

---

# Redução de Dimensionalidade

Fabício J. Barth

fabricao.barth@gmail.com

Agosto de 2016

---

---

# Sumário

- Justificativa
- Ideia principal
- Algoritmo PCA
- Escolhendo o número de componentes principais
- Alguns exemplos
- Considerações finais

---

## Justificativa

- Em muitas situações temos que trabalhar com datasets com muitas dimensões.

- 
- **Tais datasets são difíceis de serem visualizados.**

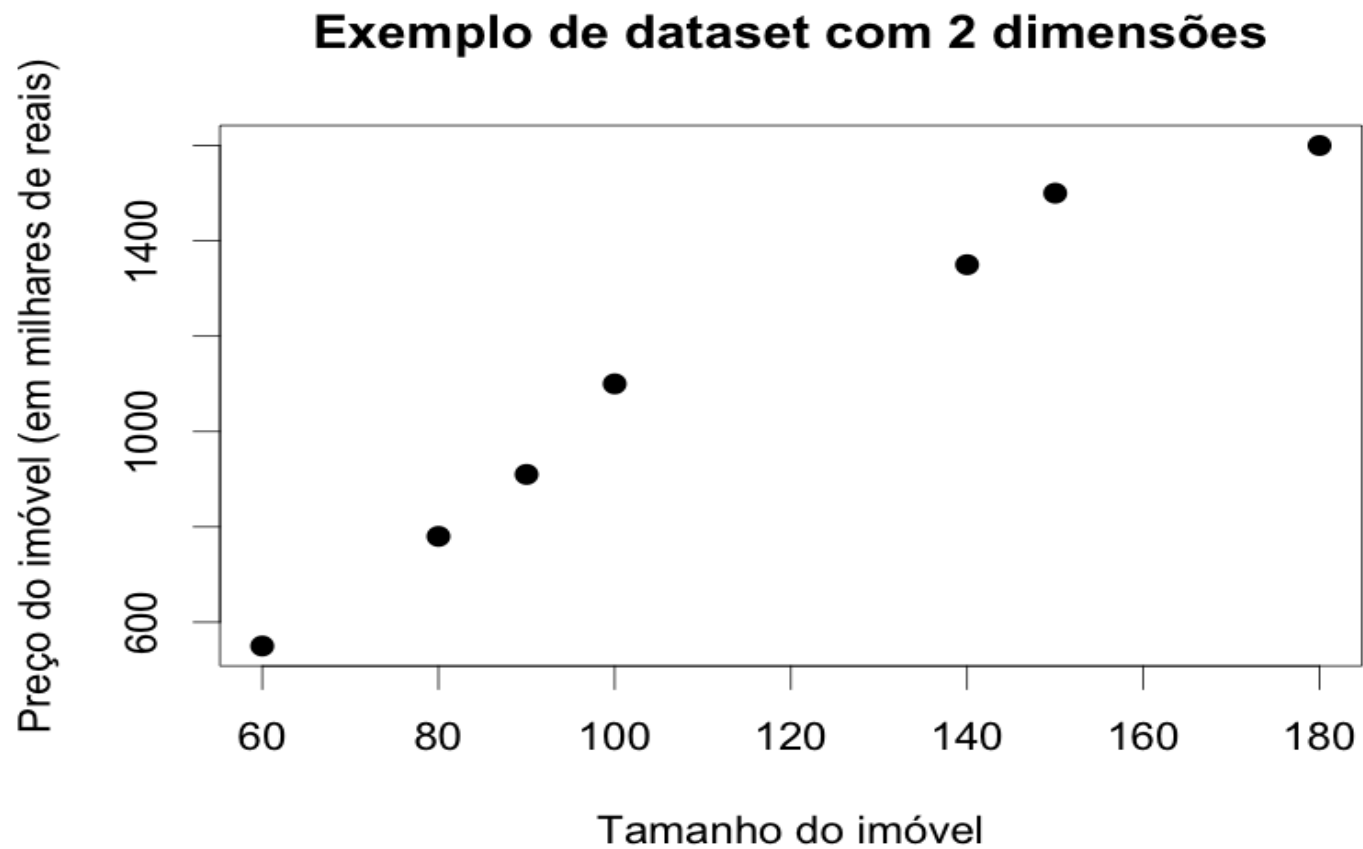
Conseguimos visualizar bem apenas datasets com duas ou três dimensões. Eventualmente, conseguimos fazer uso do *scatterplot* para visualizar mais dimensões. No entanto, quanto maior o número de visualizações, mais difícil é esta visualização.

- **Podem não ser adequados** para a criação de alguns modelos preditivos.
- **Ocupam muito espaço.** Ou seja, poderiam ser compactados.

---

# Ideia principal

# Exemplo

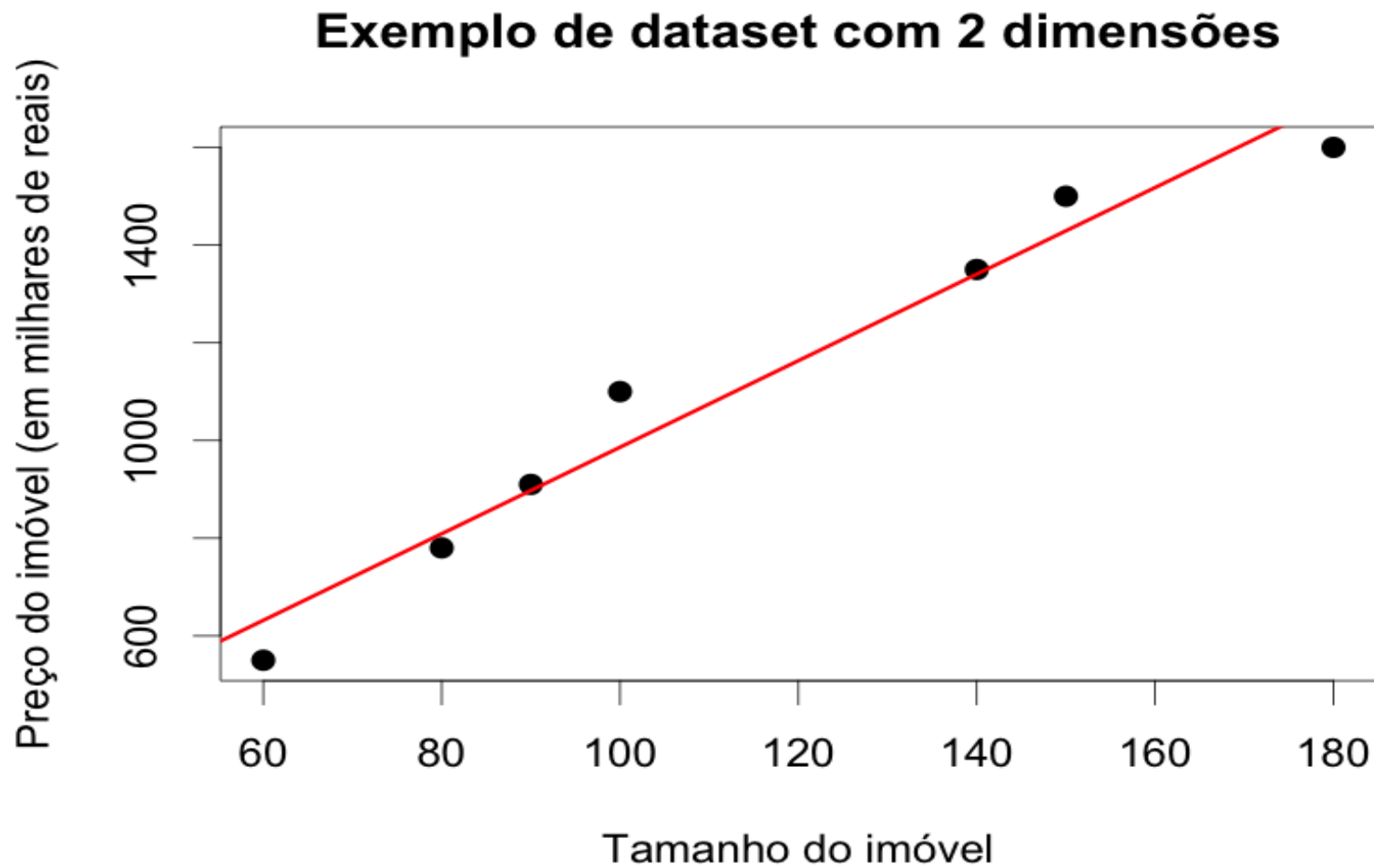


---

	tamanho	preco
1	60.00	550.00
2	80.00	780.00
3	90.00	910.00
4	100.00	1100.00
5	140.00	1350.00
6	150.00	1500.00

---

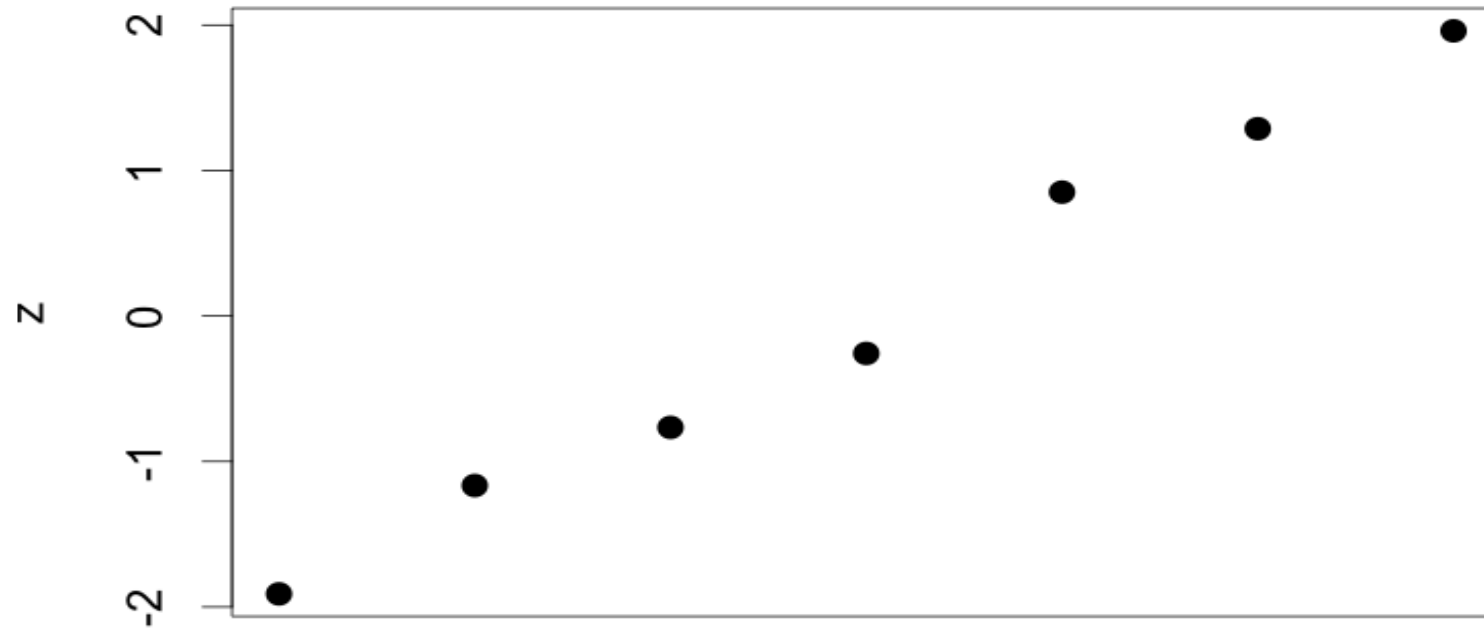
$$X \in \mathbb{R}^2$$



É necessário projetar os pontos  $X \in \mathbb{R}^2$  para  $z \in \mathbb{R}^1$  de tal forma que a distância entre os pontos em  $X$  para  $z$  seja a mínima possível.



## Pontos do vetor $z$



$$z = \{-1.9102, -1.1654, -0.7658, -0.2574, 0.8514, 1.2872, 1.9602\}$$

---

```
casas <- data.frame(tamanho=c(60,80,90,100,140,150,180),
                    preco=c(550,780,910,1100,1350,1500,1600))
pca <- prcomp(casas, scale. = TRUE)
summary(pca)
z <- pca$x[,1]
```

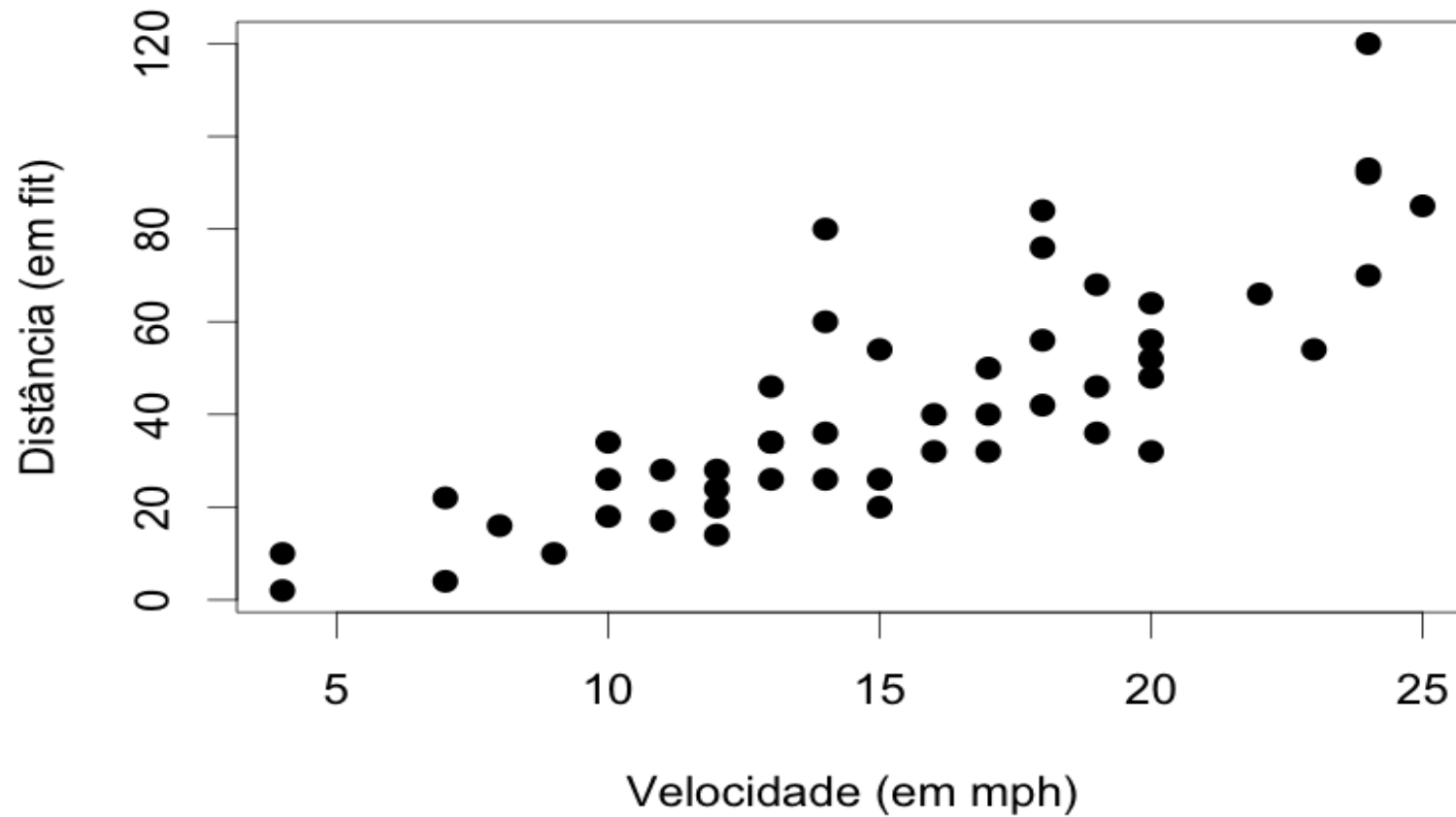
	PC1	PC2
Standard deviation	1.4074	0.1387
Proportion of Variance	0.9904	0.0096
Cumulative Proportion	0.9904	1.0000

```
> z
[1] -1.9102 -1.1654 -0.7658 -0.2574  0.8514  1.2872  1.9602
```

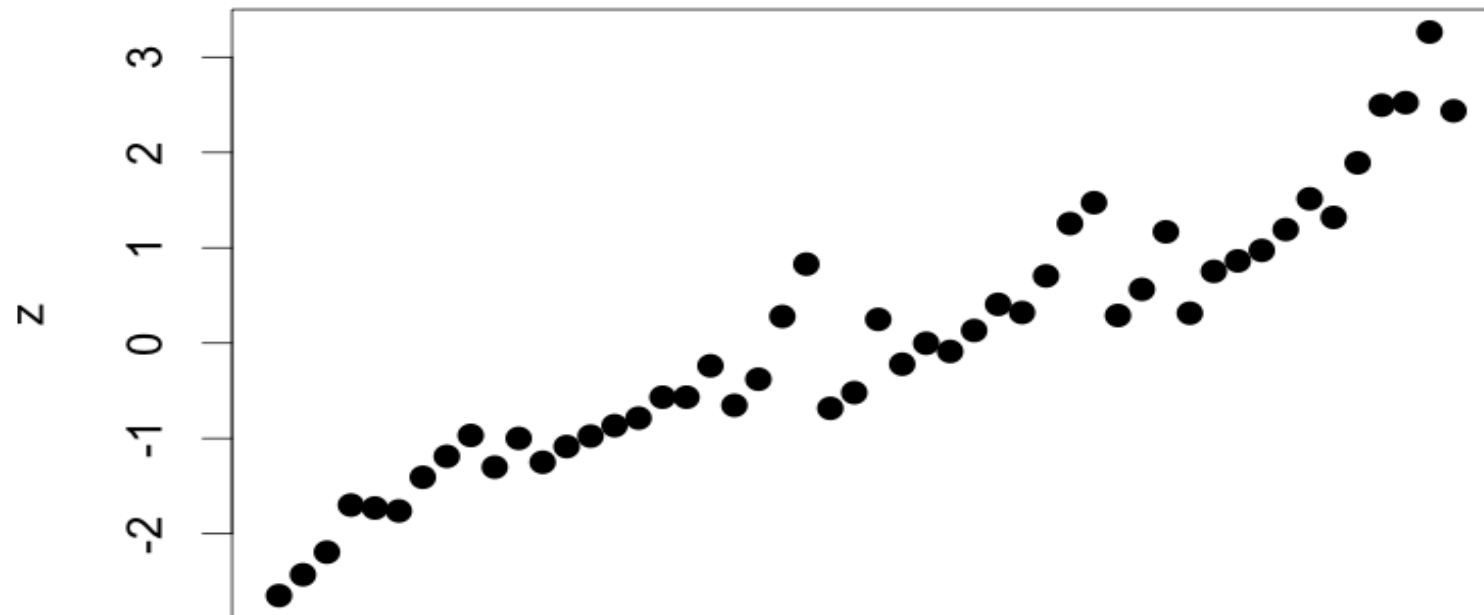
---

## Outro exemplo com duas dimensões

## Dados sobre velocidade e distância de frenagem de carro



## Pontos do vetor $z$ para o exemplo dos carros



---

```
data(cars)
pca <- prcomp(cars, scale. = TRUE)
summary(pca)
```

	PC1	PC2
Standard deviation	1.3442	0.4394
Proportion of Variance	0.9034	0.0965
Cumulative Proportion	0.9034	1.0000

---

## Um exemplo com 4 dimensões

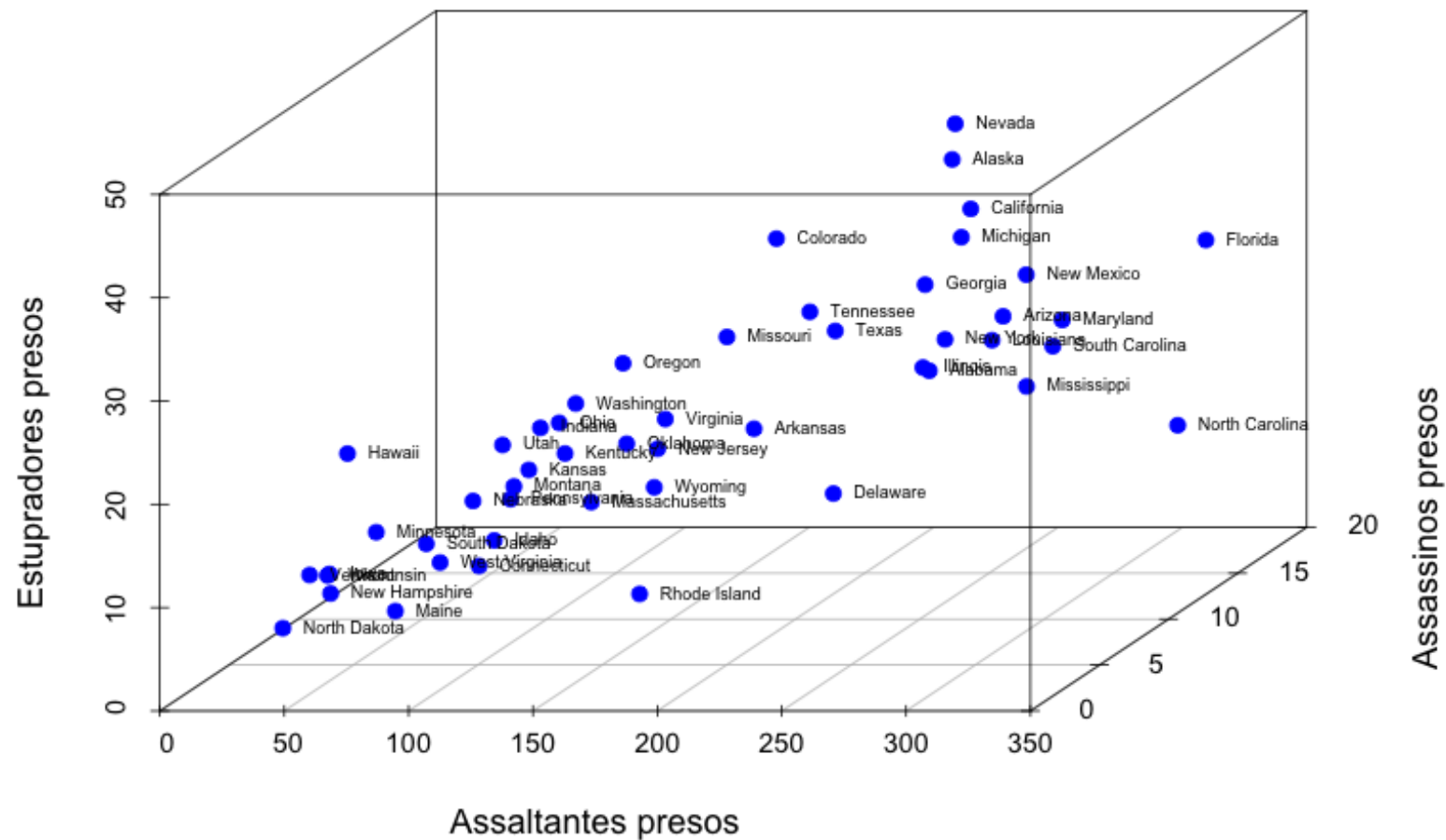
	Murder	Assault	UrbanPop	Rape
Alabama	13.20	236	58	21.20
Alaska	10.00	263	48	44.50
Arizona	8.10	294	80	31.00
Arkansas	8.80	190	50	19.50
California	9.00	276	91	40.60
Colorado	7.90	204	78	38.70
. . .	. . .	. . .	. . .	. . .

Dataset com informações sobre assassinatos presos (murder), de assaltantes presos (assault), estupradores presos (rape) e o percentual da população que é urbana (UrbanPop) de 50 estados americanos<sup>a</sup>.

---

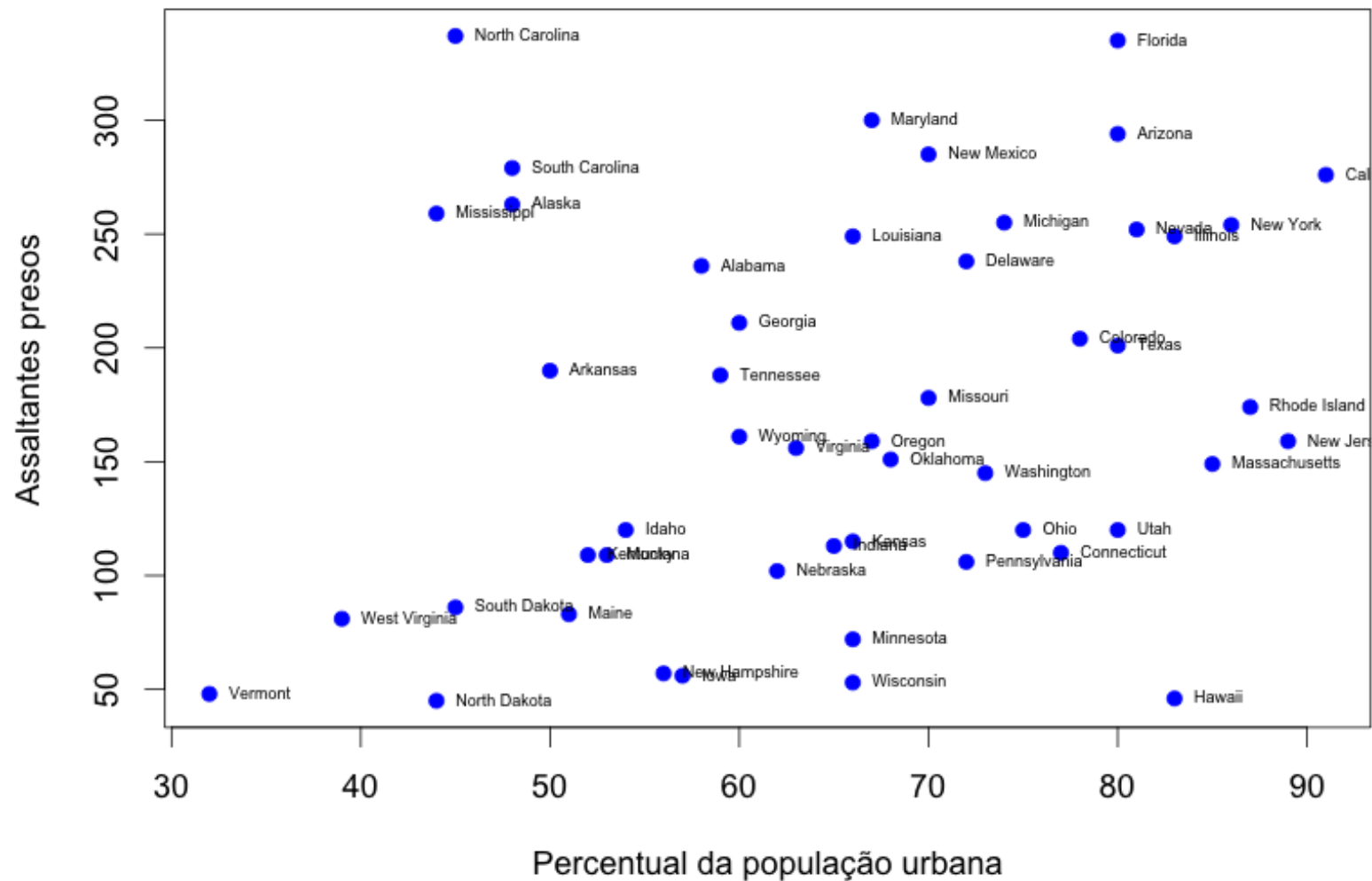
<sup>a</sup>Os números em murder, assault e rape são os números de ocorrências para cada 100.000 habitantes.

## Dados sobre 50 estados americanos em 1973



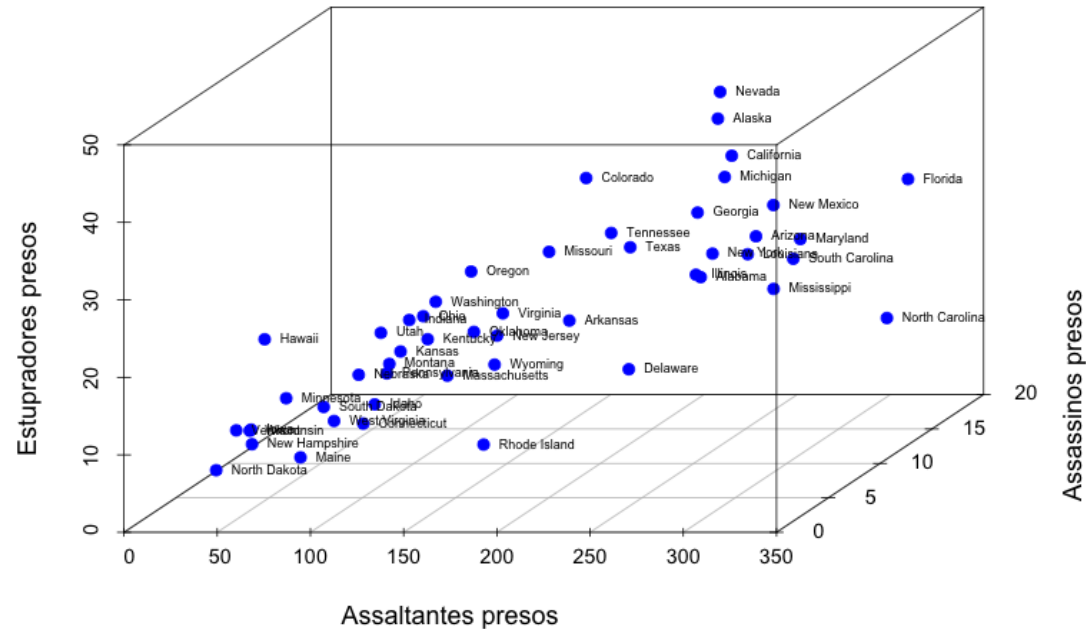


## Outra visão sobre o dataset



# Como projetar os pontos $X \in \mathbb{R}^3$ para $z \in \mathbb{R}^2$ ?

Dados sobre 50 estados americanos em 1973



Fazer o desenho da projeção.

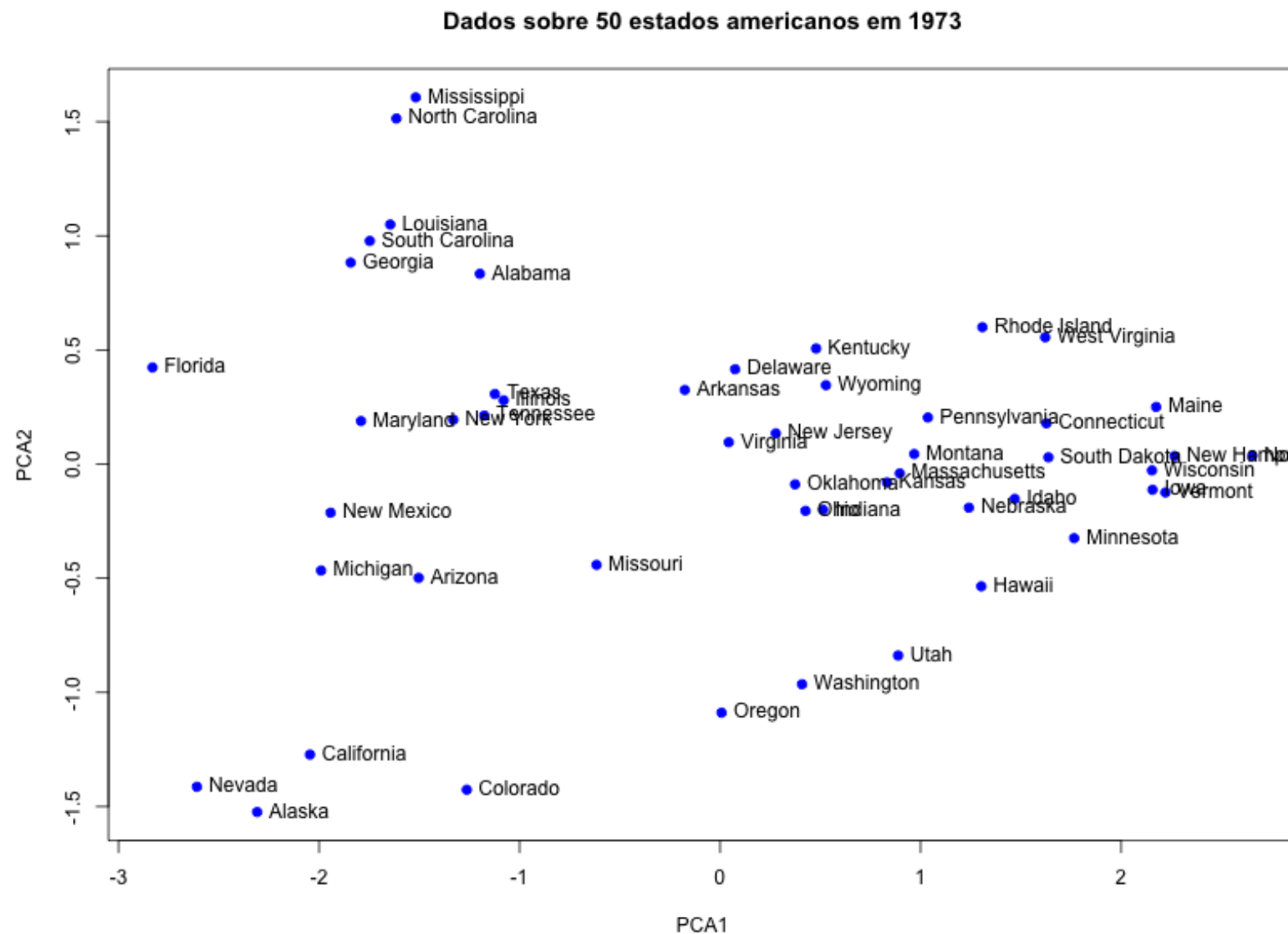
# Fazendo a projeção para $\mathbb{R}^2$

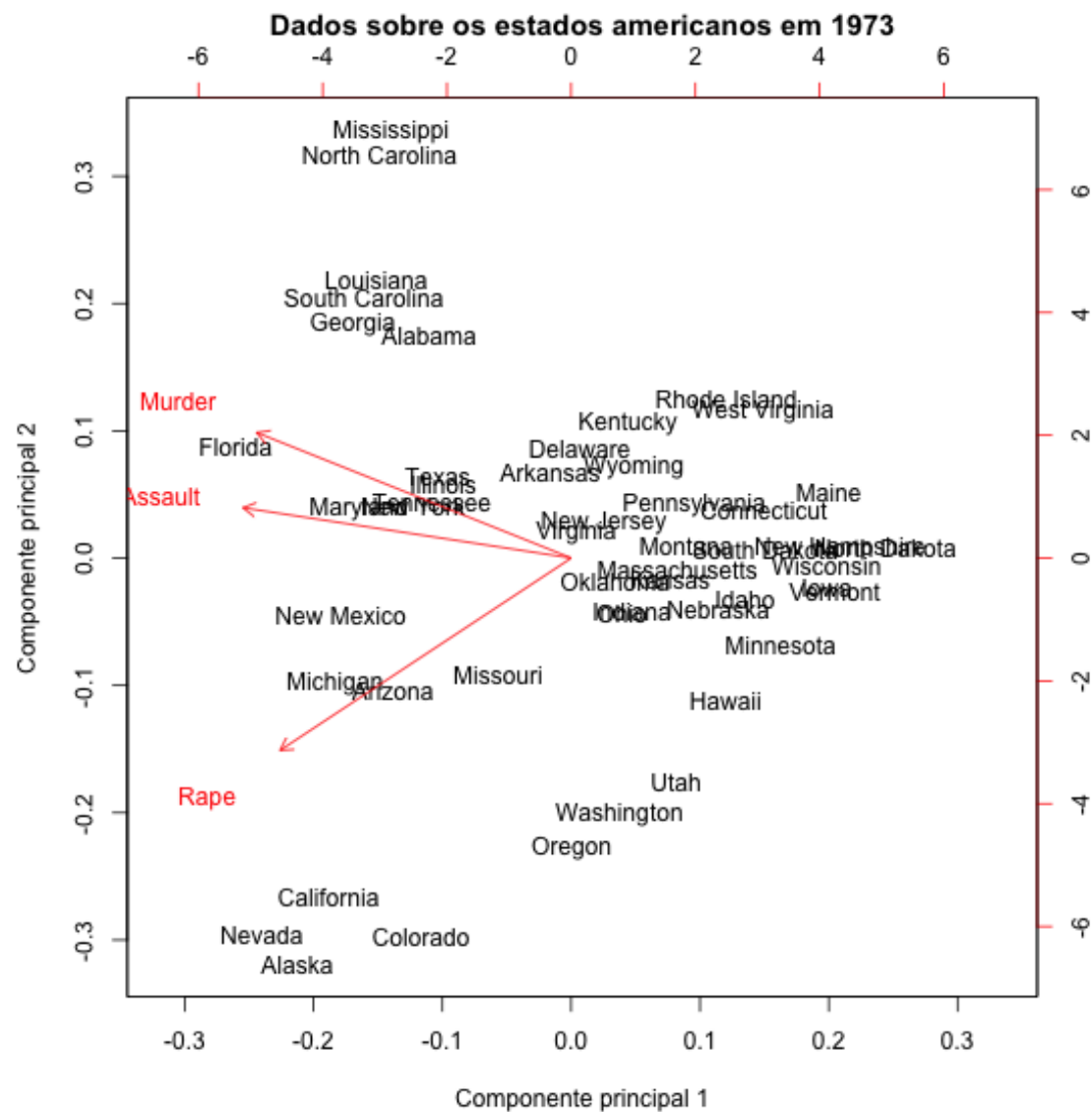
```
# Levando-se em consideracao apenas Murder, Assault e Rape  
pca <- prcomp(USArrests[,c(1,2,4)], scale. = TRUE)  
summary(pca)
```

	PC1	PC2	PC3
Standard deviation	1.5358	0.6768	0.4282
Proportion of Variance	0.7862	0.1527	0.0611
Cumulative Proportion	0.7862	0.9389	1.0000

```
head(pca$x[,1:2])
```

	PC1	PC2
Alabama	-1.20	0.83
Alaska	-2.31	-1.52
Arizona	-1.50	-0.50
Arkansas	-0.18	0.32
California	-2.05	-1.27
Colorado	-1.26	-1.43

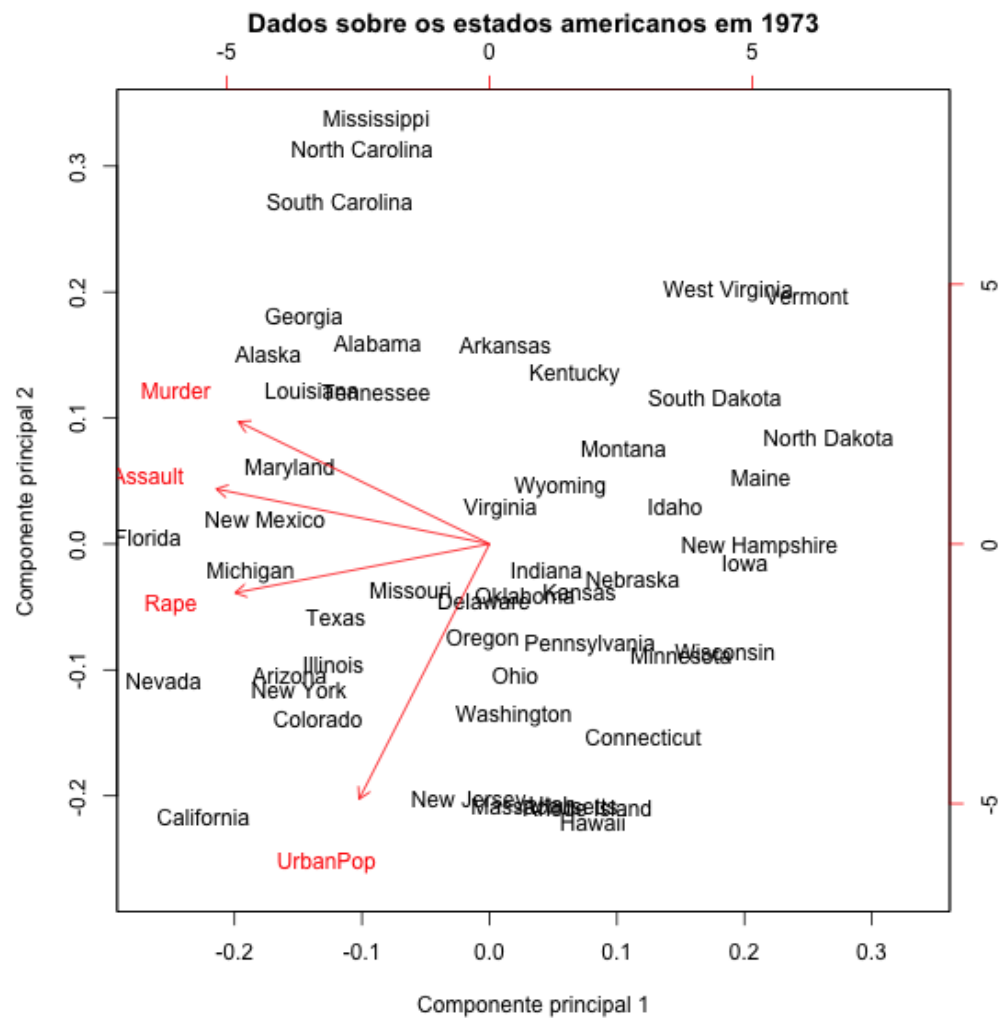




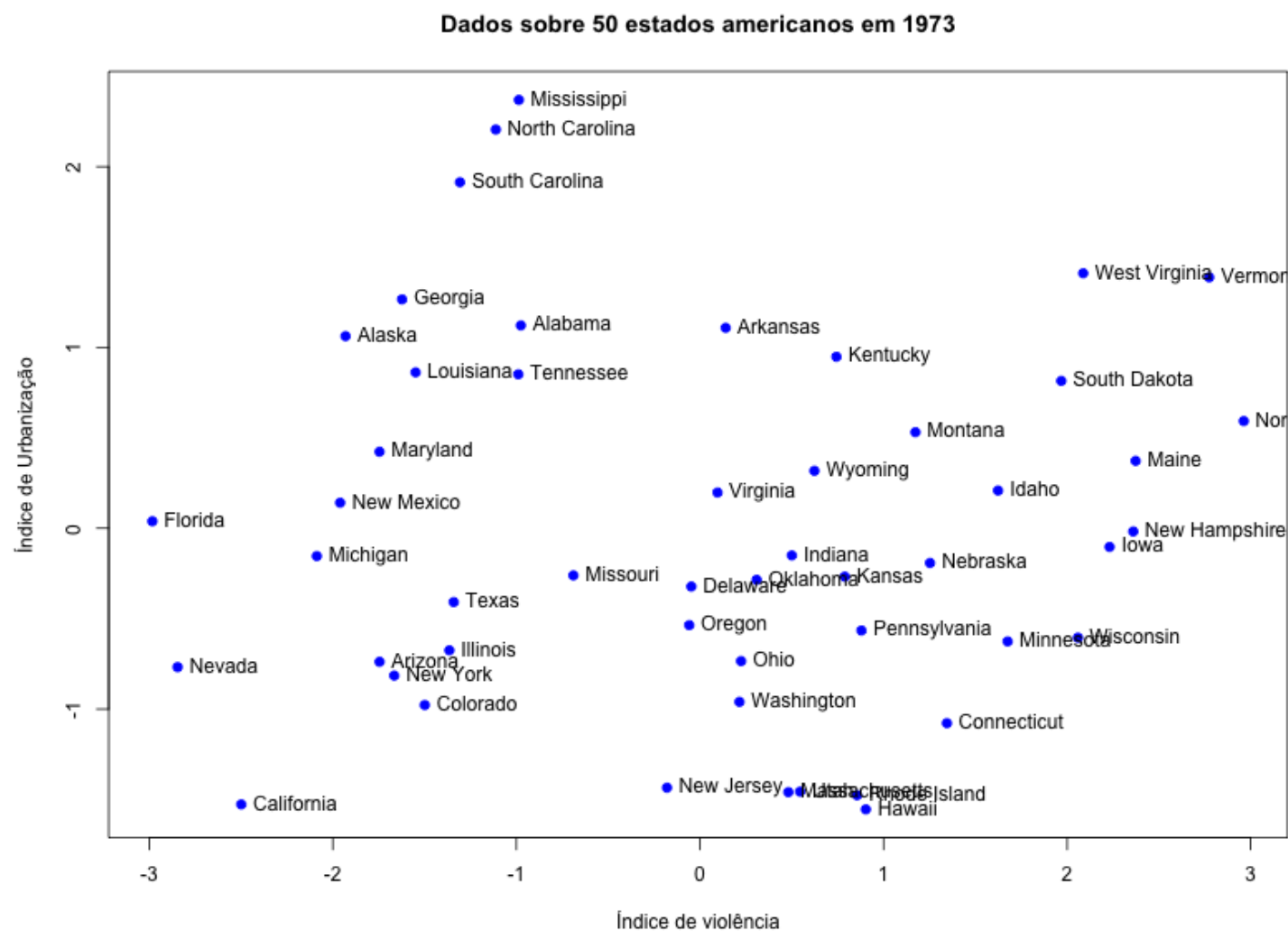
```
# Para todo o dataset
pca <- prcomp(USArrests, scale. = TRUE)
summary(pca)
pca$rotation
```

	PC1	PC2	PC3	PC4
Standard deviation	1.5749	0.9949	0.5971	0.4164
Proportion of Variance	0.6201	0.2474	0.0891	0.0434
Cumulative Proportion	0.6201	0.8675	0.9566	1.0000

	PC1	PC2	PC3	PC4
Murder	-0.54	0.42	-0.34	0.65
Assault	-0.58	0.19	-0.27	-0.74
UrbanPop	-0.28	-0.87	-0.38	0.13
Rape	-0.54	-0.17	0.82	0.09



Os atributos de murder, assault e rape estão correlacionadas, enquanto que a variável UrbanPop não está correlacionada com as outras.



No índice de urbanização, quanto mais em baixo do gráfico, mais urbano. Para o índice de violência, quanto mais a esquerda, mais violento



---

# Principal Component Analysis (PCA)

---

# Algoritmo PCA

- Reduz dados de n-dimensões para k-dimensões.
- Calcular a matriz de covariância:

$$Sigma = \frac{1}{m} \sum_{i=1}^n (X[i,])(X[i,])^T \quad (1)$$

- Calcular os autovetores<sup>a</sup> da matriz Sigma  
 $U = SVD(Sigma)$ , onde  $U \in \mathbb{R}^{n \times n}$ .
- Calcular os PCAs, na forma de  $z$ :

$$U_{reduce} = U[:, 1 : K] \quad (2)$$

$$z = (U_{reduce})^T \times X^T \quad (3)$$

---

<sup>a</sup>Singular Value Decomposition (SVD) - algebra linear

---

## Algoritmo PCA: pré-processamento dos dados

- Os dados precisam estar normalizados! Mesma situação do *k – means*.
- No *R*, a maioria das implementações de PCA já fornecem esta opção como parâmetro.
- No *prcomp*, basta informar que o parâmetro *scale.* é verdadeiro.

```
pca <- prcomp(dataset, scale. = TRUE)
```

---

## Escolhendo o número de componentes principais

```
> summary(pca)
```

```
Importance of components:
```

	PC1	PC2	PC3	PC4
Standard deviation	1.5749	0.9949	0.59713	0.41645
Proportion of Variance	0.6201	0.2474	0.08914	0.04336
Cumulative Proportion	0.6201	0.8675	0.95664	1.00000

---

# Alguns exemplos

Exemplos de uso de PCA

---

## Revisando a aplicabilidade

- Reduzir memória e disco necessário para armazenar os dados.
- Minimizar o tempo de processamento dos algoritmos de aprendizagem.
- Visualização de dados.

---

## Material de consulta

- Capítulo 10 do livro Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. An Introduction to Statistical Learning with Applications in R. Springer, 4th edition, 2014.