Executable Jupyter notebook 1: Simulated data

In [1]:

```
# imports and plotting utility functions
%matplotlib inline
import warnings

import numpy as np
from sklearn.datasets import make_regression
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import ShuffleSplit
from sklearn.linear_model import Lasso
import seaborn as sns
from matplotlib import pylab as plt
import matplotlib as mpl
from statsmodels.regression.linear_model import OLS

warnings.simplefilter('ignore')

def plot_lr(true_coefs, est_coefs, pvals):
    n_feat = len(est_coefs)
    where_sign = pvals < 0.05
    plt.figure(figsize=(15, 7))
    plt.scatter(np.arange(X.shape[1]), true_coefs, color='black', label='true be
tas', alpha=0.5)
    # print non-significant betas
    plt.scatter(np.arange(X.shape[1]), est_coefs, color='red', label='estimated
betas', alpha=0.5)
    # print significant betas
    axes = plt.gca()
    y_max, _ = axes.get_ylim()
    sign_y = np.sum(where_sign) * [y_max]
    plt.scatter(np.arange(X.shape[1])[where_sign], sign_y, color='red', label='s
ignificant at p<0.05', s=150, marker=(5, 1), alpha=0.75, linewidth=3)

    plt.xlabel('input variables')
    #plt.xticks(np.arange(n_feat)[::2], (np.arange(n_feat) + 1)[::2])
    plt.xticks(np.arange(n_feat), (np.arange(n_feat) + 1))
    plt.grid(True)
    plt.title('Linear regression', fontsize=16)
    plt.legend(loc='upper right')

def plot_regr_paths(coefs, accs, nonzeros, C_grid):
    n_cols = 2
    n_rows = 1
    n_verticals = len(coefs)
    n_feat = coefs.shape[1]
    print(n_feat)
```

```
    my_palette = np.array([
        '#F47D7D', '#FBEF69', '#98E466', '#000000',
        '#A7794F', '#CCCCCC', '#85359C', '#FF9300', '#FF0030', 'grey', 'blue', '
salmon', '#4BBCF6',
        'green', 'tomato', 'darkred', 'black', 'cyan', 'lime'
    ])
    my_colors = np.array(['???????'] * coefs.shape[-1])
    i_col = 0
    new_grp_pts_x = []
    new_grp_pts_y = []
    new_grp_pts_col = []
    new_grp_pts_total = []

    for i_vertical, (params, acc, C) in enumerate(zip(
        coefs, accs, C_grid)):
        b_notset = my_colors == '???????'
        b_nonzeros = params == 0
        b_coefs_of_new_grp = np.logical_and(b_notset, b_nonzeros)

        if np.sum(b_coefs_of_new_grp) > 0:
            i_col += 1

            # we found a new subset that became 0
            for new_i in np.where(b_coefs_of_new_grp == True)[0]:
                # color all coefficients of the current group
                cur_col = my_palette[i_col]
                my_colors[new_i] = cur_col

            new_grp_pts_x.append(C)
            new_grp_pts_y.append(acc)
            new_grp_pts_col.append(cur_col)
            new_grp_pts_total.append(np.sum(b_nonzeros))

X_colnames = np.arange(0, n_feat + 1, 1)
subplot_xlabel = '#nonzero coefficients'

f, axarr = plt.subplots(nrows=n_rows, ncols=n_cols,
    figsize=(15, 10), facecolor='white')
t, i_col = 0, 0

for i_line in range(X.shape[-1]):
    axarr[i_col].plot(np.log10(C_grid),
        coefs[:, i_line], label=X_colnames[i_line],
            color=my_colors[i_line], linewidth=1.5)

axarr[i_col].set_xlabel(subplot_xlabel, fontsize=10)
axarr[i_col].legend(loc='lower left', fontsize=8.5, markerscale=10)
axarr[0].grid(True)

axarr[0].set_title('LASSO: Groups of selected variables', fontsize=16)
axarr[0].set_xticks(np.log10(C_grid))
axarr[0].set_xticklabels(nonzeros)
```

```
    axarr[1].plot(np.arange(len(accs)), accs, color='#000000',
                      linewidth=3)
    axarr[1].set_ylim(0.0, 1.05)
    axarr[1].grid(True)
    axarr[1].set_xticks(np.arange(n_verticals))
    axarr[1].set_xticklabels(nonzeros)
    axarr[1].set_xlabel(subplot_xlabel, fontsize=10)
    axarr[1].set_title('LASSO: Out-of-sample accuracy ($R^2$ score)', fontsize=1
6)
    return my_colors

def clip_pvals(pvals):
    pvals[pvals == 0] = np.finfo(pvals.dtype).eps
    return pvals
```

In [2]:

```
# statistical helper functions
def compute_Lasso_regpath(X, y, C_grid):
    coef_list2 = []
    acc_list2 = []
    nonzero_list2 = []
    for i_step, my_C in enumerate(C_grid):
        sample_accs = []
        sample_coef = []
        for i_subsample in range(100):
            folder = ShuffleSplit(n_splits=100, test_size=0.1,
                                  random_state=i_subsample)
            train_inds, test_inds = next(iter(folder.split(X)))

            clf = Lasso(alpha=my_C, random_state=i_subsample)

            clf.fit(X[train_inds, :], y[train_inds])
            acc = clf.score(X[test_inds, :], y[test_inds])

            sample_accs.append(acc)
            sample_coef.append(clf.coef_)

        mean_coefs = np.mean(np.array(sample_coef), axis=0)
        coef_list2.append(mean_coefs)
        acc_list2.append(np.mean(sample_accs))
        notzero = np.count_nonzero(mean_coefs)
        print("alpha: %.4f acc: %.2f active_coefs: %i" % (my_C, acc, notzero))
        nonzero_list2.append(notzero)
    return np.array(coef_list2), np.array(acc_list2), np.array(nonzero_list2)
```

In [3]:

```
def infpred_plot(unbiased_acc_list, lr_pvalues, coef_list, feat_names, acc_offse
t=0.1, annot_ha='center'):
    fig = plt.figure(figsize=(9, 9))
    sorter = unbiased_acc_list.argsort()[::-1]
```

```
colors = plt.cm.viridis_r(np.linspace(0.1, 0.9, len(sorter)))


unique_nonzero = {}
size = 20
for ii, idx in enumerate(sorter):
    acc = unbiased_acc_list[idx]
    non_zero = np.where(coef_list[idx])[0]
    if tuple(non_zero) not in unique_nonzero:
        unique_nonzero[tuple(non_zero)] = non_zero
    else:
        print('skipping', ii)
        continue

    xx = -np.log10(lr_pvalues[non_zero])
    this_acc = np.array([acc] * len(xx))
    size *= 0.9
    plt.plot(xx + np.random.sample(len(xx)) * 0.01,
             this_acc,
             marker='o', linestyle='None',
             color=colors[ii], zorder=-ii,
             alpha=0.9,
             mfc='None',
             mew=1,
             markersize=size)
    if ii == 0:
        psorter = np.argsort(lr_pvalues)
        feat_names_ = [feat_names[kk] for kk in psorter]
        xx2 = -np.log10(lr_pvalues[psorter])
        for jj, (this_name, this_x) in enumerate(zip(feat_names_, xx2)):
            print(this_x)
            plt.annotate(
                this_name, xy=(this_x, acc + acc_offset),
                xycoords='data', rotation=90,
                verticalalignment='bottom' if jj % 2 else 'top',
                ha=annot_ha,
                fontsize=14)
plt.axvline(
    -np.log10(0.05), color='red', linestyle='--', linewidth=3)
plt.annotate('p < 0.05', xy=(-np.log10(0.045), 0.03), color='red', fontsize=
16)
plt.xlabel(r'significance [$-log_{10}(p)$]', fontsize=20, fontweight=150)
plt.ylabel(r'prediction [$R^2$]', fontsize=20, fontweight=150)
plt.ylim(0, 1)
plt.grid(True)
ax = plt.gca()
ax.set_yticks([0., 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1])
ax.set_yticks(np.arange(0.01, 1, 0.01), minor=True);
return fig
```

In [4]:

```
simulations = []

def run_lasso(X, y, comment, n_samples, n_feat,
              n_feat_relevant, C_grid):
    # ordinary least squares
    model = OLS(y, X)
    res = model.fit()
    lr_coefs = res.params
    lr_pvalues = clip_pvals(res.pvalues)

    # compute Lasso regularization paths
    coef_list, acc_list, nonzero_list = compute_Lasso_regpath(X, y, C_grid)

    plot_lr(true_coefs, lr_coefs, lr_pvalues)
    path_colors = plot_regr_paths(coef_list, acc_list, nonzero_list, C_grid)
    feat_names = ['' for _ in range(X.shape[1])]
    infpred_plot(unbiased_acc_list=acc_list,
                 lr_pvalues=lr_pvalues, coef_list=coef_list,
                 feat_names=feat_names, acc_offset=0.1)
    simulations.append(
        dict(X=X, y=y, comment=comment,
             n_samples=n_samples, n_feat=n_feat,
             n_feat_relevant=n_feat_relevant,
             pvals=lr_pvalues,
             coefs=coef_list,
             acc_list=acc_list))
```

100 samples, 40 variables, error = N(0, 1)

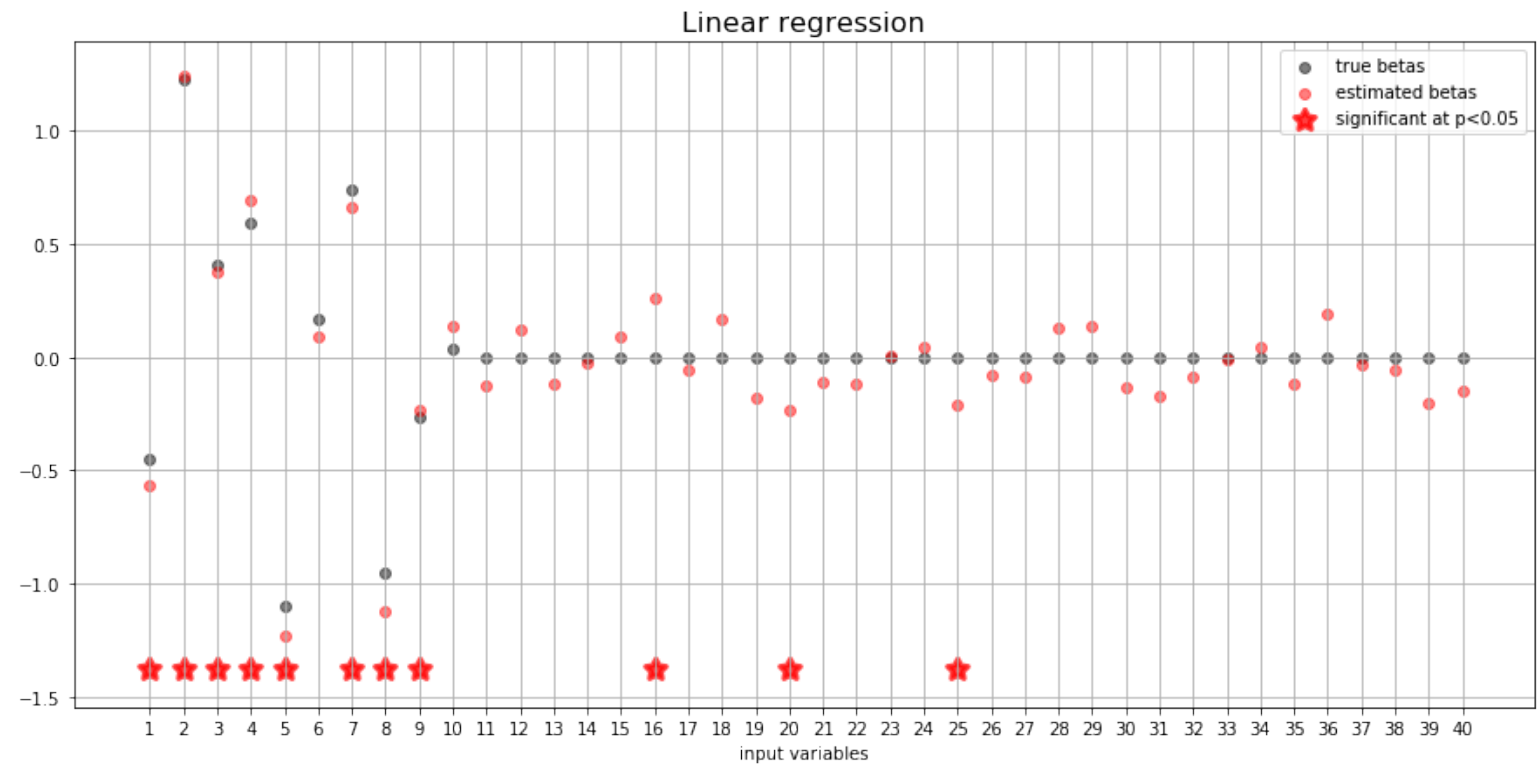
In [5]:

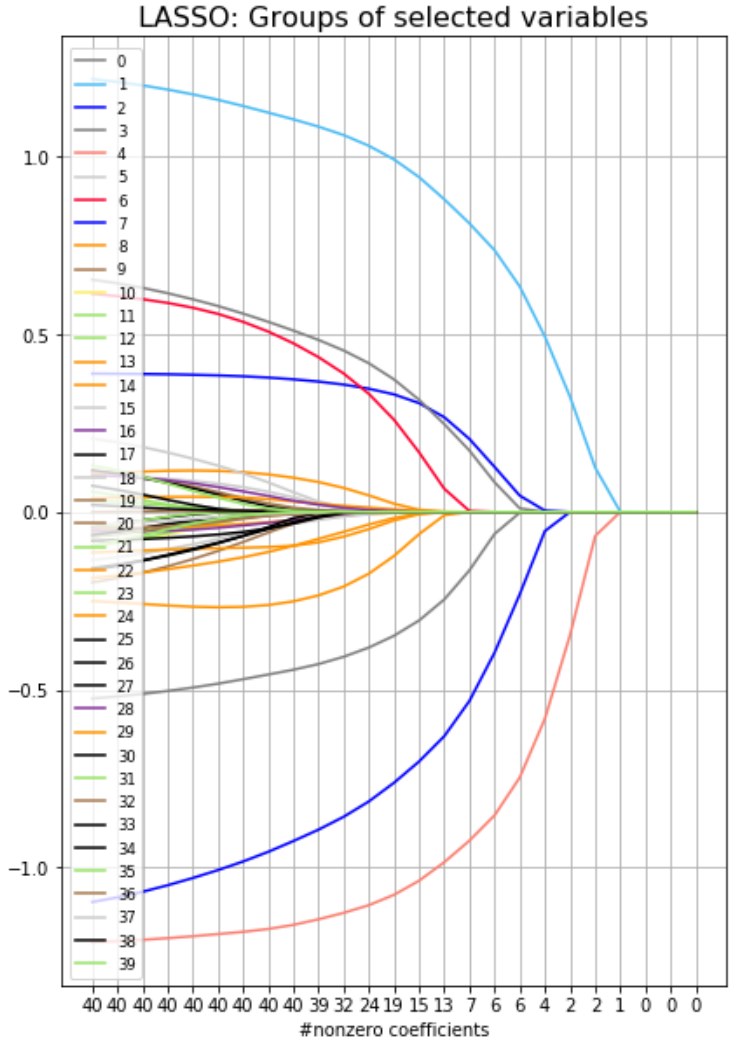
```
comment = "dataset: 10/40 variables relevant, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 100
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)
y = (true_coefs * X).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 0.5, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

```
alpha: 0.0100 acc: 0.57 active_coefs: 40
```

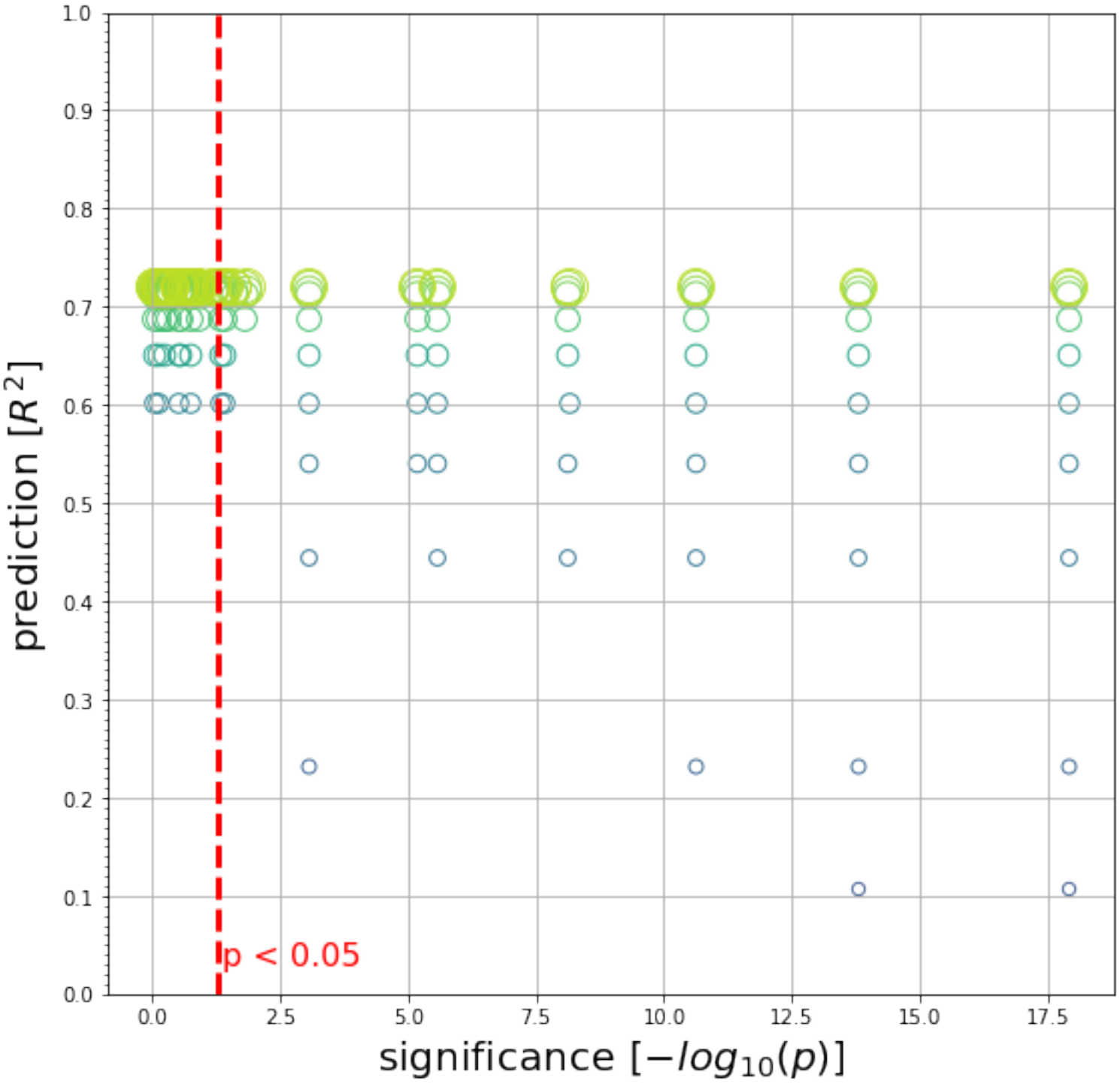
```
alpha: 0.0127 acc: 0.60 active_coefs: 40
alpha: 0.0162 acc: 0.64 active_coefs: 40
alpha: 0.0205 acc: 0.68 active_coefs: 40
alpha: 0.0261 acc: 0.72 active_coefs: 40
alpha: 0.0332 acc: 0.76 active_coefs: 40
alpha: 0.0422 acc: 0.79 active_coefs: 40
alpha: 0.0536 acc: 0.80 active_coefs: 40
alpha: 0.0681 acc: 0.80 active_coefs: 40
alpha: 0.0866 acc: 0.80 active_coefs: 39
alpha: 0.1101 acc: 0.80 active_coefs: 32
alpha: 0.1399 acc: 0.78 active_coefs: 24
alpha: 0.1778 acc: 0.76 active_coefs: 19
alpha: 0.2260 acc: 0.70 active_coefs: 15
alpha: 0.2873 acc: 0.62 active_coefs: 13
alpha: 0.3652 acc: 0.55 active_coefs: 7
alpha: 0.4642 acc: 0.44 active_coefs: 6
alpha: 0.5900 acc: 0.32 active_coefs: 6
alpha: 0.7499 acc: 0.22 active_coefs: 4
alpha: 0.9532 acc: 0.04 active_coefs: 2
alpha: 1.2115 acc: -0.19 active_coefs: 2
alpha: 1.5399 acc: -0.22 active_coefs: 1
alpha: 1.9573 acc: -0.22 active_coefs: 0
alpha: 2.4879 acc: -0.22 active_coefs: 0
alpha: 3.1623 acc: -0.22 active_coefs: 0
40
17.8985881323
13.7654376568
10.6001082352
8.11300856465
5.54904001773
5.17676654541
3.04473973171
1.81241157401
1.57344918174
1.39634148784
1.30972257689
1.15425855311
0.902375816637
0.833811642171
0.823161052389
0.72767976368
0.727464102672
0.655842393759
0.648187468074
0.636416192402
0.58041470526
0.559717763899
0.547163763916
0.531714903269
0.504594418577
0.495013119587
0.484304813059
0.350920961614
```

```
0.350231694279
0.30542518911
0.304754254536
0.286598938232
0.223505333064
0.207928336501
0.144791264036
0.141352670361
0.134192620259
0.110019292558
0.0494182670069
0.0238859141628
skipping 4
skipping 5
skipping 7
skipping 8
skipping 9
skipping 11
skipping 12
skipping 13
skipping 17
skipping 20
skipping 22
skipping 23
```




LASSO: Groups of selected variables
1.0
0.5
0.0
−0.5
−1.0
40 40 40 40 40 40 40 40 40 39 32 24 19 15 13 7 6 6 4 2 2 1 0 0 0
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
1.0
0.8
0.6
0.4
0.2
0.0
40 40 40 40 40 40 40 40 40 39 32 24 19 15 13 7 6 6 4 2 2 1 0 0 0
#nonzero coefficients



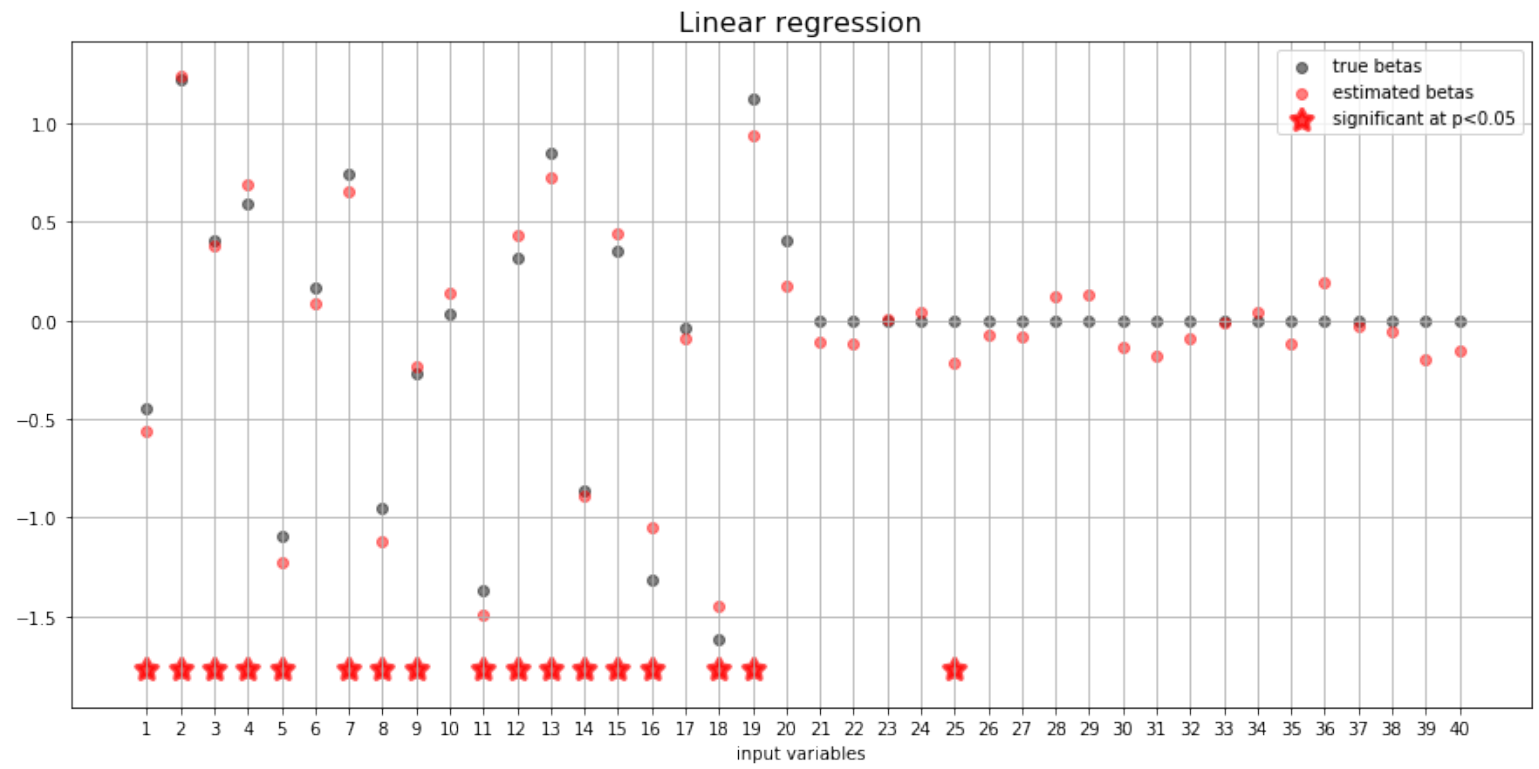
In [6]:

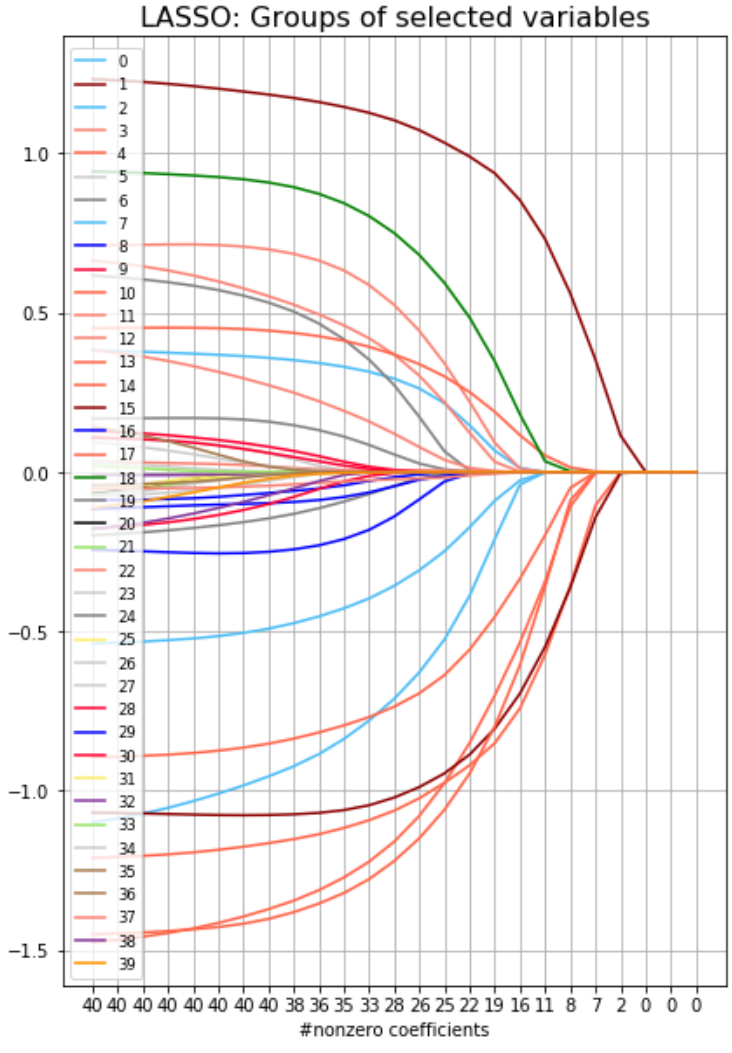
```
comment = "dataset: 20/40 variables relevant, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 100
n_feat = 40
n_feat_relevant = 20
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)
y = (true_coefs * X).sum(axis=1) + epsilon

C_grid = np.logspace(-2, 0.5, 25)
run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

```
alpha: 0.0100 acc: 0.85 active_coefs: 40
alpha: 0.0127 acc: 0.86 active_coefs: 40
alpha: 0.0162 acc: 0.87 active_coefs: 40
```

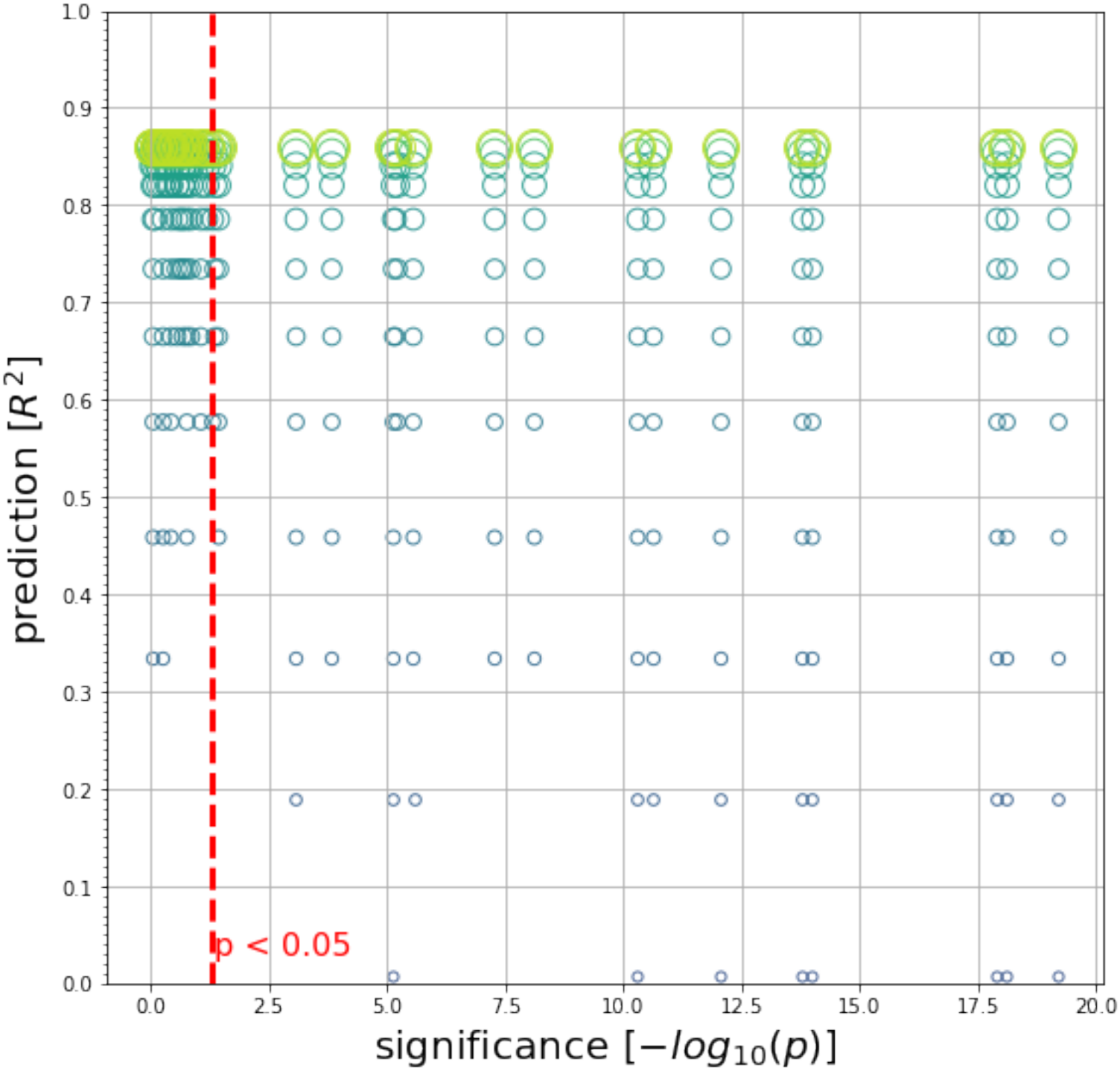
```
alpha: 0.0205 acc: 0.87 active_coefs: 40
alpha: 0.0261 acc: 0.88 active_coefs: 40
alpha: 0.0332 acc: 0.89 active_coefs: 40
alpha: 0.0422 acc: 0.91 active_coefs: 40
alpha: 0.0536 acc: 0.91 active_coefs: 40
alpha: 0.0681 acc: 0.91 active_coefs: 38
alpha: 0.0866 acc: 0.90 active_coefs: 36
alpha: 0.1101 acc: 0.88 active_coefs: 35
alpha: 0.1399 acc: 0.85 active_coefs: 33
alpha: 0.1778 acc: 0.79 active_coefs: 28
alpha: 0.2260 acc: 0.71 active_coefs: 26
alpha: 0.2873 acc: 0.63 active_coefs: 25
alpha: 0.3652 acc: 0.52 active_coefs: 22
alpha: 0.4642 acc: 0.43 active_coefs: 19
alpha: 0.5900 acc: 0.31 active_coefs: 16
alpha: 0.7499 acc: 0.18 active_coefs: 11
alpha: 0.9532 acc: 0.00 active_coefs: 8
alpha: 1.2115 acc: -0.07 active_coefs: 7
alpha: 1.5399 acc: -0.07 active_coefs: 2
alpha: 1.9573 acc: -0.04 active_coefs: 0
alpha: 2.4879 acc: -0.04 active_coefs: 0
alpha: 3.1623 acc: -0.04 active_coefs: 0
40
19.2001348711
18.1041494893
17.8985881323
13.9890792566
13.7654376568
12.0549348919
10.6001082352
10.283474044
8.11300856465
7.243931239
5.54904001773
5.17676654541
5.1181412403
3.79538075164
3.04473973171
1.39634148784
1.30972257689
1.15425855311
1.03145382869
0.833811642171
0.72767976368
0.727464102672
0.655842393759
0.648187468074
0.636416192402
0.547163763916
0.531714903269
0.484304813059
0.403881475164
0.350920961614
```

```
0.350231694279
0.30542518911
0.304754254536
0.286598938232
0.223505333064
0.144791264036
0.141352670361
0.134192620259
0.0494182670069
0.0238859141628
skipping 1
skipping 3
skipping 4
skipping 6
skipping 7
skipping 8
skipping 9
skipping 23
skipping 24
```




LASSO: Groups of selected variables
1.0
0.5
0.0
−0.5
−1.0
−1.5
40 40 40 40 40 40 40 40 38 36 35 33 28 26 25 22 19 16 11 8 7 2 0 0 0
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
1.0
0.8
0.6
0.4
0.2
0.0
40 40 40 40 40 40 40 40 38 36 35 33 28 26 25 22 19 16 11 8 7 2 0 0 0
#nonzero coefficients



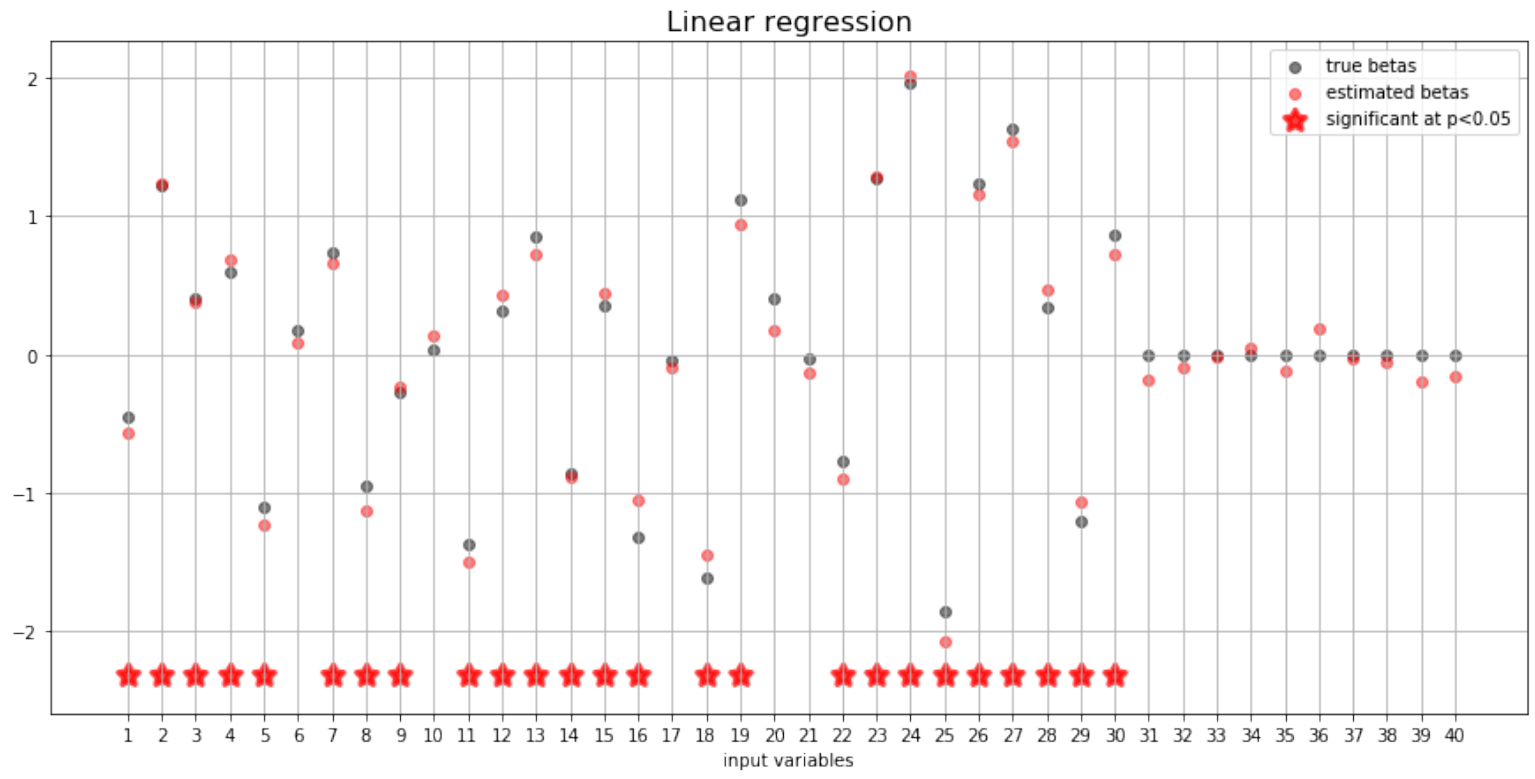
In [7]:

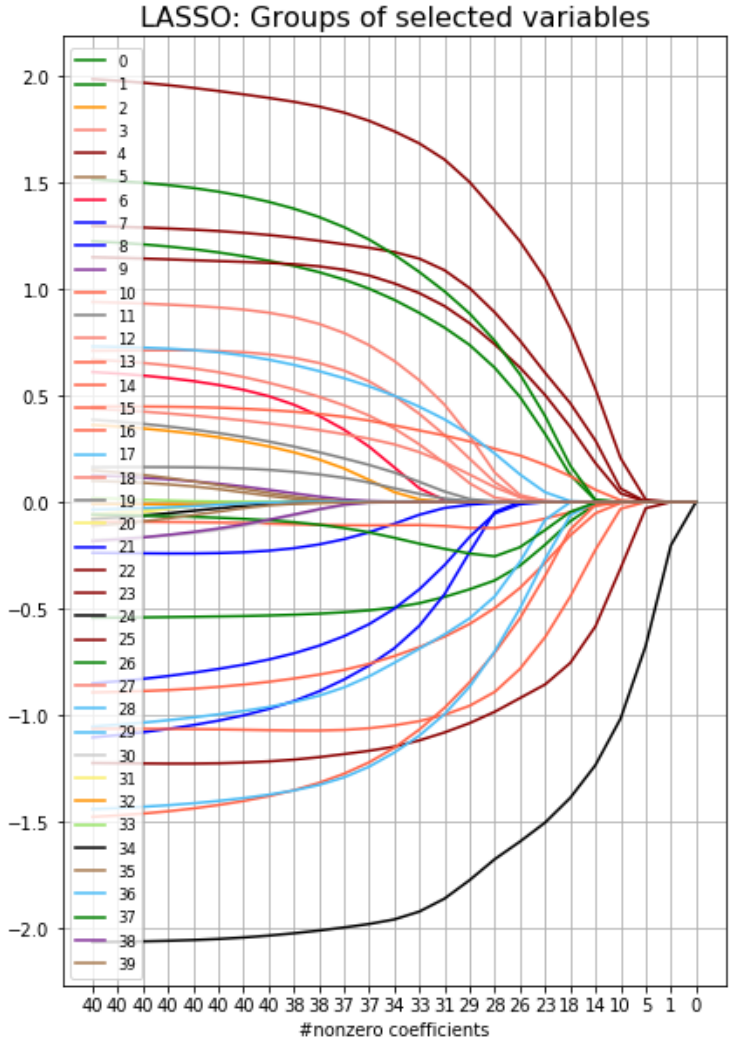
```
comment = "dataset: 30/40 variables relevant, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 100
n_feat = 40
n_feat_relevant = 30
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)
y = (true_coefs * X).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 0.5, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

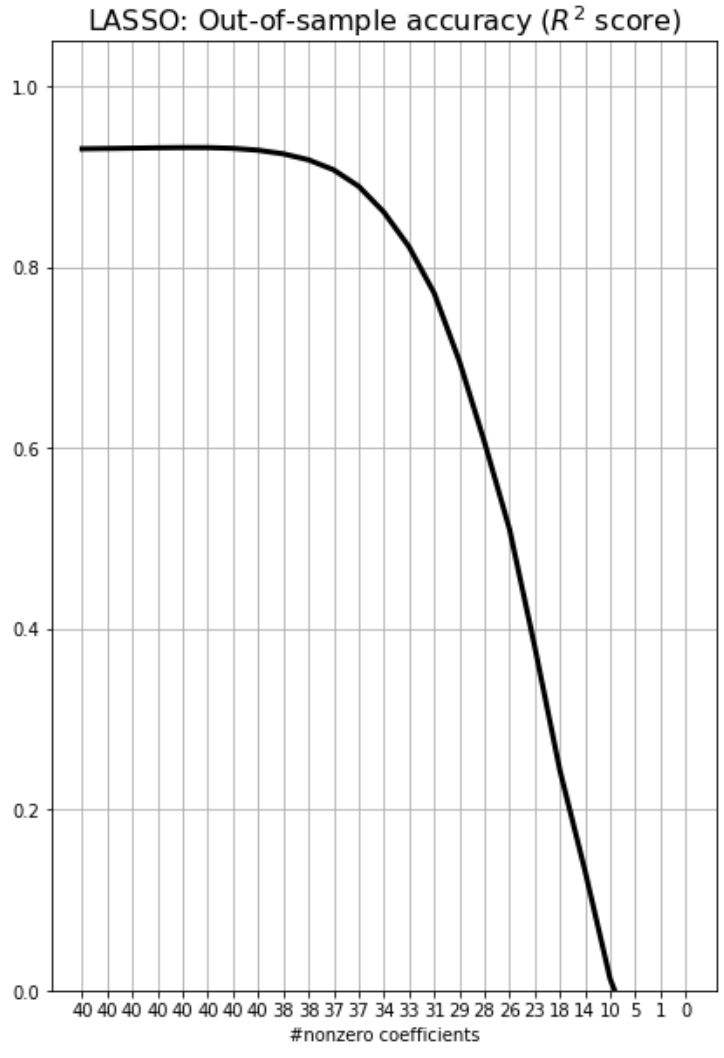
```
alpha: 0.0100 acc: 0.95 active_coefs: 40
alpha: 0.0127 acc: 0.95 active_coefs: 40
alpha: 0.0162 acc: 0.95 active_coefs: 40
```

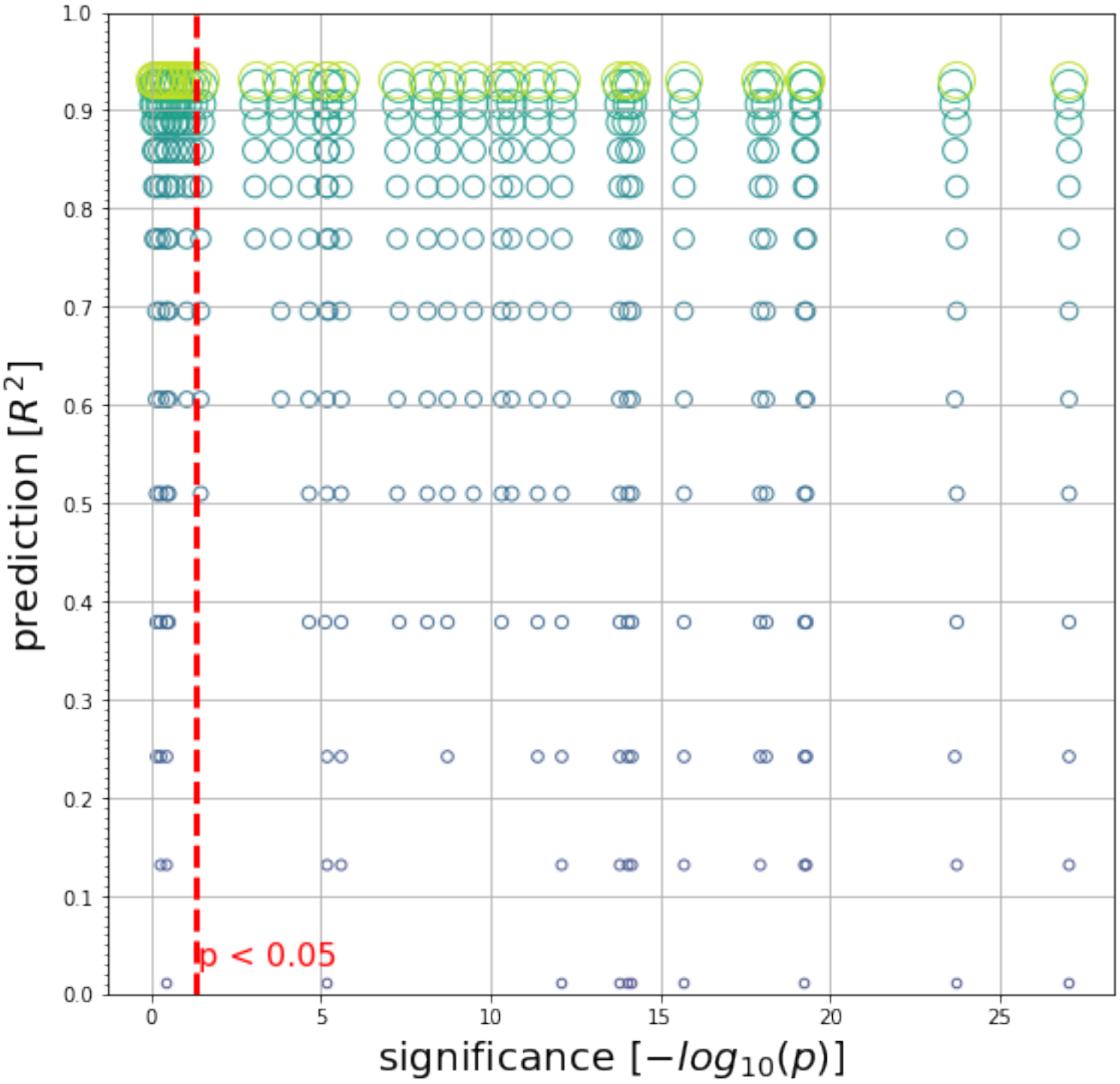
```
alpha: 0.0205 acc: 0.95 active_coefs: 40
alpha: 0.0261 acc: 0.96 active_coefs: 40
alpha: 0.0332 acc: 0.96 active_coefs: 40
alpha: 0.0422 acc: 0.96 active_coefs: 40
alpha: 0.0536 acc: 0.96 active_coefs: 40
alpha: 0.0681 acc: 0.96 active_coefs: 38
alpha: 0.0866 acc: 0.96 active_coefs: 38
alpha: 0.1101 acc: 0.95 active_coefs: 37
alpha: 0.1399 acc: 0.93 active_coefs: 37
alpha: 0.1778 acc: 0.90 active_coefs: 34
alpha: 0.2260 acc: 0.86 active_coefs: 33
alpha: 0.2873 acc: 0.81 active_coefs: 31
alpha: 0.3652 acc: 0.75 active_coefs: 29
alpha: 0.4642 acc: 0.68 active_coefs: 28
alpha: 0.5900 acc: 0.58 active_coefs: 26
alpha: 0.7499 acc: 0.43 active_coefs: 23
alpha: 0.9532 acc: 0.22 active_coefs: 18
alpha: 1.2115 acc: -0.06 active_coefs: 14
alpha: 1.5399 acc: -0.25 active_coefs: 10
alpha: 1.9573 acc: -0.36 active_coefs: 5
alpha: 2.4879 acc: -0.45 active_coefs: 1
alpha: 3.1623 acc: -0.45 active_coefs: 0
40
27.0435011033
23.6884396179
19.2837128916
19.2001348711
18.1041494893
17.8985881323
15.6634560403
14.1567924463
13.9890792566
13.7654376568
12.0549348919
11.3353540548
10.6001082352
10.283474044
9.45165943022
8.7061775904
8.11300856465
7.243931239
5.54904001773
5.17676654541
5.1181412403
4.62032233443
3.79538075164
3.04473973171
1.39634148784
1.15425855311
1.03145382869
0.833811642171
0.727464102672
0.648187468074
```

```
0.636416192402
0.531714903269
0.454963495849
0.403881475164
0.304754254536
0.286598938232
0.223505333064
0.144791264036
0.134192620259
0.049418267007
skipping 1
skipping 2
skipping 3
skipping 4
skipping 5
skipping 6
skipping 7
skipping 9
```




LASSO: Groups of selected variables
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
40 40 40 40 40 40 40 40 38 38 37 37 34 33 31 29 28 26 23 18 14 10 5 1 0
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
1.0
0.8
0.6
0.4
0.2
0.0
40 40 40 40 40 40 40 40 38 38 37 37 34 33 31 29 28 26 23 18 14 10 5 1 0
#nonzero coefficients



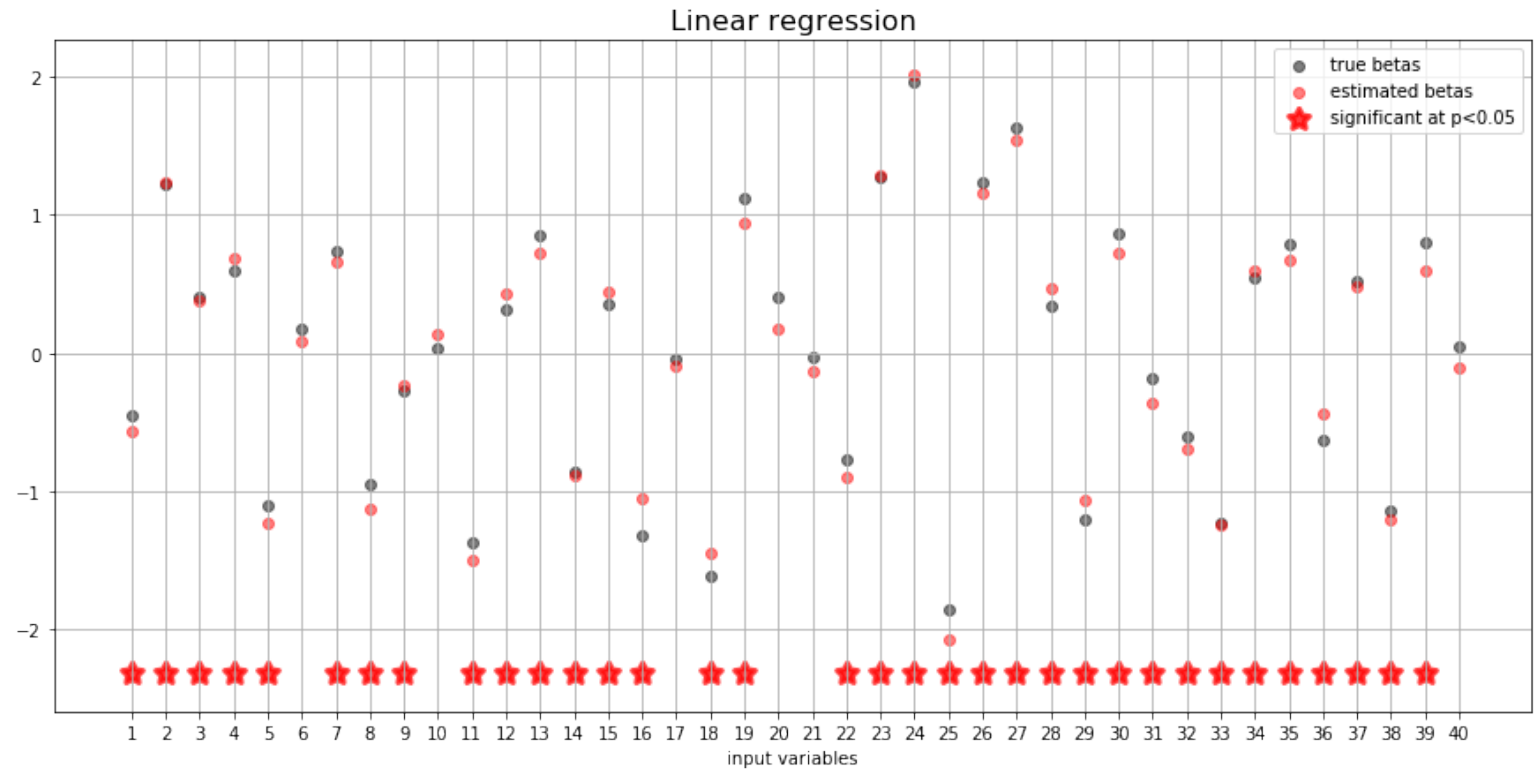
In [8]:

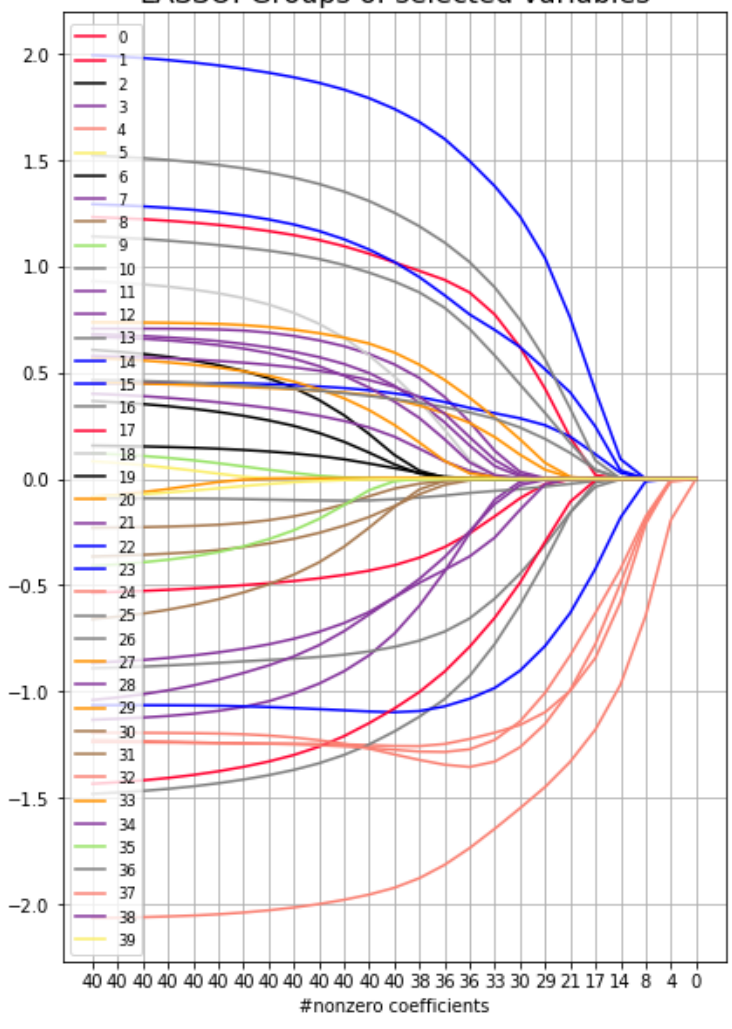
```
comment = "dataset: 40/40 variables relevant, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 100
n_feat = 40
n_feat_relevant = 40
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)
y = (true_coefs * X).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 0.5, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

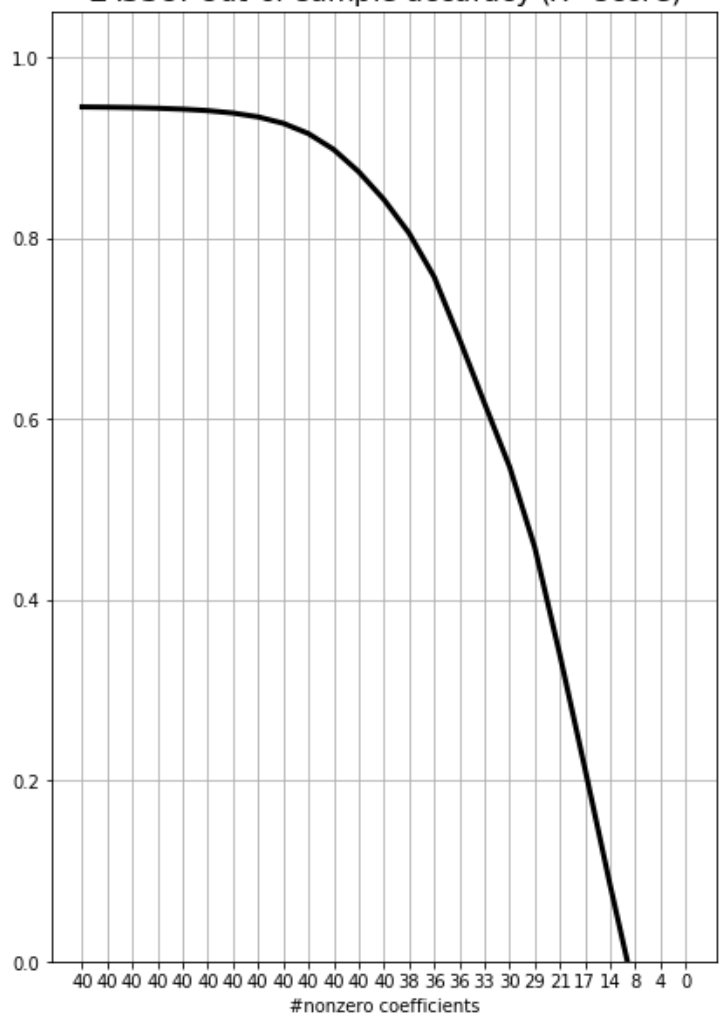
```
alpha: 0.0100 acc: 0.97 active_coefs: 40
alpha: 0.0127 acc: 0.97 active_coefs: 40
alpha: 0.0162 acc: 0.97 active_coefs: 40
```

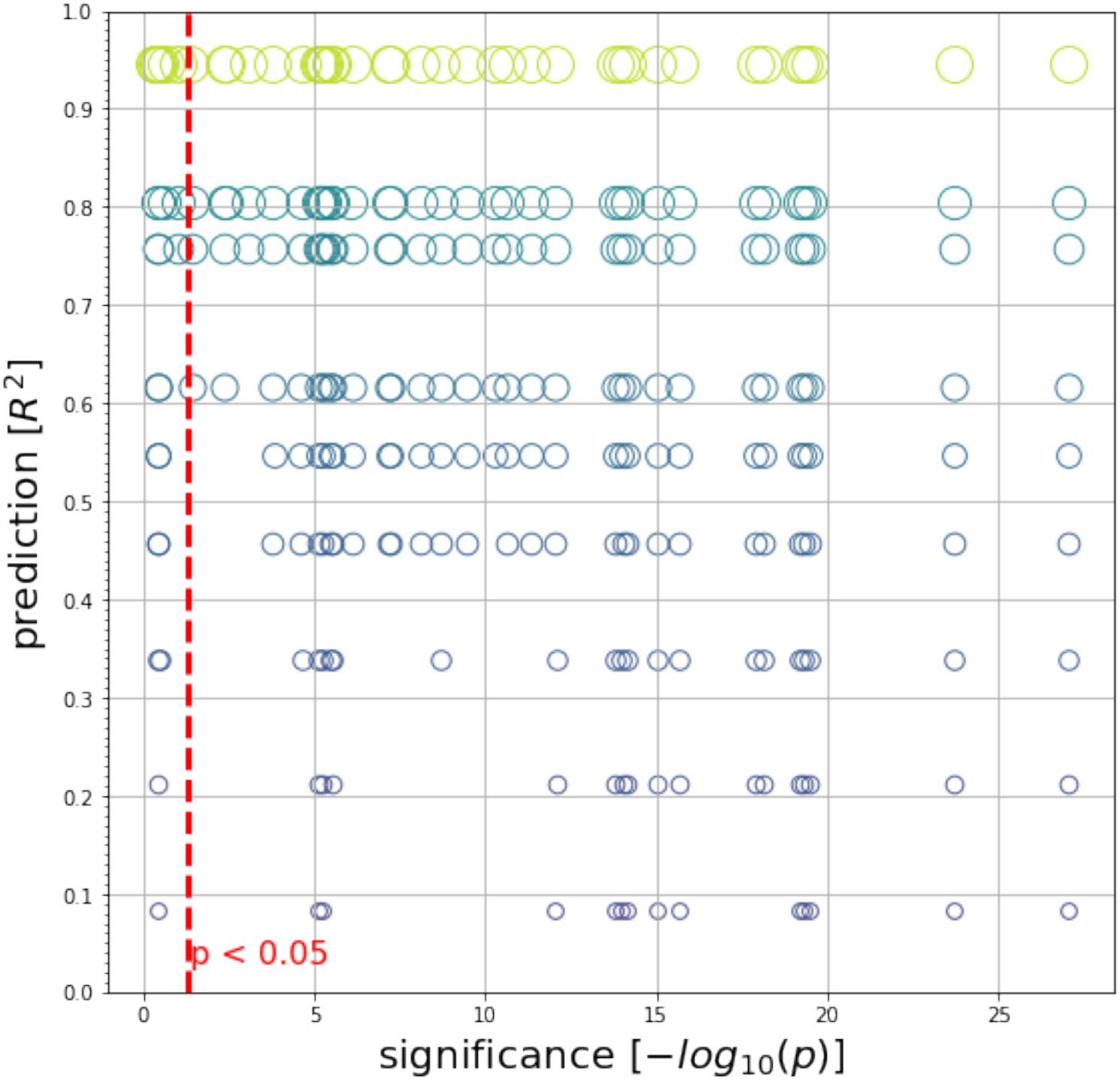
```
alpha: 0.0205 acc: 0.97 active_coefs: 40
alpha: 0.0261 acc: 0.97 active_coefs: 40
alpha: 0.0332 acc: 0.98 active_coefs: 40
alpha: 0.0422 acc: 0.98 active_coefs: 40
alpha: 0.0536 acc: 0.98 active_coefs: 40
alpha: 0.0681 acc: 0.98 active_coefs: 40
alpha: 0.0866 acc: 0.97 active_coefs: 40
alpha: 0.1101 acc: 0.96 active_coefs: 40
alpha: 0.1399 acc: 0.94 active_coefs: 40
alpha: 0.1778 acc: 0.92 active_coefs: 40
alpha: 0.2260 acc: 0.88 active_coefs: 38
alpha: 0.2873 acc: 0.82 active_coefs: 36
alpha: 0.3652 acc: 0.75 active_coefs: 36
alpha: 0.4642 acc: 0.68 active_coefs: 33
alpha: 0.5900 acc: 0.59 active_coefs: 30
alpha: 0.7499 acc: 0.47 active_coefs: 29
alpha: 0.9532 acc: 0.30 active_coefs: 21
alpha: 1.2115 acc: 0.09 active_coefs: 17
alpha: 1.5399 acc: -0.11 active_coefs: 14
alpha: 1.9573 acc: -0.25 active_coefs: 8
alpha: 2.4879 acc: -0.31 active_coefs: 4
alpha: 3.1623 acc: -0.31 active_coefs: 0
40
27.0435011033
23.6884396179
19.4601541668
19.2837128916
19.2001348711
18.1041494893
17.8985881323
15.6634560403
15.0310351063
14.1567924463
13.9890792566
13.7654376568
12.0549348919
11.3353540548
10.6001082352
10.283474044
9.45165943022
8.7061775904
8.11300856465
7.243931239
7.17773431258
6.08559747734
5.54904001773
5.45082906449
5.29573007252
5.25207388611
5.17676654541
5.1181412403
4.62032233443
3.79538075164
```

```
3.04473973171
2.42409762423
2.38320513046
1.39634148784
1.03145382869
0.636416192402
0.454963495849
0.403881475164
0.3995473636
0.304754254536
skipping 1
skipping 2
skipping 3
skipping 4
skipping 5
skipping 6
skipping 7
skipping 8
skipping 9
skipping 10
skipping 11
skipping 12
skipping 15
```




LASSO: Groups of selected variables
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
40 40 40 40 40 40 40 40 40 40 40 40 40 38 36 36 33 30 29 21 17 14 8 4 0
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
1.0
0.8
0.6
0.4
0.2
0.0
40 40 40 40 40 40 40 40 40 40 40 40 40 38 36 36 33 30 29 21 17 14 8 4 0
#nonzero coefficients



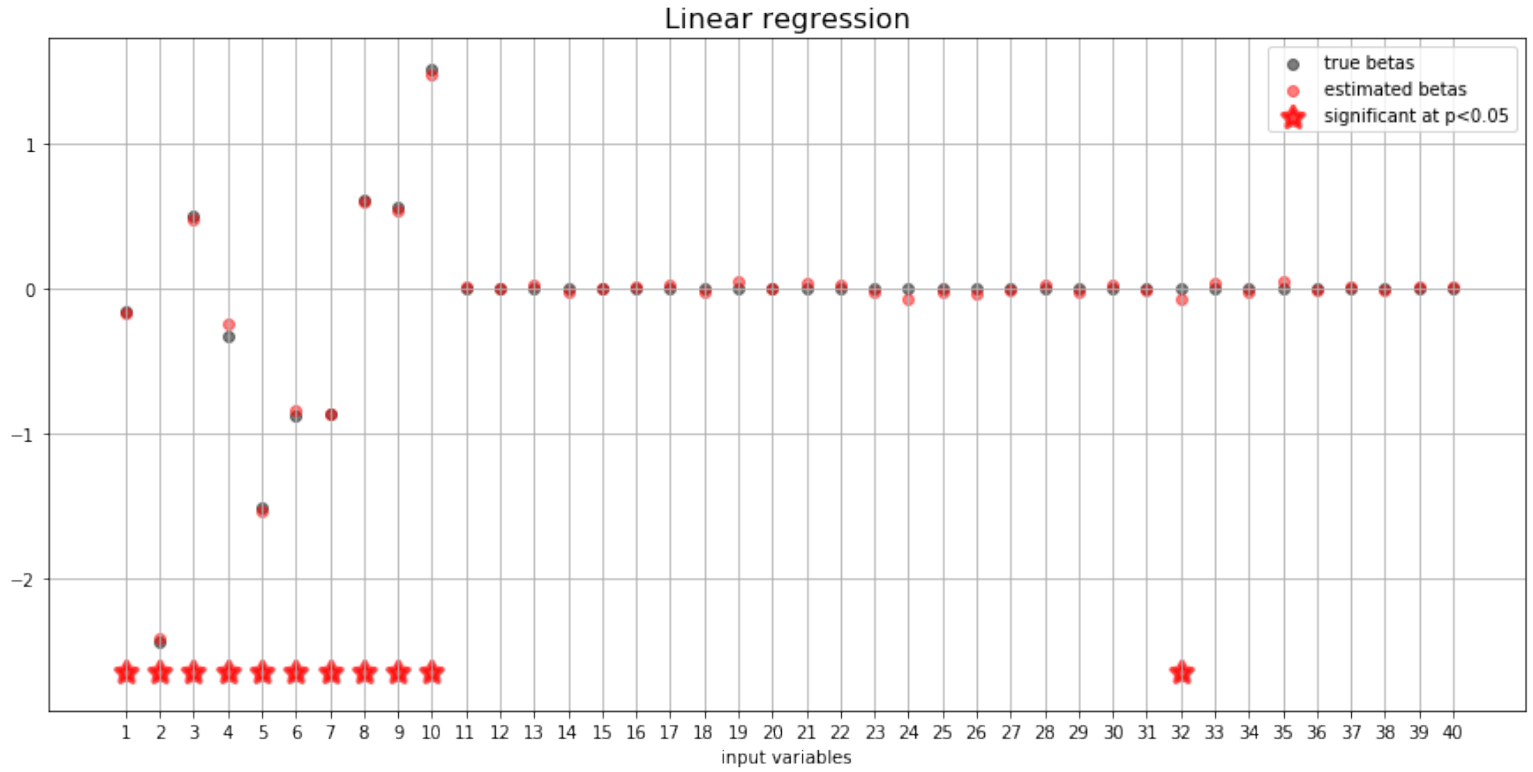
1000 samples, 40 variables, error = N(0, 1)

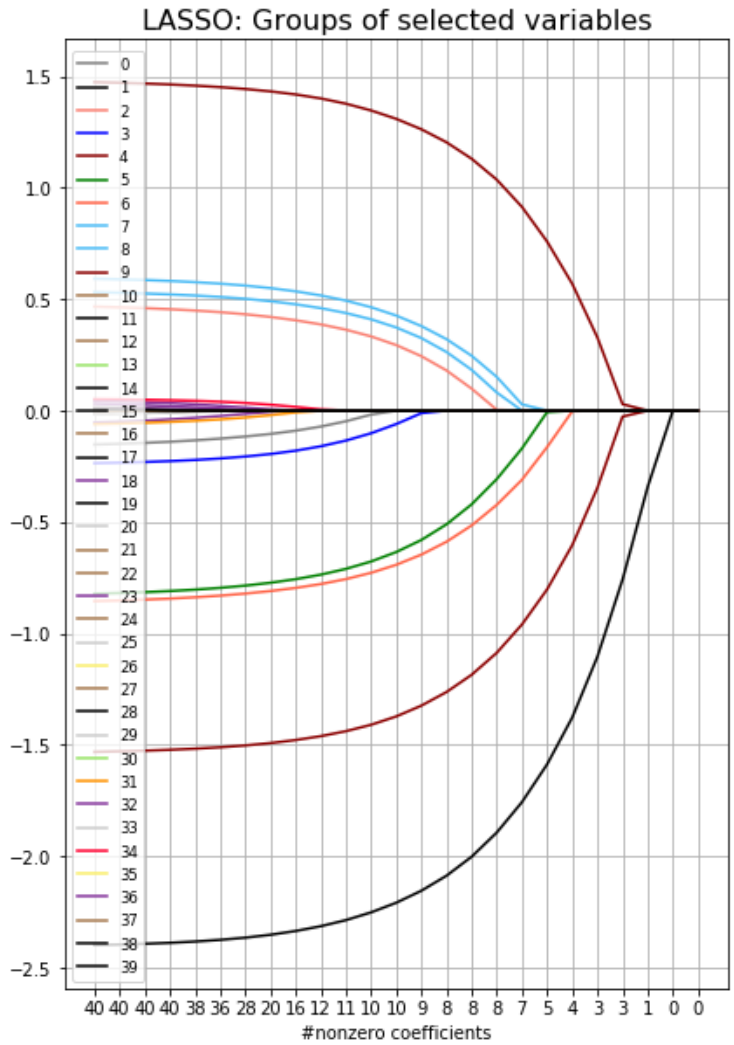
In [9]:

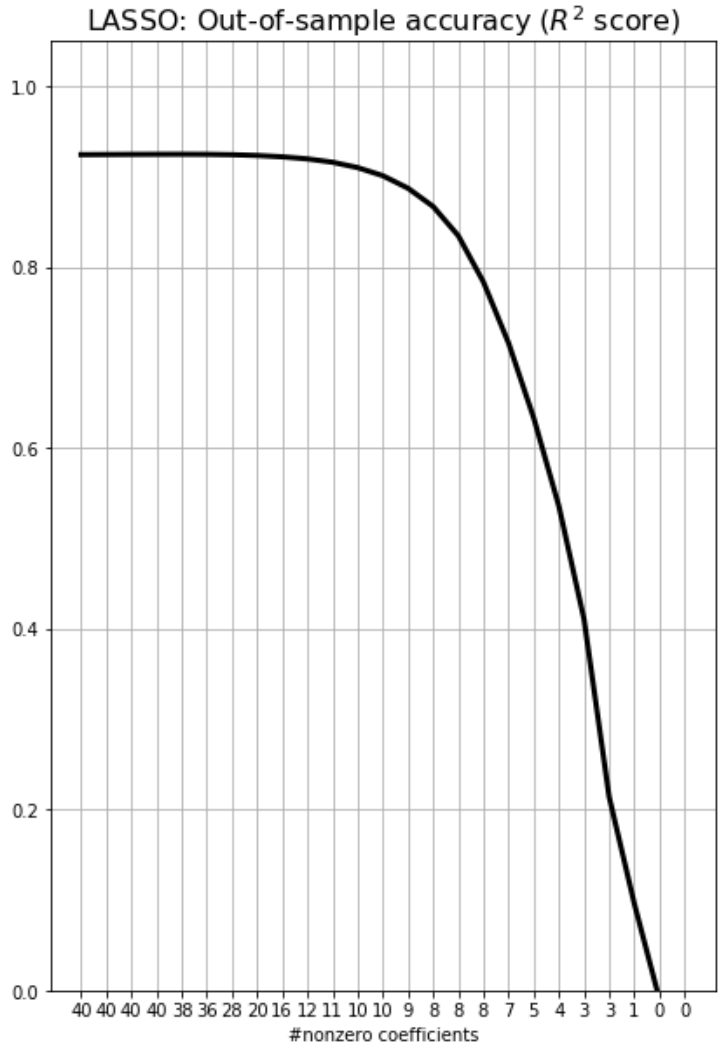
```
comment = "dataset: 10/40 variables relevant, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)
y = (true_coefs * X).sum(axis=1) + epsilon

C_grid = np.logspace(-2, 0.5, 25)
run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

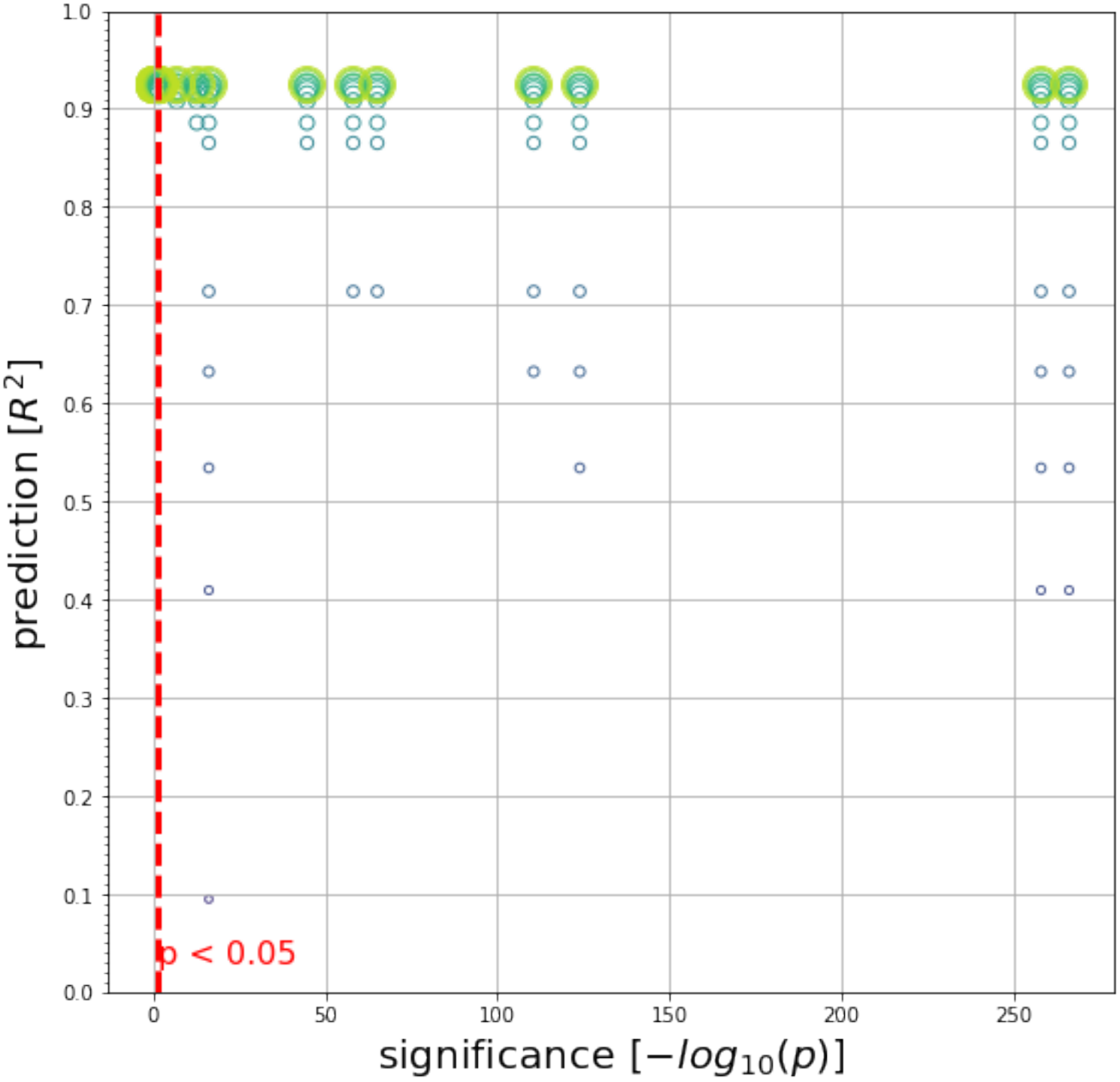
```
alpha: 0.0100 acc: 0.93 active_coefs: 40
alpha: 0.0127 acc: 0.93 active_coefs: 40
alpha: 0.0162 acc: 0.94 active_coefs: 40
alpha: 0.0205 acc: 0.94 active_coefs: 40
alpha: 0.0261 acc: 0.94 active_coefs: 38
alpha: 0.0332 acc: 0.94 active_coefs: 36
alpha: 0.0422 acc: 0.94 active_coefs: 28
alpha: 0.0536 acc: 0.94 active_coefs: 20
alpha: 0.0681 acc: 0.93 active_coefs: 16
alpha: 0.0866 acc: 0.93 active_coefs: 12
alpha: 0.1101 acc: 0.93 active_coefs: 11
alpha: 0.1399 acc: 0.92 active_coefs: 10
alpha: 0.1778 acc: 0.91 active_coefs: 10
alpha: 0.2260 acc: 0.90 active_coefs: 9
alpha: 0.2873 acc: 0.88 active_coefs: 8
alpha: 0.3652 acc: 0.86 active_coefs: 8
alpha: 0.4642 acc: 0.81 active_coefs: 8
alpha: 0.5900 acc: 0.75 active_coefs: 7
alpha: 0.7499 acc: 0.67 active_coefs: 5
alpha: 0.9532 acc: 0.56 active_coefs: 4
alpha: 1.2115 acc: 0.41 active_coefs: 3
alpha: 1.5399 acc: 0.19 active_coefs: 3
alpha: 1.9573 acc: 0.09 active_coefs: 1
alpha: 2.4879 acc: -0.00 active_coefs: 0
alpha: 3.1623 acc: -0.00 active_coefs: 0
40
266.203836749
257.580615163
123.564764097
110.581643142
64.6911966587
58.0763677229
44.5382808274
15.6535597745
12.4367201993
6.63051580291
```

```
1.6133005526
1.29300187988
1.14848609089
1.06006598496
0.817489578238
0.642098922498
0.493990500501
0.453427787712
0.422264239499
0.382988313694
0.371271160633
0.365409504085
0.349212515172
0.322311370704
0.309894806882
0.303032990969
0.301528778114
0.272578017003
0.270285843655
0.251741956379
0.247745297292
0.19677190609
0.186386617028
0.144977583921
0.0605753890562
0.0594680417561
0.0292018341059
0.0286690261293
0.0196918758999
0.000825533770864
skipping 3
skipping 4
skipping 5
skipping 12
skipping 15
skipping 16
skipping 21
skipping 24
```


Linear regression
true betas
estimated betas
significant at p<0.05
input variables


LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients



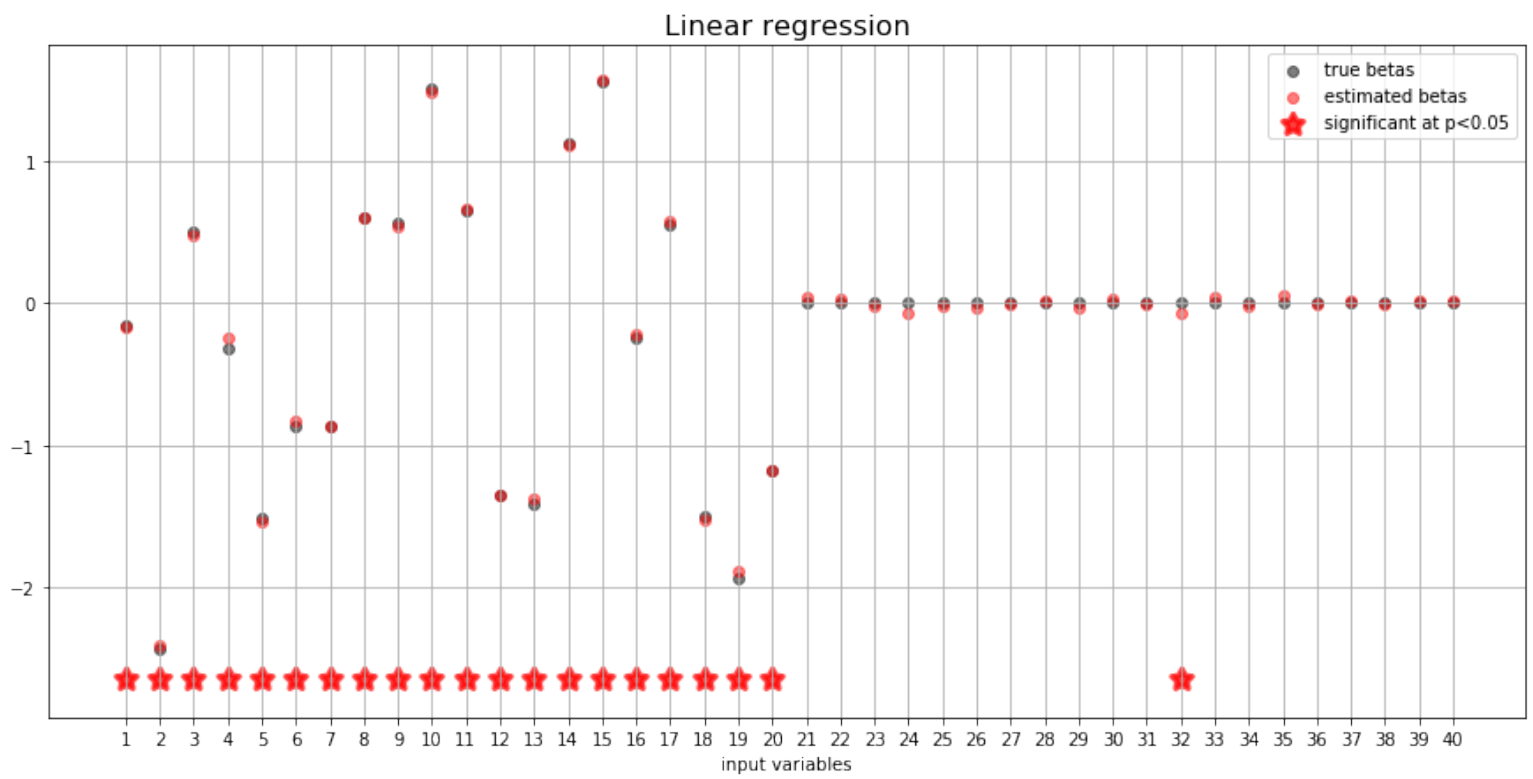
In [10]:

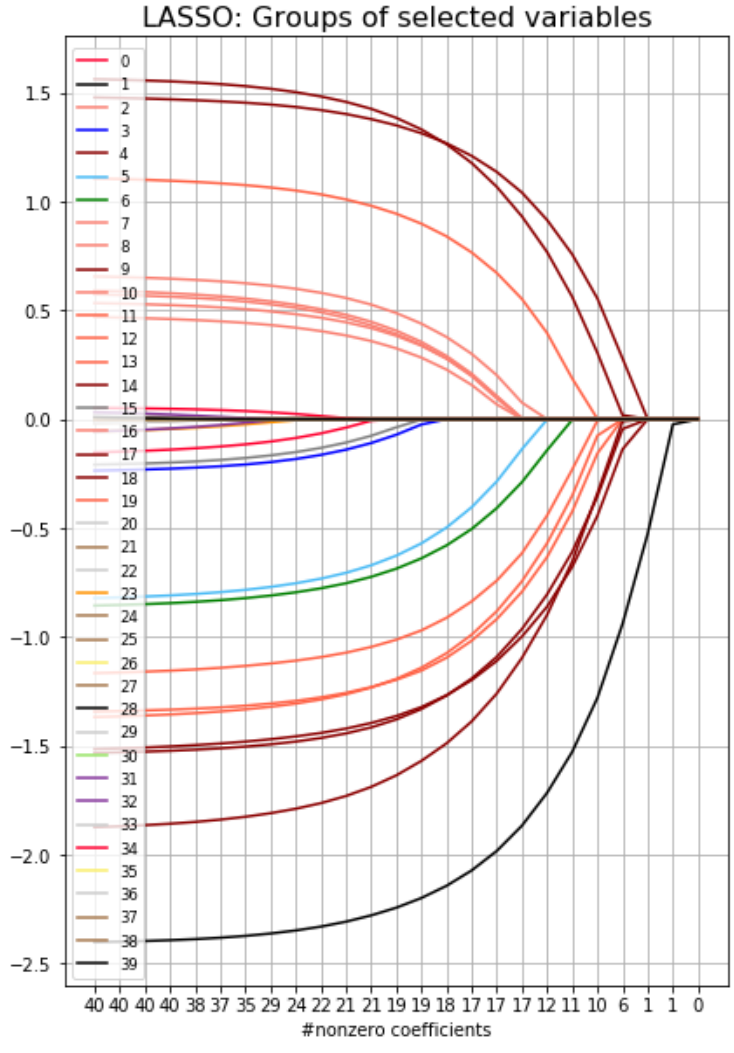
```
comment = "dataset: 20/40 variables relevant, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samplebs = 1000
n_feat = 40
n_feat_relevant = 20
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)
y = (true_coefs * X).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 0.5, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

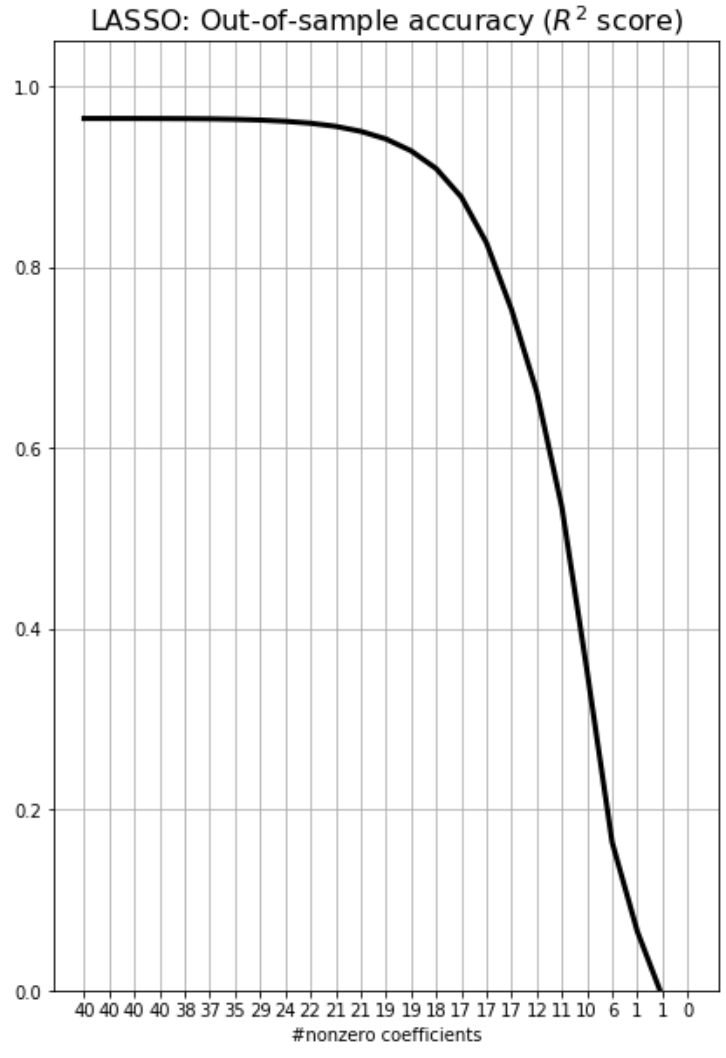
```
alpha: 0.0100 acc: 0.96 active_coefs: 40
alpha: 0.0127 acc: 0.96 active_coefs: 40
alpha: 0.0162 acc: 0.97 active_coefs: 40
```

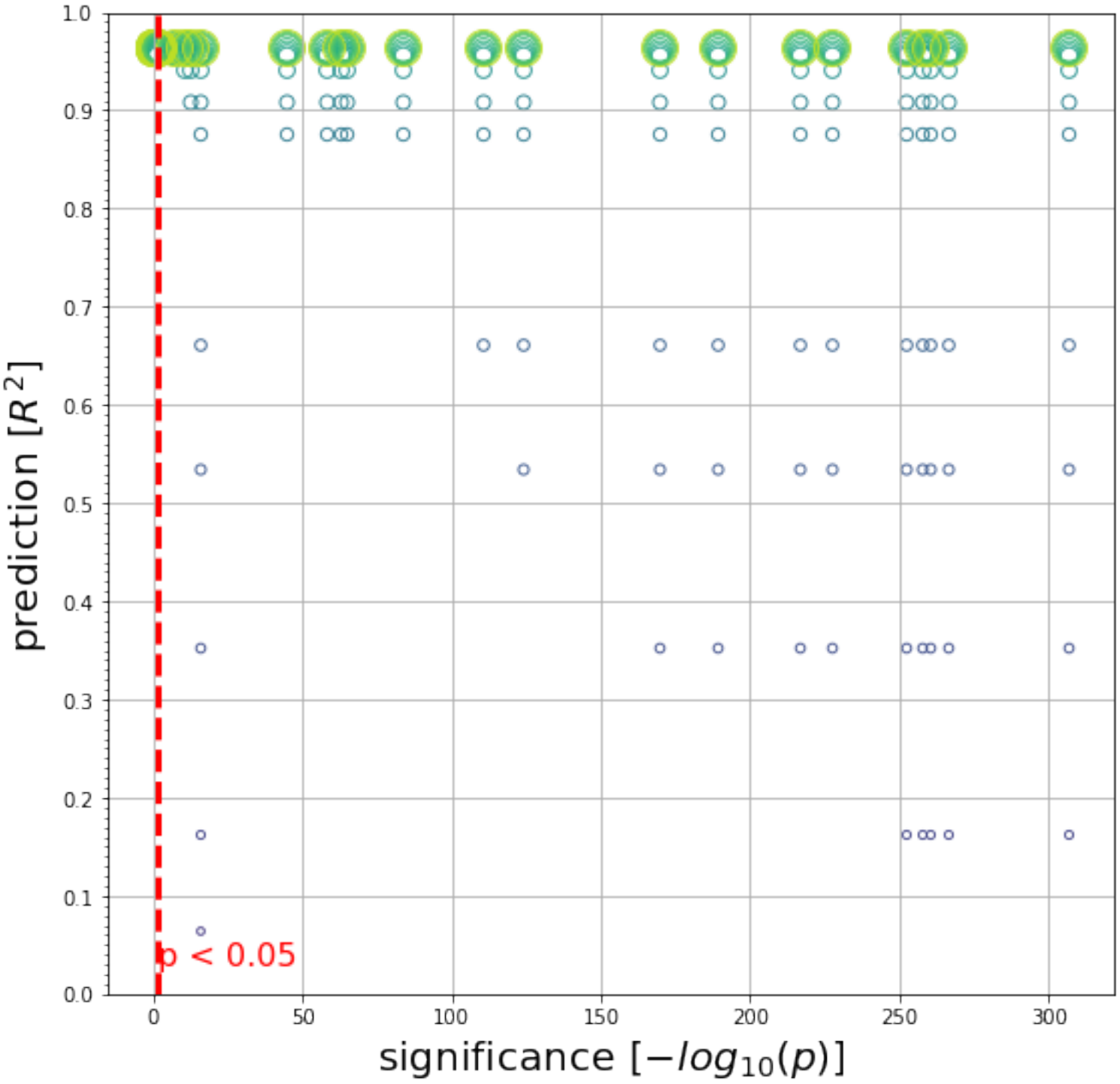
```
alpha: 0.0205 acc: 0.97 active_coefs: 40
alpha: 0.0261 acc: 0.97 active_coefs: 38
alpha: 0.0332 acc: 0.97 active_coefs: 37
alpha: 0.0422 acc: 0.97 active_coefs: 35
alpha: 0.0536 acc: 0.97 active_coefs: 29
alpha: 0.0681 acc: 0.97 active_coefs: 24
alpha: 0.0866 acc: 0.96 active_coefs: 22
alpha: 0.1101 acc: 0.96 active_coefs: 21
alpha: 0.1399 acc: 0.96 active_coefs: 21
alpha: 0.1778 acc: 0.95 active_coefs: 19
alpha: 0.2260 acc: 0.94 active_coefs: 19
alpha: 0.2873 acc: 0.93 active_coefs: 18
alpha: 0.3652 acc: 0.90 active_coefs: 17
alpha: 0.4642 acc: 0.86 active_coefs: 17
alpha: 0.5900 acc: 0.81 active_coefs: 17
alpha: 0.7499 acc: 0.73 active_coefs: 12
alpha: 0.9532 acc: 0.61 active_coefs: 11
alpha: 1.2115 acc: 0.43 active_coefs: 10
alpha: 1.5399 acc: 0.19 active_coefs: 6
alpha: 1.9573 acc: 0.08 active_coefs: 1
alpha: 2.4879 acc: -0.00 active_coefs: 1
alpha: 3.1623 acc: -0.00 active_coefs: 0
40
306.70250279
266.203836749
259.985298685
257.580615163
251.750476271
227.390605979
216.542219335
189.032138743
169.767726896
123.564764097
110.581643142
83.4078299178
64.6911966587
62.3908395227
58.0763677229
44.5382808274
15.6535597745
12.4367201993
10.5126810224
6.63051580291
1.6133005526
1.29300187988
1.14848609089
0.817489578238
0.642098922498
0.453427787712
0.422264239499
0.382988313694
0.371271160633
0.365409504085
```

```
0.349212515172
0.309894806882
0.301528778114
0.272578017003
0.251741956379
0.19677190609
0.144977583921
0.0594680417562
0.0292018341059
0.0286690261292
skipping 1
skipping 2
skipping 3
skipping 11
skipping 13
skipping 16
skipping 17
skipping 23
```




LASSO: Groups of selected variables
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
40 40 40 40 38 37 35 29 24 22 21 21 19 19 18 17 17 17 12 11 10 6 1 1 0
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
1.0
0.8
0.6
0.4
0.2
0.0
40 40 40 40 38 37 35 29 24 22 21 21 19 19 18 17 17 17 12 11 10 6 1 1 0
#nonzero coefficients



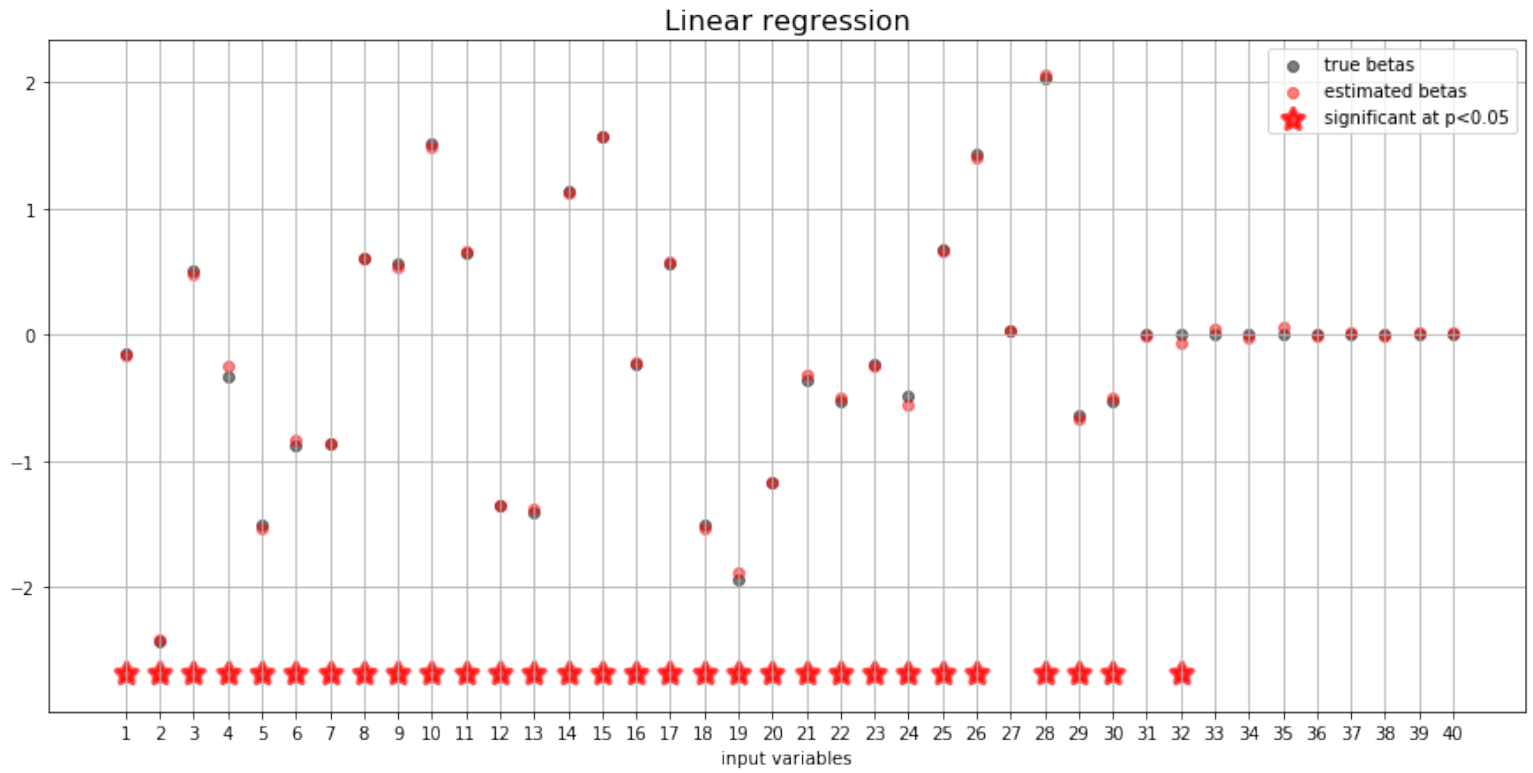
In [11]:

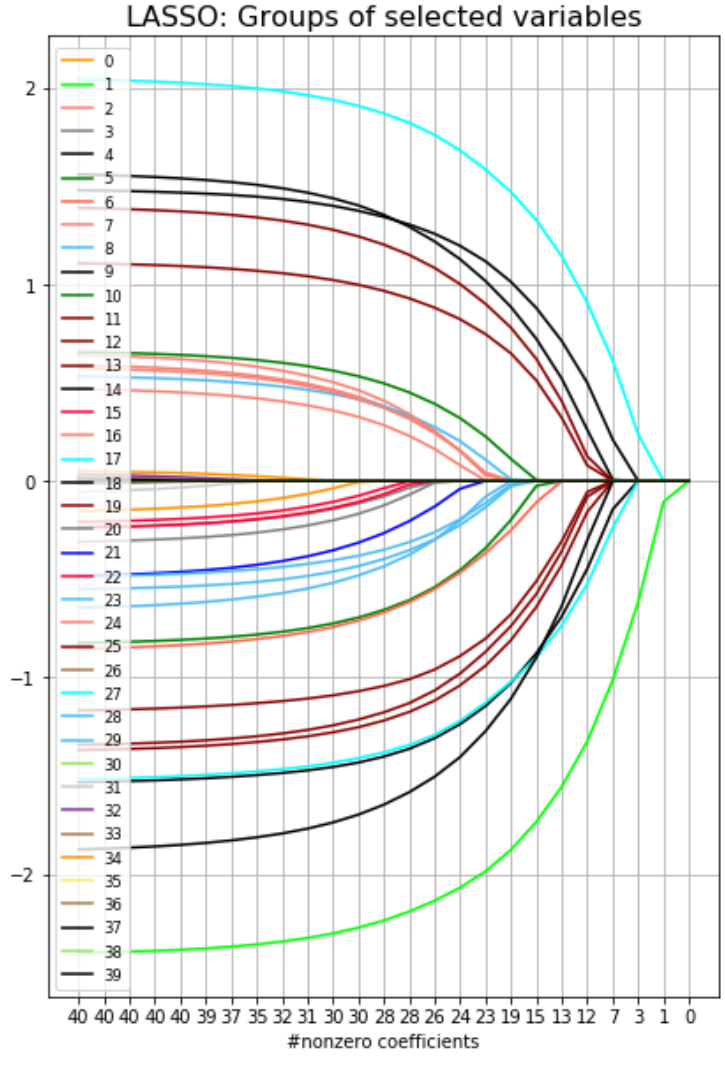
```
comment = "dataset: 30/40 variables relevant, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 30
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)
y = (true_coefs * X).sum(axis=1) + epsilon

C_grid = np.logspace(-2, 0.5, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

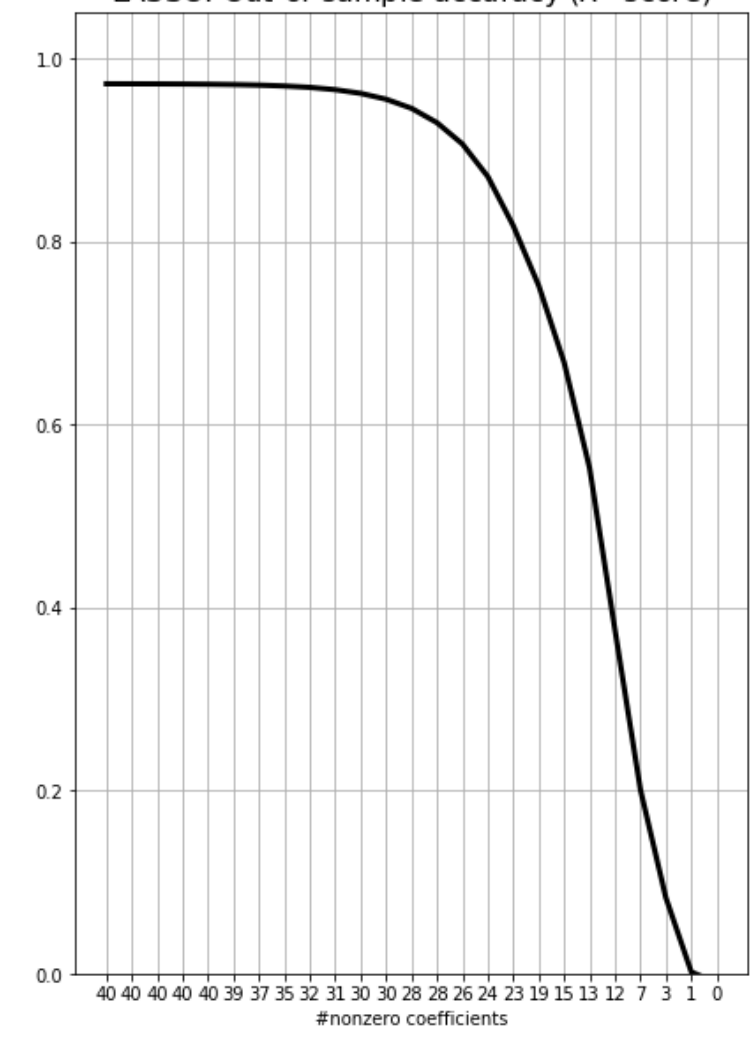
```
alpha: 0.0100 acc: 0.97 active_coefs: 40
alpha: 0.0127 acc: 0.97 active_coefs: 40
```

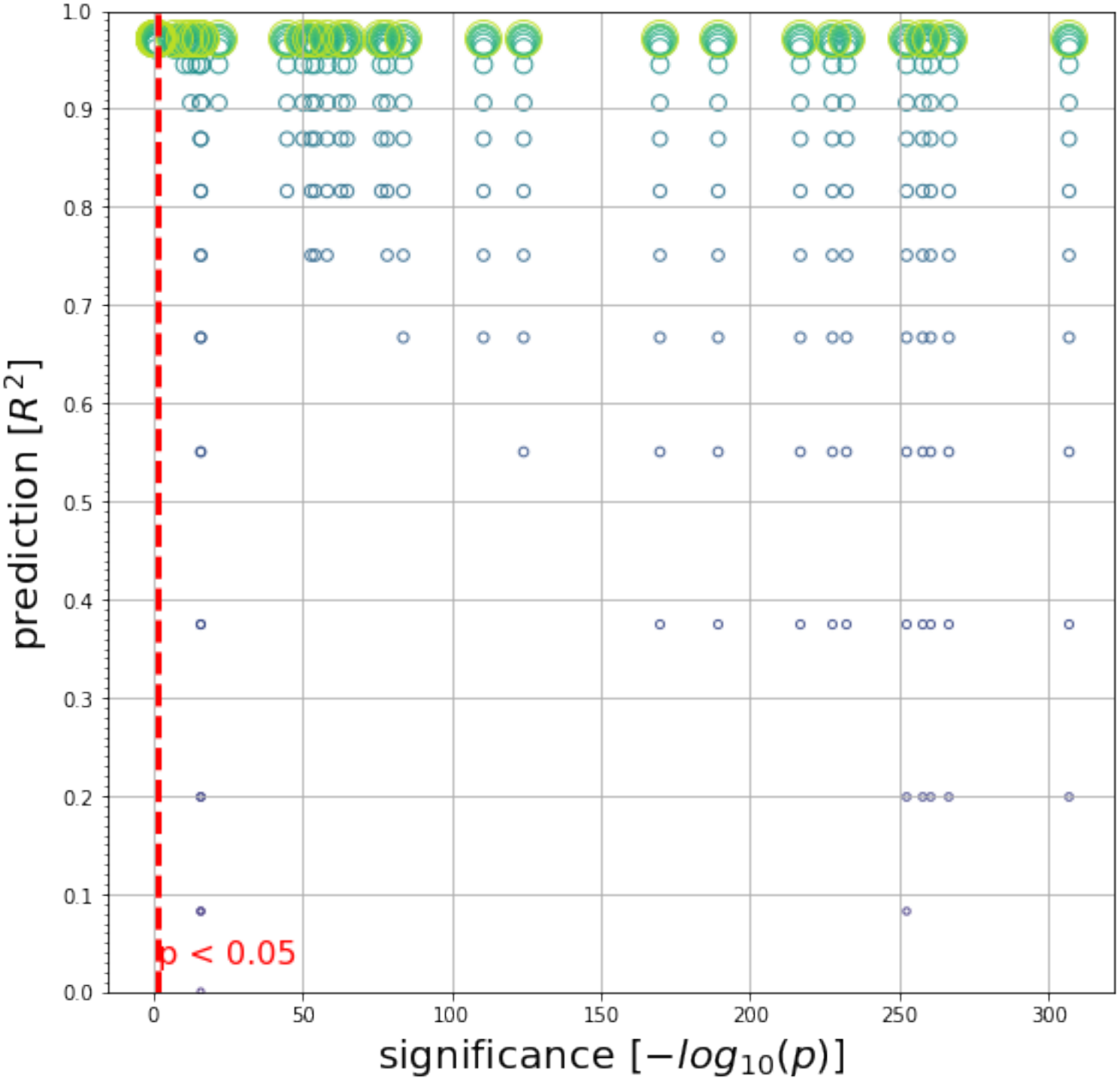
```
alpha: 0.0162 acc: 0.97 active_coefs: 40
alpha: 0.0205 acc: 0.97 active_coefs: 40
alpha: 0.0261 acc: 0.97 active_coefs: 40
alpha: 0.0332 acc: 0.97 active_coefs: 39
alpha: 0.0422 acc: 0.97 active_coefs: 37
alpha: 0.0536 acc: 0.97 active_coefs: 35
alpha: 0.0681 acc: 0.97 active_coefs: 32
alpha: 0.0866 acc: 0.97 active_coefs: 31
alpha: 0.1101 acc: 0.97 active_coefs: 30
alpha: 0.1399 acc: 0.96 active_coefs: 30
alpha: 0.1778 acc: 0.95 active_coefs: 28
alpha: 0.2260 acc: 0.94 active_coefs: 28
alpha: 0.2873 acc: 0.92 active_coefs: 26
alpha: 0.3652 acc: 0.90 active_coefs: 24
alpha: 0.4642 acc: 0.85 active_coefs: 23
alpha: 0.5900 acc: 0.80 active_coefs: 19
alpha: 0.7499 acc: 0.72 active_coefs: 15
alpha: 0.9532 acc: 0.62 active_coefs: 13
alpha: 1.2115 acc: 0.44 active_coefs: 12
alpha: 1.5399 acc: 0.23 active_coefs: 7
alpha: 1.9573 acc: 0.10 active_coefs: 3
alpha: 2.4879 acc: 0.01 active_coefs: 1
alpha: 3.1623 acc: -0.00 active_coefs: 0
40
306.70250279
266.203836749
259.985298685
257.580615163
251.750476271
232.177575665
227.390605979
216.542219335
189.032138743
169.767726896
123.564764097
110.581643142
83.4078299178
78.3958987363
76.2136424952
64.6911966587
62.3908395227
58.0763677229
53.6639002678
52.6305743525
49.951198593
44.5382808274
21.9049462821
15.6535597745
15.6535597745
14.4524983925
12.4367201993
10.5126810224
6.63051580291
```

```
1.6133005526
1.14848609089
0.817489578238
0.542505420164
0.365409504085
0.301528778114
0.251741956379
0.19677190609
0.144977583921
0.0594680417562
0.0292018341059
skipping 1
skipping 2
skipping 3
skipping 4
skipping 11
skipping 13
```




LASSO: Groups of selected variables
2
1
0
−1
−2
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
40 40 40 40 40 39 37 35 32 31 30 30 28 28 26 24 23 19 15 13 12 7 3 1 0
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
1.0
0.8
0.6
0.4
0.2
0.0
40 40 40 40 40 39 37 35 32 31 30 30 28 28 26 24 23 19 15 13 12 7 3 1 0
#nonzero coefficients



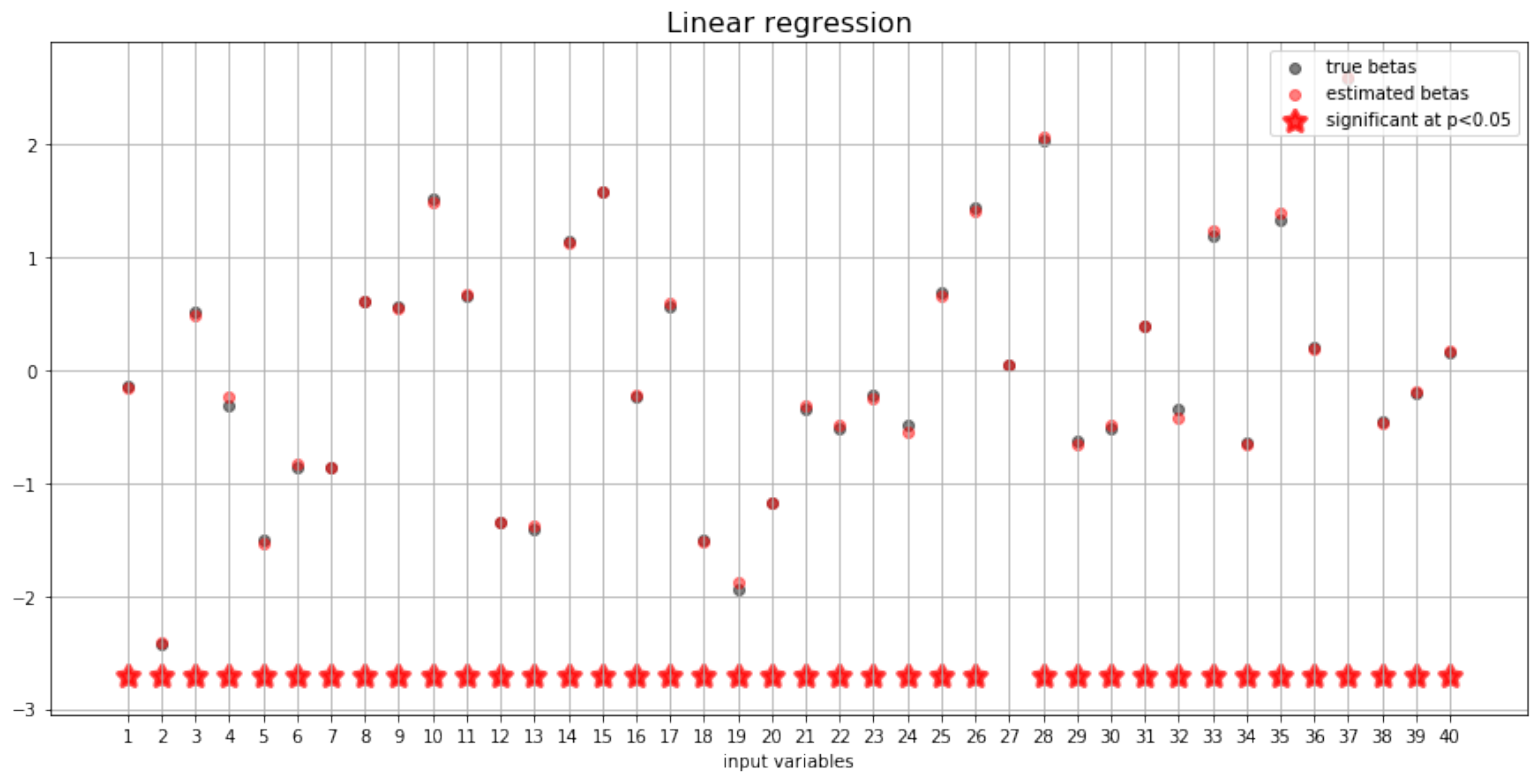
In [12]:

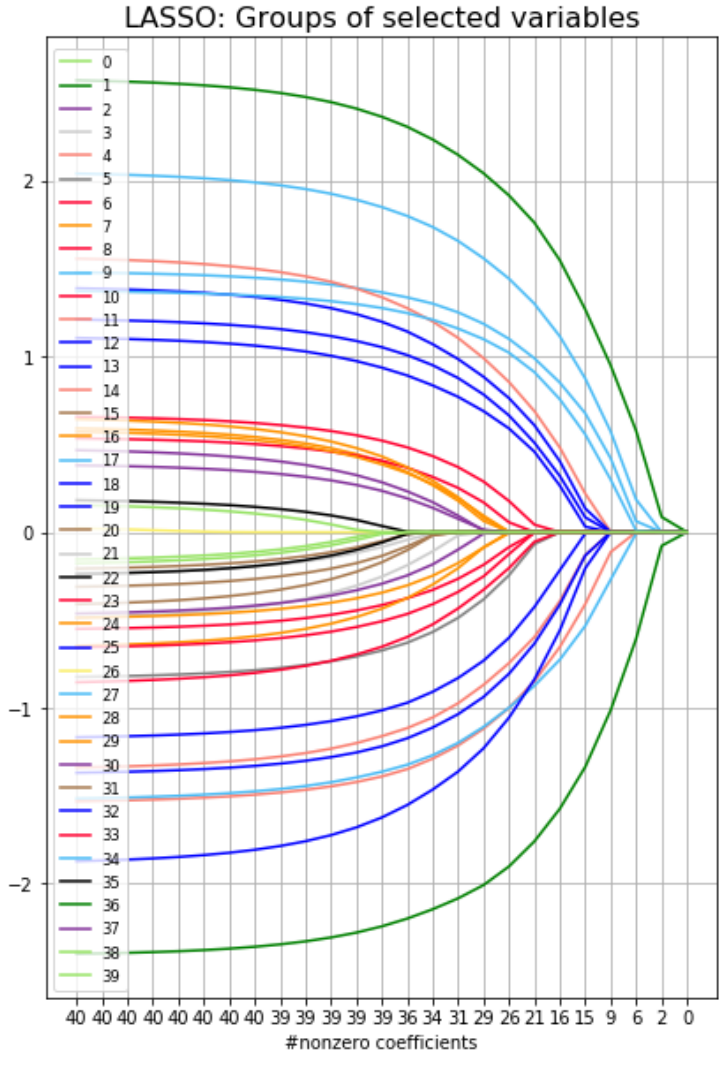
```
comment = "dataset: 40/40 variables relevant, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 40
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)
y = (true_coefs * X).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 0.5, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

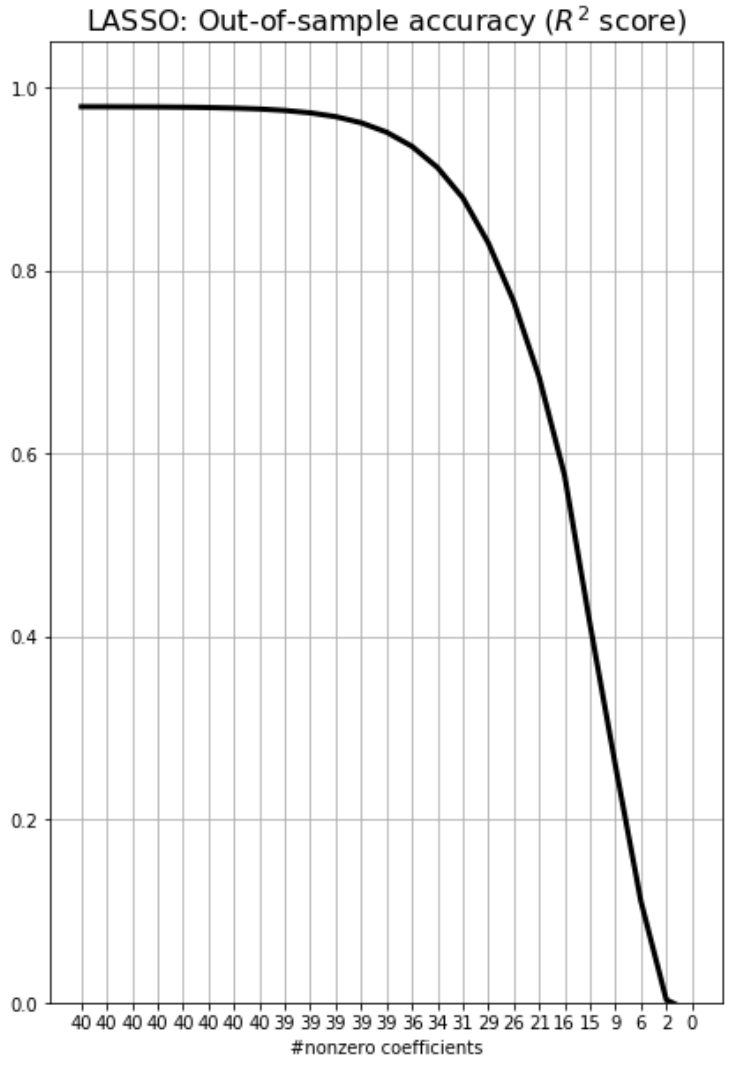
```
alpha: 0.0100 acc: 0.98 active_coefs: 40
alpha: 0.0127 acc: 0.98 active_coefs: 40
alpha: 0.0162 acc: 0.98 active_coefs: 40
```

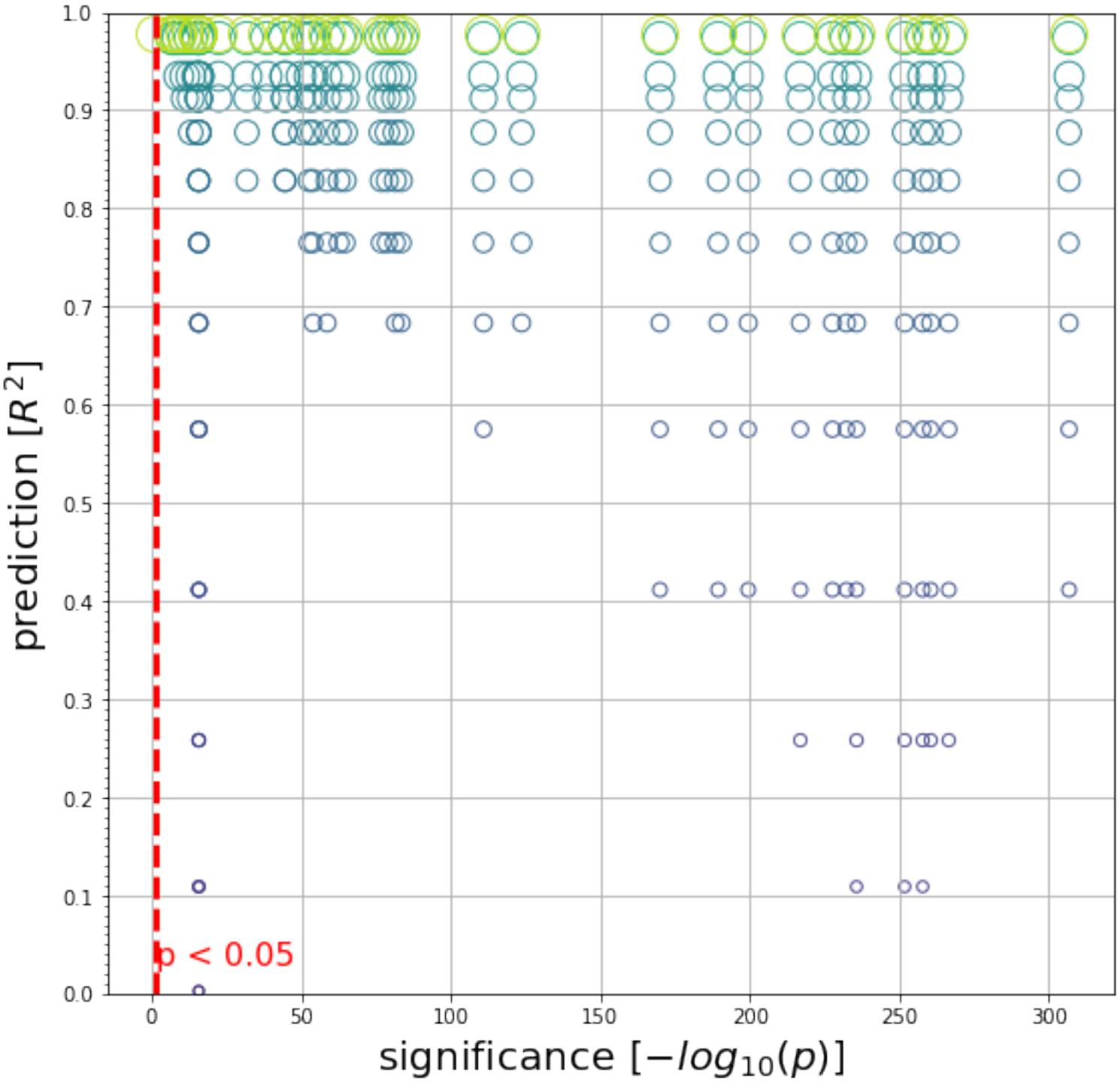
```
alpha: 0.0205 acc: 0.98 active_coefs: 40
alpha: 0.0261 acc: 0.98 active_coefs: 40
alpha: 0.0332 acc: 0.98 active_coefs: 40
alpha: 0.0422 acc: 0.98 active_coefs: 40
alpha: 0.0536 acc: 0.98 active_coefs: 40
alpha: 0.0681 acc: 0.98 active_coefs: 39
alpha: 0.0866 acc: 0.97 active_coefs: 39
alpha: 0.1101 acc: 0.97 active_coefs: 39
alpha: 0.1399 acc: 0.96 active_coefs: 39
alpha: 0.1778 acc: 0.95 active_coefs: 39
alpha: 0.2260 acc: 0.94 active_coefs: 36
alpha: 0.2873 acc: 0.92 active_coefs: 34
alpha: 0.3652 acc: 0.89 active_coefs: 31
alpha: 0.4642 acc: 0.85 active_coefs: 29
alpha: 0.5900 acc: 0.79 active_coefs: 26
alpha: 0.7499 acc: 0.71 active_coefs: 21
alpha: 0.9532 acc: 0.61 active_coefs: 16
alpha: 1.2115 acc: 0.45 active_coefs: 15
alpha: 1.5399 acc: 0.28 active_coefs: 9
alpha: 1.9573 acc: 0.12 active_coefs: 6
alpha: 2.4879 acc: 0.01 active_coefs: 2
alpha: 3.1623 acc: -0.00 active_coefs: 0
40
306.70250279
266.203836749
259.985298685
257.580615163
251.750476271
235.534081448
232.177575665
227.390605979
216.542219335
199.38808419
189.032138743
169.767726896
123.564764097
110.581643142
83.4078299178
81.1335521336
78.3958987363
76.2136424952
64.6911966587
62.3908395227
58.0763677229
53.6639002678
52.6305743525
49.951198593
44.5382808274
44.0642093481
38.0245732995
31.6565063265
21.9049462821
15.6535597745
```

```
15.6535597745
15.6535597745
14.4524983925
12.4367201993
10.5126810224
8.53876677309
7.98828655843
6.89162459463
6.63051580291
0.542505420164
skipping 1
skipping 2
skipping 3
skipping 4
skipping 5
skipping 6
skipping 7
skipping 9
skipping 10
skipping 11
skipping 12
```




LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients



100 samples, 40 variables, error = none

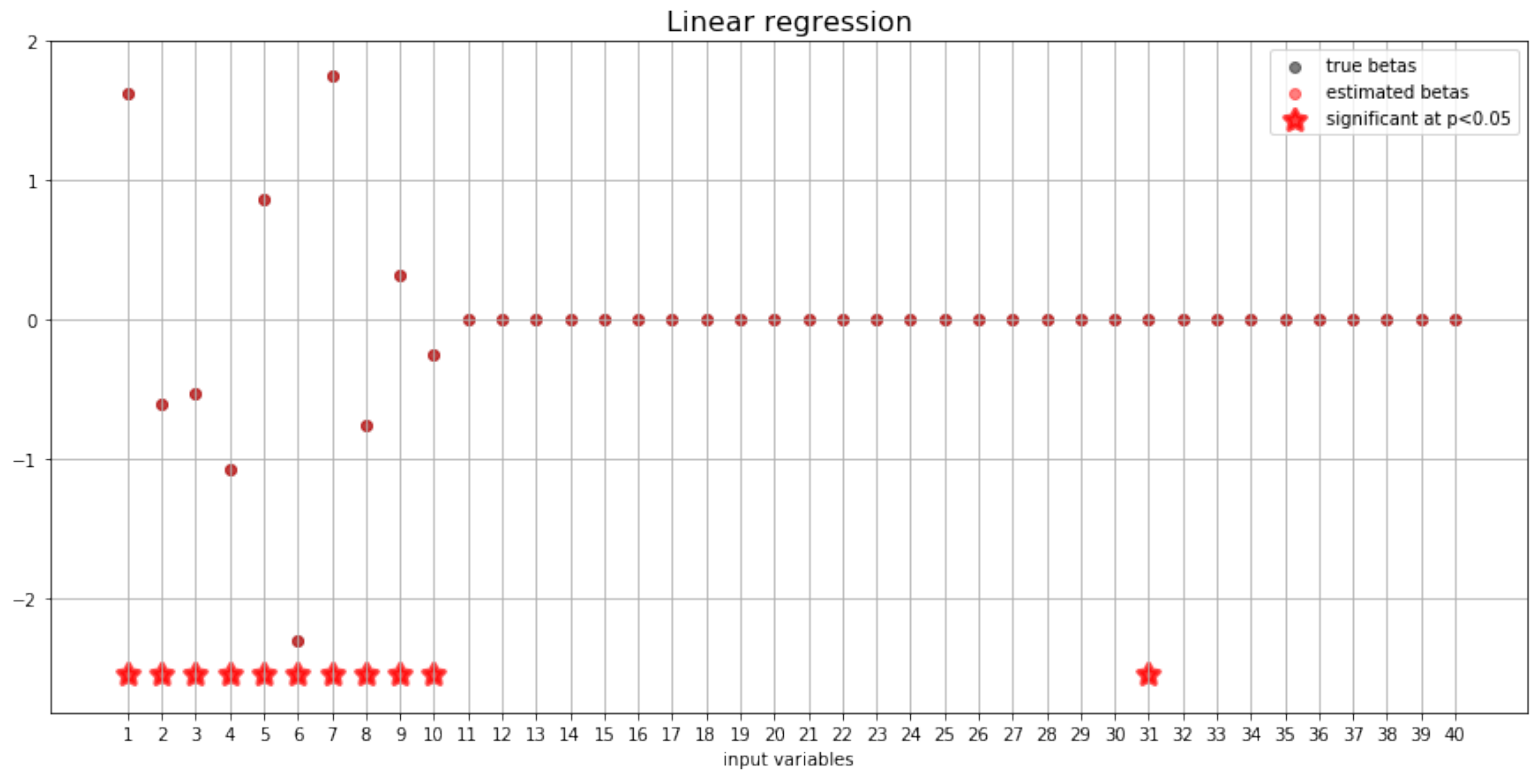
In [13]:

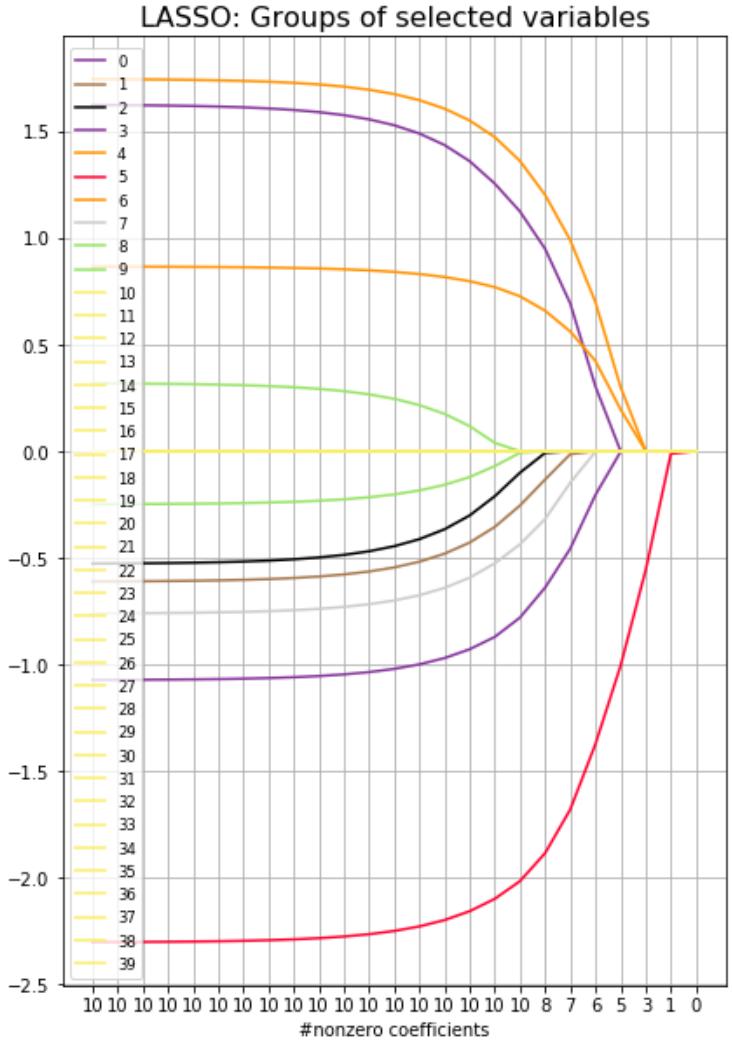
```
comment = "dataset: 10/40 variables relevant, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 100
n_feat = 40
n_feat_relevant = 10
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)
y = (true_coefs * X).sum(axis=1)
C_grid = np.logspace(-3, 0.5, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

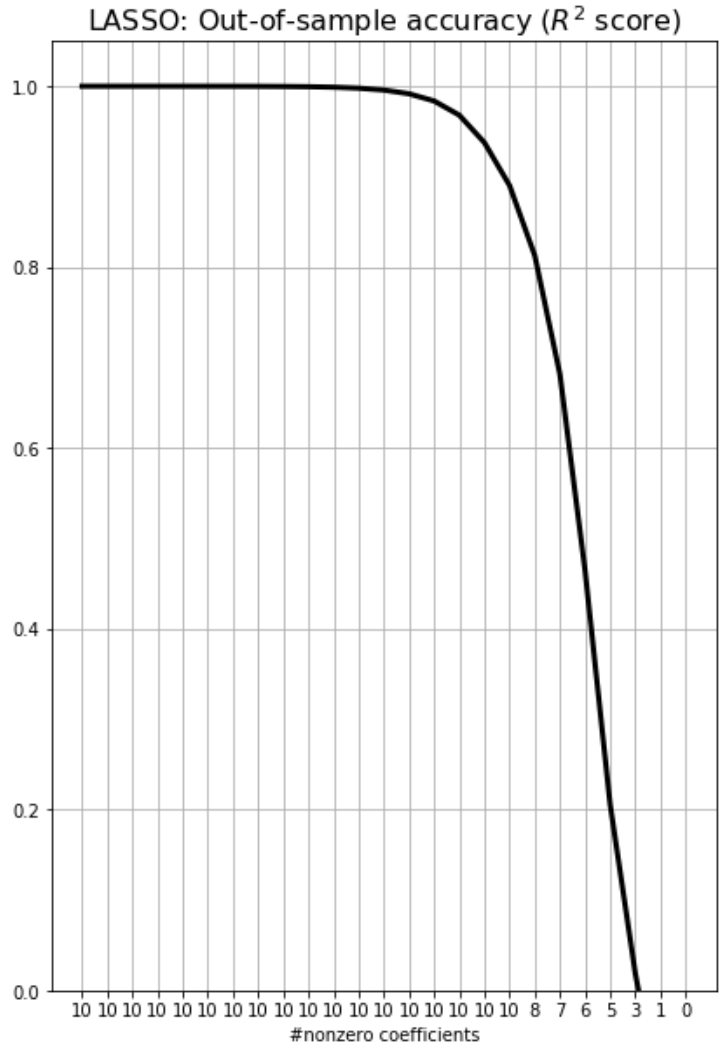
alpha: 0.0010 acc: 1.00 active coefs: 10

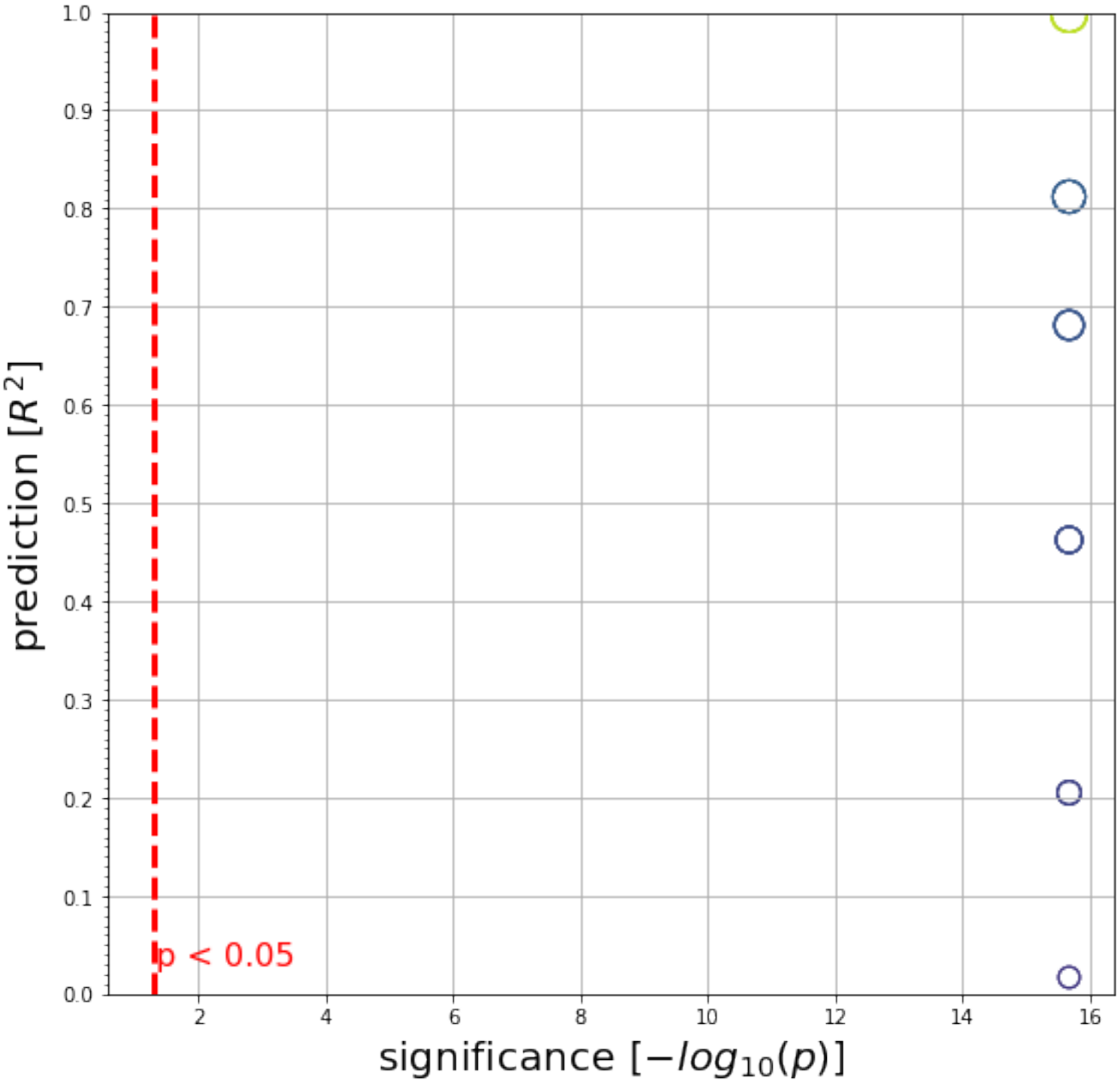
alpha: 0.0010 acc: 1.00 active_coefs: 10
alpha: 0.0014 acc: 1.00 active_coefs: 10
alpha: 0.0020 acc: 1.00 active_coefs: 10
alpha: 0.0027 acc: 1.00 active_coefs: 10
alpha: 0.0038 acc: 1.00 active_coefs: 10
alpha: 0.0054 acc: 1.00 active_coefs: 10
alpha: 0.0075 acc: 1.00 active_coefs: 10
alpha: 0.0105 acc: 1.00 active_coefs: 10
alpha: 0.0147 acc: 1.00 active_coefs: 10
alpha: 0.0205 acc: 1.00 active_coefs: 10
alpha: 0.0287 acc: 1.00 active_coefs: 10
alpha: 0.0402 acc: 1.00 active_coefs: 10
alpha: 0.0562 acc: 1.00 active_coefs: 10
alpha: 0.0787 acc: 0.99 active_coefs: 10
alpha: 0.1101 acc: 0.99 active_coefs: 10
alpha: 0.1540 acc: 0.98 active_coefs: 10
alpha: 0.2154 acc: 0.96 active_coefs: 10
alpha: 0.3014 acc: 0.93 active_coefs: 10
alpha: 0.4217 acc: 0.89 active_coefs: 8
alpha: 0.5900 acc: 0.80 active_coefs: 7
alpha: 0.8254 acc: 0.68 active_coefs: 6
alpha: 1.1548 acc: 0.47 active_coefs: 5
alpha: 1.6156 acc: 0.18 active_coefs: 3
alpha: 2.2603 acc: -0.05 active_coefs: 1
alpha: 3.1623 acc: -0.05 active_coefs: 0
40
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
1.61739026323
0.879563955841
0.790306054233
0.778937583676
0.725370757479
0.635983280364
0.557566256501
0.489958458313
0.406985390651
0.375985990843
0.375406530898
0.359171519063
0.29825310897
0.29289624102
0.279531275712
0.276687594927
0.275025663466
0.26829732566

```
0.243293378855
0.173644697436
0.129666046034
0.111371440451
0.110064425583
0.0834983032796
0.0584021802216
0.0480890142989
0.0434083446419
0.0309774591073
0.0270615649803
0.022174180266
skipping 1
skipping 2
skipping 3
skipping 4
skipping 5
skipping 6
skipping 7
skipping 8
skipping 9
skipping 10
skipping 11
skipping 12
skipping 13
skipping 14
skipping 15
skipping 16
skipping 17
```




LASSO: Groups of selected variables
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 8 7 6 5 3 1 0
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
1.0
0.8
0.6
0.4
0.2
0.0
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 8 7 6 5 3 1 0
#nonzero coefficients



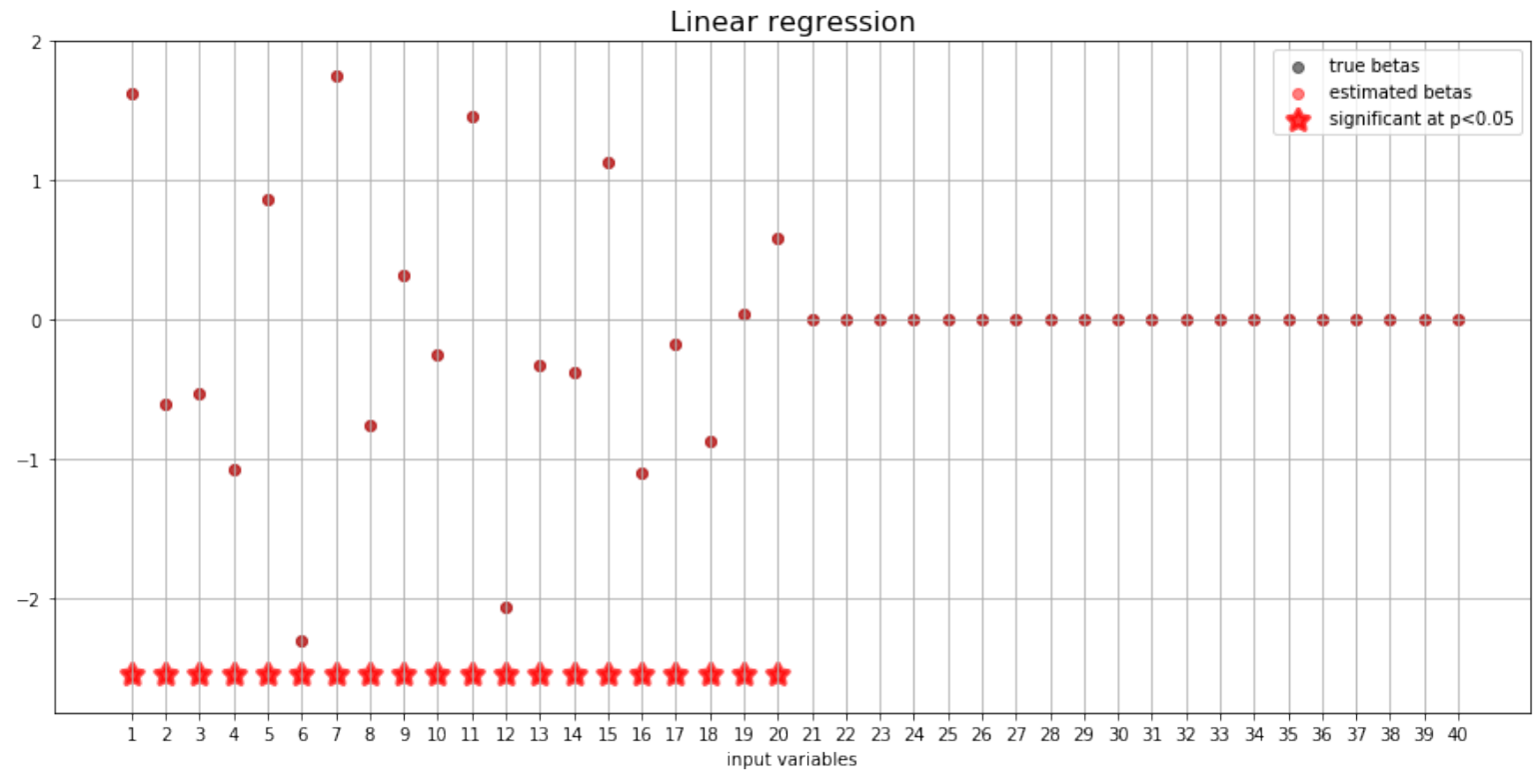
In [14]:

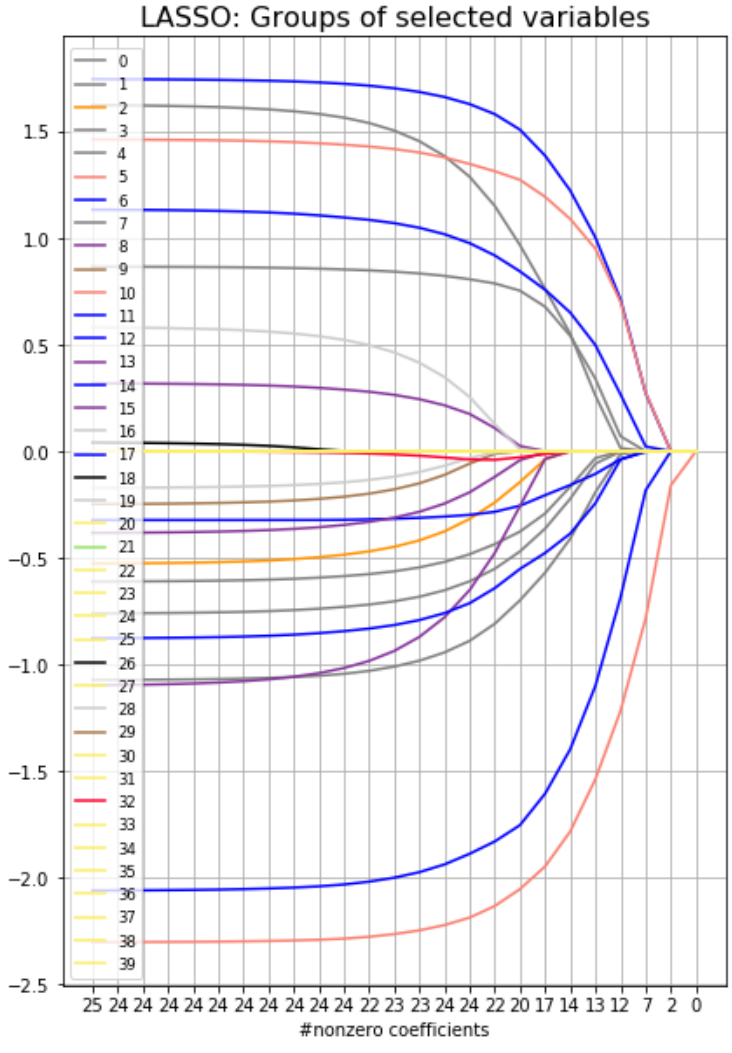
```
comment = "dataset: 20/40 variables relevant, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 100
n_feat = 40
n_feat_relevant = 20
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)
y = (true_coefs * X).sum(axis=1)
C_grid = np.logspace(-3, 0.5, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

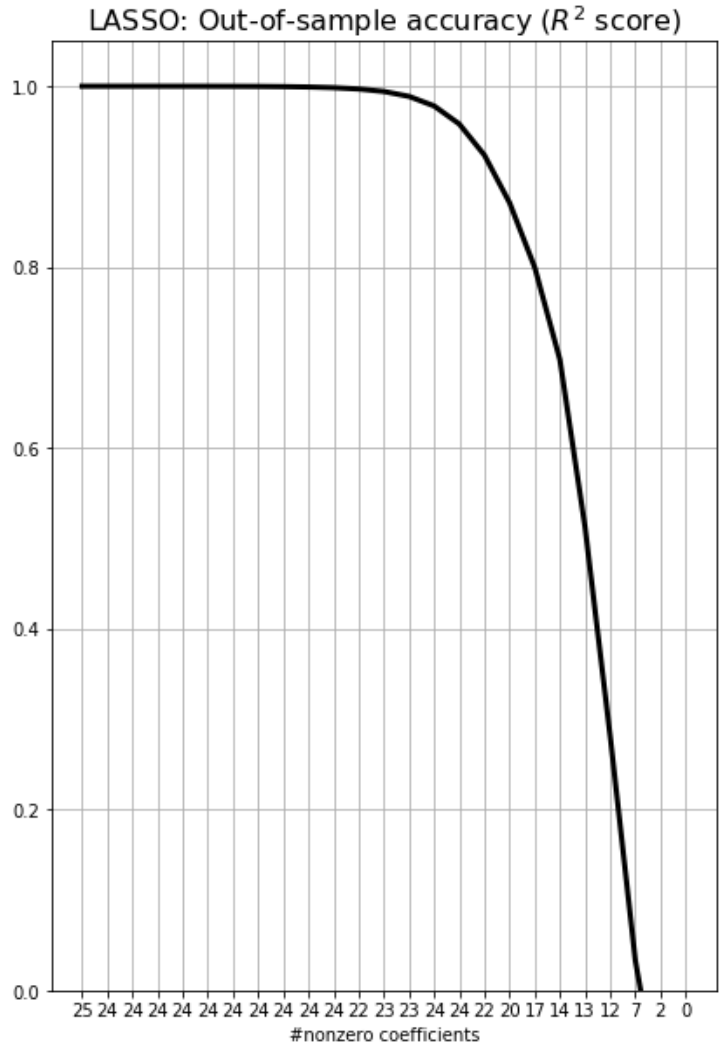
```
alpha: 0.0010 acc: 1.00 active_coefs: 25
alpha: 0.0014 acc: 1.00 active_coefs: 24
alpha: 0.0020 acc: 1.00 active_coefs: 24
alpha: 0.0027 acc: 1.00 active_coefs: 24
```

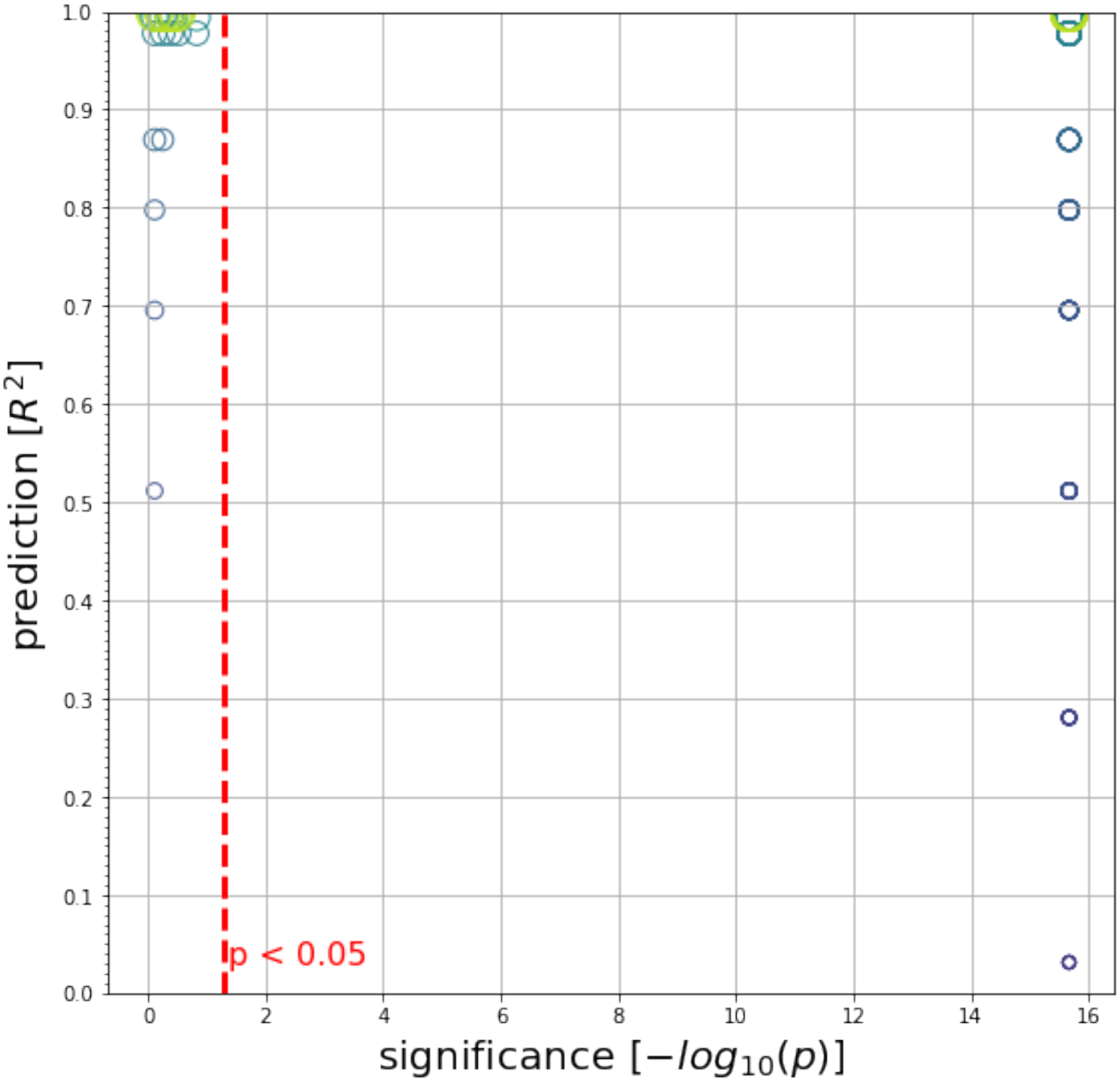
```
alpha: 0.0038 acc: 1.00 active_coefs: 24
alpha: 0.0054 acc: 1.00 active_coefs: 24
alpha: 0.0075 acc: 1.00 active_coefs: 24
alpha: 0.0105 acc: 1.00 active_coefs: 24
alpha: 0.0147 acc: 1.00 active_coefs: 24
alpha: 0.0205 acc: 1.00 active_coefs: 24
alpha: 0.0287 acc: 1.00 active_coefs: 24
alpha: 0.0402 acc: 1.00 active_coefs: 22
alpha: 0.0562 acc: 1.00 active_coefs: 23
alpha: 0.0787 acc: 0.99 active_coefs: 23
alpha: 0.1101 acc: 0.98 active_coefs: 24
alpha: 0.1540 acc: 0.97 active_coefs: 24
alpha: 0.2154 acc: 0.94 active_coefs: 22
alpha: 0.3014 acc: 0.90 active_coefs: 20
alpha: 0.4217 acc: 0.83 active_coefs: 17
alpha: 0.5900 acc: 0.69 active_coefs: 14
alpha: 0.8254 acc: 0.39 active_coefs: 13
alpha: 1.1548 acc: 0.00 active_coefs: 12
alpha: 1.6156 acc: -0.69 active_coefs: 7
alpha: 2.2603 acc: -1.09 active_coefs: 2
alpha: 3.1623 acc: -1.15 active_coefs: 0
40
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
0.886567580113
0.81864295227
0.734424735799
0.590584025429
0.51475776808
0.482841439493
0.474298849976
0.37474412589
0.344564335639
0.300080499518
0.220048996732
```

```
0.218906262707
0.19226119604
0.165193496298
0.112394696422
0.086732755483
0.0843127739337
0.0801228464764
0.0722840806337
0.0529873237368
skipping 2
skipping 3
skipping 4
skipping 5
skipping 6
skipping 7
skipping 8
skipping 9
skipping 10
skipping 13
skipping 15
skipping 16
```




LASSO: Groups of selected variables
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
25 24 24 24 24 24 24 24 24 24 24 22 23 23 24 24 22 20 17 14 13 12 7 2 0
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
1.0
0.8
0.6
0.4
0.2
0.0
25 24 24 24 24 24 24 24 24 24 24 22 23 23 24 24 22 20 17 14 13 12 7 2 0
#nonzero coefficients



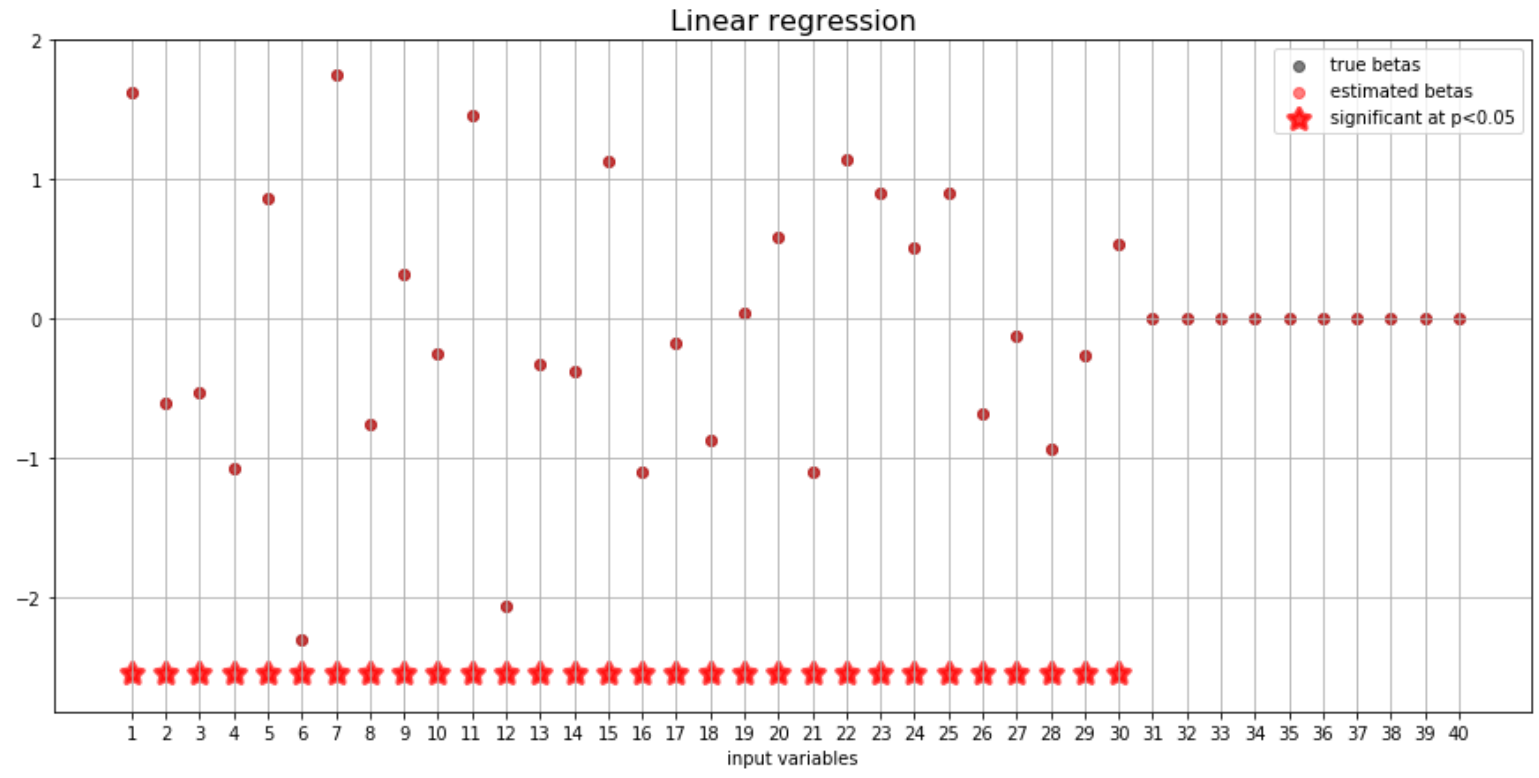
In [15]:

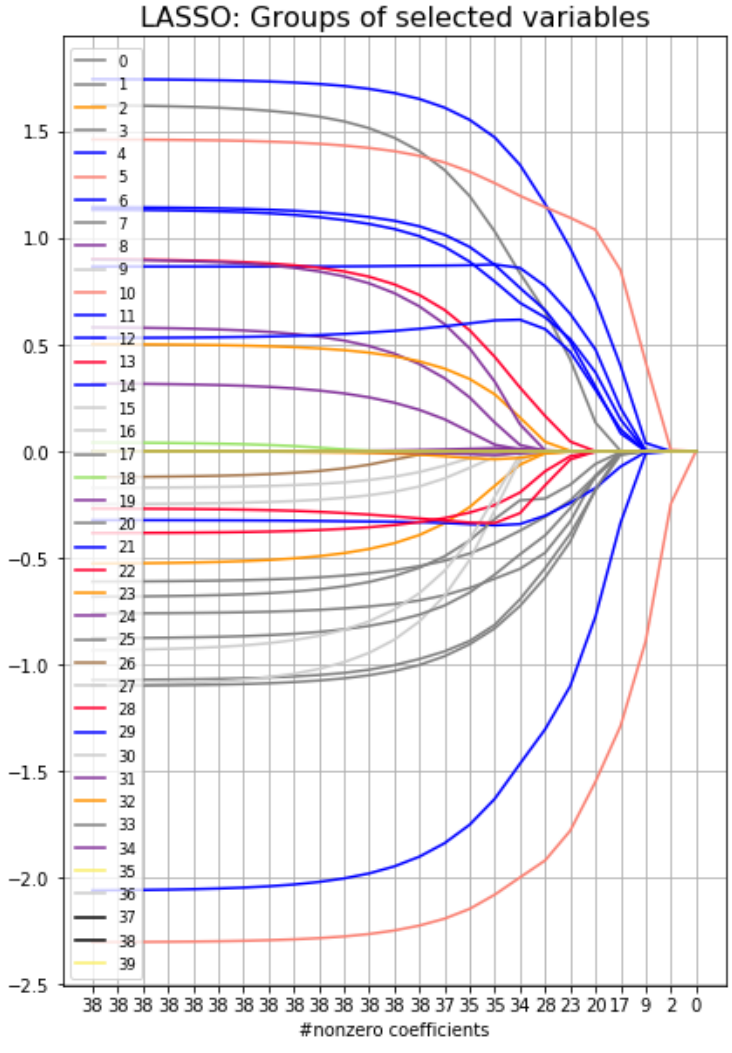
```
comment = "dataset: 30/40 variables relevant, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 100
n_feat = 40
n_feat_relevant = 30
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)
y = (true_coefs * X).sum(axis=1)
C_grid = np.logspace(-3, 0.5, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

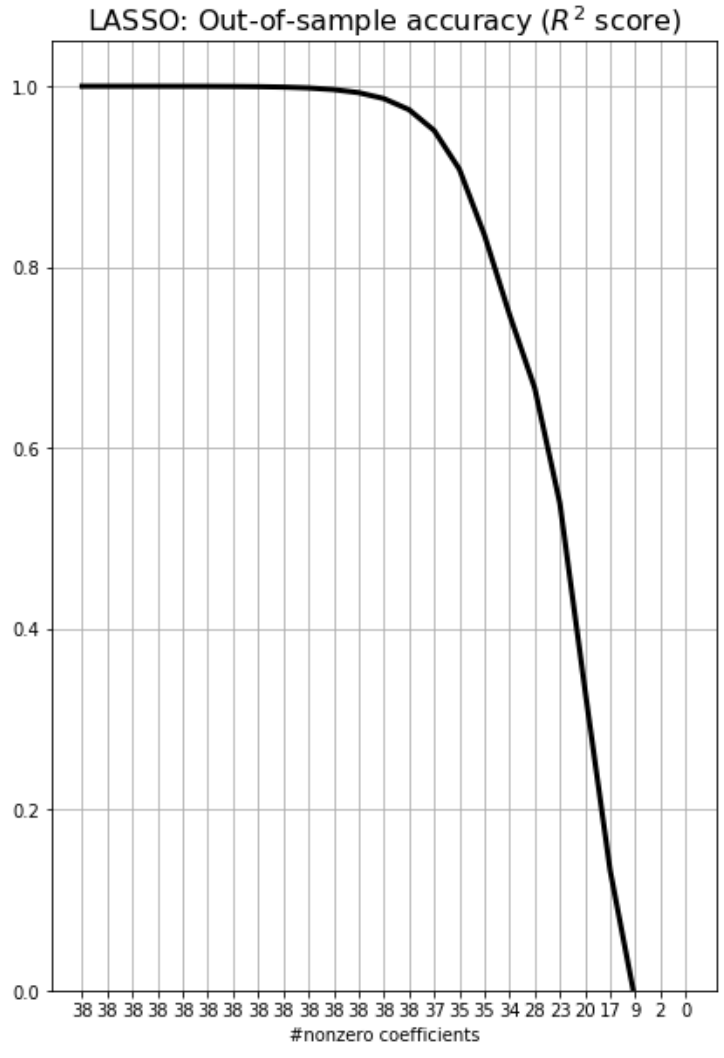
```
alpha: 0.0010 acc: 1.00 active_coefs: 38
alpha: 0.0014 acc: 1.00 active_coefs: 38
alpha: 0.0020 acc: 1.00 active_coefs: 38
alpha: 0.0027 acc: 1.00 active_coefs: 38
```

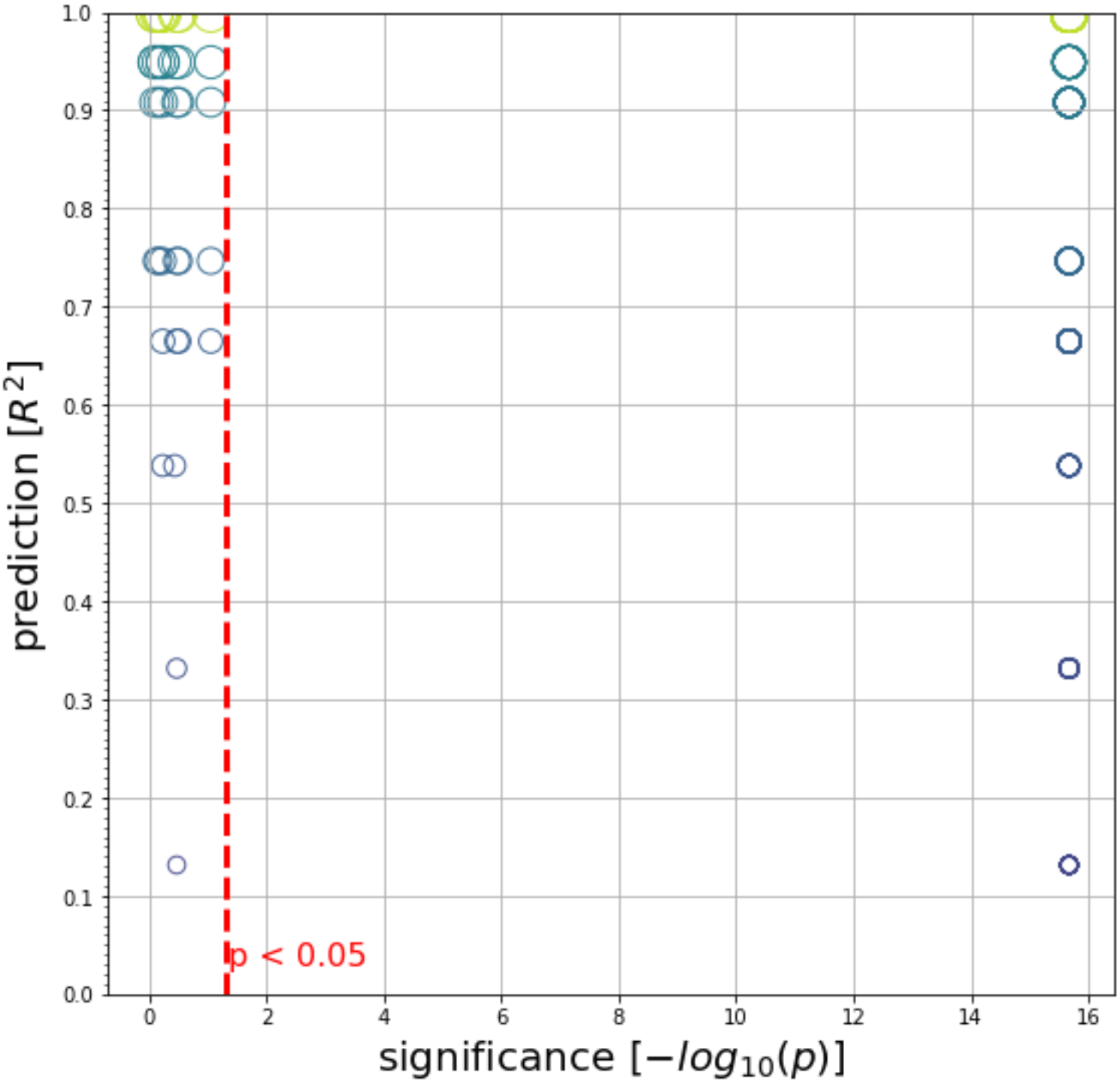
```
alpha: 0.0038 acc: 1.00 active_coefs: 38
alpha: 0.0054 acc: 1.00 active_coefs: 38
alpha: 0.0075 acc: 1.00 active_coefs: 38
alpha: 0.0105 acc: 1.00 active_coefs: 38
alpha: 0.0147 acc: 1.00 active_coefs: 38
alpha: 0.0205 acc: 1.00 active_coefs: 38
alpha: 0.0287 acc: 0.99 active_coefs: 38
alpha: 0.0402 acc: 0.99 active_coefs: 38
alpha: 0.0562 acc: 0.97 active_coefs: 38
alpha: 0.0787 acc: 0.95 active_coefs: 38
alpha: 0.1101 acc: 0.91 active_coefs: 37
alpha: 0.1540 acc: 0.83 active_coefs: 35
alpha: 0.2154 acc: 0.69 active_coefs: 35
alpha: 0.3014 acc: 0.50 active_coefs: 34
alpha: 0.4217 acc: 0.23 active_coefs: 28
alpha: 0.5900 acc: -0.25 active_coefs: 23
alpha: 0.8254 acc: -0.74 active_coefs: 20
alpha: 1.1548 acc: -0.97 active_coefs: 17
alpha: 1.6156 acc: -1.18 active_coefs: 9
alpha: 2.2603 acc: -1.40 active_coefs: 2
alpha: 3.1623 acc: -1.45 active_coefs: 0
40
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
1.04259815874
```

```
0.542099041852
0.495852910005
0.49151759281
0.437191821452
0.219545825874
0.196203276168
0.143810303355
0.0920859920704
0.0708849363353
skipping 1
skipping 2
skipping 3
skipping 4
skipping 5
skipping 6
skipping 7
skipping 8
skipping 9
skipping 10
skipping 11
skipping 12
skipping 13
skipping 16
```




LASSO: Groups of selected variables
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
38 38 38 38 38 38 38 38 38 38 38 38 38 37 35 35 34 28 23 20 17 9 2 0
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
1.0
0.8
0.6
0.4
0.2
0.0
38 38 38 38 38 38 38 38 38 38 38 38 38 37 35 35 34 28 23 20 17 9 2 0
#nonzero coefficients



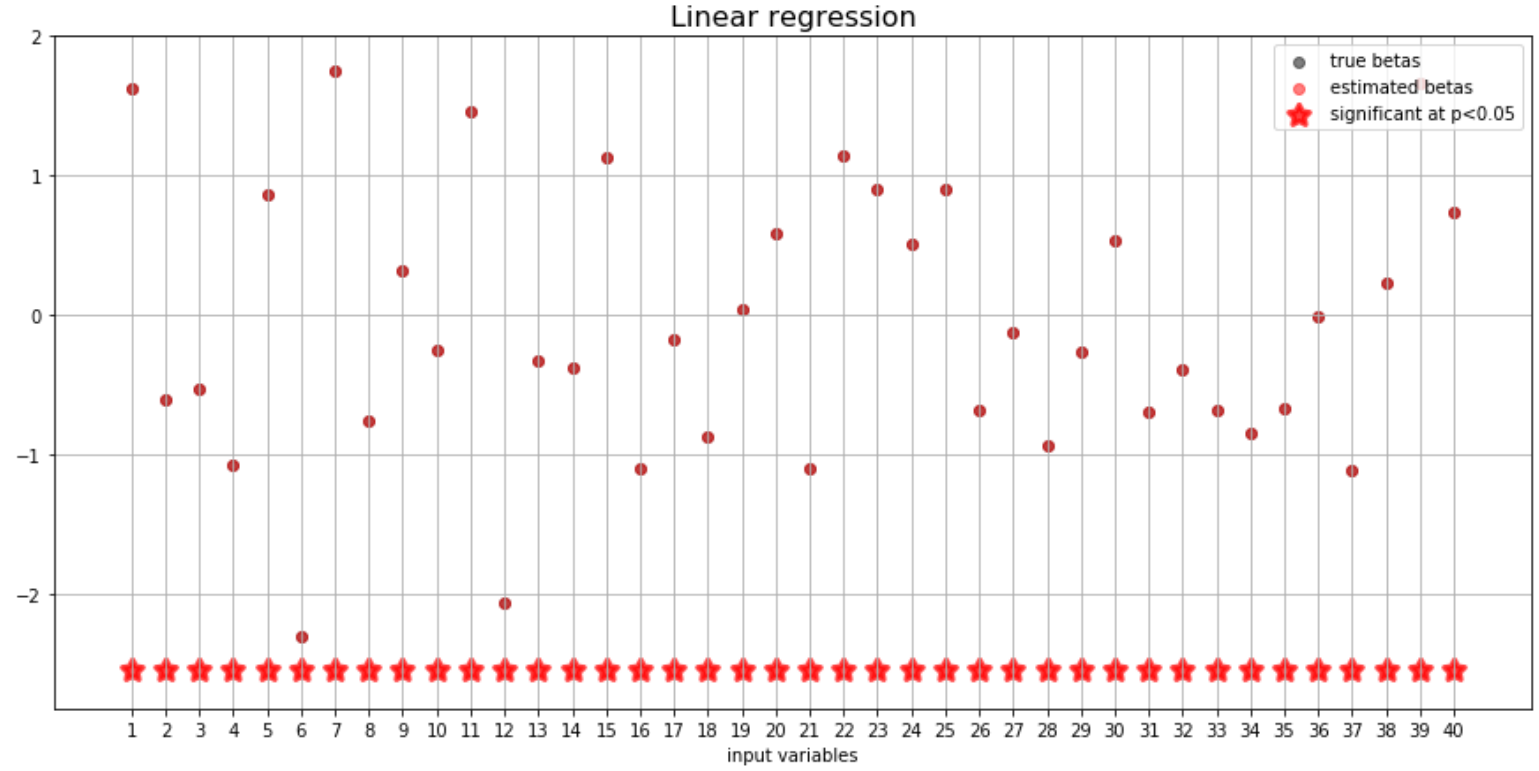
In [16]:

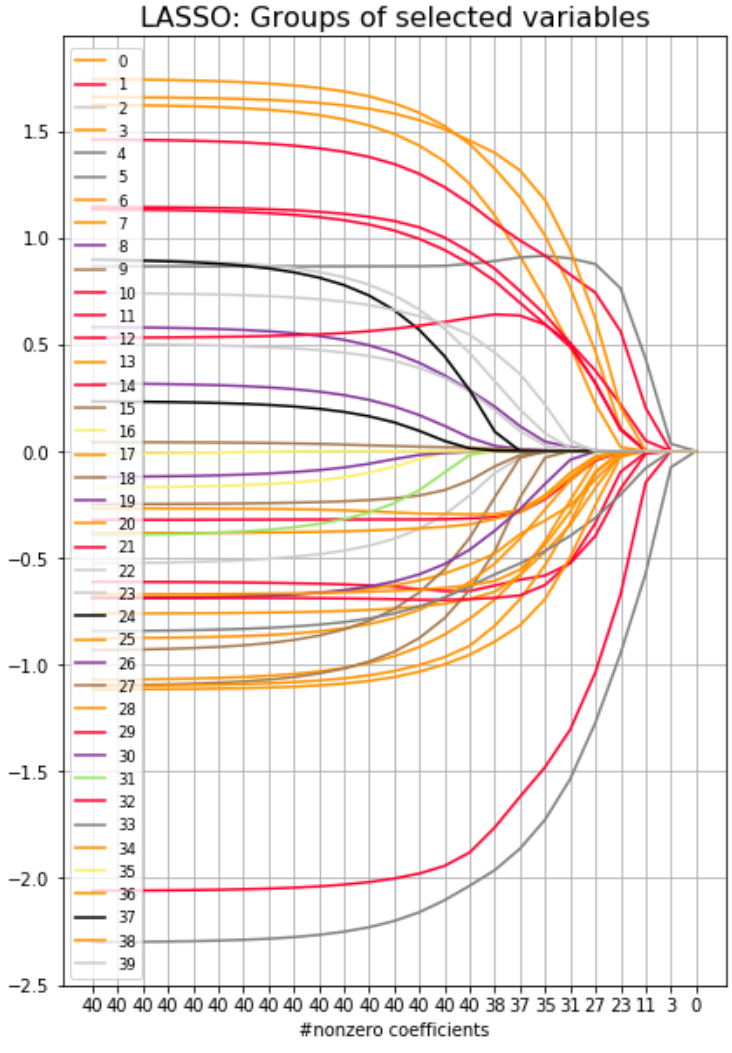
```
comment = "dataset: 40/40 variables relevant, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 100
n_feat = 40
n_feat_relevant = 40
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)
y = (true_coefs * X).sum(axis=1)
C_grid = np.logspace(-3, 0.5, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

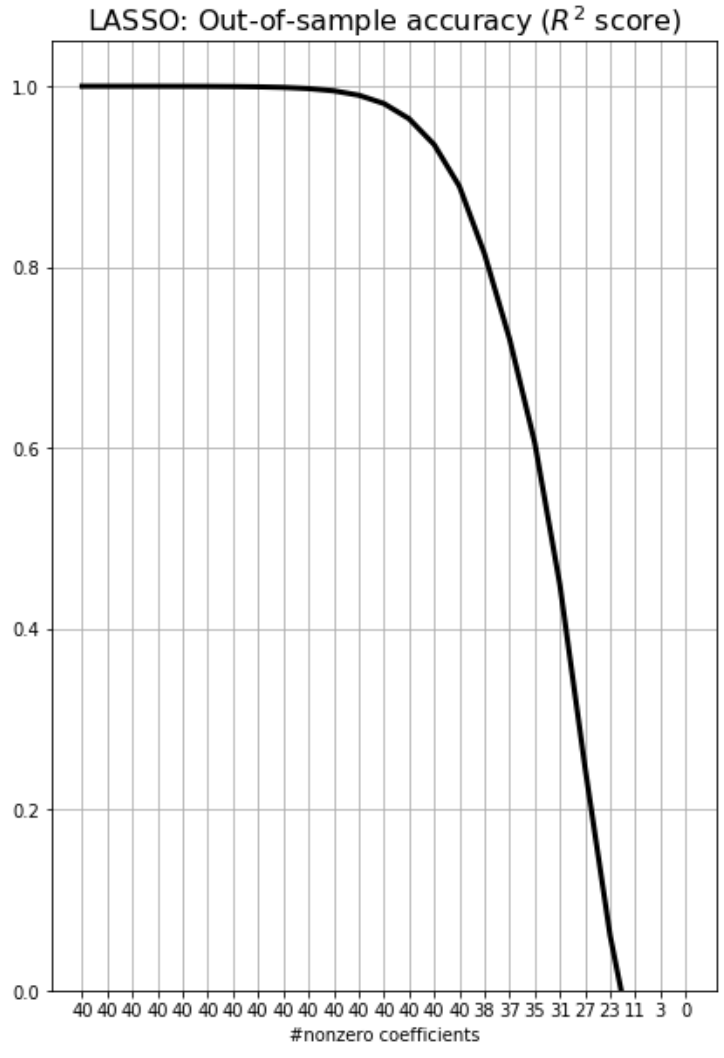
```
alpha: 0.0010 acc: 1.00 active_coefs: 40
alpha: 0.0014 acc: 1.00 active_coefs: 40
alpha: 0.0020 acc: 1.00 active_coefs: 40
alpha: 0.0027 acc: 1.00 active_coefs: 40
```

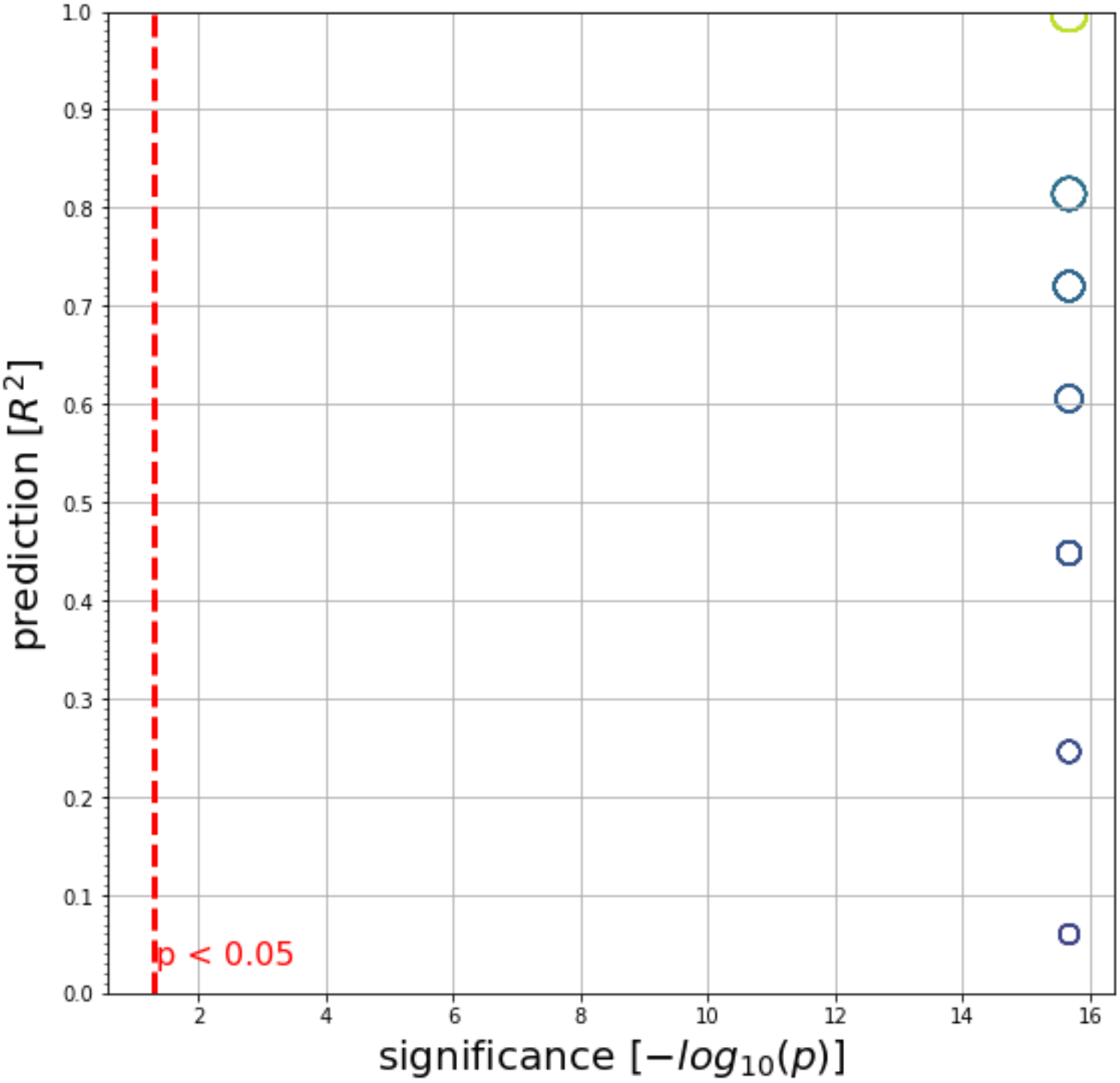
```
alpha: 0.0038 acc: 1.00 active_coefs: 40
alpha: 0.0054 acc: 1.00 active_coefs: 40
alpha: 0.0075 acc: 1.00 active_coefs: 40
alpha: 0.0105 acc: 1.00 active_coefs: 40
alpha: 0.0147 acc: 1.00 active_coefs: 40
alpha: 0.0205 acc: 1.00 active_coefs: 40
alpha: 0.0287 acc: 0.99 active_coefs: 40
alpha: 0.0402 acc: 0.98 active_coefs: 40
alpha: 0.0562 acc: 0.97 active_coefs: 40
alpha: 0.0787 acc: 0.94 active_coefs: 40
alpha: 0.1101 acc: 0.88 active_coefs: 40
alpha: 0.1540 acc: 0.81 active_coefs: 40
alpha: 0.2154 acc: 0.68 active_coefs: 38
alpha: 0.3014 acc: 0.53 active_coefs: 37
alpha: 0.4217 acc: 0.37 active_coefs: 35
alpha: 0.5900 acc: 0.13 active_coefs: 31
alpha: 0.8254 acc: -0.17 active_coefs: 27
alpha: 1.1548 acc: -0.21 active_coefs: 23
alpha: 1.6156 acc: -0.30 active_coefs: 11
alpha: 2.2603 acc: -0.51 active_coefs: 3
alpha: 3.1623 acc: -0.52 active_coefs: 0
40
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
```

```
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
skipping 1
skipping 2
skipping 3
skipping 4
skipping 5
skipping 6
skipping 7
skipping 8
skipping 9
skipping 10
skipping 11
skipping 12
skipping 13
skipping 14
skipping 15
```




LASSO: Groups of selected variables
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 38 37 35 31 27 23 11 3 0
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
1.0
0.8
0.6
0.4
0.2
0.0
40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 38 37 35 31 27 23 11 3 0
#nonzero coefficients



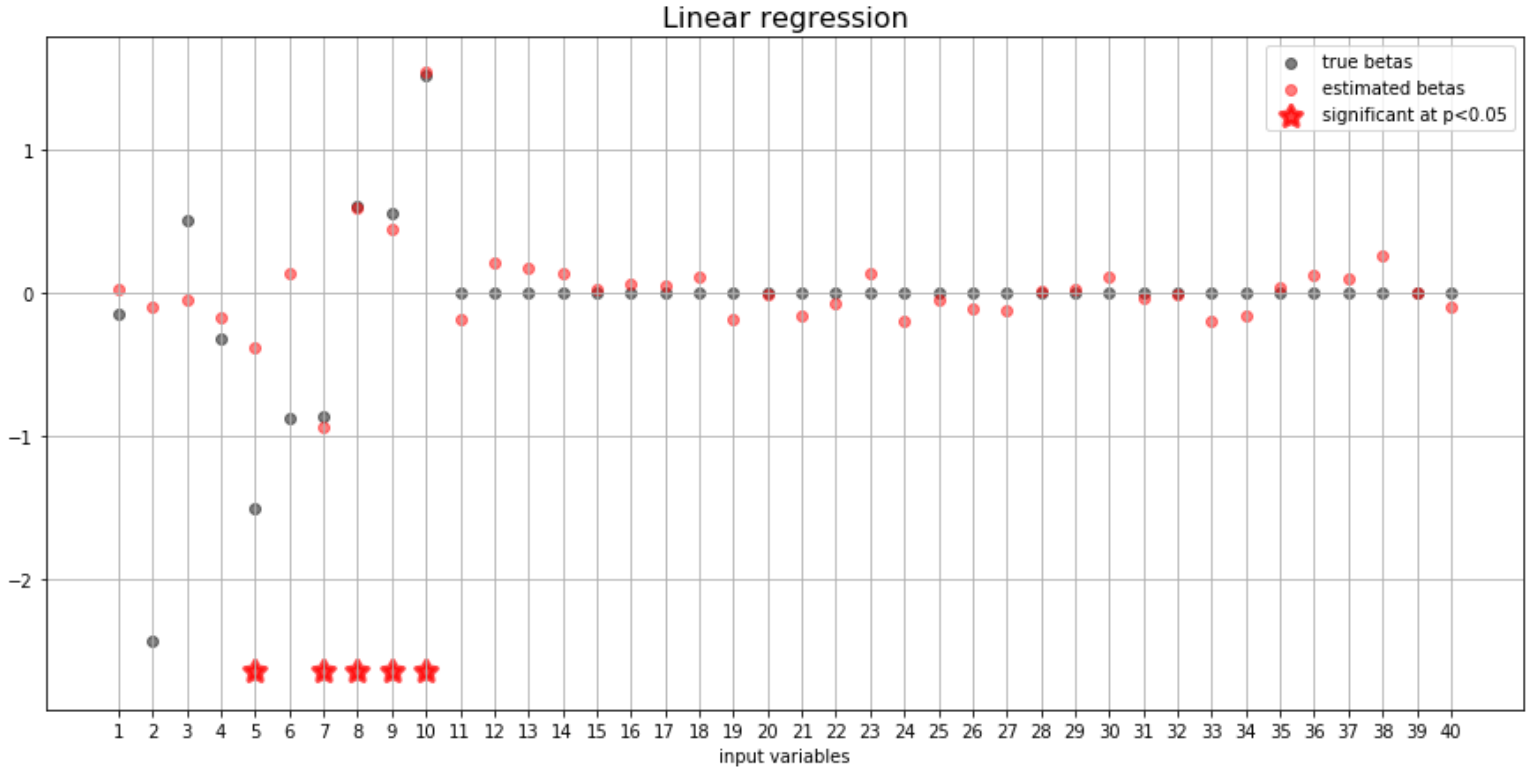
Pathological cases with diverging ground-truth model (abs, log, exp, sqrt, 1/x): 1000 samples, 40 variables, error = N(0,1)

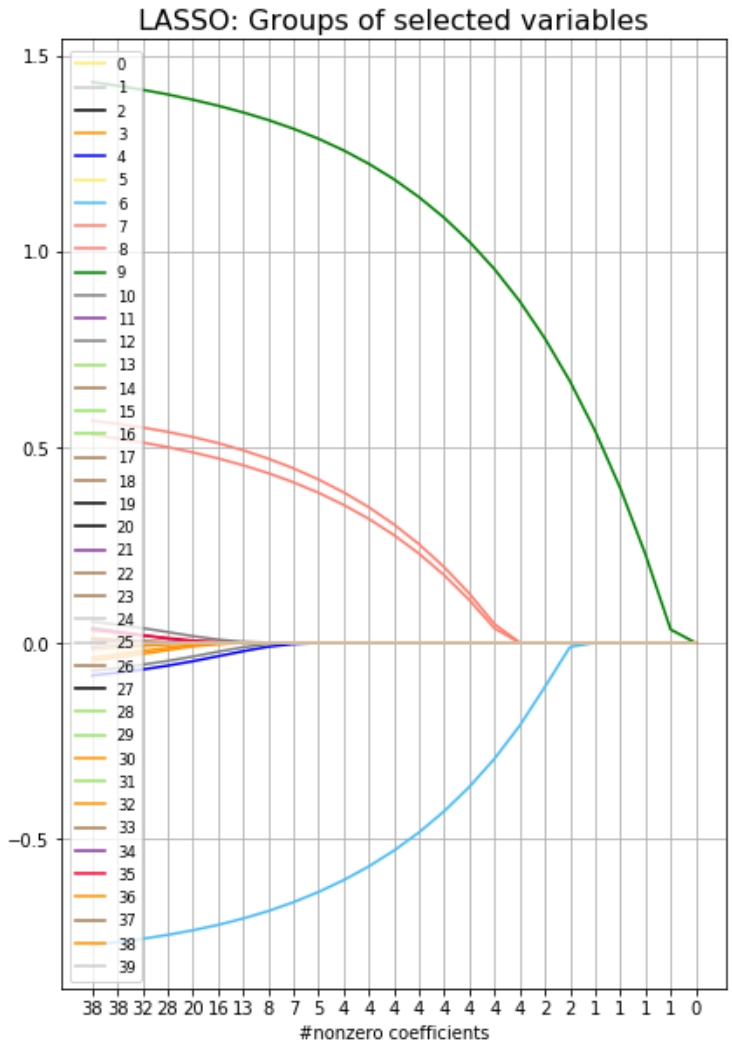
In [17]:

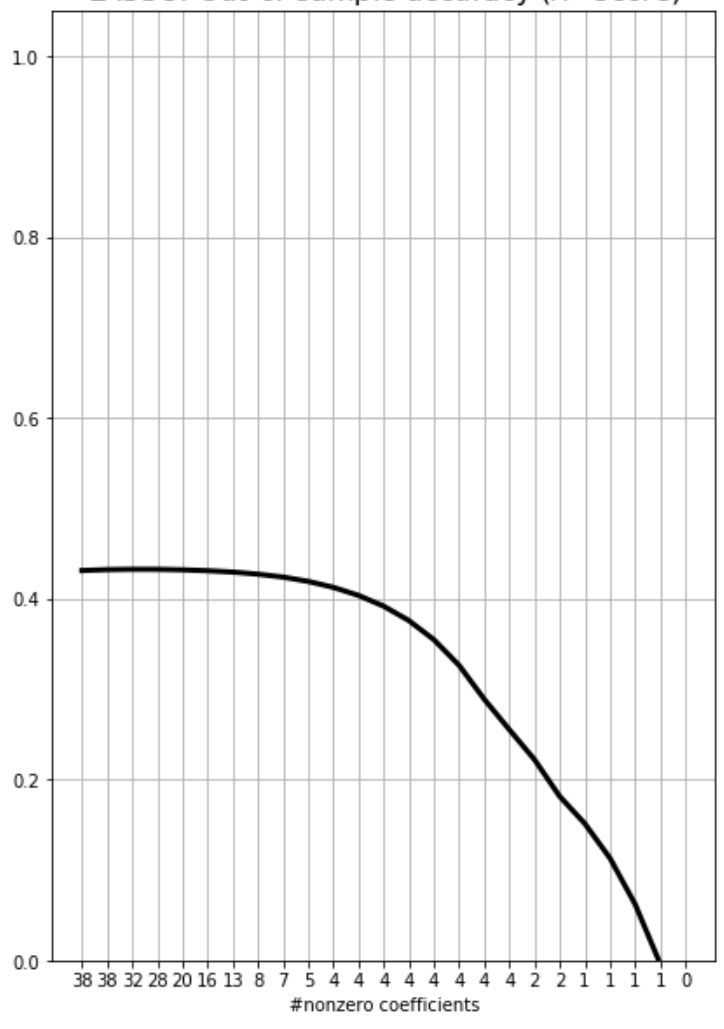
```
comment = "abs / dataset: 10/40 relevant variables, including 5 pathological one
s, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

X_abs5 = X.copy()
X_abs5[:, 0:6] = np.abs(X_abs5[:, 0:6])  # introduce pathological transformation
, NOT captured by ground-truth model
y = (true_coefs * X_abs5).sum(axis=1) + epsilon
C_grid = np.logspace(-1.25, 0.25, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

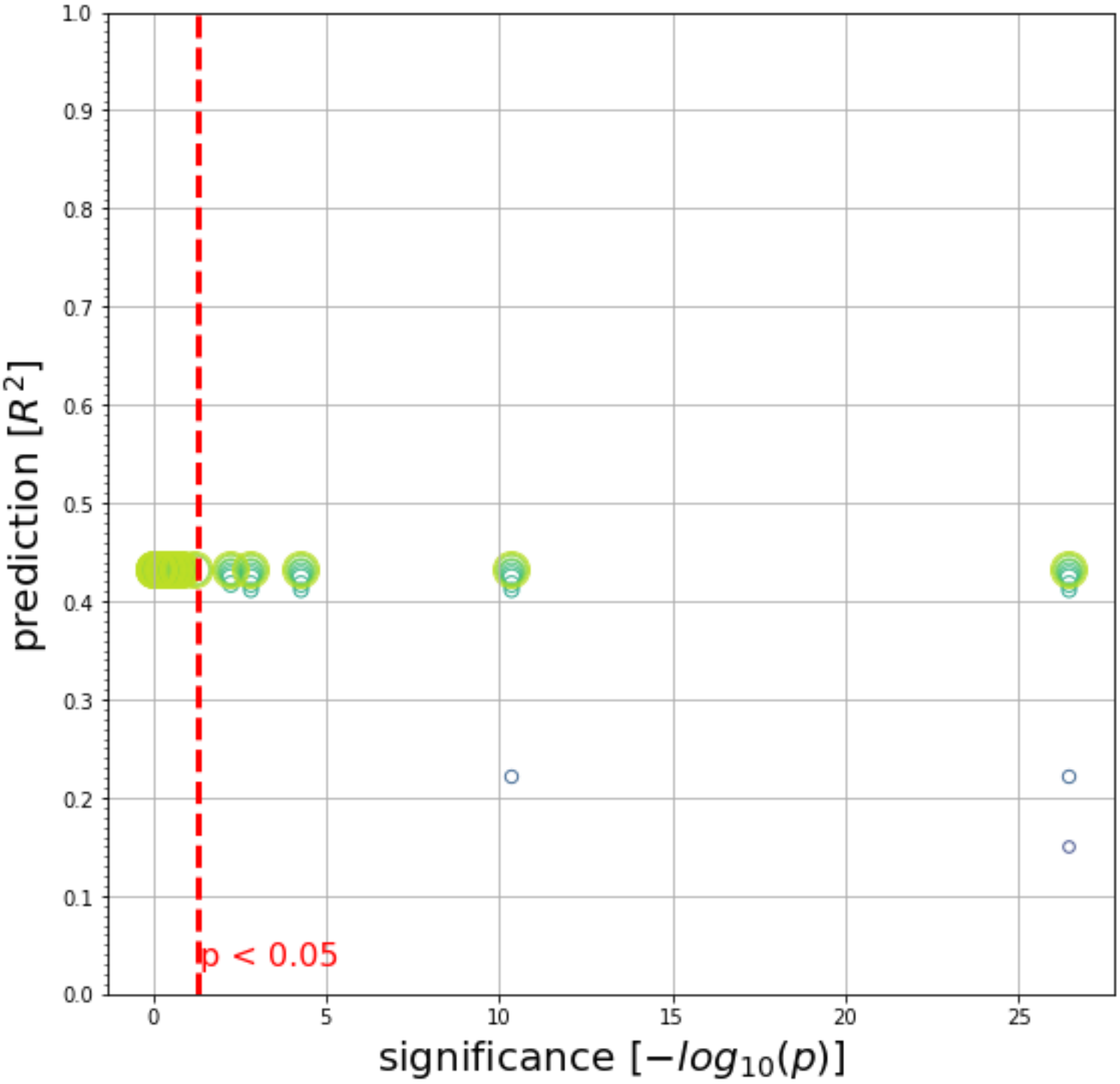
```
alpha: 0.0562 acc: 0.47 active_coefs: 38
alpha: 0.0649 acc: 0.47 active_coefs: 38
alpha: 0.0750 acc: 0.47 active_coefs: 32
alpha: 0.0866 acc: 0.48 active_coefs: 28
alpha: 0.1000 acc: 0.48 active_coefs: 20
alpha: 0.1155 acc: 0.48 active_coefs: 16
alpha: 0.1334 acc: 0.48 active_coefs: 13
alpha: 0.1540 acc: 0.48 active_coefs: 8
alpha: 0.1778 acc: 0.48 active_coefs: 7
alpha: 0.2054 acc: 0.47 active_coefs: 5
alpha: 0.2371 acc: 0.47 active_coefs: 4
alpha: 0.2738 acc: 0.46 active_coefs: 4
alpha: 0.3162 acc: 0.45 active_coefs: 4
alpha: 0.3652 acc: 0.43 active_coefs: 4
alpha: 0.4217 acc: 0.41 active_coefs: 4
alpha: 0.4870 acc: 0.37 active_coefs: 4
alpha: 0.5623 acc: 0.33 active_coefs: 4
alpha: 0.6494 acc: 0.30 active_coefs: 4
alpha: 0.7499 acc: 0.27 active_coefs: 2
alpha: 0.8660 acc: 0.23 active_coefs: 2
alpha: 1.0000 acc: 0.18 active_coefs: 1
alpha: 1.1548 acc: 0.13 active_coefs: 1
alpha: 1.3335 acc: 0.05 active_coefs: 1
alpha: 1.5399 acc: -0.04 active_coefs: 1
alpha: 1.7783 acc: -0.04 active_coefs: 0
40
26.4383494167
10.3462209366
4.23466877222
2.78571500508
```

```
2.20846067323
1.18370072913
0.81815340767
0.788407715192
0.742723127962
0.709485761504
0.69292276907
0.627588808255
0.594158647105
0.566090925226
0.561877613573
0.48538106739
0.466614227898
0.433916655934
0.426898127196
0.399545782116
0.369619063469
0.365035125402
0.35338066123
0.346051523773
0.297972029669
0.294571646309
0.241003713854
0.170642232506
0.154803276805
0.154775808307
0.154184528639
0.11805336642
0.0915052577244
0.0757410600265
0.068502592059
0.0674567479775
0.0437747763522
0.0416921815627
0.0354366878206
0.0151945430138
skipping 4
skipping 11
skipping 12
skipping 13
skipping 14
skipping 15
skipping 16
skipping 17
skipping 19
skipping 21
skipping 22
skipping 23
```


Linear regression
true betas
estimated betas
significant at p<0.05
input variables


LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients



In [18]:

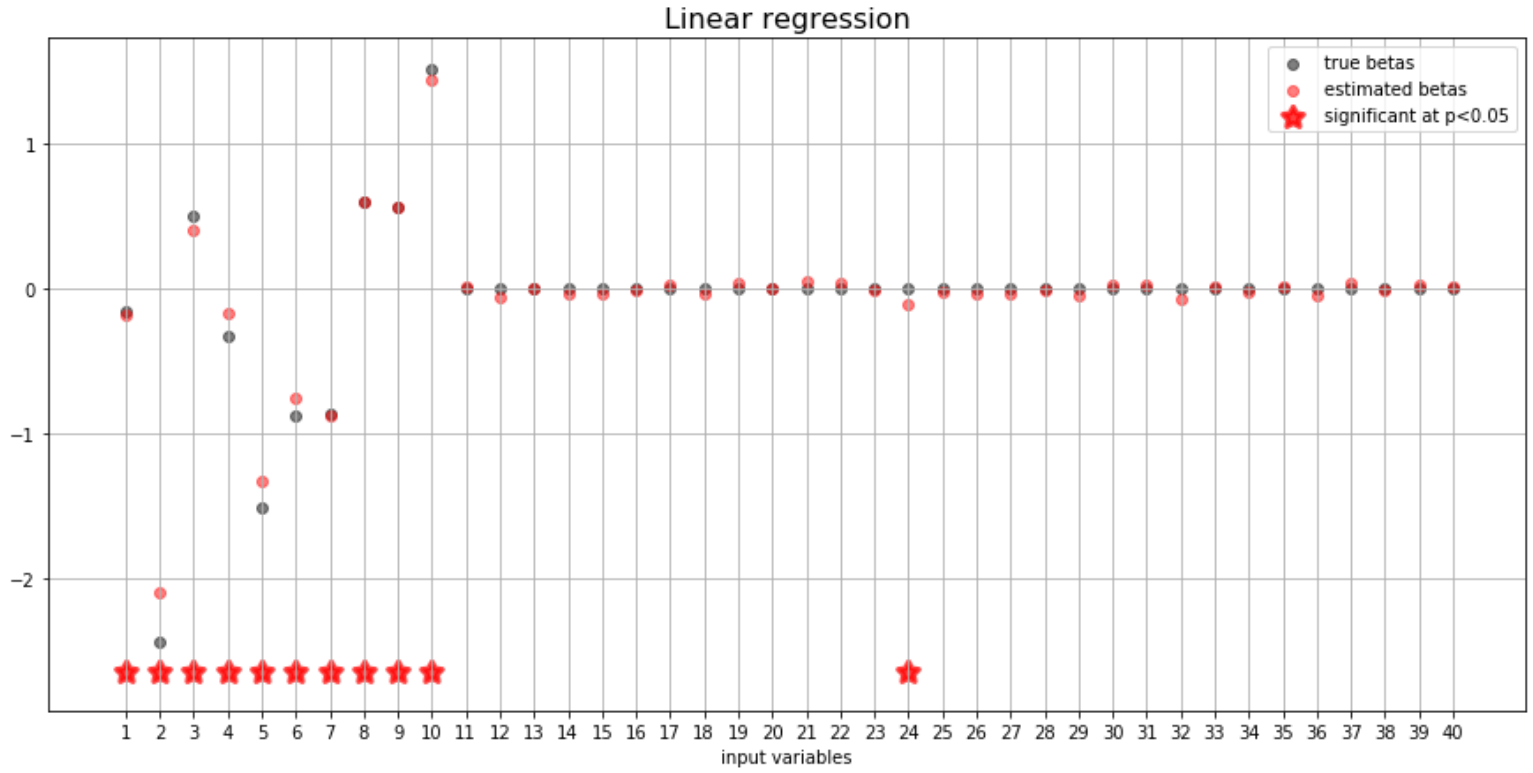
```
### observation: 1/5 path. variables are significant, the same 1/5 path. (?) var
iables are selected as predictive
```

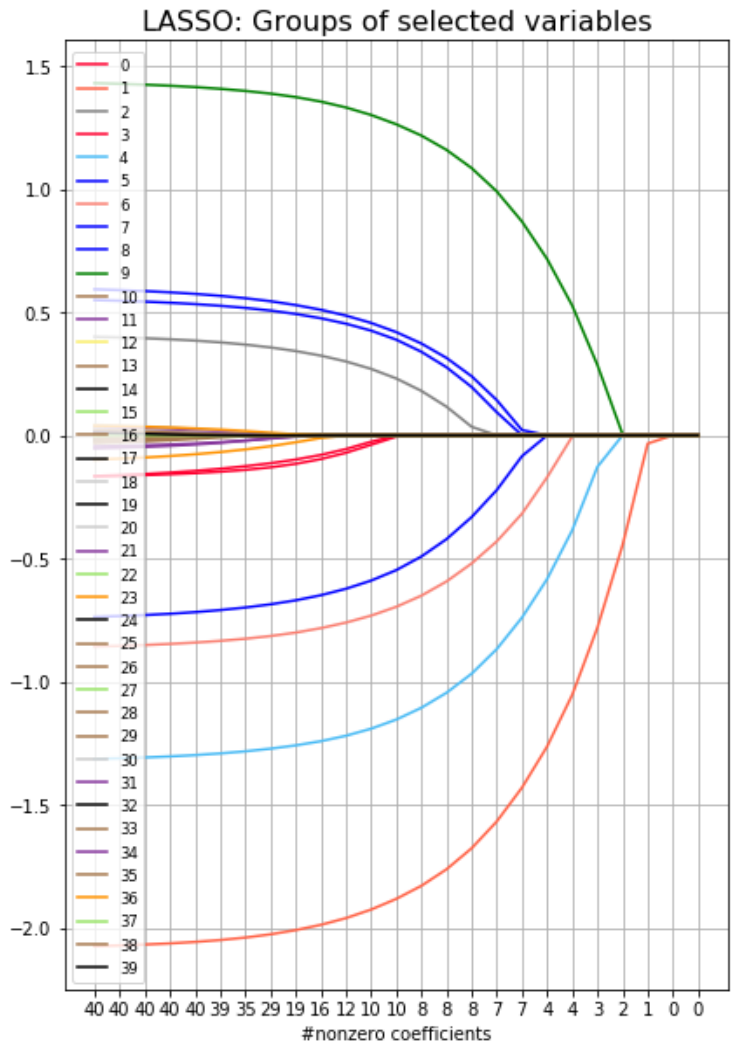
In [19]:

```
comment = "square root / dataset: 10/40 relevant variables, including 5 patholog
ical ones, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

X_sqrt5 = X.copy()
signs = np.sign(X)
X_sqrt5[:, 0:6] = signs[:, 0:6] * np.sqrt(np.abs(X_sqrt5[:, 0:6]))  # introduce
pathological transformation, NOT captured by ground-truth model
y = (true_coefs * X_sqrt5).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 0.5, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

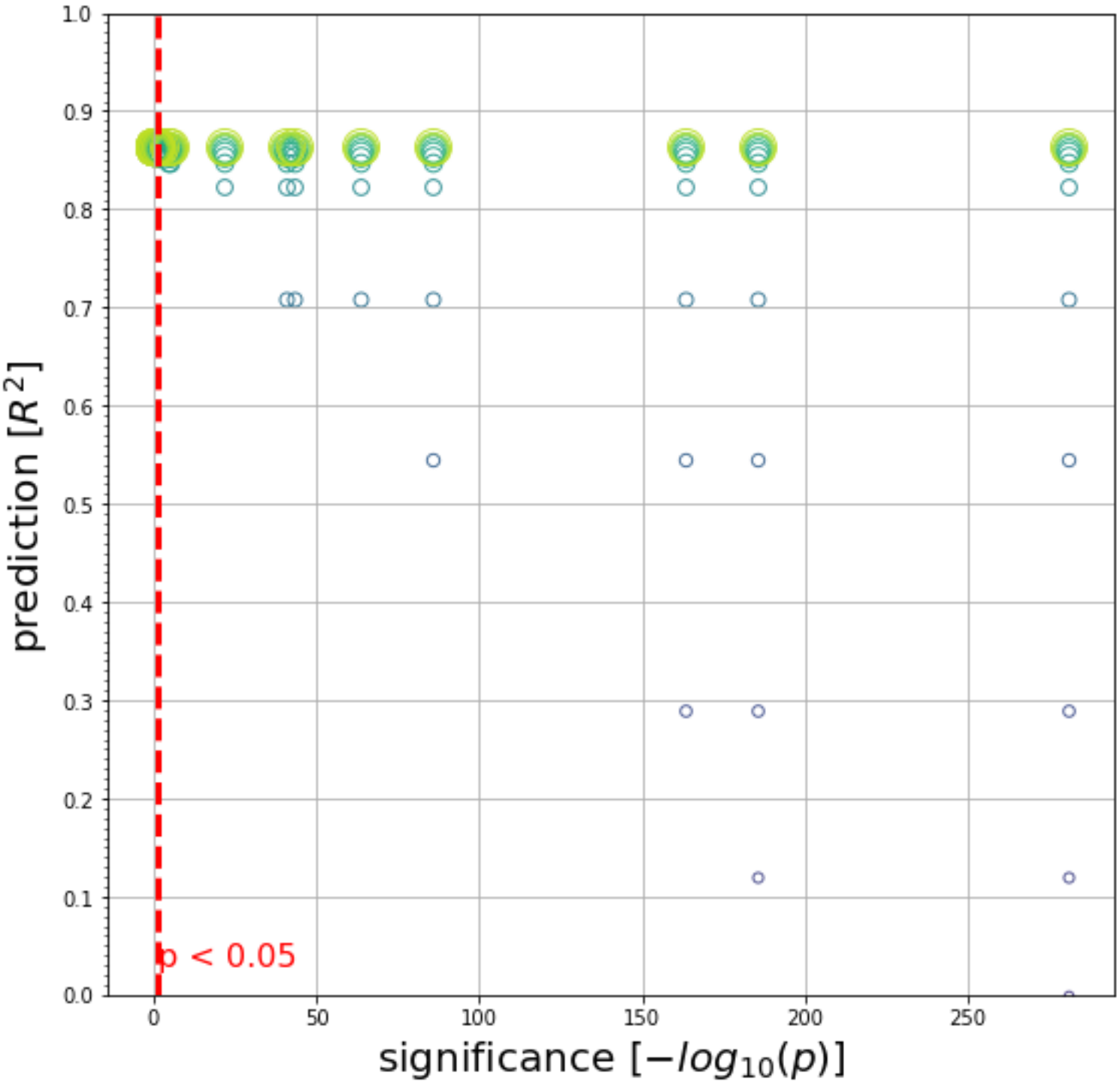
```
alpha: 0.0100 acc: 0.88 active_coefs: 40
alpha: 0.0127 acc: 0.88 active_coefs: 40
alpha: 0.0162 acc: 0.88 active_coefs: 40
alpha: 0.0205 acc: 0.88 active_coefs: 40
alpha: 0.0261 acc: 0.88 active_coefs: 40
alpha: 0.0332 acc: 0.88 active_coefs: 39
alpha: 0.0422 acc: 0.88 active_coefs: 35
alpha: 0.0536 acc: 0.88 active_coefs: 29
alpha: 0.0681 acc: 0.88 active_coefs: 19
alpha: 0.0866 acc: 0.88 active_coefs: 16
alpha: 0.1101 acc: 0.87 active_coefs: 12
alpha: 0.1399 acc: 0.87 active_coefs: 10
alpha: 0.1778 acc: 0.86 active_coefs: 10
alpha: 0.2260 acc: 0.84 active_coefs: 8
alpha: 0.2873 acc: 0.82 active_coefs: 8
alpha: 0.3652 acc: 0.79 active_coefs: 8
alpha: 0.4642 acc: 0.74 active_coefs: 7
alpha: 0.5900 acc: 0.67 active_coefs: 7
alpha: 0.7499 acc: 0.58 active_coefs: 4
alpha: 0.9532 acc: 0.46 active_coefs: 4
alpha: 1.2115 acc: 0.28 active_coefs: 3
alpha: 1.5399 acc: 0.12 active_coefs: 2
alpha: 1.9573 acc: 0.00 active_coefs: 1
alpha: 2.4879 acc: -0.00 active_coefs: 0
alpha: 3.1623 acc: -0.00 active_coefs: 0
40
280.81838128
185.051650597
162.88201745
85.3330991348
```

```
63.4917864069
43.377949246
40.80446038
21.9604985602
5.04246713604
4.26929741094
1.96706005172
0.966373233503
0.778172567731
0.745373058737
0.569681434625
0.55625387768
0.531978428786
0.524489358896
0.432711005565
0.378456321842
0.376749381122
0.370754243673
0.359919320587
0.333398247406
0.282094917425
0.281170933348
0.275016710205
0.271553772295
0.237934528659
0.237372937007
0.202448397809
0.197458054233
0.161273189598
0.151824327378
0.078635501981
0.0456266473681
0.0442190444804
0.0241547741538
0.0221343750767
0.0141042018173
skipping 3
skipping 4
skipping 6
skipping 7
skipping 12
skipping 14
skipping 15
skipping 17
skipping 19
skipping 24
```


Linear regression
true betas
estimated betas
significant at p<0.05
input variables


LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients



In [20]:

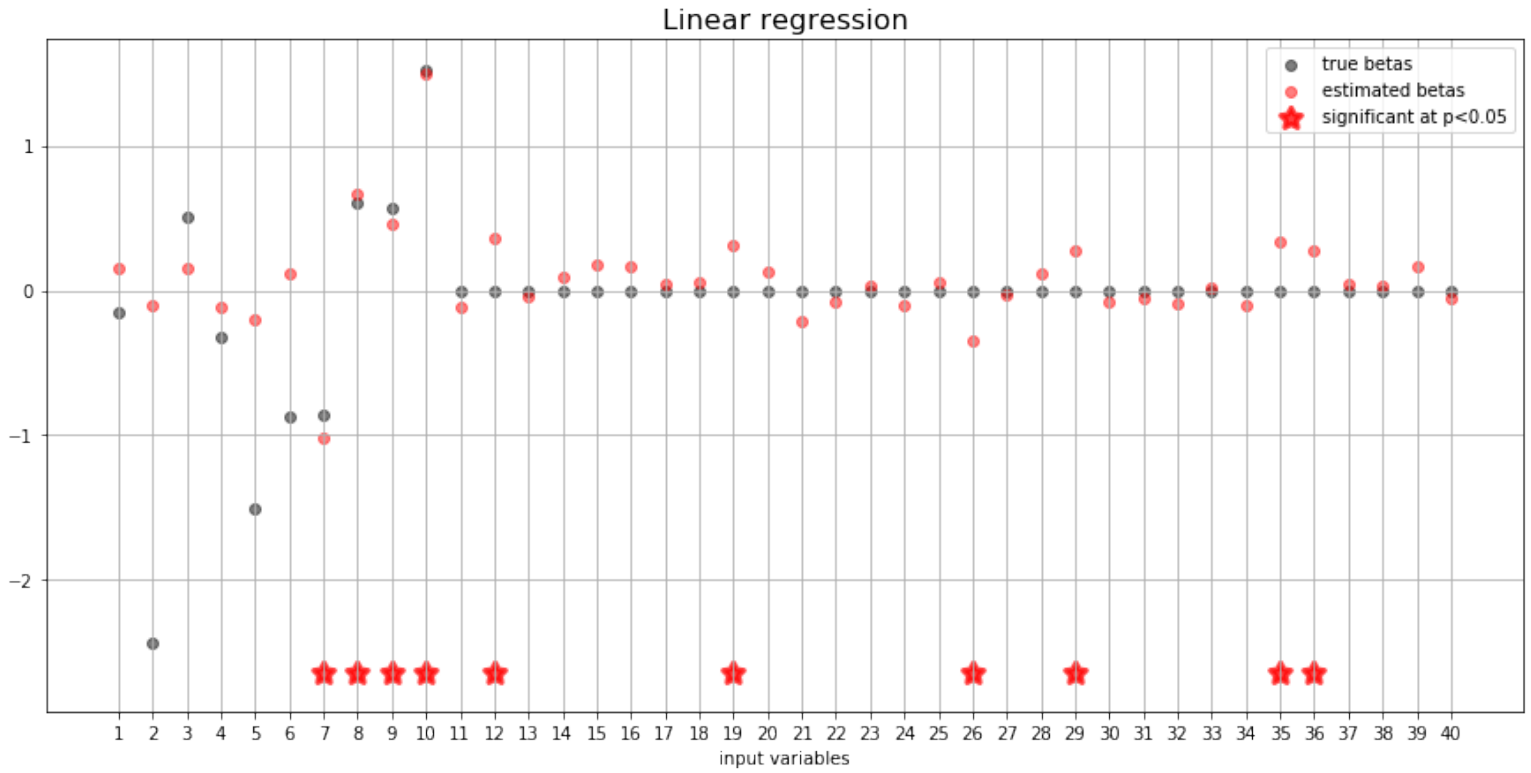
```
### observation: 5/5 path. variables are significant, the same 5/5 path. (?) var
iables are selected as predictive
```

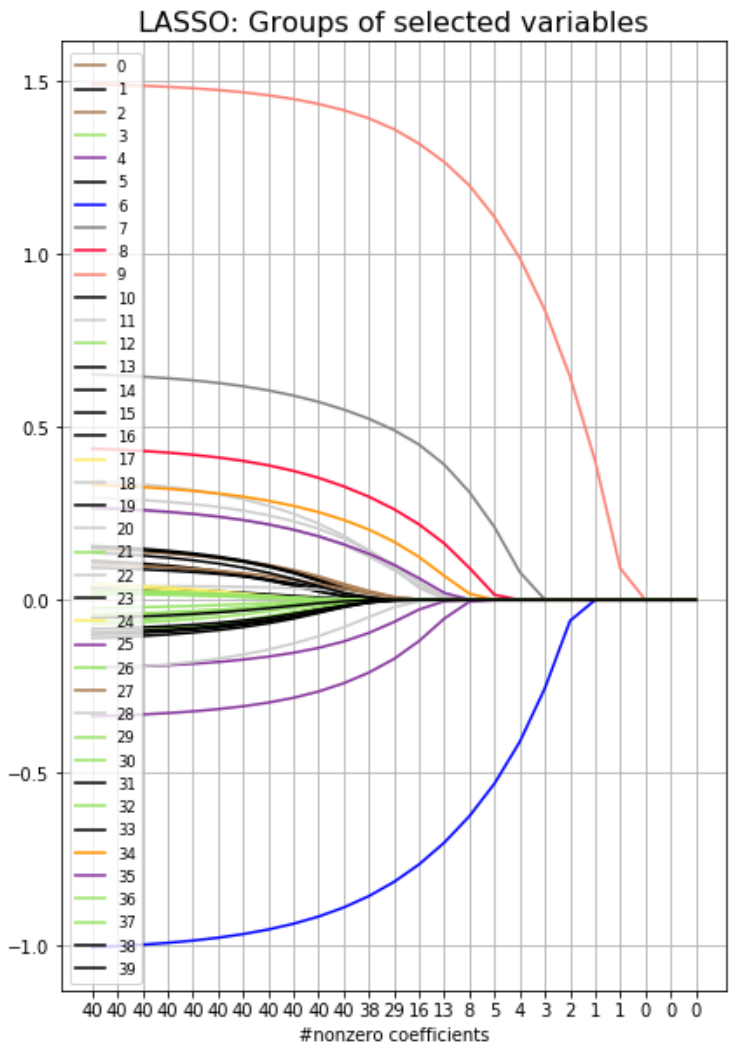
In [21]:

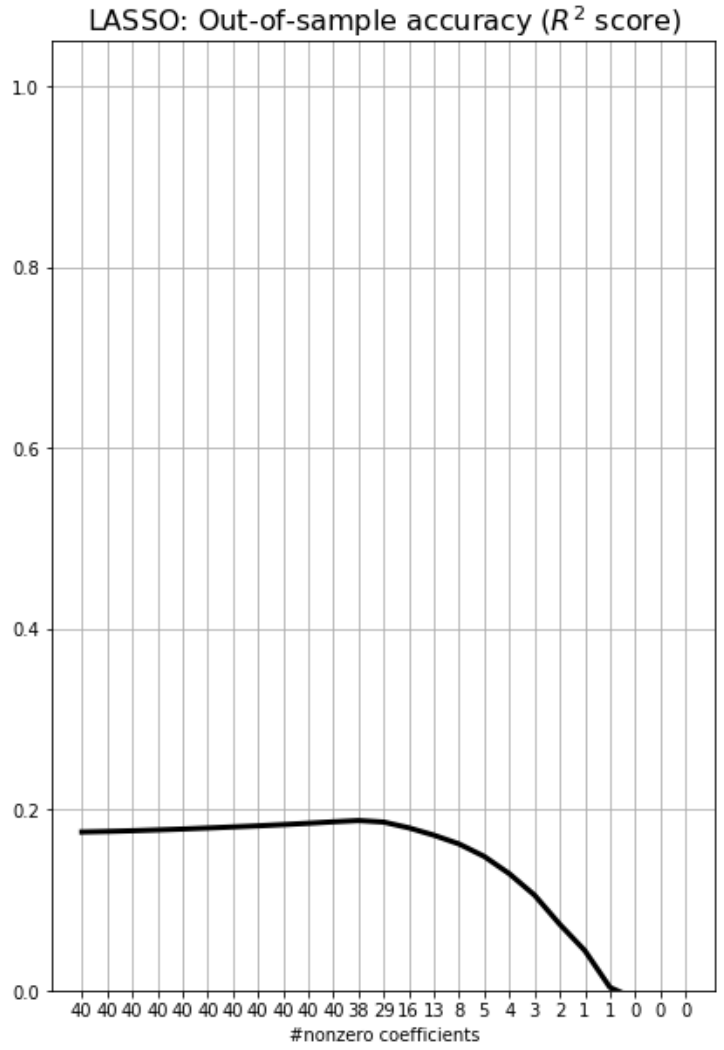
```
comment = "logn / dataset: 10/40 relevant variables, including 5 pathological on
es, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

X_log5 = X.copy()
signs = np.sign(X)
# Bertrand: no sign update
# Bertrand: rather log (X + constant); such as constant=4
X_log5[:, 0:6] = signs[:, 0:6] * np.log(np.abs(X_log5[:, 0:6]))  # introduce pat
hological transformation, NOT captured by ground-truth model
y = (true_coefs * X_log5).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 0.5, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

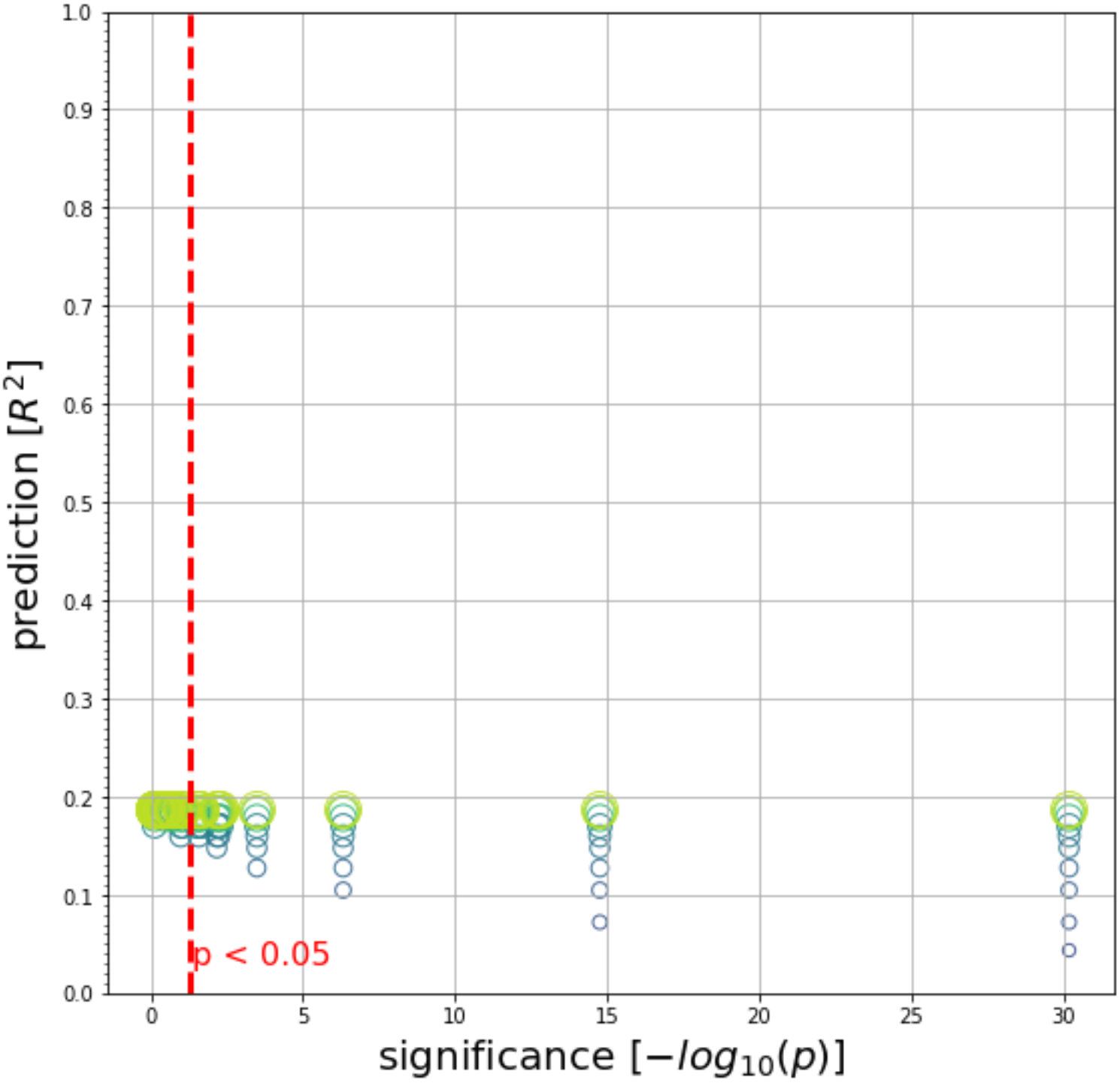
```
alpha: 0.0100 acc: 0.13 active_coefs: 40
alpha: 0.0127 acc: 0.13 active_coefs: 40
alpha: 0.0162 acc: 0.13 active_coefs: 40
alpha: 0.0205 acc: 0.13 active_coefs: 40
alpha: 0.0261 acc: 0.13 active_coefs: 40
alpha: 0.0332 acc: 0.13 active_coefs: 40
alpha: 0.0422 acc: 0.13 active_coefs: 40
alpha: 0.0536 acc: 0.14 active_coefs: 40
alpha: 0.0681 acc: 0.14 active_coefs: 40
alpha: 0.0866 acc: 0.14 active_coefs: 40
alpha: 0.1101 acc: 0.14 active_coefs: 40
alpha: 0.1399 acc: 0.15 active_coefs: 38
alpha: 0.1778 acc: 0.16 active_coefs: 29
alpha: 0.2260 acc: 0.15 active_coefs: 16
alpha: 0.2873 acc: 0.15 active_coefs: 13
alpha: 0.3652 acc: 0.15 active_coefs: 8
alpha: 0.4642 acc: 0.14 active_coefs: 5
alpha: 0.5900 acc: 0.13 active_coefs: 4
alpha: 0.7499 acc: 0.12 active_coefs: 3
alpha: 0.9532 acc: 0.09 active_coefs: 2
alpha: 1.2115 acc: 0.06 active_coefs: 1
alpha: 1.5399 acc: 0.01 active_coefs: 1
alpha: 1.9573 acc: -0.00 active_coefs: 0
alpha: 2.4879 acc: -0.00 active_coefs: 0
alpha: 3.1623 acc: -0.00 active_coefs: 0
40
30.1690564789
14.7235400932
```

```
6.29702760204
3.44313078711
2.27256983837
2.17952649457
2.14748133614
1.63682352844
1.53512098893
1.49830164697
0.984371184175
0.965015596298
0.748898813483
0.660653418414
0.657087217632
0.655709193224
0.649315170553
0.510438462762
0.459474927836
0.451883676344
0.418696576568
0.403180556243
0.377593082023
0.362440684088
0.350750473413
0.335545717201
0.33446417349
0.27989359913
0.257622766449
0.198178076446
0.192683829352
0.169184836979
0.151540381614
0.143814167025
0.143062010504
0.122489759224
0.104997363716
0.0947527121556
0.0788067961437
0.0710569600832
skipping 3
skipping 4
skipping 5
skipping 6
skipping 8
skipping 9
skipping 10
skipping 11
skipping 12
skipping 13
skipping 21
skipping 23
skipping 24
```


Linear regression
true betas
estimated betas
significant at p<0.05
input variables


LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients



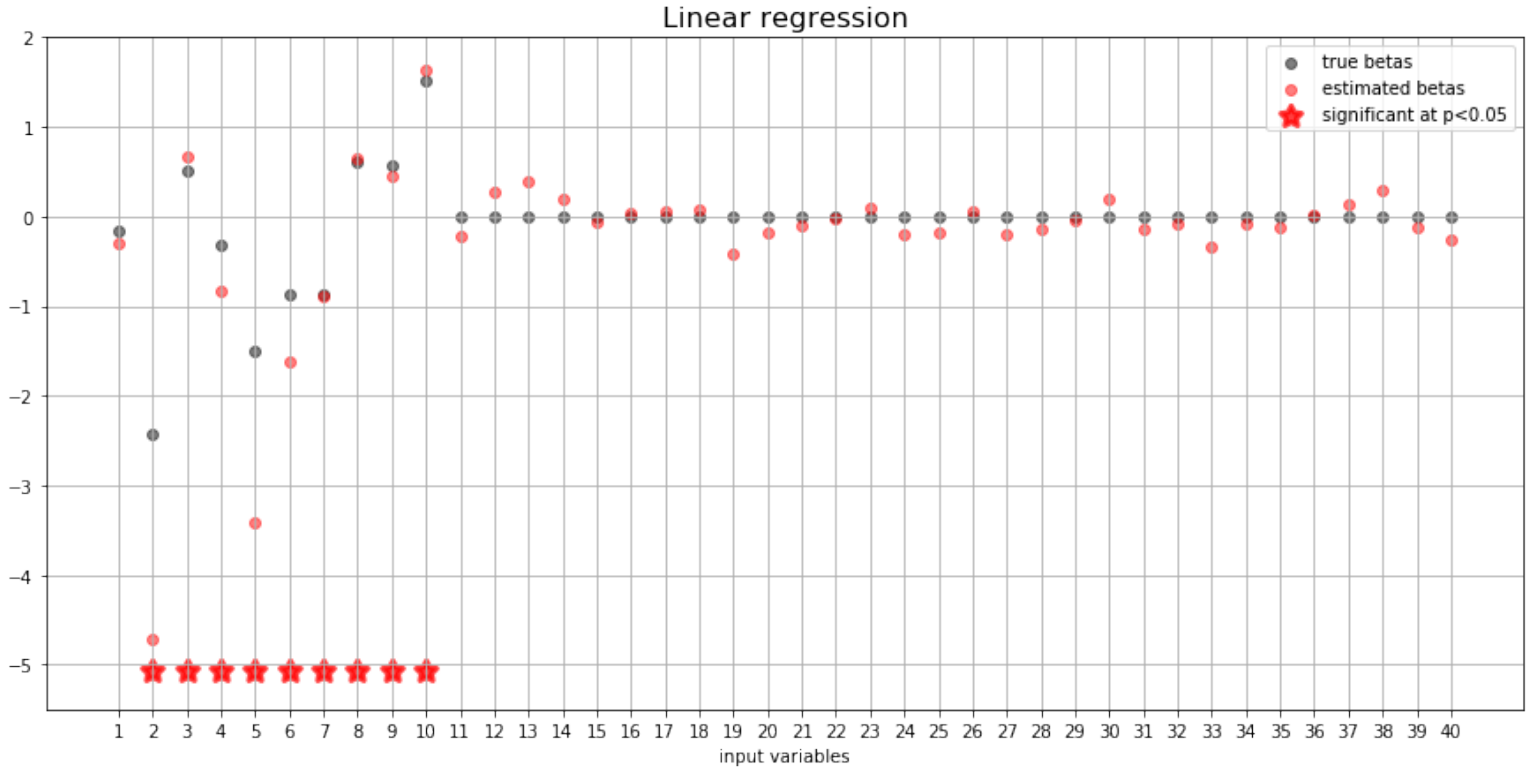
observation: 0/5 path. variables are significant, the same 1/5 path. variables are selected as predictive ->

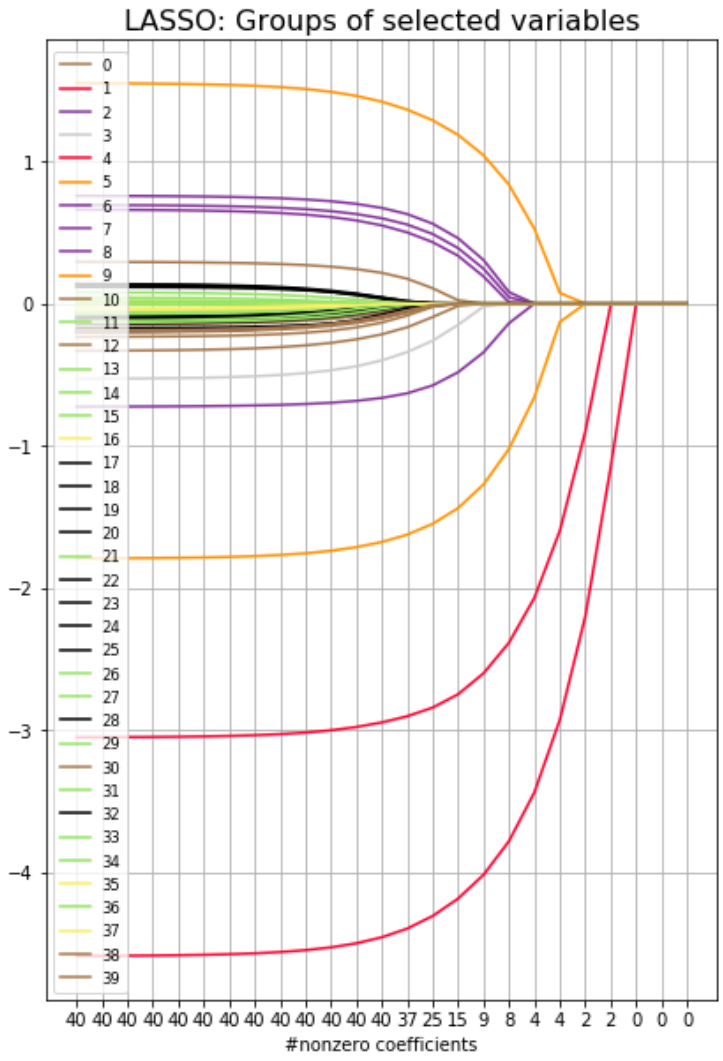
In [22]:

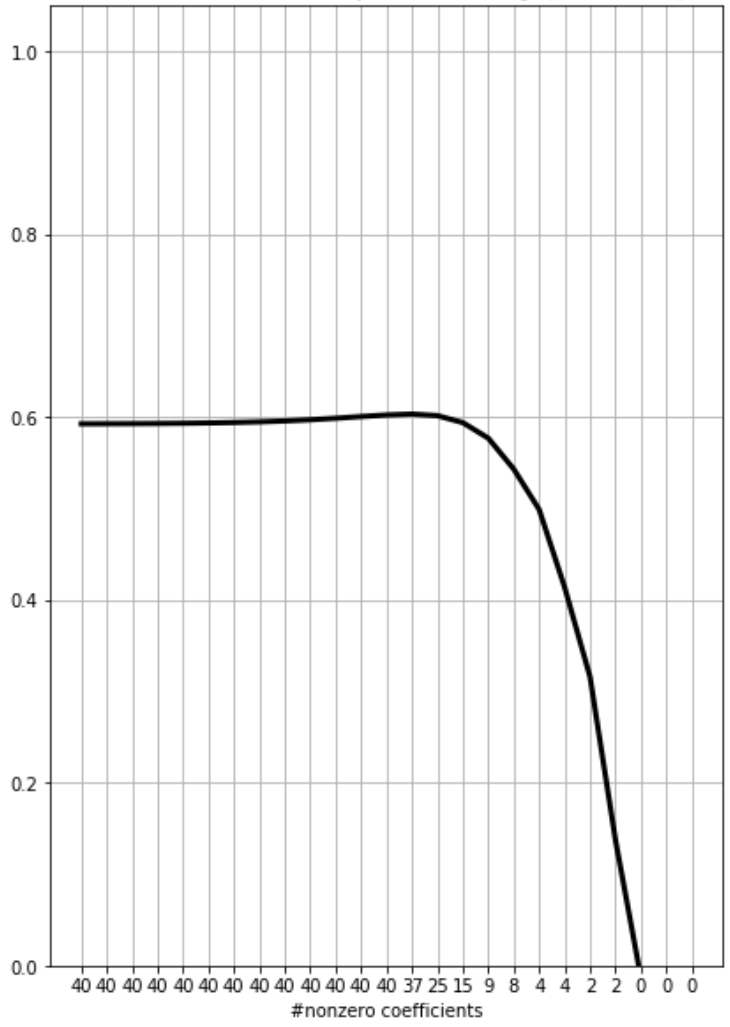
```
comment = "exp / dataset: 10/40 relevant variables, including 5 pathological one
s, linear ground truth, some noise only monotone transformation"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

X_exp = X.copy()
signs = np.sign(X)
X_exp[:, 0:6] = signs[:, 0:6] * np.exp(X_exp[:, 0:6])  # introduce pathological
transformation, NOT captured by ground-truth model
y = (true_coefs * X_exp).sum(axis=1) + epsilon
C_grid = np.logspace(-3, 1, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

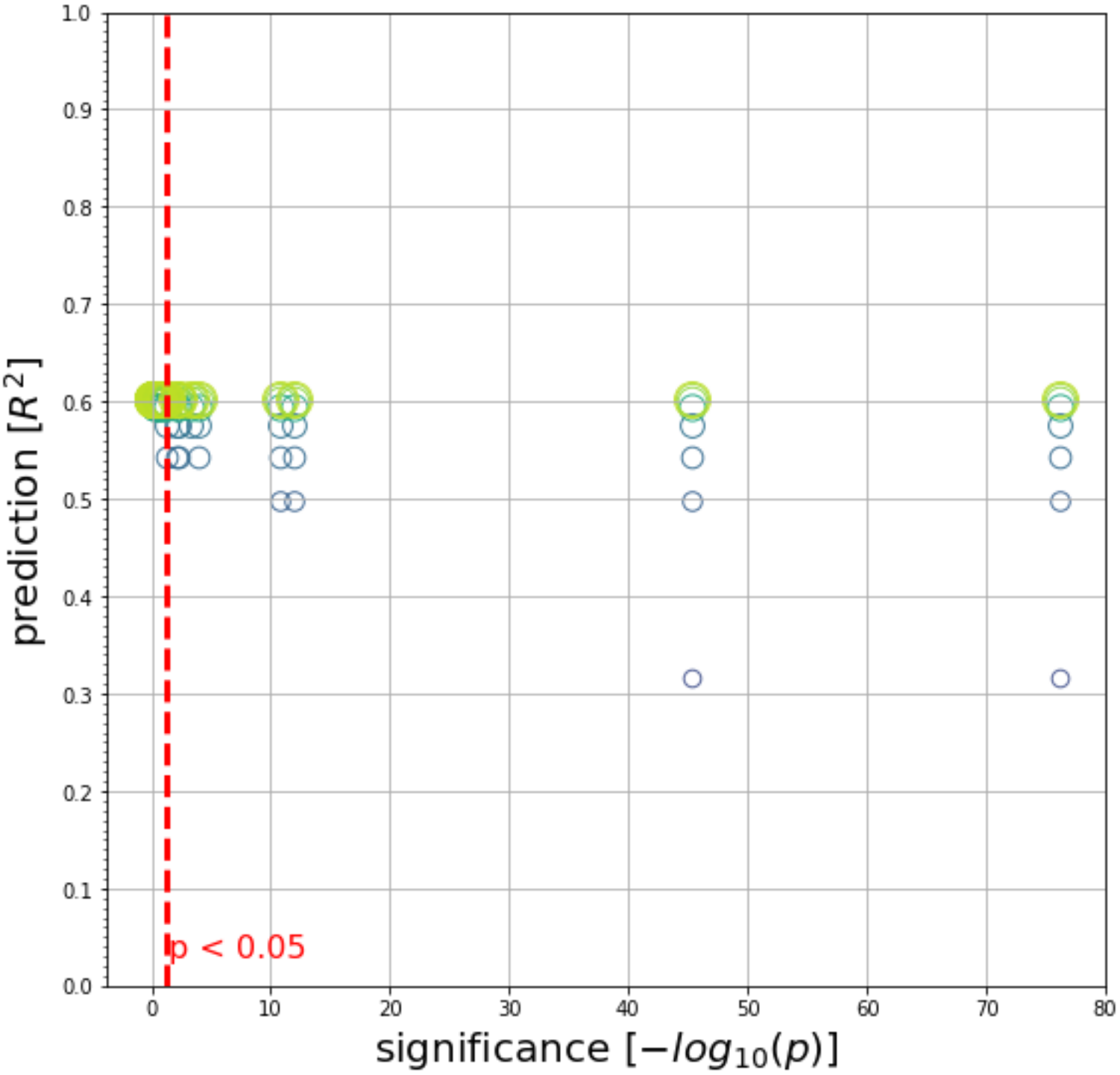
```
alpha: 0.0010 acc: 0.70 active_coefs: 40
alpha: 0.0015 acc: 0.70 active_coefs: 40
alpha: 0.0022 acc: 0.70 active_coefs: 40
alpha: 0.0032 acc: 0.70 active_coefs: 40
alpha: 0.0046 acc: 0.70 active_coefs: 40
alpha: 0.0068 acc: 0.70 active_coefs: 40
alpha: 0.0100 acc: 0.70 active_coefs: 40
alpha: 0.0147 acc: 0.70 active_coefs: 40
alpha: 0.0215 acc: 0.70 active_coefs: 40
alpha: 0.0316 acc: 0.70 active_coefs: 40
alpha: 0.0464 acc: 0.71 active_coefs: 40
alpha: 0.0681 acc: 0.71 active_coefs: 40
alpha: 0.1000 acc: 0.71 active_coefs: 40
alpha: 0.1468 acc: 0.71 active_coefs: 37
alpha: 0.2154 acc: 0.72 active_coefs: 25
alpha: 0.3162 acc: 0.72 active_coefs: 15
alpha: 0.4642 acc: 0.71 active_coefs: 9
alpha: 0.6813 acc: 0.67 active_coefs: 8
alpha: 1.0000 acc: 0.62 active_coefs: 4
alpha: 1.4678 acc: 0.51 active_coefs: 4
alpha: 2.1544 acc: 0.39 active_coefs: 2
alpha: 3.1623 acc: 0.17 active_coefs: 2
alpha: 4.6416 acc: -0.00 active_coefs: 0
alpha: 6.8129 acc: -0.00 active_coefs: 0
alpha: 10.0000 acc: -0.00 active_coefs: 0
40
76.2143531819
45.3613686825
11.8860592524
10.7770231965
```

```
3.92670584568
3.31257575664
2.29754305425
2.15258273589
1.30573326531
1.06197768944
1.00006512521
0.852205601628
0.718481807765
0.660739203119
0.612899107741
0.583045619179
0.507725234888
0.436234962749
0.435416377275
0.374203408051
0.365593619403
0.362775096156
0.36079077124
0.262820523178
0.261969847886
0.254061694889
0.229051548961
0.206583221557
0.198821191253
0.179707832984
0.155608965768
0.126833265463
0.123219303399
0.102081170546
0.101839208832
0.0803103243867
0.0585020869233
0.0523148663045
0.04916679132
0.0323112879968
skipping 3
skipping 4
skipping 5
skipping 6
skipping 7
skipping 8
skipping 10
skipping 11
skipping 12
skipping 13
skipping 14
skipping 15
skipping 19
skipping 21
skipping 23
skipping 24
```


Linear regression
true betas
estimated betas
significant at p<0.05
input variables


LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients

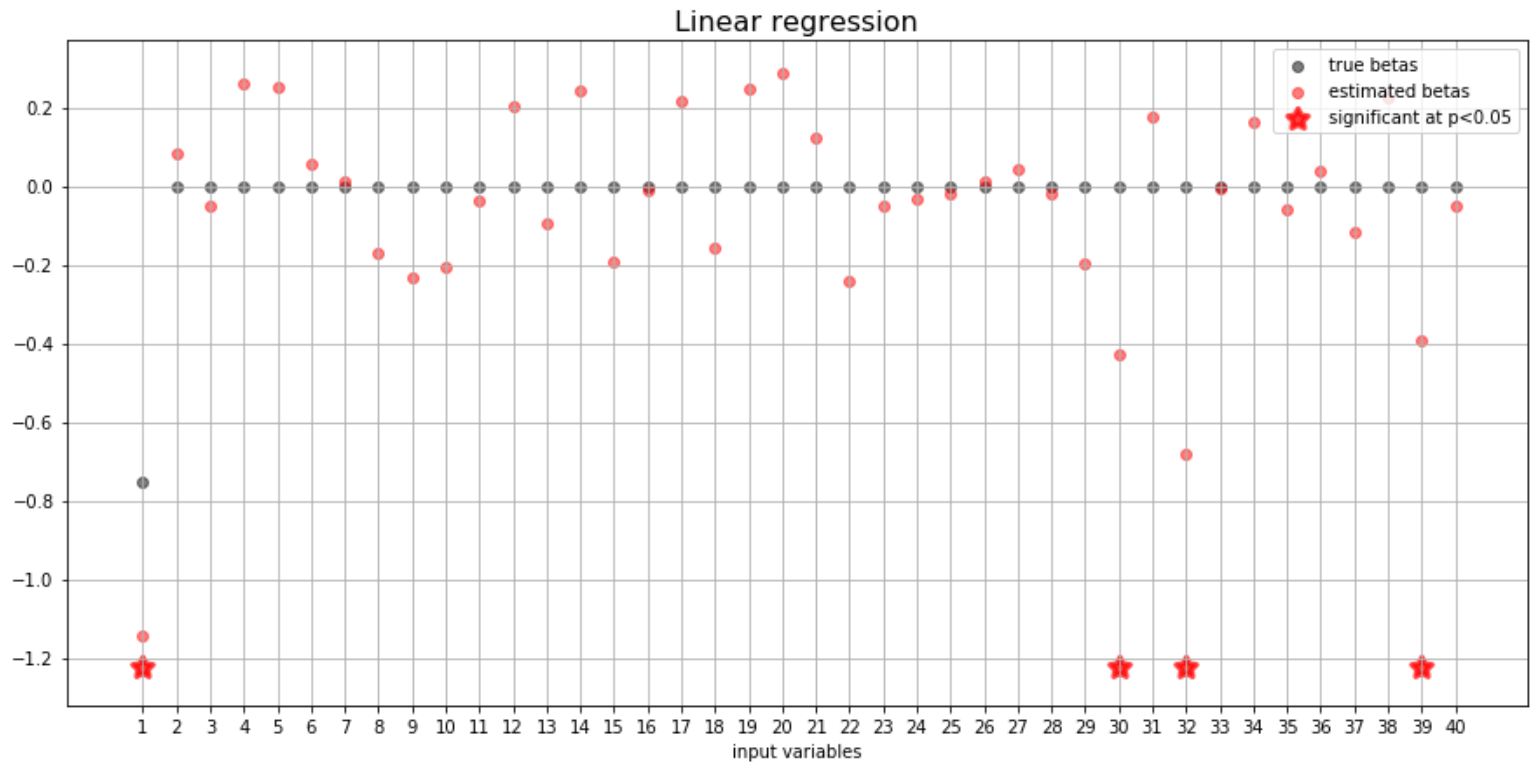


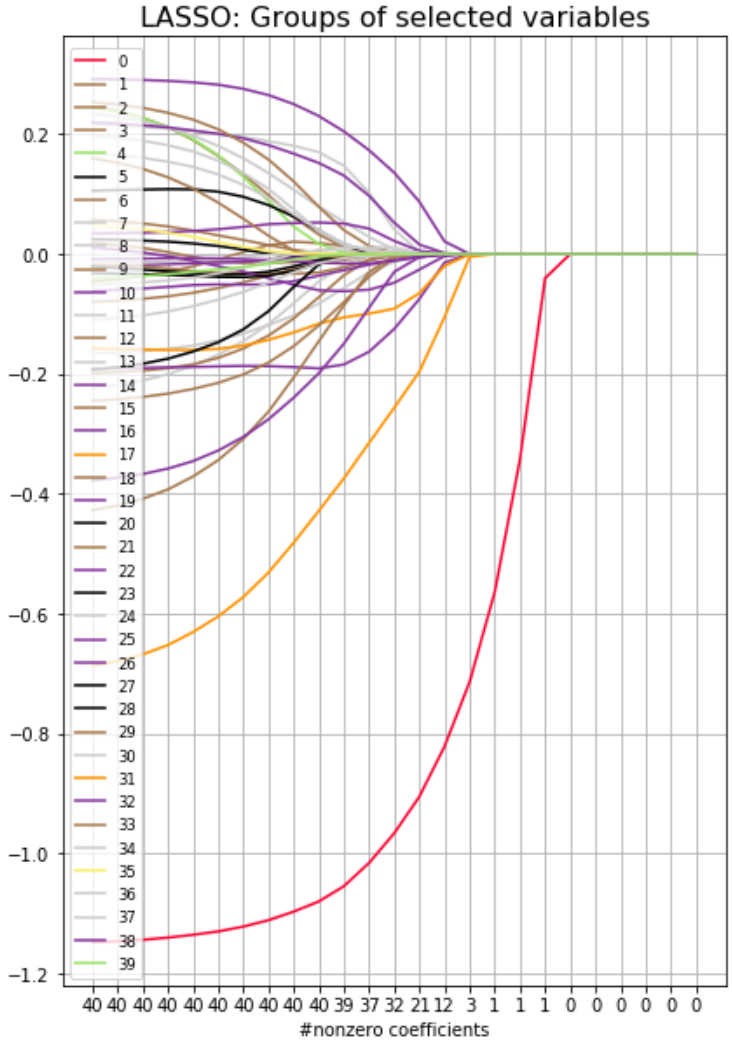
In [23]:

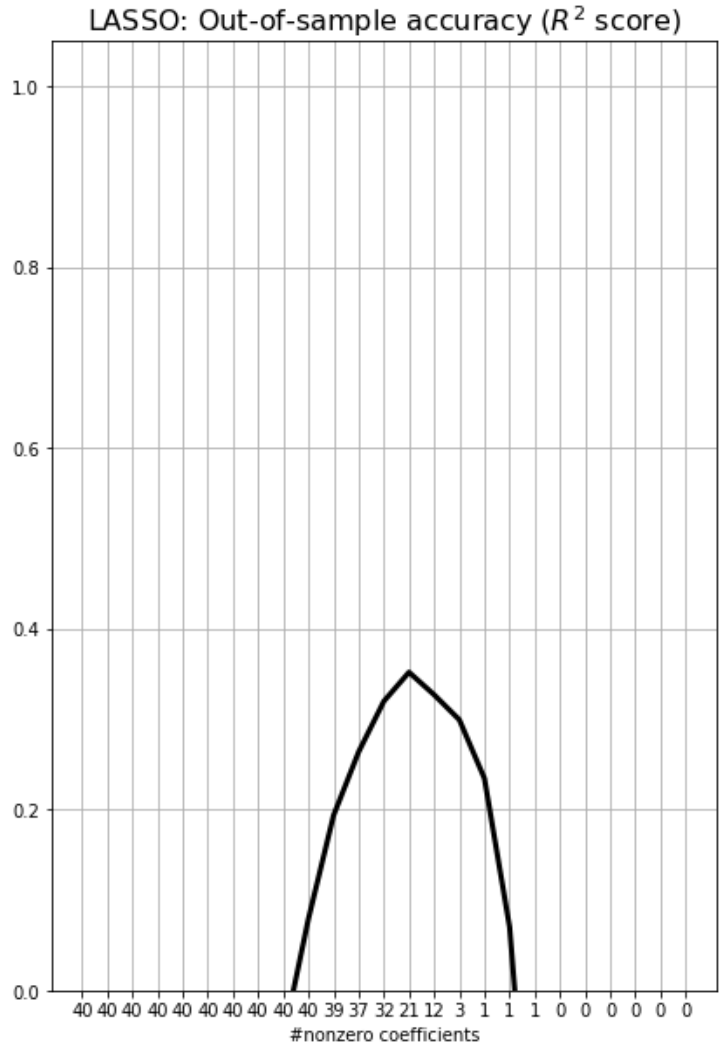
```
comment = "dataset: 1/40 relevant variables linear ground truth + some noise"
rs = np.random.RandomState(1)
n_samples = 60
n_feat = 40
n_feat_relevant = 1
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

X_exp = X.copy()
signs = np.sign(X)
y = (true_coefs * X).sum(axis=1) + epsilon * 1
C_grid = np.logspace(-3, 1, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

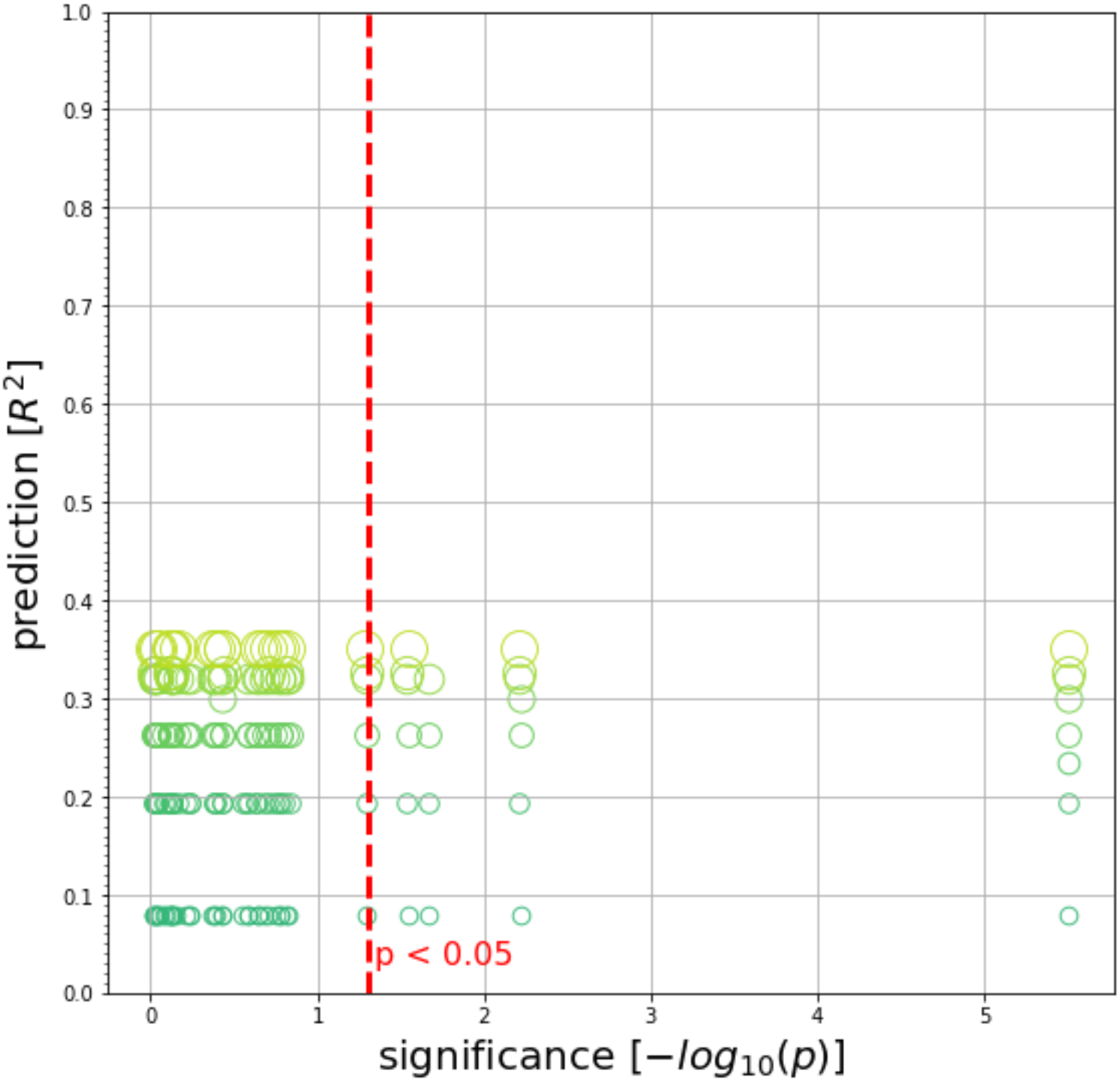
```
alpha: 0.0010 acc: -1.25 active_coefs: 40
alpha: 0.0015 acc: -1.06 active_coefs: 40
alpha: 0.0022 acc: -0.80 active_coefs: 40
alpha: 0.0032 acc: -0.47 active_coefs: 40
alpha: 0.0046 acc: -0.07 active_coefs: 40
alpha: 0.0068 acc: 0.24 active_coefs: 40
alpha: 0.0100 acc: 0.37 active_coefs: 40
alpha: 0.0147 acc: 0.47 active_coefs: 40
alpha: 0.0215 acc: 0.51 active_coefs: 40
alpha: 0.0316 acc: 0.52 active_coefs: 40
alpha: 0.0464 acc: 0.46 active_coefs: 39
alpha: 0.0681 acc: 0.45 active_coefs: 37
alpha: 0.1000 acc: 0.49 active_coefs: 32
alpha: 0.1468 acc: 0.48 active_coefs: 21
alpha: 0.2154 acc: 0.46 active_coefs: 12
alpha: 0.3162 acc: 0.43 active_coefs: 3
alpha: 0.4642 acc: 0.35 active_coefs: 1
alpha: 0.6813 acc: 0.23 active_coefs: 1
alpha: 1.0000 acc: 0.02 active_coefs: 1
alpha: 1.4678 acc: -0.02 active_coefs: 0
alpha: 2.1544 acc: -0.02 active_coefs: 0
alpha: 3.1623 acc: -0.02 active_coefs: 0
alpha: 4.6416 acc: -0.02 active_coefs: 0
alpha: 6.8129 acc: -0.02 active_coefs: 0
alpha: 10.0000 acc: -0.02 active_coefs: 0
40
5.49843999918
2.20862842615
1.6627727626
1.53716686016
1.28478081045
0.828063403002
0.803738383855
0.770219548481
0.767916261689
0.740917805793
0.702491287288
0.668384117993
0.631945268853
0.630014593113
0.57585460771
0.57562708848
0.550387508027
0.426464076979
0.423766813883
0.383521053994
0.374288322857
0.367539894884
0.226933568061
0.221980057228
0.207508177611
0.155989296473
0.119922628961
0.119745646911
```

```
0.116770699723
0.113798549995
0.111410358147
0.0939433931034
0.0732957588312
0.0555086334971
0.040113224744
0.0392843730584
0.022928790449
0.0228765856354
0.0153164947934
0.0139348330708
skipping 8
skipping 9
skipping 10
skipping 11
skipping 13
skipping 14
skipping 15
skipping 16
skipping 17
skipping 18
skipping 19
skipping 20
skipping 21
skipping 22
skipping 23
skipping 24
```




LASSO: Groups of selected variables
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−1.2
40 40 40 40 40 40 40 40 40 40 39 37 32 21 12 3 1 1 1 0 0 0 0 0 0
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
1.0
0.8
0.6
0.4
0.2
0.0
40 40 40 40 40 40 40 40 40 40 39 37 32 21 12 3 1 1 1 0 0 0 0 0 0
#nonzero coefficients

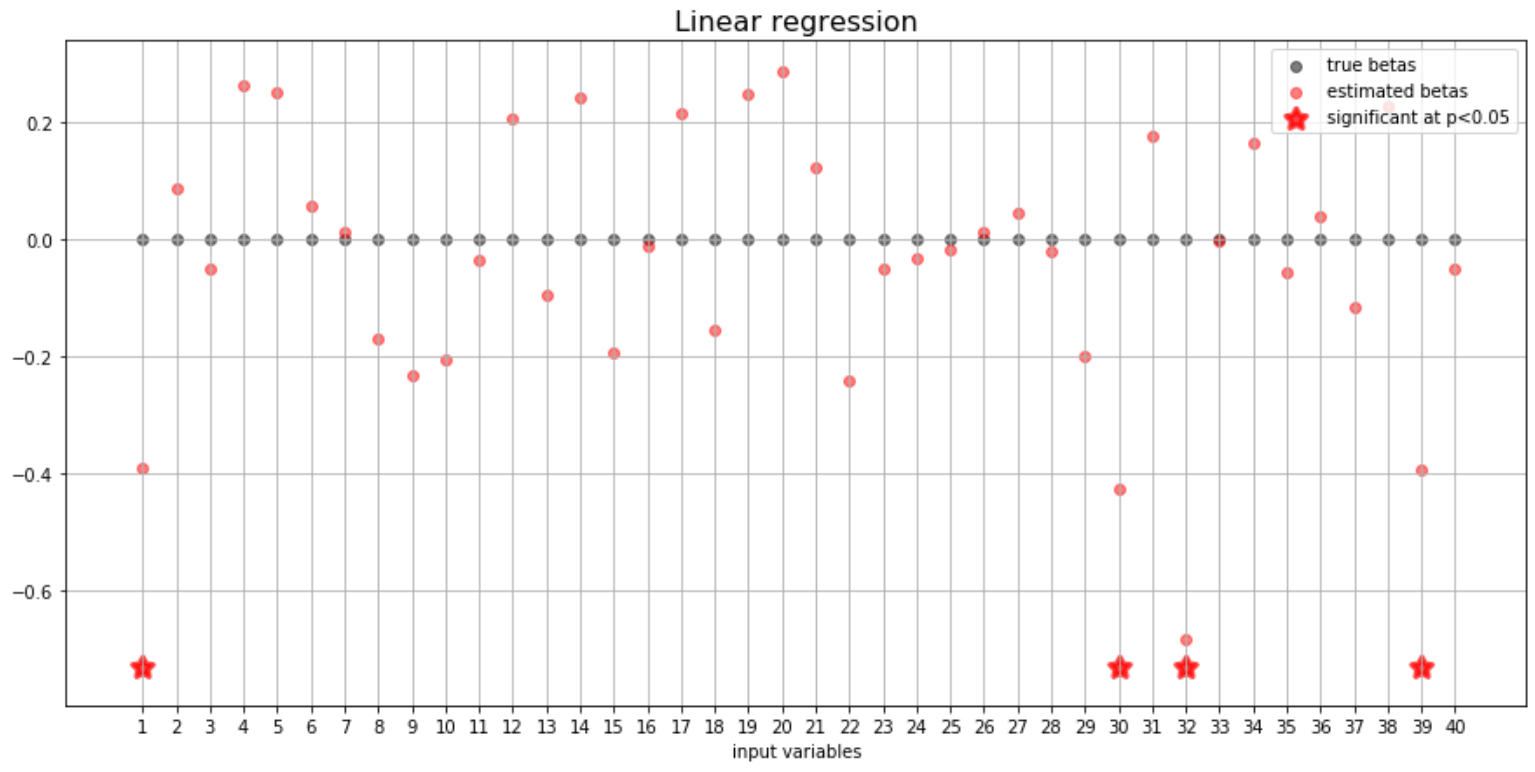


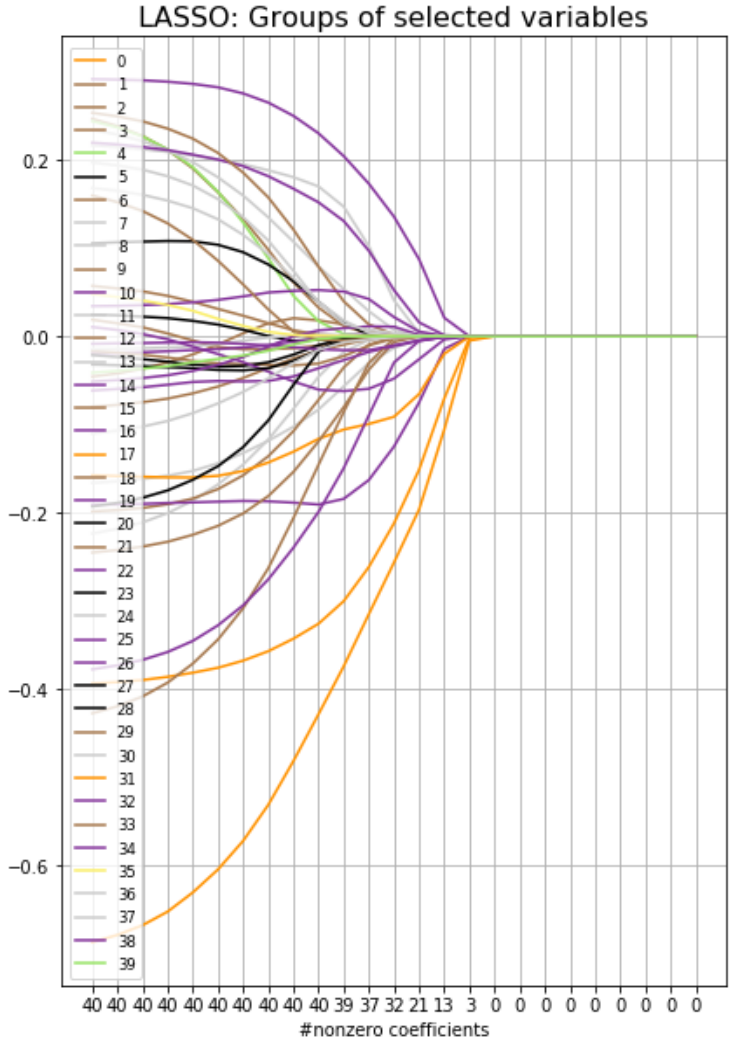
In [24]:

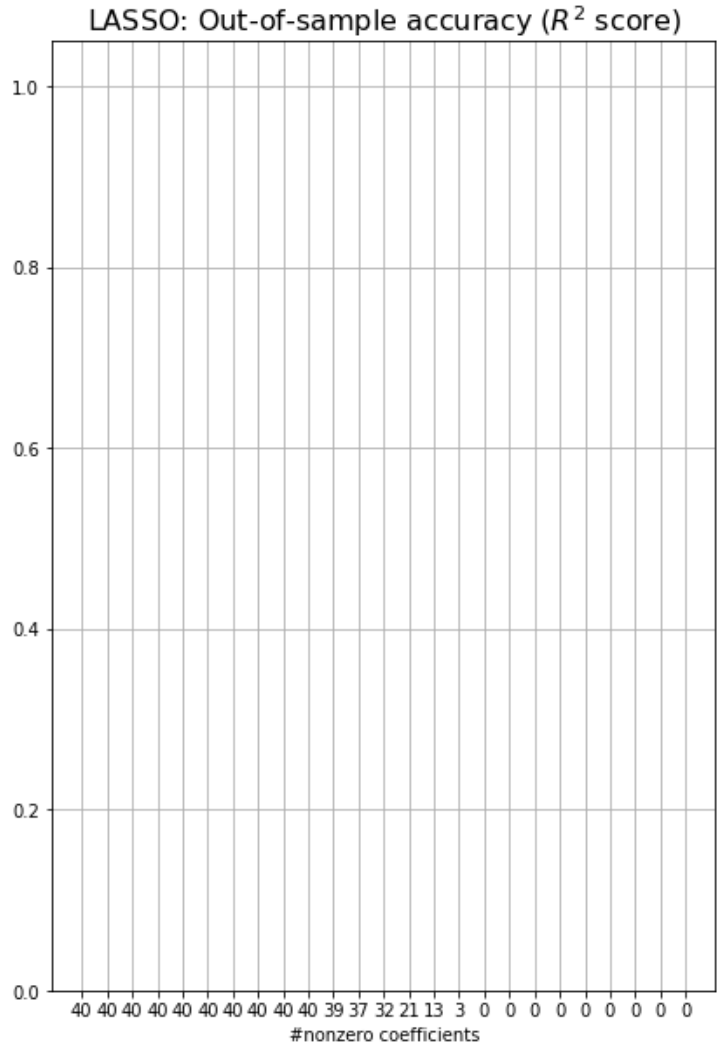
```
comment = "dataset: 0/40 relevant variables linear ground truth + some noise"
rs = np.random.RandomState(1)
n_samples = 60
n_feat = 40
n_feat_relevant = 0
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

X_exp = X.copy()
signs = np.sign(X)
y = (true_coefs * X).sum(axis=1) + epsilon * 1
C_grid = np.logspace(-3, 1, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

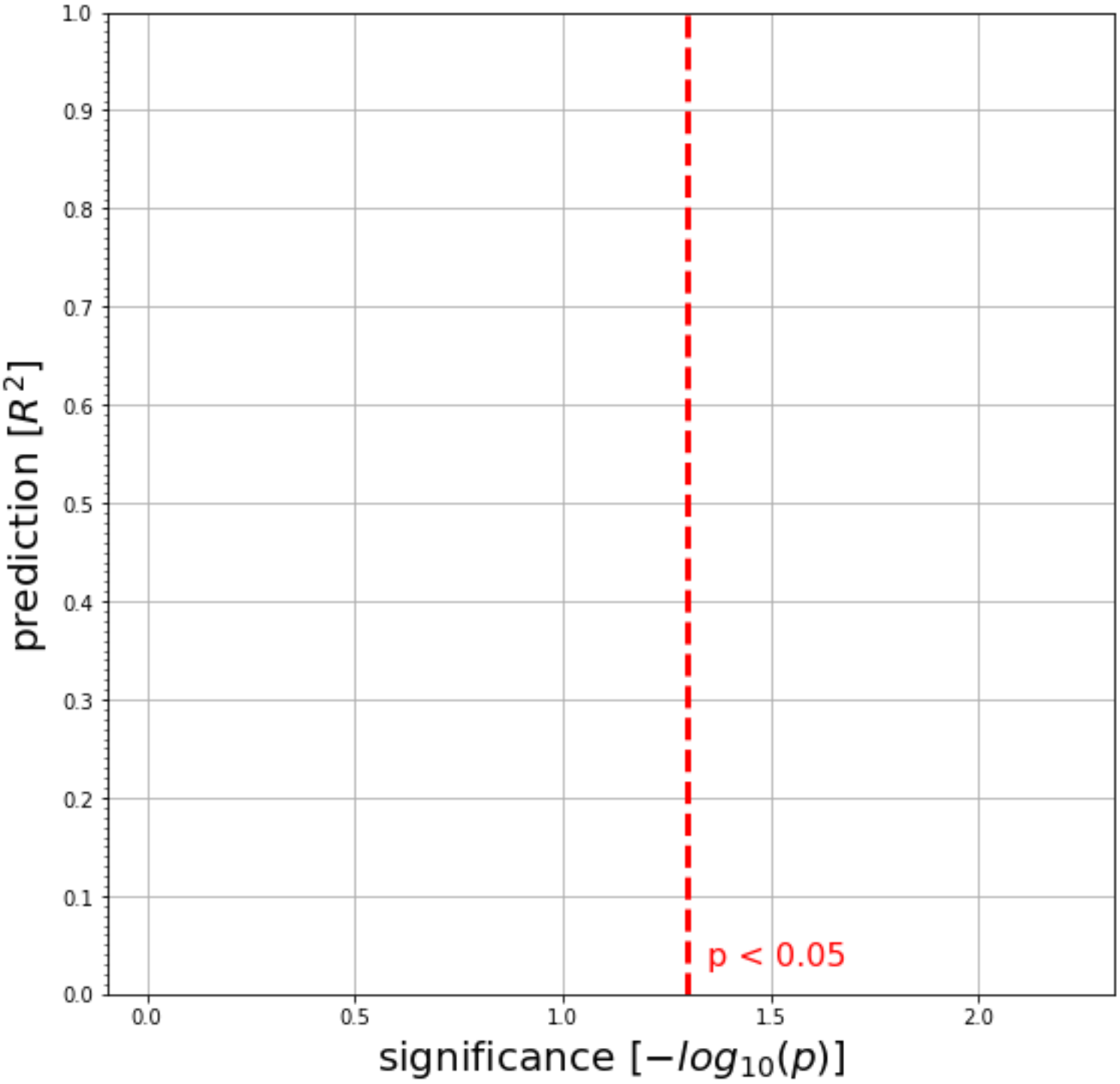
```
alpha: 0.0010 acc: -3.11 active_coefs: 40
alpha: 0.0015 acc: -2.76 active_coefs: 40
alpha: 0.0022 acc: -2.29 active_coefs: 40
alpha: 0.0032 acc: -1.68 active_coefs: 40
alpha: 0.0046 acc: -0.95 active_coefs: 40
alpha: 0.0068 acc: -0.39 active_coefs: 40
alpha: 0.0100 acc: -0.16 active_coefs: 40
alpha: 0.0147 acc: 0.03 active_coefs: 40
alpha: 0.0215 acc: 0.11 active_coefs: 40
alpha: 0.0316 acc: 0.12 active_coefs: 40
alpha: 0.0464 acc: 0.01 active_coefs: 39
alpha: 0.0681 acc: -0.01 active_coefs: 37
alpha: 0.1000 acc: 0.06 active_coefs: 32
alpha: 0.1468 acc: 0.06 active_coefs: 21
alpha: 0.2154 acc: 0.02 active_coefs: 13
alpha: 0.3162 acc: -0.00 active_coefs: 3
alpha: 0.4642 acc: -0.00 active_coefs: 0
alpha: 0.6813 acc: -0.00 active_coefs: 0
alpha: 1.0000 acc: -0.00 active_coefs: 0
alpha: 1.4678 acc: -0.00 active_coefs: 0
alpha: 2.1544 acc: -0.00 active_coefs: 0
alpha: 3.1623 acc: -0.00 active_coefs: 0
alpha: 4.6416 acc: -0.00 active_coefs: 0
alpha: 6.8129 acc: -0.00 active_coefs: 0
alpha: 10.0000 acc: -0.00 active_coefs: 0
40
2.20862842615
1.6627727626
1.53716686016
1.37947545309
1.28478081045
0.828063403002
0.803738383855
0.770219548481
0.767916261689
0.740917805793
0.702491287288
0.668384117993
0.631945268853
0.630014593113
0.57585460771
0.57562708848
0.550387508027
0.426464076979
0.423766813883
0.383521053994
0.374288322857
0.367539894884
0.226933568061
0.221980057228
0.207508177611
0.155989296473
0.119922628961
0.119745646911
```

```
0.116770699723
0.113798549995
0.111410358147
0.0939433931034
0.0732957588312
0.0555086334971
0.040113224744
0.0392843730584
0.022928790449
0.0228765856354
0.0153164947934
0.0139348330708
skipping 4
skipping 5
skipping 6
skipping 7
skipping 8
skipping 9
skipping 10
skipping 11
skipping 16
skipping 17
skipping 18
skipping 19
skipping 20
skipping 21
skipping 22
skipping 23
skipping 24
```




LASSO: Groups of selected variables
0.2
0.0
−0.2
−0.4
−0.6
40 40 40 40 40 40 40 40 40 40 39 37 32 21 13 3 0 0 0 0 0 0 0 0
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
1.0
0.8
0.6
0.4
0.2
0.0
40 40 40 40 40 40 40 40 40 40 39 37 32 21 13 3 0 0 0 0 0 0 0 0
#nonzero coefficients



observation: 4/5 path. variables are significant, the ?/5 path. variables are selected as predictive

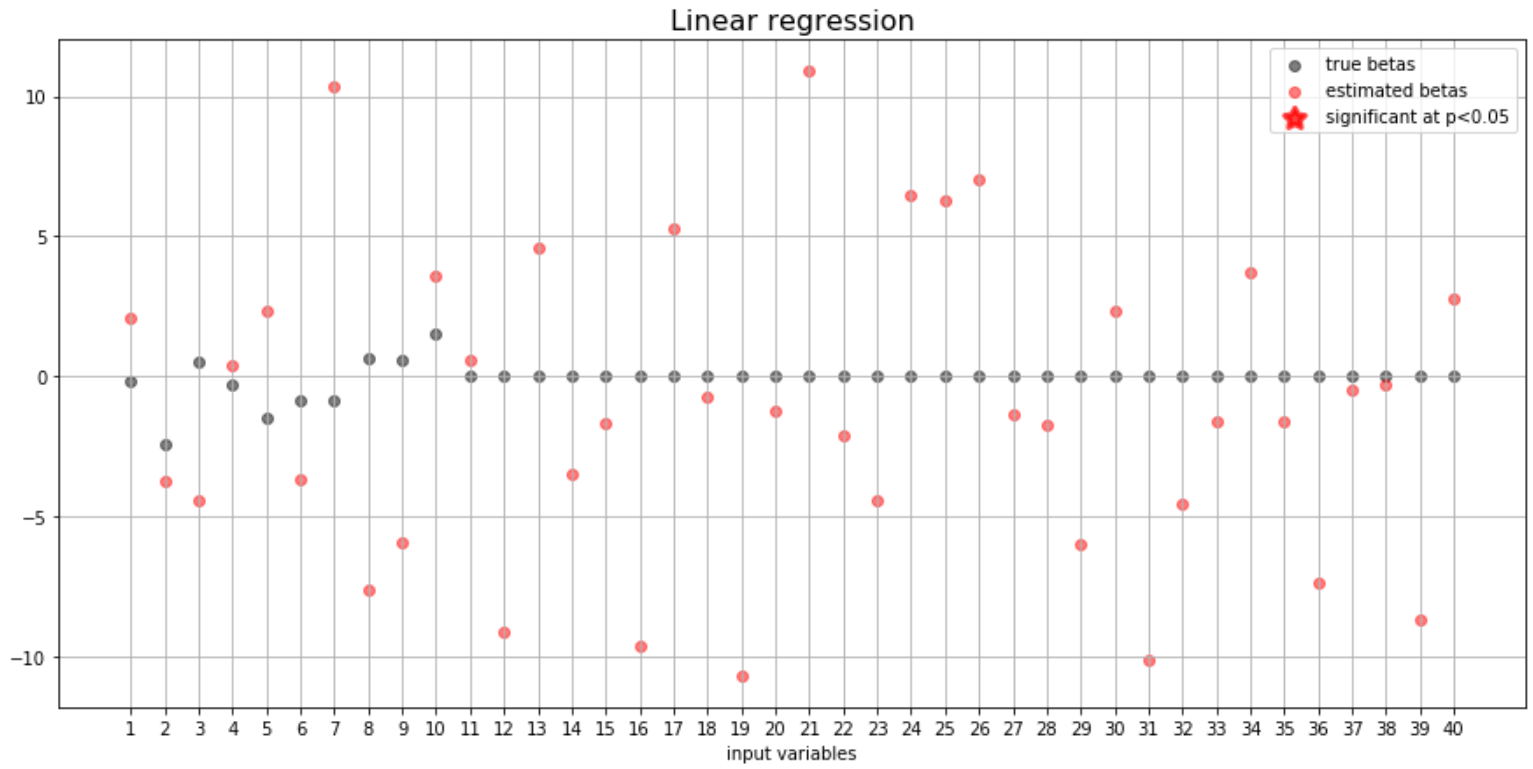
In [25]:

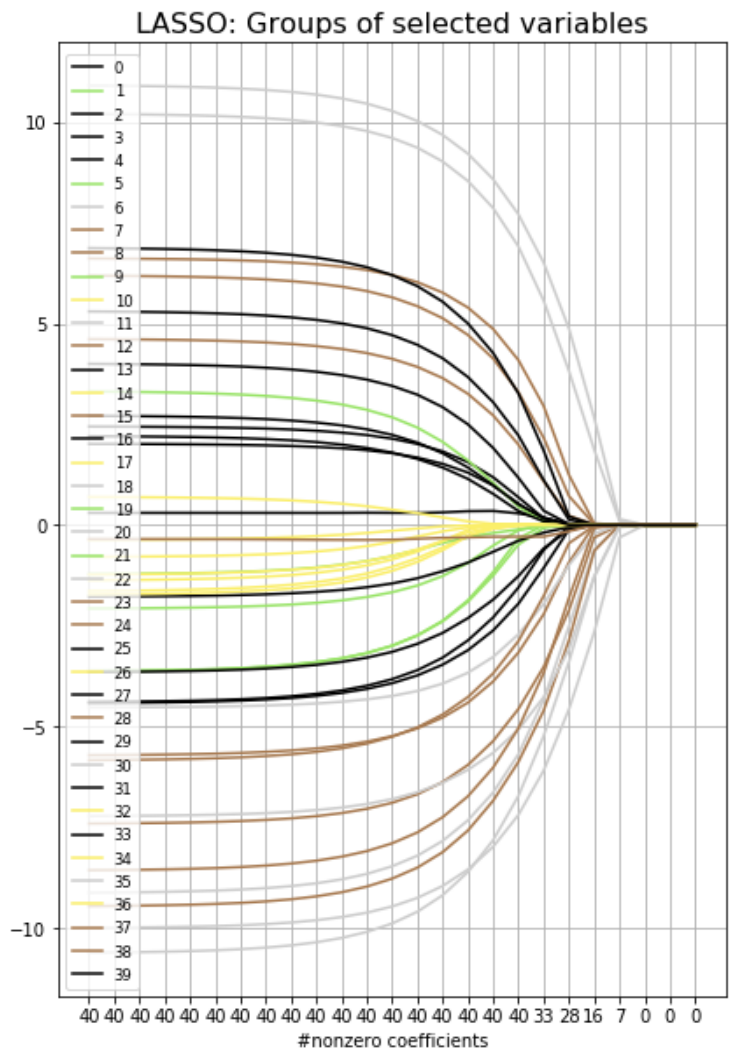
```
comment = "1/x / dataset: 10/40 relevant variables, including 5 pathological one
s, linear ground truth, some noise opposite effect of log roughly; small values
become gigantic"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

X_inv = X.copy()
X_inv[:, 0:6] = 1. / X_inv[:, 0:6]  # introduce pathological transformation, NOT
captured by ground-truth model
y = (true_coefs * X_inv).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 1.5, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

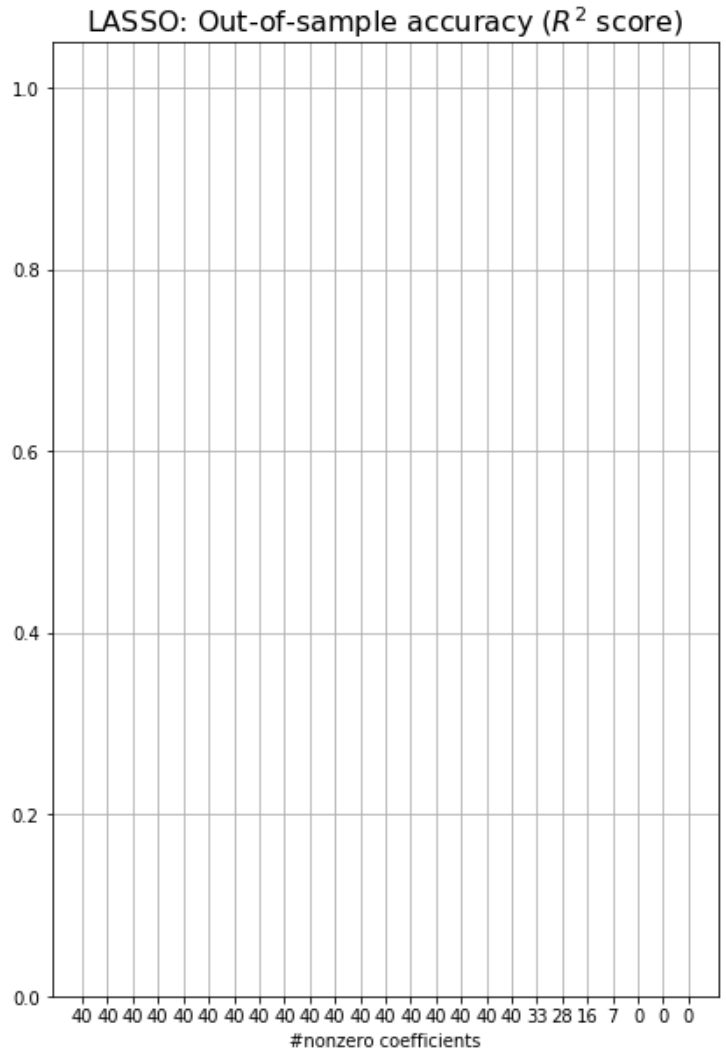
```
alpha: 0.0100 acc: -0.13 active_coefs: 40
alpha: 0.0140 acc: -0.13 active_coefs: 40
alpha: 0.0196 acc: -0.13 active_coefs: 40
alpha: 0.0274 acc: -0.13 active_coefs: 40
alpha: 0.0383 acc: -0.13 active_coefs: 40
alpha: 0.0536 acc: -0.13 active_coefs: 40
alpha: 0.0750 acc: -0.13 active_coefs: 40
alpha: 0.1049 acc: -0.13 active_coefs: 40
alpha: 0.1468 acc: -0.13 active_coefs: 40
alpha: 0.2054 acc: -0.13 active_coefs: 40
alpha: 0.2873 acc: -0.12 active_coefs: 40
alpha: 0.4019 acc: -0.12 active_coefs: 40
alpha: 0.5623 acc: -0.12 active_coefs: 40
alpha: 0.7867 acc: -0.11 active_coefs: 40
alpha: 1.1007 acc: -0.11 active_coefs: 40
alpha: 1.5399 acc: -0.10 active_coefs: 40
alpha: 2.1544 acc: -0.09 active_coefs: 40
alpha: 3.0142 acc: -0.08 active_coefs: 40
alpha: 4.2170 acc: -0.06 active_coefs: 33
alpha: 5.8997 acc: -0.05 active_coefs: 28
alpha: 8.2540 acc: -0.04 active_coefs: 16
alpha: 11.5478 acc: -0.02 active_coefs: 7
alpha: 16.1560 acc: -0.02 active_coefs: 0
alpha: 22.6030 acc: -0.02 active_coefs: 0
alpha: 31.6228 acc: -0.02 active_coefs: 0
40
1.23328886799
1.1642438789
1.12861866505
1.11143662299
```

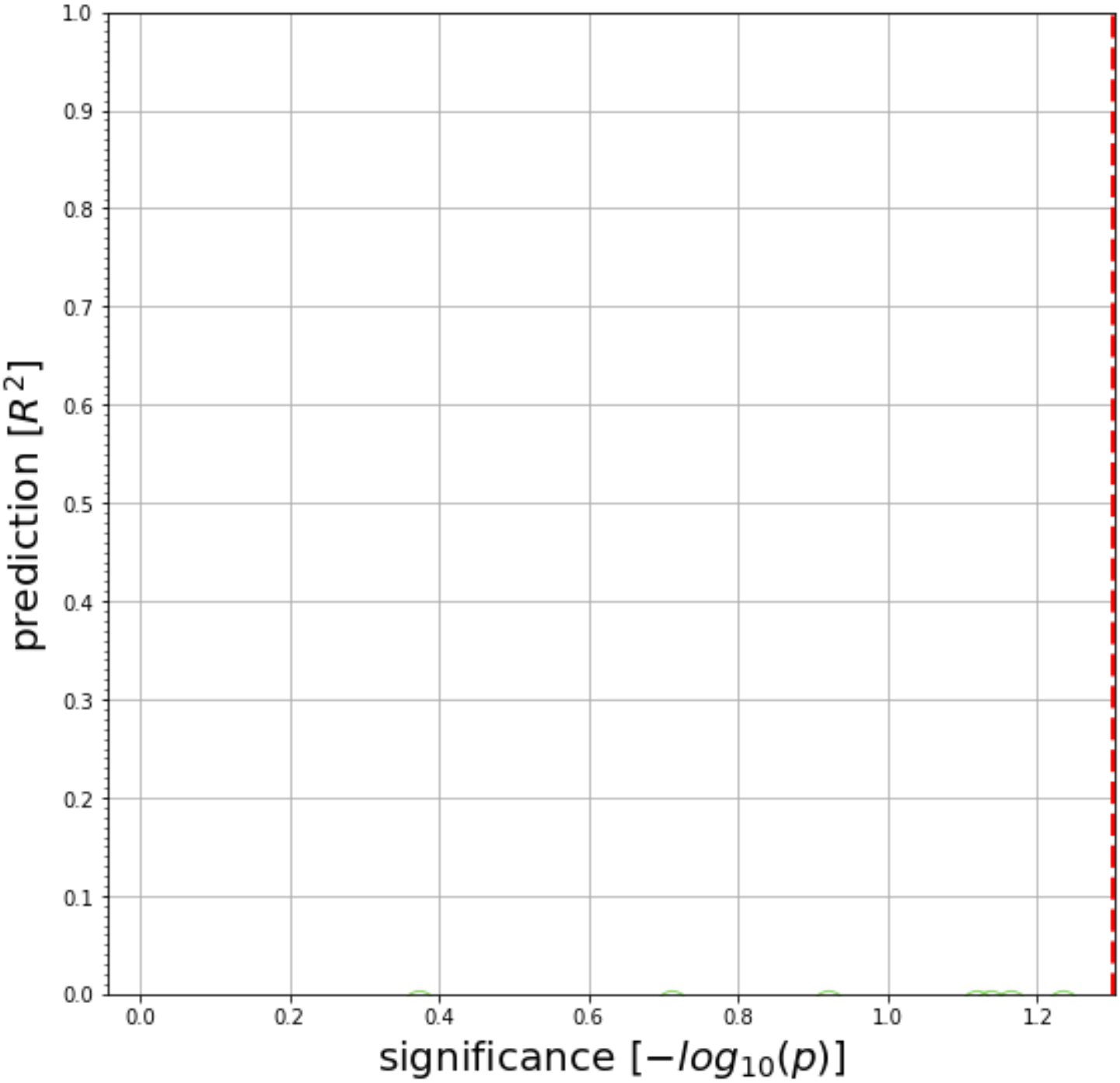
```
0.984953226432
0.919861684347
0.855082099196
0.708081111681
0.707125873183
0.653618213156
0.55025454591
0.540136715094
0.528939731724
0.523176775359
0.43870431882
0.37469548558
0.369528596131
0.367385028417
0.350506966043
0.289671628796
0.288371099717
0.281743191572
0.276873699717
0.259611725841
0.201319017302
0.170246564109
0.166991254372
0.149804006458
0.143716583718
0.118217638043
0.112994561554
0.11182729728
0.108555807666
0.0896232839141
0.0826176298745
0.0452607575379
0.0387188200396
0.0318337572718
0.0228396412336
0.0166646523702
skipping 1
skipping 2
skipping 8
skipping 9
skipping 10
skipping 11
skipping 12
skipping 13
skipping 14
skipping 15
skipping 16
skipping 17
skipping 18
skipping 19
skipping 20
skipping 21
skipping 22
```

```
skipping 23
skipping 24
```









observation: 0/5 path. variables are significant, the 0/5 path. variables are selected as predictive -> B: un peu extreme

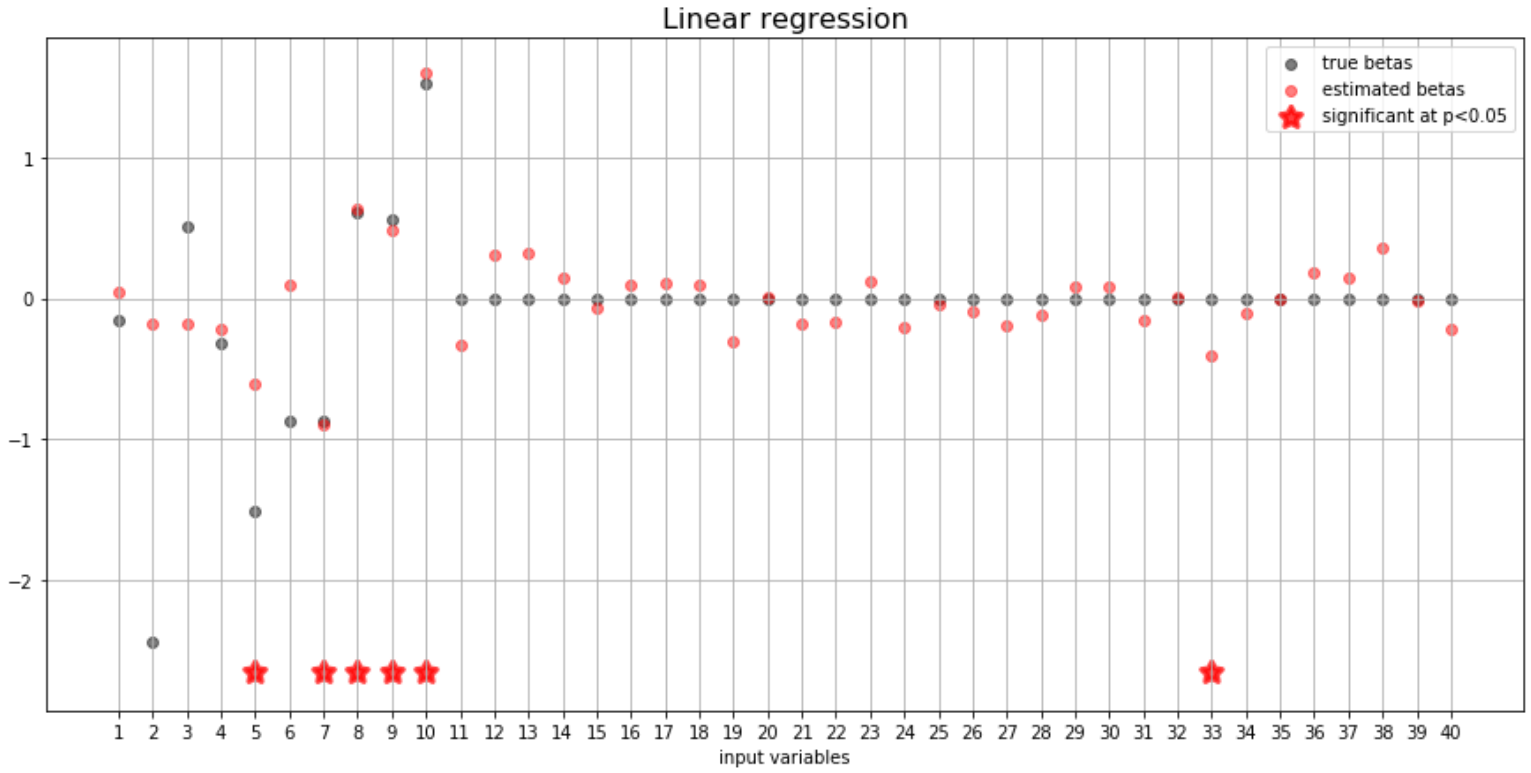
Polynomial transformations 1000 samples, 40 variables, error = N(0, 1)

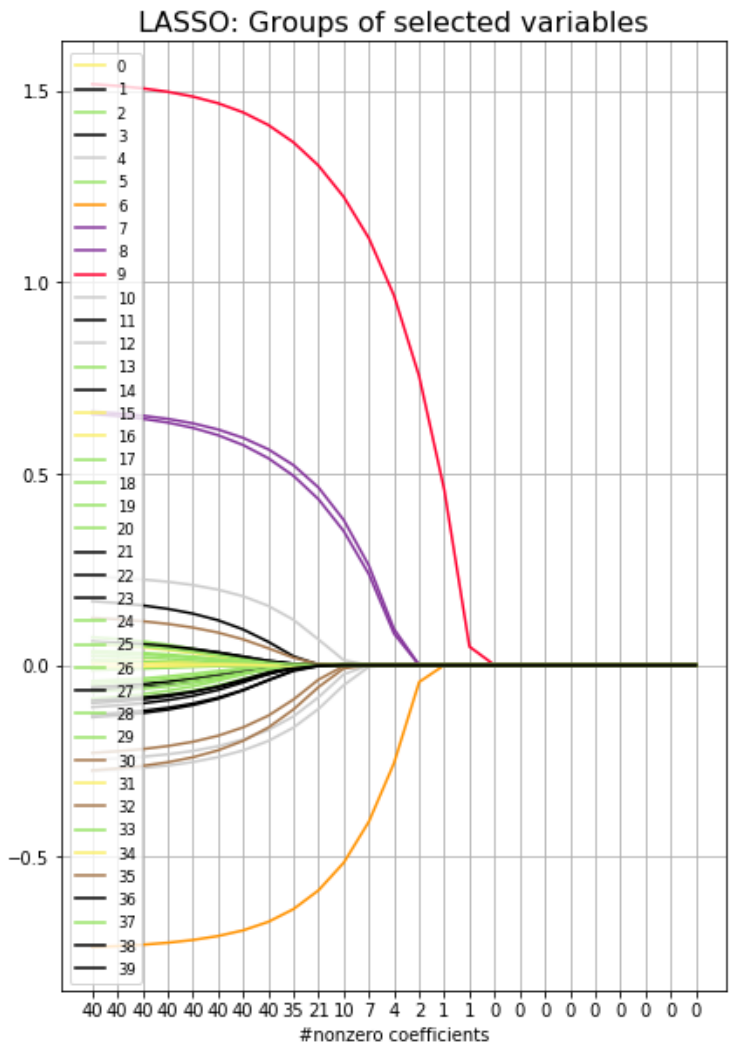
In [26]:

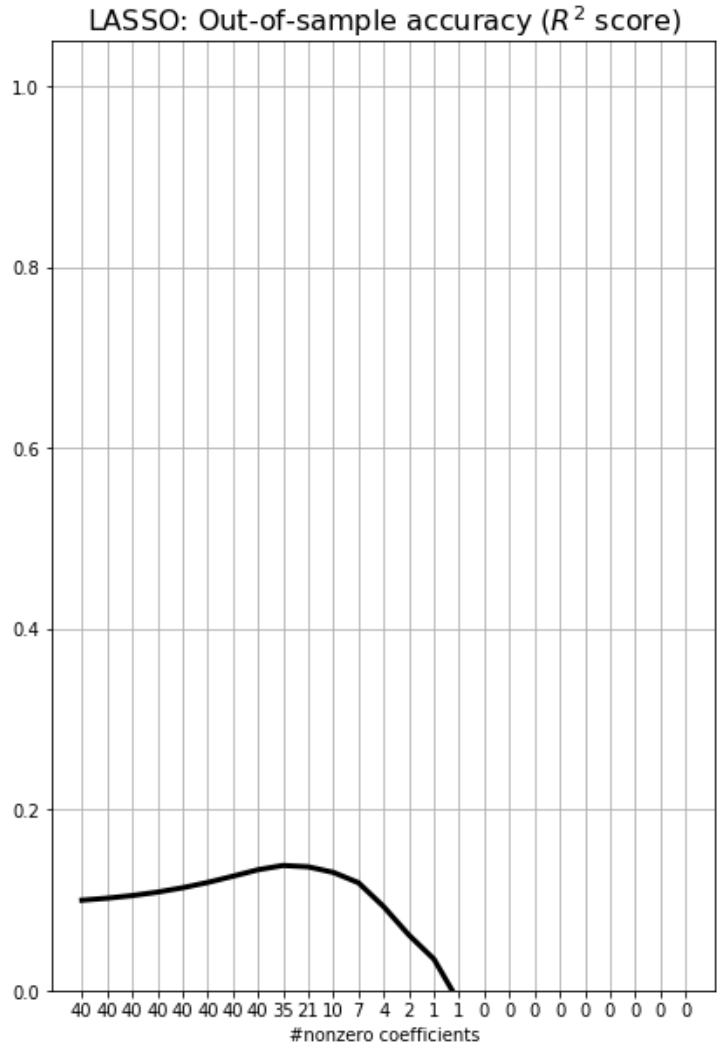
```
comment = "x^2 / dataset: 10/40 relevant variables, including 5 pathological one
s, linear ground truth, some noise cube -> keep monotony; otherwise not -> "
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

X_2 = X.copy()
X_2[:, 0:6] = np.square(X_2[:, 0:6])  # introduce pathological transformation, N
OT captured by ground-truth model
y = (true_coefs * X_2).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 1.5, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

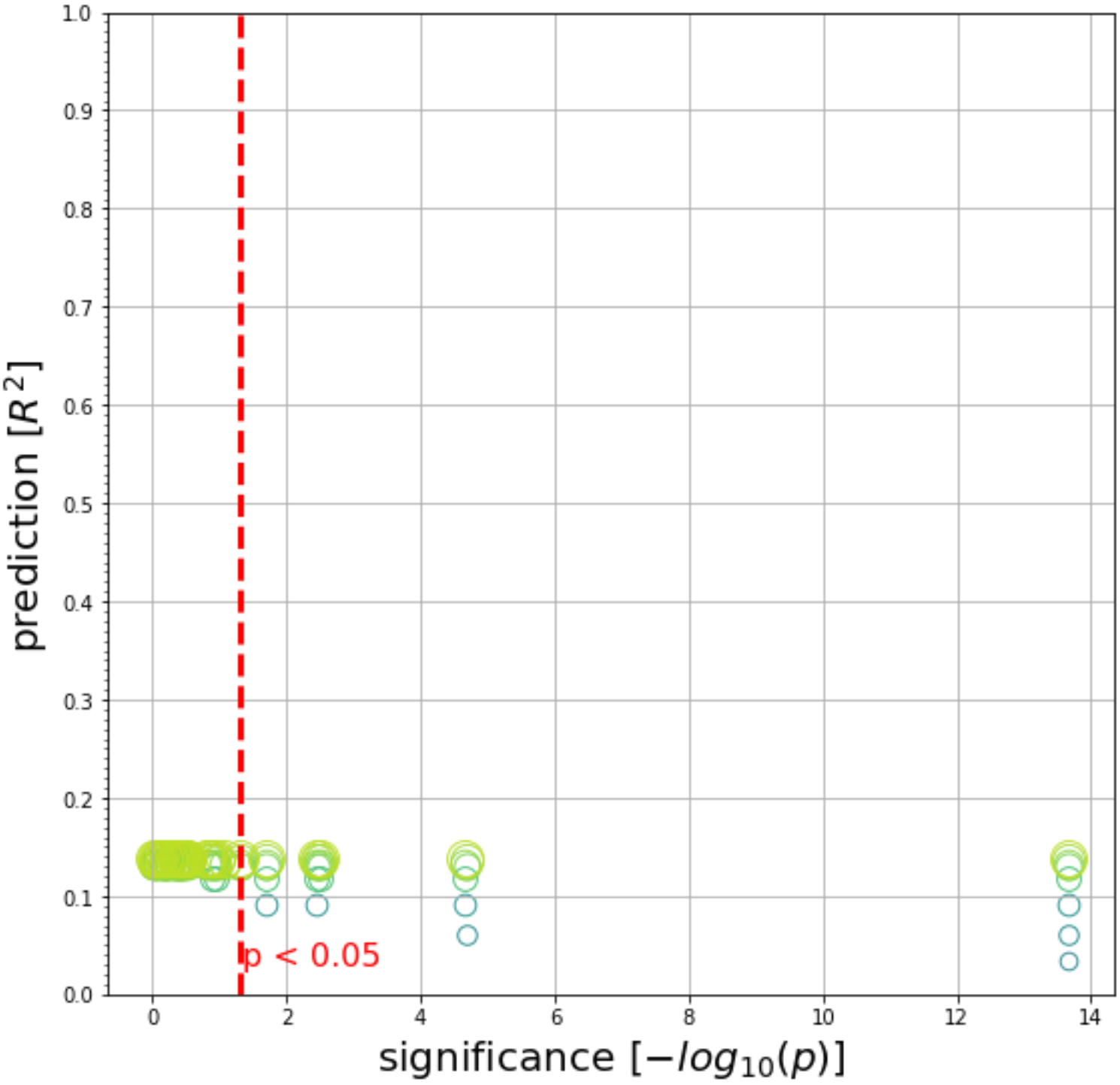
```
alpha: 0.0100 acc: 0.10 active_coefs: 40
alpha: 0.0140 acc: 0.10 active_coefs: 40
alpha: 0.0196 acc: 0.10 active_coefs: 40
alpha: 0.0274 acc: 0.11 active_coefs: 40
alpha: 0.0383 acc: 0.11 active_coefs: 40
alpha: 0.0536 acc: 0.12 active_coefs: 40
alpha: 0.0750 acc: 0.12 active_coefs: 40
alpha: 0.1049 acc: 0.13 active_coefs: 40
alpha: 0.1468 acc: 0.14 active_coefs: 35
alpha: 0.2054 acc: 0.14 active_coefs: 21
alpha: 0.2873 acc: 0.15 active_coefs: 10
alpha: 0.4019 acc: 0.15 active_coefs: 7
alpha: 0.5623 acc: 0.13 active_coefs: 4
alpha: 0.7867 acc: 0.10 active_coefs: 2
alpha: 1.1007 acc: 0.05 active_coefs: 1
alpha: 1.5399 acc: -0.03 active_coefs: 1
alpha: 2.1544 acc: -0.03 active_coefs: 0
alpha: 3.0142 acc: -0.03 active_coefs: 0
alpha: 4.2170 acc: -0.03 active_coefs: 0
alpha: 5.8997 acc: -0.03 active_coefs: 0
alpha: 8.2540 acc: -0.03 active_coefs: 0
alpha: 11.5478 acc: -0.03 active_coefs: 0
alpha: 16.1560 acc: -0.03 active_coefs: 0
alpha: 22.6030 acc: -0.03 active_coefs: 0
alpha: 31.6228 acc: -0.03 active_coefs: 0
40
13.659806738
4.66152493206
2.50215172889
2.43369759722
1.67610947252
```

```
1.30514390134
1.04378641578
0.953155031622
0.877840222358
0.808479659606
0.79008994489
0.531379374082
0.495837310808
0.452914579975
0.443459511339
0.419351702886
0.41834354119
0.410735633449
0.400939373898
0.363995922809
0.330936198292
0.330213317472
0.307688499341
0.253256677675
0.238763636337
0.220195455476
0.196212398691
0.184377967691
0.174320534055
0.172474515603
0.169895832654
0.166000066243
0.152902783997
0.114169548555
0.086096020092
0.0681634636715
0.0392594094376
0.0192412595491
0.01066194249
0.00472249626506
skipping 4
skipping 5
skipping 7
skipping 8
skipping 9
skipping 10
skipping 11
skipping 15
skipping 17
skipping 18
skipping 19
skipping 20
skipping 21
skipping 22
skipping 23
skipping 24
```


Linear regression
true betas
estimated betas
significant at p<0.05
input variables


LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients



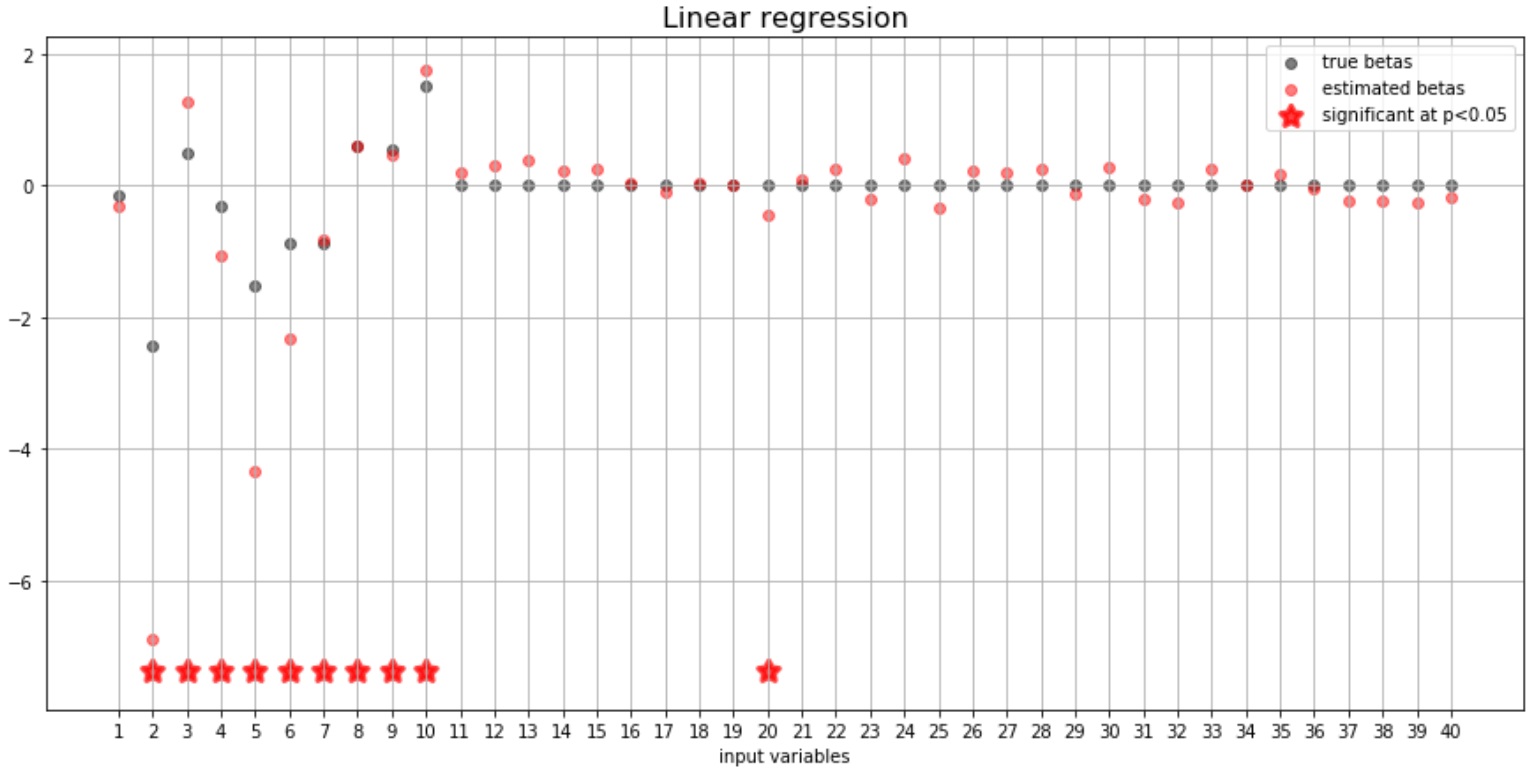
observation: 1/5 path. variables are significant, the same 1/5 path. variables are selected as predictive

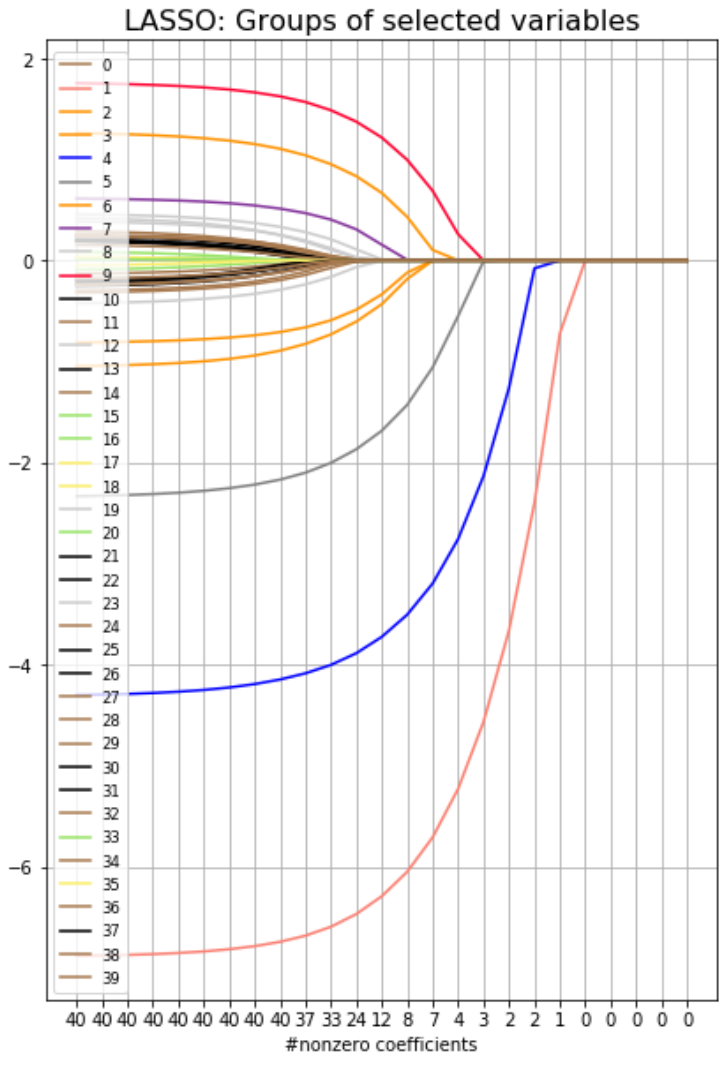
In [27]:

```
comment = "x^3 / dataset: 10/40 relevant variables, including 5 pathological one
s, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

X_3 = X.copy()
X_3[:, 0:6] = X_3[:, 0:6]**3  # introduce pathological transformation, NOT captu
red by ground-truth model
y = (true_coefs * X_3).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 1.5, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

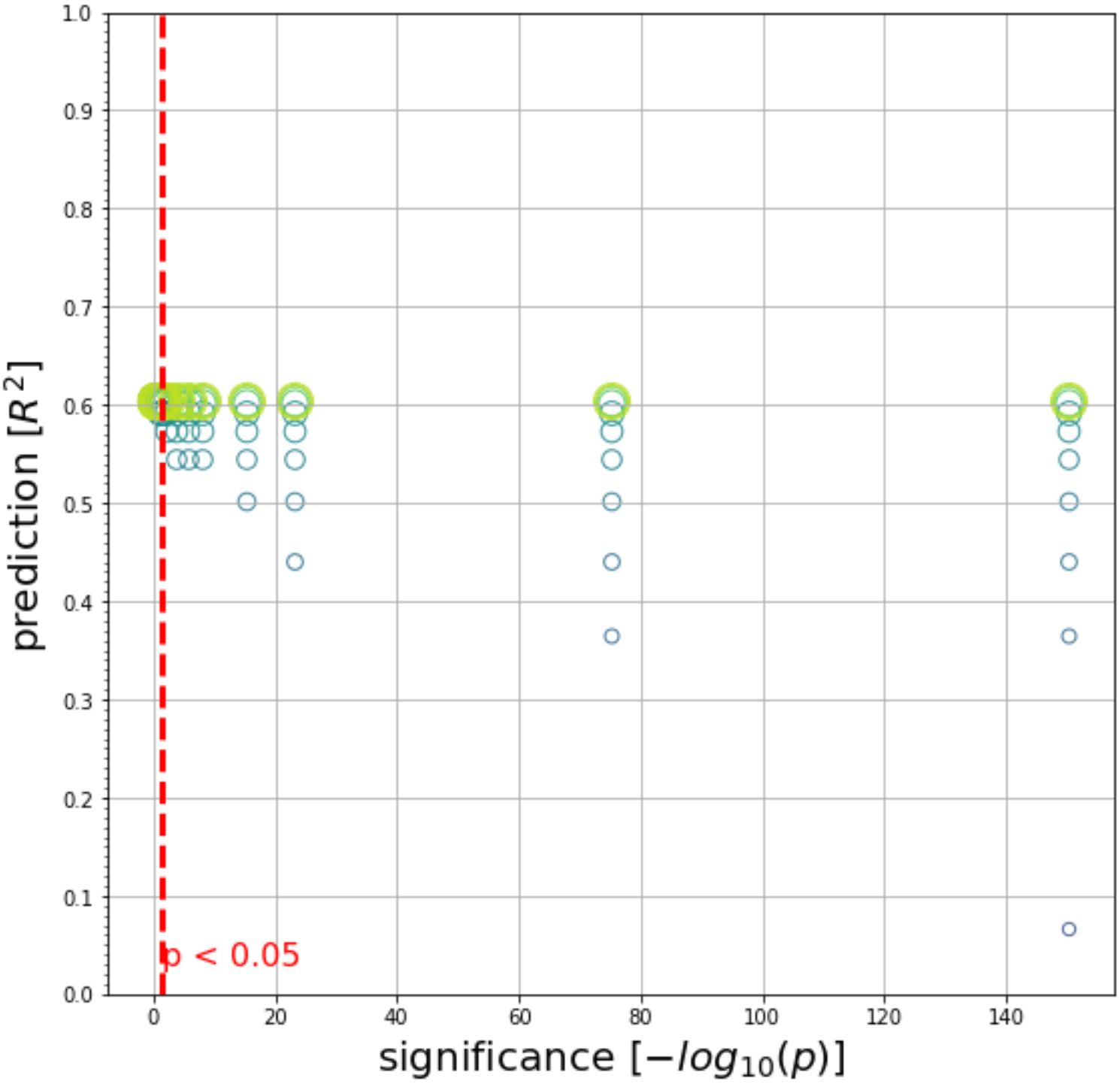
```
alpha: 0.0100 acc: 0.70 active_coefs: 40
alpha: 0.0140 acc: 0.70 active_coefs: 40
alpha: 0.0196 acc: 0.70 active_coefs: 40
alpha: 0.0274 acc: 0.70 active_coefs: 40
alpha: 0.0383 acc: 0.70 active_coefs: 40
alpha: 0.0536 acc: 0.70 active_coefs: 40
alpha: 0.0750 acc: 0.71 active_coefs: 40
alpha: 0.1049 acc: 0.71 active_coefs: 40
alpha: 0.1468 acc: 0.71 active_coefs: 40
alpha: 0.2054 acc: 0.71 active_coefs: 37
alpha: 0.2873 acc: 0.71 active_coefs: 33
alpha: 0.4019 acc: 0.71 active_coefs: 24
alpha: 0.5623 acc: 0.70 active_coefs: 12
alpha: 0.7867 acc: 0.68 active_coefs: 8
alpha: 1.1007 acc: 0.64 active_coefs: 7
alpha: 1.5399 acc: 0.57 active_coefs: 4
alpha: 2.1544 acc: 0.50 active_coefs: 3
alpha: 3.0142 acc: 0.41 active_coefs: 2
alpha: 4.2170 acc: 0.23 active_coefs: 2
alpha: 5.8997 acc: 0.08 active_coefs: 1
alpha: 8.2540 acc: -0.00 active_coefs: 0
alpha: 11.5478 acc: -0.00 active_coefs: 0
alpha: 16.1560 acc: -0.00 active_coefs: 0
alpha: 22.6030 acc: -0.00 active_coefs: 0
alpha: 31.6228 acc: -0.00 active_coefs: 0
40
150.243108154
75.0283749478
23.272204574
15.3187159372
7.87503220061
```

```
5.61870855681
3.81530402359
2.12069438148
1.46170253145
1.36882744084
1.12680810745
1.108092383
0.840109831874
0.788400497084
0.761307405398
0.739485213624
0.642302070955
0.630742505569
0.619784968806
0.618575918143
0.600478768004
0.588333972587
0.567855483538
0.491926495158
0.489941867261
0.481027300636
0.470793838239
0.459169120441
0.423885028547
0.41032864251
0.379035383086
0.342683717811
0.265901243367
0.185562455302
0.166495244096
0.0941668480445
0.0648977457284
0.0404564612609
0.0297791289573
0.0166129303079
skipping 3
skipping 4
skipping 6
skipping 7
skipping 8
skipping 9
skipping 10
skipping 11
skipping 18
skipping 21
skipping 22
skipping 23
skipping 24
```


Linear regression
true betas
estimated betas
significant at p<0.05
input variables


LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients



In [28]:

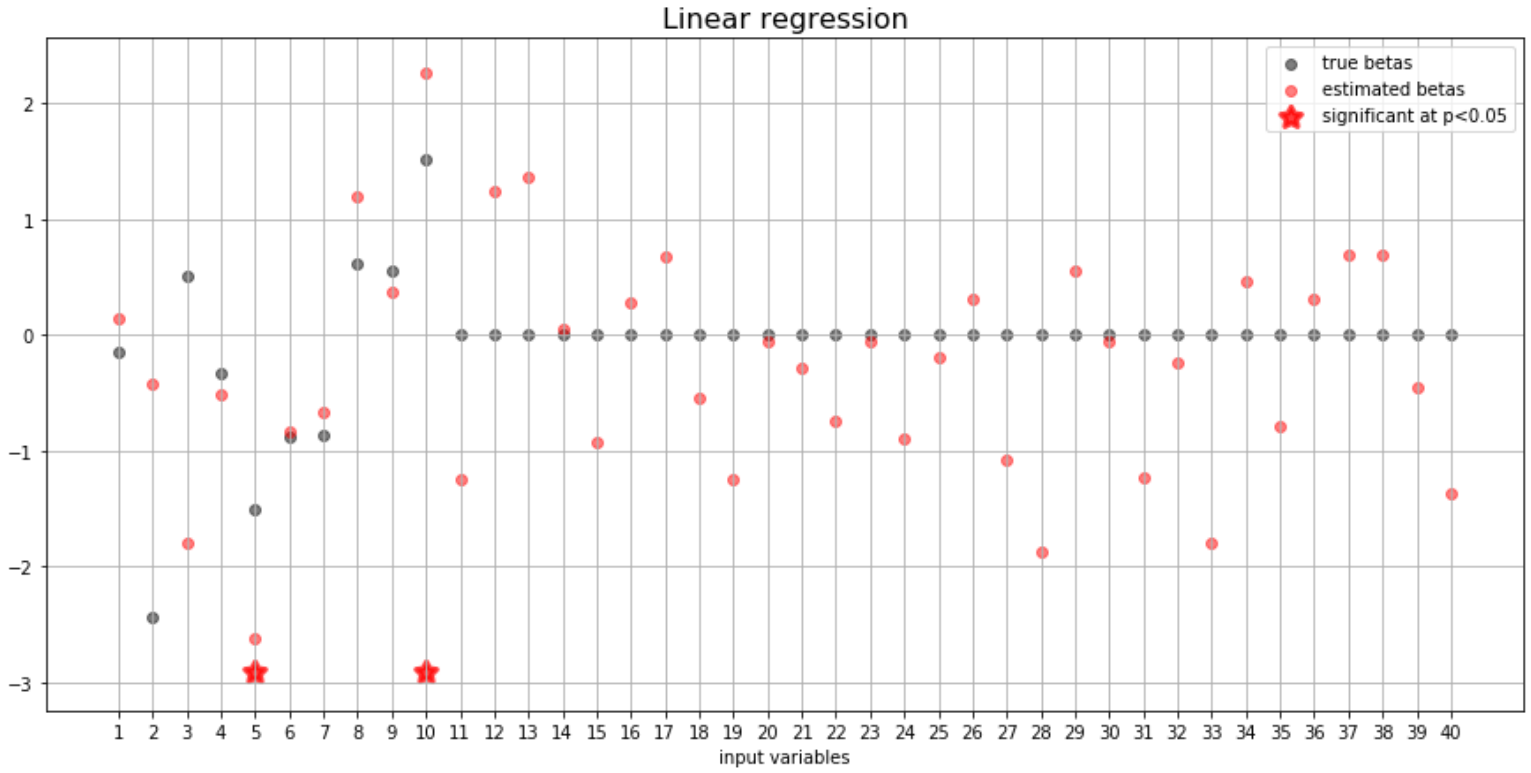
```
### observation: 4/5 path. variables are significant, the same 4/5 path. variabl
es are selected as predictive
```

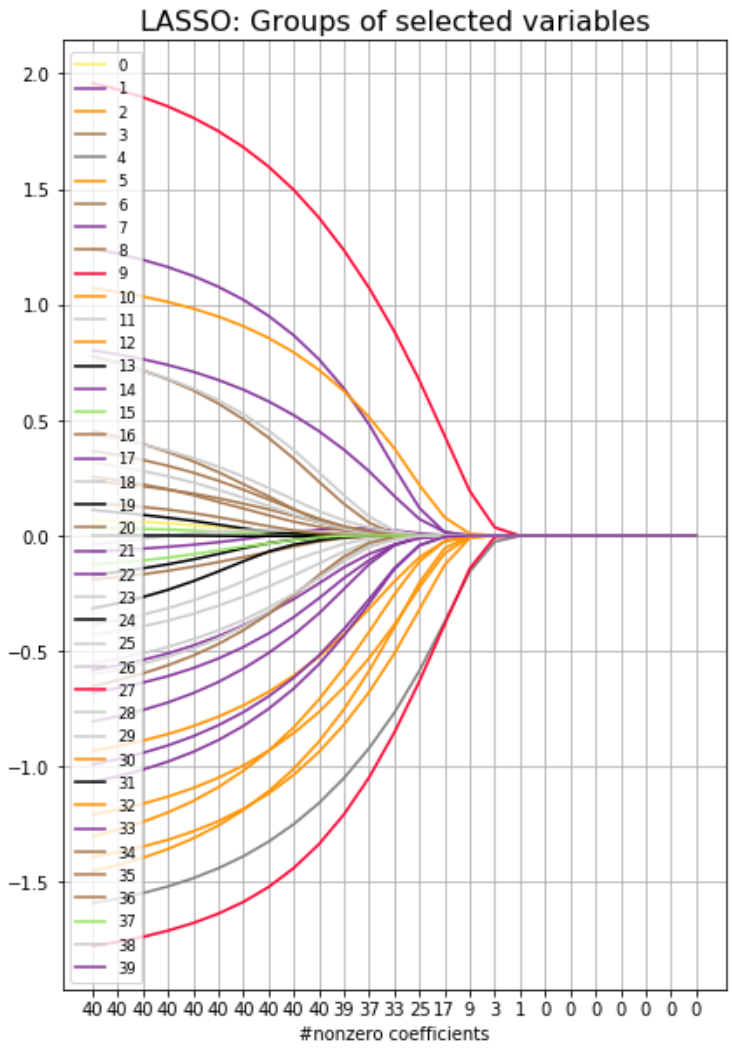
In [29]:

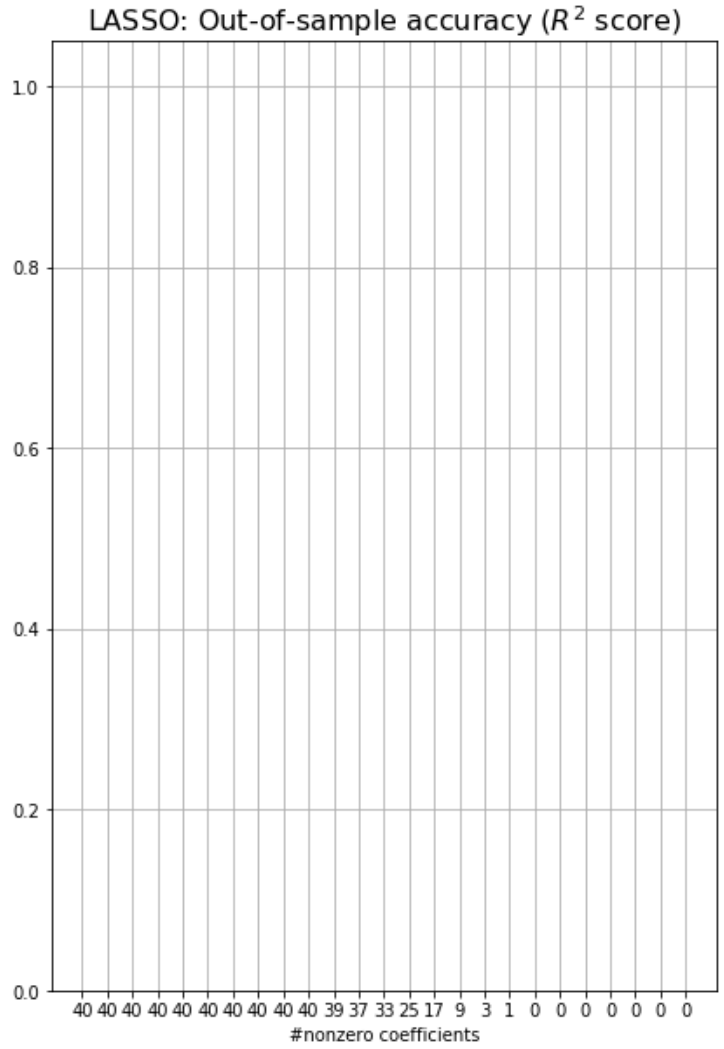
```
comment = "x^4 / dataset: 10/40 relevant variables, including 5 pathological one
s, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

X_4 = X.copy()
X_4[:, 0:6] = X_4[:, 0:6]**4  # introduce pathological transformation, NOT captu
red by ground-truth model
y = (true_coefs * X_4).sum(axis=1) + epsilon
C_grid = np.logspace(-1, 1.0, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

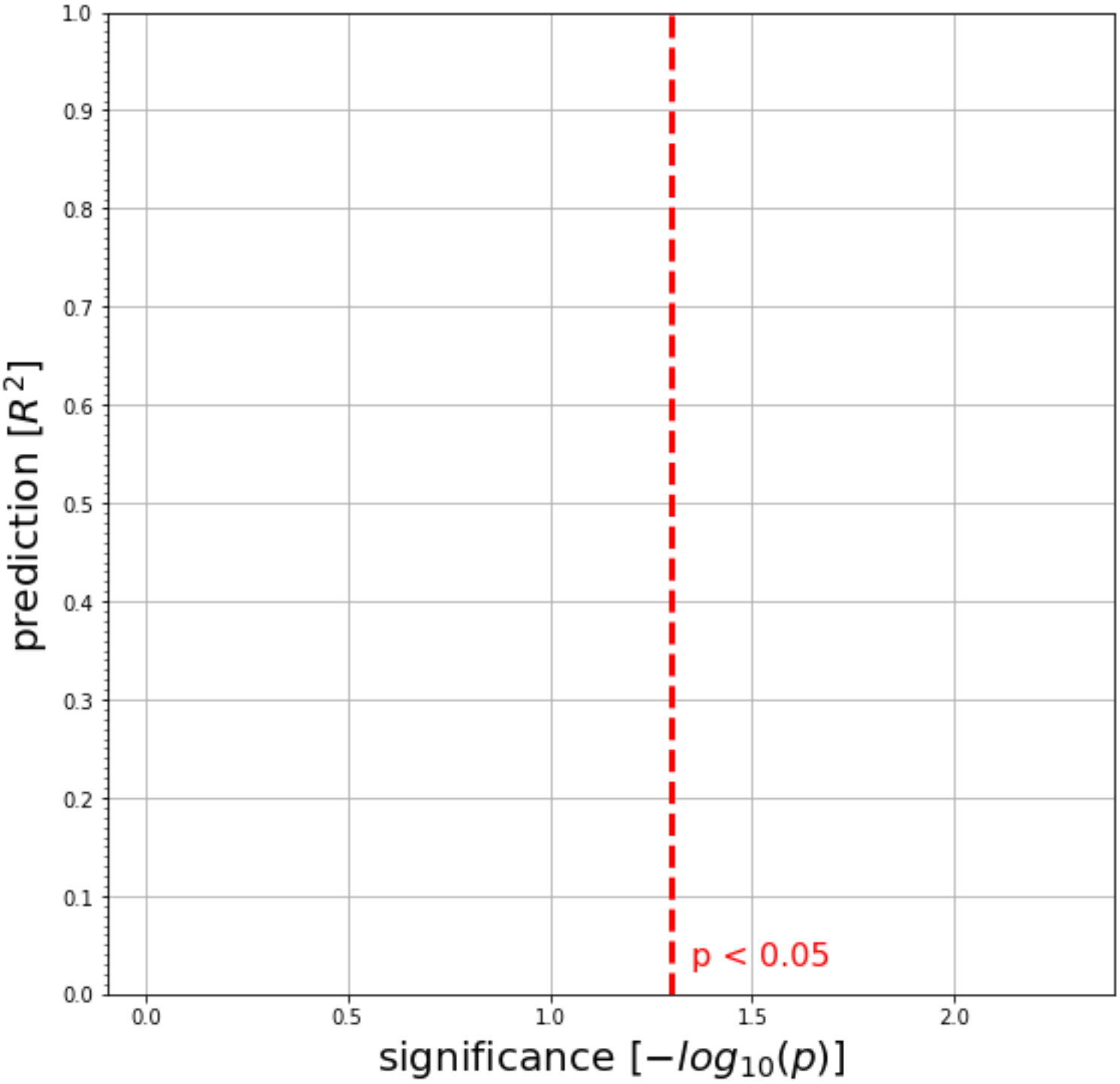
```
alpha: 0.1000 acc: -0.13 active_coefs: 40
alpha: 0.1212 acc: -0.13 active_coefs: 40
alpha: 0.1468 acc: -0.12 active_coefs: 40
alpha: 0.1778 acc: -0.12 active_coefs: 40
alpha: 0.2154 acc: -0.11 active_coefs: 40
alpha: 0.2610 acc: -0.11 active_coefs: 40
alpha: 0.3162 acc: -0.10 active_coefs: 40
alpha: 0.3831 acc: -0.09 active_coefs: 40
alpha: 0.4642 acc: -0.08 active_coefs: 40
alpha: 0.5623 acc: -0.07 active_coefs: 40
alpha: 0.6813 acc: -0.06 active_coefs: 39
alpha: 0.8254 acc: -0.05 active_coefs: 37
alpha: 1.0000 acc: -0.04 active_coefs: 33
alpha: 1.2115 acc: -0.02 active_coefs: 25
alpha: 1.4678 acc: -0.01 active_coefs: 17
alpha: 1.7783 acc: -0.00 active_coefs: 9
alpha: 2.1544 acc: -0.00 active_coefs: 3
alpha: 2.6102 acc: -0.00 active_coefs: 1
alpha: 3.1623 acc: -0.00 active_coefs: 0
alpha: 3.8312 acc: -0.00 active_coefs: 0
alpha: 4.6416 acc: -0.00 active_coefs: 0
alpha: 5.6234 acc: -0.00 active_coefs: 0
alpha: 6.8129 acc: -0.00 active_coefs: 0
alpha: 8.2540 acc: -0.00 active_coefs: 0
alpha: 10.0000 acc: -0.00 active_coefs: 0
40
2.27312640394
1.81146459557
1.28792716039
1.23638187293
1.19728979578
```

```
0.823951430378
0.8059726792
0.731878206262
0.705922719017
0.68986927677
0.657107742186
0.644955707324
0.586704013568
0.457372465623
0.41761560617
0.402731665316
0.393307161277
0.362987419564
0.3464016722
0.324458303681
0.31708272733
0.308342189407
0.251868690387
0.234421959485
0.217933699178
0.201723699943
0.193755453118
0.181325877158
0.156509331924
0.128717941181
0.125950774089
0.116441899229
0.108004883958
0.0963564816516
0.0729641005321
0.0575923934638
0.0231724384812
0.0218756211694
0.0200828890486
0.0183867576655
skipping 1
skipping 2
skipping 3
skipping 4
skipping 5
skipping 6
skipping 16
skipping 17
skipping 18
skipping 19
skipping 20
skipping 21
skipping 22
skipping 23
skipping 24
```


Linear regression
true betas
estimated betas
significant at p<0.05
input variables


LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients



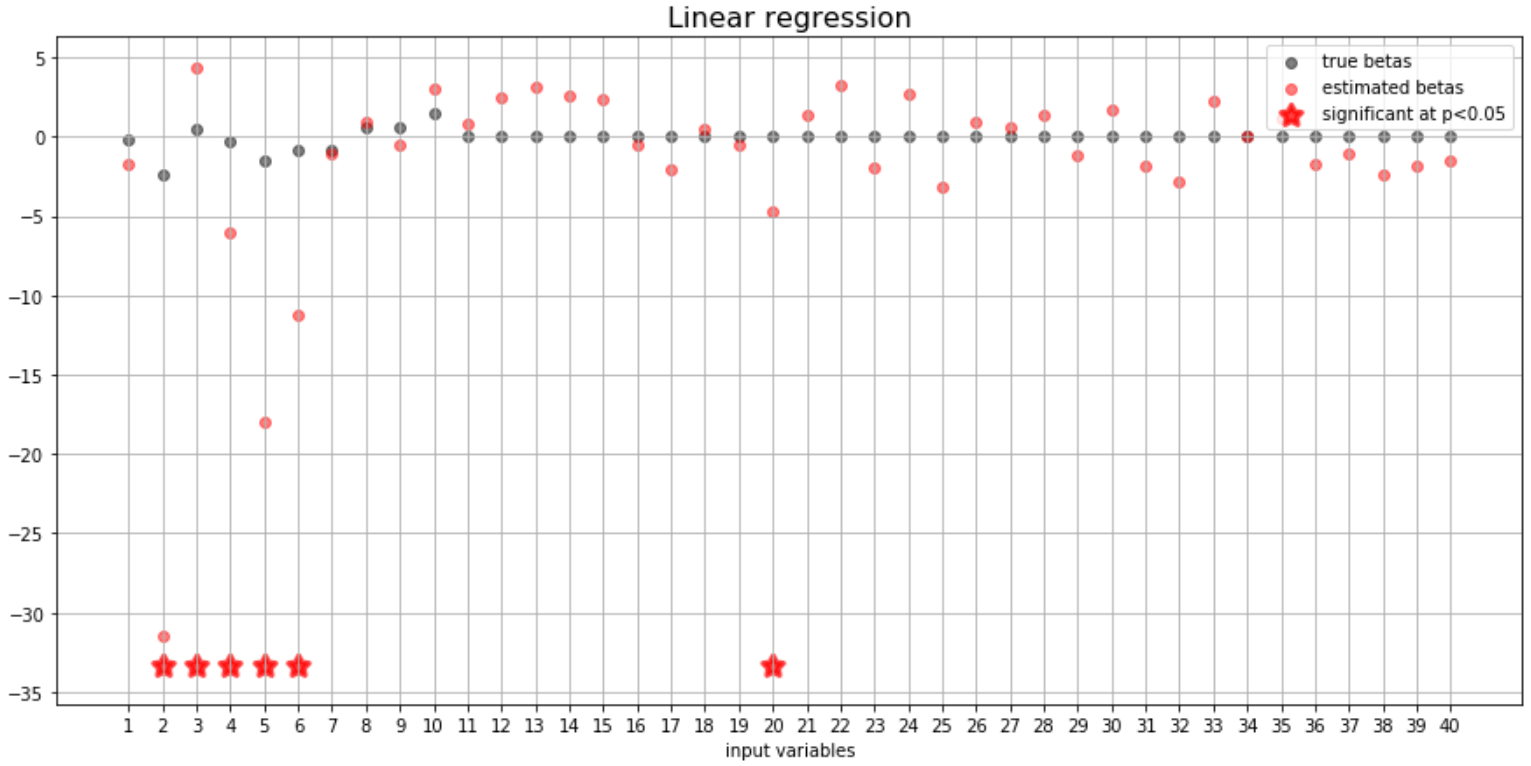
observation: 1/5 path. variables are significant, the same 1/5 path. variables are selected as predictive + coefficients for other 4/5 relevant variables are messed up

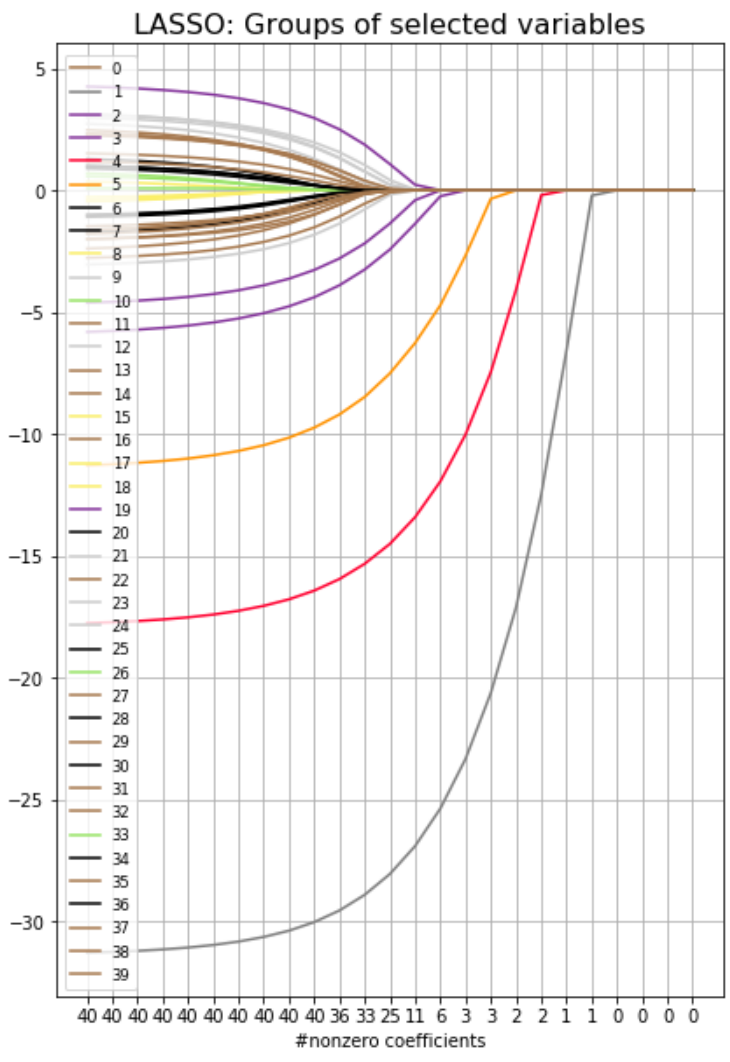
In [30]:

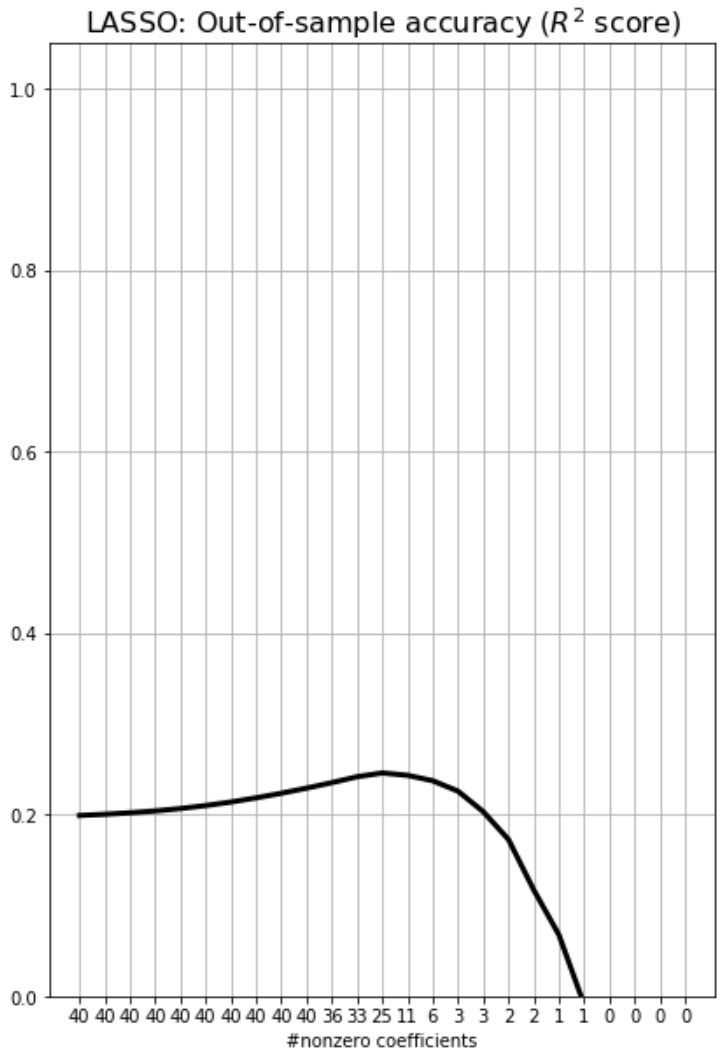
```
comment = "x^5 / dataset: 10/40 relevant variables, including 5 pathological one
s, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

X_5 = X.copy()
X_5[:, 0:6] = X_5[:, 0:6]**5  # introduce pathological transformation, NOT captu
red by ground-truth model
y = (true_coefs * X_5).sum(axis=1) + epsilon
C_grid = np.logspace(-1, 2, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

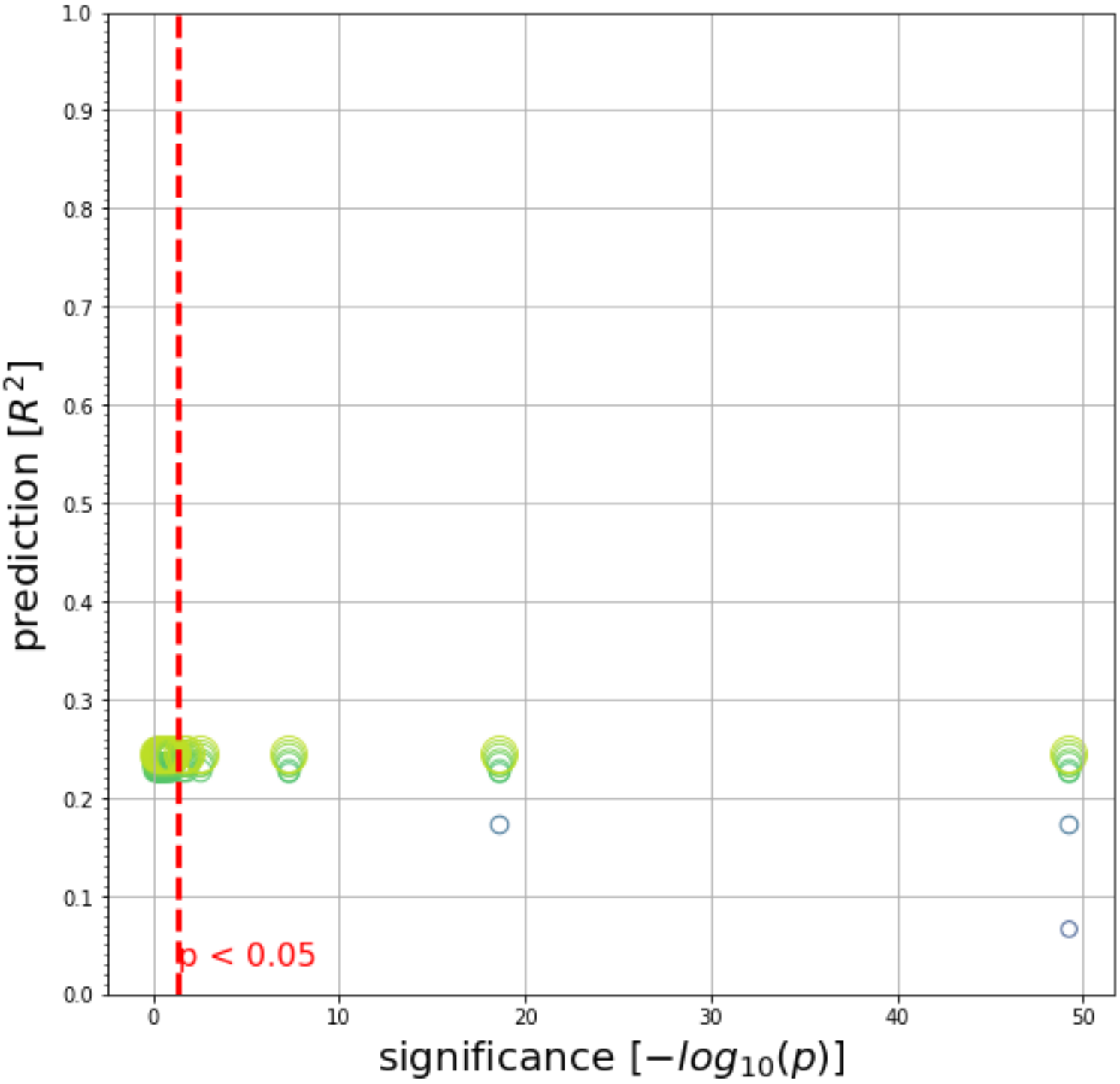
```
alpha: 0.1000 acc: 0.28 active_coefs: 40
alpha: 0.1334 acc: 0.28 active_coefs: 40
alpha: 0.1778 acc: 0.29 active_coefs: 40
alpha: 0.2371 acc: 0.29 active_coefs: 40
alpha: 0.3162 acc: 0.29 active_coefs: 40
alpha: 0.4217 acc: 0.30 active_coefs: 40
alpha: 0.5623 acc: 0.30 active_coefs: 40
alpha: 0.7499 acc: 0.31 active_coefs: 40
alpha: 1.0000 acc: 0.31 active_coefs: 40
alpha: 1.3335 acc: 0.32 active_coefs: 40
alpha: 1.7783 acc: 0.33 active_coefs: 36
alpha: 2.3714 acc: 0.34 active_coefs: 33
alpha: 3.1623 acc: 0.35 active_coefs: 25
alpha: 4.2170 acc: 0.35 active_coefs: 11
alpha: 5.6234 acc: 0.33 active_coefs: 6
alpha: 7.4989 acc: 0.30 active_coefs: 3
alpha: 10.0000 acc: 0.26 active_coefs: 3
alpha: 13.3352 acc: 0.22 active_coefs: 2
alpha: 17.7828 acc: 0.15 active_coefs: 2
alpha: 23.7137 acc: 0.10 active_coefs: 1
alpha: 31.6228 acc: -0.00 active_coefs: 1
alpha: 42.1697 acc: -0.00 active_coefs: 0
alpha: 56.2341 acc: -0.00 active_coefs: 0
alpha: 74.9894 acc: -0.00 active_coefs: 0
alpha: 100.0000 acc: -0.00 active_coefs: 0
40
49.2371190602
18.584028178
7.21483498789
2.47120034517
1.77517671435
```

```
1.50111176216
0.996707702412
0.941775600473
0.932778004676
0.902291768925
0.821245862361
0.676404064949
0.676265814707
0.62985792574
0.628353937742
0.621159743954
0.610128413389
0.523606146004
0.504442347076
0.455136106657
0.435947510042
0.428599490922
0.409290482757
0.403699906469
0.346030679159
0.299699876642
0.294899094667
0.241158419627
0.239141190831
0.238607265024
0.232912077744
0.200913881714
0.18408167703
0.159837282161
0.107336533579
0.103838877059
0.100530824106
0.0992206235634
0.0932097845463
0.0151452277154
skipping 7
skipping 8
skipping 9
skipping 10
skipping 11
skipping 12
skipping 13
skipping 14
skipping 15
skipping 16
skipping 18
skipping 20
skipping 22
skipping 23
skipping 24
```


Linear regression
true betas
estimated betas
significant at p<0.05
input variables


LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients



observation: similar to X^3 4/5 path. variables are significant, the same 4/5 path. variables are selected as predictive; other than X^3 but similar to X^4 coefficients for other 4/5 relevant variables are messed up

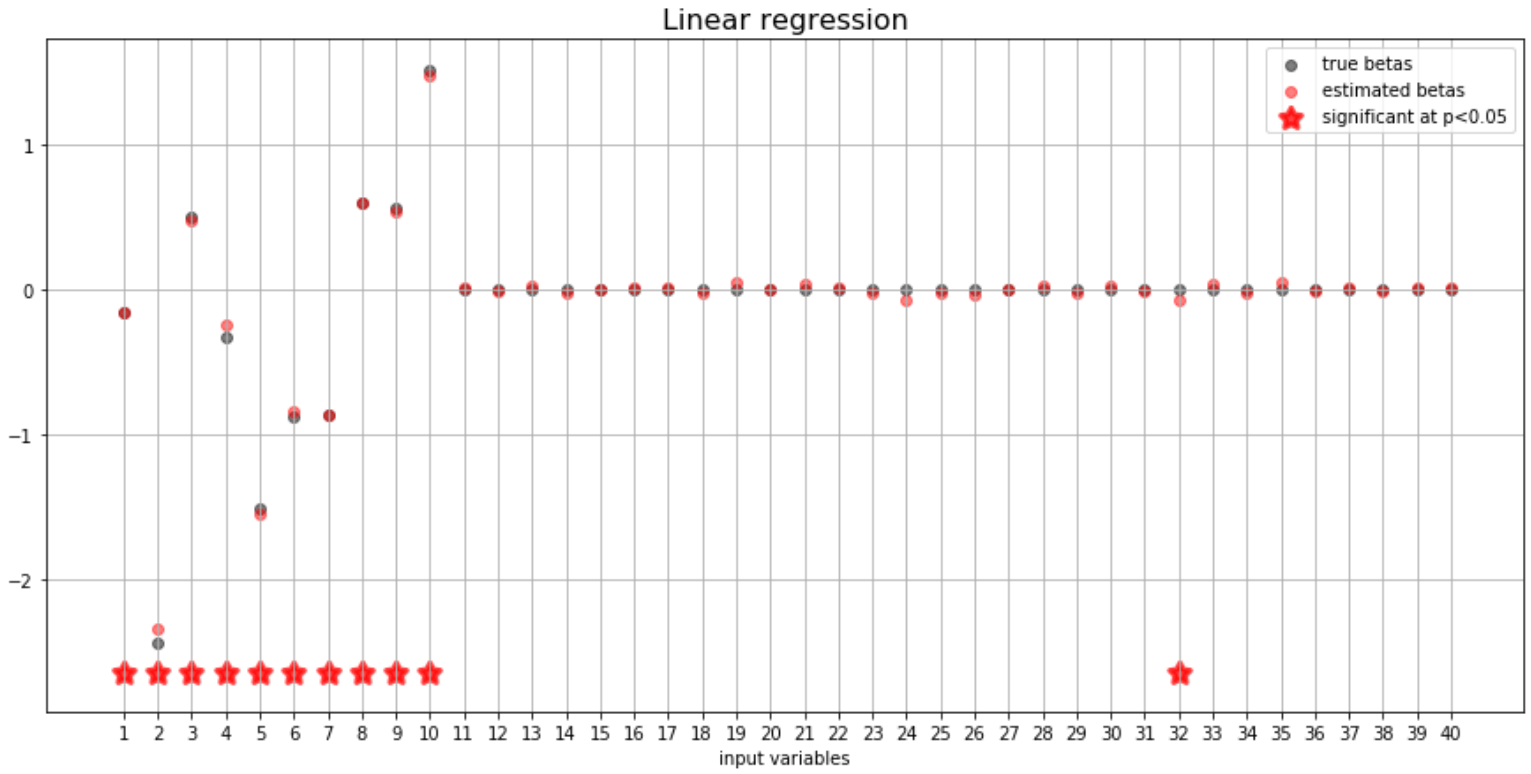
Multicollinearity with correct model, 1000 samples, 40 variables, error = N(0, 1)__

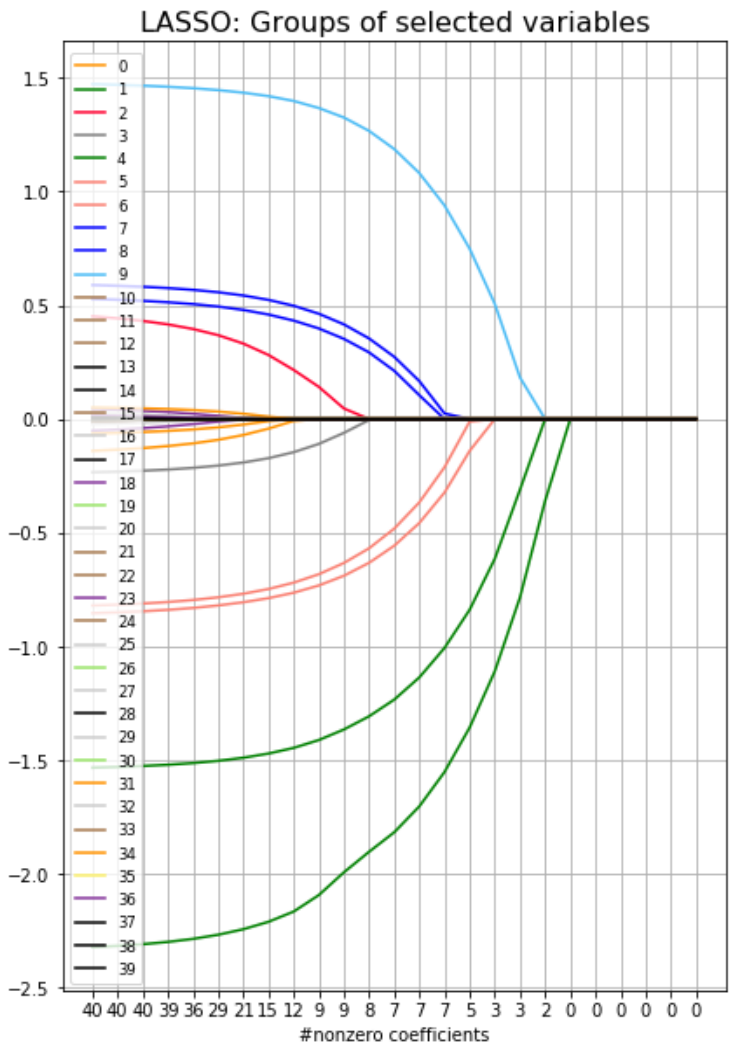
In [31]:

```
comment = "3 correlated vars at ~50% / dataset: 10/40 relevant variables, linear
ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

n_corr_feat = 3
cov = np.ones((n_corr_feat, n_corr_feat)) * .5
cov[np.diag_indices(n_corr_feat)] = 1

X_corr = rs.multivariate_normal(mean=np.zeros((n_corr_feat)), cov=cov, size=n_sa
mples)
X[:, 0:n_corr_feat] = X_corr
y = (true_coefs * X).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 1, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

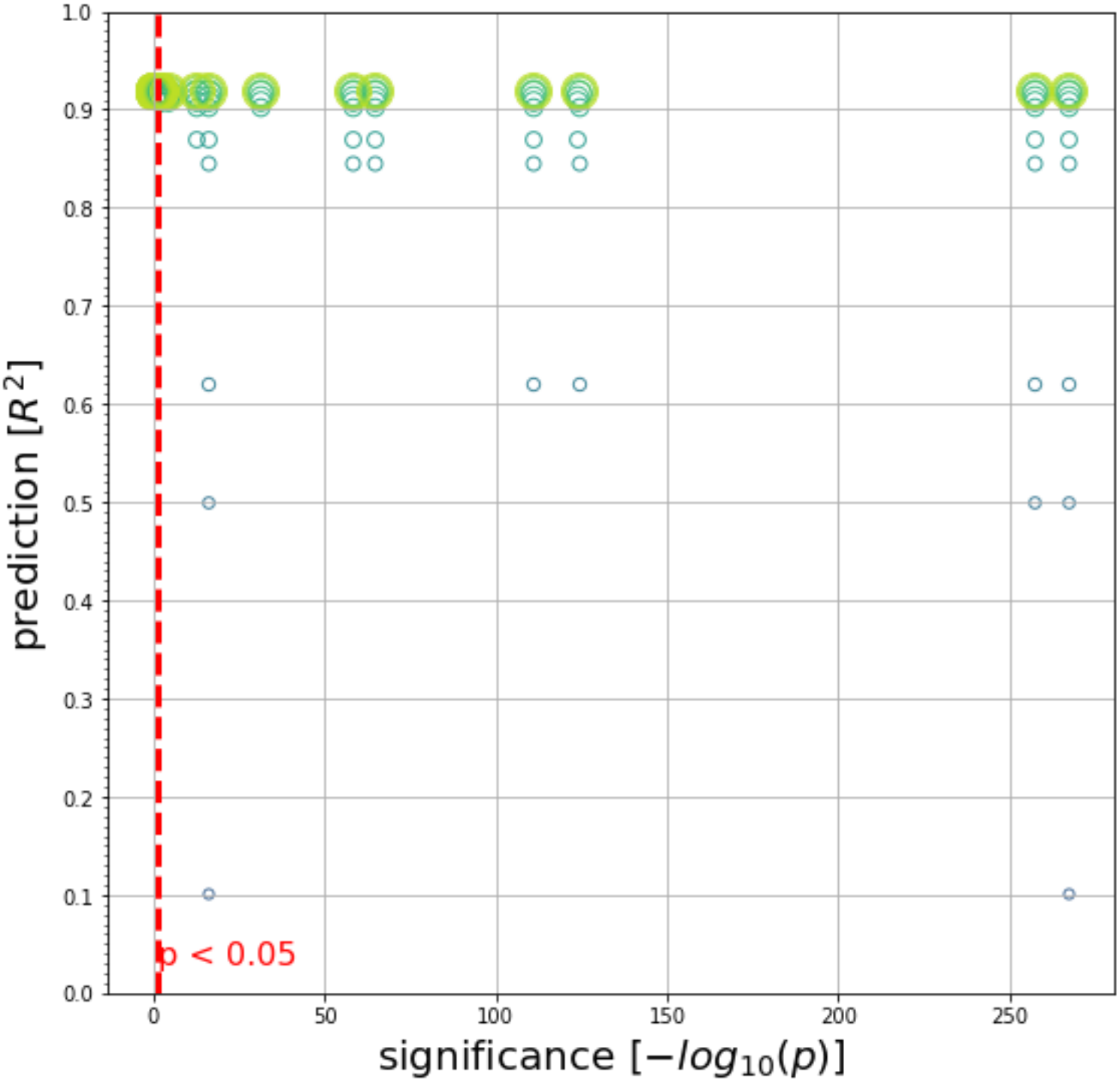
```
alpha: 0.0100 acc: 0.93 active_coefs: 40
alpha: 0.0133 acc: 0.93 active_coefs: 40
alpha: 0.0178 acc: 0.93 active_coefs: 40
alpha: 0.0237 acc: 0.93 active_coefs: 39
alpha: 0.0316 acc: 0.93 active_coefs: 36
alpha: 0.0422 acc: 0.93 active_coefs: 29
alpha: 0.0562 acc: 0.92 active_coefs: 21
alpha: 0.0750 acc: 0.92 active_coefs: 15
alpha: 0.1000 acc: 0.91 active_coefs: 12
alpha: 0.1334 acc: 0.90 active_coefs: 9
alpha: 0.1778 acc: 0.89 active_coefs: 9
alpha: 0.2371 acc: 0.87 active_coefs: 8
alpha: 0.3162 acc: 0.85 active_coefs: 7
alpha: 0.4217 acc: 0.81 active_coefs: 7
alpha: 0.5623 acc: 0.73 active_coefs: 7
alpha: 0.7499 acc: 0.64 active_coefs: 5
alpha: 1.0000 acc: 0.51 active_coefs: 3
alpha: 1.3335 acc: 0.30 active_coefs: 3
alpha: 1.7783 acc: 0.10 active_coefs: 2
alpha: 2.3714 acc: -0.00 active_coefs: 0
alpha: 3.1623 acc: -0.00 active_coefs: 0
alpha: 4.2170 acc: -0.00 active_coefs: 0
alpha: 5.6234 acc: -0.00 active_coefs: 0
alpha: 7.4989 acc: -0.00 active_coefs: 0
alpha: 10.0000 acc: -0.00 active_coefs: 0
40
```

267.297057962
257.077329698
123.953958013
110.714409952
64.5552706964
58.1514318377
31.4184117078
15.6535597745
12.3603261008
3.93613852142
1.72093092201
1.23638678616
1.13629145185
1.04180914503
0.777051875767
0.612476971947
0.529772284355
0.48723140239
0.47010409933
0.377690974262
0.331074148568
0.325426743346
0.323438450552
0.321005746635
0.320191060137
0.311990939412
0.299338839163
0.294236590827
0.287545269398
0.268444087709
0.25449472524
0.246914692328
0.145628291764
0.121930395595
0.101743752358
0.0519583947804
0.0384864960574
0.0313702428645
0.028852048216
0.0132349880354
skipping 3
skipping 4
skipping 10
skipping 13
skipping 14
skipping 17
skipping 20
skipping 21
skipping 22
skipping 23
skipping 24


Linear regression
true betas
estimated betas
significant at p<0.05
input variables


LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients



In [32]:

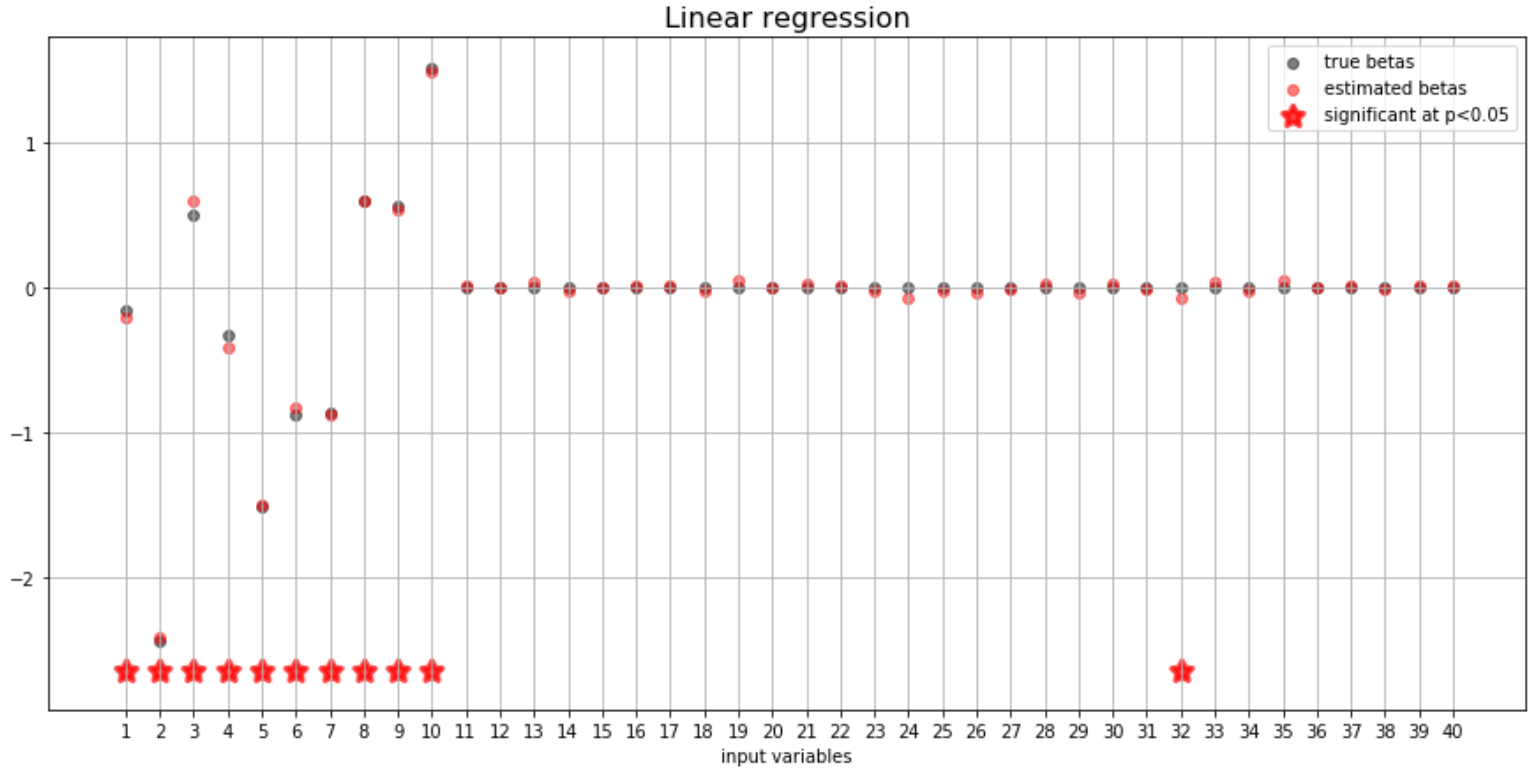
```
### observation: 10/10 variables are significant, the 2/3 corr. variables are se
lected as predictive
```

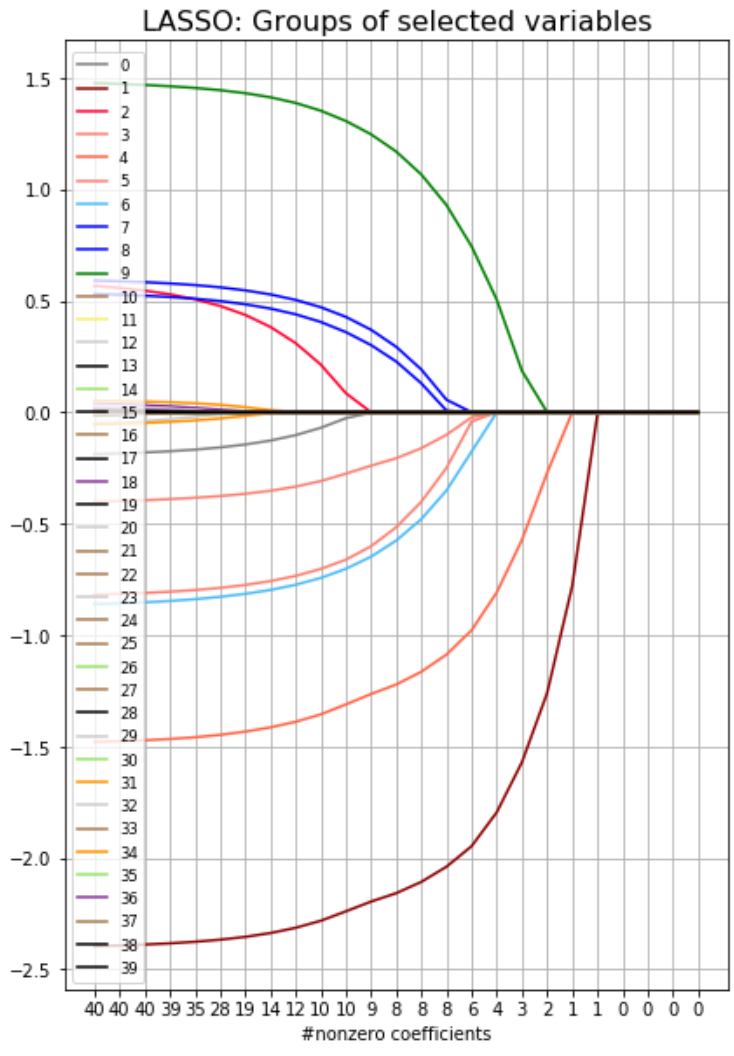
In [33]:

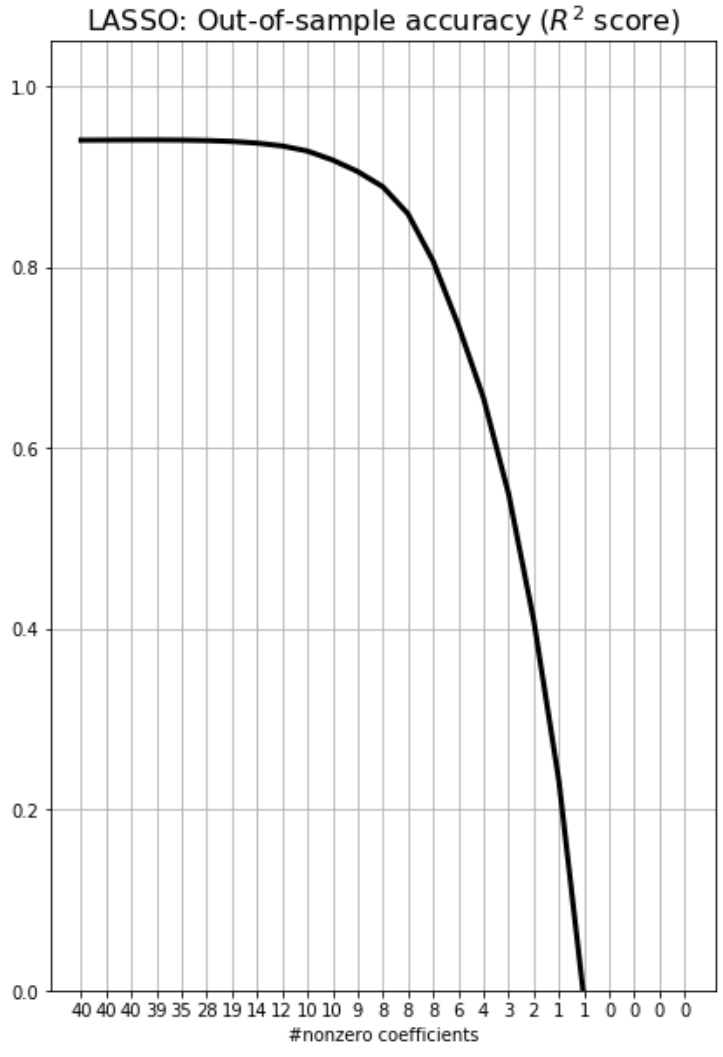
```
comment = "5 correlated vars at ~50% / dataset: 10/40 relevant variables, linear
ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

n_corr_feat = 5
cov = np.ones((n_corr_feat, n_corr_feat)) * .5
cov[np.diag_indices(n_corr_feat)] = 1

X_corr = rs.multivariate_normal(mean=np.zeros((n_corr_feat)), cov=cov, size=n_sa
mples)
X[:, 0:n_corr_feat] = X_corr
y = (true_coefs * X).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 1, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

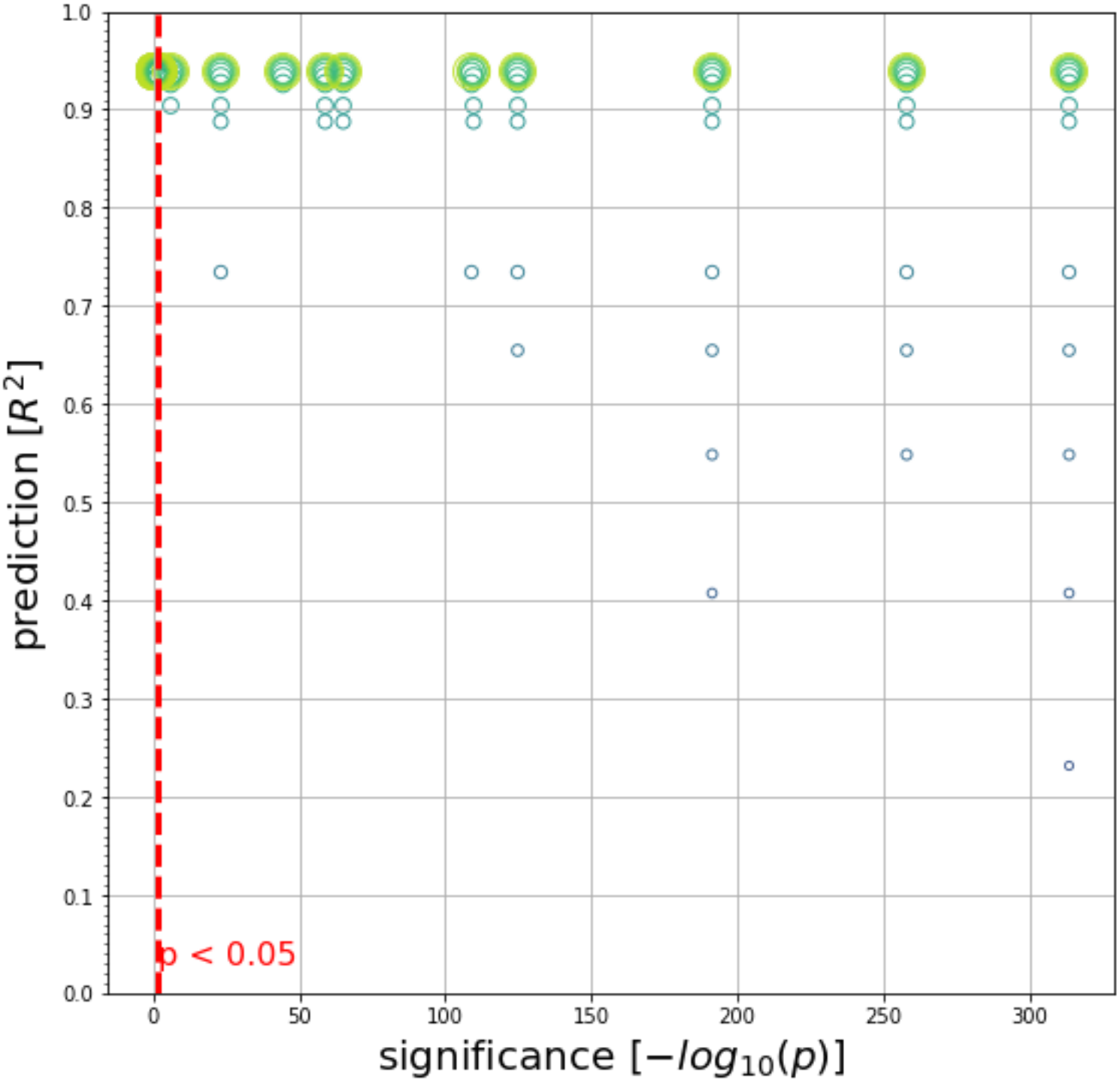
```
alpha: 0.0100 acc: 0.94 active_coefs: 40
alpha: 0.0133 acc: 0.94 active_coefs: 40
alpha: 0.0178 acc: 0.94 active_coefs: 40
alpha: 0.0237 acc: 0.94 active_coefs: 39
alpha: 0.0316 acc: 0.94 active_coefs: 35
alpha: 0.0422 acc: 0.94 active_coefs: 28
alpha: 0.0562 acc: 0.94 active_coefs: 19
alpha: 0.0750 acc: 0.94 active_coefs: 14
alpha: 0.1000 acc: 0.94 active_coefs: 12
alpha: 0.1334 acc: 0.93 active_coefs: 10
alpha: 0.1778 acc: 0.92 active_coefs: 10
alpha: 0.2371 acc: 0.91 active_coefs: 9
alpha: 0.3162 acc: 0.89 active_coefs: 8
alpha: 0.4217 acc: 0.85 active_coefs: 8
alpha: 0.5623 acc: 0.79 active_coefs: 8
alpha: 0.7499 acc: 0.71 active_coefs: 6
alpha: 1.0000 acc: 0.62 active_coefs: 4
alpha: 1.3335 acc: 0.47 active_coefs: 3
alpha: 1.7783 acc: 0.34 active_coefs: 2
alpha: 2.3714 acc: 0.20 active_coefs: 1
alpha: 3.1623 acc: 0.01 active_coefs: 1
alpha: 4.2170 acc: -0.00 active_coefs: 0
alpha: 5.6234 acc: -0.00 active_coefs: 0
alpha: 7.4989 acc: -0.00 active_coefs: 0
alpha: 10.0000 acc: -0.00 active_coefs: 0
40
313.549517688
```

257.926430562
191.06404283
124.531244884
108.997057085
64.4613934571
58.3681547567
44.0200256324
22.708161523
5.84026475176
1.38440748688
1.26029580981
1.18641696721
0.872586989065
0.822187337005
0.642076888446
0.583364864781
0.532313902143
0.37487020594
0.369299326617
0.365493311854
0.363450021274
0.330079475045
0.321739404554
0.31357492096
0.306167086407
0.261738855109
0.257064875627
0.201814622868
0.193021207977
0.182120408073
0.170821703447
0.170264349871
0.0927879134536
0.0813540679751
0.0781248126016
0.044205632203
0.0314982467255
0.0249758350177
0.0060603287728
skipping 2
skipping 4
skipping 10
skipping 13
skipping 14
skipping 20
skipping 22
skipping 23
skipping 24


Linear regression
true betas
estimated betas
significant at p<0.05
input variables


LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients



In [34]:

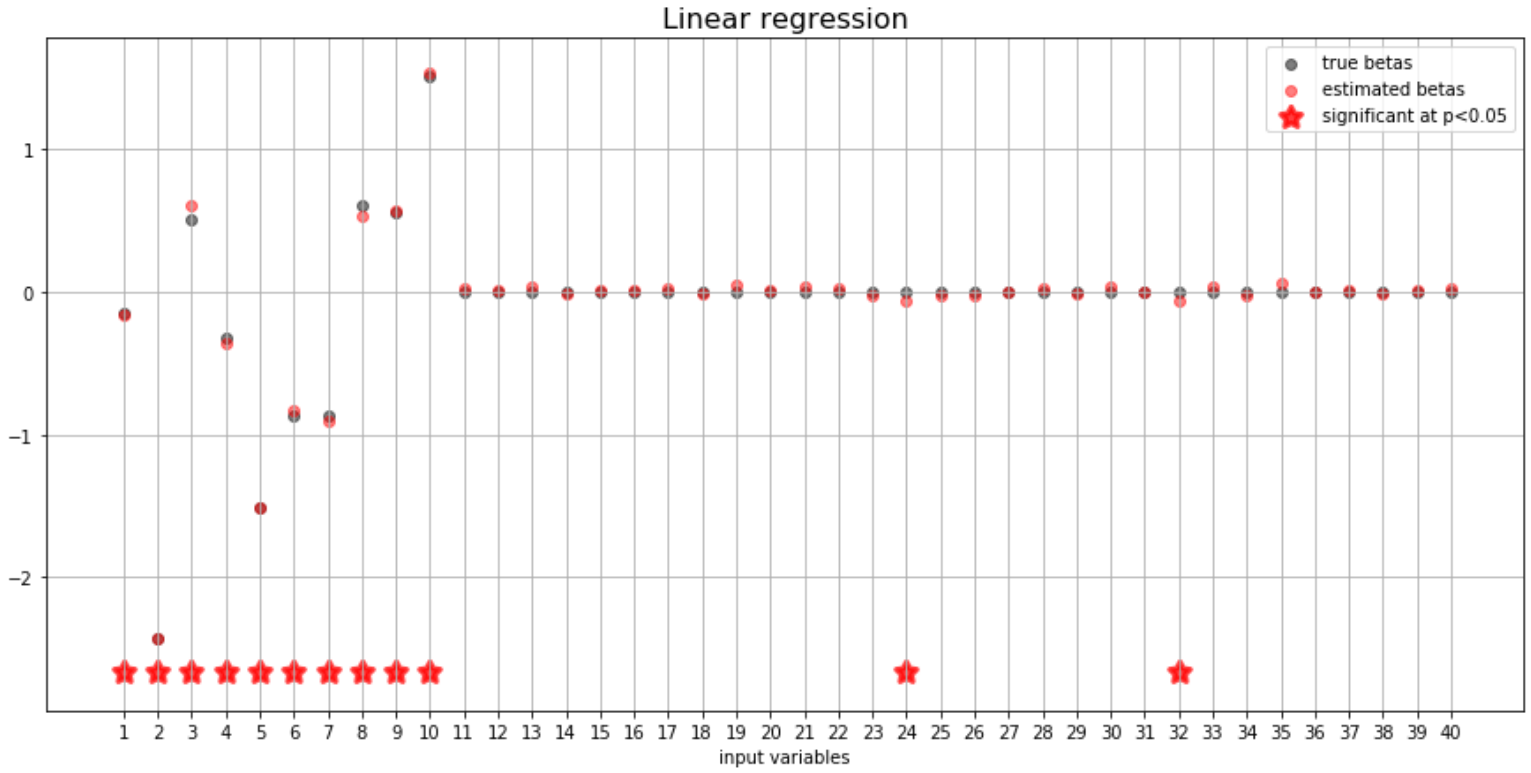
```
### observation: 10/10 variables are significant, the 5/5 (?) corr. variables ar
e selected as predictive
```

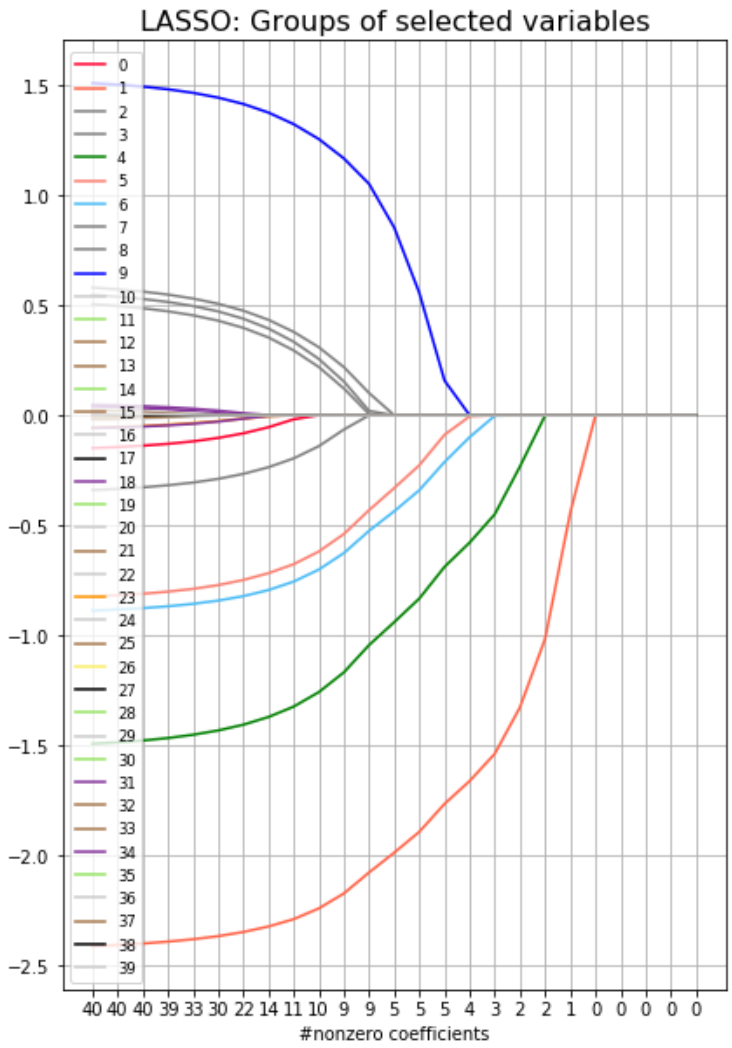
In [35]:

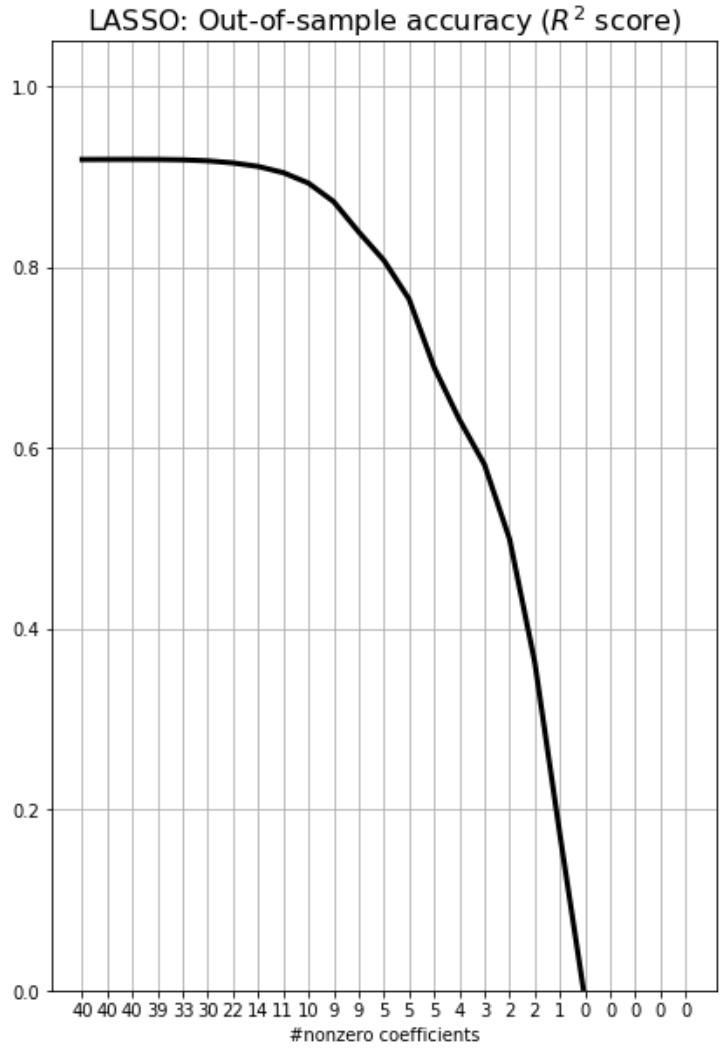
```
comment = "10 correlated vars at ~50% / dataset: 10/40 relevant variables, linea
r ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

n_corr_feat = 10
cov = np.ones((n_corr_feat, n_corr_feat)) * .5
cov[np.diag_indices(n_corr_feat)] = 1

X_corr = rs.multivariate_normal(mean=np.zeros((n_corr_feat)), cov=cov, size=n_sa
mples)
X[:, 0:n_corr_feat] = X_corr
y = (true_coefs * X).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 1, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

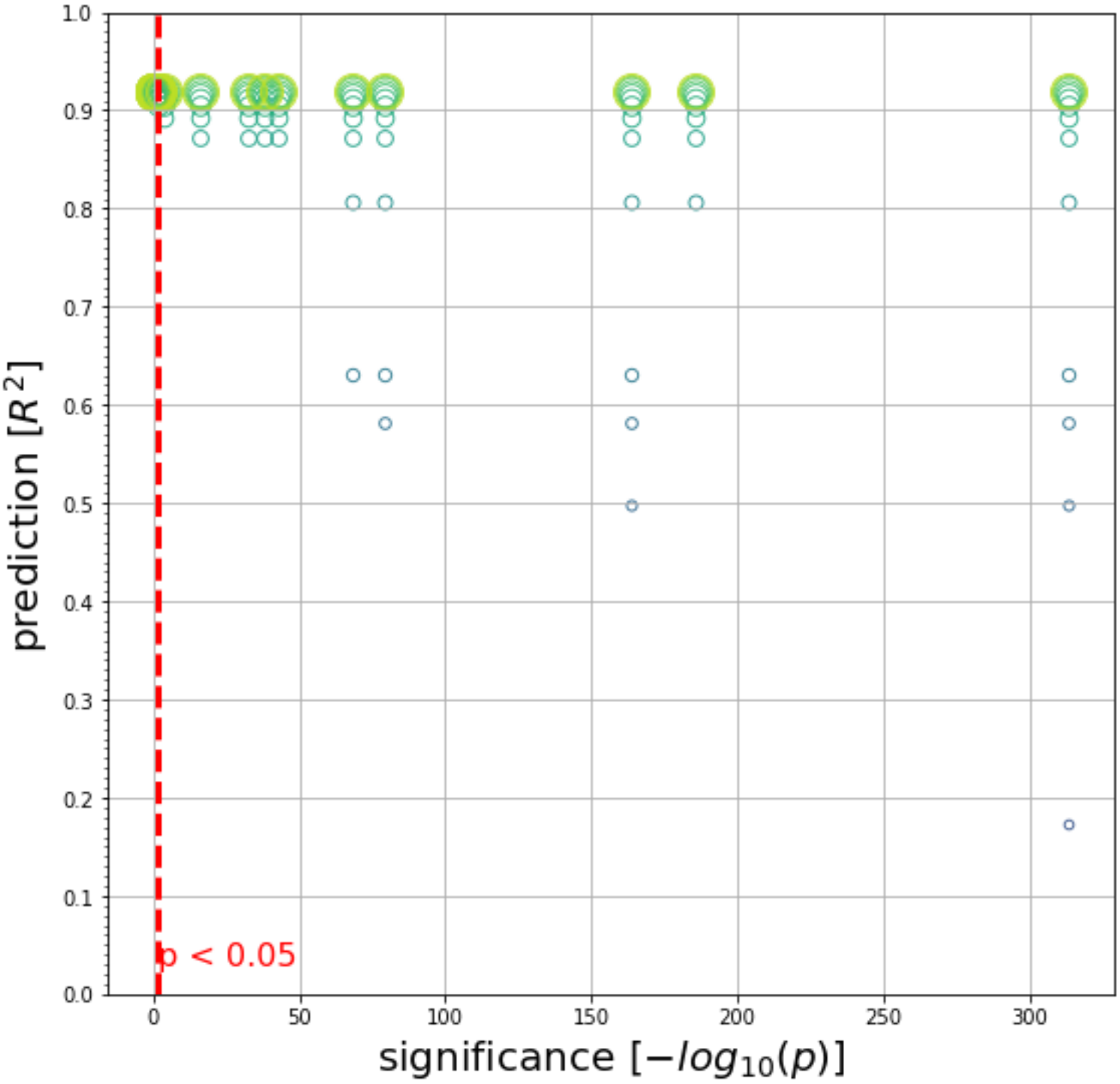
```
alpha: 0.0100 acc: 0.91 active_coefs: 40
alpha: 0.0133 acc: 0.91 active_coefs: 40
alpha: 0.0178 acc: 0.91 active_coefs: 40
alpha: 0.0237 acc: 0.91 active_coefs: 39
alpha: 0.0316 acc: 0.91 active_coefs: 33
alpha: 0.0422 acc: 0.91 active_coefs: 30
alpha: 0.0562 acc: 0.91 active_coefs: 22
alpha: 0.0750 acc: 0.90 active_coefs: 14
alpha: 0.1000 acc: 0.89 active_coefs: 11
alpha: 0.1334 acc: 0.88 active_coefs: 10
alpha: 0.1778 acc: 0.85 active_coefs: 9
alpha: 0.2371 acc: 0.82 active_coefs: 9
alpha: 0.3162 acc: 0.78 active_coefs: 5
alpha: 0.4217 acc: 0.74 active_coefs: 5
alpha: 0.5623 acc: 0.66 active_coefs: 5
alpha: 0.7499 acc: 0.61 active_coefs: 4
alpha: 1.0000 acc: 0.57 active_coefs: 3
alpha: 1.3335 acc: 0.50 active_coefs: 2
alpha: 1.7783 acc: 0.38 active_coefs: 2
alpha: 2.3714 acc: 0.18 active_coefs: 1
alpha: 3.1623 acc: -0.01 active_coefs: 0
alpha: 4.2170 acc: -0.01 active_coefs: 0
alpha: 5.6234 acc: -0.01 active_coefs: 0
alpha: 7.4989 acc: -0.01 active_coefs: 0
alpha: 10.0000 acc: -0.01 active_coefs: 0
40
313.604664308
```

185.342557077
163.286474962
78.8073408038
68.0425959965
42.8118588326
38.1680054874
32.1663345522
16.1010294856
3.88226847491
1.51908251505
1.31671870707
1.13331713008
0.942980337959
0.671334026061
0.527527958679
0.516166975827
0.515060297813
0.45746543492
0.454561489988
0.398878294933
0.390650496877
0.35151437345
0.328316074233
0.304453208307
0.286492169873
0.240840246415
0.23451903018
0.233926076692
0.204222907567
0.199829895486
0.199651203146
0.152414067699
0.14480598684
0.128502593025
0.0738539509642
0.058806834594
0.0433177491567
0.0175832712395
0.000532831109096
skipping 1
skipping 3
skipping 11
skipping 13
skipping 14
skipping 18
skipping 21
skipping 22
skipping 23
skipping 24


Linear regression
true betas
estimated betas
significant at p<0.05
input variables


LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients



In [36]:

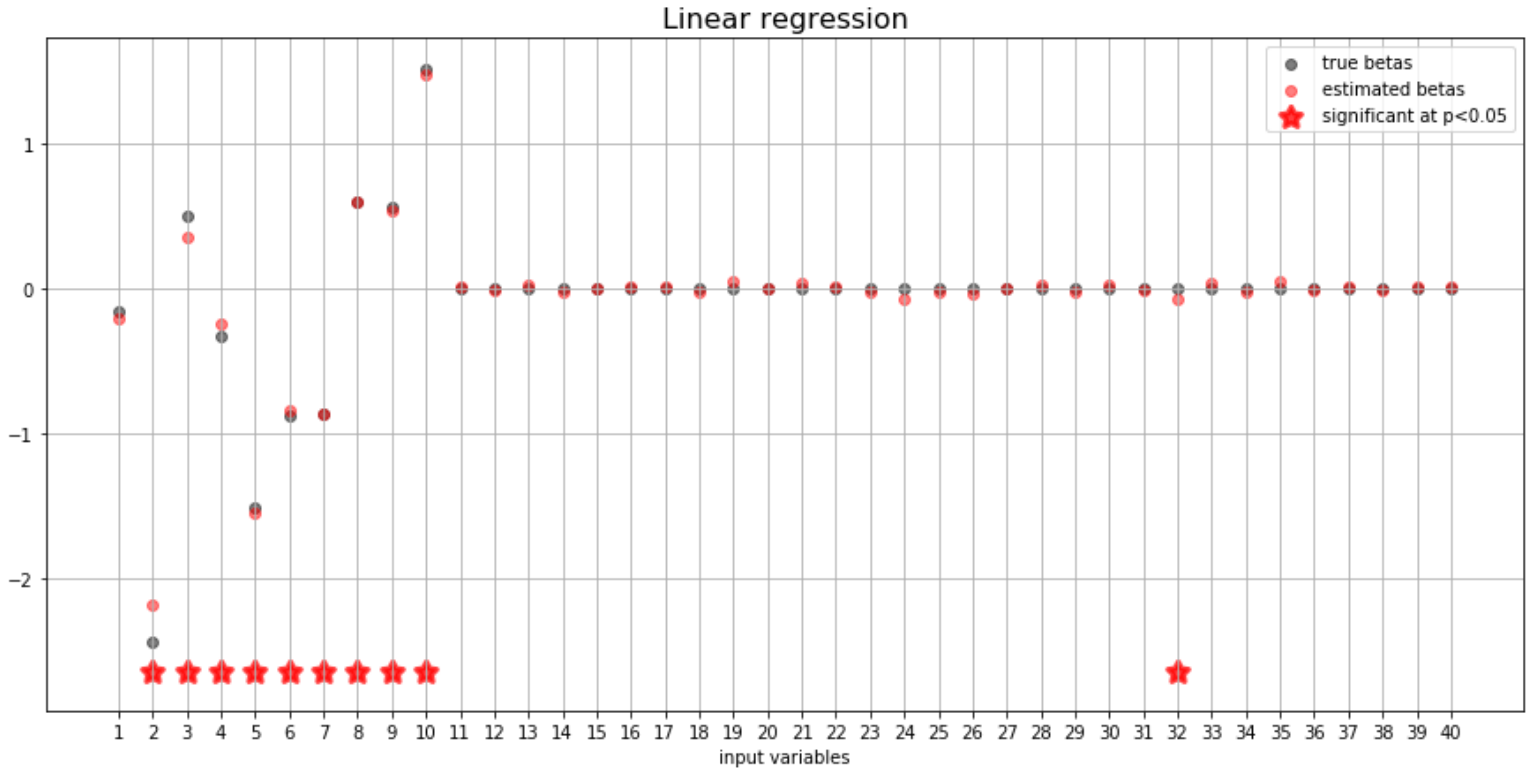
```
### observation: 10/10 variables are significant, the 9/10 (?) corr. variables a
re selected as predictive
```

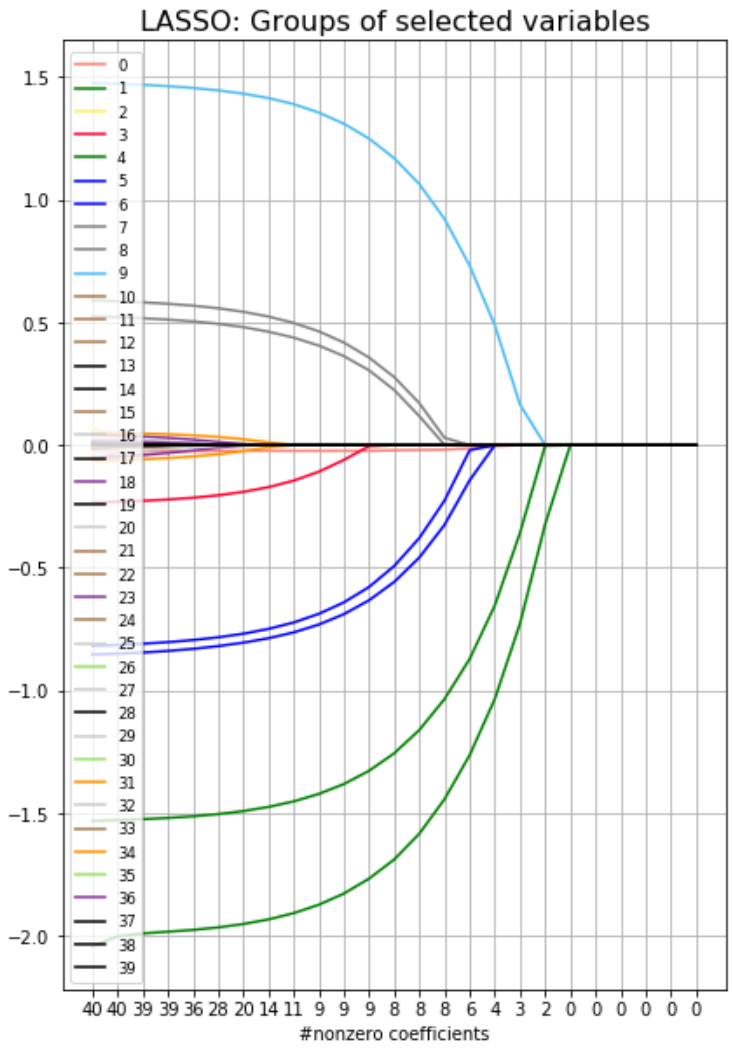
In [37]:

```
comment = "3 correlated vars at ~95% / dataset: 10/40 relevant variables, linear
ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

n_corr_feat = 3
cov = np.ones((n_corr_feat, n_corr_feat)) * .95
cov[np.diag_indices(n_corr_feat)] = 1

X_corr = rs.multivariate_normal(mean=np.zeros((n_corr_feat)), cov=cov, size=n_sa
mples)
X[:, 0:n_corr_feat] = X_corr
y = (true_coefs * X).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 1, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

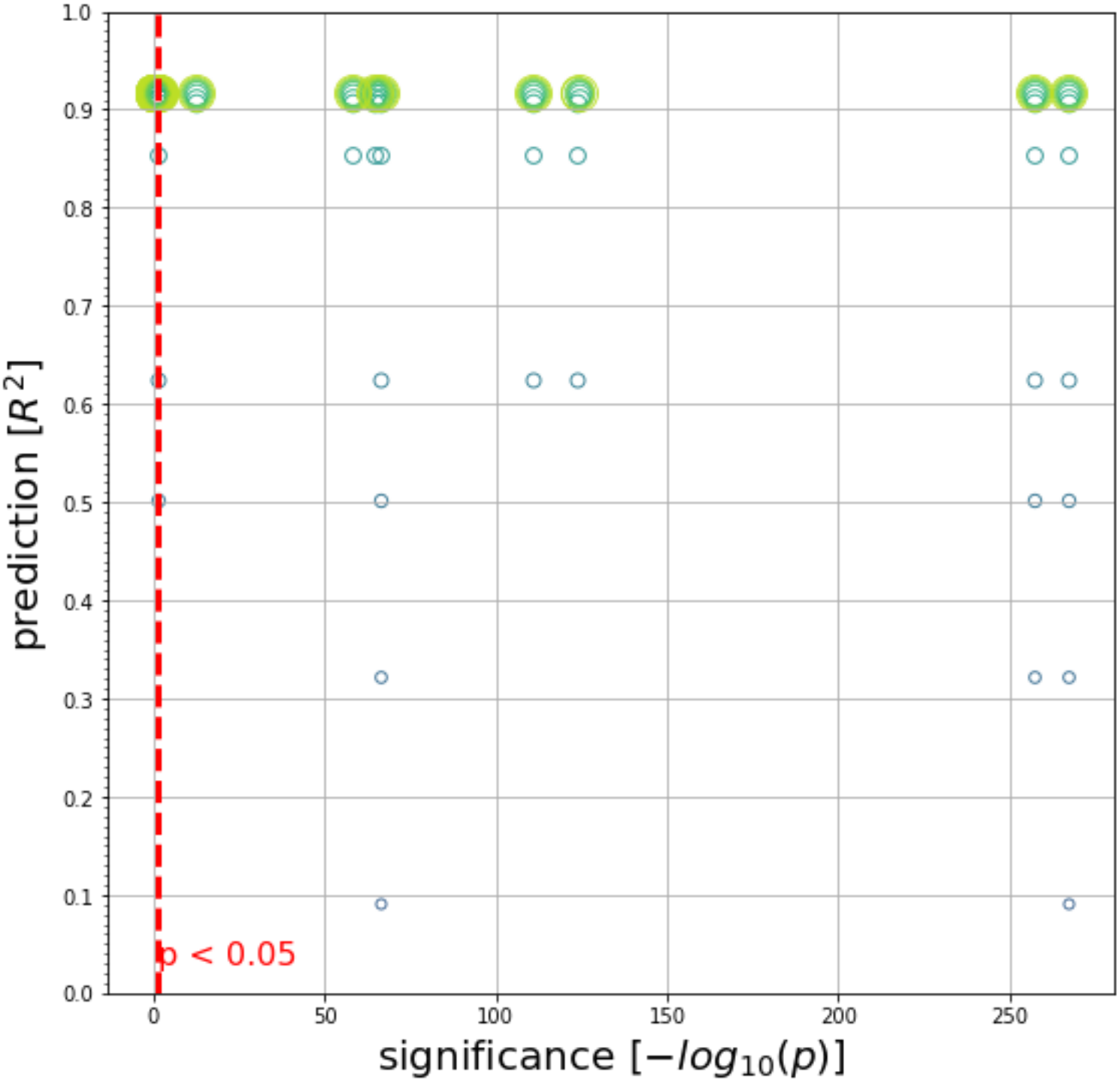
```
alpha: 0.0100 acc: 0.92 active_coefs: 40
alpha: 0.0133 acc: 0.92 active_coefs: 40
alpha: 0.0178 acc: 0.92 active_coefs: 39
alpha: 0.0237 acc: 0.92 active_coefs: 39
alpha: 0.0316 acc: 0.93 active_coefs: 36
alpha: 0.0422 acc: 0.93 active_coefs: 28
alpha: 0.0562 acc: 0.92 active_coefs: 20
alpha: 0.0750 acc: 0.92 active_coefs: 14
alpha: 0.1000 acc: 0.92 active_coefs: 11
alpha: 0.1334 acc: 0.92 active_coefs: 9
alpha: 0.1778 acc: 0.91 active_coefs: 9
alpha: 0.2371 acc: 0.89 active_coefs: 9
alpha: 0.3162 acc: 0.87 active_coefs: 8
alpha: 0.4217 acc: 0.82 active_coefs: 8
alpha: 0.5623 acc: 0.75 active_coefs: 8
alpha: 0.7499 acc: 0.65 active_coefs: 6
alpha: 1.0000 acc: 0.52 active_coefs: 4
alpha: 1.3335 acc: 0.31 active_coefs: 3
alpha: 1.7783 acc: 0.09 active_coefs: 2
alpha: 2.3714 acc: -0.01 active_coefs: 0
alpha: 3.1623 acc: -0.01 active_coefs: 0
alpha: 4.2170 acc: -0.01 active_coefs: 0
alpha: 5.6234 acc: -0.01 active_coefs: 0
alpha: 7.4989 acc: -0.01 active_coefs: 0
alpha: 10.0000 acc: -0.01 active_coefs: 0
40
267.297057962
```

257.077329698
123.953958013
110.714409952
66.0147751544
64.5552706964
58.1514318377
12.3603261008
2.63357566384
1.72093092201
1.23638678616
1.13629145185
1.05893116802
1.04180914503
0.777051875767
0.612476971947
0.529772284355
0.48723140239
0.47010409933
0.377690974262
0.331074148568
0.325426743346
0.323438450552
0.321005746635
0.320191060137
0.311990939412
0.299338839163
0.294236590827
0.287545269398
0.268444087709
0.25449472524
0.246914692328
0.145628291764
0.121930395595
0.101743752358
0.0519583947804
0.0384864960574
0.0313702428645
0.0288520482161
0.0132349880354
skipping 2
skipping 5
skipping 10
skipping 11
skipping 13
skipping 14
skipping 20
skipping 21
skipping 22
skipping 23
skipping 24


Linear regression
true betas
estimated betas
significant at p<0.05
input variables


LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients



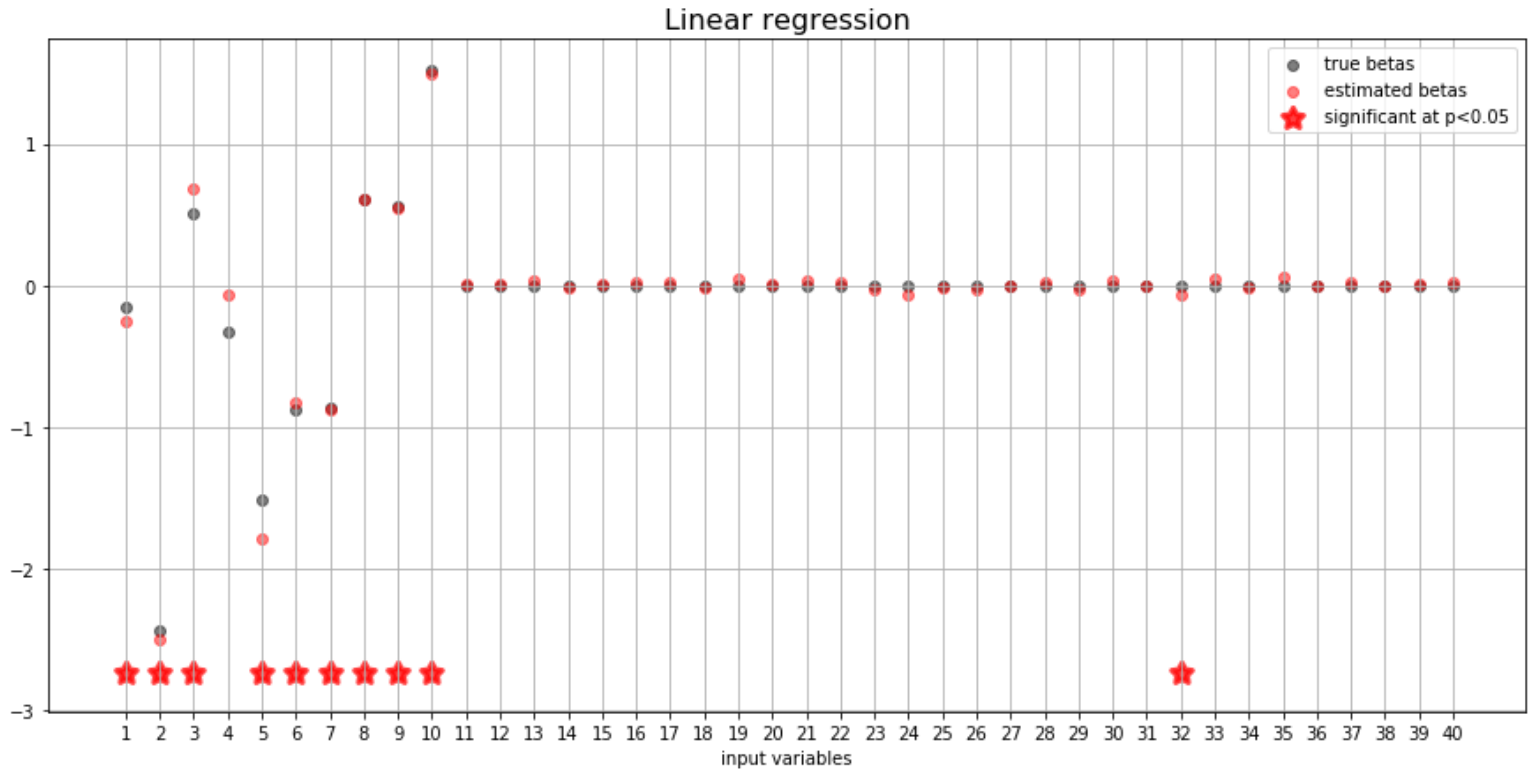
observation: 2/3 corr. variables are significant, different (!) 2/3 highly corr. variables are selected as predictive

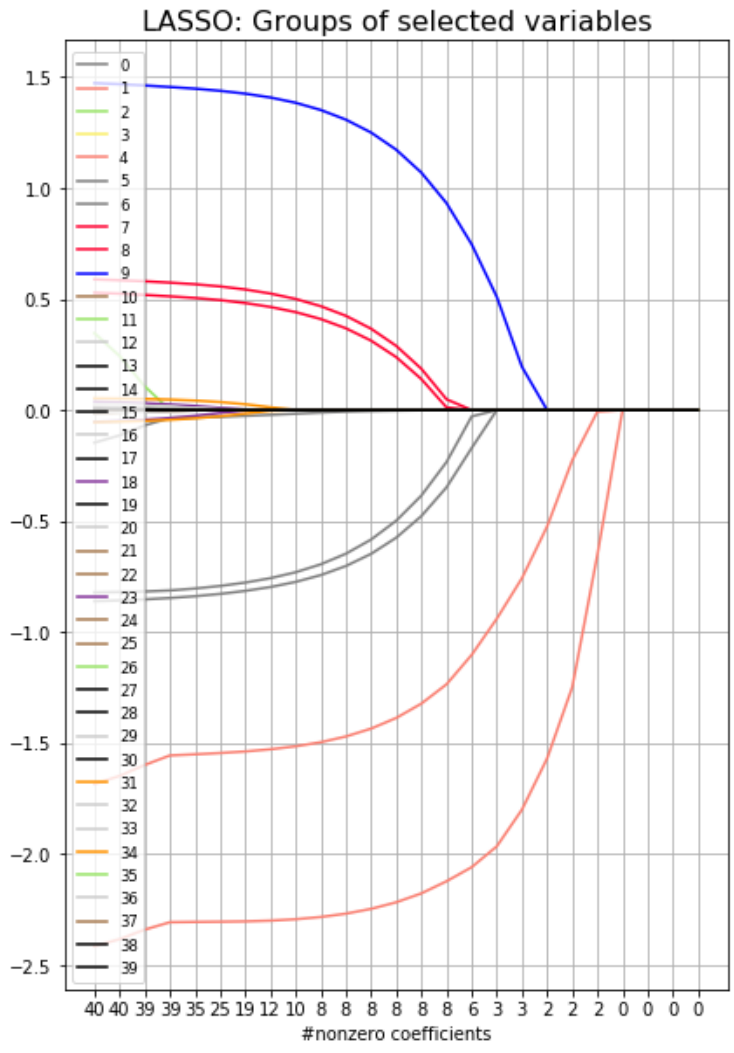
In [38]:

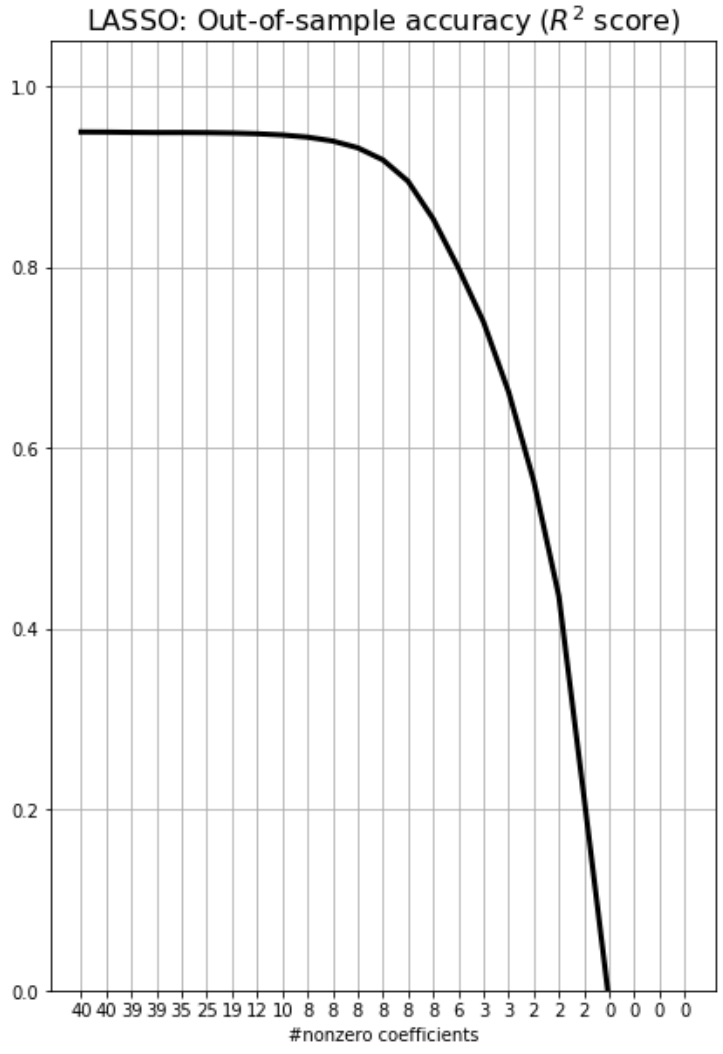
```
comment = "5 correlated vars at ~95% / dataset: 10/40 relevant variables, linear
ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

n_corr_feat = 5
cov = np.ones((n_corr_feat, n_corr_feat)) * .95
cov[np.diag_indices(n_corr_feat)] = 1

X_corr = rs.multivariate_normal(mean=np.zeros((n_corr_feat)), cov=cov, size=n_sa
mples)
X[:, 0:n_corr_feat] = X_corr
y = (true_coefs * X).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 1, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

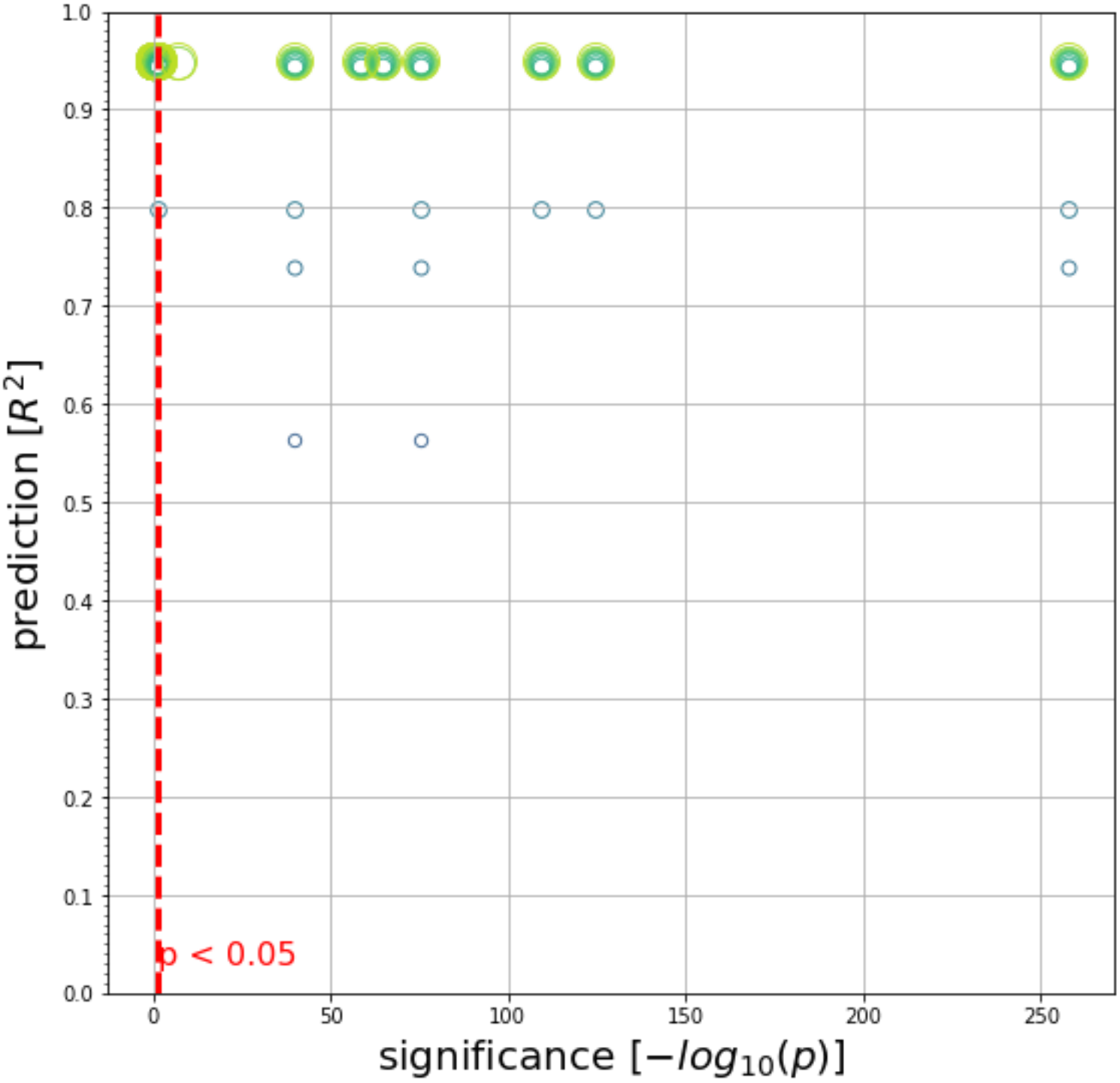
```
alpha: 0.0100 acc: 0.94 active_coefs: 40
alpha: 0.0133 acc: 0.94 active_coefs: 40
alpha: 0.0178 acc: 0.94 active_coefs: 39
alpha: 0.0237 acc: 0.95 active_coefs: 39
alpha: 0.0316 acc: 0.95 active_coefs: 35
alpha: 0.0422 acc: 0.95 active_coefs: 25
alpha: 0.0562 acc: 0.95 active_coefs: 19
alpha: 0.0750 acc: 0.95 active_coefs: 12
alpha: 0.1000 acc: 0.94 active_coefs: 10
alpha: 0.1334 acc: 0.94 active_coefs: 8
alpha: 0.1778 acc: 0.94 active_coefs: 8
alpha: 0.2371 acc: 0.93 active_coefs: 8
alpha: 0.3162 acc: 0.92 active_coefs: 8
alpha: 0.4217 acc: 0.89 active_coefs: 8
alpha: 0.5623 acc: 0.84 active_coefs: 8
alpha: 0.7499 acc: 0.78 active_coefs: 6
alpha: 1.0000 acc: 0.71 active_coefs: 3
alpha: 1.3335 acc: 0.61 active_coefs: 3
alpha: 1.7783 acc: 0.51 active_coefs: 2
alpha: 2.3714 acc: 0.40 active_coefs: 2
alpha: 3.1623 acc: 0.21 active_coefs: 2
alpha: 4.2170 acc: -0.01 active_coefs: 0
alpha: 5.6234 acc: -0.01 active_coefs: 0
alpha: 7.4989 acc: -0.01 active_coefs: 0
alpha: 10.0000 acc: -0.01 active_coefs: 0
40
257.926430562
```

```
124.531244884
108.997057085
75.3178395861
64.4613934571
58.3681547567
39.705434732
6.76541315005
1.38440748688
1.3295185464
1.26029580981
1.18641696721
0.872586989065
0.822187337005
0.642076888446
0.583364864781
0.532313902143
0.37487020594
0.369299326617
0.365493311854
0.363450021274
0.330079475045
0.321739404554
0.31357492096
0.306167086407
0.261738855109
0.257064875627
0.201814622868
0.20029789534
0.193021207977
0.182120408073
0.170821703447
0.17026434987
0.0927879134536
0.0813540679751
0.0781248126016
0.044205632203
0.0314982467255
0.0249758350177
0.00606032877281
skipping 1
skipping 4
skipping 10
skipping 11
skipping 12
skipping 13
skipping 14
skipping 17
skipping 19
skipping 20
skipping 22
skipping 23
skipping 24
```


Linear regression
true betas
estimated betas
significant at p<0.05
input variables


LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients



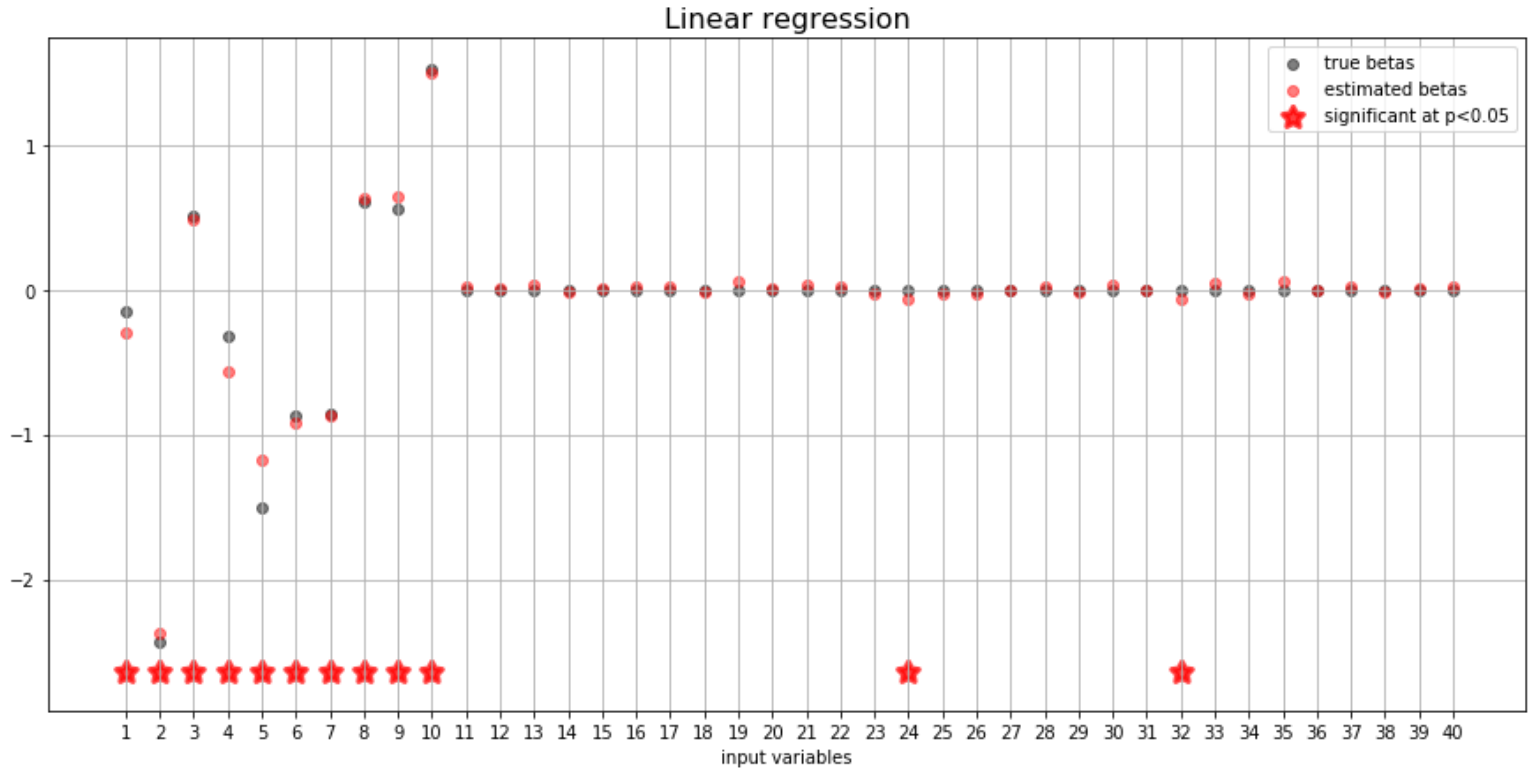
observation: 4/5 highly corr. variables are significant, different (!) 3/5 highly corr. variables are selected as predictive

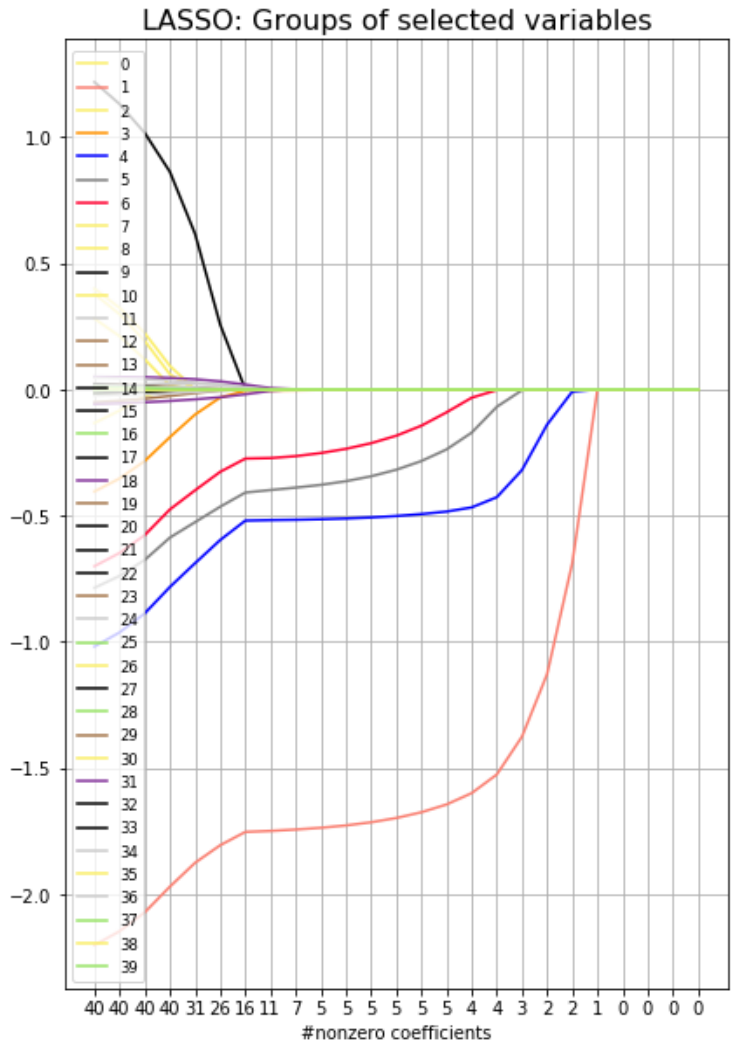
In [39]:

```
comment = "10 correlated vars at ~95% / dataset: 10/40 relevant variables, linea
r ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 10
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

n_corr_feat = 10
cov = np.ones((n_corr_feat, n_corr_feat)) * .95
cov[np.diag_indices(n_corr_feat)] = 1

X_corr = rs.multivariate_normal(mean=np.zeros((n_corr_feat)), cov=cov, size=n_sa
mples)
X[:, 0:n_corr_feat] = X_corr
y = (true_coefs * X).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 1, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

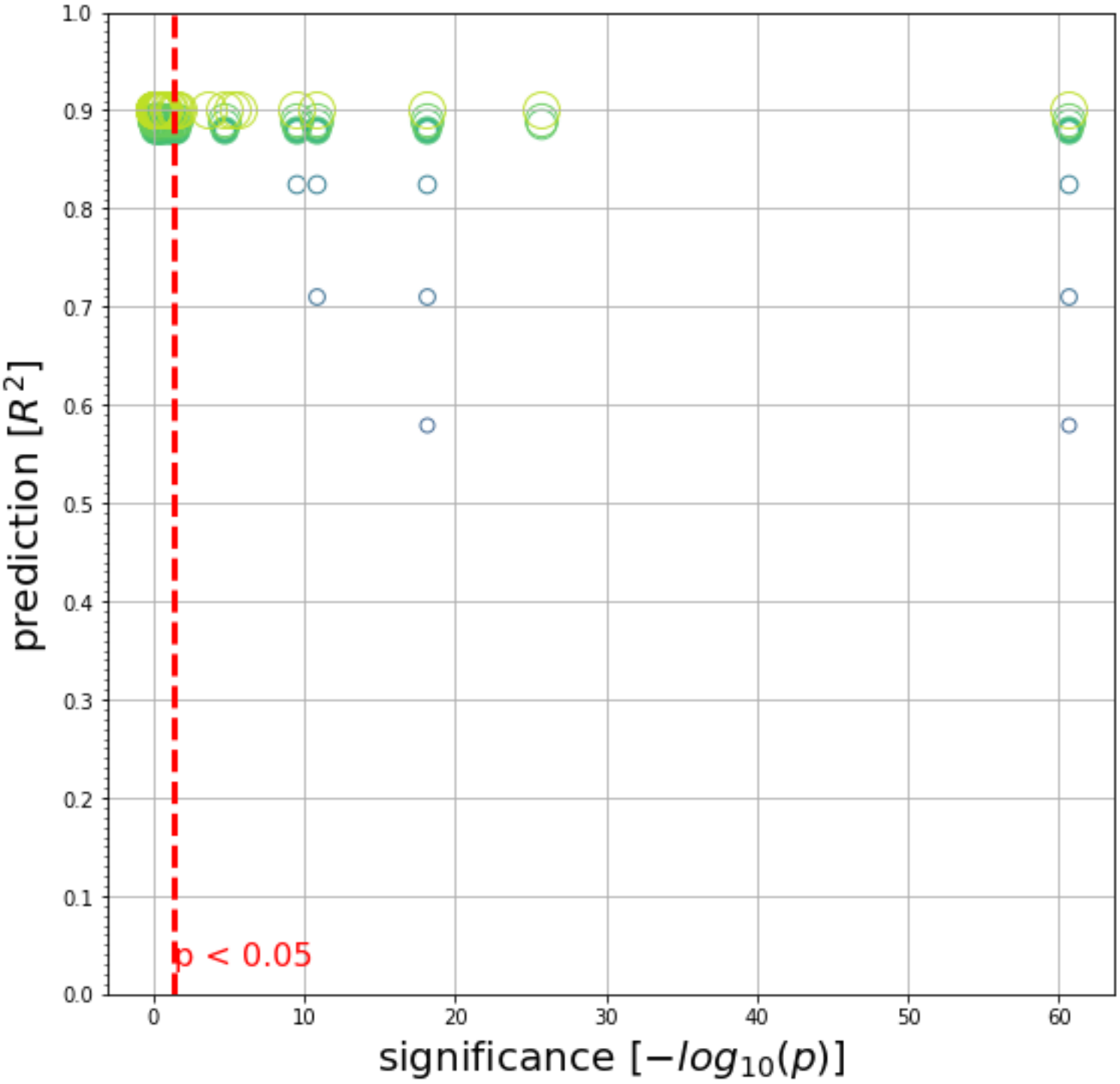
```
alpha: 0.0100 acc: 0.90 active_coefs: 40
alpha: 0.0133 acc: 0.90 active_coefs: 40
alpha: 0.0178 acc: 0.90 active_coefs: 40
alpha: 0.0237 acc: 0.90 active_coefs: 40
alpha: 0.0316 acc: 0.90 active_coefs: 31
alpha: 0.0422 acc: 0.90 active_coefs: 26
alpha: 0.0562 acc: 0.90 active_coefs: 16
alpha: 0.0750 acc: 0.90 active_coefs: 11
alpha: 0.1000 acc: 0.90 active_coefs: 7
alpha: 0.1334 acc: 0.90 active_coefs: 5
alpha: 0.1778 acc: 0.90 active_coefs: 5
alpha: 0.2371 acc: 0.90 active_coefs: 5
alpha: 0.3162 acc: 0.90 active_coefs: 5
alpha: 0.4217 acc: 0.89 active_coefs: 5
alpha: 0.5623 acc: 0.88 active_coefs: 5
alpha: 0.7499 acc: 0.87 active_coefs: 4
alpha: 1.0000 acc: 0.83 active_coefs: 4
alpha: 1.3335 acc: 0.76 active_coefs: 3
alpha: 1.7783 acc: 0.63 active_coefs: 2
alpha: 2.3714 acc: 0.39 active_coefs: 2
alpha: 3.1623 acc: -0.00 active_coefs: 1
alpha: 4.2170 acc: -0.00 active_coefs: 0
alpha: 5.6234 acc: -0.00 active_coefs: 0
alpha: 7.4989 acc: -0.00 active_coefs: 0
alpha: 10.0000 acc: -0.00 active_coefs: 0
40
60.6621438734
```

```
25.6344653953
18.044090156
10.8009979743
9.4966963324
5.58853513924
5.27526463266
4.70238054406
3.56092827632
1.5709739911
1.51908251505
1.31671870707
1.13331713008
0.942980337959
0.671334026061
0.527527958679
0.516166975827
0.515060297813
0.45746543492
0.454561489988
0.398878294933
0.390650496877
0.35151437345
0.328316074233
0.304453208307
0.286492169873
0.240840246415
0.23451903018
0.233926076692
0.204222907567
0.199829895486
0.199651203146
0.152414067698
0.14480598684
0.128502593025
0.0738539509642
0.058806834594
0.0433177491567
0.0175832712395
0.00053283110911
skipping 1
skipping 2
skipping 3
skipping 10
skipping 11
skipping 12
skipping 13
skipping 14
skipping 16
skipping 19
skipping 22
skipping 23
skipping 24
```


Linear regression
true betas
estimated betas
significant at p<0.05
input variables


LASSO: Groups of selected variables
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
#nonzero coefficients



observation: 10/10 highly corr. variables are significant, different (!) 4/10 highly corr. variables are selected as predictive with reasonable accuracy around R2=0.9 -> a few representative variables are easily found by Lasso without false positive while sign. testing yields 2 false positives -> corelation among variables appears to inflate Type-1 error

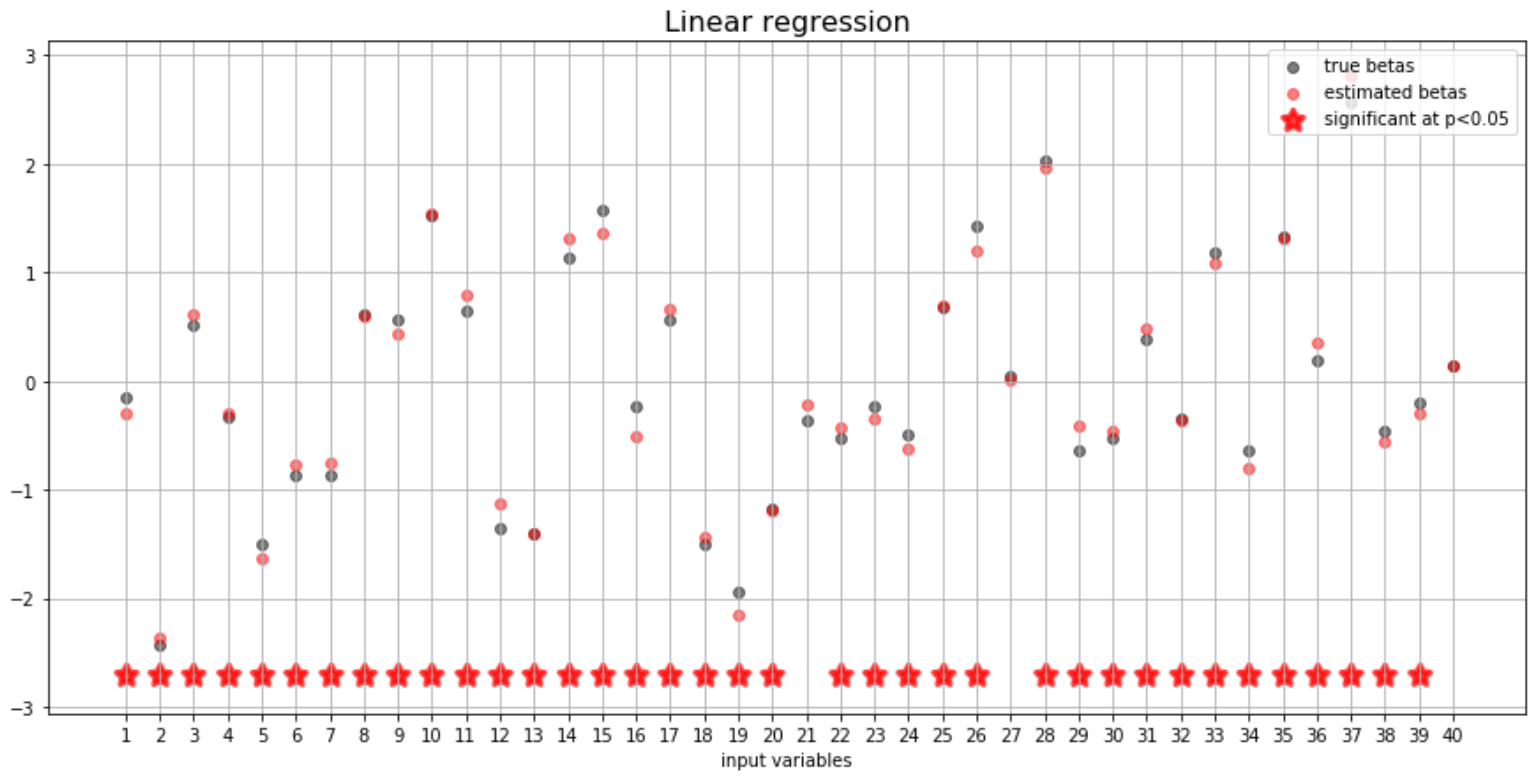
In [40]:

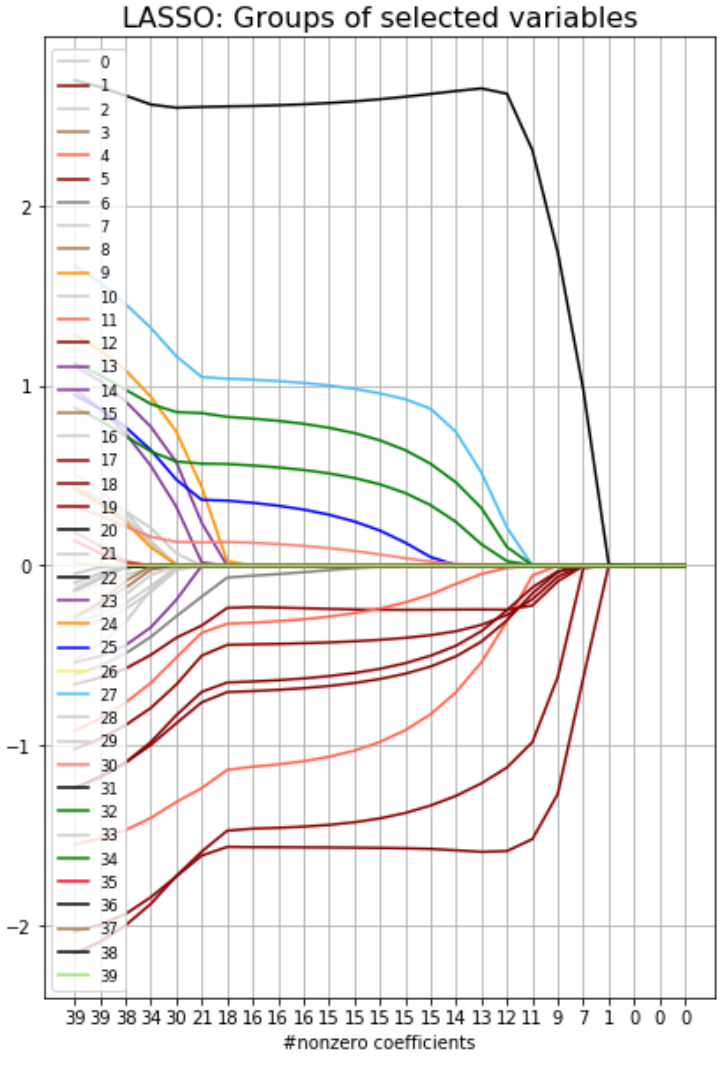
```
comment = "4 groups of 10 correlated vars at ~95% / dataset: 40/40 relevant vari
ables, linear ground truth, some noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 40
epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
#X = rs.randn(n_samples, n_feat)

n_corr_feat = 10
cov = np.ones((n_corr_feat, n_corr_feat)) * .95
cov[np.diag_indices(n_corr_feat)] = 1
X_corr = rs.multivariate_normal(mean=np.zeros((n_corr_feat)), cov=cov, size=n_sa
mples)
X[:, 0:10] = X_corr

n_corr_feat = 10
cov = np.ones((n_corr_feat, n_corr_feat)) * .95
cov[np.diag_indices(n_corr_feat)] = 1
X_corr = rs.multivariate_normal(mean=np.zeros((n_corr_feat)), cov=cov, size=n_sa
mples)
X[:, 10:20] = X_corr

n_corr_feat = 10
cov = np.ones((n_corr_feat, n_corr_feat)) * .95
cov[np.diag_indices(n_corr_feat)] = 1
X_corr = rs.multivariate_normal(mean=np.zeros((n_corr_feat)), cov=cov, size=n_sa
mples)
X[:, 20:30] = X_corr

n_corr_feat = 10
cov = np.ones((n_corr_feat, n_corr_feat)) * .95
cov[np.diag_indices(n_corr_feat)] = 1
X_corr = rs.multivariate_normal(mean=np.zeros((n_corr_feat)), cov=cov, size=n_sa
mples)
X[:, 30:40] = X_corr

y = (true_coefs * X).sum(axis=1) + epsilon
C_grid = np.logspace(-2, 1, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

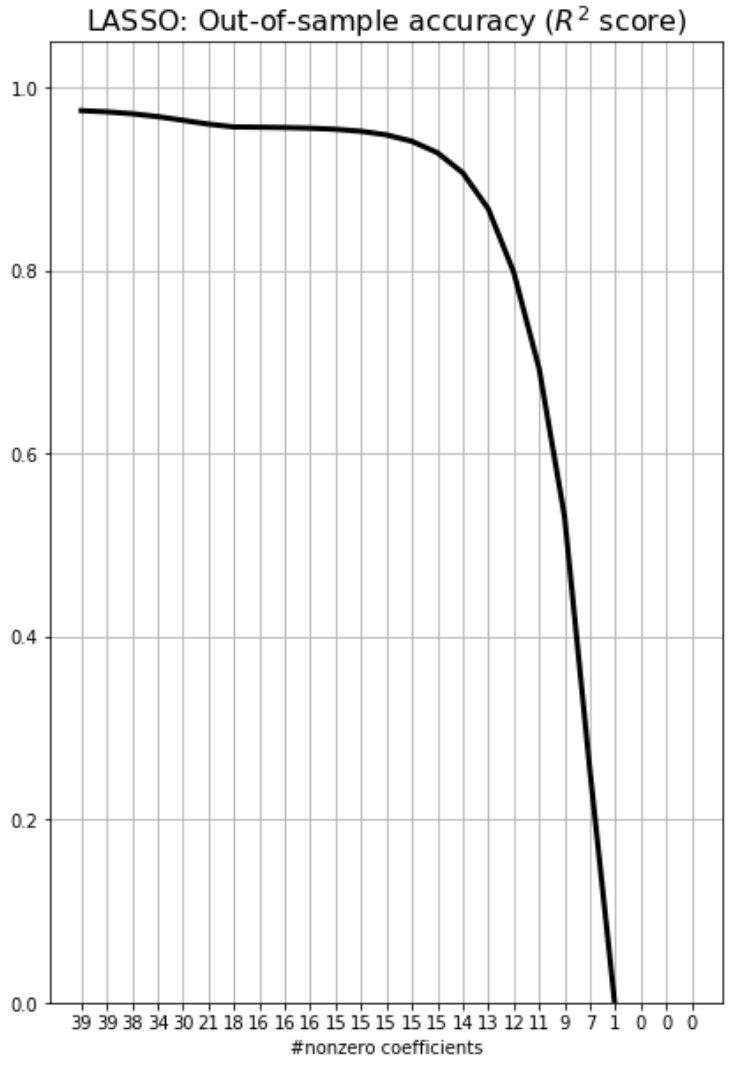
```
alpha: 0.0100 acc: 0.97 active_coefs: 39
alpha: 0.0133 acc: 0.97 active_coefs: 39
alpha: 0.0178 acc: 0.97 active_coefs: 38
alpha: 0.0237 acc: 0.97 active_coefs: 34
alpha: 0.0316 acc: 0.96 active_coefs: 30
alpha: 0.0422 acc: 0.96 active_coefs: 21
```

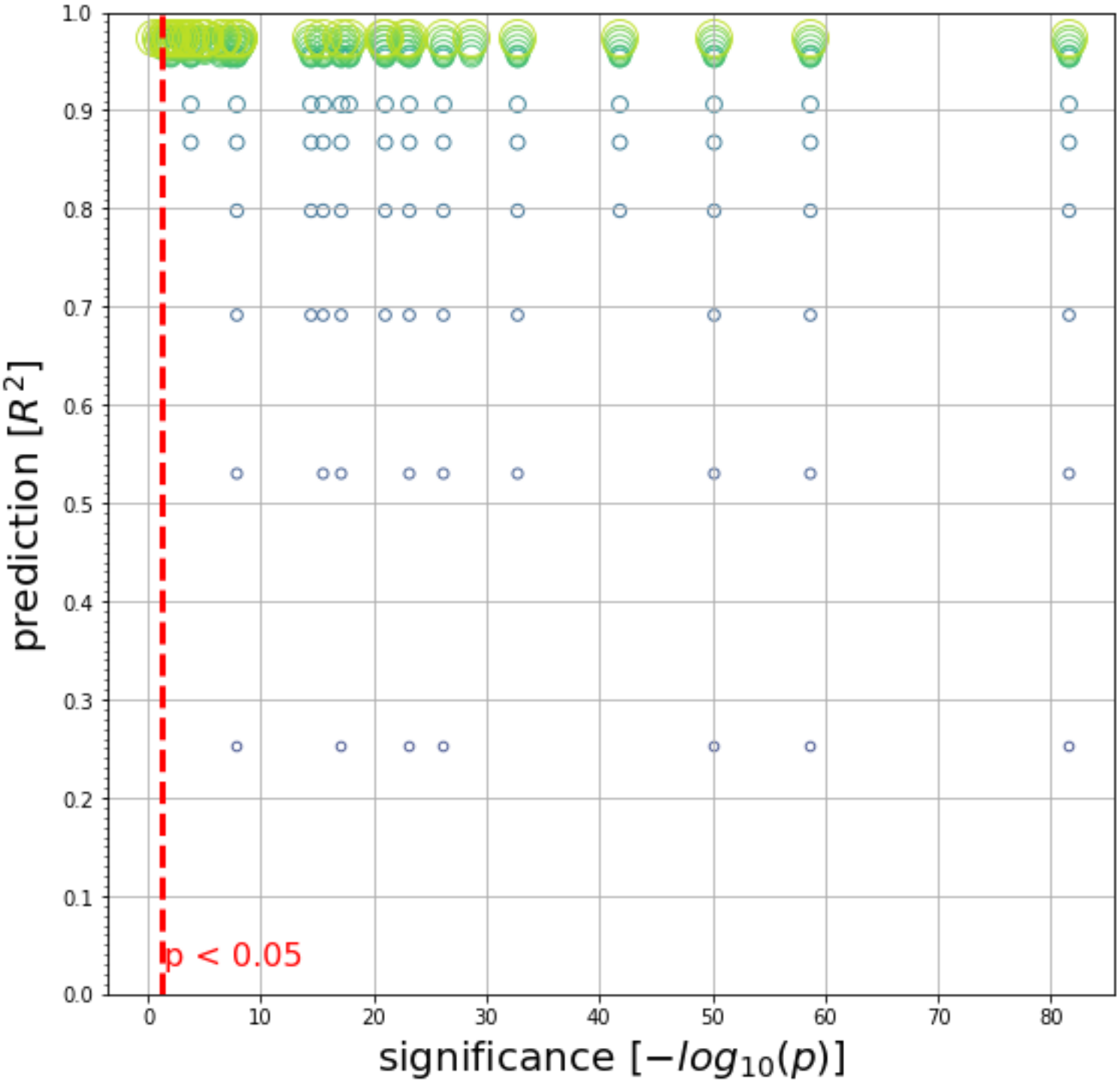
```
alpha: 0.0562 acc: 0.96 active_coefs: 18
alpha: 0.0750 acc: 0.96 active_coefs: 16
alpha: 0.1000 acc: 0.96 active_coefs: 16
alpha: 0.1334 acc: 0.96 active_coefs: 16
alpha: 0.1778 acc: 0.95 active_coefs: 15
alpha: 0.2371 acc: 0.95 active_coefs: 15
alpha: 0.3162 acc: 0.94 active_coefs: 15
alpha: 0.4217 acc: 0.94 active_coefs: 15
alpha: 0.5623 acc: 0.92 active_coefs: 15
alpha: 0.7499 acc: 0.90 active_coefs: 14
alpha: 1.0000 acc: 0.86 active_coefs: 13
alpha: 1.3335 acc: 0.79 active_coefs: 12
alpha: 1.7783 acc: 0.69 active_coefs: 11
alpha: 2.3714 acc: 0.54 active_coefs: 9
alpha: 3.1623 acc: 0.29 active_coefs: 7
alpha: 4.2170 acc: -0.00 active_coefs: 1
alpha: 5.6234 acc: -0.00 active_coefs: 0
alpha: 7.4989 acc: -0.00 active_coefs: 0
alpha: 10.0000 acc: -0.00 active_coefs: 0
40
81.5686355203
58.6753218022
50.0765350181
41.8061746787
32.6343114648
28.6260353174
26.1839299049
23.0200828076
22.6768253623
21.0208291367
20.8076278124
17.6690442253
16.975203842
15.379432348
14.3841389163
8.02913430806
7.9107278857
7.87871232811
7.3431442434
6.54663563282
6.34458270121
5.00539018141
4.97859099563
4.71238681166
4.56811960159
3.76896169615
3.74942146693
3.37186480891
2.7923976555
2.75676190176
2.56485061035
2.17410595426
1.94214277231
```

```
1.92139368417
1.62265205629
1.52795148566
1.49487709407
1.07161445818
0.492519587797
0.0303378711099
skipping 1
skipping 8
skipping 9
skipping 11
skipping 12
skipping 13
skipping 14
skipping 23
skipping 24
```




LASSO: Groups of selected variables
2
1
0
−1
−2
39 39 38 34 30 21 18 16 16 16 15 15 15 15 15 14 13 12 11 9 7 1 0 0 0
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
1.0
0.8
0.6
0.4
0.2
0.0
39 39 38 34 30 21 18 16 16 16 15 15 15 15 15 14 13 12 11 9 7 1 0 0 0
#nonzero coefficients



observation: significance yields 3 false negatives; Lasso yields stable R2>0.9 with roughly 16/40 active variables capturing 4 real sources of variation

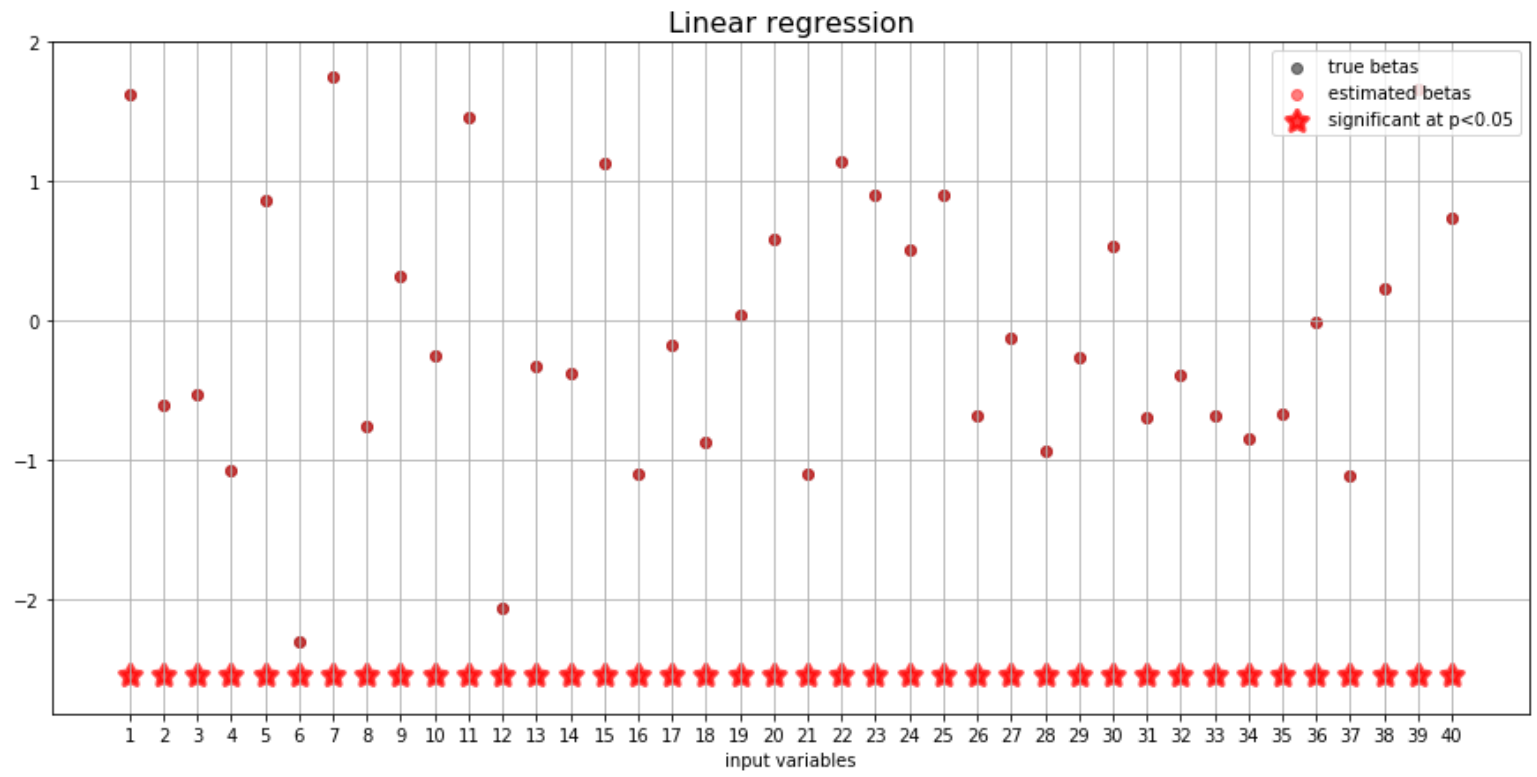
In [41]:

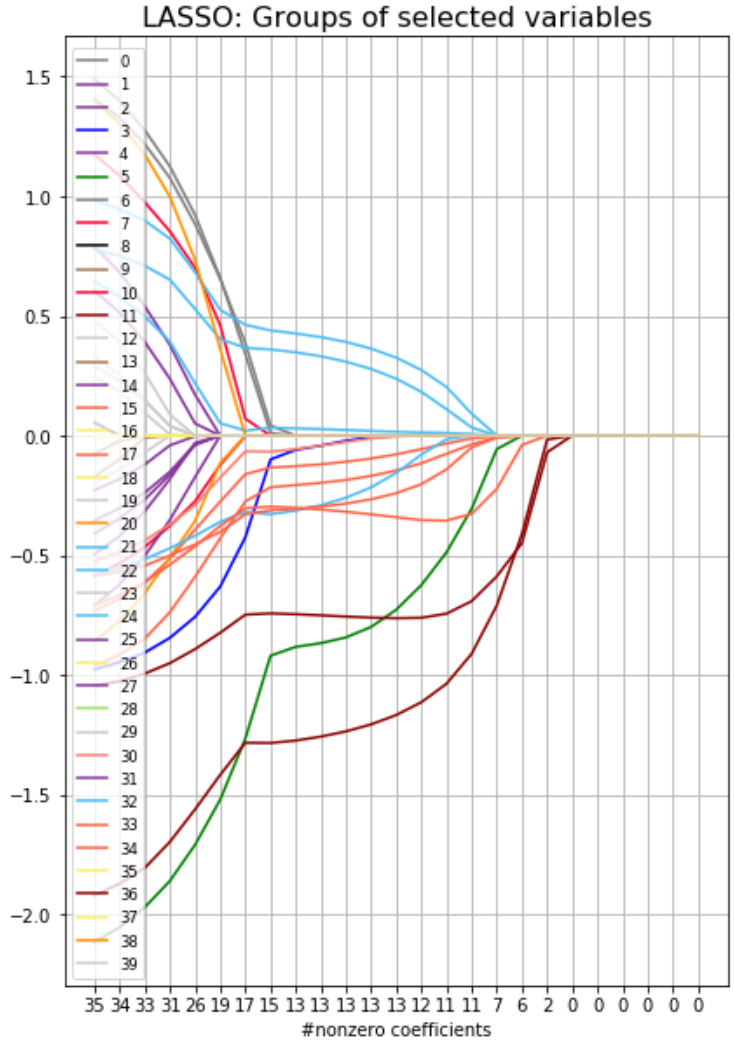
```
comment = "4 groups of 10 correlated vars at ~95% / dataset: 40/40 relevant vari
ables, linear ground truth, NO noise"
rs = np.random.RandomState(1)
n_samples = 1000
n_feat = 40
n_feat_relevant = 40
#epsilon = rs.randn(n_samples)
true_coefs = rs.randn(n_feat)
true_coefs[n_feat_relevant:] = 0
X = rs.randn(n_samples, n_feat)

n_corr_feat = 10
cov = np.ones((n_corr_feat, n_corr_feat)) * .95
cov[np.diag_indices(n_corr_feat)] = 1
X_corr = rs.multivariate_normal(mean=np.zeros((n_corr_feat)), cov=cov, size=n_sa
mples)
X[:, 0:10] = X_corr

n_corr_feat = 10
cov = np.ones((n_corr_feat, n_corr_feat)) * .95
cov[np.diag_indices(n_corr_feat)] = 1
X_corr = rs.multivariate_normal(mean=np.zeros((n_corr_feat)), cov=cov, size=n_sa
mples)
X[:, 10:20] = X_corr

n_corr_feat = 10
cov = np.ones((n_corr_feat, n_corr_feat)) * .95
cov[np.diag_indices(n_corr_feat)] = 1
X_corr = rs.multivariate_normal(mean=np.zeros((n_corr_feat)), cov=cov, size=n_sa
mples)
X[:, 20:30] = X_corr

n_corr_feat = 10
cov = np.ones((n_corr_feat, n_corr_feat)) * .95
cov[np.diag_indices(n_corr_feat)] = 1
X_corr = rs.multivariate_normal(mean=np.zeros((n_corr_feat)), cov=cov, size=n_sa
mples)
X[:, 30:40] = X_corr

#y = (true_coefs * X).sum(axis=1) + epsilon
y = (true_coefs * X).sum(axis=1)
C_grid = np.logspace(-2, 1, 25)

run_lasso(X, y, comment, n_samples, n_feat,
          n_feat_relevant, C_grid=C_grid)
```

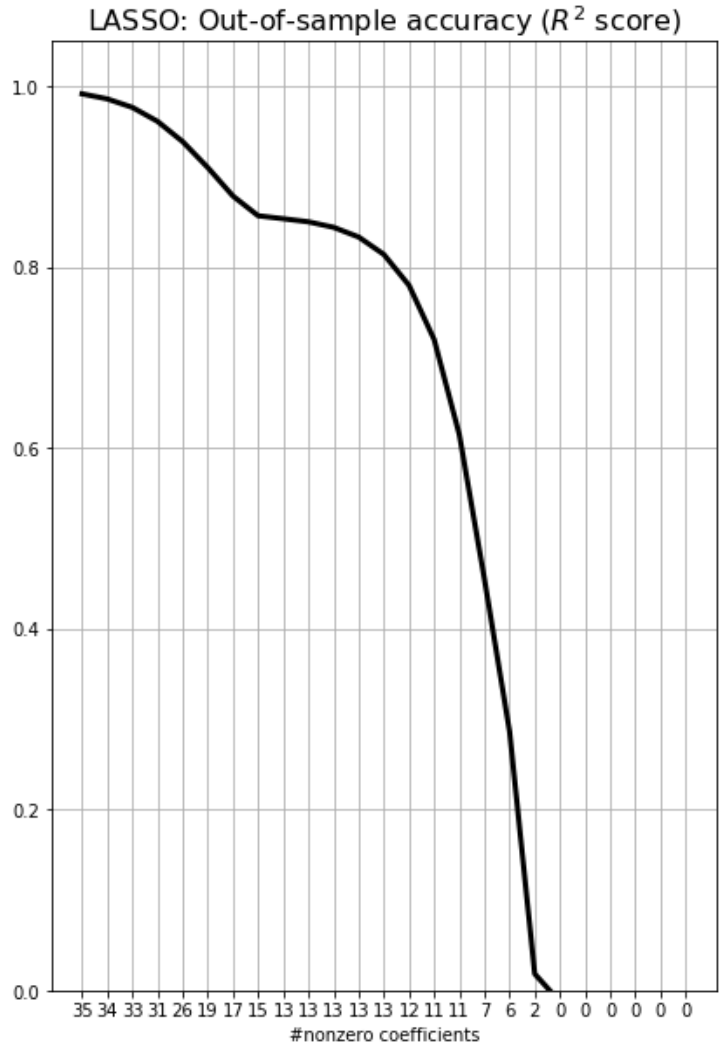
```
alpha: 0.0100 acc: 0.99 active_coefs: 35
alpha: 0.0133 acc: 0.98 active_coefs: 34
alpha: 0.0178 acc: 0.96 active_coefs: 33
alpha: 0.0237 acc: 0.94 active_coefs: 31
alpha: 0.0316 acc: 0.91 active_coefs: 26
```

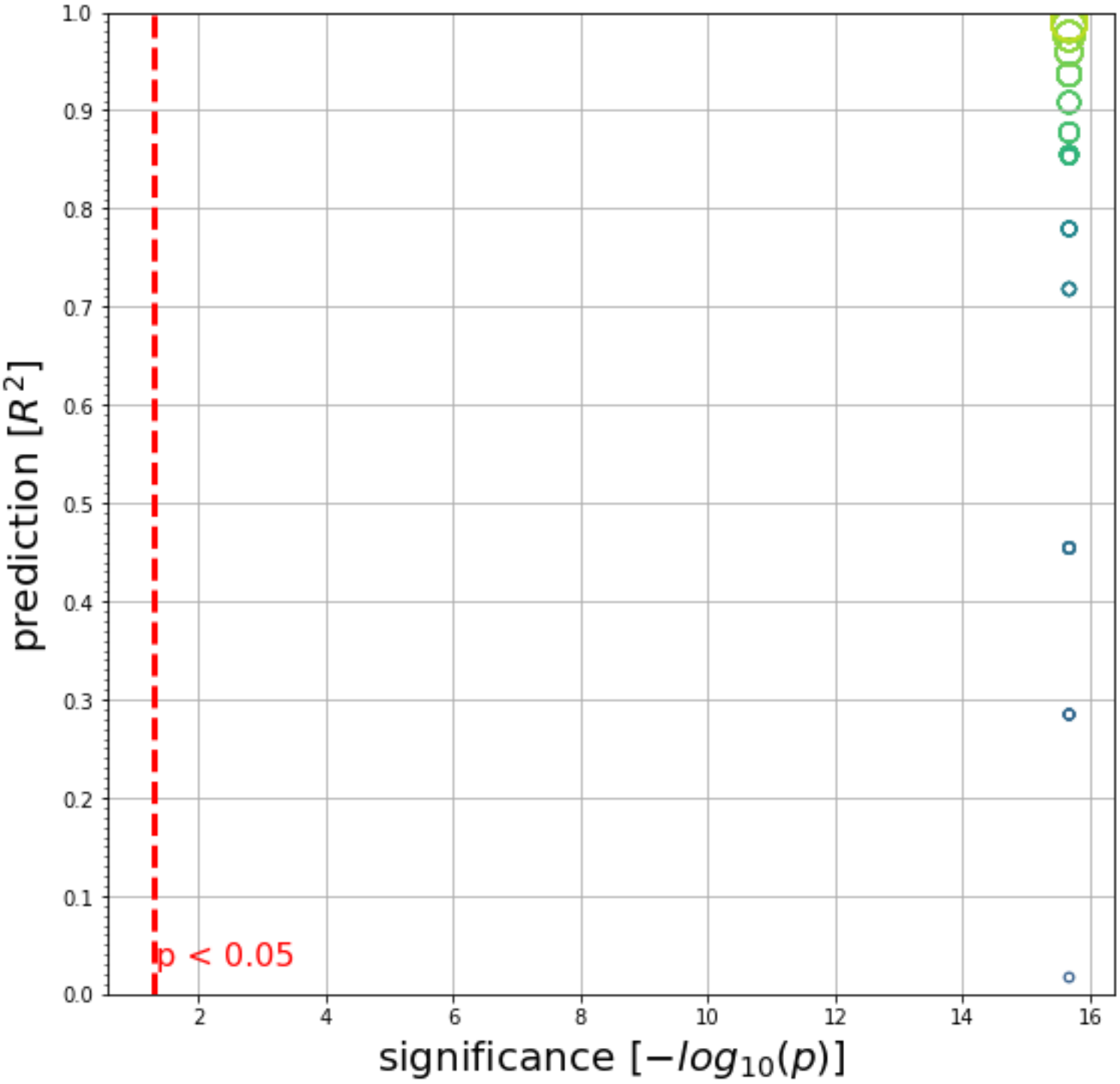
```
alpha: 0.0422 acc: 0.88 active_coefs: 19
alpha: 0.0562 acc: 0.83 active_coefs: 17
alpha: 0.0750 acc: 0.80 active_coefs: 15
alpha: 0.1000 acc: 0.80 active_coefs: 13
alpha: 0.1334 acc: 0.80 active_coefs: 13
alpha: 0.1778 acc: 0.79 active_coefs: 13
alpha: 0.2371 acc: 0.78 active_coefs: 13
alpha: 0.3162 acc: 0.76 active_coefs: 13
alpha: 0.4217 acc: 0.73 active_coefs: 12
alpha: 0.5623 acc: 0.67 active_coefs: 11
alpha: 0.7499 acc: 0.58 active_coefs: 11
alpha: 1.0000 acc: 0.44 active_coefs: 7
alpha: 1.3335 acc: 0.29 active_coefs: 6
alpha: 1.7783 acc: 0.04 active_coefs: 2
alpha: 2.3714 acc: -0.01 active_coefs: 0
alpha: 3.1623 acc: -0.01 active_coefs: 0
alpha: 4.2170 acc: -0.01 active_coefs: 0
alpha: 5.6234 acc: -0.01 active_coefs: 0
alpha: 7.4989 acc: -0.01 active_coefs: 0
alpha: 10.0000 acc: -0.01 active_coefs: 0
40
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
```

```
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
15.6535597745
skipping 9
skipping 10
skipping 11
skipping 12
skipping 15
skipping 20
skipping 21
skipping 22
skipping 23
skipping 24
```




LASSO: Groups of selected variables
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
35 34 33 31 26 19 17 15 13 13 13 13 13 12 11 11 7 6 2 0 0 0 0 0 0
#nonzero coefficients


LASSO: Out-of-sample accuracy (R^2 score)
1.0
0.8
0.6
0.4
0.2
0.0
35 34 33 31 26 19 17 15 13 13 13 13 13 12 11 11 7 6 2 0 0 0 0 0 0
#nonzero coefficients


prediction [R^2]
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
p < 0.05
2
4
6
8
10
12
14
16
significance [$-log_{10}(p)$]

Summary

In [43]:

```
# print summary
for sim in simulations:
    pval = sim['pvals'].min()
    n_pval = np.sum(sim['pvals'] <= 0.05)
    r2 = sim['acc_list'].max()
    print(sim['comment'])
    print(pval >= 0.05, pval, n_pval, r2)
    print("")

# plot 1: r2 scores given the p-value
plt.figure(figsize=(6, 6))
for sim in simulations:
    n_rel = sim['n_feat_relevant']
    if n_rel == 0:
        continue
    pval = sim['pvals'][:n_rel].min()
    n_pval = np.sum(sim['pvals'][:n_rel] <= 0.05)
    r2 = sim['acc_list'].max()
    plt.scatter(-np.log10(pval), r2)

plt.xlabel(r'significance [$-log_{10}(p)$]', fontsize=20, fontweight=150)
plt.ylabel(r'prediction [$R^2$]', fontsize=20, fontweight=150)

plt.grid(True)
ax = plt.gca()
plt.axvline(
    -np.log10(0.05), color='red', linestyle='--', linewidth=3)
plt.annotate('p < 0.05', xy=(-np.log10(0.045), 0.03), color='red', fontsize=14)
ax.set_yticks([0., 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1])
ax.set_yticks(np.arange(0.01, 1, 0.01), minor=True);

# plot 2: r2 scores given the number of significant p-values relative
# to number relevant features
plt.figure(figsize=(6, 6))
for sim in simulations:
    n_rel = sim['n_feat_relevant']
    if n_rel == 0:
        continue
    pval = sim['pvals'][:n_rel].min()
    n_pval = np.sum(sim['pvals'][:n_rel] <= 0.05)
    r2 = sim['acc_list'].max()
    plt.scatter(n_pval / n_rel, r2)

plt.xlabel(r'Proportion of p-values < 0.05', fontsize=20, fontweight=150)
plt.ylabel(r'prediction [$R^2$]', fontsize=20, fontweight=150)
plt.grid(True)
ax = plt.gca()
ax.set_yticks([0., 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1])
ax.set_yticks(np.arange(0.01, 1, 0.01), minor=True);
```

```
dataset: 10/40 variables relevant, linear ground truth, some noise
```

```
False 1.26302476958e-18 11 0.722118442333

dataset: 20/40 variables relevant, linear ground truth, some noise
False 6.30761429734e-20 17 0.861085349967

dataset: 30/40 variables relevant, linear ground truth, some noise
False 9.04688139435e-28 25 0.932098782862

dataset: 40/40 variables relevant, linear ground truth, some noise
False 9.04688139435e-28 34 0.944988438592

dataset: 10/40 variables relevant, linear ground truth, some noise
False 6.25407738595e-267 11 0.924743235144

dataset: 20/40 variables relevant, linear ground truth, some noise
False 1.98379691076e-307 21 0.964327764623

dataset: 30/40 variables relevant, linear ground truth, some noise
False 1.98379691076e-307 30 0.972198689331

dataset: 40/40 variables relevant, linear ground truth, some noise
False 1.98379691076e-307 39 0.978898424293

dataset: 10/40 variables relevant, linear ground truth, some noise
False 2.22044604925e-16 11 0.999998653576

dataset: 20/40 variables relevant, linear ground truth, some noise
False 2.22044604925e-16 20 0.999997887566

dataset: 30/40 variables relevant, linear ground truth, some noise
False 2.22044604925e-16 30 0.999995333766

dataset: 40/40 variables relevant, linear ground truth, some noise
False 2.22044604925e-16 40 0.999993155536

abs / dataset: 10/40 relevant variables, including 5 pathological on
es, linear ground truth, some noise
False 3.64460597918e-27 5 0.43249201706

square root / dataset: 10/40 relevant variables, including 5 patholo
gical ones, linear ground truth, some noise
False 1.51921318069e-281 11 0.863204401872

logn / dataset: 10/40 relevant variables, including 5 pathological o
nes, linear ground truth, some noise
False 6.77553387848e-31 10 0.187750767237

exp / dataset: 10/40 relevant variables, including 5 pathological on
es, linear ground truth, some noise only monotone transformation
False 6.10445389692e-77 9 0.603196772268

dataset: 1/40 relevant variables linear ground truth + some noise
False 3.17365709646e-06 4 0.351770845517
```

dataset: 0/40 relevant variables linear ground truth + some noise
False 0.00618545389122 4 -0.232977557021

1/x / dataset: 10/40 relevant variables, including 5 pathological on
es, linear ground truth, some noise opposite effect of log roughly;
small values become gigantic
True 0.0584401244368 0 -0.0120716048324

x^2 / dataset: 10/40 relevant variables, including 5 pathological on
es, linear ground truth, some noise cube -> keep monotony; otherwise
not ->
False 2.18873539944e-14 6 0.13808398418

x^3 / dataset: 10/40 relevant variables, including 5 pathological on
es, linear ground truth, some noise
False 5.71336336946e-151 10 0.604666552499

x^4 / dataset: 10/40 relevant variables, including 5 pathological on
es, linear ground truth, some noise
False 0.00533179687848 2 -0.0260270483268

x^5 / dataset: 10/40 relevant variables, including 5 pathological on
es, linear ground truth, some noise
False 5.79269869944e-50 6 0.245828888167

3 correlated vars at ~50% / dataset: 10/40 relevant variables, linea
r ground truth, some noise
False 5.04593948679e-268 11 0.918941812044

5 correlated vars at ~50% / dataset: 10/40 relevant variables, linea
r ground truth, some noise
False 2.82151466631e-314 11 0.940517588658

10 correlated vars at ~50% / dataset: 10/40 relevant variables, line
ar ground truth, some noise
False 2.48505320757e-314 12 0.919058355574

3 correlated vars at ~95% / dataset: 10/40 relevant variables, linea
r ground truth, some noise
False 5.04593948679e-268 10 0.9177647

5 correlated vars at ~95% / dataset: 10/40 relevant variables, linea
r ground truth, some noise
False 1.18459375281e-258 10 0.949337491812

10 correlated vars at ~95% / dataset: 10/40 relevant variables, line
ar ground truth, some noise
False 2.17698845853e-61 12 0.901747976192

4 groups of 10 correlated vars at ~95% / dataset: 40/40 relevant var
iables, linear ground truth, some noise
False 2.7000044484e-82 37 0.974570549784

4 groups of 10 correlated vars at ~95% / dataset: 40/40 relevant var
iables, linear ground truth, NO noise
False 2.22044604925e-16 40 0.991527659962