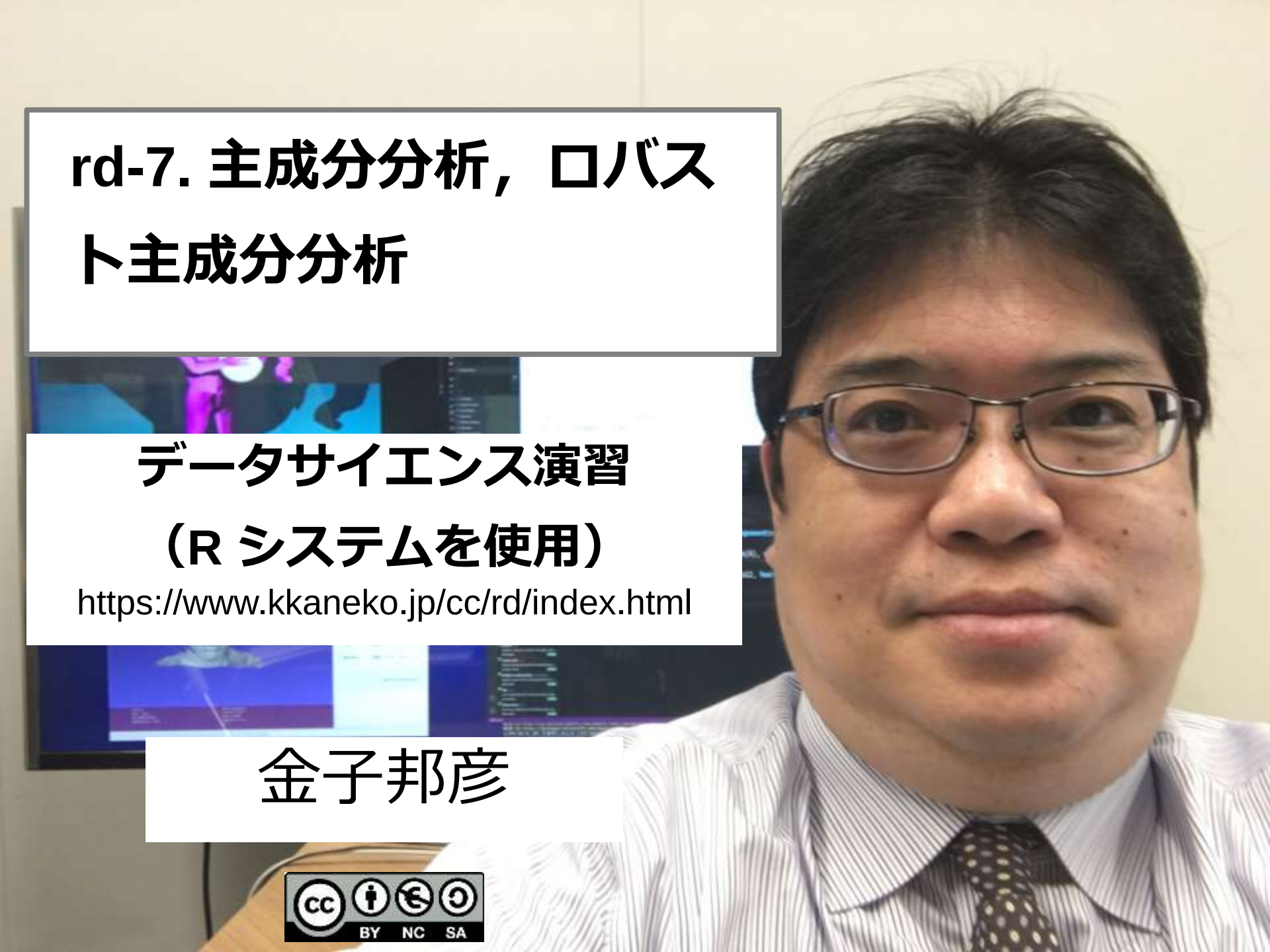
rd-7. 主成分分析，ロバスト主成分分析



データサイエンス演習

（R システムを使用）

https://www.kkaneko.jp/cc/rd/index.html

金子邦彦

アウトライン

7-1. 主成分分析と次元削減

7-2. パッケージの追加インストール

7-3. 主成分分析の実行

7-4. 外れ値と主成分分析

7-1 主成分分析と次元削減

次元と次元削減

属性 z を削除

x	y	z
0	-20	0
10	20	0.1

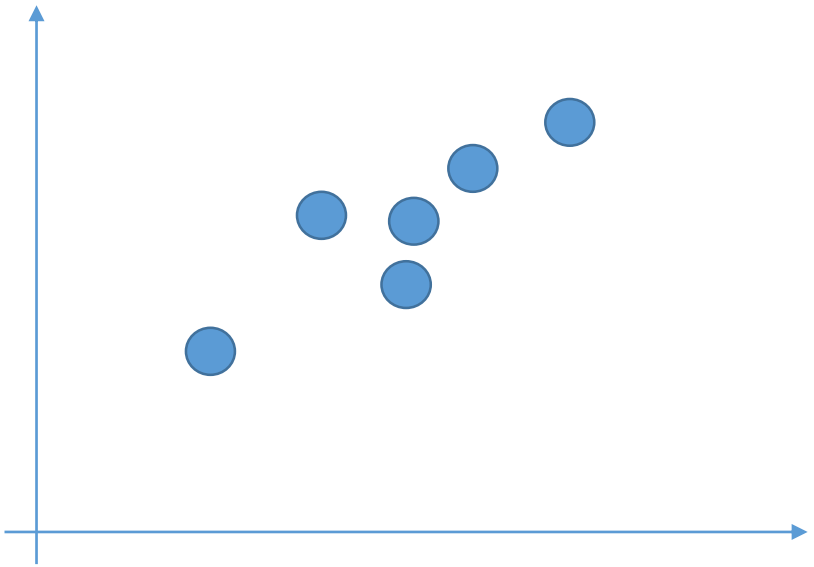
x	y
0	-20
10	20

元データ：　次元は**3**

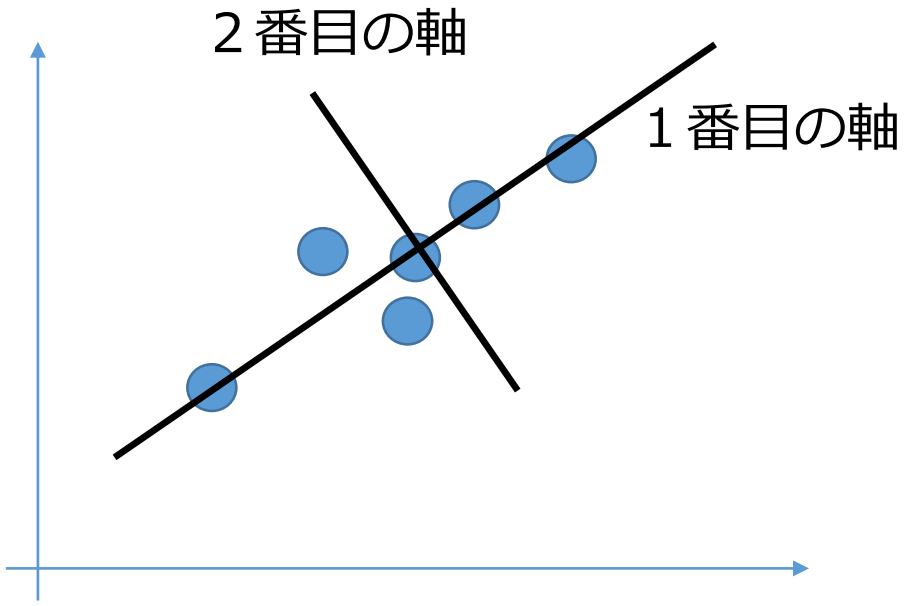
次元は**2**

次元数は，レコードの中の数値属性の数

データの中の主軸



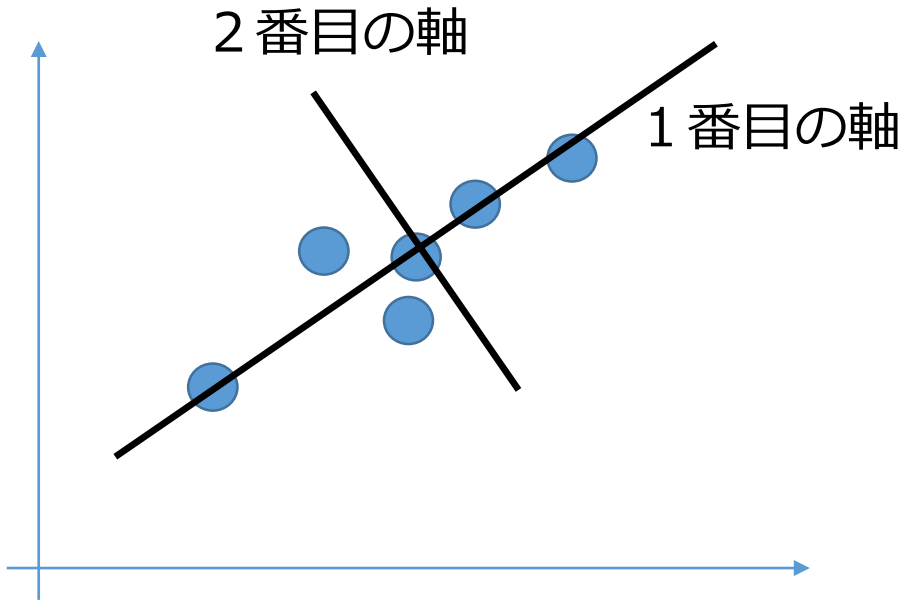
元データ：　次元は**2**



元データ：　次元は**2**

主成分分析 (principle component analysis)

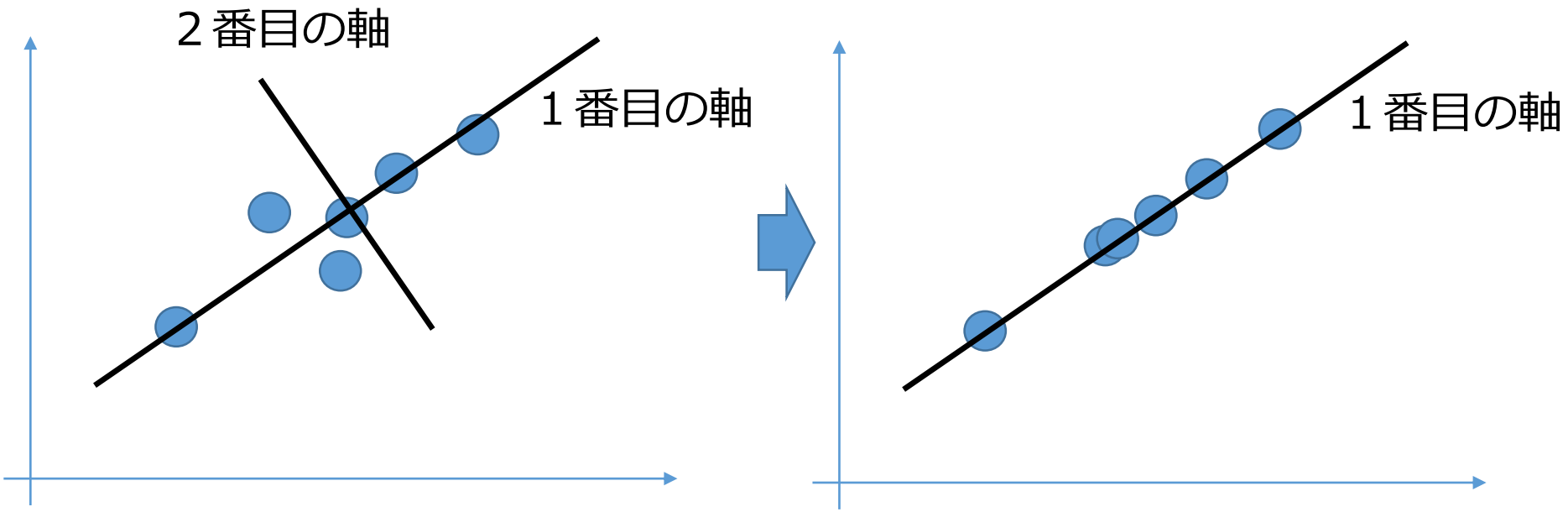
- 複数の変数から得られた標本をもとに，軸を得るための方式
- 得られた「１番目の軸（主軸）」は，標本群の**分散が最大になる**ような軸である.



元データ：　次元は**2**

主成分分析と次元削減

主成分分析で得られた上位の軸を残し、下位の軸を削除

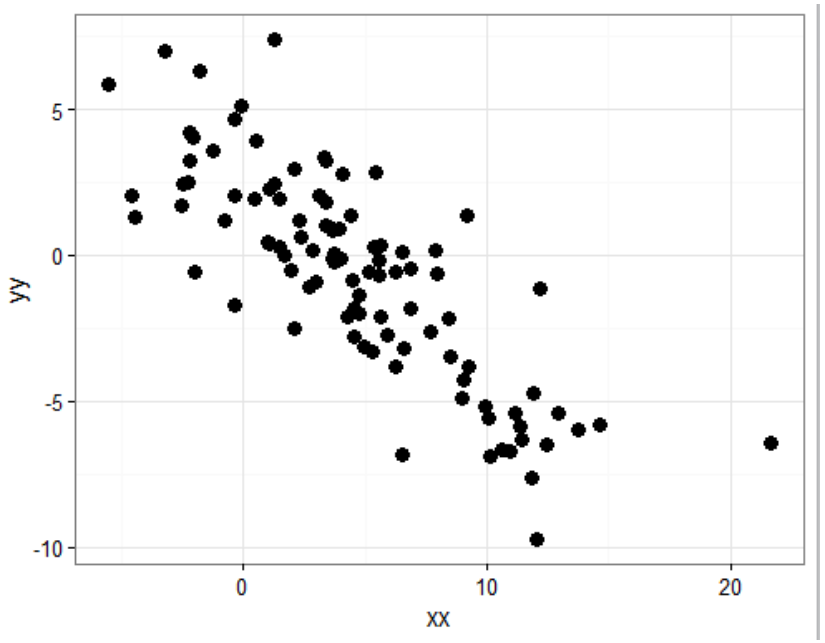


元データ：　次元は**2**

次元は**1**

主成分分析の例

- 元データ



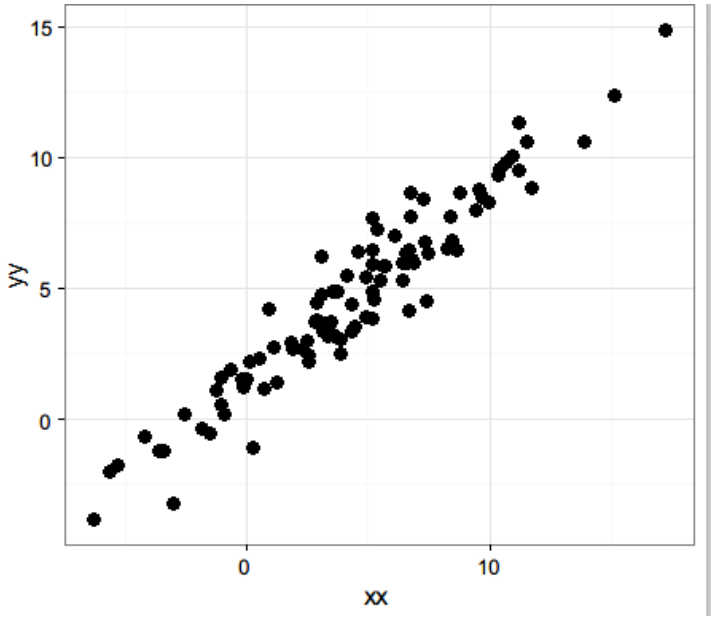

```
> print(afrotation)
            PC1       PC2
xx -0.8229231 0.5681528
yy  0.5681528 0.8229231
>
```

1番目の軸　２番目の軸

主成分分析の結果

主成分分析の例

- 元データ



	PC1	PC2
xx	-0.7954104	0.6060712
yy	-0.6060712	-0.7954104

1番目の軸　　２番目の軸

主成分分析の結果

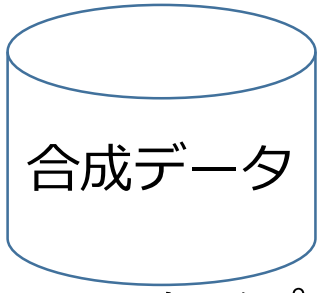
7-2 パッケージの追加インストール

パッケージの設定

- 次の手順で，必要なパッケージをインストール
- パッケージをインストールするのにインターネット接続が必要
- install.packages("**ggplot2**") を実行
- install.packages("**pcaPP**") を実行

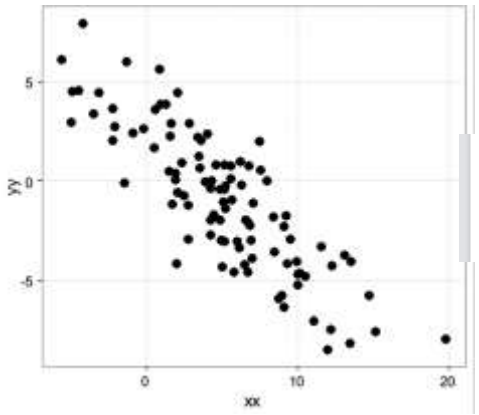
7-3 主成分分析の実行

合成データからランダムに100個選び，主成分分析を実施



サイズ１００
の標本を２セット

タイプ：数値（整数化しない）
サイズ：100,000




```
> print(a$rotation)
          PC1       PC2
xx -0.8229231 0.5681528
yy  0.5681528 0.8229231
>
```

主成分分析

```
x <- rnorm(100000, mean=5, sd=5)
y <- rnorm(100000, mean=5, sd=5)
n <- floor( runif(100, 1, 100000+1) )
d8 <- data.frame( xx=x[n],  yy=y[n] )
d8$yy <- d8$yy - (d8$xx + d8$yy) * 0.6
library(ggplot2)
ggplot(d8, aes(x=xx)) +
  geom_point( aes(y=yy), size=3 ) + theme_bw()
a <- prcomp(d8)
print(a$rotation)
```

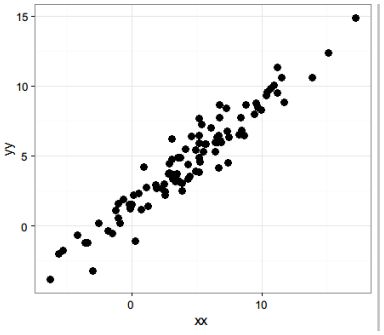
合成データの生成

合成データに
相関関係をもたせる

この２行が主成分分析

合成データからランダムに100個選び標本を作る

サイズ１００
の標本を２セッ




```
          PC1        PC2
xx -0.7954104  0.6060712
yy -0.6060712 -0.7954104
>
```

主成分分析

タイプ：数値（整数化しない）
サイズ：100,000

```
x <- rnorm(100000, mean=5, sd=5)
y <- rnorm(100000, mean=5, sd=5)
n <- floor( runif(100, 1, 100000+1) )
d9 <- data.frame( xx=x[n],  yy=y[n] )
d9$yy <- d9$yy + (d9$xx - d9$yy) * 0.8
library(ggplot2)
ggplot(d9, aes(x=xx)) +
  geom_point( aes(y=yy), size=3 ) + theme_bw()
a <- prcomp(d9)
print(a$rotation)
```

合成データの生成

合成データに
相関関係をもたせる

この２行が主成分分析

主成分分析 (principle component analysis)

- 次元数が n のとき，n 個の軸が得られる

- **1番目の軸（主軸）**は，**分散が最大になるような軸**.
- 2番目の軸は，1番目の軸方向の成分を取り除いた残りで，分散が最大になる軸
- 3番目の軸は，1, 2 番目の軸方向の成分を取り除いた残りで，分散が最大になる軸

  以上を n 個の軸を得るまで繰り返す
- 得られた軸は，互いに直交（**垂直に交わる**）

7-4 外れ値と主成分分析

主成分分析は万能というわけではない

◆ 標本には，必ず「ノイズ」が混入する

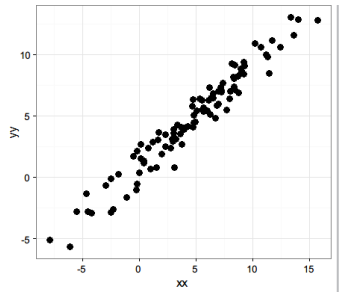
◆ ノイズがランダム（無作為）のときは，求まる軸の向きに影響を及ぼさない

◆ **ノイズがランダムでない**ときは，求まる軸の**向き**に**影響を及ぼす**

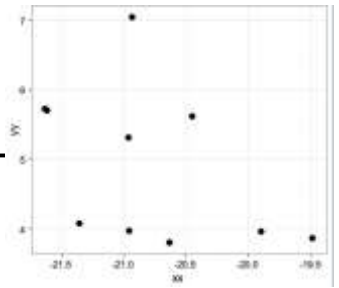
- **外れ値**： 明らかにおかしな値
- **計測もれ**： 値が 0 や 空 になっている

- 手作業で「外れ値や計測もれなどの不正なデータを取り除く」のが基本だが，自動で取り除くのが困難な状況もある

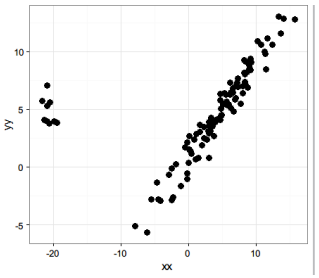
外れ値を含むデータの合成の例



d9 ＋



d10
外れ値

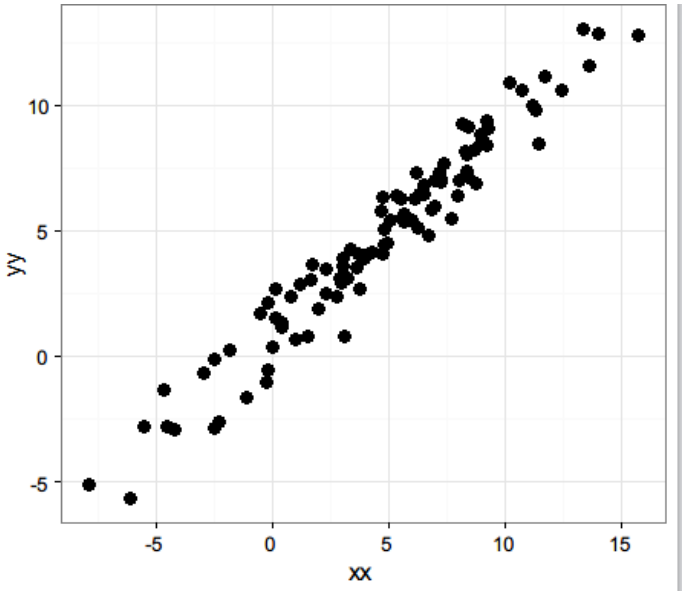


d11
外れ値が混入

```
x <- rnorm(100000, mean=5, sd=5)
y <- rnorm(100000, mean=5, sd=5)
n <- floor( runif(100, 1, 100000+1) )
d9 <- data.frame( xx=x[n],  yy=y[n] )
d9$yy <- d9$yy + (d9$xx - d9$yy) * 0.8
d10 <- data.frame( xx=rnorm(10, mean=-20, sd=1),
  yy=rnorm(10, mean=5, sd = 1) )
d11 <- rbind( d9, d10 )
library(ggplot2)
ggplot(d11, aes(x=xx)) +
  geom_point( aes(y=yy), size=3 ) + theme_bw()
```

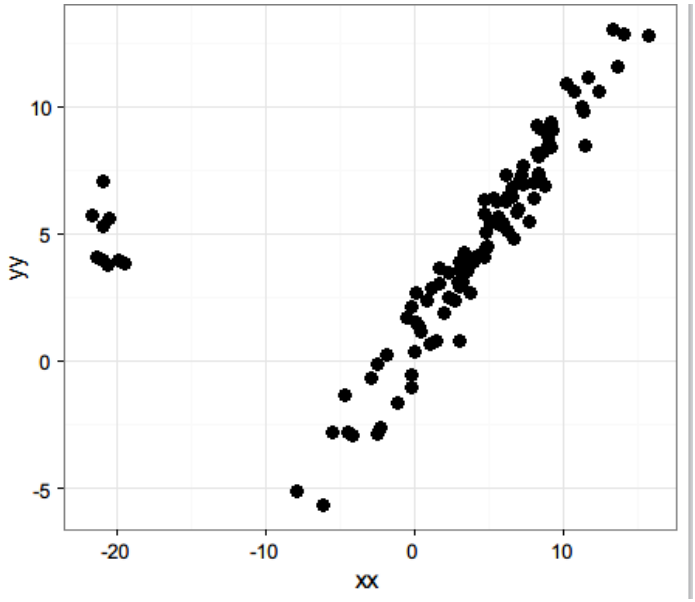
外れ値の混入

主成分分析は外れ値に弱い



d9

```
> a <- prcomp(d9)
> print(a$rotation)
           PC1        PC2
xx -0.7638487 -0.6453954
yy -0.6453954  0.7638487
>
```

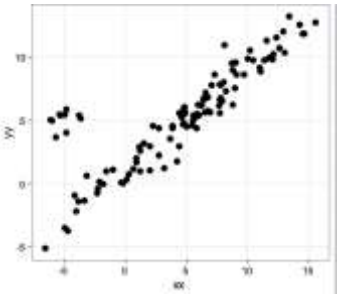


d11
外れ値が混入

```
> a <- prcomp(d11)
> print(a$rotation)
           PC1        PC2
xx -0.9681328 -0.2504373
yy -0.2504373  0.9681328
>
```

全データから
忠実に軸を
算出

主成分分析は外れ値に弱い



d11

```
> a <- prcomp(d11)
> print(a$rotation)
            PC1        PC2
xx -0.9681328 -0.2504373
yy -0.2504373  0.9681328
>
```

主成分分析

```
x <- rnorm(100000, mean=5, sd=5)
y <- rnorm(100000, mean=5, sd=5)
n <- floor( runif(100, 1, 100000+1) )
d9 <- data.frame( xx=x[n],  yy=y[n] )
d9$yy <- d9$yy + (d9$xx - d9$yy) * 0.8
d10 <- data.frame( xx=rnorm(10, mean=-20, sd=1),
  yy=rnorm(10, mean=5, sd = 1) )
d11 <- rbind( d9, d10 )
library(ggplot2)
ggplot(d11, aes(x=xx)) +
  geom_point( aes(y=yy), size=3 ) + theme_bw()
a <- prcomp(d11)
print(a$rotation)
```

外れ値の混入

主成分分析のバリエーション
- robust PCA -

- 外れ値があるデータでも，主成分分析したい

　　→　この問題に取り組んだ手法が多数ある

例えば

C. Croux, P. Filzmoser, M. Oliveira, (2007). Algorithms for Projection-Pursuit Robust Principal Component Analysis, Chemometrics and Intelligent Laboratory Systems, Vol. 87, pp. 218-225.

主成分分析のバリエーション
- Principle Component Pursuit -

- 計測漏れがあるデータでも，主成分分析したい

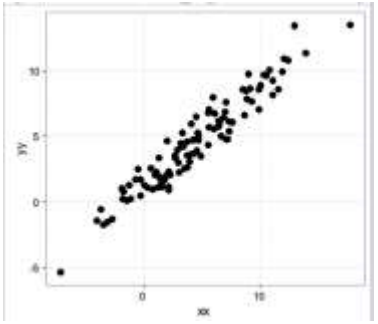
　　→　この問題に取り組んだ手法が多数ある

例えば

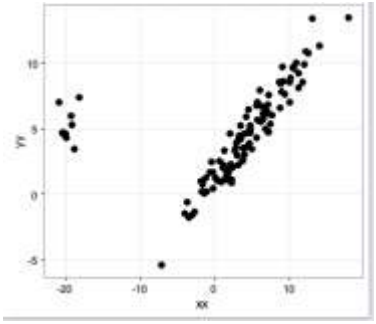
Candes, E. J., Li, X., Ma, Y., & Wright, J. (2011). Robust principal component analysis?. Journal of the ACM (JACM), 58(3), 11

robust PCA の実行結果例

pcaPP パッケージを使用



d9



d11
外れ値が混入

- PCA

```
> a <- prcomp(d9)
> print(a$rotation)
           PC1        PC2
xx -0.7937832 -0.6082008
yy -0.6082008  0.7937832
>
```

```
> a <- prcomp(d11)
> print(a$rotation)
           PC1        PC2
xx -0.9776248 -0.2103561
yy -0.2103561  0.9776248
> a <- prcomp(d9)
```

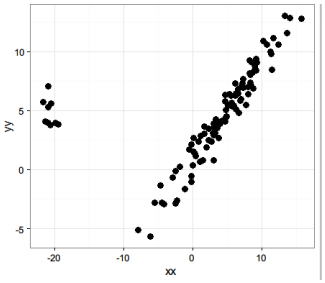
- Robust PCA

```
> a2 <- PCAgrid(d9)
> print(a2$loadings)

Loadings:
   Comp.1 Comp.2
xx  0.725 -0.688
yy  0.688  0.725
```

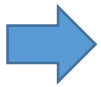
```
> a2 <- PCAgrid(d11)
> print(a2$loadings)

Loadings:
   Comp.1 Comp.2
xx  0.805 -0.594
yy  0.594  0.805
```

外れ値に対して
ある程度の
耐性がある

pcaPP パッケージを用いて robust PCA



d11



```
> library(pcaPP)
> a2 <- PCAgrid(d11)
> print(a2$loadings)

Loadings:
   Comp.1 Comp.2
xx  0.849 -0.529
yy  0.529  0.849
```

```
x <- rnorm(100000, mean=5, sd=5)
y <- rnorm(100000, mean=5, sd=5)
n <- floor( runif(100, 1, 100000+1) )
d9 <- data.frame( xx=x[n],  yy=y[n] )
d9$yy <- d9$yy + (d9$xx - d9$yy) * 0.8
d10 <- data.frame( xx=rnorm(10, mean=-20, sd=1),
  yy=rnorm(10, mean=5, sd = 1) )
d11 <- rbind( d9, d10 )
library(ggplot2)
ggplot(d11, aes(x=xx)) +
  geom_point( aes(y=yy), size=3 ) + theme_bw()
library(pcaPP)
a2 <- PCAgrid(d11)
print(a2$loadings)
```

外れ値の混入