Getting and Cleaning Data - Data Science - Quiz 4 - Coursera

Getting and Cleaning Data Quiz 4

This is Quiz 4 from the Getting and Cleaning Data course within the Data Science Specialization on Coursera.

Questions

---

1. The American Community Survey distributes downloadable data about United States communities. Download the 2006 microdata survey about housing for the state of Idaho using download.file() from here:

https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2Fss06hid.csv

and load the data into R. The code book, describing the variable names is here:

https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FPUMSDataDict06.pdf

Apply strsplit() to split all the names of the data frame on the characters "wgtp". What is the value of the 123 element of the resulting list?

- **"" "15"**

```
library(data.table)

download.file("https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2Fss06hid.csv", destf
ile = "quiz4data.csv")

d <- read.csv("quiz4data.csv")

strsplit(names(d), split = "wgtp")[123]
```

```
## [[1]]

## [1] ""   "15"
```

---

2. Load the Gross Domestic Product data for the 190 ranked countries in this data set:

https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FGDP.csv

Remove the commas from the GDP numbers in millions of dollars and average them. What is the average?

Original data sources:

http://data.worldbank.org/data-catalog/GDP-ranking-table

- **377652.4**

```
download.file("https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FGDP.csv", destfile
= "quiz4data.csv")

data <- read.csv("quiz4data.csv")


clean <- gsub(",","",data[,5])


mean(as.numeric(clean[1:215]),na.rm = TRUE)
```

---

3. In the data set from Question 2 what is a regular expression that would allow you to count the number of countries whose name begins with “United”? Assume that the variable with the country names in it is named countryNames. How many countries begin with United?

- **grep(“^United”,countryNames), 3**

---

4. Load the Gross Domestic Product data for the 190 ranked countries in this data set:

https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FGDP.csv

Load the educational data from this data set:

https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FEDSTATS_Country.csv

Match the data based on the country shortcode. Of the countries for which the end of the fiscal year is available, how many end in June?

- **13**

```
download.file("https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FGDP.csv", destfile
= "quiz4data2.csv")
```

```
data2 <- read.csv("quiz4data2.csv")


download.file("https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FEDSTATS_Country.csv
", destfile = "quiz4data3.csv")


data3 <- read.csv("quiz4data3.csv")


setnames(data2, c("X", "X.1", "X.3", "X.4"), c("CountryCode", "rankingGDP", "Long.Name",
"gdp"))


all <- merge(data2, data3, by = "CountryCode")


table(grepl("june", tolower(all$Special.Notes)), grepl("fiscal year end", tolower(all$Spe
cial.Notes)))[4]
```

```
## [1] 13
```

5.

You can use the quantmod (http://www.quantmod.com/) package to get historical stock prices for publicly traded companies on the NASDAQ and NYSE. Use the following code to download data on Amazon's stock price and get the times the data was sampled.

```
library(quantmod)
```

```
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
## Attaching package: 'xts'
## The following object is masked from 'package:data.table':
##
##     last
## Loading required package: TTR
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
amzn = getSymbols("AMZN",auto.assign=FALSE)
```

```
##     As of 0.4-0, 'getSymbols' uses env=parent.frame() and
```

```
##  auto.assign=TRUE by default.
##
##  This  behavior  will be  phased out in 0.5-0  when the call  will
##  default to use auto.assign=FALSE. getOption("getSymbols.env") and
##  getOptions("getSymbols.auto.assign") are now checked for alternate defaults
##
##  This message is shown once per session and may be disabled by setting
##  options("getSymbols.warning4.0"=FALSE). See ?getSymbols for more details.
sampleTimes = index(amzn)
```

How many values were collected in 2012? How many values were collected on Mondays in 2012? head(data)

- **250, 47**

```
length(grep("^2012",sampleTimes))
## [1] 250
library(lubridate)
##
## Attaching package: 'lubridate'
## The following objects are masked from 'package:data.table':
##
##     hour, mday, month, quarter, wday, week, yday, year
## The following object is masked from 'package:base':
##
##     date
sampleTimes[grep("^2012",sampleTimes)]
##   [1] "2012-01-03" "2012-01-04" "2012-01-05" "2012-01-06" "2012-01-09"
##   [6] "2012-01-10" "2012-01-11" "2012-01-12" "2012-01-13" "2012-01-17"
##  [11] "2012-01-18" "2012-01-19" "2012-01-20" "2012-01-23" "2012-01-24"
##  [16] "2012-01-25" "2012-01-26" "2012-01-27" "2012-01-30" "2012-01-31"
##  [21] "2012-02-01" "2012-02-02" "2012-02-03" "2012-02-06" "2012-02-07"
##  [26] "2012-02-08" "2012-02-09" "2012-02-10" "2012-02-13" "2012-02-14"
##  [31] "2012-02-15" "2012-02-16" "2012-02-17" "2012-02-21" "2012-02-22"
##  [36] "2012-02-23" "2012-02-24" "2012-02-27" "2012-02-28" "2012-02-29"
##  [41] "2012-03-01" "2012-03-02" "2012-03-05" "2012-03-06" "2012-03-07"
##  [46] "2012-03-08" "2012-03-09" "2012-03-12" "2012-03-13" "2012-03-14"
```

```
##  [51] "2012-03-15" "2012-03-16" "2012-03-19" "2012-03-20" "2012-03-21"
##  [56] "2012-03-22" "2012-03-23" "2012-03-26" "2012-03-27" "2012-03-28"
##  [61] "2012-03-29" "2012-03-30" "2012-04-02" "2012-04-03" "2012-04-04"
##  [66] "2012-04-05" "2012-04-09" "2012-04-10" "2012-04-11" "2012-04-12"
##  [71] "2012-04-13" "2012-04-16" "2012-04-17" "2012-04-18" "2012-04-19"
##  [76] "2012-04-20" "2012-04-23" "2012-04-24" "2012-04-25" "2012-04-26"
##  [81] "2012-04-27" "2012-04-30" "2012-05-01" "2012-05-02" "2012-05-03"
##  [86] "2012-05-04" "2012-05-07" "2012-05-08" "2012-05-09" "2012-05-10"
##  [91] "2012-05-11" "2012-05-14" "2012-05-15" "2012-05-16" "2012-05-17"
##  [96] "2012-05-18" "2012-05-21" "2012-05-22" "2012-05-23" "2012-05-24"
## [101] "2012-05-25" "2012-05-29" "2012-05-30" "2012-05-31" "2012-06-01"
## [106] "2012-06-04" "2012-06-05" "2012-06-06" "2012-06-07" "2012-06-08"
## [111] "2012-06-11" "2012-06-12" "2012-06-13" "2012-06-14" "2012-06-15"
## [116] "2012-06-18" "2012-06-19" "2012-06-20" "2012-06-21" "2012-06-22"
## [121] "2012-06-25" "2012-06-26" "2012-06-27" "2012-06-28" "2012-06-29"
## [126] "2012-07-02" "2012-07-03" "2012-07-05" "2012-07-06" "2012-07-09"
## [131] "2012-07-10" "2012-07-11" "2012-07-12" "2012-07-13" "2012-07-16"
## [136] "2012-07-17" "2012-07-18" "2012-07-19" "2012-07-20" "2012-07-23"
## [141] "2012-07-24" "2012-07-25" "2012-07-26" "2012-07-27" "2012-07-30"
## [146] "2012-07-31" "2012-08-01" "2012-08-02" "2012-08-03" "2012-08-06"
## [151] "2012-08-07" "2012-08-08" "2012-08-09" "2012-08-10" "2012-08-13"
## [156] "2012-08-14" "2012-08-15" "2012-08-16" "2012-08-17" "2012-08-20"
## [161] "2012-08-21" "2012-08-22" "2012-08-23" "2012-08-24" "2012-08-27"
## [166] "2012-08-28" "2012-08-29" "2012-08-30" "2012-08-31" "2012-09-04"
## [171] "2012-09-05" "2012-09-06" "2012-09-07" "2012-09-10" "2012-09-11"
## [176] "2012-09-12" "2012-09-13" "2012-09-14" "2012-09-17" "2012-09-18"
## [181] "2012-09-19" "2012-09-20" "2012-09-21" "2012-09-24" "2012-09-25"
## [186] "2012-09-26" "2012-09-27" "2012-09-28" "2012-10-01" "2012-10-02"
## [191] "2012-10-03" "2012-10-04" "2012-10-05" "2012-10-08" "2012-10-09"
## [196] "2012-10-10" "2012-10-11" "2012-10-12" "2012-10-15" "2012-10-16"
## [201] "2012-10-17" "2012-10-18" "2012-10-19" "2012-10-22" "2012-10-23"
## [206] "2012-10-24" "2012-10-25" "2012-10-26" "2012-10-31" "2012-11-01"
## [211] "2012-11-02" "2012-11-05" "2012-11-06" "2012-11-07" "2012-11-08"
## [216] "2012-11-09" "2012-11-12" "2012-11-13" "2012-11-14" "2012-11-15"
## [221] "2012-11-16" "2012-11-19" "2012-11-20" "2012-11-21" "2012-11-23"
## [226] "2012-11-26" "2012-11-27" "2012-11-28" "2012-11-29" "2012-11-30"
## [231] "2012-12-03" "2012-12-04" "2012-12-05" "2012-12-06" "2012-12-07"
## [236] "2012-12-10" "2012-12-11" "2012-12-12" "2012-12-13" "2012-12-14"
```

```
## [241] "2012-12-17" "2012-12-18" "2012-12-19" "2012-12-20" "2012-12-21"
## [246] "2012-12-24" "2012-12-26" "2012-12-27" "2012-12-28" "2012-12-31"
```

```
sum(weekdays(as.Date(sampleTimes[grep("^2012",sampleTimes)]))=="Monday")
```

```
## [1] 47
```
