Lab 9

Stat 131A

Welcome to the Lab 9! In this lab, we will review PCA and implement multivariate linear regressions in `R`.

A simple PCA example

This is a simple example for PCA from R Bloggers. The iris dataset is perhaps the best known dataset for classification. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.

```
# Load data
data(iris)
head(iris, 3)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
```

There are five variables in the dataset `iris`, where `Sepal.Length`, `Sepal.Width`, `Petal.Length` and `Petal.Width` are continuous and `Species` is categorical.

```
names(iris)
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

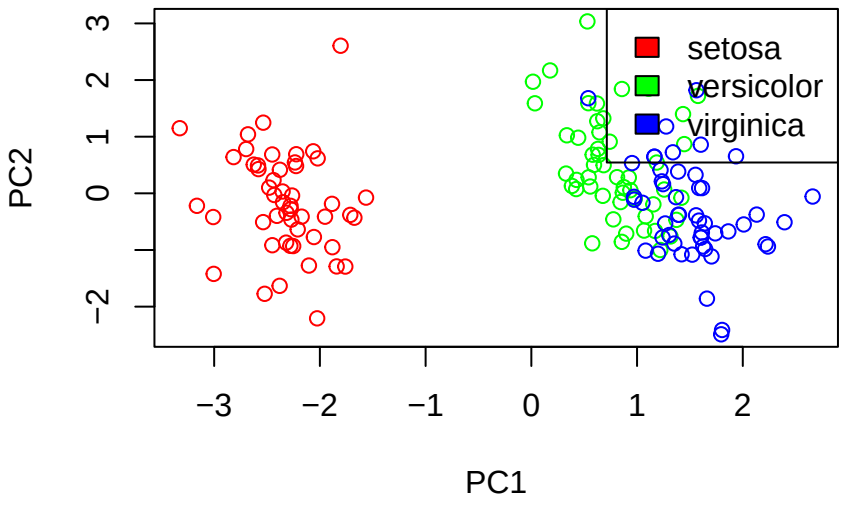
Excercise 1: PCA on the iris dataset

(a) Apply PCA to the four continuous variables. Be sure to transform these variables by taking the log and then converting those log values to standard units. **Note.** The centering and scaling may be done by the user or by changing the arguments of `prcomp()`. See the manual page `?prcomp()` for details.

```
# log transform on the four continuous variable
log.ir <- log(iris[, 1:4])
# the Species variable
ir.species <- iris[, 5]

# Apply PCA
# center and scale is used to standardize the variables prior to the application of PCA.
ir.pca <- prcomp(log.ir, center = TRUE, scale. = TRUE)
```

(b) Create a scatter plot of the transformed data projected onto PC1 and PC2 with PC2 on the vertical axis and PC1 on the horizontal axis. Color the points on the scatter plot by species. **Hint.** If plotting with `plot()`, change the `col` argument.

```
colorvec <- c("red", "green", "blue")
names(colorvec) = unique(ir.species)
plot(ir.pca$x[, 1], ir.pca$x[, 2], col = unname(colorvec[ir.species]), xlab =
"PC1", ylab = "PC2")
legend("topright", legend = names(colorvec), fill = unname(colorvec))
```



Now, lets add a few missing values to the data.

```
iris2 <- iris
iris2[2,3] <- NA
iris2[5,2] <- NA
iris2[10,1] <- NA
```

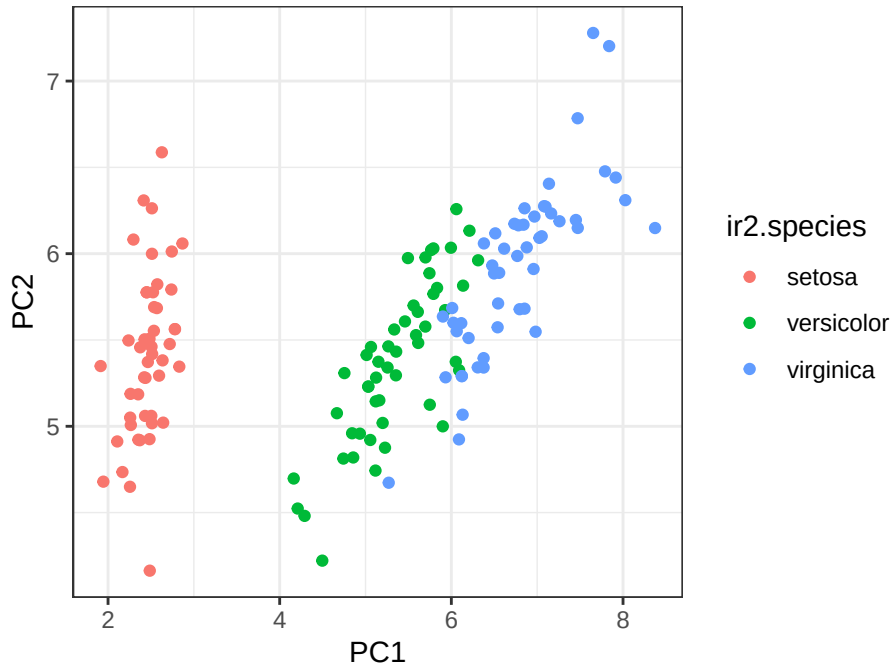
(c) Apply PCA as before, but make change to code to omit rows with missing values.

```
ir.omit <-  na.omit(log(iris2[, 1:4]))
ir2.species <- iris2[, 5]
ir2.pca <- prcomp(ir.omit, center = TRUE, scale. = TRUE)
```

(d) Project the full data (no missing values) onto the principal components space computed from the data with missingness. Create a scatter plot of the full data projected onto the space formed by PC1 and PC2 (as calculated from the data with missingness). Can you identify the species corresponding to the rows with missing data?

```
ir.rot <- as.matrix(iris2[,1:4]) %*%
  ir2.pca$rotation
ir.species_with_missing <- ir2.species
ir.species_with_missing[!complete.cases(iris2)] <- NA

ir.rot %>% as.data.frame %>% ggplot(
  aes(x = PC1, y=PC2, color = ir2.species)
) + geom_point() + theme_bw()
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



(e) **Extra.** Re-create the plot from (b), but now with the `biplot()` function. Interpret the lines in the biplot: what to the lengths and directions tell you?

Multiple regression with diamond price data

This is a very large data set showing various factors of over 50,000 diamonds including price, cut, color, clarity, etc. We are interested in diamond price `price` and how different factors influence it.

Variable	Description
price	price in US dollars ($326–$18,823)
carat	weight of the diamond (0.2–5.01)
cut	quality of the cut (Fair, Good, Very Good, Premium, Ideal)
color	diamond colour, from J (worst) to D (best)
clarity	how clear the diamond is (I1 (worst), SI1, SI2, VS1, VS2, VVS1, VVS2, IF (best))
length.in.mm	length in mm (0–10.74)
width.of.mm	width in mm (0–58.9)
depth.in.mm	depth in mm (0–31.8)
depth	total depth percentage $ = z / mean(x, y) = 2 * z / (x+ y) (43–79)$
table	width of top of diamond relative to widest point (43–95)

```
diamonds <- read.csv("diamonds.csv")
head(diamonds)
```

```
##   carat       cut color clarity depth table price length.in.mm width.of.mm
## 1  0.23     Ideal     E     SI2  61.5    55   326         3.95        3.98
## 2  0.21   Premium     E     SI1  59.8    61   326         3.89        3.84
## 3  0.23      Good     E     VS1  56.9    65   327         4.05        4.07
## 4  0.29   Premium     I     VS2  62.4    58   334         4.20        4.23
## 5  0.31      Good     J     SI2  63.3    58   335         4.34        4.35
## 6  0.24 Very Good     J    VVS2  62.8    57   336         3.94        3.96
```

```
##   depth.in.mm
## 1        2.43
## 2        2.31
## 3        2.31
## 4        2.63
## 5        2.75
## 6        2.48
```

Exercise 2: Exploratory Data Analysis before Regression

First, to better understand the relationships between the variables, we will generate scatter plots. Create scatter plots between the response variable (`price`) and all the continuous variables. The function `is.numeric()` might be helpful to check whether a variable is numeric. For example:

```
vec1 = 1:10
vec2 = as.character(1:10)
vec1
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
vec2
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10"
```

```
# The following line of code would return TRUE
is.numeric(vec1)
```

```
## [1] TRUE
```

```
# The following line of code would return FALSE
is.numeric(vec2)
```

```
## [1] FALSE
```

```
# The following line of code would return TRUE
is.character(vec2)
```

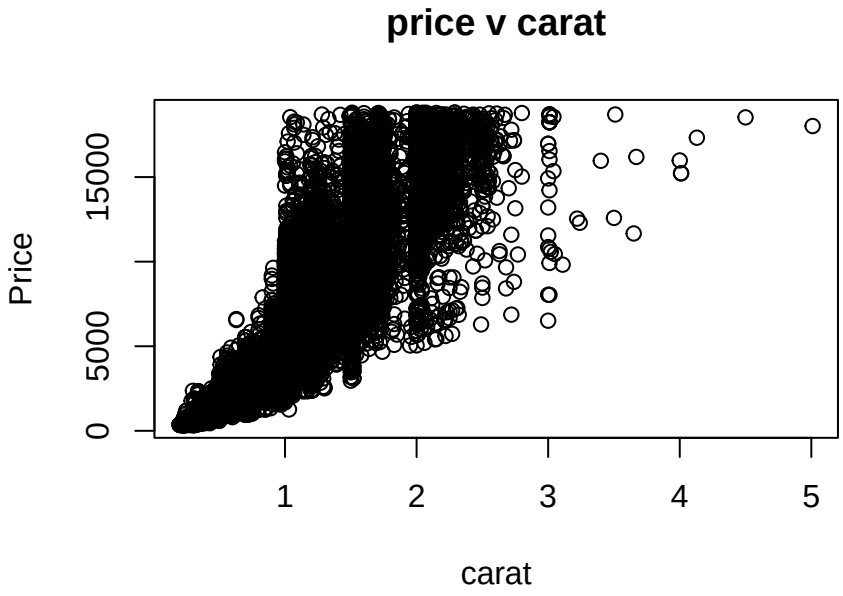
```
## [1] TRUE
```

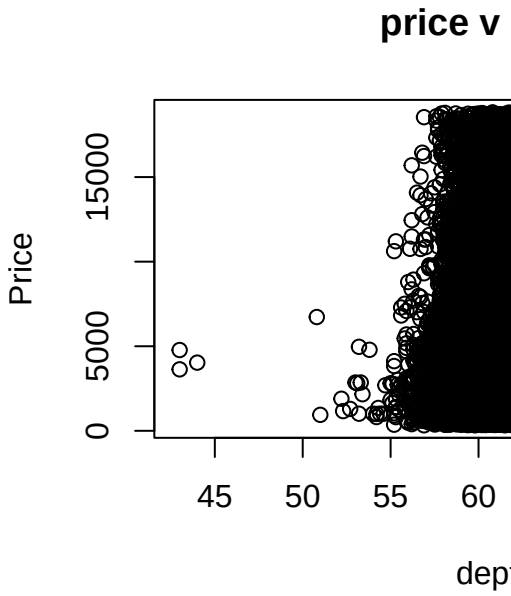
The function `which()` can help you to locate the column indices of the numeric vectors. (In fact, function `which()` is a super useful function in R.)

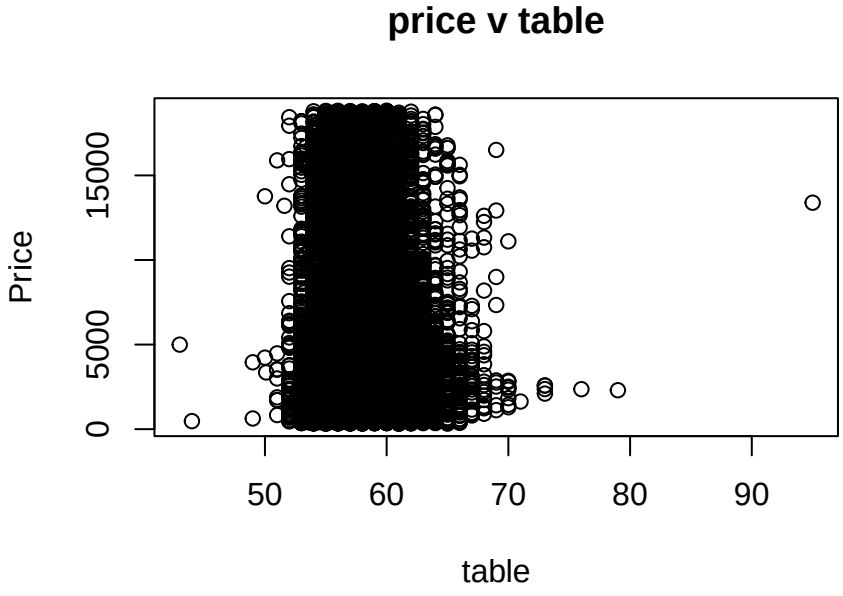
```
# function `which` give the TRUE indices of a logical object
which(c(TRUE, FALSE, TRUE, FALSE, TRUE))
```

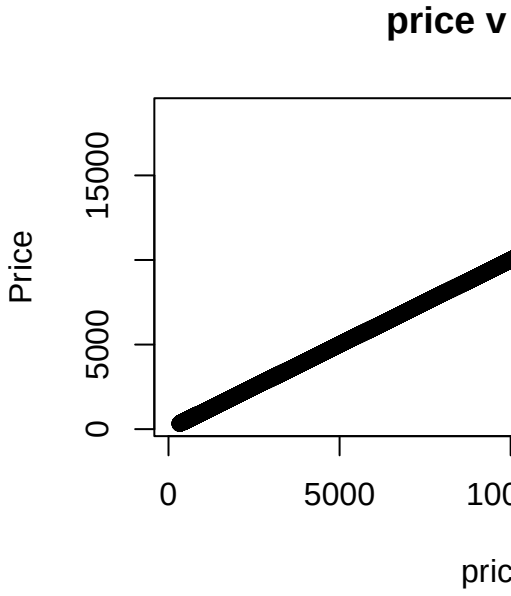
```
## [1] 1 3 5
```

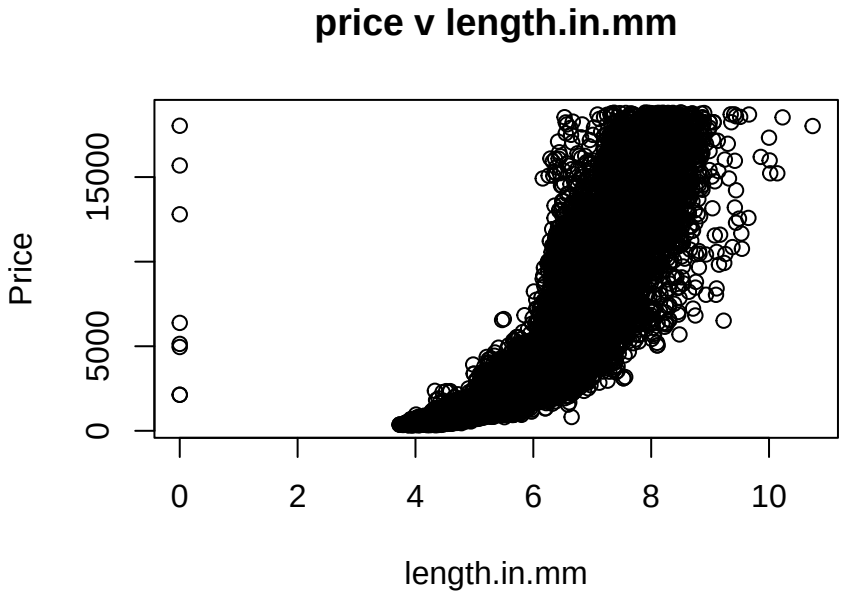
```
# Insert your code here, use for loop to loop through each numerical variable
x=sapply(diamonds, is.numeric)
y=which(x==TRUE)
for (i in y) {
 plot(diamonds[,i], diamonds$price, ylab="Price", main=paste("price v" ,
colnames(diamonds)[i]), xlab=colnames(diamonds)[i])
}
```

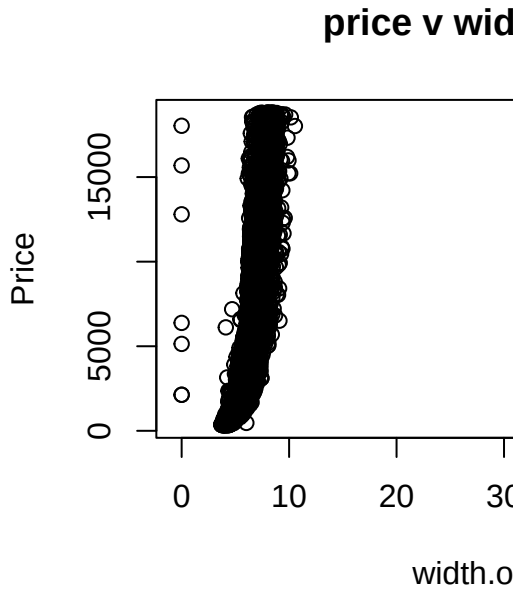

price v carat
Price
0
5000
15000
1
2
3
4
5
carat

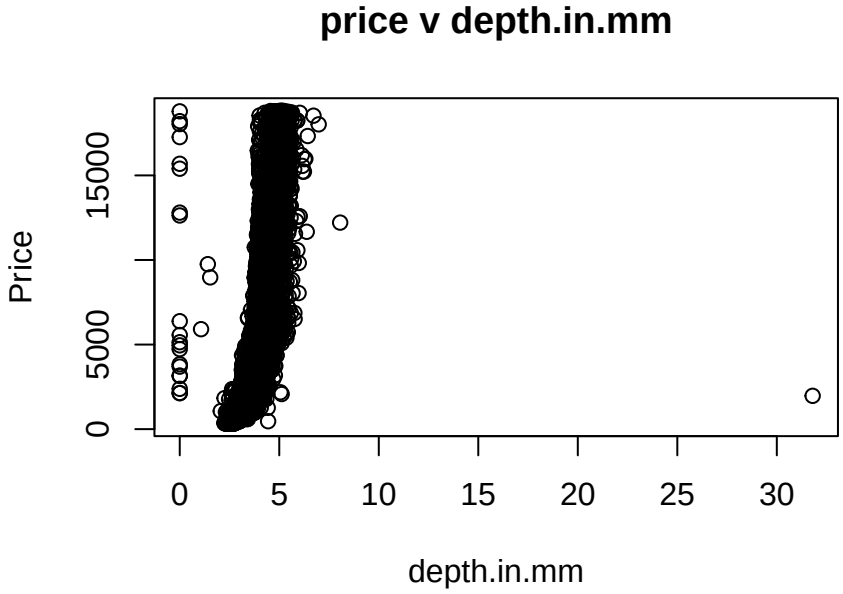

price v
Price
0
5000
15000
45
50
55
60
dep


price v table
Price
0
5000
15000
50
60
70
80
90
table


price v
Price
0
5000
15000
0
5000
pric







Multiple regression with continuous variable

Exercise 3 Fit the model and Calculate the Statistics

(a) Fit a linear model to price with all the continuous variables as explanatory variables. Print the summary of your model.

```
# Insert you code here, save your model as `fit`
fit <- (lm(price ~ carat + depth + table + length.in.mm + width.of.mm+ depth.in.mm,
data = diamonds))
```

```
summary(fit)
```

```
##
## Call:
## lm(formula = price ~ carat + depth + table + length.in.mm + width.of.mm +
##     depth.in.mm, data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23878.2   -615.0    -50.7    347.9  12759.2
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  20849.316    447.562  46.584  < 2e-16 ***
## carat        10686.309     63.201 169.085  < 2e-16 ***
## depth         -203.154      5.504 -36.910  < 2e-16 ***
## table         -102.446      3.084 -33.216  < 2e-16 ***
## length.in.mm -1315.668     43.070 -30.547  < 2e-16 ***
## width.of.mm     66.322     25.523   2.599  0.00937 **
## depth.in.mm     41.628     44.305   0.940  0.34744
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1497 on 53933 degrees of freedom
## Multiple R-squared:  0.8592, Adjusted R-squared:  0.8592
## F-statistic: 5.486e+04 on 6 and 53933 DF,  p-value: < 2.2e-16
```

(b) Calculate the fitted values.

```
# Insert you code here, save your results as `fitted.values`
fitted.value <- fitted.values(fit)
```

(c) Calculate the residuals, the residual sum of squares (RSS), and the total sum of squares (TSS) using the `fitted.value()` from the above chunk.

```
# Insert you code here
residual <- residuals(fit)
RSS <- sum(residual^2)
RSS
```

```
## [1] 1.20857e+11
```

```
TSS <- sum(((diamonds$price) - mean(diamonds$price))^2)
TSS
```

```
## [1] 858473135517
```

(d) Calculate the R-square (R^2) using RSS and TSS. What is the interpretation of R^2?

```
# Insert you code here, save your results as `Rsq`
Rsq <- 1-(RSS/TSS)
Rsq
```

```
## [1] 0.8592187
```

Exercise 4 Think deeper. Is the model reasonable?

(a) Using the fitted model, we can write the estimated model formula. *Hint.* Use `summary()` on the model object.

$$\begin{aligned}\texttt{earnings} = 20849.316 &+ 10686.309 \cdot \texttt{carat} - 203.154 \cdot \texttt{depth} \\ &- 102.446 \cdot \texttt{table} - 1315.668 \cdot \texttt{length.in.mm} \\ &+ 66.322 \cdot \texttt{width.of.mm} + 41.628 \cdot \texttt{depth.in.mm} \\ &+ \hat{\varepsilon}\end{aligned}$$

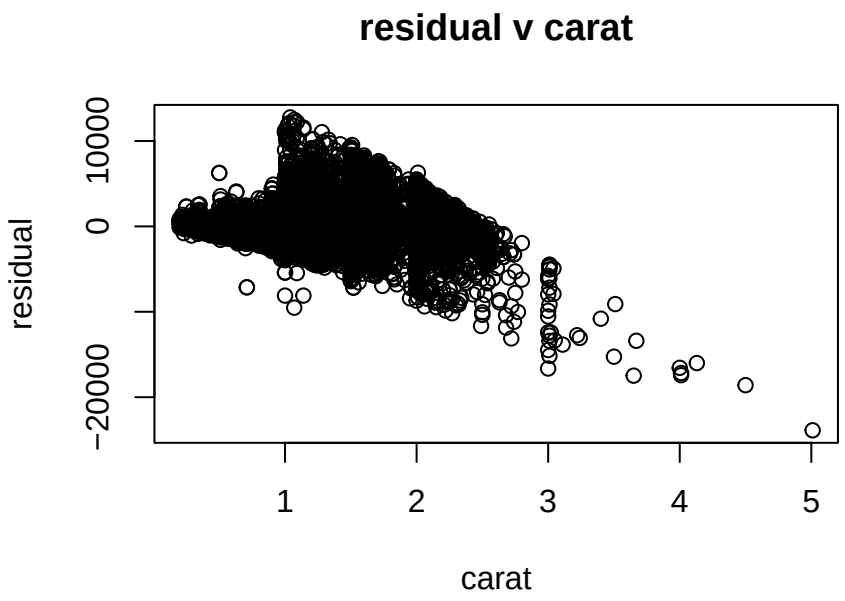
How do we interpret this equation? By looking at the p-values, we know that the coefficients we estimated are significant except for that of `depth.in.mm`. Take `length.in.mm` for example, the coefficient -1315.668 tells us that for a unit increase in `length.in.mm` is associated with a reduction in price of \$1315.67 dollars on average. Isn't that weird? Fit another multivariate model, but this time, drop `length.in.mm`, `width.of.mm` and `depth.in.mm`? Is that a good idea? *Hint.* Check the correlation between `length.in.mm`, `width.of.mm`, `depth.in.mm` and `carat`.

```
# insert your code here
cor(diamonds[, c("length.in.mm", "width.of.mm", "depth.in.mm" ,"carat")])
```

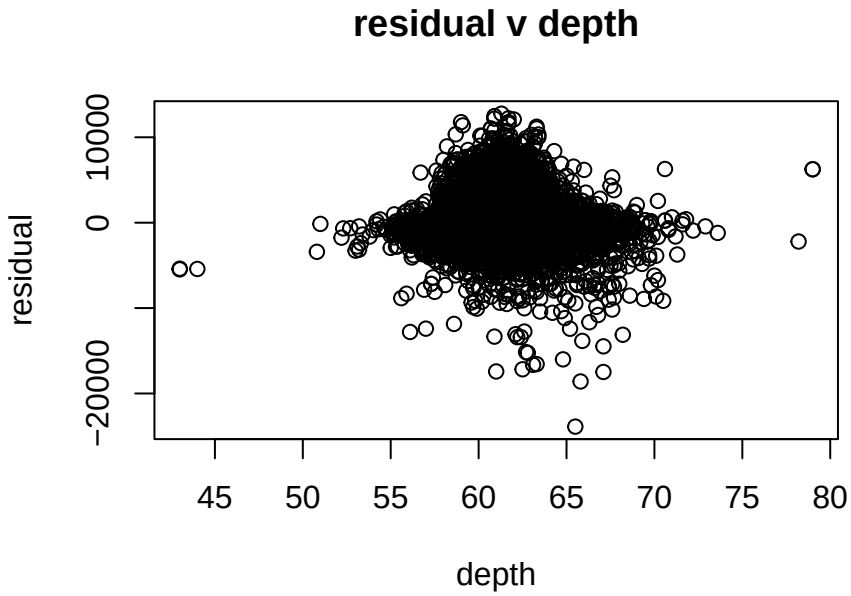
```
##              length.in.mm width.of.mm depth.in.mm     carat
## length.in.mm    1.0000000   0.9747015   0.9707718 0.9750942
## width.of.mm     0.9747015   1.0000000   0.9520057 0.9517222
## depth.in.mm     0.9707718   0.9520057   1.0000000 0.9533874
## carat           0.9750942   0.9517222   0.9533874 1.0000000
```

(b) Plot the residuals from the restricted model from part (a) against the variables and calculate their correlations. Can you find any problem in your model by looking at these scatter plots? If you're asked to add some terms to improve the model, what will you do? (*Hint.* Consider the scatter plot in Exercise 1: are the relationships linear?)

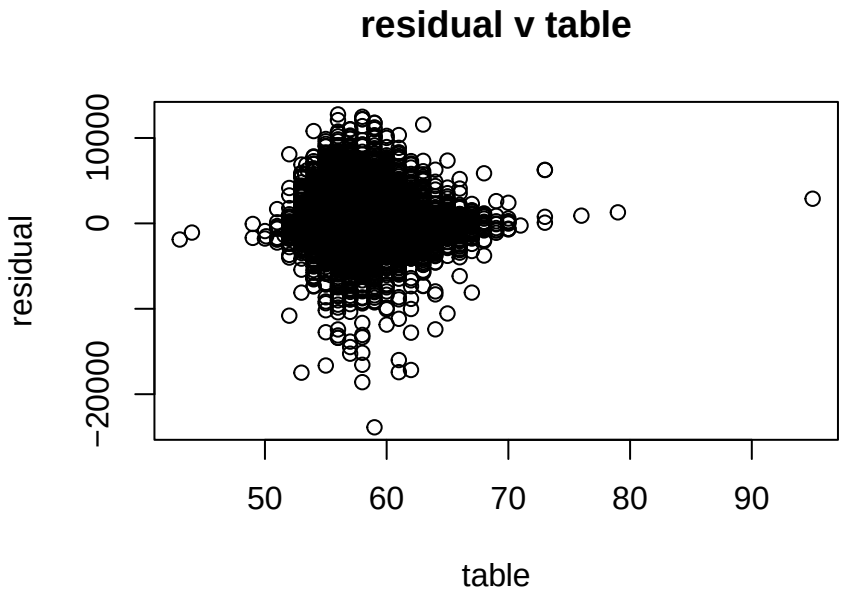
```
# Insert your code here, use for loop to loop through each numeric variable
x=sapply(diamonds, is.numeric)
y=which(x==TRUE)
for (i in y) {
 plot(diamonds[,i], residual,main=paste("residual v" , colnames(diamonds)[i]),
xlab=colnames(diamonds)[i])
 print (cor(diamonds[,i], residual))

}
```



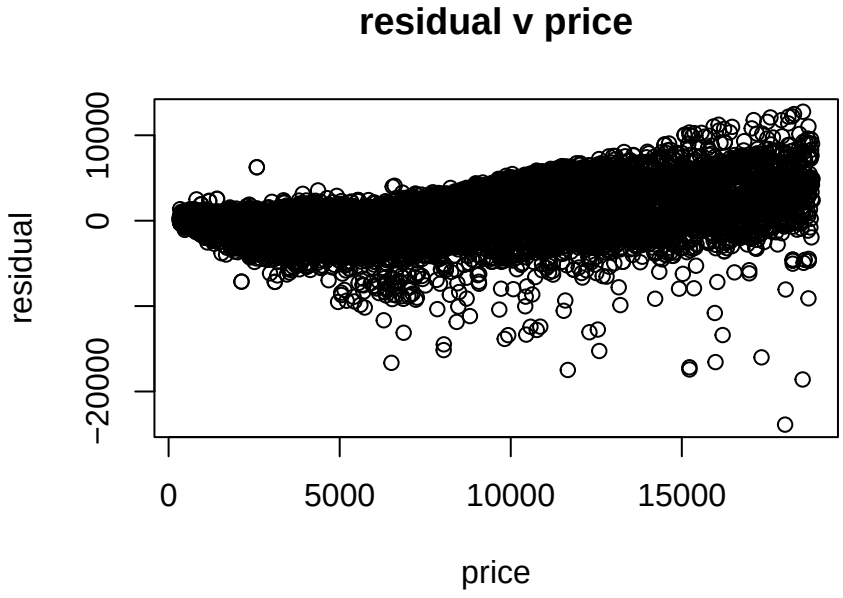

```
## [1] 5.956029e-17
```



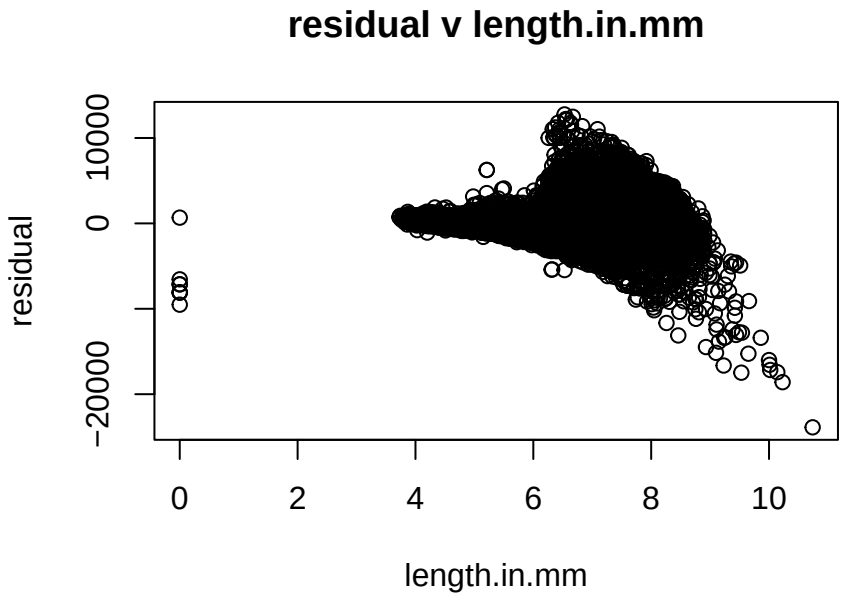

```
## [1] -1.899877e-16
```



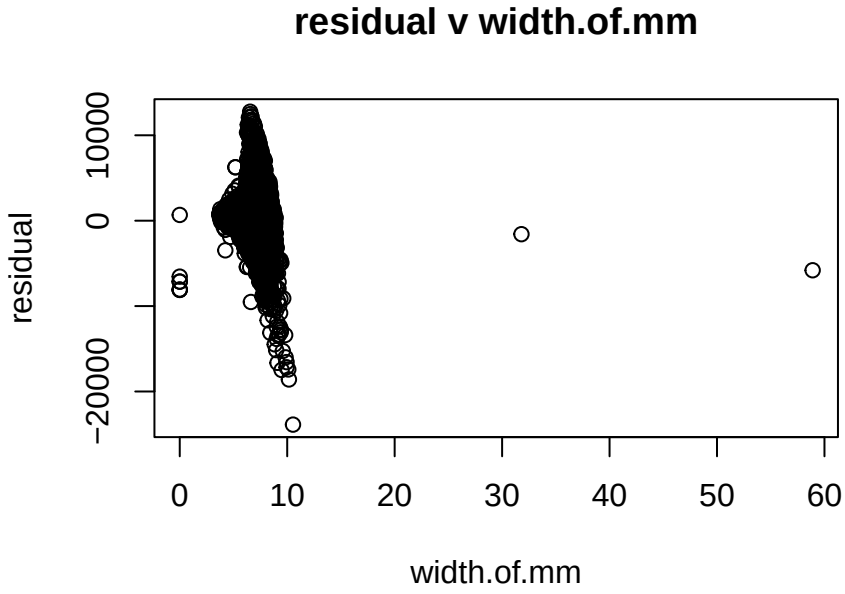

```
## [1] -1.329821e-16
```



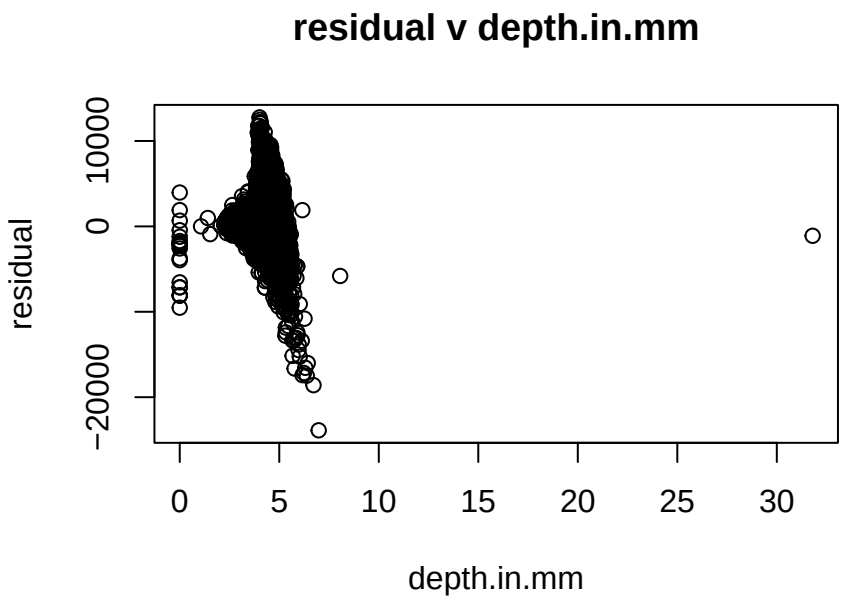

```
## [1] 0.3752084
```




```
## [1] -4.275645e-16
```




```
## [1] -4.114601e-16
```




```
## [1] -3.361076e-16
```

Multiple regression with continuous and categorical variables

Exercise 5

(a) Fit a linear regression model with explanatory variable `carat`, `depth`, `table`, `clarity`, `color` and `cut`.

```
levels(diamonds$clarity)
```

```
## NULL
```

```
levels(diamonds$color)
```

```
## NULL
```

```
levels(diamonds$cut)
```

```
## NULL
```

```
# Insert you code here, save your model as `fit.categorical`
fit.categorical <- (lm(price ~ carat+ depth + table + as.factor(clarity) +
as.factor(color) + as.factor(cut), data = diamonds))
summary(fit.categorical)
```

```
##
## Call:
## lm(formula = price ~ carat + depth + table + as.factor(clarity) +
##     as.factor(color) + as.factor(cut), data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16828.8   -678.7   -199.4    464.6  10341.2
##
## Coefficients:
```

```
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -4555.171    373.482 -12.197  < 2e-16 ***
## carat                      8895.194     12.079 736.390  < 2e-16 ***
## depth                       -21.024      4.079  -5.154 2.56e-07 ***
## table                       -24.803      2.978  -8.329  < 2e-16 ***
## as.factor(clarity)IF       5404.237     52.174 103.582  < 2e-16 ***
## as.factor(clarity)SI1      3567.794     44.587  80.020  < 2e-16 ***
## as.factor(clarity)SI2      2619.004     44.788  58.476  < 2e-16 ***
## as.factor(clarity)VS1      4525.400     45.547  99.356  < 2e-16 ***
## as.factor(clarity)VS2      4210.194     44.840  93.893  < 2e-16 ***
## as.factor(clarity)VVS1     5061.734     48.224 104.964  < 2e-16 ***
## as.factor(clarity)VVS2     4957.310     46.901 105.697  < 2e-16 ***
## as.factor(color)E          -210.849     18.304 -11.519  < 2e-16 ***
## as.factor(color)F          -304.288     18.498 -16.450  < 2e-16 ***
## as.factor(color)G          -506.964     18.116 -27.984  < 2e-16 ***
## as.factor(color)H          -977.974     19.269 -50.754  < 2e-16 ***
## as.factor(color)I         -1438.277     21.642 -66.459  < 2e-16 ***
## as.factor(color)J         -2322.565     26.715 -86.940  < 2e-16 ***
## as.factor(cut)Good          614.424     34.337  17.894  < 2e-16 ***
## as.factor(cut)Ideal         877.569     34.152  25.696  < 2e-16 ***
## as.factor(cut)Premium       806.024     32.954  24.459  < 2e-16 ***
## as.factor(cut)Very Good     778.428     32.936  23.635  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1156 on 53919 degrees of freedom
## Multiple R-squared:  0.9161, Adjusted R-squared:  0.916
## F-statistic: 2.942e+04 on 20 and 53919 DF,  p-value: < 2.2e-16
```

(b) Write the equation when

i. Clarity is VS2, color is H, and cut is Premium. Replace ??? with numerical values.

$$\text{price} = -18171.89 + 8699.36 \cdot \text{carat} + 168.42 \cdot \text{depth} + 79.18 \cdot \text{table}$$

ii. clarity is I1, color is D and cut is Fair. Replace ??? by numerical values.

$$\text{price} = -60252.5 + 5553.0 \cdot \text{carat} + 734.6 \cdot \text{depth} + 158.9 \cdot \text{table}$$

```
v1=subset(diamonds,clarity=="VS2" & color=="H" & cut=="Premium")
v1=(lm(price ~ carat + depth + table, data = v1))
summary(v1)
```

```
##
## Call:
## lm(formula = price ~ carat + depth + table, data = v1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2578.0  -728.6  -298.4   868.2  4049.6
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -18171.89    3752.79  -4.842 1.69e-06 ***
## carat         8699.36      96.55  90.106  < 2e-16 ***
## depth          168.42      43.56   3.866 0.000124 ***
## table           79.18      32.29   2.453 0.014507 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1044 on 528 degrees of freedom
## Multiple R-squared:  0.9394, Adjusted R-squared:  0.9391
## F-statistic:  2729 on 3 and 528 DF,  p-value: < 2.2e-16
```

```
v2=subset(diamonds,clarity=="I1" & color=="D" & cut=="Fair")
v2=(lm(price ~ carat + depth + table, data = v2))
summary(v2)
```

```
##
## Call:
## lm(formula = price ~ carat + depth + table, data = v2)
##
## Residuals:
## ALL 4 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -60252.5        NaN     NaN      NaN
## carat         5553.0        NaN     NaN      NaN
## depth          734.6        NaN     NaN      NaN
## table          158.9        NaN     NaN      NaN
##
## Residual standard error: NaN on 0 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:    NaN
## F-statistic:   NaN on 3 and 0 DF,  p-value: NA
```