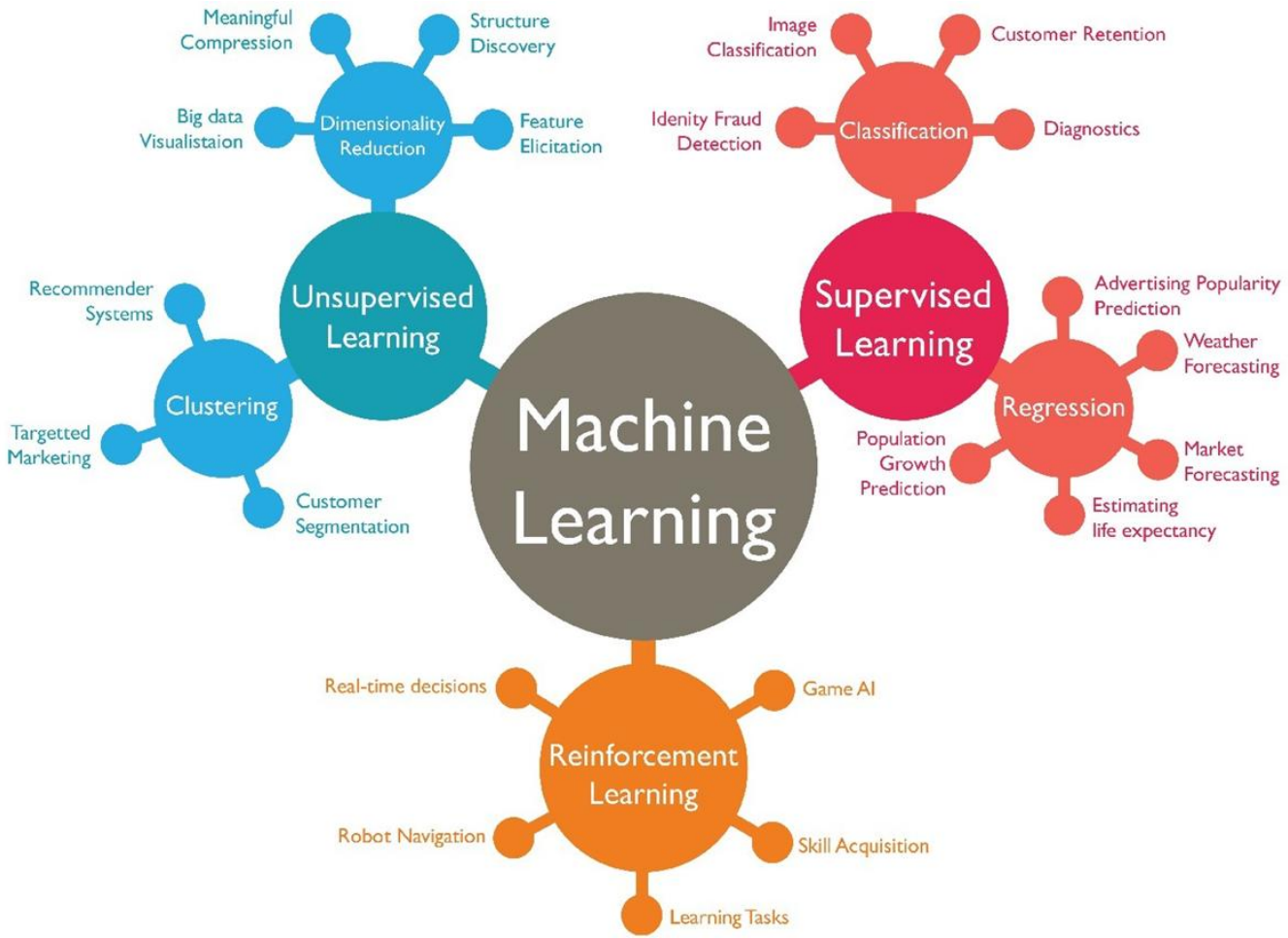
Анализ данных и машинное обучение, ч. 2

Лекция 3. Обучение без учителя. Факторный анализ. Метод главных компонент. Диагностика и оценка результатов

Киреев В.С.,
к.т.н., доцент

Москва, 2025

Задачи машинного обучения



Обучение без учителя

Обучение без учителя, также известное как машинное обучение без учителя, использует алгоритмы машинного обучения для анализа и кластеризации неразмеченных наборов данных. Эти алгоритмы обнаруживают скрытые шаблоны или группы данных без необходимости вмешательства человека.

Его способность обнаруживать сходства и различия в информации делает его идеальным решением для исследовательского анализа данных, стратегий перекрестных продаж, сегментации клиентов и распознавания изображений, создания признаков (фич) для моделей классификации и регрессии.

Обучение без учителя. Частые задачи

- автоматическое разделение исходных данных на группы в соответствии с выявленным сходством;
- обнаружение аномалий и нетипичных показателей в исходных данных
- определение наборов элементов, показателей и признаков, которые часто встречаются в исходных данных
- сокращение размерности исходных данных

Задача сокращения размерности

Сокращение размерности — это метод машинного обучения, используемый, когда количество признаков (или измерений) в заданном наборе данных слишком велико. Это уменьшает количество входных данных до управляемого размера, сохраняя при этом целостность данных.

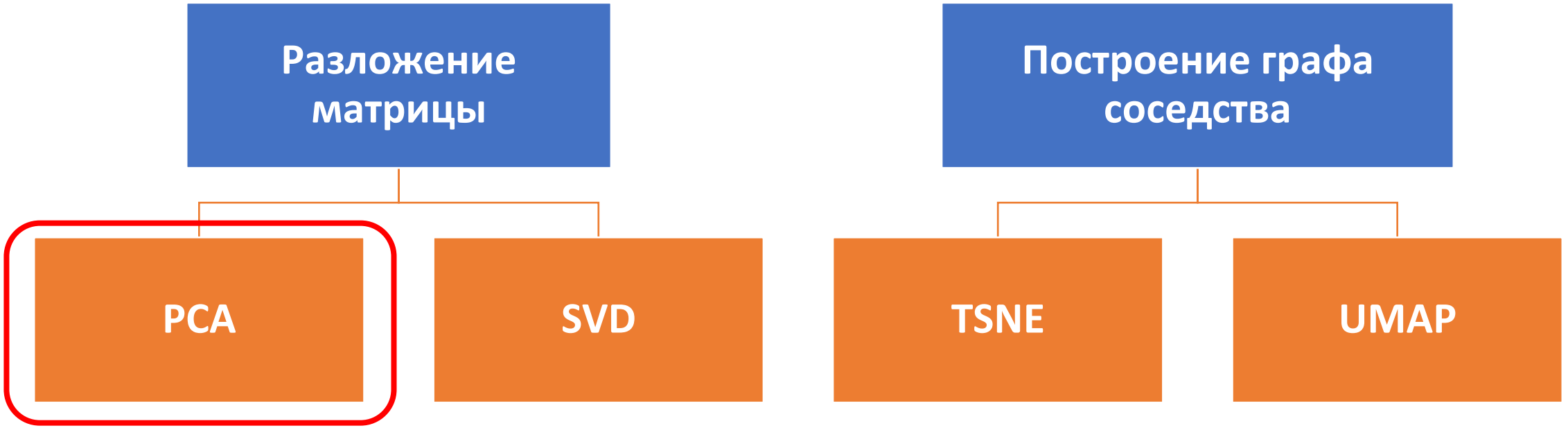
Часто этот метод используется на этапе предварительной обработки данных, например, для удаления шума из визуальных данных для улучшения качества изображения.

Задача сокращения размерности. Типы

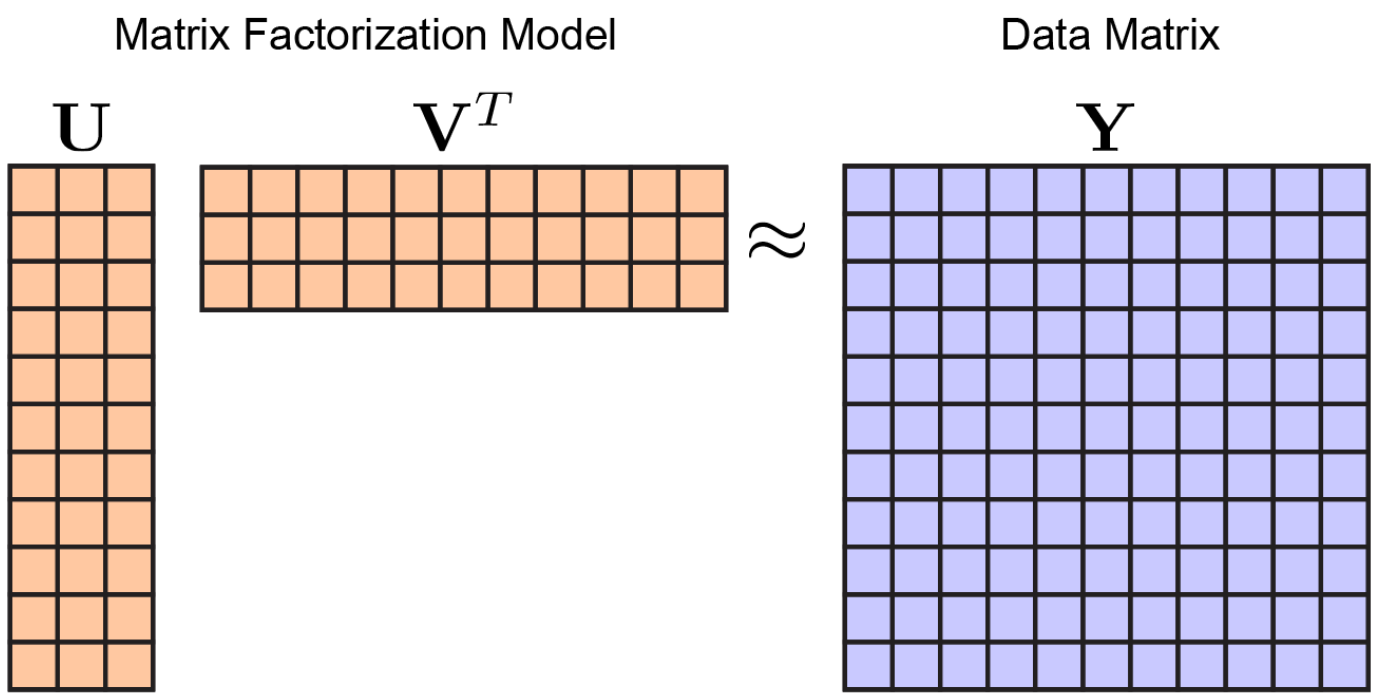
Задачу сокращения размерности можно разделить на два типа:

- отбор признаков (фич) – в этом случае применяются статистические тесты, чтобы сделать вывод о важности признаков и отобрать какое-либо их подмножество. Подразумевает потерю информации и меньшую стабильность.
- извлечение признаков (фич) – в этом случае создаются новых независимые признаки в виде комбинации исходных признаков. Применяются линейные и нелинейные методологии.

Обучение без учителя. Методы сокращения размерности. Извлечение признаков



Задача сокращения размерности. Разложение матрицы

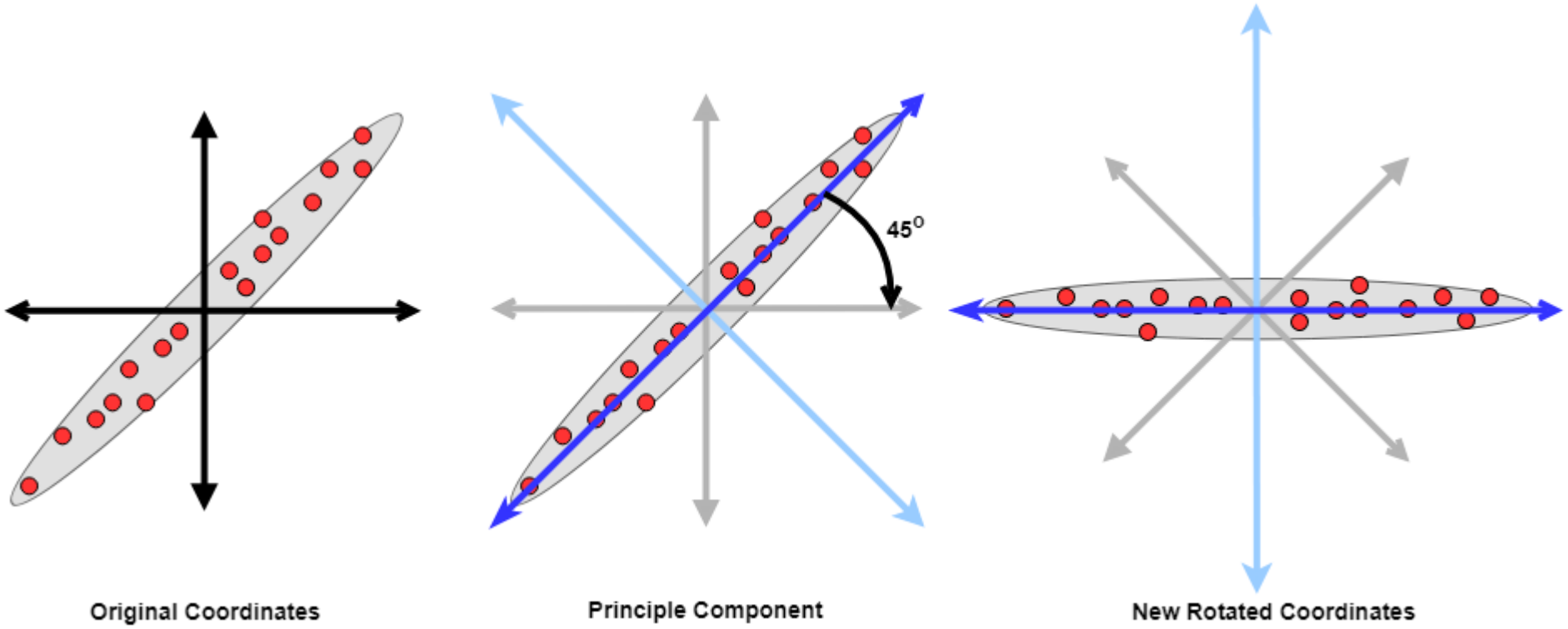


Метод главных компонент (PCA)

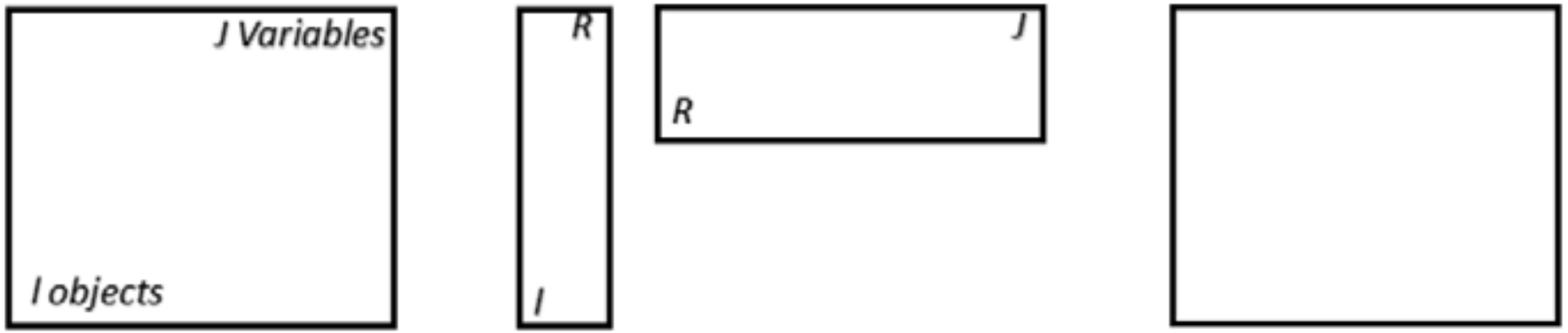
Метод главных компонент (МГК) применяется для снижения размерности пространства наблюдаемых векторов, не приводя к существенной потере информативности. Предпосылкой МГК является нормальный закон распределения многомерных векторов.

В МГК линейные комбинации случайных величин определяются характеристическими векторами ковариационной матрицы. Главные компоненты представляют собой ортогональную систему координат, в которой дисперсии компонент характеризуют их статистические свойства.

Метод главных компонент (PCA). Принципиальная схема работы

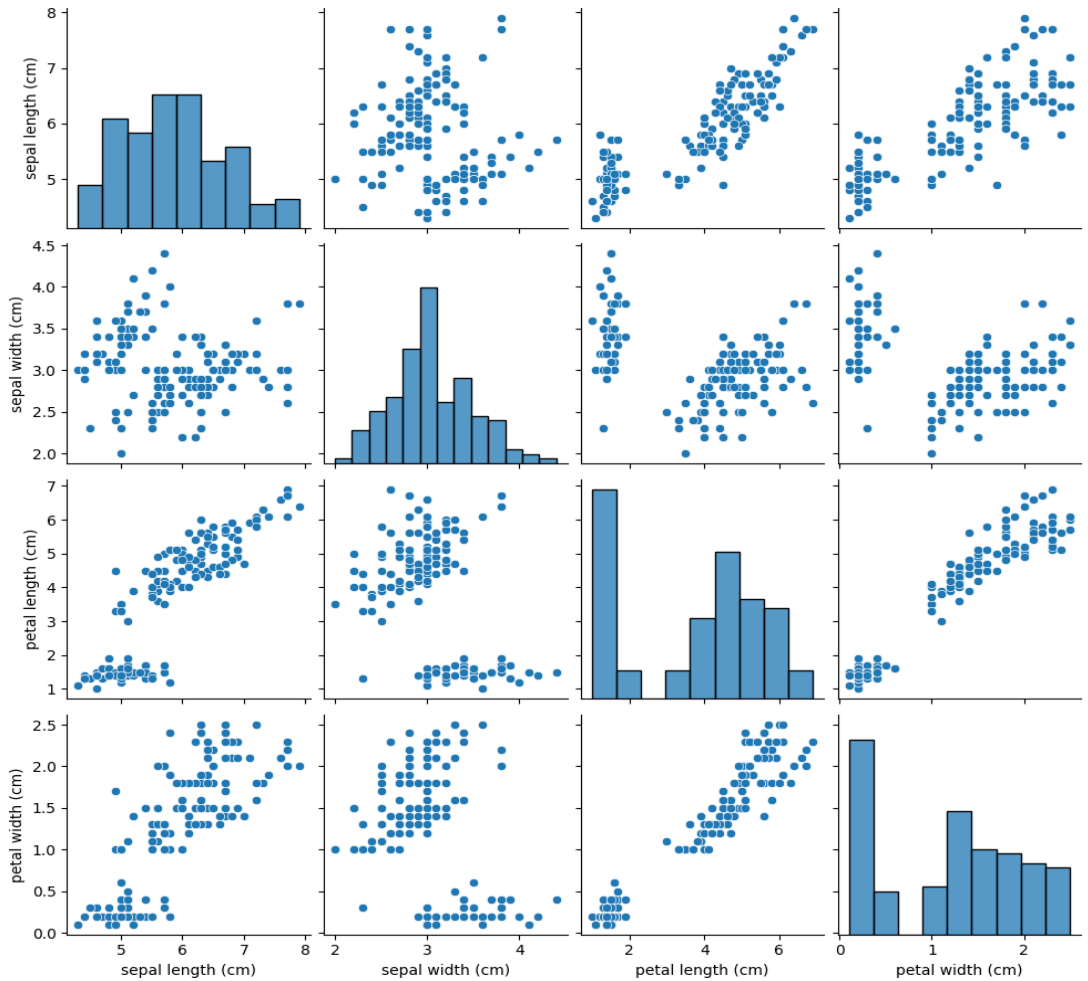


Метод главных компонент (PCA). Схема матриц



$$X = T P^{T} + E$$

Метод главных компонент (PCA). Пример. Ирисы Фишера



Метод главных компонент (PCA). Матрица ковариации

$$\operatorname{cov}[X, X] = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_x)(x_i - \mu_x) =$$

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_x)^2 = \operatorname{var}(X)$$

Метод главных компонент (PCA). Пример. Ирисы Фишера. Расчет матрицы ковариации

```
1 sc=StandardScaler()
2 dfz=pd.DataFrame(sc.fit_transform(df),columns=[x+'_Z' for x in load_iris()['feature_names']])
3 dfz.head(3)
```

	sepal length (cm)_Z	sepal width (cm)_Z	petal length (cm)_Z	petal width (cm)_Z
0	-0.900681	1.019004	-1.340227	-1.315444
1	-1.143017	-0.131979	-1.340227	-1.315444
2	-1.385353	0.328414	-1.397064	-1.315444

```
1 cov=1/(df.shape[0]-1)*np.dot(dfz.values.T,dfz.values)
2 cov
```

```
array([[ 1.00671141, -0.11835884,  0.87760447,  0.82343066],
       [-0.11835884,  1.00671141, -0.43131554, -0.36858315],
       [ 0.87760447, -0.43131554,  1.00671141,  0.96932762],
       [ 0.82343066, -0.36858315,  0.96932762,  1.00671141]])
```

Метод главных компонент (PCA). Собственные вектора и собственные значения

$$\mathrm{A}\,\mathbf{X} = \lambda\,\mathbf{X}$$

Метод главных компонент (PCA). Пример. Ирисы Фишера. Расчет собственных значений

```
1 eigenvalues, eigenvectors = np.linalg.eig(cov)
2 eigenvalues
```

```
array([4.22824171, 0.24267075, 0.0782095 , 0.02383509])
```

```
1 eigenvectors.T
```

```
array([[ 0.36138659, -0.08452251,  0.85667061,  0.3582892 ],
       [-0.65658877, -0.73016143,  0.17337266,  0.07548102],
       [-0.58202985,  0.59791083,  0.07623608,  0.54583143],
       [ 0.31548719, -0.3197231 , -0.47983899,  0.75365743]])
```

Матрица нагрузок (loadings matrix)

Матрица нагрузок P – это матрица перехода из исходного пространства переменных x1, … xJ (J-мерного) в пространство главных компонент (A-мерное). Каждая строка матрицы P состоит из коэффициентов, связывающих переменные t и x.

Например, a-я строка – это проекция всех переменных x1, …xJ на a-ю ось главных компонент. Каждый столбец P – это проекция соответствующей переменной xj на новую систему координат.

Метод главных компонент (PCA). Пример. Ирисы Фишера. Расчет матрицы нагрузок

```
1 eigenvalues, eigenvectors = np.linalg.eig(cov)
2 eigenvalues
```

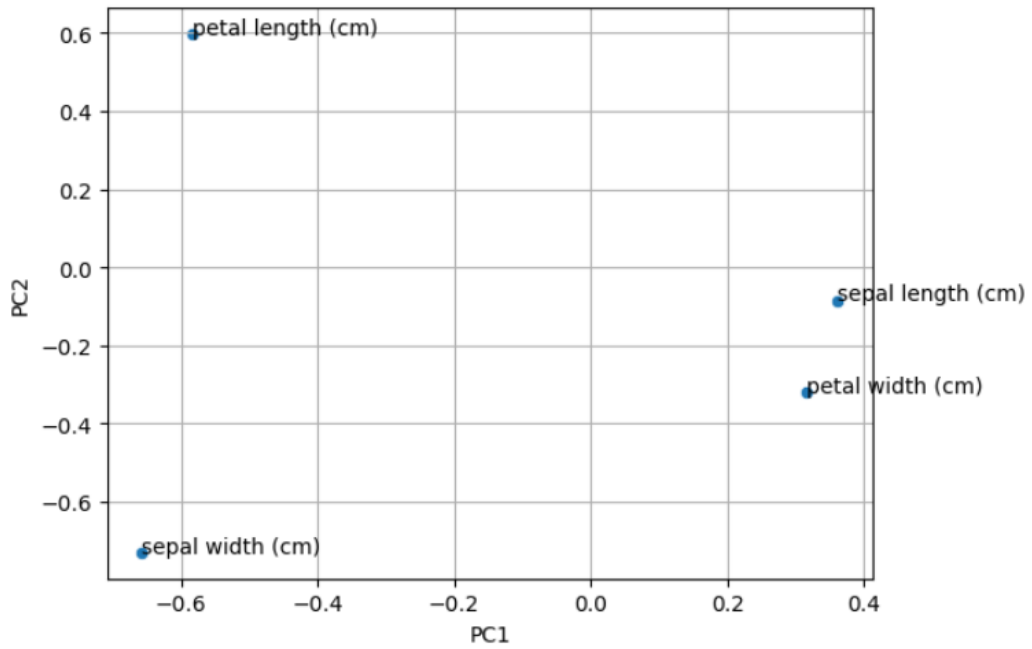
```
array([4.22824171, 0.24267075, 0.0782095 , 0.02383509])
```

```
1 eigenvectors.T
```

```
array([[ 0.36138659, -0.08452251,  0.85667061,  0.3582892 ],
       [-0.65658877, -0.73016143,  0.17337266,  0.07548102],
       [-0.58202985,  0.59791083,  0.07623608,  0.54583143],
       [ 0.31548719, -0.3197231 , -0.47983899,  0.75365743]])
```

Метод главных компонент (PCA). Пример. Ирисы Фишера. Визуализация нагрузок

```
f=sns.scatterplot(pd.DataFrame(eigenvectors.T,index=df.columns,columns=x_labels),x='PC1',y='PC2')
f.grid()
pc=pd.DataFrame(eigenvectors.T,index=df.columns,columns=x_labels)
for i in range(pc.shape[0]):
  f.text(pc.iloc[i,0],
         pc.iloc[i,1],s=pc.index[i])
```



Матрица счетов (score matrix)

Матрица счетов T дает нам проекции исходных образцов (J –мерных векторов x1,…,xI) на подпространство главных компонент (A-мерное). Строки t1,…,tI матрицы T – это координаты образцов в новой системе координат. Столбцы t1,…,tA матрицы T – ортогональны и представляют проекции всех образцов на одну новую координатную ось.

Метод главных компонент (PCA). Пример. Ирисы Фишера. Расчет матрицы счетов

```
1 pd.DataFrame(dfV.values.dot(P),columns=x_labels)
```

	PC1	PC2	PC3	PC4
0	-2.684126	-0.319397	-0.027915	0.002262
1	-2.714142	0.177001	-0.210464	0.099027
2	-2.888991	0.144949	0.017900	0.019968
3	-2.745343	0.318299	0.031559	-0.075576
4	-2.728717	-0.326755	0.090079	-0.061259
...	...	...	...	...
145	1.944110	-0.187532	0.177825	0.426196
146	1.527167	0.375317	-0.121898	0.254367
147	1.764346	-0.078859	0.130482	0.137001
148	1.900942	-0.116628	0.723252	0.044595
149	1.390189	0.282661	0.362910	-0.155039

150 rows × 4 columns

Отбор главных компонент. График «осыпи» (scree plot)

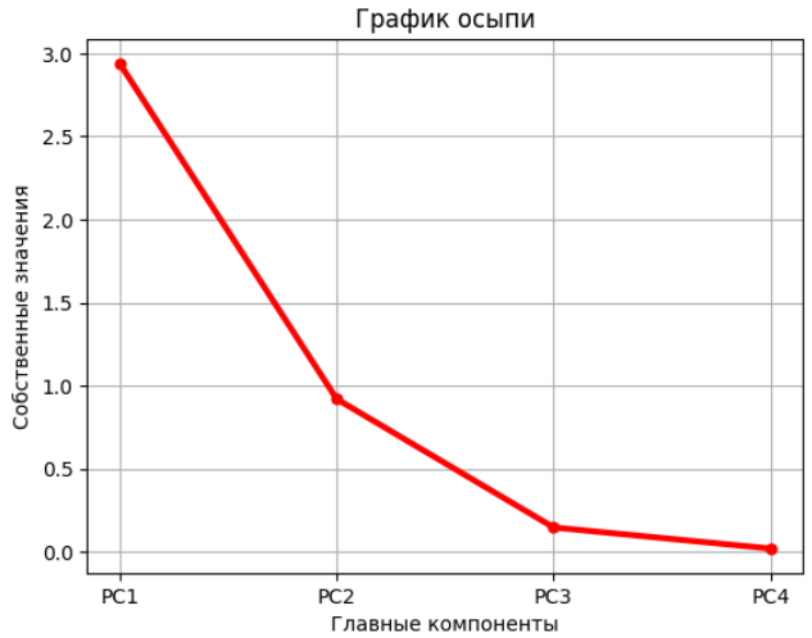
Можно показать, что дисперсия главной компоненты равна соответствующему собственному значению. Более того, поскольку след корреляционной матрицы равен сумме ее собственных значений и, очевидно, равен p - количеству переменных, а кроме того, главные компоненты некоррелированы, то сразу следует, что доля общей дисперсии которая объясняется j-той компонентой — это просто λj/p.

График «осыпи» представляет собой график, показывающий объясненную дисперсию для вновь определенной компоненты (главной компоненты). Мерой графика может быть процент или абсолютное значение объясненной дисперсии (собственные значения).

Метод главных компонент (PCA). Пример. Ирисы Фишера. График «осыпи»

```
sum_eigenvalues = np.sum(eigenvalues)
prop_var = [i for i in eigenvalues]
x_labels = ['PC{}'.format(i+1) for i in range(len(prop_var))]

plt.plot(x_labels, prop_var, marker='o', markersize=5, color='red', linewidth=3);
plt.grid()
plt.title('График осыпи')
plt.xlabel('Главные компоненты')
plt.ylabel('Собственные значения')
plt.show()
```



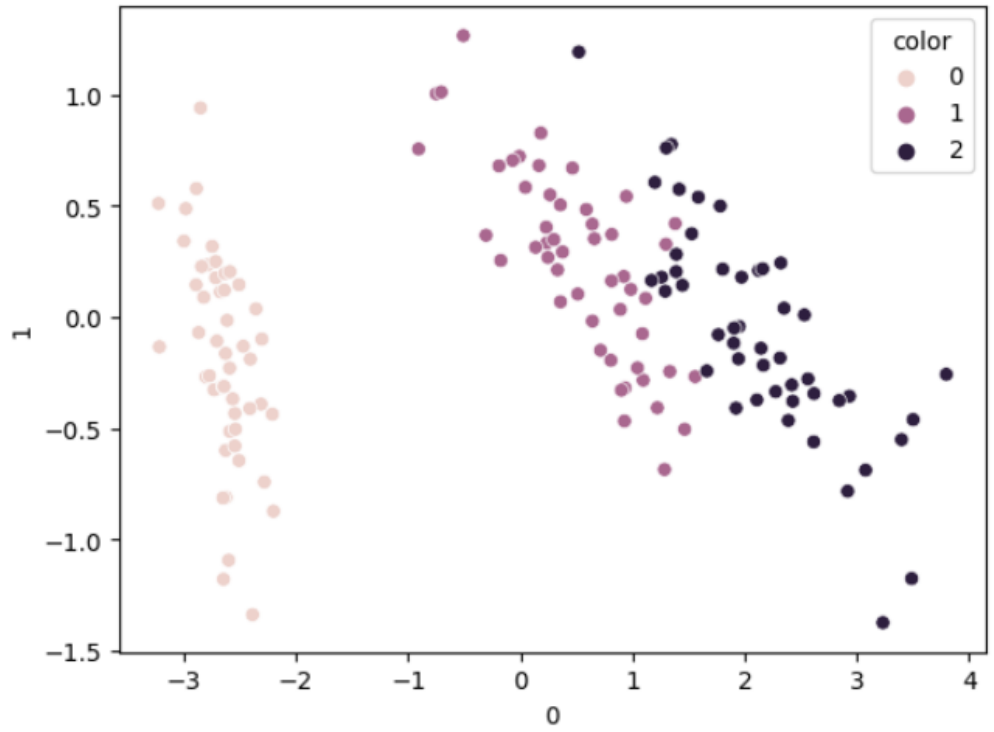
Отбор главных компонент. Критерий Кайзера

Критерий Кайзера-Гуттмана гласит, что компоненты, основанные на собственных значениях больше 1, должны быть отобраны. Это основано на представлении о том, что, поскольку сумма собственных значений равна p, собственное значение больше 1 представляет компоненту «выше среднего».

Метод главных компонент (PCA). Пример. Ирисы Фишера. Визуализация

```
1 sns.scatterplot(data=pd.DataFrame(dfV.values.dot(P)).assign(color=load_iris()['target']),x=0,y=1,hue='color')
```

```
<Axes: xlabel='0', ylabel='1'>
```



Особенности метода главных компонент

В основе метода главных компонент лежат следующие допущения:

- допущение о том, что размерность данных может быть эффективно понижена путем линейного преобразования;
- допущение о том, что больше всего информации несут те направления, в которых дисперсия входных данных максимальна.

Недостатки метода главных компонент

- Можно легко видеть, что эти условия далеко не всегда выполняются. Например, если точки входного множества располагаются на поверхности гиперсферы, то никакое линейное преобразование не сможет понизить размерность (но с этим легко справится нелинейное преобразование, опирающееся на расстояние от точки до центра сферы). Это недостаток в равной мере свойственен всем линейным алгоритмам и может быть преодолен за счет использования дополнительных фиктивных переменных, являющихся нелинейными функциями от элементов набора входных данных (т.н. kernel trick).
- Второй недостаток метода главных компонент состоит в том, что направления, максимизирующие дисперсию, далеко не всегда максимизируют информативность. Например, переменная с максимальной дисперсией может не нести почти никакой информации, в то время как переменная с минимальной дисперсией позволяет полностью разделить классы. Метод главных компонент в данном случае отдаст предпочтение первой (менее информативной) переменной. Вся дополнительная информация, связанная с вектором (например, принадлежность образа к одному из классов), игнорируется.

Вращение системы компонент (факторов)

Вращение — это процесс, который позволяет аналитикам сделать решение факторного анализа более интерпретируемым. Ортогональные повороты требуют, чтобы фактор не был коррелирован.

Целью ортогонального вращения является обобщаемость и простота. Напротив, наклонные повороты учитывают коррелированные факторы, и их цель состоит в том, чтобы обеспечить наилучшее соответствие.

Вращение системы компонент (факторов). Виды

Ортогональное вращение: основным преимуществом является простота. Нагрузки представляют собой корреляции между факторами и наблюдаемыми функциями.

- Varimax: мало больших и много нагрузок, близких к 0
- Quartimax: заставляет данную переменную сильно коррелировать с одним фактором. Делает большие нагрузки очень большими, а маленькие загрузки очень маленькими. Максимизирует дисперсию по строкам факторной матрицы, возводя нагрузки в четвертую степень.
- Облимакс: когда предположение об однородно распределенной ошибке не может быть применено, но может быть заменено принципом максимального эксцесса, согласно Д. Р. Сондерсу (обоснование метода преобразования «облимакс» в факторном анализе).
- Equimax: Попытка улучшения варимакс. Вращение подстраивается под количество вращаемых факторов, что приводит к более равномерному распределению набора факторов по сравнению с варимаксом. Создает менее общие факторы.

Варимаксное вращение

Этот критерий использует формализацию сложности фактора через дисперсию квадратов нагрузок переменной:

$$v_j = \frac{n\sum_{i=1}^{n}(b_{ij}^4 - (\sum_{i=1}^{n} b_i^2)^2}{n^2}$$

Тогда критерий в общем виде можно записать как:

$$V = \frac{\sum_{j=1}^{r}(n\sum_{i=1}^{n}(b_{ij}^4) - \sum_{j=1}^{r}(\sum_{i=1}^{n} b_i^2)^2}{n^2}$$

При этом, факторные нагрузки могут нормироваться для избавления от влияния отдельных переменных.

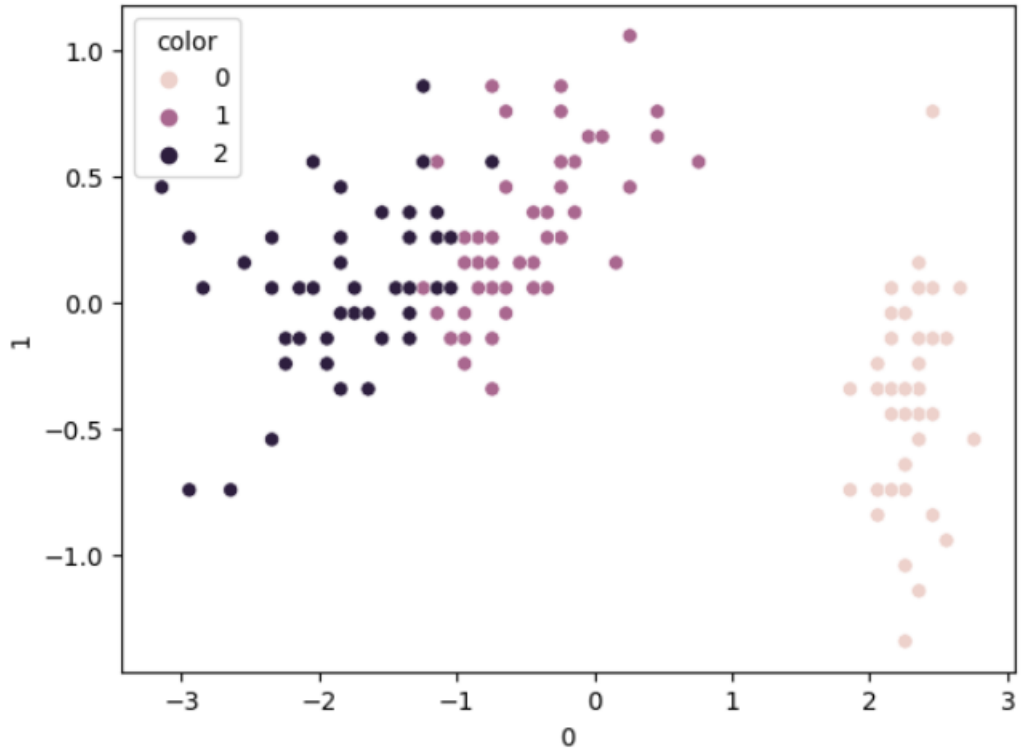
Варимаксное вращение

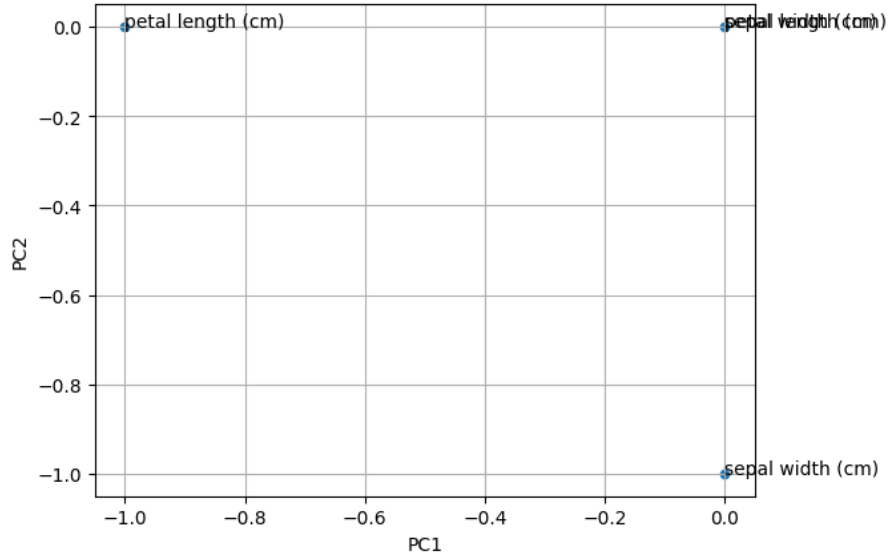
```
def varimax(Phi, gamma = 1, q = 30, tol = 1e-6):
    from numpy import eye, asarray, dot, sum, diag
    from numpy.linalg import svd
    p,k = Phi.shape
    R = eye(k)
    d=0
    for i in range(q):
        d_old = d
        Lambda = dot(Phi, R)
        u,s,vh = svd(dot(Phi.T,asarray(Lambda)**3 - (gamma/p) * dot(Lambda, diag(diag(dot(Lambda.T,Lambda))))))
        R = dot(u,vh)
        d = sum(s)
        if d/d_old < tol: break
    return dot(Phi, R)
```

Метод главных компонент (PCA). Пример. Ирисы Фишера. Варимакс. Визуализация

```
1 sns.scatterplot(data=pd.DataFrame(dfV.values.dot(varimax(eigenvectors.T))).assign(color=load_iris()['target']),x=0,y=1,hue='color')
```

```
<ipython-input-125-1fd5853a2d70>:13: RuntimeWarning: divide by zero encountered in double_scalars
  if d/d_old < tol: break
<Axes: xlabel='0', ylabel='1'>
```





Пример данных для МГК

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
39			**Autoscaled Data**											
40			**Height**	**Weight**	**Hair**	**Shoes**	**Age**	**Income**	**Beer**	**Wine**	**Sex**	**Strength**	**Region**	**IQ**
41	**1**	**MN**	2.47	1.81	-0.98	2.08	1.43	1.97	1.88	-0.34	-0.98	2.25	-0.98	-1.24
42	**2**	**MN**	1.08	1.29	-0.98	1.05	-0.15	0.62	1.11	-0.60	-0.98	1.43	-0.98	1.22
43	**3**	**MN**	0.98	1.22	-0.98	1.05	0.27	0.73	0.78	-0.68	-0.98	1.30	-0.98	0.98
44	**4**	**MN**	0.88	1.02	-0.98	0.54	0.06	0.29	1.64	-1.35	-0.98	0.48	-0.98	2.04
45	**5**	**MN**	0.68	1.02	-0.98	0.79	0.16	0.29	1.53	-1.39	-0.98	0.34	-0.98	1.14
46	**6**	**MN**	0.98	1.09	-0.98	0.54	0.27	0.85	1.05	-1.75	-0.98	1.16	-0.98	-0.83
47	**7**	**MN**	0.68	1.15	-0.98	1.05	0.90	1.07	1.16	-1.00	-0.98	0.89	-0.98	-0.50
48	**8**	**MN**	0.68	1.09	-0.98	1.05	1.21	1.63	1.24	-0.84	-0.98	0.61	-0.98	-0.17
49	**9**	**MS**	1.18	1.15	-0.98	1.31	-0.89	-1.28	0.50	0.98	-0.98	1.43	0.98	-0.50
50	**10**	**MS**	1.38	1.29	-0.98	1.56	-0.78	-1.22	0.55	0.94	-0.98	1.84	0.98	0.32
51	**11**	**MS**	0.39	0.03	-0.98	0.28	-0.89	-1.06	-0.45	0.57	-0.98	0.61	0.98	0.40
52	**12**	**MS**	0.68	0.50	-0.98	0.79	-0.15	-0.94	-0.15	0.88	-0.98	0.48	0.98	-0.01
53	**13**	**MS**	0.78	0.69	-0.98	0.79	0.79	0.40	-0.57	0.59	-0.98	0.20	0.98	-0.83
54	**14**	**MS**	0.29	0.23	-0.98	0.54	1.64	0.96	-0.60	0.92	-0.98	0.07	0.98	-1.57
55	**15**	**MS**	0.19	0.17	0.98	0.54	2.16	1.18	-0.71	1.12	-0.98	-0.20	0.98	-0.83
56	**16**	**MS**	0.48	0.69	-0.98	0.54	-0.47	-0.38	-0.51	1.54	-0.98	-0.07	0.98	0.24
57	**17**	**FN**	-0.71	-1.15	-0.98	-1.00	-0.26	0.06	0.23	-1.08	0.98	-0.89	-0.98	-0.26
58	**18**	**FN**	-0.31	-0.29	0.98	-0.49	-1.20	-0.83	0.69	-0.66	0.98	-0.07	-0.98	-0.42
59	**19**	**FN**	-0.11	-0.03	0.98	-0.23	-1.10	-0.61	0.65	-0.82	0.98	0.07	-0.98	-1.08
60	**20**	**FN**	-0.41	-0.89	0.98	-1.00	-1.10	-0.50	0.01	-0.86	0.98	-0.48	-0.98	-1.41
61	**21**	**FN**	-0.51	-0.82	0.98	-0.75	-0.78	-0.44	0.12	-0.92	0.98	-0.48	-0.98	-1.24
62	**22**	**FN**	-1.60	-1.15	0.98	-1.00	-0.26	0.51	-0.16	-0.80	0.98	-1.57	-0.98	0.98
63	**23**	**FN**	-0.91	-0.95	0.98	-0.49	0.69	0.73	0.06	0.05	0.98	-0.75	-0.98	-1.16
64	**24**	**FN**	-1.11	-1.02	0.98	-0.75	0.58	0.73	0.17	-0.15	0.98	-0.89	-0.98	-0.59
65	**25**	**FS**	-0.51	-0.95	0.98	-0.75	1.53	0.73	-0.88	0.61	0.98	-0.75	0.98	1.63
66	**26**	**FS**	-0.71	-1.02	0.98	-1.00	-1.41	-1.50	-1.10	2.29	0.98	-0.89	0.98	0.65
67	**27**	**FS**	-1.50	-1.22	0.98	-1.52	-0.47	-1.06	-1.43	-0.23	0.98	-1.57	0.98	0.32
68	**28**	**FS**	-1.01	-0.95	0.98	-1.00	-1.73	-1.84	-1.18	0.09	0.98	-0.89	0.98	-1.08
69	**29**	**FS**	-1.11	-0.95	0.98	-1.00	-1.52	-1.78	-1.29	0.29	0.98	-1.02	0.98	1.39
70	**30**	**FS**	-0.81	-0.89	0.98	-1.00	0.16	-0.16	-1.42	-0.05	0.98	-0.75	0.98	0.89
71	**31**	**FS**	-1.21	-1.08	0.98	-1.26	0.69	0.45	-1.47	1.30	0.98	-0.89	0.98	0.40
72	**32**	**FS**	-1.31	-1.08	0.98	-1.26	0.58	0.40	-1.45	1.34	0.98	-1.02	0.98	1.14
73	**mean**		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
74	**STD**		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
75														

Пример данных для МГК. Обозначения

Height	Рост: в сантиметрах
Hair	Волосы: короткие: –1, или длинные: +1
Age	Возраст: в годах
Beer	Пиво: потребление в литрах в год
Sex	Пол: мужской: –1, или женский: +1
Region	Регион: север : –1, или юг: +1

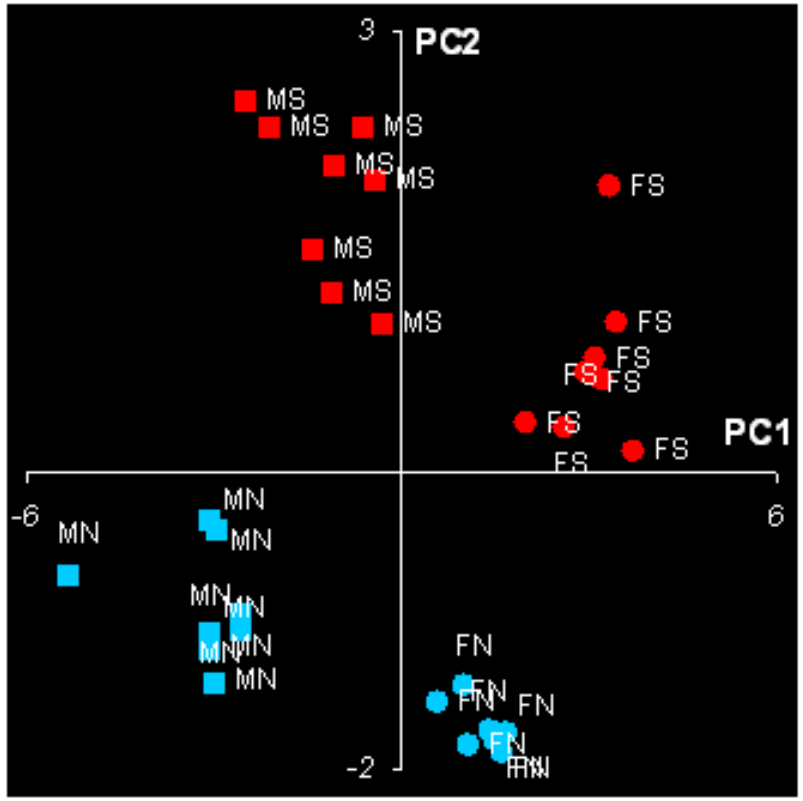
Матрица счетов

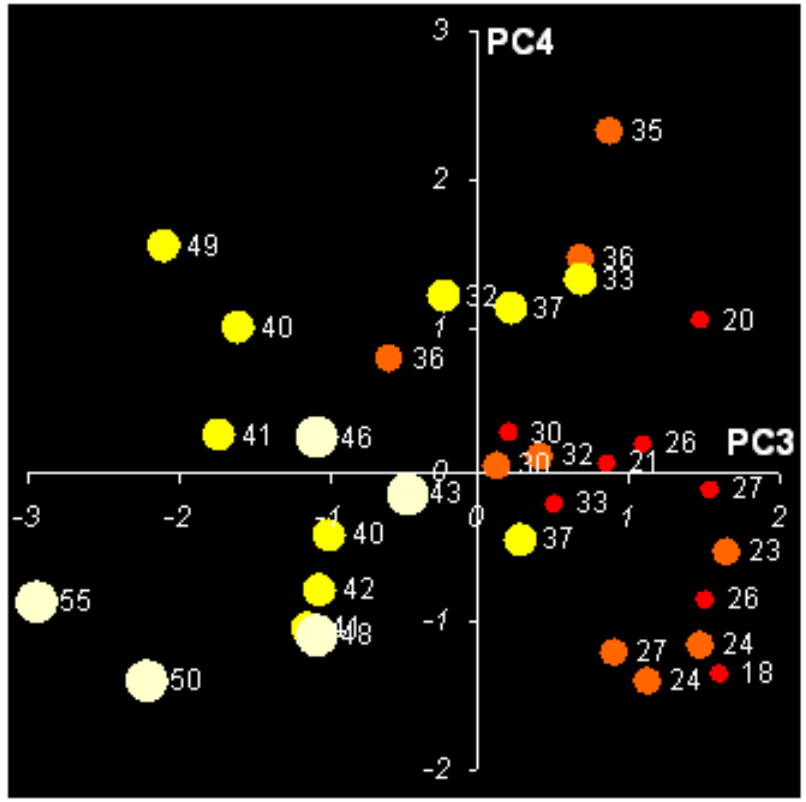
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		**PCA**												
2		T Scores												
3		PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	
4	1	=ScoresPCA(Xraw,12,3)				1.060	-0.017	0.563	0.085	0.280	0.078	0.109	0.133	
5	2	-3.114	-0.293	0.671	1.310	0.435	0.119	0.109	-0.456	-0.088	-0.034	0.051	-0.009	
6	3	-2.997	-0.360	0.212	1.117	0.204	-0.017	0.120	-0.469	-0.149	-0.018	0.184	-0.208	
7	4	-2.591	-0.928	0.863	2.321	-0.095	-0.076	-0.270	0.306	0.240	-0.192	-0.168	0.005	
8	5	-2.588	-1.037	0.686	1.461	-0.347	-0.098	-0.447	0.345	0.083	0.021	-0.043	-0.005	
9	6	-3.027	-1.400	0.284	-0.440	-0.633	-0.538	0.252	-0.343	0.042	-0.132	-0.247	0.020	
10	7	-3.095	-1.069	-0.472	-0.144	-0.304	-0.059	-0.147	-0.025	-0.217	0.150	-0.077	-0.088	
11	8	-3.113	-1.181	-1.068	0.242	-0.232	0.141	-0.211	-0.008	-0.038	0.203	-0.060	0.087	
12	9	-2.130	2.356	1.503	-0.858	0.284	0.070	0.020	0.246	-0.171	0.095	-0.123	0.060	
13	10	-2.510	2.525	1.542	-0.105	0.653	-0.087	0.222	0.228	-0.305	0.078	0.024	0.035	
14	11	-0.463	1.992	1.090	0.206	-0.617	0.084	0.138	-0.186	-0.202	-0.421	0.069	0.174	
15	12	-1.098	2.094	0.496	-0.198	-0.376	0.127	-0.126	0.299	-0.071	-0.122	0.112	-0.115	
16	13	-1.438	1.527	-1.059	-0.788	-0.734	-0.124	0.015	-0.060	0.264	0.185	0.156	-0.057	
17	14	-1.137	1.241	-2.200	-1.404	-0.953	0.200	0.033	0.063	-0.075	-0.002	-0.058	-0.009	
18	15	-0.334	1.034	-2.931	-0.866	0.420	-0.654	-0.781	0.037	0.016	-0.283	0.060	-0.004	
19	16	-0.652	2.360	0.125	0.055	-0.334	0.674	-0.398	-0.283	0.359	0.192	-0.007	-0.007	
20	17	1.084	-1.845	0.409	0.123	-1.323	0.872	0.704	0.328	0.032	-0.029	0.078	-0.026	
21	18	0.981	-1.434	1.645	-0.526	0.714	-0.035	-0.096	0.127	-0.005	0.003	-0.124	-0.053	
22	19	0.567	-1.551	1.474	-1.154	0.677	-0.234	-0.047	0.031	0.068	0.213	-0.058	-0.052	
23	20	1.663	-1.762	1.122	-1.394	0.018	-0.081	0.085	-0.218	0.224	-0.284	0.028	0.004	
24	21	1.486	-1.813	0.904	-1.208	0.070	-0.147	-0.003	-0.023	0.043	-0.160	0.109	-0.034	
25	22	2.464	-2.040	-0.222	1.202	-0.140	0.268	-0.491	-0.185	-0.091	0.166	0.191	0.233	
26	23	1.396	-1.736	-1.130	-1.041	0.367	0.487	-0.170	0.133	-0.220	-0.007	0.123	-0.074	
27	24	1.622	-1.894	-0.992	-0.419	0.287	0.463	-0.204	0.141	-0.225	-0.048	-0.033	0.015	
28	25	2.005	0.344	-2.085	1.549	0.603	-0.377	0.509	0.471	0.105	-0.061	0.070	0.020	
29	26	3.335	1.956	0.851	0.063	0.884	0.897	-0.141	-0.048	0.252	-0.128	-0.057	-0.046	
30	27	3.711	0.147	0.195	0.279	-0.774	-0.624	-0.091	0.120	0.002	0.081	-0.114	-0.074	
31	28	3.207	0.630	1.602	-1.358	-0.420	-0.480	-0.057	-0.047	-0.001	0.168	-0.026	0.112	
32	29	3.423	1.021	1.482	1.036	0.016	-0.395	-0.084	-0.024	-0.035	0.124	0.177	-0.027	
33	30	2.615	0.320	-0.600	0.784	-0.038	-0.789	0.455	-0.109	0.045	0.093	0.108	-0.007	
34	31	2.958	0.688	-1.728	0.267	0.268	0.181	0.320	-0.246	-0.082	0.016	-0.249	-0.012	
35	32	3.102	0.788	-1.603	0.988	0.359	0.249	0.220	-0.232	-0.078	0.055	-0.203	0.009	
36														

Матрица нагрузок

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
2			P Loadings												
3			PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	
4		Height	=LoadingsPCA(Xraw,12,3)				0.186	-0.124	0.268	0.118	0.729	-0.307	0.255	-0.065	
5		Weight	-0.381	0.111	0.068	0.033	0.100	-0.192	-0.224	-0.219	0.190	0.572	-0.415	-0.395	
6		Hair	0.338	-0.150	-0.079	-0.114	0.660	-0.489	-0.368	-0.081	0.041	-0.140	0.007	0.078	
7		Shoes	-0.378	0.151	0.001	-0.066	0.152	-0.031	-0.234	0.171	-0.280	0.376	0.685	0.183	
8		Age	-0.143	-0.061	-0.720	0.055	-0.029	-0.165	0.043	0.435	-0.174	-0.134	-0.036	-0.430	
9		Income	-0.190	-0.287	-0.586	0.085	0.063	0.137	0.129	-0.434	0.180	0.167	-0.038	0.492	
10		Beer	-0.325	-0.308	0.188	0.040	0.231	0.239	-0.170	0.567	-0.015	-0.049	-0.420	0.350	
11		Wine	0.124	0.554	-0.212	-0.125	0.415	0.638	-0.120	-0.040	0.024	-0.054	-0.095	-0.093	
12		Sex	0.352	-0.232	0.052	-0.051	0.313	0.098	0.580	0.254	0.078	0.529	0.084	-0.124	
13		Strength	-0.365	0.112	0.135	-0.081	0.336	-0.160	0.512	-0.258	-0.530	-0.232	-0.165	-0.001	
14		Region	0.144	0.595	-0.130	-0.022	-0.151	-0.402	0.161	0.265	0.050	0.180	-0.255	0.476	
15		IQ	0.044	0.123	0.062	0.969	0.180	-0.010	0.024	0.001	-0.006	-0.033	0.076	-0.010	
16															

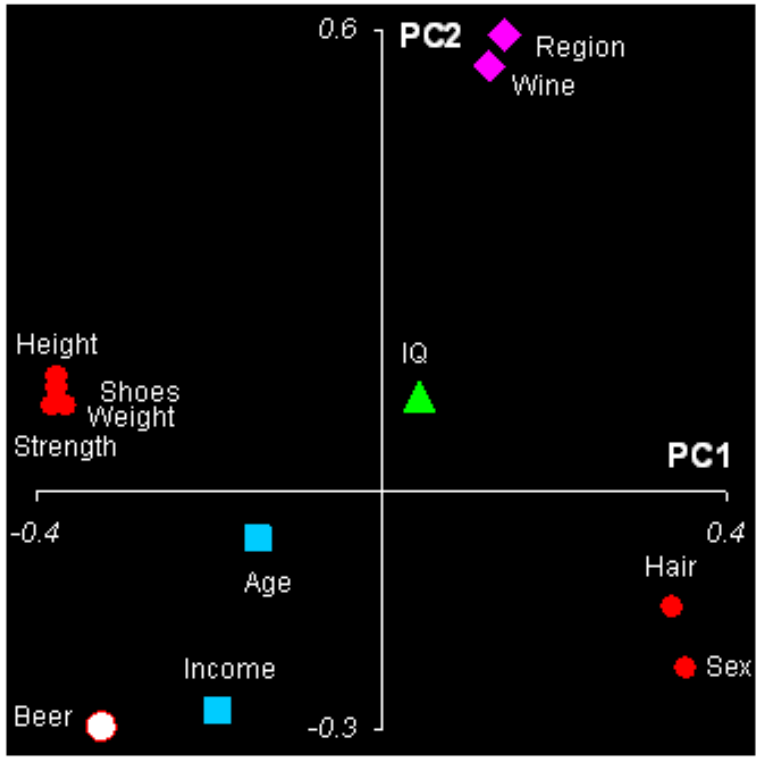
Недостатки метода главных компонент





Женщины (F) обозначены кружками ● и ●, а мужчины (M) – квадратами ■ и ■. Север (N) представлен голубым ■, а юг (S) – красным цветом ●.

Исходные переменные в пространстве новых компонент



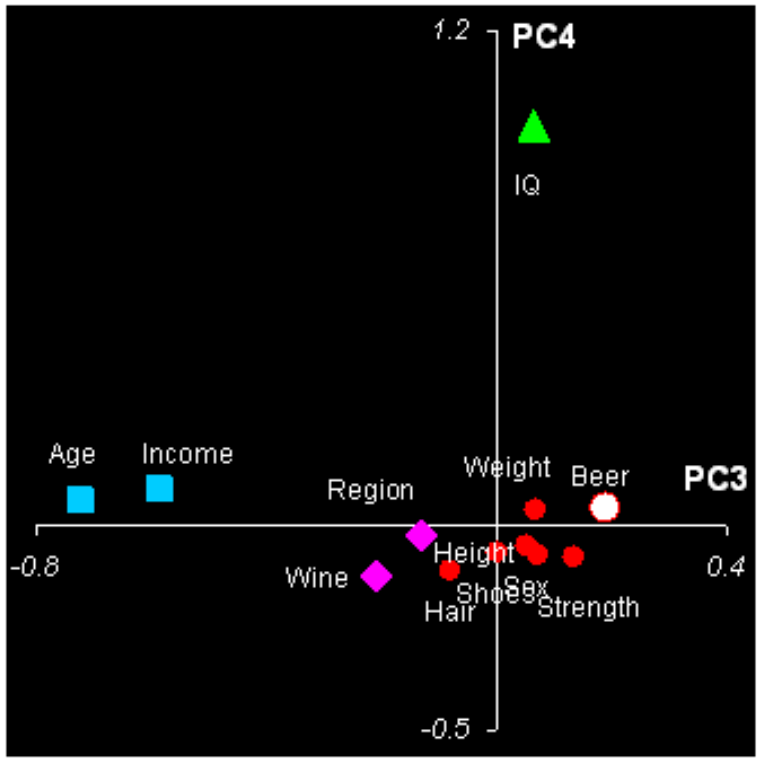
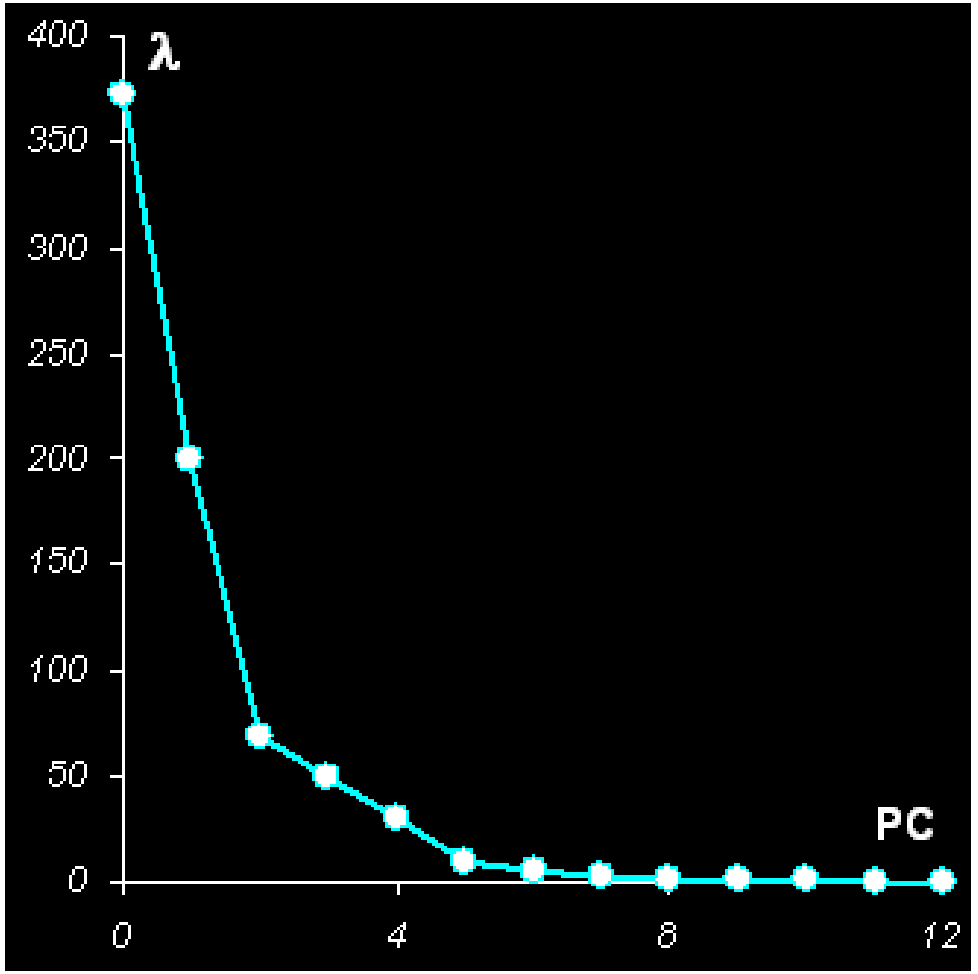


График «каменистой осыпи» (scree plot)



Диагностика наблюдения

Одной из основных диагностических задач является обнаружение разного рода выбросов, потому что они могут:

- искажать классический (ненадежный) PCA
- раскрыть что-то интересное о процессе (выбросы не должны быть ошибками измерения)

Диагностика наблюдения

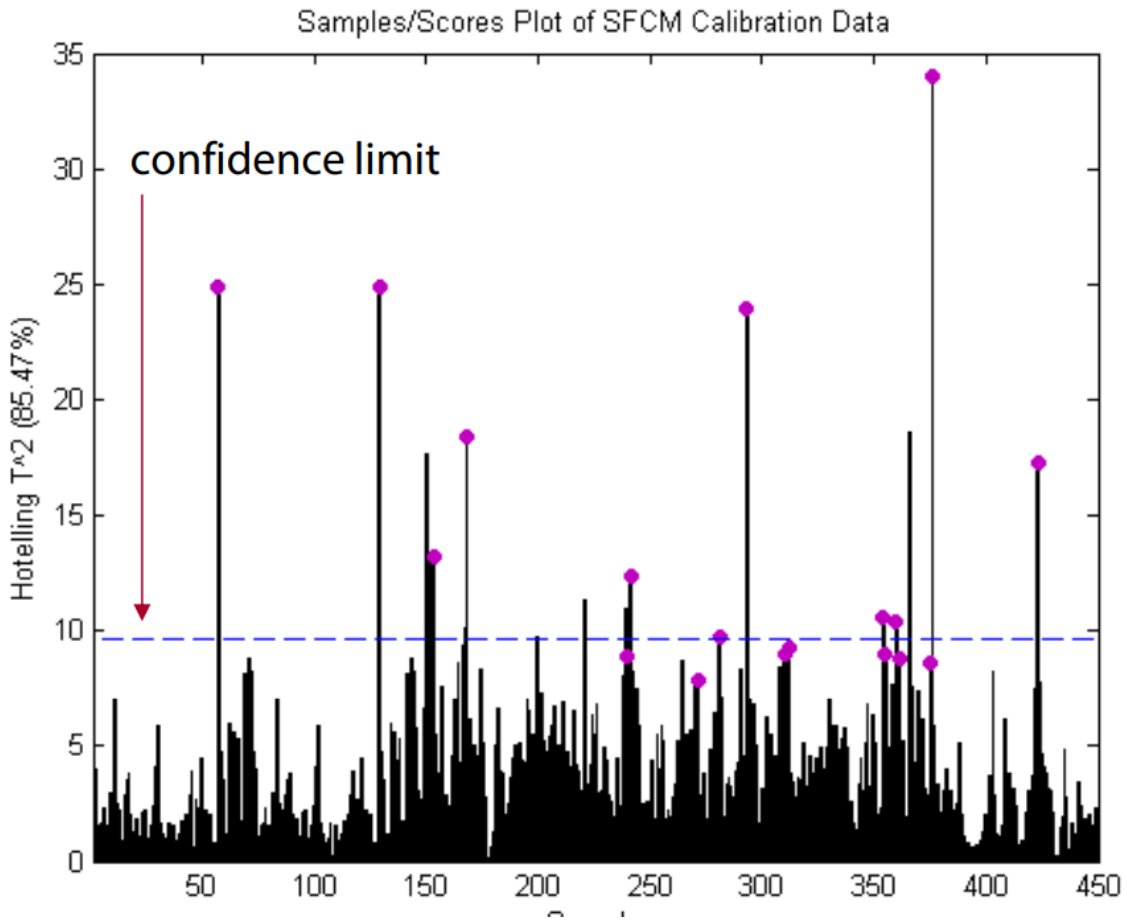
В соответствии с интерпретацией PCA, по существу существует два вида выбросов, выбросы которые приводят к :

- исключительно большие значения главных компонент для некоторого наблюдения
- большой остаток Q для некоторых наблюдений, такие выбросы ортогональны пространству, покрытому главными компонентами

Статистика Хотелинга

Статистика Хотеллинга является многомерным аналогом Т - статистики Стьюдента. Она играет важную роль в многомерной проверке гипотез и построении доверительной области, так же как и критерий Стьюдента.

Пример нахождения выборки с помощью статистики Хотелинга



Анализ качества PCA модели

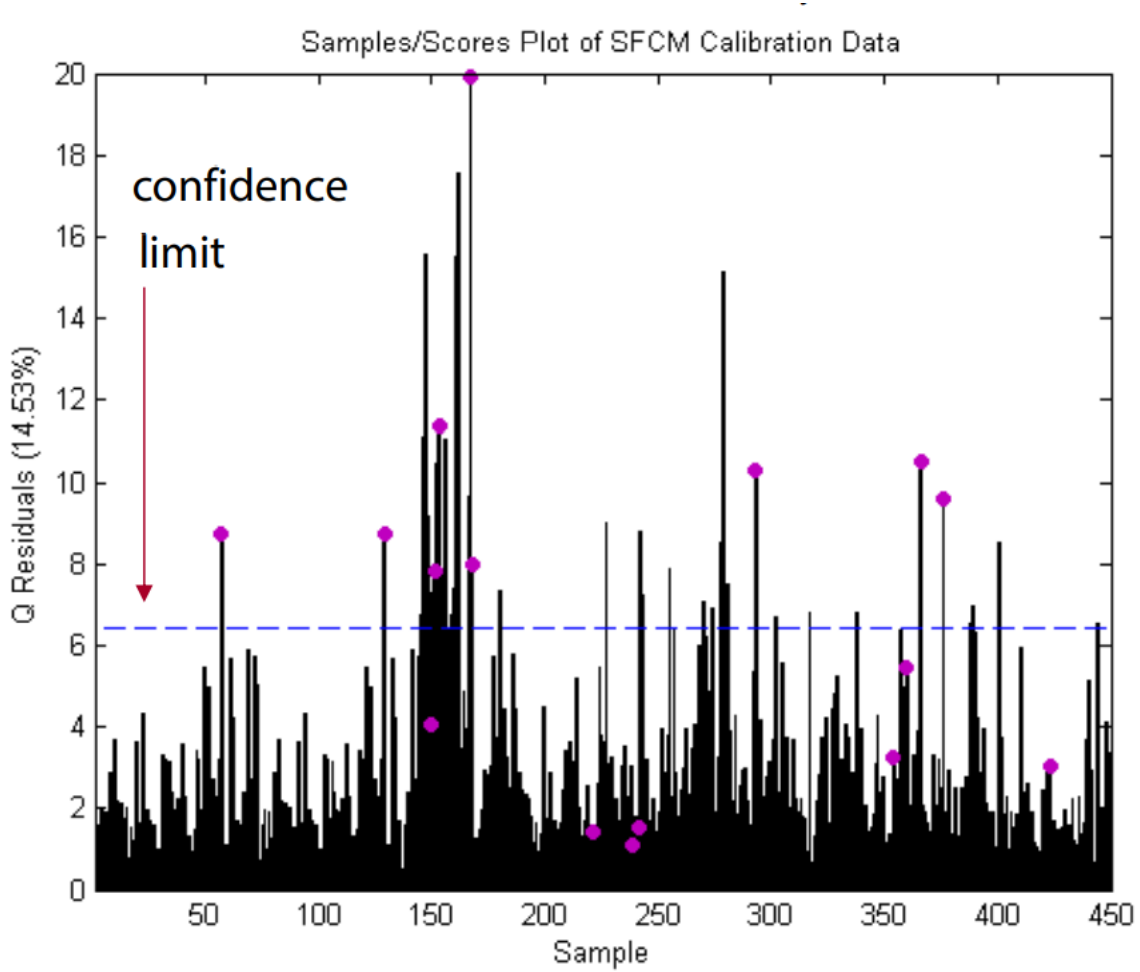
Большие остаточные выбросы могут быть обнаружены путем построения графика остатков Q. Для конкретного i-ого наблюдения, остаток(смещение) Q определяется как:

$$Q_i = \mathbf{e}_i^{\mathrm{T}}\mathbf{e}_i = \mathbf{x}_i^{\mathrm{T}}(\mathbf{I} - \mathbf{P}\mathbf{P}^{\mathrm{T}})\mathbf{x}_i$$

где $\mathbf{e}_i^{\mathrm{T}}$ - является элементом остаточной матрицы $\mathbf{E} = \mathbf{X} - \mathbf{T}\mathbf{P}^{\mathrm{T}}$

Статистика Q показывает насколько достоверно каждое наблюдение соответствует модели PCA.

Пример нахождения выборки с помощью статистики Q



Диагностика модели

С помощью диагностики модели мы пытаемся оценить, насколько модель будет соответствовать новым данным, которые не используются для построения модели. Из-за проблем, связанных с переобучением, важно найти хороший компромисс между соответствием модели и предсказуемостью.

В случае главных компонент

- этот компромисс включает в себя количество основных компонентов
- основным инструментом принятия решений является перекрестная проверка (CV)

Кросс-валидация

Кросс-валидация выполняет две важные функции:

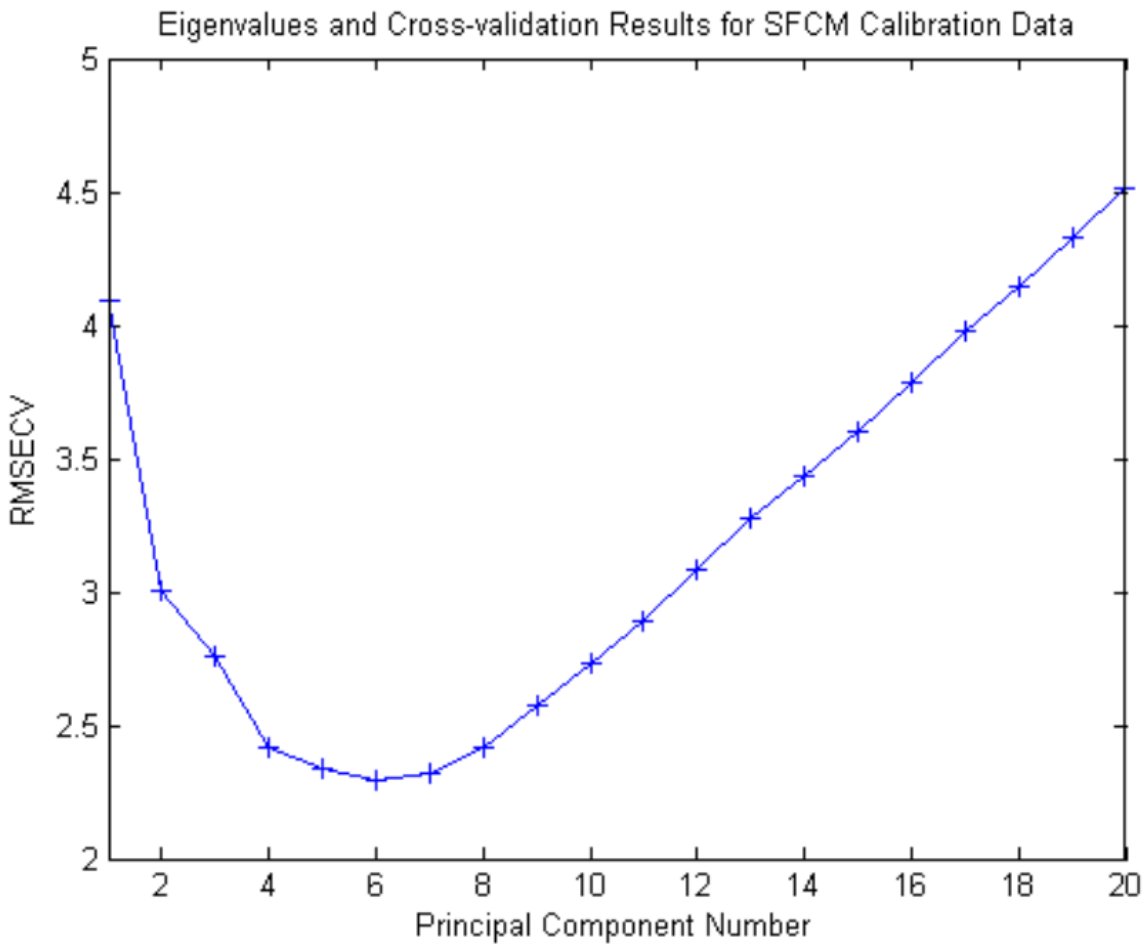
- позволяет оценить оптимальную сложность модели
- позволяет оценить производительность модели при ее применении к новым данным

Основная идея перекрестной проверки состоит в том, чтобы разделить данные на

- Обучающий набор наблюдений, который используется для оптимизации параметров
- Валидационный набор наблюдений, который используется для структурных решений
- тестовый набор , который используется для оценки производительности

Кросс-валидация. Среднеквадратическая ошибка

$$E_{\mathrm{RMS}}^{\mathrm{CV}} = \sqrt{\frac{1}{np}\sum_{m=1}^{s}\sum_{i=1}^{n/s}\sum_{j=1}^{p} e_{ij}^{2}}$$



Спасибо за внимание!