



International Soccer Player Ratings

DSC 530 FINAL PROJECT

EDRIS SAFARI

Fédération Internationale de Football Association(FIFA) publishes Player rating data set which contains over 85 features about 18000 players. The features and players are updated annually and players are rated from 1(worst) to 100(best) . The data set can be used for various purposes, one of which is in the gaming industry.

The goal of this study is to explore the 2019 publication of this dataset using statistical and programatic techniques.

Dataset description

- Single csv file
- 18209 observations
- 89 features
- 34 features recording each player's skills with rating of 1(bad) to 100(good)
- 'Potential', 'Overall' features show the ratings based on various features such as skills, age, etc.. Their value ranges from 1(bad) to 5(good)
- Other features of interest:
 - Age, weight, height

Data Preparation

- Import Dataset
- Replace 'lbs' from the weight and convert to integer
- Convert height from “ft’inch” to Inches as type integer place in new column(Height_Inch)
- Compute experience in years from signing date

Question

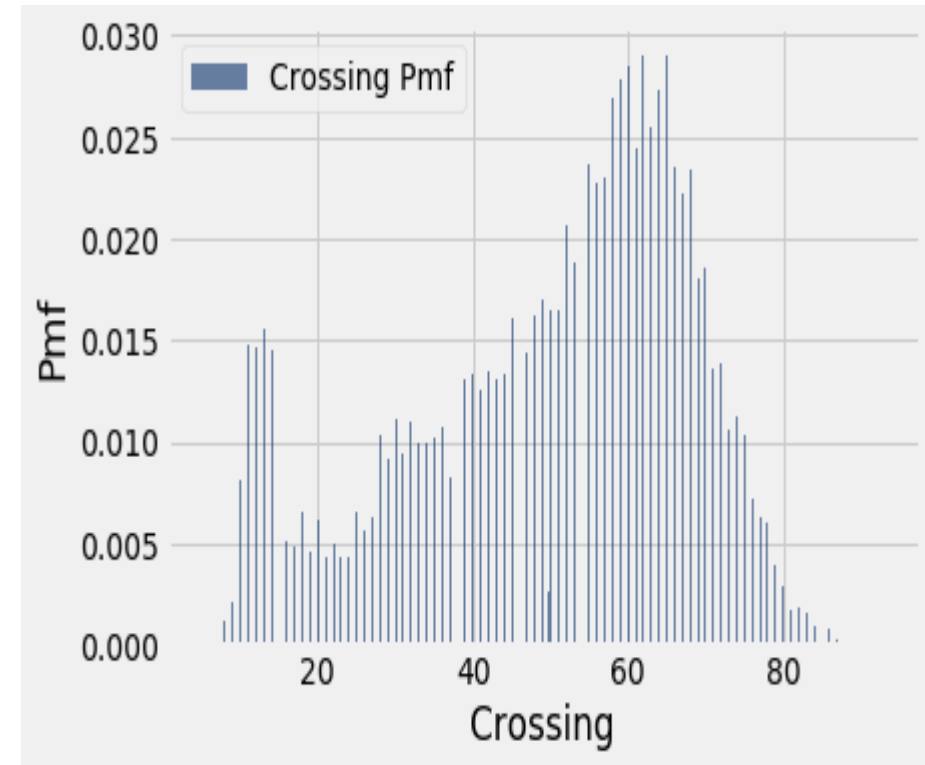
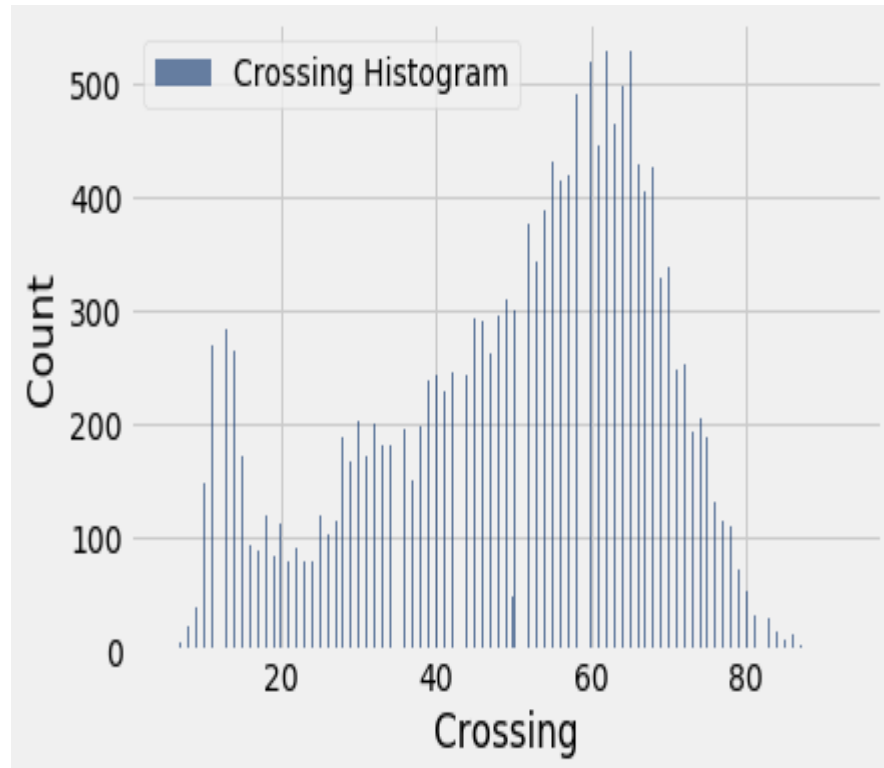
- Is the overall rating of players govern by their skill set.

Data Exploration-Best in skill

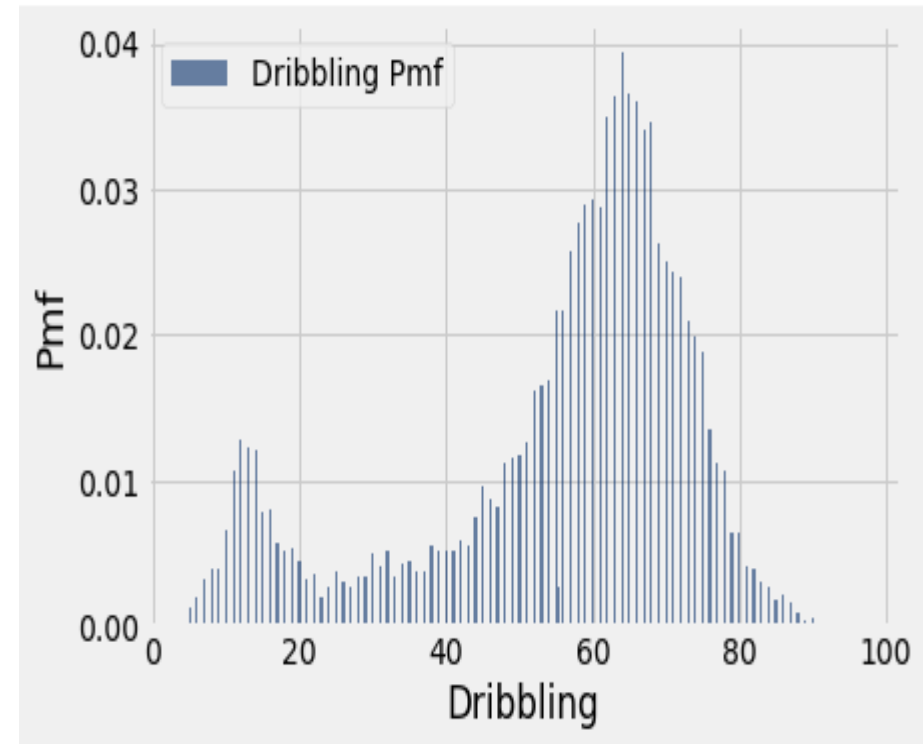
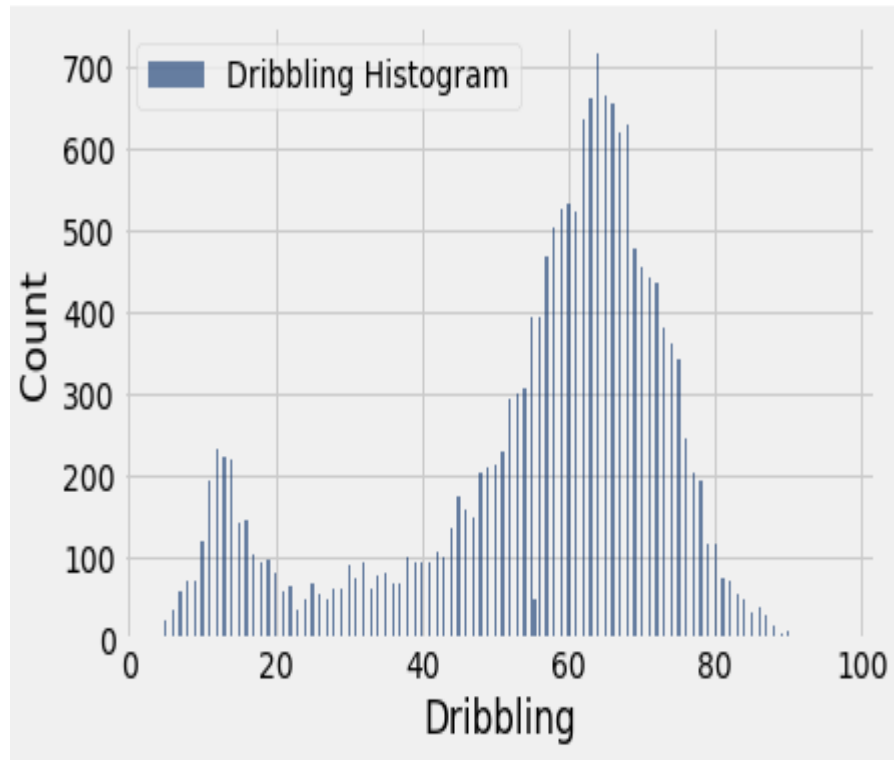
Best Crossing : K. De Bruyne from Belgium
Best Finishing : L. Messi from Argentina
Best HeadingAccuracy : Naldo from Brazil
Best ShortPassing : L. Modrić from Croatia
Best Volleys : E. Cavani from Uruguay
Best Dribbling : L. Messi from Argentina
Best Curve : Quaresma from Portugal
Best FKAccuracy : L. Messi from Argentina
Best LongPassing : T. Kroos from Germany
Best BallControl : L. Messi from Argentina
Best Acceleration : Douglas Costa from Brazil
Best SprintSpeed : K. Mbappé from France
Best Agility : Neymar Jr from Brazil
Best Reactions : Cristiano Ronaldo from Portugal
Best Balance : Bernard from Brazil
Best ShotPower : Cristiano Ronaldo from Portugal
Best Jumping : Cristiano Ronaldo from Portugal
Best Stamina : N. Kanté from France
Best Strength : A. Akinfenwa from England
Best LongShots : L. Messi from Argentina
Best Aggression : B. Pearson from England
Best Interceptions : N. Kanté from France
Best Positioning : Cristiano Ronaldo from Portugal
Best Vision : L. Messi from Argentina
Best Penalties : M. Balotelli from Italy
Best Composure : L. Messi from Argentina
Best Marking : A. Barzagli from Italy
Best StandingTackle : G. Chiellini from Italy
Best SlidingTackle : Sergio Ramos from Spain
Best GK Diving : De Gea from Spain
Best GK Handling : J. Oblak from Slovenia
Best GK Kicking : M. Neuer from Germany
Best GK Positioning : G. Buffon from Italy
Best GK Reflexes : De Gea from Spain

Best player in each skill category. Lionel Messi is the best overall player and is best rated in more skills than others.

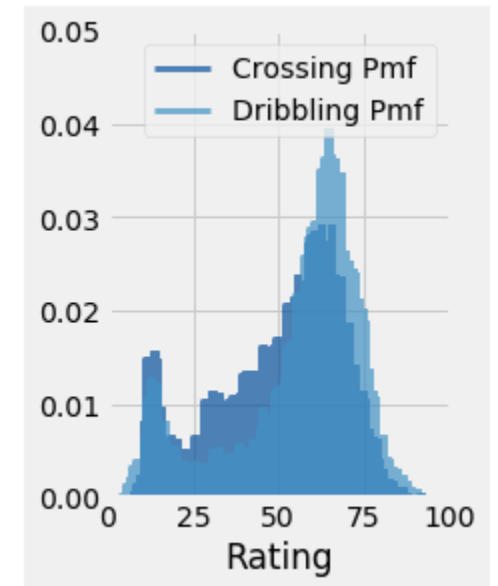
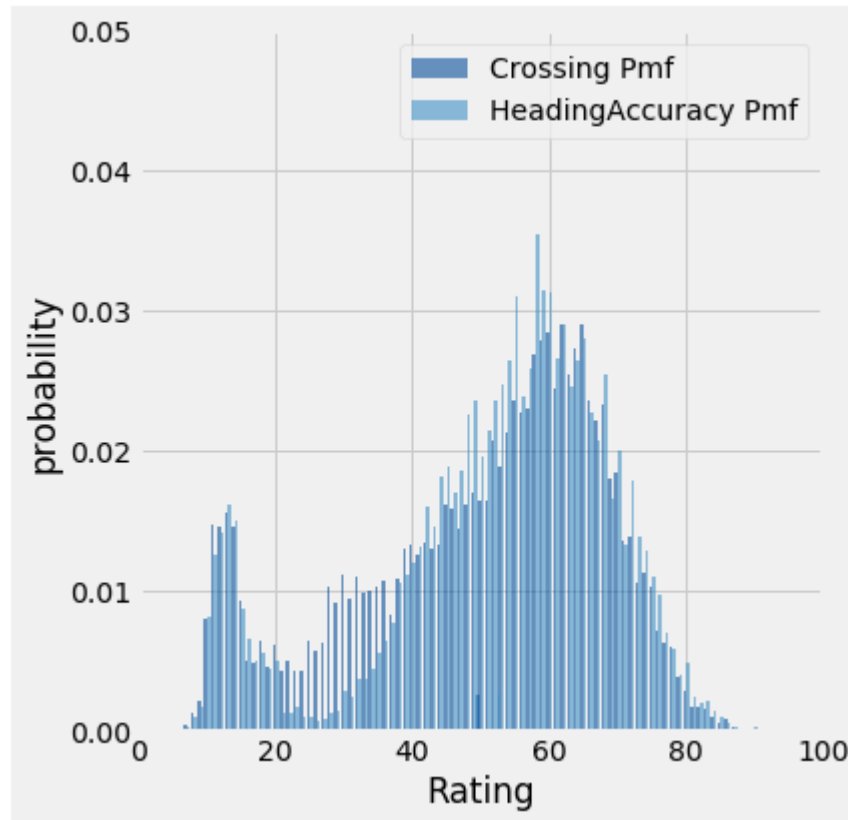
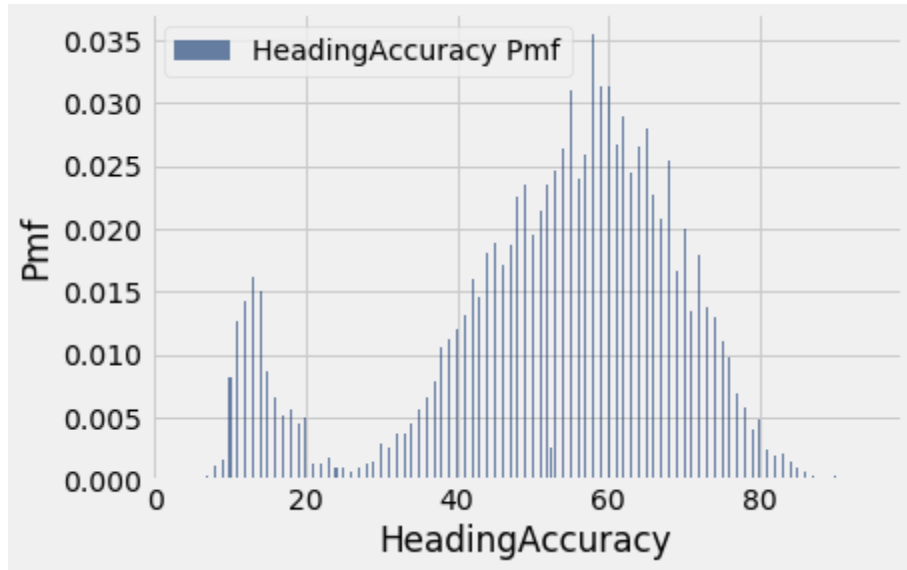
Data Exploration- Distributions



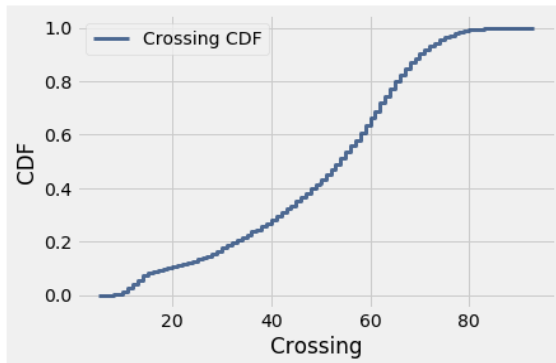
Data Exploration- Distributions



Data Exploration- Distribution comparison



Data Exploration- Distribution

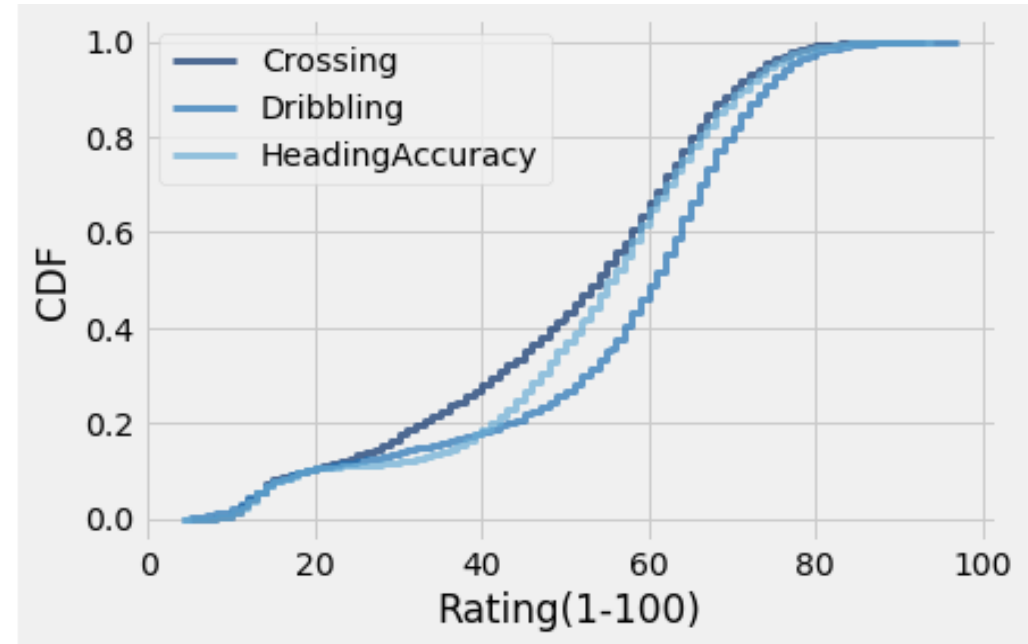


```
: cdf.Prob(40),cdf.Prob(80)  
#this means that 28 percent of the players have rating of 40 and 99 percent have rating of 80
```

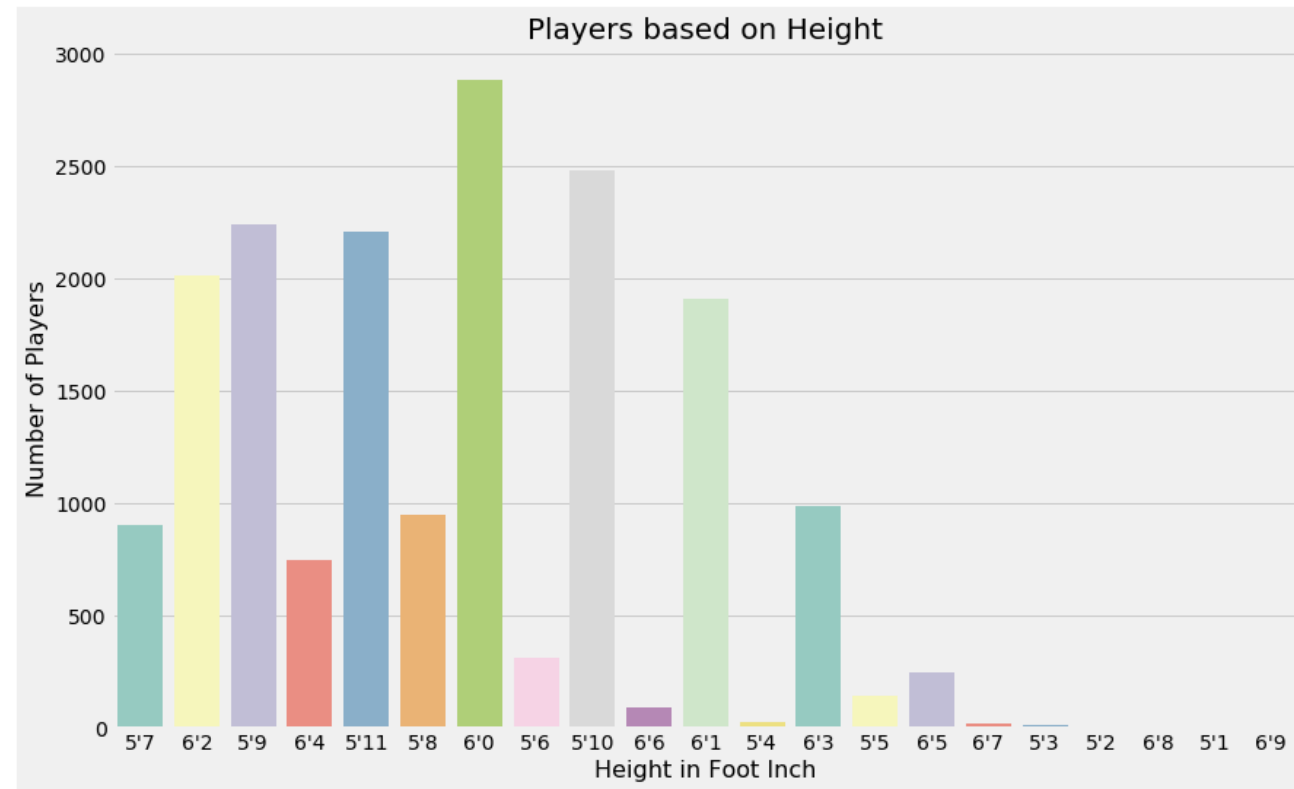
```
: (0.28280331740539355, 0.9913220190036799)
```

```
: cdf.Value(.5)  
# This means that 50 percent have rating of 54  
cdf.Prob(54)  
# this shows that.
```

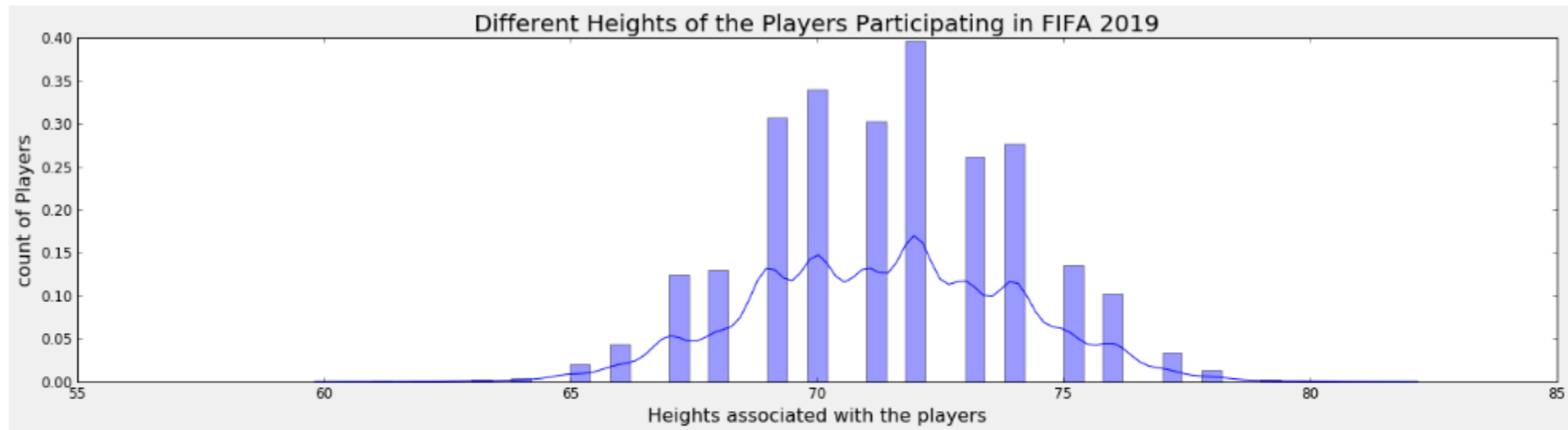
```
: 0.5113967155489647
```



Data Exploration- Distribution



Data Exploration- Distribution

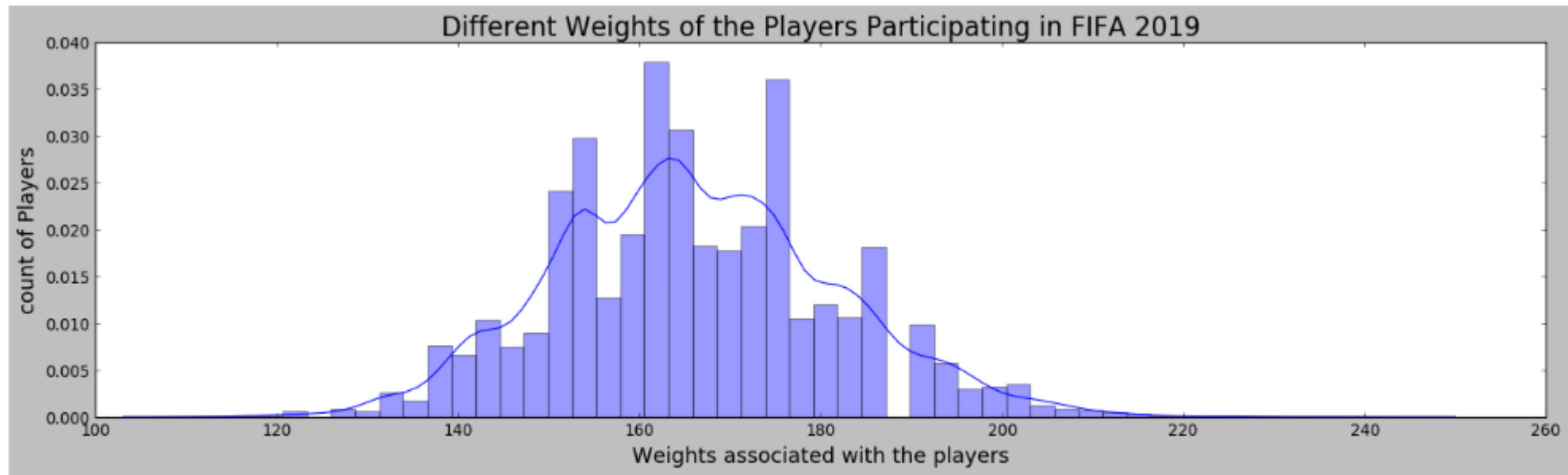


Normal distribution in above graph shows to be 71.36 as shown here

```
statistics.mean(data['Height_Inch'])
```

71.3612533729831

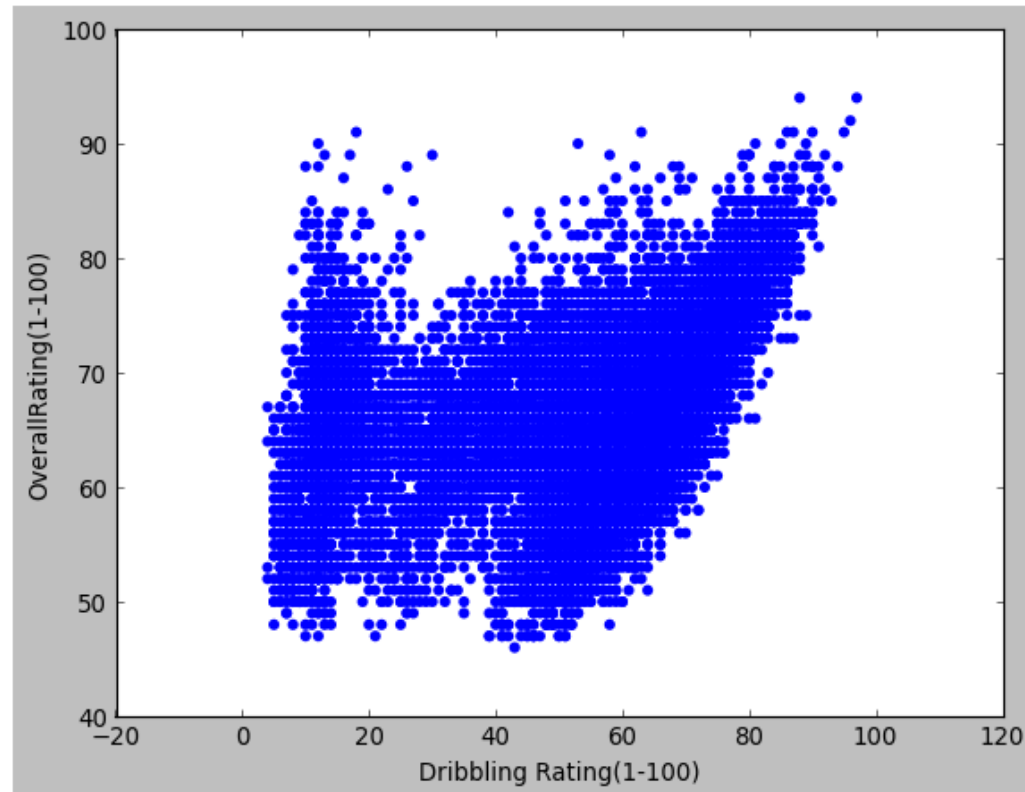
Data Exploration- Distribution



```
: statistics.mean(data['Weight'])
```

```
: 165.97912880665234
```

Data Exploration- Scatter plot



Modeling Coefficients

```
statsmodels.formula  
  
a = 'Overall ~ Cross  
= smf.ols(formula, d  
s = model.fit()  
s.summary()
```

FKAccuracy	0.0005	0.002	0.223	0.824	-0.004	0.005
BallControl	0.1459	0.005	28.653	0.000	0.136	0.156
Acceleration	0.0325	0.004	8.332	0.000	0.025	0.040
SprintSpeed	0.0314	0.004	8.624	0.000	0.024	0.038
Agility	-0.0086	0.003	-3.073	0.002	-0.014	-0.003
Reactions	0.2833	0.004	75.435	0.000	0.276	0.291
ShotPower	0.0146	0.003	5.468	0.000	0.009	0.020
Jumping	0.0012	0.002	0.594	0.552	-0.003	0.005
Stamina	0.0030	0.002	1.278	0.201	-0.002	0.008
Strength	0.0434	0.002	19.147	0.000	0.039	0.048
Penalties	0.0008	0.003	0.297	0.767	-0.005	0.006
Composure	0.1164	0.003	38.078	0.000	0.110	0.122
Marking	0.0343	0.002	13.851	0.000	0.029	0.039
StandingTackle	0.0055	0.002	2.202	0.028	0.001	0.010
GKDivining	0.0744	0.006	12.536	0.000	0.063	0.086
GKHandling	0.0768	0.006	12.790	0.000	0.065	0.089
GKKicking	0.0334	0.006	6.035	0.000	0.023	0.044
GKPositioning	0.0602	0.006	11.628	0.000	0.057	0.080

Modeling Correlations

Acceleration	0.0292	0.004	7.771	0.000	0.022	0.037
SprintSpeed	0.0306	0.004	8.442	0.000	0.023	0.038
Reactions	0.2824	0.004	75.374	0.000	0.275	0.290
ShotPower	0.0141	0.003	5.345	0.000	0.009	0.019
Jumping	-0.0001	0.002	-0.056	0.955	-0.004	0.004
Stamina	0.0023	0.002	0.984	0.325	-0.002	0.007
Strength	0.0450	0.002	20.304	0.000	0.041	0.049
Penalties	0.0003	0.003	0.124	0.902	-0.005	0.006
Composure	0.1154	0.003	37.927	0.000	0.109	0.121
Marking	0.0346	0.002	13.967	0.000	0.030	0.039
StandingTackle	0.0063	0.002	2.533	0.011	0.001	0.011
GKDividing	0.0744	0.006	12.540	0.000	0.063	0.086
GKHandling	0.0770	0.006	12.832	0.000	0.065	0.089
GKKicking	0.0337	0.006	6.085	0.000	0.023	0.045
GKPositioning	0.0683	0.006	11.633	0.000	0.057	0.080
GKReflexes	0.0775	0.006	13.162	0.000	0.066	0.089
Omnibus:	45.524	Durbin-Watson:	1.680			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	57.150			
Skew:	0.027	Prob(JB):	3.89e-13			
Kurtosis:	3.269	Cond. No.	2.89e+03			

```
Remove features with negative coefficient
port statsmodels.formula.api as smf
```

```
rmula = 'Overall ~ Crossing+Finishing+HeadingAccuracy+ShortPassing+Curve+FKAccuracy+BallControl+Acceleration+SprintSpeed+Reactions'
del = smf.ols(formula, data=data)
sults = model.fit()
sults.summary()
```

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 2.89e+03. This might indicate that there are strong multicollinearity or other numerical problems.

Modeling

```
] : #Split dataset into test and train datasets
    from sklearn.model_selection import train_test_split
    # Create test and train data set. Training data set is 80% of the total and test is 20%
    X_train, X_test, y_train, y_test = train_test_split(Model_Independent_Variables, Model_Dependent_Variable, test_size=0.2, random_s

    #One Hot Encoding
    X_train = pd.get_dummies(X_train)
    X_test = pd.get_dummies(X_test)
    print(X_test.shape, X_train.shape)
    print(y_test.shape, y_train.shape)

(3642, 34) (14565, 34)
(3642,) (14565,)
```

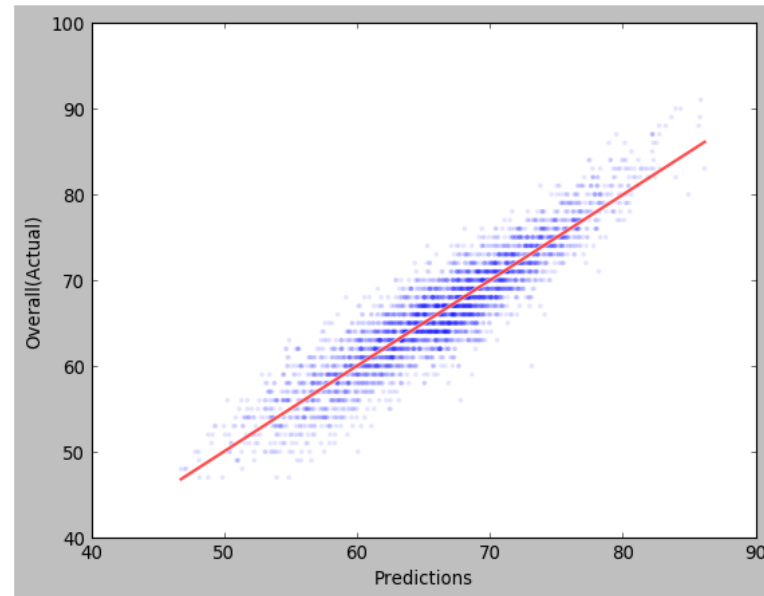
```
: #Apply Linear Regression
    from sklearn.linear_model import LinearRegression
    model = LinearRegression()
    model.fit(X_train, y_train)
    predictions = model.predict(X_test)

    #Finding the r2 score and root mean squared error
    from sklearn.metrics import r2_score, mean_squared_error
    print('r2 score: ' + str(r2_score(y_test, predictions)))
    print('RMSE : ' + str(np.sqrt(mean_squared_error(y_test, predictions))))

r2 score: 0.8514783362139567
RMSE : 2.6485407085072126
```

Modeling – All Skills

```
: thinkplot.Scatter(predictions, y_test, color='blue', alpha=0.1, s=10)  
thinkplot.Plot(fit_xs, fit_ys, color='white', linewidth=3)  
thinkplot.Plot(fit_xs, fit_ys, color='red', linewidth=2)  
thinkplot.Config(xlabel="Predictions",  
                  ylabel='Overall(Actual)',  
                  legend=False)
```



Modeling – Less Skills

```
# reduce features
```

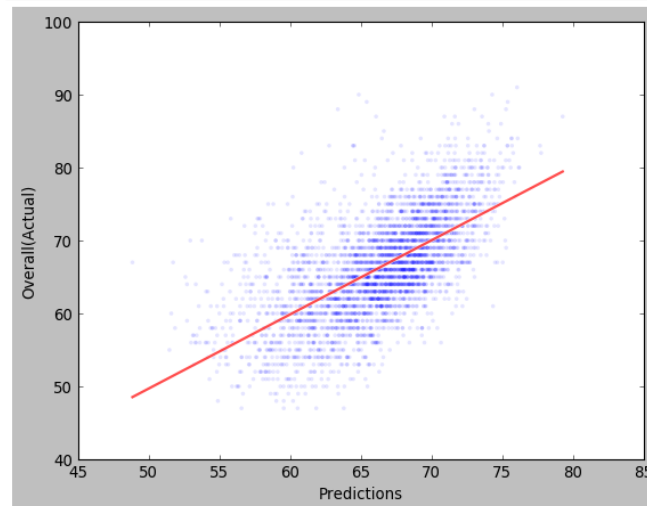
```
Model_Independent_Variables = data[["Crossing", "Finishing", "HeadingAccuracy", "ShortPassing", "Volleys", "Dribbling", "Curve", "LongPa
```

```
inter, slope = LeastSquares(predictions, y_test)  
inter, slope
```

```
(-1.1423557380204983, 1.0164462486903685)
```

```
fit_xs, fit_ys = FitLine(predictions, inter, slope)
```

```
thinkplot.Scatter(predictions, y_test, color='blue', alpha=0.1, s=10)  
thinkplot.Plot(fit_xs, fit_ys, color='white', linewidth=3)  
thinkplot.Plot(fit_xs, fit_ys, color='red', linewidth=2)  
thinkplot.Config(xlabel="Predictions",  
                  ylabel='Overall(Actual)',  
                  legend=False)
```



Conclusions

You need all the skills you can get to be number one.

The skills rating had multicollinearity. We can eliminate them using a backward elimination technique by evaluating the coefficients, R^2 and p-value at each iteration.

The prediction model was run with all skills and a subset. The prediction suffered when some skills were removed.

The dataset and its analysis can answer many other questions related to whether a player should be signed, what rate should the compensated, how long a contract to offer, can be a starter or better be on the bench until certain point in the game.