

Beginning Perl Programming (X52.9543.001)**An Implementation of Homework 3**

Here's **one** implementation of Homework #3. The **bold** items highlight changes/improvements from `monte1.pl`:

monte2.pl

```
#!/App/Perl/bin/perl -w
#
# monte2.pl
#
# Perl script to project returns from investments using Monte Carlo
# simulation over historical data.
#
# (c) Copyright 2000 by Clint Goss, All Rights Reserved.

$progVersion = 2;

# Pick up our options

use Getopt::Std;
use vars qw($opt_a $opt_d $opt_D $opt_h $opt_m $opt_y);
if (! getopts("a:dDhm:y:")) { &usage(); exit; }

sub usage {
    print <<END_OF_HELP;
usage: $0 [options]

Project the returns from investments using Monte Carlo simulation using
historical data.

Options:
  -a INTEGER    Starting amount of dollars for simulation. Default is \$10,000.
  -d            Turn on moderate debugging
  -D            Turn on very verbose debugging
  -h            Print this help message
  -m INTEGER    Number of Monte Carlo simulation to run. Default is 10.
  -y INTEGER    Number of years to run simulation. Default is 5.
END_OF_HELP
}

if ($opt_h) { &usage(); exit; }

# Parameters to this program

$debug = $opt_D ? 2 : $opt_d ? 1 : 0;      # Set level of debugging, default to 0
$yearsToCalculate = $opt_y ? $opt_y : 5;
$initialAssets = $opt_a ? $opt_a : 10_000;
$monteTrials = $opt_m ? $opt_m : 10;      # Number of Monte Carlo simulations
```

```

# Read data lines of the form: YEAR, RETURN, YEAR, RETURN, ...
# Build up the global @yearlyReturns array

$minYear = 3000;
$maxYear = 0;
while (<DATA>) {
    next if substr($_, 0, 1) eq "#";    # Ignore comment lines
    @data = split(",");
    for ($i = 0; $data[$i]; $i += 2) {

        ($year, $totalReturn) = @data[$i .. $i+1];
        print "Input year $year: return $totalReturn\n" if $debug;

        if ($year < $minYear) { $minYear = $year; }
        if ($year > $maxYear) { $maxYear = $year; }
        push @yearlyReturns, $totalReturn;
    }
}

# Remove \n from any and ALL elements of @yearlyReturns
chomp @yearlyReturns;

print ("Data items: ", scalar @yearlyReturns, "\n") if $debug;

# Loop to perform each requested Monte Carlo simulation.
foreach $monteTrial (1 .. $monteTrials) {

    # Starting with the initial amount of assets, we loop over the
    # requested number of years, modifying the assets by a randomly
    # picked return.

    $assets = &cumulativeAssetsMonte ($yearsToCalculate, $initialAssets);

    printf ("run #%d, assets=%0.2f\n", $monteTrial, $assets) if $debug;
    push @assetResults, $assets;
}

# Here when all runs completed. We sort the array so we can easily
# pull out the results at the appropriate percentile locations.

@assetResultsSorted = sort numerically @assetResults;

if ($debug) {
    print "Sorted Results:\n";
    foreach $i (0..$#assetResultsSorted) {
        printf " %0.2f\n", $assetResultsSorted[$i];
    }
}

# Select the interesting percentile results.

$result10 = &pickResult (10);
$result50 = &pickResult (50);
$result90 = &pickResult (90);

```

```

# Print results

printf <<END_OF_HEADER,
Monte Carlo simulation (version %d) of investment returns for
large company stock using historical data from %d to %d.

Initial assets: \$.%0.2f

Using %d Monte Carlo simulations, each running for %d years,
the median ending assets were \$.%0.2f.
The ending assets were between \$.%0.2f and \$.%0.2f on 80%% of the runs.
END_OF_HEADER

    $progVersion, $minYear, $maxYear,
    $initialAssets, $monteTrials, $yearsToCalculate,
    $result50, $result10, $result90;

# End of main program
exit;

# Run a single Monte Carlo simulation for a given number of years,
# against a given starting asset amount.
# Returns the ending asset amount for the single run.
# NOTE: This routine REQUIRES the global array @yearlyReturns to contain a
# set of returns as percentages where 100 means a yearly return of 100%.

sub cumulativeAssetsMonte {
    my ($years, $startingAssets) = @_;

    my ($assets) = $startingAssets;

    foreach $year (1 .. $years) {

        # Randomly select a year from our data to use as a rate of
        # return for this year.

        $currentYearReturn = $yearlyReturns[
            int (rand (scalar @yearlyReturns))];
        $assets *= 1 + $currentYearReturn / 100;

        if ($debug >= 2) {
            printf "year %0.2f, return=%0.2f, assets=%0.2f\n",
                $year, $currentYearReturn, $assets;
        }
    }
    return $assets;
}

# Service routine for sort function.

sub numerically { $a <=> $b; }

```

```
# Pick an array element from the global array @assetResultsSorted which is at
# the given percentile in the list.
# For example, if the array has 200 elements, and the requested percentile
# is 12, this routine will return $assetResultsSorted [24];
```

```
sub pickResult {
    my ($percentile) = @_ ;

    $numResults = scalar @assetResultsSorted;

    $index = int (($numResults * $percentile)/ 100);
    printf ("index%d=%d\n", $percentile, $index) if $debug;
    return $assetResultsSorted[$index];
}
```

__END__

```
# Yearly total return of large company stocks
# Source: Ibbotson SBBI Yearbook
# Typed in by Clint Goss - data has not been verified!
1926, -11.62, 1927, +37.49, 1928, +43.61, 1929, -08.42, 1930, -24.90
1931, -43.34, 1932, -08.19, 1933, +53.99, 1934, -01.44, 1935, +47.67
1936, +33.92, 1937, -35.03, 1938, +31.12, 1939, -00.41, 1940, -09.78
1941, -11.59, 1942, +20.34, 1943, +25.90, 1944, +19.75, 1945, +36.44
1946, -08.07, 1947, +05.71, 1948, +05.50, 1949, +18.79, 1950, +31.71
1951, +24.02, 1952, +18.37, 1953, -00.99, 1954, +52.62, 1955, +31.56
1956, +06.56, 1957, -10.78, 1958, +43.36, 1959, +11.96, 1960, +00.47
1961, +26.89, 1962, -08.73, 1963, +22.80, 1964, +16.48, 1965, +12.45
1966, -10.06, 1967, +23.98, 1968, +11.06, 1969, -08.50, 1970, +04.01
1971, +14.31, 1972, +18.98, 1973, -14.66, 1974, -26.47, 1975, +37.20
1976, +23.84, 1977, -07.18, 1978, +06.56, 1979, +18.44, 1980, +32.42
1981, -04.91, 1982, +21.41, 1983, +22.51, 1984, +06.27, 1985, +32.16
1986, +18.47, 1987, +05.23, 1988, +16.81, 1989, +31.49, 1990, -03.17
1991, +30.55, 1992, +07.67, 1993, +09.99, 1994, +01.31, 1995, +37.43
1996, +23.07
```