Foundations of Probability and Statistics project

Fabrizio D'Intinosante, Massimiliano Perletti

Contents

Introduction

In order to try to determine the relation between the footballer's performance and the price at which they where sold we scraped two dataframes:

- From Transfer Market we obtain the one containing the information about the selling price for each football player.
- From Who Scored we obtain the one containing the players' perfomance, in the year preceding the market operation.

Import packages

```
library(readxl)
library(dplyr)
library(gsubfn)
library(NLP)
library(pander)
library(ggplot2)
library(GGally)
library(ggthemes)
library(nortest)
library(EnvStats)
library(coefplot)
library(forestmodel)
library(lsmeans)
library(knitr)
library(kableExtra)
```

Instance matching

In order to obtain a unique, large dataset, we need to apply an instance matching procedure so we can make the analysis.

Import datasets

First of all we start importing the singles datasets and giving them a first look

```
transfer <- read_excel("transfer_serie_A.xlsx")
scored <- read_excel("TransferMarket_WhoScored_Data_Seria_A_v1.xlsx")
pander(head(transfer), caption = "Transfer Market")
```

Table 1: Transfer Market (continued below)

type	name	role	age	season	nation
Cessione	Lucas Castro	Centrale	29	18/19"	Argentina
Cessione	Samuel Bastien	Centrale	21	18/19"	Belgio
Cessione	Dario Dainelli	Difensore centrale	39	18/19"	Italia
Cessione	Radoslav Kirilov	Ala sinistra	26	18/19"	Bulgaria
Cessione	Massimo Gobbi	Terzino sinistro	37	18/19"	Italia
Cessione	Alessio Sestu	Ala destra	34	18/19"	Italia

from	to	market_value	value
AC Chievo Verona	Cagliari	7,00 mln	7,00 mln 6,50 mln
AC Chievo Verona	Standard Liegi	2,50 mln	2,50 mln 3,00 mln
AC Chievo Verona	Livorno	300 mila	300 mila gratuito
AC Chievo Verona	Vis Pesaro	300 mila	300 mila gratuito
AC Chievo Verona	Parma	300 mila	300 mila gratuito
AC Chievo Verona	Piacenza	150 mila	150 mila gratuito

```
pander(head(scored), caption = "Who Scored")
```

Table 3: Who Scored (continued below)

_id	aerialWonPerGame	age	apps	assistTotal	firstName	goal
114863	0.2121	24	33	5	Paulo	22
85077	0.4118	25	34	14	Luis Alberto	11
22546	0.0303	30	33	10	Alejandro	6
12267	1.543	32	35	8	Aleksandar	2
83390	0.08108	27	37	11	Lorenzo	8
100995	1.769	27	26	4	Alex Sandro	4

height	isActive	isManOfTheMatch	isOpta	lastName
177	true	false	true	Dybala
182	true	false	true	Romero Alconchel
165	true	false	true	GÃ³mez
187	true	false	true	Kolarov
163	true	false	true	Insigne
180	true	false	true	Lobo Silva

manOfTheMatch	minsPlayed	name	passSuccess
8	2358	Paulo Dybala	87.33
5	2677	Luis Alberto	79.9
6	2758	Alejandro GÃ³mez	81.91
6	3061	Aleksandar Kolarov	81.19
7	3104	Lorenzo Insigne	85.16
3	2115	Alex Sandro	86.36

playedPositions	playedPositionsShort	playerId	positionText	ranking
-AMC-FW-	AM(C),FW	114863	Forward	1
-AMC-AML-FW-	AM(CL),FW	85077	Midfielder	2
-FW-MC-ML-	M(CL),FW	22546	Midfielder	3
-DC-DL-ML-	D(CL),M(L)	12267	Defender	4
-AMC-AML-FW-	AM(CL),FW	83390	Forward	5
-DL-ML-	D(L),M(L)	100995	Defender	6

rating	redCard	regionCode	seasonId	seasonName	shotsPerGame	subOn
7.767	0	ar	6974	2017/2018	3.455	7
7.692	0	es	6974	2017/2018	2.382	2
7.649	0	ar	6974	2017/2018	2.97	2
7.516	0	rs	6974	2017/2018	1.8	1
7.511	0	it	6974	2017/2018	4.784	1
7.467	0	br	6974	2017/2018	0.6923	3

teamId	teamName	teamRegionName	tournamentId	tournamentName
87	Juventus	Italy	5	Serie A
77	Lazio	Italy	5	Serie A
300	Atalanta	Italy	5	Serie A
84	Roma	Italy	5	Serie A
276	Napoli	Italy	5	Serie A
87	Juventus	Italy	5	Serie A

tournamentRegionCode	tournamentRegionId	tournamentRegionName
it	108	Italy
it	108	Italy
it	108	Italy
it	108	Italy
it	108	Italy
it	108	Italy

tournamentShortName	weight	yellowCard
ISA	75	0
ISA	70	5
ISA	68	2
ISA	81	3
ISA	59	4
ISA	80	8

First data cleaning operation

Now we proceed to eliminate duplicates, to add a new names column to work with, and to clean this one from white spaces, accents ect.

```
#for transfer
transfer1 <- distinct(transfer,transfer$name,transfer$role,
                      transfer$age,transfer$nation, .keep_all=T)
transfer1$`transfer$name`<-NULL
transfer1$`transfer$role`<-NULL
transfer1$`transfer$age`<-NULL
transfer1$`transfer$nation`<-NULL
transfer1$newname <- transfer1$name
transfer1$newname <- tolower(transfer1$name)
transfer1$newname<-gsub(" ","",transfer1$newname)
transfer1$newname<-iconv(transfer1$newname,
```

```
                         from = 'UTF-8', to = 'ASCII//TRANSLIT')

#for who scored
scored1 <- scored
scored1$newname <- scored1$name
scored1$newname <- tolower(scored1$newname)
scored1$newname<-gsub(" ","",scored1$newname)
scored1$newname<-iconv(scored1$newname,
                       from = 'UTF-8', to = 'ASCII//TRANSLIT')
```

Algorithm application

Now that we have two quite cleaned dataset we use a partial matching algorithm to merge the two datasets that shows differences in the players' names encoding.

```
signature=function(x){
  sig=paste(sort(unlist(strsplit(tolower(x)," "))),collapse='')
  return(sig)
}

partialMatch=function(x,y,levDist=0.1){
  xx=data.frame(sig=sapply(x, signature),row.names=NULL)
  yy=data.frame(sig=sapply(y, signature),row.names=NULL)
  xx$raw=x
  yy$raw=y
  xx=subset(xx,subset=(sig!=''))
  xy=merge(xx,yy,by='sig',all=T)
  matched=subset(xy,subset=(!(is.na(raw.x)) & !(is.na(raw.y))))
  matched$pass="Duplicate"
  todo=subset(xy,subset=(is.na(raw.y)),select=c(sig,raw.x))
  colnames(todo)=c('sig','raw')
  todo$partials= as.character(sapply(todo$sig, agrep, yy$sig,
                                     max.distance = levDist,value=T))
  todo=merge(todo,yy,by.x='partials',by.y='sig')
  partial.matched=subset(todo,subset=(!(is.na(raw.x)) & !(is.na(raw.y))),
                         select=c("sig","raw.x","raw.y"))
  partial.matched$pass="Partial"
  matched=rbind(matched,partial.matched)
  un.matched=subset(todo,subset=(is.na(raw.x)),
                    select=c("sig","raw.x","raw.y"))
  if (nrow(un.matched)>0){
    un.matched$pass="Unmatched"
    matched=rbind(matched,un.matched)
  }
  matched=subset(matched,select=c("raw.x","raw.y","pass"))

  return(matched)
}
matches = partialMatch(scored1$name,transfer1$name)
a = scored1
b = transfer1
matched2 = merge(a,matches,by.x='name',by.y='raw.x',all.x=T)
matched2 = merge(matched2,b,by.x='raw.y',by.y='name',all.x=T)
matched2 <- na.omit(matched2)
```

```
matched2 <- matched2 %>% distinct(name,age.x,role, .keep_all = T)
pander(data.frame(dim (matched2), row.names = c("N of players", "N of columns")),
       caption = "Dimesion of merged dataset")
```

Table 11: Dimesion of merged dataset

	dim.matched2.
N of players	192
N of columns	52

With this partial matching procedure we obtained the complete dataset with the players' perfomance at the $Year_{t-1}$ and the price at which they where sold at the $Year_t$. Now we can proceed to apply some procedures to preprocess the data.

Preprocessing

During the scraping procedure some elements of the tables are positioned incorrectly into the the column, in order to obtain well formed data we need to apply some transformations to our merged dataset.

```
data <- matched2
data$market_value<-gsub(" ","",data$market_value)
data$market_value<-gsub("mln","0000",data$market_value)
data$market_value<-gsub("mila","000",data$market_value)
data$market_value<-gsub(",","",data$market_value)
data$market_value <- as.numeric(data$market_value)
data$value <- gsub(".*(gratuito)","Vendita secca",data$value)
data$value <- gsub(".*(Fine prestito).*","\\1",data$value)
data$value <- gsub(".*(Prestito)","\\1",data$value)
data$value <- gsub(".*mln.*mln","Diritto di riscatto",data$value)
data$value <- gsub(".*mln.*mila","Diritto di riscatto",data$value)
data$value <- gsub(".*-","Svincolato o ritirato",data$value)
data$value <- gsub(".*mln.*","Sconosciuto",data$value)
data$value <- gsub(".*mila.*","Sconosciuto",data$value)
data$value <- as.character(data$value)
data <- data[data$value != "Svincolato o ritirato",]
data <- data[data$value != "Sconosciuto",]
data$value <-gsub("Fine prestito", "Prestito", data$value)
data$value <- as.factor(data$value)
```

Now that we have cleaned up our data we can remove useless variables as *firstName* and *lastName* because we already have the *name* variable that includes the other two and, as this one, others.

The new datasets appears like

```
pander(head(data), caption = "Final dataset")
```

Table 12: Final dataset (continued below)

name	aerialWonPerGame	age.x	apps	assistTotal	goal
Adam Masina	3.382	24	34	1	0
Adel Taarabt	0.2273	29	22	2	2
Adem Ljajic	0.03704	26	27	10	6
Afriyie Acquah	0.5909	26	22	0	1

name	aerialWonPerGame	age.x	apps	assistTotal	goal
Albano Bizzarri	0.25	40	32	0	0
Alberto Brignoli	0.7692	27	13	0	1

height	isActive	isManOfTheMatch	isOpta	manOfTheMatch	minsPlayed
189	false	false	true	1	2841
178	true	false	true	1	1619
182	false	false	true	4	2155
179	false	false	true	0	936
193	false	false	true	0	2880
188	false	false	true	0	1124

passSuccess	positionText	ranking	rating	redCard	shotsPerGame	subOn
75.57	Defender	101	6.865	1	0.5	1
80.53	Midfielder	148	6.76	1	1.5	4
82.08	Forward	24	7.209	0	2.259	4
87.28	Midfielder	279	6.224	1	0.4091	13
60.95	Goalkeeper	215	6.592	1	0	0
56.42	Goalkeeper	337	6.512	0	0.07692	2

teamName	weight	yellowCard	type	role	nation
Bologna	78	8	Cessione	Terzino sinistro	Italia
Genoa	77	5	Cessione	Trequartista	Marocco
Torino	74	2	Cessione	Trequartista	Serbia
Torino	70	3	Acquisto	Centrale	Ghana
Udinese	89	0	Cessione	Portiere	Argentina
Benevento	79	2	Acquisto	Portiere	Italia

from	to	market_value	value
Bologna FC 1909	Watford	7e+06	Diritto di riscatto
Genoa CFC	Benfica	1500000	Prestito
Torino FC	Besiktas	1.3e+07	Diritto di riscatto
Torino	Empoli FC	2e+06	Diritto di riscatto
Udinese Calcio	Foggia	2e+05	Vendita secca
Benevento	Juventus FC	1e+06	Prestito

With dimensions:

```
pander(data.frame(dim (data), row.names = c("N of players", "N of columns")),
       caption = "Dimesion final dataset")
```

Table 17: Dimesion final dataset

	dim.data.
N of players	178

	dim.data.
N of columns	29

Low level analysis

Descriptive

In first place we can create a correlation matrix that compute the value of correlation between every numeric variable.

```
numeric_var <- names(data[,c(2,3,4,5,6,7,11,12,13,15,16,18,19,21,22,28)])
pander(cor(data[, numeric_var]), big.mark = ",",
       caption = "Correlation between numeric variables")
```

Table 18: Correlation between numeric variables (continued below)

	aerialWonPerGame	age.x	apps	assistTotal
aerialWonPerGame	1	0.09253	0.1986	-0.2046
age.x	0.09253	1	0.04933	-0.02283
apps	0.1986	0.04933	1	0.3682
assistTotal	-0.2046	-0.02283	0.3682	1
goal	0.1781	-0.07725	0.455	0.4931
height	0.2959	0.1389	0.012	-0.3383
manOfTheMatch	0.1664	0.02042	0.455	0.518
minsPlayed	0.2558	0.1208	0.9218	0.2981
passSuccess	-0.07444	0.03045	-0.1348	0.03813
ranking	-0.2908	-0.09755	-0.5795	-0.3777
rating	0.2313	0.1004	0.4012	0.4238
shotsPerGame	0.1641	-0.2189	0.3846	0.5177
subOn	-0.1654	-0.1969	0.064	0.1024
weight	0.18	0.2683	0.02202	-0.2726
yellowCard	0.378	0.05939	0.5458	0.08808
market__value	-0.07794	-0.1203	0.4248	0.4756

	goal	height	manOfTheMatch	minsPlayed
aerialWonPerGame	0.1781	0.2959	0.1664	0.2558
age.x	-0.07725	0.1389	0.02042	0.1208
apps	0.455	0.012	0.455	0.9218
assistTotal	0.4931	-0.3383	0.518	0.2981
goal	1	-0.1239	0.5706	0.3612
height	-0.1239	1	-0.04955	0.09304
manOfTheMatch	0.5706	-0.04955	1	0.4725
minsPlayed	0.3612	0.09304	0.4725	1
passSuccess	-0.1082	-0.287	-0.108	-0.1322
ranking	-0.3506	-0.02468	-0.5214	-0.6444
rating	0.3263	0.002975	0.5482	0.5112
shotsPerGame	0.7487	-0.2159	0.4677	0.2645
subOn	0.1449	-0.167	-0.1009	-0.3179
weight	-0.07608	0.7635	0.003242	0.09838
yellowCard	0.1332	-0.03581	0.1735	0.5527
market__value	0.4673	-0.1061	0.4005	0.4047

	goal	height	manOfTheMatch	minsPlayed

	passSuccess	ranking	rating	shotsPerGame
aerialWonPerGame	-0.07444	-0.2908	0.2313	0.1641
age.x	0.03045	-0.09755	0.1004	-0.2189
apps	-0.1348	-0.5795	0.4012	0.3846
assistTotal	0.03813	-0.3777	0.4238	0.5177
goal	-0.1082	-0.3506	0.3263	0.7487
height	-0.287	-0.02468	0.002975	-0.2159
manOfTheMatch	-0.108	-0.5214	0.5482	0.4677
minsPlayed	-0.1322	-0.6444	0.5112	0.2645
passSuccess	1	-0.04506	0.1025	-0.08644
ranking	-0.04506	1	-0.9343	-0.2801
rating	0.1025	-0.9343	1	0.2358
shotsPerGame	-0.08644	-0.2801	0.2358	1
subOn	-0.005746	0.2428	-0.3307	0.2203
weight	-0.2245	-0.04118	0.04743	-0.1985
yellowCard	0.0517	-0.3379	0.2207	0.1949
market_value	0.1602	-0.4639	0.504	0.316

	subOn	weight	yellowCard	market_value
aerialWonPerGame	-0.1654	0.18	0.378	-0.07794
age.x	-0.1969	0.2683	0.05939	-0.1203
apps	0.064	0.02202	0.5458	0.4248
assistTotal	0.1024	-0.2726	0.08808	0.4756
goal	0.1449	-0.07608	0.1332	0.4673
height	-0.167	0.7635	-0.03581	-0.1061
manOfTheMatch	-0.1009	0.003242	0.1735	0.4005
minsPlayed	-0.3179	0.09838	0.5527	0.4047
passSuccess	-0.005746	-0.2245	0.0517	0.1602
ranking	0.2428	-0.04118	-0.3379	-0.4639
rating	-0.3307	0.04743	0.2207	0.504
shotsPerGame	0.2203	-0.1985	0.1949	0.316
subOn	1	-0.1695	-0.1176	-0.004954
weight	-0.1695	1	-0.08951	-0.06094
yellowCard	-0.1176	-0.08951	1	0.07447
market_value	-0.004954	-0.06094	0.07447	1

Now we can inspect also a variables' summary

```
pander(summary(data[,numeric_var]), big.mark = ",",
       caption = "Summary of numeric variables")
```

Table 22: Summary of numeric variables (continued below)

aerialWonPerGame	age.x	apps	assistTotal
Min. :0.0000	Min. :18.00	Min. : 1.00	Min. : 0.00
1st Qu.:0.3489	1st Qu.:24.00	1st Qu.:12.25	1st Qu.: 0.00
Median :0.7500	Median :26.00	Median :21.00	Median : 0.00

aerialWonPerGame	age.x	apps	assistTotal
Mean :1.0693	Mean :26.89	Mean :20.37	Mean : 1.18
3rd Qu.:1.5657	3rd Qu.:29.75	3rd Qu.:29.00	3rd Qu.: 2.00
Max. :6.2121	Max. :40.00	Max. :38.00	Max. :12.00

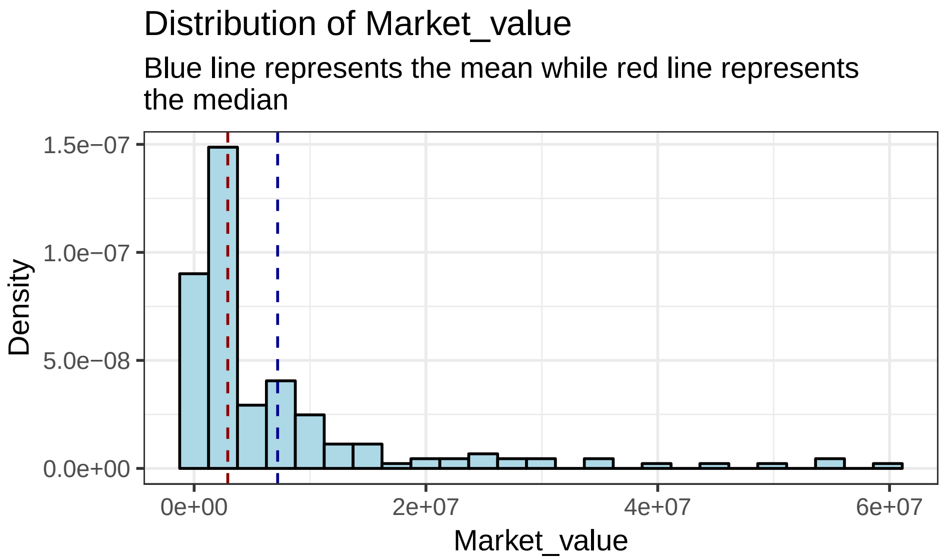
goal	height	manOfTheMatch	minsPlayed
Min. : 0.000	Min. :167.0	Min. :0.0000	Min. : 9.0
1st Qu.: 0.000	1st Qu.:180.0	1st Qu.:0.0000	1st Qu.: 626.2
Median : 0.500	Median :184.0	Median :0.0000	Median :1191.0
Mean : 1.567	Mean :183.5	Mean :0.6854	Mean :1438.7
3rd Qu.: 2.000	3rd Qu.:188.0	3rd Qu.:1.0000	3rd Qu.:2242.0
Max. :16.000	Max. :196.0	Max. :6.0000	Max. :3420.0

passSuccess	ranking	rating	shotsPerGame
Min. : 49.02	Min. : 5.0	Min. :5.730	Min. :0.0000
1st Qu.: 74.19	1st Qu.:109.0	1st Qu.:6.424	1st Qu.:0.1557
Median : 80.03	Median :206.5	Median :6.661	Median :0.5076
Mean : 78.54	Mean :226.6	Mean :6.646	Mean :0.7213
3rd Qu.: 83.89	3rd Qu.:308.8	3rd Qu.:6.858	3rd Qu.:1.0446
Max. :100.00	Max. :550.0	Max. :7.630	Max. :2.9167

subOn	weight	yellowCard	market_value
Min. : 0.000	Min. :60.00	Min. : 0.000	Min. : 150000
1st Qu.: 1.000	1st Qu.:72.00	1st Qu.: 1.000	1st Qu.: 1500000
Median : 3.000	Median :76.50	Median : 2.000	Median : 2900000
Mean : 4.376	Mean :76.63	Mean : 2.854	Mean : 7204213
3rd Qu.: 6.000	3rd Qu.:80.00	3rd Qu.: 4.000	3rd Qu.: 8000000
Max. :25.000	Max. :92.00	Max. :12.000	Max. :60000000

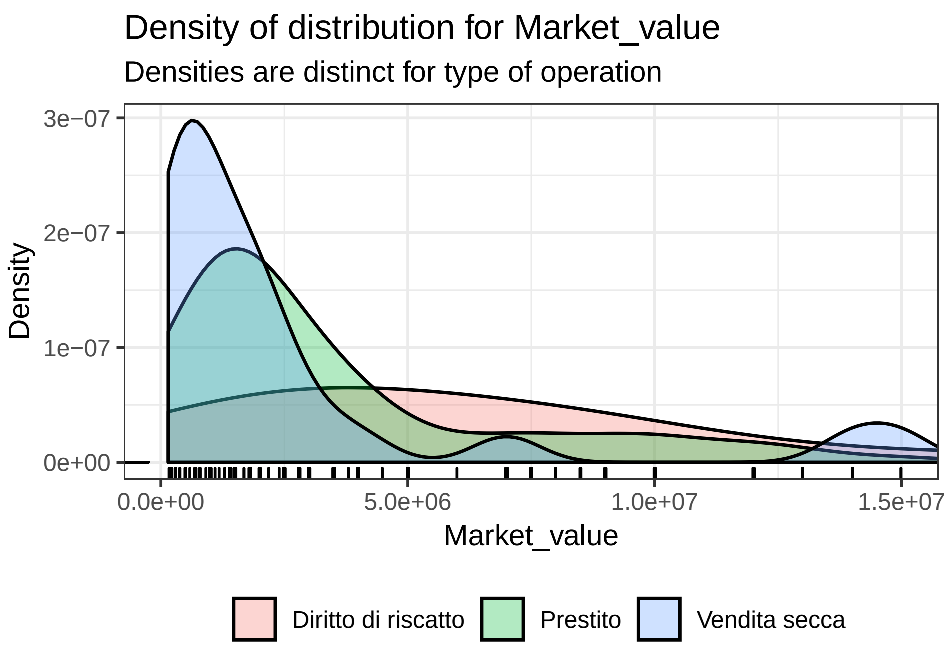
Some plot

We need to take a first look to the distribution of *market_value* variable that represent the price at which the football's players were sold



We can observe that there is a strong positive skewness because the meadian is lower that the mean, this is due to the presence of much more expensive market operations than the average. This means that our Y variable doesn't present a **normal distribution**.

We can focus on our Y distribution but conditionally to the type of operation the players were involved to.



Tests

We know from Transfer Market that the mean of total market operation for the season 2017/2018 is equal to **1.090.607 euro**

```
market_mean = 1090607
pander(t.test(data$market_value, mu = market_mean))
```

Table 26: One Sample t-test: `data$market_value`

Test statistic	df	P value	Alternative hypothesis	mean of x
7.507	177	2.857e-12 * * *	two.sided	7204213

From this test we learn that the mean of the operation of our dataset is significantly different from the mean we found on Transfer Market for the previous year. WE already seen graphically that the distribution of our Y variable isn't normal but we can also use some test to verify this.

```
pander(ad.test(data$market_value))
```

Table 27: Anderson-Darling normality test: `data$market_value`

Test statistic	P value
22.79	3.7e-24 * * *

```
pander(shapiro.test(data$market))
```

Table 28: Shapiro-Wilk normality test: `data$market`

Test statistic	P value
0.6206	1.13e-19 * * *

```
pander(wilcox.test(data$market_value, conf.int = TRUE, mu = market_mean))
```

Table 29: Wilcoxon signed rank test with continuity correction: `data$market_value` Our test confirms that *market_value* isn't normally distributed. We can check also the association between the players' market_price and the operation with they have been bought

Test statistic	P value	Alternative hypothesis	(pseudo)median
14790	3.562e-23 * * *	two.sided	4500000

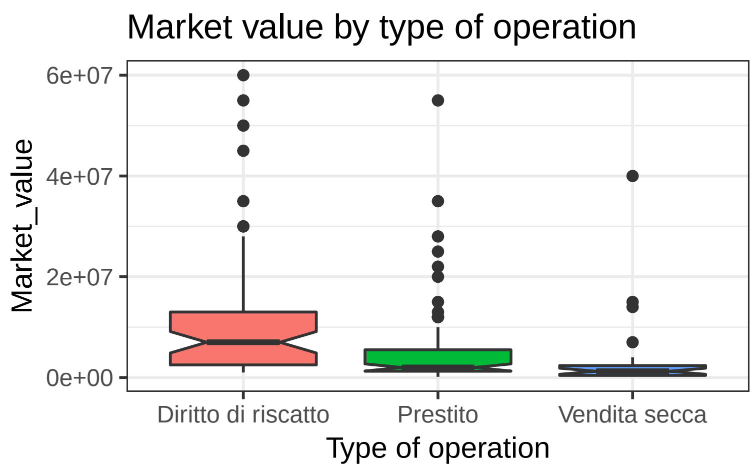
```
a.table <- table(data$market_value,data$value)
chi.a = chisq.test(a.table)
pander(chi.a)
```

Table 30: Pearson's Chi-squared test: `a.table`

Test statistic	df	P value
136.6	102	0.01253 *

```
chi.norm.a = chi.a$statistic/(nrow(data)*min(nrow(a.table)-1,ncol(a.table)-1))
pander(chi.norm.a)
```

X-squared
0.3838



ANOVA models

In order to try to explain our Y variable *market_value* we try to use some ANOVA models.

- One with *value* as explaining variable
- One with the players' role

```
lm_value <- lm(market_value ~ value, data = data)
pander(drop1(lm_value, test = 'F'))
```

Table 32: Single term deletions

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
	NA	NA	1.92e+16	5758	NA	NA
value	2	1.7e+15	2.09e+16	5769	7.748	0.000597

```
pander(summary(lm_value))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	11437705	1341061	8.529	6.779e-15
valuePrestito	-6141551	1733204	-3.543	0.0005065
valueVendita secca	-7487705	2453136	-3.052	0.002625

Table 34: Fitting linear model: market_value ~ value

Observations	Residual Std. Error	R^2	Adjusted R^2
178	10474025	0.08134	0.07084

```r
pander(anova(lm_value, test = 'F'))
```

Table 35: Analysis of Variance Table *value* results meaningful with F-value and t-value. This simple model explain 0.07 of the explicative variable.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
value	2	1.7e+15	8.5e+14	7.748	0.000597
Residuals	175	1.92e+16	1.097e+14	NA	NA

```r
lm_role <- lm(market_value ~ positionText, data = data)
pander(drop1(lm_role, test = 'F'))
```

Table 36: Single term deletions

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
	NA	NA	2.045e+16	5771	NA	NA
positionText	3	4.467e+14	2.09e+16	5769	1.267	0.2874

On the other side position text is not able to explain our explicative variable, so we don't proceed with this model.

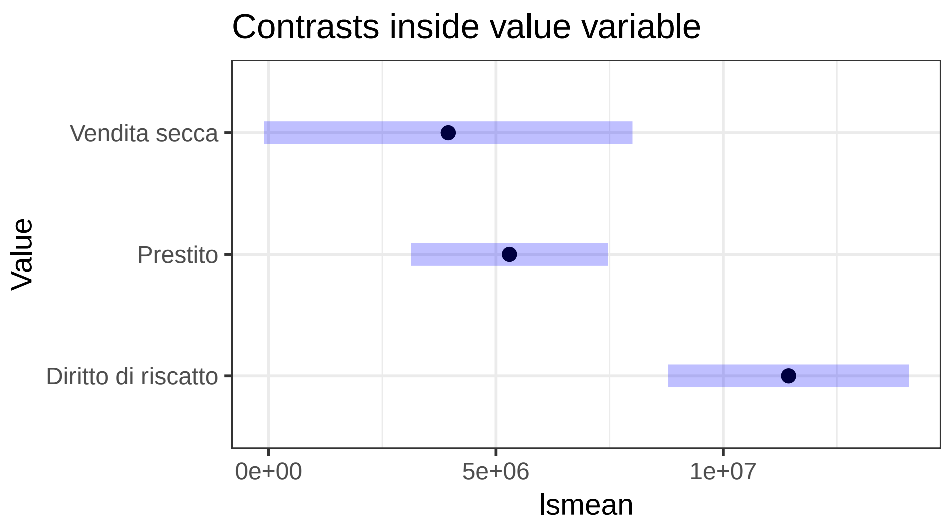
```r
library(lsmeans)
ls_value = lsmeans(lm_value,
                   pairwise ~ value,
                   adjust = 'tukey')
kable(ls_value$contrasts, format = "latex", align = "c") %>%
  kable_styling(position = "center")
```

contrast	estimate	SE	df	t.ratio	p.value
Diritto di riscatto - Prestito	6141551	1733204	175	3.543466	0.0014635
Diritto di riscatto - Vendita secca	7487705	2453136	175	3.052299	0.0073683
Prestito - Vendita secca	1346154	2329159	175	0.577957	0.8320699

```r
kable(ls_value$lsmeans, format = "latex", align = "c") %>%
  kable_styling(position = "center")
```

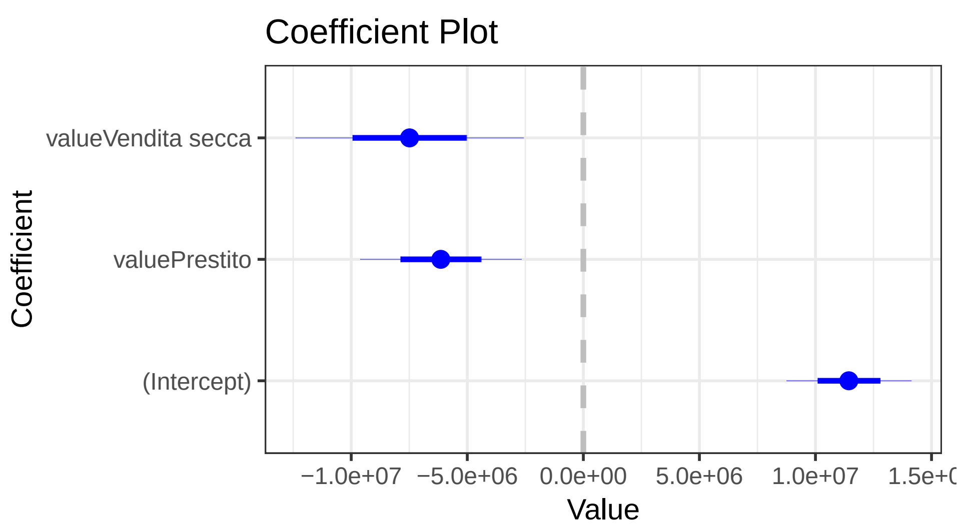
value	lsmean	SE	df	lower.CL	upper.CL
Diritto di riscatto	11437705	1341061	175	8790969.3	14084441
Prestito	5296154	1097976	175	3129174.4	7463133
Vendita secca	3950000	2054125	175	-104047.2	8004047

```r
plot(ls_value$lsmeans, alpha = .05) + theme_bw() + labs(y = "Value",
                                                        title = "Contrasts inside value variable")
```

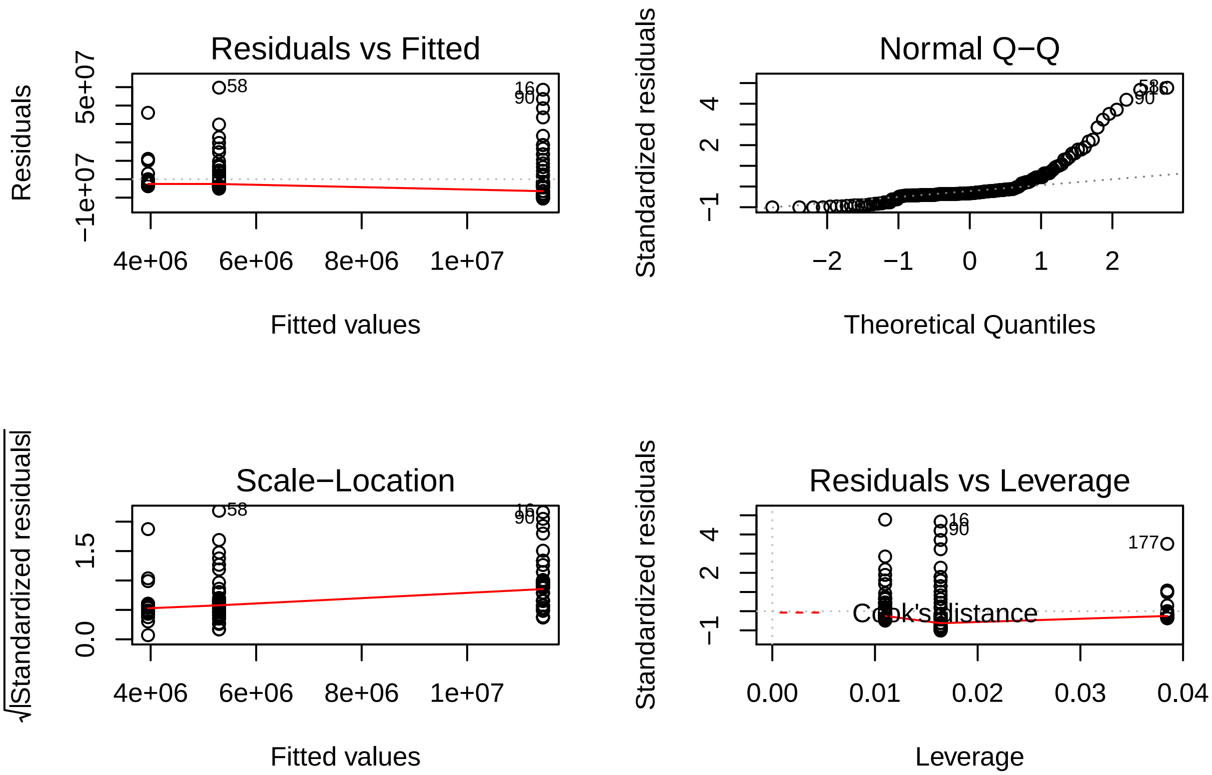


Altogether speaking *value* is meaningful but only *Diritto di riscatto* is significantly different from the other values into *value* variable.

```
coefplot(lm_value, intercept = TRUE) + theme_bw()
```




```
par(mfrow = c(2,2))
plot(lm_value)
```


Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Residuals vs Leverage
Cook's distance
Leverage