$$Z_k = \phi_{1k}X_1 + \cdots + \phi_{pk}X_p \qquad \text{with } \sum \phi_{i1}^2 = 1 \text{ and } \mathrm{Corr}(Z_j, Z_2) = 0,\ 1 \leq j \leq k-1$$

Alternate Geometric Perspective

Perspective 1: Principal components are directions in feature space along which data vary the most.

Alternate Geometric Perspective

Perspective 1: Principal components are directions in feature space along which data vary the most.

Perspective 2: The first M principal components are the best M-dimensional approximation to the p-dimensional data set.

Alternate Geometric Perspective

Perspective 1: Principal components are directions in feature space along which data vary the most.

Perspective 2: The first M principal components are the best M-dimensional approximation to the p-dimensional data set.

- Observe that the loading vector ϕ_1 is the line in p-dim space that is *closest* to the n observations in the data set.

Alternate Geometric Perspective

Perspective 1: Principal components are directions in feature space along which data vary the most.

Perspective 2: The first M principal components are the best M-dimensional approximation to the p-dimensional data set.

- Observe that the loading vector ϕ_1 is the line in p-dim space that is *closest* to the n observations in the data set.
- Together, the loading vectors ϕ_1, ϕ_2 generate a plane in p-dim space that is closest to the n observations

Alternate Geometric Perspective

Perspective 1: Principal components are directions in feature space along which data vary the most.

Perspective 2: The first M principal components are the best M-dimensional approximation to the p-dimensional data set.

- Observe that the loading vector ϕ_1 is the line in p-dim space that is *closest* to the n observations in the data set.
- Together, the loading vectors ϕ_1, ϕ_2 generate a plane in p-dim space that is closest to the n observations
- Generally, the first M loading vectors $\phi_1, \ldots, \phi_p$ generate an M-dimensional hyperplane in p-dim space that is closest to the n observations.

Alternate Geometric Perspective

Perspective 1: Principal components are directions in feature space along which data vary the most.

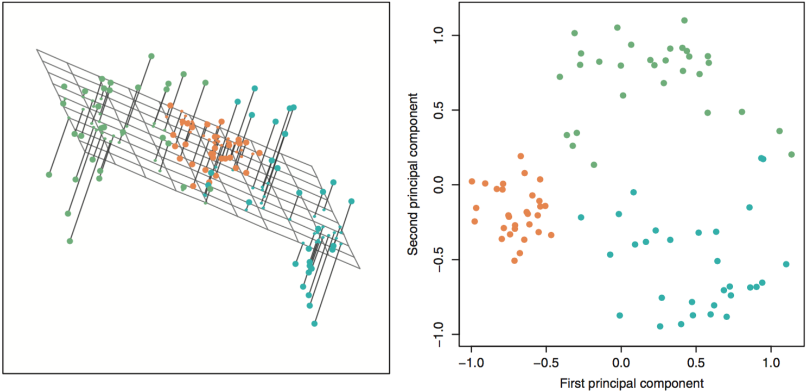
Perspective 2: The first M principal components are the best M-dimensional approximation to the p-dimensional data set.

- Observe that the loading vector ϕ_1 is the line in p-dim space that is *closest* to the n observations in the data set.
- Together, the loading vectors ϕ_1, ϕ_2 generate a plane in p-dim space that is closest to the n observations
- Generally, the first M loading vectors $\phi_1, \ldots, \phi_p$ generate an M-dimensional hyperplane in p-dim space that is closest to the n observations.

$$x_{ij} \approx \sum_{m=1}^{M} z_{im}\phi_{jm} \qquad \text{where } z_{im} = \phi_{1m}x_{im} + \cdots + \phi_{pm}x_{ip}$$

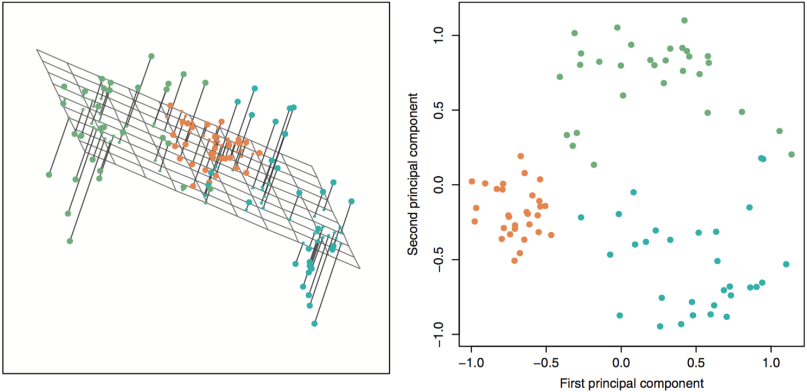
Visual

Reduction from $p = 3$ to $p = 2$ via principal components.



Visual

Reduction from $p = 3$ to $p = 2$ via principal components.



How does this differ from least squares regression?

Properties of PCA

How much information is lost when we project the data set onto the hyperplane spanned by the first M principal component loading vectors?

Properties of PCA

How much information is lost when we project the data set onto the hyperplane spanned by the first M principal component loading vectors?

- The *Total Variance* (TV) of the data set is

$$\mathrm{TV} = \sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$$

Properties of PCA

How much information is lost when we project the data set onto the hyperplane spanned by the first M principal component loading vectors?

- The *Total Variance* (TV) of the data set is

$$\mathrm{TV} = \sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$$

- While the variance explained by the mth principal component V_m is

$$V_m = \frac{1}{n} \sum_{i=1}^{n} z_{im}^2 = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{jm} x_{ij} \right)^2$$

Properties of PCA

How much information is lost when we project the data set onto the hyperplane spanned by the first M principal component loading vectors?

- The *Total Variance* (TV) of the data set is

$$\mathrm{TV} = \sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$$

- While the variance explained by the mth principal component V_m is

$$V_m = \frac{1}{n} \sum_{i=1}^{n} z_{im}^2 = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{jm} x_{ij} \right)^2$$

- Thus, the *Proportion of Variance Explained* by the mth principal component PVE_m is

$$\mathrm{PVE}_m = \frac{V_m}{TV} = \frac{\sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{jm} x_{ij} \right)^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2}$$

How many principal components?

In Principal Component Regression, we can use CV to decide how many principal components should be used in a model.

How many principal components?

In Principal Component Regression, we can use CV to decide how many principal components should be used in a model.

- But in unsupervised exploratory analysis, CV is not available (why?)

How many principal components?

In Principal Component Regression, we can use CV to decide how many principal components should be used in a model.

- But in unsupervised exploratory analysis, CV is not available (why?)

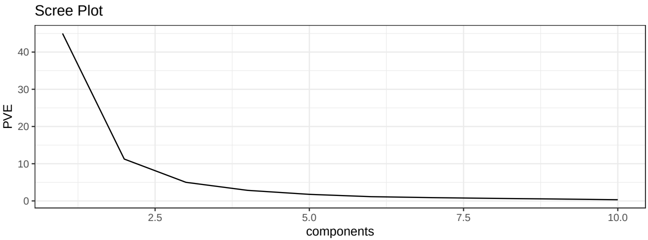
Instead, we can create the *scree plot* of PVE_m versus m and look for the point of diminishing returns (called the *elbow*)

How many principal components?

In Principal Component Regression, we can use CV to decide how many principal components should be used in a model.

- But in unsupervised exploratory analysis, CV is not available (why?)

Instead, we can create the *scree plot* of PVE_m versus m and look for the point of diminishing returns (called the *elbow*)

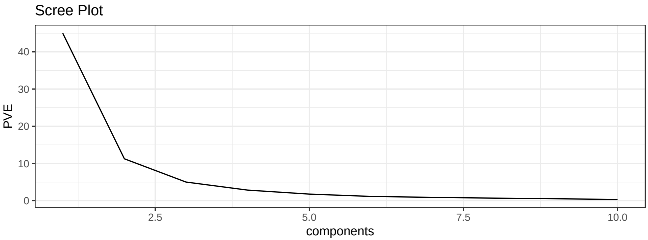


How many principal components?

In Principal Component Regression, we can use CV to decide how many principal components should be used in a model.

- But in unsupervised exploratory analysis, CV is not available (why?)

Instead, we can create the *scree plot* of PVE_m versus m and look for the point of diminishing returns (called the *elbow*)



An alternative is to investigate the data structure present in the first several principal components, and then continue adding further components until the structures of interest no longer change substantially

Section 2

PCA in R

PCA Example

12 perfumers were asked to rate 12 perfumes on 11 scent adjectives

```
##  [1] "spicy"    "heady"    "fruity"   "green"    "vanilla"  "floral"
##  [7] "woody"    "citrus"   "marine"   "greedy"   "oriental"
```

PCA Example

12 perfumers were asked to rate 12 perfumes on 11 scent adjectives

```
##  [1] "spicy"    "heady"    "fruity"   "green"    "vanilla"  "floral"
##  [7] "woody"    "citrus"   "marine"   "greedy"   "oriental"
```

Each was rated on a scale of 1-10, and ratings for each perfume were averaged across experts.

```r
head(experts)
```

```
## # A tibble: 6 x 12
##   perfume spicy heady fruity green vanilla floral woody citrus marine greedy
##   <fct>   <dbl> <dbl>  <dbl> <dbl>   <dbl>  <dbl> <dbl>  <dbl>  <dbl>  <dbl>
## 1 "Angel" 3.22  8.26   1.9   0.133   7.75    2.09 1.05   0.142 0.125   8.28
## 2 "Aroma~ 7.41  8.17   0.575 0.35    1.75    3.71 3.39   0.375 0.0583  0.258
## 3 "Chane~ 3.93  8.42   1.18  0.5     1.73    4.66 1.02   0.6   0.05    0.458
## 4 "Cin\x~ 0.983 2.07   5.2   0.267   4.18    5.32 1.25   0.775 1.02    3.66
## 5 "Coco ~ 0.925 0.717  4.58  1.2     2.02    7.31 1.13   1.17  1.14    2.72
## 6 "J'ado~ 0.108 1.03   6.85  1.62    0.183   8.51 0.925  2.13  1.91    1.47
## # ... with 1 more variable: oriental <dbl>
```

Fitting the PCA

To fit a pca model, we use the `prcomp` function in `Base R`.

```
pca1 <- prcomp(experts[, -1], scale = TRUE)
```

Fitting the PCA

To fit a pca model, we use the `prcomp` function in `Base` R.

```
pca1 <- prcomp(experts[, -1], scale = TRUE)
```

The output of `prcomp` contains a number of useful quantities

```
names(pca1)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

Fitting the PCA

To fit a pca model, we use the `prcomp` function in `Base` R.

```
pca1 <- prcomp(experts[, -1], scale = TRUE)
```

The output of `prcomp` contains a number of useful quantities

```
names(pca1)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

The rotation value contains the principal component loadings

```
kable(pca1$rotation)
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11
spicy	-0.32	-0.31	0.15	-0.10	0.21	0.00	0.29	-0.17	0.12	-0.77	0.00
heady	-0.35	-0.11	0.25	0.16	-0.21	-0.47	0.36	0.48	0.19	0.22	-0.23
fruity	0.34	0.15	-0.36	-0.17	0.26	-0.49	0.17	-0.21	-0.01	-0.07	-0.57
green	0.30	-0.15	0.62	0.27	0.36	0.31	0.05	-0.06	-0.04	0.14	-0.42
vanilla	-0.19	0.51	0.17	-0.28	-0.09	0.17	-0.29	0.40	-0.26	-0.32	-0.38
floral	0.34	-0.20	-0.27	0.07	-0.17	0.28	-0.13	0.39	0.63	-0.22	-0.18
woody	-0.25	-0.37	-0.14	-0.59	0.48	0.15	-0.10	0.22	0.04	0.35	-0.05
citrus	0.33	-0.18	0.38	-0.18	0.07	-0.54	-0.51	0.14	0.04	-0.17	0.28
marine	0.32	-0.08	0.27	-0.61	-0.51	0.12	0.39	-0.13	-0.02	0.06	0.01
greedy	-0.09	0.58	0.23	-0.16	0.26	-0.02	0.09	-0.17	0.65	0.11	0.20
oriental	-0.35	-0.18	0.08	-0.04	-0.35	-0.05	-0.47	-0.51	0.25	0.12	-0.39

Visualize

How can we visualize in R?

Visualize

How can we visualize in R?

- Representing the data set itself requires 11 dimesions.

Visualize

How can we visualize in R?

- Representing the data set itself requires 11 dimesions.
- Representing all pairwise structure requires $\binom{55}{2} = 55$ pairwise scatterplots

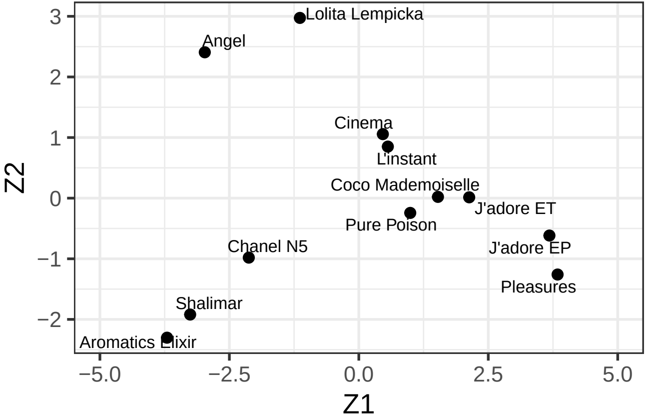
Visualize

How can we visualize in R?

- Representing the data set itself requires 11 dimesions.
- Representing all pairwise structure requires $\binom{55}{2} = 55$ pairwise scatterplots

We can use principal components to focus our attention on small dimensional representation which describes most of the structure.

Scatterplot



Interpretation

Effectively interpreting principal the loading vector for principal components usually requires domain knowledge. But we can try!

Interpretation

Effectively interpreting principal the loading vector for principal components usually requires domain knowledge. But we can try!

What does Z_1 represent? (i.e for what values of x is Z_1 large? small?)

```
##    spicy    heady   fruity    green  vanilla   floral    woody   citrus
##   -0.324   -0.352    0.340    0.304   -0.192    0.344   -0.252    0.330
##   marine   greedy oriental
##    0.322   -0.085   -0.353
```

Interpretation

Effectively interpreting principal the loading vector for principal components usually requires domain knowledge. But we can try!

What does Z_1 represent? (i.e for what values of x is Z_1 large? small?)

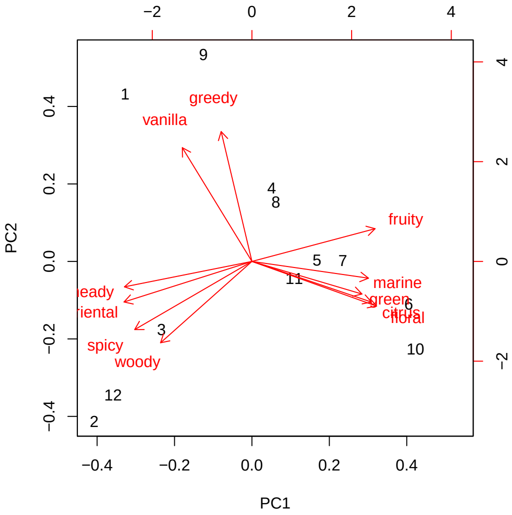
```
##    spicy    heady   fruity    green  vanilla   floral    woody   citrus
##   -0.324   -0.352    0.340    0.304   -0.192    0.344   -0.252    0.330
##   marine   greedy oriental
##    0.322   -0.085   -0.353
```

What does Z_2 represent?

```
##    spicy    heady   fruity    green  vanilla   floral    woody   citrus
##   -0.307   -0.114    0.147   -0.147    0.512   -0.201   -0.366   -0.183
##   marine   greedy oriental
##   -0.075    0.584   -0.182
```

Another Visualization

```
biplot(pca1)
```



Scree Plot

```
d <- data.frame(PC = 1:11, PVE = pca1$sdev^2 / sum(pca1$sdev^2))

ggplot(d, aes(x = PC, y = PVE)) + geom_line() + geom_point() +
  theme_bw(base_size = 18)
```