Homework3Q1

David Li

November 13, 2017

Stats 506: Homework 3 Question 1
David Li
Data Used For This Question: RECS2009 Data: http://www.eia.gov/consumption/residential/data/2009/csv/recs2009_public.csv

```
library("data.table", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("ggplot2", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("knitr", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("rmarkdown", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("curl", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("rvest", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("tidyverse", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
```

```
# Begin Functions Section
decode_state = function(x){ # Decodes the numeric representation of states into their actual names
  if(!is.numeric(x)){
    stop('decode_state expects numeric input indexed from 1!')
  }
  switch(x,
         "CT, ME, NH, RI, VT", "MA", "NY", "NJ", "PA", "IL", "IN, OH", "MI", "WI",
         "IA, MN, ND, SD", "KS, NE", "MO", "VA", "DE, DC, MD, WV", "GA",
         "NC, SC" , "FL", "AL, KY, MS", "TN", "AR, LA, OK",
         "TX", "CO", "ID, MT, UT, WY", "AZ", "NV, NM",
         "CA", "AK, HI, OR, WA"
  )
}
decode_all_states = function(x){ # Applies decoding to a vector instead of a single value
  sapply(x, decode_state)
}
decode_roof_type = function(x){
  # Decodes the numeric representation of rooftypes into their actual names
  if(!is.numeric(x)){
    stop('decode_roof_type expects numeric input indexed from 1!')
  }
  # Not Applicable values will be filtered out later
  switch(x,
         "CeramicClay", "WoodShing", "Metal",
         "Slate", "CompShing", "Asphalt",
         "Concrete_Tiles", "Other"
  )
}
decode_all_roof_types = function(x){ # Applies decoding to a vector instead of a single value
  sapply(x, decode_roof_type)
}
decode_decade_range = function(x){
  # Decodes the numeric representation of decades into their actual names
  if(!is.numeric(x)){
    stop('decode_decade_range expects numeric input indexed from 1!')
```

```
  }
  switch(x,
         "1950Less", "1950-59", "1960-69", "1970-79",
         "1980-89", "1990-99", "2000-09", "2000-09"
         # This effectively combines '00 - '04 and '05 - '09 together
  )
}
decode_all_decade_ranges = function(x){ # Applies decoding to a vector instead of a single value
  sapply(x, decode_decade_range)
}

Increase_percent = function(x){
  # Calculate relative increase percentage, for a vector with a start and end percent
  return(100*(x[2]-x[1]) / x[1])
}
# End Functions section
```

- Here are some functions that prove useful for the steps later

```
# Import whole RECS dataset, using data.table
recsorig = fread("~/Desktop/Stats506/Datasets/recs2009_public.csv")

# First keeping relevant columns, then doing decoding using data.table
recs_roof = recsorig[, .(UniqueId = DOEID, State=REPORTABLE_DOMAIN, RoofType = ROOFTYPE,
                         YearMade = YEARMADE, YearMadeDecade = YEARMADERANGE, Weight = NWEIGHT)
  ][RoofType != -2 # Discarding N/A data
  ][, ':=' (State = decode_all_states(State), RoofType = decode_all_roof_types(RoofType)) # Decoding
  ][, .(Totalweight = sum(Weight)), keyby = .(State, RoofType)
    # Summarizing, by weights and by grouping
  ]
# Column bind for retaining data structure after re-shaping the data
recs_roof = cbind(State = unique(recs_roof$State), as.data.table(tapply(recs_roof$Totalweight,list(recs
# Computing Proportions
recs_roof = recs_roof[, ':=' (Total = rowSums(.SD, na.rm = T)), .SDcols = 2:9
  ][, .(State = State,
        Asphalt = 100*Asphalt/Total,
        CeramicClay = 100*CeramicClay/Total,
        CompShing = 100*CompShing/Total,
        Concrete_Tiles = 100*Concrete_Tiles/Total,
        Metal = 100*Metal/Total,
        Other = 100*Other/Total,
        Slate = 100*Slate/Total,
        WoodShing = 100*WoodShing/Total)
  ][order(-WoodShing) # Greatest Wood Shingle usage State first
  ]
# Table for display
kable(recs_roof, digits=2, caption='Proportion of roof types by State(s).')
```

Table 1: Proportion of roof types by State(s).

State	Asphalt	CeramicClay	CompShing	Concrete_Tiles	Metal	Other	Slate	WoodShing
NC, SC	20.54	NA	51.29	0.32	11.17	0.35	1.60	14.72
CA	8.85	16.34	51.66	5.38	3.15	1.10	2.52	10.99
NV, NM	18.07	24.29	23.47	2.65	11.20	8.12	1.29	10.90

State	Asphalt	CeramicClay	CompShing	Concrete_Tiles	Metal	Other	Slate	WoodShing
CO	19.77	0.65	55.11	2.05	9.80	1.77	0.28	10.56
ID, MT, UT, WY	45.08	NA	34.37	0.60	8.83	0.69	0.60	9.83
TX	2.91	0.91	77.39	0.42	8.21	0.32	0.53	9.31
FL	18.56	7.60	41.49	3.16	18.05	1.42	1.59	8.12
AK, HI, OR, WA	6.55	1.23	73.50	0.61	9.12	1.32	0.19	7.46
IN, OH	19.79	0.50	62.51	NA	8.50	0.30	1.63	6.77
DE, DC, MD, WV	13.15	0.59	64.73	NA	9.81	0.86	4.10	6.77
PA	19.04	NA	58.65	NA	5.92	6.66	3.00	6.73
GA	13.93	1.85	71.99	0.24	4.13	0.50	0.74	6.62
AZ	7.42	31.87	23.20	14.76	11.94	4.12	0.56	6.14
MA	54.64	0.54	34.47	NA	1.62	1.58	1.09	6.06
NY	36.36	NA	48.58	0.34	4.46	2.54	1.79	5.94
MO	13.04	0.71	68.05	0.56	9.78	0.92	1.26	5.69
KS, NE	20.57	0.61	68.87	NA	4.15	0.25	0.33	5.21
CT, ME, NH, RI, VT	33.46	NA	48.25	0.25	11.37	0.22	1.66	4.78
IA, MN, ND, SD	44.57	0.16	45.30	NA	4.42	0.52	0.42	4.61
IL	30.73	NA	61.18	0.44	1.93	0.50	0.93	4.28
VA	14.78	0.38	65.62	NA	14.43	NA	1.00	3.80
WI	51.05	2.20	39.30	NA	2.33	0.51	0.82	3.79
AR, LA, OK	14.22	1.19	67.53	NA	12.08	NA	1.48	3.49
AL, KY, MS	9.96	0.34	60.53	0.24	25.26	0.28	NA	3.39
MI	16.72	0.41	66.63	NA	9.48	1.42	1.99	3.36
NJ	19.38	NA	74.42	2.62	0.72	NA	NA	2.86
TN	25.01	NA	57.68	0.45	13.96	0.68	0.46	1.77

```
# First keeping relevant columns, then doing decoding using data.table
recs_decade = recsorig[, .(UniqueId = DOEID, State=REPORTABLE_DOMAIN,
                           RoofType = ROOFTYPE, YearMade = YEARMADE,
                           YearMadeDecade = YEARMADERANGE, Weight = NWEIGHT)
  ][RoofType != -2 # Discarding N/A data
  ][, ':=' (State = decode_all_states(State), RoofType = decode_all_roof_types(RoofType), YearMadeDecad
  ][, .(Totalweight = sum(Weight)), keyby = .(YearMadeDecade, RoofType)
    # Summarizing, by weights and by grouping
  ]
# Column bind for retaining data structure after re-shaping the data
recs_decade = cbind(YearMadeDecade = unique(recs_decade$YearMadeDecade), as.data.table(tapply(recs_deca
# Computing Proportions
recs_decade = recs_decade[, ':=' (Total = rowSums(.SD, na.rm = T)), .SDcols = 2:9
  ][, .(YearMadeDecade = YearMadeDecade,
         Asphalt = 100*Asphalt/Total,
         CeramicClay = 100*CeramicClay/Total,
         CompShing = 100*CompShing/Total,
         Concrete_Tiles = 100*Concrete_Tiles/Total,
         Metal = 100*Metal/Total,
         Other = 100*Other/Total,
         Slate = 100*Slate/Total,
         WoodShing = 100*WoodShing/Total)
  ]
recs_decade_rorder = recs_decade[c(2,1,3,4,5,6,7),] # Reorder-ing for proper chronological order

# Table for display
kable(recs_decade_rorder, digits=2, caption='Proportion of roof types by Decade(s).')
```

Table 2: Proportion of roof types by Decade(s).

YearMadeDecade	Asphalt	CeramicClay	CompShing	Concrete_Tiles	Metal	Other	Slate	WoodShing
1950Less	26.70	1.16	53.95	0.39	6.86	2.37	2.71	5.85
1950-59	21.41	1.05	61.20	0.84	4.54	1.35	1.58	8.03
1960-69	21.16	2.02	58.92	0.70	6.28	1.35	1.34	8.23
1970-79	17.68	3.17	56.03	1.29	13.06	1.70	0.76	6.31
1980-89	17.40	5.01	52.12	1.44	13.78	0.66	1.20	8.39
1990-99	16.32	6.63	55.94	1.89	11.56	0.58	0.98	6.12
2000-09	15.52	5.75	64.05	2.96	4.50	0.35	0.45	6.42

```
# First keeping relevant columns, then doing decoding using data.table
recs_1950_2000 = recsorig[, .(UniqueId = DOEID, State=REPORTABLE_DOMAIN, RoofType = ROOFTYPE,
                              YearMade = YEARMADE, YearMadeDecade = YEARMADERANGE, Weight = NWEIGHT)
  ][RoofType != -2 # Discarding N/A data
  ][YearMade == 1950 | YearMade == 2000 # Only care about years 1950 and 2000
  ][, ':=' (State = decode_all_states(State), RoofType = decode_all_roof_types(RoofType),
            YearMadeDecade = decode_all_decade_ranges(YearMadeDecade)) # Decoding
  ][, .(Totalweight = sum(Weight)), keyby = .(YearMade, RoofType)
    # Summarizing, by weights and by grouping
  ]
# Column bind for retaining data structure after re-shaping the data
recs_1950_2000 = cbind(YearMade = unique(recs_1950_2000$YearMade),
                       as.data.table(tapply(recs_1950_2000$Totalweight,list(recs_1950_2000$YearMade,
                                                                            recs_1950_2000$RoofType), FU
# Computing Proportions
recs_1950_2000 = recs_1950_2000[, ':=' (Total = rowSums(.SD, na.rm = T)), .SDcols = 2:9
  ][, .(YearMade = YearMade,
        Asphalt = 100*Asphalt/Total,
        CeramicClay = 100*CeramicClay/Total,
        CompShing = 100*CompShing/Total,
        Concrete_Tiles = 100*Concrete_Tiles/Total,
        Metal = 100*Metal/Total,
        Other = 100*Other/Total,
        Slate = 100*Slate/Total,
        WoodShing = 100*WoodShing/Total)
  ]
# Computing Relative Percent increases into a row, then binding this to the data.table
percentages = recs_1950_2000[, lapply(.SD, Increase_percent), .SDcols = 2:9]
recs_1950_2000 = rbind(recs_1950_2000, percentages, fill = TRUE)
# Table for display
kable(recs_1950_2000, digits=2, caption='Relative Increase of RoofType Usage from 1950 to 2000.')
```

Table 3: Relative Increase of RoofType Usage from 1950 to 2000.

YearMade	Asphalt	CeramicClay	CompShing	Concrete_Tiles	Metal	Other	Slate	WoodShing
1950	19.59	1.38	61.31	0.70	5.86	1.39	2.09	7.69
2000	17.99	5.55	62.85	1.42	5.83	NA	NA	6.37
NA	-8.17	303.27	2.50	101.59	-0.49	NA	NA	-17.08

- The steps for data.table processing of the RECS dataset is very much similar to dplyr
- Decoding needed to be done first to have meaningful names, followed a same "decoding key" as similar to last hw
- Computing proportions was similar to last homework too, a table was created for each of the 3 questions from the question set.
- data.table does an excellent job of summarizing data very quickly and efficiently, as well as parsing data
- Most of the explanations are in the R code as comments, best described there