Multiple Statistical Models and Bootstrapping Unit Summary

Yogindra Raghav

11/16/2018

Load package `tidyverse`. We will use `diamonds` data set in this quiz.

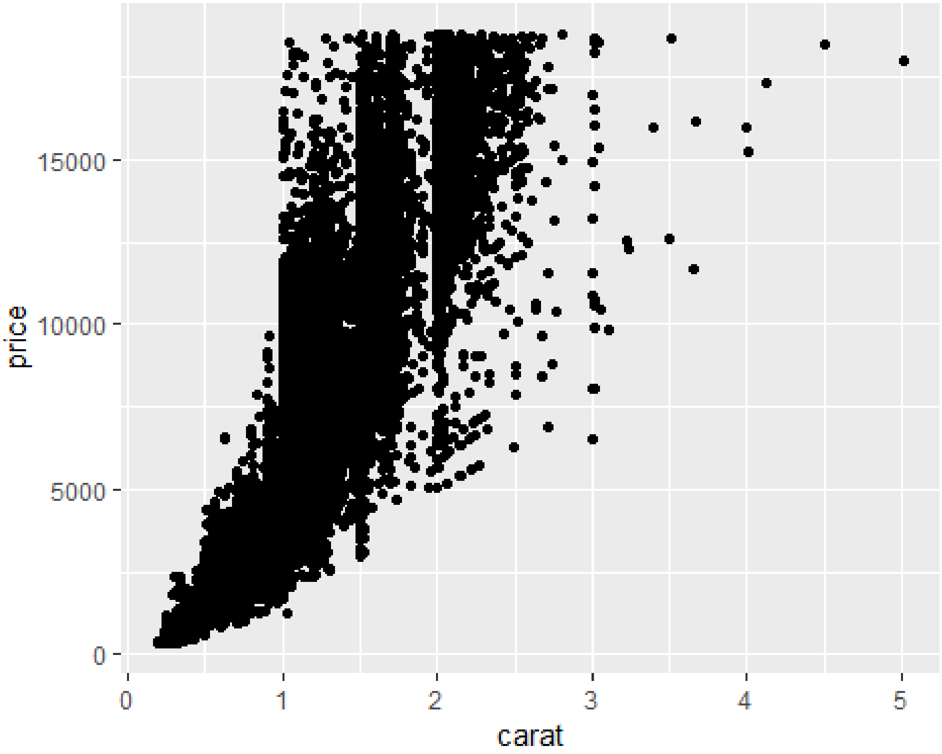
```r
library(tidyverse)
```

Suppose we are interested in the relationship of diamond's weight (carat) and price.

First, create a scatterplot using `ggplot()` to visualize the relationship between carat and price.

```r
diamond_scatter = ggplot(data = diamonds)+ geom_point(aes(carat, price))

diamond_scatter
```



The scatterplot shows a non-linear relationship. Transform both variables by taking log and add them to the original dataset. Name the new dataset `diamonds1`. Keep only the two new variables and discard all other variables. Show the first few rows of `diamonds1` using `head()` function.

```
diamonds1 = diamonds %>% mutate(log2carat = log2(carat), log2price = log2(pri
ce)) %>% select(log2price, log2carat)
```

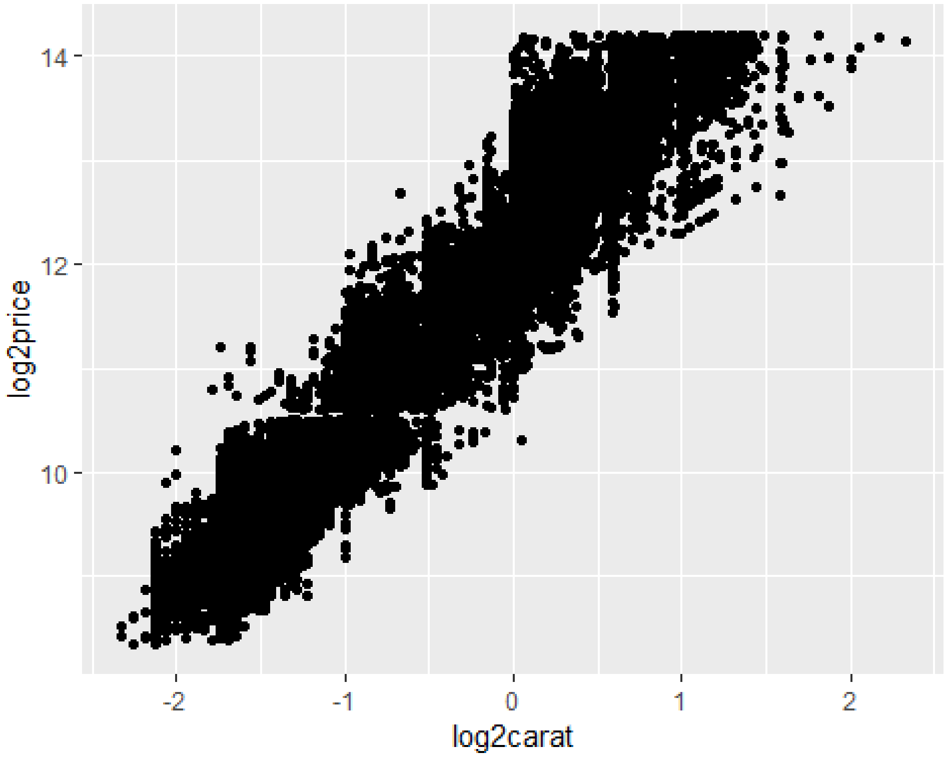
```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
head(diamonds1)
```

```
## # A tibble: 6 x 2
##   log2price log2carat
##       <dbl>     <dbl>
## 1      8.35     -2.12
## 2      8.35     -2.25
## 3      8.35     -2.12
## 4      8.38     -1.79
## 5      8.39     -1.69
## 6      8.39     -2.06
```

Create a scatterplot to show the relationship between the transformed variables.

```
ggplot(data = diamonds1) + geom_point(aes(log2carat, log2price))
```



Now you can observe a very strong linear association between log(carat) and log(price). Randomly sample 1000 observations from diamonds1 (using sample_n() function) and name it sample. Set seed to be 100 so that everyone's result will be the same.

```
set.seed(100)
sample = diamonds1 %>% sample_n(size = 1000)
sample
```

```
## # A tibble: 1,000 x 2
##    log2price log2carat
##        <dbl>     <dbl>
##  1     12.7     0.0144
##  2     12.5    -0.152
##  3      9.47   -1.74
##  4      9.14   -1.74
##  5     13.8     1.04
##  6     13.9     0.642
##  7     10.5    -0.971
##  8     13.1     0.214
##  9      9.46   -1.64
## 10     12.1     0.111
## # ... with 990 more rows
```

Fit a simple linear regression model on the sample using lm() and use summary() to show the regression results.

```
mod = lm(log2price~log2carat, data = sample)
summary(mod)
```

```
##
## Call:
## lm(formula = log2price ~ log2carat, data = sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.55725 -0.25264  0.00037  0.25787  1.48372
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.19245    0.01456   837.5   <2e-16 ***
## log2carat    1.69046    0.01450   116.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3819 on 998 degrees of freedom
## Multiple R-squared:  0.9316, Adjusted R-squared:  0.9315
## F-statistic: 1.359e+04 on 1 and 998 DF,  p-value: < 2.2e-16
```

Construct a 95% confidence interval for the slope of log(carat) using traditional way (assuming the error follows normal distribution with mean 0). Does the confidence interval imply that the slope is significantly different from 0?

```
1.69046+2*0.01450 *c(-1,1)

## [1] 1.66146 1.71946
```

The confidence interval does imply that the slope is significantly different from 0 because 0 is not contained within the confidence interval.

Use bootstrap to create a 95% confidence interval. You may use 1000 iterations of bootstraping. You may either using the first or second method in Slide #20 of Lecture 19. You'll need to load package `broom`.

```
library(broom)

boot_diamonds = diamonds1 %>%  bootstrap(1000) %>% do(tidy(lm(log2price ~ log
2carat, .)))

boot_diamonds %>%  ungroup() %>% filter(term == "log2carat") %>% summarize(LL
= quantile(estimate,0.025), UL= quantile(estimate,0.975))

## # A tibble: 1 x 2
##      LL    UL
##   <dbl> <dbl>
## 1  1.67  1.68
```