統計的方法への疑問

Rで計算して解決

かすや
粕谷　英一
（九大・理・生物）

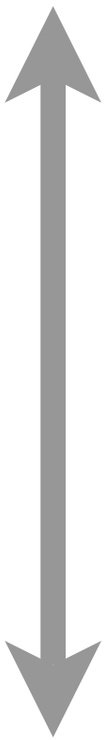
2008.3（生態学会大会、福岡）

統計的方法

『○●は、××のときに使うのはよくない』

『×△のときには、 ●○は 使うといい』

統計学の研究テーマでもある



使える”生活の知恵”

誤ってはいないが抜けている

おおざっぱで役立たず

”誤った信仰”

目的変数が割合のとき直線回帰はまずい

ブロックは作れる限り作った方がいい

検定で有意にならなくても共変量としていれるといい

正規分布＆等分散だったらt検定

分割表はカイ2乗検定

分割表は小さい値があればFisherの検定

割合のデータ解析ではoverdispersion無視はありえない

『分散に大きな差があるときは、ノンパラメトリクス』

『ノンパラメトリクスを使えば分布は気にしないでいい』

ほとんど都市伝説

解析的に解決すれば見通しもいいが・・・

解析的に解決しないことも多い

実際に計算して判断しよう

統計学の論文でも数値的に計算しているものが沢山

Rでやる”ご利益”

関数の計算結果の再利用

Rの豊富な関数たち
やりたいことのかなりをカバー

すでにある関数　働かせる

ど～んと

R：組み合わせでできることが広がる

乱数

目次

統計的方法の代表的な”良い・悪い”

とても簡単な例．相関係数rの検定

かなり簡単な例．ロジスティック回帰的な状況

モデル選択（予測の最適化）

AIC　赤池情報量基準

-2×最大対数尤度＋2×自由パラメーター数

正しいモデルの採用率

検定　　よい検定・悪い検定

第１種の誤り　　有意確率・危険率に対応

本当は帰無仮説が正しいとき
帰無仮説を捨て対立仮説を採用

第２種の誤り　　（１－検出力）

本当は対立仮説が正しいとき
帰無仮説を捨てない

検定　　**よい検定・悪い検定**

第１種の誤りの率が宣言した有意水準以下

のものの中で

なるべく

第２種の誤りが小さい（検出力が高い）

Neyman-Pearson基準

正規分布のときの相関係数r

検定としての”良い・悪い”

第１段階

有意なものが出る率は宣言した水準以下か？

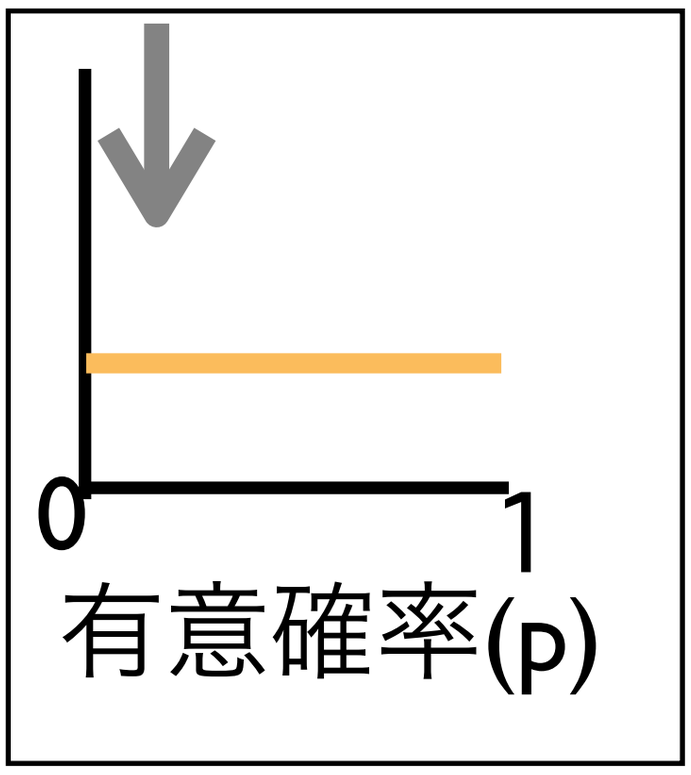
5%水準と言ってるのに10%水準とかに

なっていないか

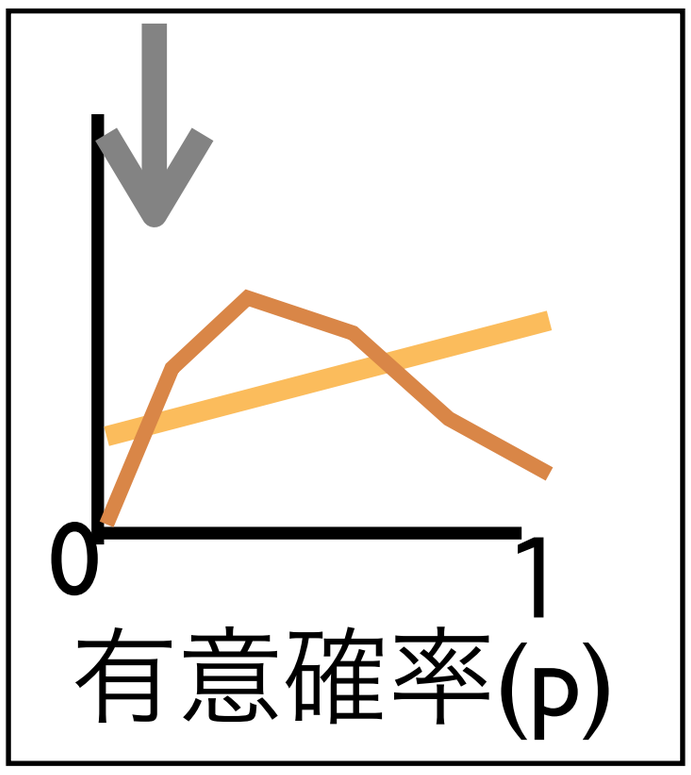
帰無仮説のモデルのもとで　有意確率の分布

帰無仮説のモデルが正しいとき　　一般論

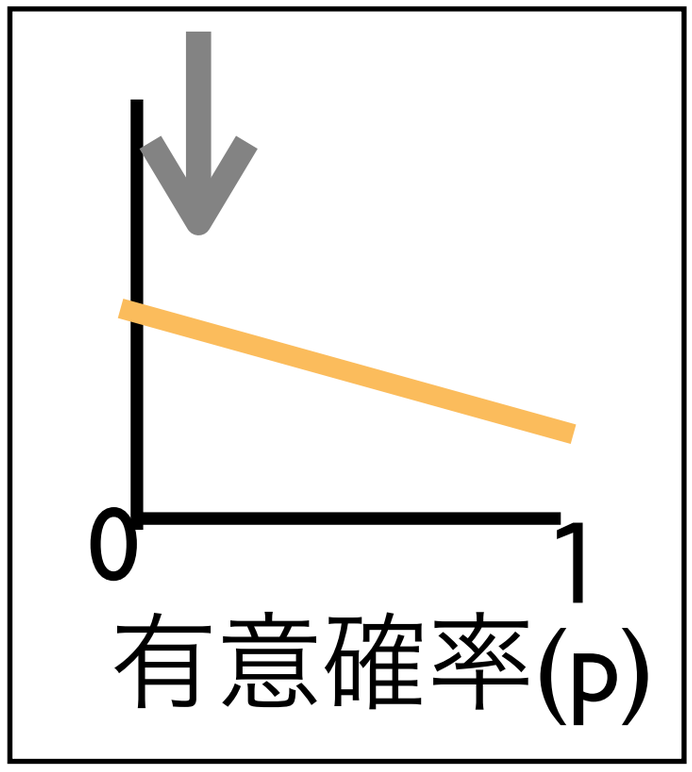
有意確率の分布　　理想的には一様分布



理想的



鈍感



だめ
（公称より甘い）

相関係数rの検定での有意確率の分布

（母集団：無相関、正規分布）サンプル数10

50000回繰り返し

```
pr1<-numeric(50000)
xx1<-numeric(10)
yy1<-numeric(10)
for (i1 in 1:50000) {
xx1<-rnorm(10,mean=9,sd=2)
yy1<-rnorm(10,mean=7,sd=4)
rr1<-cor.test(xx1,yy1)
pr1[i1]<-rr1$p.value
}
sum(pr1<=0.05)
sum(pr1<=0.01)
```

数値を入れておく場所（ベクトル）用意

繰り返し計算

計算結果集計

numeric(個数)　個数だけ数値の入れ場所を用意」

```
pr1<-numeric(50000)
xx1<-numeric(10)
yy1<-numeric(10)
```

10　サンプルサイズ
（データ点個数）

50000　繰り返し数

数値入れ場所用意

繰り返し計算

計算結果集計

for (変数in範囲){}　変数を１ずつ変化させ繰り返し

```
for (i1 in 1:50000) {
xx1<-rnorm(10,mean=9,sd=2)
yy1<-rnorm(10,mean=7,sd=4)
rr1<-cor.test(xx1,yy1)
pr1[i1]<-rr1$p.value
}
```

数値入れ場所用意

繰り返し計算

計算結果集計

```
xx1<-rnorm(10,mean=9,sd=2)
```

母平均9、母標準偏差2の正規乱数10個をxx1に入れる

```
for (i1 in 1:50000) {
xx1<-rnorm(10,mean=9,sd=2)
yy1<-rnorm(10,mean=7,sd=4)
rr1<-cor.test(xx1,yy1)
pr1[i1]<-rr1$p.value
}
```

数値入れ場所用意

繰り返し計算

計算結果集計

```
rr1<-cor.test(xx1,yy1)
```

rr1に相関係数の検定結果

```
pr1[i1]<-rr1$p.value
```

rr1から有意確率を取り出してpr1に

```
for (i1 in 1:50000) {
xx1<-rnorm(10,mean=9,sd=2)
yy1<-rnorm(10,mean=7,sd=4)
rr1<-cor.test(xx1,yy1)
pr1[i1]<-rr1$p.value
}
```

数値入れ場所用意

繰り返し計算

計算結果集計

sum()　合計を計算する

sum(条件式)　条件式に合うデータの個数

```
sum(pr1<=0.05)
sum(pr1<=0.01)
```

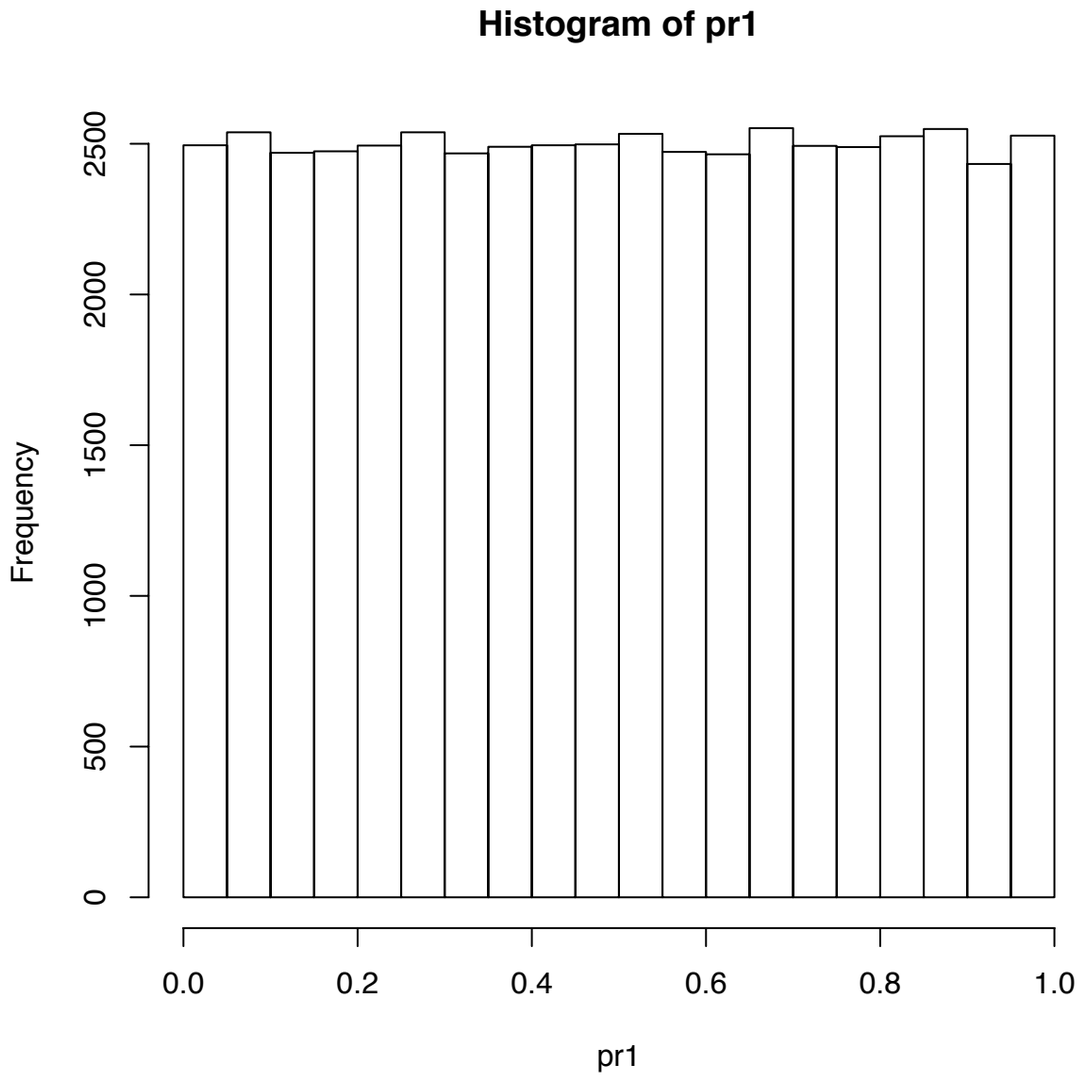
数値入れ場所用意

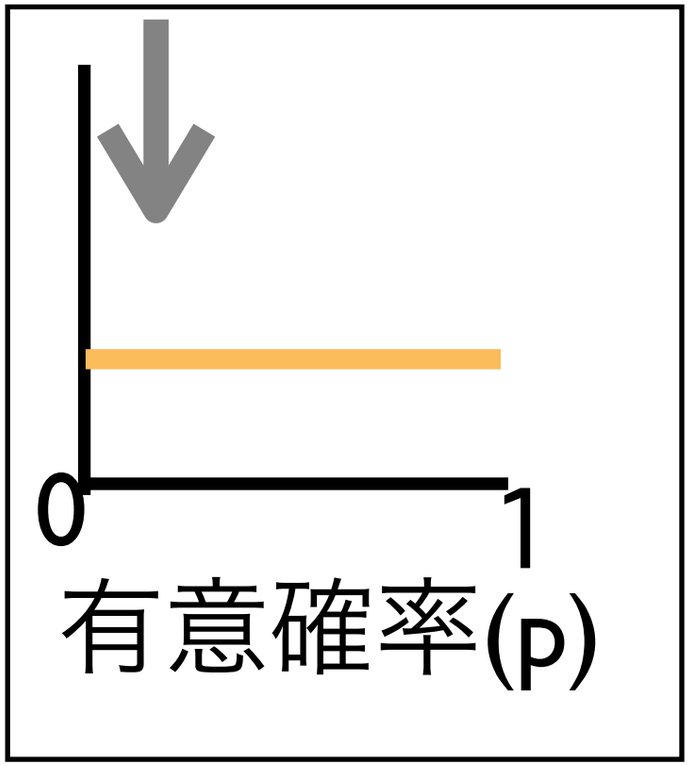
繰り返し計算

計算結果集計

帰無仮説が正しい（母集団が無相関）とき

回数





理想的

相関係数rを検定したときの有意確率

hist(pr1)

hist(pr1)で作ったものをPostscript保存してPDF化

どのくらいのばらつきが生じる？

5%　　二項分布　　分散

10000回　±0.4%程度に0.95の確率で入る

1000回　　±1.4%程度に0.95の確率で入る

50%　　二項分布　　分散

10000回　　±1%程度に0.95の確率で入る

1000回　　±3%程度に0.95の確率で入る

ある条件　確率的なばらつき

計算上の表現　**乱数**

```
xx1<-rnorm(10,mean=9,sd=2)
```

乱数 → r＋norm ← 正規分布

d　確率密度関数

q　分布関数　p　確率点

正規分布でなく
対数正規分布

rnorm→rlnorm

binom
lnorm　cauchy
t
exp　gamma
f　nbinom

出てきそうな確率分布は
そろっている

相関係数rの検定での有意確率の分布

（母集団：無相関、正規分布）サンプル数10

50000回繰り返し

```
pr1<-numeric(50000)
xx1<-numeric(10)
yy1<-numeric(10)
for (i1 in 1:50000) {
xx1<-rnorm(10,mean=9,sd=2)
yy1<-rnorm(10,mean=7,sd=4)
rr1<-cor.test(xx1,yy1)
pr1[i1]<-rr1$p.value
}
sum(pr1<=0.05)
sum(pr1<=0.01)
```

数値を入れておく場所（ベクトル）用意

繰り返し計算

計算結果集計

```
rr1<-cor.test(xx1,yy1)
```

```
> rr1

        Pearson's product-moment correlation
（中略）
t = 0.1516, df = 48, p-value = 0.8801
（中略）
sample estimates:
cor   0.02187758
```

rr1から、

ほしいもの（ここでは有意確率）を取り出す

計算結果オブジェクトの中身調べ

関数attributes()

```
> attributes(rr1)
$names
[1] "statistic"   "parameter"   "p.value"     "estimate"
"null.value"
[6] "alternative" "method"      "data.name"
"conf.int"
```

```
> rr1$p.vaue
[1] 0.880131
```

計算結果オブジェクトの中身調べ

関数str()

```
> str(rr1)
List of 9
 $ statistic  : Named num 0.152
  ..- attr(*, "names")= chr "t"
 $ parameter  : Named num 48
  ..- attr(*, "names")= chr "df"
 $ p.value    : num 0.88
$ estimate   : Named num 0.0219
 ..- attr(*, "names")= chr "cor"
$ null.value : Named num 0
 ..- attr(*, "names")= chr "correlation"
$ alternative: chr "two.sided"
$ method     : chr "Pearson's product-moment correlation"
$ data.name  : chr "xx1 and yy1"
$ conf.int   : atomic [1:2] -0.258  0.298
 ..- attr(*, "conf.level")= num 0.95
- attr(*, "class")= chr "htest"
```

```
rr1<-cor.test(xx1,yy1)
```

rr1から、有意確率を取り出す

```
> rr1$p.vaue
[1] 0.880131
```

計算結果オブジェクトの中身調べ

attributes()やstr()で調べて

計算結果オブジェクト名$項目の名前

新しい関数では

計算結果オブジェクト名@項目の名前

S4オブジェクト　　スロット

相関係数rの検定での有意確率の分布

（母集団：無相関、正規分布）サンプル数10

50000回繰り返し

```
pr1<-numeric(50000)
xx1<-numeric(10)
yy1<-numeric(10)
for (i1 in 1:50000) {
xx1<-rnorm(10,mean=9,sd=2)
yy1<-rnorm(10,mean=7,sd=4)
rr1<-cor.test(xx1,yy1)
pr1[i1]<-rr1$p.value
}
sum(pr1<=0.05)
sum(pr1<=0.01)
```

数値を入れておく場所（ベクトル）用意

繰り返し計算

計算結果集計

相関係数rの検定

乱数は使っているが
計算できない場合が生じない

xが全部同じ値
yが全部同じ値

乱数＋繰り返し計算　　予想しない数値

エラーが出て途中で止まる

関数try()　　関数tryCatch()

目次

統計的方法の代表的な”良い・悪い”

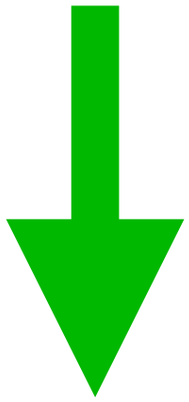
とても簡単な例．相関係数rの検定

かなり簡単な例．ロジスティック回帰的な状況

量的変数

体サイズ　　温度

標高　　など



２値変数

生存／死亡　　メス／オス

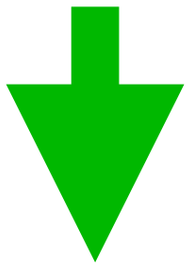
勝ち／負け　　など

ロジスティック回帰　　量的変数が確率に影響

２処理の比較　　生存個体と死亡個体の体サイズ

原因／結果が逆転？

量的変数　　体サイズ　温度　標高　など



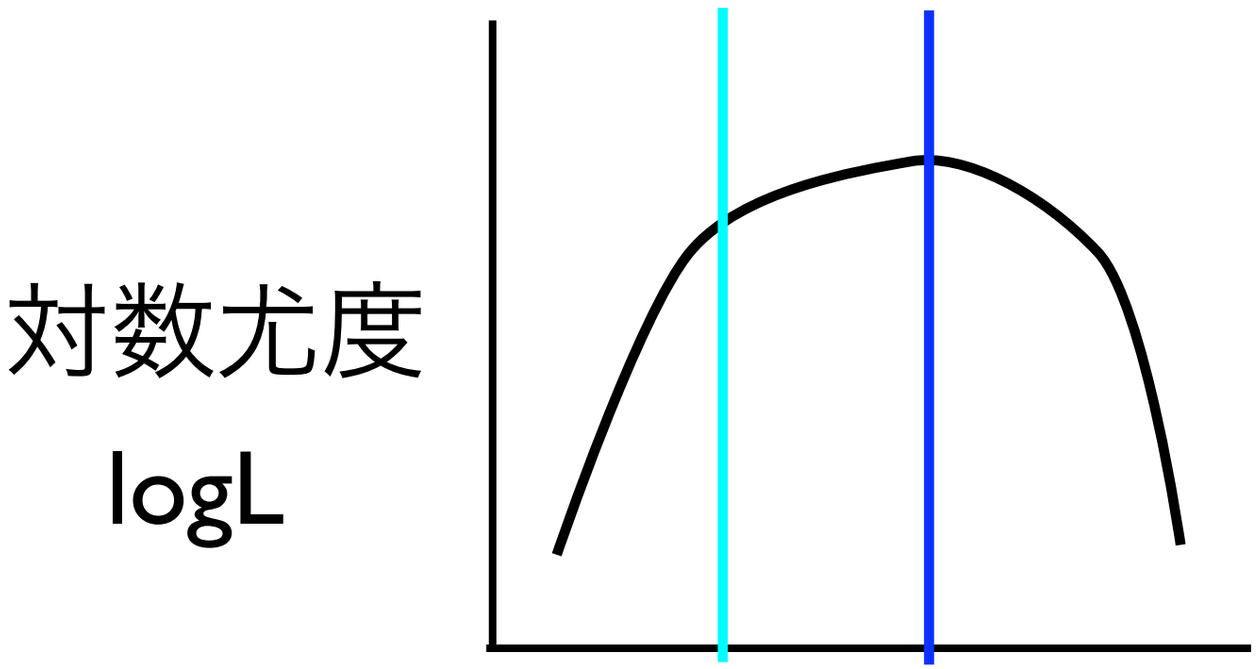
２値変数　　生存／死亡　　メス／オス
勝ち／負け　　など

ロジスティック回帰　　尤度比検定　　Wald検定

２処理の比較　　中央値検定　　Wilcoxon検定
（Mann-WhitneyのU検定）

t検定（Welch）

最尤法（maximum likelihood）



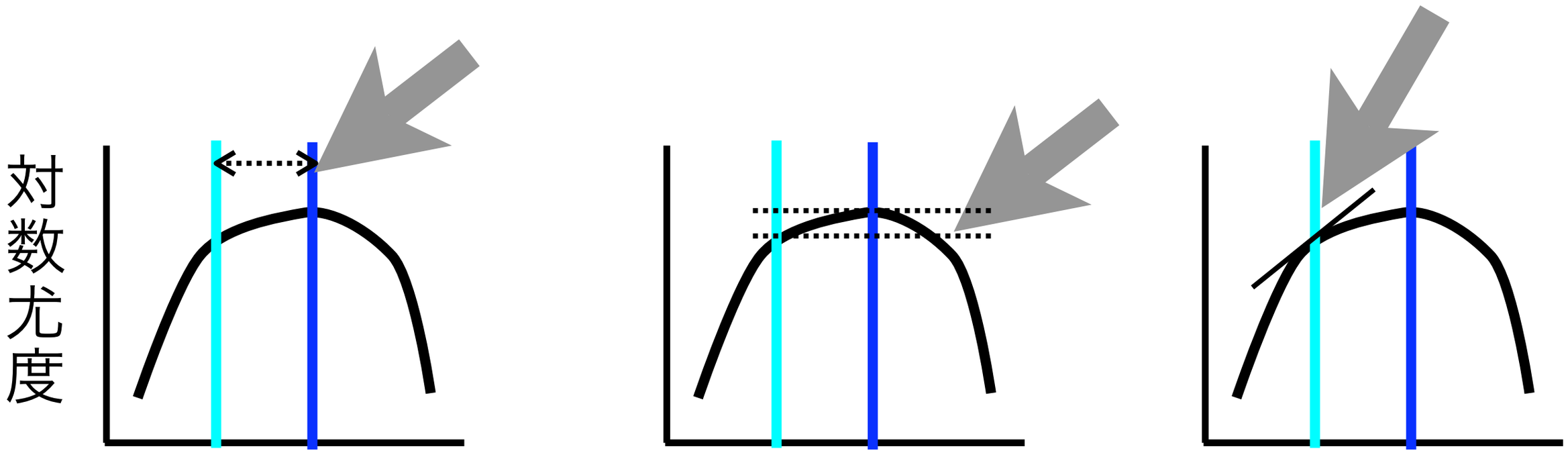
帰無仮説のモデル

最尤推定値

直線回帰なら0

直線回帰なら
推定された傾き

最尤法と３つの検定



Wald

最尤推定値

尤度比

対数尤度の差

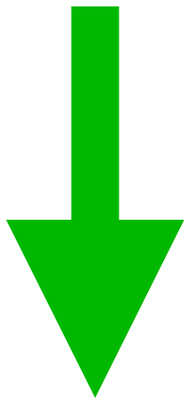
スコア

（Lagrange乗数）

帰無仮説の
モデルでの
対数尤度の傾き

サンプル数無限大では同じ

量的変数　体サイズ　温度　標高　など



２値変数　生存／死亡　メス／オス　など

ベルヌイ過程（overdispersionなし）

サンプルサイズ100　（1000までやった）

使ったプログラム

やや長いのはいろいろな方法を評価したから

```
fp1<-numeric(10000)
wp1<-numeric(10000)
lp1<-numeric(10000)
tp1<-numeric(10000)
lrp1<-numeric(10000)
cnt1<-numeric(10000)
fcnt1<-numeric(10000)
aic1<-numeric(10000)
n1<-100
x1<-numeric(n1)
for (i in 1:n1) {
x1[i]<-qnorm(i/(n1+1),mean=20.0,sd=4.0)
}

d1<-numeric(n1)
for (i in 1:n1) {
d1[i]<-0.5
}
y1<-numeric(n1)

for (i1 in 1:10000) {
for (i in 1:n1) {
 if (d1[i]>=runif(1)) {y1[i]<-1} else {y1[i]<-0}
}
xa<-sum(head(y1,n1/2))
xc<-(n1/2-xa)
xb<-sum(tail(y1,n1/2))
xd<-(n1/2-xb)
fm1<-matrix(c(xa,xb,xc,xd),ncol=2)
fp1[i1]<-fisher.test(fm1)$p.value
fcnt1[i1]<-(xa+xb+xc+xd)
yf1<-factor(y1)
ww1<-wilcox.test(x1~y1)
wp1[i1]<-ww1$p.value
lgt0<-glm(yf1~1,family=binomial)
lgt1<-glm(yf1~x1,family=binomial)
slgt1<-summary(lgt1)
lp1[i1]<-slgt1$coef[2,4]
lrt1<-2*abs(logLik(lgt1)[1]-logLik(lgt0)[1])
lrp1[i1]<-(1-pchisq(lrt1,1))
tr1<-t.test(x1~yf1)
tp1[i1]<-tr1$p.value
cnt1[i1]<-1
aic1[i1]<-(lgt1$aic-lgt0$aic)
}
sum(fp1<=0.05)
sum(lp1<=0.05)
sum(wp1<=0.05)
sum(lrp1<=0.05)
sum(tp1<=0.05)
sum(fp1<=0.01)
sum(lp1<=0.01)
sum(wp1<=0.01)
sum(lrp1<=0.01)
sum(tp1<=0.01)
sum(aic1<=0.0)
sum(fcnt1)/10000
```

尤度比、Wald、AICのみ

```
lp1<-numeric(10000)
lrp1<-numeric(10000)
aic1<-numeric(10000)
n1<-100
x1<-numeric(n1)
for (i in 1:n1) {
x1[i]<-qnorm(i/(n1+1),mean=20.0,sd=4.0)
}

d1<-numeric(n1)
for (i in 1:n1) {
d1[i]<-0.5
}
y1<-numeric(n1)

for (i1 in 1:10000) {
for (i in 1:n1) {
 if (d1[i]>=runif(1)) {y1[i]<-1} else {y1[i]<-0}
}

yf1<-factor(y1)
lgt0<-glm(yf1~1,family=binomial)
lgt1<-glm(yf1~x1,family=binomial)
slgt1<-summary(lgt1)
lp1[i1]<-slgt1$coef["x1","Pr(>|z|)"]
lrt1<-2*abs(logLik(lgt1)[1]-logLik(lgt0)[1])
lrp1[i1]<-(1-pchisq(lrt1,1))
aic1[i1]<-(lgt1$aic-lgt0$aic)
}
sum(lp1<=0.05)
sum(lrp1<=0.05)
sum(aic1<=0.0)
```

数値を入れておく場所用意

量的変数の値と確率との準備

繰り返し計算

計算結果集計

glm()の結果オブジェクトから検定結果を取り出す（例、有意確率）
attributes()やstr()でさがす

> glmの結果のsummaryオブジェクト$coefficients

```
                Estimate Std. Error    z value      Pr(>|z|)
(Intercept) 18.23579822    3.967524 4.5962662 3.140334e-05
x01          0.06201217    0.409028 0.1516086 8.801311e-01
```

glmの結果のsummaryオブジェクトのcoefficients 行や列に名前のついた行列

```
coefficients["x01","Pr(>|z|)"]
```

↑久保さん指摘による

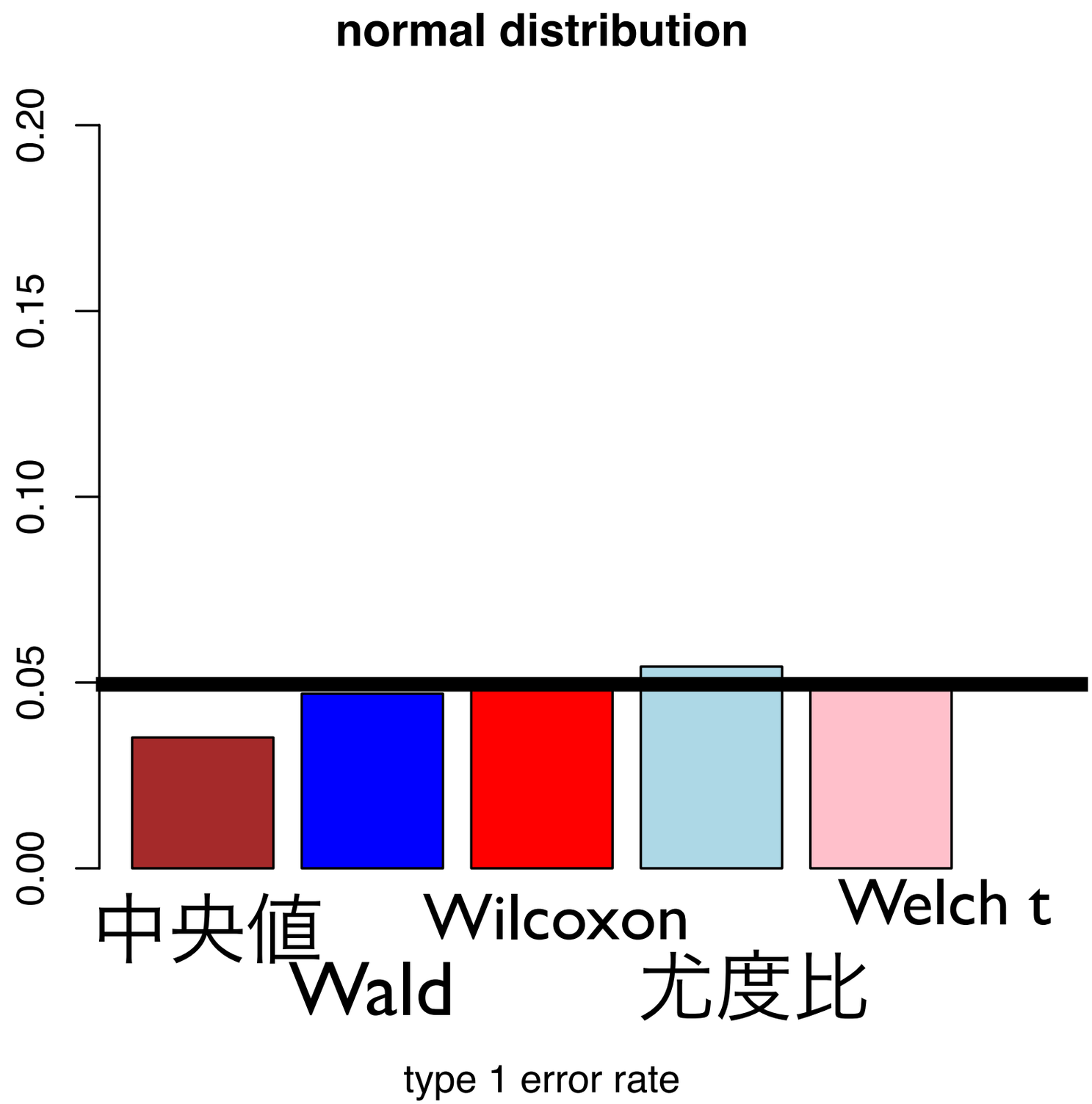
あるいは

glmの結果のsummaryオブジェクトcoefficients[8]
または
glmの結果のsummaryオブジェクト$coefficients[2,4]

説明変数（量的）ー正規分布

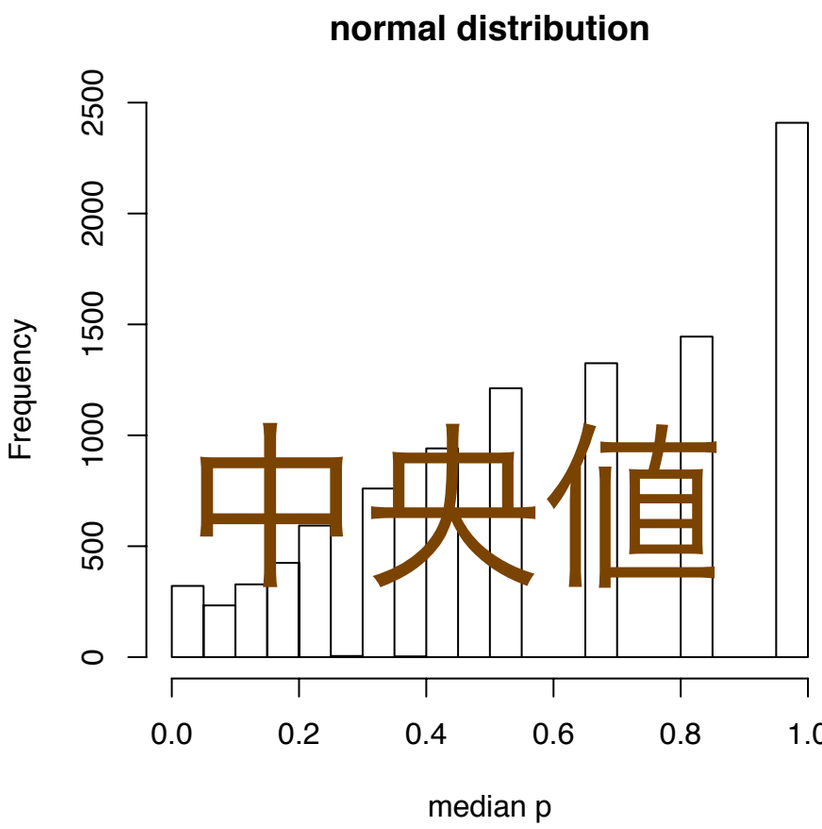
帰無仮説が正しいとき

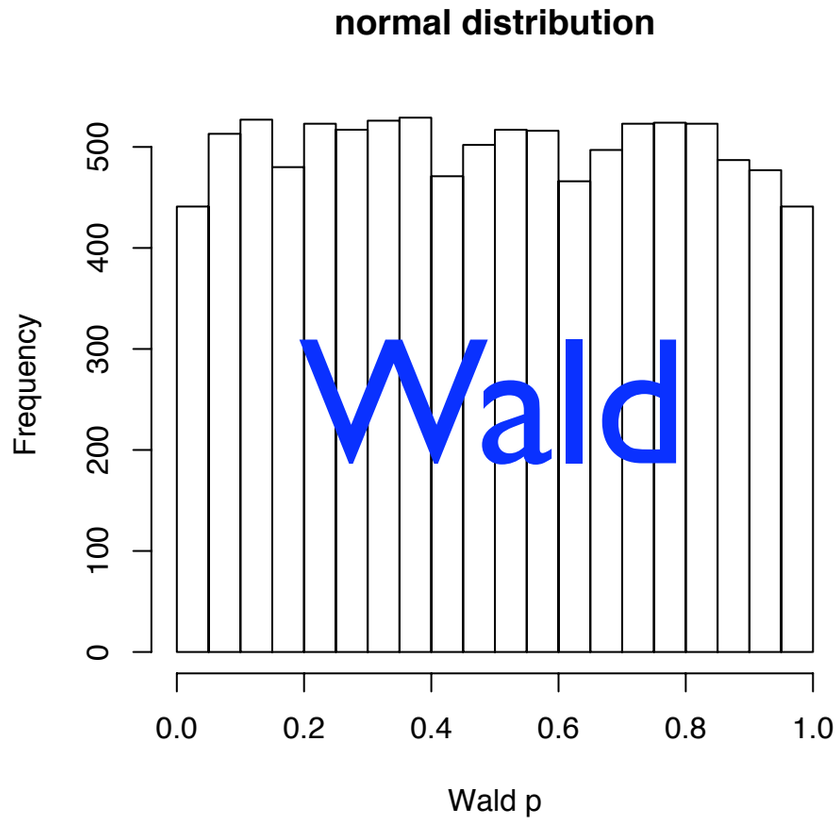
5%水準で検定したとき有意と判定する率

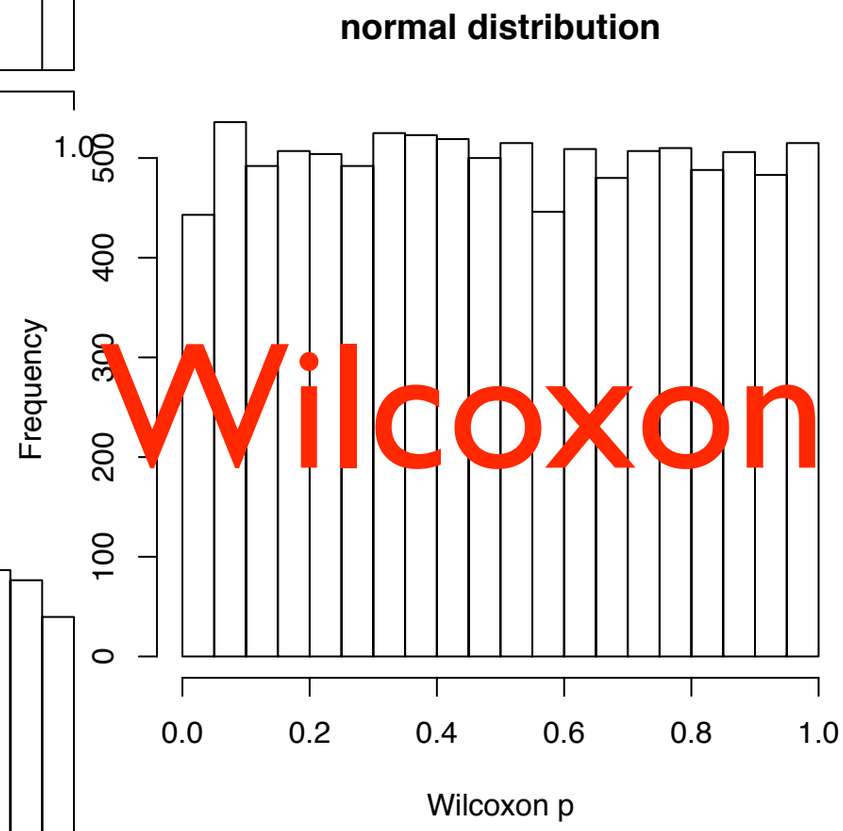


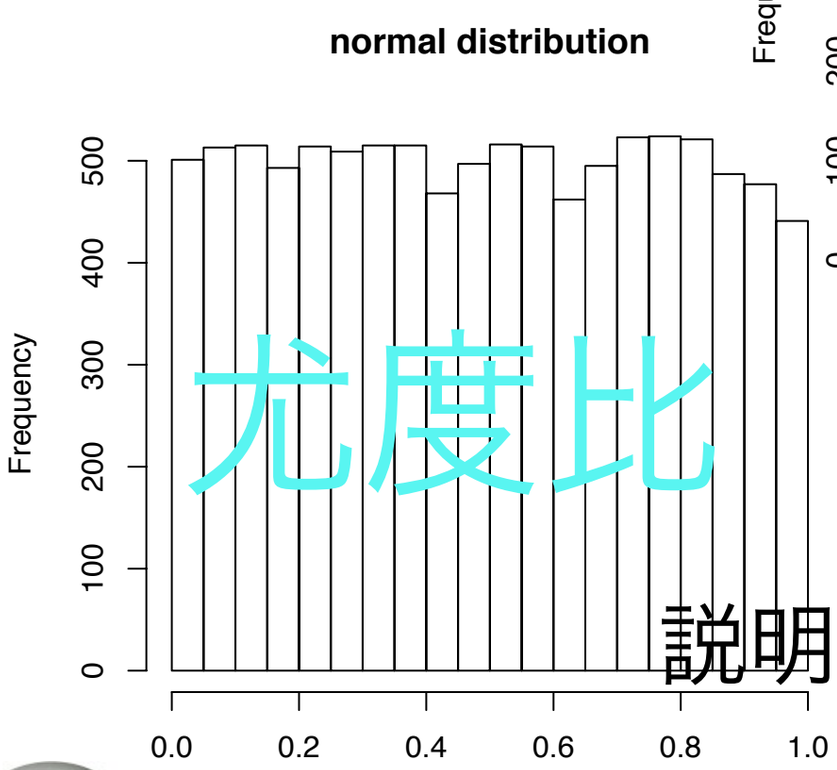
barplot()

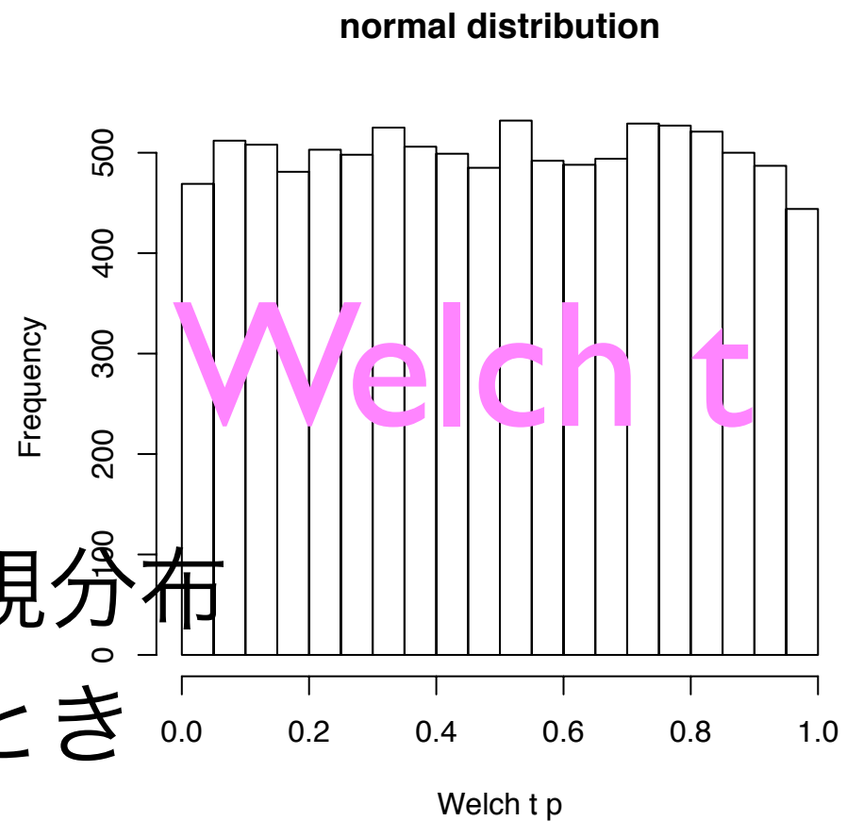
有意確率の
ヒストグラム











説明変数（量的）ー正規分布

帰無仮説が正しいとき

hist()

説明変数の分布も解析結果に影響
（目的変数の分布は注意を払われるが）

説明変数ー目的変数をひっくり返すのは要注意

検定specific：単に第１種の誤りの率が宣言値以下かだけでは不十分

正規線形モデル（誤差＝等分散の正規分布）では

エラーが出て途中で止まる

計算できない場合　目的変数が全部同じ値

完全分離

x＞ある値　yがすべて1

xくある値　yがすべて0

関数try()　　関数tryCatch()

関数の計算結果の再利用

その他の場合にも

面倒なところはRの既存の関数で

例．パラメトリック・ブートストラップ