Data Exploration: Problem Set 4

Robert Chen

December 14, 2015

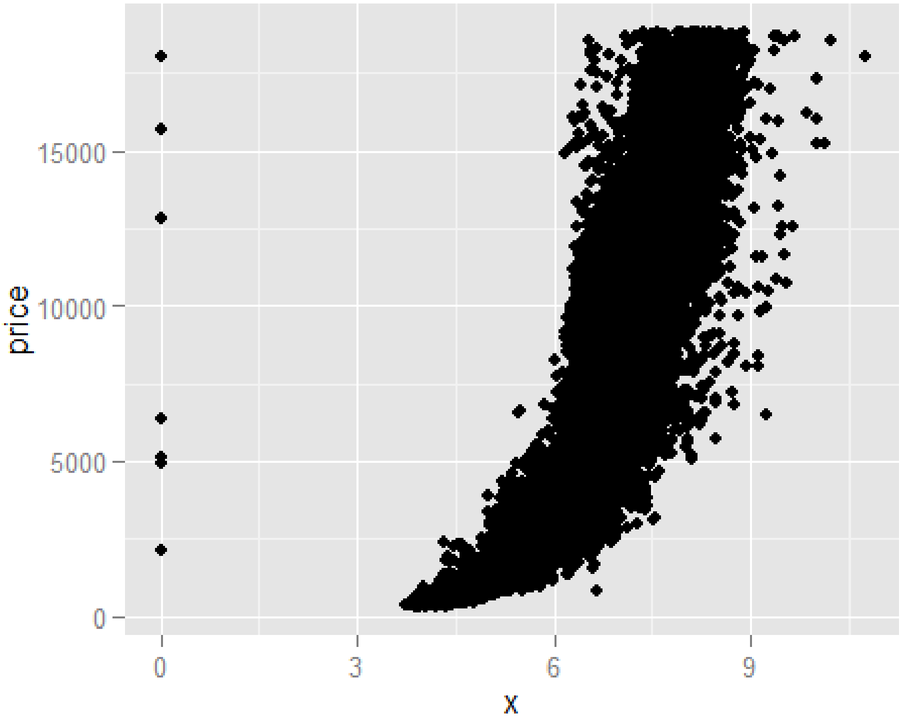
Summary

This is a summary of the code to complete Problem Set 4 of UD561 for the Exploratory Data Analysis Unit (Part 2) of the Sliderule "Fundamentals of Data Science" course.

Problem 1a: Price vs x -- Scatterplot

The assignment is to create a scatterplot of price vs x in the "diamonds" dataset using ggplot. Here is the code to do this:

```
library(ggplot2)
ggplot(aes(x = x, y = price), data = diamonds) + geom_point()
```



One can see from this graph that in general price seems to increase as X increases after the value of x=3. Also, there appear to be some outlier values at x = 0.

Problem 1b: Price vs x -- Correlation

One can get the correlation by using the cor.test command:

```
cor.test(diamonds$x, diamonds$price)
```

```
##
##  Pearson's product-moment correlation
##
## data:  diamonds$x and diamonds$price
## t = 440.16, df = 53938, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8825835 0.8862594
## sample estimates:
##       cor
## 0.8844352
```

```
cor.test(diamonds$y, diamonds$price)
```

```
##
##  Pearson's product-moment correlation
##
## data:  diamonds$y and diamonds$price
## t = 401.14, df = 53938, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8632867 0.8675241
## sample estimates:
##       cor
## 0.8654209
```

```
cor.test(diamonds$z, diamonds$price)
```

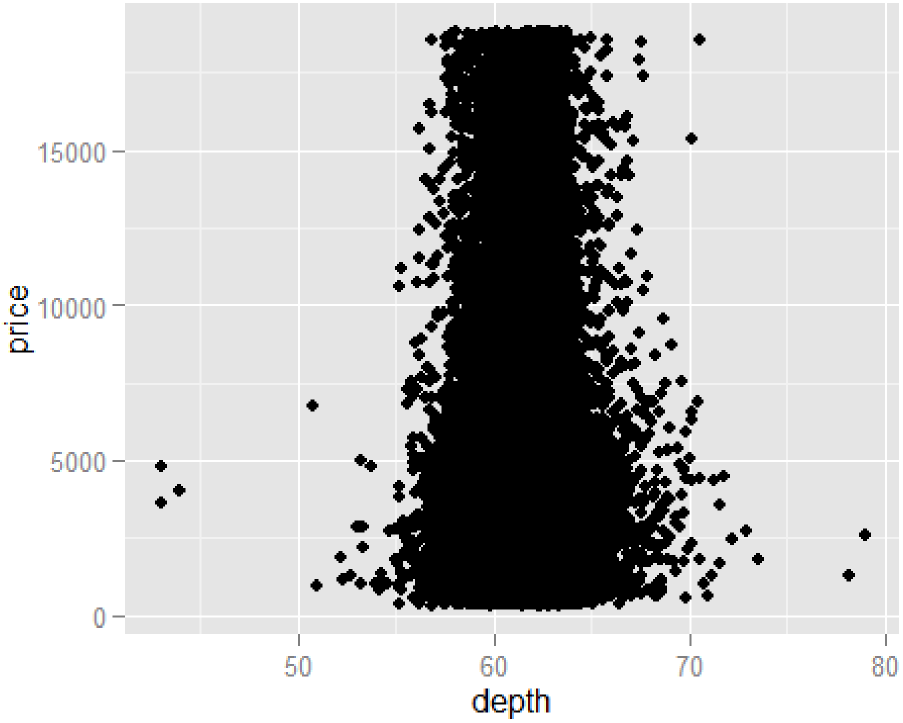
```
##
##  Pearson's product-moment correlation
##
## data:  diamonds$z and diamonds$price
## t = 393.6, df = 53938, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8590541 0.8634131
## sample estimates:
##       cor
## 0.8612494
```

From this one can see the correlation between price and x is 0.88, between price and y 0.87, and between price and z 0.86.

Problem 2a: Price vs Depth -- Scatter Plot

The next assignment is to create a simple scatter plot of price vs depth. This is easily done with the basic ggplot command:

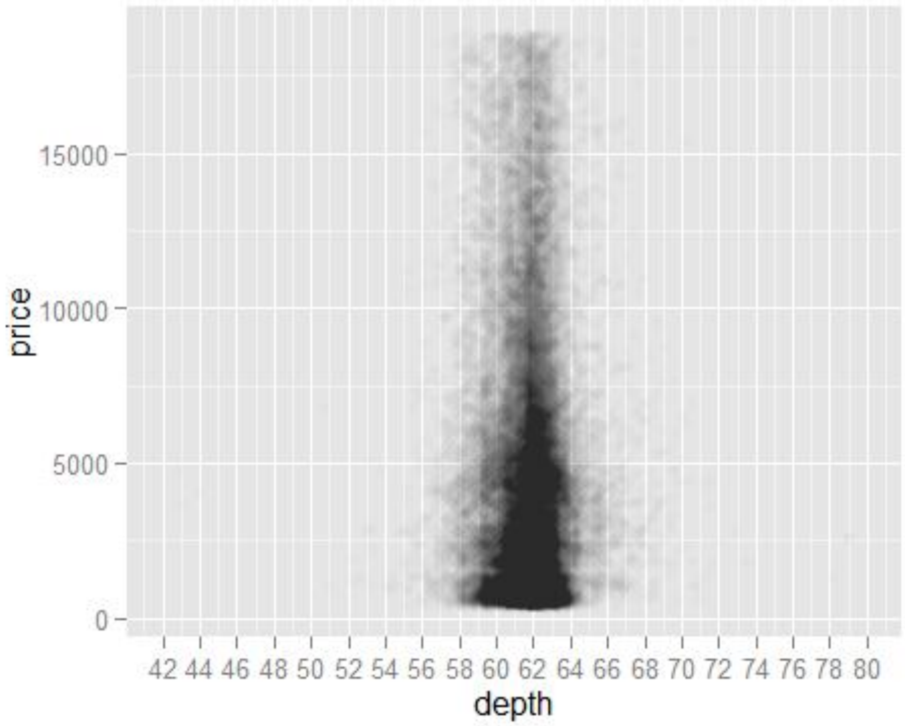
```
ggplot(aes(x = depth, y = price), data = diamonds) + geom_point()
```



Problem 2b: Price vs Depth -- Alpha and x-axis Refinements

The next step is to make the transparency of the points 1/100 and mark the x-axis every 2 units This is done with the "alpha" option in geom_point and by using "scale_x_continuous." "range" is first used to determine the high and low values for the x-axis:

```
range(diamonds$depth)

## [1] 43 79

ggplot(aes(x = depth, y = price), data = diamonds) +
  geom_point(alpha = .01) +
  scale_x_continuous(lim = c(42, 80), breaks = seq(42, 80, 2))
```



Based on these values, most diamonds seem to fall between about 58 and 64.

Problem 2c: Price vs Depth -- Correlation

Using cor.test we find the correlation is -.01:

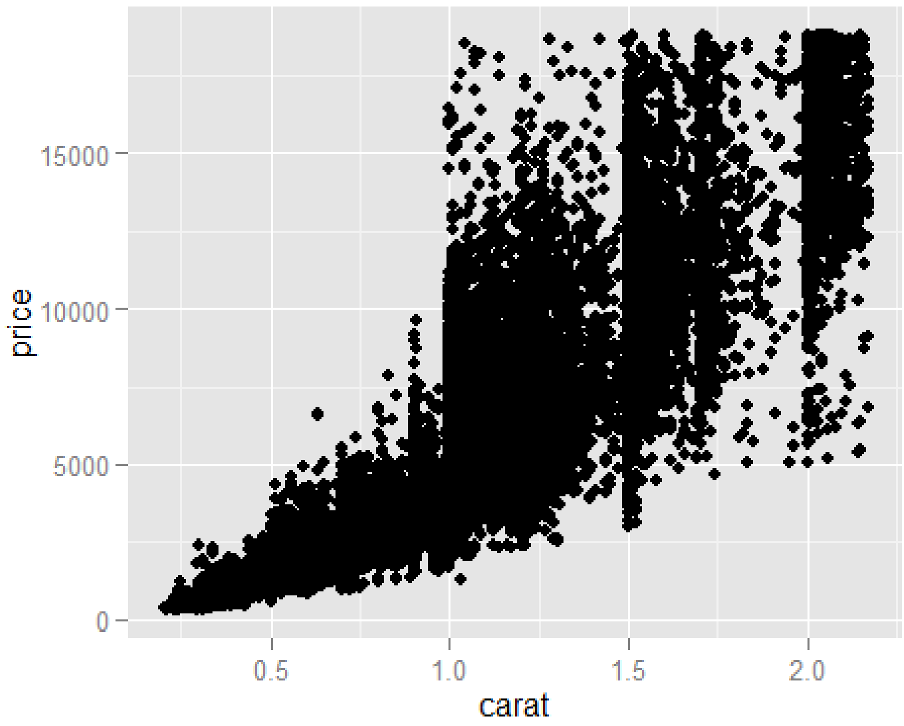
```
cor.test(diamonds$depth, diamonds$price)

##
##  Pearson's product-moment correlation
##
## data:  diamonds$depth and diamonds$price
## t = -2.473, df = 53938, p-value = 0.0134
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.019084756 -0.002208537
## sample estimates:
##        cor
## -0.0106474
```

Based on this value, I would not use depth to predict the price. The very low value of correlation shows there is almost no relationship between the two.

Problem 3: Price vs Carat -- Scatterplot

The next assignment is to create a scatterplot of Price vs Carat, omitting the top 1% of price and carat. The "quantile" command is used to find the value at which the top 1% occurs. Those values are then used in "subset" to create the scatterplot:

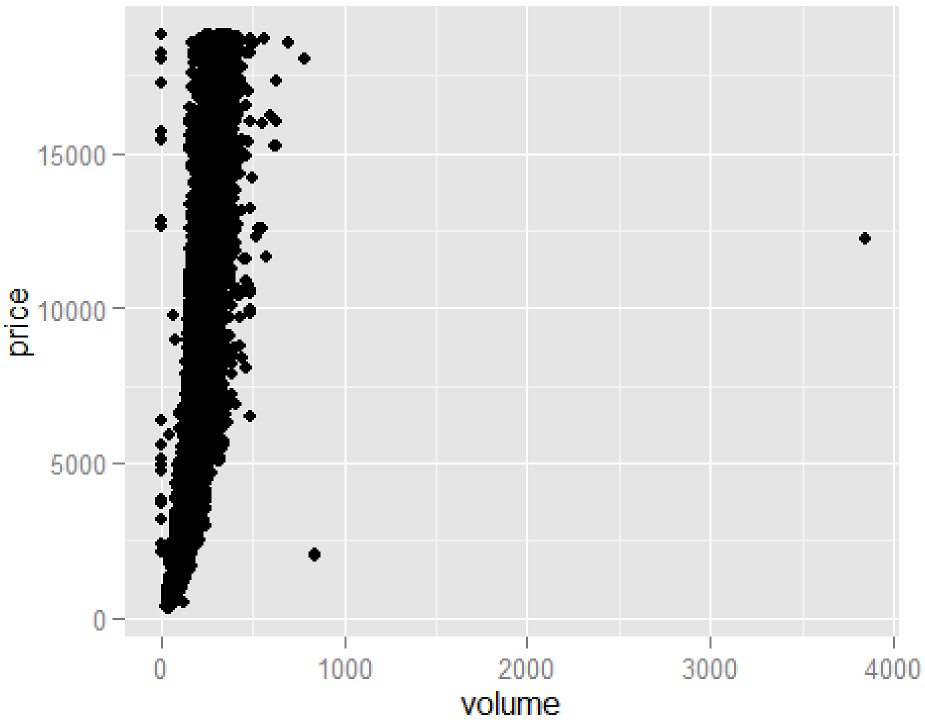
```
quantile(diamonds$price,.99)

##      99%
## 17378.22

quantile(diamonds$carat,.99)

##  99%
## 2.18

ggplot(aes(x = carat, y = price),
       data = subset(diamonds, carat < 2.18, price < 17378.22)) +
  geom_point()
```



Problem 4a: Price vs Volume -- Scatterplot

The next assignment is to create a new value for volume (equal to x * y * z) and then use that for a scatterplot of Price vs Volume:

```
diamonds$volume <- diamonds$x * diamonds$y * diamonds$z
ggplot(aes(x = volume, y = price), data = diamonds) + geom_point()
```



One can see that there seems to be a general positive correlation between volume and price, with two clear outlier values.

Problem 4b: Price vs Volume -- Correlation

The assignment is to find the correlation of Price vs Volume, where 0 < Price < 800. "subset" is used to narrow the values before calculating the correlation:

```
d2 <- subset(diamonds, volume > 0)
d3 <- subset(d2,       volume < 800)
cor.test(d3$price, d3$volume)

##
##  Pearson's product-moment correlation
##
## data:  d3$price and d3$volume
## t = 559.19, df = 53915, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9222944 0.9247772
## sample estimates:
##       cor
## 0.9235455
```

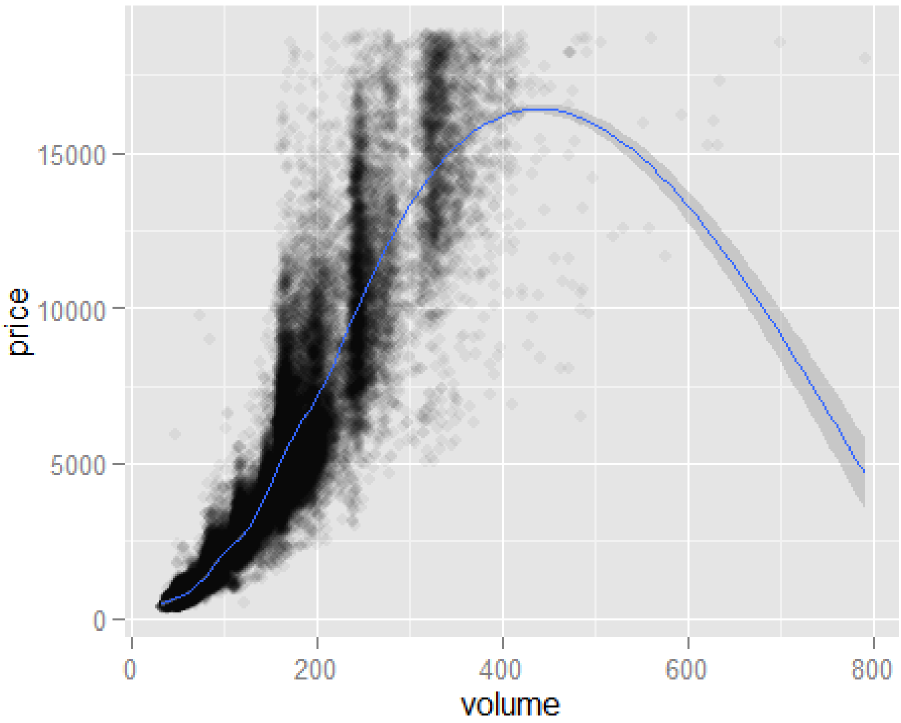
The value is 0.92, which is very high.

Problem 4c: Price vs Volume -- Transparency / Linear Model

The assignment is to adjust the transparency and add a linear model, while retaining the 0 < Price < 800 limit of before. "alpha" is used to make the transparency 1/20, while "geom_smooth()" is used to add the linear model line:

```
ggplot(aes(x = volume, y = price), data = d3) +
  geom_point(alpha = .05) +
  geom_smooth()
```

```
## 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.
```



The model looks like it could be useful up to a value of volume = 400, though after that it's not clear if it remains useful since the point causing the subsequent downturn could potentially be an invalid outlier.

Problem 5: Summarizing Clarity

The assignment is to use dplyr to create a new data frame containing mean_price, median_price, min_price, max_price, and n for each value of clarity. This is done by using "group_by()" 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
```

```
diamondsByClarity <- diamonds %>%
  group_by(clarity) %>%
  summarise(mean_price   = mean(price),
            median_price = median(price),
            min_price    = min(price),
            max_price    = max(price),
            n            = n())
head(diamondsByClarity)
```

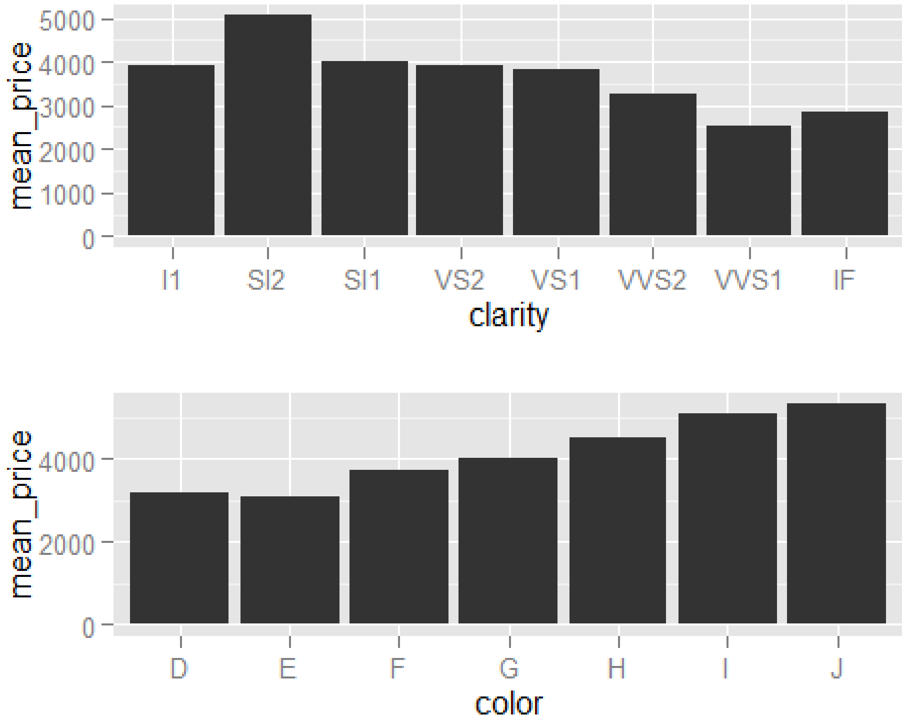
```
## Source: local data frame [6 x 6]
##
##   clarity mean_price median_price min_price max_price     n
##    (fctr)      (dbl)        (dbl)     (int)     (int) (int)
## 1      I1   3924.169         3344       345     18531   741
## 2     SI2   5063.029         4072       326     18804  9194
## 3     SI1   3996.001         2822       326     18818 13065
## 4     VS2   3924.989         2054       334     18823 12258
## 5     VS1   3839.455         2005       327     18795  8171
## 6    VVS2   3283.737         1311       336     18768  5066
```

Problem 6: Clarity and Color Bar Charts

The assignment is to summarize mean_price by clarity and color and produce bar charts for both. This is done with the following code:

```
diamonds_by_clarity <- group_by(diamonds, clarity)
diamonds_mp_by_clarity <- summarise(diamonds_by_clarity, mean_price =
mean(price))
diamonds_by_color <- group_by(diamonds, color)
diamonds_mp_by_color <- summarise(diamonds_by_color, mean_price =
mean(price))
d1 <- ggplot(aes(x = clarity, y = mean_price), data = diamonds_mp_by_clarity)
+
  geom_bar(stat="identity")
d2 <- ggplot(aes(x = color,   y = mean_price), data = diamonds_mp_by_color)
+
```

```r
  geom_bar(stat="identity")
library(gridExtra)
grid.arrange(d1, d2, ncol = 1)
```



The 'stat="identity"' is used to prevent ggplot from making a default histogram.

One gets the counter-intuitive result that the worst clarity ratings (I1, SI1, SI2 ...) have the highest mean price, and the worst colors (J, I ..) also have the highest mean price.

Problem 7: Gapminder data -- Income vs Internet Use

The final problem is to take data from the Gapminder site and create 2-5 plots using the scatter plot / correlation techniques in this unit.

The Income Per Person and Internet Users (per 100 people) datasets were used.

Let's say that you're a software business and you want to look into potential new countries to start a design center. Assuming countries with relatively high Internet usage would tend to have more programmers, you are looking for countries that have a relatively high Internet use with a corresponding lower-than-expected average income.

The first step is to read in the data files:

```r
internet <- read.csv("Internet user per 100.csv", header = TRUE)
income   <- read.csv("indicator gapminder gdp_per_capita_ppp.csv", header = 
TRUE)
```

Next we will do some data wrangling. The first thing we do is change the name of column 1 to the more useful "Country". Then we select just the year 2011 (2011 is the most recent year for the Internet usage dataset):

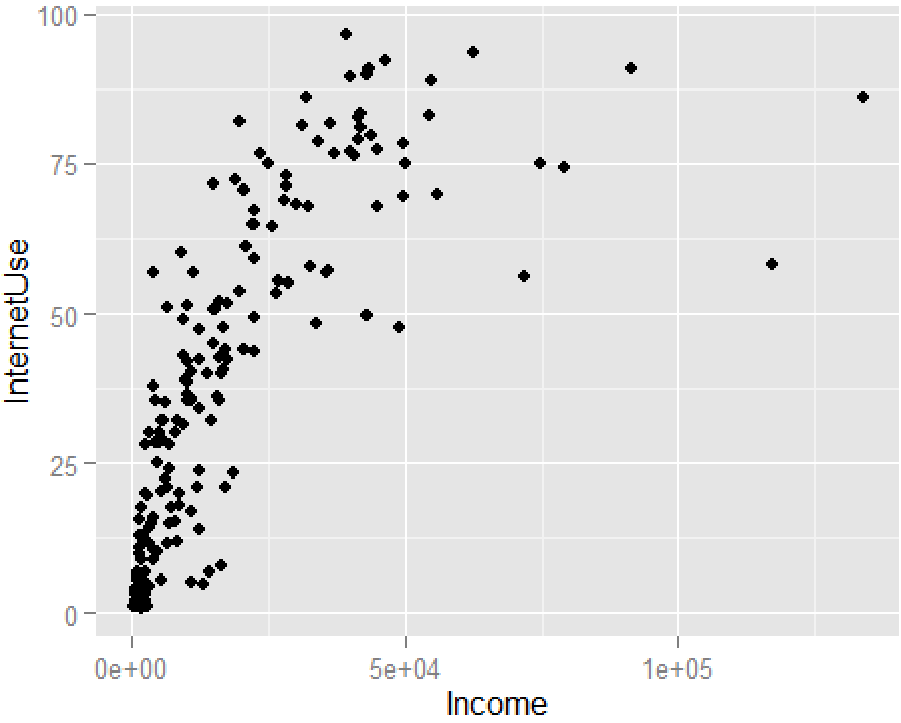
```
colnames(internet)[1] = "Country"
colnames(income)[1]   = "Country"
internet2011 <- internet %>% select(Country, X2011)
income2011   <- income   %>% select(Country, X2011)
```

Next we remove any countries with NA values. Also, since both datasets have columns named "X2011", we rename those columns to prevent a conflict later:

```
internet2011 <- internet2011 %>% filter(!is.na(X2011))
income2011   <- income2011   %>% filter(!is.na(X2011))
colnames(internet2011)[2] = "InternetUse"
colnames(income2011)[2]   = "Income"
```

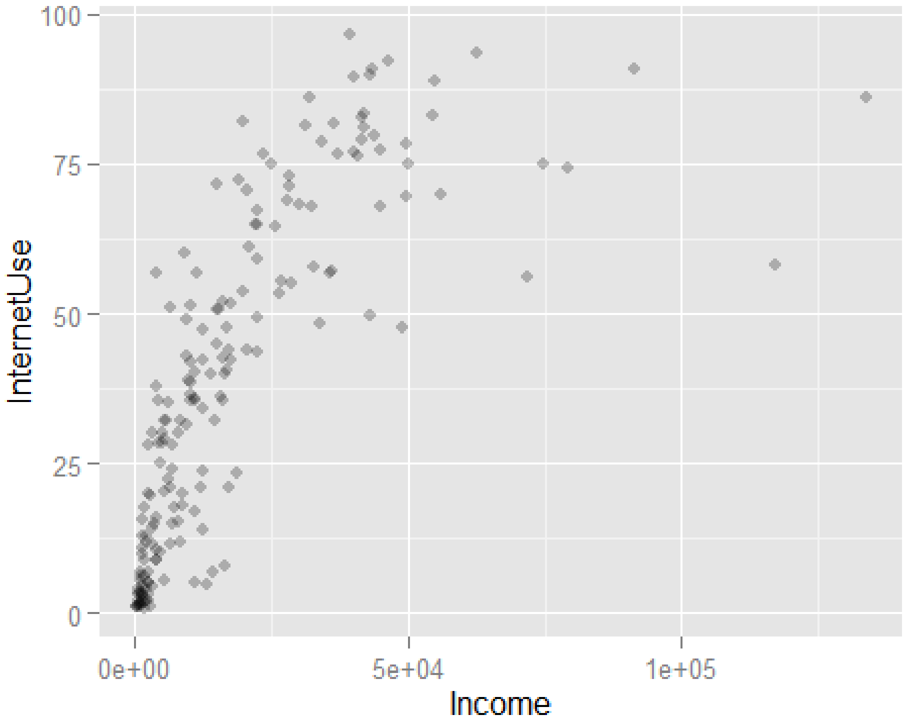
Now we are ready to do some analysis. We join the 2 datasets (using "Country" as the common field) and then do a simple scatterplot. In the join, we do an inner join to just use countries where both Internet and Income data are available. When we do this we get:

```
all2011 <- inner_join(internet2011, income2011)

## Joining by: "Country"

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

ggplot(aes(x = Income, y = InternetUse), data = all2011) + geom_point()
```



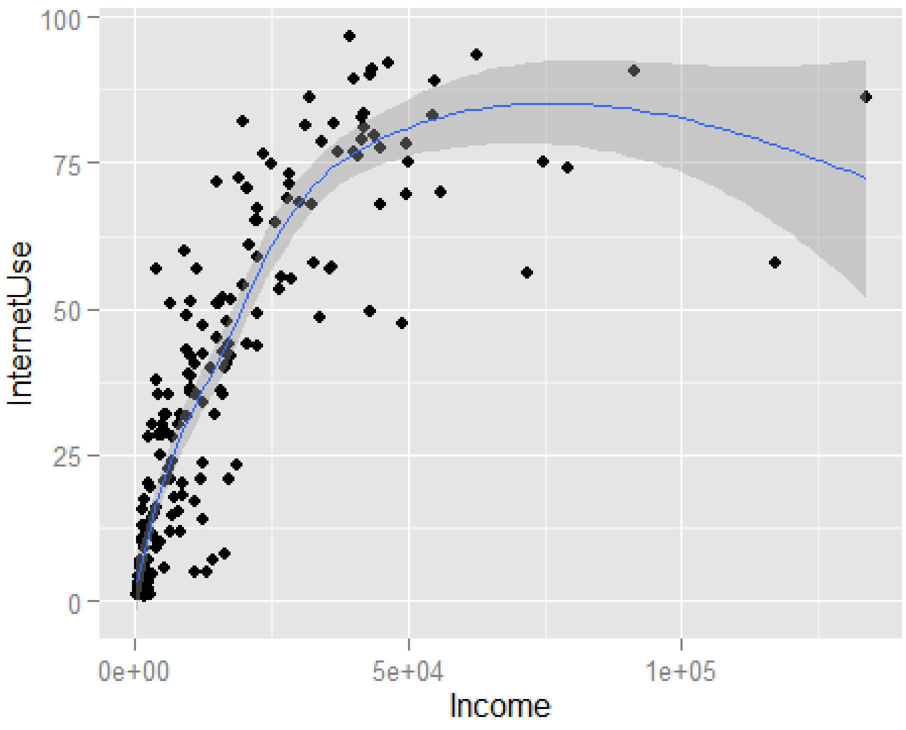
We see a cluster of points near the bottom left. To see if overplotting is involved, let's add an alpha value of 0.25:

```
ggplot(aes(x = Income, y = InternetUse), data = all2011) + geom_point(alpha =
0.25)
```



We see that this looks about the same -- so there isn't a lot of overplotting occurring. Let's go back to alpha = 1 so the dots are darker. Now let's add a correlation line using "geom_smooth()" and find the correlation using "cor.test":

```
ggplot(aes(x = Income, y = InternetUse), data = all2011) +
  geom_point() +
  geom_smooth()

## geom_smooth: method="auto" and size of largest group is <1000, so using
loess. Use 'method = x' to change the smoothing method.
```



```
cor.test(all2011$Income, all2011$InternetUse)
```

```
## 
##  Pearson's product-moment correlation
## 
## data:  all2011$Income and all2011$InternetUse
## t = 15.326, df = 181, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6806562 0.8084402
## sample estimates:
##       cor 
## 0.7515132
```

The correlation value of .75 points to a fairly strong relationship between the 2.

Now we are ready to do some analysis. One thing we see is a cluster of points near 100% usage and just below $50,000 per year. We can find those by listing all countries above 85% usage and then looking at the income:

```
all2011 %>% filter(InternetUse > 85) %>% arrange(desc(InternetUse))
```

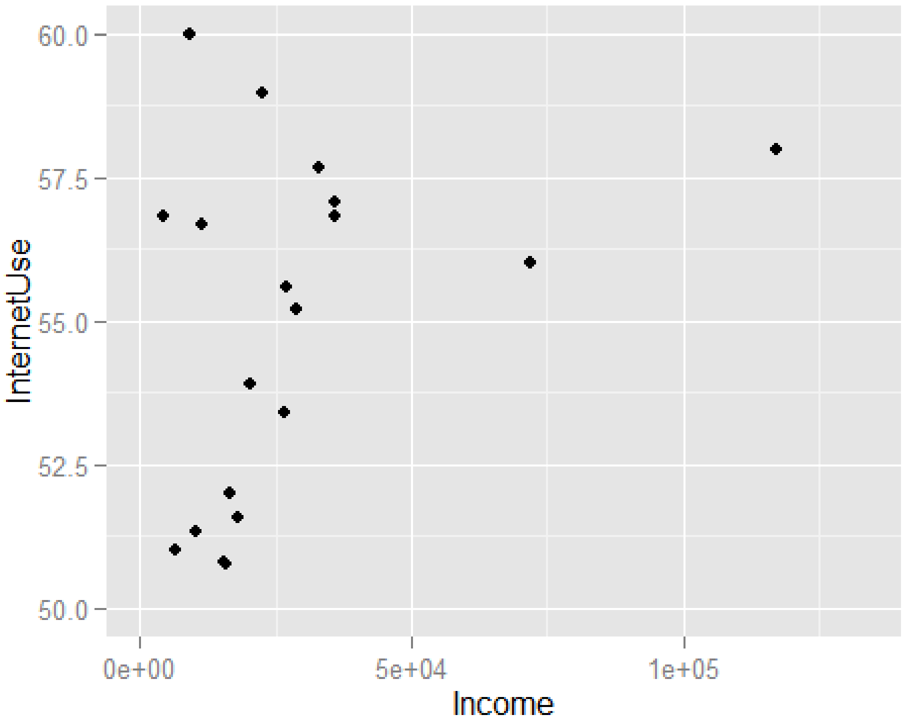
```
##        Country InternetUse Income
## 1      Iceland    96.61836  39619
## 2       Norway    93.45476  62737
## 3  Netherlands    92.12722  46388
## 4       Sweden    90.88205  43709
## 5   Luxembourg    90.70382  91469
## 6      Denmark    89.97730  43314
## 7      Finland    89.33300  40251
## 8      Bermuda    88.85072  54985
## 9        Qatar    86.20000 133734
## 10 New Zealand    86.18173  32283
```

From this we see that several countries in the Scandinavian area (Iceland, Sweden, Denmark, Finland) and New Zealand could be worth looking into. The high Internet usage rate for Finland and Sweden could also be explained by the prevalence of mobile phones in the area. They probably already have quite a few software design centers, so maybe Denmark, Iceland and New Zealand would be better to look into first.

We also see from the data that Qatar is the outlier country with an income greater than $100,000 but Internet Usage at "only" 86%.

Another area of potential interest is a cluster of countries with just above 50% Internet usage that are fairly far from the curve. Let's zoom in on that area using ylim:

```
  ggplot(aes(x = Income, y = InternetUse), data = all2011) +
  geom_point() +
  ylim(50, 60)

## Warning: Removed 164 rows containing missing values (geom_point).
```



We see that the countries of interest are roughly at 56 to 60% Internet usage. Let's list the countries that fit that:

```
all2011 %>% filter(InternetUse > 56, InternetUse < 60 ) %>%
arrange(desc(InternetUse))
```

```
##               Country InternetUse Income
## 1             Hungary    58.97110  22524
## 2        Macao, China    58.00000 117188
## 3              Cyprus    57.68000  32983
## 4               Aruba    57.06836  36016
## 5 West Bank and Gaza    56.81903   4359
## 6               Italy    56.81747  35901
## 7     Macedonia, FYR    56.70000  11431
```

7 countries are listed. We see that Hungary, Cyprus, Aruba, the West Bank, Italy, and Macedonia might be interesting areas to check out. We also see that Macao is the outlier country with > $100,000 income but only 58% usage.

From this and the 3 countries listed before we have 10 countries we could look at as potential candidates for the design center.

Finally, from the graph, we see that there are 4 countries that have relatively high incomes but lower Internet usage numbers than you would expect. As a sanity check of the data, let's look at those countries:

```
all2011 %>% filter(Income > 50000, InternetUse < 75 ) %>%
arrange(desc(InternetUse))
```

```
##                 Country InternetUse Income
## 1                Kuwait        74.2  79102
## 2 United Arab Emirates        70.0  56192
## 3          Macao, China        58.0 117188
## 4                Brunei        56.0  71991
```

We see that they consist of 3 countries that have relatively large incomes from oil, plus a country which might have a relatively large income due to gambling income (Macao).