Assignment 1

Problem 1

Calculate the remainder after dividing 31079 into 170166719.

```
# use the modulo operator %% to calculate the remainder
remainder <- 170166719 %% 31079
remainder
```

```
## [1] 9194
```

Problem 2

Calculate the respective areas of circles having radii 3, 4, . . . , 100.

```
# area of circle is radius sqaured times pi
areas <- pi * (3:100)^2
# output will have 97 values since the sequence 3:100 consists of 97 values
areas
```

```
##  [1]    28.27433    50.26548    78.53982   113.09734   153.93804   201.06193
##  [7]   254.46900   314.15927   380.13271   452.38934   530.92916   615.75216
## [13]   706.85835   804.24772   907.92028  1017.87602  1134.11495  1256.63706
## [19]  1385.44236  1520.53084  1661.90251  1809.55737  1963.49541  2123.71663
## [25]  2290.22104  2463.00864  2642.07942  2827.43339  3019.07054  3216.99088
## [31]  3421.19440  3631.68111  3848.45100  4071.50408  4300.84034  4536.45979
## [37]  4778.36243  5026.54825  5281.01725  5541.76944  5808.80482  6082.12338
## [43]  6361.72512  6647.61005  6939.77817  7238.22947  7542.96396  7853.98163
## [49]  8171.28249  8494.86654  8824.73376  9160.88418  9503.31778  9852.03456
## [55] 10207.03453 10568.31769 10935.88403 11309.73355 11689.86626 12076.28216
## [61] 12468.98124 12867.96351 13273.22896 13684.77760 14102.60942 14526.72443
## [67] 14957.12262 15393.80400 15836.76857 16286.01632 16741.54725 17203.36137
## [73] 17671.45868 18145.83917 18626.50284 19113.44970 19606.67975 20106.19298
## [79] 20611.98940 21124.06900 21642.43179 22167.07776 22698.00692 23235.21927
## [85] 23778.71480 24328.49351 24884.55541 25446.90049 26015.52876 26590.44022
## [91] 27171.63486 27759.11269 28352.87370 28952.91790 29559.24528 30171.85585
## [97] 30790.74960 31415.92654
```

Problem 3

Calculate the interest earned on an investment of \$2000, assuming an interest rate of 3% compounded annually, for terms of 1, 2, . . . , 30 years.

```
net_worth <- 2000 * (1.03^(1:30))
# but what we're really interested is the "interest earned", so the profit
interest <- net_worth - 2000
interest
```

```
##  [1]   60.0000  121.8000  185.4540  251.0176  318.5481  388.1046  459.7477
##  [8]  533.5402  609.5464  687.8328  768.4677  851.5218  937.0674 1025.1794
## [15] 1115.9348 1209.4129 1305.6953 1404.8661 1507.0121 1612.2225 1720.5891
## [22] 1832.2068 1947.1730 2065.5882 2187.5559 2313.1825 2442.5780 2575.8554
## [29] 2713.1310 2854.5249
```

Problem 4

Using the rep() and seq() functions as needed, create the following vectors: { 0 0 0 0 0 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 } { 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 } { 1 2 3 4 5 2 3 4 5 6 3 4 5 6 7 4 5 6 7 8 5 6 7 8 9 }

```
# the default setting of the rep() function is as follows: rep(x, times = 1, length.out = NA, each = 1)
# for this problem we just need to manipulate the "times" and "each" argument
x <- rep(0:4, each = 5)
x
```

```
##  [1] 0 0 0 0 0 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4
```

```
y <- rep(1:5, times = 5)
y
```

```
##  [1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

```
z <- rep(1:5, times = 5) + rep(0:4, each = 5) # no need to worry about recycling since total # of value
z
```

```
##  [1] 1 2 3 4 5 2 3 4 5 6 3 4 5 6 7 4 5 6 7 8 5 6 7 8 9
```

Problem 5

Find the value of r^1 + r^2 + … + r^n, for all values of n between 1 and 100, when r = 1.08.

```
r <- 1.08 # the default r value set by the problem

exponents <- 1:100 # this is a sequence of values from 1 to 100

# sum of a geometric series uses the formula S = a * (1 - r^n) / (1 - r)
sum_values <- (r * (1 - r^exponents)) / (1 - r)
sum_values
```

```
##   [1]     1.080000     2.246400     3.506112     4.866601     6.335929
##   [6]     7.922803     9.636628    11.487558    13.486562    15.645487
##  [11]    17.977126    20.495297    23.214920    26.152114    29.324283
```

```
## [16]    32.750226    36.450244    40.446263    44.761964    49.422921
## [21]    54.456755    59.893296    65.764759    72.105940    78.954415
## [26]    86.350768    94.338830   102.965936   112.283211   122.345868
## [31]   133.213537   144.950620   157.626670   171.316804   186.102148
## [36]   202.070320   219.315945   237.941221   258.056519   279.781040
## [41]   303.243523   328.583005   355.949646   385.505617   417.426067
## [46]   451.900152   489.132164   529.342737   572.770156   619.671769
## [51]   670.325510   725.031551   784.114075   847.923201   916.837058
## [56]   991.264022  1071.645144  1158.456755  1252.213296  1353.470360
## [61]  1462.827988  1580.934227  1708.488966  1846.248083  1995.027929
## [66]  2155.710164  2329.246977  2516.666735  2719.080074  2937.686480
## [71]  3173.781398  3428.763910  3704.145023  4001.556624  4322.761154
## [76]  4669.662047  5044.315011  5448.940211  5885.935428  6357.890263
## [81]  6867.601484  7418.089602  8012.616770  8654.706112  9348.162601
## [86] 10097.095609 10905.943258 11779.498718 12722.938616 13741.853705
## [91] 14842.282002 16030.744562 17314.284127 18700.506857 20197.627405
## [96] 21814.517598 23560.759006 25446.699726 27483.515704 29683.276961
```

Problem 6

Identify the elements of the sequence {2^1, . . . , 2^15} that exceed the corresponding elements of the sequence {1^3, . . . , 15^3}.

```
# Let's first create the sequences and then compare them
seq1 <- 2^(1:15)
seq2 <- (1:15)^3

result <- seq1 > seq2 # this inequality will spit out a boolean value
# if seq1 > seq2 is true, it will output "true"
result
```

```
##  [1]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
## [13]  TRUE  TRUE  TRUE
```

```
# so the 1st & 10~15th element of the first seq. exceeds the corresponding elements of the second seq
```

Problem 7

The following are a sample of observations on incoming solar radiation at a greenhouse: {11.1, 10.6, 6.3, 8.8, 10.7, 11.2, 8.9, 12.2 }

Assign the data to an object called solar.radiation.

```
solar.radiation <- c(11.1, 10.6, 6.3, 8.8, 10.7, 11.2, 8.9, 12.2)
```

Find the mean, median, range, and variance of the radiation observations.

```r
mean_solar <- mean(solar.radiation)
median_solar <- median(solar.radiation)
range_solar <- range(solar.radiation)
variance_solar <- var(solar.radiation)
mean_solar
```

```
## [1] 9.975
```

```r
median_solar
```

```
## [1] 10.65
```

```r
range_solar
```

```
## [1]  6.3 12.2
```

```r
variance_solar
```

```
## [1] 3.525
```

Add 10 to each observation of solar.radiation and assign the result to sr10. Find the mean, median, range, and variance of sr10. Which statistics change, and by how much?

```r
sr10 <- solar.radiation + 10

mean_sr10 <- mean(sr10)
median_sr10 <- median(sr10)
range_sr10 <- range(sr10)
variance_sr10 <- var(sr10)
mean_sr10
```

```
## [1] 19.975
```

```r
median_sr10
```

```
## [1] 20.65
```

```r
range_sr10
```

```
## [1] 16.3 22.2
```

```r
variance_sr10
```

```
## [1] 3.525
```

```
# Adding 10 increases the mean, median, and range (both lower & upper limit) by that amount
# Variance is unchanged
```

Multiply each observation of solar.radiation by -2 and assign the result to srm2. Find the mean, median, range, and variance of srm2. How do the statistics change now?

```
srm2 <- solar.radiation * -2

mean_srm2 <- mean(srm2)
median_srm2 <- median(srm2)
range_srm2 <- range(srm2)
variance_srm2 <- var(srm2)
mean_srm2
```

```
## [1] -19.95
```

```
median_srm2
```

```
## [1] -21.3
```

```
range_srm2
```

```
## [1] -24.4 -12.6
```

```
variance_srm2
```

```
## [1] 14.1
```

```
# Multiplying by -2 multiplies the mean, median by that amount
# Similar thing happens for range, but since the sign is changed, lower & upper limits swap places
# Variance multiplies by the square of the (-2), that is 4
```