Exploratory Data Analysis Problem Set 3

Robert Chen

December 13, 2015

Summary

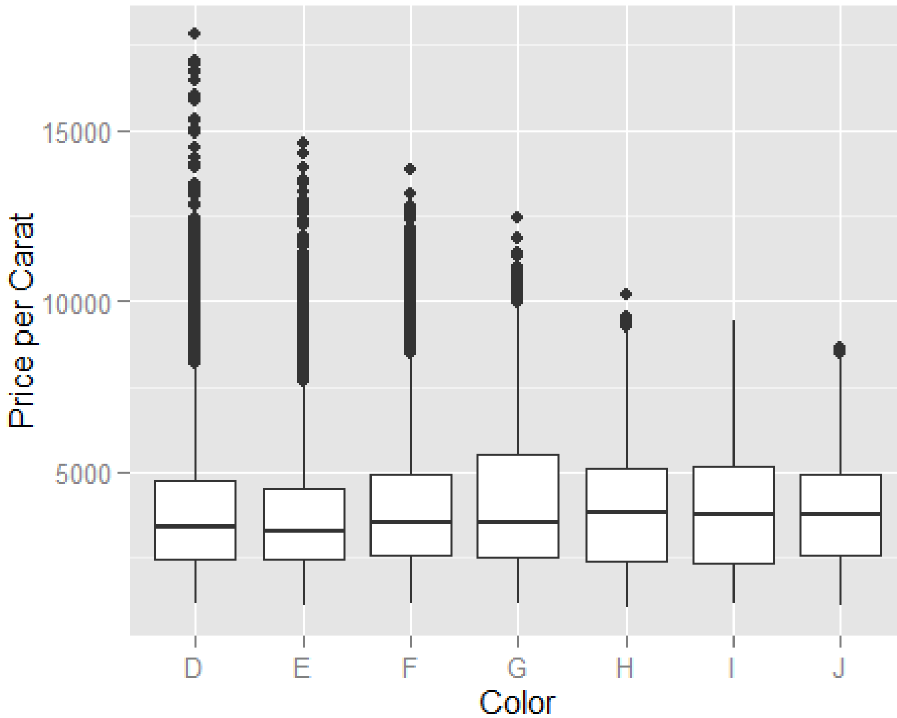
This is a summary of the code to complete Problem Set 3 of UD561 for the Exploratory Data Analysis Unit (Part 1) of the Sliderule "Fundamentals of Data Science" course.

Problem 1: Diamonds -- Price per Carat vs Color

The first part of the assignment is to explore price per carat vs color in the "diamonds" dataset using boxplots ("diamonds" comes with the library "ggplot2"). color ranges from D to J, with D being the best grade of color.

The code follows:

```r
library(ggplot2)
diamonds$ppc <- NA
diamonds$ppc <- diamonds$price / diamonds$carat
qplot(x = color, y = ppc, data = diamonds, geom="boxplot",
   xlab = "Color",
   ylab = "Price per Carat")
```



The first line loads the ggplot2 library (including the "diamonds" dataset). The next 2 lines create a price per carat variable. The third line creates the box plots.

This can also be sanity-checked using the "summary" command using the ppc variable across color:

```
      by(diamonds$ppc, diamonds$color, summary)

## diamonds$color: D
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1128    2455    3411    3953    4749   17830
## --------------------------------------------------------
## diamonds$color: E
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1078    2430    3254    3805    4508   14610
## --------------------------------------------------------
## diamonds$color: F
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1168    2587    3494    4135    4947   13860
## --------------------------------------------------------
## diamonds$color: G
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1139    2538    3490    4163    5500   12460
## --------------------------------------------------------
## diamonds$color: H
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1051    2397    3819    4008    5127   10190
```

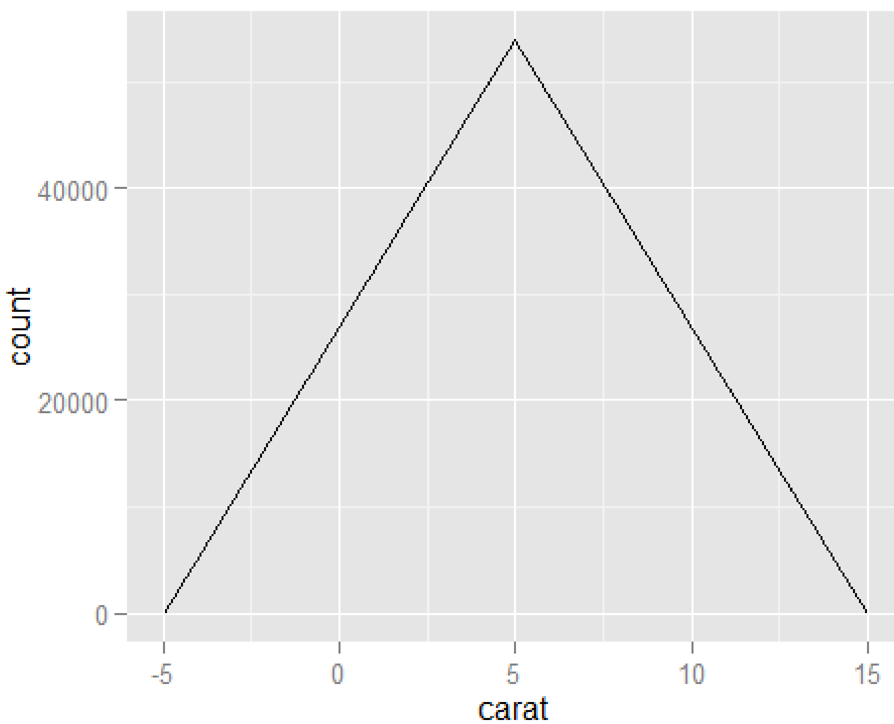
```
## ----------------------------------------------------------
## diamonds$color: I
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1152    2345    3780    3996    5197    9398
## ----------------------------------------------------------
## diamonds$color: J
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1081    2563    3780    3826    4928    8647
```

From the data, one can see that all colors roughly have the same median and middle quantiles, but the maximum value increases as the color grade gets better.

Problem 2: Diamonds -- Distribution of Carats

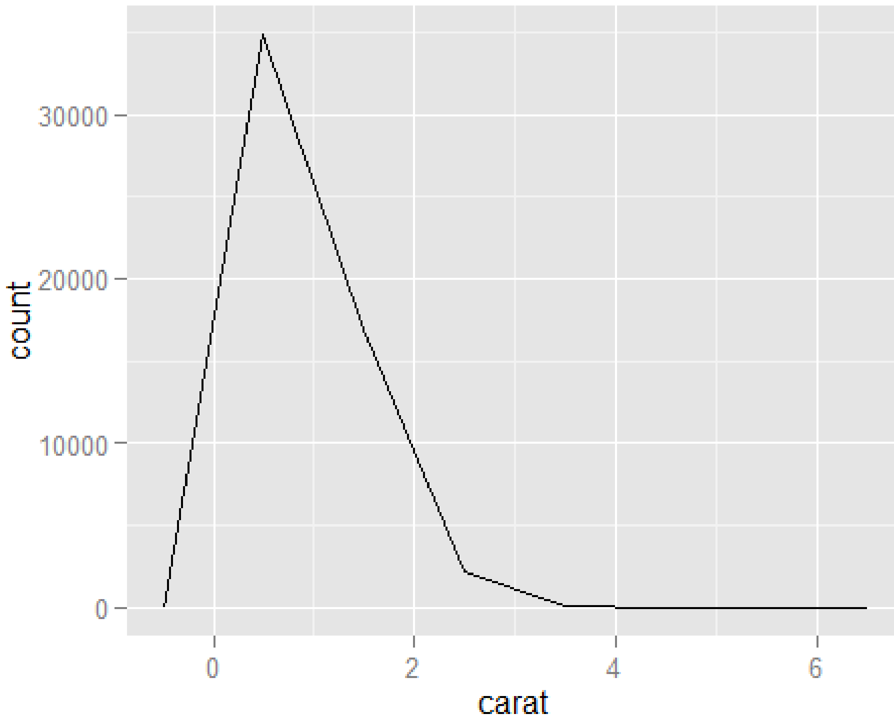
The second part is to explore weight (carats) using frequency polygons. First we try with a binwidth of 10:

```
qplot(x = carat, data = diamonds, binwidth = 10, geom="freqpoly")
```



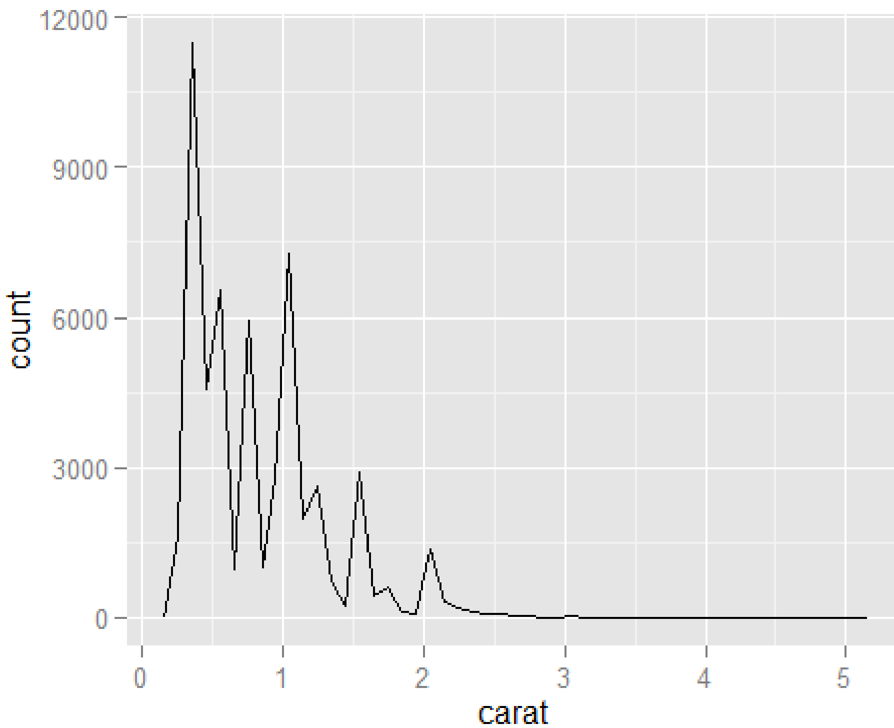
This data isn't very helpful. Let's try lowering the binwidth to 1:

```
qplot(x = carat, data = diamonds, binwidth = 1, geom="freqpoly")
```



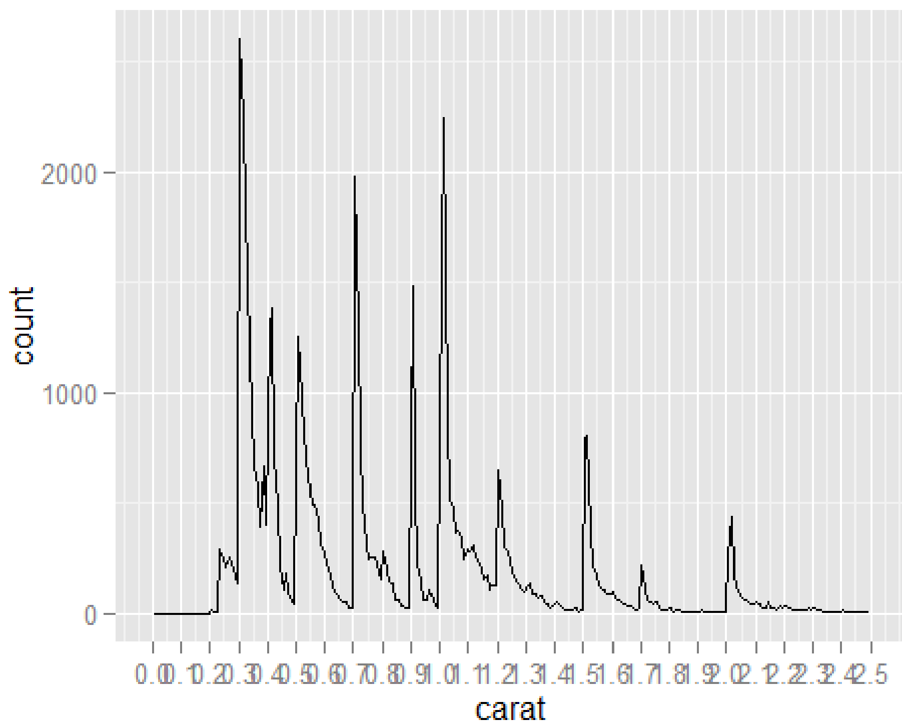
That's a little better, but it still seems like information is lost. Let's try to go by tenths:

```
qplot(x = carat, data = diamonds, binwidth = .1, geom="freqpoly")
```



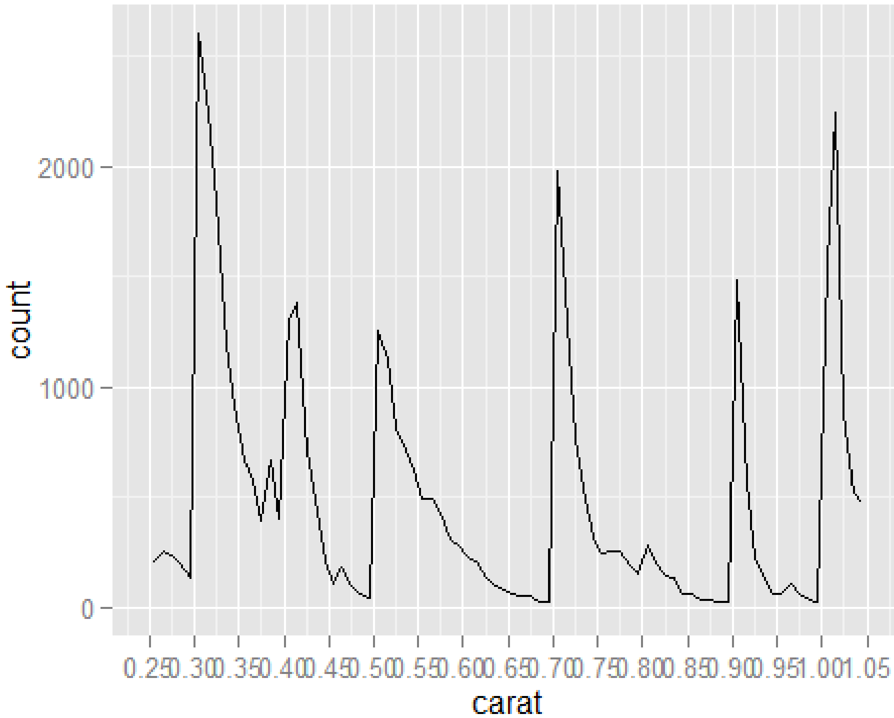
This is better. To better answer the question of which carats have quantities greater than 2000 we cause the x-axis values to increment by 0.1 and focus on the area between 0 and 2.5. We also decrease the binwidth to .01 to be able to check the value 1.01 carats:

```
    qplot(x = carat, data = diamonds, binwidth = .01, geom="freqpoly") +
    scale_x_continuous(lim = c(0, 2.5), breaks = seq(0, 2.5, .1))

## Warning: Removed 2 rows containing missing values (geom_path).
```



From this, one can see that values greater than 2000 start after .25 and stop after 1.05. We then zoom in on that portion and increase the resolution of the x-axis to 0.05:

```
    qplot(x = carat, data = diamonds, binwidth = .01, geom="freqpoly") +
    scale_x_continuous(lim = c(0.25, 1.05), breaks = seq(0.25, 1.05, .05))

## Warning: Removed 2 rows containing missing values (geom_path).
```



Roughly we see that 0.3 and 1.01 have values above 2000.

This can be confirmed numerically using the table command:

```r
table(diamonds$carat)
```

```
##
##  0.2 0.21 0.22 0.23 0.24 0.25 0.26 0.27 0.28 0.29  0.3 0.31 0.32 0.33 0.34
##   12    9    5  293  254  212  253  233  198  130 2604 2249 1840 1189  910
## 0.35 0.36 0.37 0.38 0.39  0.4 0.41 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49
##  667  572  394  670  398 1299 1382  706  488  212  110  178   99   63   45
##  0.5 0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59  0.6 0.61 0.62 0.63 0.64
## 1258 1127  817  709  625  496  492  430  310  282  228  204  135  102   80
## 0.65 0.66 0.67 0.68 0.69  0.7 0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79
##   65   48   48   25   26 1981 1294  764  492  322  249  251  251  187  155
##  0.8 0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89  0.9 0.91 0.92 0.93 0.94
##  284  200  140  131   64   62   34   31   23   21 1485  570  226  142   59
## 0.95 0.96 0.97 0.98 0.99    1 1.01 1.02 1.03 1.04 1.05 1.06 1.07 1.08 1.09
##   65  103   59   31   23 1558 2242  883  523  475  361  373  342  246  287
##  1.1 1.11 1.12 1.13 1.14 1.15 1.16 1.17 1.18 1.19  1.2 1.21 1.22 1.23 1.24
##  278  308  251  246  207  149  172  110  123  126  645  473  300  279  236
## 1.25 1.26 1.27 1.28 1.29  1.3 1.31 1.32 1.33 1.34 1.35 1.36 1.37 1.38 1.39
##  187  146  134  106  101  122  133   89   87   68   77   50   46   26   36
##  1.4 1.41 1.42 1.43 1.44 1.45 1.46 1.47 1.48 1.49  1.5 1.51 1.52 1.53 1.54
##   50   40   25   19   18   15   18   21    7   11  793  807  381  220  174
## 1.55 1.56 1.57 1.58 1.59  1.6 1.61 1.62 1.63 1.64 1.65 1.66 1.67 1.68 1.69
##  124  109  106   89   89   95   64   61   50   43   32   30   25   19   24
```

```
##  1.7 1.71 1.72 1.73 1.74 1.75 1.76 1.77 1.78 1.79  1.8 1.81 1.82 1.83 1.84 
##  215  119   57   52   40   50   28   17   12   15   21    9   13   18    4 
## 1.85 1.86 1.87 1.88 1.89  1.9 1.91 1.92 1.93 1.94 1.95 1.96 1.97 1.98 1.99 
##    3    9    7    4    4    7   12    2    6    3    3    4    4    5    3 
##    2 2.01 2.02 2.03 2.04 2.05 2.06 2.07 2.08 2.09  2.1 2.11 2.12 2.13 2.14 
##  265  440  177  122   86   67   60   50   41   45   52   43   25   21   48 
## 2.15 2.16 2.17 2.18 2.19  2.2 2.21 2.22 2.23 2.24 2.25 2.26 2.27 2.28 2.29 
##   22   25   18   31   22   32   23   27   13   16   18   15   12   20   17 
##  2.3 2.31 2.32 2.33 2.34 2.35 2.36 2.37 2.38 2.39  2.4 2.41 2.42 2.43 2.44 
##   21   13   16    9    5    7    8    6    8    7   13    5    8    6    4 
## 2.45 2.46 2.47 2.48 2.49  2.5 2.51 2.52 2.53 2.54 2.55 2.56 2.57 2.58 2.59 
##    4    3    3    9    3   17   17    9    8    9    3    3    3    3    1 
##  2.6 2.61 2.63 2.64 2.65 2.66 2.67 2.68  2.7 2.71 2.72 2.74 2.75 2.77  2.8 
##    3    3    3    1    1    3    1    2    1    1    3    3    2    1    2 
##    3 3.01 3.02 3.04 3.05 3.11 3.22 3.24  3.4  3.5 3.51 3.65 3.67    4 4.01 
##    8   14    1    2    1    1    1    1    1    1    1    1    1    1    2 
## 4.13  4.5 5.01 
##    1    1    1
```

Problem 3: Data Wrangling Review

The assignment is to review Garrett Grolemund's "Data Wrangling with R" slides. This was done.

Problem 4: Gapminder

The assignment here is to take a dataset from the Gapminder website (www.gapminder.org) and create 2-5 plots making use of the histogram / frequency polygon / boxplot techniques taught in this lesson.

The dataset used is total cell phones per country per year. 275 countries are listed, with cell phone usage for each country from 1965 to 2011.

The question I wanted to find out from this data was, what regions of the world show the most cell phone usage and show the largest increase in cell phone usage?

The first step was to add another column of data for the 275 entries called "region." The following regions were assigned:

```
1   North America
2   South America / Central America
3   Europe
4   Asia
5   Middle East
6   Africa
7   Russia and former Soviet Republics ("Central Asia")
8   Australia / Oceania
9   Carribean
```

The next step was to read in the data. So they wouldn't skew the results, any countries that showed no cell phone usage in 2011 were removed. Also, since the first cell phone usage numbers start to appear in 1980, only columns after 1980 are selected. Finally, column 1 was renamed to give a more meaningful name:

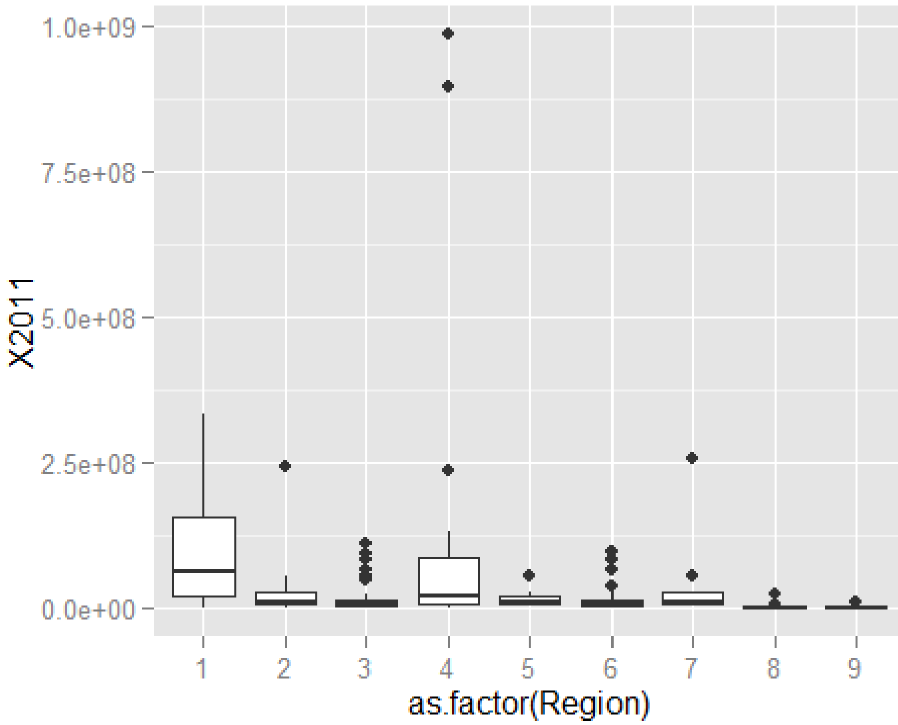
```r
# Set up libraries
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)

# Read data file
cp <- read.csv("cell_phone_total.csv", header = TRUE)

# Rename column 1 title
colnames(cp)[1] = "Country"

# Filter out any countries without 2011 data, any data before 1980
cpf <- cp %>% filter(!is.na(X2011))
cpf <- cpf %>% select(Country, Region,
                      starts_with("X198"), starts_with("X199"),
                      starts_with("X20") )
```

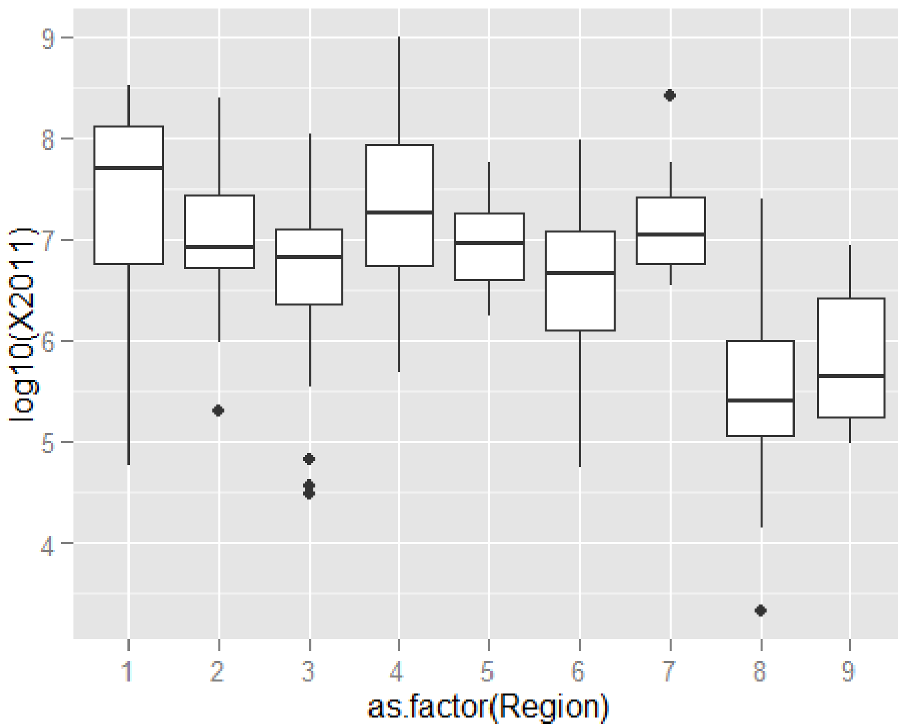
First we try a boxplot for each region of the total cell phone usage in the latest year (2011). To generate a valid boxplot, the "Region" column has to converted from a numeric to a factor:

```r
# Compare overall usage for most recent year for each region
qplot(x=as.factor(Region), y=X2011, data=cpf, geom="boxplot")
```



It is hard to interpret this data because of the very large scope of the data. We can get a slightly better sense by using log scales to get a sense of order-of-magnitude usage:

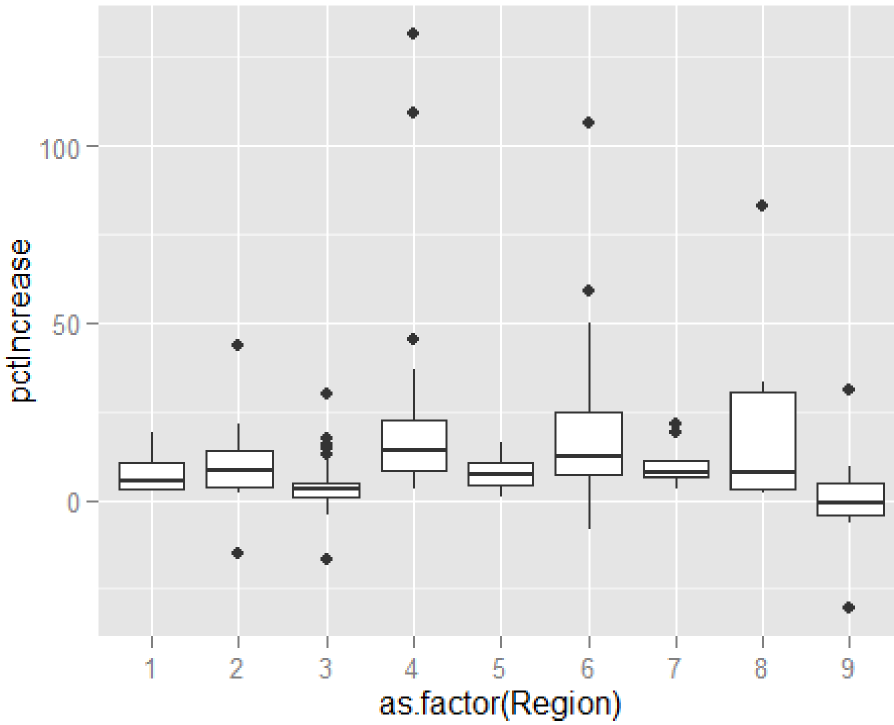
```
qplot(x=as.factor(Region), y=log10(X2011), data=cpf, geom="boxplot")
```



Here we get a slightly better sense of things. Looking at the medians, the region with the highest median use was North America (no surprise there), followed by Asia, with South/Central America, the Middle East, and (somewhat surprisingly) Russia with the same value, and Europe and Africa slightly behind them. Oceania and the Carribean are significantly lower (also not surprising). Looking at the overall ranges, Asia has the country with the highest usage, followed by North America and Russia.

A lot of this might reflect the difference in populations of the countries. To eliminate this, another column showing the year-on-year increase from 2010 to 2011 ("pctIncrease") was added and a new boxplot was generated:

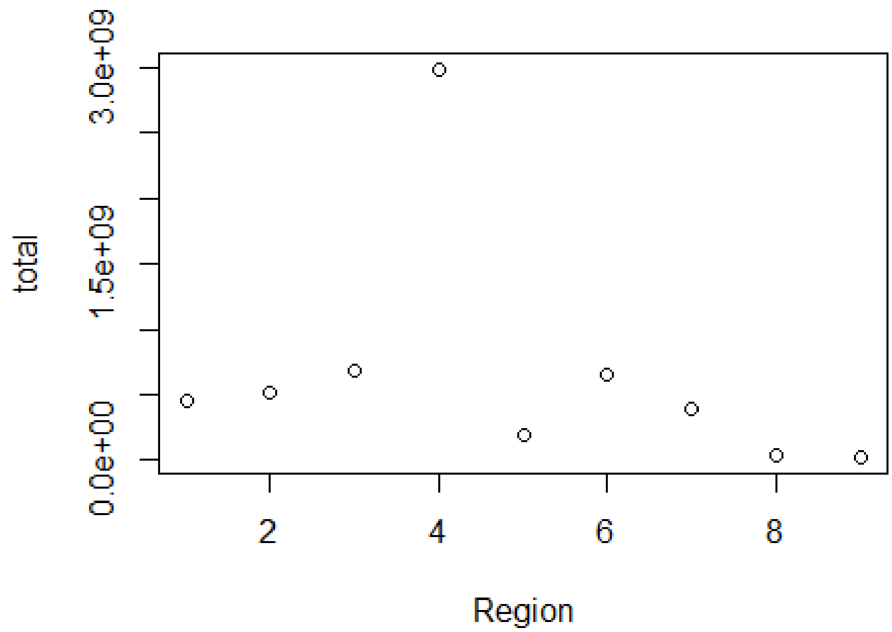
```
# Generate year-on-year increase for 2011 and do a boxplot by region
cpf <- mutate(cpf, pctIncrease = ((X2011 - X2010)/X2010)*100)
qplot(x=as.factor(Region), y=pctIncrease, data=cpf, geom="boxplot")

## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```



Here we see the median and middle quantiles grouped closer together in the 10-25% range, with Asia and Africa showing the largest median growth. The regions with the highest maximum values are Asia, Africa, and Australia.

We can also calculate the total for each region and do a simple plot:

```
# Calculate total usage by region
regions <- cpf %>% group_by(Region) %>%
summarise(total=sum(as.numeric(X2011)))
regions

## Source: local data frame [9 x 2]
##
##   Region      total
##    (int)      (dbl)
## 1      1  452082365
## 2      2  516274151
## 3      3  688307902
## 4      4 2983362972
## 5      5  188950217
## 6      6  653127967
## 7      7  390841464
## 8      8   34023961
## 9      9   24076904

plot(regions)
```

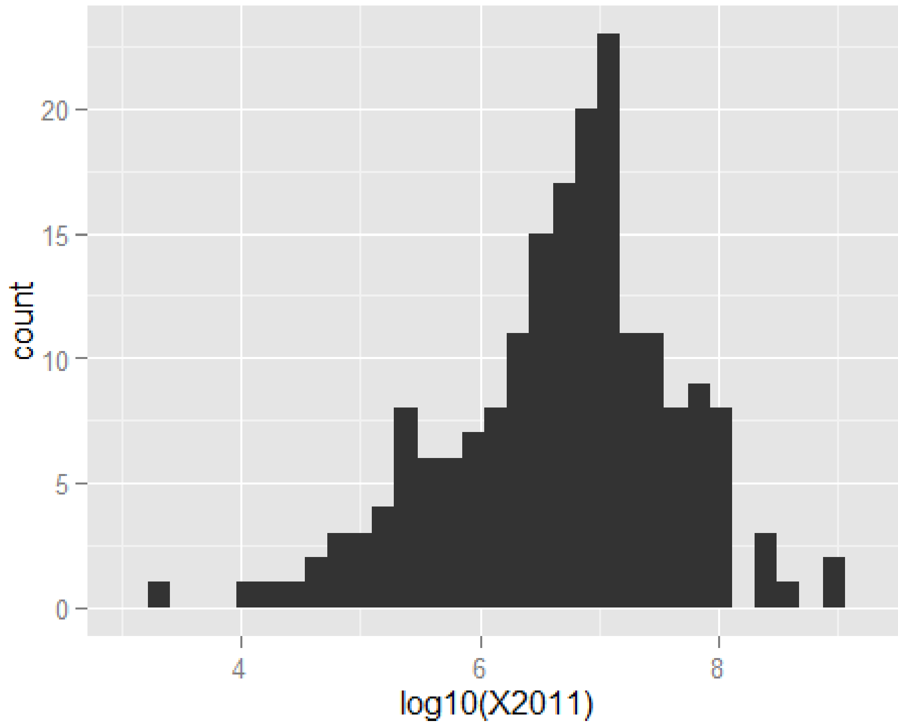


The as.numeric() was used to prevent an overflow for the value of Asia. One can see from the graph and the table that Asia has by far the most overall users (with it containing China and India that makes sense), followed by Europe and Africa, South and Central America, North America, and Russia. The Middle East, Oceania and the Carribean were a good deal behind. Two surprises were the relatively low ranking of North America (taking into account overall country populations that is a little less of a surprise), and the relatively high usage for Russia. Drilling down into the data, one can see that Russia itself has quite a few cell phone users (250m), with Ukraine contributing another 50m and Uzbekistan contributing another 25m. Given the population of Russia is around 143m, the 250m value is a particularly surprising number.

Finally, leaving out the regions, one can do a simple histogram to get a sense of the the distribution of cell phone usage among the countries:

```
qplot(x=log10(X2011), data=cpf)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust
this.
```



One can see that most of the countries fall in the 10^7 (10,000,000) range. The largest is just under 1 billion (China) and the smallest is in the 1000s (Tuvalu).

Birthdays

The task is to investigate my friends' birthdays on Facebook. The goal is to find out 2 things: Which day has the most friends' birthdays, and what month has the most friends' birthdays?

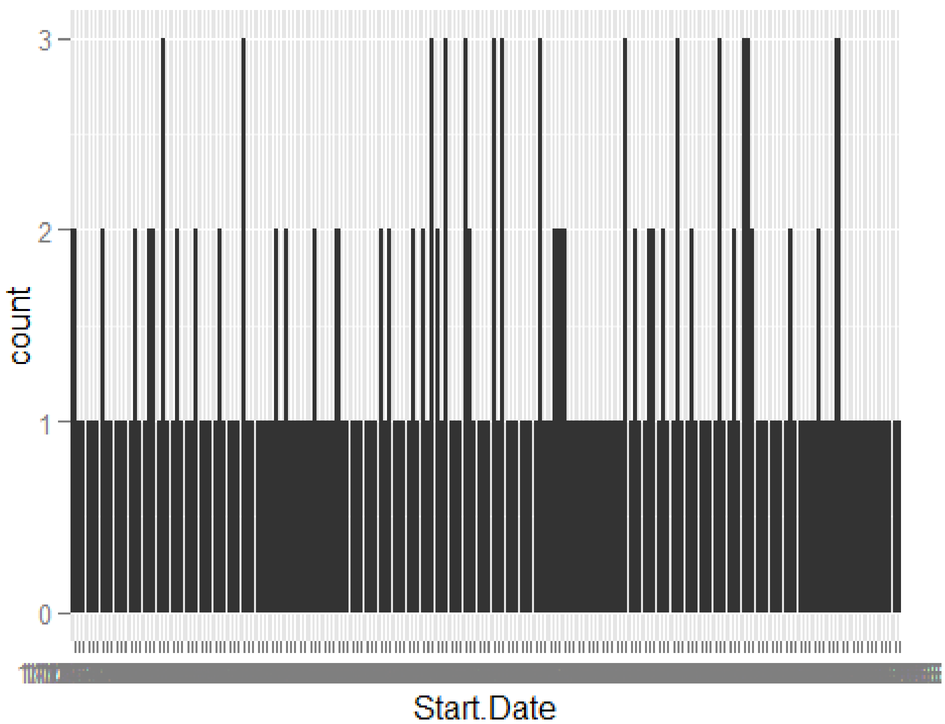
Data was extracted by exporting the birthdays from Facebook to an Outlook calendar. The "export" feature in Outlook was then used to output the results to a CSV file. This was then read into R using "read.csv":

```
# Set up libraries
#install.packages("lubridate")
library(dplyr)
library(ggplot2)
library(lubridate)

## Warning: package 'lubridate' was built under R version 3.2.3

# Read data file
bd <- read.csv("friends_birthdays.csv", header = TRUE)
```

"lubridate" will be explained later.

To get a sense of which day has the most birthdays, first we do a simple plot of the date, where the date column is currently treated as a factor:

```
# Find birthday most commonly used
qplot(x = Start.Date, data=bd, binwidth = 1)
```



From this we see there is a 14-way tie for first, each having 3 friends with that birthday. To find out those days we can do a group_by() followed by a summarise() to get the count per day and then a filter() to only show those days with a value of 3:

```
bd %>% group_by(Start.Date) %>% summarise(num_friends = n()) %>%
  filter(num_friends == 3)

## Source: local data frame [14 x 2]
##
##    Start.Date num_friends
##        (fctr)       (int)
## 1  10/15/2016           3
## 2  11/21/2016           3
## 3    2/5/2016           3
## 4    3/1/2016           3
## 5   3/18/2016           3
## 6   3/25/2016           3
## 7   3/29/2016           3
## 8   4/16/2016           3
## 9   5/27/2016           3
## 10  6/20/2016           3
```

```
## 11  7/12/2016          3
## 12  7/20/2016          3
## 13  7/21/2016          3
## 14  8/27/2016          3
```

Finding the distribution by month is a little more involved. We use the "lubridate" package recommended by the Udacity team to process the date column. This library has already been installed and loaded from previous commands. We then use the "mdy()" function in lubridate to convert the "date" column that was formerly factors into official R dates (mdy means that the factor data is stored as month/day/year):

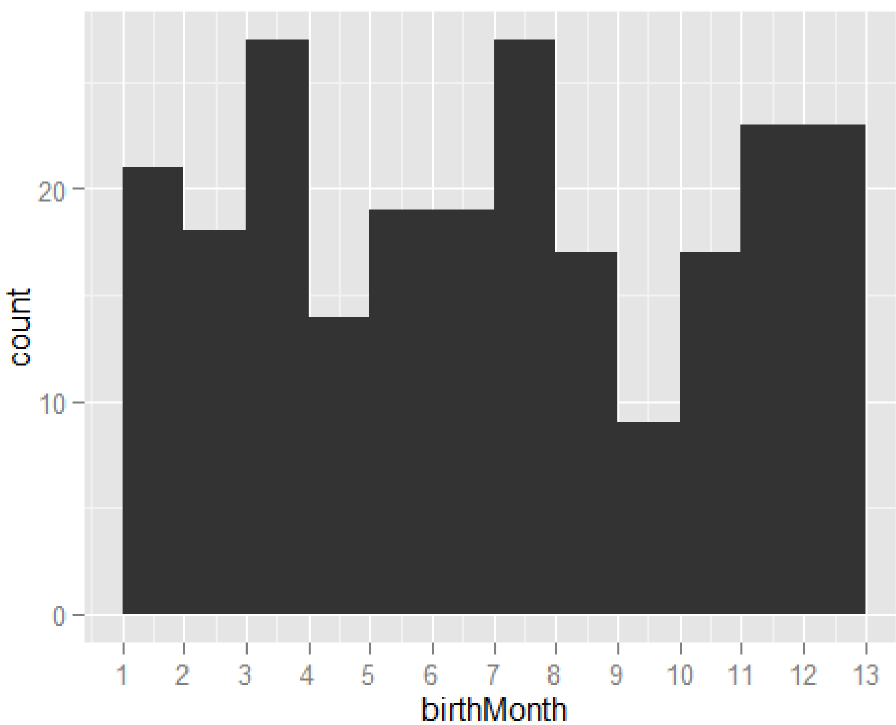
```
# Add month column
bd[,2] <- mdy(bd[,2])
```

One we do this, we can then use the month() function to extract the month:

```
bd <- mutate(bd, birthMonth = month(Start.Date))
```

And we can then do a histogram by month:

```
qplot(x = birthMonth, data=bd, binwidth = 1) +
  scale_x_continuous(limit = c(1, 13), breaks = seq(1, 13, 1))
```



From this we see March and July have the most friends' birthdays, with 24 friends each.