Quiz 1 Solutions

APPM 2720 Doug Nychka

2/23/2017

(1) ***Load the data set BoulderJuneTemperature convert the temperatures from Fahrenheit to Kelvin.*** $(\mathrm{T(K)} = (\mathrm{T(°F)} + 459.67) \times 5/9)$

```
load( "BoulderJuneTemperature.rda")
#this data set has two columns!
temperatureKelvin <- (5/9)* (BoulderJuneTemperature[,2] + 459.67)
  max(temperatureKelvin)
```

```
## [1] 296.5574
```

```
 (5/9)*(max(BoulderJuneTemperature[,2]) + 459.67)
```

```
## [1] 296.5574
```

(2) ***If you made a boxplot of these data and the Y axis did not have a scale could you tell if the units were Kelvin or Fahrenheit?***

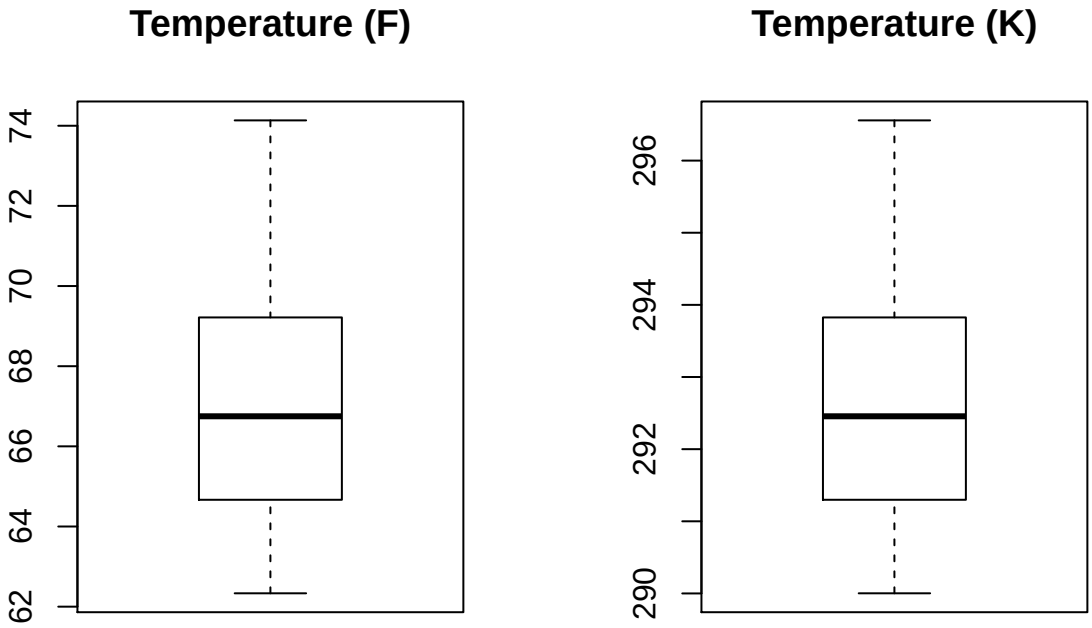
No because adding a constant to a data set and scaling it by a constant does not change the *relative* relationships among the statistics used to compute a boxplot and the way outliers are identified. Another way of expressing this is after drawing the boxplot in Fahrenheit if one just relabled the Y axis in Kelvin then that figure would be **identical** to drawing the boxplot with all the temperatures first converted to Kelvin.

```
library(fields)
```

```
set.panel(1,2)
```

```
plot window will lay out plots in a 1 by 2 matrix
```

```
boxplot(BoulderJuneTemperature[,2], main ="Temperature (F)")
boxplot( temperatureKelvin, main ="Temperature (K)")
```



(3) ***How many rows and columns are in the data set*** *ufo*. ***Is this a matrix or a data frame?***

64506 Rows and 8 Columns. This is a **data.frame** because some columns are character strings.

```
load("ufoQuiz1.rda")
dim( ufo)
```

```
[1] 64506     8
```

```
head( ufo)
```

```
           datetime state    shape duration.seconds latitude longitude
1  10/10/1949 20:30    tx cylinder             2700 29.88306 -97.94111
4  10/10/1956 21:00    tx   circle               20 28.97833 -96.64583
6  10/10/1961 19:00    tn   sphere              300 36.59500 -82.18889
8  10/10/1965 23:45    ct     disk             1200 41.11750 -73.40833
9  10/10/1966 20:00    al     disk              180 33.58611 -86.28611
10 10/10/1966 21:00    fl     disk              120 30.29472 -82.98417
   year month
1  1949    10
4  1956    10
6  1961    10
8  1965    10
9  1966    10
10 1966    10
```

(4) ***How many different UFO shapes are reported? What is the most common shape reported?***

There are 29 different shapes with "light" being the most common.

```
tableUfo<- table( ufo$shape)
length( tableUfo)
```

```
[1] 29
```

(5) ***Do you find any interesting relationships between the shape and the duration of the event?***

Most of the median durations for shapes are (suspiciously) around 2-4 minutes suggesting the reporting rounds to the minute often and the actual duration may be a guess. It seems that the "changing" category has the greatest range of durations. The specific shapes of "fireball", "chevron", "flash" have reduced ranges.

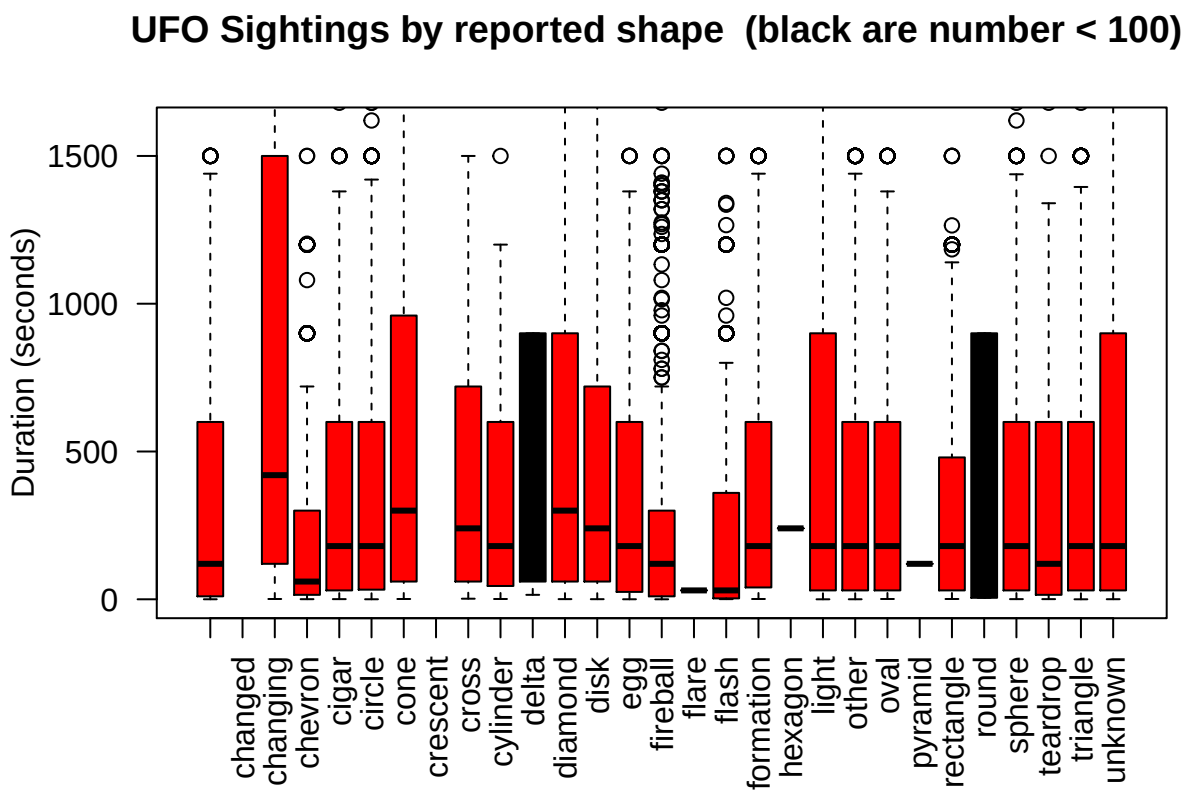
```
Q2Ufo<- tapply( ufo$duration, ufo$shape, FUN="median")
Q2Ufo
```

```
           changed  changing   chevron     cigar    circle      cone
    120.0    3600.0     420.0      60.0     180.0     180.0     300.0
 crescent     cross  cylinder     delta   diamond      disk       egg
  37800.0     240.0     180.0     360.0     300.0     240.0     180.0
 fireball     flare     flash formation   hexagon     light     other
    120.0      30.0      30.0     180.0     240.0     180.0     180.0
     oval   pyramid rectangle     round    sphere  teardrop  triangle
    180.0     120.0     180.0     452.5     180.0     120.0     180.0
  unknown
    180.0
```

```
iqrUfo<- tapply( ufo$duration, ufo$shape, FUN="IQR")
countUfo<-  tapply( ufo$duration, ufo$shape, FUN="length")
ind<- countUfo >100 # only look cases for larger sample sizes
# this plot not too helpful!
# plot(Q2Ufo[ind],iqrUfo[ind], xlab="median", ylab="interquartile range"  )

boxplot(ufo$duration ~ ufo$shape, ylim = c(0,1600),
```

```
        col= (countUfo >100) +1 ,
        las=2, axis.cex=.8,   # las rotates labels to be perp to axis
        ylab="Duration (seconds)",
        xlab="")
title("UFO Sightings by reported shape  (black are number < 100) ")
```



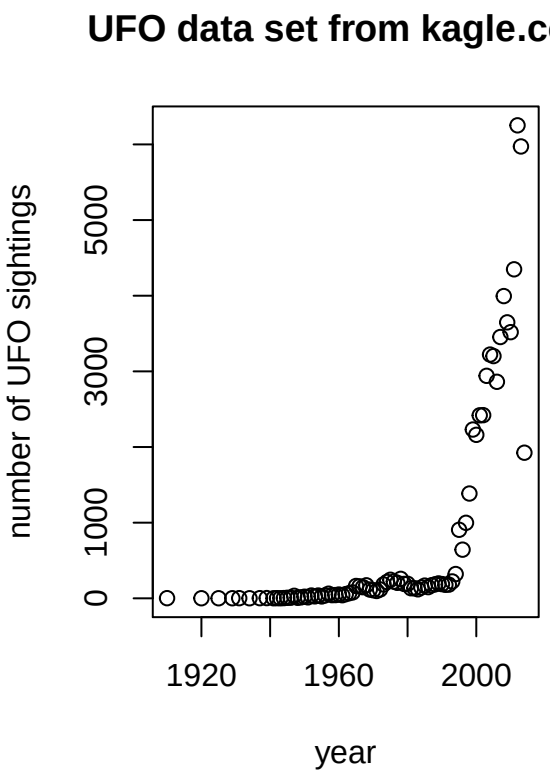
(6) ***Do you see any patterns in the number of sightings per year? How about as a function of the month?***

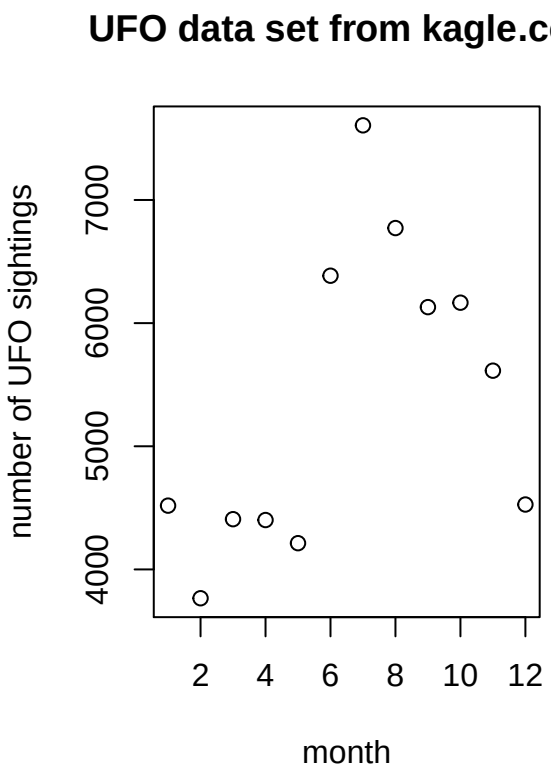
Rapidly increasing since about 1990. – This must be when the invasion started ;-). Many more sightings in summer months but also more in Fall than Spring.

```
countY<- table( ufo$year)
countM<- table( ufo$month)
set.panel(1,2)
```

```
## plot window will lay out plots in a 1 by 2 matrix
```

```
plot(names(countY ), as.numeric(countY), xlab="year", ylab="number of UFO sightings" )
title( "UFO data set from kagle.com")
plot(names(countM ), as.numeric(countM), xlab="month", ylab="number of UFO sightings" )
title( "UFO data set from kagle.com")
```





(7) ***Plot the locations of ufo sightings for the Rocky Mountain Region (Rectangle of longitudes of -113 to -100 and latitudes of 33 to 46).***

There is a choice here; I actually do not like axes on plots with a map so omitted them.

```
set.panel( 1,3)
```

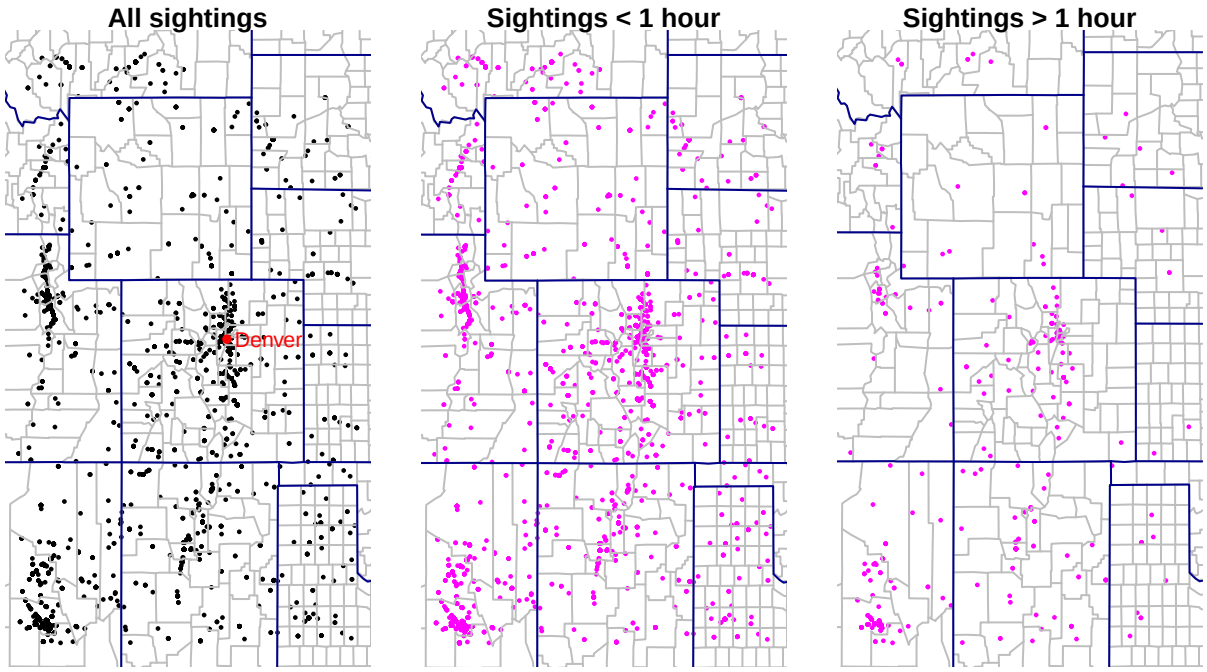
```
## plot window will lay out plots in a 1 by 3 matrix
```

```
par( mar=c( 8,2,1,0))
ind<- (ufo$longitude < -100) & (ufo$longitude > -113) &
                     (ufo$latitude < 46) & (ufo$latitude > 33 )
plot( ufo$longitude[ind],ufo$latitude[ind], pch=16, cex=.5, col="black",
      axes=FALSE, xlab="", ylab="")
map( "county", add=TRUE, col="grey")# add county boundaries
map( "state", add=TRUE, col="blue4")# and overlay state ones too.
points(-105,39.7,pch = 16, col="red")
text(-106,39.7,"Denver",col = "red", adj=-.5)
title("All sightings")

 # create a square plot so map looks right
ind2<-  ind & ufo$duration < 3600
plot( ufo$longitude[ind2],ufo$latitude[ind2], pch=16, cex=.5, col="magenta",
      axes=FALSE, xlab="", ylab="")
map( "county", add=TRUE, col="grey")# add county boundaries
map( "state", add=TRUE, col="blue4")# and overlay state ones too.
title("Sightings < 1 hour")

 # create a square plot so map looks right
ind2<-  ind & ufo$duration >= 3600
plot( ufo$longitude[ind2],ufo$latitude[ind2], pch=16, cex=.5, col="magenta",
```

```
      axes=FALSE, xlab="", ylab="")
map( "county", add=TRUE, col="grey")# add county boundaries
map( "state", add=TRUE, col="blue4")# and overlay state ones too.
title("Sightings > 1 hour")
```



(8) ***What kind of patterns do you see for this region? Do the patterns persist if you only look at the very long duration events (e.g. greater than 3600 seconds == 1 hour)?***

The sightings tend to concentrate around population centers. For example they follow the population density in the Front Range of Colorado. However, there are also some sights in remote areas. There does not seem to be any strong differences between the patterns based on duration of 1 hour.

(9) Generate some R code that converts the strings of dates in the **datetime** column to the numeric years as given in the **year** column.

```
library( lubridate) # only needed for first solution
# R elegant way
dates<- as.Date(ufo$datetime, format="%m/%d/%Y" )
# see also the function: parse_date_time which has better examples.
justYear<- as.numeric(year( dates))

# another equally good way
nLength<- nchar(ufo$datetime)
# days and months with single digits have a shorter strings -- a nightmare!
# luckily the time always takes up the last 4 spaces so we count from the end.
year<- substr( ufo$datetime, nLength -9, nLength - 6 )
justYear<- as.numeric( year)
table( justYear) #check that this is OK
```

```
justYear
1910 1920 1925 1929 1931 1934 1937 1939 1941 1942 1943 1944 1945 1946 1947
   2    1    1    1    2    1    2    3    1    2    1    3    7    8   33
1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962
   7   13   21   13   37   26   37   25   39   61   42   43   50   41   55
```

```
1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977
  68   78  163  158  147  172  116  107   97  117  183  216  245  221  205
1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992
 257  190  192  133  133  119  143  168  148  173  185  197  188  179  181
1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007
 223  321  906  643  998 1385 2231 2161 2419 2421 2941 3222 3202 2862 3454
2008 2009 2010 2011 2012 2013 2014
3995 3648 3518 4349 6252 5973 1925
```

```
# using scan this is essentially building your own date function.
work<- gsub("/", " ",ufo$datetime) # now dates and time separarted with white spaces
work2<- scan( text=work, what=" ") # reads this in as if from a file!
# examine this result
work2[1:20]
```

```
 [1] "10"    "10"    "1949"  "20:30" "10"    "10"    "1956"  "21:00"
 [9] "10"    "10"    "1961"  "19:00" "10"    "10"    "1965"  "23:45"
[17] "10"    "10"    "1966"  "20:00"
```

```
# we want every 4th string starting with the  3rd
ind<- seq( 3, length( work2),4)
year<- work2[ind]
justYear<- as.numeric( year)
```