Week 8. Logistic, Poisson regressions, and PCA

Confidence band for prediction

The true probability is

$$p = \Pr(Y_i = 1|X = x) = \frac{e^{\alpha+\beta x}}{1 + e^{\alpha+\beta x}},$$

and the estimated probability is

$$\widehat{p} = \frac{e^{a+bx}}{1 + e^{a+bx}},$$

where x is where the prediction is made (it may be $x = x_i$ but not necessarily). To find the CI for the probability p we find the CI for $a + bx$ and then use transformation $\exp/(1 + \exp)$ to get the CI for p. We have

$$\begin{bmatrix} a \\ b \end{bmatrix} \simeq \mathcal{N}\left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \mathbf{C} \right)$$

where

$$\mathbf{C}^{2\times 2} = \begin{bmatrix} \sum d_i & \sum d_i x_i \\ \sum d_i x_i & \sum d_i x_i^2 \end{bmatrix}^{-1}$$

$$d_i = \frac{e^{a+bx_i}}{(1 + e^{a+bx_i})^2}.$$

Then

$$a + bx \simeq \mathcal{N}\left(\alpha + \beta x, \mathbf{x}'\mathbf{C}\mathbf{x}\right)$$

where

$$\mathbf{x}^{2\times 1} = \begin{bmatrix} 1 \\ x \end{bmatrix}$$

Thus, the $(1 - \alpha)$ confidence interval for $\alpha + \beta x$ is

$$a + bx \pm Z_{1-\alpha/2}\sqrt{\mathbf{x}'\mathbf{C}\mathbf{x}}$$

and finally

$$\left(\frac{e^{a+bx-Z_{1-\alpha/2}\sqrt{\mathbf{x}'\mathbf{C}\mathbf{x}}}}{1 + e^{a+bx-Z_{1-\alpha/2}\sqrt{\mathbf{x}'\mathbf{C}\mathbf{x}}}}, \frac{e^{a+bx+Z_{1-\alpha/2}\sqrt{\mathbf{x}'\mathbf{C}\mathbf{x}}}}{1 + e^{a+bx+Z_{1-\alpha/2}\sqrt{\mathbf{x}'\mathbf{C}\mathbf{x}}}} \right)$$

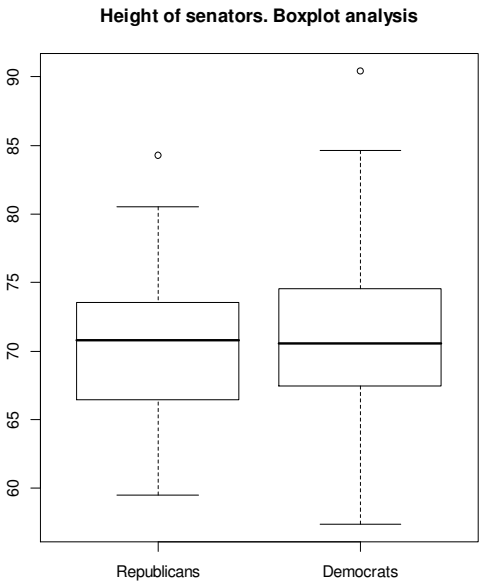
is the confidence band for the prediction.

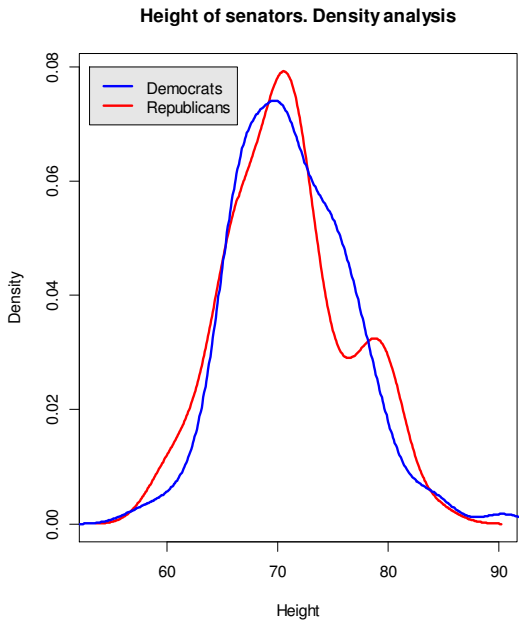
Example 1 *Is it true that democrats are taller than republicans?*

Solution. R code: drLOG, data yx=read.table("c:\\M7021\\RDheight.txt",header=T)
drLOG(job=1)

```
 Welch Two Sample t-test
data:  hr and hd
t = -0.5386, df = 143.646, p-value = 0.591
alternative hypothesis:  true difference in means is not equal to 0
95 percent confidence interval:
 -1.989471 1.137388
sample estimates:
mean of x mean of y
```

```
 70.85833 71.28437
```






```
> drLOG(job=2)
#LOGISTIC REGRESSION=======================================
Call:
glm(formula = y ~x, family = binomial)
Deviance Residuals:
 Min 1Q Median 3Q Max
-1.5201 -1.4088 0.9204 0.9533 1.0261
Coefficients:
 Estimate Std.  Error z value Pr(>|z|)
(Intercept) -0.50835 1.99202 -0.255 0.799
x 0.01525 0.02798 0.545 0.586
(Dispersion parameter for binomial family taken to be 1)
 Null deviance:  261.37 on 199 degrees of freedom
Residual deviance:  261.07 on 198 degrees of freedom
AIC: 265.07
Number of Fisher Scoring iterations:  4
#PROBIT REGRESSION=========================================
Call:
glm(formula = y ~x, family = binomial(probit))
Deviance Residuals:
 Min 1Q Median 3Q Max
-1.5201 -1.4089 0.9205 0.9534 1.0253
Coefficients:
 Estimate Std.  Error z value Pr(>|z|)
(Intercept) -0.305375 1.224271 -0.249 0.803
x 0.009339 0.017181 0.544 0.587
(Dispersion parameter for binomial family taken to be 1)
 Null deviance:  261.37 on 199 degrees of freedom
Residual deviance:  261.07 on 198 degrees of freedom
```

```
AIC: 265.07
Number of Fisher Scoring iterations:  4
```

How to test the validity of GLM

Deviance = -2 times log likelihood function

Null deviance = no predictors, just an intercept term

To test the validity we test that all slopes are zero: $H_0 : \boldsymbol{\beta} = \mathbf{0}$.

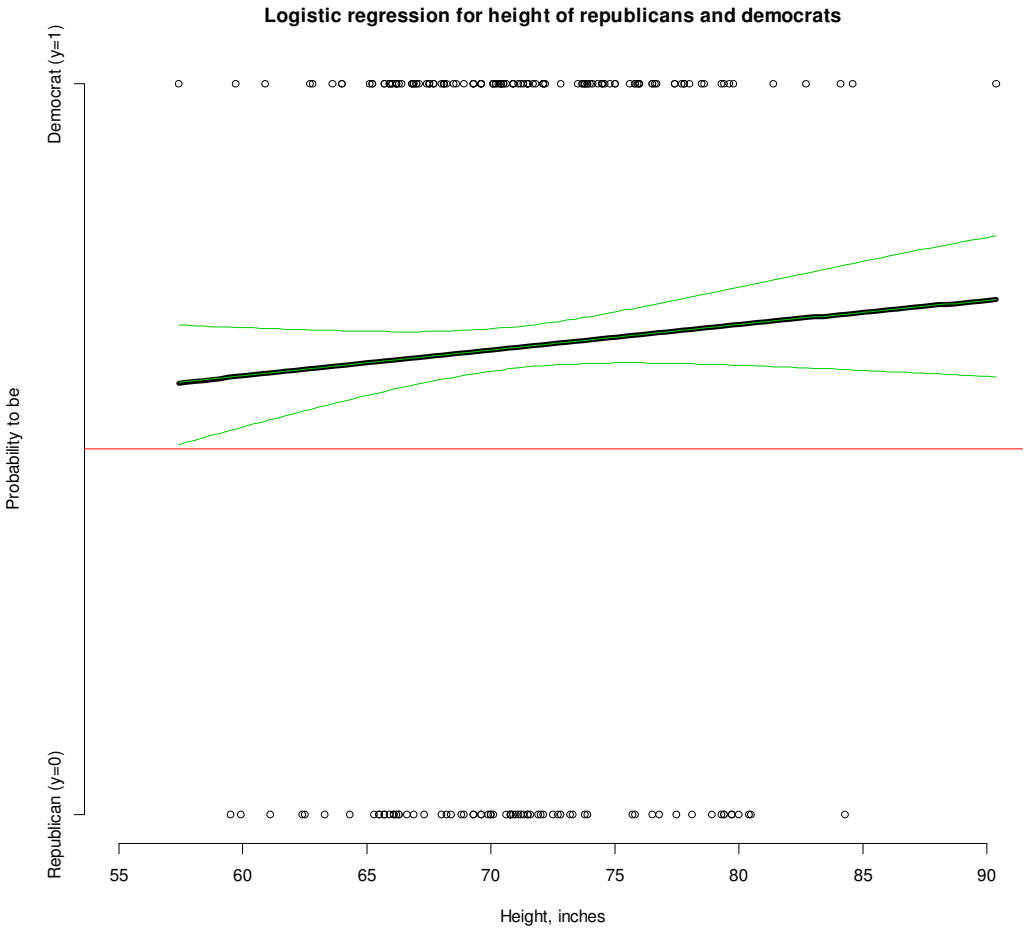
According to the LR test, if H_0 holds,

$$\text{Null deviance - Deviance } \simeq \chi^2(m).$$

The p-value:

$$p = \text{pchisq(261.37 - 261.07,df=1,lower.tail=F)} = 0.5838824$$

The height is not statistically significant.



Logistic regression for discrimination of two groups

Supervised learning: number of observations (n) > dimension (m)

Group 1: $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_{n_1} \in R^m$

Group 2: $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{n_2} \in R^m$

Assumption: $m < n_1 + n_2$.

Combine the data into a $(n_1+n_2)\times m$ matrix $(n_1+n_2=n)$:

$$\mathbf{X}^{n\times m}=\begin{bmatrix}\mathbf{u}_1\\ \vdots\\ \mathbf{u}_{n_1}\\ \mathbf{v}_1\\ \vdots\\ \mathbf{v}_{n_2}\end{bmatrix}=\begin{bmatrix}\mathbf{x}_1\\ \vdots\\ \mathbf{x}_{n_1}\\ \mathbf{x}_{n_1+1}\\ \vdots\\ \mathbf{x}_n\end{bmatrix}$$

and create the indicator variable

$$y_i=\begin{cases}1 \text{ if } i \text{ belongs to Group 1 (cancer)}\\ 0 \text{ if } i \text{ belongs to Group 2 (normal)}\end{cases}, \quad i=1,2,..,n.$$

Run the logistic regression

$$\Pr(y_i=1)=\frac{\exp(\beta_0+\boldsymbol{\beta}'\mathbf{x}_i)}{1+\exp(\beta_0+\boldsymbol{\beta}'\mathbf{x}_i)}.$$

The discrimination rule:

$$\begin{aligned}\beta_0+\boldsymbol{\beta}'\mathbf{x}_i &< c: \text{ observation } i \text{ belongs to group 2,}\\ \beta_0+\boldsymbol{\beta}'\mathbf{x}_i &> c: \text{ observation } i \text{ belongs to group 1}\end{aligned}$$

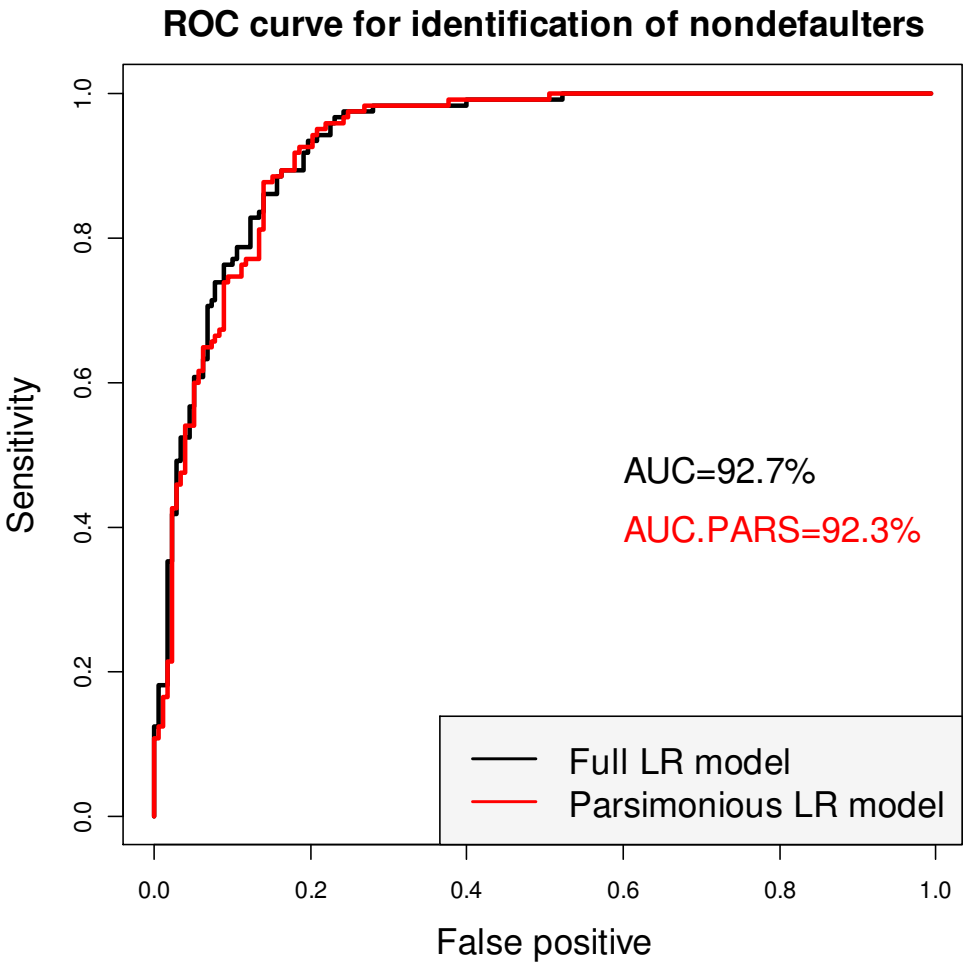
where $-\infty<c<\infty$.

$$\begin{aligned}\text{Sensitivity}(c) &= \frac{\text{number of } y_i=1 \text{ and } \beta_0+\boldsymbol{\beta}'\mathbf{x}_i>c}{\text{number of cancers}=\sum_{i=1}^n y_i}=\text{proportion of identified cancers,}\\ \text{Specificity}(c) &= \frac{\text{number of } y_i=0 \text{ and } \beta_0+\boldsymbol{\beta}'\mathbf{x}_i<c}{\text{number of controls}=\sum_{i=1}^n(1-y_i)}=\text{proportion of identified controls.}\end{aligned}$$

ROC curve: x-axis=1-Specificity(c) and y-axis=Sensitivity(c), where $-\infty<c<\infty$.

Example 2 *Identification of nondefaulters using logistic regression. Estimate LR to identify/classify non-defaullters using the applicant-specific variables age, sex, mariStat, income, yearsW, ownCar, Savings from the file mortgageAPP_DEF.csv. Reduce to parsimonious model and plot the two ROC curves along with the respective AUCs.*

See: mortgAPP()



Poisson regression

Section 8.8.3

Counts follow a Poisson distribution

$$\Pr(Y = y; \lambda) = \frac{\lambda^y}{y!} e^{-\lambda}, \quad y = 0, 1, 2,$$

where y is the observed count and $E(Y) = \lambda$.

Poisson regression is a statistical model for counts with λ_i dependent on a linear combination of predictors, that is, $\boldsymbol{\beta}'\mathbf{x}_i$.

It is convenient to model

$$E(Y) = e^{\boldsymbol{\beta}'\mathbf{x}}$$

because λ is positive. In a special case ($m = 1$) we may have $x_i = 1$ and then $\lambda = e^{\beta}$ – regular Poisson distribution.

Let observed counts $y_1, y_2, ..., y_n$ be independent (n = sample size) and follow Poisson distribution with

$$\lambda_i = e^{\boldsymbol{\beta}'\mathbf{x}_i},$$

where $\mathbf{x}_i^{m\times 1}$ and $\boldsymbol{\beta}^{m\times 1}$.

For any GLM

$$E(Y) = \mu(\boldsymbol{\beta}'\mathbf{x})$$

and μ^{-1}, the inverse function, is called the `link`. The link for Poisson regression is log ("log link"), $\mu^{-1} = \ln$. We have

$$Y_i \simeq \mu^{-1}(\boldsymbol{\beta}'\mathbf{x}_i).$$

This means that running a linear regression with $\ln Y_i$ must be close to Poisson regression. Sometimes use $\ln(1 + Y_i)$.

The likelihood and log-likelihood functions are

$$\begin{aligned} L(\boldsymbol{\beta}) &= \prod_{i=1}^{n} \frac{\lambda_i^{y_i}}{y_i!} e^{-\lambda_i}, \\ l(\boldsymbol{\beta}) &= const + \left(\sum_{i=1}^{n} y_i \mathbf{x}_i\right)' \boldsymbol{\beta} - \sum_{i=1}^{n} e^{\boldsymbol{\beta}'\mathbf{x}_i}. \end{aligned}$$

Let $\mathbf{r} = \sum_{i=1}^{n} y_i \mathbf{x}_i$ and rewrite

$$l(\boldsymbol{\beta}) = \boldsymbol{\beta}'\mathbf{r} - \sum_{i=1}^{n} e^{\boldsymbol{\beta}'\mathbf{x}_i}$$

We have

$$\begin{aligned} \frac{\partial l}{\partial \boldsymbol{\beta}} &= \mathbf{r} - \sum_{i=1}^{n} \mathbf{x}_i e^{\boldsymbol{\beta}'\mathbf{x}_i}, \\ \frac{\partial^2 l}{\partial \boldsymbol{\beta}^2} &= -\sum_{i=1}^{n} e^{\boldsymbol{\beta}'\mathbf{x}_i} \mathbf{x}_i \mathbf{x}_i' \end{aligned}$$

Therefore, the Fisher information matrix is

$$\sum_{i=1}^{n} e^{\boldsymbol{\beta}'\mathbf{x}_i} \mathbf{x}_i \mathbf{x}_i' = \mathbf{X}'\mathbf{E}\mathbf{X}$$

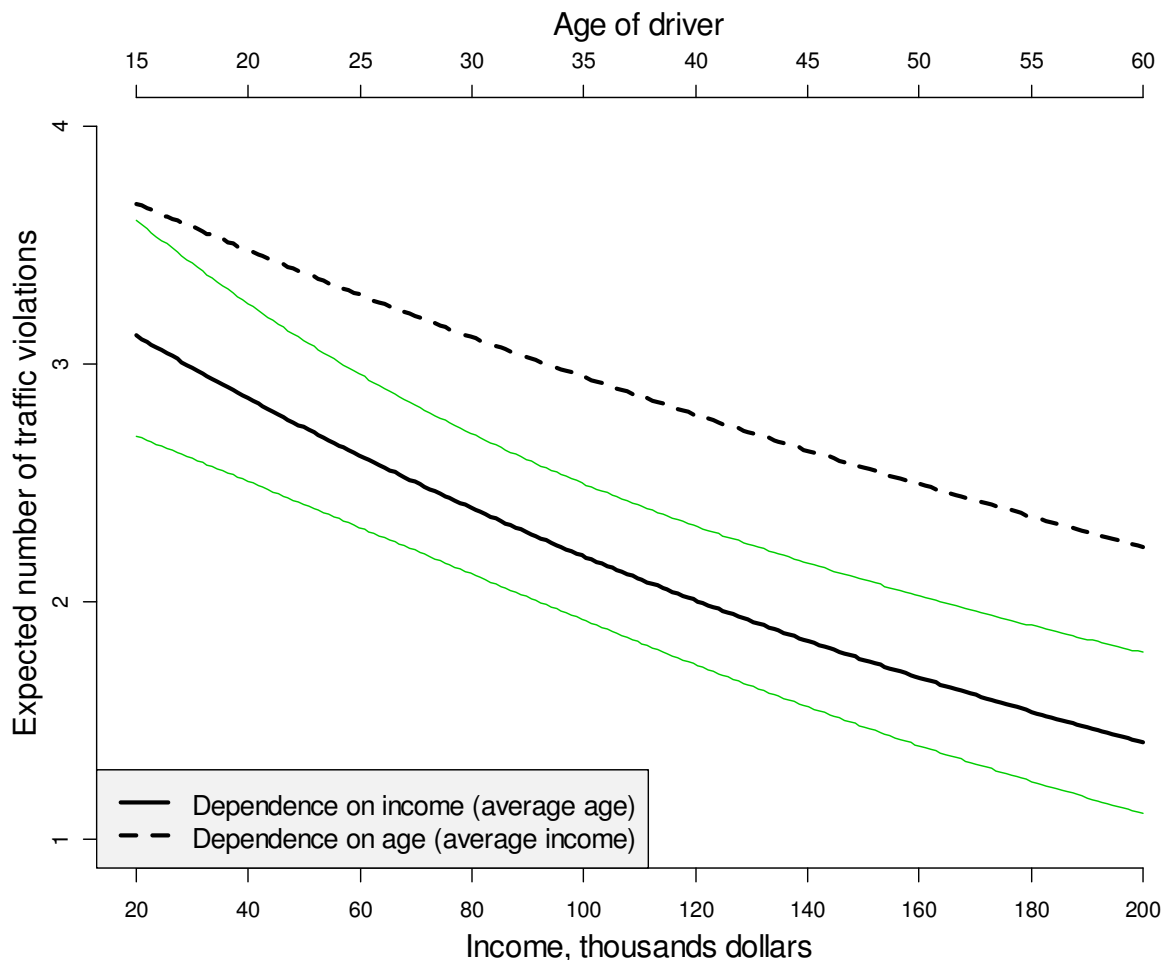
where

$$\mathbf{E}^{n\times n} = \mathbf{E}(\boldsymbol{\beta}) = \begin{bmatrix} e^{\boldsymbol{\beta}'\mathbf{x}_1} & 0 & 0 & 0 \\ 0 & e^{\boldsymbol{\beta}'\mathbf{x}_2} & 0 & 0 \\ 0 & 0 & & 0 \\ 0 & 0 & 0 & e^{\boldsymbol{\beta}'\mathbf{x}_2} \end{bmatrix}.$$

Statistical inference is based on the fact that

$$\widehat{\boldsymbol{\beta}}_{ML} \simeq \mathcal{N}(\boldsymbol{\beta}, (\mathbf{X}'\mathbf{E}\mathbf{X})^{-1})$$

Use: `glm(...,family=poisson)`



Example 3 *The number of traffic violation citation is modeled using Poisson regression*

$$\lambda = e^{\beta_0+\beta_1 marriage+\beta_2 gender+\beta_3 age+\beta_4 income}.$$

See R function `poisR.r`*. Estimate parameters using glm with family=Poisson.*

Solution. (a) Use the R function `vcov` to extract the covariance matrix for five parameters $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$. (b) Check the validity of the Poisson model (report the p-value) by testing that all slope coefficients equal zero. (c) Display the expected number of violations for a married man of the average age as a function of income from \$20K to \$200K, and as a function of age with average income using `axis(side=3)`. (d) Display the 95% confidence band for the expected number of violations.

CB: if $\mathbf{V}^{m\times m}$ is the covariance matrix, $\mathbf{V} = (\mathbf{X}'\mathbf{E}\mathbf{X})^{-1}$ then

$$var(\mathbf{p}'\widehat{\boldsymbol{\beta}}_{ML}) = \mathbf{p}'\mathbf{V}\mathbf{p}.$$

If

$$\mathbf{p} = (1, marriage, gender, age, income)$$

the 95% CB on the linear sclale is

$$\mathbf{p}'\widehat{\boldsymbol{\beta}}_{ML} \pm 1.96\sqrt{\mathbf{p}'\mathbf{V}\mathbf{p}}$$

and on the probability scale

PCA

Principal component analysis (PCA) for dimension reduction in the variable space. PCA projection onto line: using PCA with objects/subjects ranking and application to college admission. PCA projection onto plane and 3D and application to banknote counterfeit. Quality of PCA projection as explained variance. Comparison of logistic regression, LDA, and PCA for binary classification.

R codes: `collegePCA, swiss`

Data: `CollegeAdmData.csv, SwissBankNotes100+100.txt, iris`

The goal of PCA is to project multidimensional data onto the space of fewer dimension: line (1D), plane (2D), or space (3D) for viewing.

Data matrix is given as $\mathbf{X}^{n\times m}$ where n is the number of observations or the number of objects/subjects and m is the number of features/attributes/variables ($m < n$). In the vector representation denote $\mathbf{x}_i$ the vector of features of the ith object. Then the data is the collection of n feature vectors:

$$\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\} \in R^m.$$

When $m > 3$ the data are impossible to see. We want to project the R^m data onto R^k, where $k = 1, 2$, or 3, with minimum distortion.

PCA works for **unsupervised** or **supervised** data. In the case we know the groups/classes we display the n points and color to reflect grouping.

The goal of the PCA is to derive synthetic feature, as a linear combination of m features, that most informatively reflect all features.

The first principal component reflects the most informative synthetic feature, as a linear combination of m features, the second principal component reflects the next most informative synthetic feature, etc.

Projection on the line and its application to ranking

To be specific, we refer to the ith row of $\mathbf{X}$ as the ith subject with m attributes $x_{i1}, x_{i2}, ..., x_{im}$ combined into vector $\mathbf{x}_i$. We want to project the data on line using vector $\mathbf{p}^{m\times 1} = (p_1, p_2, ..., p_m)'$ as a vector of coefficients. Sometimes, coefficients p_j are called loadings. In other words,

$$y_i = \sum_{j=1}^{m} p_j x_{ij}, \quad i = 1, 2, ..., n.$$

We treat vector $\mathbf{y} = (y_1, ..., y_n)'$ as a vector which represents n subjects with a single synthetic attribute (the linear combination of the original attributes). In vector/matrix form

$$\mathbf{y}^{n\times 1} = \mathbf{X}\mathbf{p}^{m\times 1}.$$

There are several ways to arrive at the PCA solution: (a) maximum variance, (b) optimal projection on the line.

(a) Justification of PCA as the maximum variance. First, we derive the PCA solution in the probabilistic framework and then apply it to the data matrix $\mathbf{X}$. Let us assume that $\mathbf{x}$ is a random m-dimensional vector with the $m \times m$ covariance matrix $\mathbf{W}$. We aim to find a one-dimensional random variable Y as a linear combination of $\mathbf{x}$ in the form

$$Y = \mathbf{x}'\mathbf{p} = \sum_{j=1}^{m} p_j x_j,$$

where $p_1, p_2, ..., p_m$ are the coefficients to be determined. Among all $\mathbf{p}$ we want to find such vector that makes Y as scattered as possible, i.e. having maximum variance, because otherwise values will be difficult to distinguish and the linear combination is non-informative.

Layman language: given a random vector with known covariance matrix find a linear combination with minimum variance.

Since the variance is unbounded if the norm of $\mathbf{p}$ is not limited we may assume that $\mathbf{p}$ has unit length. One may say that we put Y on the unit scale. In mathematical language we want to solve the following optimization problem:

$$\max_{\|\mathbf{p}\|^2=1} var(Y) = \max_{\|\mathbf{p}\|^2=1} var(\mathbf{x}'\mathbf{p}).$$

Now we use a well known fact that

$$var(\mathbf{x}'\mathbf{p}) = \mathbf{p}'\mathbf{W}\mathbf{p}.$$

Then the problem simplifies to

$$\max_{\|\mathbf{p}\|^2=1} \mathbf{p}'\mathbf{W}\mathbf{p}.$$

As we know from linear algebra $\mathbf{p}'\mathbf{W}\mathbf{p}$ reaches its maximum at $\mathbf{p}$ as the maximum eigenvector

$$\mathbf{p} = \max \text{ eigenvector } \mathbf{W}.$$

This can also be seen from Lagrange multiplier function

$$\mathcal{L}(\mathbf{p}; \lambda) = \mathbf{p}'\mathbf{\Omega}\mathbf{p} - \lambda(\|\mathbf{p}\|^2 - 1).$$

Indeed,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{p}} = 2\mathbf{\Omega}\mathbf{p} - 2\lambda\mathbf{p} = \mathbf{0}$$

and

$$\mathbf{\Omega}\mathbf{p} = \lambda\mathbf{p}.$$

Since

$$\mathbf{p}'\mathbf{\Omega}\mathbf{p} = \lambda \|\mathbf{p}\|^2 = \lambda$$

we pick $\mathbf{p} = \mathbf{p}_{\max}$ the maximum eigenvector with $\lambda = \lambda_{\max}$ as the maximum eigenvalue.

In summary, the maximum eigenvector of matrix $\mathbf{W}$ is the optimal vector of coefficients.

Now we turn our attention to matrix $\mathbf{X}$ treating rows $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ as observation vectors with the same population covariance matrix $\mathbf{\Omega}$. An estimator of this matrix (discrete version of the covariance matrix) is

$$\mathbf{W} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})'.$$

Solution: the first principal component of matrix $\mathbf{X}$ is the linear combination of vector columns $\mathbf{X}\mathbf{p}_{\max}$ = synthetic feature:

$$\mathbf{X} \to \mathbf{X}\mathbf{p}_{\max},$$

where

$$\begin{aligned} \mathbf{p}_{\max} &= \text{maximum eigenvector of the covariance matrix } \mathbf{W},\ \|\mathbf{p}_{\max}\| = 1 \\ \lambda_{\max} &= \max_{\|\mathbf{p}\|^2=1} \mathbf{p}'\mathbf{W}\mathbf{p} > 0 \end{aligned}$$

The larger $\lambda_{\max}$ the better.

Computation in R:

1. `W=var(X)`
2. `eigenW=eigen(W,sym=T)`
3. `p.max=eigenW$vectors[,1]`
4. `lambda.max=eigenW$values[1]`
5. `proj1=(X-rep(1,nrow(X)))%*%t(colMeans(X))%*%p.max`

(b) Justification of PCA as the optimal projection on the line. We want to project points $\{\mathbf{x}_i \in R^m, i = 1, 2, ..., n\}$ onto the straight line in R^m. The straight line is defined as $\{\mathbf{a}+\lambda\mathbf{p}, -\infty < \lambda < \infty\}$, where $\mathbf{a} \in R^m$ is the translation vector and $\mathbf{p} \in R^m$ is the line direction vector. Without loss of generality, we can assume that $\|\mathbf{p}\|^2 = 1$. We want to find **a, p**, and projection coordinates λ_i such that the line represents vectors $\{\mathbf{x}_i, i = 1, 2, ..., n\}$ in the closest way meaning that

$$\min_{\lambda_i, \mathbf{a}, \|\mathbf{p}\|^2=1} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{a} - \lambda_i \mathbf{p}\|^2 .$$

First we eliminate $\mathbf{a}$ when $\lambda_i\mathbf{p}$ is held fixed:

$$\mathbf{a} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i - \lambda\mathbf{p} = \overline{\mathbf{x}} - \lambda\mathbf{p}.$$

This gives

$$\sum_{i=1}^{n} \|(\mathbf{x}_i - \overline{\mathbf{x}}) - \lambda_i \mathbf{p}\|^2 .$$

Now we find optimal λ_i by differentiation:

$$-2((\mathbf{x}_i - \overline{\mathbf{x}}) - \lambda_i \mathbf{p})'\mathbf{p} = 0,$$

which gives

$$\lambda_i = (\mathbf{x}_i - \overline{\mathbf{x}})'\mathbf{p}.$$

Plugging this back into the criterion function we obtain

$$\|(\mathbf{x}_i - \overline{\mathbf{x}}) - \lambda_i \mathbf{p}\|^2 = (\mathbf{x}_i - \mathbf{a})'(\mathbf{I} - \mathbf{p}\mathbf{p}')(\mathbf{x}_i - \mathbf{a}) = \|(\mathbf{x_i} - \overline{\mathbf{x}})\|^{\mathbf{2}} - ((\mathbf{x}_i - \overline{\mathbf{x}})'\mathbf{p})^2.$$

The minimum attains when

$$\max_{\|\mathbf{p}\|^2=1} ((\mathbf{x}_i - \overline{\mathbf{x}})'\mathbf{p})^2 = \max_{\|\mathbf{p}\|^2=1} \mathbf{p}' \left(\sum_{i=1}^{n} (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})' \right) \mathbf{p}$$

This means that $\mathbf{p}$ is the maximum eigenvector of matrix

$$\sum_{i=1}^{n} (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})'.$$

Sometimes $\{p_j, j = 1, ..., m\}$ are called the loadings.

Conclusion 4 *Since the eigenvectors are defined up to ± you can get ranks in ascending or descending order (from worst to best or other way around). Be careful!*

A way to check if the direction of the found maximum eigenvector is correct is to normalize

$$\text{synthetic score} = p_1x_1 + p_2x_2 + ... + p_mx_m.$$

Synthetic score must match common sense (total rank), see below.

Application of PCA to subjects/objects ranking

Trivial ranking based on the total rank: sum the ranks over the features: `rank(y)` or `rank(-y)`

Example:

```
> y=c(3.3,1.3,4.1,2.9)
> rank(y)
[1] 3 1 4 2 #1=min,4=max, from min to max
> rank(-y) #1=max,4=min, from max to min
[1] 2 4 1 3
```

Alternative: 1st principal component PCA: projection onto line

Example 5 *College admission. Five thousand students applied for a prestigious college, but only one thousand can be admitted. Select students for admission based on the data presented in CollegeAdmData.csv.*

Solution. Seven score criteria are used to rank students:

1. HSC=High school curriculum
2. SAT=SAT score
3. CI=College interview
4. OSA=Out of school activity
5. SR=Sport & research programs
6. ES=Essay
7. LR=Letters of recommendation

1st principal component PCA:

$$\text{synthetic score} = p_1 \times \text{HSC} + p_2 \times \text{SAT} + p_3 \times \text{CI} + p_4 \times \text{OSA} + p_5 \times \text{SR} + p_6 \times \text{ES} + p_7 \times \text{LR}$$

```
collegePCA(job=1)
collegePCA(job=2)
collegePCA(job=3)
```

Wrong direction of the max eigenvector, turn by 180 degree

Projection onto plane

Centerize given data matrix $\mathbf{X}$ as

$$\mathbf{Z} = \mathbf{X} - \mathbf{1}\overline{\mathbf{x}}',$$

that is, subtract the means from the columns:

```
X.means=colMeans(X)
Z=X-rep(1,nrow(X))%*%t(X.means)
```

$$[\mathbf{p}_1, \mathbf{p}_2] = \text{first two max eigenvectors } \mathbf{Z}'\mathbf{Z}$$

Projection

$$\mathbf{y}^{n\times 2} = \mathbf{Z}[\mathbf{p}_1, \mathbf{p}_2]$$

Plot $\{(y_{1i}, y_{2i}), i = 1, ..., n\}$
Alternatively, use `var`:
Computation in R:

1. `W=var(X)`
2. `eigenW=eigen(W,sym=T)`
3. `p.max12=eigenW$vectors[,1:2]`
4. `lambda.max12=eigenW$values[1:2]`
5. `Z=X-rep(1:nrow(X))%*%t(colMeans(X))`
6. `proj12=Z%*%p.max12`

Normalized PCA: PCA on the correlation matrix:

$$h_{ij} = \frac{x_{ij} - \overline{x}_j}{SD_j}$$

where

$$SD_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2},$$

the SD of the jth column (row-vectors are iid).
Computation in R for normalized PCA:

1. `R=cor(X)`
2. `eigenW=eigen(R,sym=T)`
3. `p.max12=eigenW$vectors[,1:2]`
4. `lambda.max12=eigenW$values[1:2]`
5. `SD=sqrt(diag(var(X)))`
6. `proj12=((X-rep(1,nrow(X)))%*%t(X.means))/SD)%*%p.max12`

Example: PCA against counterfeit (R function `swiss`) The first half of these measurements are from genuine bank notes, the other half are from counterfeit bank notes.

X1 = length of the bill
X2 = height of the bill (left)
X3 = height of the bill (right)
X4 = distance of the inner frame to the lower border
X5 = distance of the inner frame to the upper border
X6 = length of the diagonal of the central picture.

Flury, B. and Riedwyl, H. (1988). *Multivariate Statistics, A practical Approach*, Cambridge, University Press.

Forged bank notes.
Dataset: SwissBankNotes100+100.txt, R function `swiss`
swiss(job=1)
The maximum eigenvector:
-0.6033324 -0.1339827 -0.1330675 0.5343121 0.5274588 -0.1913882
The projection line in R^6 is defined
X1=-0.6033324$\times\lambda$
X2=-0.1339827$\times\lambda$
...
X6=-0.1913882$\times\lambda$
`swiss(job=1)`
`swiss(job=1.1)`

PCA for binary classification

The maximum eigenvector $\mathbf{p}_1$ defines a plane in R^m

$$\left\{\mathbf{x} \in R^m : (\mathbf{x}-\boldsymbol{\mu})'\mathbf{p}_1 = 0\right\}.$$

This plane separates n points into two groups. We can treat

$$S_i = (\mathbf{x}_i - \boldsymbol{\mu})'\mathbf{p}_1$$

as a synthetic feature with the rule for classification as follows:

$$\begin{array}{l}\text{vector } \mathbf{x}_i \text{ belongs to group 1 if } S_i < 0 \\ \text{vector } \mathbf{x}_i \text{ belongs to group 2 if } S_i > 0\end{array}.$$

This classification algorithm is an **unsupervised** learning technique.

If clases are known, as in supervised learning, the ROC curve can be derived following the rule

$$\begin{array}{l}\text{vector } \mathbf{x}_i \text{ belongs to group 1 if } S_i < c \\ \text{vector } \mathbf{x}_i \text{ belongs to group 2 if } S_i > c\end{array}\ ,$$

where c is the theshold running from $-\infty$ to $+\infty$.. Then PCA becomes a supervised learning algorithm. Warning: since $\mathbf{p}_1$ is defined up to 180°, we must change $\mathbf{p}_1$ for $-\mathbf{p}_1$ if the ROC curve below the 45° line.

Quality of projection

Definition 6 *Let* $\mathbf{Y}$ *be an* $m \times 1$ *random vector with the* $m \times m$ *covariance matrix* $\mathbf{\Omega}$*. The total variance of* $\mathbf{Y}$ *is the sum of component variances,*

$$\sum_{j=1}^{m} var(Y_j) = \sum_{j=1}^{m} \Omega_{jj} = tr(\mathbf{\Omega}).$$

Lemma 7 *For any* $m \times m$ *symmetric matrix* $\mathbf{M}$ *with eigenvalues* λ_j *we have*

$$tr(\mathbf{M}) = \sum_{j=1}^{m} \lambda_j.$$

Proof. Since $\mathbf{M}$ is symmetric we can apply spectral Jordan decomposition

$$\mathbf{M} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}' = \sum_{j=1}^{m} \lambda_j \mathbf{p}_j \mathbf{p}_j'$$

where λ_j is the eigenvalue with the corresponding eigenvector $\mathbf{p}_j$, and

$$\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_m], \quad \mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_m \end{bmatrix}$$

Note that λ_j and $\mathbf{p}_j$ can be arranged is any order, random, ascending, or descending.

Lemma 8 *Let* $\mathbf{Y}_k$ *be the PCA projection of the* $m \times 1$ *random vector* $\mathbf{Y}$ *onto* R^k *where* $k < m$ *such as* $k = 1, 2,$ *or* $3,$ *that is,*

$$\mathbf{Y}_k = \mathbf{P}_k \mathbf{Y},$$

where

$$\mathbf{P}_k = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_k].$$

Then the total variance of $\mathbf{Y}_k$ *is*

$$\sum_{j=1}^{k} \lambda_j$$

where the eigenvalues and eigenvectors are arranged is descending order: $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_k$.

Definition 9 *The variance explained by PCA projection of* R^m *onto* R^k *is defined as*

$$\frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{m} \lambda_j},$$

assuming that eigenvalues and eigenvectors are arranged in descending order: $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_k$.

When projecting on the line the variance explained is

$$\frac{\lambda_{\max}}{\sum_{j=1}^{m}\lambda_j}.$$

For discrete PCA the covariance matrix is estimated as

$$\mathbf{M}^{m\times m}=\frac{1}{n}\mathbf{Z}'\mathbf{Z}.$$

Then for projection onto line $\mathbf{y}=\mathbf{Z}\mathbf{p}_{\max}$ we have

$$\text{Variance explained}_1=\frac{var(\mathbf{y})}{tr(\mathbf{Z}'\mathbf{Z})}=\frac{\mathbf{p}'_{\max}\mathbf{Z}'\mathbf{Z}\mathbf{p}_{\max}}{tr(\mathbf{Z}'\mathbf{Z})}=\frac{\lambda_{\max}}{\sum_{j=1}^{m}\lambda_j},$$

the quality of the projection onto line, the proportion of total variance captured by the 1st component. The quantity

$$\frac{\sum_{j=2}^{m}\lambda_j}{\sum_{j=1}^{m}\lambda_j}$$

is the proportion of the variance leftover (unexplained by PCA).

For projection onto plane we have

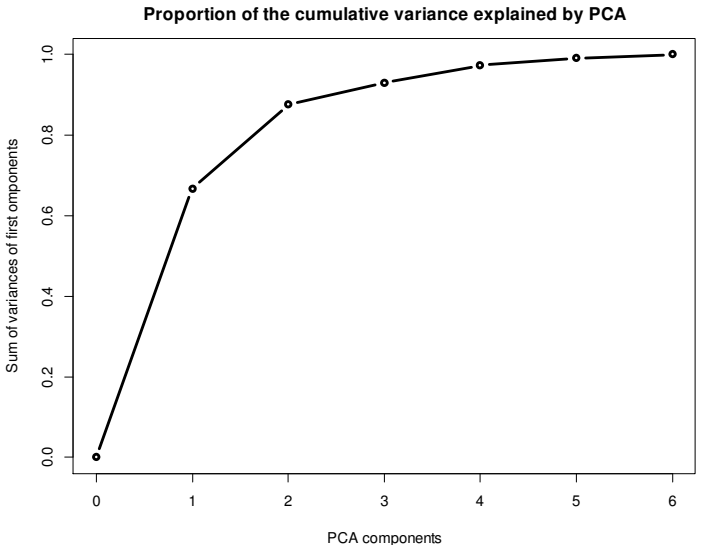
$$\text{Variance explained}_2=\frac{\lambda_1+\lambda_2}{\sum_{j=1}^{m}\lambda_j}$$

as the proportion of the variance quality of the projection onto plane.

In general case

$$\text{Variance explained by projection on the first } k \text{ components}=\frac{\sum_{j=1}^{k}\lambda_j}{\sum_{j=1}^{m}\lambda_j}$$

`swiss(3)`



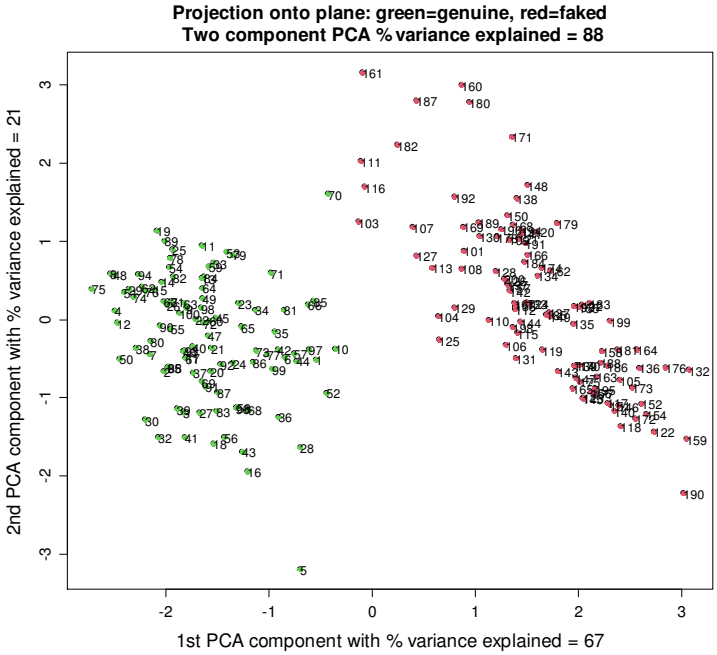
It is a good practice to show % variance explained by the two PCA components as labels at the x- and y-axis separately, that is,

$$\frac{\lambda_1}{\sum_{j=1}^{m}\lambda_j}\times 100\% \text{ and } \frac{\lambda_2}{\sum_{j=1}^{m}\lambda_j}\times 100\%,$$

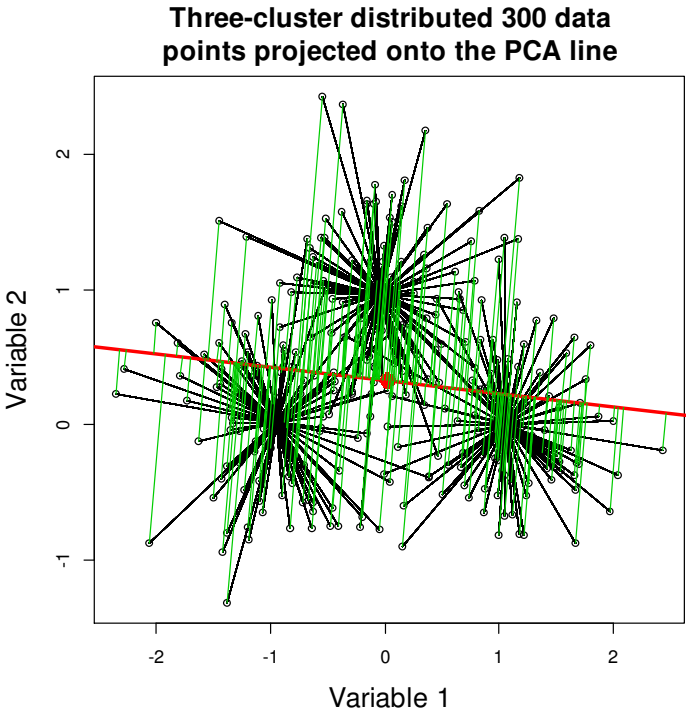
and on the top the total variance explained by the two components, that is

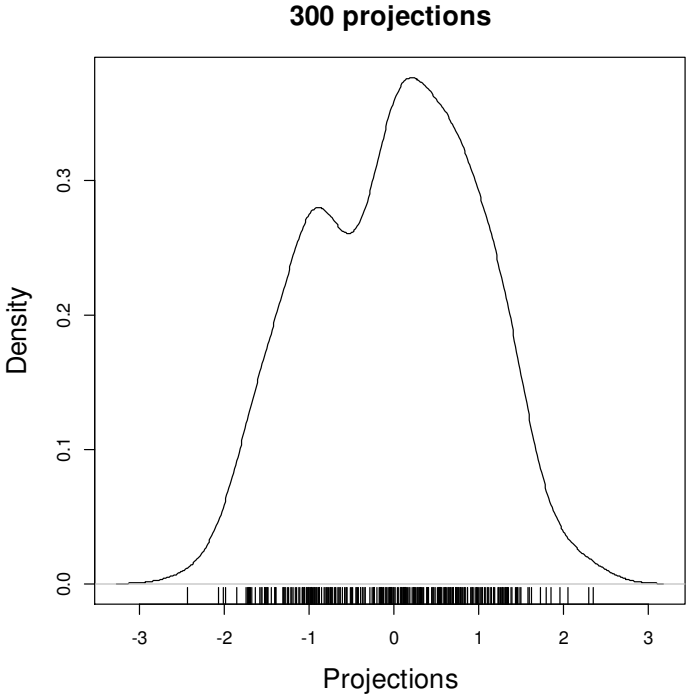
$$\frac{\lambda_1 + \lambda_2}{\sum_{j=1}^{m} \lambda_j} \times 100\%.$$

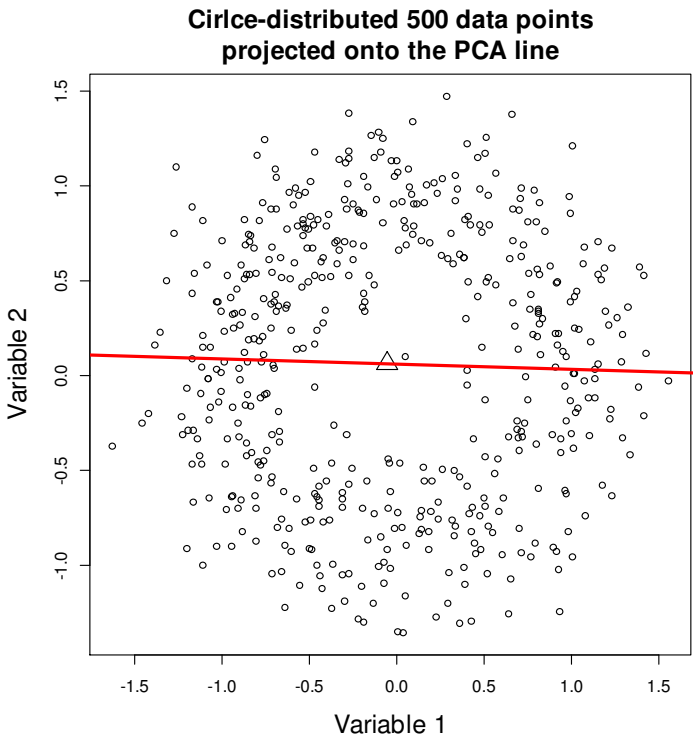
`swiss(job=4)`

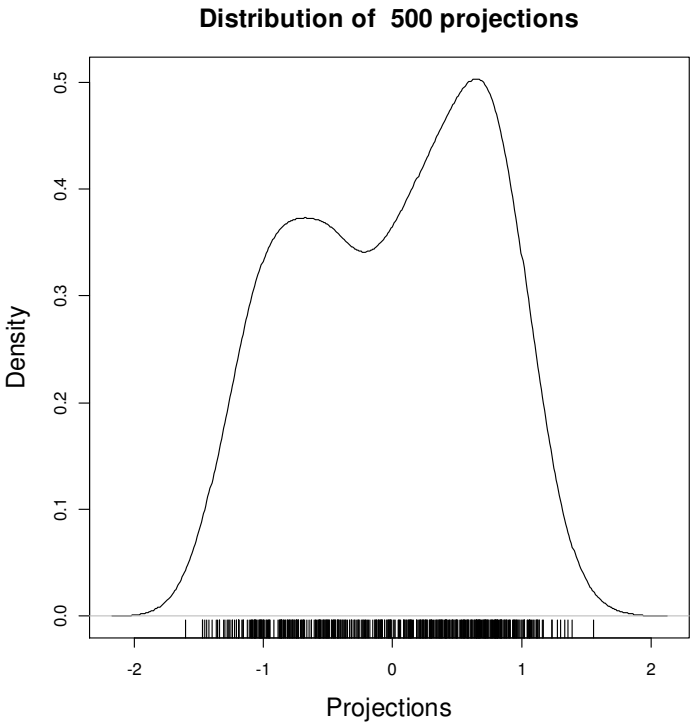


When PCA does not work









Homework 8

1. (15 points). Problem 8.8.10. In addition, (e) Compute and display the ROC curves along with AUCs for identification of an inactive shopper for full and parsimonious model using different color.

2. (15 points). (a) Apply PCA to project the `iris` data onto the plane, display and color each flower. (b) Compute and display on each axis the proportion of variance explained and two components together on the top of the graph. Use `legend` for flower colors. (c) Display the proportion of variance explained by PCA projections as in `swiss(job=3)`.

Solution

1. We know that $\mathbf{x}_i$ are fixed/nonrandom and

$$\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \mathbf{r} - \sum_{i=1}^{n} \frac{\mathrm{e}^{\boldsymbol{\beta}'\mathbf{x}_i}}{1+\mathrm{e}^{\boldsymbol{\beta}'\mathbf{x}_i}}\mathbf{x}_i$$

where $\mathbf{r} = \sum_{i=1}^{n} Y_i\mathbf{x}_i$. We start with showing that

$$I(\boldsymbol{\theta}) = cov\left[\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right],$$

where $I(\boldsymbol{\theta})$ is given by (**??**). Since $\sum_{i=1}^{n} \frac{\mathrm{e}^{\boldsymbol{\beta}'\mathbf{x}_i}}{1+\mathrm{e}^{\boldsymbol{\beta}'\mathbf{x}_i}}\mathbf{x}_i$ is a fixed vector and Y_i are independent we have

$$cov\left[\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right] = cov(\mathbf{r}) = \sum_{i=1}^{n} var(Y_i)\mathbf{x}_i\mathbf{x}_i'.$$

Since Y_i is a Bernoulli random variable we find

$$var(Y_i) = \Pr(Y_i = 1) \times \Pr(Y_i = 0) = \frac{\mathrm{e}^{\boldsymbol{\beta}'\mathbf{x}_i}}{(1+\mathrm{e}^{\boldsymbol{\beta}'\mathbf{x}_i})^2}.$$

Finally,

$$cov\left[\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right] = \sum_{i=1}^{n} \frac{\mathrm{e}^{\boldsymbol{\beta}'\mathbf{x}_i}}{(1+\mathrm{e}^{\boldsymbol{\beta}'\mathbf{x}_i})^2}\mathbf{x}_i\mathbf{x}_i'$$

as we intended to prove.

Now we prove that

$$I(\boldsymbol{\theta}) = E\left[\left(\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right)\left(\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right)'\right] \tag{1}$$

using the fact that

$$E(\mathbf{U}\mathbf{U}') = cov(\mathbf{U}) + E(\mathbf{U})E'(\mathbf{U}).$$

Letting

$$\mathbf{U} = \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$$

and using the fact that

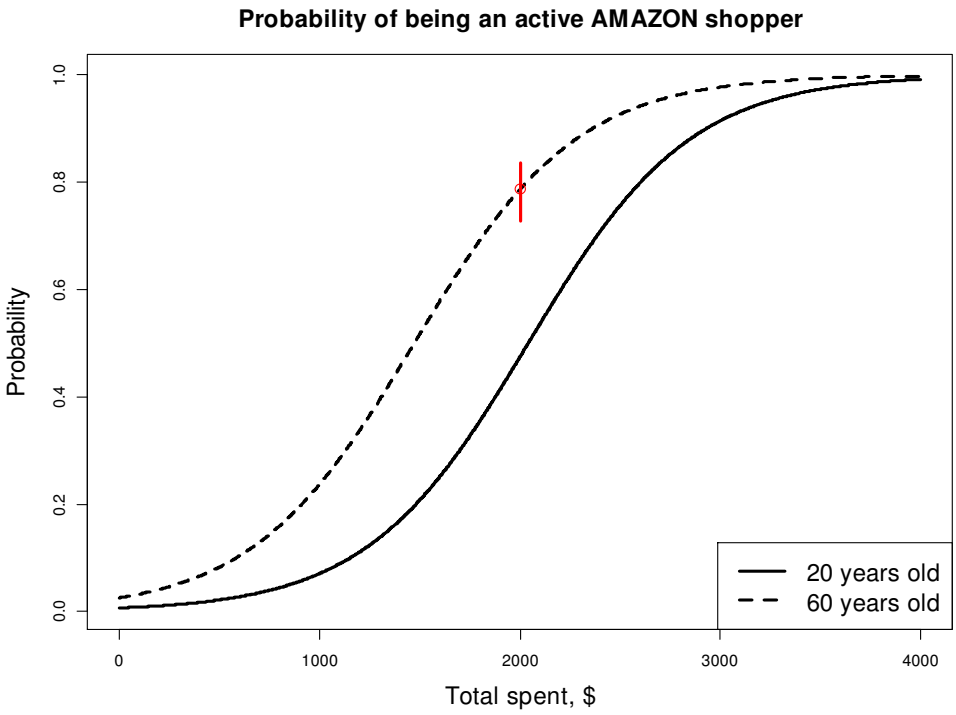
$$E(\mathbf{U}) = E\left(\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right) = \mathbf{0}$$

we arrive at (1).

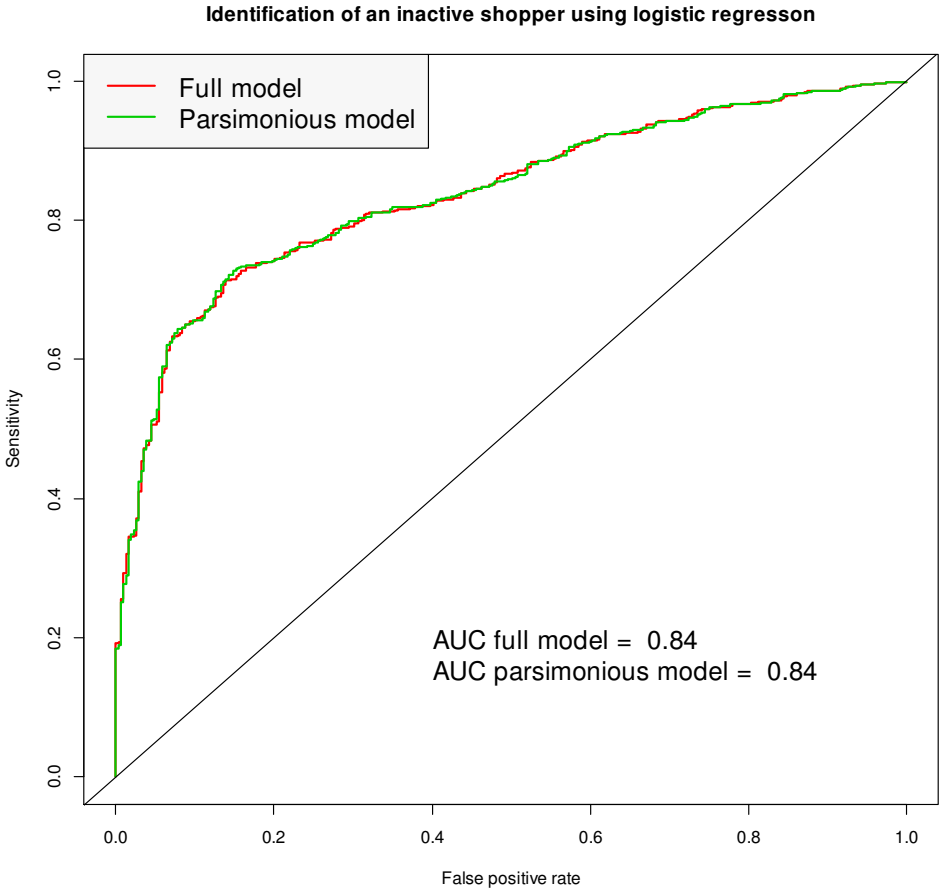
2.

```
> hw7.2()
[1] "Full model:"
Call:
glm(formula = shop ~age + sex + total + npurch, family = binomial,
 data = d)
Deviance Residuals:
 Min 1Q Median 3Q Max
```

```
-3.4774 -0.6796 -0.4417 0.7636 2.2849
Coefficients:
 Estimate Std.  Error z value Pr(>|z|)
(Intercept) -5.7238470 0.4175261 -13.709 < 2e-16 ***
age 0.0353099 0.0056171 6.286 3.25e-10 ***
sex 0.0318172 0.1647212 0.193 0.847
total 0.0024731 0.0001929 12.823 < 2e-16 ***
npurch -0.0075134 0.0161670 -0.465 0.642
---
Signif.  codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'  0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
 Null deviance:  1236.60 on 999 degrees of freedom
Residual deviance:  938.28 on 995 degrees of freedom
AIC: 948.28
Number of Fisher Scoring iterations:  5
[1] "Parsimonious model:"
Call:
glm(formula = shop ~age + total, family = binomial, data = d)
Deviance Residuals:
 Min 1Q Median 3Q Max
-3.4891 -0.6737 -0.4447 0.7641 2.2919
Coefficients:
 Estimate Std.  Error z value Pr(>|z|)
(Intercept) -5.7523797 0.4039476 -14.240 < 2e-16 ***
age 0.0351829 0.0056064 6.275 3.49e-10 ***
total 0.0024735 0.0001928 12.831 < 2e-16 ***
---
Signif.  codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'  0.1 ' ' 1
(Dispersion parameter for binomial family taken to be 1)
 Null deviance:  1236.60 on 999 degrees of freedom
Residual deviance:  938.54 on 997 degrees of freedom
AIC: 944.54
Number of Fisher Scoring iterations:  5
[1] "LHS = 298.056225716662 P-value all coefs are zero = 1.8963257274995e-65"
[1] "Probability CI for 60 yo who spends $2000:  0.727671703035311 0.835943249666849"
```



2.



The R code:
hw7.2 <-function(job=1)

```
{
dump("hw7.2","c:\\M7021\\hw7.2.r")
d=read.csv("c:\\M7021\\amazshop.csv")
if(job==1)
{
#(a)
    print("Full model:")
    o=glm(shop~age+sex+total+npurch,family=binomial,data=d)
    print(summary(o))
    print("Parsimonious model (sex and npurch are eliminated):")
    o=glm(shop~age+total,family=binomial,data=d)
    beta=coef(o)
    covb=vcov(o) #covariance matrix for betas
    so=summary(o)
    print(so)
#(b)
    nul.dev=so[[8]] #null deviance=-2*log-lik of const model = 1236.6
    dev=so[[4]] #residual deviance=-2*log-lik of full model = 938.54
    chi1.obs=nul.dev-dev
    p.val=pchisq(chi1.obs,df=2,lower.tail=F)
    print(paste("LHS =",chi1.obs," P-value all coefs are zero =",p.val))
#(c)
    xtot=0:4000;nx=length(xtot)
    pr=matrix(ncol=2,nrow=nx)
    age=c(20,60)
    for(i in 1:2)
    {
        linp=beta[1]+beta[2]*age[i]+beta[3]*xtot
        pr[,i]=exp(linp)/(1+exp(linp))
    }
    par(mfrow=c(1,1),mar=c(4.5,4.5,4,1),cex.lab=1.5,cex.main=1.5)
    matplot(xtot,pr,type="l",col=1,lwd=3,xlab="Total spent, $",ylab="Probability",
            main="Probability of being an active AMAZON shopper")
    legend("bottomright",c("20 years old","60 years old"),lty=1:2,lwd=3,cex=1.5)
#(d)
    linc=c(1,60,2000)
    lin60=sum(beta*linc)
    se.lin=sqrt(t(linc)%*%covb%*%linc)
    lin60.UP=lin60+qnorm(1-0.05/2)*se.lin
    lin60.LOW=lin60-qnorm(1-0.05/2)*se.lin
    points(2000,exp(lin60)/(1+exp(lin60)),cex=1.5,col=2)
    segments(2000,exp(lin60.LOW)/(1+exp(lin60.LOW)),2000,exp(lin60.UP)/(1+exp(lin60.UP))
            ,col=2,lwd=3)
    print(paste("Probability CI for 60 yo who spends $2000:",exp(lin60.LOW)/(1+exp(lin60.LOW)),
            exp(lin60.UP)/(1+exp(lin60.UP))))
}
if(job==2)#(e) ROC curves
```

```
{
    #full
    full=glm(shop~age+sex+total+npurch,family=binomial,data=d)
    bf=coef(full)
    X=cbind(d$age,d$sex,d$total,d$npurch)
    lp=bf[1]+X%*%bf[2:5]
    lps=sort(lp)
    n=length(lp)
    fp=sens=rep(0,n)
    AUCfull=0
    for(i in 1:n)
    {
        sens[i]=sum((lp<lps[i]) & d$shop==0)/sum(d$shop==0)
        fp[i]=sum((lp<lps[i]) & d$shop==1)/sum(d$shop==1)
        if(i>1) AUCfull=AUCfull+(fp[i]-fp[i-1])*sens[i]
    }
    plot(fp,sens,type="s",col=2,lwd=2,xlab="False positive rate",ylab="Sensitivity")
    title("Identification of an inactive shopper using logistic regresson")
    #"Parsimonious model
    pars=glm(shop~age+total,family=binomial,data=d)
    bf=coef(pars)
    X=cbind(d$age,d$total)
    lp=bf[1]+X%*%bf[2:3]
    lps=sort(lp)
    n=length(lp)
    fp=sens=rep(0,n)
    AUCpars=0
    for(i in 1:n)
    {
        sens[i]=sum((lp<lps[i]) & d$shop==0)/sum(d$shop==0)
        fp[i]=sum((lp<lps[i]) & d$shop==1)/sum(d$shop==1)
        if(i>1) AUCpars=AUCpars+(fp[i]-fp[i-1])*sens[i]
    }
    lines(fp,sens,type="s",col=3,lwd=2)
    segments(-1,-1,2,2)
    text(.4,.2,paste("AUC full model = ",round(AUCfull,2)),adj=0,cex=1.5)
    text(.4,.15,paste("AUC parsimonious model = ",round(AUCpars,2)),adj=0,cex=1.5)
    legend("topleft",c("Full model","Parsimonious model"),col=2:3,lty=1,lwd=2,cex=1.5,
            bg="gray97")
}
}
```