Applied Statistics 2020 - Exercises week 10

1. Maximum likelihood estimator for geometric random variables (Theory)

The geometric random variable, as presented in the textbook, has the following probability mass function

$$Pr[X = k] = (1 - p)^{k-1} \cdot p$$

which can be described as the probability of requiring k trials to obtain the first success in a sequence of Bernoulli trials. For this random variable, we have seen that the maximum likelihood estimator for the parameter p is

$$\hat{p} = \frac{n}{\sum_{i=0}^{n} x_i} = \frac{1}{\bar{x}}$$

However, in some contexts [1] a slightly different definition of geometric random variable is used:

$$Pr[X = k] = (1 - p)^{k} \cdot p$$

This second formulation can be described as the probability of experiencing k consecutive failures before the first success.

We shall see, with this exercise, that this small change leads to a different maximum likelihood estimator for p!

a. Derive the loglikelihood function $\ell(p)$

Solution: Likelihood function from the textbook (Dekking, p. 317): $L(\theta) = P(X_1 = x_1, \ldots, X_n = x_n) = p_\theta(x_1) \cdots p_\theta(x_n)$. Pluging $Pr[X = k] = (1 - p)^k \cdot p$ in gives:

$$L(p) = \prod_{i=0}^{n} (1 - p)^{k_i} \cdot p = (1 - p)^{\sum_{i=0}^{n} k_i} \cdot p^n$$

We know from the book (Dekking, p. 319) that $\ell(\theta) = \ln(L(\theta))$, so:

$$\ell(p) = \ln(L(p)) = n \ln(p) + \ln(1 - p) \sum_{i=0}^{n} k_i$$

b. Compute the derivative $\ell'(p)$ of the loglikelihood function

Solution: Here we need to remember some calculus. Derivative of $f(x) = ln(x)$ is $\frac{d}{dx} f(x) = \frac{1}{x}$, so $\ell(p)$ derivative is:

$$\frac{d}{dp} \ell(p) = \frac{n}{p} - \frac{1}{1 - p} \sum_{i=0}^{n} k_i$$

[1] Including the R implementation of the geometric random distribution

c. Show that the maximum likelihood estimator for p is

$$\hat{p} = \frac{n}{n + \sum_{i=0}^{n} x_i}$$

Solution: To find the maximum likelihood estimator for p we need to find the point where $\ell'(p)$ is equal to 0. Since we cannot divide by zero the following is valid for $0 < p < 1$.

Setting $\frac{d}{dp}\ell(p) = 0$ gives us

$$\frac{n}{\hat{p}} = \frac{1}{1-\hat{p}} \sum_{i=0}^{n} k_i$$

Multiplying through with $\hat{p}(1-\hat{p})$ then gives us

$$n(1-\hat{p}) = \hat{p} \sum_{i=0}^{n} k_i$$

rearranging terms and factoring out $\hat{p}$ then gives the result

$$n = \hat{p}(n + \sum_{i=0}^{n} k_i)$$

Which gives the expected result

$$\hat{p} = \frac{n}{n + \sum_{i=0}^{n} k_i}$$

Therefore, *pay attention* to the distribution you are dealing with, always read carefully the definitions and the documentation!

2. Maximum likelihood estimators for the Pareto distribution (Theory)

The Pareto distribution is used in a wide variety of contexts, ranging from describing the size of meteorites to the error rates of disk drives. The expression of the probability density function of the Pareto distribution is

$$f(x) = \frac{\alpha}{x^{\alpha+1}} \quad \text{for } x \geq 1$$

Given the numerous applications, it is important to be able to estimate the value of α from random samples. In this exercise you will derive a maximum likelihood estimator for α.

a. Derive the loglikelihood function $\ell(\alpha)$ *Solution:* Again, we use likelihood function from the text book (Dekking, p. 317), this time for a continuous distribution: $L(\theta) = f_\theta(x_1) \cdots f_\theta(x_n)$.

$$L(\alpha) = f_\alpha(x_1) \cdots f_\alpha(x_n) = \frac{\alpha}{x_1^{\alpha+1}} \cdots \frac{\alpha}{x_n^{\alpha+1}} = \alpha^n \prod_{i=1}^{n} \frac{1}{x_i^{\alpha+1}} = \alpha^n \prod_{i=1}^{n} x_i^{-\alpha-1}$$

$$\ell(\alpha) = n\ln(\alpha) - \sum_{i=1}^{n}(\alpha+1)\ln(x_i) = n\ln(\alpha) - \sum_{i=1}^{n}\ln(x_i) - \sum_{i=1}^{n}\alpha\ln(x_i) = n\ln(\alpha) - \sum_{i=1}^{n}\ln(x_i) - \alpha\sum_{i=1}^{n}\ln(x_i)$$

b. Compute the derivative $\ell'(\alpha)$ of the loglikelihood function

Solution:

$$\frac{d}{d\alpha}\ell(\alpha) = \frac{n}{\alpha} - \sum_{i=1}^{n} \ln(x_i)$$

c. Derive the maximum likelihood estimator for α

Solution:

$$\frac{n}{\hat{\alpha}} = \sum_{i=1}^{n} \ln(x_i)$$

Dividing by n we get

$$\frac{1}{\hat{\alpha}} = \frac{\sum_{i=1}^{n} \ln(x_i)}{n}$$

Now taking the reciprocal on both sides we end up with the maximum likelihood estimator

$$\hat{\alpha} = \frac{n}{\sum_{i=1}^{n} \ln(x_i)}$$

3. Linear models (Theory)

In some situations we may know that the linear model should have some peculiarities, like having no slope, or having intercept equals to zero[2]. Answer to the two following separate questions (i.e. the answer to one doesn't depend on the answer to the other). Let U_i be random variables with expectation zero and variance σ^2.

a. Consider the case $\alpha = 0$. The model then becomes $Y_i = \beta x_i + U_i$, for $i = 1, 2, \ldots, n$. Find the least squares estimate $\hat{\beta}$ for β.

Solution: We take the expression

$$\alpha \sum_{i=1}^{n} x_i + \beta \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i$$

which is already derived with respect to β (Dekking, p. 330), and rearrange the terms, for $\alpha = 0$:

$$\hat{\beta} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}$$

b. Consider the case $\beta = 0$. The model is then $Y_i = \alpha + U_i$, for $i = 1, 2, \ldots, n$. Find the least squares estimate $\hat{\alpha}$ for α.

Solution: We take the expression

$$n\alpha + \beta \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$$

which is already derived with respect to α (Dekking, p. 330), and rearrange the terms, for $\beta = 0$:

$$\hat{\alpha} = \frac{\sum_{i=1}^{n} y_i}{n}$$

Alternative more direct solution:

In the method of least squares we want to minimize the sum of squares.

$$S(\alpha, \beta) = \sum_{i} (y_i - \alpha - \beta x_i)^2$$

[2]For instance we may know that when one quantity of the bivariate dataset is 0 then the other *must* be zero.

if $\alpha = 0$

$$\frac{d}{d\beta}\sum_{i=1}^{n}(y_i - \beta x_i)^2 = -2\sum_{i=1}^{n}(y_i - \beta x_i)x_i = 0$$

which means that

$$\sum_{i=1}^{n} y_i x_i = \hat{\beta}\sum_{i=1}^{n} x_i^2$$

So

$$\hat{\beta} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}$$

if $\beta = 0$

$$\frac{d}{d\alpha}\sum_{i=1}^{n}(y_i - \alpha)^2 = -2\sum_{i=1}^{n}(y_i - \alpha) = 0$$

So we get

$$\sum_{i=1}^{n} y_i = \sum_{i=1}^{n}\hat{\alpha} = n\hat{\alpha}$$

So

$$\hat{\alpha} = \frac{\sum_{i=1}^{n} y_i}{n}$$

4. Maximum likelihood estimator for geometric random variables (R)

In exercise 1, you did show that the geometric distribution defined as

$$Pr[X = k] = (1-p)^k \cdot p$$

has the following maximum likelihood estimator for p:

$$\hat{p}^* = \frac{n}{n + \sum_{i=0}^{n} x_i}$$

This definition of geometric random variable is the one use by R, as state at the beginning of the "Details" section of `help(rgeom)`. In this exercise you will verify that, in this case, using the inverse of the sample mean as the estimator for p leads to heavily biased estimations.

Let $n = 200$. First of all, define a function `estimate_p` that, given the realization of a random sample of n elements it returns the estimate of p using the estimator $\hat{p}^* = \frac{n}{n+\sum_{i=0}^{n} x_i}$.

Then, define $p = 0.3$, and take a random sample of n elements using the `rgeom` function. From this random sample, estimate p using first the estimator $\hat{p} = \frac{1}{\bar{x}}$ and then using the estimator $\hat{p}^* = \frac{n}{n+\sum_{i=0}^{n} x_i}$. Compute the two values $\hat{p} - p$ and $\hat{p}^* - p$. What do the resulting numbers suggest?

Repeat the above sampling and estimation procedure 1000 times, accumulating the values $\hat{p} - p$ and $\hat{p}^* - p$ in two separate lists. Plot the two distributions, possibly overlaying them on the same plot. What can you conclude by observing the plot?

Solution:

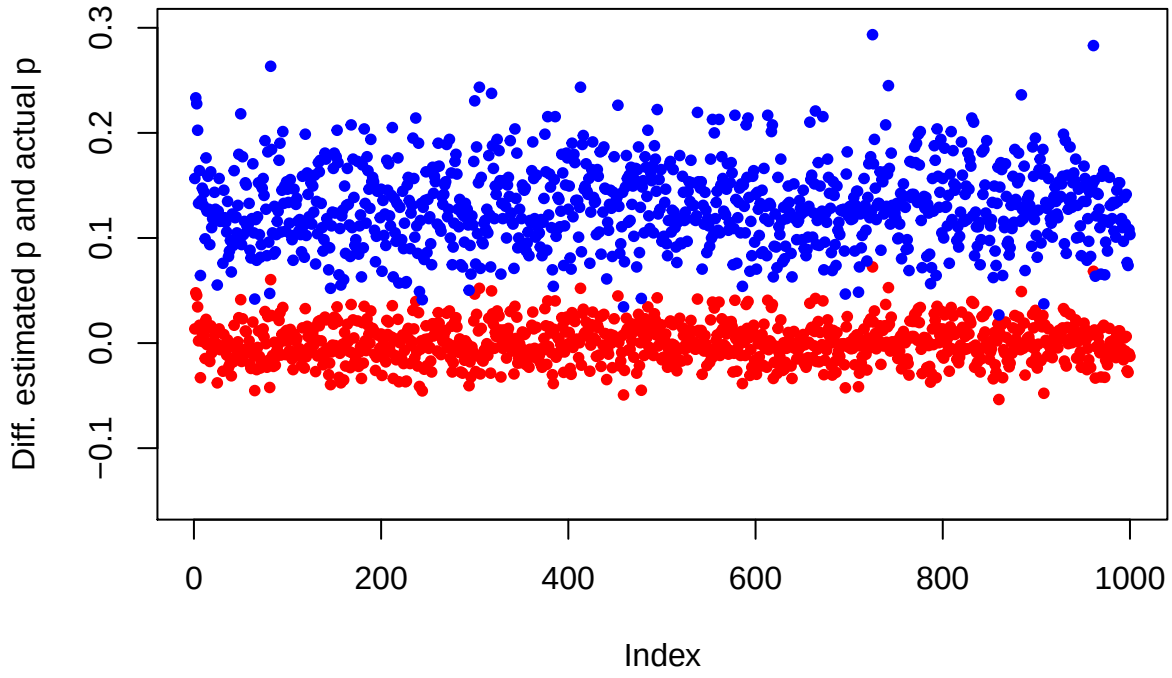
```r
estimate_p <- function(sample){
  p <- 200/(200+sum(sample))
  return(p-0.3)
}

estimate_phat <- function(sample){
  phat <- 1/mean(sample)
  return(phat-0.3)
}

p = 0
phat = 0

for(i in (1:1000)){
  sample <- rgeom(200,0.3)
  p[i] <- estimate_p(sample)
  phat[i] <- estimate_phat(sample)
}

plot(p, col = 'red', pch = 20, ylim=c(-0.15,0.3), ylab = 'Diff. estimated p and actual p')
points(phat, col = 'blue', pch = 20)
```



We conclude that the estimator $\hat{p} = \frac{1}{\bar{x}}$ is positively biased when using R's definition of the geometric distribution.

5. Linear regression model and residuals (R)

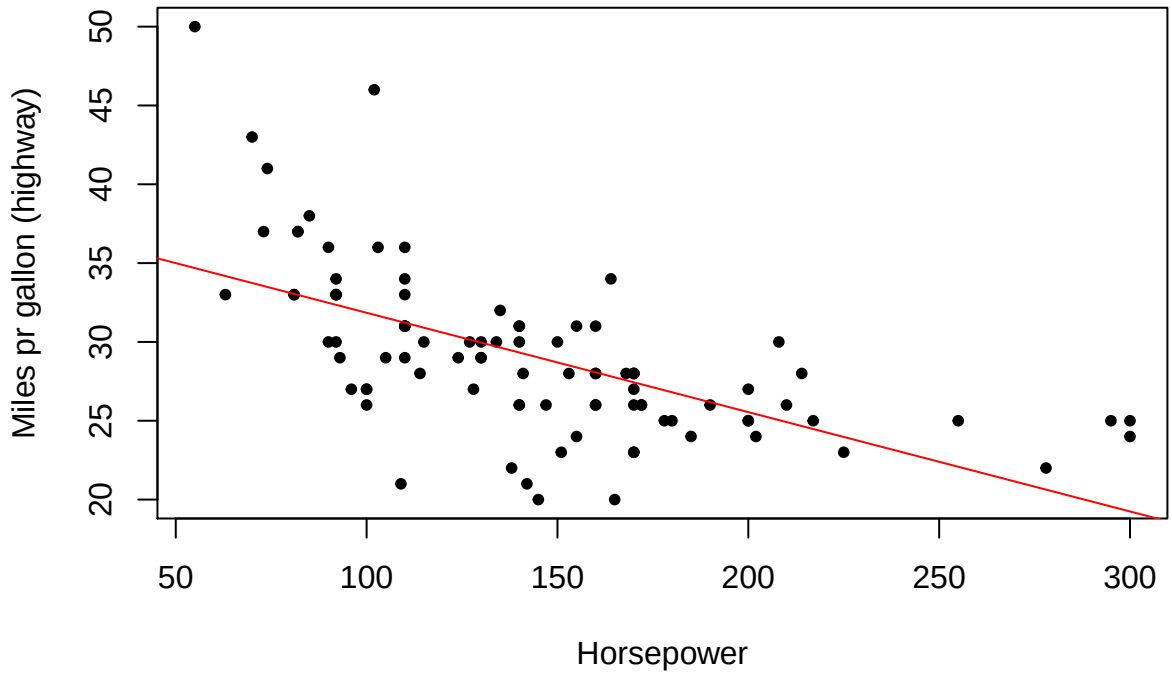
Let us take a look at the `Cars93 (MASS)` data set.

(a) Plot the mileage `MPG.highway` in the function of `Horsepower`. Compute the least-squares estimate for the regression line and add it to the plot.

```
require(MASS)
```

```
## Loading required package: MASS
```

```
plot(MPG.highway ~ Horsepower, data=Cars93, pch=20, ylab = 'Miles pr gallon (highway)', xlab='Horsepowe
fittedlm <- lm(MPG.highway ~ Horsepower, data=Cars93)
abline(fittedlm, col='red')
```



(b) What the predicted mileage for a car with 225 horsepower?

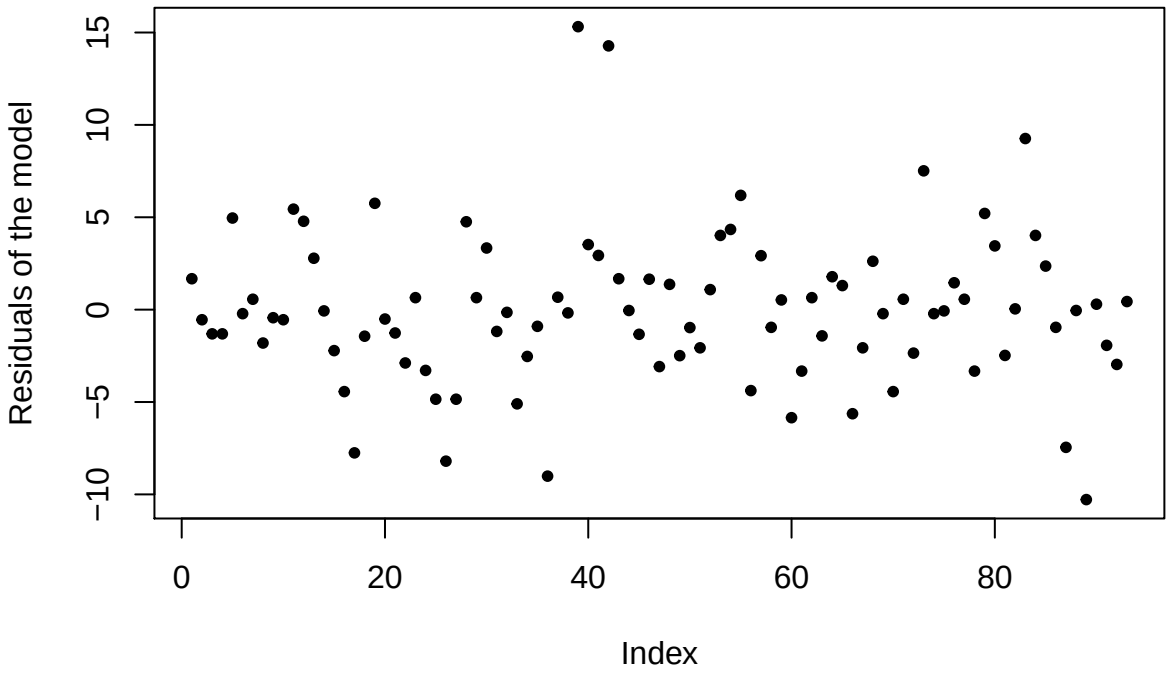
```
predict(fittedlm, data.frame(Horsepower=c(225) ))
```

```
##        1
## 23.97066
```

Our model predicts that a car with horsepowe 255 would have a miles pr gallon of about 24.

(c) Compute and plot the residuals in the function of horsepower. On the basis of the residuals, is the linear model assumption reasonable?

```
diffs <- resid(fittedlm)
plot(diffs, pch = 20, ylab = 'Residuals of the model')
```



The residuals seem to be centered around 0 without any trend or pattern. Thus the assumption of a linear model is reasonable.

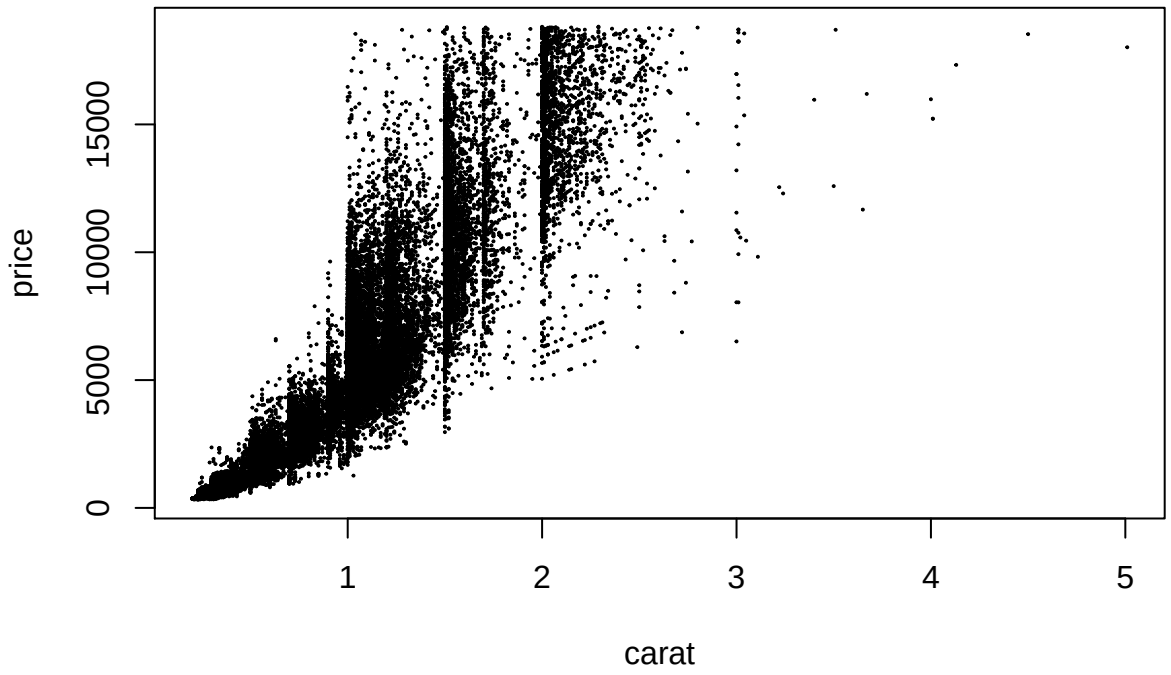
6. Diamond prices (R, Home Exercise)

The `diamonds` dataset[3] (available by loading the `UsingR` package) contains prices of more than 50000 diamonds, along with other information like the carats, the clarity of the diamond, the cut, etc...

1. First of all, plot the scatterplot of diamond prices versus their carats. Can you spot a pattern?

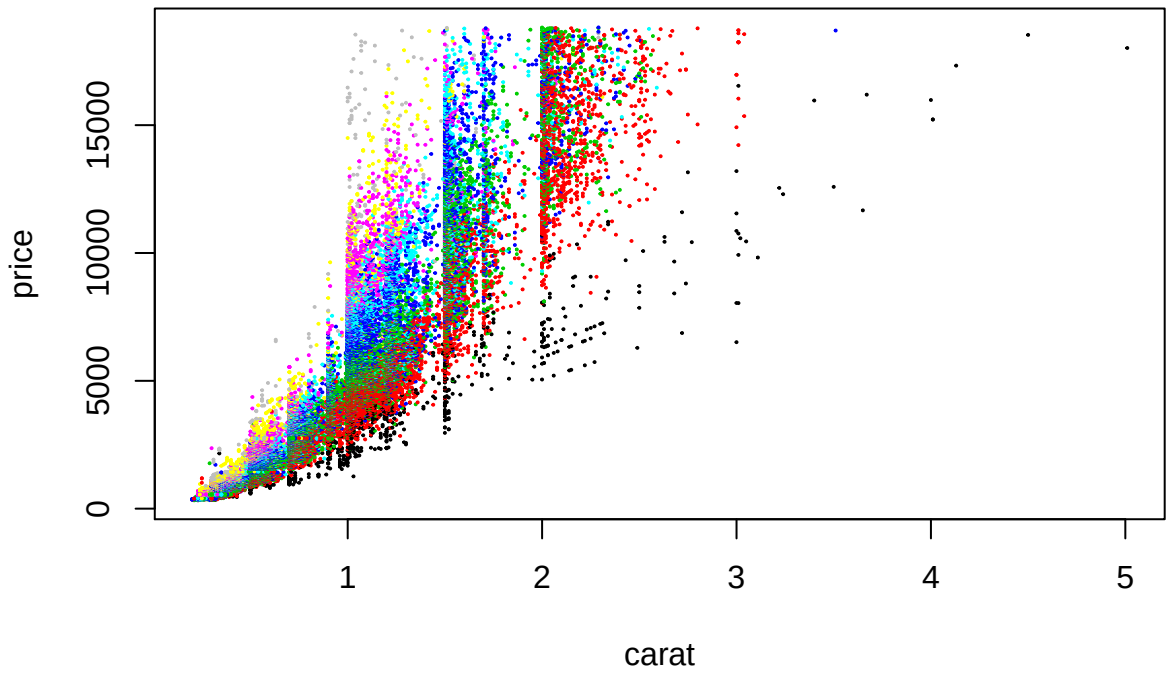
```
plot(price ~ carat, data=diamonds, pch=20, cex = .2)
```

[3]Note that this time the name is plural rather than singular. In a past exercise we used the `diamond` dataset, which containts far less information



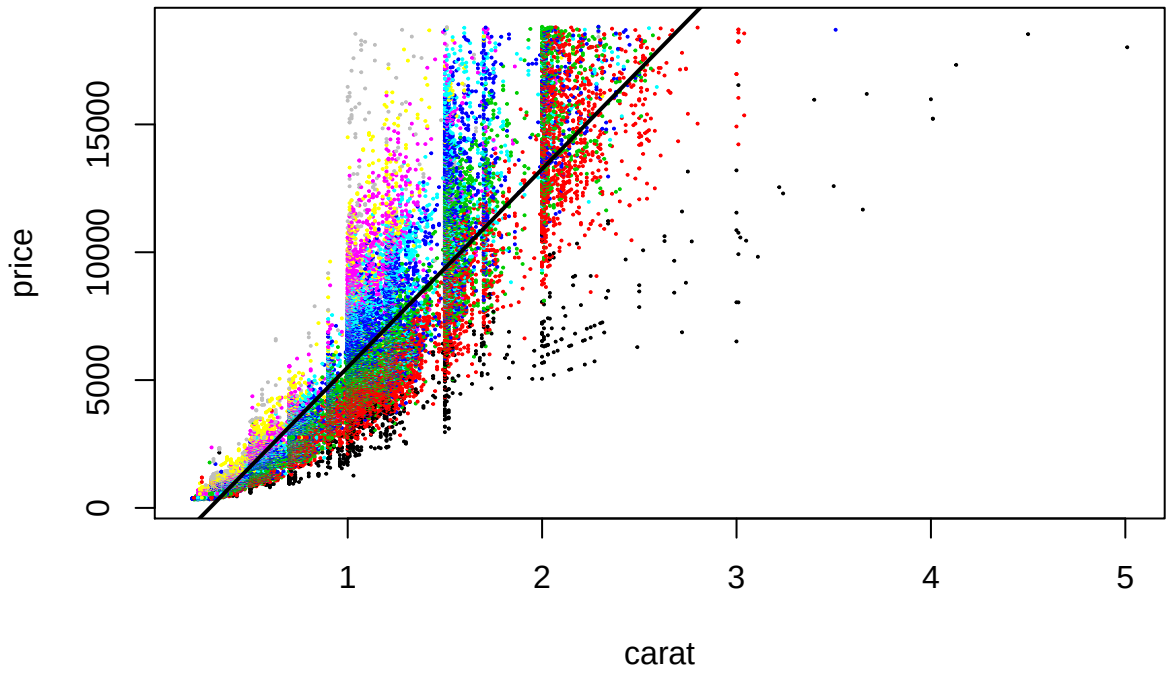
2. Try to add some color, to see if some patterns unveil! In particular, plot the price against the carats, with a different color for each level of the `clarity` column. Does the resulting plot reveal some structure of the dataset?

```
plot(price ~ carat, data=diamonds, pch=20, cex = .2, col = diamonds$clarity)
```



3. Train a linear regression model on the entire dataset, and plot the resulting regression line on top of the previous plot. In your opinion, does it represent summarise accurately the data?

```
fittedlm <- lm(price ~ carat, data=diamonds)
plot(price ~ carat, data=diamonds, pch=20, cex = .2, col = diamonds$clarity)
abline(fittedlm, lwd=2)
```



4. Try to plot a regression line for some subsets of the dataset defined by clarity levels (e.g. all diamonds with clarity "I1") and plot them. Do they provide a better fit for the subsets of data?

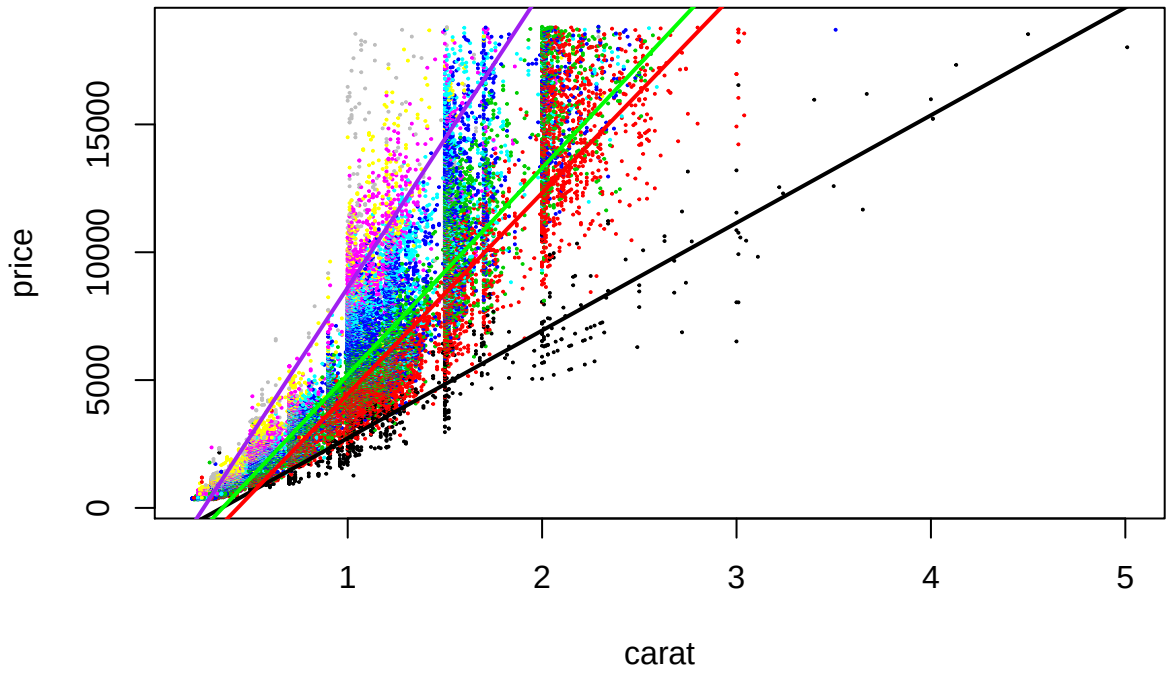
```
plot(price ~ carat, data=diamonds, pch=20, cex = .2, col = diamonds$clarity)

sub_I1 <- subset(diamonds, clarity == 'I1')
lm_I1 <- lm(price ~ carat, data=sub_I1)
abline(lm_I1, lwd=2)

sub_SI2 <- subset(diamonds, clarity == 'SI2')
lm_SI2 <- lm(price ~ carat, data=sub_SI2)
abline(lm_SI2, lwd=2, col = 'red')

sub_SI1 <- subset(diamonds, clarity == 'SI1')
lm_SI1 <- lm(price ~ carat, data=sub_SI1)
abline(lm_SI1, lwd=2, col = 'green')

sub_IF <- subset(diamonds, clarity == 'IF')
lm_IF <- lm(price ~ carat, data=sub_IF)
abline(lm_IF, lwd=2, col = 'purple')
```



Much better.

5. Using the linear models built in the previous step, predict the price of two diamonds of `1.5` carats: the first with clarity `"I1"`, the other with clarity `"IF"`. Also predict the price of a `1.5` carats diamond using the model trained on the entire dataset. Comparing your results with the previous plots, are the models trained taking into account the clarity more powerful in their prediction?

```
cat(sprintf("Price prediction of a diamond of 1.5 carats with clarity I1: %.2f",
            predict(lm_I1, data.frame(carat=c(1.5)))))
```

```
## Price prediction of a diamond of 1.5 carats with clarity I1: 4834.13
```

```
cat(sprintf("\nPrice prediction of a diamond of 1.5 carats with clarity IF: %.2f",
            predict(lm_IF, data.frame(carat=c(1.5)))))
```

```
##
## Price prediction of a diamond of 1.5 carats with clarity IF: 14430.72
```

```
cat(sprintf("\nPrice prediction of a diamond of 1.5 by the model trained on the
            entire dataset: %.2f", predict(fittedlm, data.frame(carat=c(1.5)))))
```

```
##
## Price prediction of a diamond of 1.5 by the model trained on the
##             entire dataset: 9378.28
```