Метод главных компонент

1 Метод главных компонент в случае $k \geq 2$ переменных

1.1 Основные понятия

Рассмотрим ситуацию с числом признаков, большим двух. В большинстве случаев эти признаки коррелированы, поскольку являются результатами частных измерений некоторых интегральных характеристик. В медицине эти скрытые признаки чаще всего отражают степень тяжести того или иного заболевания и мощность защитных сил организма и складываются из влияния целого комплекса характеристик. Часто оказывается, что эти интегральные характристики описываются главными компонентами. Идея метода главных компонент состоит в том, чтобы заменить k-мерную случайную величину при наименьшей потере информативности на m- мерную ($m < k$).

Что означает наименьшая потеря информации? Эффективность или информативность этих m линейных функций зависит от того, в какой степени они дают возможность реконструировать или восстановить k первоначальных величин. Один из методов реконструкции этой первоначальной случайной величины состоит в построении ее наилучшего предиктора на основе m линейных функций. Оказывается, что наилучшим выбором линейных функций является выбор первых m главных компонент.

Определим понятие главной компоненты. Пусть имеется k признаков $X_1, \ldots, X_k$. Первой главной компонентой Y_1 называется сохраняющая расстояние между точками линейная комбинация исходных признаков

$$Y_1 = \alpha_{11}X_1 + \ldots + \alpha_{k1}X_k,$$

где коэффициенты $\alpha_{11}, \ldots, \alpha_{k1}$ выбираются таким образом, чтобы дисперсия $D(Y_1) = \lambda_1$ была максимальной. Это соответствует тому, что по первой главной компоненте индивиды должны отличаться наибольшим образом. Вторая главная компонента также является линейной комбинацией исходных признаков:

$$Y_2 = \alpha_{12}X_1 + \ldots + \alpha_{k2}X_k,$$

где коэффициенты $\alpha_{12}, \ldots, \alpha_{k2}$ выбираются таким образом, что компоненты Y_1 и Y_2 некоррелированы, а дисперсия $D(Y_2) = \lambda_2$ является максимальной из всех линейных комбинаций, некоррелированных с Y_1, то есть вторая компонента должна нести наибольшую новую информацию, не имеющую отношения к первой главной компоненте. Аналогично строятся остальные главные компоненты

$$Y_j = \sum_{i=1}^{k} \alpha_{ij}X_i, \;\; j = 1, \ldots, k.$$

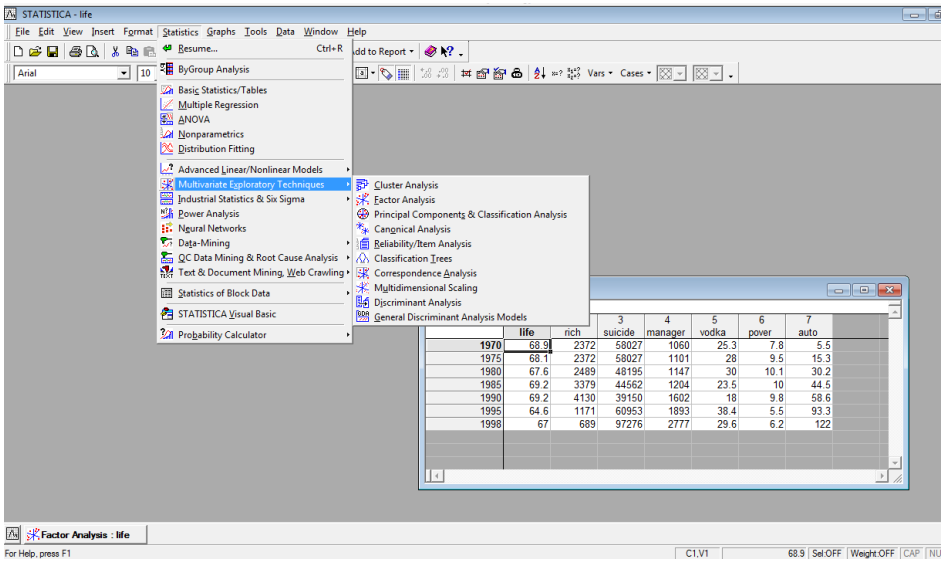
Суммарная дисперсия остается неизменной:

$$V = D(X_1) + \ldots D(X_k) = \lambda_1 + \ldots + \lambda_k.$$

Значимость главных компонент определяется долей объясняемой ими дисперсии, равной $\frac{\lambda_i}{V} \cdot 100\%$.

Факторами называются нормированные главные компоненты: $Y_j/\sqrt{\lambda_i}$.

Для того чтобы сделать факторную обработку в *StatSoft*, нужно запустить программу *Statistics - Multivariate Exploratory Techniques - Factor Analysis*. Далее нужно выбрать переменные, по которым строятся факторы (*Variables*). Нажимаем *OK*.



	life	rich	3 suicide	4 manager	5 vodka	6 pover	7 auto
1970	68.9	2372	58027	1060	25.3	7.8	5.5
1975	68.1	2372	58027	1101	28	9.5	15.3
1980	67.6	2489	48195	1147	30	10.1	30.2
1985	69.2	3379	44562	1204	23.5	10	44.5
1990	69.2	4130	39150	1602	18	9.8	58.6
1995	64.6	1171	60953	1893	38.4	5.5	93.3
1998	67	689	97276	2777	29.6	6.2	122

Рис. 1: Пример организации данных и начала вычислений в Statistica.

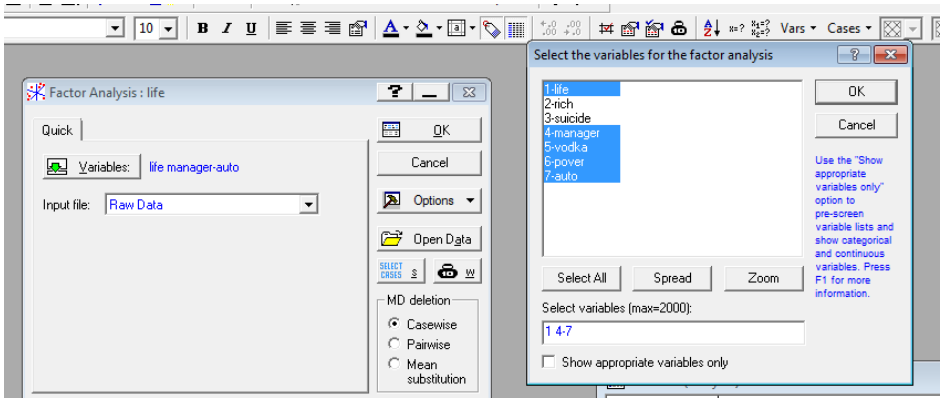


Рис. 2: Отбор переменных для анализа главных компонент.

В окошке $Max.no.of\,factors$ нужно указать необходимое для анализа число факторов, оно должно быть больше нуля и не больше количества участвующих в анализе признаков.

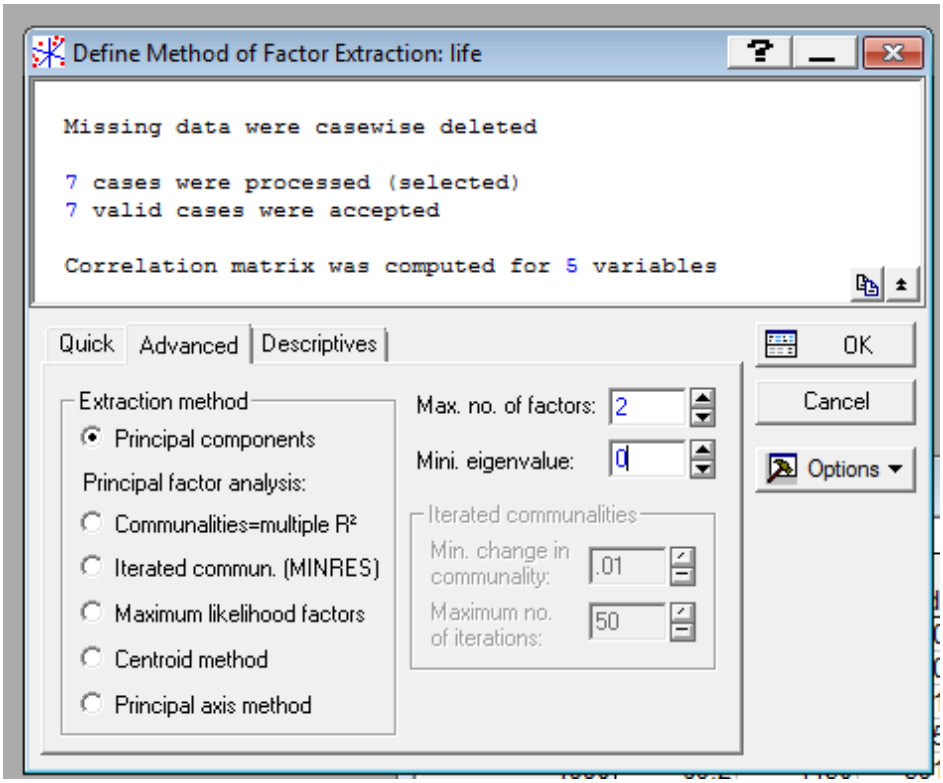


Рис. 3: Выбор количества первых главных компонент и наименьшего собственного числа.

В окошке *Mini. eigenvalue* лучше сразу поставить 0. Читаем, сколько переменных и сколько наблюдений участвует в анализе. Далее *OK*. Для правильного чтения результатов необходимыми являются сведения, излагаемые далее.

1.2 Вычисление коэффициентов главных компонент

Пусть признаки $X = (X_1, \ldots, X_k)^T$ центрированы $EX_i = 0$ и имеют ковариационную матрицу $\Sigma = EXX^T$. Обозначим через A_i собственные векторы матрицы Σ, соответствующие собственным числам λ_i

$$\Sigma A_i = \lambda_i A_i, \quad A_i^T A_j = \begin{cases} 1, & \text{при } i = j, \\ 0, & \text{при } i \neq j \end{cases}, \quad i = 1, \ldots, k. \tag{1}$$

Упорядочим собственные числа по убыванию: $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k$. Соберем все собственные вектора в одну ортогональную матрицу

$$\mathcal{A} = [A_1, \ldots, A_k] = \begin{bmatrix} a_{11}, & \ldots, & a_{1k} \\ \cdot & \cdot & \cdot \\ a_{k1}, & \ldots, & a_{kk} \end{bmatrix}, \quad \mathcal{A}^T\mathcal{A} = I = \begin{bmatrix} 1, & \ldots, & 0 \\ \cdot & \cdot & \cdot \\ 0, & \ldots, & 1 \end{bmatrix}$$

и определим главную компоненту как

$$Y_j = A_j^T X = \sum_{i=1}^{k} a_{ij} X_i. \tag{2}$$

Для всех главных компонент справедливо выражение

$$Y = \begin{bmatrix} Y_1 \\ \cdot \\ Y_k \end{bmatrix} = \begin{bmatrix} A_1^T X \\ \cdot \\ A_k^T X \end{bmatrix} = \mathcal{A}^T X, \quad \text{откуда} \quad X = \mathcal{A} Y. \tag{3}$$

Теорема (о некоррелированности главных компонент и инвариантности суммарной дисперсии).

$$1.\ EY_iY_j = \begin{cases} \lambda_j, & \text{при } i = j, \\ 0, & \text{при } i \neq j \end{cases}, \qquad 2.\ \sum_{i=1}^{k} DX_i = \sum_{j=1}^{k} DY_j.$$

Доказательство.

$$1. EY_iY_j = EY_iY_j^T \overset{(2)}{=} EA_i^T X (A_j^T X)^T = A_i^T \Sigma A_j \overset{(1)}{=}$$
$$= A_i^T \lambda_j A_j = \begin{cases} \lambda_j, & \text{при } i = j, \\ 0, & \text{при } i \neq j. \end{cases}$$

Следовательно, дисперсия главной компоненты равна соответственному собственному числу ковариационной матрицы: $DY_j = \lambda_j$. Из упорядоченности собственных чисел получаем упорядоченные по убыванию дисперсии главные компоненты:

$$DY_1 \geq DY_2 \geq \ldots \geq DY_k.$$

$$2.\ DX_1 + \ldots + DX_k = \sum_{i=1}^{k} EX_i^2 = EX^T X = \overset{(3)}{=}$$
$$= EY^T \mathcal{A}^T \mathcal{A} Y = EY^T Y = \sum_{j=1}^{k} EY_j^2 = DY_1 + \ldots + DY_k.$$

1.3 Факторные нагрузки и формула восстановления

Нормированные главные компоненты $G_i = \frac{Z_i}{\sqrt{\lambda_i}},\ i = 1, \ldots, k$, называются факторами.

Корреляция $\beta_{ij} = cor(X_i, Y_j)$ между признаком X_i и главной компонентой Y_j называется факторной нагрузкой. Пусть σ_i – стандартное отклонение X_i, λ_j – дисперсия главной компоненты Y_j. Тогда факторная нагрузка имеет вид:

$$\beta_{ij} = cor(X_i, Y_j) = \frac{EX_iY_j}{\sigma_i \sqrt{\lambda_j}} \overset{(3)}{=} \frac{E \sum\limits_{t=1}^{k} \alpha_{it} Y_t Y_j}{\sigma_i \sqrt{\lambda_j}} = \frac{\alpha_{ij} \sqrt{\lambda_j}}{\sigma_i} \tag{4}$$

Формулу восстановления X_i по главным компонентам можно выразить через факторы $G_j = Y_j/\sqrt{\lambda_j}$:

$$X_i = \sum_{j=1}^{k} \alpha_{ij} Y_j = \sigma_i \sum_{j=1}^{k} \frac{\beta_{ij}}{\sqrt{\lambda_j}} G_j \sqrt{\lambda_j} = \sigma_i \sum_{j=1}^{k} \beta_{ij} G_j \,.$$

При восстановлении переменных по m главным компонентам, меньшему количества исходных признаков k, значения признаков могут восстанавливаться с ошибками. Чем больше вклад используемых в восстановлении главных компонент, тем меньше ошибки восстановления.

2 Пример вычислений в R

2.1 Счет по формулам

Считываем данные.

```
> life <- read.csv(
+   "~/Documents/share/R/lessions/A_PrincipleComponents/life.csv", sep=";")
> data<-data.frame(L=life$life,
+                  M=life$manager,
+                  P=life$pover,
+                  A=life$auto,
+                  V=life$vodka)
> row.names(data)<-life$X
```

	L	M	P	A	V
1970	68.90	1060	7.80	5.50	25.30
1975	68.10	1101	9.50	15.30	28.00
1980	67.60	1147	10.10	30.20	30.00
1985	69.20	1204	10.00	44.50	23.50
1990	69.20	1602	9.80	58.60	18.00
1995	64.60	1893	5.50	93.30	38.40
1998	67.00	2777	6.20	122.00	29.60

Таблица 1: Данные о средней продолжительности жизни и сопутствующих факторах.

Функция в R для факторного анализа.

```
> pc<-princomp(scale(data),cor=TRUE)
```

Проверим, что вычисляется при помощи функции *princomp*. Нормируем и центрируем данные, вычисляем корреляционную матрицу

```
> data.0<-apply(data,2,function(x)(x-mean(x))/sd(x))
> Sigma<-cov(data.0)
> xtable(Sigma,caption="Корреляционная матрица.")
```

	L	M	P	A	V
L	1.00	-0.50	0.77	-0.60	-0.93
M	-0.50	1.00	-0.70	0.95	0.30
P	0.77	-0.70	1.00	-0.67	-0.68
A	-0.60	0.95	-0.67	1.00	0.37
V	-0.93	0.30	-0.68	0.37	1.00

Таблица 2: Корреляционная матрица.

Вычисляем собственные числа корреляционной матрицы и собственные векторы.

```
> ei<-eigen(Sigma)
> xtable(rbind(ei$values,cumsum(ei$values)/sum(ei$values)*100),
+ caption="Собственные числа и суммарный  вклад компонент в общую дисперсию.")
```

	1	2	3	4	5
1	3.60	1.09	0.24	0.05	0.02
2	72.00	93.80	98.70	99.63	100.00

Таблица 3: Собственные числа и суммарный вклад компонент в общую дисперсию.

```
> xtable(ei$vectors,caption="Собственные векторы.")
```

	1	2	3	4	5
1	0.47	-0.38	0.30	0.50	-0.54
2	-0.43	-0.54	-0.03	0.54	0.48
3	0.48	-0.04	-0.85	0.17	0.13
4	-0.45	-0.47	-0.38	-0.33	-0.57
5	-0.41	0.59	-0.20	0.57	-0.35

Таблица 4: Собственные векторы.

Убедимся в том, что сумма собственных чисел совпадает с суммарной дисперсий нормированных признаков.

```
> c(sum(ei$values),sum(diag(cov(data.0))) )
```

```
[1] 5 5
```

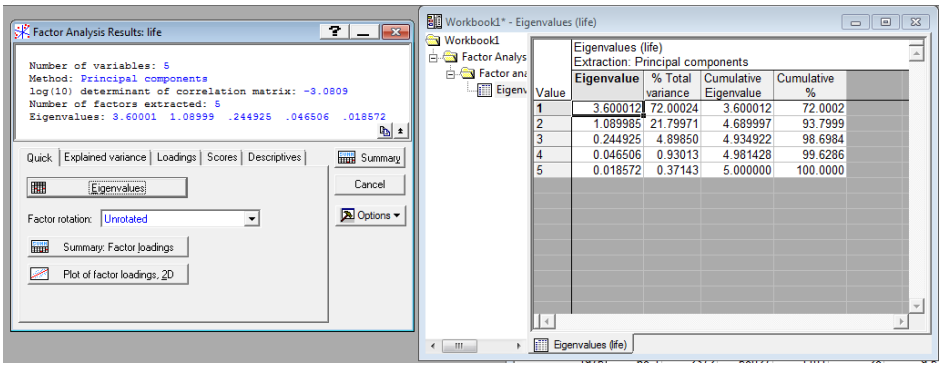


Рис. 4: Собственные числа корреляционной матрицы в Statistica.

Вычисляем главные компоненты,

```
> Scores<-data.0 %*% ei$vectors
```

проверяем, действительно ли дисперсии главных компонент совпадают с собственными числами корреляционной матрицы,

```
> apply(Scores,2,var)
```

```
[1] 3.60001200 1.08998530 0.24492480 0.04650638 0.01857152
```

```
> ei$values
```

```
[1] 3.60001200 1.08998530 0.24492480 0.04650638 0.01857152
```

затем факторы (нормированные главные компоненты)

```
> factors<-apply(Scores,2, function(x) x/sd(x))
> xtable(factors, caption="Значения главных компонент")
```

	1	2	3	4	5
1970	0.60	0.47	2.01	0.17	-0.06
1975	0.54	0.71	-0.16	0.66	0.89
1980	0.37	0.79	-1.28	0.67	0.36
1985	0.72	-0.34	-0.46	-0.11	-2.02
1990	0.66	-1.30	-0.23	-1.41	0.98
1995	-1.59	1.02	-0.04	-1.23	-0.21
1998	-1.31	-1.34	0.16	1.26	0.07

Таблица 5: Значения главных компонент

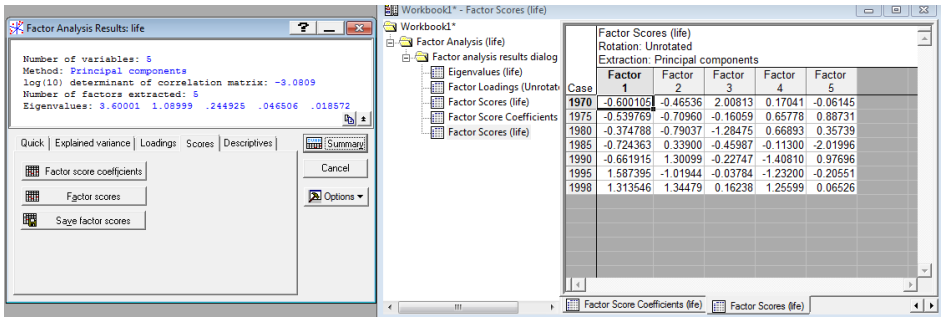


Рис. 5: Значения факторов в Statistica.

и факторные нагрузки как коэффициенты корреляции между признаками и факторами.

```
> Matr<-apply(Scores,2,function(y)apply(data.0,2,function(x)cor(x,y)))
> xtable(Matr, caption="Матрица факторных нагрузок")
```

	1	2	3	4	5
L	0.90	-0.40	0.15	0.11	-0.07
M	-0.82	-0.56	-0.02	0.12	0.07
P	0.90	-0.04	-0.42	0.04	0.02
A	-0.85	-0.49	-0.19	-0.07	-0.08
V	-0.77	0.61	-0.10	0.12	-0.05

Таблица 6: Матрица факторных нагрузок

Variable	Factor Loadings (Unrotated) (life) Extraction: Principal components (Marked loadings are >.700000) **Factor 1**	Factor 2	Factor 3	Factor 4	Factor 5
life	-0.896110	0.397903	0.147006	0.107621	-0.073943
manager	0.815034	0.563601	-0.016136	0.116128	0.065783
vodka	0.772424	-0.613259	-0.099476	0.122703	-0.048198
pover	-0.904844	0.044736	-0.421464	0.036020	0.018072
auto	0.846958	0.485721	-0.188483	-0.071307	-0.078275
Expl.Var	3.600012	1.089985	0.244925	0.046506	0.018572
Prp.Totl	0.720002	0.217997	0.048985	0.009301	0.003714

Рис. 6: Факторные нагрузки в Statistica.

2.2 Использование функции princomp

Проверяем счет при помощи встроенной функции princomp. В R собственные числа или дисперсии главных компонент те же

```
> c(pc$sdev^2,sum(pc$sdev^2))
```

```
    Comp.1     Comp.2     Comp.3     Comp.4     Comp.5
3.60001200 1.08998530 0.24492480 0.04650638 0.01857152

5.00000000
```

Вместо нагрузок *pc$loadings* выдаются собственные вектора.

```
> A<-pc$loadings[seq(ncol(data.0)),seq(ncol(data.0))]
> xtable(A, caption="Собственные вектора  princomp")
```

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5
L	0.47	-0.38	0.30	0.50	-0.54
M	-0.43	-0.54	-0.03	0.54	0.48
P	0.48	-0.04	-0.85	0.17	0.13
A	-0.45	-0.47	-0.38	-0.33	-0.57
V	-0.41	0.59	-0.20	0.57	-0.35

Таблица 7: Собственные вектора princomp

Чтобы получить факторные нагрузки, нужно элементы с.в. умножить на корень из соответствующего собственного числа.

```
> MatrPC<-apply(rbind(A,pc$sdev),2,function(x)x[-length(x)]*x[length(x)])
> xtable(MatrPC,caption="Вычисление факторных нагрузок в  princomp")
```

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5
L	0.90	-0.40	0.15	0.11	-0.07
M	-0.82	-0.56	-0.02	0.12	0.07
P	0.90	-0.04	-0.42	0.04	0.02
A	-0.85	-0.49	-0.19	-0.07	-0.08
V	-0.77	0.61	-0.10	0.12	-0.05

Таблица 8: Вычисление факторных нагрузок в princomp

Чтобы получить значения факторов в *princomp*, нужно нормировать *scores*.

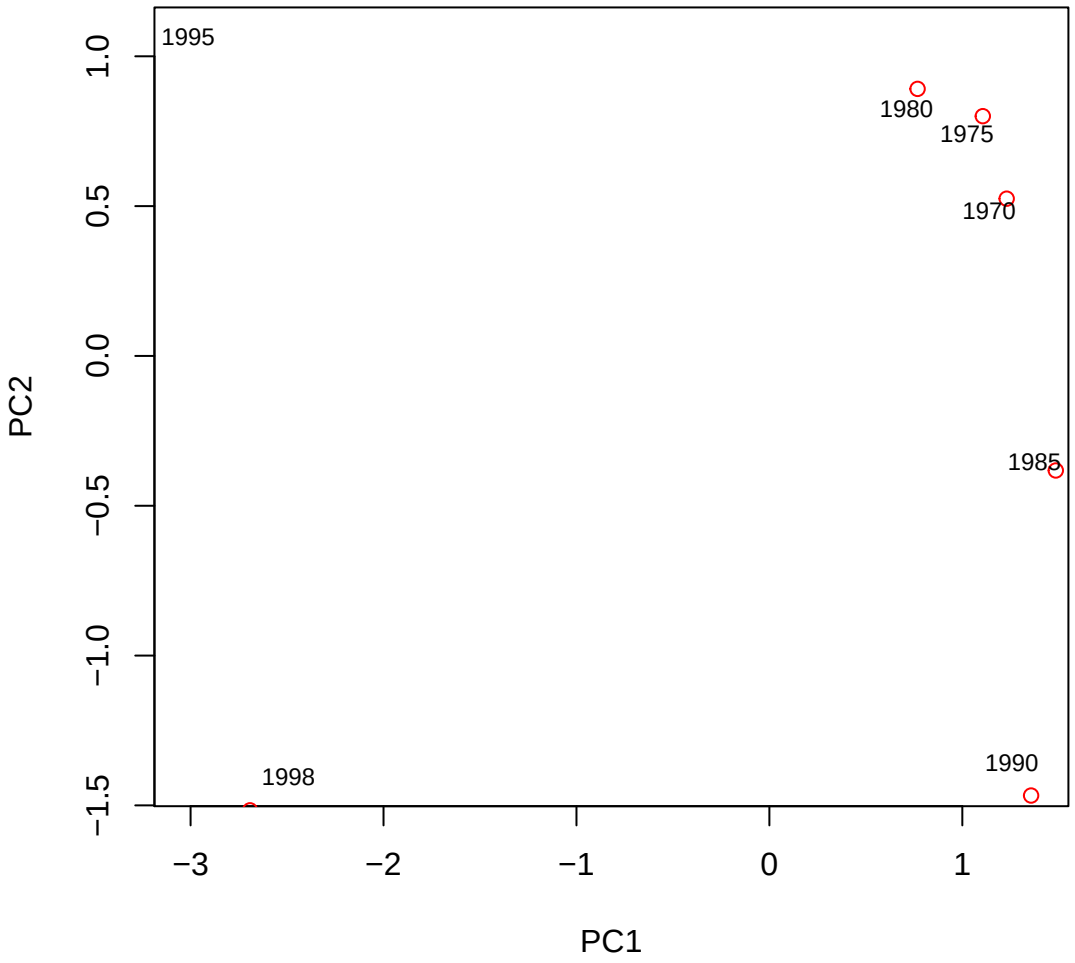
```
> xtable(scale(pc$scores), caption="Значения факторов по  princomp")
```

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5
1970	0.60	0.47	2.01	0.17	-0.06
1975	0.54	0.71	-0.16	0.66	0.89
1980	0.37	0.79	-1.28	0.67	0.36
1985	0.72	-0.34	-0.46	-0.11	-2.02
1990	0.66	-1.30	-0.23	-1.41	0.98
1995	-1.59	1.02	-0.04	-1.23	-0.21
1998	-1.31	-1.34	0.16	1.26	0.07

Таблица 9: Значения факторов по princomp

Значения главных компонент указывают на более "благополучные"в смысле повышения доходов бедных годы и годы, отличающиеся объемами продаваемой водки.

```
> plot(Scores[,seq(2)],type="n",xlab="PC1",ylab="PC2")
> lines(pc$scores[,seq(2)],col=2,type="p")
> text(Scores[,seq(2)],as.character(life[,1]),cex=0.75)
```



2.3 Восстановление переменных по первым двум факторам

Для восстановления данных по главным компонентам используем формулу произведения матрицы факторных нагрузок на значения факторов.

```
> sigma2<-apply(data,2,var)
> Rest<-function(k,sigma2,Matr,factors,data)
+   {
+   XX<-Matr[,seq(k)]%*%t(factors[,seq(k)])
+ XX.1<-apply(cbind(XX,sqrt(sigma2)),1,function(x)x[-length(x)]*x[length(x)])
+ XX.2<-apply(rbind(XX.1,colMeans(data)),2,function(x)x[-length(x)]+x[length(x)])
+ return(XX.2)}
```

```
> plot(data[,1],type="b",ylim=c(60,70))
> lines(Rest(2,sigma2,Matr,factors,data)[,1],lty=2,col=2)
> lines(Rest(2,sigma2,MatrPC,scale(pc$scores),data)[,1],lty=2,col=2)
```

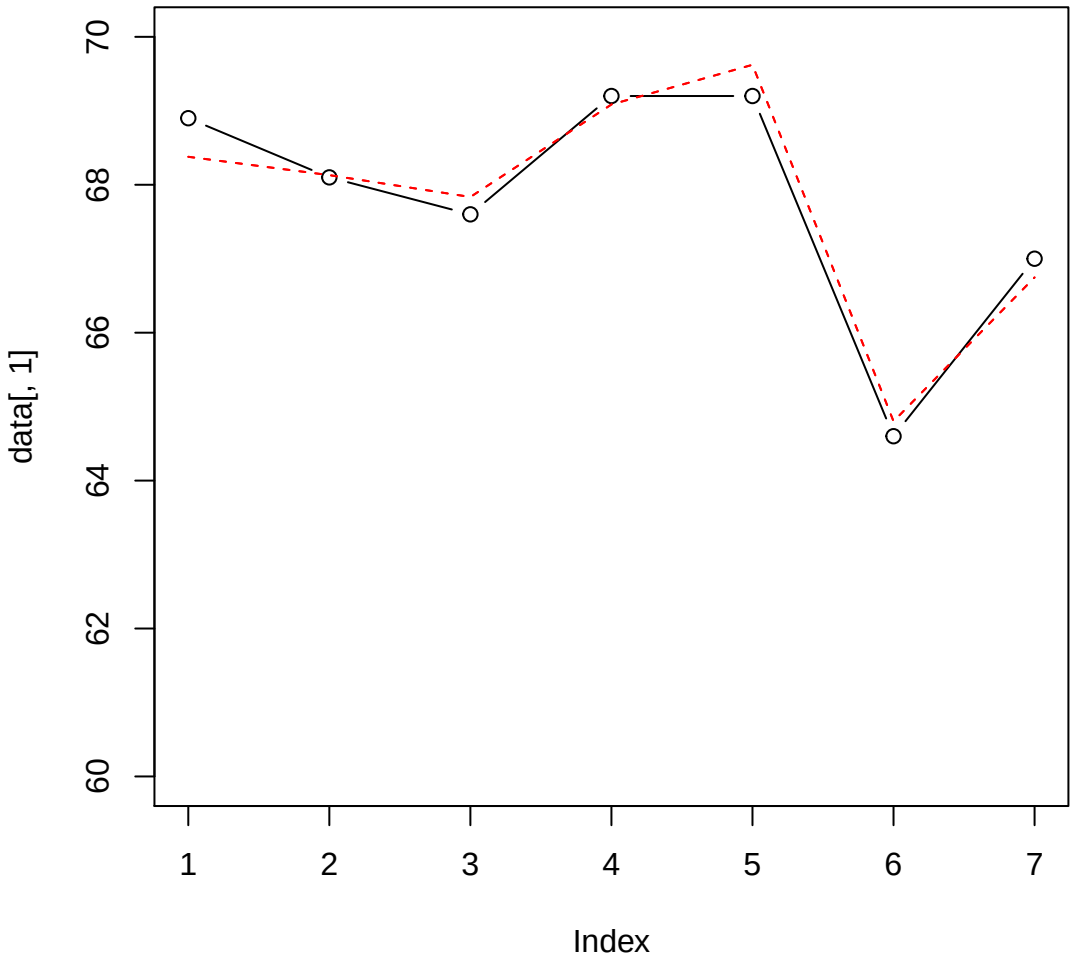


Рис. 7: Восстановление средней продолжительности жизни по первому и второму факторам.

2.4 Интерпретация факторов

Признаки: L – средняя продолжительность жизни; M – количество чиновников; A – количество автомобилей; P – доходы бедных; V – объемы продажи водки. Вклад первого фактора равен 72%. Значения в i-й строке и j-м столбце соответствуют коэффициенту

корреляции между i-м признаком и j-й главной компонентой. Чем больше первый фактор, тем больше продолжительность жизни и доходы бедных, меньше чиновников и автомобилей и не много водки – фактор какого-то благополучия. Во всяком случае корреляция первого фактора с числом самоубийц отрицательная.

```
> xtable(cor(cbind(life[,c("rich","auto","suicide")],factors[,seq(2)])),
+        caption="Корреляционная матрица факторов с другими признаками.")
```

	rich	auto	suicide	1	2
rich	1.00	-0.55	-0.87	0.85	-0.21
auto	-0.55	1.00	0.63	-0.85	-0.49
suicide	-0.87	0.63	1.00	-0.72	-0.25
1	0.85	-0.85	-0.72	1.00	0.00
2	-0.21	-0.49	-0.25	0.00	1.00

Таблица 10: Корреляционная матрица факторов с другими признаками.

Для интерпретации второго фактора, в который самый значимый вклад вносит продажа водки, добавим признаки, означающие число браков и разводов. Регрессионный анализ указывает на отрицательную зависимость фактора продаж водки с браками и положительную с разводами.

```
> ost_life0 <- read.table(
+ "~/Documents/share/R/lessions/A_PrincipleComponents/ost_life0.csv",
+ header=TRUE, quote="\"")
> data.lm<-cbind(f2=factors[,2],data,
+                marriages=ost_life0[,5],
+                devorce=ost_life0[,6])
> LM<-lm(f2~.,subset(data.lm,select=c(f2,marriages,devorce)))
> summary(LM)
```

```
Call:
lm(formula = f2 ~ ., data = subset(data.lm, select = c(f2, marriages,
    devorce)))

Residuals:
    1970     1975     1980     1985     1990     1995
 0.74390  0.33086 -0.05068 -0.61651 -1.23760  1.03813
    1998
-0.20810

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) -4.8960     3.4195 -1.432    0.225
marriages    0.2610     0.1983  1.316    0.258
devorce      0.6604     0.7555  0.874    0.431

Residual standard error: 0.9615 on 4 degrees of freedom
Multiple R-squared:  0.3836,        Adjusted R-squared:  0.07546
F-statistic: 1.245 on 2 and 4 DF,  p-value: 0.3799
```