Assignment 2

Averill Cate, Jr

February 23, 2018

Assignment 2

1. What are the different data types supported by R?

R's data types are vectors, arrays, data frames, lists, factors, matrices, numeric, character and logical and integer.

Create a vector consisting of the first thousand numbers.

```
v = 1L:1000L
tail(v, 10)
```

```
##  [1]  991  992  993  994  995  996  997  998  999 1000
```

2. Using the control structures, write a program to

a. find the odd numbers from the above created vector

I did this two ways.

```
# With control structures
find.odds <- function(values) {
  l = c(0)
  for(v in values) {
    if (values[[v]] %% 2 == 1) {
        l = c(l, values[[v]])
    }
  }
  return(l)
}
head(find.odds(v), 20)
```

```
##  [1]  0  1  3  5  7  9 11 13 15 17 19 21 23 25 27 29 31 33 35 37
```

```
tail(find.odds(v), 20)
```

```
##  [1] 961 963 965 967 969 971 973 975 977 979 981 983 985 987 989 991 993
## [18] 995 997 999
```

```
# Without control structures
find.odds <- function(values) {
  result <- values[c(T,F)]
  return(result)
}
head(find.odds(v), 20)
```

```
##  [1]  1  3  5  7  9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
```

```
tail(find.odds(v), 20)
```

```
##  [1] 961 963 965 967 969 971 973 975 977 979 981 983 985 987 989 991 993
## [18] 995 997 999
```

b. Find all the numbers divisible by 7.

I did this two ways.

```
# With control structures
divby.seven <- function(values) {
  l = c(0)
  for(v in values) {
    if (values[[v]] %% 7 == 0) {
        l = c(l, values[[v]])
    }
  }
  return(l)

}
head(divby.seven(v), 20)
```

```
##  [1]   0   7  14  21  28  35  42  49  56  63  70  77  84  91  98 105 112
## [18] 119 126 133
```

```
tail(divby.seven(v), 20)
```

```
##  [1] 861 868 875 882 889 896 903 910 917 924 931 938 945 952 959 966 973
## [18] 980 987 994
```

```
# Without control structures
divby.seven <- function(values) {
  indexes = values %% 7
  return(values[!indexes])
}
head(divby.seven(v), 20)
```

```
##  [1]   7  14  21  28  35  42  49  56  63  70  77  84  91  98 105 112 119
## [18] 126 133 140
```

```
tail(divby.seven(v), 20)
```

```
##  [1] 861 868 875 882 889 896 903 910 917 924 931 938 945 952 959 966 973
## [18] 980 987 994
```

3. Write a function in R which

a. Takes an input of three sides and tells whether it is an equilateral triangle, isosceles or scalene triangle.

```
triangle.type <- function(s1, s2, s3) {
  data = c(s1, s2, s3)
  fdata = factor(data)
  levs = nlevels(fdata)
  if ( levs == 2) {
```

```
    return("isoceles")
  } else if (levs == 1) {
    return("equilateral")
  } else {
    return("scalene")
  }
}

print(triangle.type(1,2,3))
```

```
## [1] "scalene"
```

```
print(triangle.type(2,2,3))
```

```
## [1] "isoceles"
```

```
print(triangle.type(2,2,2))
```

```
## [1] "equilateral"
```

```
print(triangle.type(4,1,2))
```

```
## [1] "scalene"
```

```
print(triangle.type(4,4,4))
```

```
## [1] "equilateral"
```

b. Takes two points (x,y) and finds the Cartesian distance between the 2D plane.

```
cartesian.distance <- function(p1, p2) {
  dist = NULL
  t = (p2 - p1)^2
  s = sum(t)
  d = sqrt(s)
  return(d)
}
print(cartesian.distance(c(1,1), c(2,2)))
```

```
## [1] 1.414214
```

```
print(cartesian.distance(c(1,1), c(4,7)))
```

```
## [1] 6.708204
```