Problem Set 2 Key

Due February 13, 10:00 AM (Before Class)

for loops, if else, while

1. Write a for loop that iterates over the numbers 1 to 7 and prints the cube of each number using print().

```
for (i in 1:7) {
  print(i^3)
}
```

   - Write the loop such that if the two dice total to values 8,9,10,11,12 the game ends immediately
   - If the first roll does not equal one of those five values continue to roll the dice until you roll either a 2 or a 6
   - What is the average number of dice casts per game

```
set.seed(14)

n.sim = 1000 # number of simulations

cts = rep(0, n.sim) # counts of dice casts

x = NULL # container for the current cast total

for (i in 1:n.sim) {
   start = c(sample(1:6,1) + sample(1:6,1)) # a pair of dice
   cts[i] = cts[i] + 1 # update of cast counts
   x = start
     if(x %in% c(8,9,10,11,12)){ # initial target totals
     next} # stop the game if this total is casted

   repeat {
     x = c(sample(1:6,1) + sample(1:6,1))
     cts[i] = cts[i] + 1
     if(x %in% c(2, 6)){ # new target of the repeat loop
       break
     }}}
```

2. Load the following data: http://politicaldatascience.com/PDS/Problem%20Sets/Problem%20Set%202/GSS-data.csv

   Now create a function called `vote.choice` which can take one of three commands: "Trump", "Clinton", or "Other". The function should return the number of participants who voted for Trump when you input "Trump" into the function; the number of participants who voted for Clinton when you input "Clinton" into the function; and the number of participants that voted for neither when you input "Other".

   - Now edit this function so that if a pre-defined object, numeric value or misspelled word is entered, the function returns the message"Please enter either 'Trump' 'Clinton' or 'Other' into the function to return valid response".

```
gss<- read.csv('http://politicaldatascience.com/PDS/Problem%20Sets/Problem%20Set%202/GSS-data.csv')
vote.choice <- function(x){
  if((x =="Trump")){
    cat(sum(gss$pres16=="Trump"), "voted for Donald Trump")
```

```
  }else if(x =="Clinton"){
    cat(sum(gss$pres16=="Clinton"), "voted for Hilary Clinton")
  }else if(x =="Other") {cat(sum(gss$pres16!= "Clinton"&gss$pres16 != "Trump"),
"did not report voting for either candidate")
  }else if(x!= "Clinton" & x!= "Trump" & x!= "Other"){
print("Please enter either 'Trump' 'Clinton' or
'Other' into the function to return valid response")
  }

}

vote.choice('Trump')
```

```
## 577 voted for Donald Trump
```

```
vote.choice('Clinton')
```

```
## 764 voted for Hilary Clinton
```

```
vote.choice('Other')
```

```
## 1009 did not report voting for either candidate
```

3. Run the following code:

```
install.packages('fivethrityeight')
library(fivethirtyeight)
```

4. Now review the data in the `cabinet_turnover` object (this is loaded into your space when you load the library even though you cannot see it in the global space. You can also assign it to your own object if you'd like.).

   Write a function which allows you to type in the name of a president and returns the proportion of time appointees spent serving each adminstration i.e the number of days appointees served for each administration, on average, divided by the number of days the particular president served.

   To illustrate the average number of days all appointees served in the Reagan administration was 2140.959. Below you can see that Reagan served 2922 days. So appointees served 73% of Reagan's administration, on average (2140.959/2922).

   For simplicity, here are the number of days each president served:

   Carter: 1461

   Reagan: 2922

   Bush 41: 1461

   Clinton: 2922

   Bush 43: 2922

   Obama: 2922

   Trump: 1105[1]

```
library(fivethirtyeight)

average.days <- function(administration){
  if(administration == 'Reagan'){
```

[1] As of January 30, 2020

```
    mean(cabinet_turnover$days[cabinet_turnover$president=='Reagan'], na.rm=T)/2922
  }else if(administration == 'Carter'){
    mean(cabinet_turnover$days[cabinet_turnover$president=='Carter'], na.rm=T)/1461
  }else if(administration == 'Bush 41'){
    mean(cabinet_turnover$days[cabinet_turnover$president=='Bush 41'], na.rm=T)/1461
  }else if(administration == 'Clinton'){
    mean(cabinet_turnover$days[cabinet_turnover$president=='Clinton'], na.rm=T)/2922
  }else if(administration == 'Bush 43'){
    mean(cabinet_turnover$days[cabinet_turnover$president=='Bush 43'], na.rm=T)/2922
  }else if(administration == 'Obama'){
    mean(cabinet_turnover$days[cabinet_turnover$president=='Obama'], na.rm=T)/2922
  }else if(administration == 'Trump'){
    mean(cabinet_turnover$days[cabinet_turnover$president=='Trump'], na.rm=T)/1105
  }
}


average.days('Reagan')
```

```
## [1] 0.7327033
```

```
average.days('Carter')
```

```
## [1] 0.8683757
```

```
average.days('Bush 41')
```

```
## [1] 0.8839634
```

```
average.days('Clinton')
```

```
## [1] 0.6972091
```

```
average.days('Bush 43')
```

```
## [1] 0.7516721
```

```
average.days('Obama')
```

```
## [1] 0.7250932
```

```
average.days('Trump')
```

```
## [1] 0.4524887
```

5. Now you will use the `congress_age` data set. Create a function called `congress_stats` that has two commands "congress" and "state".

   When you enter "congress" into the function it should return the average age of congressmembers for each congressional era. Your function should return 34 results which display the average age of congressmembers of an era as well as the congress. For example the most recent congress is the 113 Congress so your last result should be `57.6 113`.

   Similarly, when you input "state" into the function, it should return the average age of congressmembers by state. The function will then return 50 results an example of one of the 50 is `53.4 TX`.

```
congress_stats <- function(stats){
  if(stats == 'state'){
    for(i in 1:50){
      a<-  round(mean(congress_age$age[congress_age$state == state.abb[i]]), digits=1)
```

```
      print(paste(a,state.abb[i]), sep= " ")}
  }else if(stats =="congress"){
      for(i in 1:34){
      b<-round(mean(congress_age$age[congress_age$congress == unique(congress_age$congress)[i]]), d
      print(paste(b, unique(congress_age$congress)[i], sep= " "))}
    }}

congress_stats('congress')
```

```
## [1] "52.5 80"
## [1] "52.6 81"
## [1] "53.2 82"
## [1] "53.2 83"
## [1] "53.4 84"
## [1] "54.2 85"
## [1] "53.2 86"
## [1] "53.6 87"
## [1] "53 88"
## [1] "52.2 89"
## [1] "52.4 90"
## [1] "52.7 91"
## [1] "53.1 92"
## [1] "52.4 93"
## [1] "51.3 94"
## [1] "50.6 95"
## [1] "50.1 96"
## [1] "49.5 97"
## [1] "50.1 98"
## [1] "51.1 99"
## [1] "51.9 100"
## [1] "52.7 101"
## [1] "53.8 102"
## [1] "53 103"
## [1] "52.7 104"
## [1] "53.1 105"
## [1] "54.2 106"
## [1] "54.7 107"
## [1] "55.4 108"
## [1] "56.5 109"
## [1] "57.3 110"
## [1] "57.8 111"
## [1] "57.6 112"
## [1] "57.6 113"
```

```
congress_stats('state')
```

```
## [1] "53 AL"
## [1] "57.6 AK"
## [1] "54.7 AZ"
## [1] "52.8 AR"
## [1] "53.6 CA"
## [1] "52.3 CO"
## [1] "49.7 CT"
## [1] "53.1 DE"
```

```
## [1] "53.4 FL"
## [1] "53.4 GA"
## [1] "59.5 HI"
## [1] "50.8 ID"
## [1] "54.7 IL"
## [1] "49.6 IN"
## [1] "54 IA"
## [1] "53.9 KS"
## [1] "55.6 KY"
## [1] "50.1 LA"
## [1] "50.6 ME"
## [1] "53.9 MD"
## [1] "53.4 MA"
## [1] "52.2 MI"
## [1] "50.4 MN"
## [1] "53.5 MS"
## [1] "52.5 MO"
## [1] "53.8 MT"
## [1] "55 NE"
## [1] "54.5 NV"
## [1] "51.2 NH"
## [1] "53.9 NJ"
## [1] "53.9 NM"
## [1] "53 NY"
## [1] "56.4 NC"
## [1] "58.1 ND"
## [1] "53.7 OH"
## [1] "49.4 OK"
## [1] "52.5 OR"
## [1] "53.8 PA"
## [1] "52.1 RI"
## [1] "54.6 SC"
## [1] "50.2 SD"
## [1] "52 TN"
## [1] "53.4 TX"
## [1] "54.8 UT"
## [1] "59 VT"
## [1] "55.2 VA"
## [1] "52.4 WA"
## [1] "56.6 WV"
## [1] "50.7 WI"
## [1] "56.8 WY"
```