Statistical Tests and Parameter Estimations in R

Candong Chen cc4766

Moment Estimation

- List the estimations

$$E[X^k] = g_k(\theta_1, \cdots, \theta_m), k = 1, \cdots, m$$

- Solve the equation set

$$\theta_k = \theta_k(E[X], E[X^2], \cdots, E[X^m])$$

- Replace $E[X^k]$ with $M_k = \frac{1}{n}\sum_{i=1}^{n} X_i^n$, we get the moment estimations

$$\hat{\theta}_k = \theta_k(M_1, \cdots, M_m)$$

```
rootSolve::multiroot(f, start, maxiter =
100)
```

Given n (nonlinear) equations, we use multiroot to solve for n roots.
1. **f** is the euqation set w.r.t. estimators.
2. **start** is the initial guesses for unknown variables of **f**.
3. **maxiter** is maximal number of iterations allowed.

MLE

- Build likelihood function

$$L = P(X_1 = x_1, \cdots, X_n = x_n) = \Pi_{i=1}^{n} f(x_i; \theta)$$

- MLE of θ is $\hat{\theta} = \arg\max_\theta \log L$
- Solve the equation $\frac{d \log L}{d\theta} = 0$ and verify if $\frac{d^2 \log L}{d\theta^2} < 0$

```
optim(par, fn, method = c("Nelder-Mead",
"BFGS", "CG", "L-BFGS-B", "SANN", "Brent"),
lower = -Inf, upper = Inf)
```

1. **optim** calculates maximum likelihood estimates for multiple parameters distributions.
2. **par** sets the initial value of parameter.
3. **fn** is the likelihood function.
4. **method** provides six ways to calculate extremum.
5. **lower** and **upper** are the lower and upper bounds of parameters respectively

EM Algorithm

$L(\theta; X) = \int p(Z|X, \theta) p(X|\theta)$
Iteratively applying these two steps:

- Expectation step (E step):

$$Q(\theta|\theta^{(t)}) = E_{Z|X,\theta^{(t)}}[\log L(\theta; X, Z)]$$

- Maximization step (M step):

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta|\theta^{(t)})$$

```
mclust::Mclust(data, G = NULL, modelNames =
NULL, prior = NULL, control = emControl(),
initialization = NULL)
```

1. **G** specifies the number of categories.The default is G=1:9.
2. **modelNames** specifies the fitting model during the EM algorithm.
3. **prior** allows specification of a conjugate prior on the means and variances through the function priorControl.
4. **control** specifies the control parameters for the EM algorithm.

Distribution Estimation

Empirical Distribution	`ecdf(x)`
Histogram	`hist(x, breaks)`
KDE	`density(x, bw = "nrd0", adjust, kernel)`

In KDE:
1. **bw** is the smoothing bandwidth to be used.
2. the bandwidth used is actually **adjust*bw**.
3. **kernel** specifies the smoothing kernel to be used.

Parametric Test

Normal Distribution

t test (mean):
```
t.test(x, y=NULL, alternative=c("two.sid-
ed","less","greater"), mu=0, paired=FALSE,
var.equal=FALSE, conf.level=0.95)
```

F test (variance):
```
var.test(x, y, ratio = 1, alternative
= c("two.sided", "less", "greater"),
conf.level = 0.95)
```

Binomial Distribution

binomial test(p):
```
binom.test(x, n, p = 0.5, alternative
= c("two.sided", "less", "greater"),
conf.level = 0.95)
```

Bivariate Correlation Test

```
cor.test(x, y, alternative = c("two.sided",
"less", "greater"), method = c("pearson",
"kendall", "spearman"), exact = NULL,
conf.level = 0.95, ...)
```

1. Pearson's correlation coefficient has a precondition that x and y are normally distributed.
2. Spearman's correlation coefficient is used for continuous variables.
3. Kendall's correlation coefficient is used for ordinal categorical variables.

Nonparametric Test (completely known)

Pearson's chi-squared test

```
chisq.test(x, y = NULL, correct = TRUE, p =
rep(1/length(x), length(x)), rescale.p =
FALSE
```

which is used to test if x and y have the same distribution.
1. **correct** is a logical variable that indicates whether it is used for continuous correction.
2. **p** is the theoretical probability that the original hypothesis falls in the intervals, and the default value represents uniform distri- bution.
3. **rescale.p** is a logical variable. FALSE (default) requires that sum of the input **p** equals 1. When TRUE is selected, this condition is not required and the program recalculates the **p** value.

Shapirowilk test

```
shapiro.test(x)
```

which is used to test if x is normally distributed.

Nonparametric Test (unknown parameters)

Kolmogorov-Smirnov test

```
ks.test(x, y, ..., alternative =
c("two.sided", "less", "greater"), exact
= NULL)
```

which is used to test if x and y have the same distribution.
1. **y** can be either data or a character string specifying distribution such as **"pnorm"** or **"pexp"**.
2. **...** are parameters of the distribution specified (as a character string) by y.
3. **exact** is a logical indicating whether an exact p-value should be computed.

Contingency table test

To test if the two columns in x are independent

```
chisq.test(x, correct = False)
```

the minimum theoretical frequency: T
the total frequency: N

Pearson's chi-squared test can be used only if $T \geq 5$ and $N \geq 40$.

```
chisq.test(x,correct = True)
```

Pearson's chi-squared modified test should be used if $1 \leq T < 5$ and $N \geq 40$.

```
fisher.test(x, alternative =''two.sided''
```

Fisher exact test should be used if $T < 1$ or $N < 40$.

Nonparametric Test (completely known)

Sign test

```
binom.test(sum(x>M), length(x), alternative
= c("less", "greater"), conf.level=0.95)
```

which is used to test if the median of X is greater (less) than M.

In the sign test method, only the number of symbols is counted, without considering the magnitude of the absolute value of each sign difference.

Wilcoxon signed-rank test (univariate)

```
wilcox.test(x, mu=M, alternative =
c("two.sided", "less", "greater"),
exact=FALSE, correct=FALSE)
```

which is used to test if the median of x is equal (greater or less) to M.

Wilcoxon signed-rank test (multivariate)

```
wilcox.test(x, y, alternative =
c("two.sided", "less", "greater"), paired
= TRUE, correct = TRUE, conf.level = 0.95)
```

which is used to test if the median of x is equal (greater or less) to the median of y.