

# Predict FPPG on FanDuel: OLS Regression, Aug. 2016

## loadData

In [1]:

```
import pandas as pd

xls_file = pd.ExcelFile('nfl.xls')
xls_file
```

Out[1]:

<pandas.io.excel.ExcelFile at 0x3f02fd0>

In [4]:

```
xls_file.sheet_names
```

Out[4]:

[u'nfl']

In [5]:

```
df = xls_file.parse('nfl')
df.head()
```

Out[5]:

|   | Position | Player           | AvgPointsPerGame | Salary | SalaryPerPointsPerGame | Yards | Touchdowns | Turnovers | Played |
|---|----------|------------------|------------------|--------|------------------------|-------|------------|-----------|--------|
| 0 | QB       | A Rodgers        | 19.33            | 9000   | 465.597517             | 3821  | 31         | 8         | 16     |
| 1 | QB       | A Luck           | 20.41            | 8700   | 426.261637             | 1881  | 15         | 12        | 7      |
| 2 | QB       | D Brees          | 21.01            | 8700   | 414.088529             | 4870  | 32         | 11        | 15     |
| 3 | QB       | R Wilson         | 21.52            | 8500   | 394.981413             | 4024  | 34         | 8         | 16     |
| 4 | QB       | B Roethlisberger | 20.34            | 8500   | 417.895772             | 3938  | 21         | 16        | 12     |

## cleanData

In [21]:

```
df_clean = df.dropna()
df_clean.head(1)
df_clean.ix[:,1:8].head(10)
df_clean[['Position', 'Player', 'AvgPointsPerGame', 'Salary', 'SalaryPerPointsPerGame', 'Yards', 'Touchdowns',
          'Turnovers',
          'Played']].corr().ix[:,1:7].head(10)
```

Out[21]:

|                        | Salary    | SalaryPerPointsPerGame | Yards     | Touchdowns | Turnovers | Played    |
|------------------------|-----------|------------------------|-----------|------------|-----------|-----------|
| AvgPointsPerGame       | 0.659661  | -0.807686              | 0.727751  | 0.664923   | 0.648305  | 0.027106  |
| Salary                 | 1.000000  | -0.163150              | 0.390614  | 0.320289   | 0.325995  | 0.007989  |
| SalaryPerPointsPerGame | -0.163150 | 1.000000               | -0.548603 | -0.532759  | -0.474905 | -0.106049 |
| Yards                  | 0.390614  | -0.548603              | 1.000000  | 0.933032   | 0.837973  | 0.295199  |
| Touchdowns             | 0.320289  | -0.532759              | 0.933032  | 1.000000   | 0.778434  | 0.321167  |
| Turnovers              | 0.325995  | -0.474905              | 0.837973  | 0.778434   | 1.000000  | -0.002512 |
| Played                 | 0.007989  | -0.106049              | 0.295199  | 0.321167   | -0.002512 | 1.000000  |

## buildModel

In [22]:

```
import statsmodels.api as sm

X = df_clean[['Salary', 'SalaryPerPointsPerGame', 'Yards', 'Touchdowns', 'Turnovers', 'Played']]
y = df_clean[['AvgPointsPerGame']]

X1 = sm.add_constant(X)
est = sm.OLS(y, X1).fit()

est.summary()
```

Out[22]:

OLS Regression Results

|                          |                  |                            |          |
|--------------------------|------------------|----------------------------|----------|
| <b>Dep. Variable:</b>    | AvgPointsPerGame | <b>R-squared:</b>          | 0.975    |
| <b>Model:</b>            | OLS              | <b>Adj. R-squared:</b>     | 0.967    |
| <b>Method:</b>           | Least Squares    | <b>F-statistic:</b>        | 127.4    |
| <b>Date:</b>             | Thu, 11 Aug 2016 | <b>Prob (F-statistic):</b> | 7.31e-15 |
| <b>Time:</b>             | 14:37:33         | <b>Log-Likelihood:</b>     | -23.831  |
| <b>No. Observations:</b> | 27               | <b>AIC:</b>                | 61.66    |
| <b>Df Residuals:</b>     | 20               | <b>BIC:</b>                | 70.73    |
| <b>Df Model:</b>         | 6                |                            |          |
| <b>Covariance Type:</b>  | nonrobust        |                            |          |

|                               | coef    | std err | t       | P> t  | [95.0% Conf. Int.] |
|-------------------------------|---------|---------|---------|-------|--------------------|
| <b>const</b>                  | 11.9447 | 1.553   | 7.693   | 0.000 | 8.706 15.184       |
| <b>Salary</b>                 | 0.0018  | 0.000   | 11.769  | 0.000 | 0.001 0.002        |
| <b>SalaryPerPointsPerGame</b> | -0.0190 | 0.001   | -14.161 | 0.000 | -0.022 -0.016      |
| <b>Yards</b>                  | 0.0007  | 0.000   | 2.000   | 0.059 | -3.03e-05 0.001    |
| <b>Touchdowns</b>             | -0.0036 | 0.040   | -0.089  | 0.930 | -0.088 0.080       |
| <b>Turnovers</b>              | 0.0048  | 0.055   | 0.088   | 0.930 | -0.110 0.119       |
| <b>Played</b>                 | -0.1245 | 0.049   | -2.561  | 0.019 | -0.226 -0.023      |

|                       |       |                          |          |
|-----------------------|-------|--------------------------|----------|
| <b>Omnibus:</b>       | 2.213 | <b>Durbin-Watson:</b>    | 1.798    |
| <b>Prob(Omnibus):</b> | 0.331 | <b>Jarque-Bera (JB):</b> | 1.621    |
| <b>Skew:</b>          | 0.597 | <b>Prob(JB):</b>         | 0.445    |
| <b>Kurtosis:</b>      | 2.878 | <b>Cond. No.</b>         | 9.91e+04 |

## showRegressors

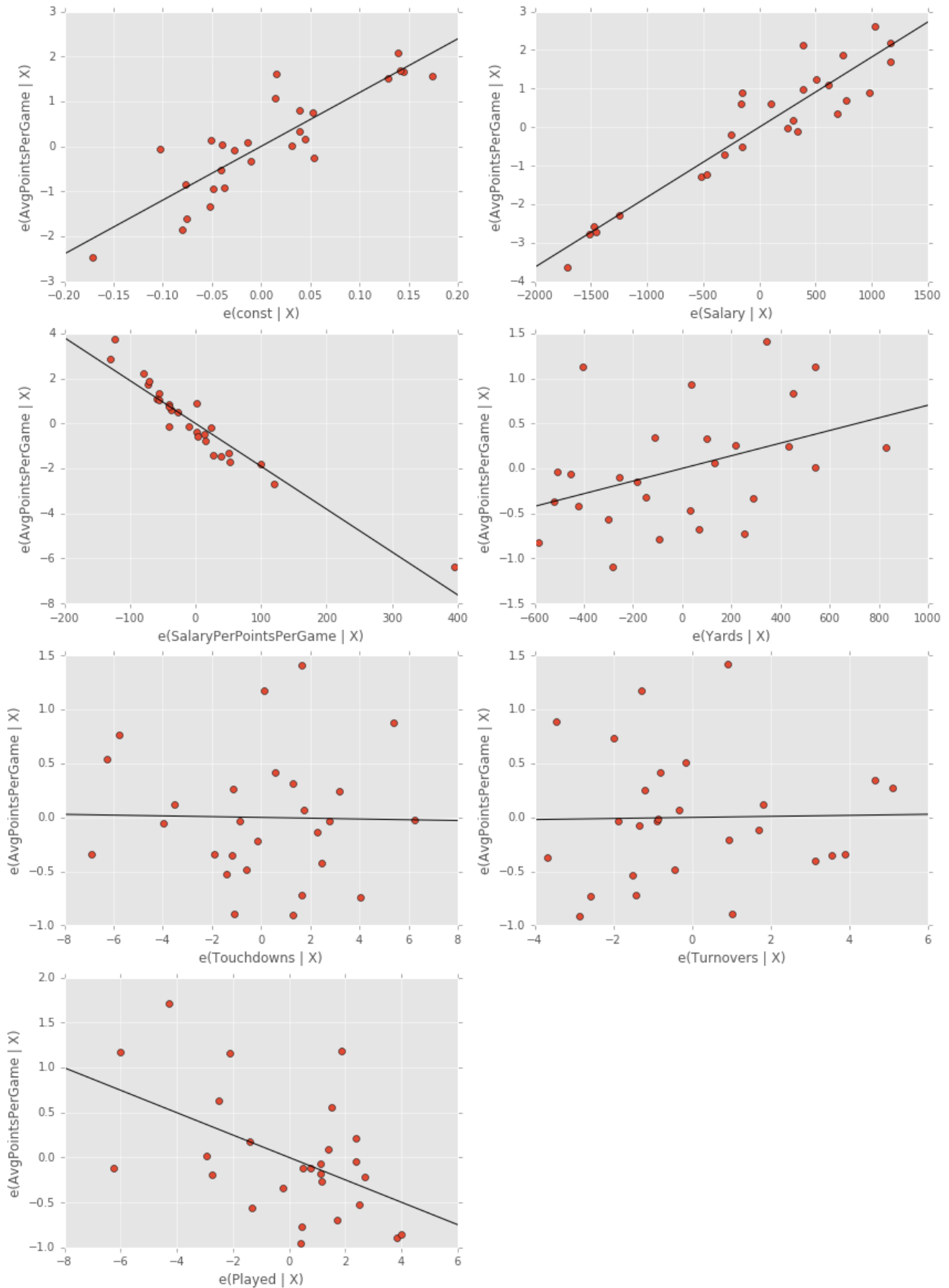
In [23]:

```
import matplotlib.pyplot as plt
import statsmodels.api as sm
%matplotlib inline

with plt.style.context('ggplot'):
    fig = plt.figure(figsize=(12,17))
    fig = sm.graphics.plot_partregress_grid(est, fig=fig)

plt.show()
```

Partial Regression Plot



## writeUp

This is an Ordinary Least Squares (OLS) regression model that results in predicted values close to the observed data. The R-squared value in the *OLS Regression Results* is a relative measure of fit, and improvement in the regression model results in proportional increases in R-squared. One pitfall of R-squared is that it can only increase as predictors are added to the regression model. R-squared value for this model is 0.975 or 97.5%, with an adjusted R-squared value of 0.967 or 96.7%.