Recitation 3 DA5030 Fall 2020

Martin Schedlbauer

Random Numbers, Sampling

The series of values generated by such algorithms is generally determined by a fixed number called a *seed*.

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
##  [1] 15  4 13  8  9 14 19  6 18  2 32 12 21  7 26 27 31 10  5 28 16 20 22 25  3
```

```
##                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
```

```
##                    mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Merc 280C         17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## AMC Javelin       15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28        13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Ford Pantera L    15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino      19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
```

```
## [1]  4 13 18 22 12  7
```

```
##                  mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Hornet 4 Drive   21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Merc 450SL       17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Fiat 128         32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Dodge Challenger 15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## Merc 450SE       16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Duster 360       14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
```

Random Number Generating Functions

One of the most common PRNG is the linear congruential generator, which uses the recurrence

$X_{n+1} = (aX_n + b) \mod m$

to generate numbers, where a, b and m are large integers, and X_{n+1} is the next in X as a series of *pseudo-random* numbers.

Loops, Iteration

for loops

```r
# Create a vector filled with random normal values
u1 <- rnorm(30)
print("This loop calculates the square of the first 10 elements of vector u1")
```

```
## [1] "This loop calculates the square of the first 10 elements of vector u1"
```

```
# Initialize ‘usq‘
usq <- 0

for(i in 1:10) {
  # i-th element of ‘u1‘ squared into ‘i‘-th position of ‘usq‘
  usq[i] <- u1[i]*u1[i]
  print(usq[i])
}
```

```
## [1] 4.890346
## [1] 0.3842461
## [1] 1.669929
## [1] 0.4353283
## [1] 1.535001
## [1] 0.3475597
## [1] 2.096267
## [1] 0.6040312
## [1] 0.06150101
## [1] 0.1815778
```

```
print(i)
```

```
## [1] 10
```

Games with Loops

Usual loops: for, while, “do” repeat

```
set.seed(4432)
v1 <- c(3,7,1,9,5,3,7,2)
v2 <- floor(runif(8,min=1,max=100))

v1[3]
```

```
## [1] 1
```

```
v2
```

```
## [1] 87 70 85 33  7 50 13 64
```

```
v3 <- 1:8

n <- min(length(v1),length(v2))

for (i in 1:n)
{
  v3[i] <- v1[i] / v2[i]
}

v3
```

```
## [1] 0.03448276 0.10000000 0.01176471 0.27272727 0.71428571 0.06000000 0.53846154
## [8] 0.03125000
```

```
v3<- v1 / v2
v3
```

```
## [1] 0.03448276 0.10000000 0.01176471 0.27272727 0.71428571 0.06000000 0.53846154
## [8] 0.03125000
```

Time Series Analysis

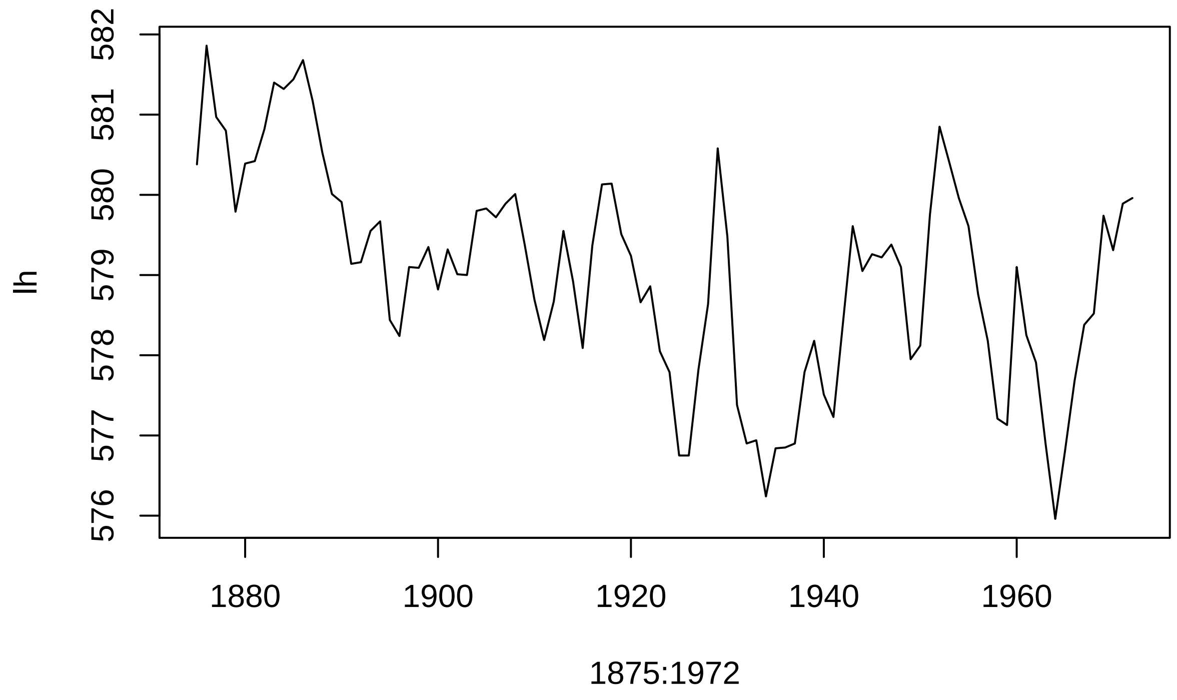
```
lh <- LakeHuron
head(lh)
```

```
## [1] 580.38 581.86 580.97 580.80 579.79 580.39
```

```
str(lh)
```

```
##  Time-Series [1:98] from 1875 to 1972: 580 582 581 581 580 ...
```

```
plot(x=1875:1972,y=lh,type="l")
```



```r
n <- length(lh)

# three-year simple moving average
t <- (lh[n] + lh[n-1] + lh[n-2])/3

lh[n]
```

```
## [1] 579.96
```

```r
t1972_98 <- (lh[97]+lh[96]+lh[95])/3
t1972_98
```

```
## [1] 579.6467
```

```r
e <- t1972_98 - lh[n]
e
```

```
## [1] -0.3133333
```

```r
p <- 1:n
e <- 1:n
mad <- 1:n

x <- 10

for (k in 1:x)
{
  for (i in (k+1):n)
  {
    # p[i] <- (lh[i-1]+lh[i-2]+lh[i-3]) / k
    p[i] <- mean(lh[(i-1):(i-k)])
    e[i] <- abs(p[i] - lh[i])
  }
  mad[k] <- mean(e)
}

min_mad <- min(mad)
best_k <- which(mad == min_mad)

best_k
```

```
## [1] 1
```

```r
v <- c(7,4,2,5,6,1,7,4,2,9)

a <- 1
b <- 3
mean(v[a:b])
```

```
## [1] 4.333333
```

Load data: CSV (Comma Separated Values)

```r
usedcars <- read.csv("usedcars.csv", header = TRUE, stringsAsFactors = TRUE)
head(usedcars, 4)
```

```
##   year model price mileage  color transmission
## 1 2011   SEL 21992    7413 Yellow         AUTO
## 2 2011   SEL 20995   10926   Gray         AUTO
## 3 2011   SEL 19995    7351 Silver         AUTO
## 4 2011   SEL 17809   11613   Gray         AUTO
```

```r
str(usedcars)
```

```
## 'data.frame':    150 obs. of  6 variables:
##  $ year        : int  2011 2011 2011 2011 2012 2010 2011 2010 2011 2010 ...
##  $ model       : Factor w/ 3 levels "SE","SEL","SES": 2 2 2 2 1 2 2 2 3 3 ...
##  $ price       : int  21992 20995 19995 17809 17500 17495 17000 16995 16995 16995 ...
##  $ mileage     : int  7413 10926 7351 11613 8367 25125 27393 21026 32655 36116 ...
##  $ color       : Factor w/ 9 levels "Black","Blue",..: 9 4 7 4 8 7 2 7 7 7 ...
##  $ transmission: Factor w/ 2 levels "AUTO","MANUAL": 1 1 1 1 1 1 1 1 1 1 ...
```