Chapter 4 –Dimension Reduction

Data Mining for Business Analytics in R

Shmueli, Bruce, Yahav, Patel & Lichtendahl

Exploring the data

Statistical summary of data: common metrics

- Average
- Median
- Minimum
- Maximum
- Standard deviation
- Counts & percentages

Summary Statistics for Boston Housing Data

	mean	sd	min	max	median	length	miss.val
CRIM	3.61352356	8.6015451	0.00632	88.9762	0.25651	506	0
ZN	11.36363636	23.3224530	0.00000	100.0000	0.00000	506	0
INDUS	11.13677866	6.8603529	0.46000	27.7400	9.69000	506	0
CHAS	0.06916996	0.2539940	0.00000	1.0000	0.00000	506	0
NOX	0.55469506	0.1158777	0.38500	0.8710	0.53800	506	0
RM	6.28463439	0.7026171	3.56100	8.7800	6.20850	506	0
AGE	68.57490119	28.1488614	2.90000	100.0000	77.50000	506	0
DIS	3.79504269	2.1057101	1.12960	12.1265	3.20745	506	0
RAD	9.54940711	8.7072594	1.00000	24.0000	5.00000	506	0
TAX	408.23715415	168.5371161	187.00000	711.0000	330.00000	506	0
PTRATIO	18.45553360	2.1649455	12.60000	22.0000	19.05000	506	0
LSTAT	12.65306324	7.1410615	1.73000	37.9700	11.36000	506	0
MEDV	22.53280632	9.1971041	5.00000	50.0000	21.20000	506	0
CAT.MEDV	0.16600791	0.3724560	0.00000	1.0000	0.00000	506	0

Correlation Matrix for Boston Housing Data

TABLE 4.4 CORRELATION TABLE FOR BOSTON HOUSING DATA

```
> round(cor(boston.housing.df),2)
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	LSTAT	MEDV	CAT.MEDV
CRIM	1.00	-0.20	0.41	-0.06	0.42	-0.22	0.35	-0.38	0.63	0.58	0.29	0.46	-0.39	-0.15
ZN	-0.20	1.00	-0.53	-0.04	-0.52	0.31	-0.57	0.66	-0.31	-0.31	-0.39	-0.41	0.36	0.37
INDUS	0.41	-0.53	1.00	0.06	0.76	-0.39	0.64	-0.71	0.60	0.72	0.38	0.60	-0.48	-0.37
CHAS	-0.06	-0.04	0.06	1.00	0.09	0.09	0.09	-0.10	-0.01	-0.04	-0.12	-0.05	0.18	0.11
NOX	0.42	-0.52	0.76	0.09	1.00	-0.30	0.73	-0.77	0.61	0.67	0.19	0.59	-0.43	-0.23
RM	-0.22	0.31	-0.39	0.09	-0.30	1.00	-0.24	0.21	-0.21	-0.29	-0.36	-0.61	0.70	0.64
AGE	0.35	-0.57	0.64	0.09	0.73	-0.24	1.00	-0.75	0.46	0.51	0.26	0.60	-0.38	-0.19
DIS	-0.38	0.66	-0.71	-0.10	-0.77	0.21	-0.75	1.00	-0.49	-0.53	-0.23	-0.50	0.25	0.12
RAD	0.63	-0.31	0.60	-0.01	0.61	-0.21	0.46	-0.49	1.00	0.91	0.46	0.49	-0.38	-0.20
TAX	0.58	-0.31	0.72	-0.04	0.67	-0.29	0.51	-0.53	0.91	1.00	0.46	0.54	-0.47	-0.27
PTRATIO	0.29	-0.39	0.38	-0.12	0.19	-0.36	0.26	-0.23	0.46	0.46	1.00	0.37	-0.51	-0.44
LSTAT	0.46	-0.41	0.60	-0.05	0.59	-0.61	0.60	-0.50	0.49	0.54	0.37	1.00	-0.74	-0.47
MEDV	-0.39	0.36	-0.48	0.18	-0.43	0.70	-0.38	0.25	-0.38	-0.47	-0.51	-0.74	1.00	0.79
CAT.MEDV	-0.15	0.37	-0.37	0.11	-0.23	0.64	-0.19	0.12	-0.20	-0.27	-0.44	-0.47	0.79	1.00

Computing Summary Statistics

```
# compute mean, standard dev., min, max, median,
# length, and missing values for all variables

data.frame(mean=sapply(boston.housing.df, mean), +
+ sd=sapply(boston.housing.df, sd), +
+ min=sapply(boston.housing.df, min), +
+ max=sapply(boston.housing.df, max), +
+ median=sapply(boston.housing.df, median), +
+ length=sapply(boston.housing.df, length) +
+ miss.val=sapply(boston.housing.df, function(x)
+ sum(length(which(is.na(x)))))) )
```

Using `table` to tabulate counts

```
> boston.housing.df <- read.csv("BostonHousing.csv")
> table(boston.housing.df$CHAS)

0   1
471 35
```

35 neighborhoods have a CHAS value of "1," i.e. they border the Charles River

Using `aggregate` to tabulate counts using multiple variables

```
# create bins of size 1
boston.housing.df$RM.bin <- .bincode(boston.housing.df$RM, c(1:9))

# compute the average of MEDV by (binned) RM and CHAS
# in aggregate() use the argument by= to define the list of
# aggregating variables,and FUN= as an aggregating function.

aggregate(boston.housing.df$MEDV,
by=list(RM=boston.housing.df$RM.bin,
CHAS=boston.housing.df$CHAS), FUN=mean)
```

```
   RM CHAS  x
1   3   0   25.30000
2   4   0   15.40714
3   5   0   17.2000
4   6   0   21.76917
5   7   0   35.96444
6   8   0   45.70000
7   5   1   22.21818
8   6   1   25.91875
9   7   1   44.06667
10  8   1   35.95000
```

In neighborhoods where houses averaged 3 rooms and did not border the Charles, median value was 25.3 ($000)

Use functions `melt` and `cast` in `reshape` for pivot tables

```
# use melt() to stack a set of columns into a single column of data.
# stack MEDV values for each combination of (binned) RM and CHAS
mlt <- melt(boston.housing.df, id=c("RM.bin", "CHAS"), measure=c("MEDV"))
head(mlt, 5)
```

output:

```
 RM.bin CHAS variable value
1    6    0    MEDV   24.0
2    6    0    MEDV   21.6
3    7    0    MEDV   34.7
4    6    0    MEDV   33.4
5    7    0    MEDV   36.2
```

```
# use cast() to reshape data and generate pivot table
cast(mlt, RM.bin ~ CHAS, subset=variable=="MEDV",
margins=c("grand_row", "grand_col"), mean)
```

```
 RM.bin   0         1          (all)
1  3  25.30000    NaN        25.30000
2  4  15.40714    NaN        15.40714
3  5  17.20000    22.21818  17.55159
4  6  21.76917    25.91875  22.01599
5  7  35.96444    44.06667  36.91765
6  8  45.70000    35.95000  44.20000
7  (all) 22.09384 28.44000  22.53281
```

Reducing Categories

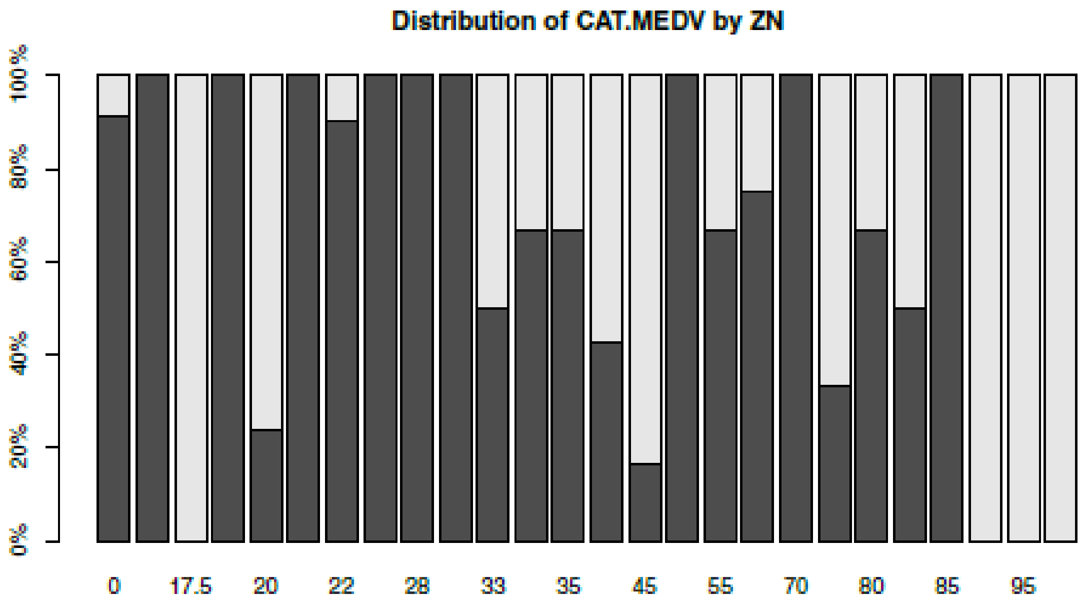
- A single categorical variable with *m* categories is typically transformed into *m* or *m-1* dummy variables (handled automatically by most R modeling functions
- Each dummy variable takes the values 0 or 1

  0 = “no” for the category

  1 = “yes”
- Problem: Can end up with too many variables
- Solution: Reduce by combining categories that are close to each other
- Use pivot tables to assess outcome variable sensitivity to the dummies
- Exception: Naïve Bayes can handle categorical variables without transforming them into dummies

Combining Categories

Many zoning categories are the same or similar with respect to CATMEDV



Principal Components Analysis

Goal: Reduce a set of numerical variables.

The idea: Remove the overlap of information between these variable. ["Information" is measured by the sum of the variances of the variables.]

Final product: A smaller number of numerical variables that contain most of the information

Principal Components Analysis

How does PCA do this?

- Create new variables that are linear combinations of the original variables (i.e., they are weighted averages of the original variables).

- These linear combinations are uncorrelated (no information overlap), and only a few of them contain most of the original information.

- The new variables are called *principal components*.

Example – Breakfast Cereals (excerpt)

name	mfr	type	calories	protein	…	rating
100%_Bran	N	C	70	4	…	68
100%_Natural_Bran	Q	C	120	3	…	34
All-Bran	K	C	70	4	…	59
All-Bran_with_Extra_Fiber	K	C	50	4	…	94
Almond_Delight	R	C	110	2	…	34
Apple_Cinnamon_Cheerios	G	C	110	2	…	30
Apple_Jacks	K	C	110	2	…	33
Basic_4	G	C	130	3	…	37
Bran_Chex	R	C	90	2	…	49
Bran_Flakes	P	C	90	3	…	53
Cap'n'Crunch	Q	C	120	1	…	18
Cheerios	G	C	110	6	…	51
Cinnamon_Toast_Crunch	G	C	120	1	…	20

Description of Variables

Name: name of cereal

mfr: manufacturer

type: cold or hot

calories: calories per serving

protein: grams

fat: grams

sodium: mg.

fiber: grams

carbo: grams complex carbohydrates

sugars: grams

potass: mg.

vitamins: % FDA rec

shelf: display shelf

weight: oz. 1 serving

cups: in one serving

rating: consumer reports

Consider calories & ratings covariance matrix

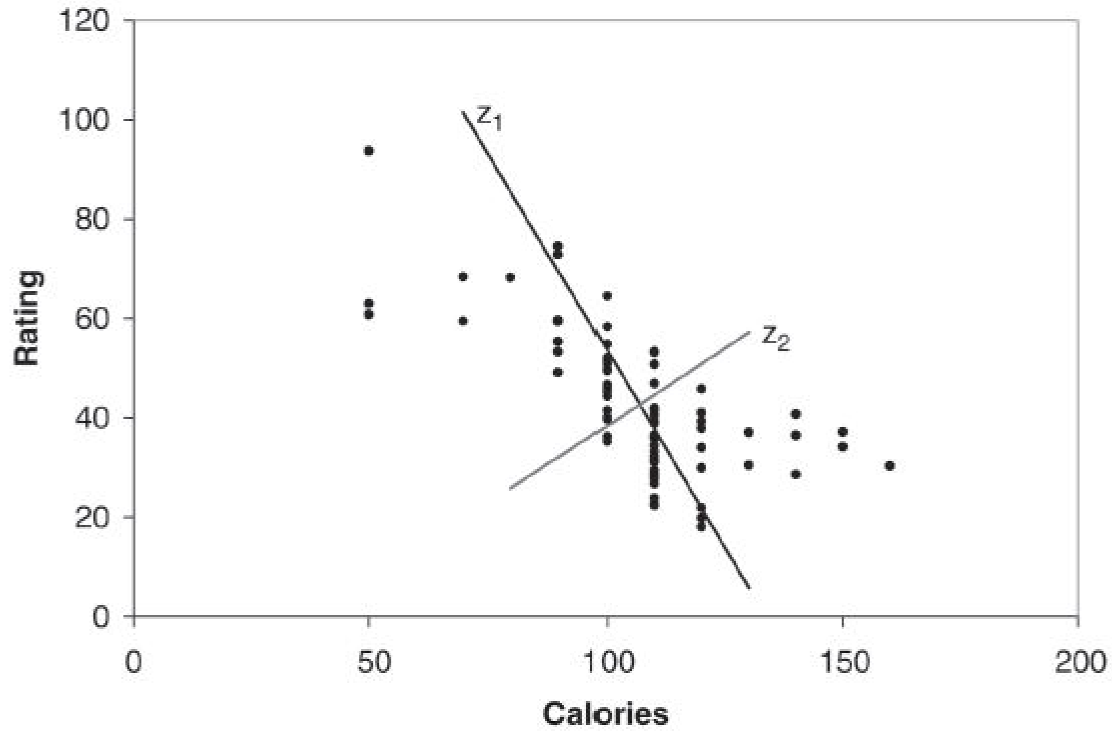
	calories	ratings
calories	379.63	-189.68
ratings	-189.68	197.32

- Total variance (="information") is sum of individual variances: 379.63 + 197.32
- Calories accounts for 379.63/577 = 66%
- If we want to make do with just calories, we lose 34% of the variation

Using linear combinations to redistribute the variability in a more polarized way

Z_1 and Z_2 are two linear combinations.

- Z_1 has the highest variation (spread of values)
- Z_2 has the lowest variation



PCA output for these 2 variables

```
pcs <- prcomp(data.frame(cereals.df$calories, cereals.df$rating))
summary(pcs)
```

Weights to project original data onto Z_1 & $Z_{2,}$ e.g. (0.847, -0.532) are weights for Z_1

```
                         PC1        PC2
cereals.df.calories   0.8470535  0.5315077
cereals.df.rating    -0.5315077  0.8470535
```

```
Importance of components:
                          PC1     PC2
Standard deviation      22.3165  8.8844
Proportion of Variance   0.8632  0.1368
Cumulative Proportion    0.8632  1.0000
```

86% of the total variance is accounted for by component 1

Principal Component Scores for the First Five Records

```
              PC1         PC2
[1,]  -44.921528   2.1971833
[2,]   15.725265  -0.3824165
[3,]  -40.149935  -5.4072123
[4,]  -75.310772  12.9991256
[5,]    7.041508  -5.3576857
```

PCA for the 13 Numerical Variables in the Cereals Data

```
> pcs <- prcomp(na.omit(cereals.df[,-c(1:3)]))
> summary(pcs)
```

```
Importance of components:
                           PC1     PC2      PC3      PC4     PC5     PC6     PC7
Standard deviation     83.7641 70.9143 22.64375 19.18148 8.42323 2.09167 1.69942
Proportion of Variance  0.5395  0.3867  0.03943  0.02829 0.00546 0.00034 0.00022
Cumulative Proportion   0.5395  0.9262  0.96560  0.99389 0.99935 0.99968 0.99991
                           PC8     PC9    PC10   PC11    PC12      PC13
Standard deviation     0.77963 0.65783 0.37043 0.1864 0.06302 5.334e-08
Proportion of Variance 0.00005 0.00003 0.00001 0.0000 0.00000 0.000e+00
Cumulative Proportion  0.99995 0.99999 1.00000 1.0000 1.00000 1.000e+00
```

The first two components account for 93% of the total variance, so using 2-3 components in further modeling would probably be sufficient

The Weightings for the First Five Components

	PC1	PC2	PC3	PC4	PC5
calories	0.0779841812	0.0093115874	-0.6292057595	-0.6010214629	0.454958508
protein	-0.0007567806	-0.0088010282	-0.0010261160	0.0031999095	0.056175970
fat	-0.0001017834	-0.0026991522	-0.0161957859	-0.0252622140	-0.016098458
sodium	0.9802145422	-0.1408957901	0.1359018583	-0.0009680741	0.013948118
fiber	-0.0054127550	-0.0306807512	0.0181910456	0.0204721894	0.013605026
carbo	0.0172462607	0.0167832981	-0.0173699816	0.0259482087	0.349266966
sugars	0.0029888631	0.0002534853	-0.0977049979	-0.1154809105	-0.299066459
potass	-0.1349000039	-0.9865619808	-0.0367824989	-0.0421757390	-0.047150529
vitamins	0.0942933187	-0.0167288404	-0.6919777623	0.7141179984	-0.037008623
shelf	-0.0015414195	-0.0043603994	-0.0124888415	0.0056471836	-0.007876459
weight	0.0005120017	-0.0009992138	-0.0038059565	-0.0025464145	0.003022113
cups	0.0005101111	0.0015910125	-0.0006943214	0.0009853800	0.002148458
rating	-0.0752962922	-0.0717421528	0.3079471212	0.3345338994	0.757708025

Generalization

$X_1, X_2, X_3, \ldots X_p$, original *p* variables

$Z_1, Z_2, Z_3, \ldots Z_p$, weighted averages of original variables

All pairs of Z variables have 0 correlation

Order Z's by variance (z_1 largest, Z_p smallest)

Usually the first few Z variables contain most of the information, and so the rest can be dropped.

Normalizing data

- In these results, sodium dominates first PC
- Just because of the way it is measured (mg), its scale is greater than almost all other variables
- Hence its variance will be a dominant component of the total variance
- Normalize each variable to remove scale effect
  - Divide by std. deviation (may subtract mean first)
- Normalization (= standardization) is usually performed in PCA; otherwise measurement units affect results

```
> pcs.cor <- prcomp(na.omit(cereals.df[,-c(1:3)]), scale. = T)
```

Normalize the variables

PCA Output Using all 13 ***Normalized*** Numerical Variables

```
> pcs.cor <- prcomp(na.omit(cereals.df[,-c(1:3)]), scale. = T)
> summary(pcs.cor)

Importance of components:
                          PC1    PC2    PC3     PC4    PC5     PC6     PC7     PC8
Standard deviation     1.9062 1.7743 1.3818 1.00969 0.9947 0.84974 0.81946 0.64515
Proportion of Variance 0.2795 0.2422 0.1469 0.07842 0.0761 0.05554 0.05166 0.03202
Cumulative Proportion  0.2795 0.5217 0.6685 0.74696 0.8231 0.87861 0.93026 0.96228
                           PC9    PC10    PC11    PC12      PC13
Standard deviation     0.56192 0.30301 0.25194 0.13897 1.499e-08
Proportion of Variance 0.02429 0.00706 0.00488 0.00149 0.000e+00
Cumulative Proportion  0.98657 0.99363 0.99851 1.00000 1.000e+00

> pcs.cor$rot[,1:5]
```

Weightings for the First Five ***Normalized*** Components

```
> pcs.cor$rot[,1:5]
```

	PC1	PC2	PC3	PC4	PC5
calories	0.29954236	0.3931479	-0.114857453	0.20435870	0.20389885
protein	-0.30735632	0.1653233	-0.277281953	0.30074318	0.31974897
fat	0.03991542	0.3457243	0.204890102	0.18683311	0.58689327
sodium	0.18339651	0.1372205	-0.389431009	0.12033726	-0.33836424
fiber	-0.45349036	0.1798119	-0.069766079	0.03917361	-0.25511906
carbo	0.19244902	-0.1494483	-0.562452458	0.08783547	0.18274252
sugars	0.22806849	0.3514345	0.355405174	-0.02270716	-0.31487243
potass	-0.40196429	0.3005442	-0.067620183	0.09087843	-0.14836048
vitamins	0.11598020	0.1729092	-0.387858660	-0.60411064	-0.04928672
shelf	-0.17126336	0.2650503	0.001531036	-0.63887859	0.32910135
weight	0.05029930	0.4503085	-0.247138314	0.15342874	-0.22128334
cups	0.29463553	-0.2122479	-0.139999705	0.04748909	0.12081645
rating	-0.43837841	-0.2515389	-0.181842433	0.03831622	0.05758420

PCA in Classification/Prediction

- Apply PCA to training data
- Decide how many PC's to use
- Use variable weights in those PC's with validation/new data
- This creates a new reduced set of predictors in validation/new data

Regression-Based Dimension Reduction

- Multiple Linear Regression or Logistic Regression
- Use subset selection
- Algorithm chooses a subset of variables
- This procedure is integrated directly into the predictive task

Summary

- **Data summarization** is an important for data exploration
- **Data summaries** include numerical metrics (average, median, etc.) and graphical summaries
- **Data reduction** is useful for compressing the information in the data into a smaller subset
  - Categorical variables can be reduced by combining similar categories
  - Principal components analysis transforms an original set of numerical data into a smaller set of weighted averages of the original data that contain most of the original information in less variables.