R Notebook Prob/Stats - Bryan Honeck 02.08.2021

Code

This data was taken from a fictional problem located on the Kaggle website-an online data science community. The problem involves a businessman who is attempting to start up his own phone company. He has been gathering data on thousands of different types of phones over the years. His goal is to be able to predict what price range a phone is in given the remaining 20 attributes listed in the structure of this data frame (i.e. battery capacity, phone weight, RAM, etc.).

Hide

```
#phone.data <- train

head(phone.data)
```

	battery_power <int>	bl… <fctr>	clock_speed <dbl>	dual_sim <fctr>	fc <int>	four_g <fctr>	int_memory <int>	m_… <dbl>	mobile_wt <int>
1	842	No	2.2	No	1	No	7	0.6	188
2	1021	Yes	0.5	Yes	0	Yes	53	0.7	136
3	563	Yes	0.5	Yes	2	Yes	41	0.9	145
4	615	Yes	2.5	No	0	No	10	0.8	131
5	1821	Yes	1.2	No	13	Yes	44	0.6	141
6	1859	No	0.5	Yes	3	No	22	0.7	164

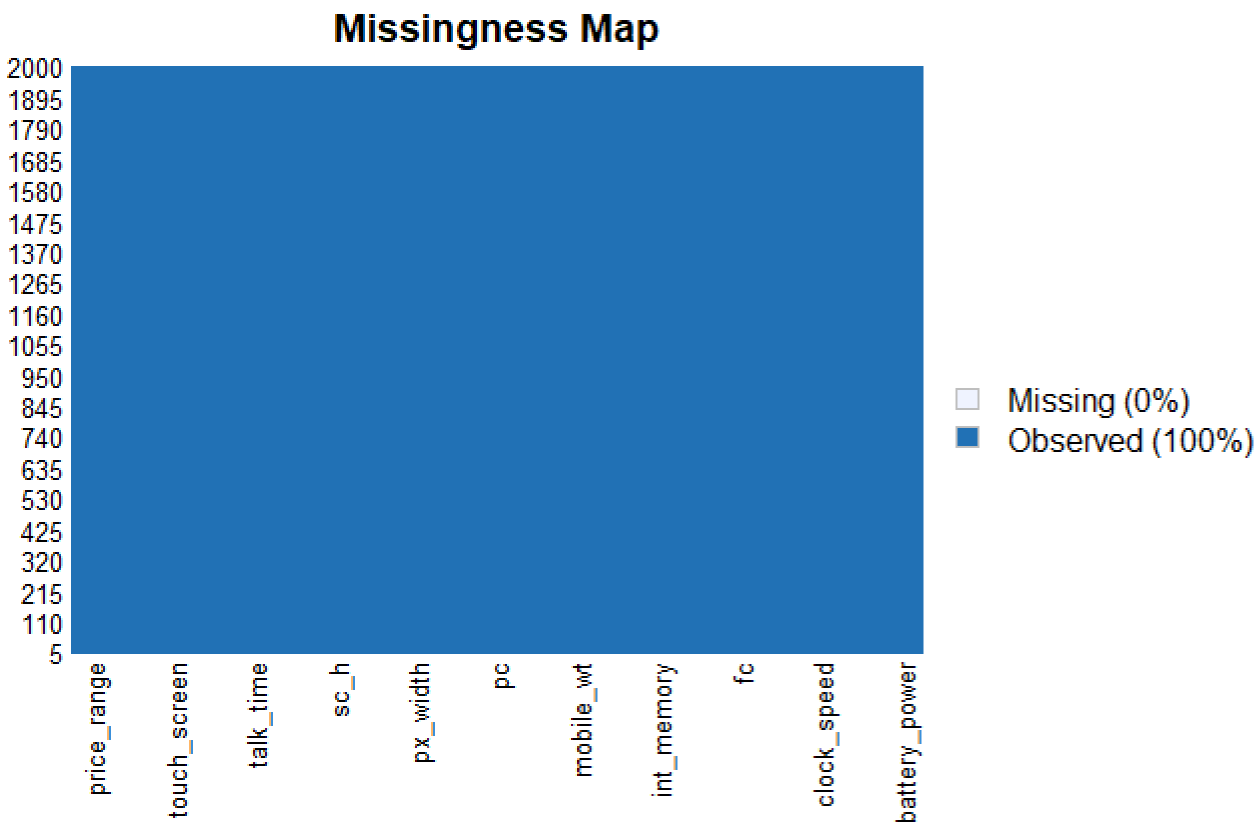
6 rows | 1-10 of 21 columns

It would be good to check and see if there is any missing data, just in case.

Hide

```
library(Amelia)

missmap(phone.data)
```



Hide

```
str(phone.data)
```

```
'data.frame':	2000 obs. of  21 variables:
 $ battery_power: int  842 1021 563 615 1821 1859 1821 1954 1445 509 ...
 $ blue         : Factor w/ 2 levels "No","Yes": 1 2 2 2 2 1 1 1 2 2 ...
 $ clock_speed  : num  2.2 0.5 0.5 2.5 1.2 0.5 1.7 0.5 0.5 0.6 ...
 $ dual_sim     : Factor w/ 2 levels "No","Yes": 1 2 2 1 1 2 1 2 1 2 ...
 $ fc           : int  1 0 2 0 13 3 4 0 0 2 ...
 $ four_g       : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 1 2 1 1 2 ...
 $ int_memory   : int  7 53 41 10 44 22 10 24 53 9 ...
 $ m_dep        : num  0.6 0.7 0.9 0.8 0.6 0.7 0.8 0.8 0.7 0.1 ...
 $ mobile_wt    : int  188 136 145 131 141 164 139 187 174 93 ...
 $ n_cores      : int  2 3 5 6 2 1 8 4 7 5 ...
 $ pc           : int  2 6 6 9 14 7 10 0 14 15 ...
 $ px_height    : int  20 905 1263 1216 1208 1004 381 512 386 1137 ...
 $ px_width     : int  756 1988 1716 1786 1212 1654 1018 1149 836 1224 ...
 $ ram          : int  2549 2631 2603 2769 1411 1067 3220 700 1099 513 ...
 $ sc_h         : int  9 17 11 16 8 17 13 16 17 19 ...
 $ sc_w         : int  7 3 2 8 2 1 8 3 1 10 ...
 $ talk_time    : int  19 7 9 11 15 10 18 5 20 12 ...
 $ three_g      : Factor w/ 2 levels "No","Yes": 1 2 2 2 2 2 2 2 2 2 ...
 $ touch_screen : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 1 1 2 1 1 ...
 $ wifi         : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 2 2 1 1 ...
 $ price_range  : Factor w/ 4 levels "Very Cheap","Cheap",..: 2 3 3 3 2 2 4 1 1 1 ...
```

Change 0s and 1s to Yes and No; also update price_range since it is naturally categorical as well.

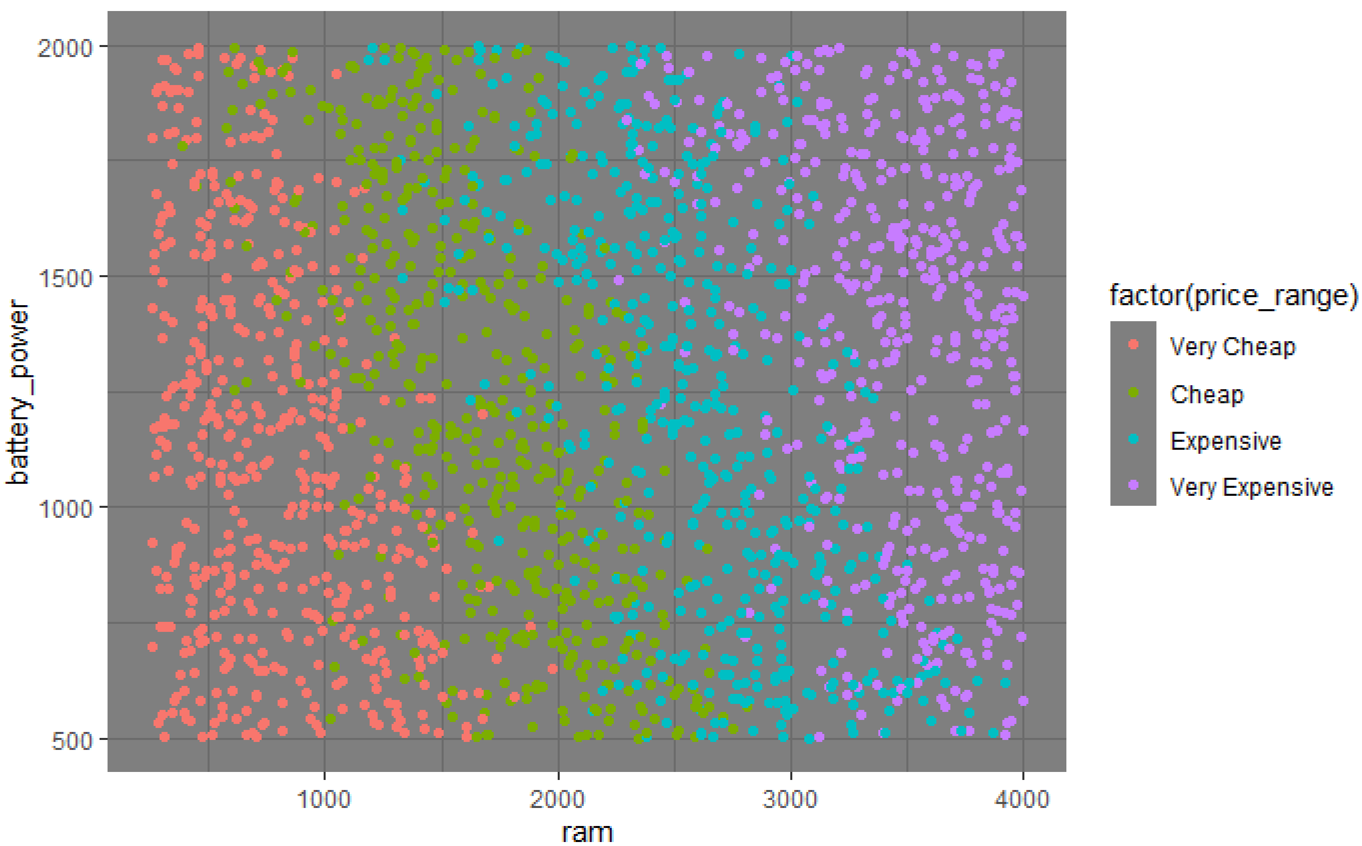
Hide

```
#phone.data$blue <- factor(phone.data$blue, levels=c(0,1), labels=c("No", "Yes"))
#phone.data$dual_sim <- factor(phone.data$dual_sim, levels=c(0,1), labels=c("No", "Yes"))
#phone.data$four_g <- factor(phone.data$four_g, levels=c(0,1), labels=c("No", "Yes"))
#phone.data$three_g <- factor(phone.data$three_g, levels=c(0,1), labels=c("No", "Yes"))
#phone.data$touch_screen <- factor(phone.data$touch_screen, levels=c(0,1), labels=c("No", "Ye
s"))
#phone.data$wifi <- factor(phone.data$wifi, levels=c(0,1), labels=c("No", "Yes"))
#phone.data$price_range <- factor(phone.data$price_range, levels=c(0,1, 2, 3), labels=c("Very Ch
eap", "Cheap", "Expensive", "Very Expensive"))
```

Hide

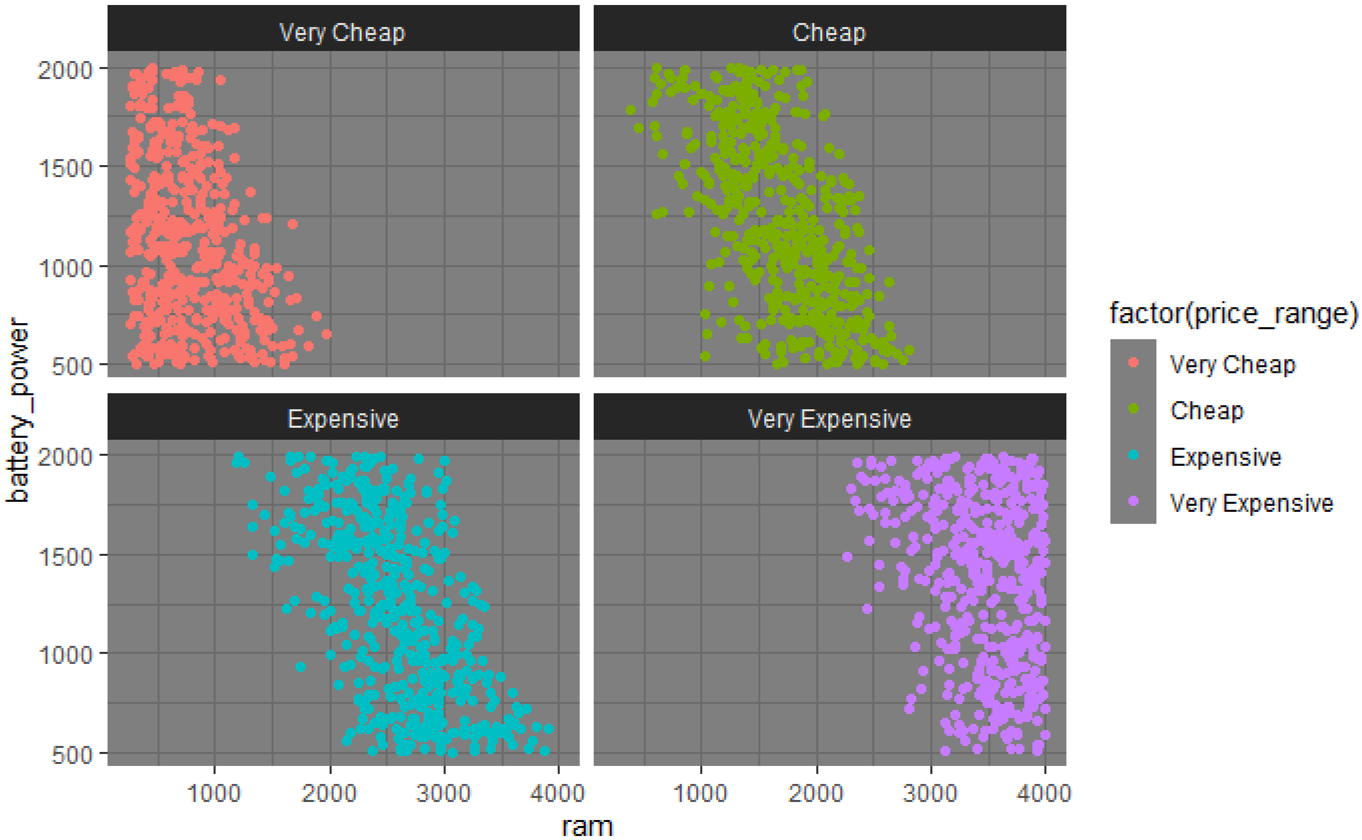
```
library(ggplot2)
library(ggthemes)
library(plotly)

pl <- ggplot(phone.data, aes(x=ram, y=battery_power)) + geom_point(aes(color = factor(price_rang
e))) + theme_dark()
pl
```



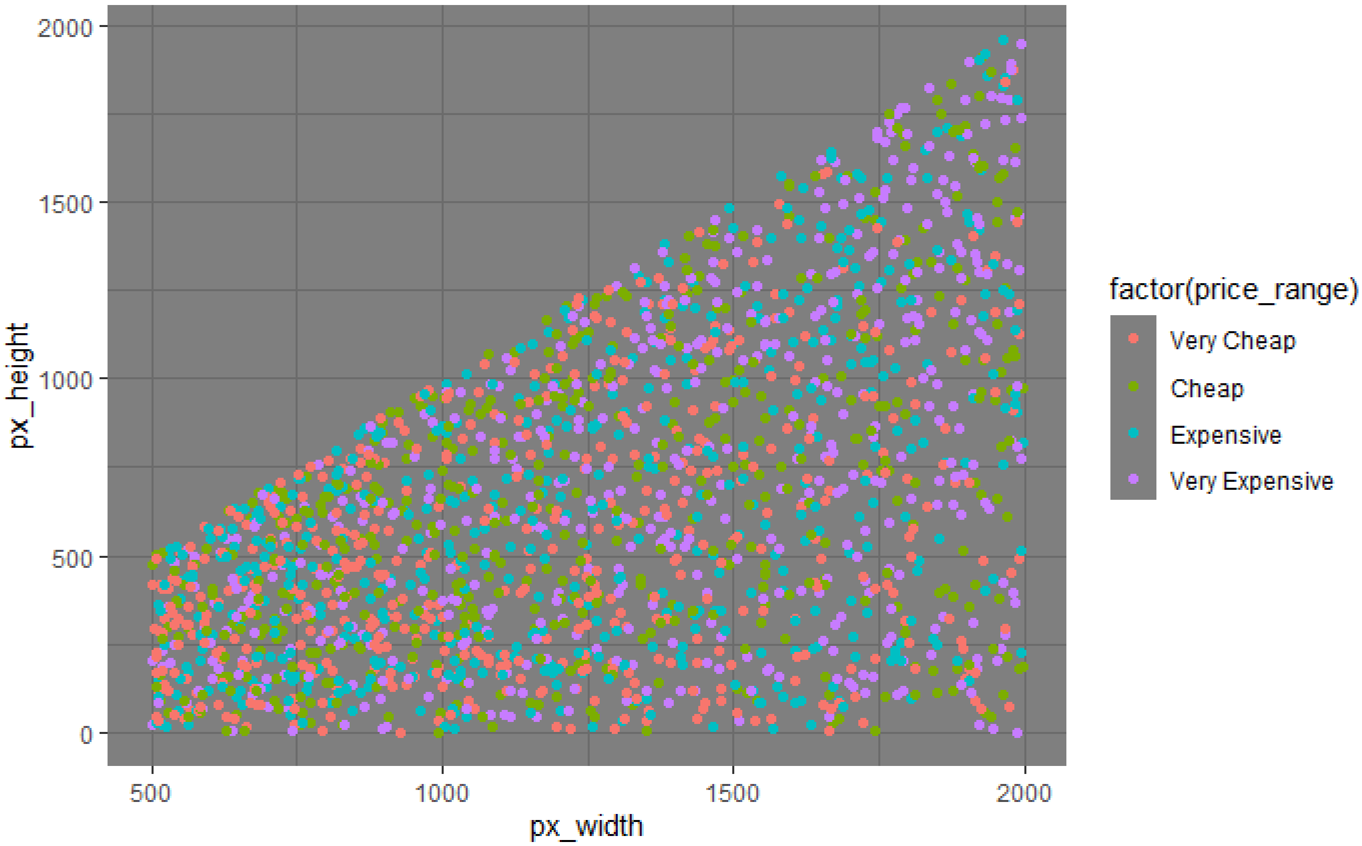
Hide

```
pl.quad <- ggplot(phone.data, aes(x=ram, y=battery_power)) + geom_point(aes(color = factor(price
_range))) + theme_dark() + facet_wrap(~price_range)
pl.quad
```



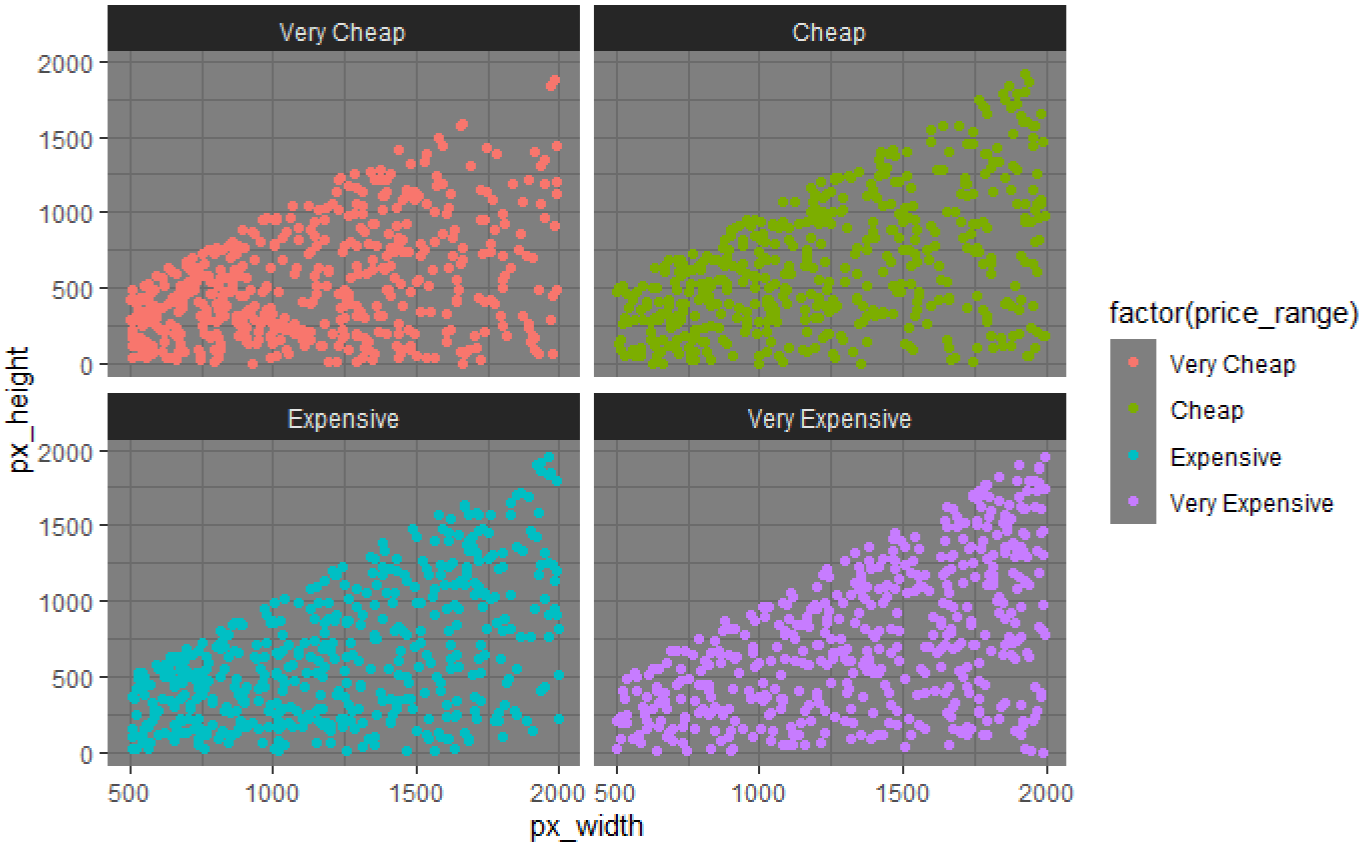
Hide

```
pl2 <- ggplot(phone.data, aes(x=px_width, y=px_height)) + geom_point(aes(color = factor(price_ra
nge))) + theme_dark()
pl2
```



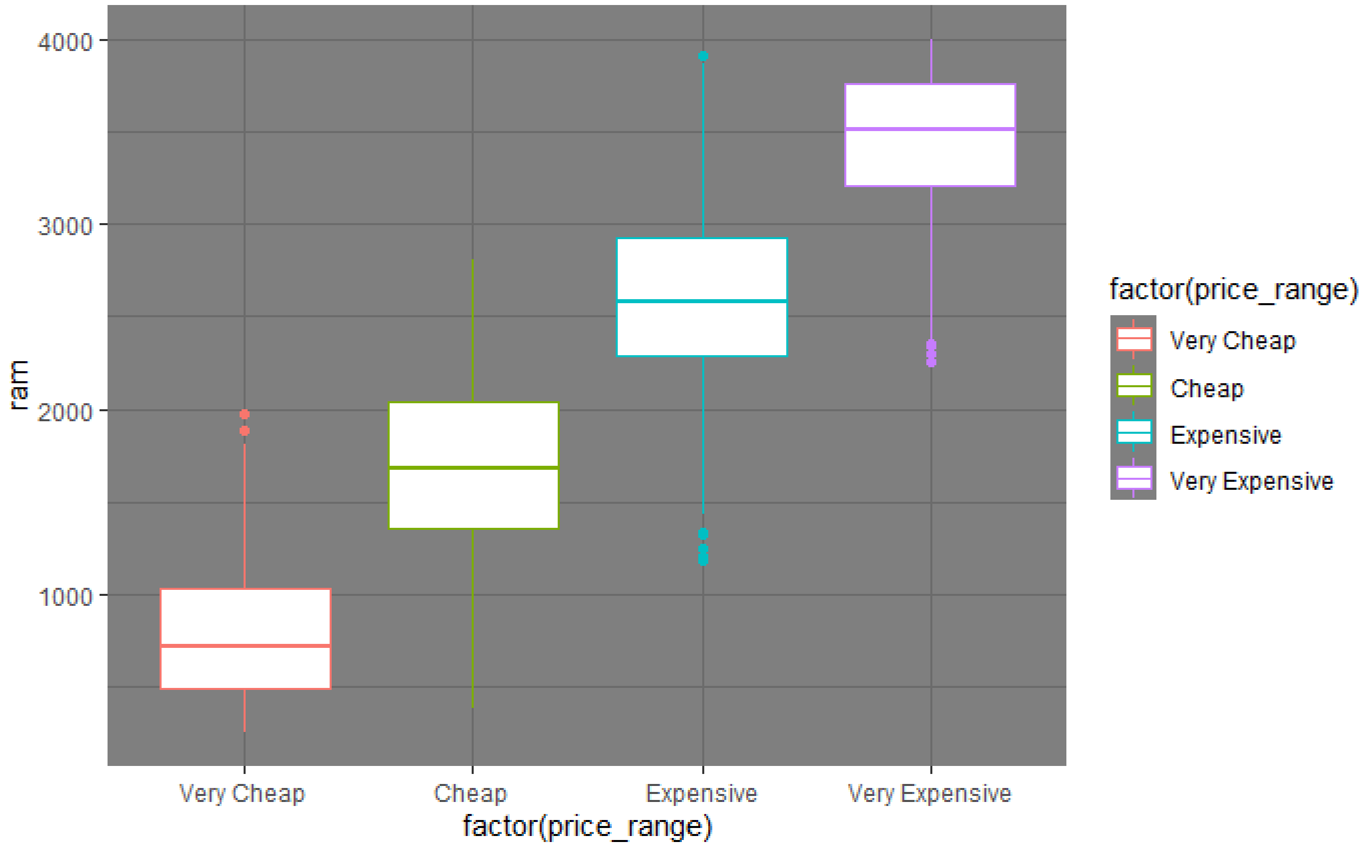
Hide

```
pl2.quad <- ggplot(phone.data, aes(x=px_width, y=px_height)) + geom_point(aes(color = factor(pri
ce_range))) + theme_dark() + facet_wrap(~price_range)
pl2.quad
```



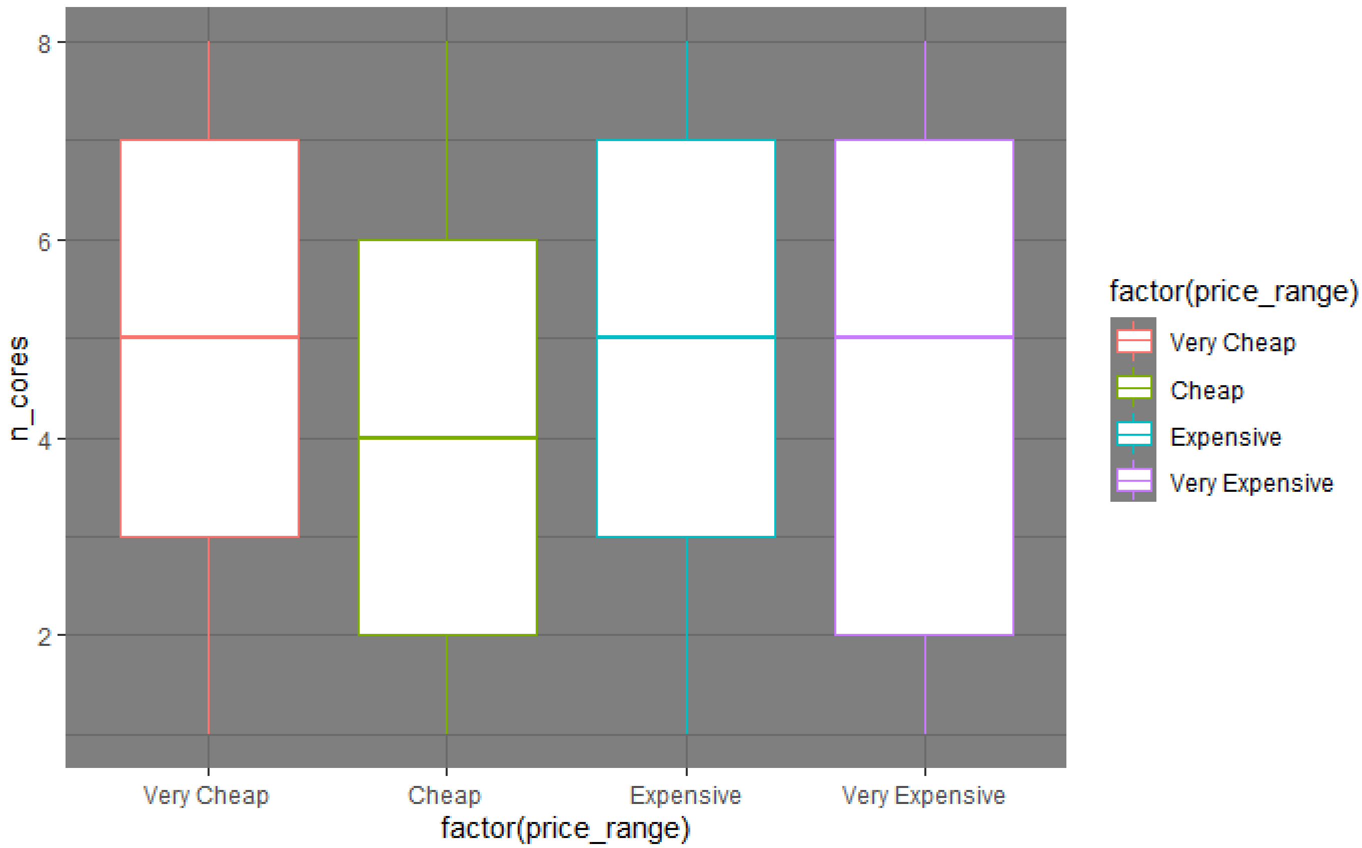
Hide

```
pl3 <- ggplot(phone.data, aes(factor(price_range), ram)) + geom_boxplot(aes(color = factor(price
_range))) + theme_dark()
(pl3)
```



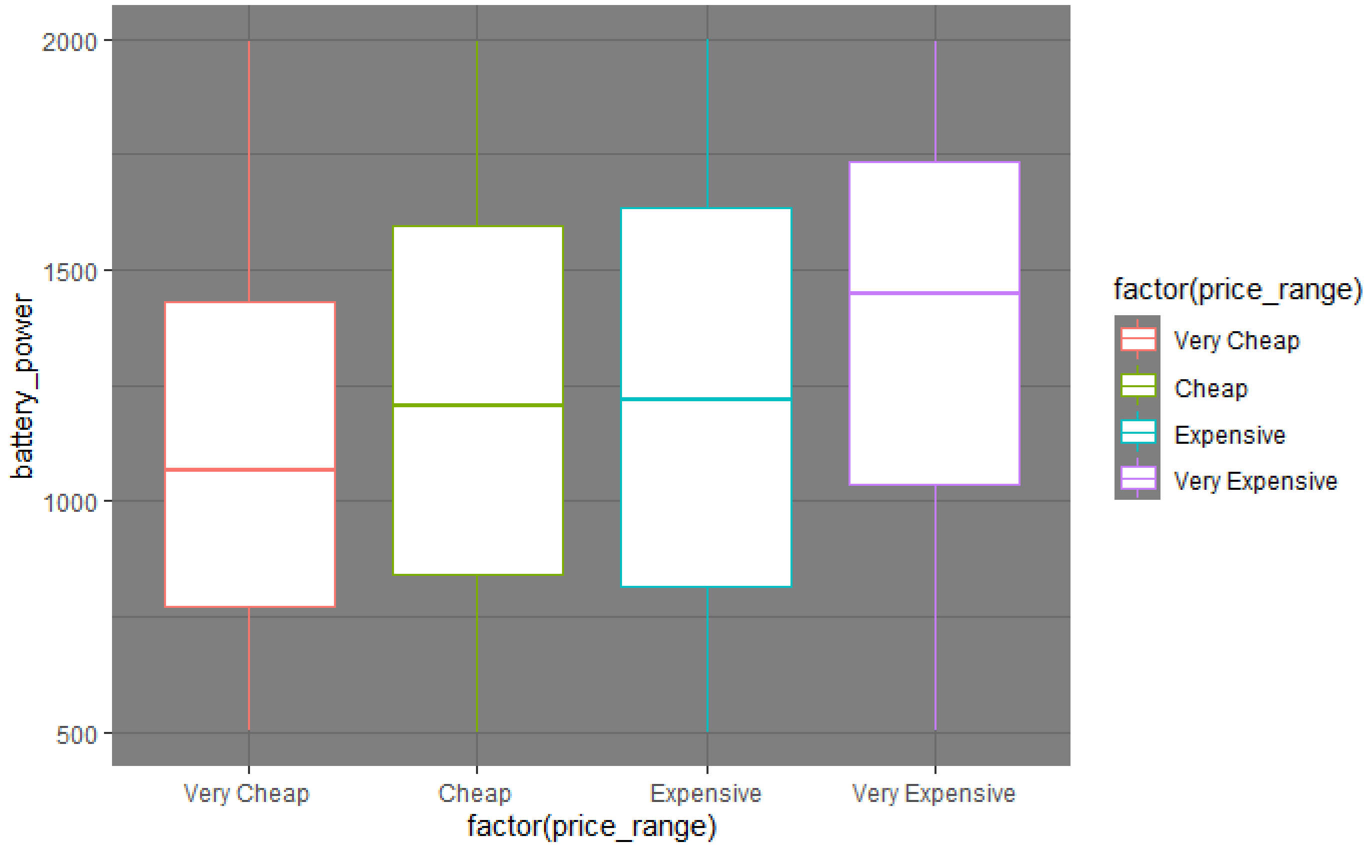
Hide

```
pl4 <- ggplot(phone.data, aes(factor(price_range), n_cores)) + geom_boxplot(aes(color = factor(p
rice_range))) + theme_dark()
(pl4)
```



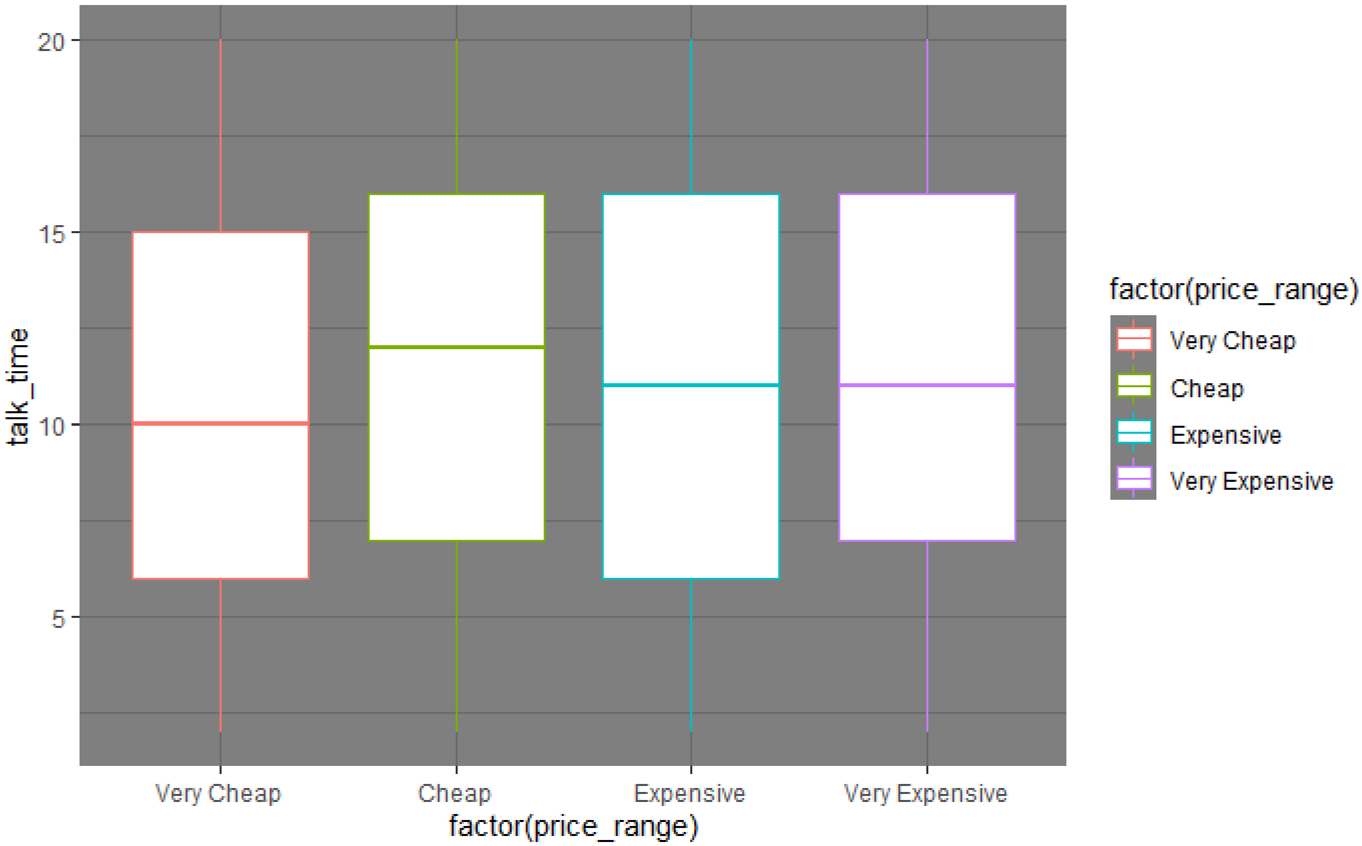
Hide

```
pl4.a <- ggplot(phone.data, aes(factor(price_range), battery_power)) + geom_boxplot(aes(color =
factor(price_range))) + theme_dark()
(pl4.a)
```



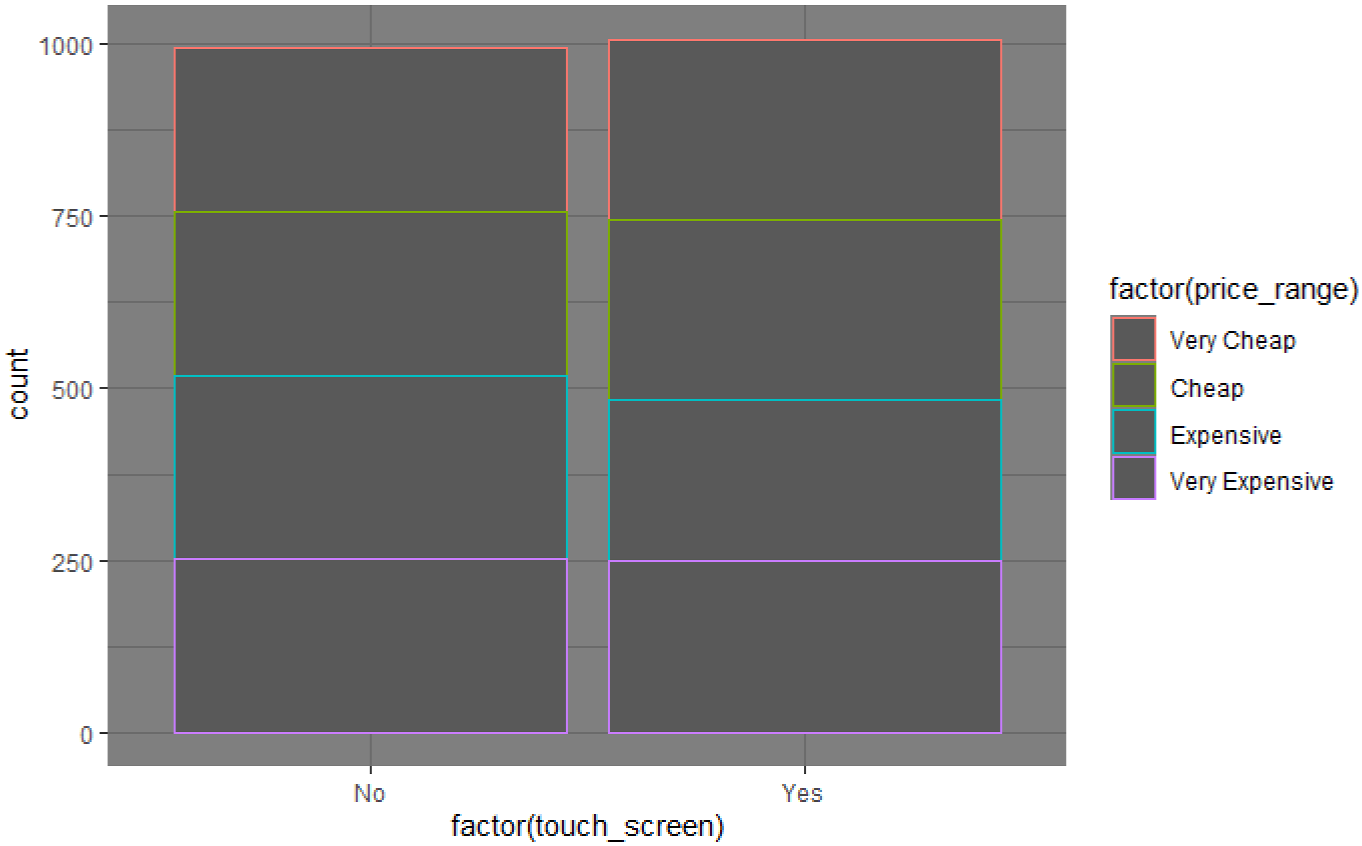
Hide

```
pl4.b <- ggplot(phone.data, aes(factor(price_range), talk_time)) + geom_boxplot(aes(color = fact
or(price_range))) + theme_dark()
(pl4.b)
```



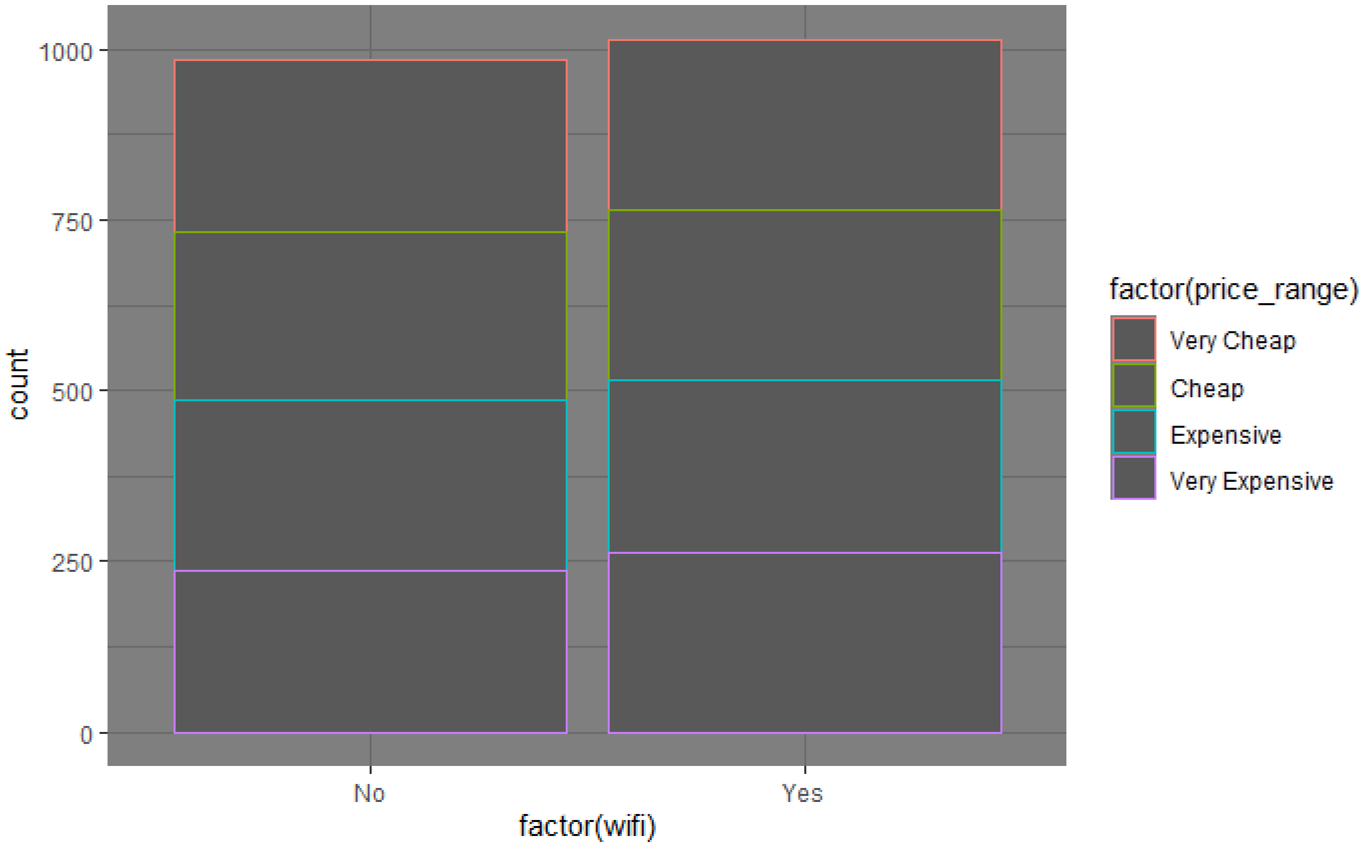
Hide

```
pl5 <- ggplot(phone.data, aes(x=factor(touch_screen))) + geom_bar(aes(color = factor(price_rang
e)))+ theme_dark() #+ facet_wrap(~touch_screen)
(pl5)
```



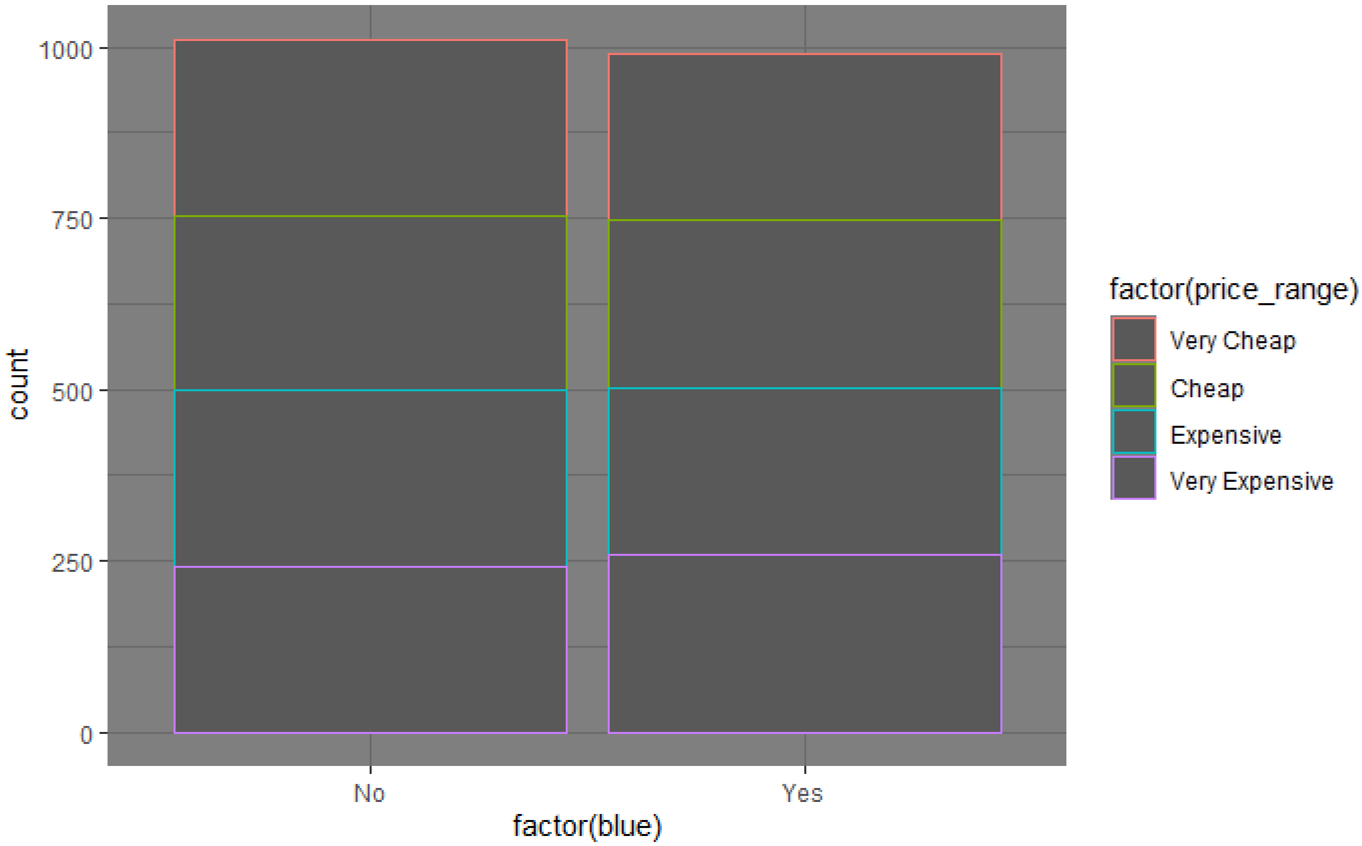
Hide

```
pl6 <- ggplot(phone.data, aes(x=factor(wifi))) + geom_bar(aes(color = factor(price_range)))+ the
me_dark()
(pl6)
```



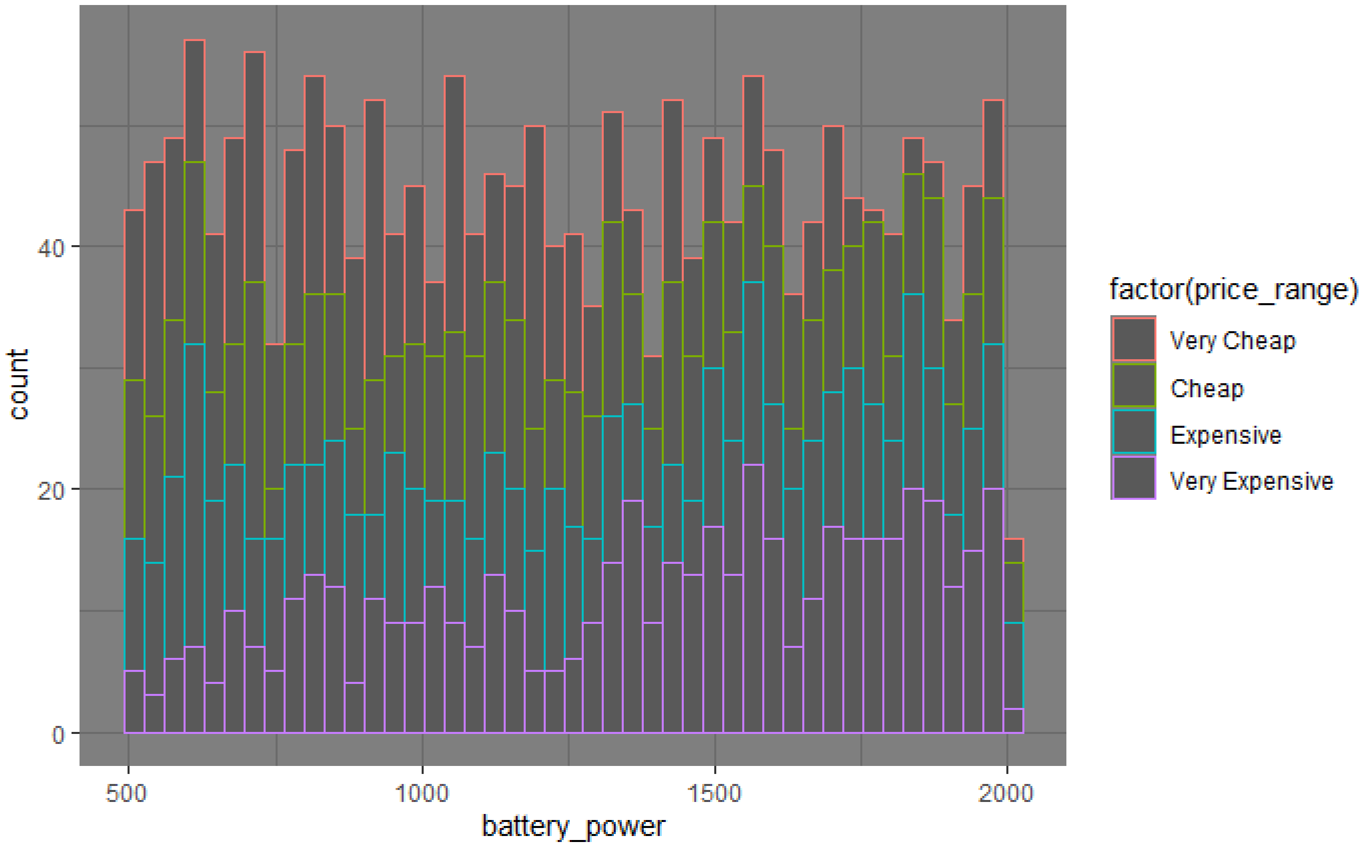
Hide

```
pl6.a <- ggplot(phone.data, aes(x=factor(blue))) + geom_bar(aes(color = factor(price_range)))+ t
heme_dark()
(pl6.a)
```



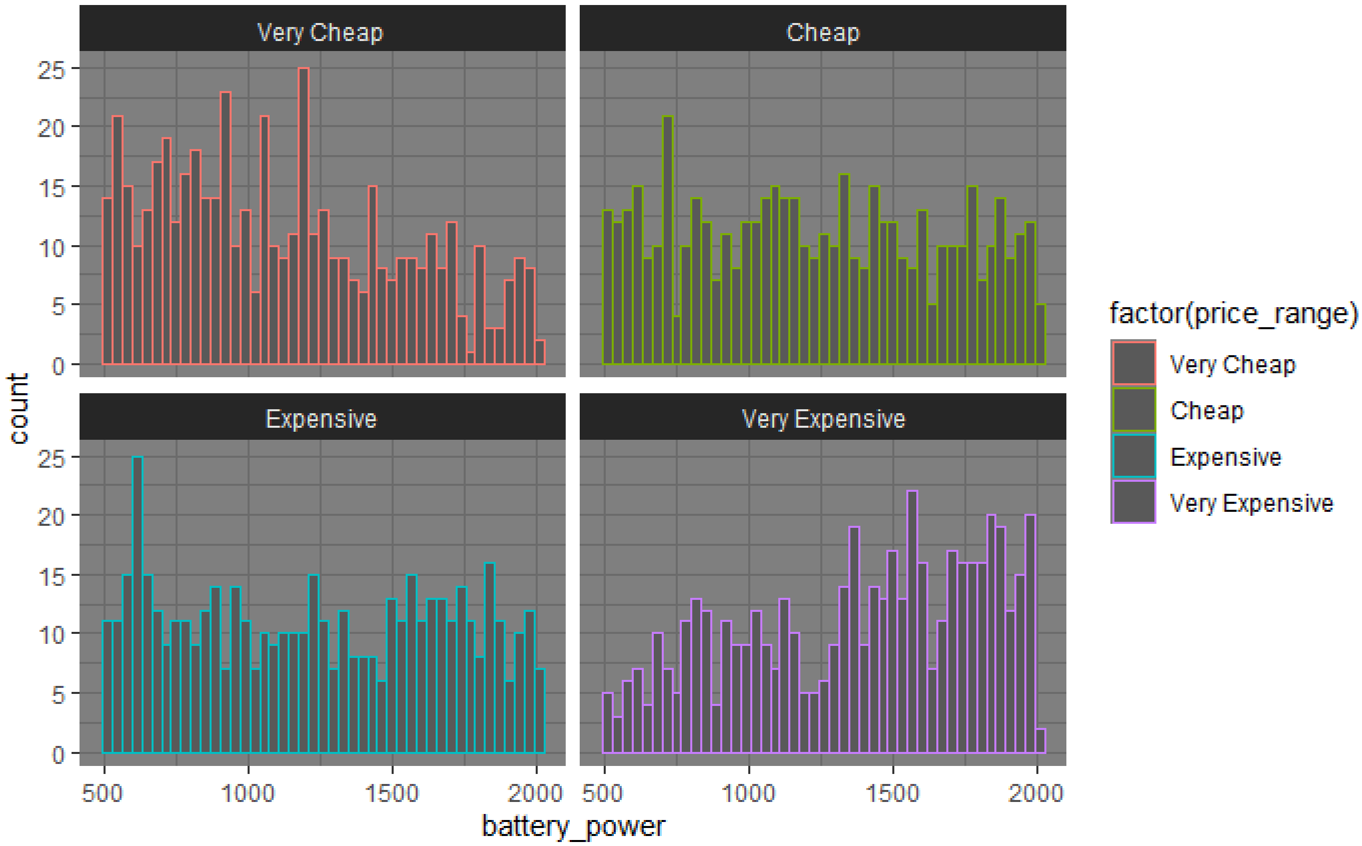
Hide

```
pl7 <- ggplot(phone.data, aes(battery_power)) + geom_histogram(aes(color = factor(price_range)),
bins = 45) + theme_dark() #+ facet_wrap(~price_range)
pl7
```



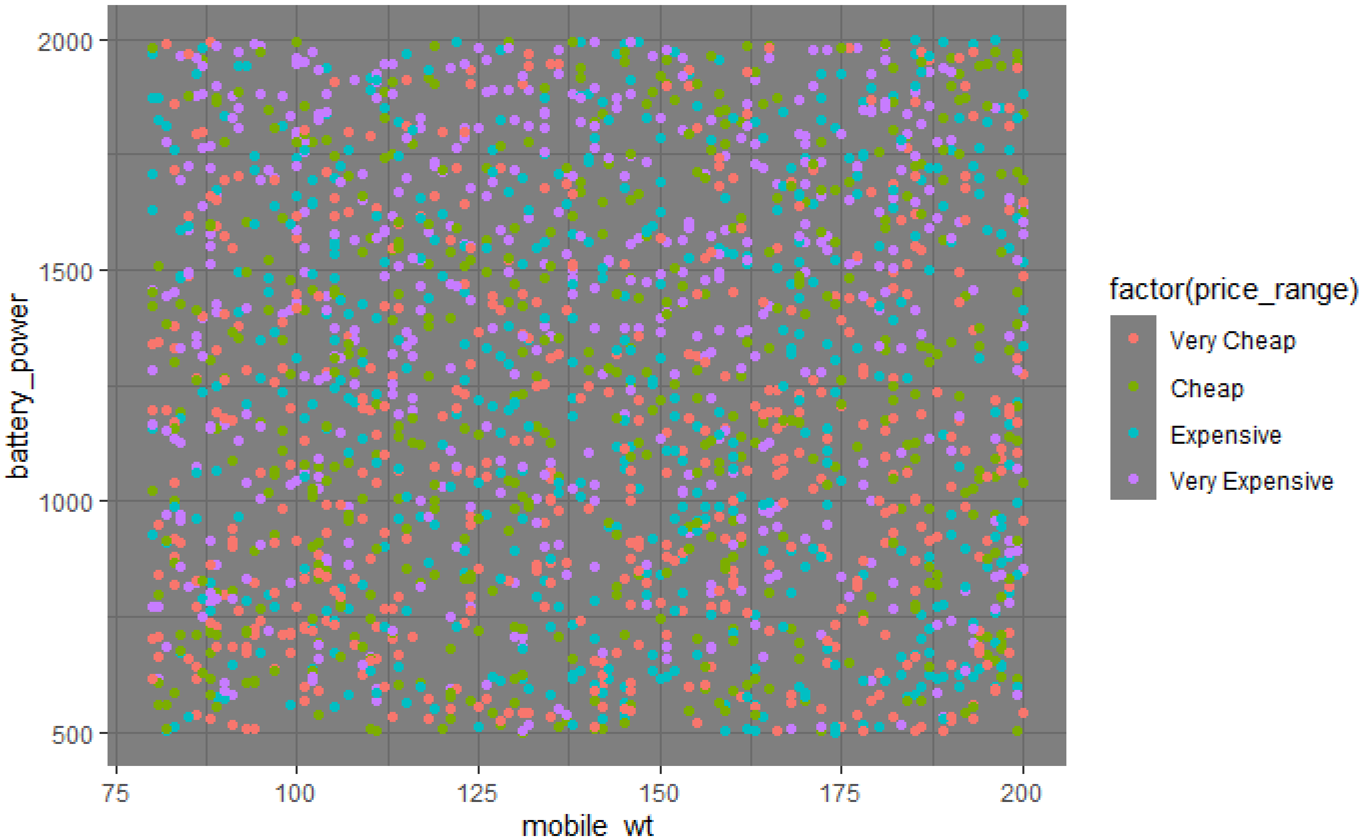
Hide

```
pl7.quad <- ggplot(phone.data, aes(battery_power)) + geom_histogram(aes(color = factor(price_ran
ge)),bins = 45) + theme_dark() + facet_wrap(~price_range)
pl7.quad
```



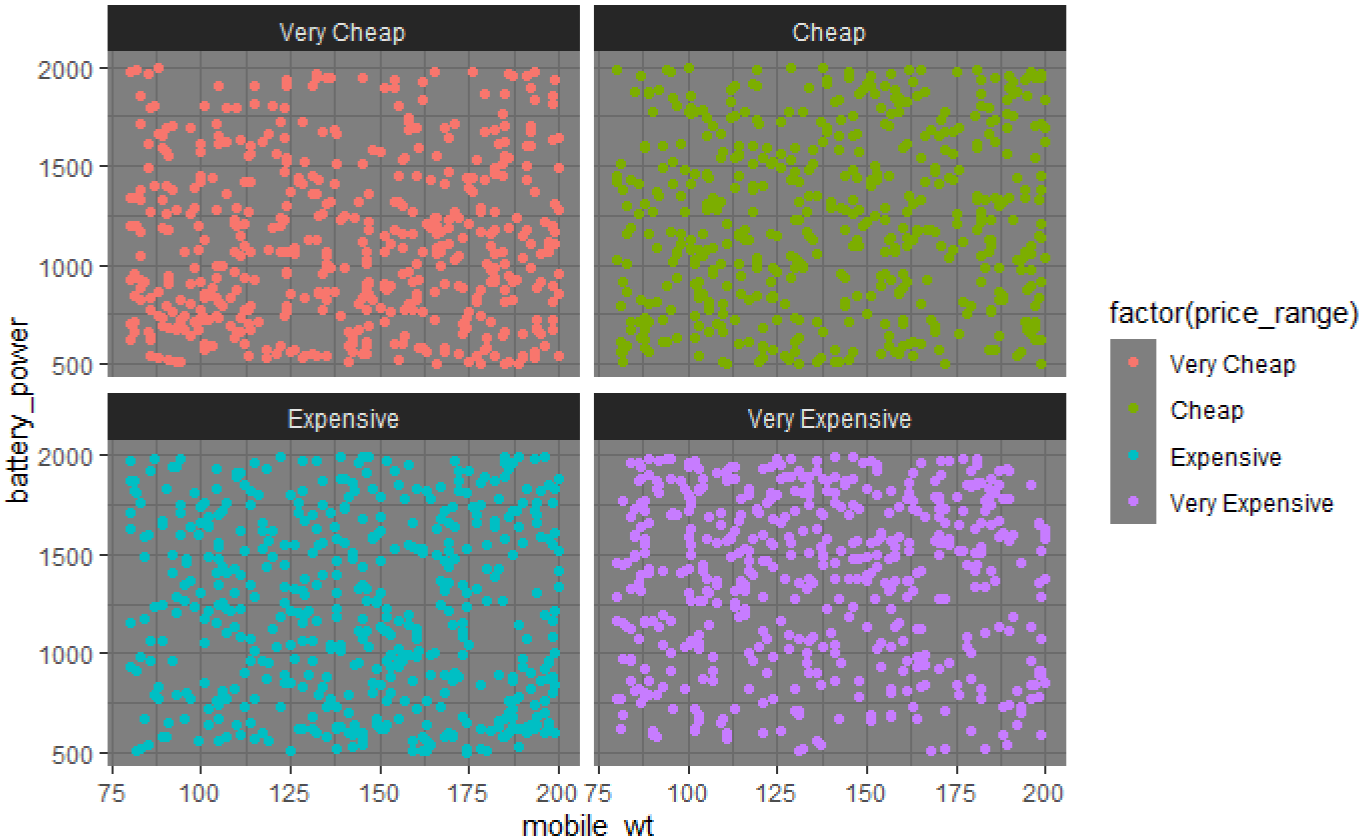
Hide

```
pl8 <- ggplot(phone.data, aes(x=mobile_wt, y=battery_power)) + geom_point(aes(color = factor(pri
ce_range))) + theme_dark()
pl8
```



Hide

```
pl8.quad <- ggplot(phone.data, aes(x=mobile_wt, y=battery_power)) + geom_point(aes(color = facto
r(price_range))) + theme_dark() + facet_wrap(~price_range)
pl8.quad
```



There is plenty to sift through with the visualizations above. Let’s look at some numerical attributes in the data.

The summary command gives quite a lot of information for each attribute.

Hide

```
summary(phone.data)
```

```
 battery_power     blue      clock_speed    dual_sim        fc          four_g   
 Min.   : 501.0   No :1010   Min.   :0.500   No : 981   Min.   : 0.000   No : 957  
 1st Qu.: 851.8   Yes: 990   1st Qu.:0.700   Yes:1019   1st Qu.: 1.000   Yes:1043  
 Median :1226.0              Median :1.500              Median : 3.000             
 Mean   :1238.5              Mean   :1.522              Mean   : 4.309             
 3rd Qu.:1615.2              3rd Qu.:2.200              3rd Qu.: 7.000             
 Max.   :1998.0              Max.   :3.000              Max.   :19.000             
   int_memory        m_dep          mobile_wt        n_cores           pc        
 Min.   : 2.00   Min.   :0.1000   Min.   : 80.0   Min.   :1.000   Min.   : 0.000  
 1st Qu.:16.00   1st Qu.:0.2000   1st Qu.:109.0   1st Qu.:3.000   1st Qu.: 5.000  
 Median :32.00   Median :0.5000   Median :141.0   Median :4.000   Median :10.000  
 Mean   :32.05   Mean   :0.5018   Mean   :140.2   Mean   :4.521   Mean   : 9.916  
 3rd Qu.:48.00   3rd Qu.:0.8000   3rd Qu.:170.0   3rd Qu.:7.000   3rd Qu.:15.000  
 Max.   :64.00   Max.   :1.0000   Max.   :200.0   Max.   :8.000   Max.   :20.000  
   px_height        px_width          ram            sc_h            sc_w       
 Min.   :   0.0   Min.   : 500.0   Min.   : 256   Min.   : 5.00   Min.   : 0.000  
 1st Qu.: 282.8   1st Qu.: 874.8   1st Qu.:1208   1st Qu.: 9.00   1st Qu.: 2.000  
 Median : 564.0   Median :1247.0   Median :2146   Median :12.00   Median : 5.000  
 Mean   : 645.1   Mean   :1251.5   Mean   :2124   Mean   :12.31   Mean   : 5.767  
 3rd Qu.: 947.2   3rd Qu.:1633.0   3rd Qu.:3064   3rd Qu.:16.00   3rd Qu.: 9.000  
 Max.   :1960.0   Max.   :1998.0   Max.   :3998   Max.   :19.00   Max.   :18.000  
   talk_time     three_g    touch_screen  wifi              price_range 
 Min.   : 2.00   No : 477   No : 994     No : 986   Very Cheap    :500  
 1st Qu.: 6.00   Yes:1523   Yes:1006     Yes:1014   Cheap         :500  
 Median :11.00                                      Expensive     :500  
 Mean   :11.01                                      Very Expensive:500  
 3rd Qu.:16.00                                                          
 Max.   :20.00                                                          
```

We can actually create a model to predict what price range a phone will be in based on certain characteristics. There was some obvious clustering in one of the first plots produced in this document.

We can use k-means clustering to see how similar some points are to others in their clusters.

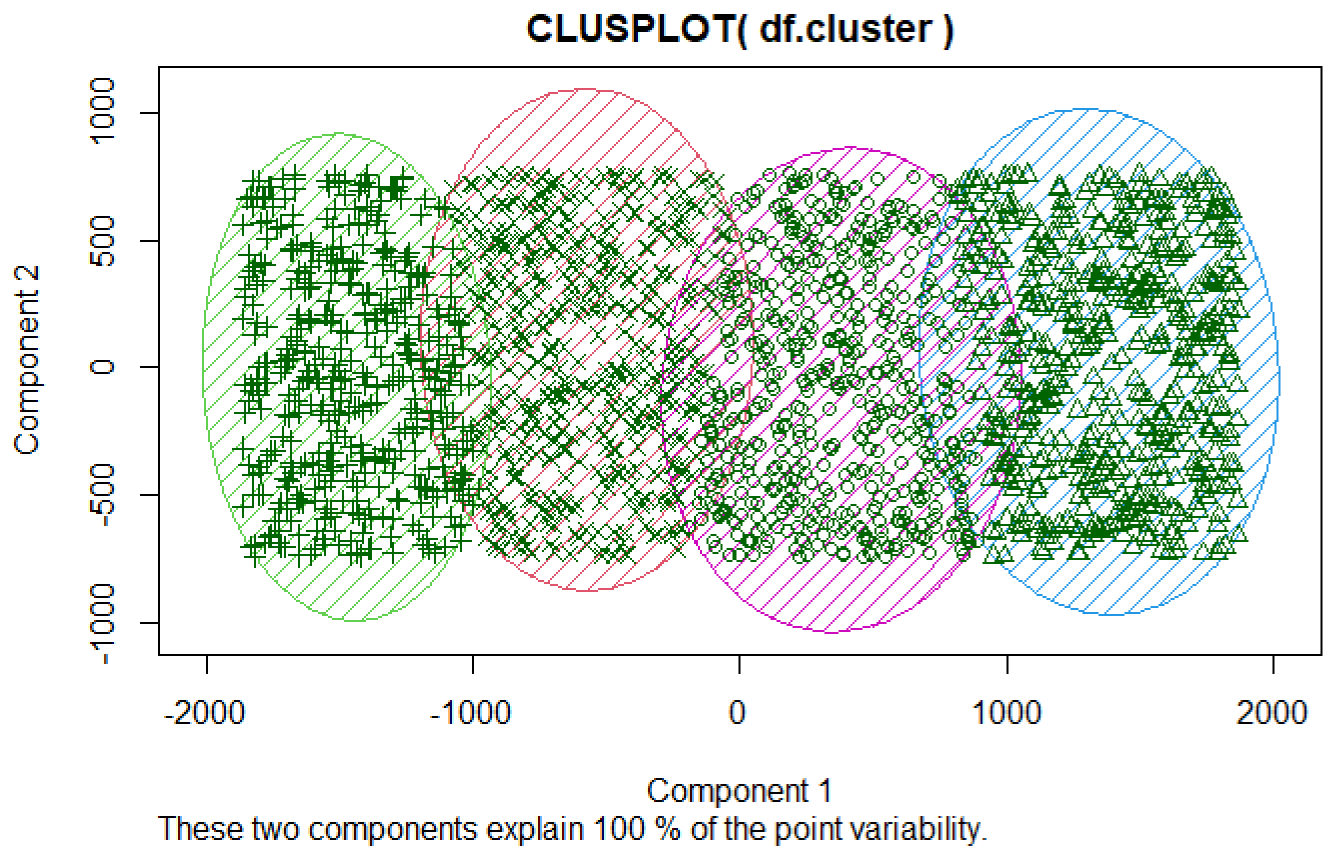
Hide

```
library(cluster)

df.cluster <- data.frame(phone.data$ram, phone.data$battery_power)
cluster.price <- kmeans(df.cluster, 4, nstart = 20)
```

Hide

```
clusplot(df.cluster, cluster.price$cluster, color = TRUE, shade = TRUE, labels = 0, lines = 0)
```



Hide

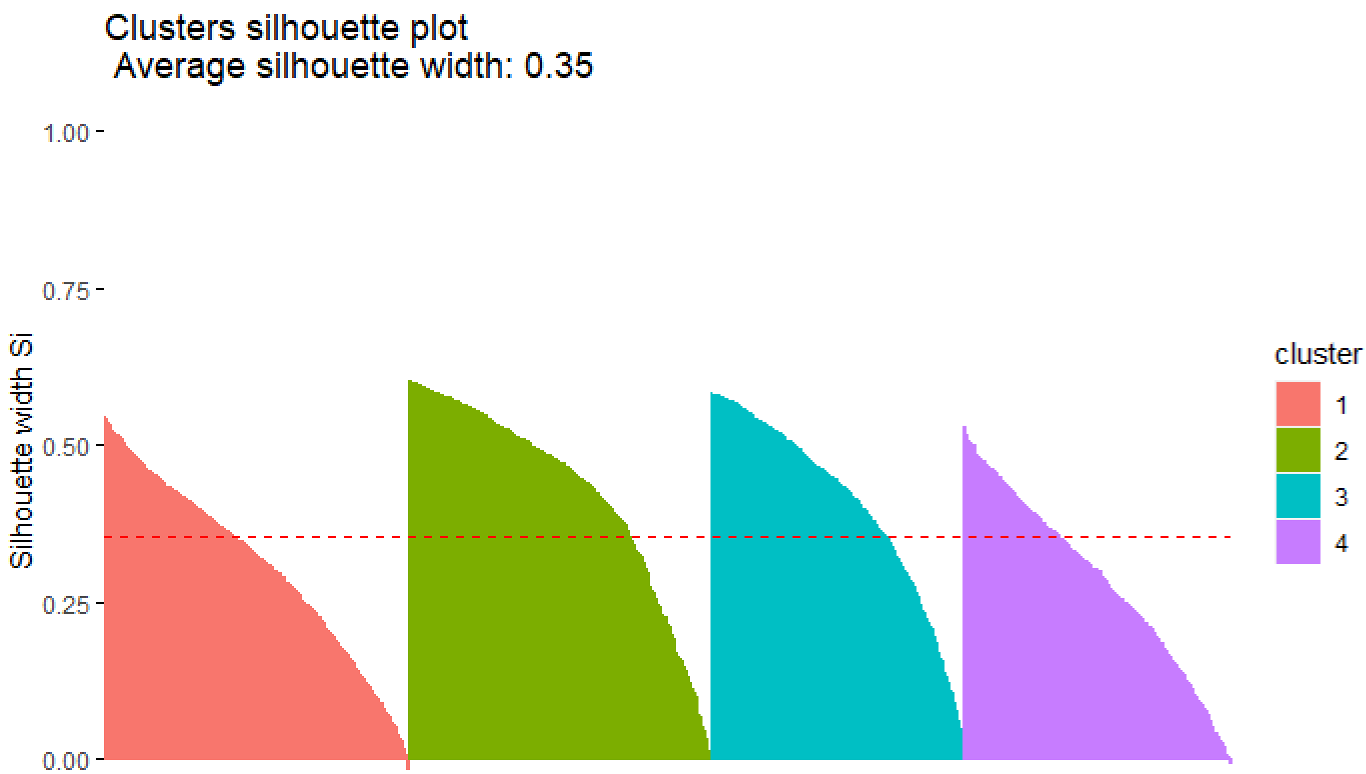
```
library(factoextra)

sil <- silhouette(cluster.price$cluster, dist(df.cluster))


fviz_silhouette(sil)
```

	cluster <fctr>	size <int>	ave.sil.width <dbl>
1	1	540	0.31
2	2	536	0.42
3	3	449	0.40
4	4	475	0.28

4 rows



0.35 average silhouette width is not ideal, but it still shows there is pretty significant clustering. An average above 0.5 is what we would hope to find to make an even stronger case for the model. Based on the data frame that comes with our silhouette, we can actually see that clusters 2 and 3 were more sound than 1 and 4. There was some overlap in the model's classification between those two clusters.

Hide

```
table(cluster.price$cluster, phone.data$price_range)
```

```
   Very Cheap Cheap Expensive Very Expensive
  1          1   141       337             61
  2          0     0        97            439
  3        404    45         0              0
  4         95   314        66              0
```

This table is a good way to show the overlap in the model. It shows how the k-means clustering compares with the actual data. This helps our case in showing that RAM is a good predictor variable for finding a price range of a phone. As shown in the visualizations above, many of the other variables seem to have little or no effect on whether the phone is expensive or cheap.

For each of the plots, I have colored each individual phone by their categorical price range to make it easier on the viewer for interpreting the plots. The price range originally came in an integer form, from 0 to 3, with 0 being the least expensive and 3 being the most expensive. I updated all of the categorical variables in the data frame to make the visualizations more readable as well. For example, when examining the bar graph that shows the number of phones with or without touch screen capabilities, instead of having two integer values (0, 1), it makes significantly more sense to label them as factors ("No", "Yes"). This is good practice for professional data visualization and also for readability in general.

Given each of the visualizations above, along with the statistical information retrieved, it is fair to say that RAM is the best predictor for the price range of a phone. This means that, given a phone's RAM, we are able to give a relatively accurate prediction for what price range that phone may lie within. There was clear and obvious clustering occurring in that plot. Battery power appeared to have very little effect on the price range of the phones. As seen in the histograms involving battery power, the "Very Expensive" phones tend to have larger batteries than the "Very Cheap" devices, but this is not a significant predictor. As the phones had larger batteries, for each price range, the amount of RAM on the machine would drop. For instance, a device in the "Cheap" price range with a battery power of 500 would likely have between 1500 MB and 2500 MB of of RAM. On the other hand, another phone within the same "Cheap" price range with battery power of 2000 would most likely have between 750 MB and 1250 MB of RAM. This can clearly be seen by the pattern/clustering in the plot. Individually, each price range has a relatively linear correlation in relation to RAM.

In regards to screen sizes, there are no obvious correlations based on price range there. When split by price range, the data is almost uniform. The data had an interesting shape, however. Pixel height and pixel length have a certain cutoff; for example, a screen with a pixel width of 1000 will not have a pixel height greater than 1000. Based on the plot, all the data from this set follow this pattern.

Something interesting I noticed in the numerical summaries of each variable is that all but one of the categorical attributes had seemingly very even splits. The one variable that did not have an even split between its categories is the three_g attribute. 3 times as many phones in the dataset had 3G compared to those that did not. Every other categorical variable was split nearly perfectly down the line. In the case of the price range, each category had 500 observations. Given the background story behind the data, this makes sense.

Some other oddities within the data include the following: a device with a screen width of 0 was found. This must mean that the phone does not have a screen, otherwise, that tuple is invalid. On the other hand, the minimum screen height was 5. Something strange is occurring behind the scenes on the far minimum end of this attribute in the dataset. This would not have been brought to light without viewing the five-number summaries of this variable. Another oddity is the minimum pixel height found on a device turned out to be 0. This is questionable, much like the preceding few statements.

It should also be noted that, in general, most of the numeric types of data in the dataset appear to be relatively uniformly distributed. That is, the data is spread pretty evenly regardless of how many standard deviations each point resides from the mean of each attribute. There are fewer observations beyond 2 or 3 standard deviations away from the mean, but this number is not significantly less than what can be found within 1 or 2 standard deviations. This uniform distribution was absolutely the case for battery power, wifi, touch screen, weight, bluetooth, and more. Notice that in many of the visualizations above, price range was not affected whatsoever by these variables. Each price range had a similar amount of each level in each attribute. This is actually quite revealing for many other reasons. It goes to show that the only main difference between the most expensive and least expensive phones is the amount of RAM installed on the device. Otherwise, a customer is basically getting the exact same specifications regardless of how much they pay.