Final Assignment

Simone Dal Ben

2023-03-27

1) Introduction

The study is based on the dataset that refers to NBA players from kaggle [kaggle] (https://www.kaggle.com/datasets/justinas/nba-players-data). This data set contains over two decades of data on each player who has been part of an NBA teams' roster. It captures demographic variables such as age, height, weight and place of birth, biographical details like the team played for, draft year and round. In addition, it has basic box score statistics such as games played, average number of points, rebounds, assists, etc.. The dataset has 12305 observations and 22 variables. The pull initially contained 52 rows of missing data. The gaps have been filled using data from Basketball Reference. I am not aware of any other data quality issues. I decided to take only the statistics of the player that played in the last three seasons because the dataset has to many observations and I can't compute some operations (for example the "shapiro test"). So now my dataset has 1674 observations. I decided to create two categorical variables that are the "age_class" (I divided the "age" variable in 3 levels: "olders", "middle" and "younger") and the "Country" (a dummy variable: "USA" and "foreign"). So, with this manipulations of the dataset, it has the response and the 13 predictors. The variables are:

```
names(NBA[1,])
```

```
##  [1] "player_height" "player_weight" "gp"            "pts"
##  [5] "reb"           "ast"           "net_rating"   "oreb_pct"
##  [9] "dreb_pct"      "usg_pct"       "ts_pct"       "ast_pct"
## [13] "age_class"     "Country"
```

- **player_height**= height of the players (in centimeters).
- **player_weight** = weight of the players (in kilograms).
- **gp** = games played throughout the season
- **reb** = average number of rebounds grabbed
- **ast** = average number of assists distributed
- **net_rating** = team's point differential per 100 possessions while the player is on the court
- **oreb_pct** = percentage of available offensive rebounds the player grabbed while he was on the floor
- **dreb_pct** = percentage of available defensive rebounds the player grabbed while he was on the floor
- **usg_pct** = percentage of team plays used by the player while he was on the floor (FGA + Possession Ending FTA + TO) / POSS)
- **ts_pct** = measure of the player's shooting efficiency that takes into account free throws, 2 and 3 point shots (PTS / (2x(FGA + 0.44 x FTA)))
- **ast_pct** = percentage of teammate field goals the player assisted while he was on the floor

2) Applied goals

In this assignment, the goal is to rely on a linear regression model to explore associations between the response (the dependent variable) and some predictors (independent). I choose the variable "pts" as my response variable and it represents the average number of points scored. In other words the aim of this analysis is to identify which statistics of the players have a significant impact on the average number of points scored and the magnitude of their impact.

3) Graphical representation of the data

Graphs a) and b) in figure 1 show the histogram and the boxplot of the response variable **points**: both of them highlight a right-skewed distribution. From the histogram it's immediate to see that the majority of the observations is condensed between 0 and 10 average number of points; as the response grows, the number of observations decreases. The blue vertical line represents the mean of the distribution and is placed around 8.5. The same conclusion can be reached by looking at the boxplot, that shows a median value equal to 7: it's a lower value than the mean one, and this confirms the asymmetry of the distribution of the response. In the boxplot all the observation with points higher than 24 are extreme points. Plot c) in figure 1 represents the relationship between the categorical **age_class** and **points**: what can be said by looking at this graph is that, on average, players that are in the olders class have a higher average number of points than the others.

```
par(mfrow=c(1,3))
hist(NBA$pts,main = "histogram of points",xlab = "NBA average points",
     ylab = "Relative frequenties",freq = FALSE,breaks = 20, col = "red",
     xlim = c(0,35),ylim = c(0,0.1))
abline(v=mean(NBA$pts),lwd=3,col="blue")
est_density=density(NBA$pts)
lines(est_density,lwd=3,col="black")

boxplot(NBA$pts,main="box-plot of points",xlab="values",col="red",horizontal = TRUE)

plot(NBA$age_class,NBA$pts,xlab="Age class",ylab="Average number of points",
     main="box-plot of points\n for age class",col=c(2,3,4),pch=3)
```

Now I graph the *scatterplot matrix* that underline the relationship between all the quantitative variables:

```
plot(~ player_height+player_weight+gp+pts+reb+oreb_pct+dreb_pct+ast+net_rating+usg_pct+ts_pct+ast_pct,
     data = NBA,col="yellow")
```

As we can see, the most striking case appears to be the link between *ast* and *ast_pct* which resembles almost a straight line: since both these variables offer a measure of the assist it's logical that the two of them have a strong correlation, that will be addressed later.

4) Linear regression model

```
ols=lm(pts ~ .,data=NBA)
sum=summary(ols)
```

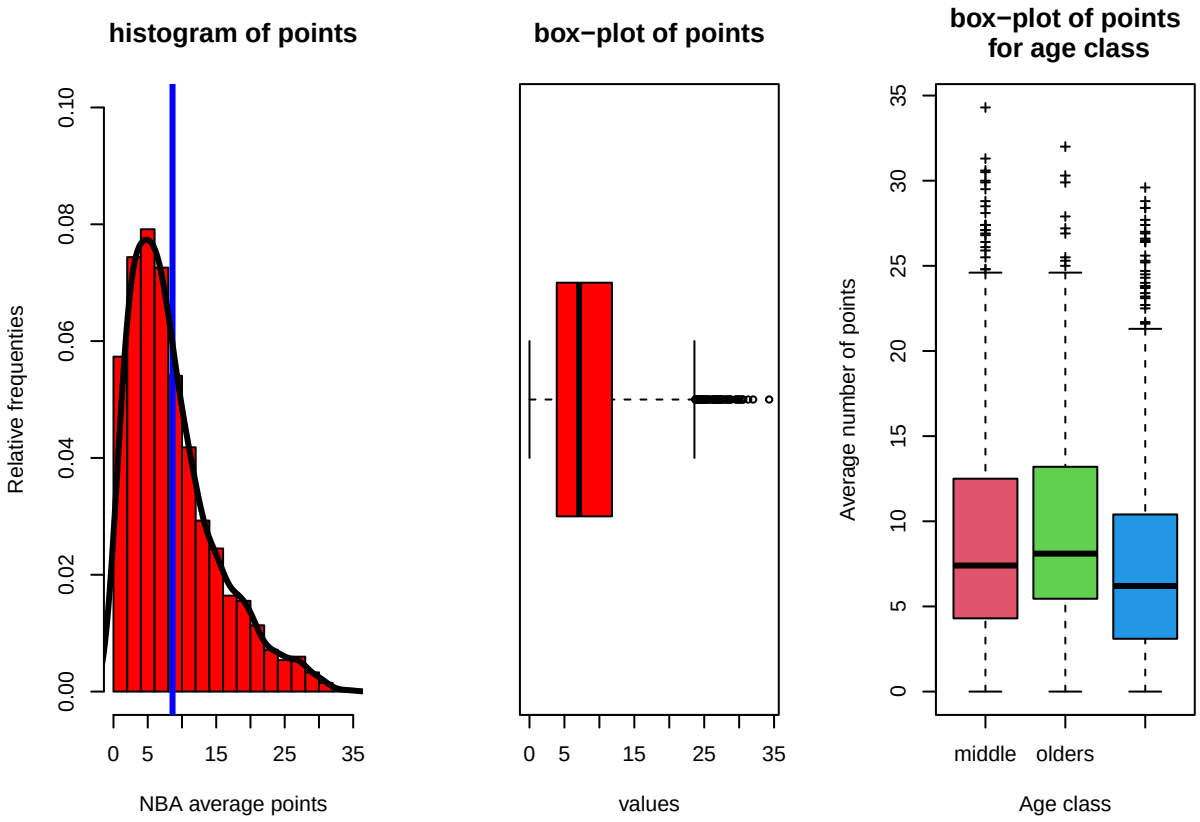


Figure 1: Visualization of points distribution

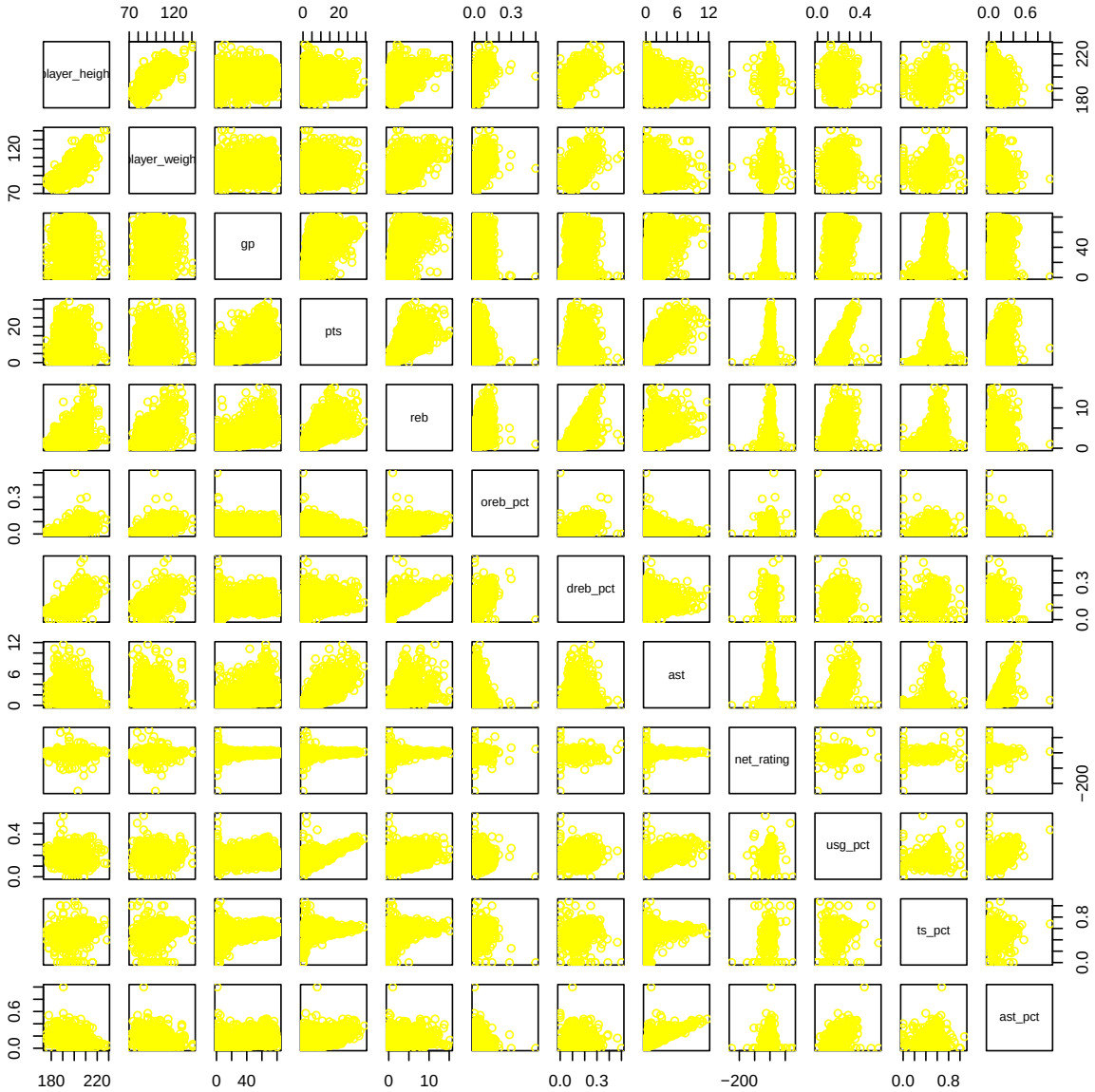


Figure 2: Scatterplot matrix

This is the model with all the covariates; most of them are statistically significant (a low p-value) and so they can be used for the best model. But, we must perform a best subset selection to identify the set of predictors that are most relevant for predicting the response for each possible number of parameters (in this case from 1 to 13).

```
ols_bss=regsubsets(pts ~ .,data = NBA,nvmax = 13) #by default the maximum number of cov is 8, in my cas
summ_bss=summary(ols_bss)
```

5) BEST MODEL

Once the best model for each number of parameters are known, the following step is to determine which is the optimal number of regressors that must be used to fit the model. This decision can be taken by using different criteria: the BIC, the AIC, the Cp Mallow's and the adjusted R^2.

```
par(pty = "s", mfrow = c(1, 4),mar = c(2, 1, 2, 1))
#BIC
plot(summ_bss$bic, type = "b", pch = 19,
     xlab = "Number of predictors", ylab = "", main = "Drop in BIC")
abline (v = which.min(summ_bss$bic),col = 2, lty = 2)
#Cp
plot(summ_bss$cp, type = "b", pch = 19,
     xlab = "Number of predictors", ylab = "", main = "Mallow' Cp")
abline (v = which.min(summ_bss$cp), col = 2, lty = 2)
# Adj R^2
plot(summ_bss$adjr2, type = "b", pch = 19,
     xlab = "Number of predictors", ylab = "", main = "Adjusted R²")
abline (v = which.max(summ_bss$adjr2), col = 2, lty = 2)
# AIC
p = 13
n = nrow(NBA)
j = 1
aic = matrix(NA, p, 1)
for(j in 1:p){
  aic[j]= summ_bss$bic[j] - (j + 2) * log(n) + 2 * (j + 2)
  #aic[j] <- n*log(summ_bss$rss[j]/n)+(j+2)*2
}
plot(aic, type = "b", pch = 19, xlab = "Number of predictors", ylab = "", main = "Drop in AIC")
abline (v = which.min(aic), col = 2, lty = 2)
```

From the graphs obtained the model with 13 predictors results to have the lowest drop in AIC, the higher adjusted R^2 and the lowest Mallow's Cp, while the lowest drop in BIC corresponds to the model with 9 predictors. However it can be seen from all the plots that there is a plateau and therefore there is no big difference between the model with 13 parameters and the one with 9. For Occam's razor principle, it is better to take the model with less predictors and so I take the one with 7 predictors (the values of criterias are similar to the one with 9 predictors). We now want to find the best model according to the cross-validation approach (normally it is used for prediction that isn't the goal of this analysis and so the previous 4 criterias are more suited to determine the optimal number of regressors). This approach requires to perform the best subset selection within each of the k training sets (in this case we set k=10).

```
k=10
folds = sample(1:k,nrow(NBA), replace =TRUE)
cv.errors = matrix(NA,k,p,dimnames=list(NULL,paste(1:p)))
```

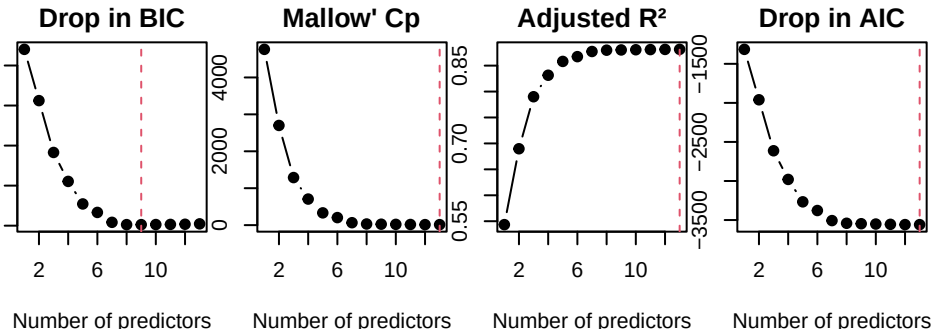


Figure 3: Best model according to different crietria

```
j=1
for(j in 1:k){
  best.fit =regsubsets(pts ~ ., data=NBA[folds!=j,], nvmax = 13)
  for(i in 1:p) {
    mat = model.matrix(pts ~ .,NBA[folds==j,])
    coefi <- coef(best.fit ,id = i)
    xvars <- names(coefi)
    pred <- mat[,xvars]%*% coefi
    cv.errors[j,i] <- mean((NBA$pts[folds==j] - pred)**2)
    }
}
cv.mean <- colMeans(cv.errors)
# round(cv.mean, 4)
par(mfrow = c(1, 1))
plot(cv.mean ,type = "b", pch = 19, xlab = "Number of predictors", ylab = "CV error")
abline(v = which.min(cv.mean), col = 2, lty = 2)
```

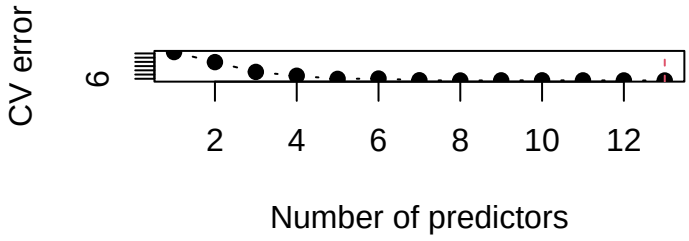


Figure 4: Cross-validation error of each size for the best model

According to the CV errors the optimal choice is to fit the model with 13 parameters. But, for the Occam's razor principle (like before), the best model is the one with 7 predictors. So the final best model is the following:

```
ols_7=lm(pts~ast+ast_pct+reb+dreb_pct+oreb_pct+usg_pct+ts_pct,data = NBA)
summ_7=summary(ols_7)
```

6) COLLINEARITY ISSUE

The scatterplot matrix in figure 2 showed a linear relationship between ast and ast_pct: to solve this problem the latter variable can be dropped from the model.

```
NBA=NBA[,-12]
```

To study the correlation between the variables selected to fit the model it's possible to represent for each of them their **VIF**. The choice of representing the **VIF** instead of the correlation matrix was made because the **VIF** takes into account the correlation between one regressor and all the other ones in the dataset, while a correlation matrix focuses only on the bivariate correlations.

```
VIF=vif(ols_7)
plot(VIF, pch=16,ylim=c(0,6), ylab="Vif values", main="Variance Inflation plot")
abline(h = 5, col = 2, lty = 2, lwd = 2)
text(x = VIF, labels = names(VIF), cex = 0.8, pos = 3)
```

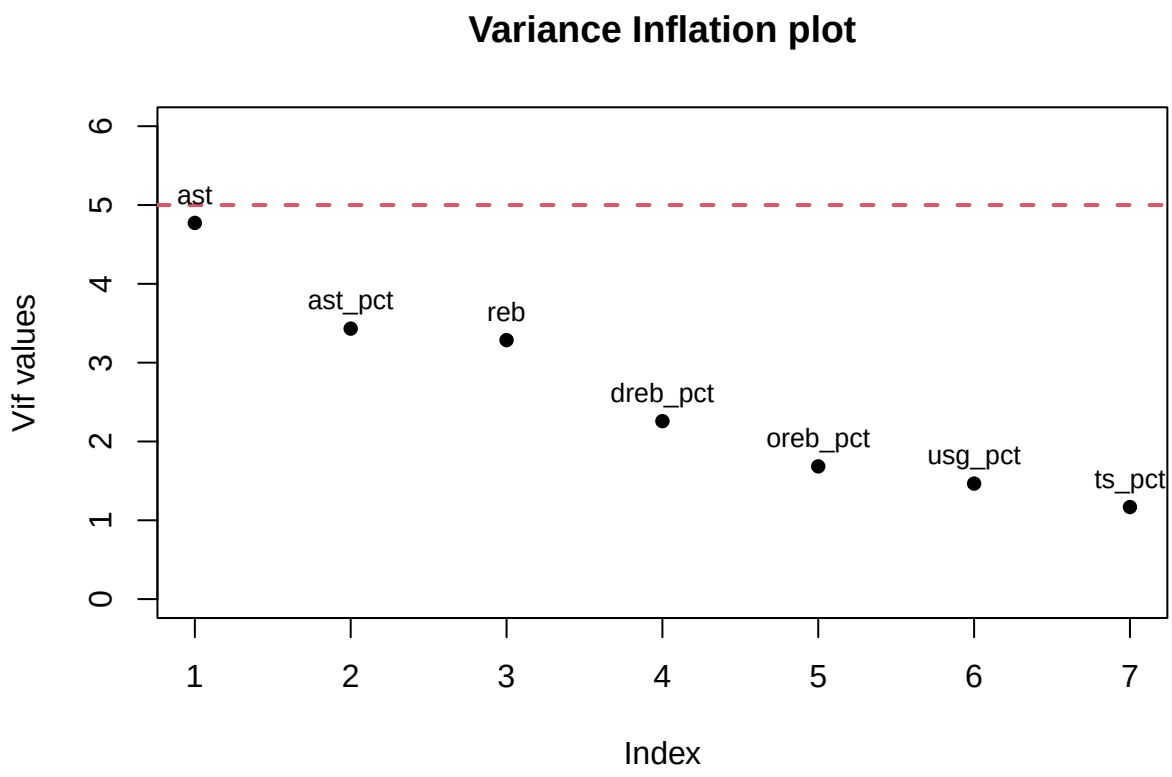


Figure 5: VIF plot

The smallest possible value for VIF is 1, which indicates the complete absence of collinearity. Typically, in practice there is a small amount of collinearity among the predictors. As a rule of thumb, a VIF value that exceeds 5 or 10 indicates a problematic amount of collinearity. As we can see from Figure 5 the VIF values are all below 5, therefore it can be stated that there are no collinearity issues.

7) Diagnostic

7.1 Linearity

The first step of the diagnostic analysis is to verify how the residuals plotted versus the coefficients and the fitted values are distributed: the results are showed in figure 6.

```
ols_6=lm(pts~ast+reb+dreb_pct+oreb_pct+usg_pct+ts_pct,data = NBA)
residualPlots(ols_6,test=F)
```

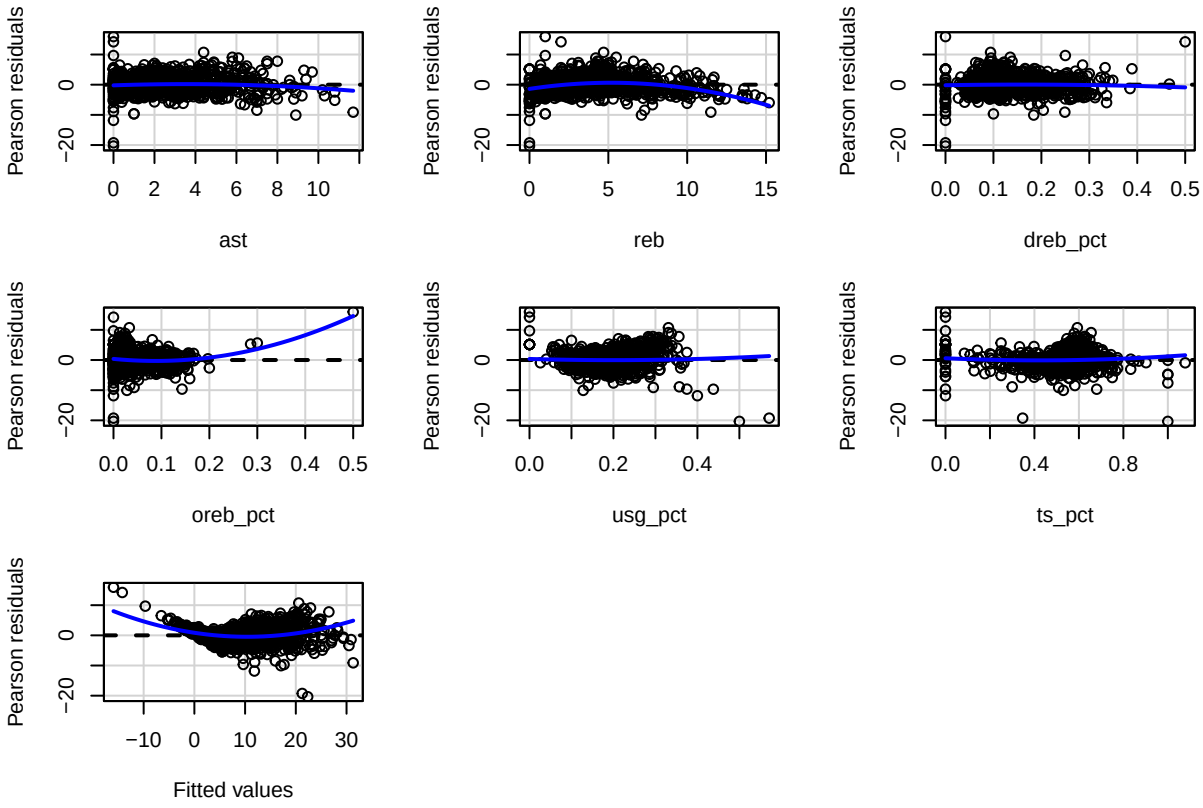


Figure 6: Residuals vs regressors and fitted values

Figure 5 shows that there are problems concerning the linearity assumption: the variables *ast*, *oreb_pct*(the worst one) and *reb* have a plot where the Pearsons residuals follows a non perfectly linear trend. A possible solution would be to transform those variables.

7.2 Homoschedasticity

The variance of the residuals, which should be constant, is clearly heterogeneous: the plot of the fitted values against the residuals (the last graph in figure 6) isn't a *null plot*, the residuals are not randomly located. A remedy for non constant variance is to transform the response variable.

7.3 Normality

To check whether the residuals are normally distributed, one can rely on the QQ-plot (plot b in figure 7) and on the Shapiro-Wilk test: both this tools shows that this property is not satisfied, since form the graph it's clear that the residuals distribution has long tails and the p-value obtained from the test is close to 0; this problem can be addressed by implementing a non parametric test.

```
par(pty = "s", mfrow = c(2, 2),mar = c(2, 1, 2, 1))
plot(fitted(ols_6),residuals(ols_6),pch=16,cex=0.8,xlab = "Fitted values",
     ylab = "Residuals",col=4, main = "a) Residuals vs Fitted Values",
     ylim = c(-10,10))
abline(h=0,col=2,lwd=2)
qqnorm(residuals(ols_6),plot.it = TRUE,col=12,main = "b) Q-Q plot of the residuals")
qqline(residuals(ols_6),col=2,lwd=2)
#solve both issues of non linearity and non constant variance
#first I take the players that played at least 10 matches in the season because it is reasonable and so
NBA10=filter(NBA,gp>10)
best_model=lm(log(pts)~log(reb)+oreb_pct+log(dreb_pct)+log(usg_pct)+log(ts_pct),data = NBA10)
plot(fitted(best_model),residuals(best_model),pch=16,cex=0.8,xlab = "Fitted values",
     ylab = "Residuals",col=4, main = "c) Residuals vs Fitted Values\n with transformations")
abline(h=0,col=2,lwd=2)

qqnorm(residuals(best_model),plot.it = TRUE,col=12,main = "d) Q-Q plot of the residuals\n with transfor
qqline(residuals(best_model),col=2,lwd=2)
```

So, I do both transformations of predictors and the response. I do also a new best subset selection because now the response and some covariates are in a logarithmic scale. After this, I conclude that the *best model* has the 5 number of predictors, *ast* is removed. In the plot c) we see that the shape is better than before: it is more similar to a null plot. But the problem of non linearity remains, in fact the Q-Q plot (plot d) still has a long-tailed distribution and the shapiro test has a p-value very close to 0.

8) Outliers, high leverage points and influential points

To plot the outliers, the standardized residual *rsta* must be taken into account: a point is regarded as outliers if its studentized residual is higher than 3 or lower than -3. A point is considered to be a leverage point if its leverage *hat* it's higher than 2(p+1)/2: for this dataset the threshold is equal to 2 x (5 + 1)/1426 = 0.0084. Finally, a point is influential if its Cook's Distance *cook* it's higher than 0.5.

```
rsta=rstandard(best_model)
hat=hatvalues(best_model)
cook=cooks.distance(best_model)
```

```
par(mfrow=c(1,3))
# Outliers
plot(rsta,ylab = "studentized residuals",main = "Outliers",col="antiquewhite",pch=16)
```

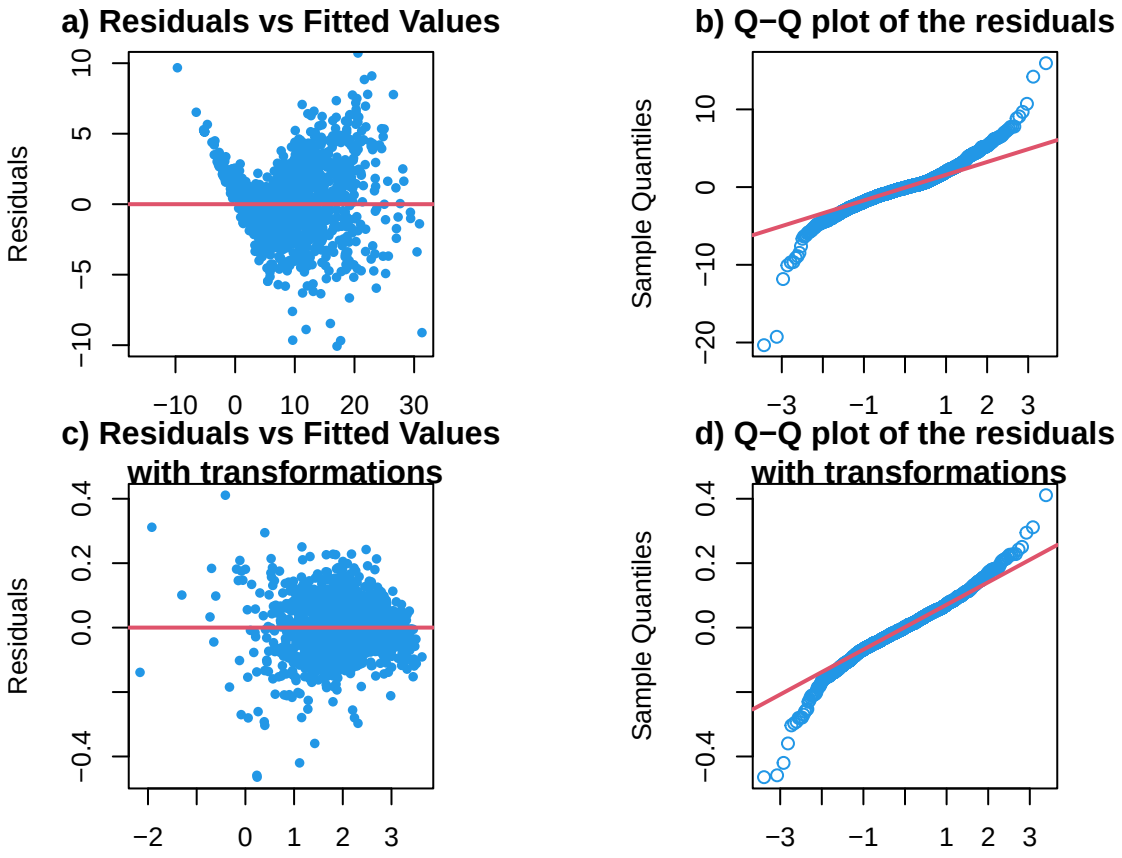


Figure 7: representation of the residuals

```
abline(h=3,lty=2,col=2,lwd=2)
abline(h=-3,lty=2,col=2,lwd=2)
out=(abs(rsta)>3)
x_o=which(out)
y_o=(rsta)[out]
text(x_o,y_o,labels=x_o,cex=1,pos=2)
# high leverage points
plot(hat,ylab = "Leverages",main = "High leverage points",col="antiquewhite",pch=16)
abline(h=12/nrow(NBA10),lty=2,col=2,lwd=2)
high_lev=hat >0.06
x_hl=which(high_lev)
y_hl=(hat)[high_lev]
text(x_hl,y_hl,labels=x_hl,cex=1,pos=2)

plot(best_model,which = 4, col="antiquewhite")
abline(h=1,col=10)
```

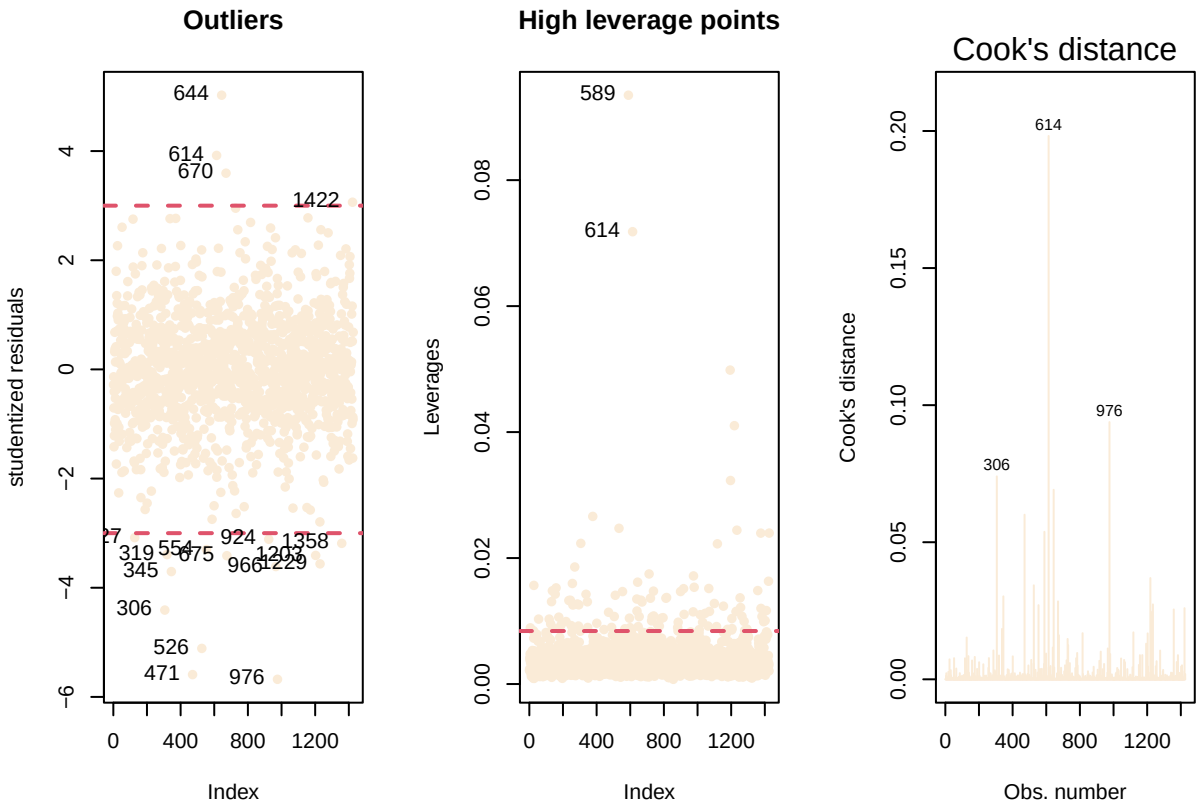


Figure 8: Outliers, high leverage points and influential points

From the graph in figure 8 emerge that there are a lot of outliers looking at the rule of thumb, but the points below the -3 line are closely to that value so they aren't a problem. So I consider as outliers the observations *644,306,526,471* and *976*. Therefore, I remove this data points. Looking the plot of high leverage points, we see that there are several points with a leverage higher than 0.0084, but none of them is an influential point because the Cook's distance of all is below the treshold 0.5. The point with a Cook's Distance higher than 0.1 is the 614th: it is below the treshold and it does not have any significant impact on the model. So,

removing the outliers, the model become:

```
NBA10_o=NBA10[-c(306,644,526,471,976),]
best_model_o=lm(log_pts~log_reb+oreb_pct+log_dreb_pct+log_usg_pct+log_ts_pct,data = NBA10_o)
summ_best_o=summary(best_model_o)
```

9) Coefficients

Now we look at the coefficients of the best model obtained:

```
summ_best_o$coefficients
```

```
##                Estimate  Std. Error   t value      Pr(>|t|)
## (Intercept)   1.8311477 0.031310047  58.48435  0.000000e+00
## log_reb       0.9822984 0.004880400 201.27414  0.000000e+00
## oreb_pct     -4.4109663 0.086387882 -51.06001 1.976263e-323
## log_dreb_pct -0.8348050 0.008833768 -94.50158  0.000000e+00
## log_usg_pct   1.0524945 0.008032226 131.03398  0.000000e+00
## log_ts_pct    1.0801582 0.016642199  64.90478  0.000000e+00
```

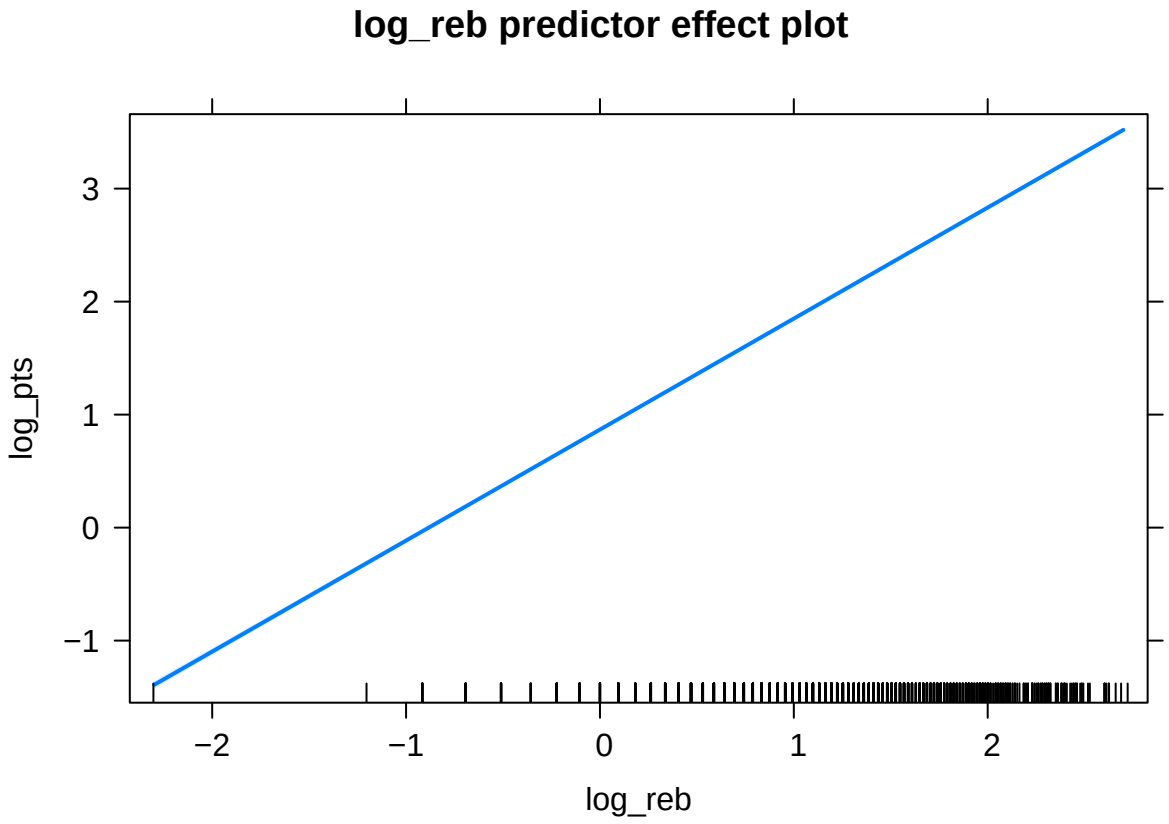
This table contains the 95% confidence interval of the estimated β:

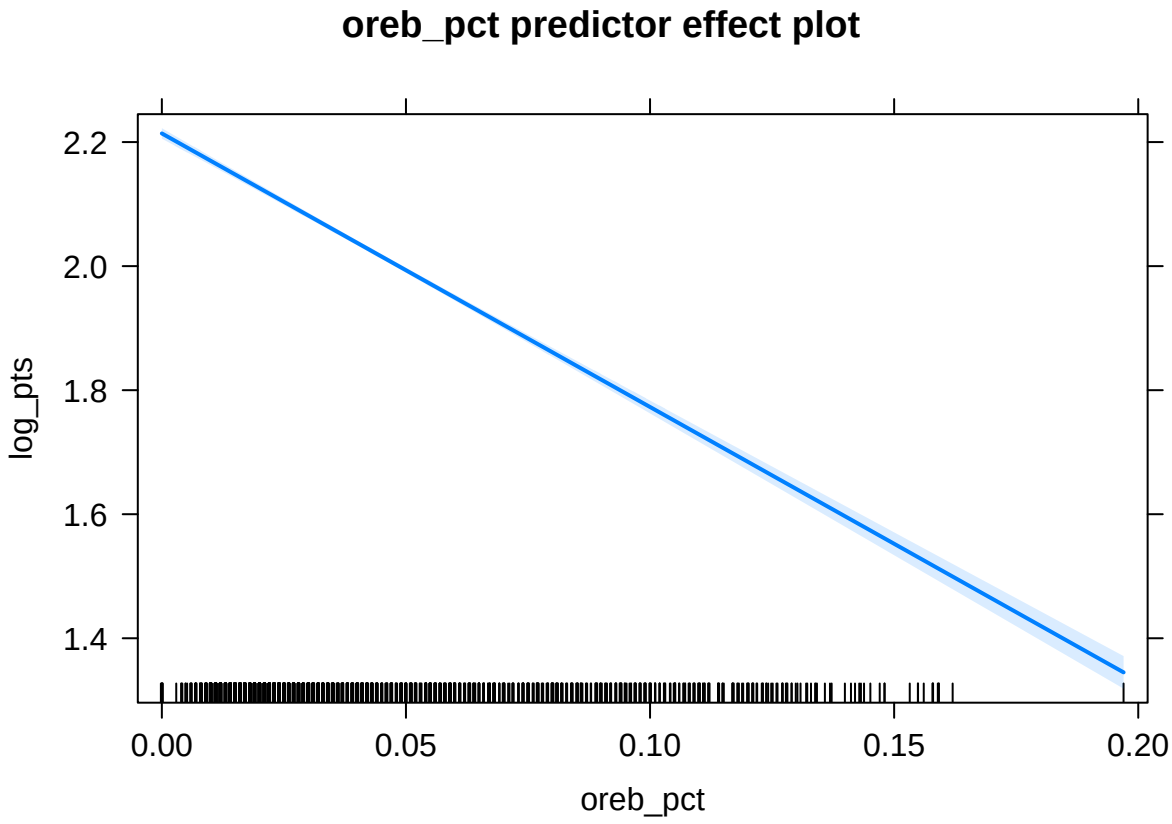
```
t(confint(best_model_o))
```

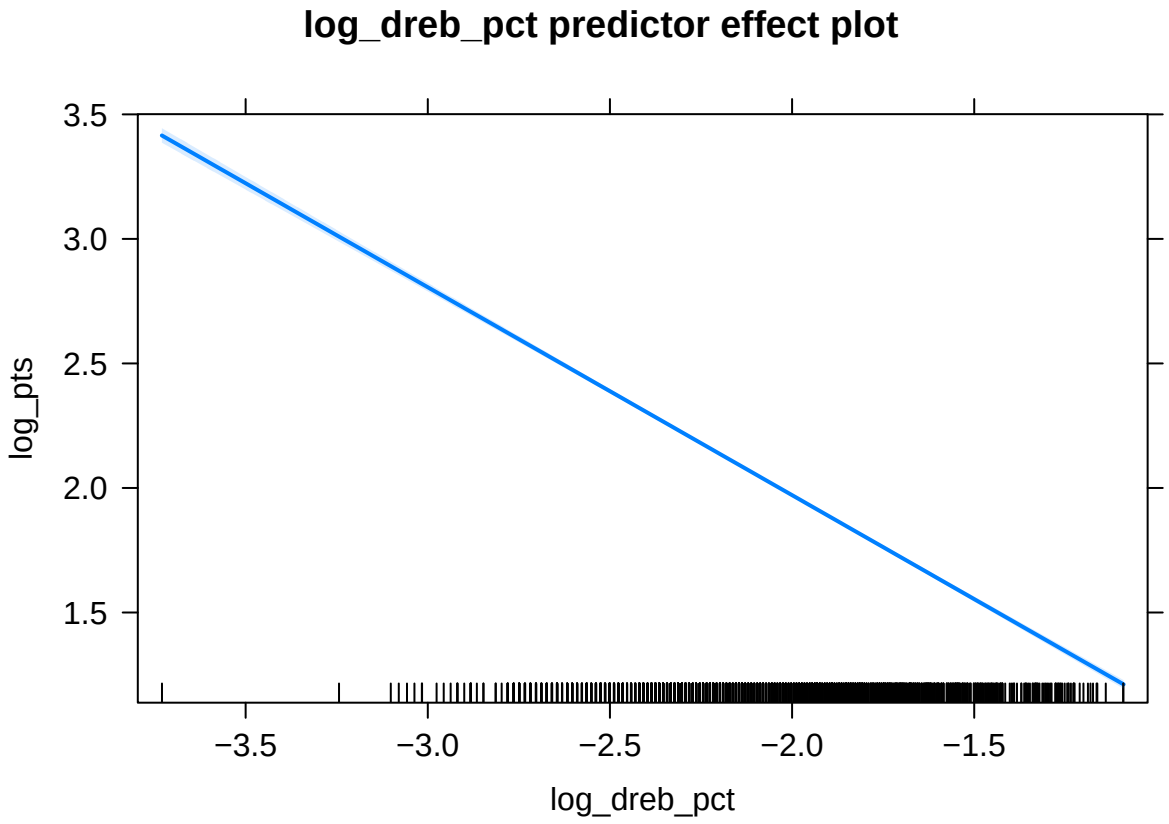
```
##        (Intercept)   log_reb  oreb_pct log_dreb_pct log_usg_pct log_ts_pct
## 2.5 %     1.769729 0.9727248 -4.580428   -0.8521337    1.036738   1.047512
## 97.5 %    1.892567 0.9918720 -4.241504   -0.8174763    1.068251   1.112804
```

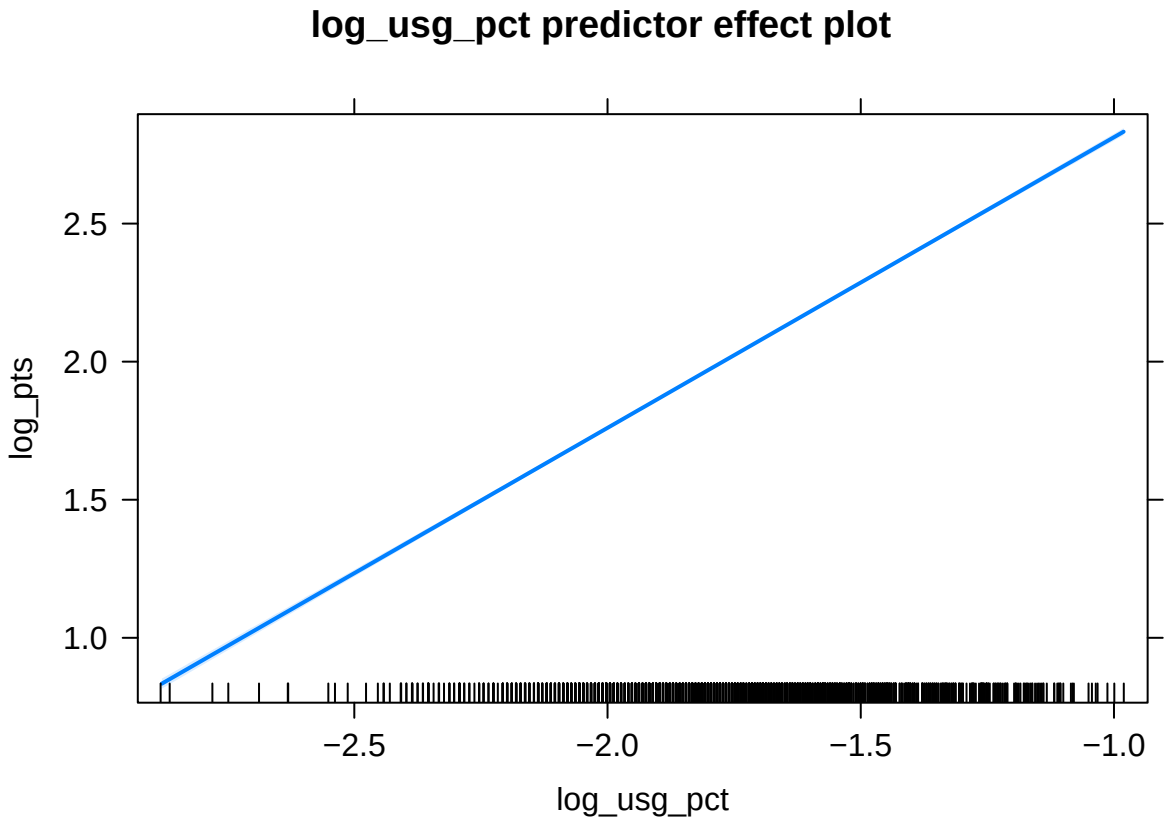
$\hat{\beta}_0$=1.83: the intercept represents the estimated value of log_pts when all the covariates (with the transformations) are equal to 0. If we want to see the average number of points we do the exp of log_pts: $e^{\hat{\beta}_0}$ =6.234 $\hat{\beta}_1$=0.98: this coefficient represents the difference in the log(pts) with a unit increase of log(reb) keeping constant the other variables. So, the average number of points increases by 2.66 (=exp(0.98)) when there is an increase of one unit of the log(reb) and the other covariates are keeping constant. All the others coefficients can be interpreted in the same way because they are all quantitative variables. So, in general, a coefficient β_j is interpreted as the change in the mean function (in the response) for a unit increase in the corresponding X_j variable keeping constant all the other variables.

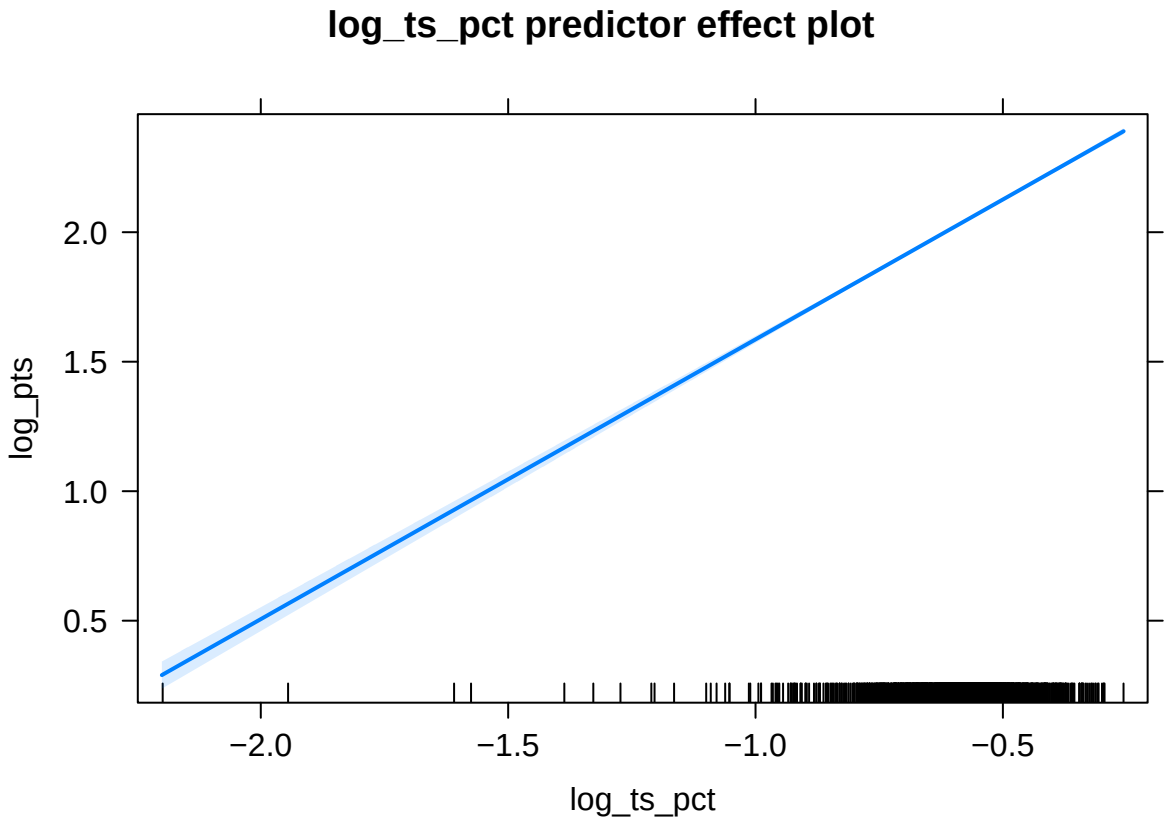
```
par(mfrow=c(1,3))
e1.best=plot(predictorEffect("log_reb",best_model_o,main=" "))
e2.best=plot(predictorEffect("oreb_pct",best_model_o,main=" "))
e3.best=plot(predictorEffect("log_dreb_pct",best_model_o,main=" "))
e4.best=plot(predictorEffect("log_usg_pct",best_model_o,main=" "))
e5.best=plot(predictorEffect("log_ts_pct",best_model_o,main=" "))
grid.arrange(plot(e1.best),plot(e2.best),plot(e3.best),plot(e4.best),plot(e5.best))
```


log_reb predictor effect plot
log_pts
3
2
1
0
−1
−2
−1
0
1
2
log_reb


oreb_pct predictor effect plot
log_pts
2.2
2.0
1.8
1.6
1.4
0.00
0.05
0.10
0.15
0.20
oreb_pct


log_dreb_pct predictor effect plot
3.5
3.0
2.5
2.0
1.5
log_pts
−3.5
−3.0
−2.5
−2.0
−1.5
log_dreb_pct


log_usg_pct predictor effect plot
log_pts
2.5
2.0
1.5
1.0
−2.5
−2.0
−1.5
−1.0
log_usg_pct



10) Test of the coefficients

The previous "table" of the coefficients displays for each regressor the p-value relative to the t-test computed to determine whether the estimated coefficient is equal to 0 (H_0) or different from 0 (H_1): since all of them are below the threshold of 0.05, for all the predictors there is evidence against the null hypothesis, so the conclusion is that all the estimated coefficient are different from 0.

11) Test a group of predictors

ols_t is a model fitted without the 2 predictors with the smallest estimated coefficients, *log_reb* and *log_dreb_pct*: due to their small $\hat{\beta}$, a unit change of these 2 regressors won't have a large impact on the response, so the idea is to fit a model that contains only the variables with larger $\hat{\beta}$ and see if it's different from the previously fitted *best_model_o*.

```
ols_t=lm(log_pts~oreb_pct+log_usg_pct+log_ts_pct,data = NBA10_o)
anova(ols_t,best_model_o)
```

```
## Analysis of Variance Table
##
## Model 1: log_pts ~ oreb_pct + log_usg_pct + log_ts_pct
## Model 2: log_pts ~ log_reb + oreb_pct + log_dreb_pct + log_usg_pct + log_ts_pct
##   Res.Df     RSS Df Sum of Sq     F    Pr(>F)
```

```
## 1   1417 260.017
## 2   1415   8.747  2    251.27 20324 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value of the ANOVA test is almost equal to 0: this suggest that there is evidence against the null hypothesis (according to which the full and nested model are equivalent) and so we accept the alternative hypothesis. Therefore, the model with all the covariates is better than the nested one.

12) Goodness of fit

An indication of the goodness of fit of the model is offered by the adjusted R^2, the ratio between the variablity explained by the regression model and the total variability of the response.

```
summ_best_o$sigma
```

```
## [1] 0.07862288
```

```
summ_best_o$adj.r.squared
```

```
## [1] 0.9887274
```

The 98.87% of the variability of $log(pts)$ is explained by the best model. The standard deviation of the model is 0.078, therefore every prediction on the response should be considered in the range ± 0.078.

13) Prediction

new_obs contains information regarding a player which was not included in the dataset NBA used to fit the model: one can predict the average number of points of this player using the estimated coefficients of best_model_o and the values of the predictors from new_obs. The fit value is the log(points) estimated by the model, and the other two values represent the lower and the upper extreme of the 95% confidence interval.

```
new_obs=data.frame("log_reb"=1.1234,"oreb_pct"=0.064,"log_dreb_pct"=-1.221,
                   "log_usg_pct"=-1.478,"log_ts_pct"=-0.582)
predict(best_model_o,newdata = new_obs,level = 0.95,interval = "prediction")
```

```
##        fit      lwr      upr
## 1 1.487418 1.332473 1.642362
```

The predicted average number of points is 4.425654 and the prediction interval is [3.790405, 5.16736].

14) Simulation of data points

```
n_sim=167
set.seed(8)
beta=coefficients(best_model_o)
X=model.matrix(best_model_o)
y=X %*% beta + rnorm(n_sim,0,sigma(best_model_o))
```

The vector y contains 167 fitted values of log(pts), predicted using the model *best_model_o*. To check how far are the predictions made by the model from the actual data points contained in the dataset, the mean square error can be computed as follows:

```
MSE=mean((NBA10_o$log_pts-y)^2)
MSE
```

```
## [1] 0.01280108
```

MSE offers a measure of the average squared difference between the actual and simulated data points: in this case the model provides a relatively good fit to the data, because, on average, the difference between actual and simulated points is relatively small.