# ProblemSet 3

Ricky Lim

November 19, 2014

## 1 Dataset

```
library(ggplot2)
data(diamonds)
head(diamonds)

##   carat       cut color clarity depth table price    x    y    z
## 1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
## 2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
## 3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31
## 4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
## 5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
## 6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48

summary(diamonds)

##      carat               cut        color       clarity          depth
##  Min.   :0.2000   Fair     : 1610   D: 6775   SI1    :13065   Min.   :43.00
##  1st Qu.:0.4000   Good     : 4906   E: 9797   VS2    :12258   1st Qu.:61.00
##  Median :0.7000   Very Good:12082   F: 9542   SI2    : 9194   Median :61.80
##  Mean   :0.7979   Premium  :13791   G:11292   VS1    : 8171   Mean   :61.75
##  3rd Qu.:1.0400   Ideal    :21551   H: 8304   VVS2   : 5066   3rd Qu.:62.50
##  Max.   :5.0100                     I: 5422   VVS1   : 3655   Max.   :79.00
##                                     J: 2808   (Other): 2531
##      table           price             x                y                z
##  Min.   :43.00   Min.   :  326   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
##  Median :57.00   Median : 2401   Median : 5.700   Median : 5.710   Median : 3.530
##  Mean   :57.46   Mean   : 3933   Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
##  3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
##  Max.   :95.00   Max.   :18823   Max.   :10.740   Max.   :58.900   Max.   :31.800
##

dim(diamonds)

## [1] 53940    10

names(diamonds)

##  [1] "carat"   "cut"     "color"   "clarity" "depth"   "table"   "price"   "x"
##  [9] "y"       "z"

class(diamonds$color)
```

```
## [1] "ordered" "factor"

class(diamonds$cut)

## [1] "ordered" "factor"

class(diamonds$clarity)

## [1] "ordered" "factor"

summary(diamonds$color)

##     D     E     F     G     H     I     J
##  6775  9797  9542 11292  8304  5422  2808
```

## 1.1 Price Histogram

```
summary(diamonds$price)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     326     950    2401    3933    5324   18820

p <- ggplot(aes(x=price), data=diamonds)+
        geom_histogram(binwidth=100)+
        scale_x_continuous(breaks=seq(100,20000, 500), limits=c(300, 10000))
p

p + facet_wrap(~cut, ncol=1, scales='free_y')


nrow(subset(diamonds, price < 500))

## [1] 1729

head(subset(diamonds, price < 500))

##   carat       cut color clarity depth table price    x    y    z
## 1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
## 2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
## 3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31
## 4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
## 5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
## 6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48

nrow(subset(diamonds, price < 250))

## [1] 0

nrow(subset(diamonds, price >= 15000))

## [1] 1656

head(subset(diamonds, price >= 15000))
```

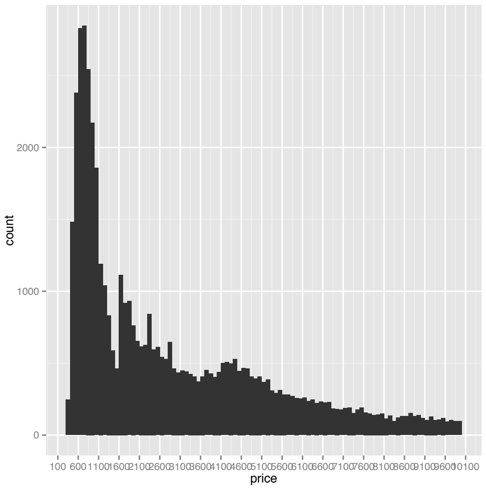
```
##       carat       cut color clarity depth table price    x    y    z
## 25885  1.60     Ideal     G     VS2  61.9    56 15000 7.53 7.47 4.64
## 25886  1.54   Premium     E     VS2  62.3    58 15002 7.31 7.39 4.58
## 25887  1.19     Ideal     F    VVS1  61.5    55 15005 6.82 6.84 4.20
## 25888  2.10   Premium     I     SI1  61.5    57 15007 8.25 8.21 5.06
## 25889  1.69     Ideal     D     SI1  60.8    57 15011 7.69 7.71 4.68
## 25890  1.50 Very Good     G    VVS2  62.9    56 15013 7.22 7.32 4.57
```

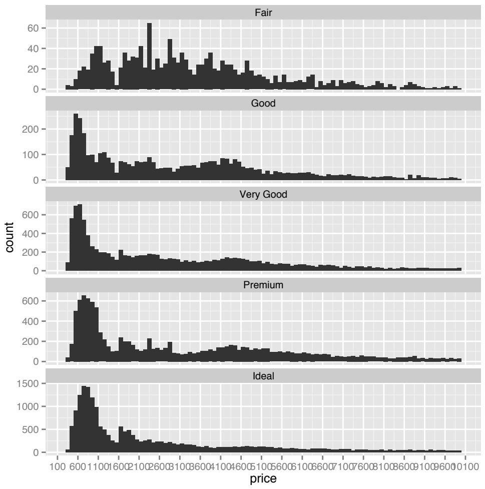
```
# cut prices
by(diamonds$price, diamonds$cut, summary)
```

```
## diamonds$cut: Fair
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     337    2050    3282    4359    5206   18570
## -------------------------------------------------------------------
## diamonds$cut: Good
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     327    1145    3050    3929    5028   18790
## -------------------------------------------------------------------
## diamonds$cut: Very Good
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     336     912    2648    3982    5373   18820
## -------------------------------------------------------------------
## diamonds$cut: Premium
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     326    1046    3185    4584    6296   18820
## -------------------------------------------------------------------
## diamonds$cut: Ideal
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     326     878    1810    3458    4678   18810
```

```
by(diamonds$price, diamonds$cut, max)
```

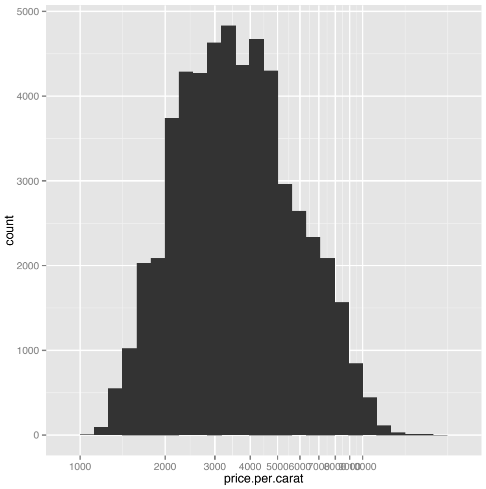
```
## diamonds$cut: Fair
## [1] 18574
## -------------------------------------------------------------------
## diamonds$cut: Good
## [1] 18788
## -------------------------------------------------------------------
## diamonds$cut: Very Good
## [1] 18818
## -------------------------------------------------------------------
## diamonds$cut: Premium
## [1] 18823
## -------------------------------------------------------------------
## diamonds$cut: Ideal
## [1] 18806
```

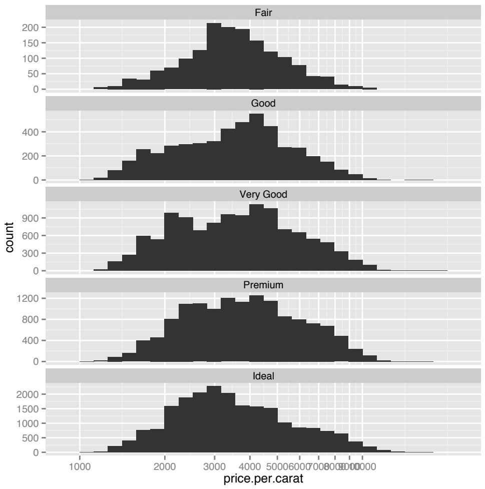




## 1.2 Price per Carat Histogram

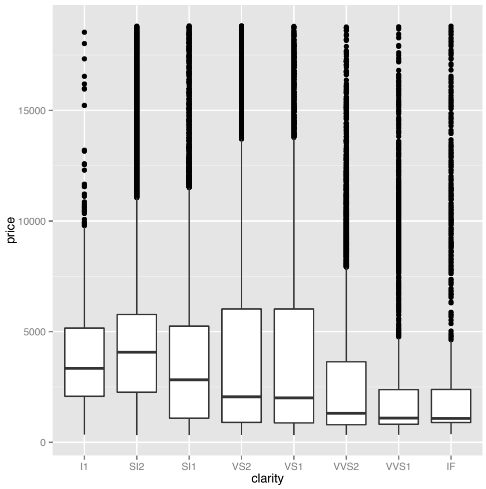
```
head(diamonds)

##   carat       cut color clarity depth table price    x    y    z
## 1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
## 2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
## 3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31
## 4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
## 5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
## 6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48

summary(log10(diamonds$price.per.carat))

## Error in log10(diamonds$price.per.carat):  non-numeric argument to mathematical function

diamonds$price.per.carat <- diamonds$price/diamonds$carat
p <- ggplot(aes(x=price.per.carat), data=diamonds)+
        geom_histogram(binwidth=0.05)+
        scale_x_log10(breaks=seq(1000,10000, 1000))
        #scale_x_continuous(breaks=seq(100,20000, 500), limits=c(300, 10000))
p

p + facet_wrap(~cut, ncol=1, scales='free_y')
```





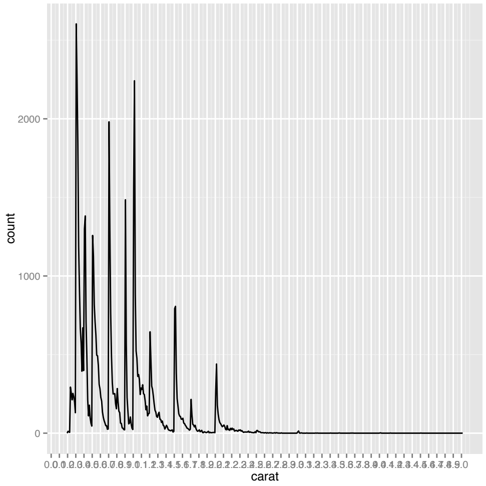
## 1.3 Price BoxPlots

```
p <- ggplot(aes(x=clarity, y=price), data=diamonds) +
      geom_boxplot()
p
head(diamonds)

##   carat       cut color clarity depth table price    x    y    z price.per.carat
## 1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43        1417.391
## 2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31        1552.381
## 3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31        1421.739
## 4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63        1151.724
## 5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75        1080.645
## 6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48        1400.000

head(subset(diamonds, color=='D'))

##    carat       cut color clarity depth table price    x    y    z price.per.carat
## 29  0.23 Very Good     D     VS2  60.5    61   357 3.96 3.97 2.40        1552.174
## 35  0.23 Very Good     D     VS1  61.9    58   402 3.92 3.96 2.44        1747.826
## 39  0.26 Very Good     D     VS2  60.8    59   403 4.13 4.16 2.52        1550.000
## 43  0.26      Good     D     VS2  65.2    56   403 3.99 4.02 2.61        1550.000
## 44  0.26      Good     D     VS1  58.4    63   403 4.19 4.24 2.46        1550.000
## 55  0.22   Premium     D     VS2  59.3    62   404 3.91 3.88 2.31        1836.364

summary(subset(diamonds, color=='D')$price)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     357     911    1838    3170    4214   18690

summary(subset(diamonds, color=='J')$price)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     335    1860    4234    5324    7695   18710

IQR(subset(diamonds, color=='D')$price) # the best color

## [1] 3302.5

IQR(subset(diamonds, color=='J')$price) # the worst color

## [1] 5834.5
```



### 1.4 Carat Frequency Polygon

```
summary(diamonds$carat)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2000  0.4000  0.7000  0.7979  1.0400  5.0100

p <- ggplot(aes(x=carat), data=diamonds) +
        geom_freqpoly(binwidth=0.01)+
        scale_x_continuous(breaks=seq(0,5,0.1))

p
```



## 2 Birthday

Questions

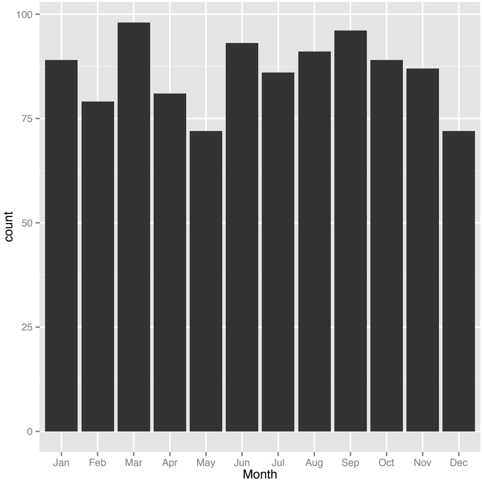
- Which month contains the most number of birthdays?
- How many birthdays are in each month?

- Which day of the year has the most number of birthdays?
- Do you have at least 365 friends that have birthdays on everyday of the year?

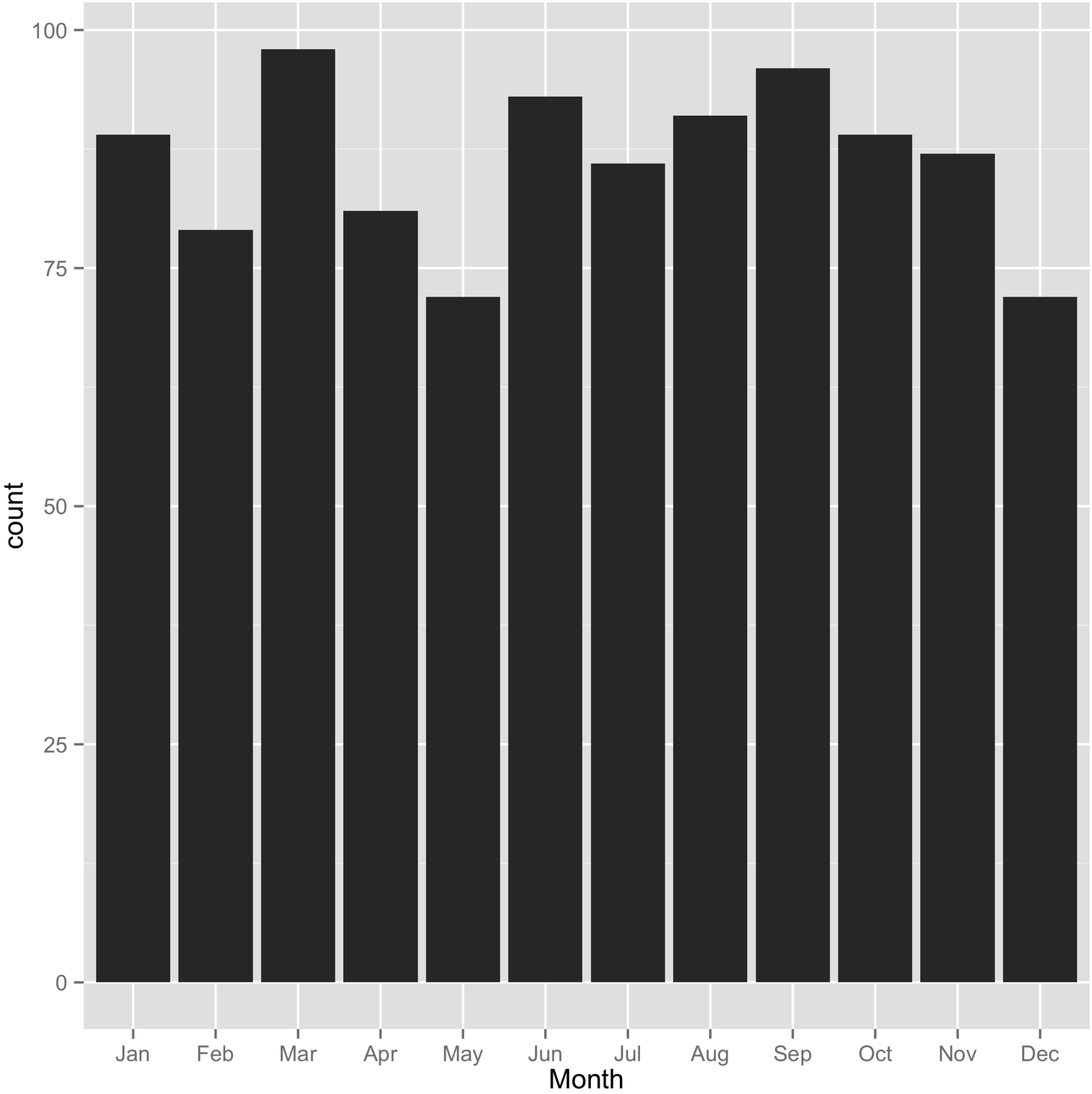
```
library(ggplot2)
library(lubridate)
work_dir='/Users/RickyLim/Documents/OnlineLearning/DataAnalysisR/'
birthdays <- read.csv(paste0(work_dir, 'Data/birthdaysExample.csv'),
                      header=TRUE)
dim(birthdays)

## [1] 1033    1

head(birthdays)

##      dates
## 1 11/25/14
## 2   6/8/14
## 3  9/12/14
## 4  5/26/14
## 5  2/20/14
## 6  6/19/14

tail(birthdays)

##         dates
## 1028  3/22/14
## 1029  3/29/14
## 1030  8/26/14
## 1031 12/28/14
## 1032  9/27/14
## 1033  8/26/14

birthdays$Date <- as.Date(birthdays$dates,format='%m/%d/%y')
birthdays$Month <- as.numeric(format(birthdays$Date, '%m'))
birthdays$Day <- as.numeric(format(birthdays$Date, '%d'))
birthdays$Year<- as.numeric(format(birthdays$Date, '%y'))

birthdays <- subset(birthdays, select=c(Date, Day, Month, Year))
birthdays$Month<- factor(birthdays$Month,levels=as.character(1:12),
                         labels=c("Jan","Feb","Mar","Apr","May","Jun",
                                  "Jul","Aug","Sep","Oct","Nov","Dec"),
                         ordered=TRUE)
```

### 2.1 Which month contains the most number of birthdays?

```
head(birthdays)

##         Date Day Month Year
## 1 2014-11-25  25   Nov   14
## 2 2014-06-08   8   Jun   14
## 3 2014-09-12  12   Sep   14
## 4 2014-05-26  26   May   14
## 5 2014-02-20  20   Feb   14
## 6 2014-06-19  19   Jun   14

p <- ggplot(aes(x=Month), data=birthdays) +
        geom_histogram() +
        scale_x_discrete()
p

ggsave('figs/Month_bod.png', p)

## Saving 7 x 7 in image
```



March is the most number of birthdays.


100
75
50
25
0
count
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month

## 2.2 How many birthdays are in each month?

| | Month | Freq |
|---|---|---|
| 1 | Jan | 89 |
| 2 | Feb | 79 |
| 3 | Mar | 98 |
| 4 | Apr | 81 |
| 5 | May | 72 |
| 6 | Jun | 93 |
| 7 | Jul | 86 |
| 8 | Aug | 91 |
| 9 | Sep | 96 |
| 10 | Oct | 89 |
| 11 | Nov | 87 |
| 12 | Dec | 72 |

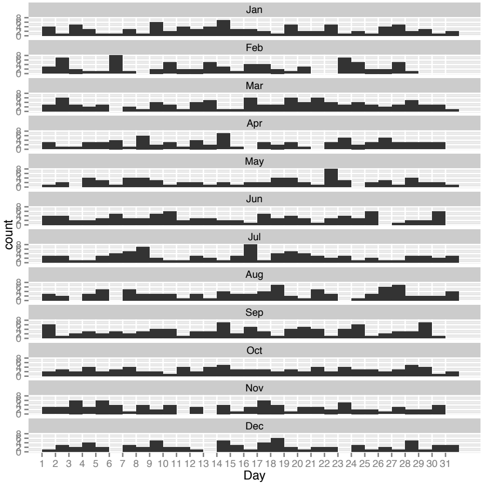
## 2.3 Which day of the year has the most number of birthdays?

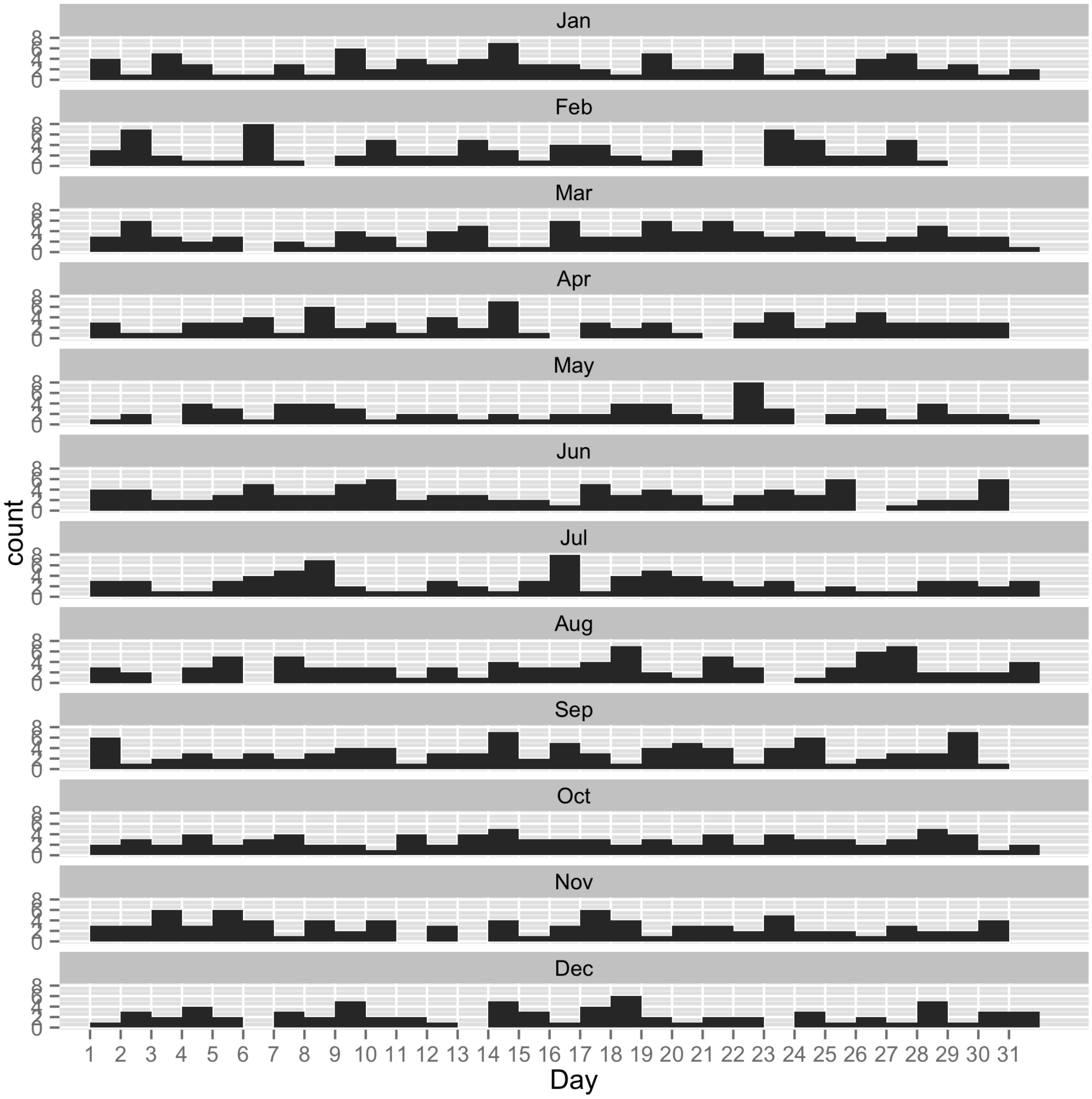
```
Day_bod <- as.data.frame(table(birthdays$Day))
colnames(Day_bod) <- c('Day', 'Freq')
subset(Day_bod, Freq == max(Day_bod$Freq))

##    Day Freq
## 14  14   48
```

14 is the day of the year that has the most number of birthdays.

## 2.4 Do you have at least 365 friends that have birthdays on everyday of the year?

```
p <- ggplot(aes(x=Day), data=birthdays) +
        geom_histogram() +
        scale_x_discrete(breaks=1:31)+
        facet_wrap(~Month, ncol=1)
p

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

ggsave('figs/Day_bod.png', p)

## Saving 7 x 7 in image
## stat_bin:  binwidth defaulted to range/30.  Use ’binwidth = x’ to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```




Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
count
0
2
4
6
8
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Day

No, as some days in several months, such as 13 Dec, 6 Dec, and so on.

```
Filename: problemSet3.Rnw
Working directory: /Users/RickyLim/Documents/OnlineLearning/DataAnalysisR/Codes/ProblemSet3
```

## 3 Metainfo

```
sessionInfo()

## R version 3.1.1 (2014-07-10)
## Platform: x86_64-apple-darwin13.3.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] xtable_1.7-4    lubridate_1.3.3 ggplot2_1.0.0   knitr_1.7
##
## loaded via a namespace (and not attached):
##  [1] Cairo_1.5-6      codetools_0.2-8  colorspace_1.2-4 digest_0.6.4     evaluate_0.5.5
##  [6] formatR_1.0      grid_3.1.1       gtable_0.1.2     highr_0.3        labeling_0.3
## [11] MASS_7.3-33      memoise_0.2.1    munsell_0.4.2    plyr_1.8.1       proto_0.3-10
## [16] Rcpp_0.11.2      reshape2_1.4     scales_0.2.4     stringr_0.6.2    tools_3.1.1
```

```
library(knitr)
knit("problemSet3.Rnw" ) # compile to tex

##
##
## processing file:  problemSet3.Rnw
## Error in parse_block(g, patterns):  duplicate label 'setup'

purl("problemSet3.Rnw", documentation = 0) # extract R code only

##
##
## processing file:  problemSet3.Rnw
## Error in parse_block(g, patterns):  duplicate label 'setup'

knit2pdf("problemSet3.Rnw")

##
##
## processing file:  problemSet3.Rnw
## Error in parse_block(g, patterns):  duplicate label 'setup'
```