Correlation and Regression - CS51

National Basketball Association Player Stats Analytics

César Emanuel Castro García

Minerva Schools at KGI

Introduction

In this report, we explore the correlation between minutes playing per game and field goal attempts of basketball players. We estimate how accurately our independent variable can predict our response variable of interest. In other words, is there convincing evidence that there exists a correlation between this predictor variable and the response variable? This analysis will provide some insight into the time playing per game of NBA players.

Dataset

This dataset is based on data from Basketball Reference (2017-2018 NBA Player Stats: Per Game, 2018). Here we can discover different player stats such as their age, position, minutes played per game, field goal attempts, and many more. For this report, the focus will be on minutes played per game and searching for a predictor variable that could accurately model real-world stats for NBA players. There are 664 players in the dataset. The data can be found **here** as a .csv file or **here** on the Basketball Reference website.

We are interested in using this sample data to find an appropriate correlation between our response variable and possible predictor variables. The names of the players are not crucial for our analyses, so we will focus on minutes played per game and field goal attempts as our variables of interest. We will try to answer the question: Is there any relationship between minutes played per game and the number of goal attempts per game? The minutes played per game, number of field goal attempts and many other variables found here are quantitative and continuous variables because they can be counted and can be any real number.[1]

[1] **#variables:** This application identifies and classifies the variables of interest in this report which are different from each other and can be of more than one type (quantitative and continuous).

Methods

The dataset was read into Python using the pandas package for analysis as seen on **Figure 1**. The descriptive statistics for the relevant variables explored in this analysis (Player, MP, Age, FGA, ORB, FT, AST, and STL) are in **Figure 2**. The calculations for these can be found in **Appendix A**.

	Player	MP	Age	FGA	ORB	FT	AST	STL
0	Alex Abrines	15.1	24	3.9	0.3	0.5	0.4	0.5
1	Quincy Acy	19.4	27	5.2	0.6	0.7	0.8	0.5
2	Steven Adams	32.7	24	9.4	5.1	2.1	1.2	1.2
3	Bam Adebayo	19.8	20	4.9	1.7	1.9	1.5	0.5
...	...	...	...	...	...	...	...	...
6	LaMarcus Aldridge	33.5	32	18.0	3.3	4.5	2.0	0.6
7	Jarrett Allen	20.0	19	5.5	2.0	1.6	0.7	0.4
8	Kadeem Allen	5.9	25	1.2	0.2	0.4	0.7	0.2
9	Tony Allen	12.4	36	4.1	0.9	0.5	0.4	0.5

Figure 1. Imported Data

Only the first 10 rows of data printed in the interest of space.

	MP	Age	FGA	ORB	FT	AST	STL
count	664.00	664.00	664.00	664.00	664.00	664.00	664.00
mean	18.64	26.20	6.54	0.75	1.24	1.74	0.60
std	9.31	4.13	4.47	0.71	1.23	1.67	0.44
min	1.00	19.00	0.00	0.00	0.00	0.00	0.00
25%	11.30	23.00	3.10	0.30	0.40	0.60	0.30
50%	18.60	26.00	5.50	0.50	0.90	1.20	0.50
75%	26.10	29.00	9.30	1.00	1.70	2.30	0.80
max	36.90	41.00	21.10	5.10	8.70	10.30	2.40

Figure 2. Descriptive Statistics

Relevant variables explored in the analysis (MP, Age, FGA, ORB, FT, AST, and STL) shown.

The summary statistics and the corresponding histogram for our two most relevant variables minutes played and field goal attempts per game are shown below. We see that the former has a similar mean and median, so the histogram is nearly normal. While the latter has a larger mean than median, so it is right skewed.[2] For the purpose of this analysis, we will consider these to be sample distributions from a total population.[3] These calculations are in **Appendix B**.

	Table 1: Summary statistics for variables of interest	
	Minutes played per game (min)	Field goal attempts per game
Count	$n_1 = 664$	$n_2 = 664$
Mean	$\bar{x}_{MP} = 18.211$	$\bar{x}_{FGA} = 6.113$
Median	18.0	5.0
Mode	16	3
Standard Deviation	$s_{MP} = 9.29$	$s_{FGA} = 4.46$
Range	36 – 1 = 35	21 – 0 = 21

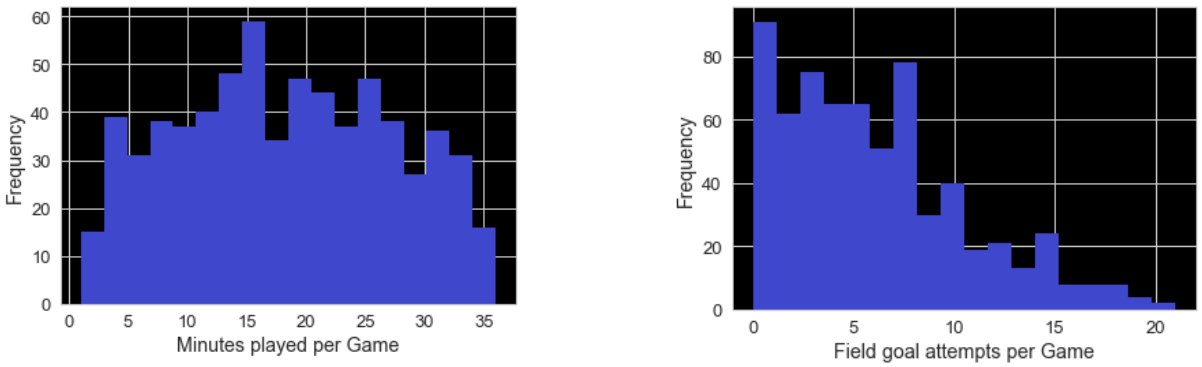


Figure 3. Histograms of the Variables of Interest

Histogram of Minutes Played per Game (left) and Histogram of Field Goal Attempts per Game (right).

[2] **#descriptivestats:** This application describes the use of descriptive statistics and justifies the choice, interpretation of the statistical parameters is given as well. Relevant variables are considered for purpose of creating a robust analysis of the data.

[3] **#distributions:** This application shows how different type of distribution have different features and how we can identify them and assume a normal distribution to make further progress with the statistical tools. For the purpose of the analysis, many inferences can be drawn when all these characteristics are considered.

The least square method to build a simple linear regression model uses some specific assumptions, which should be tested before making a model. For any regression model, we should examine some diagnostics of its success and validity. A scatterplot, a residual plot, a histogram, and a QQ Plot are all used to evaluate linearity, constant variability of the residuals, and normal distribution of the residuals. **Appendix C**.[4]

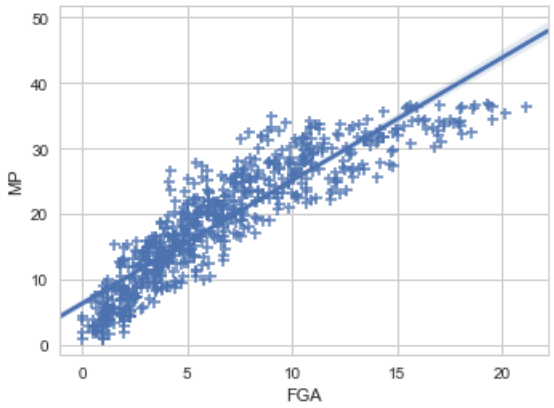


Figure 4. Scatter plot of FGA vs MP

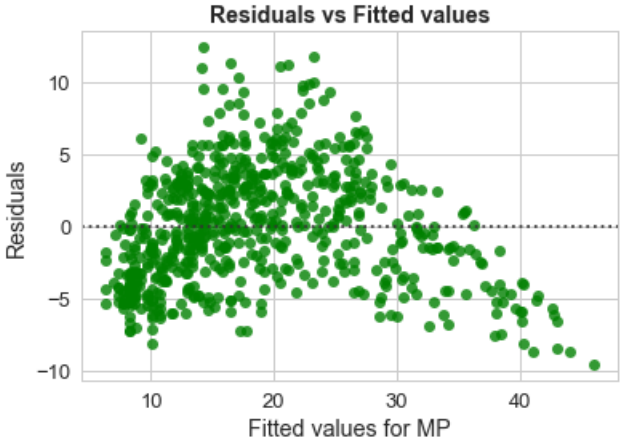


Figure 5. Residual plot for checking constant variability

[4] **#dataviz:** This application allows for a detailed data visualization correctly chosen to visualize the type of variables we have. Rules for creating a graph are followed and analysis is given afterwards.

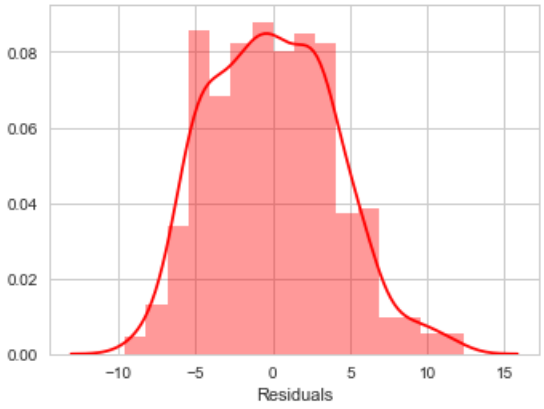


Figure 6. Histogram of Residuals

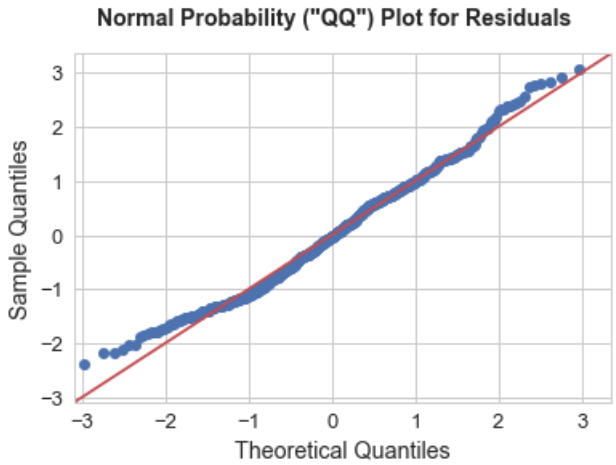


Figure 7. Q-Q Plot for checking normal distribution of Residuals

The correlation coefficient, Pearson's r shows how tightly a set of data can be linearly associated. Here the value of r is 0.9 which implies a very strong correlation between the number of field goal attempts and minutes played per game. Also, the sign of Pearson's r shows the sign of the slope; so, they are positively correlated. **Appendix D**.[5]

The coefficient of determination, R^2 shows how much variation of the response variable (minutes played per game) can be explained by the predictor (field goal attempts). The value of

[5] **#correlation:** This application allows for an appropriate use of Pearson correlation coefficient by correctly identifying that the collection of bivariate data (FGA and MP) have a high value and interpret the implications of this calculation. The calculation of this coefficient helps make my analysis even more robust.

R^2 is 0.813 that means that 81.3% of variance of the response variable can be explained by the predictor. **Appendix C**.

From the regression model calculated by statsmodel library functions, the equation of the linear regression is the following:

$$MP = b_0 + b_1 FGA$$

$$MP = 6.3634 + 1.8769 * FGA$$

According to our model, an increase in a field goal attempt will most likely result in 1.8769 more minutes played per game for an NBA player. This model also predicts that someone that doesn't attempt a single field goal will play only for 6.3634 minutes.[6]

The hypotheses that will be used to measure the statistical significance of the slope, b1, are the following (Appendix E)[7]:

Null Hypothesis: $H_0: b_1 = 0$, there is no correlation between the field goal attempts and minutes played.

Alternative Hypothesis: $H_A: b_1 \neq 0$, a non-zero correlation exists between the field goal attempts and minutes played.

A two-tailed test with the significance level, $\alpha = 0.05$, can be computed. The p-value and the 95% confidence interval are calculated using the regression model function of statsmodel library.

The 95% confidence interval of the slope is [1.808, 1.945]. It means that if we took 100 different datasets of the same sample size from the actual population and each time, we calculate

[6] **#regression:** This application allows for the use of the regression equation to provide a well justified interpretation of the relation between field goal attempts and minutes played per game. The calculation of the regression equation helps guide our inferences, predictions and explains the result in the given context.
[7] **#significance:** This application applies statistical significance tests to interpret and making inferences regarding the research question. An interpretation is given on the calculations obtained, i.e. interpreting the null and alternative hypothesis and how they relate to p-values obtained.

a 95% confidence interval of the slope, 95 of the time, the interval range will overlap with the population slope. **Appendix C**.[8] We can reject the null hypothesis because our p-value for our model is less than the assigned significance level.

Results and Conclusions

The results for the linear regression models provide insights to our study. The calculations for different models can be found in **Appendix E**. Thus, we can say that we have enough evidence that the time playing per games in minutes can be predicted somewhat accurately only with the number of field goal attempts which was the main calculation discussed in this paper. In the context of this dataset and using a one predictor model, this means that as more field goal attempts an NBA player has per game, they will likely have more time to play. Also, other predictors that are independent from each other can increase the accuracy of our model and with the variables in this dataset we found a five-predictor model that is 87.0% accurate.

These conclusions are inductive because we are not 100% certain about our result, we cannot say that our analysis proved that there is causation but there definitely is correlation. However, they are still reliable because they are based on different analysis of samples and using statistical formulas.[9]

In summary, we've shown that the time playing per game in minutes positively correlates with the number of field goal attempts per game based on the data.

WORD COUNT: 979

[8] **#confidenceintervals:** This application shows how we considered the assumptions and use appropriate formula while calculating a plausible range for the value of b1, the slope of our regression equation model.
[9] **#induction:** This application allows for the use of evidence to reach a conclusion and going beyond the direct implications of the premises. The weaknesses and reliability of the argument are also explored.

References

Basketball Reference. (2018). *2017-2018 NBA Player Stats: Per Game*, 1. Retrieved January 24th, 2019, from Basketball Reference: https://www.basketball-reference.com/leagues/NBA_2018_per_game.html

Appendix

The full Jupyter notebook file can be accessed **here**; the pdf is found **here**.

Appendix A: Import and Analyze Data

In [1]:
```
#import relevant packages and libraries
import pandas as pd
import numpy as np
import scipy as sc
from scipy import stats
import matplotlib.pyplot as plt
```

In [2]:
```
#import data using pandas dataframe
mnts = pd.read_csv('assignment.csv')
#choose relevant column
mnts = mnts.loc[:, ['Player','MP','Age','FGA','ORB','FT','AST','STL']]
mnts.head(10)
#printing 1st 10 rows of data
```

Out[2]:

	Player	MP	Age	FGA	ORB	FT	AST	STL
0	Alex Abrines	15.1	24	3.9	0.3	0.5	0.4	0.5
1	Quincy Acy	19.4	27	5.2	0.6	0.7	0.8	0.5
2	Steven Adams	32.7	24	9.4	5.1	2.1	1.2	1.2
3	Bam Adebayo	19.8	20	4.9	1.7	1.9	1.5	0.5
4	Arron Afflalo	12.9	32	3.1	0.1	0.4	0.6	0.1
5	Cole Aldrich	2.3	29	0.7	0.1	0.1	0.1	0.1
6	LaMarcus Aldridge	33.5	32	18.0	3.3	4.5	2.0	0.6
7	Jarrett Allen	20.0	19	5.5	2.0	1.6	0.7	0.4
8	Kadeem Allen	5.9	25	1.2	0.2	0.4	0.7	0.2
9	Tony Allen	12.4	36	4.1	0.9	0.5	0.4	0.5

In [3]:
```
np.round(mnts.describe(),2)
#describe the summary statistics
```

Out[3]:

	MP	Age	FGA	ORB	FT	AST	STL
count	664.00	664.00	664.00	664.00	664.00	664.00	664.00
mean	18.64	26.20	6.54	0.75	1.24	1.74	0.60
std	9.31	4.13	4.47	0.71	1.23	1.67	0.44
min	1.00	19.00	0.00	0.00	0.00	0.00	0.00
25%	11.30	23.00	3.10	0.30	0.40	0.60	0.30
50%	18.60	26.00	5.50	0.50	0.90	1.20	0.50
75%	26.10	29.00	9.30	1.00	1.70	2.30	0.80
max	36.90	41.00	21.10	5.10	8.70	10.30	2.40

In [4]:
```
mnts_game = []
start = np.array(mnts['MP'])
for i in range(len(start)):
    start[i] = float(start[i])
    mnts_game.append(int(start[i]))
print(mnts_game)
```

```
[15, 19, 32, 19, 12, 2, 33, 20, 5, 12, 30, 13, 26, 26, 2, 36, 32, 20, 12, 33, 7, 18, 10, 8, 15, 23, 14, 15, 11, 13, 13, 11, 34,
23, 34, 33, 31, 23, 18, 27, 36, 9, 22, 24, 23, 26, 14, 17, 25, 14, 30, 13, 8, 18, 20, 10, 16, 31, 27, 31, 9, 27, 30, 9, 6, 34,
17, 21, 15, 15, 1, 31, 31, 27, 3, 16, 12, 28, 29, 5, 28, 27, 4, 5, 30, 9, 7, 14, 2, 4, 27, 21, 16, 36, 14, 8, 3, 10, 16, 33, 2
1, 4, 22, 27, 29, 17, 16, 15, 14, 28, 13, 21, 25, 31, 21, 19, 23, 23, 22, 9, 7, 6, 10, 24, 15, 15, 29, 5, 31, 18, 22, 2, 12, 8,
36, 31, 29, 16, 20, 10, 26, 25, 27, 21, 21, 20, 32, 20, 36, 15, 18, 24, 12, 18, 18, 13, 33, 11, 16, 28, 17, 10, 12, 1, 31, 7,
5, 7, 33, 14, 29, 34, 3, 3, 3, 8, 26, 30, 22, 23, 20, 12, 16, 5, 30, 16, 14, 28, 8, 9, 3, 17, 16, 12, 27, 21, 19, 32, 27, 14, 1
3, 12, 16, 18, 32, 14, 33, 23, 21, 36, 5, 10, 33, 20, 32, 32, 31, 25, 16, 20, 22, 17, 25, 32, 22, 28, 23, 34, 34, 33, 4, 33, 3
5, 21, 17, 18, 18, 19, 34, 25, 33, 32, 34, 25, 23, 16, 23, 5, 13, 5, 3, 21, 5, 2, 18, 25, 11, 10, 9, 11, 22, 13, 25, 14, 27, 2
6, 27, 15, 6, 36, 31, 7, 28, 15, 26, 27, 25, 8, 31, 17, 30, 14, 9, 1, 27, 25, 25, 25, 24, 31, 32, 33, 32, 19, 16, 25, 35, 5, 5,
5, 25, 22, 26, 36, 19, 20, 4, 13, 8, 14, 15, 15, 5, 4, 6, 5, 26, 21, 21, 22, 18, 27, 28, 20, 32, 5, 12, 5, 15, 11, 4, 13, 17, 3
1, 27, 23, 25, 20, 25, 12, 11, 8, 9, 23, 16, 5, 16, 21, 19, 31, 2, 20, 24, 14, 9, 27, 4, 8, 30, 26, 7, 20, 23, 7, 17, 26, 12, 1
5, 9, 36, 25, 15, 13, 23, 26, 28, 32, 15, 2, 19, 19, 12, 14, 16, 8, 9, 8, 29, 22, 18, 5, 33, 25, 6, 16, 36, 22, 2, 4, 21, 21, 2
2, 9, 16, 1, 3, 19, 3, 14, 12, 12, 36, 19, 23, 8, 25, 30, 27, 24, 29, 33, 1, 13, 20, 15, 23, 19, 4, 4, 31, 12, 16, 26, 27, 8, 1
6, 11, 19, 17, 22, 9, 9, 10, 3, 21, 31, 20, 10, 21, 22, 20, 20, 20, 20, 16, 12, 3, 5, 15, 8, 24, 21, 26, 23, 10, 16, 18, 15, 1
3, 12, 12, 12, 34, 23, 22, 11, 27, 14, 5, 7, 7, 4, 24, 19, 19, 15, 4, 9, 31, 8, 23, 28, 28, 29, 9, 8, 10, 15, 4, 11, 6, 19, 16,
18, 8, 31, 22, 32, 21, 15, 4, 2, 18, 6, 4, 14, 26, 25, 23, 30, 11, 10, 10, 3, 33, 12, 12, 5, 33, 26, 13, 14, 26, 16, 19, 12, 2
5, 25, 29, 25, 24, 15, 29, 22, 31, 18, 21, 19, 14, 12, 15, 19, 20, 3, 33, 29, 20, 4, 29, 29, 24, 28, 8, 4, 27, 13, 12, 7, 13, 2
2, 3, 7, 2, 7, 30, 17, 33, 24, 15, 24, 25, 16, 14, 26, 27, 26, 18, 34, 20, 15, 22, 35, 27, 25, 28, 12, 23, 22, 27, 20, 7, 7, 4,
7, 16, 14, 19, 29, 22, 23, 22, 30, 34, 34, 28, 30, 9, 9, 33, 13, 12, 13, 9, 23, 13, 36, 13, 8, 13, 11, 25, 36, 6, 8, 14, 18, 4,
32, 25, 3, 14, 4, 17, 3, 18, 24, 4, 3, 13, 13, 15, 20, 7, 10, 10, 17, 32, 19, 16, 16, 16, 15, 6, 9]
```

In [5]:
```
#summary statistics
def all_stats(sample):
    mean = round(np.mean(sample),3)
    median = np.median(sample)
    mode = stats.mode(sample)[0][0]
    min_value = min(sample)
    max_value = max(sample)
    sd = round(np.std(sample, ddof = 1),2) #bassel's correction: using n-1 as denominator
    print(" - Mean =",mean)
    print(" - Median =",median)
    print(" - Mode =",mode)
    print(" - SD =",sd)
    print(" - Min =",min_value)
    print(" - Max =",max_value)
```

In [6]:
```
#printing the summary statistics
print("Minutes played per Game")
all_stats(mnts_game)
```

```
Minutes played per Game
 - Mean = 18.211
 - Median = 18.0
 - Mode = 16
 - SD = 9.29
 - Min = 1
 - Max = 36
```

In [8]:
```
fgas_game = []
start = np.array(mnts['FGA'])
for i in range(len(start)):
    start[i] = float(start[i])
    fgas_game.append(int(start[i]))
print(fgas_game)
```

```
[3, 5, 9, 4, 3, 0, 18, 5, 1, 4, 8, 5, 5, 7, 0, 18, 15, 4, 1, 9, 2, 5, 1, 1, 1, 7, 4, 4, 3, 3, 2, 3, 10, 10, 15, 12, 10, 6, 5, 1
0, 18, 2, 10, 9, 9, 10, 3, 4, 5, 4, 10, 3, 2, 4, 5, 2, 7, 13, 13, 13, 1, 9, 10, 1, 1, 19, 5, 8, 3, 4, 1, 13, 14, 9, 1, 4, 3, 8,
10, 2, 9, 14, 1, 2, 11, 2, 1, 3, 0, 1, 8, 10, 6, 15, 6, 2, 1, 2, 3, 10, 6, 1, 7, 9, 10, 4, 4, 3, 3, 10, 3, 6, 4, 8, 6, 6, 11, 1
2, 10, 2, 1, 1, 1, 7, 4, 3, 9, 1, 14, 4, 7, 1, 3, 1, 18, 10, 11, 3, 9, 5, 8, 7, 10, 5, 4, 6, 16, 7, 19, 4, 3, 7, 3, 5, 4, 2, 1
7, 3, 4, 10, 6, 3, 4, 1, 14, 2, 2, 2, 11, 2, 12, 18, 0, 0, 1, 3, 9, 16, 5, 4, 6, 4, 5, 1, 15, 6, 4, 9, 4, 4, 2, 3, 6, 2, 8, 5,
6, 14, 10, 2, 4, 3, 4, 7, 12, 5, 14, 8, 9, 17, 1, 5, 9, 7, 7, 14, 14, 6, 3, 5, 6, 8, 8, 8, 10, 8, 7, 17, 17, 16, 1, 15, 20, 5,
7, 6, 6, 6, 13, 8, 15, 14, 16, 7, 7, 5, 6, 1, 3, 1, 1, 5, 2, 0, 5, 6, 3, 3, 3, 3, 8, 4, 11, 4, 7, 7, 7, 3, 1, 15, 10, 2, 10, 5,
12, 14, 9, 1, 10, 5, 11, 2, 4, 1, 10, 5, 8, 8, 9, 8, 8, 12, 18, 5, 3, 7, 9, 1, 1, 1, 12, 6, 12, 19, 8, 9, 2, 5, 1, 5, 4, 3, 2,
1, 3, 1, 8, 6, 7, 5, 5, 8, 9, 5, 13, 1, 2, 0, 3, 4, 1, 5, 4, 7, 7, 9, 10, 6, 7, 5, 5, 3, 4, 11, 4, 2, 6, 6, 5, 13, 2, 7, 10, 4,
3, 14, 1, 2, 9, 10, 3, 5, 12, 2, 4, 10, 1, 2, 1, 19, 12, 4, 2, 10, 10, 12, 12, 5, 0, 7, 6, 2, 3, 4, 3, 4, 3, 12, 6, 7, 1, 11,
5, 2, 3, 18, 5, 0, 2, 6, 6, 6, 3, 4, 0, 1, 6, 0, 5, 2, 3, 15, 8, 5, 1, 8, 11, 12, 12, 12, 17, 1, 6, 7, 6, 7, 7, 0, 0, 10, 1, 5,
11, 9, 2, 6, 3, 8, 7, 9, 4, 3, 6, 1, 7, 13, 5, 3, 6, 6, 6, 7, 4, 4, 5, 3, 1, 1, 3, 1, 9, 6, 12, 6, 4, 4, 5, 2, 3, 4, 4, 5, 17,
8, 5, 3, 9, 3, 2, 2, 2, 1, 10, 7, 6, 3, 2, 2, 13, 2, 8, 10, 10, 11, 3, 2, 3, 2, 0, 3, 0, 5, 3, 4, 2, 11, 11, 18, 5, 5, 1, 0, 6,
1, 1, 4, 11, 12, 8, 12, 3, 2, 3, 0, 10, 3, 3, 2, 13, 4, 3, 3, 7, 7, 8, 6, 8, 10, 10, 14, 8, 3, 11, 5, 17, 6, 6, 7, 4, 4, 4, 4,
6, 1, 12, 11, 5, 1, 9, 14, 9, 7, 3, 1, 5, 6, 3, 0, 4, 8, 1, 0, 1, 2, 10, 5, 11, 5, 5, 7, 8, 3, 3, 13, 12, 13, 3, 16, 4, 3, 6, 1
4, 5, 7, 9, 1, 7, 8, 9, 7, 2, 2, 0, 1, 4, 2, 6, 14, 10, 9, 11, 13, 17, 16, 8, 11, 1, 4, 16, 4, 2, 2, 1, 4, 5, 21, 4, 1, 2, 5, 1
0, 15, 0, 2, 3, 5, 2, 16, 7, 2, 5, 2, 5, 0, 5, 7, 1, 1, 3, 3, 3, 6, 1, 2, 3, 6, 10, 4, 5, 5, 4, 4, 2, 2]
```

In [9]:
```
#printing the summary statistics
print("Field Goal attempts per Game")
all_stats(fgas_game)
```

```
Field Goal attempts per Game
 - Mean = 6.113
 - Median = 5.0
 - Mode = 3
 - SD = 4.46
 - Min = 0
 - Max = 21
```

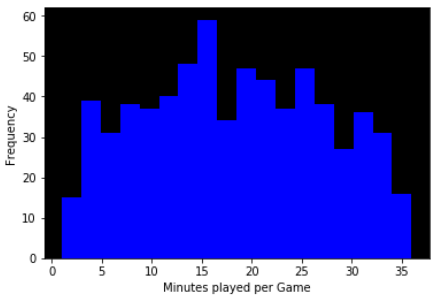
**** This is where the relevant information of Appendix A ends, in the Jupyter notebook there is further exploration of the data that is not specific for this analysis.**

Appendix B: Visualize Data

In [10]:
```
#changing the background color
ax = plt.gca()
ax.set_facecolor('xkcd:black')

#creating histogram
plt.hist(mnts_game,18,facecolor='b')
plt.xlabel('Minutes played per Game')
plt.ylabel('Frequency')
```

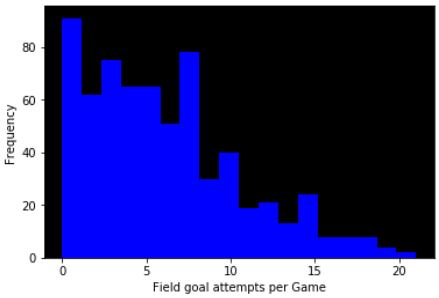
Out[10]:



In [11]:
```
#changing the background color
ax = plt.gca()
ax.set_facecolor('xkcd:black')

#creating histogram
plt.hist(fgas_game,18,facecolor='b')
plt.xlabel('Field goal attempts per Game')
plt.ylabel('Frequency')
```

Out[11]:



Appendix C: Regression Model (Assumptions)

In [29]:
```
def mult_regression(column_x, column_y):
    ''' this function uses built in library functions to construct a linear
    regression model with potentially multiple predictor variables. It outputs
    two plots to assess the validity of the model.'''

    #If there is only one predictor variable, plot the regression line
    if len(column_x)==1:
        plt.figure()
        sns.regplot(x=column_x[0], y=column_y, data=data, marker="+",fit_reg=True,color='orange')

    #define predictors X and response Y:
    X = data[column_x]
    X = statsmodels.add_constant(X)
    Y = data[column_y]

    #construct model:
    global regressionmodel
    regressionmodel = statsmodels.OLS(Y,X).fit() #OLS stands for "ordinary least squares"

    #residual plot:
    plt.figure()
    residualplot = sns.residplot(x=regressionmodel.predict(), y=regressionmodel.resid, color='green')
    residualplot.set(xlabel='Fitted values for '+column_y, ylabel='Residuals')
    residualplot.set_title('Residuals vs Fitted values',fontweight='bold',fontsize=14)

    #QQ plot:
    qqplot = statsmodels.qqplot(regressionmodel.resid,fit=True,line='45')
    qqplot.suptitle("Normal Probability (\"QQ\") Plot for Residuals",fontweight='bold',fontsize=14)
```

A simple model to start

Below are the results for a simple linear regression model, using **the average number of field goal attempts per game** (code `FGA`) to predict **the time played per game in minutes** (code `MP`).

$$\text{MP} = b_0 + b_1\text{FGA}$$

1. How well can we predict the time played in minutes from the average number of field goal attempts per game? (How much of the total variation in time played in minutes among the 664 samples is explained by field goal attempts?)
2. What are b_0 and b_1? Are these values statistically different than 0 at the $\alpha = 0.05$ significance level? Would you reject the null hypothesis that $H_0 : \beta_1 = 0$?
3. What do you infer from the plots about the assumptions for a linear model?

Notes here:

1. We can predict the time played in minutes really well from the average number of field goal attempts per game. The total variation in time played in minutes among the 664 samples can be explained by 81.3% of the field goal attempts.
2. In this case b_0 is 6.3634 and b_1 is 1.8769. These values are indeed statistically different than 0 at the $a = 0.05$ significance level. The most appropriate thing to do here would be to reject the null hypothesis that $H_0 : \beta_1 = 0$.
3. From the plots about the assumptions for a linear model, I infer that it is correct to make a linear regression model because they show linearity and reasonably similar variance of the residuals.

In [30]:
```
mult_regression(['FGA'],'MP')
regressionmodel.summary()
```

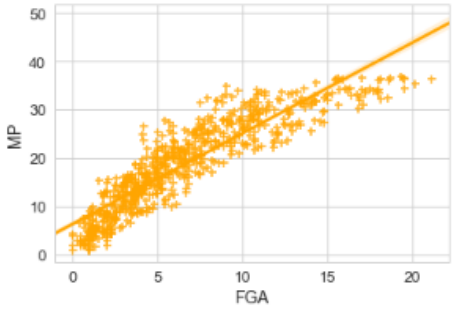
Out[30]:

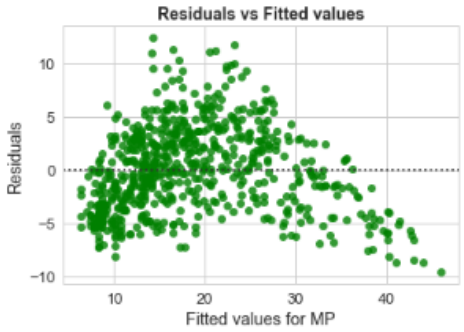
OLS Regression Results

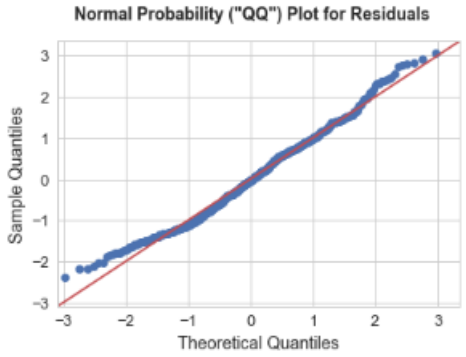
Dep. Variable:	MP	**R-squared:**	0.813
Model:	OLS	**Adj. R-squared:**	0.813
Method:	Least Squares	**F-statistic:**	2886.
Date:	Mon, 04 Feb 2019	**Prob (F-statistic):**	1.55e-243
Time:	06:22:39	**Log-Likelihood:**	-1865.4
No. Observations:	664	**AIC:**	3735.
Df Residuals:	662	**BIC:**	3744.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	6.3634	0.277	22.996	0.000	5.820	6.907
FGA	1.8769	0.035	53.723	0.000	1.808	1.945

Omnibus:	12.059	**Durbin-Watson:**	1.670
Prob(Omnibus):	0.002	**Jarque-Bera (JB):**	11.453
Skew:	0.280	**Prob(JB):**	0.00326
Kurtosis:	2.682	**Cond. No.**	14.2







Appendix D: Correlation Coefficient

In [36]:
```
data = pd.DataFrame({'Minutes per game': mnts_game,
                     'FG attempts per game': fgas_game})
data.head(16)
```

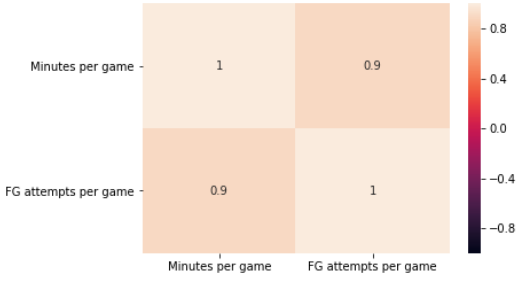
Out[36]:

	Minutes per game	FG attempts per game
0	15	3
1	19	5
2	32	9
3	19	4
...	...	...
12	26	5
13	26	7
14	2	0
15	36	18

16 rows × 2 columns

In [14]:
```
import statsmodels.api as statsmodels #package for stats modelling
import seaborn as sns
sns.heatmap(np.round(data.corr(),2), vmax=1, vmin=-1, annot = True)
```

Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x15daca76400>



Appendix E: Comparison of Regression Models

In [28]:
```
#Import useful packages
import pandas #package for data analysis
pandas.set_option('max_rows', 8)
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import matplotlib
%matplotlib inline
import statsmodels.api as statsmodels #useful stats package with linear regression functions
import seaborn as sns #very nice plotting package
sns.set(color_codes=True, font_scale=1.3)
sns.set_style("whitegrid")

#import data
filename = 'assignment.csv'
data = pandas.read_csv(filename)
data
```

Out[28]:

	Rk	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	...	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PS/G
0	1	Alex Abrines	SG	24	OKC	75	8	15.1	1.5	3.9	...	0.848	0.3	1.2	1.5	0.4	0.5	0.1	0.3	1.7	4.7
1	2	Quincy Acy	PF	27	BRK	70	8	19.4	1.9	5.2	...	0.817	0.6	3.1	3.7	0.8	0.5	0.4	0.9	2.1	5.9
2	3	Steven Adams	C	24	OKC	76	76	32.7	5.9	9.4	...	0.559	5.1	4.0	9.0	1.2	1.2	1.0	1.7	2.8	13.9
3	4	Bam Adebayo	C	20	MIA	69	19	19.8	2.5	4.9	...	0.721	1.7	3.8	5.5	1.5	0.5	0.6	1.0	2.0	6.9
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
660	537	Tyler Zeller	C	28	MIL	24	1	16.9	2.6	4.4	...	0.895	2.0	2.7	4.6	0.8	0.3	0.6	0.5	2.0	5.9
661	538	Paul Zipser	SF	23	CHI	54	12	15.3	1.5	4.3	...	0.760	0.2	2.2	2.4	0.9	0.4	0.3	0.8	1.6	4.0
662	539	Ante Zizic	C	21	CLE	32	2	6.7	1.5	2.1	...	0.724	0.8	1.1	1.9	0.2	0.1	0.4	0.3	0.9	3.7
663	540	Ivica Zubac	C	20	LAL	43	0	9.5	1.4	2.8	...	0.765	1.0	1.8	2.9	0.6	0.2	0.3	0.6	1.1	3.7

664 rows × 30 columns

In [31]:
```
mult_regression(['FGA','ORB'],'MP')
regressionmodel.summary()
```

Out[31]:

OLS Regression Results

Dep. Variable:	MP	**R-squared:**	0.831
Model:	OLS	**Adj. R-squared:**	0.830
Method:	Least Squares	**F-statistic:**	1622.
Date:	Mon, 04 Feb 2019	**Prob (F-statistic):**	1.07e-255
Time:	06:22:40	**Log-Likelihood:**	-1833.1
No. Observations:	664	**AIC:**	3672.
Df Residuals:	661	**BIC:**	3686.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	5.5815	0.280	19.909	0.000	5.031	6.132
FGA	1.7871	0.035	50.995	0.000	1.718	1.856
ORB	1.8249	0.222	8.227	0.000	1.389	2.260

Omnibus:	10.659	**Durbin-Watson:**	1.681
Prob(Omnibus):	0.005	**Jarque-Bera (JB):**	9.991
Skew:	0.255	**Prob(JB):**	0.00677
Kurtosis:	2.682	**Cond. No.**	16.1

In [32]:
```
mult_regression(['FGA','ORB','FT'],'MP')
regressionmodel.summary()
```

Out[32]:

OLS Regression Results

Dep. Variable:	MP	R-squared:	0.839
Model:	OLS	Adj. R-squared:	0.838
Method:	Least Squares	F-statistic:	1148.
Date:	Mon, 04 Feb 2019	Prob (F-statistic):	2.41e-261
Time:	06:22:40	Log-Likelihood:	-1816.2
No. Observations:	664	AIC:	3640.
Df Residuals:	660	BIC:	3658.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	5.1358	0.284	18.094	0.000	4.578	5.693
FGA	2.0750	0.060	34.722	0.000	1.958	2.192
ORB	2.0515	0.220	9.333	0.000	1.620	2.483
FT	-1.2995	0.221	-5.875	0.000	-1.734	-0.865

Omnibus:	10.583	Durbin-Watson:	1.697
Prob(Omnibus):	0.005	Jarque-Bera (JB):	9.470
Skew:	0.234	Prob(JB):	0.00878
Kurtosis:	2.649	Cond. No.	17.9

In [33]:
```
mult_regression(['FGA','ORB','FT','AST'],'MP')
regressionmodel.summary()
```

Out[33]:

OLS Regression Results

Dep. Variable:	MP	R-squared:	0.850
Model:	OLS	Adj. R-squared:	0.849
Method:	Least Squares	F-statistic:	936.1
Date:	Mon, 04 Feb 2019	Prob (F-statistic):	4.42e-270
Time:	06:22:40	Log-Likelihood:	-1792.2
No. Observations:	664	AIC:	3594.
Df Residuals:	659	BIC:	3617.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	4.9318	0.276	17.899	0.000	4.391	5.473
FGA	1.8778	0.064	29.259	0.000	1.752	2.004
ORB	2.4123	0.218	11.048	0.000	1.984	2.841
FT	-1.5552	0.217	-7.179	0.000	-1.981	-1.130
AST	0.8825	0.126	7.016	0.000	0.635	1.129

Omnibus:	4.606	Durbin-Watson:	1.742
Prob(Omnibus):	0.100	Jarque-Bera (JB):	4.644
Skew:	0.203	Prob(JB):	0.0981
Kurtosis:	2.952	Cond. No.	18.8

```
In [34]: mult_regression(['FGA','ORB','FT','AST','STL'],'MP')
         regressionmodel.summary()
```

Out[34]:

OLS Regression Results

Dep. Variable:	MP	**R-squared:**	0.870
Model:	OLS	**Adj. R-squared:**	0.869
Method:	Least Squares	**F-statistic:**	879.5
Date:	Mon, 04 Feb 2019	**Prob (F-statistic):**	1.65e-288
Time:	06:22:41	**Log-Likelihood:**	-1745.9
No. Observations:	664	**AIC:**	3504.
Df Residuals:	658	**BIC:**	3531.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	4.3695	0.263	16.595	0.000	3.852	4.887
FGA	1.6941	0.063	27.025	0.000	1.571	1.817
ORB	2.1247	0.206	10.323	0.000	1.721	2.529
FT	-1.3742	0.203	-6.770	0.000	-1.773	-0.976
AST	0.4461	0.125	3.559	0.000	0.200	0.692
STL	4.1945	0.422	9.931	0.000	3.365	5.024

Omnibus:	3.485	**Durbin-Watson:**	1.724
Prob(Omnibus):	0.175	**Jarque-Bera (JB):**	3.630
Skew:	0.092	**Prob(JB):**	0.163
Kurtosis:	3.312	**Cond. No.**	27.9

Appendix F (1): Simple Model [One Predictor]

Appendix F (2): Adding Predictor [Two Predictors]

Appendix F (3): Backward Selection [Five Predictors]