

## 2 変数の記述統計

---

1. 名義・順序変数の記述統計
2. 二変量連続変数の記述統計
3. 実用的なパッケージ、関数の紹介

## 名義・順序変数の記述統計

---

## (復習) 分割表：複数の名義・順序変数の関係

### 二つの名義・順序変数の分割表

変数 A	変数 B			合計
	水準 $B_1$	...	水準 $B_j$	
水準 $A_1$	$A_1$ & $B_1$ の 頻度 (%)	...	$A_1$ & $B_j$ の 頻度 (%)	$A_1$ の 頻度 (%)
⋮	⋮	同時分布	⋮	⋮
⋮	⋮		⋮	⋮
⋮	⋮		⋮	⋮
水準 $A_K$	$A_K$ & $B_1$ の 頻度 (%)	...	$A_K$ & $B_j$ の 頻度 (%)	$A_K$ の 頻度 (%)
合計	$B_1$ の 頻度 (%)	...	$B_j$ の 頻度 (%)	全体の 合計

周辺分布 (marginal distribution)

(注：同時分布の割合 (%) の分母は三通り。どれを報告するかはケースバイケースで判断)

21

## (復習)2×2 分割表とリスク差・リスク比・オッズ比

2 値のアウトカム変数 (例: 生存/死亡) と 2 値のカテゴリ変数 (例: 治療あり/なし) の関係が以下の様な 2×2 分割表で表されとする

頻度の表

	死亡	生存	合計
治療なし	$a$	$b$	$a + b$
治療あり	$c$	$d$	$c + d$
合計	$a + c$	$b + d$	$n$

確率の表

	死亡	生存	合計
治療なし	$p_0$	$1 - p_0$	1
治療あり	$p_1$	$1 - p_1$	1

- ・ リスク差 (Risk Difference, RD) :  $RD = p_1 - p_0$
- ・ リスク比 (Risk Ratio, RR) :  $RR = \frac{p_1}{p_0}$
- ・ オッズ比 (Odds Ratio, OR) :  $OR = \frac{p_1/(1 - p_1)}{p_0/(1 - p_0)}$
- ・ Number Needed to Treat (NNT) :  $NNT = \frac{1}{(p_0 - p_1)}$

## (復習) 二変数のクロス集計と割合

### 二変数のクロス集計

- ・ `table()` の引数に 2 つのベクトルを指定
- ・ `prop.table` の `margin` 引数を指定  
行方向・列方向の割合を計算

```
tab2 <- table("ベクトル名1", "ベクトル名2")  
print(tab2)  
prop.table(tab2) # 全体の割合  
prop.table(tab2, margin=1) # 行ごとの割合  
prop.table(tab2, margin=2) # 列ごとの割合
```

## 演習問題：pharmacoSmoking データの解析

**演習:** asaur パッケージの pharmacoSmoking の RCT データを用いて、以下の解析を行ってみましょう

- ・ relapse = 1（禁煙失敗）をイベントとして扱う
- ・ 介入群（grp = "combination"）と対照群（grp = "patchOnly"）におけるイベント発生割合  $p_1, p_0$  を計算
- ・ 以下の指標を計算：
  - ・ リスク差 (Risk Difference, RD)
  - ・ リスク比 (Risk Ratio, RR)
  - ・ オッズ比 (Odds Ratio, OR)
  - ・ Number Needed to Treat (NNT)

## コード例：禁煙失敗に対する各指標の算出

```
install.packages("asaur")  
library(asaur)  
data(pharmacoSmoking)
```

データの読み込み

```
tb <- table(pharmacoSmoking$grp,  
            pharmacoSmoking$relapse)  
ptb <- prop.table(tb, margin=1)
```

分割表の作成

```
p0 <- ptb[2,2]  
p1 <- ptb[1,2]
```

介入群と対照群のイベント発生割合の計算

```
RD <- p1 - p0  
RR <- p1 / p0  
OR <- (p1 / (1 - p1)) / (p0 / (1 - p0))  
NNT <- 1 / (p0 - p1)
```

統計量の計算

```
print(c(RD=RD,RR=RR,OR=OR,NNT=NNT))
```



- ・ 出力例 :

RD	RR	OR	NNT
-0.2059426	0.7465322	0.3557692	4.855721

- ・ 解釈例 :

- ・ リスク差 (RD) : 約  $-0.206$   $\rightarrow$  combination 群の禁煙失敗リスクは patchOnly 群より 20.6 ポイント低い
- ・ リスク比 (RR) : 約  $0.747$   $\rightarrow$  combination 群の禁煙失敗リスクは patchOnly 群の約 75%
- ・ オッズ比 (OR) : 約  $0.356$   $\rightarrow$  combination 群の禁煙失敗オッズは patchOnly 群の約 36%
- ・ NNT: 約  $5$   $\rightarrow$  combination 治療を 5 人に行うことで、平均して 1 人の禁煙失敗を防げる

## 二変量連続変数の記述統計

---

# (復習) 散布図と相関

## 散布図

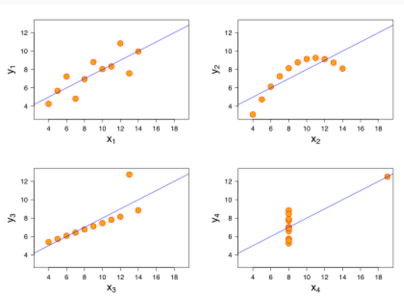
- ・ 二変数  $A$  と  $B$ . 片方を横軸、他方を縦軸.  $(A, B)$  のペアをプロットした図

## ピアソンの相関係数

- ・ 二つの変数の間の **直線的な関係** の度合いを表す
- ・ 外れ値の影響を受けやすい

## スピアマンの順位相関係数

- ・ 二つの変数の間の **単調性** の度合いを表す
- ・ **順位データに対するピアソンの相関係数と一致**
- ・ 外れ値に対して頑健



Anscombe (1973)

ピアソン相関係数がほぼ同じ 4 つの散布図

## (復習) 散布図の描画

plot 関数による散布図の作成

```
plot(x = "ベクトル名1", y = "ベクトル名2",  
     main = "タイトル", xlab = "x軸ラベル", ylab="y軸ラベル",  
     ...)
```

par() 関数とマルチパネル表示

- ・ mfrow = c(nrow, ncol) : 描画順が行優先のパネル分割
- ・ mfcoll = c(nrow, ncol) : 描画順が列優先のパネル分割

使い方:

```
par(mfrow = c(1,2)) # 2列横並びに分割  
plot(x1,y1,...) # 左パネル: 対照群  
plot(x2,y2,...) # 右パネル: CBT 群  
par(mfrow = c(1,1)) # リセット
```

# 相関係数の算出

- ・ ベース機能を用いた相関係数の算出
  - ・ 関数 `cor()` を用いて算出

```
cor(x = "ベクトル名1", y = "ベクトル名2",  
    method="相関係数の種類の指定")
```

- ・ `method =` は "pearson", "kendall", "spearman" が指定可能
- ・ パッケージ `dplyr` を用いた相関係数の算出
  - ・ 他の統計量と同様に関数 `summarise` で算出可能
  - ・ ピアソンの相関係数とスピアマンの相関係数の両方を層別に算出する例

```
data %>%  
  group_by(層別因子の列名) %>%  
  summarise(  
    表示名1 = cor(列名1, 列名2, method = "pearson"),  
    表示名2 = cor(列名1, 列名2, method = "spearman")  
  )
```

## 異なる相関を持つデータの生成

**演習:** 以下の手順で異なるパターンの関連を持つ 4 つの組  $(X_1, Y_1), (X_1, Y_2), (X_1, Y_3), (X_1, Y_4)$  をそれぞれ 200 個作成してみましょう

$$\begin{aligned} X_1, X_2 &\overset{i.i.d.}{\sim} N(0, 1) \\ Y_1 &= X_1 + X_2 \\ Y_2 &= -X_1 + X_2 \\ Y_3 &= X_2 \\ Y_4 &= X_1^2 + X_2 \end{aligned} \tag{1}$$

## 乱数生成: コード例

```
set.seed(123)
```

```
n <- 200
```

```
x1 <- rnorm(n)
```

```
x2 <- rnorm(n)
```

```
y1 <- x1 + x2
```

 →  $x_1$  が大きければ  $y_1$  も大きい: 正の相関

```
y2 <- -x1 + x2
```

 →  $x_1$  が大きければ  $y_2$  は小さい: 負の相関

```
y3 <- x2
```

 →  $x_1$  と  $y_3$  は独立

```
y4 <- x1^2 + x2
```

 →  $x_1$  と  $y_4$  は非線形の関係

# 散布図の描画と相関係数の算出

**演習:** 以下の関係があることを散布図と相関係数から確認してみましょう

- ・  $(X_1, Y_1)$ : 正の相関
- ・  $(X_1, Y_2)$ : 負の相関
- ・  $(X_1, Y_3)$ : 独立
- ・  $(X_1, Y_4)$ : 非線形な関係

## 1. 散布図の描画

- ・ `par(mfrow=c(2,2))` を用いて 2 行 2 列の複数パネル表示にすることで一枚の画像で散布図を比較可能にしてみましょう

## 2. 相関係数の算出

- ・ それぞれの組み合わせについて、ピアソンの相関係数とスピアマンの順位相関係数の双方を算出してみましょう (8 パターンの算出)



## 散布図：コード例

# レイアウト設定

```
par(mfrow = c(2, 2))
```

 → 2 行 2 列のレイアウト設定

```
plot(x1, y1, main = " 正の相関")  
plot(x1, y2, main = " 負の相関")  
plot(x1, y3, main = " 独立")  
plot(x1, y4, main = " 非線形関係")
```

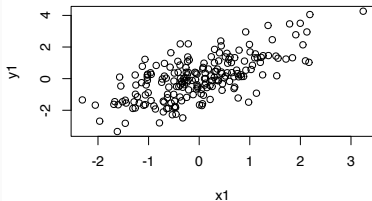
→ 4 パターンの plot

```
par(mfrow = c(1, 1))
```

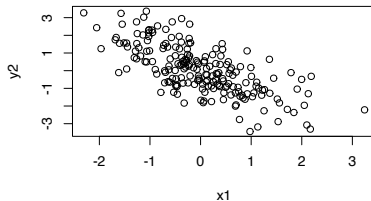
 → レイアウト設定を元に戻す

# 散布図：出力例

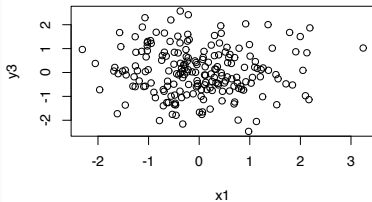
正の相関



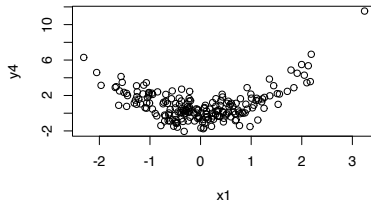
負の相関



独立



非線形な関係



## 相関係数の計算：コード例

```
types <- c("positive","negative","independent",  
          "nonlinear")  
type_factor <- factor(rep(types, each=n),  
                      levels = types)  
sim_data <- data.frame(x = c(x1,x1,x1,x1),  
                      y = c(y1,y2,y3,y4),  
                      type = type_factor)
```

long format のデータフレームの作成

```
library(dplyr)  
sim_data %>%  
  group_by(type) %>%  
  summarise(  
    Pearson = cor(x, y, method = "pearson"),  
    Spearman = cor(x, y, method = "spearman"))
```

→ 相関係数の種類の指定

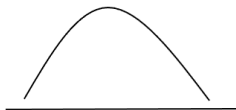
## 相関係数の出力

```
# A tibble: 4 × 3
  type          Pearson Spearman
<fct>         <dbl>     <dbl>
1 positive      0.677      0.628
2 negative    -0.698     -0.693
3 independent -0.0277    -0.0459
4 nonlinear     0.155    -0.0379
```

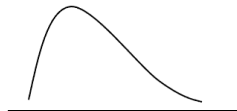
相関係数は線形的な関係の強さと向きを表す指標なので、  
非線形な関係に対して意味を持たない

## 対称な分布、非対称な（歪んだ）分布

対称な分布  
symmetric



歪んだ分布  
多くは、正（右側）に歪み  
positively/right skewed



# 1 変数の分布の確認

hist によるデータのヒストグラム作成

## ヒストグラム描画のサンプルコード

```
faithful_data <- faithful$waiting → faithfulデータのwaiting列(2列目を取り出す)  
par(cex.main = 3, cex.lab = 2, cex.axis = 2) → ヒストグラムの各種文字サイズ変更  
hist(faithful_data, breaks=20, prob=FALSE, main='Old Faithfulデータのヒストグラム', xlab='待ち時間')
```

→ ヒストグラム描画. bin数は20, 縦軸は頻度

# 正規 Q-Q プロットによる歪みの確認

**Q-Q プロット (Quantile-Quantile plot)**: 2 つの分布の分位点同士を対応させて比較する図

- ・ 任意の 2 つの分布の「形状の違い」を視覚的に評価するために用いられる

**正規 Q-Q プロット**: 観測データの分位点と標準正規分布の理論分位点を比較

- ・ Q-Q プロットの片方を標準正規分布の理論分位点としたもの
- ・ 観測値が正規分布に従うなら、Q-Q プロット上でおおむね直線上に並ぶ
- ・ 外れやすい箇所（端の方）に注目することで、歪みや外れ値を評価可能

R での使い方:

```
qqnorm(確認したい数値ベクトル) # 正規Q-Qプロットの描画  
qqline(確認したい数値ベクトル,col=2) # 赤で直線を追加
```

## 実データでの相関：歪んだ分布の例

PimaIndiansDiabetes2 の glucose と insulin について分布を確認

```
library(mlbench)
data(PimaIndiansDiabetes2)
data <- na.omit(PimaIndiansDiabetes2)
```

→ 欠損を除外した PimaIndiansDiabetes データの準備

```
hist(data$glucose, main="Histogram of Glucose",
      xlab="Glucose", breaks=20)
qqnorm(data$glucose)
qqline(data$glucose, col="red")
```

→ glucose の分布をヒストグラムで確認、正規分布からの歪みを qqnorm, qqline で確認

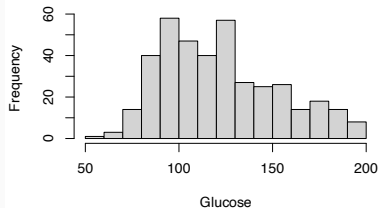
```
hist(data$insulin, main="Histogram of Insulin",
      xlab="Insulin", breaks=20)
qqnorm(data$insulin); qqline(data$insulin, col="red")
```

→ insulin についても同様の確認

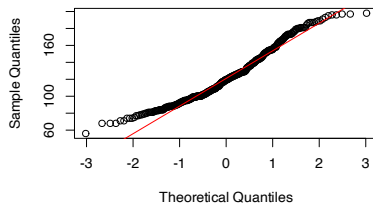


# ヒストグラムと正規 Q-Q プロット

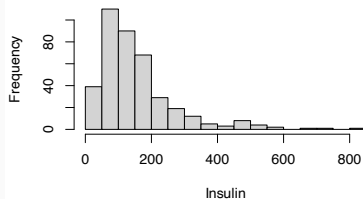
Histogram of Glucose



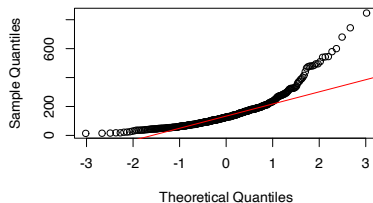
Normal Q-Q Plot



Histogram of Insulin



Normal Q-Q Plot



## 歪んだ分布の散布図と相関係数

**演習:** PimaIndiansDiabetes2 の glucose と insulin の散布図の描画と相関係数の算出を行い、歪んだ分布の散布図と相関係数を確認してみましょう

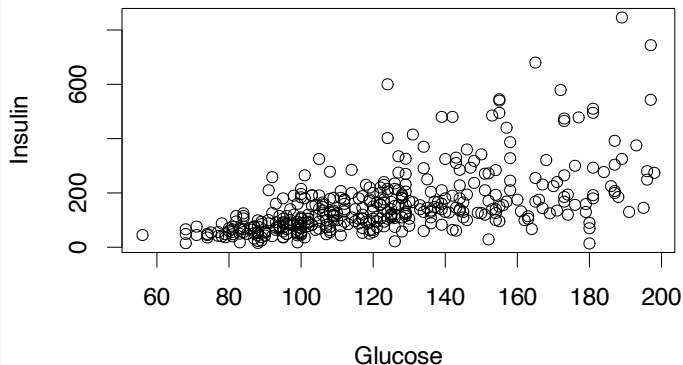
```
plot(data$glucose, data$insulin,  
      xlab = "Glucose", ylab = "Insulin")
```

→ 2変数の散布図の描画

```
data %>%  
  summarise(  
    Pearson=cor(glucose, insulin, method="pearson"),  
    Spearman=cor(glucose, insulin, method="spearman")  
  )
```

→ データから2種類の相関係数を算出

# 歪んだ分布の散布図と相関係数



相関係数

	Pearson	Spearman
1	0.581223	0.6589582

# 対数変換後の分布：ヒストグラムと正規 Q-Q プロット

対数変換後の glucose, insulin の分布を確認

```
data$log_glucose <- log(data$glucose)
data$log_insulin <- log(data$insulin)
```

→ 対数変換

```
hist(data$log_glucose, breaks=20,
      main="Histogram of log(Glucose)",
      xlab="log(Glucose)")
qqnorm(data$log_glucose)
qqline(data$log_glucose, col="red")
```

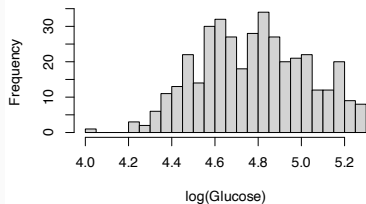
→ glucose の対数変換後の分布をヒストグラムで確認、正規分布からの歪みを qqnorm, qqline で確認

```
hist(data$log_insulin, breaks=20,
      main="Histogram of log(Insulin)",
      xlab="log(Insulin)")
qqnorm(data$log_insulin)
qqline(data$log_insulin, col="red")
```

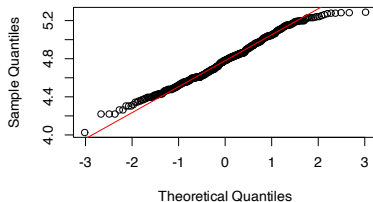
→ insulin の  
対数変換に  
についても同  
様の確認

# 対数変換後の分布：ヒストグラムと正規 Q-Q プロット

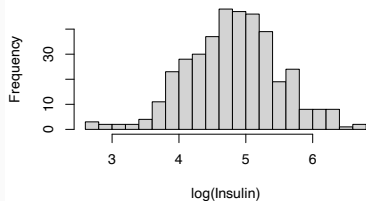
Histogram of log(Glucose)



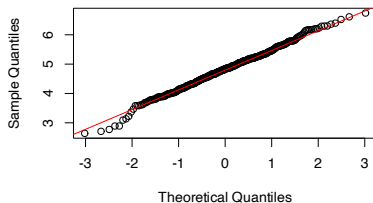
Normal Q-Q Plot



Histogram of log(Insulin)



Normal Q-Q Plot



## 対数変換後の散布図と相関係数

**演習:** 対数変換後の glucose と insulin の散布図の描画と相関係数の算出を行きましょう

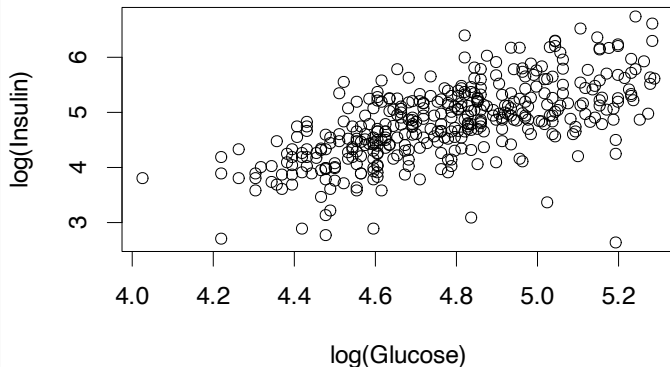
```
plot(data$log_glucose, data$log_insulin,  
      xlab = "log(Glucose)", ylab = "log(Insulin)",  
      main = " 散布図: log(glucose) vs log(insulin)")
```

→ 対数変換後の散布図の描画

```
data %>%  
  summarise(  
    Pearson = cor(log_glucose, log_insulin,  
                  method = "pearson"),  
    Spearman = cor(log_glucose, log_insulin,  
                  method = "spearman")  
  )
```

→ 対数変換後の2種類の相関係数を算出

## 対数変換後の散布図と相関係数



相関係数

	Pearson	Spearman
1	0.6345966	0.6589582

## 散布図と相関係数の比較：変換前後

変換前後での散布図および相関係数がどのように変わったか比較

```
par(mfrow=c(1,2)) → 横並びのレイアウト
```

```
plot(data$glucose, data$insulin,  
      xlab="Glucose", ylab="Insulin", main=" 変換前")  
plot(data$log_glucose, data$log_insulin,  
      xlab = "log(Glucose)", ylab = "log(Insulin)",  
      main = " 変換後")
```

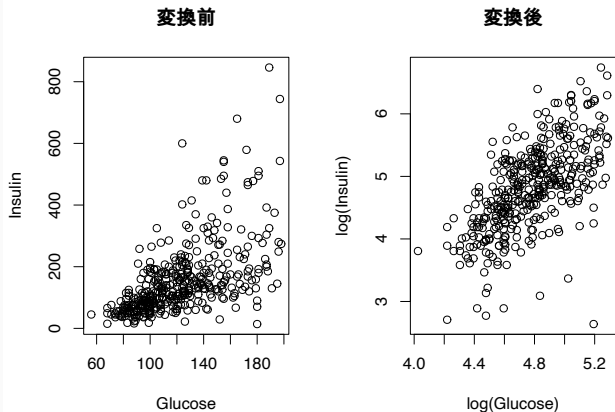
→ 変換前後の散布図の描画

```
data %>%  
  summarise(  
    Pearson_raw=cor(glucose, insulin, method="pearson"),  
    Pearson_log=cor(log_glucose, log_insulin,  
                     method = "pearson"),  
    Spearman_raw=cor(glucose,insulin,method="spearman"),  
    Spearman_log=cor(log_glucose, log_insulin,  
                      method = "spearman"))
```

→ 変換前後の相関係数の比較



# 散布図と相関係数の比較：変換前後



	Pearson_raw	Pearson_log	Spearman_raw	Spearman_log
1	0.581223	0.6345966	0.6589582	0.6589582

この例では、対数変換後のピアソンの相関係数の方が大きい  
対数変換で順位は変わらないのでスピアマンの相関係数も不変

**演習:** 1 番目の被験者の glucose の観測値に、単位誤り (10 倍) による外れ値を導入し、散布図および相関係数の変化を確認しましょう

```
data_outlier <- data
data_outlier$glucose[1] <- data$glucose[1] * 10
```

→ 外れ値の導入

```
plot(data_outlier$glucose, data_outlier$insulin,
      xlab = "Glucose", ylab = "Insulin",
      main = " 外れ値を含む散布図")
```

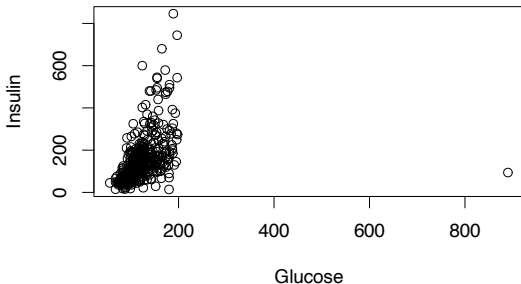
→ 散布図の描画

各データセットで相関係数を算出

```
data %>%  
  summarise(  
    Pearson = cor(glucose, insulin, method = "pearson"),  
    Spearman = cor(glucose, insulin, method = "spearman"  
    ),  
  )  
data_outlier %>%  
  summarise(  
    Pearson = cor(glucose, insulin, method = "pearson"),  
    Spearman = cor(glucose, insulin, method = "spearman"  
    ),  
  )
```

## 外れ値の影響の確認：出力例

外れ値を含む散布図



	Pearson	Spearman
1	0.581223	0.6589582

元データの相関係数

	Pearson	Spearman
1	0.3406767	0.6532737

外れ値データの相関係数

Spearman の相関は外れ値の影響を受けにくい

## 実用的なパッケージ、関数の紹介

---

# 患者背景の表の作成

table1 パッケージの関数 table1: 患者背景表 (Table 1) を簡潔に作成

- ・ 基本構文 :

```
table1(~ 項目1 + 項目2 + ... | グループ変数, data = データフレーム)
```


- ・ “~” の右に要約したい変数 (数値・カテゴリ)
- ・ “|” の右に層別する群変数 (例: 治療群、疾患の有無など)
- ・ HTML 形式で出力される (追加の手作業は必要)

(発展) 要約統計量の表示形式の変更 :

- ・ 引数 `render.continuous`, `render.categorical` で表示する統計量の詳細な設定が可能

同様の目的の他のパッケージも存在 (tableone パッケージなど)

## table1() の使用例：糖尿病データ

 **演習：** 糖尿病データの患者背景の表を作成しましょう。列は糖尿病あり、糖尿病なし、全集団の3列にしてください。

```
install.packages("table1")
library(table1)
data(PimaIndiansDiabetes2)
head(PimaIndiansDiabetes2)

table1(~ pregnant + glucose + pressure + triceps +
      insulin + mass + age | diabetes,
      data = PimaIndiansDiabetes2)
```

# 出力例：糖尿病の有無による患者背景

	neg (N=500)	pos (N=268)	Overall (N=768)
<b>pregnant</b>			
Mean (SD)	3.30 (3.02)	4.87 (3.74)	3.85 (3.37)
Median [Min, Max]	2.00 [0, 13.0]	4.00 [0, 17.0]	3.00 [0, 17.0]
<b>glucose</b>			
Mean (SD)	111 (24.8)	142 (29.6)	122 (30.5)
Median [Min, Max]	107 [44.0, 197]	140 [78.0, 199]	117 [44.0, 199]
Missing	3 (0.6%)	2 (0.7%)	5 (0.7%)
<b>pressure</b>			
Mean (SD)	70.9 (12.2)	75.3 (12.3)	72.4 (12.4)
Median [Min, Max]	70.0 [24.0, 122]	74.5 [30.0, 114]	72.0 [24.0, 122]
Missing	19 (3.8%)	16 (6.0%)	35 (4.6%)
<b>triceps</b>			
Mean (SD)	27.2 (10.0)	33.0 (10.3)	29.2 (10.5)
Median [Min, Max]	27.0 [7.00, 60.0]	32.0 [7.00, 99.0]	29.0 [7.00, 99.0]
Missing	139 (27.8%)	88 (32.8%)	227 (29.6%)
<b>insulin</b>			
Mean (SD)	130 (102)	207 (133)	156 (119)
Median [Min, Max]	103 [15.0, 744]	170 [14.0, 846]	125 [14.0, 846]
Missing	236 (47.2%)	138 (51.5%)	374 (48.7%)
<b>mass</b>			
Mean (SD)	30.9 (6.56)	35.4 (6.61)	32.5 (6.92)
Median [Min, Max]	30.1 [18.2, 57.3]	34.3 [22.9, 67.1]	32.3 [18.2, 67.1]
Missing	9 (1.8%)	2 (0.7%)	11 (1.4%)
<b>age</b>			
Mean (SD)	31.2 (11.7)	37.1 (11.0)	33.2 (11.8)
Median [Min, Max]	27.0 [21.0, 81.0]	36.0 [21.0, 70.0]	29.0 [21.0, 81.0]



# 複数変数の関係の視覚化

GGally パッケージの関数 `ggpairs`: 複数の変数間の散布図、相関係数、分布を一括表示可能


- ・ 基本構文 :

```
ggpairs(データフレーム, columns = c("変数1", "変数2",  
  ...))
```

- ・ `columns=` に表示したい変数を指定
- ・ `aes(color = グループ変数)` を指定することで、グループごとに色分けも可能

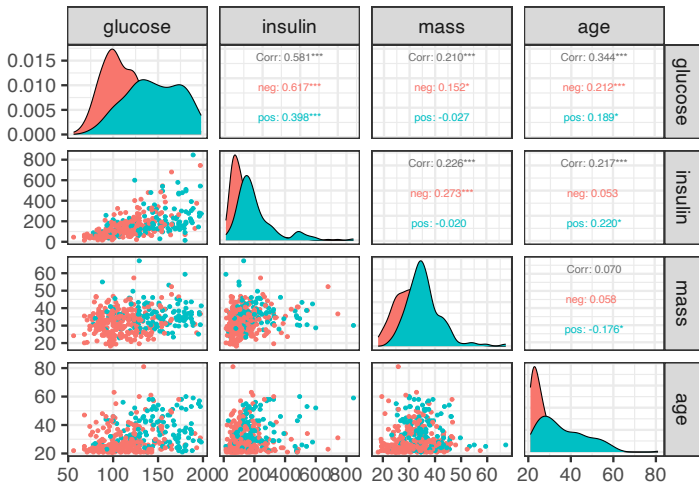
(補足) `pairs()` 関数は base R の機能で、簡易的なペアプロットを提供

## ggpairs() の使用例：糖尿病データ

 **演習:** glucose, insulin, mass, age について ggpairs を用いて各変数間の散布図、相関係数、分布を一括表示してください

```
install.packages("GGally")  
library(GGally)  
library(mlbench)  
data(PimaIndiansDiabetes2)  
data <- na.omit(PimaIndiansDiabetes2)  
  
ggpairs(data,  
        columns = c("glucose", "insulin", "mass", "age"),  
        aes(color = diabetes))
```

## 出力例 : ggpairs による多変数の関係可視化



## (発展) 標本相関係数と母相関係数

1. (1) 式の関係がある時、4つの組  $(X_1, Y_1), (X_1, Y_2), (X_1, Y_3), (X_1, Y_4)$  の母相関係数を求めて標本相関係数と比較してみましょう
2. 標本の数を増やした時に標本相関係数が母相関係数に近づくことを乱数生成により確かめてみましょう

### ヒント

1. 2つの変数  $X, Y$  の母相関係数の定義

$$r = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)}\sqrt{\text{Var}(Y)}}$$

- ・ 2つの変数  $X, Y$  が独立なとき、 $\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y)$
- ・ 標準正規分布の三次モーメントはゼロ ( $X \sim N(0, 1) \Rightarrow E[X^3] = 0$ )

2.  $n = 10000$  などで計算