Data Exploration: Problem Set 5

Robert Chen

December 16, 2015

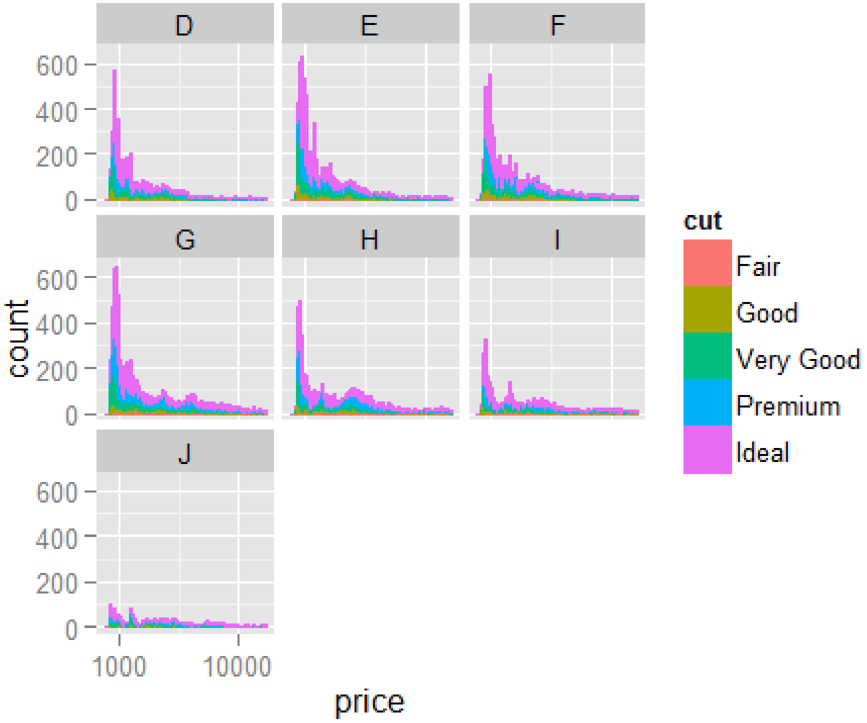
Summary

This is a summary of the code to complete Problem Set 5 of UD561 for the Exploratory Data Analysis Unit (Part 3) of Sliderule's Fundamentals of Data Science course.

Problem 1a: Prices, Cut, and Color Histogram

The assignment is to create a histogram of prices in "diamond", faceting the histogram by color and using cut to color the bars. This is done below:

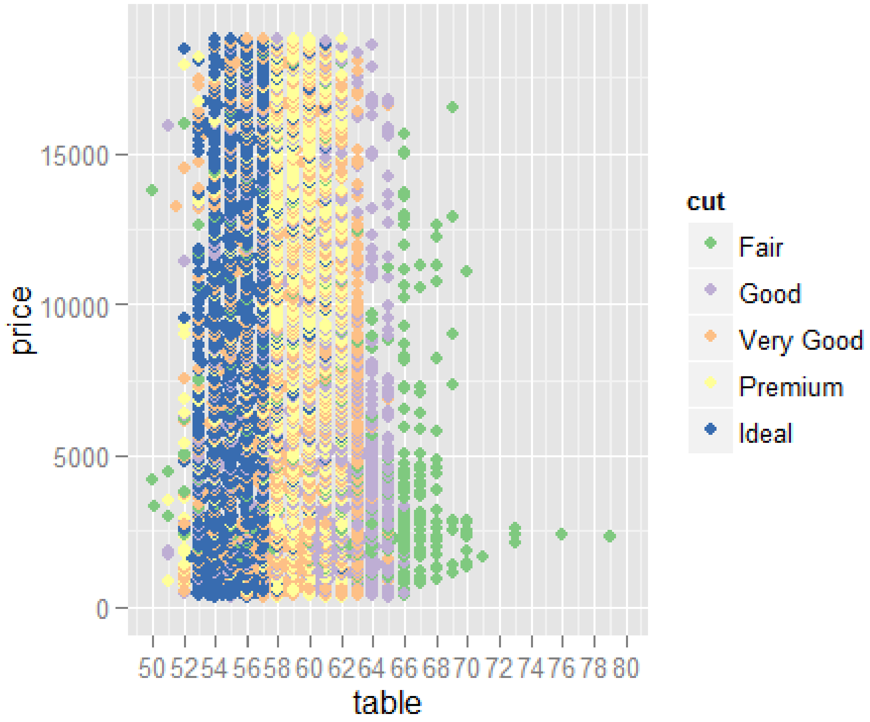
```
library(ggplot2)
ggplot(aes(x = price, color = cut), data = diamonds) +
  geom_histogram(aes(color = cut, fill = cut), binwidth = 100) +
  facet_wrap(~ color) +
  scale_x_continuous(breaks = c(1000, 10000), lim = c(0, 12000))
```



Problem 1b: Prices, Table, and Color Scatterplot

The assignment is to create a scatterplot of prices in "diamond" versus table, using cut to color the points. This is done below:

```
ggplot(aes(x = table, y = price), data = diamonds) +
  geom_point(aes(color = cut)) +
  scale_x_continuous(breaks = seq(50, 80, 2), lim = c(50, 80)) +
  scale_y_continuous(lim = c(0, 19000)) +
  scale_color_brewer(type = 'qual')

## Warning: Removed 5 rows containing missing values (geom_point).
```

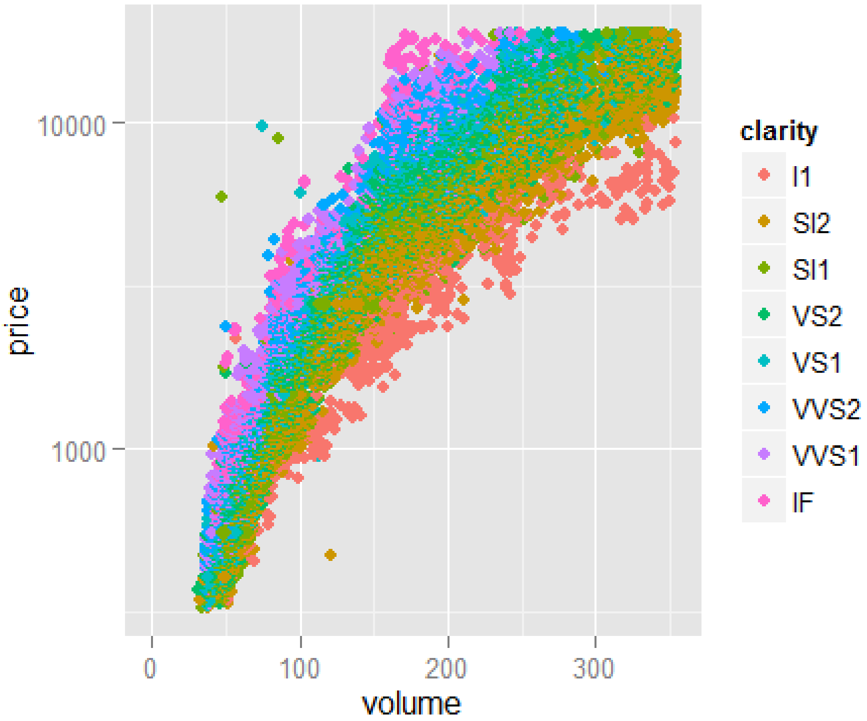


From the plot one can see that the typical table range for diamonds with an ideal cut is 53 to 57. The typical table range for diamonds with a premium cut is 58 to 62.

Problem 1c: Prices, Volume, and Clarity Scatterplot

The assignment is to create a scatterplot of prices in "diamond" versus volume (where volume is a new variable created by "volume = x * y * z"), and to use clarity to color the points. Also, a log10 scale should be used on the y-axis and the top 1% of volumes should be omitted. This is done with the following code:

```
diamonds$volume <- diamonds$x * diamonds$y * diamonds$z
ggplot(aes(x = volume, y = price), data = subset(diamonds, volume > 0)) +
  geom_point(aes(color = clarity)) +
  scale_x_continuous(lim = c(0, quantile(diamonds$volume, .99))) +
  scale_y_log10()

## Warning: Removed 540 rows containing missing values (geom_point).
```



Problem 2a: Proportion of Friendships Initiated

The assignment is use the pseudo-Facebook data and create a variable, "prop_initated", that consists of friendships_initiated divided by friend_count. This is done straightforwardly through the following code:

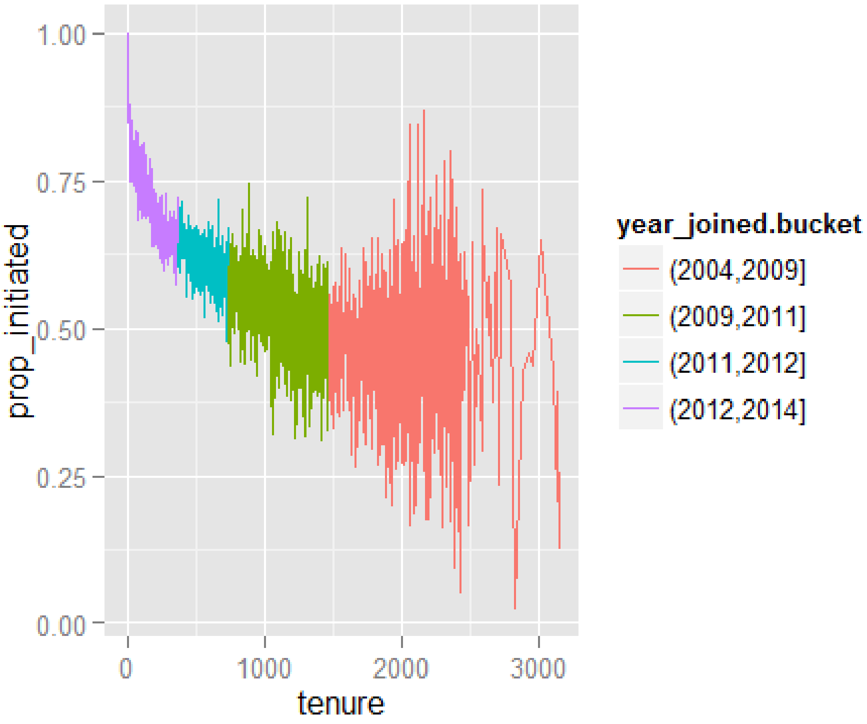
```
pf <- read.csv('pseudo_facebook.tsv', sep = '\t')
pf$prop_initiated <- pf$friendships_initiated / pf$friend_count
```

Problem 2b: Proportion of Friendships Initiated, Tenure and Year Joined

The assignment is to recreate "year_joined.bucket" from the lesson and then create a line graph containing the median of prop_initiated (created in Problem 2a) vs tenure, using the buckets to determine the line color:

```
pf$year_joined <- trunc(2014 - pf$tenure / 365)
pf$year_joined.bucket <- cut(pf$year_joined,
                             breaks=c(2004, 2009, 2011, 2012, 2014))
ggplot(aes(x = tenure, y = prop_initiated),
       data = subset(pf, !is.na(prop_initiated))) +
  geom_line(aes(color = year_joined.bucket), stat = 'summary', fun.y =
median)
```

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



"trunc" is used to cut off the decimal value in the division, and "cut" is used to create the buckets.

In the ggplot command, "stat" and "fun.y" in "geom_line" are used to define the fact that we want to take the median of our newly defined prop_initiated variable for each day of tenure.

Problem 2c: Smoothing the graph

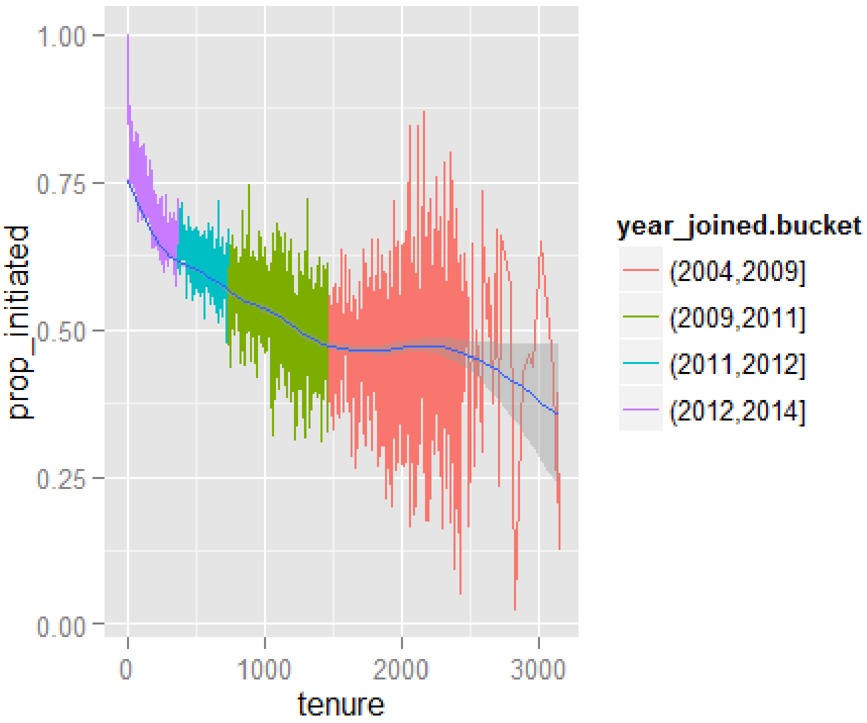
The assignment is to take the graph in 2b and smooth it. This is done using the geom_smooth() command:

```
ggplot(aes(x = tenure, y = prop_initiated),
       data = subset(pf, !is.na(prop_initiated))) +
  geom_line(aes(color = year_joined.bucket), stat = 'summary',
            fun.y = median) +
  geom_smooth()
```

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

```
## geom_smooth: method="auto" and size of largest group is >=1000, so using
gam with formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the
smoothing method.
```

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



From this, one can see that the group that initiated the greatest proportion of its Facebook friendships was people who joined after 2012.

One can find the mean proportion for this group using "group_by()", "filter()", and "summarise" in dplyr:

```
library(dplyr)

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

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

```
pf %>% group_by(year_joined.bucket) %>%
       filter(year_joined.bucket == '(2012,2014]') %>%
       summarise(mean_prop_initiated = mean(prop_initiated, na.rm = TRUE))

## Source: local data frame [1 x 2]
##
##   year_joined.bucket mean_prop_initiated
##               (fctr)               (dbl)
## 1        (2012,2014]           0.6653892
```

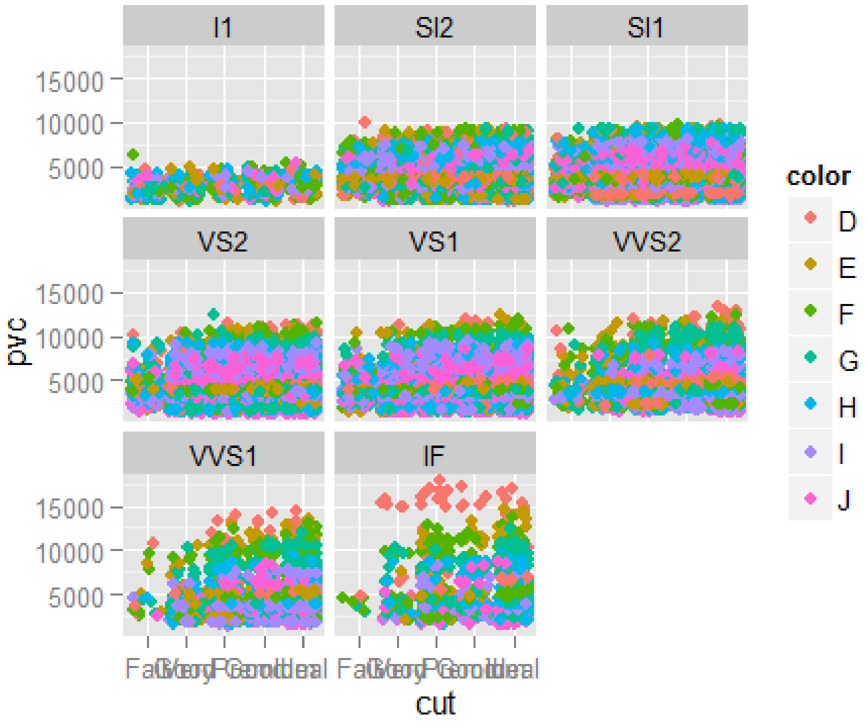
The mean proportion is .6653892. This group likely has the highest proportion of friendships initiated because they just recently started. When they start, they typically will send a lot of e-mail requests to friends they know -- so they will end up in most cases initiating the majority of friendship requests. After that the number they initiate vs the number initiated due to friends finding them is likely to even out more.

Problem 3: Price/Carat ratio, Cut, Color, and Clarity

The assignment is create a scatterplot of price/carat vs cut, with color determining the color of each point, faceted by clarity.

This is done using the following commands:

```
diamonds$pvc <- diamonds$price / diamonds$carat
ggplot(aes(x = cut, y = pvc, color = color), data = diamonds) +
  geom_point(aes(color = color), position = position_jitter()) +
  facet_wrap(~ clarity)
```



Problem 4: Life Expectancy vs Health Care Spending and Other Factors

The assignment is to use data afrom the Gapminder website and create 2-5 plots examining 3 or more variables.

Let's say that we are Healthcare policy analysts and we are asked to look at world data statistics and determine if there are some countries that would be especially interesting to examine further to glean possible best practices. To explore this, let's first chart life expectancy versus health care spending per person (HSPP). Since overall GDP might also play a role, we'll also want to classify the data in GDP per person (GDPP) buckets and use that to color the data.

First, let's read in the data. The spreadsheets were downloaded from the Gapminder website and then saved in CSV format before being read in here:

```
life <- read.csv("indicator life_expectancy_at_birth.csv", header = TRUE)
hspp <- read.csv("indicator health spending per person (US $).csv", header =
TRUE)
gdpp <- read.csv("indicator gapminder gdp_per_capita_ppp.csv", header = TRUE)
```

Next we do some data wrangling. We will rename column 1 in each dataset to the more helpful name "Country", select the most recent year of data (which happens to be 2010 in this case), remove all "NAs", and then rename the column of data from the default "X2010" to something more descriptive:

```
colnames(life)[1] = "Country"
colnames(hspp)[1] = "Country"
colnames(gdpp)[1] = "Country"
life2010 <- life %>% select(Country, X2010)
hspp2010 <- hspp %>% select(Country, X2010)
gdpp2010 <- gdpp %>% select(Country, X2010)
life2010 <- life2010 %>% filter(!is.na(X2010))
hspp2010 <- hspp2010 %>% filter(!is.na(X2010))
gdpp2010 <- gdpp2010 %>% filter(!is.na(X2010))
colnames(life2010)[2] = "Life"
colnames(hspp2010)[2] = "HSPP"
colnames(gdpp2010)[2] = "GDPP"
```

Next we join the data, using "Country" as the common key:

```
all2010 <- inner_join(life2010, hspp2010)

## Joining by: "Country"

## Warning in inner_join_impl(x, y, by$x, by$y): joining factors with
## different levels, coercing to character vector

all2010 <- inner_join(all2010, gdpp2010)

## Joining by: "Country"
## Warning in inner_join_impl(x, y, by$x, by$y): joining character vector and
## factor, coercing into character vector
```

The next step is to generate the GDPP buckets. First "summary()" is done to determine the boundary points of the 4 quartiles. These values ($3,337, $16,770, $21,710) are used by the "cut" commmand along with max/min values ($0, $128,000) to define gdpp.bucket:

```
summary(all2010$GDPP)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     632    3337   10410   16770   21710  128000

all2010$gdpp.bucket <- cut(all2010$GDPP,
                           breaks=c(0, 3337, 16770, 21710, 128000),
                           dig.lab = 10)
```

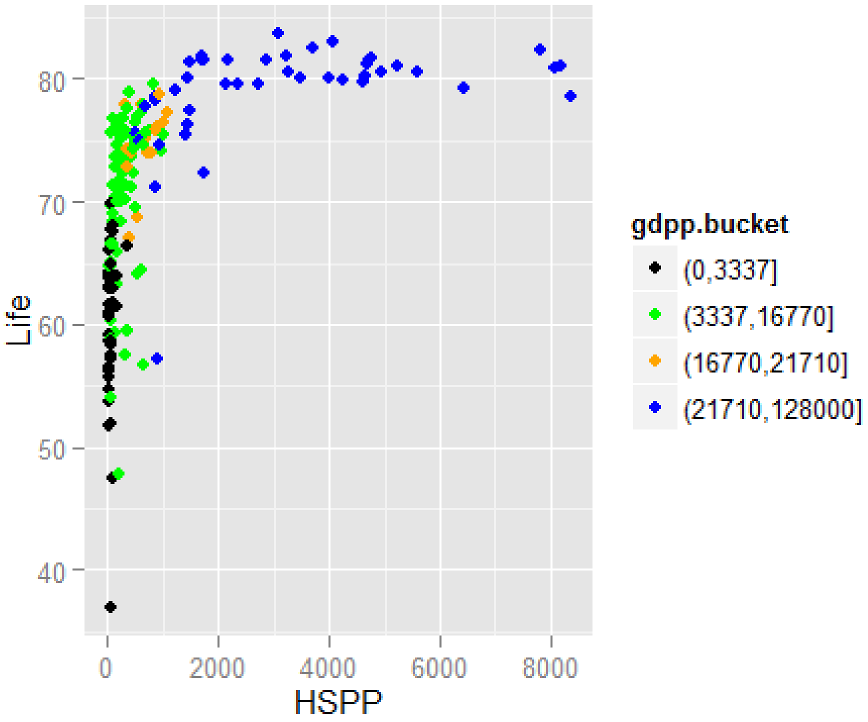
The "dig.lab = 10" option is used later to prevent scientific notation from being used in the legend for the GDPP buckets.

We also want to redefine the colors used because the default colors used are hard to distinguish at times:

```
cvalues <- c('Black', 'Green', 'Orange', 'Blue')
```

Now we can perform the plot. We are plotting Life Expectancy vs Health Spending Per Person, coloring the dots according to which quartile of GDP per Person the country falls into:

```
ggplot(aes(x = HSPP, y = Life), data = all2010) +
  geom_point(aes(color = gdpp.bucket)) +
  scale_colour_manual(values = cvalues)
```



One can see a rough correlation between HSPP and Life Expectancy at the outset, which seems to top out at 80 years / $1000. After that there is a slight rise among the richer countries to 85 years / $3000, with values more or less leveling off after that.

A few questions arise from this data. First, what is the country with a maximum value of nearly 85 years / $3000 per person? Also, even when HSPP is as low as $1000 per person, there are some upper-tier GDPP countries that have managed to keep their health care costs pretty low, and even some countries only in the second quartile of GDPP that have very high life expectancies. What are those countries?

Let's look first for the country at 85 years / $3000. We can see this by using the "filter()" command in dplyr:

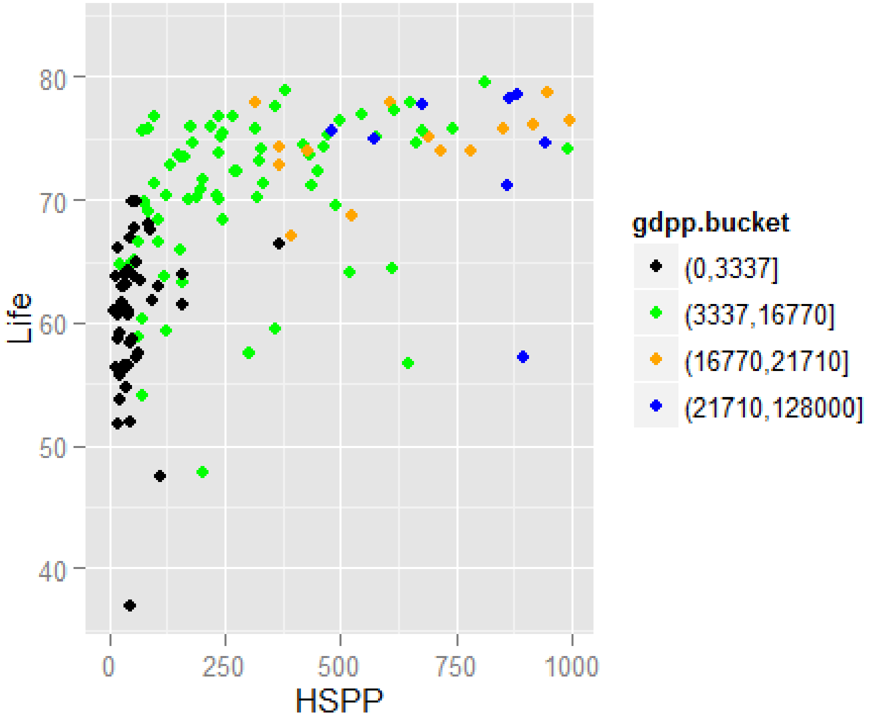
```
all2010 %>% filter(HSPP > 3000, HSPP < 3500, Life > 83)

##   Country Life     HSPP  GDPP    gdpp.bucket
## 1 Andorra 83.7 3099.413 38982 (21710,128000]
```

From this we see the country is Andorra.

Let's look for answers for the next 2 questions. To do this, let's first zoom in on the area where HSPP is between 0 and 1000:

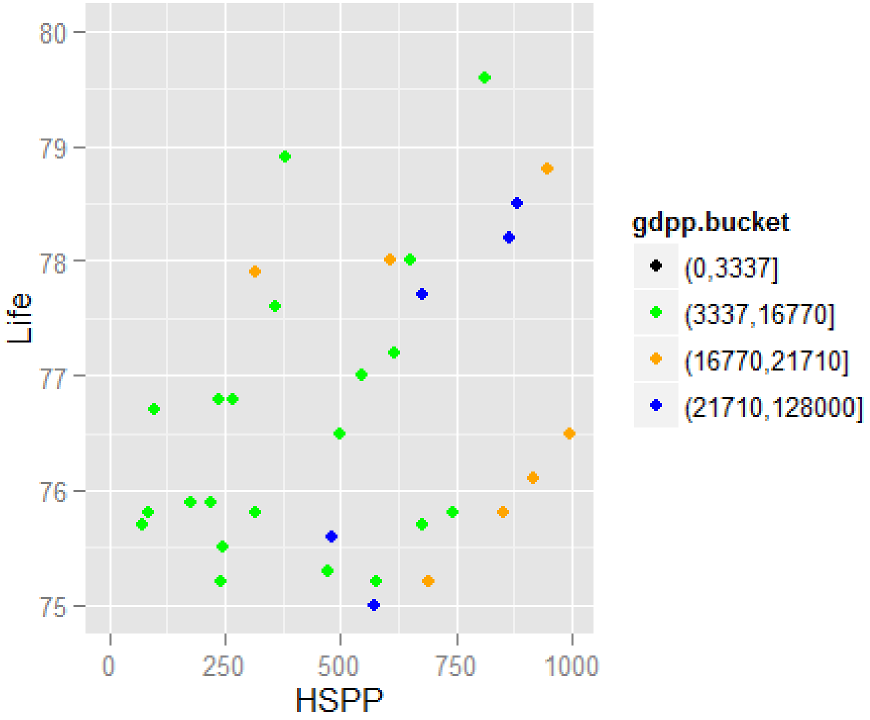
```
ggplot(aes(x = HSPP, y = Life), data = all2010) +
  geom_point(aes(color = gdpp.bucket)) +
  xlim(0, 1000) +
  scale_colour_manual(values = cvalues)
```

```
## Warning: Removed 38 rows containing missing values (geom_point).
```



and let's look even closer at the area where life expectancy is between 75 and 80:

```
ggplot(aes(x = HSPP, y = Life), data = all2010) +
  geom_point(aes(color = gdpp.bucket)) +
  xlim(0, 1000) +
  ylim(75, 80) +
  scale_colour_manual(values = cvalues)

## Warning: Removed 144 rows containing missing values (geom_point).
```



Here we see there are 5 countries in the highest quartile for GDPP that still have relatively high life expectancies with relatively low HSPP. Let's find out what they are:

```
all2010 %>% filter(HSPP < 1000, GDPP > 21710, Life >= 75) %>%
arrange(desc(Life))
```

```
##         Country Life     HSPP  GDPP    gdpp.bucket
## 1        Brunei 78.5 881.5662 70636 (21710,128000]
## 2       Bahrain 78.2 864.1476 40553 (21710,128000]
## 3  Saudi Arabia 77.7 679.6573 45598 (21710,128000]
## 4         Libya 75.6 483.7249 30261 (21710,128000]
## 5          Oman 75.0 574.3068 49188 (21710,128000]
```

We see that they consist of 4 Middle East and/or oil-rich countries (Brunei, Bahrain, Saudi Arabia, Oman) plus Libya. Libya is somewhat a surprise (both that its GDPP is in the highest quartile and that its life expectancy is so high). Since there were major upheavals in that country in 2011, it would be interesting to get the latest statistics. Given Libya's values may be very different and the low health care costs for the other countries might be explained by the presence of oil money, we won't use those countries.

We also see a cluster of 8 countries where health care costs are below $750 per person and life expectancy is 77 or more. Let's find out what those are using "filter()":

```
all2010 %>% filter(HSPP < 750, Life >= 77) %>% arrange(desc(Life))
```

```
##         Country Life     HSPP  GDPP    gdpp.bucket
## 1      Maldives 78.9 382.4750 11674   (3337,16770]
## 2         Cuba 78.0 607.0272 18477  (16770,21710]
## 3      Lebanon 78.0 651.0426 16263   (3337,16770]
## 4         Iran 77.9 316.9254 16980  (16770,21710]
## 5 Saudi Arabia 77.7 679.6573 45598 (21710,128000]
## 6       Jordan 77.6 357.4387 11256   (3337,16770]
## 7       Panama 77.2 616.3865 14921   (3337,16770]
## 8       Serbia 77.0 546.0277 12301   (3337,16770]
```

In addition to the previously seen Saudi Arabia, we see the countries Maldives, Cuba, Lebanon, Iran, Jordan, Panama, and Serbia. These can be added to Andorra for the list of recommended countries to look into.

Finally, we can compare life expectancy to other data values from the Gapminder site. In addition to HCPP and GDPP looked at above, we also look at the Corruption Index (CORR), the Democracy Index (DEMO), the Literacy Rate (LITR), and the Oil Consumption per Person rate (OILC).

The first step is to read in these new datasets:

```
corr <- read.csv("indicator ti cpi 2009.csv", header = TRUE)
demo <- read.csv("indicatorpolityiv.csv", header = TRUE)
litr <- read.csv("indicator SE_ADT_LITR_ZS.csv", header = TRUE)
oilc <- read.csv("Oil Consumption per capita.csv", header = TRUE)
```

Next we do the same data wrangling on these datasets that we did with HCPP and GDPP:

```
colnames(corr)[1] = "Country"
colnames(demo)[1] = "Country"
colnames(litr)[1] = "Country"
colnames(oilc)[1] = "Country"
corr2009 <- corr %>% select(Country, X2009)  # 2009 is most recent
demo2010 <- demo %>% select(Country, X2010)
litr2010 <- litr %>% select(Country, X2010)
oilc2010 <- oilc %>% select(Country, X2010)
corr2009 <- corr2009 %>% filter(!is.na(X2009))
demo2010 <- demo2010 %>% filter(!is.na(X2010))
litr2010 <- litr2010 %>% filter(!is.na(X2010))
oilc2010 <- oilc2010 %>% filter(!is.na(X2010))
colnames(corr2009)[2] = "CORR"
colnames(demo2010)[2] = "DEMO"
colnames(litr2010)[2] = "LITR"
colnames(oilc2010)[2] = "OILC"
```

We join this new data to our "all2010" dataset (note we use 2009 data for the Corruption index instead of 2010 since 2009 is the most recently available data):

```
all2010 <- left_join(all2010, corr2009)  #2009 close enough

## Joining by: "Country"

## Warning in left_join_impl(x, y, by$x, by$y): joining factor and character
## vector, coercing into character vector
```

```
all2010 <- left_join(all2010, demo2010)

## Joining by: "Country"

## Warning in left_join_impl(x, y, by$x, by$y): joining factor and character
## vector, coercing into character vector

all2010 <- left_join(all2010, litr2010)

## Joining by: "Country"

## Warning in left_join_impl(x, y, by$x, by$y): joining factor and character
## vector, coercing into character vector

all2010 <- left_join(all2010, oilc2010)

## Joining by: "Country"

## Warning in left_join_impl(x, y, by$x, by$y): joining factor and character
## vector, coercing into character vector
```

We don't need the GDPP bucket anymore and don't want to graph that, so let's remove it:

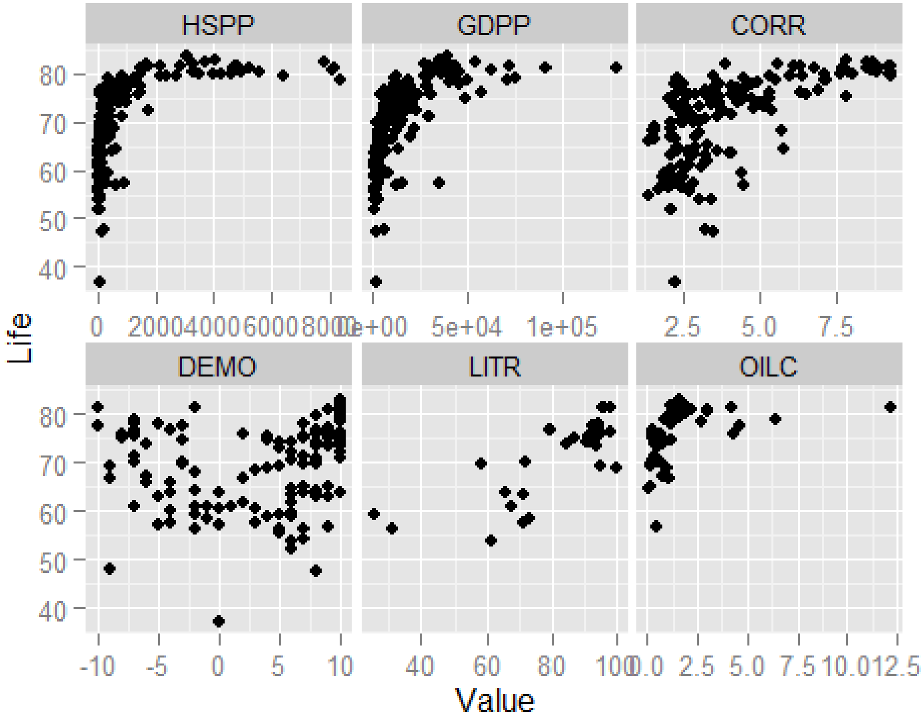
```
all2010 <- all2010[,-5]
```

In order to use facet, the data needs to be in long form. We do this by using "melt()" in reshape2. In the melting, we keep the "Country" and "Life" columns. All other columns we melt into new "Parameter" and "Value" columns:

```
library(reshape2)
all2010_long <- melt(data = all2010,
                     id = 1:2,
                     variable.name = "Parameter",
                     value.name = "Value",
                     na.rm = TRUE)
```

Now we are ready to do a series of scatterplots, faceted by the 6 parameters. The 'scales = "free_x"' is used so that each scatterplot can have its own range of X values:

```
ggplot(aes(x = Value, y = Life), data = all2010_long) +
  geom_point() +
  facet_wrap(~ Parameter, scales = "free_x")
```



HSPP, GDPP, and Oil consumption all seem to have similar curves. There also does seem to be a fairly positive correlation between life expectancy and corruption index (the less corrupt the country, the higher the life expectancy) and literacy rate. These could potentially be explained by the fact that countries with less corruption and higher literacy tend to also have higher GDP. It is difficult to draw conclusions from the democracy index. As the democracy rating increases the life expectancy does seem to tend to get higher, though there are countries with high life expectancies in almost all values of the rating.