

# R Scripts

## Statistics 427: R Programming

### Module 2

2020

## R Scripts

Even small calculation tasks can be frustrating to perform when they are typed line by line in the console. If you make a mistake, then you have to retype everything again and grrr!

R has a lovely feature that allows for easier managing of long lists of commands. The feature is called an R script. It is a text editor type window that allows you to type all the commands and run them from that script. The R script can be typed, edited, saved as a file, processed in part or in whole, changed, reprocessed, and re-saved.

## Creating and Saving an R Script

Start R. Use the mouse cursor up to **FILE** for the pull-down menu, select **New Script**. A new window, in addition to the console window, will appear for your use.

**Note:** *The R editor only exists in Mac and Windows versions of R. If you are using Linux or Unix, there is no R editor so you could use a Notepad equivalent or download a free editor program.*

## Room with a Moose

Now the moose data can be typed into an R script and saved for future uses.

.  
.

. Yes, now go type it in an R script file :-)

## Saving the moose

You can save the R script as .txt, .doc (if you have Word installed), etc. If you do not choose a filetype, the default filetype to save as is a .R file, which will open nicely in R or can be opened with Notepad, Word, etc.

## Running an R Script

You can run the code from the editor window without having to copy and paste the code into the console. You can run one line a time or you can run the whole script. With the editor window active, click **Edit** on the task bar. Find and click **Run all**.

## Back to the meese... mooses?

With the moose data and plot recreated, we can add the model equation to it in the script. What are you waiting for? :-) Add the following to the first moose data script.

```
m=2.5*(0:100)/100  
a=3.37; b=0.47
```

```
k=(a*m)/(b+m)
points(m,k,type='l')
```

## Finding Errors in an R Script

Errors are the bane of a programmer's existence, well the ones hard to find and debug are. But sometimes with the length of scripts, there are bound to be errors and they must be found and fixed. At this point, we will go through some early debugging methods, and later in the course we will cover debugging and cleaning more extensively.

### Most common errors

- (1) Misspelling: incorrect spellings, case errors, command name fails, etc.
- (2) Objects not stored in R (session ended, objects removed, etc.)
- (3) Lack of mathematical operators in appropriate places
- (4) Data types (vector types: numeric or character) are invalid for commands
- (5) More... because we are humans

### Comments in Code

The first module covered comments but anytime you want to make a comment in R, use a # sign before the comment. R will print the comment in the console when running the program but will not produce an error for trying to run the comment.

You can also use commenting for making scripts a bit more informational. You can comment on the datafile,... anything.

### Adding comments to the moose

```
#####
# Author: me
# Date: 8/11/2020
# Moose data: moose kill rate by wolves
# (avg number of moose killed per wolf per 100 days)
# Model equation included
#
# Data source: Messier, F. 1994.
# Title: Ungulate population models with predation:
# a case study with the North American moose
# Journal: Ecology 75: 478-488
#####
```

### Finance example I

Say you are ready to buy a mobile home for \$30,000 and you know you will qualify for a mortgage but want to now how large the monthly payments will be (yes you could ask a banker but this is more fun!). Let's say you have a down payment of \$6000, and that there will be extra fees (of course there are extra fees!) of around \$1000, and so the total amount of money you need to borrow is \$25,000 (the principal). The loan provider agrees to a 5% annual interest rate on any unpaid balance of the principal. We have to account for *monthly* payments based on an *annual* rate, the loan will be for 30 years (360 months).

## Finance example II

Derivations of the following equations can be found online or in the recommended book on page 30.

Equations:

$$Mp(t) = \frac{(1+r)^t r P_0}{(1+r)^t - 1}$$

$$P(m) = \frac{(1+r)^{m-1} r P_0}{(1+r)^t - 1}$$

$$Pr(m) = P_0 \left[ 1 - \frac{(1+r)^m - 1}{(1+r)^t - 1} \right]$$

### Steps I

- (1) Define and assign values for variables
  - $P_0$ : principal amount
  - $t$ : total number of monthly payments
  - $m$ : vector of  $t$ ,  $m = 1 : t$
  - $r$ : the interest rate, by appropriate time period
- (2) Calculate the vector of  $m$
- (3) Calculate the monthly payment,  $Mp(t)$

### Steps II

- (4) Calculate a vector of principal amounts of the loan paid each month using  $P(m)$
- (5) Calculate a vector of principal amounts of the loan remaining unpaid each month using  $Pr(m)$
- (6) Calculate a vector of the interest amounts paid each period (month) by subtracting the principal amounts paid from the monthly payment
- (7) Calculate the total interest paid by summing all the interest amounts paid each month. Use the `sum()` command to sum the entire vector
- (8) Print results (to console, not printer)
- (9) Graph results with principal remaining versus the vector of months

### The R script

```
# defining variables
p0 <- 25000
r = .05/12 # yearly rate adj for monthly payments
t = 12*30 # total mpnthly payments over 30 yrs
m = 1:t

# calculate mp(t)
mpt = (r*p0*(1+r)^t)/((1+r)^t-1)

# calculate principle paid
pm = ((1+r)^(m-1)*r*p0)/((1+r)^t-1)

# calculate unpaid principle
prm = p0*(1-((1+r)^m-1)/((1+r)^t-1))
```

## The R script II

```
# calculate interest paid
ip=mpt-pm

# total interest
tint=sum(ip)

# print
mpt
```

```
[1] 134.2054
```

```
pm
```

```
[1] 30.03874 30.16390 30.28958 30.41579 30.54252 30.66978 30.79757
[8] 30.92590 31.05475 31.18415 31.31408 31.44456 31.57558 31.70714
[15] 31.83926 31.97192 32.10514 32.23891 32.37324 32.50812 32.64358
[22] 32.77959 32.91617 33.05332 33.19104 33.32934 33.46821 33.60766
[29] 33.74770 33.88831 34.02951 34.17130 34.31368 34.45666 34.60023
[36] 34.74439 34.88916 35.03453 35.18051 35.32710 35.47429 35.62210
[43] 35.77053 35.91957 36.06924 36.21952 36.37044 36.52198 36.67416
[50] 36.82697 36.98041 37.13450 37.28922 37.44460 37.60061 37.75728
[57] 37.91461 38.07258 38.23122 38.39052 38.55048 38.71110 38.87240
[64] 39.03437 39.19701 39.36033 39.52433 39.68902 39.85439 40.02045
[71] 40.18720 40.35465 40.52279 40.69164 40.86119 41.03144 41.20240
[78] 41.37408 41.54647 41.71958 41.89342 42.06797 42.24325 42.41927
[85] 42.59601 42.77350 42.95172 43.13069 43.31040 43.49086 43.67207
[92] 43.85404 44.03676 44.22025 44.40450 44.58952 44.77531 44.96187
[99] 45.14921 45.33733 45.52624 45.71593 45.90642 46.09769 46.28977
[106] 46.48264 46.67632 46.87080 47.06610 47.26221 47.45913 47.65688
[113] 47.85545 48.05485 48.25508 48.45614 48.65804 48.86078 49.06437
[120] 49.26880 49.47409 49.68023 49.88723 50.09510 50.30382 50.51342
[127] 50.72390 50.93525 51.14748 51.36059 51.57459 51.78949 52.00528
[134] 52.22197 52.43956 52.65806 52.87746 53.09779 53.31903 53.54119
[141] 53.76428 53.98830 54.21325 54.43914 54.66597 54.89374 55.12246
[148] 55.35214 55.58278 55.81437 56.04693 56.28046 56.51496 56.75044
[155] 56.98690 57.22435 57.46278 57.70221 57.94263 58.18406 58.42650
[162] 58.66994 58.91440 59.15987 59.40637 59.65390 59.90246 60.15205
[169] 60.40268 60.65436 60.90709 61.16087 61.41571 61.67160 61.92857
[176] 62.18661 62.44572 62.70591 62.96718 63.22954 63.49300 63.75756
[183] 64.02321 64.28997 64.55785 64.82684 65.09695 65.36819 65.64056
[190] 65.91406 66.18870 66.46449 66.74142 67.01951 67.29876 67.57917
[197] 67.86075 68.14351 68.42744 68.71255 68.99885 69.28635 69.57504
[204] 69.86494 70.15604 70.44836 70.74189 71.03665 71.33264 71.62986
[211] 71.92831 72.22802 72.52897 72.83117 73.13463 73.43936 73.74536
[218] 74.05263 74.36118 74.67102 74.98215 75.29458 75.60830 75.92334
[225] 76.23968 76.55735 76.87634 77.19666 77.51831 77.84130 78.16564
[232] 78.49133 78.81838 79.14679 79.47657 79.80772 80.14025 80.47417
[239] 80.80948 81.14618 81.48429 81.82381 82.16474 82.50710 82.85088
[246] 83.19609 83.54274 83.89083 84.24038 84.59138 84.94384 85.29778
[253] 85.65318 86.01007 86.36845 86.72832 87.08968 87.45256 87.81694
[260] 88.18285 88.55028 88.91924 89.28973 89.66177 90.03536 90.41051
[267] 90.78722 91.16550 91.54536 91.92680 92.30983 92.69445 93.08068
[274] 93.46851 93.85797 94.24904 94.64174 95.03609 95.43207 95.82970
```

```

[281] 96.22899 96.62995 97.03257 97.43687 97.84286 98.25054 98.65992
[288] 99.07100 99.48380 99.89831 100.31455 100.73253 101.15225 101.57372
[295] 101.99694 102.42193 102.84869 103.27722 103.70755 104.13966 104.57358
[302] 105.00930 105.44684 105.88620 106.32739 106.77042 107.21530 107.66203
[309] 108.11062 108.56108 109.01342 109.46764 109.92376 110.38177 110.84170
[316] 111.30354 111.76730 112.23300 112.70064 113.17022 113.64177 114.11527
[323] 114.59075 115.06822 115.54767 116.02911 116.51257 116.99804 117.48553
[330] 117.97505 118.46662 118.96023 119.45589 119.95363 120.45343 120.95532
[337] 121.45930 121.96538 122.47357 122.98388 123.49631 124.01088 124.52759
[344] 125.04646 125.56748 126.09068 126.61606 127.14363 127.67339 128.20537
[351] 128.73955 129.27597 129.81462 130.35551 130.89866 131.44407 131.99176
[358] 132.54172 133.09398 133.64854

```

## The R script III

```
prm
```

```

[1] 24969.9613 24939.7974 24909.5078 24879.0920 24848.5495 24817.8797
[7] 24787.0821 24756.1562 24725.1015 24693.9173 24662.6032 24631.1587
[13] 24599.5831 24567.8759 24536.0367 24504.0648 24471.9596 24439.7207
[19] 24407.3475 24374.8394 24342.1958 24309.4162 24276.5000 24243.4467
[25] 24210.2557 24176.9263 24143.4581 24109.8504 24076.1027 24042.2144
[31] 24008.1849 23974.0136 23939.6999 23905.2433 23870.6431 23835.8987
[37] 23801.0095 23765.9750 23730.7945 23695.4674 23659.9931 23624.3710
[43] 23588.6004 23552.6809 23516.6116 23480.3921 23444.0217 23407.4997
[49] 23370.8255 23333.9986 23297.0182 23259.8837 23222.5944 23185.1498
[55] 23147.5492 23109.7919 23071.8773 23033.8048 22995.5735 22957.1830
[61] 22918.6325 22879.9214 22841.0490 22802.0147 22762.8177 22723.4573
[67] 22683.9330 22644.2440 22604.3896 22564.3691 22524.1819 22483.8273
[73] 22443.3045 22402.6129 22361.7517 22320.7202 22279.5178 22238.1438
[79] 22196.5973 22154.8777 22112.9843 22070.9163 22028.6731 21986.2538
[85] 21943.6578 21900.8843 21857.9325 21814.8019 21771.4915 21728.0006
[91] 21684.3285 21640.4745 21596.4377 21552.2175 21507.8130 21463.2235
[97] 21418.4482 21373.4863 21328.3371 21282.9997 21237.4735 21191.7576
[103] 21145.8512 21099.7535 21053.4637 21006.9811 20960.3047 20913.4339
[109] 20866.3678 20819.1056 20771.6465 20723.9896 20676.1342 20628.0793
[115] 20579.8242 20531.3681 20482.7101 20433.8493 20384.7849 20335.5161
[121] 20286.0420 20236.3618 20186.4746 20136.3795 20086.0757 20035.5622
[127] 19984.8383 19933.9031 19882.7556 19831.3950 19779.8204 19728.0309
[133] 19676.0257 19623.8037 19571.3641 19518.7061 19465.8286 19412.7308
[139] 19359.4118 19305.8706 19252.1063 19198.1180 19143.9048 19089.4657
[145] 19034.7997 18979.9060 18924.7835 18869.4313 18813.8486 18758.0342
[151] 18701.9873 18645.7068 18589.1919 18532.4414 18475.4545 18418.2302
[157] 18360.7674 18303.0652 18245.1225 18186.9385 18128.5120 18069.8421
[163] 18010.9277 17951.7678 17892.3614 17832.7075 17772.8050 17712.6530
[169] 17652.2503 17591.5959 17530.6889 17469.5280 17408.1123 17346.4407
[175] 17284.5121 17222.3255 17159.8798 17097.1739 17034.2067 16970.9772
[181] 16907.4842 16843.7266 16779.7034 16715.4134 16650.8556 16586.0287
[187] 16520.9318 16455.5636 16389.9230 16324.0090 16257.8203 16191.3558
[193] 16124.6144 16057.5948 15990.2961 15922.7169 15854.8562 15786.7126
[199] 15718.2852 15649.5727 15580.5738 15511.2875 15441.7124 15371.8475
[205] 15301.6914 15231.2431 15160.5012 15089.4645 15018.1319 14946.5020
[211] 14874.5737 14802.3457 14729.8168 14656.9856 14583.8509 14510.4116

```

```

[217] 14436.6662 14362.6136 14288.2524 14213.5814 14138.5992 14063.3047
[223] 13987.6964 13911.7730 13835.5333 13758.9760 13682.0997 13604.9030
[229] 13527.3847 13449.5434 13371.3777 13292.8864 13214.0680 13134.9212
[235] 13055.4447 12975.6370 12895.4967 12815.0225 12734.2131 12653.0669
[241] 12571.5826 12489.7588 12407.5940 12325.0869 12242.2361 12159.0400
[247] 12075.4972 11991.6064 11907.3660 11822.7746 11737.8308 11652.5330
[253] 11566.8798 11480.8697 11394.5013 11307.7730 11220.6833 11133.2307
[259] 11045.4138 10957.2309 10868.6807 10779.7614 10690.4717 10600.8099
[265] 10510.7746 10420.3641 10329.5768 10238.4113 10146.8660 10054.9392
[271] 9962.6293 9869.9349 9776.8542 9683.3857 9589.5277 9495.2787
[277] 9400.6370 9305.6009 9210.1688 9114.3391 9018.1101 8921.4802
[283] 8824.4476 8727.0107 8629.1679 8530.9173 8432.2574 8333.1864
[289] 8233.7026 8133.8043 8033.4897 7932.7572 7831.6050 7730.0312
[295] 7628.0343 7525.6124 7422.7637 7319.4864 7215.7789 7111.6392
[301] 7007.0657 6902.0564 6796.6095 6690.7233 6584.3959 6477.6255
[307] 6370.4102 6262.7482 6154.6376 6046.0765 5937.0631 5827.5954
[313] 5717.6717 5607.2899 5496.4482 5385.1447 5273.3773 5161.1443
[319] 5048.4437 4935.2735 4821.6317 4707.5164 4592.9257 4477.8575
[325] 4362.3098 4246.2807 4129.7681 4012.7701 3895.2846 3777.3095
[331] 3658.8429 3539.8827 3420.4268 3300.4731 3180.0197 3059.0644
[337] 2937.6051 2815.6397 2693.1661 2570.1822 2446.6859 2322.6750
[343] 2198.1475 2073.1010 1947.5335 1821.4428 1694.8268 1567.6831
[349] 1440.0097 1311.8044 1183.0648 1053.7889 923.9742 793.6187
[355] 662.7201 531.2760 399.2842 266.7425 133.6485 0.0000

```

## The R script IV

```
ip
```

```

[1] 104.1666667 104.0415053 103.9158223 103.7896157 103.6628833 103.5356228
[7] 103.4078320 103.2795088 103.1506509 103.0212561 102.8913221 102.7608468
[13] 102.6298278 102.4982629 102.3661498 102.2334862 102.1002699 101.9664985
[19] 101.8321697 101.6972812 101.5618307 101.4258158 101.2892342 101.1520834
[25] 101.0143613 100.8760652 100.7371930 100.5977421 100.4577102 100.3170948
[31] 100.1758935 100.0341039 99.8917234 99.7487498 99.6051803 99.4610127
[37] 99.3162444 99.1708729 99.0248957 98.8783103 98.7311140 98.5833045
[43] 98.4348791 98.2858352 98.1361703 97.9858818 97.8349672 97.6834237
[49] 97.5312487 97.3784397 97.2249941 97.0709090 96.9161819 96.7608102
[55] 96.6047910 96.4481218 96.2907998 96.1328223 95.9741865 95.8148897
[61] 95.6549293 95.4943023 95.3330060 95.1710377 95.0083945 94.8450736
[67] 94.6810722 94.5163875 94.3510166 94.1849566 94.0182047 93.8507581
[73] 93.6826137 93.5137687 93.3442203 93.1739653 93.0030010 92.8313243
[79] 92.6589323 92.4858220 92.3119904 92.1374345 91.9621513 91.7861377
[85] 91.6093908 91.4319074 91.2536845 91.0747190 90.8950078 90.7145478
[91] 90.5333359 90.3513689 90.1686437 89.9851572 89.8009062 89.6158875
[97] 89.4300978 89.2435340 89.0561929 88.8680712 88.6791656 88.4894729
[103] 88.2989899 88.1077131 87.9156394 87.7227654 87.5290877 87.3346031
[109] 87.1393081 86.9431993 86.7462735 86.5485271 86.3499568 86.1505590
[115] 85.9503305 85.7492677 85.5473671 85.3446253 85.1410387 84.9366039
[121] 84.7313172 84.5251751 84.3181742 84.1103107 83.9015812 83.6919819
[127] 83.4815093 83.2701597 83.0579295 82.8448150 82.6308126 82.4159184
[133] 82.2001289 81.9834403 81.7658487 81.5473506 81.3279420 81.1076193

```

[139]	80.8863785	80.6642159	80.4411276	80.2171097	79.9921585	79.7662700
[145]	79.5394402	79.3116654	79.0829415	78.8532645	78.6226306	78.3910357
[151]	78.1584758	77.9249470	77.6904451	77.4549661	77.2185059	76.9810605
[157]	76.7426257	76.5031975	76.2627716	76.0213439	75.7789104	75.5354666
[163]	75.2910085	75.0455319	74.7990324	74.5515059	74.3029479	74.0533544
[169]	73.8027208	73.5510430	73.2983165	73.0445369	72.7897000	72.5338012
[175]	72.2768362	72.0188005	71.7596896	71.4994991	71.2382245	70.9758613
[181]	70.7124048	70.4478507	70.1821942	69.9154308	69.6475559	69.3785649
[187]	69.1084530	68.8372157	68.5648483	68.2913459	68.0167040	67.7409178
[193]	67.4639824	67.1858931	66.9066452	66.6262337	66.3446538	66.0619006
[199]	65.7779694	65.4928550	65.2065528	64.9190575	64.6303644	64.3404684
[205]	64.0493645	63.7570477	63.4635128	63.1687550	62.8727689	62.5755496
[211]	62.2770919	61.9773905	61.6764405	61.3742365	61.0707733	60.7660456
[217]	60.4600483	60.1527760	59.8442233	59.5343851	59.2232558	58.9108302
[223]	58.5971028	58.2820682	57.9657210	57.6480556	57.3290667	57.0087486
[229]	56.6870958	56.3641029	56.0397641	55.7140739	55.3870267	55.0586168
[235]	54.7288385	54.3976862	54.0651540	53.7312363	53.3959272	53.0592211
[241]	52.7211120	52.3815941	52.0406615	51.6983084	51.3545289	51.0093169
[247]	50.6626665	50.3145718	49.9650266	49.6140250	49.2615610	48.9076283
[253]	48.5522209	48.1953326	47.8369573	47.4770888	47.1157208	46.7528471
[259]	46.3884614	46.0225575	45.6551289	45.2861695	44.9156726	44.5436321
[265]	44.1700414	43.7948940	43.4181835	43.0399035	42.6600472	42.2786082
[271]	41.8955799	41.5109556	41.1247287	40.7368926	40.3474404	39.9563656
[277]	39.5636612	39.1693206	38.7733370	38.3757033	37.9764129	37.5754588
[283]	37.1728340	36.7685316	36.3625446	35.9548660	35.5454888	35.1344058
[289]	34.7216100	34.3070942	33.8908512	33.4728739	33.0531550	32.6316873
[295]	32.2084635	31.7834762	31.3567182	30.9281820	30.4978602	30.0657454
[301]	29.6318302	29.1961069	28.7585682	28.3192064	27.8780139	27.4349831
[307]	26.9901063	26.5433759	26.0947841	25.6443232	25.1919854	24.7377628
[313]	24.2816476	23.8236319	23.3637079	22.9018675	22.4381027	21.9724056
[319]	21.5047681	21.0351821	20.5636395	20.0901322	19.6146519	19.1371904
[325]	18.6577395	18.1762909	17.6928362	17.2073672	16.7198754	16.2303523
[331]	15.7387896	15.2451787	14.7495111	14.2517782	13.7519714	13.2500821
[337]	12.7461016	12.2400212	11.7318321	11.2215255	10.7090927	10.1945247
[343]	9.6778127	9.1589477	8.6379208	8.1147230	7.5893451	7.0617782
[349]	6.5320131	6.0000406	5.4658516	4.9294368	4.3907869	3.8498927
[355]	3.3067447	2.7613336	2.2136500	1.6636843	1.1114271	0.5568689

## The R script V

```
tint
```

```
[1] 23313.95
```

```
# plot!
plot(m,prm,type='l')
```

