

1.

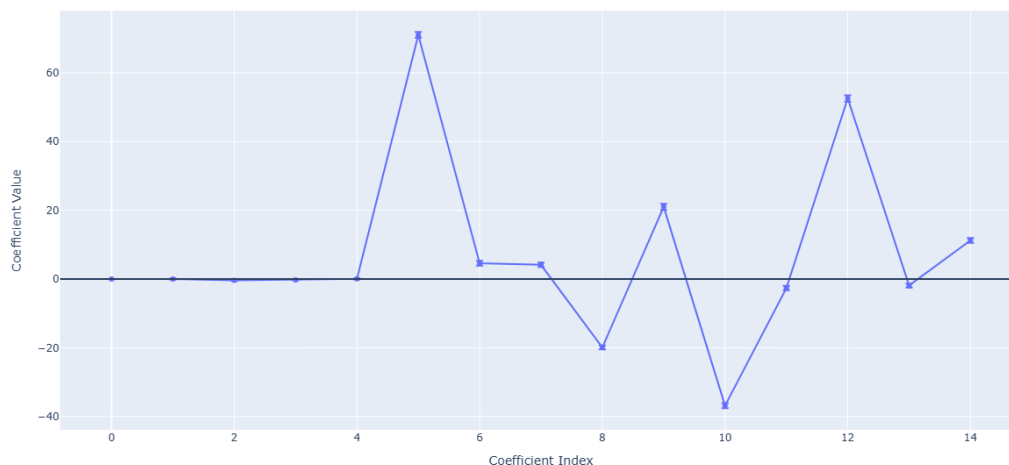
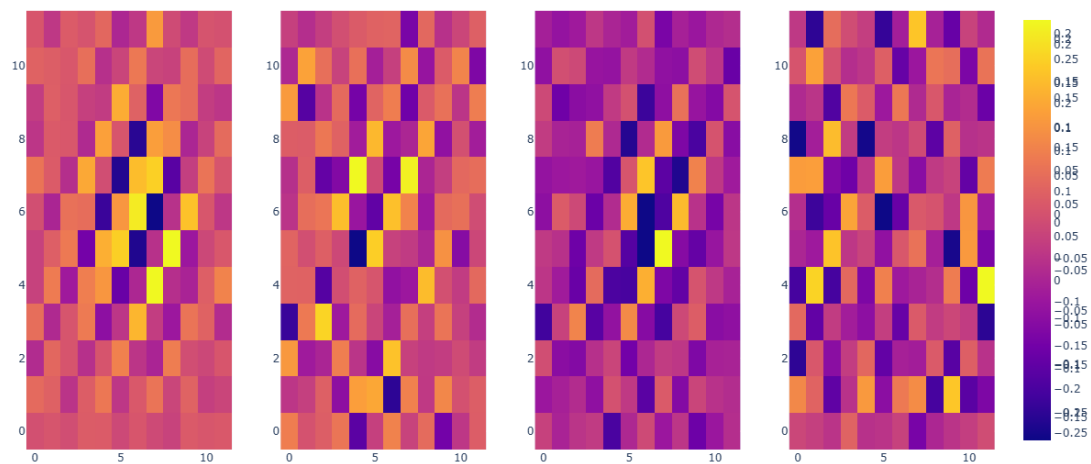
```
# calculate regression coefficients with train data
X = utils.buildDataMatrix(px=px, order=order, nRDs=n_RDs)
X_train, X_test, Y_train, Y_test = \
    sklearn.model_selection.train_test_split(X, Y,
                                              test_size=test_percentage)

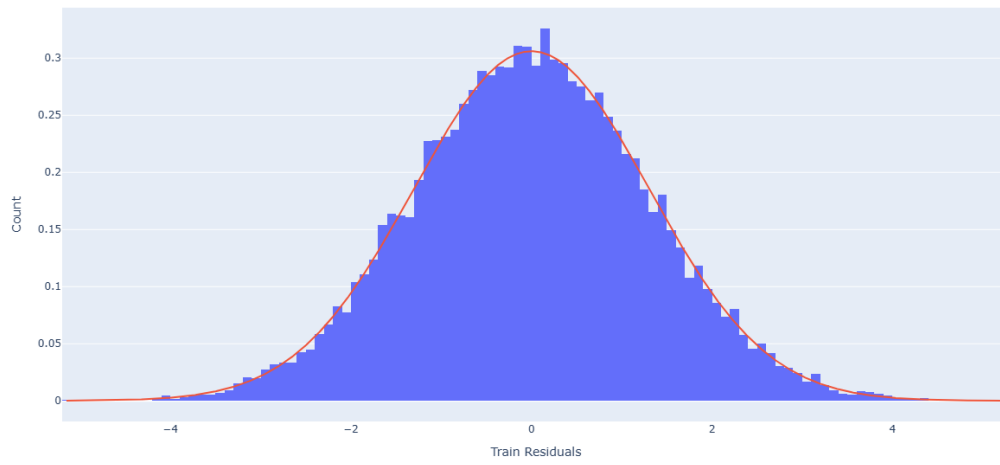
I = np.eye(X.shape[1])
#beta=np.linalg.inv(np.transpose(X) @ X + reg_coef * I) @ np.transpose(X) @ y_train
coefs = np.linalg.solve(np.transpose(X_train) @ X_train + reg_coef * I,
np.transpose(X_train) @ Y_train)
```

```
# calculate residuals
fitted_train = np.matmul(X_train, coefs)
residuals_train = Y_train - fitted_train

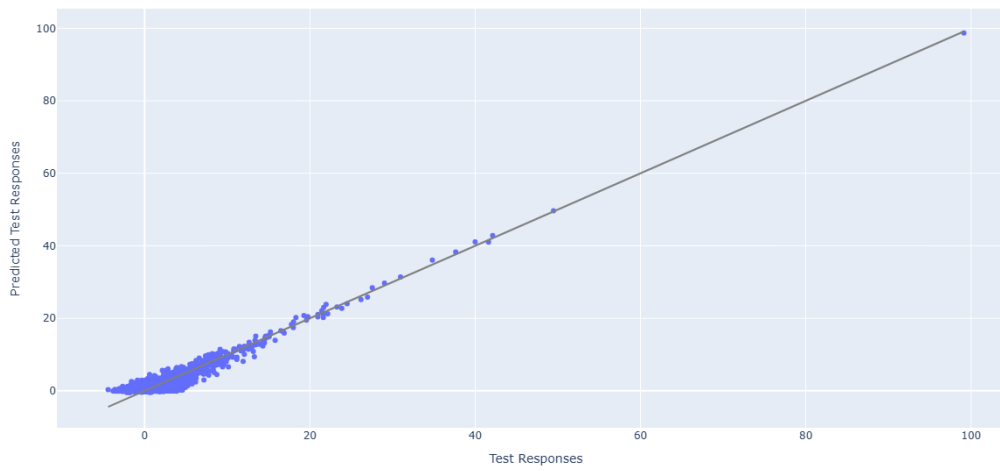
# compute correlation coefficient on test data
fitted_test = np.matmul(X_test, coefs)
rho_test = np.corrcoef(Y_test, fitted_test)[0, 1]
```

The best set of parameters: nRDs=4, order=2

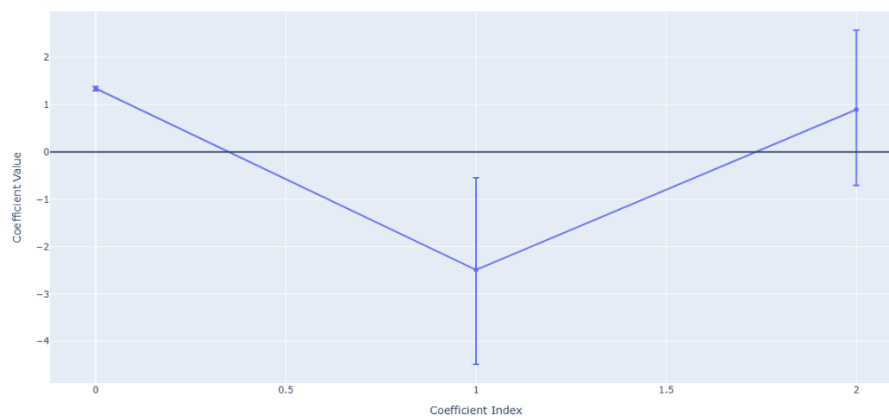
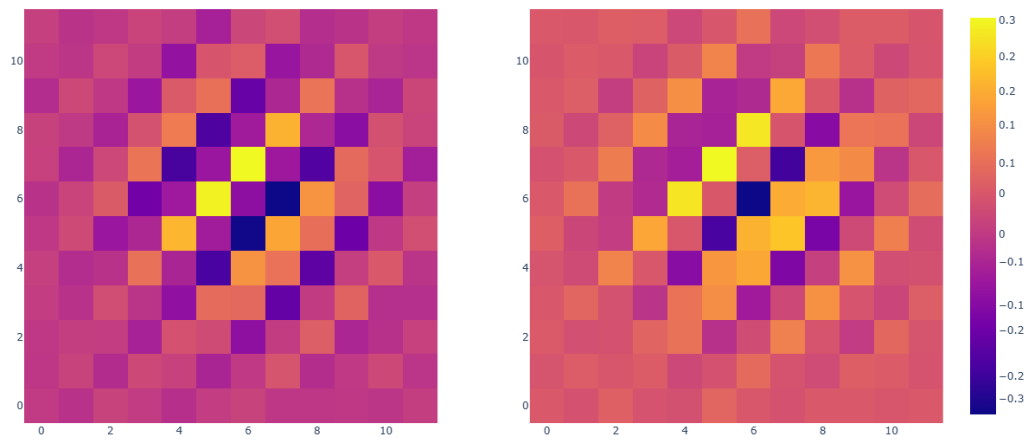


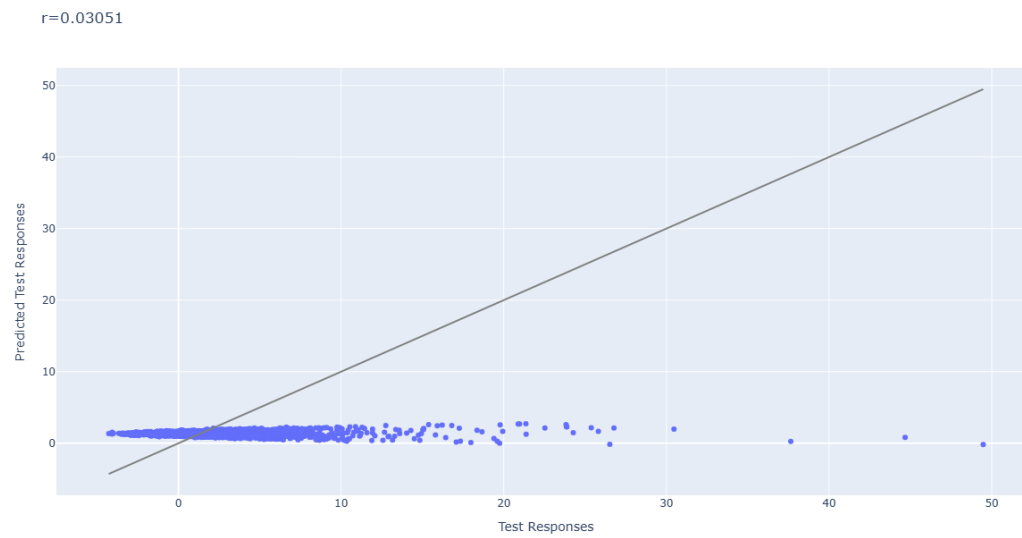
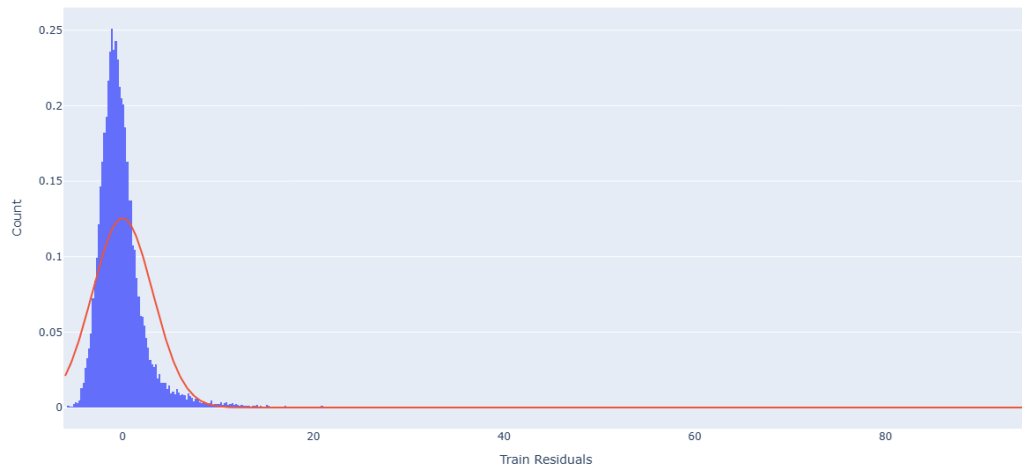


$r=0.92732$



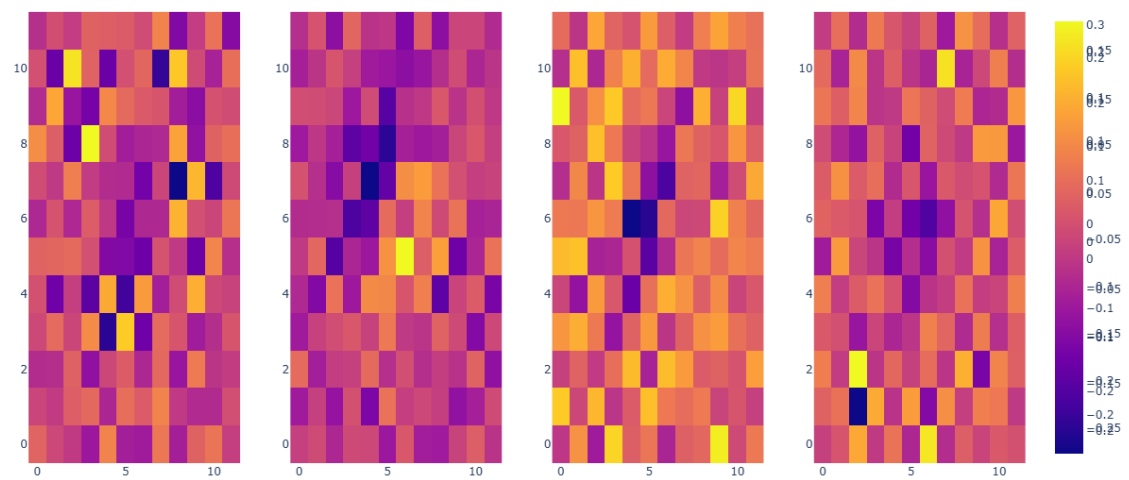
The poorest set of parameters: nRDs=2, order=1

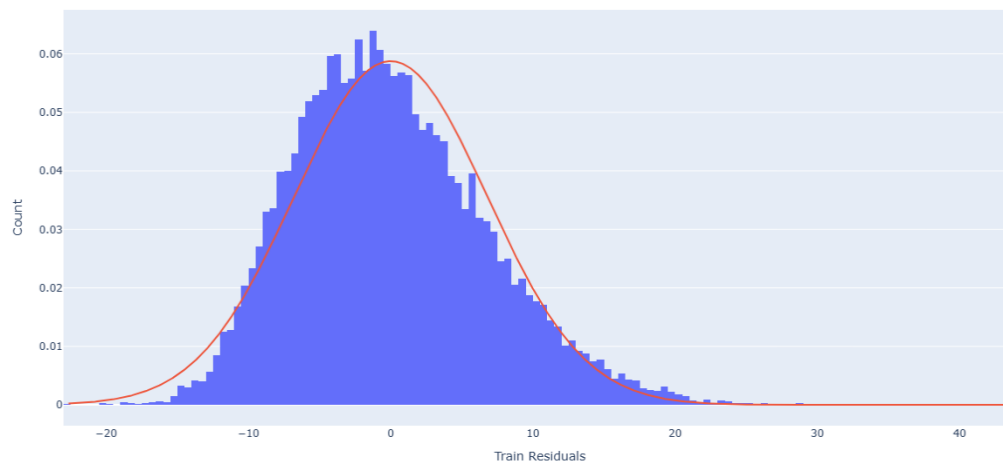
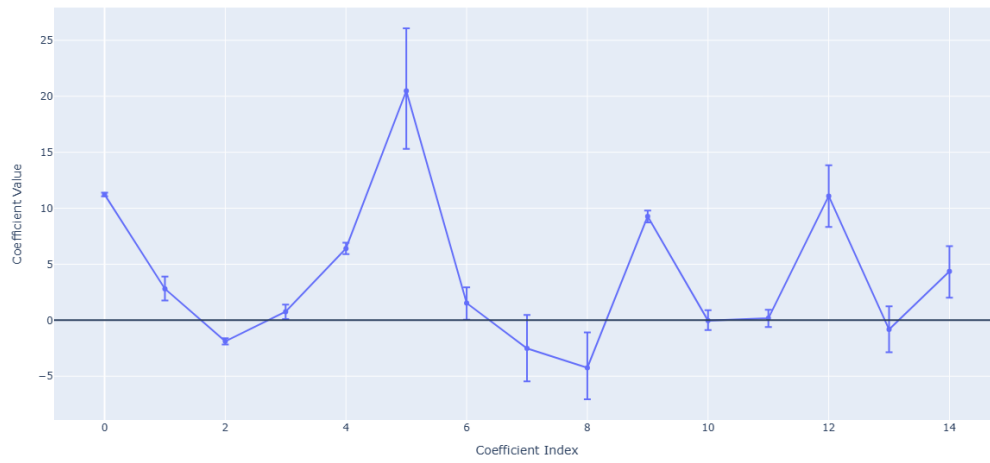




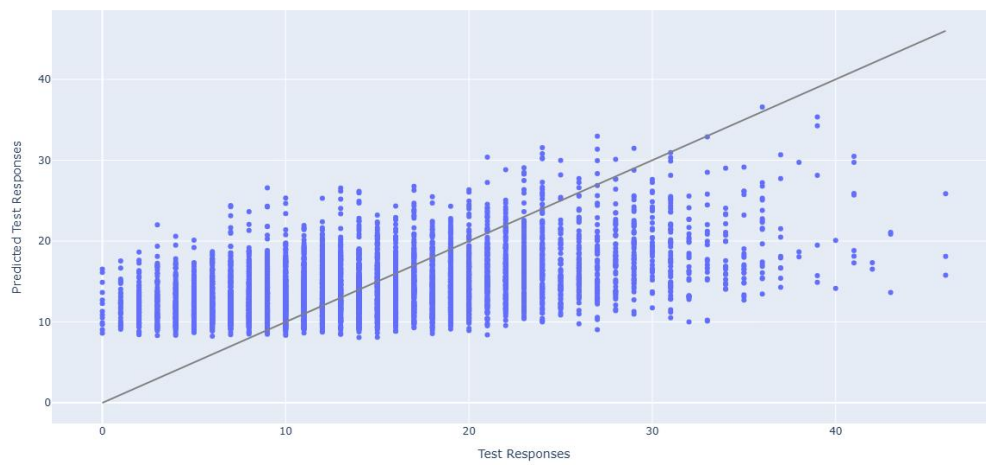
2.

The best set of parameters meet the modelled case: nRDs=4, order=2

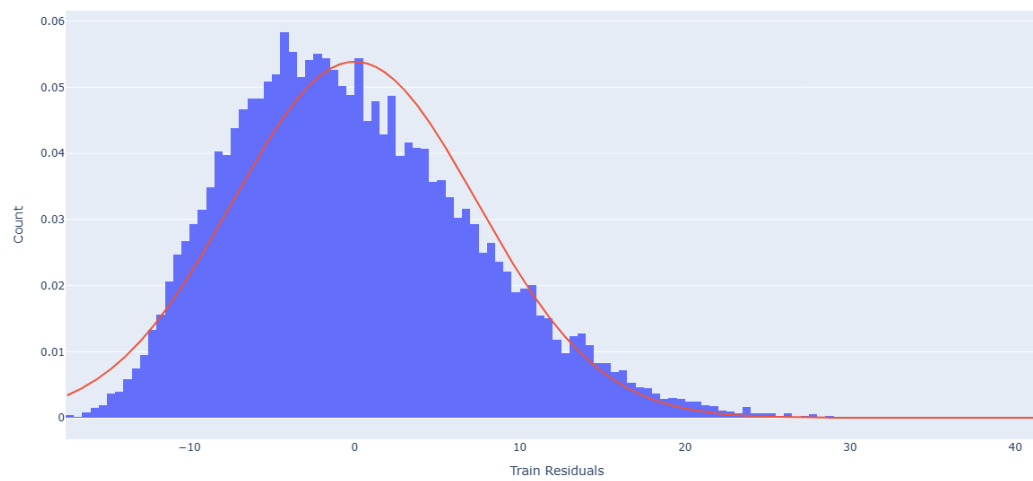
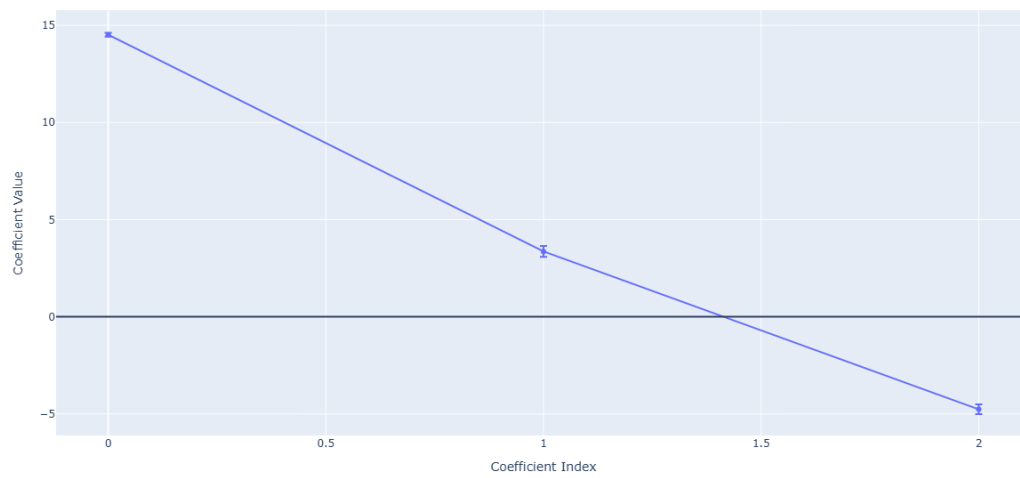
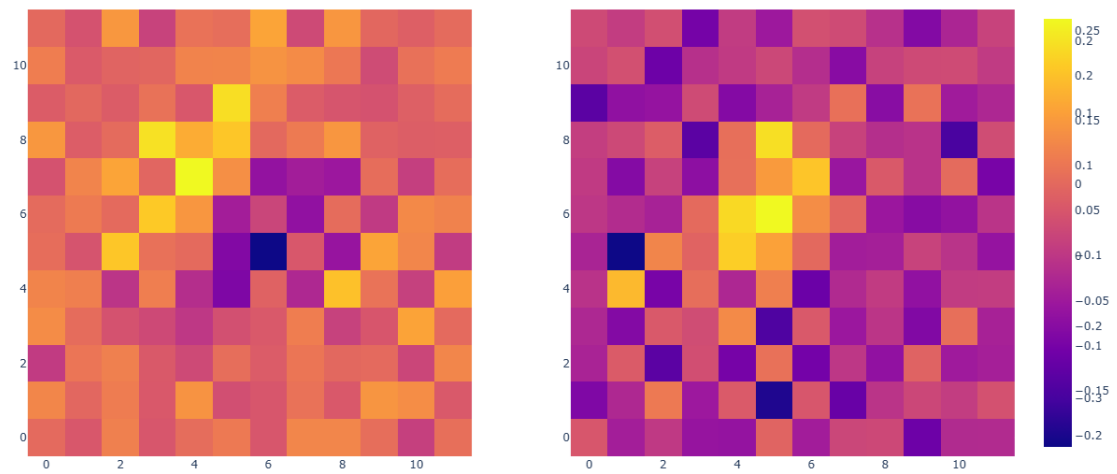




$r=0.49008$



Poor parameters: nRDs=2, order=1



r=0.31567

