```python
import pandas as pd
import numpy as np
import scipy.stats as st
import numpy as np
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
from sklearn.metrics import
mean_squared_error,mean_absolute_error,mean_absolute_percentage_error
import matplotlib.pyplot as plt
import plotly.graph_objects as go
from warnings import filterwarnings
import seaborn as sb
filterwarnings('ignore')
```

Midterm project

Congratulations! You've been hired as a data scientist at the hottest new social media startup.

Your company produces an app via which users can post short videos for anyone to view. They can also like, repost, and comment on the videos they view. The key data product is a recommendation engine that determines the order in which videos are shown to a user.

The recommendation engine has a parameter, *theta*, that affects the ordering of the videos. Recently the team of engineers that works on the recommendation engine ran it with different settings of *theta* and, for each setting, measured the amount of time users spent on the app. They have collected these measurements into a data set of 20 samples of (*theta*, *time_spent*) pairs.

Additionally, they have identified two auxiliary features (*aux1* and *aux2*) that they hypoithesize should correlate with *time_spent*. These two features are measures of time spent by users in the recent past. The engineers have not verified that the features explain *time_spent*.

(The engineers call these two features "auxiliary" because, while they might help explain *time_spent*, the engineers' ultimate interest lies in the dependence of *time_spent* on *theta*.)

Your first project at your new company is to tell the engineers which setting you think they should use for *theta*, based on the data.

1. Prepare the data

- Inspect the data. Identifiy and remove any suspicious or unusable samples.
- Put the samples in a data structure that you can work with.

```python
theta = [0.03906292, 0.05119367, 0.06004468, 0.06790036, 0.19152079,
       0.28298816, 0.294665  , 0.3578136 , 0.48352862, 0.53058676,
       0.55175137, 0.57560289, 0.59751325, 0.6375209 , 0.65241862,
       0.65633352, 0.78698546, 0.8640421 , 0.87729053, 0.94568319]
```

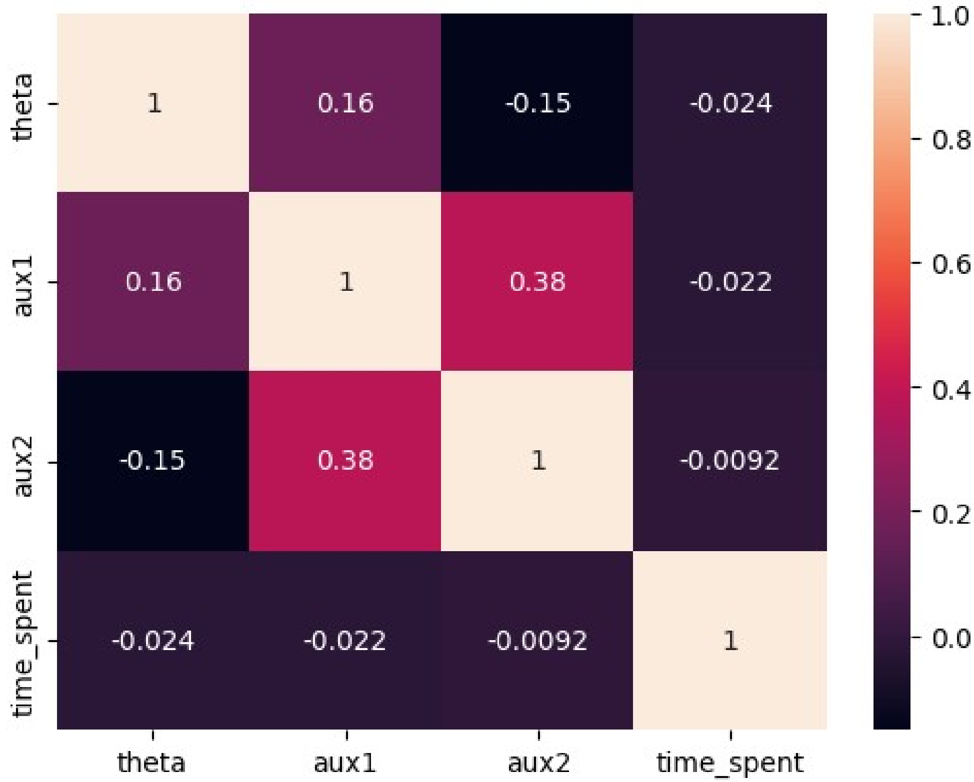
```
aux1 = [ 0.53983961, -1.77528229,  1.31487654, -0.47344805, -1.0922299
,
       -0.25002744, -0.9822943 ,  1.03126909,  0.49133378, -
0.4466466 ,
       -0.80636008,  0.13126776, -1.21256024,  0.15999085, -
0.75522304,
        0.34989599,  0.97754176, -0.13858525,  0.10385631,
0.30059104]
aux2 = [ 0.9682053 ,  0.86962384,  0.56778309,  0.46528234, -
1.16537308,
       -2.03599479, -1.15541329,  3.34515739,  0.12672721, -
0.6941789 ,
        0.55767443,  0.0991466 ,  0.63792617,  0.70311068, -
0.91609315,
       -0.78601423,  1.1191818 , -0.98339611,  0.24452002, -
0.58140974]
time_spent = [10.79768391, 10.87648065, 10.29274937, 10.78756647,
9.51844772,
        9.18078781,  9.90063639, 12.84823357, 10.92743478,
9.88927608,
       11.3373709 , 11.43996915, 11.88392171, -11.88135476,
11.73452467,
       11.18844425, 12.19144316, 11.35294826, 12.2385441 ,
11.98428985]

df =
pd.DataFrame({'theta':theta,'aux1':aux1,'aux2':aux2,'time_spent':time_
spent})
df.head()
```

	theta	aux1	aux2	time_spent
0	0.039063	0.539840	0.968205	10.797684
1	0.051194	-1.775282	0.869624	10.876481
2	0.060045	1.314877	0.567783	10.292749
3	0.067900	-0.473448	0.465282	10.787566
4	0.191521	-1.092230	-1.165373	9.518448

```
df.describe()
```

	theta	aux1	aux2	time_spent
count	20.000000	20.000000	20.000000	20.000000
mean	0.475222	-0.126610	0.069323	9.924470
std	0.292726	0.818612	1.168883	5.221924
min	0.039063	-1.775282	-2.035995	-11.881355
25%	0.260121	-0.768007	-0.818534	10.194721
50%	0.541169	-0.017364	0.185624	11.057940
75%	0.653397	0.385255	0.654222	11.771874
max	0.945683	1.314877	3.345157	12.848234

```
sb.heatmap(df.corr(),annot = True)
plt.show()
# Multi-collinearity present making aux1 variable insignificant
```




```
df1 = df[df.time_spent >= 0]
df1.head()
```

	theta	aux1	aux2	time_spent
0	0.039063	0.539840	0.968205	10.797684
1	0.051194	-1.775282	0.869624	10.876481
2	0.060045	1.314877	0.567783	10.292749
3	0.067900	-0.473448	0.465282	10.787566
4	0.191521	-1.092230	-1.165373	9.518448

2. Build a model

Write functions to run a regression, calculate the regression statistics listed below, and print a report.

- B (regressor coefficients plus one for an intercept, if appropriate)
- R2
- RSS
- RegSS

- TSS
- t statistic for each regressor coefficient

I found it useful to decompose the problem into three functions: `regress_calc()`, `regress_tstat()`, and `regress_report()`. You may write it however you see fit.

You may include either, both, or neither of *aux1* and *aux2* in your final model. Experiment. What works best? Justify your decision.

```
X = df1.iloc[:,:3]
X1 = df1.iloc[:,:2]
X2 = df1[['theta','aux2']]
X3 = df1[['theta']]
X4 = sm.add_constant(X3)
X5 = sm.add_constant(X)
X6 = sm.add_constant(X1)
X7 = sm.add_constant(X2)
y = df1.time_spent

def regress_tstat(model):
    return pd.DataFrame({'t-
statistic':model.tvalues,'pvalues':model.pvalues})

def get_coefficients(model):
    return pd.DataFrame(model.params,columns=['Coefficients'])

def regress_calc(M,x,y,m=''):
    y_pred = M.predict(x)
    SST = np.sum((y-np.mean(y))**2) #Total Sum of Squares SST =
sum((y-y_mean)**2)
    SSR = M.rsquared*SST # R2 = SSR/SST hence SSR = R2 * SST
    SSE = SST- SSR
    return pd.DataFrame({'Model':[m],
            'R2':[M.rsquared],
            'Adj_R2':[M.rsquared_adj],
            'SSE':[SSE],
            'SSR':[SSR],
            'SST':[SST],
            'MSE':[mean_squared_error(y,y_pred)],
            'RMSE':[np.sqrt(mean_squared_error(y,y_pred))],
            'MAE':[mean_absolute_error(y,y_pred)],
            'MAPE':[mean_absolute_percentage_error(y,y_pred)]})
```

Full Model without Constant

```
model = sm.OLS(y,X).fit()
model.summary()

<class 'statsmodels.iolib.summary.Summary'>
"""
```

```
                                 OLS Regression Results

=======================================================================================
Dep. Variable:             time_spent   R-squared (uncentered):
0.803
Model:                            OLS   Adj. R-squared (uncentered):
0.766
Method:                 Least Squares   F-statistic:
21.68
Date:                Wed, 06 Mar 2024   Prob (F-statistic):
7.00e-06
Time:                        13:17:00   Log-Likelihood:
-57.304
No. Observations:                  19   AIC:
120.6
Df Residuals:                      16   BIC:
123.4
Df Model:                           3

Covariance Type:            nonrobust

==============================================================================
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
theta         17.6885      2.253      7.850      0.000      12.912
22.465
aux1          -2.1228      1.598     -1.328      0.203      -5.511
1.266
aux2           1.9014      1.141      1.666      0.115      -0.518
4.320
==============================================================================
Omnibus:                        1.684   Durbin-Watson:
0.323
Prob(Omnibus):                  0.431   Jarque-Bera (JB):
1.188
Skew:                           0.367   Prob(JB):
0.552
Kurtosis:                       2.020   Cond. No.
2.23
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does
not contain a constant.
```

```
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
"""
```

```
get_coefficients(model)
```

```
       Coefficients
theta     17.688544
aux1      -2.122787
aux2       1.901389
```

```
regress_tstat(model)
```

```
       t-statistic       pvalues
theta     7.849837  7.084696e-07
aux1     -1.328086  2.027848e-01
aux2      1.666329  1.151011e-01
```

```
Full = regress_calc(model,X,y,m='Full Model without constant')
Full
```

```
                         Model        R2    Adj_R2       SSE
SSR  \
0  Full Model without constant  0.802549  0.765527  3.471452
14.109922

         SST        MSE      RMSE       MAE      MAPE
0  17.581373  24.388642  4.938486  3.956237  0.372579
```

Model with theta and aux1

```
model1 = sm.OLS(y,X1).fit()
model1.summary()
```

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                                 OLS Regression Results

======================================================================
=================
Dep. Variable:             time_spent   R-squared (uncentered):
0.768
Model:                            OLS   Adj. R-squared (uncentered):
0.741
Method:                 Least Squares   F-statistic:
28.18
Date:                Wed, 06 Mar 2024   Prob (F-statistic):
4.00e-06
Time:                        13:17:00   Log-Likelihood:
-58.824
No. Observations:                  19   AIC:
```

```
121.6
Df Residuals:                      17   BIC:
123.5
Df Model:                           2

Covariance Type:            nonrobust

======================================================================
========
                 coef    std err          t      P>|t|      [0.025
0.975]
----------------------------------------------------------------------
--------
theta         17.5237      2.366      7.407      0.000      12.532
22.515
aux1          -1.1765      1.570     -0.749      0.464      -4.489
2.136
======================================================================
========
Omnibus:                        1.597   Durbin-Watson:
0.195
Prob(Omnibus):                  0.450   Jarque-Bera (JB):
1.116
Skew:                           0.330   Prob(JB):
0.572
Kurtosis:                       2.013   Cond. No.
1.51
======================================================================
========

Notes:
[1] R² is computed without centering (uncentered) since the model does
not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
"""

get_coefficients(model1)

       Coefficients
theta     17.523653
aux1      -1.176491

regress_tstat(model1)

       t-statistic   pvalues
theta     7.406751  0.000001
aux1     -0.749249  0.463950
```

```
Model_with_theta_aux1 = regress_calc(model1,X1,y,m = 'Model with theta and aux1')
Model_with_theta_aux1
```

```
                       Model        R2    Adj_R2      SSE
SSR  \
0  Model with theta and aux1  0.768284  0.741023  4.07389  13.507483


         SST        MSE      RMSE      MAE      MAPE
0  17.581373  28.621065  5.349866  4.01667  0.370482
```

Model with theta and aux2

```
model2 = sm.OLS(y,X2).fit()
model2.summary()
```

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                                 OLS Regression Results

======================================================================
=================
Dep. Variable:             time_spent   R-squared (uncentered):
0.781
Model:                            OLS   Adj. R-squared (uncentered):
0.755
Method:                 Least Squares   F-statistic:
30.27
Date:                Wed, 06 Mar 2024   Prob (F-statistic):
2.50e-06
Time:                        13:17:00   Log-Likelihood:
-58.297
No. Observations:                  19   AIC:
120.6
Df Residuals:                      17   BIC:
122.5
Df Model:                           2

Covariance Type:            nonrobust

======================================================================
========
                 coef    std err          t      P>|t|      [0.025
0.975]
----------------------------------------------------------------------
--------
theta         17.8212      2.301      7.744      0.000      12.966
22.676
aux2           1.3630      1.090      1.250      0.228      -0.937
```

```
3.663
==============================================================================
========
Omnibus:                        3.936   Durbin-Watson:
0.081
Prob(Omnibus):                  0.140   Jarque-Bera (JB):
1.375
Skew:                           0.073   Prob(JB):
0.503
Kurtosis:                       1.690   Cond. No.
2.12
==============================================================================
========

Notes:
[1] R² is computed without centering (uncentered) since the model does
not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
"""
```

```
get_coefficients(model2)
```

```
       Coefficients
theta     17.821177
aux2       1.362967
```

```
regress_tstat(model2)
```

```
       t-statistic       pvalues
theta     7.744409  5.663004e-07
aux2      1.250068  2.282099e-01
```

```
Model_with_theta_aux2 = regress_calc(model2,X2,y,m = 'Model with theta
and aux2')
Model_with_theta_aux2
```

```
                       Model        R2    Adj_R2       SSE        SSR  \
0  Model with theta and aux2  0.780783  0.754993  3.854139  13.727235


         SST        MSE      RMSE       MAE     MAPE
0  17.581373  27.077205  5.203576  4.166276  0.39246
```

Model with theta

```
model3 = sm.OLS(y,X3).fit()
model3.summary()
```

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                                 OLS Regression Results                                
=======================================================================================
Dep. Variable:             time_spent   R-squared (uncentered):                   0.761
Model:                            OLS   Adj. R-squared (uncentered):              0.747
Method:                 Least Squares   F-statistic:                              57.20
Date:                Wed, 06 Mar 2024   Prob (F-statistic):                    5.40e-07
Time:                        13:17:00   Log-Likelihood:                         -59.133
No. Observations:                  19   AIC:                                      120.3
Df Residuals:                      18   BIC:                                      121.2
Df Model:                           1                                                  
Covariance Type:            nonrobust                                                  
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
theta         17.6374      2.332      7.563      0.000      12.738      22.537
==============================================================================
Omnibus:                        1.865   Durbin-Watson:                   0.094
Prob(Omnibus):                  0.393   Jarque-Bera (JB):                1.027
Skew:                           0.152   Prob(JB):                        0.598
Kurtosis:                       1.902   Cond. No.                         1.00
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is
```

```
correctly specified.
"""

get_coefficients(model3)

       Coefficients
theta     17.637397

regress_tstat(model3)

       t-statistic       pvalues
theta     7.562938  5.403488e-07

Model_with_theta = regress_calc(model3,X3,y,m = 'Model with theta')
Model_with_theta

              Model        R2    Adj_R2       SSE        SSR
SST  \
0  Model with theta  0.760632  0.747334  4.208418  13.372955
17.581373

         MSE     RMSE       MAE      MAPE
0  29.566189  5.43748  4.303507  0.397699
```

Model with constant and theta

```
model4 = sm.OLS(y,X4).fit()
model4.summary()

<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results

==============================================================================
========
Dep. Variable:             time_spent   R-squared:
0.327
Model:                            OLS   Adj. R-squared:
0.288
Method:                 Least Squares   F-statistic:
8.266
Date:                Wed, 06 Mar 2024   Prob (F-statistic):
0.0105
Time:                        13:17:01   Log-Likelihood:
-22.458
No. Observations:                  19   AIC:
48.92
Df Residuals:                      17   BIC:
50.81
Df Model:                           1
```

```
Covariance Type:            nonrobust

==============================================================================
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
const         10.1874      0.362     28.112      0.000       9.423
10.952
theta          1.8958      0.659      2.875      0.011       0.505
3.287
==============================================================================
Omnibus:                        1.463   Durbin-Watson:
1.900
Prob(Omnibus):                  0.481   Jarque-Bera (JB):
0.274
Skew:                           0.101   Prob(JB):
0.872
Kurtosis:                       3.553   Cond. No.
4.25
==============================================================================

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

get_coefficients(model4)

       Coefficients
const     10.187390
theta      1.895847

regress_tstat(model4)

       t-statistic       pvalues
const    28.112358  1.080093e-15
theta     2.875114  1.050165e-02

Model_with_const_theta = regress_calc(model4,X4,y,m = 'Model with
const and theta')
Model_with_theta

              Model        R2    Adj_R2       SSE        SSR
SST  \
0  Model with theta  0.760632  0.747334  4.208418  13.372955
17.581373
```

```
          MSE     RMSE       MAE      MAPE
0  29.566189  5.43748  4.303507  0.397699
```

Model with constant theta aux1 and aux2

```
model5 = sm.OLS(y,X5).fit()
model5.summary()

<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results

======================================================================
========
Dep. Variable:             time_spent   R-squared:
0.888
Model:                            OLS   Adj. R-squared:
0.866
Method:                 Least Squares   F-statistic:
39.71
Date:                Wed, 06 Mar 2024   Prob (F-statistic):
2.25e-07
Time:                        13:17:01   Log-Likelihood:
-5.4095
No. Observations:                  19   AIC:
18.82
Df Residuals:                      15   BIC:
22.60
Df Model:                           3

Covariance Type:            nonrobust

======================================================================
========
                 coef    std err          t      P>|t|      [0.025
0.975]
----------------------------------------------------------------------
--------
const          9.9201      0.167     59.334      0.000       9.564
10.276
theta          2.3877      0.299      7.982      0.000       1.750
3.025
aux1          -0.1000      0.113     -0.887      0.389      -0.340
0.140
aux2           0.6561      0.080      8.245      0.000       0.487
0.826
======================================================================
========
Omnibus:                        5.471   Durbin-Watson:
1.388
```

```
Prob(Omnibus):                   0.065   Jarque-Bera (JB):
3.845
Skew:                           -0.333   Prob(JB):
0.146
Kurtosis:                        5.101   Cond. No.
4.96
======================================================================
========

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

```
get_coefficients(model5)
```

```
       Coefficients
const      9.920095
theta      2.387676
aux1      -0.100006
aux2       0.656137
```

```
regress_tstat(model5)
```

```
       t-statistic       pvalues
const    59.334340  3.271299e-19
theta     7.982085  8.851896e-07
aux1     -0.886578  3.892997e-01
aux2      8.245170  5.925766e-07
```

```
Model_with_const_theta_aux1_aux2 = regress_calc(model5,X5,y,m = 'Model
with const theta aux1 and aux2')
Model_with_const_theta_aux1_aux2
```

```
                                  Model       R2    Adj_R2
SSE  \
0  Model with const theta aux1 and aux2  0.88818  0.865816  1.965956


         SSR        SST       MSE     RMSE       MAE     MAPE
0  15.615417  17.581373  0.103471  0.32167  0.217073  0.01999
```

Model with constant theta and aux1

```
model6 = sm.OLS(y,X6).fit()
model6.summary()
```

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
```

```
==============================================================================
========
Dep. Variable:             time_spent   R-squared:
0.381
Model:                            OLS   Adj. R-squared:
0.304
Method:                 Least Squares   F-statistic:
4.932
Date:                Wed, 06 Mar 2024   Prob (F-statistic):
0.0214
Time:                        13:17:01   Log-Likelihood:
-21.660
No. Observations:                  19   AIC:
49.32
Df Residuals:                      16   BIC:
52.15
Df Model:                           2

Covariance Type:            nonrobust

==============================================================================
========
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
--------
const         10.2836      0.367     28.000      0.000       9.505
11.062
theta          1.7740      0.660      2.689      0.016       0.375
3.173
aux1           0.2780      0.235      1.184      0.254      -0.220
0.776
==============================================================================
========
Omnibus:                        0.137   Durbin-Watson:
2.021
Prob(Omnibus):                  0.934   Jarque-Bera (JB):
0.032
Skew:                           0.006   Prob(JB):
0.984
Kurtosis:                       2.800   Cond. No.
4.38
==============================================================================
========

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

```
get_coefficients(model6)
```

```
       Coefficients
const     10.283646
theta      1.773987
aux1       0.277963
```

```
regress_tstat(model6)
```

```
       t-statistic       pvalues
const    27.999956  5.072769e-15
theta     2.688659  1.614248e-02
aux1      1.184239  2.536133e-01
```

```
Model_with_const_theta_aux1 = regress_calc(model6,X6,y,m = 'Model with
const theta and aux1')
Model_with_const_theta_aux1
```

```
                             Model        R2    Adj_R2        SSE
SSR  \
0  Model with const theta and aux1  0.381389  0.304062  10.876038
6.705335

         SST       MSE      RMSE       MAE      MAPE
0  17.581373  0.572423  0.756586  0.614386  0.057124
```

Model with constant theta and aux2

```
model7 = sm.OLS(y,X7).fit()
model7.summary()
```

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results

======================================================================
========
Dep. Variable:             time_spent   R-squared:
0.882
Model:                            OLS   Adj. R-squared:
0.868
Method:                 Least Squares   F-statistic:
59.98
Date:                Wed, 06 Mar 2024   Prob (F-statistic):
3.68e-08
Time:                        13:17:01   Log-Likelihood:
-5.8947
No. Observations:                  19   AIC:
17.79
Df Residuals:                      16   BIC:
20.62
```

```
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          9.9649      0.158     62.949      0.000       9.629      10.300
theta          2.3243      0.289      8.056      0.000       1.713       2.936
aux2           0.6275      0.072      8.688      0.000       0.474       0.781
==============================================================================
Omnibus:                        4.490   Durbin-Watson:                   1.430
Prob(Omnibus):                  0.106   Jarque-Bera (JB):                2.812
Skew:                          -0.125   Prob(JB):                        0.245
Kurtosis:                       4.868   Cond. No.                         4.52
==============================================================================

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

```
get_coefficients(model7)
```

```
       Coefficients
const      9.964893
theta      2.324257
aux2       0.627465
```

```
regress_tstat(model7)
```

```
       t-statistic       pvalues
const    62.948601  1.346041e-20
theta     8.056302  5.062555e-07
aux2      8.687911  1.871613e-07
```

```
Model_with_const_theta_aux2 = regress_calc(model7,X7,y,m = 'Model with const theta and aux2')
Model_with_const_theta_aux2
```

```
                             Model       R2   Adj_R2       SSE
SSR  \
0  Model with const theta and aux2  0.88232  0.86761  2.068975
15.512399

         SST       MSE     RMSE       MAE      MAPE
0  17.581373  0.108893  0.32999  0.236424  0.021725
```

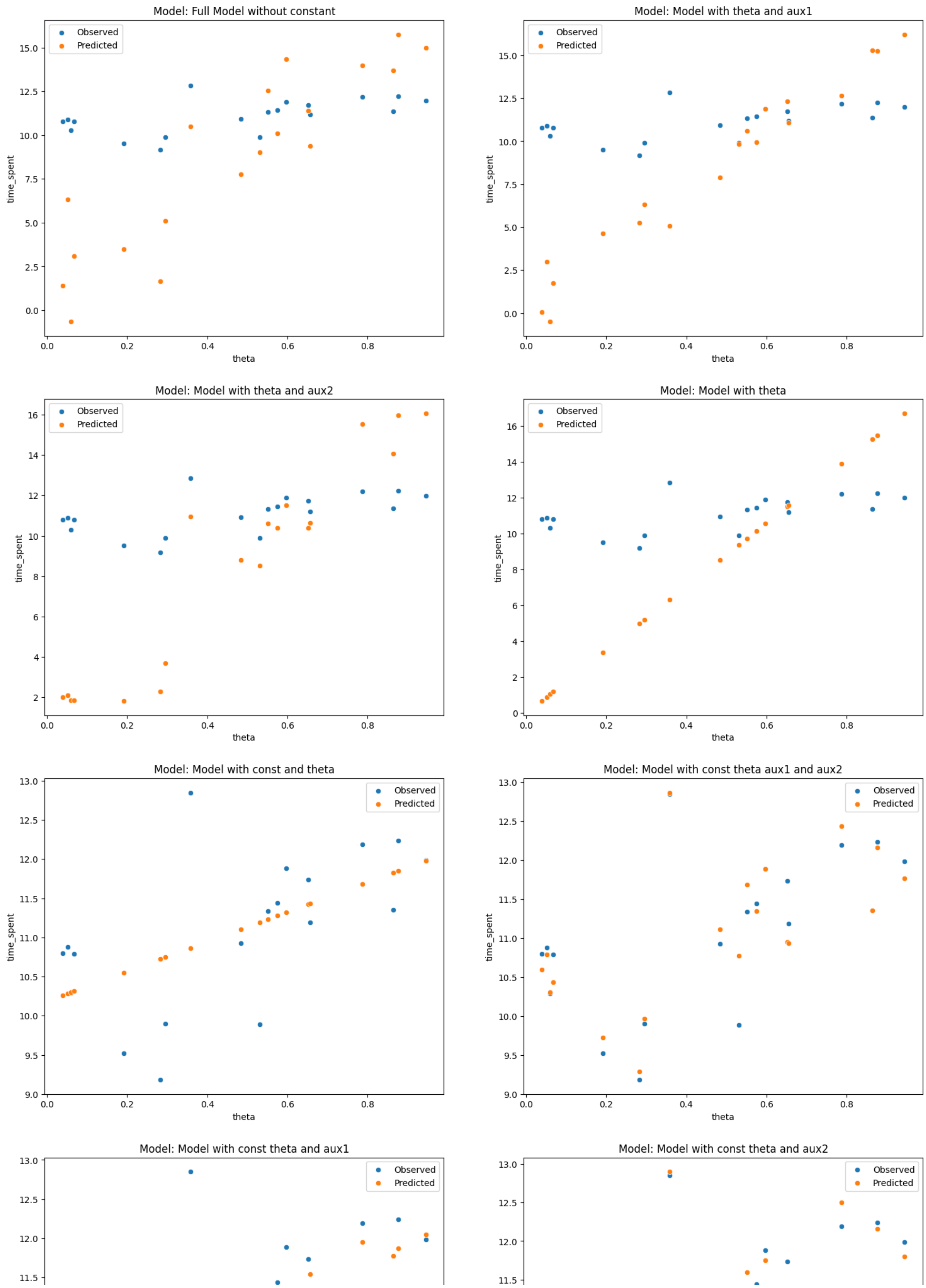
Final Results

```
Results =
pd.concat([Full,Model_with_theta_aux1,Model_with_theta_aux2,Model_with
_theta,Model_with_const_theta,Model_with_const_theta_aux1_aux2,

Model_with_const_theta_aux1,Model_with_const_theta_aux2],ignore_index=
True)

Results
```

```
                                  Model        R2    Adj_R2        SSE
\
0           Full Model without constant  0.802549  0.765527   3.471452

1             Model with theta and aux1  0.768284  0.741023   4.073890

2             Model with theta and aux2  0.780783  0.754993   3.854139

3                      Model with theta  0.760632  0.747334   4.208418

4            Model with const and theta  0.327166  0.287588  11.829337

5  Model with const theta aux1 and aux2  0.888180  0.865816   1.965956

6       Model with const theta and aux1  0.381389  0.304062  10.876038

7       Model with const theta and aux2  0.882320  0.867610   2.068975


         SSR        SST        MSE      RMSE       MAE      MAPE
0  14.109922  17.581373  24.388642  4.938486  3.956237  0.372579
1  13.507483  17.581373  28.621065  5.349866  4.016670  0.370482
2  13.727235  17.581373  27.077205  5.203576  4.166276  0.392460
3  13.372955  17.581373  29.566189  5.437480  4.303507  0.397699
4   5.752036  17.581373   0.622597  0.789048  0.592156  0.055126
5  15.615417  17.581373   0.103471  0.321670  0.217073  0.019990
6   6.705335  17.581373   0.572423  0.756586  0.614386  0.057124
7  15.512399  17.581373   0.108893  0.329990  0.236424  0.021725
```

```
models = [model,model1,model2,model3,model4,model5,model6,model7]
x = [X,X1,X2,X3,X4,X5,X6,X7]
m= Results.Model
```

```
fig,ax = plt.subplots(nrows = 4,ncols = 2,figsize = [18,30])
for i,j,k,l in zip(models,x,ax.flatten(),m):
    y_pred = i.predict(j)
    sb.scatterplot(x=j.theta,y=y,ax = k,label='Observed')
    sb.scatterplot(x=j.theta,y=y_pred,ax=k,label='Predicted')
    k.set_title(f'Model: {l}')
plt.show()
```



Model: Full Model without constant
Model: Model with theta and aux1
Model: Model with theta and aux2
Model: Model with theta
Model: Model with const and theta
Model: Model with const theta aux1 and aux2
Model: Model with const theta and aux1
Model: Model with const theta and aux2
Observed
Predicted
time_spent
theta

```
Results.sort_values(by = 'RMSE',ascending=True)
```

	Model	R2	Adj_R2	SSE
\				
5	Model with const theta aux1 and aux2	0.888180	0.865816	1.965956
7	Model with const theta and aux2	0.882320	0.867610	2.068975
6	Model with const theta and aux1	0.381389	0.304062	10.876038
4	Model with const and theta	0.327166	0.287588	11.829337
0	Full Model without constant	0.802549	0.765527	3.471452
2	Model with theta and aux2	0.780783	0.754993	3.854139
1	Model with theta and aux1	0.768284	0.741023	4.073890
3	Model with theta	0.760632	0.747334	4.208418

	SSR	SST	MSE	RMSE	MAE	MAPE
5	15.615417	17.581373	0.103471	0.321670	0.217073	0.019990
7	15.512399	17.581373	0.108893	0.329990	0.236424	0.021725
6	6.705335	17.581373	0.572423	0.756586	0.614386	0.057124
4	5.752036	17.581373	0.622597	0.789048	0.592156	0.055126
0	14.109922	17.581373	24.388642	4.938486	3.956237	0.372579
2	13.727235	17.581373	27.077205	5.203576	4.166276	0.392460
1	13.507483	17.581373	28.621065	5.349866	4.016670	0.370482
3	13.372955	17.581373	29.566189	5.437480	4.303507	0.397699

From the results we can see that model with intercept fit with all the independent variables has the least error of all the models but has aux1 variable with pvalue of t statistic >0.05 making it a insignificant variable hence we can select the model with intercept, theta and aux2 which has all the significant variables and prob(f-statistic) close to zero.

Drawbacks of the Model:

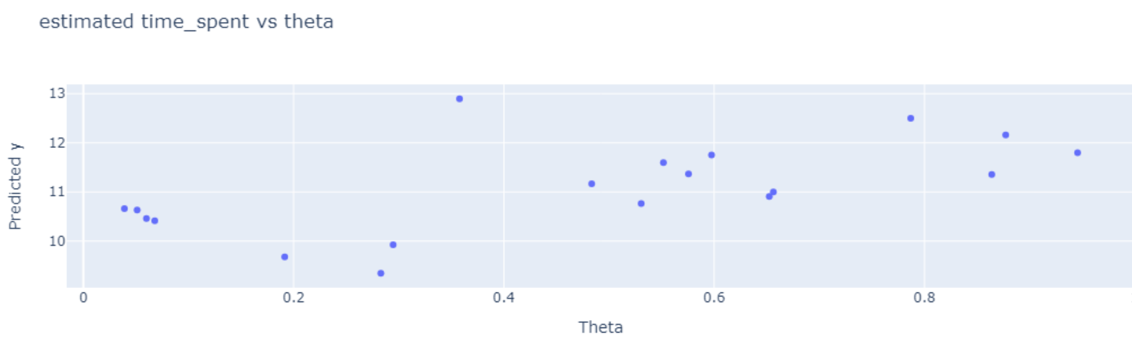
- The dataset is too small
- Overfitting is present hence the error too small for the dataset

3. Propose a setting for *theta*

Now that you have a model built, you should be able to plot estimated *time_spent* vs. *theta* over a reasonable range of *theta*. By inspecting that plot -- and knowning that the company wants to maximize the time users spend on the app -- which value of *theta* would you propose the engineers use? Explain how the data and your model support your decision.

The engineer's have capacity to take another set of measurements. Which settings of *theta* do you suggest they measure? Why?

```
y_pred = model7.predict(X7)
fig = go.Figure()
fig.add_trace(go.Scatter(x=X7.theta, y=y_pred, mode='markers',
name='time_spent'))
fig.update_layout(
    title='estimated time_spent vs theta',
    xaxis_title='Theta',
    yaxis_title='Predicted y'
)
fig.show()
```



From the plot We can see that the time spent is high i.e 12.89551 for theta of value 0.3578136 hence I choose the value of 0.3578136 to increase the time spent by the users

4. Experiment or observation?

Is this data set experimental or observational? Explain clearly. Consider how the effect of *theta* on *time_spent* differs from the effect of *aux1* or *aux2*.

The Dataset is Experimental data as the engineers ran the recommendation engine with different settings of theta and they measured the amount of time users spent on the app for each setting of theta. This suggests that the engineers manipulated the parameter theta (the independent variable) to observe its effect on the amount of time users spent on the app (the dependent variable). TThe data collection process involves controlled experiments where the engineers deliberately varied theta and observed the corresponding changes in time_spent