# Computer workshop 1 (NBS8186)

*Edu Gonzalo Almorox*

## Introduction

This document contains commented solutions to the questions of Workshop 1 for *NBS8186 Introductory Economics*. Analysis is based on `nba.csv` file.

## Question A

*Download the data set from Blackboard and save it on your h: drive. Then open the data set in R and make it the default data set.*

Prior to load the data you must set the working directory in your computer. The working directory is the place in your computer where you allocate the information that you are going to use in your data analysis. `setwd()` is the function to tell R the working directory in your computer[1]. Sometimes you may be using several working directories. In order to know the current working directory you are working on you may use `getwd()`

```
setwd("/Users/Personas/My Cloud/PhD _october_2016/teaching/NBS8186/data")
getwd() # what working directory?
```

```
## [1] "/Users/Personas/My Cloud/PhD _october_2016/teaching/NBS8186/data"
```

Once the working directory is established, it is time to load the data. The most common way to input a dataset in R consists of using the base[2] function `read.csv()`.

```
nba = read.csv("nba.csv", sep = ",", header = TRUE)
```

The data are in your computer, now may carry out some exploratory analysis of your data. For example you may want to have a look at the first and last rows. This can be done using `head()` and `tail()`.

```
head(nba, 5) # gives the first five lines
```

```
##   marr  wage exper age coll games minutes guard forward center points
## 1    1 1.002     4  27    4    77    2867     1       0      0   15.5
## 2    1 2.030     5  28    4    78    2789     1       0      0   13.3
## 3    0 0.650     1  25    4    74    1149     0       0      1    5.5
## 4    0 2.030     5  28    4    47    1178     0       1      0    7.3
## 5    0 0.755     3  24    4    82    2096     1       0      0   10.8
##   rebounds assists draft allstar avgmin black children
## 1      3.9     4.5    19       0 37.234     1        0
## 2      2.5     8.8    28       0 35.756     1        1
## 3      3.3     0.2    19       0 15.527     1        0
## 4      5.1     1.5     1       0 25.064     1        0
## 5      4.3     2.6    24       0 25.561     1        0
```

[1] Note: The way to introduce the path may differ in case you are using Windows or Mac. For Windows it looks like `"C:/Users/User Name/Documents/FOLDER"` whereas for Mac it is similar to `"/Users/User Name/Documents/FOLDER"`

[2] There are other packages such as `foreign` or `rio` that can also be used for loading data.

```
tail(nba, 5) # gives the last five lines
```

```
##     marr  wage exper age coll games minutes guard forward center points
## 263    1 3.210     7  29    4    79    2638     1       0      0   19.9
## 264    1 0.715     5  31    4    75    1084     0       1      0    4.8
## 265    1 0.600    11  33    3    67    1197     1       0      0   10.4
## 266    0 2.500     6  28    4    78    2113     0       0      1   15.7
## 267    0 2.000    12  33    3    30     282     0       1      0    2.3
##     rebounds assists draft allstar avgmin black children
## 263      2.7     3.1    11       1 33.392     1        0
## 264      2.5     0.8    54       0 14.453     1        1
## 265      1.6     2.0     4       0 17.866     1        1
## 266      6.2     1.8     2       0 27.090     0        0
## 267      2.5     0.5     5       0  9.400     1        0
```

Also, it is possible to see the structure of your data frame using `str()`

```
str(nba)
```

```
## 'data.frame':    267 obs. of  18 variables:
##  $ marr    : int  1 1 0 0 0 0 1 0 1 1 ...
##  $ wage    : num  1.002 2.03 0.65 2.03 0.755 ...
##  $ exper   : int  4 5 1 5 3 9 1 3 1 12 ...
##  $ age     : int  27 28 25 28 24 31 28 27 25 35 ...
##  $ coll    : int  4 4 4 4 4 4 0 3 4 3 ...
##  $ games   : int  77 78 74 47 82 82 80 67 60 74 ...
##  $ minutes : int  2867 2789 1149 1178 2096 1971 2303 1131 542 2700 ...
##  $ guard   : int  1 1 0 0 1 0 0 0 1 0 ...
##  $ forward : int  0 0 0 1 0 1 1 1 0 1 ...
##  $ center  : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ points  : num  15.5 13.3 5.5 7.3 10.8 11.3 15.1 6.6 3.1 26 ...
##  $ rebounds: num  3.9 2.5 3.3 5.1 4.3 4.9 7.2 4.2 0.7 6.5 ...
##  $ assists : num  4.5 8.8 0.2 1.5 2.6 1.5 1.4 0.7 2 2.3 ...
##  $ draft   : int  19 28 19 1 24 4 40 47 0 3 ...
##  $ allstar : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ avgmin  : num  37.2 35.8 15.5 25.1 25.6 ...
##  $ black   : int  1 1 1 1 1 1 0 1 1 1 ...
##  $ children: int  0 1 0 0 0 0 0 0 0 1 ...
```

## Question B

*Have a look at the summary statistics of the data set.*

`summary()` is used to get a summary statistics of the variables in your data frame.

```
summary(nba)
```

```
##       marr             wage           exper             age       
##  Min.   :0.0000   Min.   :0.150   Min.   : 1.000   Min.   :21.00  
##  1st Qu.:0.0000   1st Qu.:0.650   1st Qu.: 2.000   1st Qu.:25.00  
##  Median :0.0000   Median :1.186   Median : 4.000   Median :27.00
```

```
##  Mean   :0.4419   Mean   :1.429   Mean   : 5.116   Mean   :27.39
##  3rd Qu.:1.0000   3rd Qu.:2.022   3rd Qu.: 7.500   3rd Qu.:30.00
##  Max.   :1.0000   Max.   :5.740   Max.   :18.000   Max.   :41.00
##       coll            games           minutes           guard
##  Min.   :0.000   Min.   : 3.00   Min.   :  33.0   Min.   :0.0000
##  1st Qu.:4.000   1st Qu.:57.00   1st Qu.: 981.5   1st Qu.:0.0000
##  Median :4.000   Median :74.00   Median :1684.0   Median :0.0000
##  Mean   :3.715   Mean   :65.63   Mean   :1681.0   Mean   :0.4195
##  3rd Qu.:4.000   3rd Qu.:79.00   3rd Qu.:2443.5   3rd Qu.:1.0000
##  Max.   :4.000   Max.   :82.00   Max.   :3533.0   Max.   :1.0000
##     forward           center           points         rebounds
##  Min.   :0.0000   Min.   :0.0000   Min.   : 1.20   Min.   : 0.500
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 5.35   1st Qu.: 2.350
##  Median :0.0000   Median :0.0000   Median : 9.30   Median : 3.800
##  Mean   :0.4082   Mean   :0.1723   Mean   :10.21   Mean   : 4.403
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:14.25   3rd Qu.: 5.500
##  Max.   :1.0000   Max.   :1.0000   Max.   :29.80   Max.   :17.300
##     assists          draft          allstar          avgmin
##  Min.   : 0.00   Min.   :  0.0   Min.   :0.0000   Min.   : 2.889
##  1st Qu.: 0.90   1st Qu.:  4.5   1st Qu.:0.0000   1st Qu.:16.692
##  Median : 1.90   Median : 12.0   Median :0.0000   Median :24.925
##  Mean   : 2.41   Mean   : 18.0   Mean   :0.1161   Mean   :23.984
##  3rd Qu.: 3.40   3rd Qu.: 26.5   3rd Qu.:0.0000   3rd Qu.:33.294
##  Max.   :12.60   Max.   :139.0   Max.   :1.0000   Max.   :43.085
##      black           children
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.0000   1st Qu.:0.0000
##  Median :1.0000   Median :0.0000
##  Mean   :0.8052   Mean   :0.3483
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
```

*What is the average age of the players?*

According to the results displayed by `summary()` we can see that the average age is 27.4 years. An alternative way to obtain the average age would be by calling directly the variable `age` using the operator `$`.

```
mean(nba$age)
```

```
## [1] 27.38951
```

*How many play forwards?*

`forward` is a categorical variable. In these variables, numbers indicate qualitative characteristics that cannot be measured. This type of variables are called `factors()` in R. A simple way to summarise the number of categories in a factor is by using `table()`[3]

```
table(nba$forward)
```

```
##
##   0   1
## 158 109
```

We can see that 109 players play forwards against 158 that play in other positions i.e. center and guard.

[3]There are alternative and more efficient ways to carry out this task. `data.table` and `dplyr` are the most suited packages when there are bigger samples.

## Question C

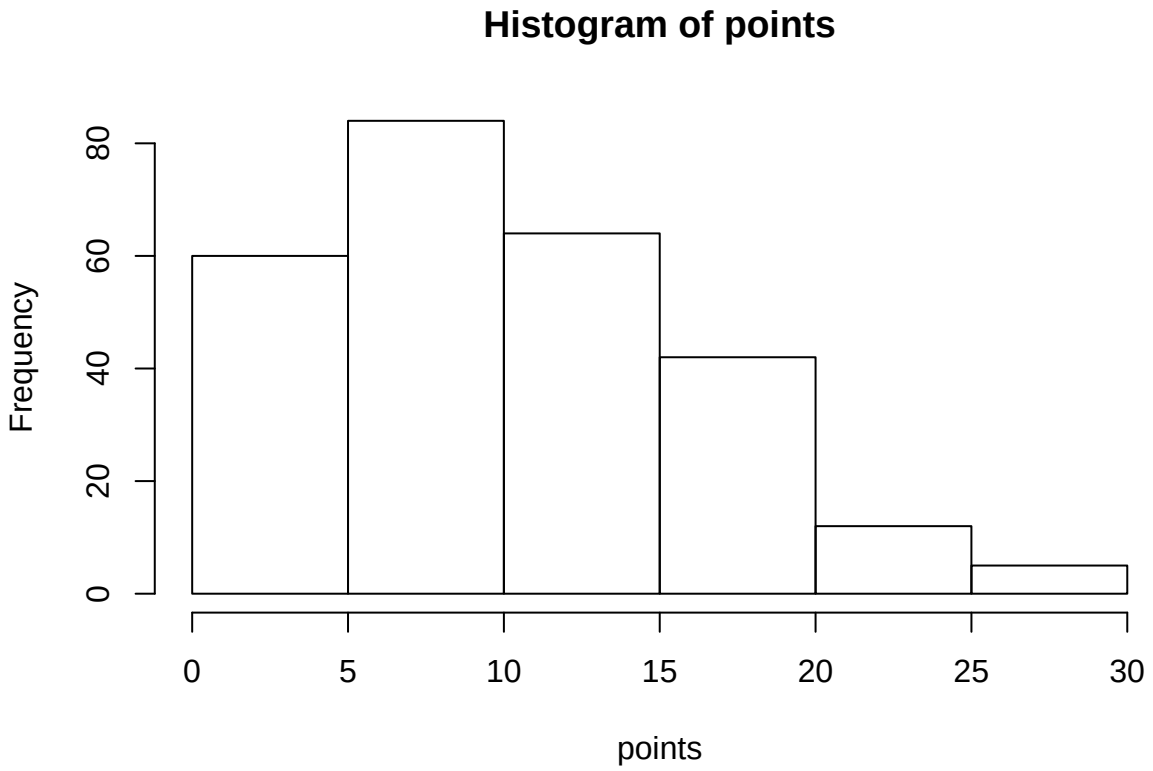
*Plot a histogram of points-per-game.*

Histograms give a visual idea of the frequency distribution of a variable. A way for plotting a simple histogram would be by using `hist()` and adding `points` variable.

```
# Histogram

    # xlab = rename the axis X
    # main = title of the plot

hist(nba$points, xlab= "points", main = "Histogram of points")
```




## Question D

*Produce a scatterplot of points-per-game versus years in league.*

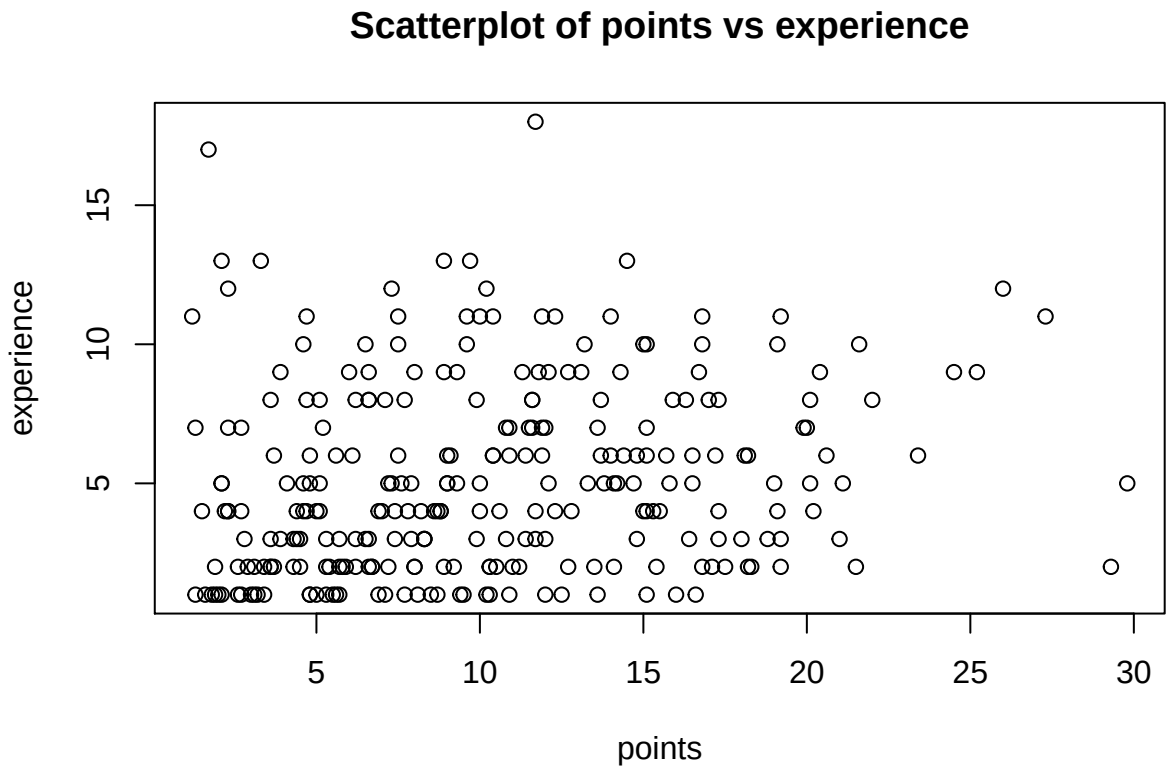
Scatterplots represent the association between two variables. A way of doing it is by using function `plot()`. Until now we have been using an object (e.g variables) that is "residing" in another object (e.g. a data frame). The easy (and natural) way to refer to them is by indexing with `$`. However, when there are more objects involved (e.g. several variables), typing `$` systematically can be confusing (specially with long names) and produce errors. In order to avoid this, it is possible to use `with()` to *attach* the data frame and use the variables independently[4].

[4] R has a `attach()` function that can be used to make objects within dataframes accessible in R without calling to the data frame. Yet the use of this function is not recommended. See the Google R Style Guide for details.

```r
# Scatterplot

    # xlab = rename the axis X
    # ylab = rename the axis Y
    # main = title of the plot

with(nba, plot(points, exper, xlab= "points", ylab = "experience",
               main = "Scatterplot of points vs experience"))
```




## Question E

*Run a regression of points-per-game on years in league, age, years played in college and position dummies.*

```r
 library(stats)
 library(stargazer)

model1 = lm(points ~ exper + age + coll + forward + center, data = nba)
summary(model1)
```

```
##
## Call:
## lm(formula = points ~ exper + age + coll + forward + center,
##     data = nba)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.2157  -4.1531  -0.8196   3.0151  23.7349
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  39.1537     6.9887   5.602 5.36e-08 ***
## exper         1.4328     0.2969   4.825 2.38e-06 ***
## age          -1.1330     0.2979  -3.803 0.000178 ***
## coll         -1.1933     0.4546  -2.625 0.009180 **
## forward      -0.8618     0.7522  -1.146 0.252976
## center       -2.6214     0.9814  -2.671 0.008034 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.574 on 261 degrees of freedom
## Multiple R-squared:  0.1303, Adjusted R-squared:  0.1136
## F-statistic: 7.817 on 5 and 261 DF,  p-value: 7.108e-07
```

`lm` estimates a linear model using ordinary least squares (OLS) and returns `model1`, a fitted-model object[5]. The variable before "`~`" indicates the dependent variable whereas the variables in the right side are considered the set of explanatory regressors. `summary()` allow to visualise the output of the regression.

```
stargazer(model1,  type = "latex",
          title = "Points per game", header = FALSE)
```

`stargazer()` produces a table (Table 1) with the results from the regression corresponding to fitted model object `model1`.

We can see that the experience (`exper`) has a statistically significant influence in the perfomance. Concretely an additional year of experience implies 1.4 additional points per game. Age (`age`) and years playing at college (`coll`) play a negative role. Whereas the fact of being a year older deteriorates the performance by 1.13 points per game, having played in college before seems to decrease the performance in about 1.2 points per game. Both negative effects are statistically significant. In general, all variables with the exception to `forward` are statistically significant at the 5% level of significance.

The $R^2$ indicates the level of variance in the data that is explained by the model. In this case is about the 13%. Likewise, the F-statistic indicates the results of an F test of the hypothesis that all regressors are jointly significant. In this case the intercept term is excluded.

## Question F

If players can be drafted in early years (e.g. college or high school in some cases) then the negative effect on the performance may be capturing the time when players are not essentially playing in the NBA and therefore not scoring points either.

## Question G

*Look at the correlation matrix*

`rcorr()` is the function for creating a correlation matrix and obtaining the levels of significance. `pander()` creates a table with the results of the correlation matrix (Table 2).

[5]The components of this object can be retrieved by using the function `names()`

Table 1: Points per game

| | *Dependent variable:* |
|---|---|
| | points |
| exper | $1.433^{***}$ |
| | (0.297) |
| age | $-1.133^{***}$ |
| | (0.298) |
| coll | $-1.193^{***}$ |
| | (0.455) |
| forward | $-0.862$ |
| | (0.752) |
| center | $-2.621^{***}$ |
| | (0.981) |
| Constant | $39.154^{***}$ |
| | (6.989) |
| Observations | 267 |
| $R^2$ | 0.130 |
| Adjusted $R^2$ | 0.114 |
| Residual Std. Error | 5.574 (df = 261) |
| F Statistic | $7.817^{***}$ (df = 5; 261) |
| *Note:* | $^{*}$p<0.1; $^{**}$p<0.05; $^{***}$p<0.01 |

```r
library(Hmisc)
library(dplyr)
library(devtools)
library(pander)

# select variables from the model
vars_mod = nba %>% select(exper, age, coll, forward, center)

# correlation matrix
cor_mat <- rcorr(as.matrix(vars_mod), type = "pearson")
emphasize.strong.cells(which(cor_mat[[3]] < 0.001, arr.ind = TRUE))
pander(cor_mat[[1]], caption = "Correlation matrix")
```

Table 2: Correlation matrix

| | exper | age | coll | forward | center |
|---|---|---|---|---|---|
| **exper** | 1 | **0.9411** | 0.08714 | -0.003706 | 0.06894 |
| **age** | **0.9411** | 1 | 0.07395 | 0.00122 | 0.07917 |
| **coll** | 0.08714 | 0.07395 | 1 | -0.05017 | -0.02503 |
| **forward** | -0.003706 | 0.00122 | -0.05017 | 1 | **-0.3789** |
| **center** | 0.06894 | 0.07917 | -0.02503 | **-0.3789** | 1 |

We can also extract the p-values associated with the significance levels of the correlations. Hence, the p-values determine whether the correlations are significant. Analogously, the p-values can be represented in a table with `pander()` (Table 3).

```r
pander(cor_mat[[3]], caption = "Correlation matrix (p-values)")
```

Table 3: Correlation matrix (p-values)

| | exper | age | coll | forward | center |
|---|---|---|---|---|---|
| **exper** | NA | 0 | 0.1556 | 0.9519 | 0.2616 |
| **age** | 0 | NA | 0.2285 | 0.9842 | 0.1972 |
| **coll** | 0.1556 | 0.2285 | NA | 0.4142 | 0.6839 |
| **forward** | 0.9519 | 0.9842 | 0.4142 | NA | 1.521e-10 |
| **center** | 0.2616 | 0.1972 | 0.6839 | 1.521e-10 | NA |

*Do you need to worry about multicollinearity?*

Normally, a strong correlation is considered when the magnitude of the Pearson correlation coefficient ($r$) is $> 0.5$. According to the results from the correlation matrix, `exper` and `age` show high positive correlation (0.94). Moreover, this correlation is significant as we can see in Table 3.

## Question H

*Now consider an extension of the basic model. Generate a new variable which is experience squared and include it in the regression.*

With `mutate` from `dplyr` it is possible to create new variables.

```
library(dplyr)
library(tibble)

# create a new variable

nba = nba %>% mutate(expersq = exper^2)
head(nba)
```

```
##   marr  wage exper age coll games minutes guard forward center points
## 1    1 1.002     4  27    4    77    2867     1       0      0   15.5
## 2    1 2.030     5  28    4    78    2789     1       0      0   13.3
## 3    0 0.650     1  25    4    74    1149     0       0      1    5.5
## 4    0 2.030     5  28    4    47    1178     0       1      0    7.3
## 5    0 0.755     3  24    4    82    2096     1       0      0   10.8
## 6    0 2.015     9  31    4    82    1971     0       1      0   11.3
##   rebounds assists draft allstar avgmin black children expersq
## 1      3.9     4.5    19       0 37.234     1        0      16
## 2      2.5     8.8    28       0 35.756     1        1      25
## 3      3.3     0.2    19       0 15.527     1        0       1
## 4      5.1     1.5     1       0 25.064     1        0      25
## 5      4.3     2.6    24       0 25.561     1        0       9
## 6      4.9     1.5     4       0 24.037     1        0      81
```

*Holding age, coll, center and forward fixed, at what value of experience does the next year of experience reduce points-per-game?*

We run first the model including the new variable `expersq`

```
  library(stats)
  library(stargazer)

 model2 = lm(points ~ exper+expersq+age+coll+center+forward, data = nba)
```

```
stargazer(model2, type = "latex",
          title = "Points per game (model 2)", header = FALSE)
```

Table 4 reflects the results of this new model (`model2`). The experience is not a positive factor for the performance at 0.072. This is a plausible result since players apart from being more experienced also get older.

## Question I

*Now you want to explain the log(wage)*

Similarly to former exercises, we create `logwage` by using `mutate()`.

```
library(dplyr)

# create a new variable

nba = nba %>% mutate(logwage = log(wage))
head(nba)
```

Table 4: Points per game (model 2)

| | *Dependent variable:* |
|---|---|
| | points |
| exper | $2.287^{***}$ |
| | (0.406) |
| expersq | $-0.072^{***}$ |
| | (0.024) |
| age | $-1.049^{***}$ |
| | (0.295) |
| coll | $-1.335^{***}$ |
| | (0.450) |
| center | $-2.324^{**}$ |
| | (0.971) |
| forward | $-0.862$ |
| | (0.741) |
| Constant | $35.676^{***}$ |
| | (6.976) |
| Observations | 267 |
| $R^2$ | 0.160 |
| Adjusted $R^2$ | 0.141 |
| Residual Std. Error | 5.488 (df = 260) |
| F Statistic | $8.256^{***}$ (df = 6; 260) |
| *Note:* | $^{*}$p<0.1; $^{**}$p<0.05; $^{***}$p<0.01 |

```
##   marr  wage exper age coll games minutes guard forward center points
## 1    1 1.002     4  27    4    77    2867     1       0      0   15.5
## 2    1 2.030     5  28    4    78    2789     1       0      0   13.3
## 3    0 0.650     1  25    4    74    1149     0       0      1    5.5
## 4    0 2.030     5  28    4    47    1178     0       1      0    7.3
## 5    0 0.755     3  24    4    82    2096     1       0      0   10.8
## 6    0 2.015     9  31    4    82    1971     0       1      0   11.3
##   rebounds assists draft allstar avgmin black children expersq
## 1      3.9     4.5    19       0 37.234     1        0      16
## 2      2.5     8.8    28       0 35.756     1        1      25
## 3      3.3     0.2    19       0 15.527     1        0       1
## 4      5.1     1.5     1       0 25.064     1        0      25
## 5      4.3     2.6    24       0 25.561     1        0       9
## 6      4.9     1.5     4       0 24.037     1        0      81
##        logwage
## 1  0.001998003
## 2  0.708035793
## 3 -0.430782916
## 4  0.708035793
## 5 -0.281037530
## 6  0.700619195
```

For estimating the new model (`model3`) we apply the following code

```
model3 <- lm(logwage~points+exper+expersq+age+coll, data = nba)
```

```
stargazer(model3, type = "latex",
          title =  "Model 3", header = FALSE)
```

*How do you interpret the results?*

The results are contained in Table 5. The interpretation of the variables differs both regressors and dependent variables are transformed. A log transformation in depedent variable in this case will interpreted as a percent change. Hence, in this case the wage seems to be more influenced by the experience than the performance. Particularly, whereas the points obtained would suppose an increase of 7% in the wage, having an additional year of experience would suppose an increase of the 22.3%.

## Question J

*Test whether age and coll are jointly significant in the regression from (i). What does this imply about whether age and education have a separate effect on wage, once productivity and senority are controlled for? (Hint: You need to estimate both the unrestricted and the restricted model. Then you can use the anova() command to get the sums of squared residuals for the two models.)*

```
library(pander)
mod_unrest <- lm(logwage~points+exper+expersq+age+coll, data = nba)
mod_rest <- lm(logwage~points+exper+expersq, data = nba)

anova(mod_rest, mod_unrest)
```

```
## Analysis of Variance Table
##
```

Table 5: Model 3

| | *Dependent variable:* |
|---|---|
| | logwage |
| points | $0.077^{***}$ |
| | (0.007) |
| exper | $0.223^{***}$ |
| | (0.049) |
| expersq | $-0.007^{***}$ |
| | (0.003) |
| age | $-0.050$ |
| | (0.035) |
| coll | $-0.038$ |
| | (0.052) |
| Constant | $-0.098$ |
| | (0.838) |
| Observations | 267 |
| $R^2$ | 0.493 |
| Adjusted $R^2$ | 0.483 |
| Residual Std. Error | 0.630 (df = 261) |
| F Statistic | $50.742^{***}$ (df = 5; 261) |
| *Note:* | $^{*}$p<0.1; $^{**}$p<0.05; $^{***}$p<0.01 |

```
## Model 1: logwage ~ points + exper + expersq
## Model 2: logwage ~ points + exper + expersq + age + coll
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    263 104.62
## 2    261 103.64  2   0.98215 1.2367  0.292
```

```
pander(anova(mod_rest, mod_unrest), caption = "Comparison of models")
```

Table 6: Comparison of models

| Res.Df | RSS | Df | Sum of Sq | F | Pr(>F) |
|---|---|---|---|---|---|
| 263 | 104.6 | NA | NA | NA | NA |
| 261 | 103.6 | 2 | 0.9821 | 1.237 | 0.292 |

`anova()` can be used for model comparison. Table 6 shows the results derived from the `ANOVA` test. The models we are comparing are nested - i.e. both models share a set of regressors and have the same outcome but one of them (e.g. the unrestricted model) has additional regressors. The results of Table 4 reveal that the p value is 0.292. This indicates that the joint variance of two variables such as `age` and `coll` is not meaningful to the model so it is not possible to reject the null hypothesis that both coefficients are 0. Therefore, the changes in the wage as a result of `age` and `coll` are 0.