Statistical Analysis of NBA Players Salary

STATISTICAL LEARNING FINAL PROJECT (MOD B)

Mahir Selek - 2041295

Joi Berberi - 2033363

Academic Year 2021-2022

OUTLINE

1. Dataset Description
2. Cleaning and Filtering the Data
3. Exploratory Data Analysis
4. Model Data
5. Conclusion

Introduction

- The goal of this project is to predict the NBA players salary
- We used the Basketball Reference database and Kaggle Repository to obtain various features regarding the players from the NBA
- We wanted to do a forecasting project because it is a big league in terms of money and because the income salary of the players arouse curiosity every year

1. Dataset Description

1. Dataset Description

- We have 3 different datasets. The "NBA Players stats since 1950" dataset which is available on Kaggle.
- However "player_salary" dataset was not provided at Kaggle. So, We scraped from " https://www.basketball-reference.com/contracts/players.html " website and created by ourselves.
- We consider a third dataset for including the info of players about their ages, heights and weights.
- The first dataset contains aggregate individual statistics for 67 NBA seasons since 1950. From basic box-score attributes such as points, assists, rebounds etc., to more advanced money-ball like features such as Value Over Replacement.
- We obtained the data for 24691 players. For every player we obtained a set with a total of 51 features
- The second dataset only contains player names, their teams and their salaries.

[redacted]

Dataset Variables

Year	Player	Pos	Age	Tm	G	GS	MP	PER	TS%	3PAr	FTr	ORB%	DRB%	TRB%	AST%	STL%	BLK%	TOV%	USG%	blanl	OWS	DWS	WS
2017	Okaro Wh	PF	24	MIA	35	0	471	7.5	0.507	0.391	0.253	5.8	13.5	9.6	6	1.1	1.7	15.7	10.8		0.1	0.5	0.6
2017	Isaiah Wh	PG	21	BRK	73	26	1643	7.5	0.487	0.293	0.222	2.1	9.7	5.9	17.7	1.2	1.7	20.3	18.2		-1.7	0.9	-0.8
2017	Hassan Wl	C	27	MIA	77	77	2513	22.6	0.579	0	0.368	12.8	35.3	24	3.8	1.1	5	12	22.7		4.2	5.3	9.5
2017	Andrew W	SF	21	MIN	82	82	3048	16.5	0.534	0.184	0.345	3.9	8.8	6.3	10.6	1.4	0.8	9.4	29		3.3	0.9	4.2
2017	C.J. Wilco:	SG	26	ORL	22	0	108	2.9	0.329	0.484	0.065	3.9	8.3	6	15.5	0.9	0.7	15.8	15.4		-0.2	0	-0.2
2017	Alan Willi	C	24	PHO	47	0	708	19.5	0.547	0.004	0.419	14	31.2	22.4	5.2	1.8	3.7	10.5	20.9		1.1	0.9	2.1
2017	Deron Wil	PG	32	TOT	64	44	1657	14	0.541	0.39	0.182	0.9	9.4	5.1	35.9	1	0.4	17.6	22.1		1.5	0.9	2.4
2017	Deron Wil	PG	32	DAL	40	40	1171	15	0.533	0.4	0.185	1.2	9.3	5.1	40.1	1.1	0.2	16.7	23.7		1.1	0.7	1.8
2017	Deron Wil	PG	32	CLE	24	4	486	11.4	0.566	0.361	0.17	0.2	9.7	5.1	25.9	0.6	1	20.2	18.1		0.4	0.2	0.6
2017	Derrick W	PF	25	TOT	50	11	804	10.6	0.537	0.398	0.365	2.6	15.1	8.9	5.1	0.9	0.7	9	17.2		0.4	0.6	1.1
2017	Derrick W	PF	25	MIA	25	11	377	10.1	0.465	0.328	0.365	4.7	16.9	10.7	5.6	1.2	1	8.1	20.4		-0.1	0.4	0.3
2017	Derrick W	PF	25	CLE	25	0	427	11.1	0.628	0.486	0.364	0.8	13.5	7.4	4.7	0.6	0.4	10.1	14.4		0.6	0.2	0.8
2017	Lou Willia	SG	30	TOT	81	1	1994	21.4	0.593	0.447	0.458	1.4	9.9	5.5	19.9	1.9	0.8	11.8	29.1		5.1	1	6.1
2017	Lou Willia	SG	30	LAL	58	1	1403	23.9	0.609	0.432	0.469	1.1	9.5	5.1	22.3	2.3	0.6	11.9	30.6		4.3	0.6	4.9
2017	Lou Willia	SG	30	HOU	23	0	591	15.4	0.547	0.489	0.428	2.2	10.7	6.5	14.3	1.2	1.2	11.3	25.3		0.8	0.4	1.2

WS/48	blank2	OBPM	DBPM	BPM	VORP	FG	FGA	FG%	3P	3PA	3P%	2P	2PA	2P%	eFG%	FT	FTA	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
0.066		-3.1	0.9	-2.1	0	33	87	0.379	12	34	0.353	21	53	0.396	0.448	20	22	0.909	25	57	82	21	10	10	18	52	98
-0.023		-4.3	-0.6	-4.9	-1.2	204	508	0.402	44	149	0.295	160	359	0.446	0.445	91	113	0.805	32	152	184	192	42	36	142	175	543
0.181		-2	1.5	-0.5	0.9	542	973	0.557	0	0		542	973	0.557	0.557	225	358	0.628	293	795	1088	57	56	161	154	226	1309
0.066		0.2	-2.9	-2.7	-0.6	709	1570	0.452	103	289	0.356	606	1281	0.473	0.484	412	542	0.76	103	226	329	189	82	30	187	183	1933
-0.09		-6.5	-2.2	-8.7	-0.2	8	31	0.258	3	15	0.2	5	16	0.313	0.306	2	2	1	4	8	12	12	2	1	6	8	21
0.142		-1.8	0.2	-1.7	0.1	138	267	0.517	0	1	0	138	266	0.519	0.517	70	112	0.625	94	198	292	23	27	32	37	125	346
0.069		0.2	-2.4	-2.3	-0.1	263	600	0.438	85	234	0.363	178	366	0.486	0.509	90	109	0.826	14	133	147	360	31	8	138	138	701
0.073		1	-2.4	-1.4	0.2	195	453	0.43	63	181	0.348	132	272	0.485	0.5	69	84	0.821	13	89	102	274	25	2	98	96	522
0.059		-1.9	-2.6	-4.5	-0.3	68	147	0.463	22	53	0.415	46	94	0.489	0.537	21	25	0.84	1	44	45	86	6	6	40	42	179
0.064		-2.4	-1.8	-4.2	-0.4	108	244	0.443	30	97	0.309	78	147	0.531	0.504	58	89	0.652	19	111	130	28	14	7	28	60	304
0.038		-3.8	-1.4	-5.2	-0.3	54	137	0.394	9	45	0.2	45	92	0.489	0.427	31	50	0.62	16	57	73	14	9	5	14	33	148
0.086		-1.1	-2.1	-3.2	-0.1	54	107	0.505	21	52	0.404	33	55	0.6	0.603	27	39	0.692	3	54	57	14	5	2	14	27	156
0.147		3.7	-3	0.8	1.4	428	998	0.429	163	446	0.365	265	552	0.48	0.511	402	457	0.88	26	176	202	239	80	19	160	92	1421
0.169		5.4	-3.2	2.2	1.5	326	734	0.444	122	317	0.385	204	417	0.489	0.527	304	344	0.884	14	118	132	183	65	10	120	67	1078
0.096		-0.1	-2.5	-2.6	-0.1	102	264	0.386	41	129	0.318	61	135	0.452	0.464	98	113	0.867	12	58	70	56	15	9	40	25	343

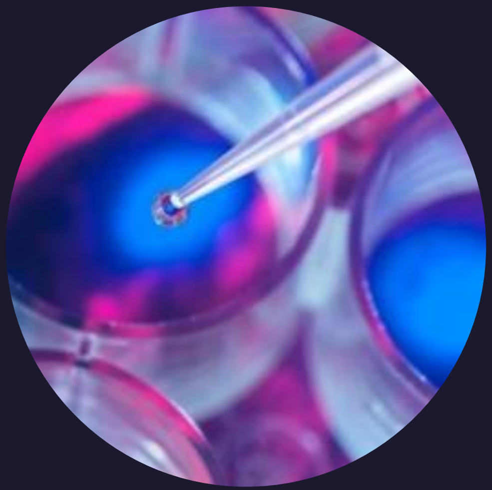
2. Cleaning and Filtering the Data

Player Features

- We obtained the data for 24691 players
- For every player we obtained a set with a total of 51 features
- The salary related features are 8 features (Age, Minutes, Points, Assist, Turnover, Block, Rebound, Steal)
- The others many of them Dummy Features for us to predict salary

Filtering

The advantage of the filtering after 2017 is that we don't have any NA or empty feature anymore. Thus, we have transformed our data into a more useful form. distinct functions help us to retain only unique/distinct rows from our input tables. Aim of the mutation is that in the seasons_stats file we don't have stats per game features. So, we mutated all of them to use in our salary prediction project. Our main purpose is to investigate how the stats effect next season's salary the players get.



Merging

Then we merged the two dataset that we have. After that we checked out new dataset and we decided to use only necessary features for our models. Our new dataset become clearer and more understandable. We prefer to use specific data belongs only on 2017. Also, we created new variables with mutate function to predict better and understandable data.

Cleaning and Filtering the Data

- After cleaning and filtering out big dataset now we have only salary related features

	Player	Year	Pos	Age	Tm	MPG	PPG	APG	RPG	TOPG	BPG	SPG	salary17_18	height	weight
	<chr>	<int>	<chr>	<int>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<int>
1	A.J. Hammons	2017	C	24	DAL	7.409091	2.1818182	0.18181818	1.6363636	0.4545455	0.59090909	0.04545455	1312611	198	99
2	Aaron Brooks	2017	PG	32	IND	13.753846	4.9538462	1.92307692	1.0615385	1.0153846	0.13846154	0.38461538	2116955	183	73
3	Aaron Gordon	2017	SF	21	ORL	28.725000	12.7375000	1.87500000	5.0625000	1.1125000	0.50000000	0.80000000	5504420	206	99
4	Al-Farouq Aminu	2017	SF	26	POR	29.065574	8.7213115	1.62295082	7.3934426	1.5409836	0.72131148	0.98360656	7319035	206	99
5	Al Horford	2017	C	30	BOS	32.250000	14.0000000	4.95588235	6.8235294	1.7058824	1.27941176	0.76470588	27734405	208	111
6	Al Jefferson	2017	C	32	IND	14.106061	8.1060606	0.86363636	4.2121212	0.5000000	0.24242424	0.28787879	9769821	208	131
7	Alan Williams	2017	C	24	PHO	15.063830	7.3617021	0.48936170	6.2127660	0.7872340	0.68085106	0.57446809	6000000	198	90
8	Alec Burks	2017	SG	25	UTA	15.547619	6.7380952	0.71428571	2.8571429	0.8333333	0.11904762	0.42857143	10845506	198	97
9	Alex Abrines	2017	SG	23	OKC	15.514706	5.9705882	0.58823529	1.2647059	0.4852941	0.11764706	0.54411765	5725000	198	86
10	Alex Len	2017	C	23	PHO	20.259740	7.9610390	0.57142857	6.6233766	1.3246753	1.27272727	0.48051948	4187599	216	117
11	Alex Poythress	2017	PF	23	PHI	26.166667	10.6666667	0.83333333	4.8333333	0.5000000	0.33333333	0.50000000	778668	201	107
12	Alexis Ajinca	2017	C	28	NOP	14.974359	5.3076923	0.30769231	4.5384615	0.7948718	0.56410256	0.51282051	4961798	218	112
13	Allen Crabbe	2017	SG	24	POR	28.531646	10.6962025	1.17721519	2.8481013	0.7848101	0.25316456	0.68354430	19332500	198	95

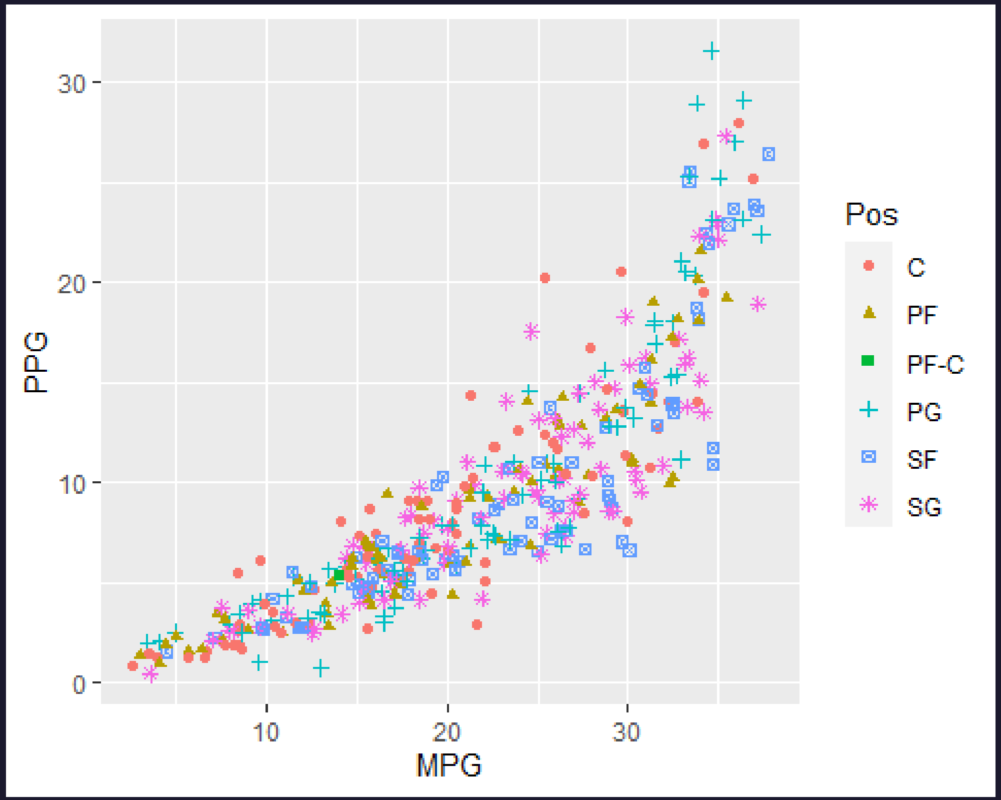
1-13 of 442 rows

Previous 1 2 3 4 5 6 ... 34 Next

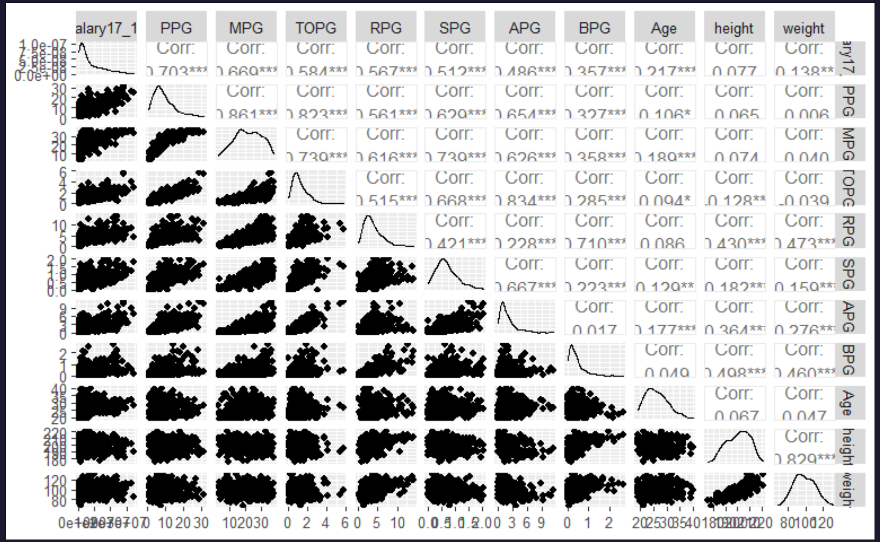
3. Exploratory Data Analysis

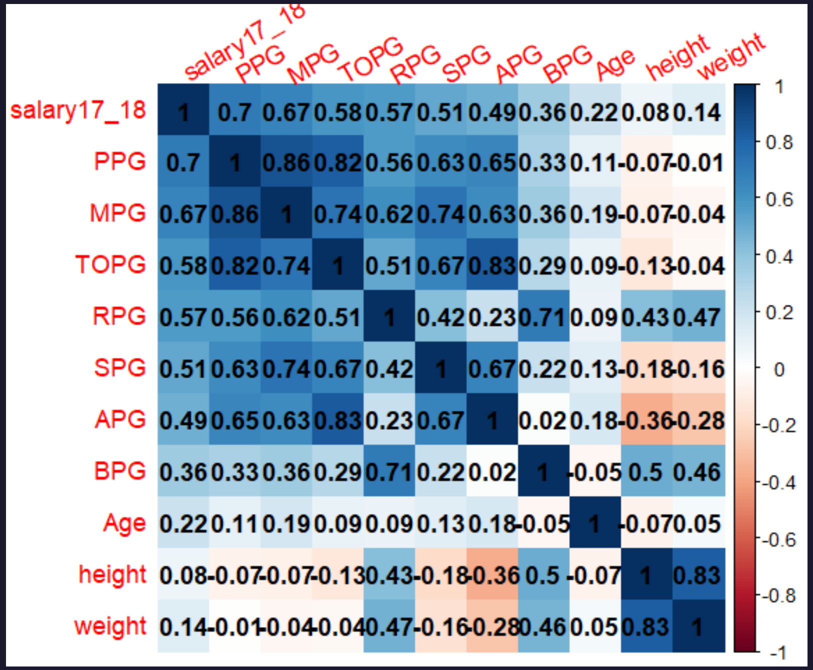
First of all before start in order to get an idea, we wanted to look at the distribution relationship of the numerical data we have with the positions of the players.

Then we did was to look at the distributions of the continuous variables conditioned on the stats salary



Correlation





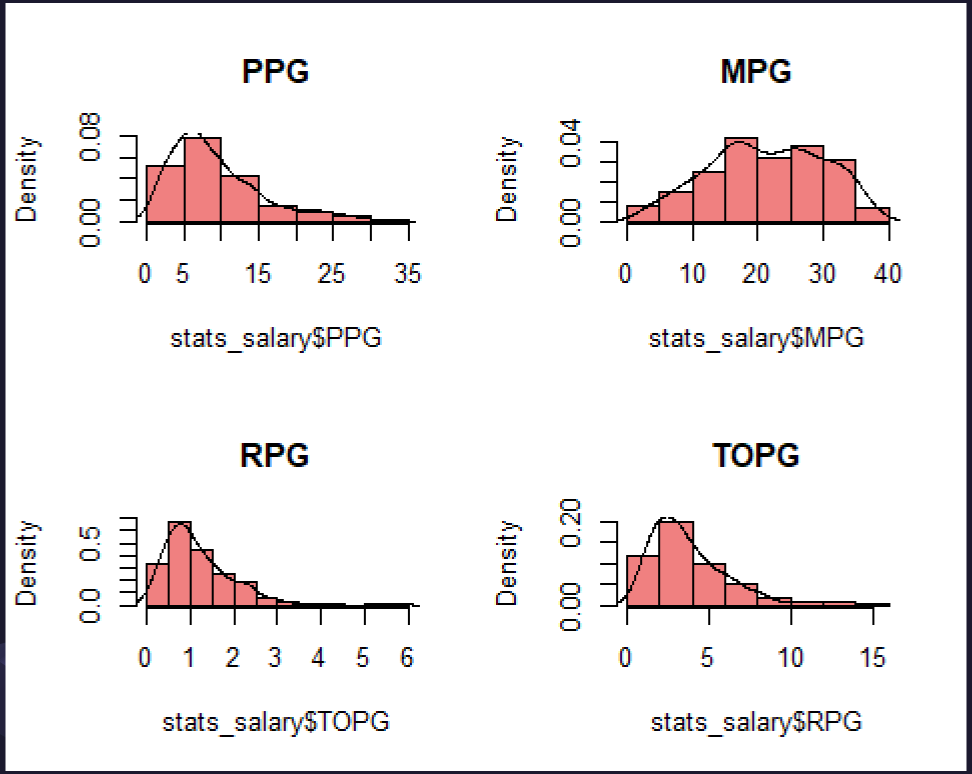
- We preferred to use correlation in order to look at the data we have from the outside. Being able to draw such a straight line helps us not only predict the unknown but also understand the relationship between the variables better
- Correlation strength: PPG > MPG > TOPG > RPG > PER > SPG > APG > Age > Weight > G
- The interesting part of this is that the number of turnover players make is linked to their salary, and the relationship has a positive correlation.

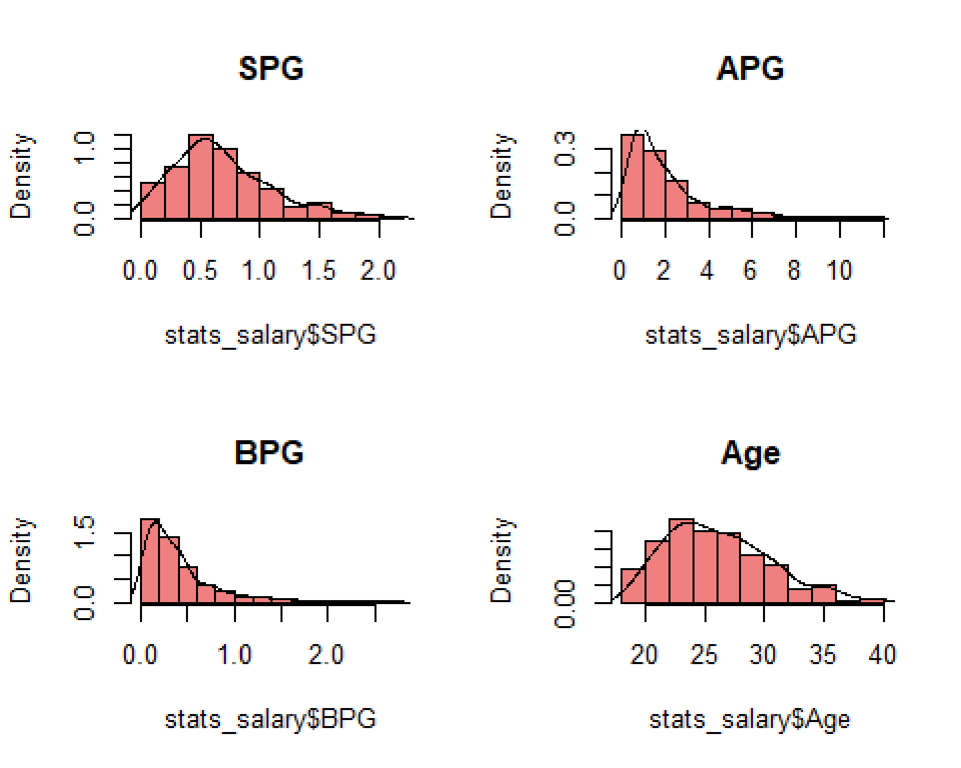
Data Visualization

Wednesday, July 20, 2022

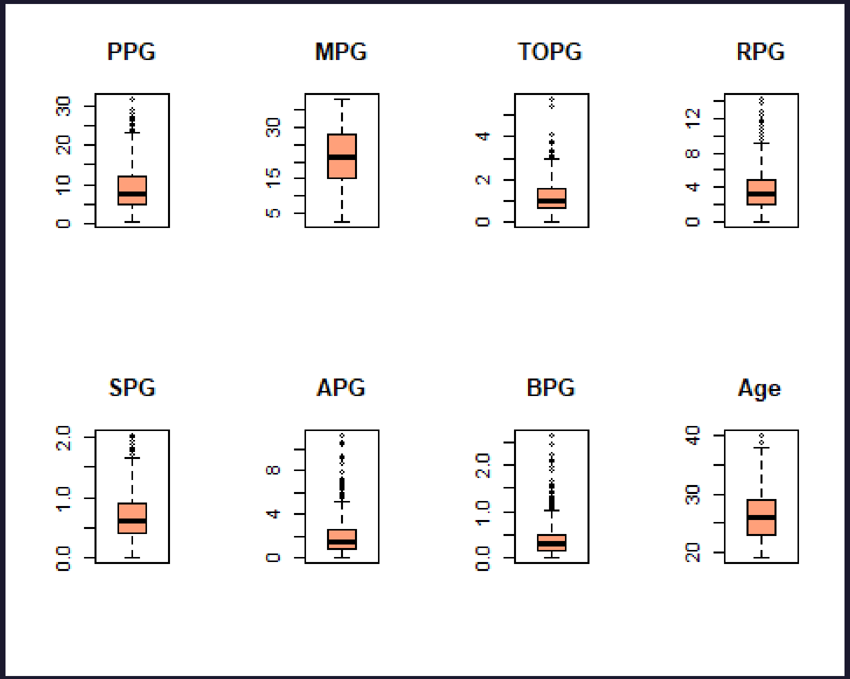
Data Distributions

- As expected, the minutes of the players are perfectly normal
- On the other hand, the other predictors are not perfectly normal

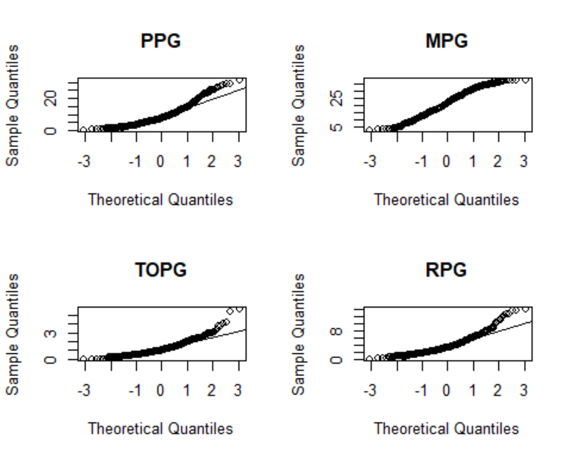


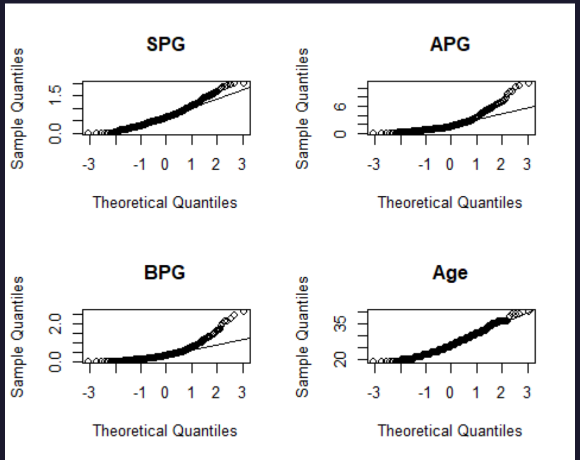


Data Distributions



We look to the features independently from the boxplot it can be noticed that the most frequent features are MPG and Age, as expected APG and BPG represent a minority.





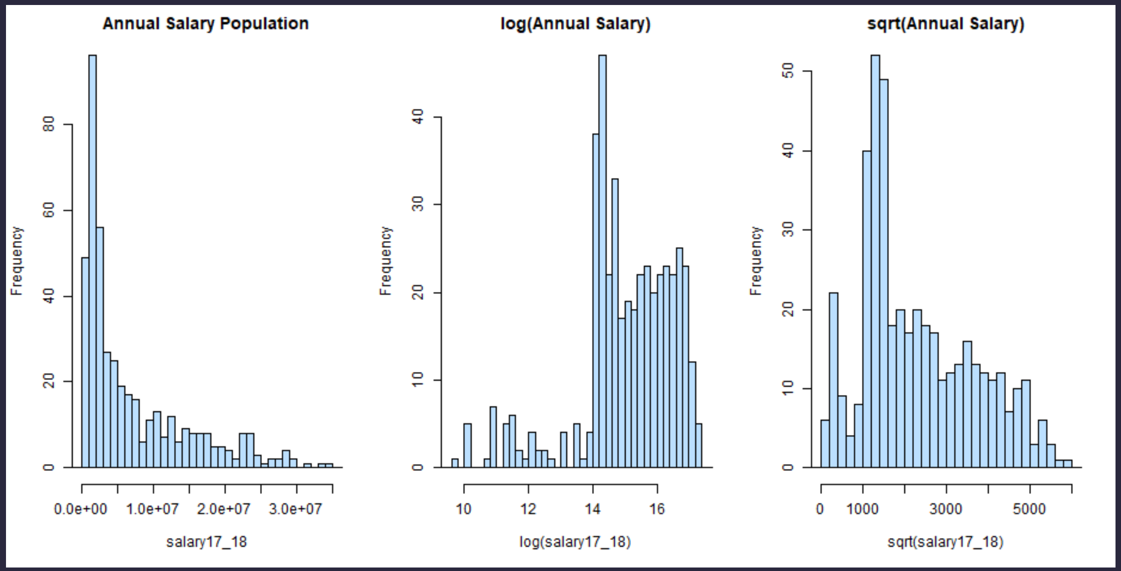
4. Model Data

Model Data: Multiple Linear Regression

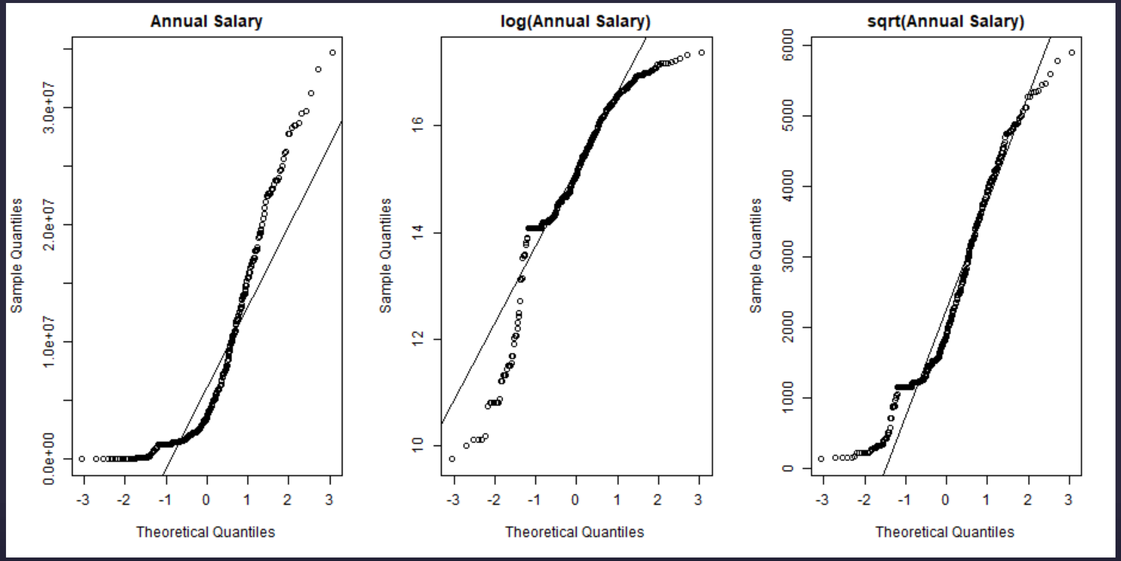
- For almost the entire project, we decided to focus the analyzes on the Salary player variable only (i.e. the annual salary that a player is going to earn, according to the statistics during the 2017 year)
- In particular our main task is to predict this response from our explanatory/predictors variables provided by the dataset using a multiple linear regression model.

Choosing the best fit distribution

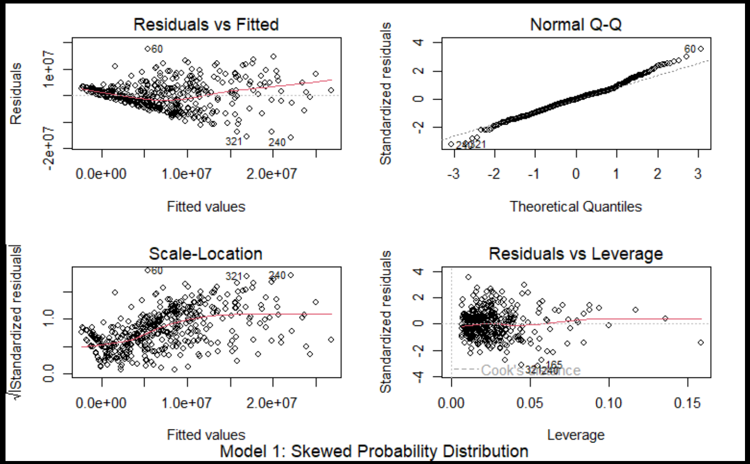
- From this histograms we can see:
  - when we use Square Root Transformation instead of Log Transformation, we got slightly better results.
  - First we tried Log transformation but the results were not satisfactory. Actually, this last one is most likely the first thing you should do to remove skewness from the predictor.
  - After that we used squared root transform, which gives us a distribution more similar to the normal distribution (The one we need)



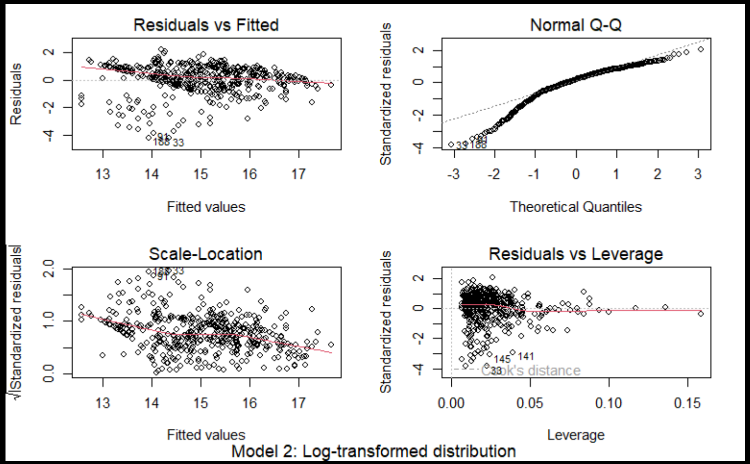
Adjusted R^2 Skewed Population Distribution	Adjusted R^2 Log Transformation	Adjusted R^2 Sqrt Transformation
0.5639	0.4708	0.5757



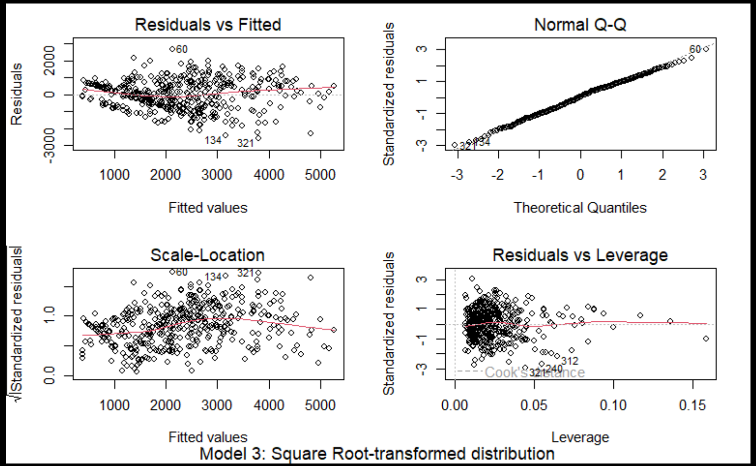
Skewed Data Probability Distribuition

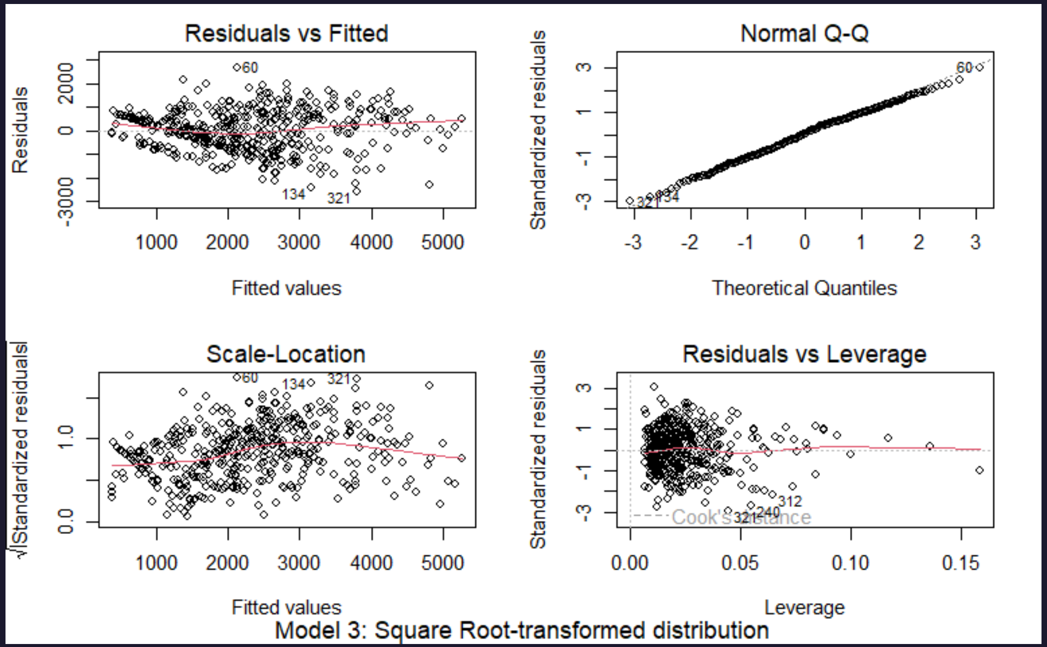


Log-Transformed Distribuition



Square Root-Transformed Distribuition






```
Call:
lm(formula = salary17_18_sqrt ~ MPG + PPG + APG + RPG + TOPG +
    BPG + SPG + Age + height + weight, data = stats_salary)

Residuals:
     Min       1Q   Median       3Q      Max
-2594.16  -628.69    44.62   607.64  2680.09

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3804.238   1439.646  -2.642  0.00853 **
MPG            35.958     12.295   2.925  0.00363 **
PPG            88.533     17.001   5.207 2.97e-07 ***
APG           123.983     53.153   2.333  0.02013 *
RPG            96.626     34.236   2.822  0.00499 **
TOPG         -321.248    144.468  -2.224  0.02669 *
BPG            82.470    157.680   0.523  0.60123
SPG           176.471    172.784   1.021  0.30767
Age            41.095     10.457   3.930 9.89e-05 ***
height         12.367      8.949   1.382  0.16768
weight          5.956      6.983   0.853  0.39421
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 886.8 on 431 degrees of freedom
Multiple R-squared:  0.5853,    Adjusted R-squared:  0.5757
F-statistic: 60.83 on 10 and 431 DF,  p-value: < 2.2e-16
```

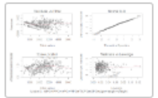
- R-Square: measures the proportion of variability of our Response Variable that can be explained using our Explanatory variables.
- Aim is to make R-Square near to one: measures of how regression predictions approximate real data points
- With the Square Root-transformed distribution: our adjusted R-squared is near to “One” (similar to a normal distribution) means that the RSS is near to 0 which in turn means that our regression predictions fit very well the data

Backward Stepwise Selection

- We applied a "Backward Stepwise Selection": technique to remove non statistically significant features

````
#TEST 2 removing "BPG" predictor

```{r}
lm.model_sqrt <- lm(formula= salary17_18_sqrt ~  MPG+PPG+APG+RPG+TOPG+SPG+Age+height+weight, data=stats_salary)
summary(lm.model_sqrt) #Adjusted R-squared:  0.5764

par(mfrow=c(2,2))
plot(lm.model_sqrt)
mtext("Model 3: MPG+PPG+APG+RPG+TOPG+SPG+Age+height+weight", side = 3, line = -28, outer = TRUE)
par(mfrow=c(1,1))
```
````

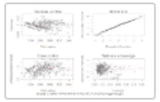
R Console



```
Call:
lm(formula = salary17_18_sqrt ~ MPG + PPG + APG + RPG + TOPG +
    SPG + Age + height + weight, data = stats_salary)

Residuals:
     Min       1Q   Median       3Q      Max
-2618.39  -628.80    45.07   613.40  2665.48

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3958.103   1408.083  -2.811 0.005164 **
MPG            36.035     12.284   2.933 0.003530 **
PPG            88.393     16.985   5.204 3.02e-07 ***
APG           120.901     52.782   2.291 0.022468 *
RPG           104.645     30.584   3.422 0.000682 ***
TOPG         -315.452    143.921  -2.192 0.028924 *
SPG           180.050    172.503   1.044 0.297186
Age            40.558     10.397   3.901 0.000111 ***
height         13.283      8.769   1.515 0.130563
weight          5.798      6.971   0.832 0.405999
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 886 on 432 degrees of freedom
Multiple R-squared:  0.585,     Adjusted R-squared:  0.5764
F-statistic: 67.67 on 9 and 432 DF,  p-value: < 2.2e-16
```

````
#TEST 3 removing "weight" predictor

```{r}
lm.model_sqrt <- lm(formula= salary17_18_sqrt ~  MPG+PPG+APG+RPG+TOPG+SPG+Age+height, data=stats_salary)
summary(lm.model_sqrt) #Adjusted R-squared:  0.5767

par(mfrow=c(2,2))
plot(lm.model_sqrt)
mtext("Model 1: MPG+PPG+APG+RPG+TOPG+SPG+Age+height", side = 3, line = -28, outer = TRUE)
par(mfrow=c(1,1))
```
````

R Console

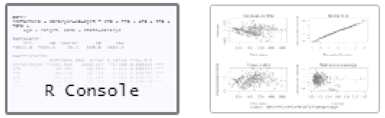


```
Call:
lm(formula = salary17_18_sqrt ~ MPG + PPG + APG + RPG + TOPG +
    SPG + Age + height, data = stats_salary)

Residuals:
     Min       1Q   Median       3Q      Max
-2602.12  -643.63    38.47   624.50  2665.95

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4474.060   1263.634  -3.541 0.000442 ***
MPG            34.488     12.138   2.841 0.004705 **
PPG            88.965     16.965   5.244 2.46e-07 ***
APG           119.290     52.727   2.262 0.024168 *
RPG           111.215     29.536   3.765 0.000189 ***
TOPG         -306.130    143.433  -2.134 0.033379 *
SPG           167.472    171.778   0.975 0.330138
Age            42.330     10.173   4.161 3.82e-05 ***
height         18.490      6.137   3.013 0.002738 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 885.7 on 433 degrees of freedom
Multiple R-squared:  0.5844,    Adjusted R-squared:  0.5767
F-statistic:  76.1 on 8 and 433 DF,  p-value: < 2.2e-16
```

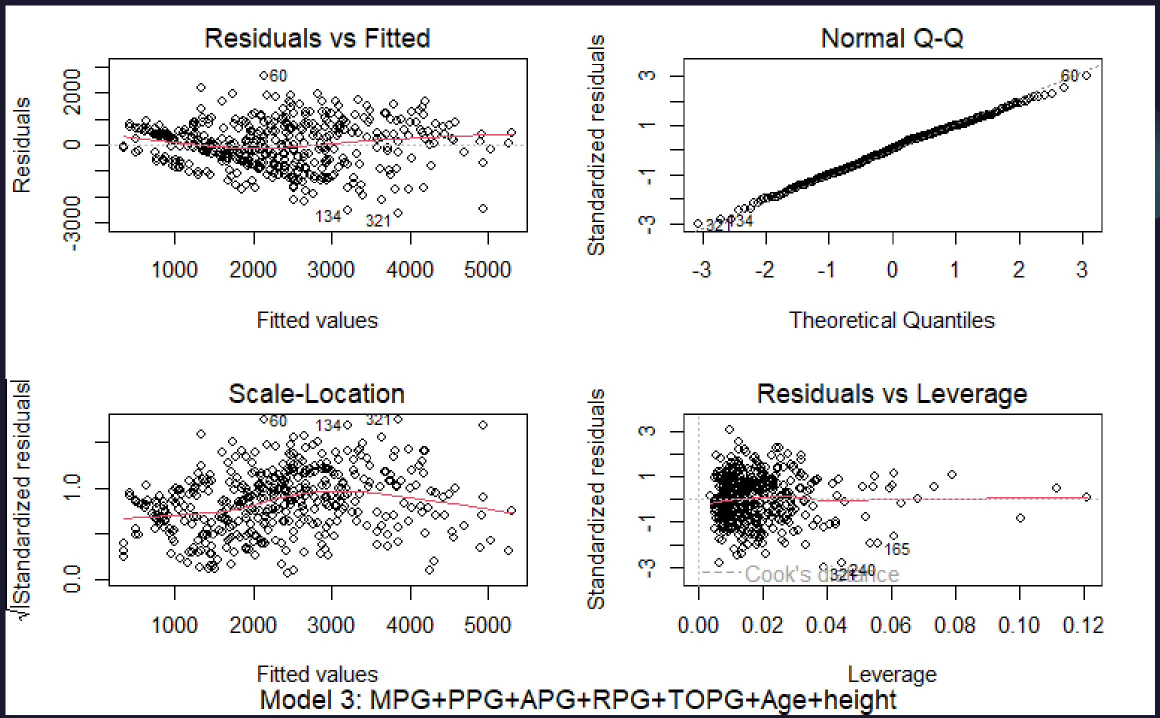
FINAL MODEL

~~~~
#TEST 4 removing "SPG" predictor to obtaining BEST MODEL

```{r}
lm.model_sqrt <- lm(formula= salary17_18_sqrt ~  MPG+PPG+APG+RPG+TOPG+Age+height, data=stats_salary)
summary(lm.model_sqrt) #Adjusted R-squared:  0.5767

par(mfrow=c(2,2))
plot(lm.model_sqrt)
mtext("Model 1: MPG+PPG+APG+RPG+TOPG+Age+height", side = 3, line = -28, outer = TRUE)
par(mfrow=c(1,1))
```
~~~~




```
Call:
lm(formula = salary17_18_sqrt ~ MPG + PPG + APG + RPG + TOPG +
    Age + height, data = stats_salary)

Residuals:
    Min      1Q  Median      3Q     Max
-2644.8  -633.1    49.4   630.0  2664.5

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4385.968   1260.327  -3.480 0.000552 ***
MPG            39.591     10.951   3.615 0.000335 ***
PPG            86.140     16.715   5.154 3.89e-07 ***
APG           131.532     51.207   2.569 0.010543 *
RPG           113.220     29.463   3.843 0.000140 ***
TOPG         -299.107    143.244  -2.088 0.037371 *
Age            41.664     10.149   4.105 4.83e-05 ***
height         18.091      6.123   2.955 0.003300 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 885.6 on 434 degrees of freedom
Multiple R-squared:  0.5835,	Adjusted R-squared:  0.5767
F-statistic: 86.84 on 7 and 434 DF,  p-value: < 2.2e-16
```



- After the “Backward Stepwise Selection” technique, we have more or less the same results, however now my model is less overfitted, less complex and more easy to interpret.

Outliers

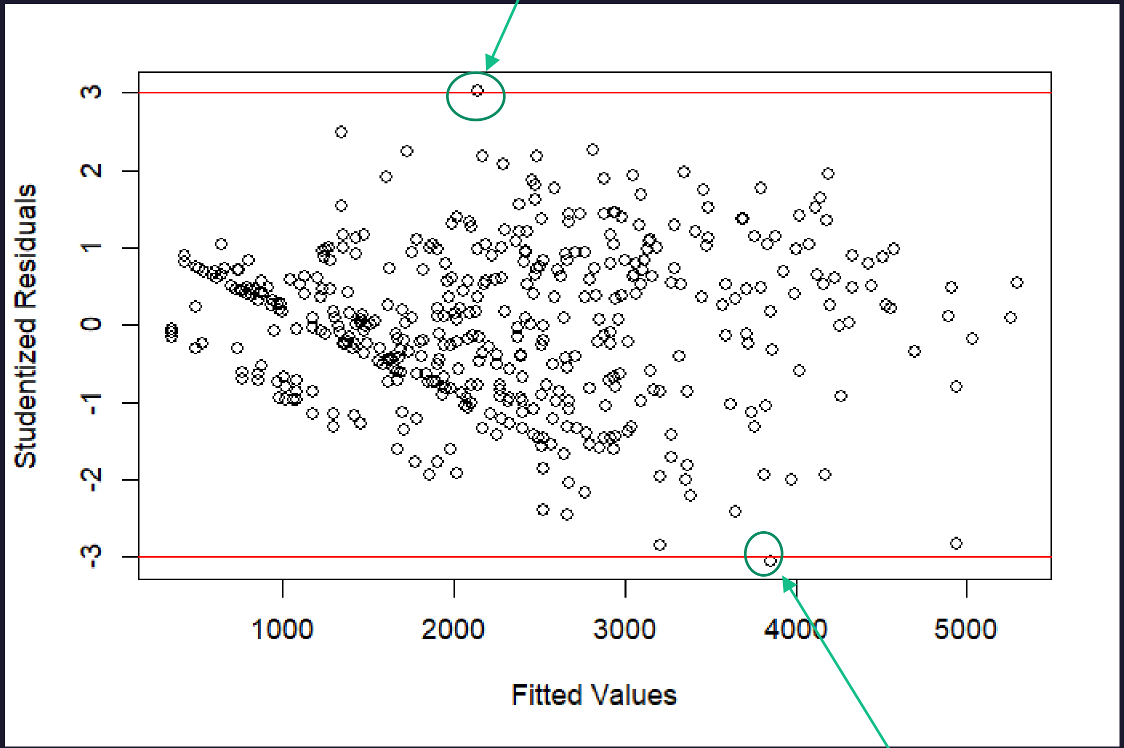
- WE HAVE TO CARRY also for 1. outliers and 2. leverage points
- The residual plot identifies some outliers. However, it can be difficult to decide how large a residual needs to be before we consider the point to be an outlier. To address this problem, instead of plotting the residuals, we can plot the studentized residuals, computed by dividing each residual Ei by its estimated standard error. Observations whose studentized residuals are greater than 3 in absolute value are possible outliers.
- Note that the empirical motivation for the value equal to 3 is that the Standardized Residuals are approximated by a N(0,1). The probability to observe a value greater than 3 is then 0.001349898.

Outliers

```{r}
1-pnorm(3)
```

```
[1] 0.001349898
```

- This norm is the value to identify that the pnorm in R is a built-in function that returns the value of the cumulative density function (cdf) of the normal distribution given a certain random variable q, and a population mean μ, and the population standard deviation σ.

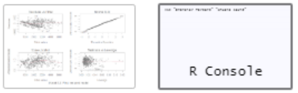


- An outlier is a data point whose response y does not follow the general trend of the rest of the data. A data point has high leverage if it has "extreme" predictor x values. With a single predictor, an extreme x value is simply one that is particularly high or low

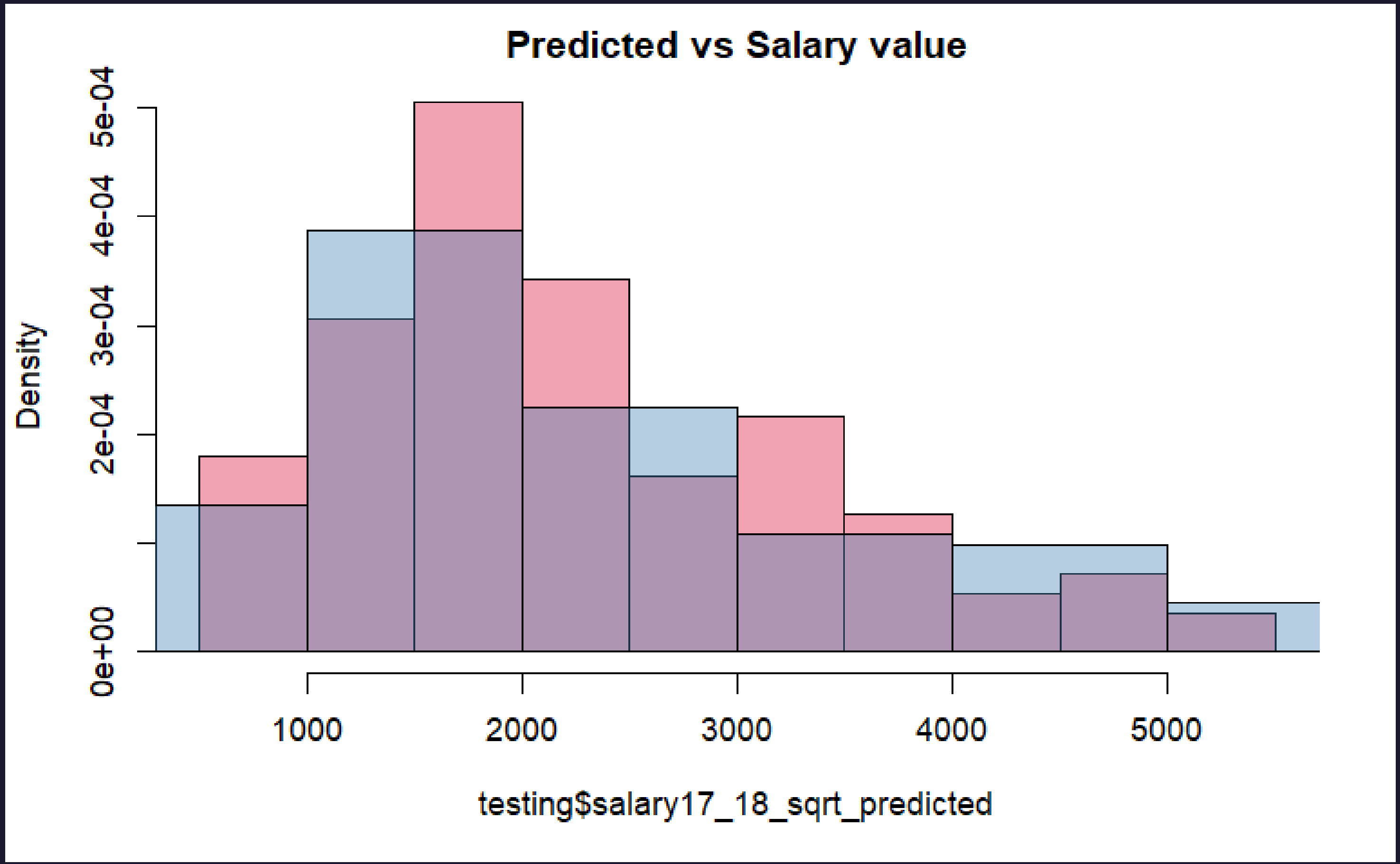
- a studentized residual is the value resulting from the division of a residual by an estimate of its standard deviation. It is a form of a Student's t-statistic, with the estimate of error varying between points. This is an important technique in the detection of outliers.

- Out two ourliers:

```
out <- names(rstandard(lm.model_sqrt)[(abs(rstandard(lm.model_sqrt)) > 3)])

# we have 442 players
playerout<-stats_salary$Player[rownames(stats_salary) %in% out]

# player I want to remove that rappresent my outliers
playerout
```




```
[1] "Chandler Parsons" "Nikola Jokic"
```



REGRESSION ON TRAINING SET PREDICTION ON TEST SET

red we present the predicted salary values and in blue we shower are our true salary values

Classification Problem

- Our goal was to predict the Annual Salary for the 2017 season, but in order to quantify the quality of our model we needed to transform our original regression into a classification problem.
- We proceeded as follows:

1. Analysis of the distribution of the square-root of the Salary
2. Definition of a list of thresholds that could divide in equal parts the distribution
3. Creation a new feature "Salary Class" generated from the thresholds applied to Salary
4. Training the linear model on training data
5. Prediction of the salary on test data
6. Application of the same thresholds to predicted Salary class
7. Building a Confusion Matrix from "Salary class" vs "Salary class predicted"

Classification: 3-Classes Model Results

Confusion Matrix:

reference\ predicted	**low**	**medium**	**high**
low	21	18	0
medium	4	30	9
high	0	8	21

```
[1] "Percentiles used:"
     33%      67%
1454.615 2828.427
[1] "Confusion Matrix ( 111 istances )"
       low medium high
low     21     18    0
medium   4     30    9
high     0      8   21
[1] "Accuracy = 64.86 %"
```

Classification : LDA & QDA comparison

We compared our model to some R built-in methods in order to prove his soundness

Salary Classes	Basic Classification Model	LDA built-in	QDA built-in
3 CLASSES	64,86%	71,17%	58,56 %

Our model reached lower accuracy compared to LDA built-in function

- QDA is not able to achieve good results

Our model vs LDA built-in

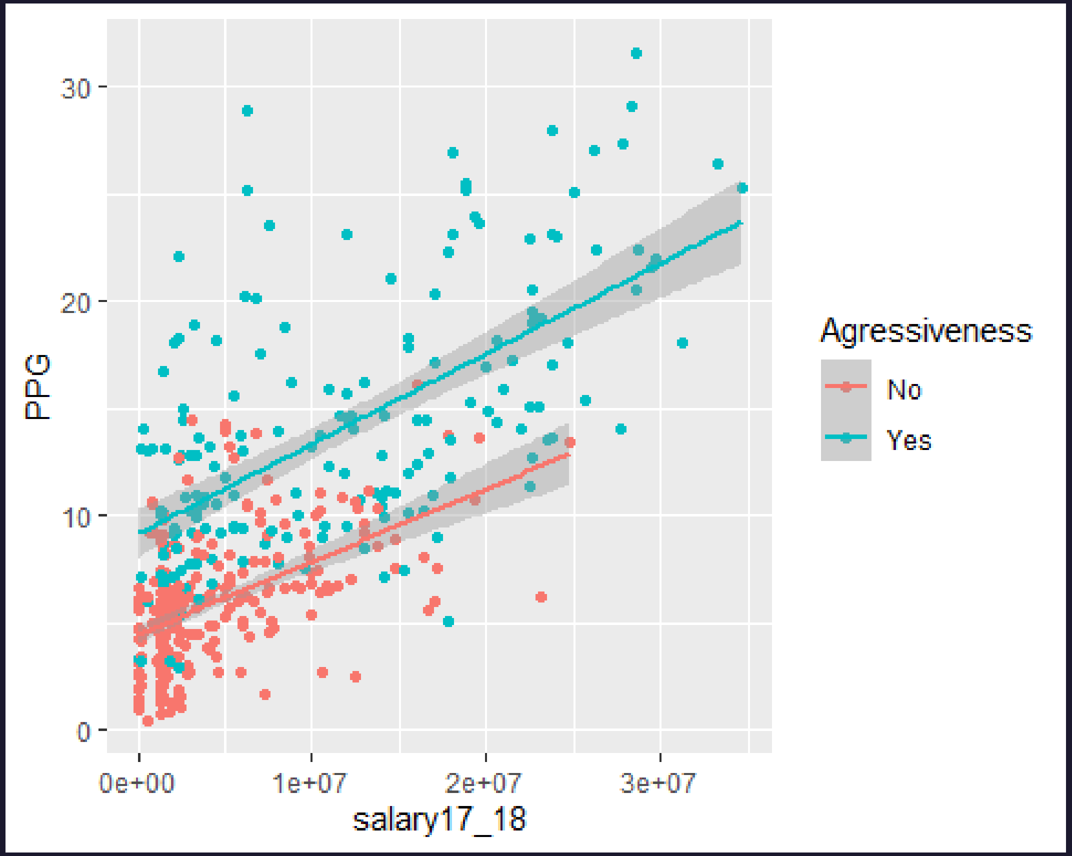
- LDA is easier to implement & reaches higher accuracies
- Our model is built from the ground up from a Multiple Linear Regression:
    1. We can inspect the diagnostic plots
    2. We can get the Adjusted R-squared of the model (and other statistics)
    3. We have a better understanding and a better control of the model

Summary of the models

- With our final model, considering 3 classes for the annual salary of 2017, we reached an Adjusted R2 of 0.5767 for the predictor and an average accuracy of 64,86% for the classifier
- Probably we would have been able to obtain a greater Adjusted R2: maybe if we had considered statistics from earlier years, we would have gotten it

5. Conclusion

- As a result, as we can clearly see, a lot of variables directly affect player salaries. However, two of them caught our attention during this project. The first of these, without doubt, is the time the basketball players take during the match. As the minutes played by the players increase, we can clearly say that their salaries also increase.
- For the second pick, our favorite was turnovers. As we mentioned at the beginning of the project, turnovers are closely related to player salaries.
- Anyway, even if the model has different possibilities for improvement, our simple model can still be used in some practical way



THANK YOU

Mahir Selek

Joi Berberi