Least squares method for Time series

Compare 3 methods:

- Pseudo inverse
- QR
- SVD

Problem description

$$\hat{x}_{k+1} = w_1 x_k + \ldots + w_M x_{k-M+1}$$

$$\|Xw - y\|_2^2 \to \min_w$$

The goal is to compare 3 methods: pseudo inverse, QR and SVD

Toeplitz

- Toeplitz matrix - Wikipec
- O(n) add
- O(n log n) vector mult
- O(n2) matrix mult
- O(n2) to solve SLAE
- O(n2) decomposition
- scipy works only with square matrices :(

$$A = \begin{bmatrix} a_0 & a_{-1} & a_{-2} & \cdots & \cdots & a_{-(n-1)} \\ a_1 & a_0 & a_{-1} & \ddots & & \vdots \\ a_2 & a_1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_{-1} & a_{-2} \\ \vdots & & \ddots & a_1 & a_0 & a_{-1} \\ a_{n-1} & \cdots & \cdots & a_2 & a_1 & a_0 \end{bmatrix}$$

Method of comparison of algorithms

1. Splitting a dataframe with different ratios
2. Training on a training samples
3. Calculating MSE on a test samples
4. Comparison MSE

Predict driver_pay

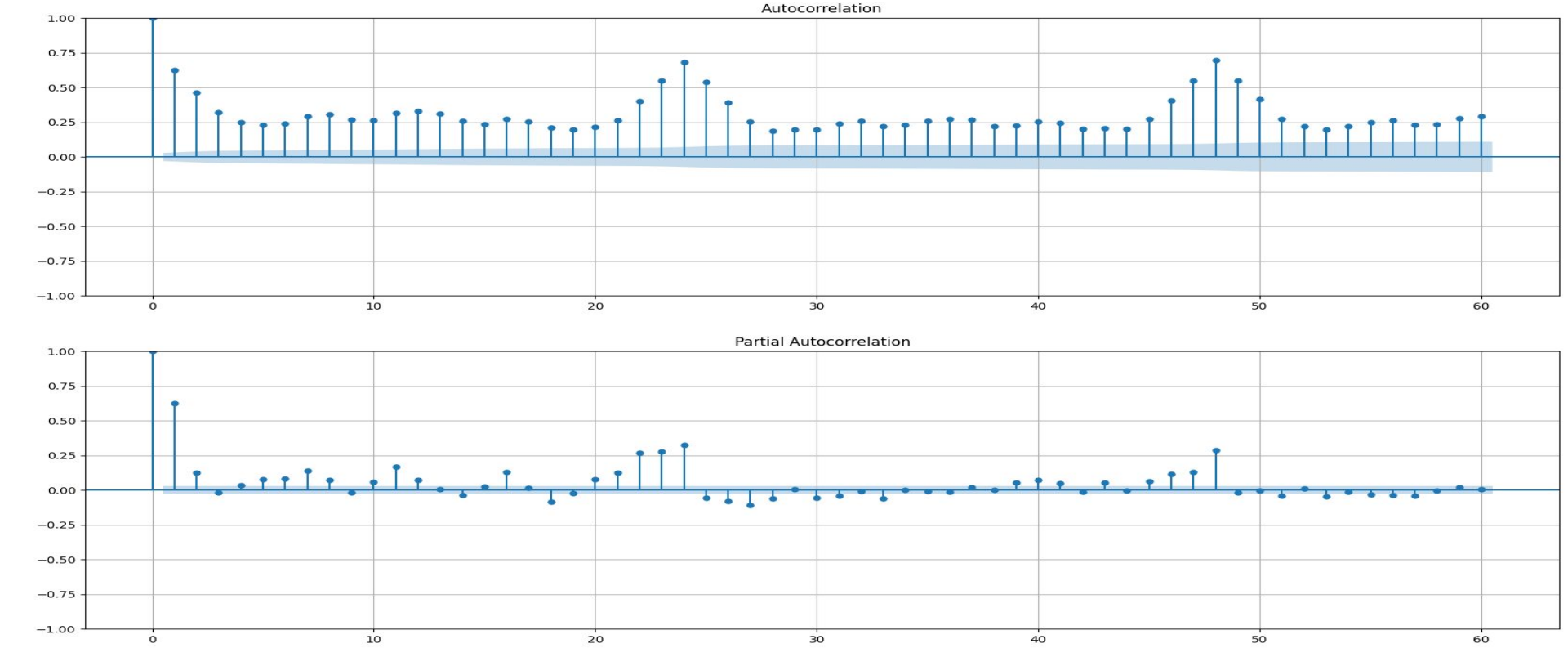
https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page

January

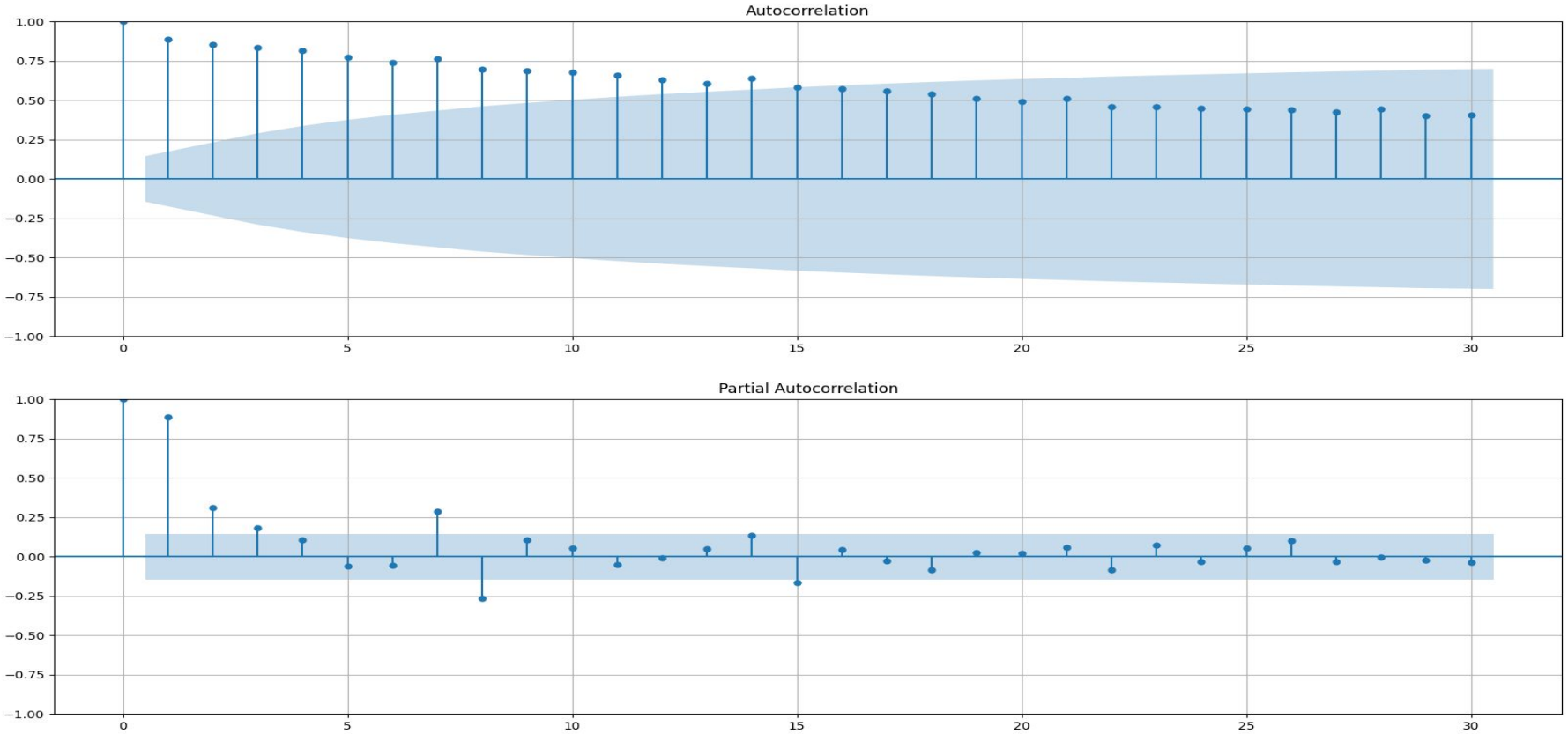
- **Yellow Taxi Trip Records** (PARQUET)
- **Green Taxi Trip Records** (PARQUET)
- **For-Hire Vehicle Trip Records** (PARQUET)
- **High Volume For-Hire Vehicle Trip Records** (PARQUET)

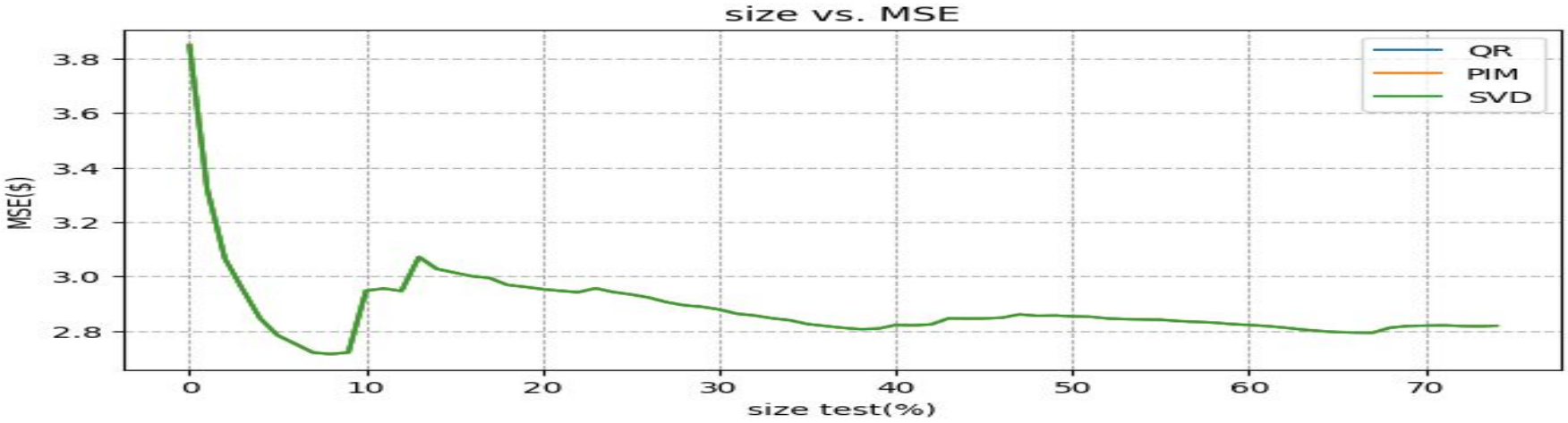
Size(**High Volume For-Hire Vehicle Trip Records**) = `(8708453, 14)`

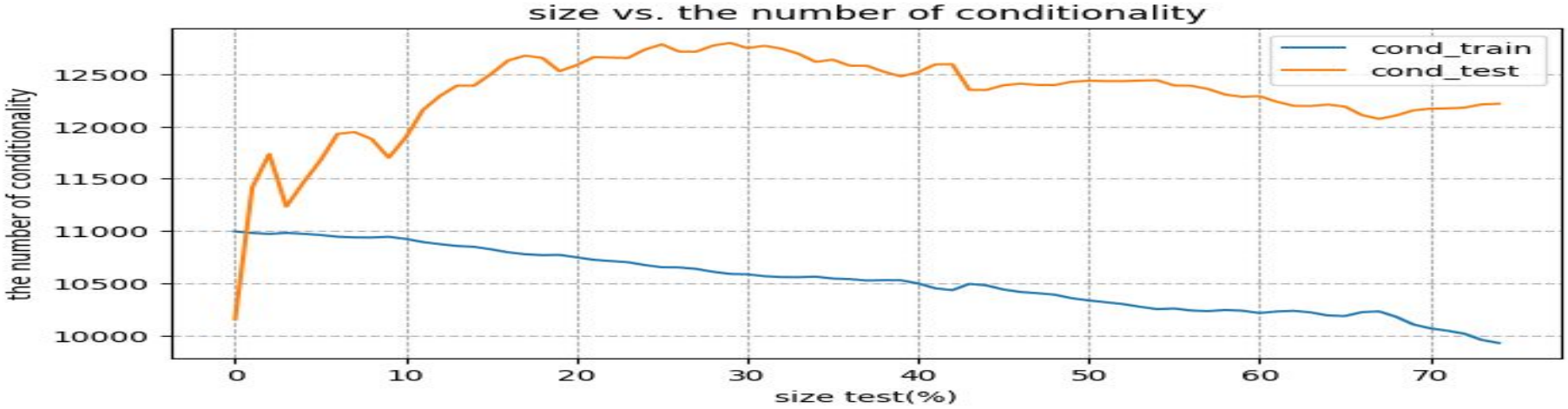
selection of lags



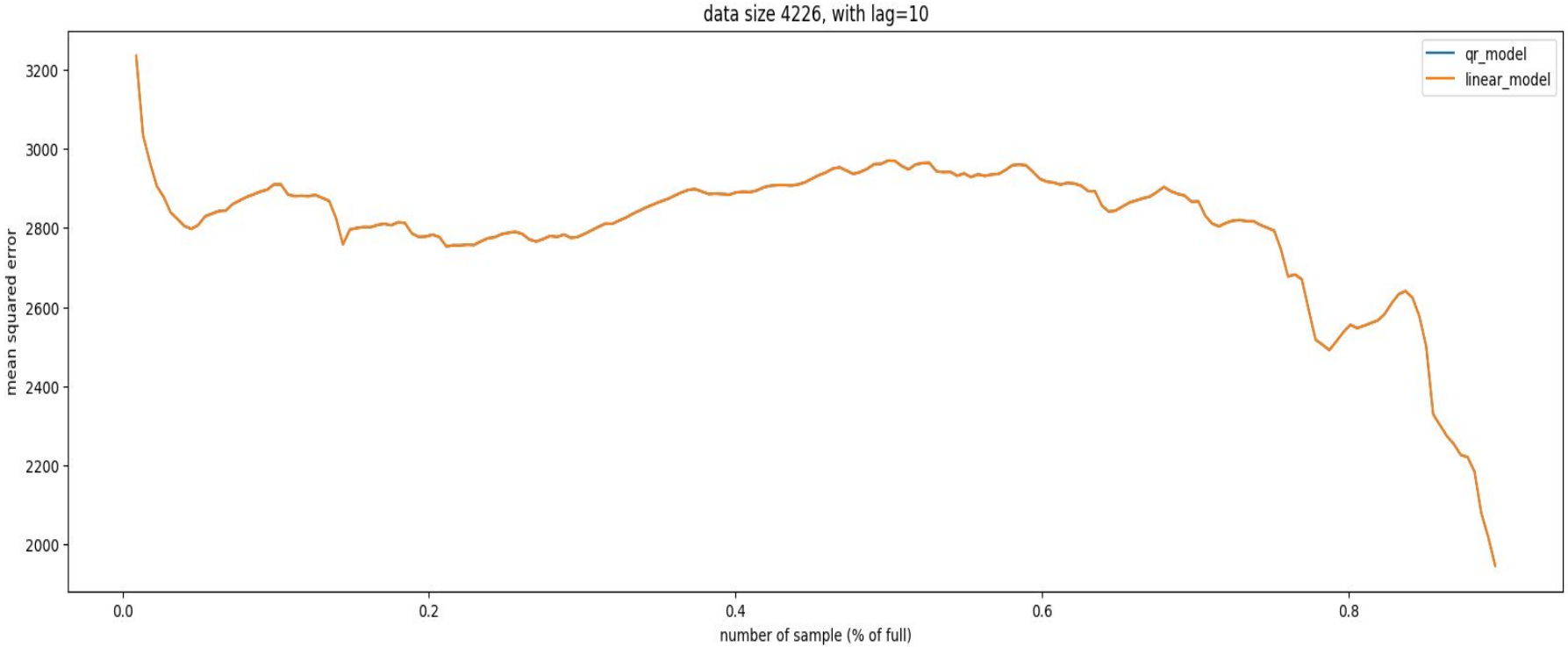
selection of lags



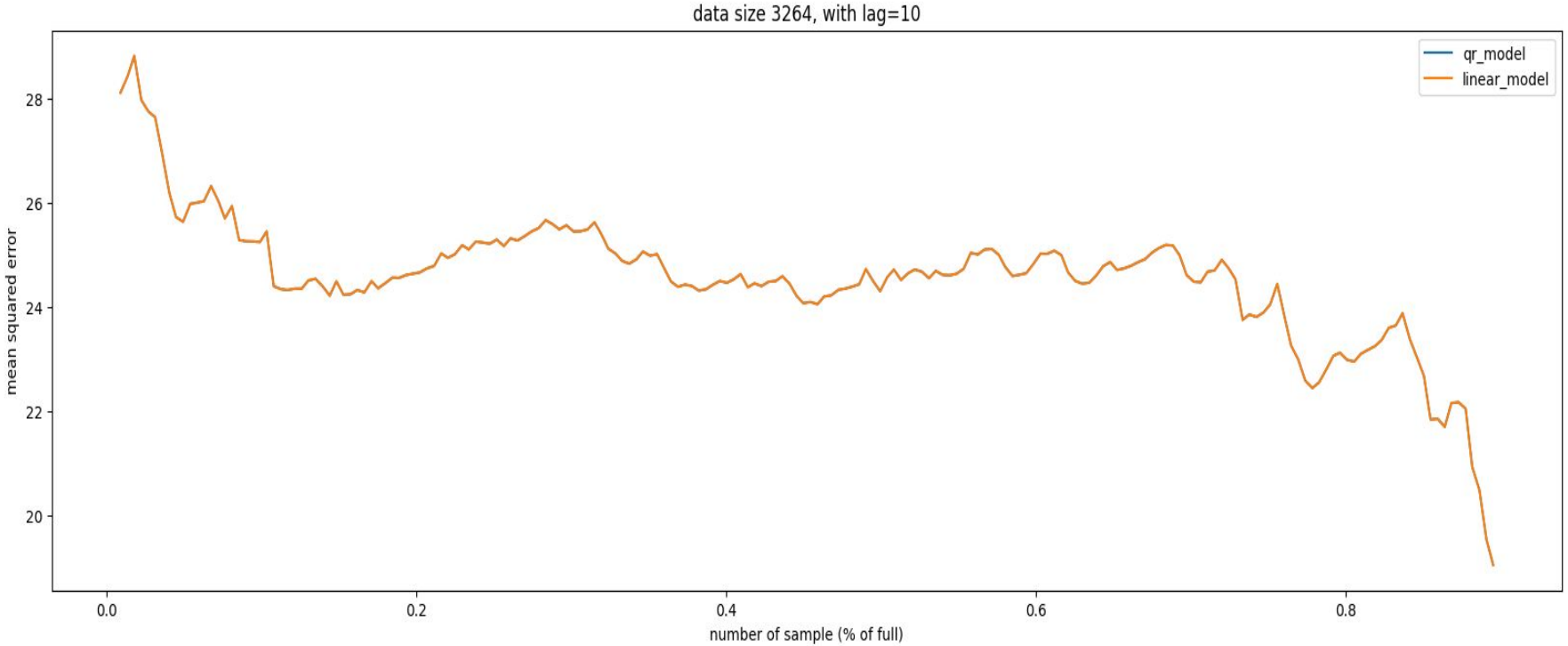

size vs. MSE
MSE($)
size test(%)
QR
PIM
SVD


size vs. the number of conditionality
the number of conditionality
size test(%)
cond_train
cond_test

https://github.com/Mcompetitions/M4-methods/blob/master/Dataset/Train/Daily-train.csv
V8

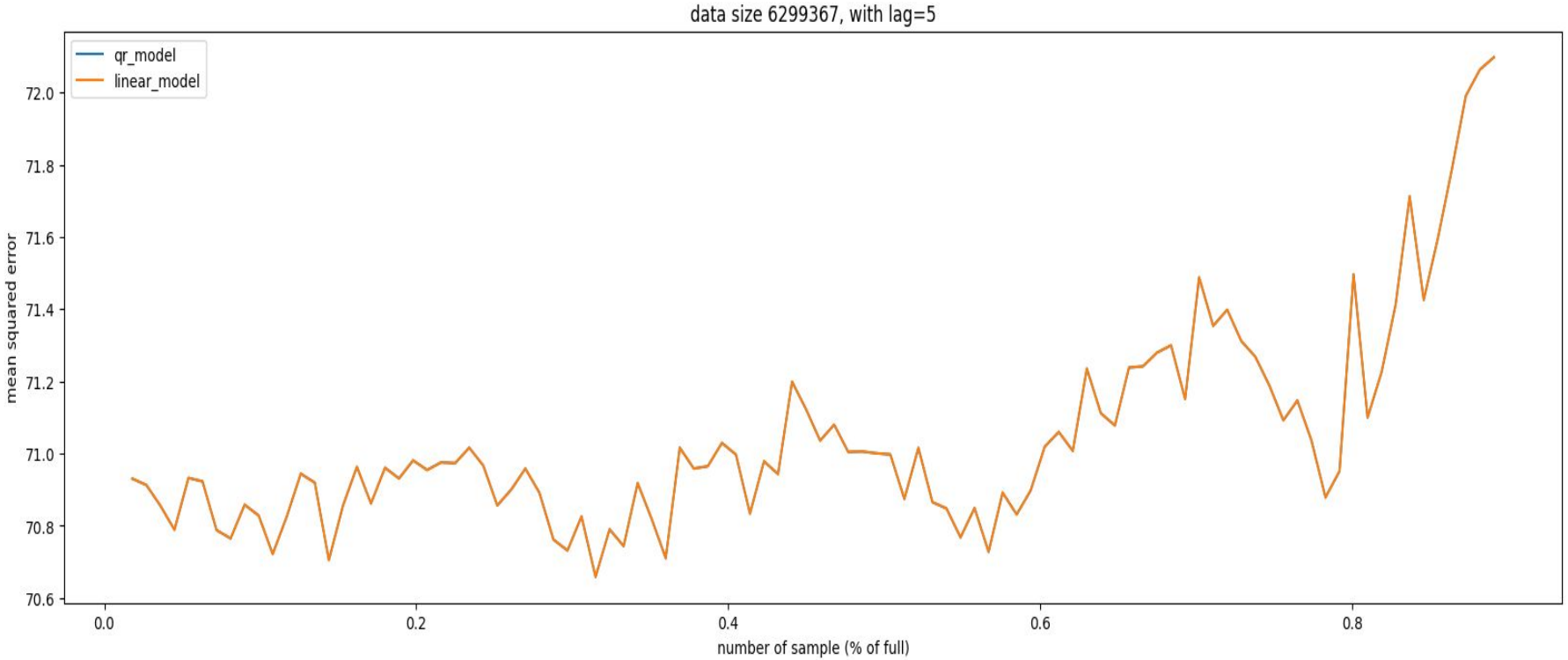


https://www.kaggle.com/datasets/robervalt/sunspots



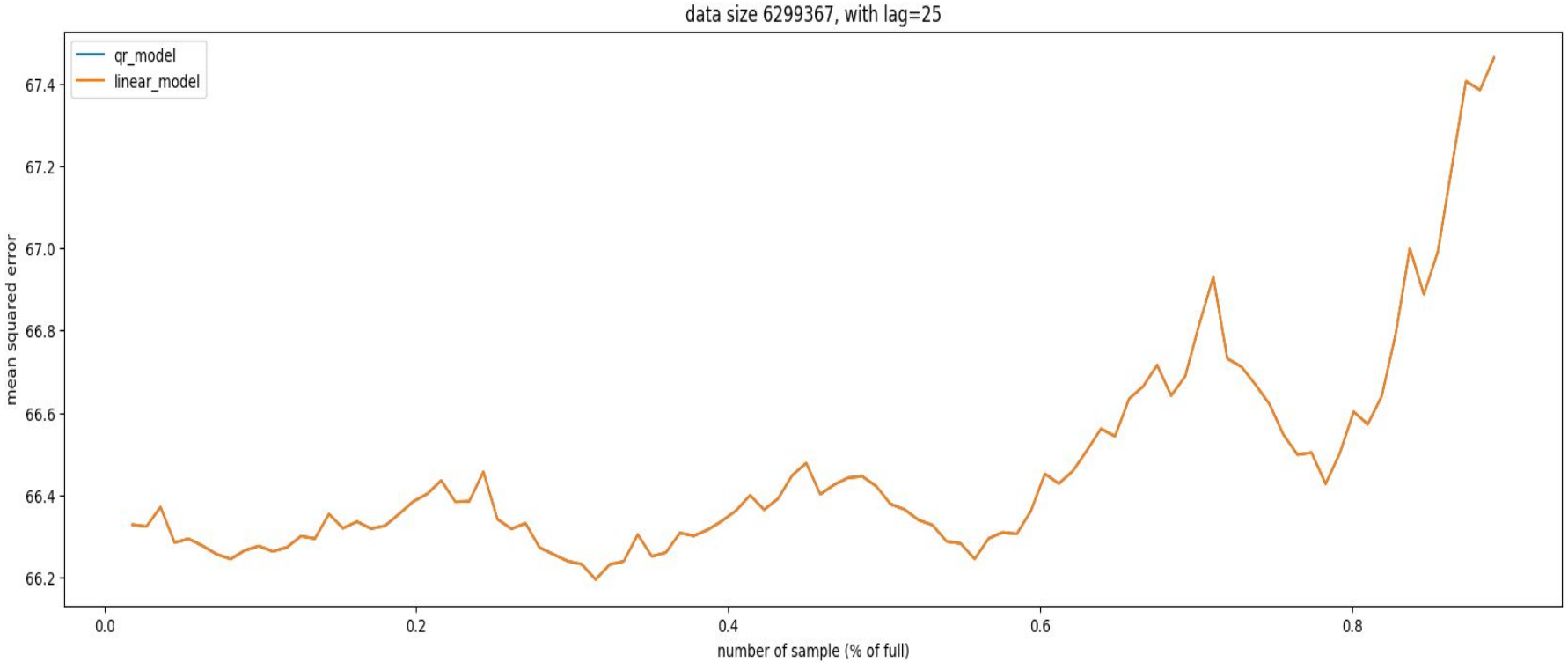
https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page

yellow_tripdata_2020-02.parquet - PULocationID



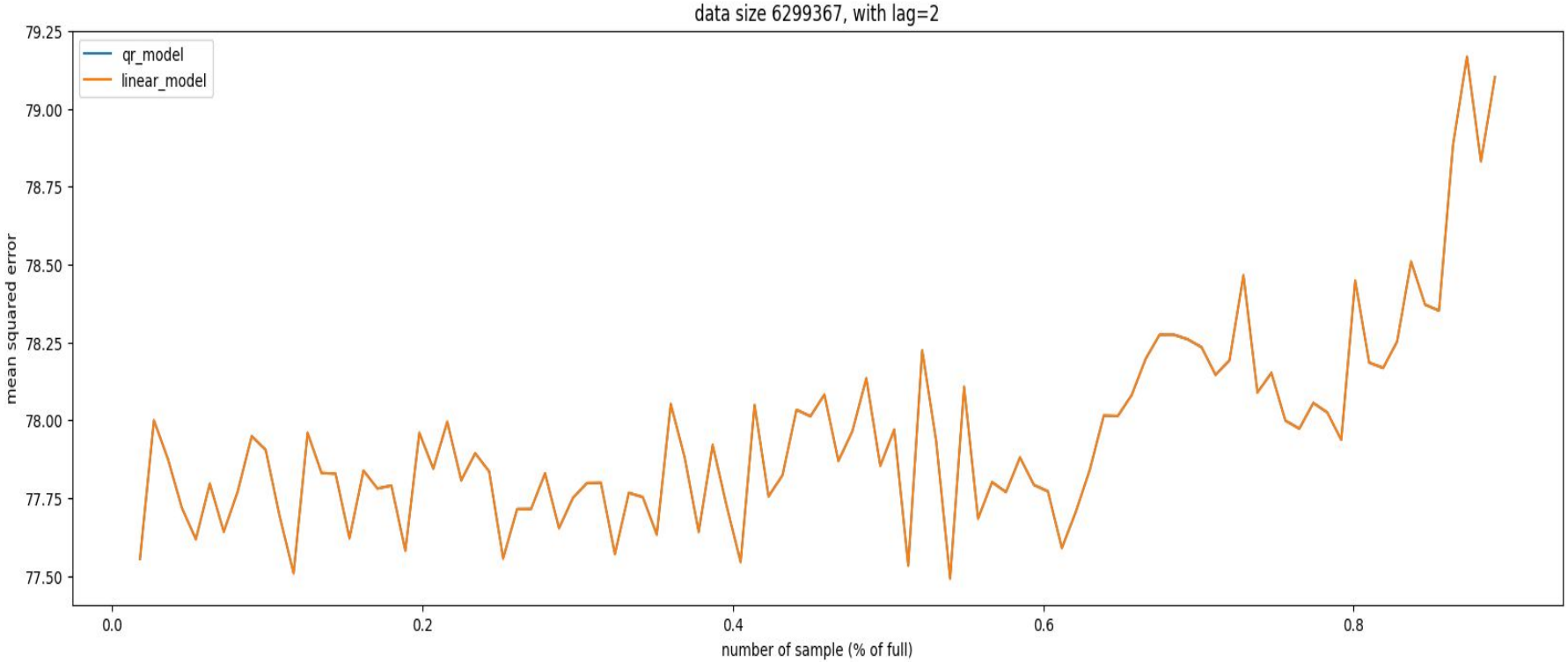
https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page

yellow_tripdata_2020-02.parquet - PULocationID

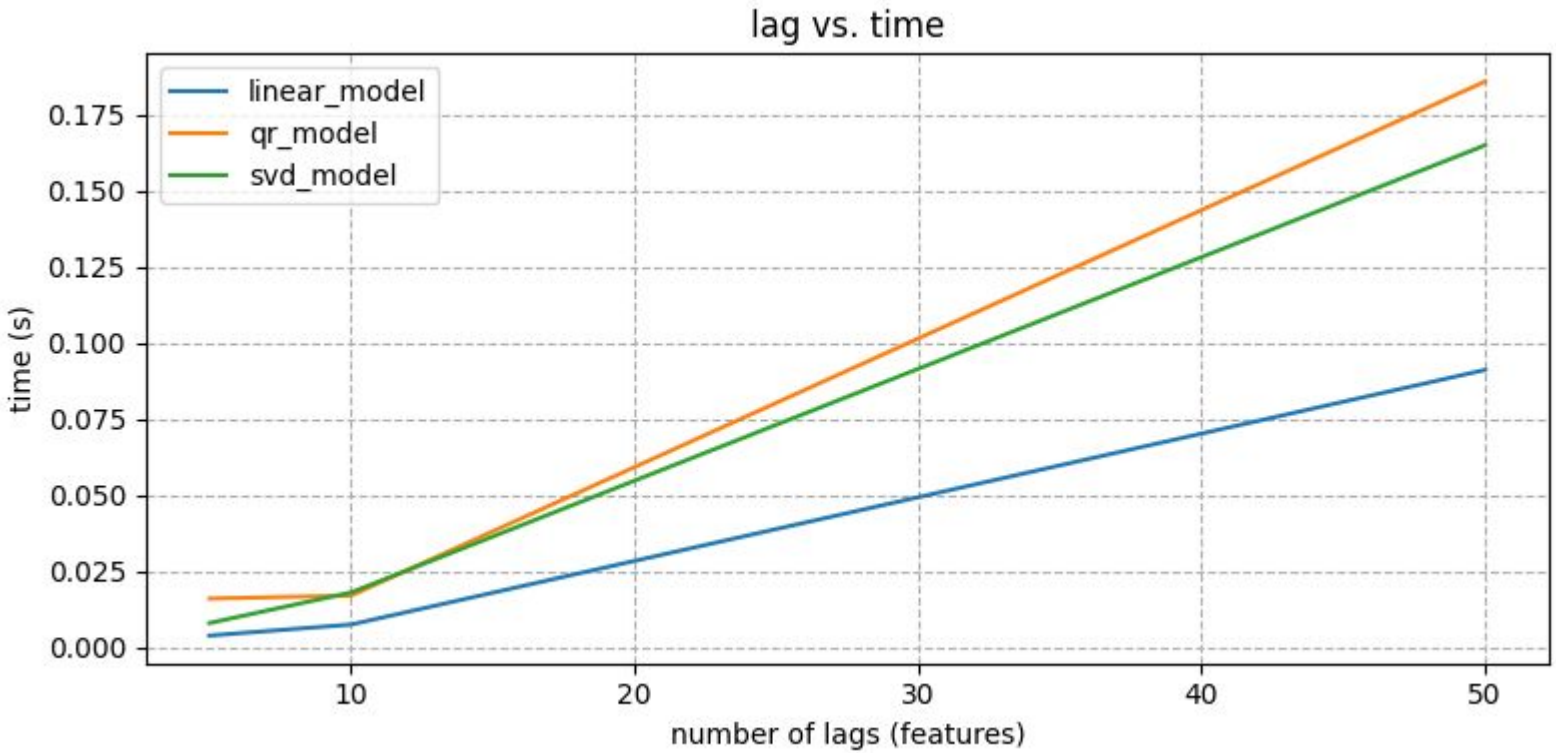


https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page

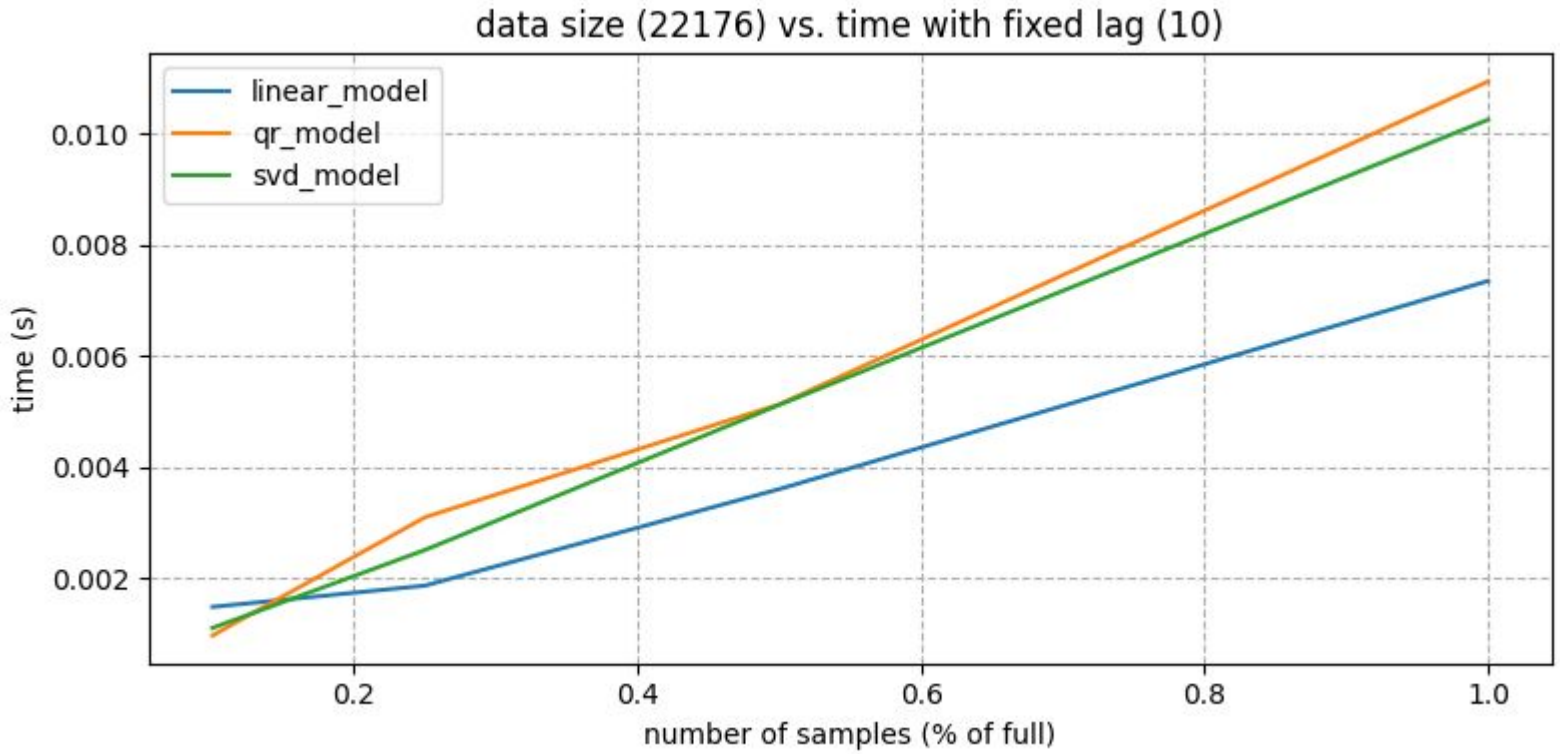
yellow_tripdata_2020-02.parquet - PULocationID



Results


lag vs. time
linear_model
qr_model
svd_model
time (s)
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
10
20
30
40
50
number of lags (features)

Results


data size (22176) vs. time with fixed lag (10)
linear_model
qr_model
svd_model
0.010
0.008
0.006
0.004
0.002
time (s)
0.2
0.4
0.6
0.8
1.0
number of samples (% of full)

Conclusions

- all the methods are numerically stable (the same solution to SLAE)
- solving via pseudo inverse is faster
- https://github.com/Ulycecc/project_nla

Plans

- Building an accurate taxi payment model
- Find the widest possible class of random processes with a full-rank matrix

https://github.com/Ulycecc/project_nla