HW4

Ting Huang

Monday, February 15, 2016

Homework 4

Problem 2

'KNNL' refers to the book by Kutner, Nachtsheim, neter and Li.

(a) The objection is inappropriate. The hardness of the plastic is not only affected by the elapsed time but also by other factors, such as the environment temperature. The simple linear regression assumes that the combination of all the other factors results in a non-systematic effect.

(b) No. $E\{Y_i\} = \beta_0 + \beta_1 X_i$. It cannot contain a random term.

(c) The least squares method doesn't require the distribution of Y to be normal.

(d) No. Because ϵ is a random variable, that has an infinite number of instances. It is characterized by the expected value, but not by the sum.

(e)

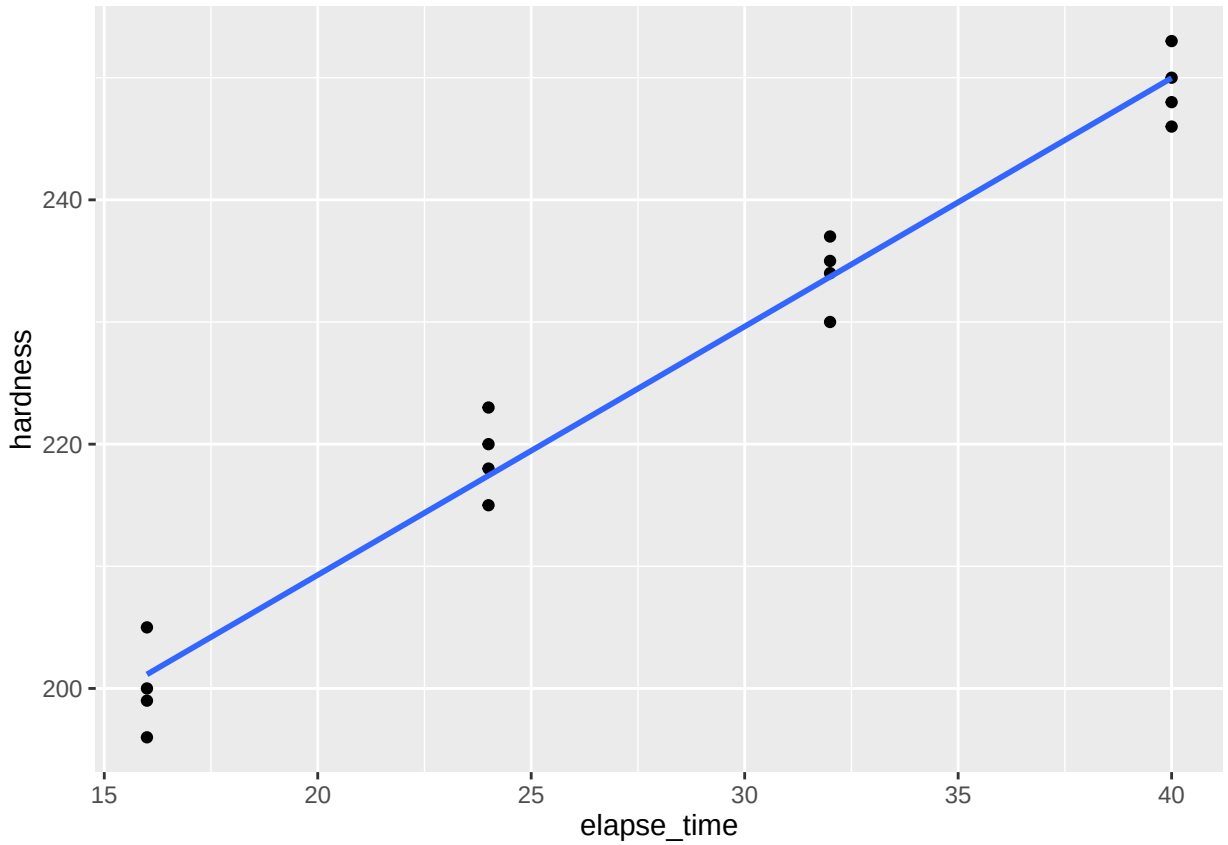
```
X<-c(16.0,16.0,16.0,16.0,24.0,24.0,24.0,24.0,32.0,32.0,32.0,32.0,40.0,40.0,40.0,40.0)
Y<-c(199.0,205.0,196.0,200.0,218.0,220.0,215.0,223.0,237.0,234.0,235.0,230.0,250.0,248.0,253.0,246.0)
plastic<-data.frame("elapse_time"=X,"hardness"=Y)
plastic.lm <- lm(hardness ~ elapse_time, data=plastic)
```

(e.1) The estimated regression function is $\hat{Y} = 168.6 + 2.034375X$.

```
#install.packages("ggplot2")
library(ggplot2)
coefficients(plastic.lm)
```

```
## (Intercept) elapse_time
##  168.600000    2.034375
```

```
ggplot(plastic, aes(elapse_time, hardness))+geom_point() + geom_smooth(method = "lm", se = FALSE)
```



The linear regression function appears to give a good fit here.

(e.2) When $X = 40, \hat{Y} = 168.6 + 2.034375 * 40 = 249.975$

(e.3) 2.034375.

(f) KNNL 1.26

```
residuals<-resid(plastic.lm)
sum(residuals)
```

```
## [1] -1.998401e-15
```

The sum of residuals is almost zero.

```
residuals<-resid(plastic.lm)
MSE=sum(residuals^2)/(16-2)
MSE
```

```
## [1] 10.45893
```

```
sqrt(MSE)
```

```
## [1] 3.234027
```

$\hat{\sigma}^2 = 10.45893, \hat{\sigma} = 3.234027$

Problem 3

```
library(tidyr, quietly = TRUE)
library(dplyr, quietly = TRUE)
sestates <- read.csv("http://samplecsvs.s3.amazonaws.com/Sacramentorealestatetransactions.csv", header =
```

(a) Replace the column " sale_date" with three coluns " day_week" , "month" , "day_month"

```
estates <- tbl_df(sestates) %>%
    separate(sale_date, c("day_week", "month", "day_month"), sep = " ", extra = "drop")
data.frame(head(estates))
```

```
##               street       city   zip state beds baths sq__ft        type
## 1       3526 HIGH ST SACRAMENTO 95838    CA    2     1    836 Residential
## 2        51 OMAHA CT SACRAMENTO 95823    CA    3     1   1167 Residential
## 3     2796 BRANCH ST SACRAMENTO 95815    CA    2     1    796 Residential
## 4   2805 JANETTE WAY SACRAMENTO 95815    CA    2     1    852 Residential
## 5    6001 MCMAHON DR SACRAMENTO 95824    CA    2     1    797 Residential
## 6 5828 PEPPERMILL CT SACRAMENTO 95841    CA    3     1   1122       Condo
##   day_week month day_month price latitude longitude
## 1      Wed   May        21 59222 38.63191 -121.4349
## 2      Wed   May        21 68212 38.47890 -121.4310
## 3      Wed   May        21 68880 38.61830 -121.4438
## 4      Wed   May        21 69307 38.61684 -121.4391
## 5      Wed   May        21 81900 38.51947 -121.4358
## 6      Wed   May        21 89921 38.66260 -121.3278
```

(b) What' s the top 10 cities with the most transactions?

```
estates %>%
    group_by(city) %>%
    summarise(n_trans = n()) %>%
    top_n(10, n_trans) %>%
    arrange(desc(n_trans))
```

```
## Source: local data frame [10 x 2]
##
##               city n_trans
##              (chr)   (int)
## 1       SACRAMENTO     439
## 2        ELK GROVE     114
## 3          LINCOLN      72
## 4        ROSEVILLE      48
## 5   CITRUS HEIGHTS      35
## 6         ANTELOPE      33
## 7   RANCHO CORDOVA      28
## 8  EL DORADO HILLS      23
## 9             GALT      21
## 10 NORTH HIGHLANDS      21
```

(c) What' s the accumulated number of transactions from May 15 to May 21 in city ELK GROVE?

```
estates %>%
    filter(city == "ELK GROVE") %>%
    group_by(month, day_month) %>%
    summarise(n_trans = n()) %>%
    mutate(n_cumtrans = cumsum(n_trans))
```

```
## Source: local data frame [5 x 4]
## Groups: month [1]
##
##   month day_month n_trans n_cumtrans
##   (chr)     (chr)   (int)      (int)
## 1   May        15      13         13
## 2   May        16      28         41
## 3   May        19      19         60
## 4   May        20      22         82
## 5   May        21      32        114
```

(d) For each type of house (Condo, Multi-Family, and Residential), what' s the highest 3 transaction prices? In which cities?

```
estates %>%
    filter(type != "Unkown") %>%
    group_by(type) %>%
    top_n(3, price) %>%
    select(type, city, price) %>%
    arrange(type, desc(price))
```

```
## Source: local data frame [9 x 3]
## Groups: type [3]
##
##           type            city  price
##          (chr)           (chr)  (int)
## 1        Condo      SACRAMENTO 360000
## 2        Condo       ROSEVILLE 350000
## 3        Condo      GOLD RIVER 300000
## 4 Multi-Family      SACRAMENTO 416767
## 5 Multi-Family      SACRAMENTO 297000
## 6 Multi-Family          AUBURN 285000
## 7  Residential          WILTON 884790
## 8  Residential EL DORADO HILLS 879000
## 9  Residential          LOOMIS 839000
```

(e) Are the values in column sq__ft looking okay? If not, what changes need to be made? For each type of house (Condo, Multi-Family, and Residential), what' s the top 3 cities with the highest price per square foot?

```
summary(estates$sq__ft)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     952    1304    1315    1718    5822
```

Not possible to have a house of square foot zero. We should set the zero values to NA (missing value). Do so and return summary output.

```
estates$sq__ft[estates$sq__ft == 0] <- NA
summary(estates$sq__ft)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     484    1144    1418    1591    1851    5822     171
```

```
estates %>%
    filter(type != "Unkown") %>%
    mutate(price_sq = price / sq__ft) %>%
    group_by(type) %>%
    top_n(3, price_sq) %>%
    select(type, city, price_sq) %>%
    arrange(type, desc(price_sq))
```

```
## Source: local data frame [9 x 3]
## Groups: type [3]
##
##           type       city price_sq
##          (chr)      (chr)    (dbl)
## 1        Condo SACRAMENTO 304.9233
## 2        Condo SACRAMENTO 214.3740
## 3        Condo     AUBURN 207.1713
## 4 Multi-Family     AUBURN 296.8750
## 5 Multi-Family SACRAMENTO 238.9719
## 6 Multi-Family SACRAMENTO 134.2070
## 7  Residential SACRAMENTO 619.6660
## 8  Residential     LOOMIS 516.6256
## 9  Residential SACRAMENTO 459.2652
```

(f) For each type of house (Condo, Multi-Family, and Residential) in city SACRAMENTO, what' s the number of transactions, average price, and average price per square foot?

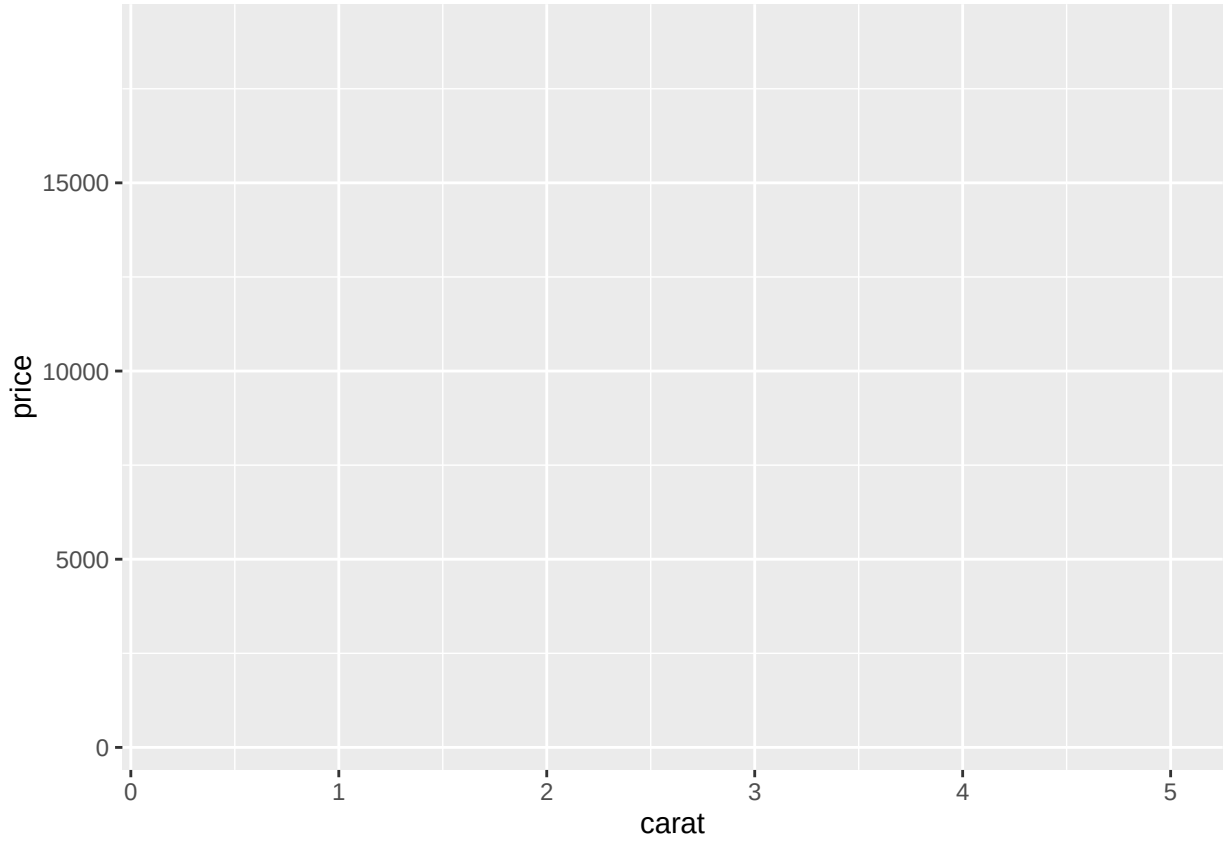
```
estates %>%
    filter(city == "SACRAMENTO") %>%
    mutate(price_sq = price / sq__ft) %>%
    group_by(type) %>%
    summarise(n_trans = n(), ave_price = mean(price),
              ave_price_sq = mean(price_sq, na.rm = TRUE))
```

```
## Source: local data frame [3 x 4]
##
##           type n_trans ave_price ave_price_sq
##          (chr)   (int)     (dbl)        (dbl)
## 1        Condo      27  137690.7     132.8999
## 2 Multi-Family      10  214189.7     104.5987
## 3  Residential     402  201359.6     137.5226
```

Problem 4

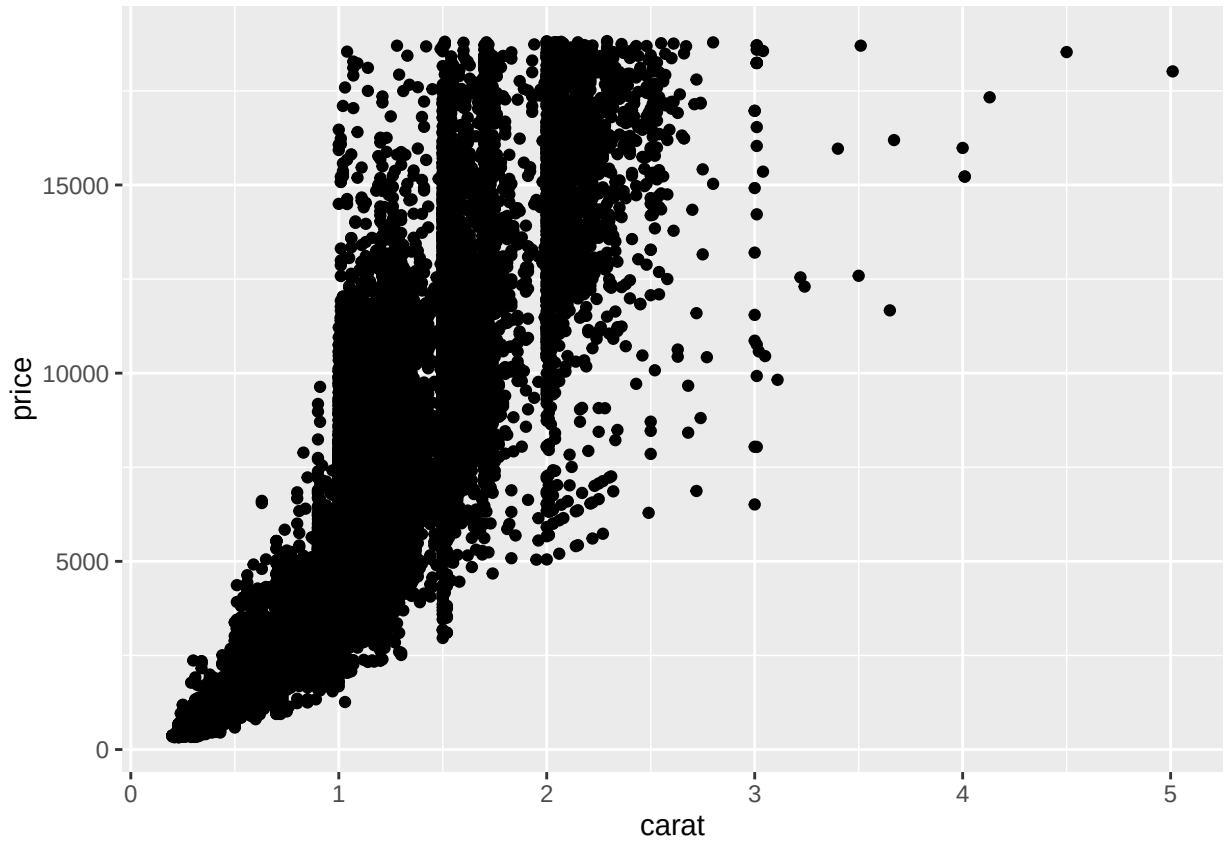
(a) Initialize a ggplot object, map carat to the x axis and price to the y axis. You should get the following output.

```
library(ggplot2, quietly = TRUE)
data(diamonds)
p <- ggplot(data=diamonds, aes(x=carat, y=price))
p
```



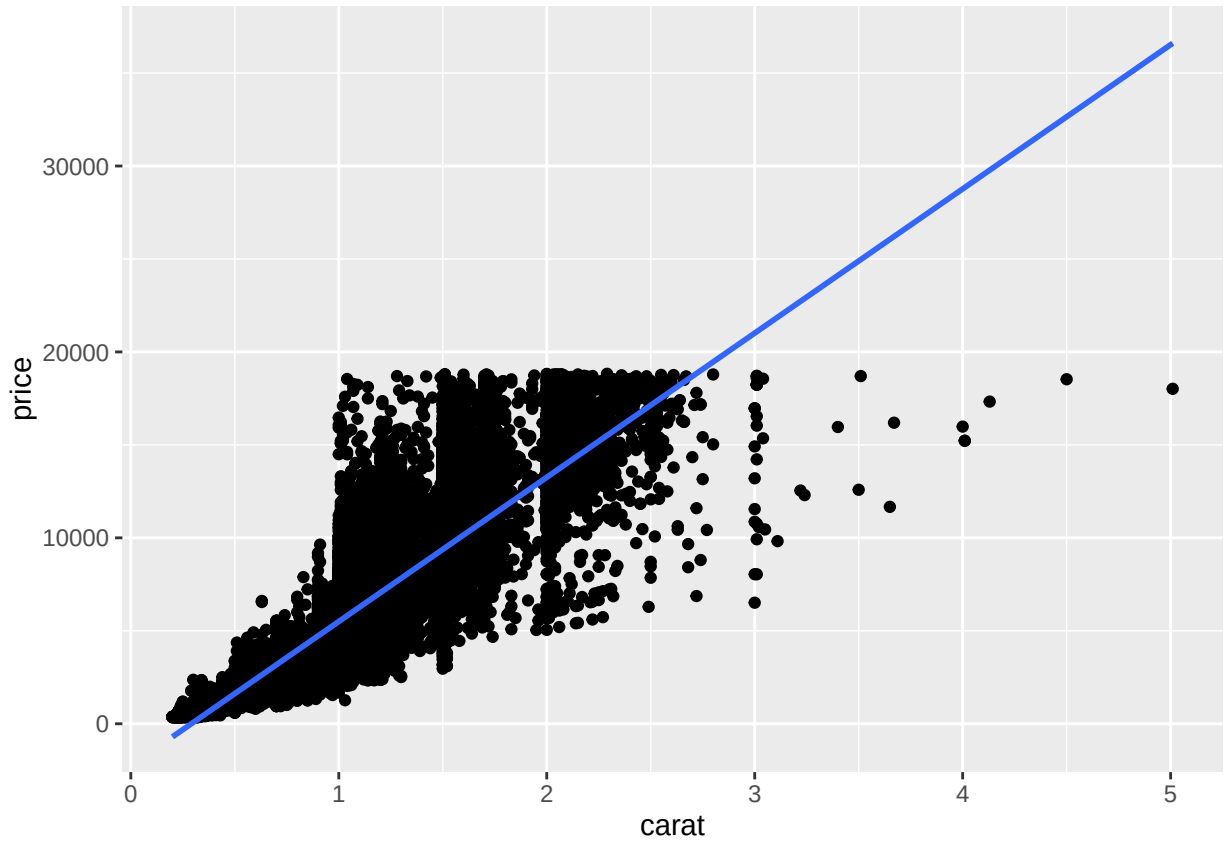
(b) Next, plot the points. (hint: "geom_points")

```
p <- p + geom_point()
p
```



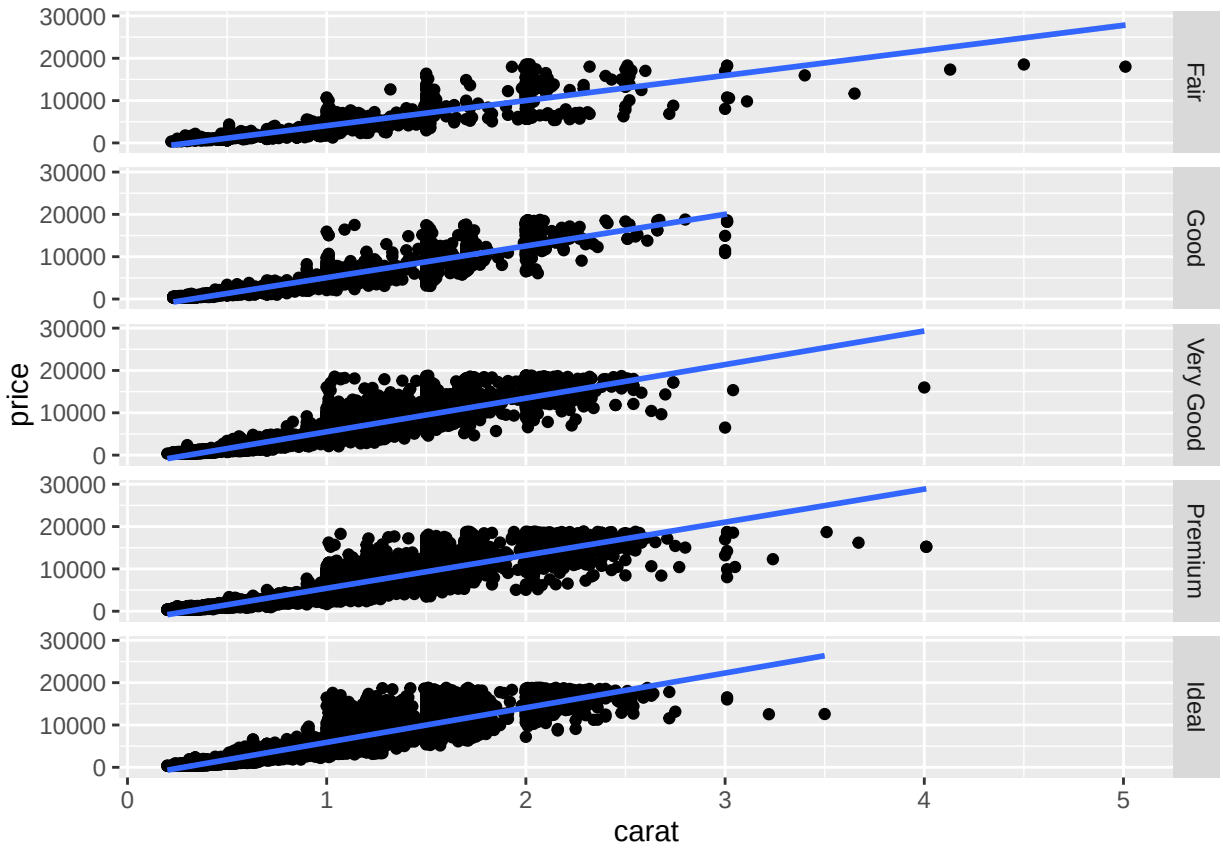
(c) Next, we need to add a linear regression line. (Hint: The "geom_smooth" function fits various lines to data. Run ?geom_smooth to check the help documentation and make sure you fit a linear regression line.)

```
p <- p + geom_smooth(method = "lm")
p
```



(d) Finally, seperate out into panels according to diamond quality (indicated by the "cut" variable). (hint: You need "facet_grid", and you will have to pass a formula object to the formula arguement. Look at the help docs and play around.)

```
p <- p + facet_grid(cut ~ .)
p
```


price
carat
Fair
Good
Very Good
Premium
Ideal
30000
20000
10000
0
0
1
2
3
4
5