MAPLE CARD

All commands and constants must be typed in the case (UPPER or lower) that is indicated

Syntax Rules

`;`	a semicolon or a colon and a RETURN must follow every command.
%	refers to previous display.

Arithmetic Operations

`+`	addition	–	subtraction
`*`	multiplication	/	division
`^`	exponentiation	!	factorial

Some Mathematical Constants

`Pi`	π	`exp(1)`	e
`-infinity`	$-\infty$	`infinity`	∞

Mathematical Functions and Expressions

`f:=expression;`	labels the expression as `f`
`g:=x->expression;`	defines a function `g` in terms of the expression
`exp(x)`,	exponential function
`log(x), ln(x)`	natural logarithms of `x`
`sqrt(x)`	square root of `x`
`abs(x)`	absolute value of `x`
`sin, cos, tan, sec, csc, cot`	trig. functions
`arcsin, arccos, arctan`	inverse trig. functions
`arccot, arcsec, arccsc`	" "
`GAMMA(n)`	the gamma function

Some Basic Commands

`choose(L,k)`[1]`;`	produces combinations of size `k` from the elements of the list `L`
`diff(f,x);`	differentiates `f` with respect to `x`
`diff(f,x$n);`	computes the `n`th derivative of `f` w.r.t. `x`
`denom(f);`	denominator of the expression `f`
`evalf(f);`	numerical evaluation of `f`
`expand(f);`	algebraic expansion of `f`
`[seq(coeff(expand((1+t)^9),t,n), n=0..9)];`	extracts coefficients of `t^n` as a list

`int(f,x);` attempts to find the antiderivative of the expression `f` w.r.t. `x`. If unsuccessful, the integral is returned without evaluation.

`int(f, x=a..b);` attempts to find the definite integral of `f` for `x` on the interval (`a,b`). If unsuccessful, the integral is returned. If a numerical solution is desired, follow up with an `evalf` command.

`numer(f);` numerator of the expression `f`

`permute(n)`[1]`;` produces a list of all `n`! permutations of {1, 2, ... n}

`plot(f,x=a..b);` plots the expression `f` (defined in terms of `x`) on the interval (`a,b`)

`plot({f1,f2,f3},x=a..b);` plots expressions `f1, f2` and `f3` (all defined in terms of `x`) on the same set of axes.

`randcomb(L,k)`[1]`;` for a list, `L`, gives a random combination of `k` elements of `L`

`randperm(n)`[1]`;` if `n` is an integer, produces a single random permutation of {1, 2, ... n}. If `n` is a list, we get a random permutation of the elements of `n`.

`solve(f,x);` solve the equation `f=0` for `x` symbolically.

`fsolve(f,x);` solve the equation `f=0` for `x` numerically.

`subs(x=a,f);` substitute `a` for `x` in the expression `f`

`eval(f,x=a);` evaluates the expression `f` at `x=a`

`sum(f,x=a..b);` sums values of `f` which correspond to `x=a, a+1, ..., b`

`taylor(y,x=a,n);` computes a truncated Taylor series (up to the `n`th degree term) of `f`, expanded about `x=a`.

Lists and Some Operations on Lists

`sort(A);` sorts the list `A` in non-decreasing order

`map(x->x^2, L);` produces a new list with entries equal to the squares of the entries from L. The first argument can be any mapping.

`X:=[i$i=1..3];` produces `X = [1,2,3]`

`Y:=[a, b, c];` makes `Y` the list of `a, b,` and `c`

`zip((x,y)->(x,y),X,Y);` produces `[1,a,2,b,3,c]`

`zip((x,y)->[x,y],X,Y);` produces `[[1,a],[2,b],[3,c]]`

`[seq(coeff((series(exp(x),x,9)),x,n),n=1..8)];` list of first eight coefficients in e^x series

Maple Supplement Commands: B.2 Descriptive Statistics

`Mean(L);` average of the elements of `L`

`StDev(L);` standard deviation of the elements of `L`

`Variance(L);` sample variance of the elements of `L`
`Max(L);` the maximum value of the list `L`
`Min(L);` the minimum value of the list `L`
`Range(L);` the range of values of the list `L`
`Percentile(L,p);` 100p-th percentile of the list `L`
`Median(L);` the median of the list `L`
`Skewness(L);` the skewness (third moment) of the list `L`
`Kurtosis(L);` the kurtosis (fourth moment) of the list `L`
`Freq(L, a..b);` the list of frequencies of `a, a+1, a+2, ..., b` in the list `L`
`ClassFreq(L,a..b,n);` frequencies of `n` equal-sized classes between `a` and `b`
`Locate(L, x);` the location of the first occurrence of `x` within the list `L`
`RunningSum(L);` running sums for the list `L`

Maple Supplement Commands: B.3 Random Samples

`RNG(n);` `n` random numbers from `[0,1)`.
`Die(m,n);` `n` rolls of an `m`-sided die.
`DiscreteS(PDF,n);` if X assumes values $k, \ldots, k+m$ with respective probabilities $p_k, \ldots, p_{k+m}$ and if `PDF` $:= [k, p_k, \ldots, k+m, p_{k+m}]$, then `DiscreteS(PDF,n)` will produce `n` random observations of X.
`ContinuousS(expr,a..b,n);` simulates `n` observations from the continuous distribution with p.d.f. `expr` on the interval `[a,b]`.
`BetaS(alpha, beta, n);` random sample (size `n`) from the beta distribution

`BernoulliS(p, n);` `BinomialS(N, p, n)`
`BivariateNormalS(muX,varX,muY,varY,rho,n);`
`CauchyS(n)` `ChisquareS(r, n);`
`DiscUniformS(a..b, n);` `ExponentialS(theta,n)`
`FS(nu1, nu2 ,n)` `GammaS(alpha, theta, n)`
`GeometricS(p, n)` `HypergeometricS(N1,N2,n,m)`
`LogisticS(n)` `NegBinomialS(r, p, n)`
`NormalS(mu,var,n);` `PoissonS(lambda, n)`
`TS(nu, n)` `UniformS(a..b, n)`

Maple Supplement Commands: B.4 Plotting

`ProbHist(expr, min..max);` probability histogram for discrete distributions with p.d.f given by `expr`
`ProbHist(PDF);` prob. hist. for discr. dist., `PDF`:=$[k, p_k, \ldots, k+m, p_{k+m}]$
`Histogram(X,xmin..xmax,NI);` hist for data list `X, NI` bins
`Histogram(X);` histogram for the list `X`, 8 bins

`PlotDiscCDF(expr, a..b);`	graph of the distribution function with p.d.f. `expr`.
`PlotEmpPDF(L, a..b, NI);`	graph of the empirical p.d.f.
`PlotEmpPDF(L);`	graph of the empirical p.d.f.
`PlotEmpCDF(L,xmin..xmax);`	graph of the empirical c.d.f.
`Ogive(L,xmin..xmax,NI);`	ogive for the data in `L`.
`PlotRunningAverage(L);`	running averages of the list `L`
`TimePlot(L1,L2,...,Lk);`	time series for the data in the lists `L1, L2, ..., Lk`
`StemLeaf(L, ND, NR);`	stem-and-leaf display of the data in the list `L`, `ND` leaf digits, `NR` rows
`BoxWhisker(L1,L2,...,Lk);`	box-and-whisker plot(s) for the list(s) `L1, L2, ..., Lk`
`QQ(X, Y);`	q-q plot of points with x and y coordinates in the lists `X` and `Y`
`QQ(L);`	q-q plot of points with x and y coordinates as sublists in the list `L`
`XbarChart(L, n, msd);`	$\overline{x}$ chart of the list of means, `L`, from samples of size `n`

Maple Supplement Commands: B.5 Regression

`L` may be replaced with `X,Y` as illustrated in `Correlation`

`Correlation(L);`	correlation coefficient of n points $[x_i, y_i]$, in the list of lists `L`
`Correlation(X, Y);`	correlation coefficient of n points in two lists, `X` and `Y`
`LinReg(L, x);`	returns the Y on X linear regression equation in the variable x
`PolyReg(L,deg,x);`	finds least squares polynomial of degree `deg` for data in list of lists `L`
`ScatPlot(L);`	produces a scatter plot of the n points in the list of lists `L`
`ScatPlotLine(L);`	produces a scatter plot of the n points in the list of lists `L` with the regression line superimposed
`PlotPolyReg(L,deg);`	produces a scatter plot of the n points in the list of lists `L` along with polynomial regression curve of degree `deg`
`Residuals(L, deg);`	displays and plots residuals calculated from degree `deg` polynomial regression
`RegBand(L,ConfLev,Type);`	for a list of lists `L`, plots a scatter plot, regression line, and a `ConfLev`% confidence (`C`) or prediction band (`P`)

Maple Supplement Commands: B.6 p.d.f.s

```
BernoulliPDF(p,x);          p.d.f. of Bernoulli dist.
BetaPDF(alpha, beta, x);    BinomialPDF(N, p, x);
CauchyPDF(x);               ChisquarePDF(r, x);
ExponentialPDF(theta, x);   FPDF(nu1, nu2, x);
GammaPDF(alpha,theta,x);    GeometricPDF(p, x);
LogisticPDF(x);             NegBinomialPDF(r,p,x);
NormalPDF(mu,var,x);        PoissonPDF(lambda,x);
TPDF(nu, x);                UniformPDF(a..b, x);
BivariateNormalPDF(muX,VarX,x,muY,VarY,y,rho);
DoubleExponentialPDF(alpha, theta, x);
HypergeometricPDF(N1, N2, n, x);
WeibullPDF(alpha, beta, theta, x);
```

Maple Supplement Commands: B.7 c.d.f.s

```
BernoulliCDF(p,x);        c.d.f. of Bernoulli dist.
BetaCDF(alpha, beta, x);  BinomialCDF(N, p, x);
ChisquareCDF(r, x);       ExponentialCDF(theta,x);
FCDF(nu1, nu2, x);        GammaCDF(alpha,theta,x);
GeometricCDF(p, x);       NegBinomialCDF(r, p, x);
NormalCDF(mu,var,x);      PoissonCDF(lambda, x);
TCDF(nu, x);              UniformCDF(a..b, x);
```

Maple Supplement Commands: B.8 pth percentile

```
BetaP(alpha, beta, p);    ChisquareP(r, p);
ExponentialP(theta, p);   FP(nu1, nu2, p);
GammaP(alpha, theta, p);  NormalP(mu, var, p);
TP(nu, p);                UniformP(a..b, p);
```

Maple Supplement Commands: B.9 Sampling Distributions

```
UniformMeanS(a..b,n,m);       m means of random samples
                              of size n from U(a,b).
UniformMedianS(a..b,n,m);     medians from uniform dist.
UniformSumS(a..b,n,m);        sums from uniform dist.
NormalMeanS(mu,var,n,m);      means from normal dist.
NormalMedianS(mu,var,n,m);    medians from normal dist.
NormalVarianceS(mu,var,n,m);  variances from normal dist.
NormalSumS(mu,var,n,m);       sums from normal dist.
NormTransVarS(mu,var,n,m);    (n-1)S^2/σ^2; from
                              normal dist.
ExponentialSumS(theta,n,m);   sums from exponential dist.
```

Maple Supplement Commands: B.10 Confidence Intervals

`ConfIntMean(ListOfSamples,CL);` produces `CL%` conf. intervals for mean of each sample
`ConfIntMean(ListOfSamples,ConfLev,var);` var known
`ConfIntVar(ListOfSamples,CL);` produces `CL%` conf. intervals for the variance
`ConfIntProp(ListOfSamples,ConfLev);` conf. int. for p
`ConfIntSuc(ListOfIntervals,v);` counts number of intervals that contain parameter `v`
`ConfIntAvLen(ListOfIntervals);` finds average length
`ConfIntPlot(ListOfIntervals);` plot of the intervals
`ConfIntPlot(ListOfIntervals,v);` adds vertical line at `v`

Maple Supplement Commands: B.11 ANOVA

`Anova1(L);` One-way ANOVA
`Anova2s(L);` Two-way ANOVA, single observation per cell
`Anova2m(L);` Two-way ANOVA, multiple observations per cell

Maple Supplement Commands: B.12 Goodness of Fit

`ChisquareFit(X,CDF,Classes);` Chi-square goodness of fit
`ChisquareFit(X,CDF,Classes,Disc);` Chi-square goodness of fit, discrete dist.
`KSFit(X,CDF,xmin..xmax);` Kolmogorov-Smirnov goodness of fit

Maple Supplement Commands: B.13 Nonparametric Tests

`SignTest(Bexpr,x=X,y=Y,...);` `Bexpr` is a boolean expression. Counts number of times list(s) satisfy it.
`Wilcoxon(L,m);` Wilcoxon statistic for testing median
`MedianTest(X,Y,k);` Median test
`Wilcoxon2(X,Y);` Two-sample Wilcoxon test
`RunTest(X,Y);` Counts runs in combined list
`RunTest(X,m);` Run test for randomness

Maple Supplement Commands: B.14 Miscellaneous

`Cards;` (list of 52 cards)
`Craps();`
`Contingency(L);`
`GraphRandWalk(pn,ps,pe,steps);`
`RandWalk2(pn,p3,steps,n);`
`GraphRandWalk2(pn,pe,steps);`
`MarginalRelFreq(A);`
`FastCraps(n);`
`RandWalk(pn,ps,pe,steps,n);`
`Convolution(X1,X2);`

[1] This command is part of the combinatorial package; to use it, you must first issue the `with(combinat):` command.