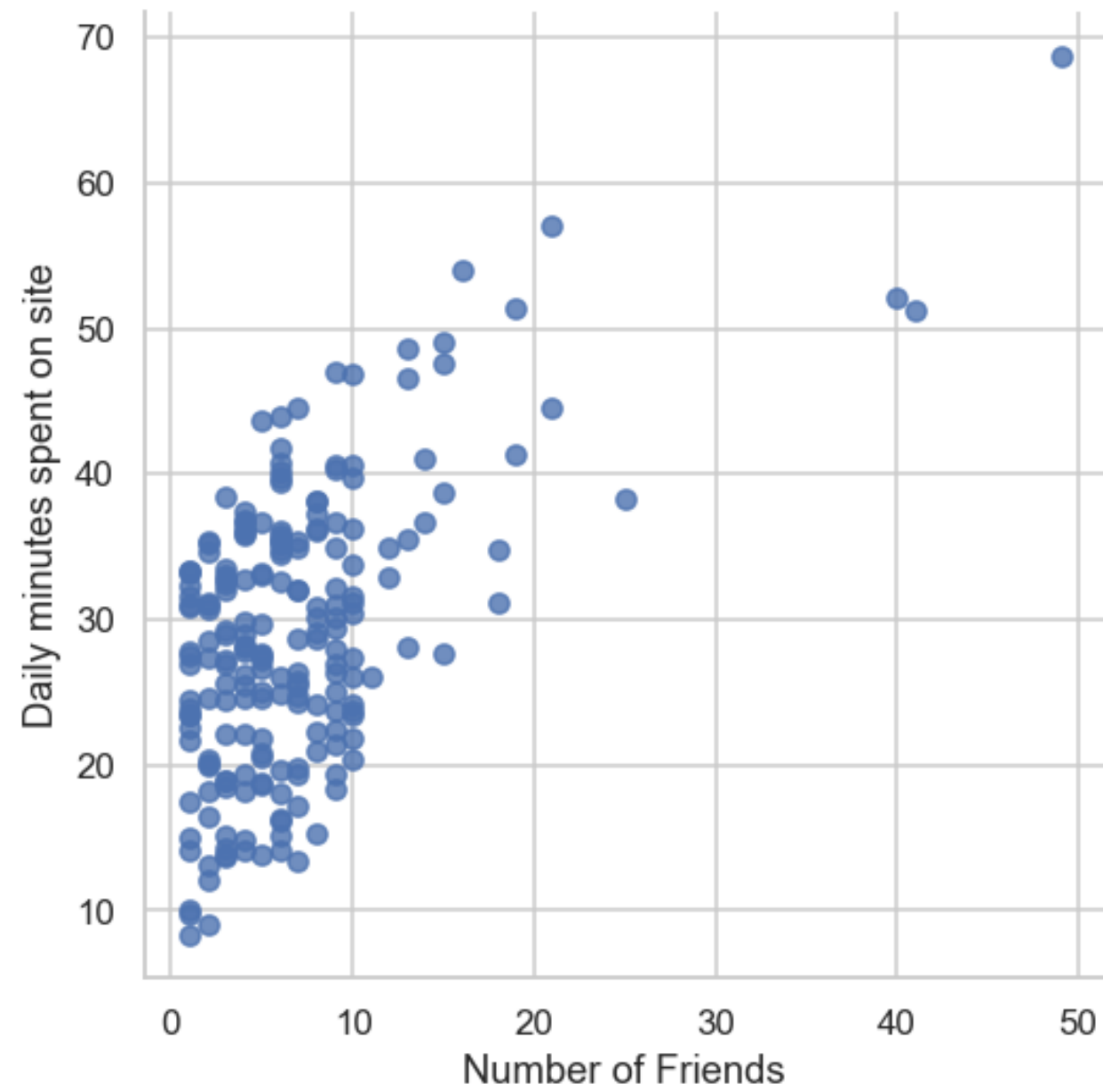


Multiple Linear Regression

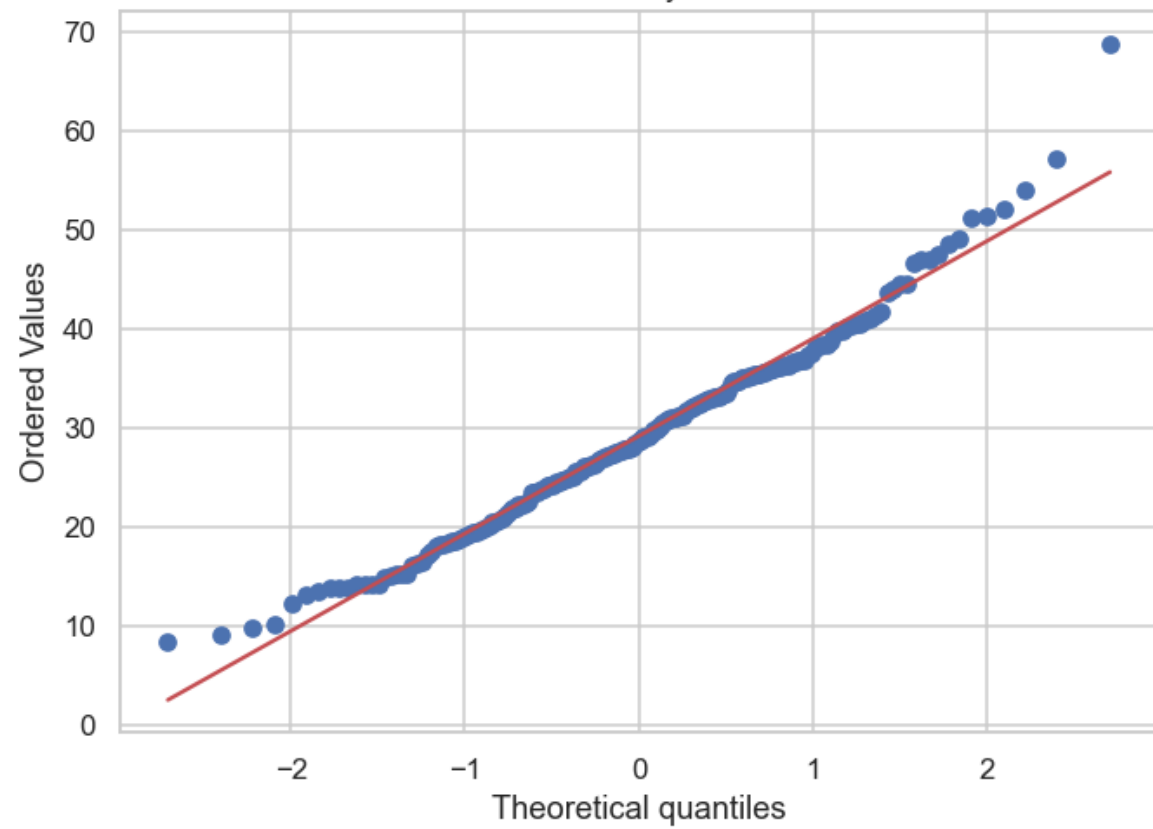
MSc Statistics

Example:

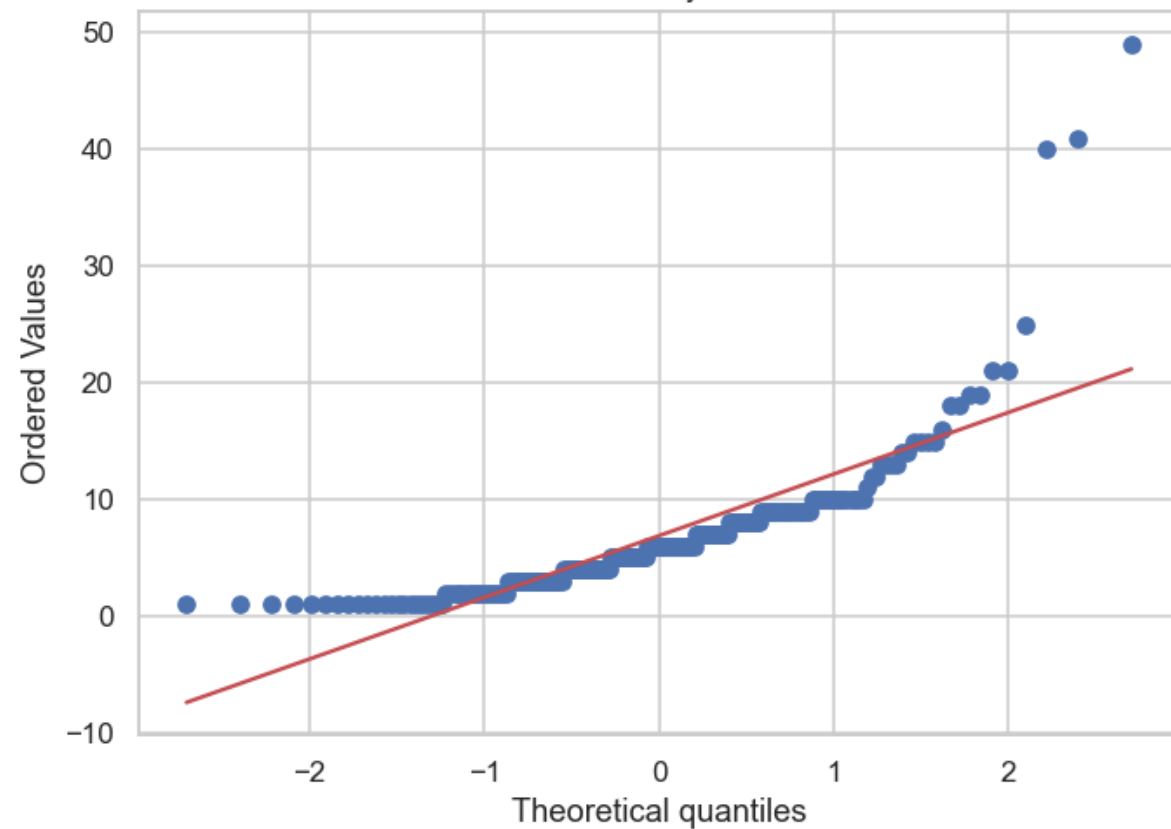
- A social media website is interested in whether the amount of time people spend on the site is related to the number of friends they have on the site.
- First we plot the data to see if any linear relationship between our two variables “Daily Minutes Spent on Site” and “Number of Friends”.
- What type of data are these two variable?
- What type of plot would we look at?
- Which is the response variable and which is the explanatory variable?

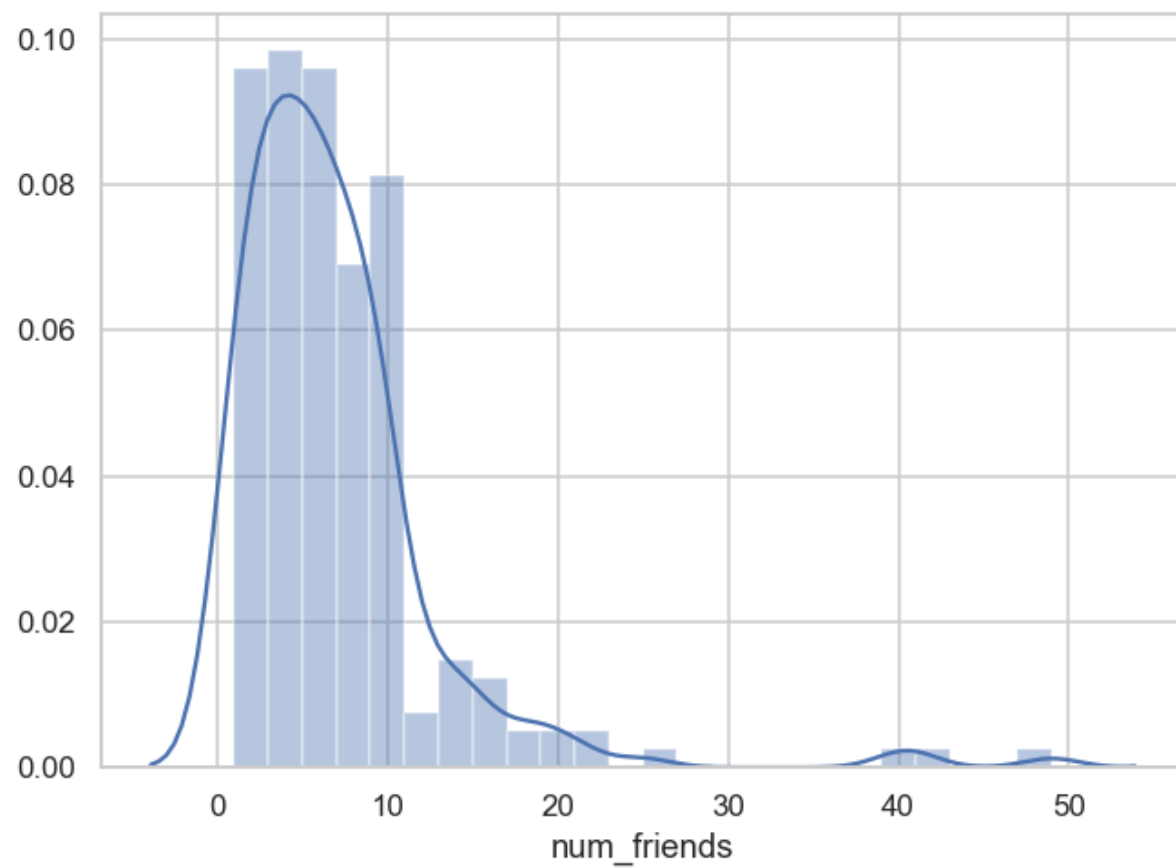
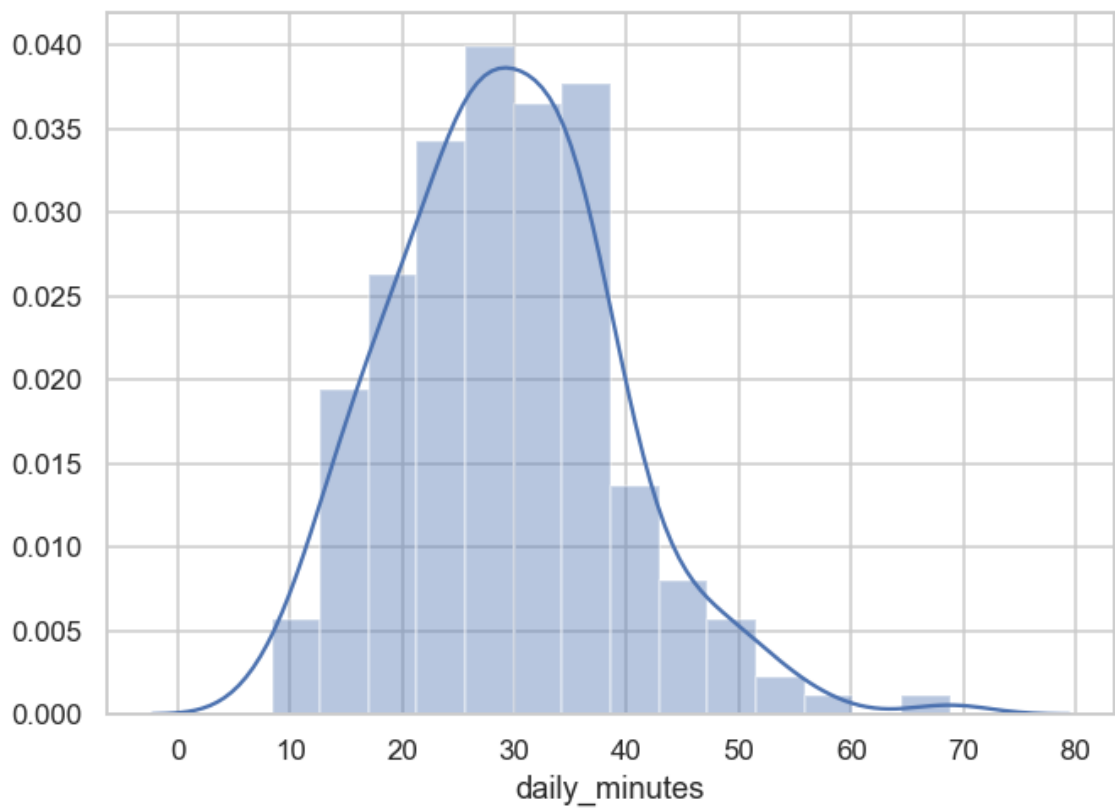


Probability Plot



Probability Plot





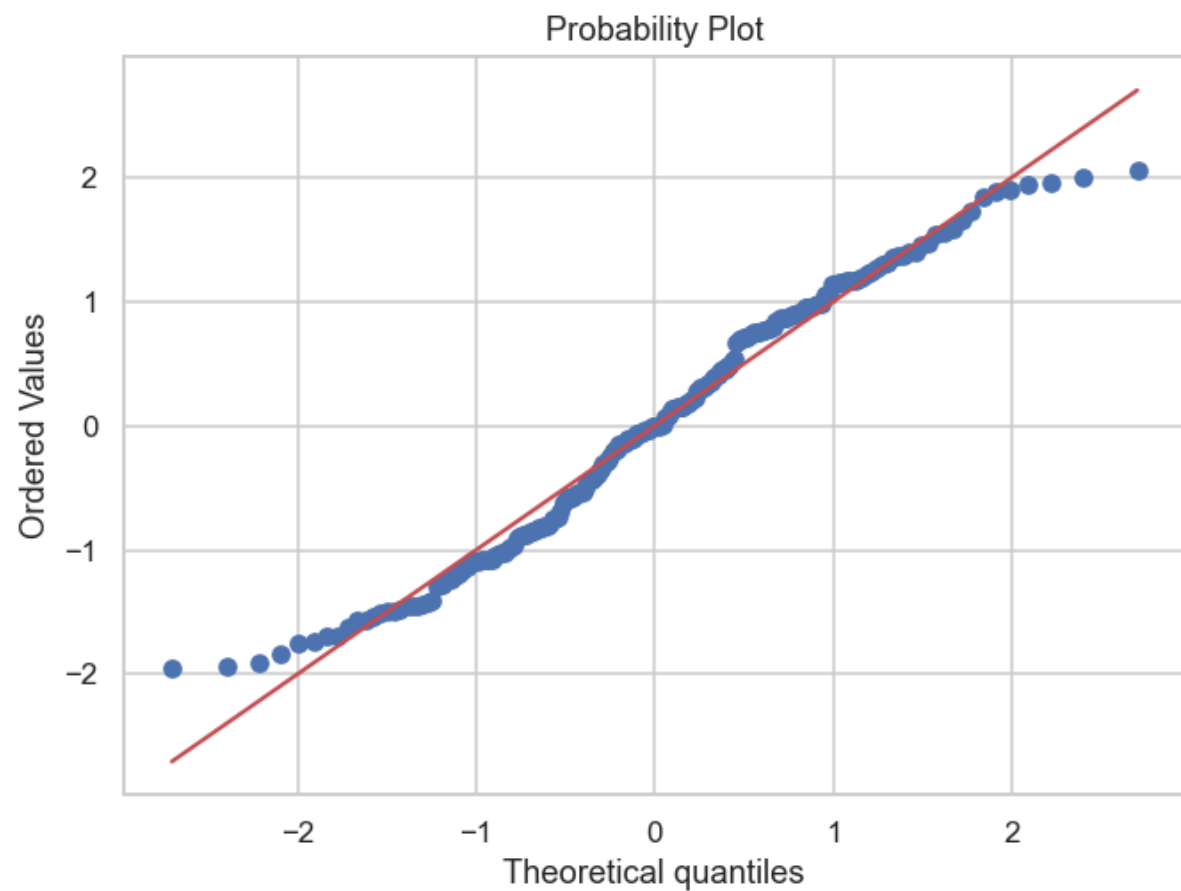
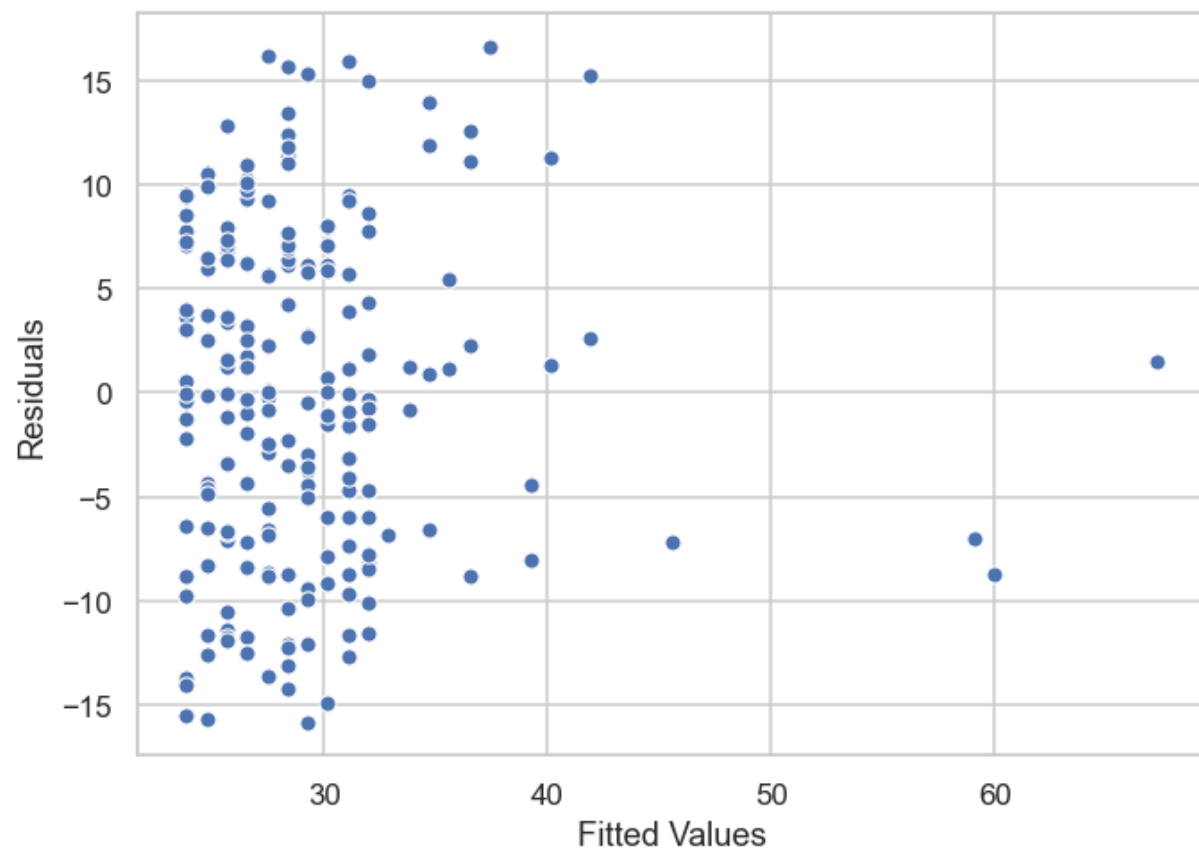
Correlation

```
#not suitable to interpret p-value from pearson correlation as x not normally distributed
corr= pearsonr(site_df["num_friends"], site_df["daily_minutes"])
corr
(0.57367921156656, 3.6768258627689324e-19)
#Looks to be a moderately strong linear relationship between time spent on site and number of friends

#Therefore can look at spearman to see if significant relationship between two variables
corrs= spearmanr(site_df["num_friends"], site_df["daily_minutes"])
corrs
SpearmanrResult(correlation=0.4118824498588302, pvalue=1.025693763528653e-09)
#H0: No association between time spent on site and number of friends
#Reject H0 and conclude evidence there is a relationship between two variables
```

Simple Linear Regression

- Decide to fit a simple linear regression model to the data, as we can infer more about the data than we can from one number obtained from correlation.
- We need to check the assumptions of the model are met before interpreting the results.
- What are the assumptions for a Simple Linear Regression?



OLS Regression Results

```

=====
Dep. Variable:          daily_minutes  R-squared:                0.329
Model:                  OLS           Adj. R-squared:            0.326
Method:                 Least Squares  F-statistic:              98.60
Date:                  Wed, 1 Jan 2020  Prob (F-statistic):        3.68e-19
Time:                  09:01:35       Log-Likelihood:           -711.76
No. Observations:      203           AIC:                     1428.
Df Residuals:          201           BIC:                     1434.
Df Model:               1
Covariance Type:       nonrobust
=====

```

```

=====
              coef      std err      t      P>|t|    [0.025    0.975]
-----
const          22.9476     0.846    27.133   0.000    21.280    24.615
num_friends     0.9039     0.091     9.930   0.000     0.724     1.083
=====

```

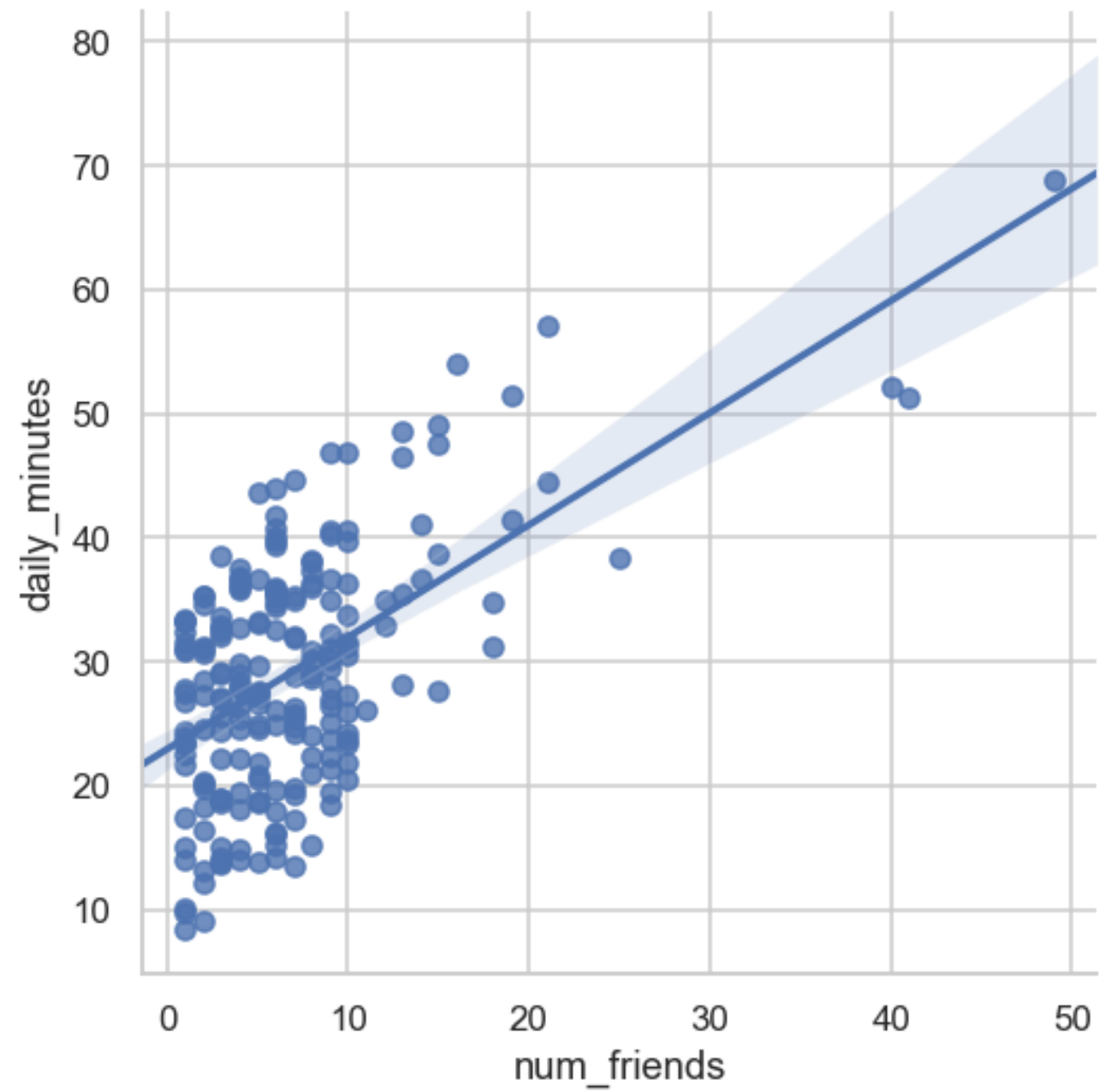
```

=====
Omnibus:          26.873  Durbin-Watson:           2.027
Prob(Omnibus):    0.000  Jarque-Bera (JB):        7.541
Skew:             0.004  Prob(JB):                0.0230
Kurtosis:         2.056  Cond. No.                 13.9
=====

```

Notes:

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



The Model for time-on-site Vs no. of friends

- Simple Linear Regression Model:
$$\text{daily_minutes} = 22.95 + 0.904 \text{ num_friends} + e$$
- The model has an R-squared of just 0.329, indicating that while it is ok, 33% of the variation in time spent on-site is being accounted for by this model.

```
np.sqrt(model.scale)  
8.102766340738382
```

- Residual standard error is 8.103
- There are clearly factors other than just number-of-friends at play.

The Model for time-on-site Vs no. of friends

Suppose additional data is available for each of your users:

1. how many hours he/she works each day, and
2. whether he/she has a Degree.

We want to use this additional data to improve our model.

We hypothesise a linear model with more independent variables:

$$\text{daily_minutes} = \beta_0 + \beta_1 \text{num_friends} + \beta_2 \text{hours_worked} + \beta_3 \text{degree} + e$$

Multiple Linear Regression Analysis

Variables:

daily_minutes (numeric - dependent)

num_friends (numeric - independent)

hours_worked (numeric – independent)

degree (non-numeric - independent)

Whether a user has a degree is not a number but we can introduce a dummy variable

that equals 1 for users with degrees and

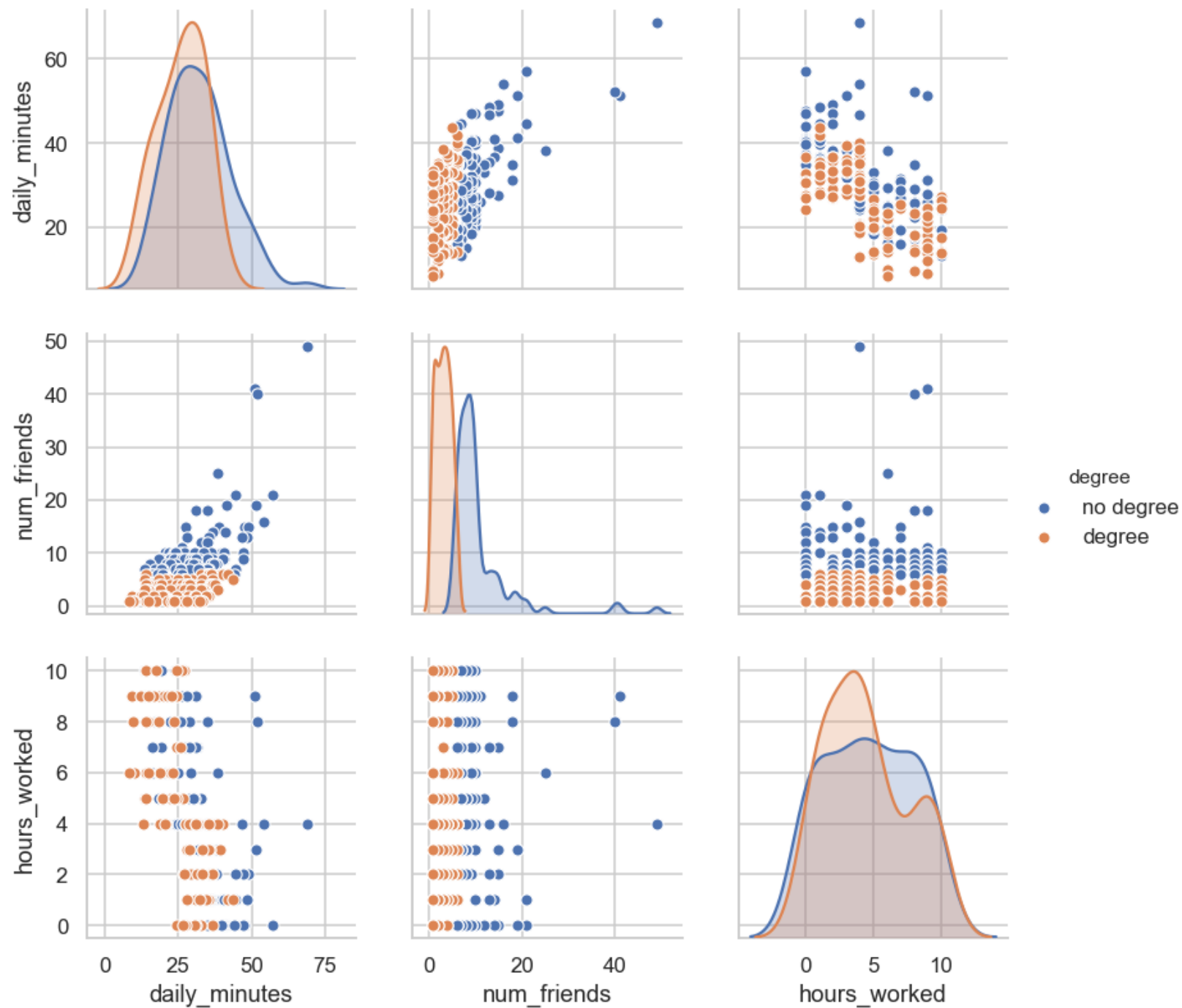
0 for users without,

after which it can be treated as numeric variable like the other variables.

Multiple Linear Regression Analysis

Assumptions:

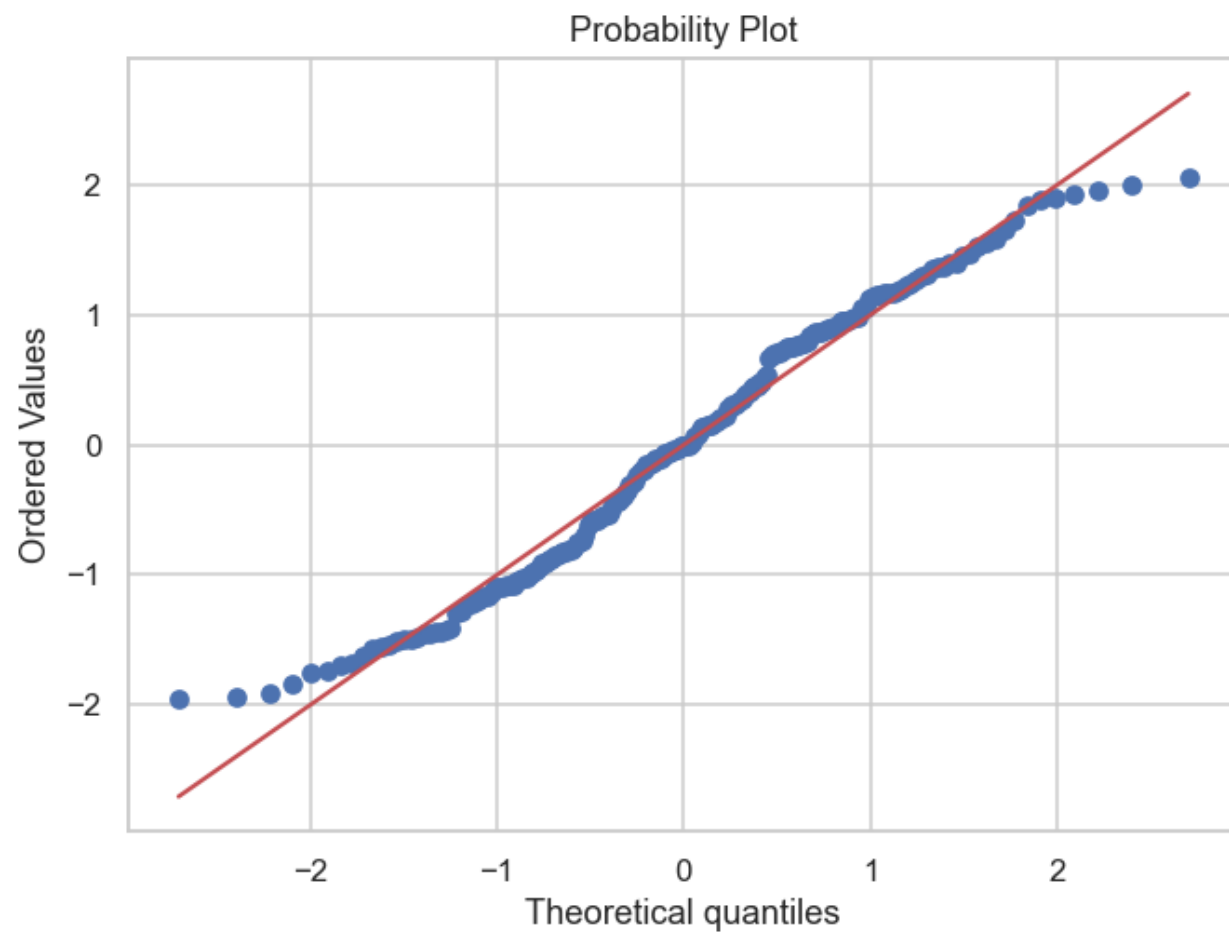
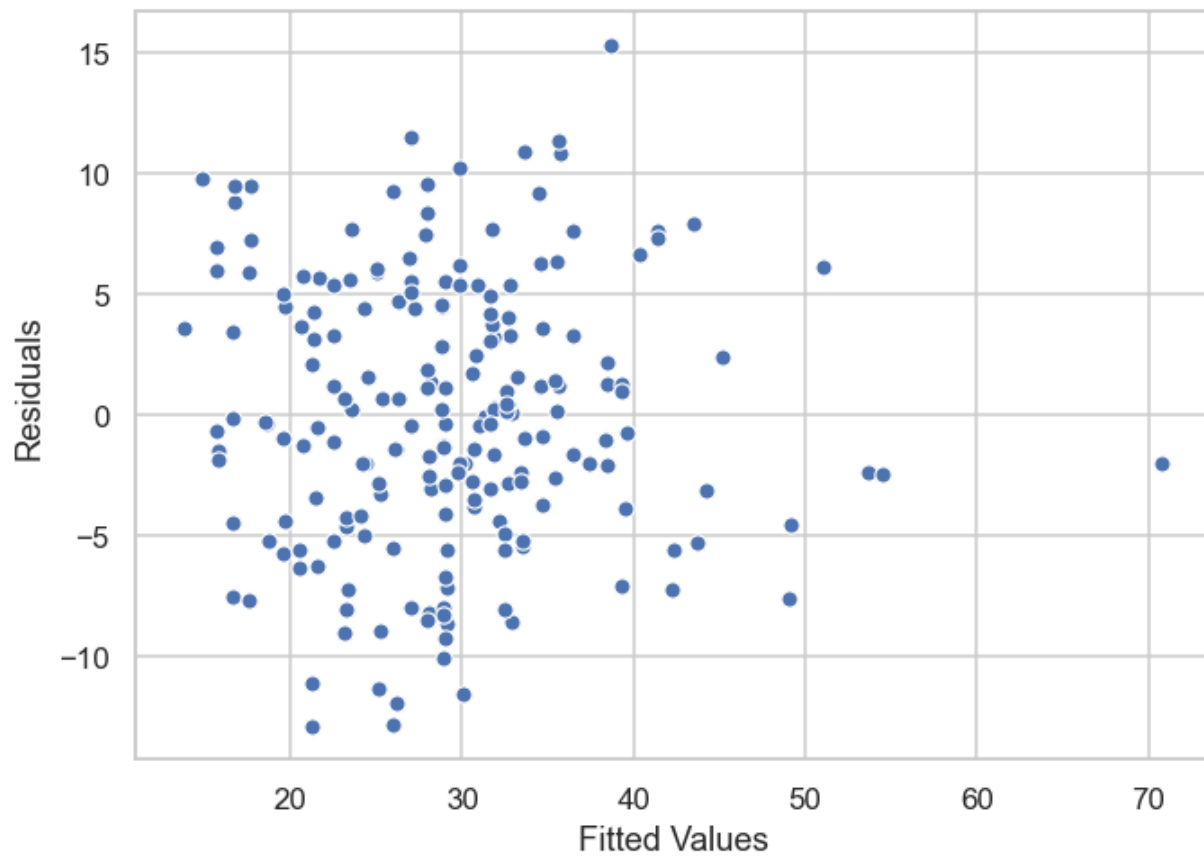
- **Linearity:** A linear relationship between the dependent variable and the independent variables.
- **Normal Distribution:** Residuals are normally distributed.
- **Constant Variance:** The variance of the residuals are similar across the values of the independent variables.
- **i.i.d:** Residuals are independently and identically distributed –random scatter.
- **No Multicollinearity**—The independent variables are not highly correlated with each other.



Running the MLR

```
X = site_df[["num_friends","hours_worked","degree"]]
X = sm.add_constant(X)
y = site_df['daily_minutes']
model_mlr = sm.OLS(y, X).fit()
```

Need to check the assumptions about the residuals first before interpreting the results.



OLS Regression Results

```

=====
Dep. Variable:          daily_minutes  R-squared:          0.680
Model:                  OLS           Adj. R-squared:      0.675
Method:                 Least Squares  F-statistic:        141.0
Date:                   Wed, 1 Jan 2020 Prob (F-statistic):  5.39e-49
Time:                   10:39:27       Log-Likelihood:     -636.61
No. Observations:      203            AIC:               1281.
Df Residuals:          199            BIC:               1294.
Df Model:               3
Covariance Type:       nonrobust
=====

```

```

=====
               coef      std err      t      P>|t|    [0.025    0.975]
-----
const          30.5790     1.190    25.692   0.000    28.232    32.926
num_friends     0.9725     0.080    12.188   0.000     0.815     1.130
hours_worked   -1.8650     0.127   -14.721   0.000    -2.115    -1.615
degree         0.9232     0.998     0.925   0.356    -1.044     2.891
=====

```

```

=====
Omnibus:          2.820  Durbin-Watson:          2.044
Prob(Omnibus):    0.244  Jarque-Bera (JB):          2.010
Skew:             0.013  Prob(JB):              0.366
Kurtosis:         2.513  Cond. No.              37.0
=====

```

	coef	std err	t	P> t	[0.025	0.975]

const	30.5790	1.190	25.692	0.000	28.232	32.926
num_friends	0.9725	0.080	12.188	0.000	0.815	1.130
hours_worked	-1.8650	0.127	-14.721	0.000	-2.115	-1.615
degree	0.9232	0.998	0.925	0.356	-1.044	2.891

Interpreting the Model:

You should think of the coefficients of the model as representing (all-else-being-equal) estimates of the impacts of each factor.

- Each additional friend corresponds to, roughly on average an extra minute spent on the site each day, keeping hours worked and degree the same
- Each additional hour in a user's workday corresponds to on average about two fewer minutes spent on the site each day, keeping number of friends and degree the same
- Having a degree is associated with spending on average an extra minute on the site each day, , keeping number of friends and hours worked the same

Fit of the model:

- We need to check if each of the independent variables are significant.
- Where do we find this information?
- What is the hypothesis being tested here?

Fit of the model:

- $H_0: \beta_i = 0$
- $H_1: \beta_i \neq 0$
- Looking at the table of coefficients results:
 - $\hat{\beta}_1$, estimate for num_friends, has t-test value = 12.188, p-value < 0.001.
Therefore, reject the null hypothesis
 - $\hat{\beta}_2$, estimate for hours_worked, has t-test value = -14.721, p-value < 0.001.
Therefore, reject the null hypothesis
 - $\hat{\beta}_3$, estimate for degree, has t-test value = 0.925, p-value = 0.356.
Therefore, fail to reject the null hypothesis

Fit of the model:

- While most of the coefficients have very small p- values (suggesting that they are indeed nonzero)
- The coefficient for “degree” has a high p-value indicating that the true coefficient is not “significantly” different from zero.
- Which makes it likely that the coefficient for “degree” is random rather than meaningful.

```
=====
Dep. Variable:          daily_minutes  R-squared:          0.680
Model:                  OLS           Adj. R-squared:     0.675
Method:                Least Squares  F-statistic:       141.0
Date:                  Wed, 1 Jan 2020 Prob (F-statistic): 5.39e-49
Time:                  10:39:27       Log-Likelihood:    -636.61
No. Observations:      203           AIC:              1281.
Df Residuals:          199           BIC:              1294.
Df Model:               3
Covariance Type:       nonrobust
```

```
model_mlr.scale
31.62947024526955
```

```
np.sqrt(model_mlr.scale)
5.6240083788406245
```

Fit of the model:

- R-squared value is 68%
- This has increased from 33% with the additional variables added to the model, suggesting a better fit than the SLR.
- Now 68% of the variability in the time spent on the site is being explained by this regression model.
- The residual standard error is 5.624, which is lower than the standard error for the SLR, again suggesting better fit.

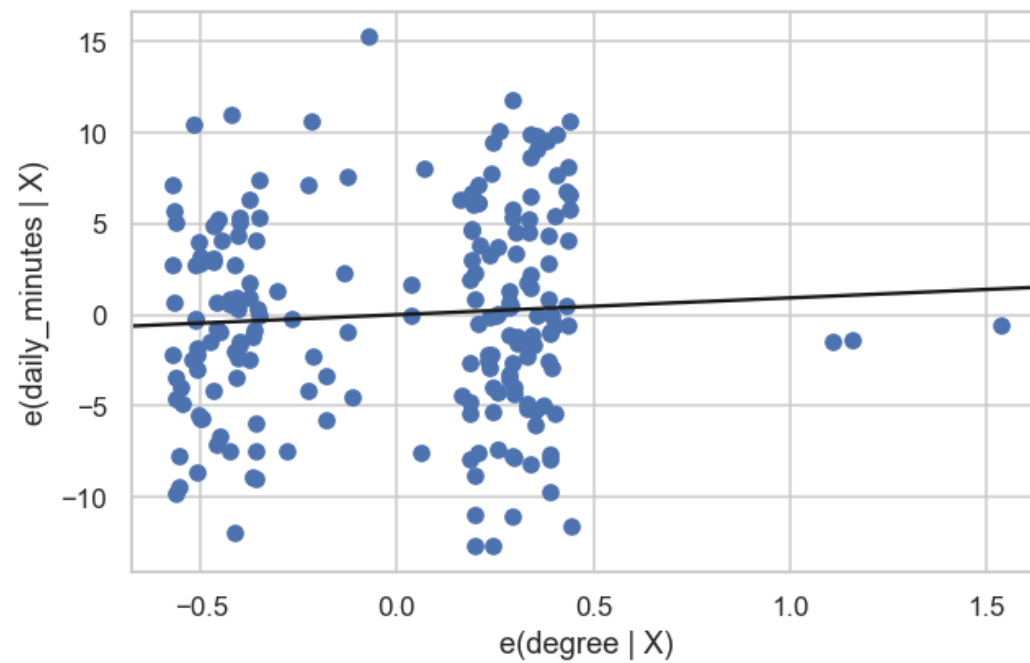
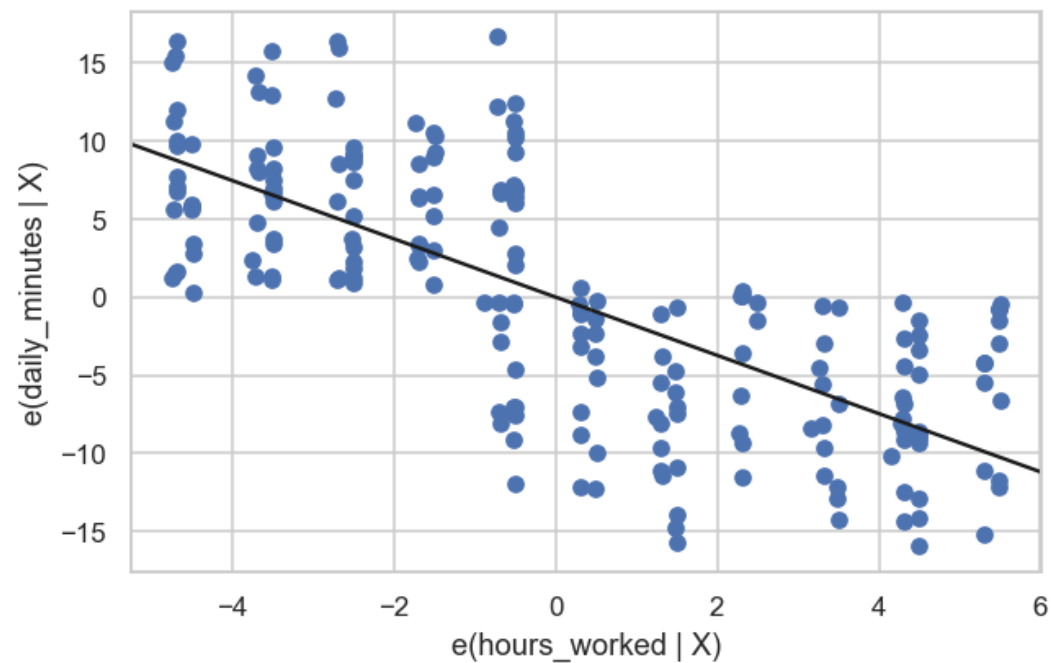
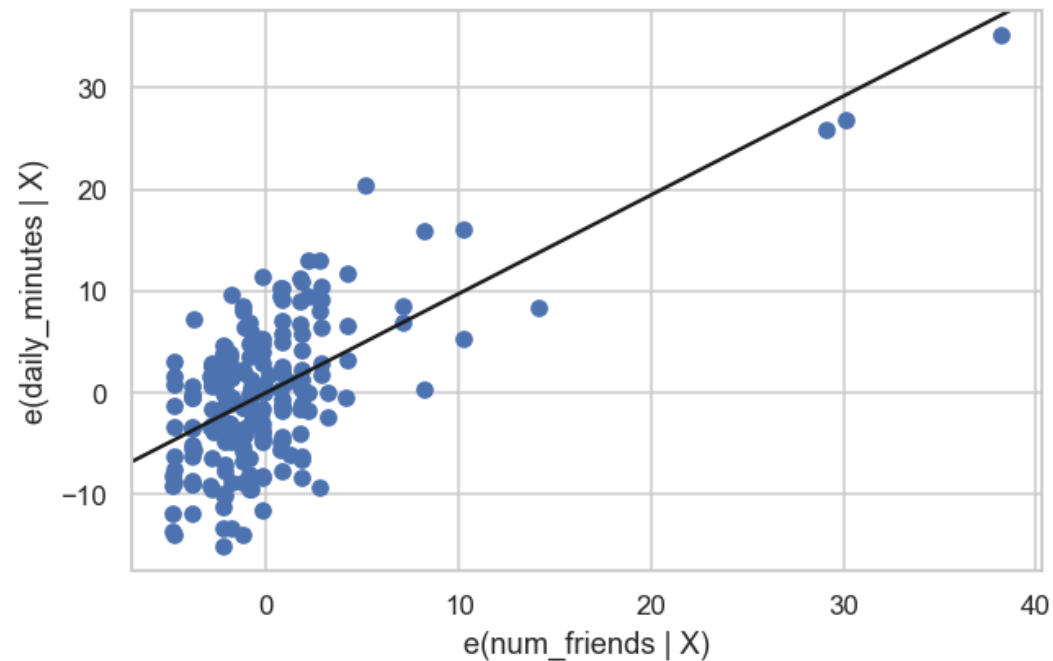
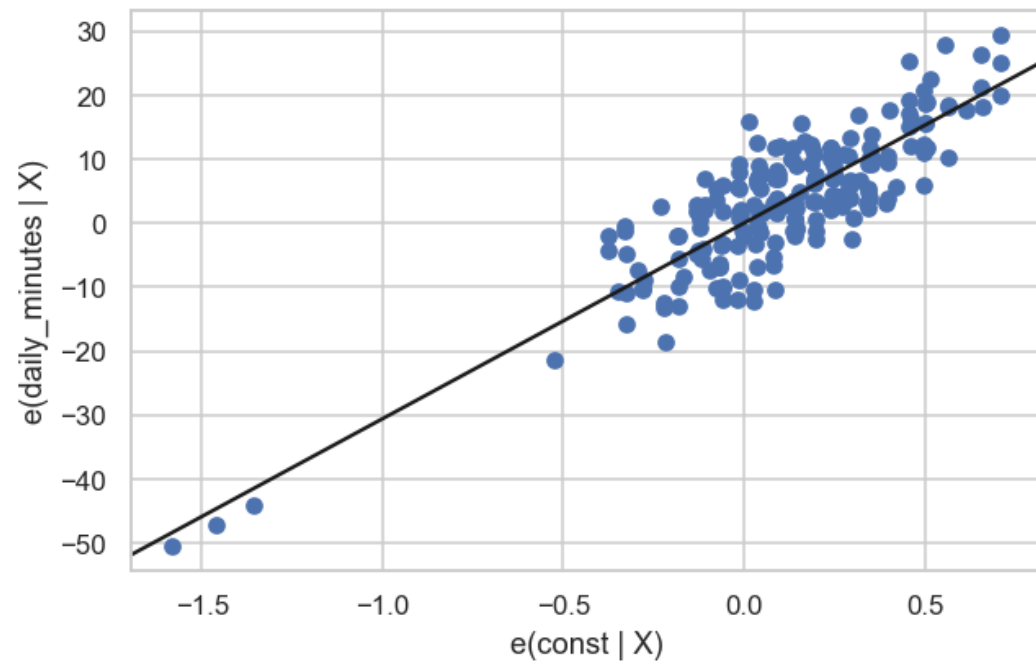
Added Variable Plots

- The Added Variable Plot helps evaluate the residuals (and coefficients) of the predictor variables in a multiple regression while holding the other variables constant.
- The Added Variable Plot shows the relationship between a response variable and a predictor, adjusting for other predictor in the model
- How to construct an added variable plot:
 1. Get residuals from regression of Y on all covariates except X_j
 2. Get residuals from regression of X_j on all other covariates
 3. Plot residuals from (1) against residuals from (2)

Creating added variable plot

```
from statsmodels.graphics.regressionplots import plot_partregress_grid  
  
plot_partregress_grid(model_mlr)  
plt.show()
```

Partial Regression Plot



Added Variable Plots

- Main use is for assessing the fit of the model. The scatter of the points and if y and x have a relationship (looking for a linear relationship in a linear regression) when considering the other variables at the same time.
- The coloured line of the plot is the slope of the line in the added variable plot is the partial regression coefficient in the full regression.
- Also useful for detecting high influence points, if there are any points quite far removed from the rest of the data points.

Unusual and Influential Data in Regression

- Outliers - An observation with large residual.
 - An observation whose dependent-variable value is unusual given its values on the predictor variables.
 - An outlier may indicate a sample peculiarity or may indicate a data entry error or other problem.
- Leverage - An observation with an extreme value on a predictor variable
 - Leverage is a measure of how far an independent variable deviates from its mean.
 - These leverage points can have an effect on the estimate of regression coefficients.
- Influence can be thought of as the product of leverage and outliers. Use cooks distance for measuring influence.

Cooks D plot

- The Cook's distance of observation i is

$$D_i = \frac{\sum_{j=1}^n (\hat{y}_j - \hat{y}_{j(i)})^2}{p \text{ MSE}}$$

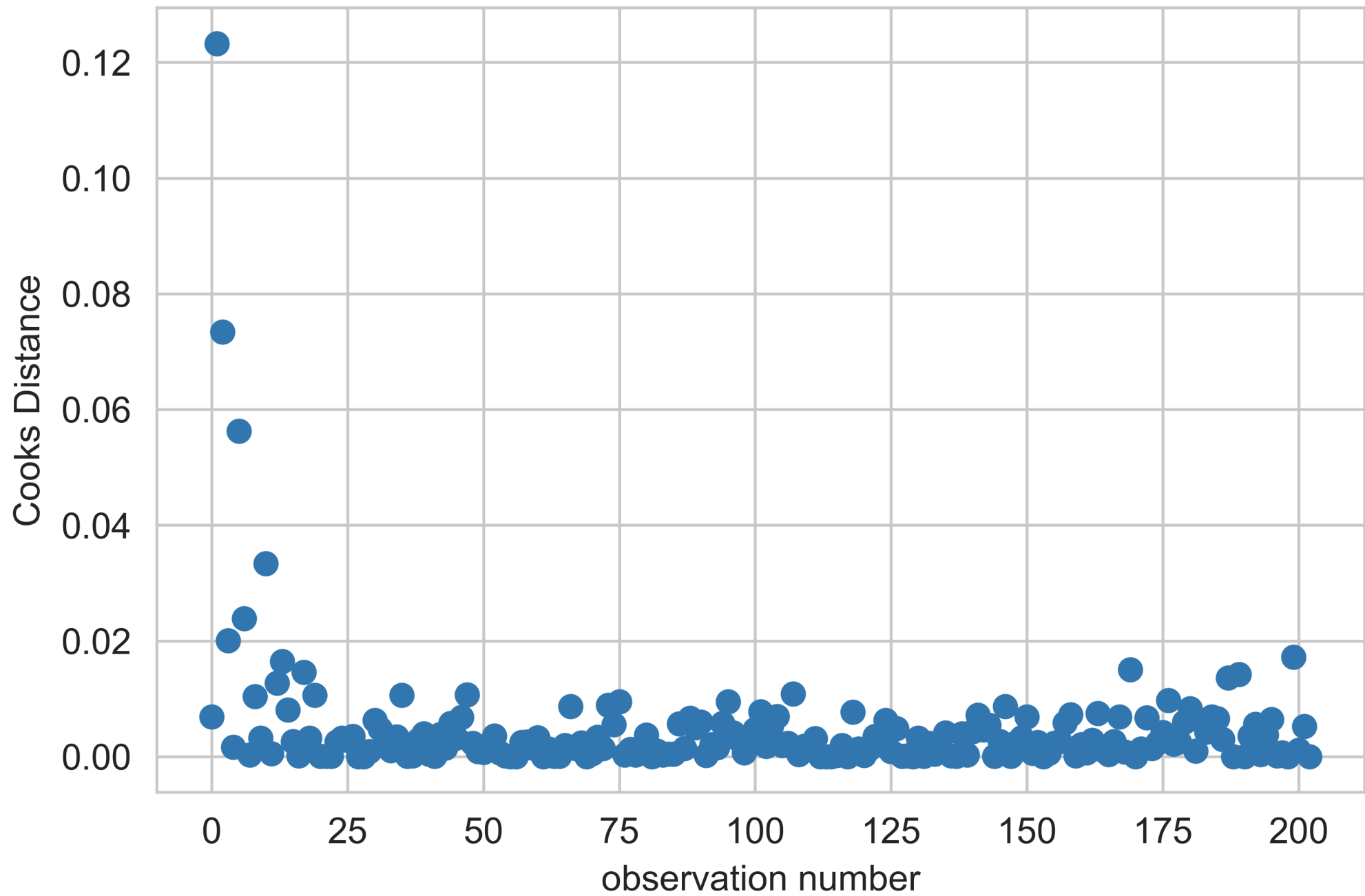
where

- \hat{y}_j is the j^{th} fitted response value.
- $\hat{y}_{j(i)}$ is the j^{th} fitted response value, where the fit does not include observation i .
- MSE is the mean squared error.
- p is the number of coefficients in the regression model.
- If removing the i^{th} observation does not effect the regression by much, D_i will be close to 0.

How to create Cook's Distance Manhattan Plot

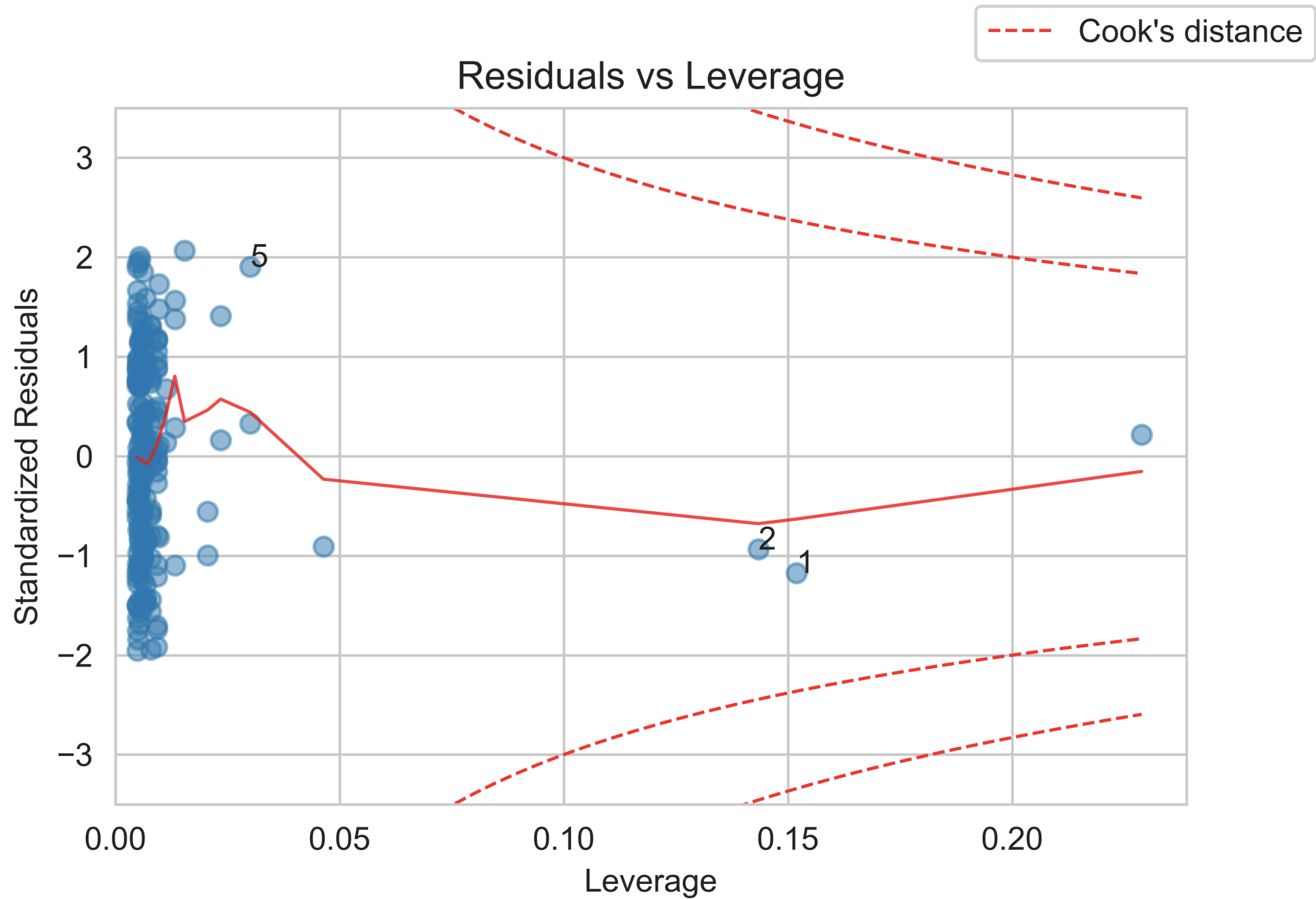
```
# cook's distance, from statsmodels internals
model_cooks = model_mlr.get_influence().cooks_distance[0]

plt.scatter(range(0,len(model_cooks)), model_cooks)
plt.xlabel('observation number')
plt.ylabel('Cooks Distance')
plt.show()
```



Outliers and Regression

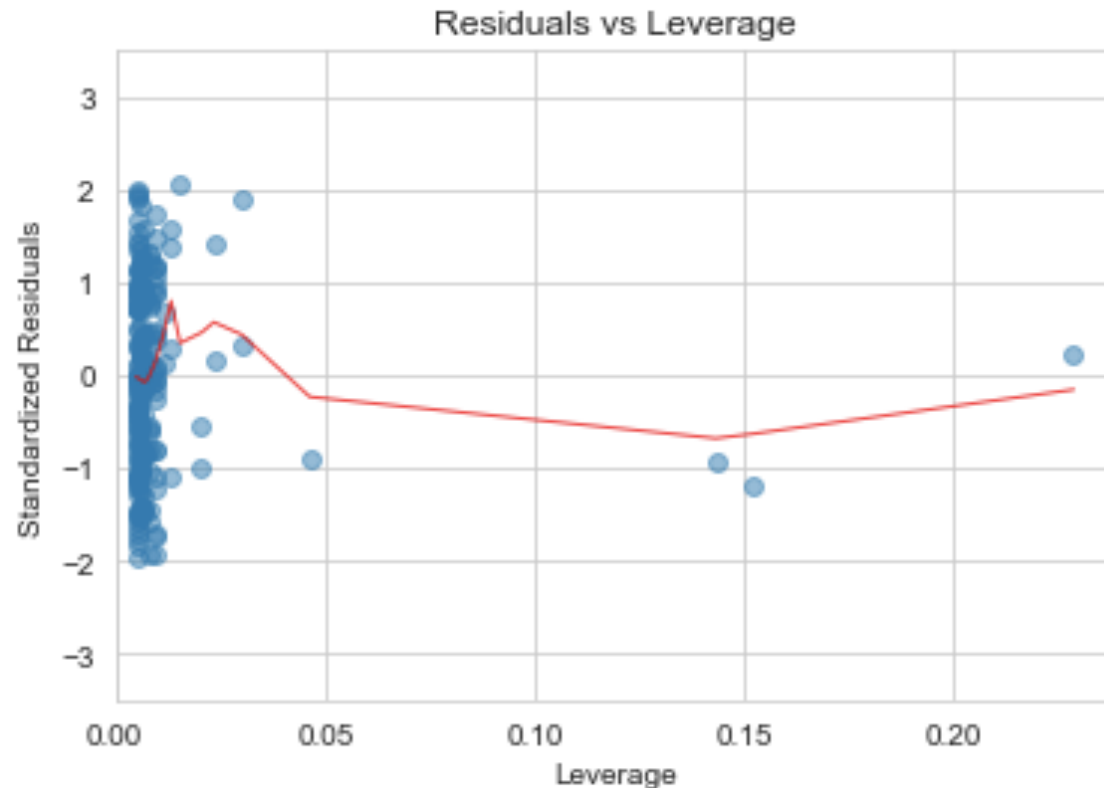
- The Residual vs Leverage plot shows contours of equal Cook's distance, for values of cook levels of 0.5 and 1.
- To find outliers or high leveraging values these values in the upper right or lower right corners, which are outside the red dashed Cook's distance line.
- These are points that would be **influential** in the model and removing them would likely noticeably alter the regression results.



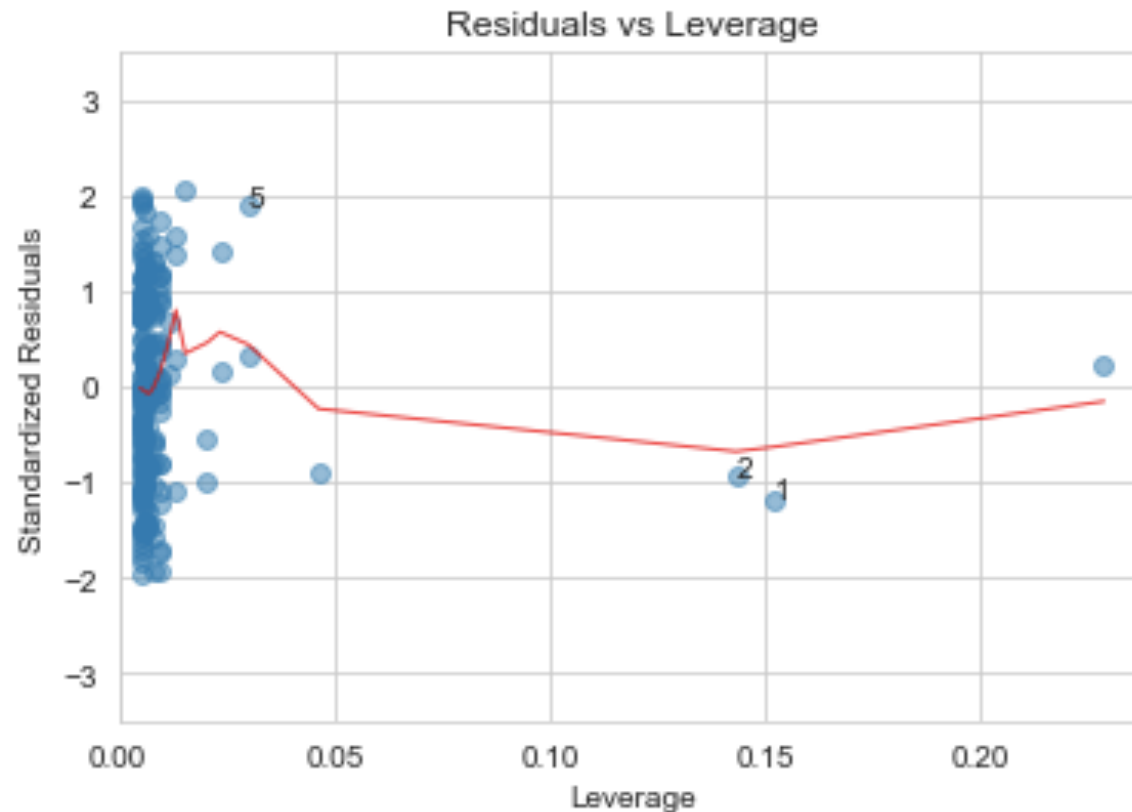
How to create Cook's Distance Plot

```
def graph(formula, x_range, label=None):  
    """  
    Helper function for plotting cook's distance lines on plot  
    """  
    x = x_range  
    y = formula(x)  
    plt.plot(x, y, label=label, lw=1, ls='--', color='red')  
  
#standardised residuals  
model_norm_residuals = model_mlr.get_influence().resid_studentized_internal  
  
# leverage, from statsmodels internals  
model_leverage = model_mlr.get_influence().hat_matrix_diag  
  
# cook's distance, from statsmodels internals  
model_cooks = model_mlr.get_influence().cooks_distance[0]
```

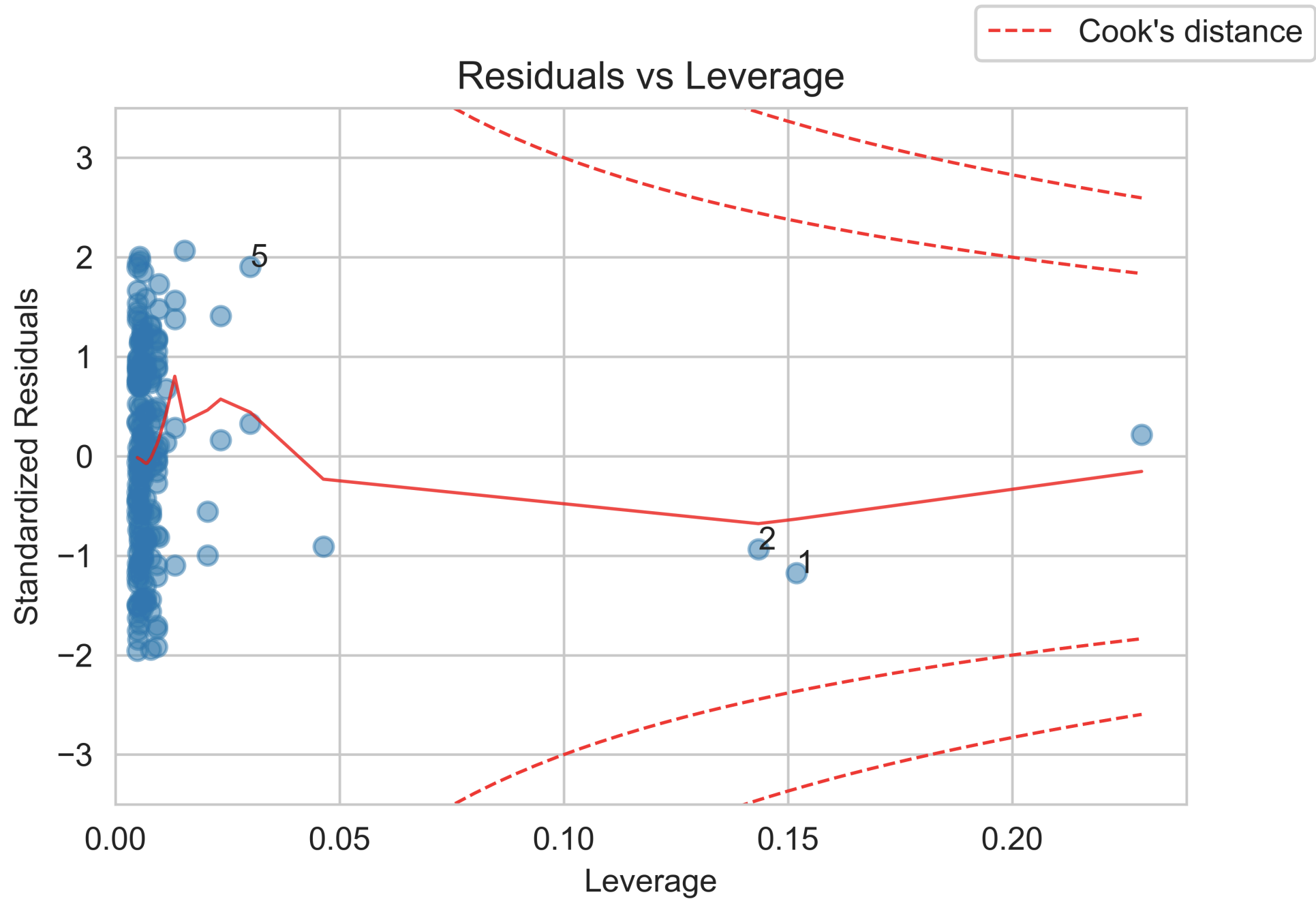
```
plot_cooks = plt.figure();
plt.scatter(model_leverage, model_norm_residuals, alpha=0.5);
sns.regplot(x=model_leverage, y=model_norm_residuals, scatter=False,
            ci=False, lowess=True, line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8});
plot_cooks.axes[0].set_xlim(0, max(model_leverage)+0.01)
plot_cooks.axes[0].set_ylim(-3.5, 3.5)
plot_cooks.axes[0].set_title('Residuals vs Leverage')
plot_cooks.axes[0].set_xlabel('Leverage')
plot_cooks.axes[0].set_ylabel('Standardized Residuals');
```



```
# annotations top 3 cooks distance points
leverage_top_3 = np.flip(np.argsort(model_cooks), 0)[:3]
for i in leverage_top_3:
    plot_cooks.axes[0].annotate(i,
                                xy=(model_leverage[i],
                                    model_norm_residuals[i]));
```



```
#create cooks distance lines
p = len(model_mlr.params) # number of model parameters
graph(lambda x: np.sqrt((0.5 * p * (1 - x)) / x),
      np.linspace(0.001, max(model_leverage), 50),
      'Cook\'s distance') # 0.5 line
graph(lambda x: np.sqrt((1 * p * (1 - x)) / x),
      np.linspace(0.001, max(model_leverage), 50)) # 1 line
graph(lambda x: -np.sqrt((0.5 * p * (1 - x)) / x),
      np.linspace(0.001, max(model_leverage), 50)) # 0.5 line
graph(lambda x: -np.sqrt((1 * p * (1 - x)) / x),
      np.linspace(0.001, max(model_leverage), 50)) # 1 line
plot_cooks.legend(loc='upper right');
plt.show()
```



Fit of the model:

- Although since $\hat{\beta}_3$, the estimate for degree, is not significant, we could look at running the model leaving this variable out.

```
X = site_df[["num_friends", "hours_worked"]]  
X = sm.add_constant(X)  
y = site_df['daily_minutes']  
model_mlr_red = sm.OLS(y, X).fit()
```

- Compare this to the previous full model, to see if this is the more appropriate model.

