DSO530 Statistical Learning Methods

Lecture 2b: Multiple Linear Regression

Dr. Xin Tong
Department of Data Sciences and Operations
Marshall School of Business
University of Southern California

Multiple linear regression

In addition to scalar input $x \in \mathbb{R}$, we can consider vector input $x \in \mathbb{R}^p$

- p is feature dimension of input (sometimes use d for *dimension*)
- Inputs of form

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$$

- Call x the *covariates*, *independent variables*, *explanatory variables*, *features*, *attributes* or *predictor variables*
  - Note that variables are usually NOT independent of one another

Example: Boston housing data

- Output (response, target, dependent) variable is medv, the median home price in neighborhoods of Suburbs of Boston
- Input variables include (but not limited to)
  - crim: per capita crime rate by town
  - rm: average number of rooms per dwelling
  - zn: proportion of large lots (zoned for $> 25,000$ feet)
  - chas: whether a home is near the Charles river ($x \in \{0, 1\}$)
  - ptratio: pupil-teacher ratio by town
- Let's do some data exploration

```
from sklearn.datasets import load_boston
boston_dataset = load_boston(); print(boston_dataset.keys())
```

```
dict_keys(['data', 'target', 'feature_names', 'DESCR', 'filename'])
```

```
print(boston_dataset.DESCR)
```

```
.. _boston_dataset:

Boston house prices dataset
---------------------------

**Data Set Characteristics:**

    :Number of Instances: 506

    :Number of Attributes: 13 numeric/categorical predictive. Median Value (attribute 14) is usually the target.

    :Attribute Information (in order):
        - CRIM     per capita crime rate by town
        - ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
        - INDUS    proportion of non-retail business acres per town
        - CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
        - NOX      nitric oxides concentration (parts per 10 million)
        - RM       average number of rooms per dwelling
        - AGE      proportion of owner-occupied units built prior to 1940
        - DIS      weighted distances to five Boston employment centres
        - RAD      index of accessibility to radial highways
        - TAX      full-value property-tax rate per $10,000
        - PTRATIO  pupil-teacher ratio by town
        - B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
        - LSTAT    % lower status of the population
        - MEDV     Median value of owner-occupied homes in $1000's

    :Missing Attribute Values: None

    :Creator: Harrison, D. and Rubinfeld, D.L.

This is a copy of UCI ML housing dataset.
https://archive.ics.uci.edu/ml/machine-learning-databases/housing/


This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University.
```

```
boston = pd.DataFrame(boston_dataset.data, columns=boston_dataset.feature_names)
boston.head()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33

```
boston['MEDV'] = boston_dataset.target
```

Figure 1: Prepare data

```
boston.isnull().sum() # checking whether there is any missing values
```

```
CRIM        0
ZN          0
INDUS       0
CHAS        0
NOX         0
RM          0
AGE         0
DIS         0
RAD         0
TAX         0
PTRATIO     0
B           0
LSTAT       0
MEDV        0
dtype: int64
```

Figure 2: Checck missing data

```
import seaborn as sns
sns.distplot(boston['MEDV']) #This function combines the matplotlib hist function
#(with automatic calculation of a good default bin size) with the seaborn kdeplot() and other functions.
# Here `kde` stands for kernel density estimate can be understood as a smooth version of hist.
plt.show()
```

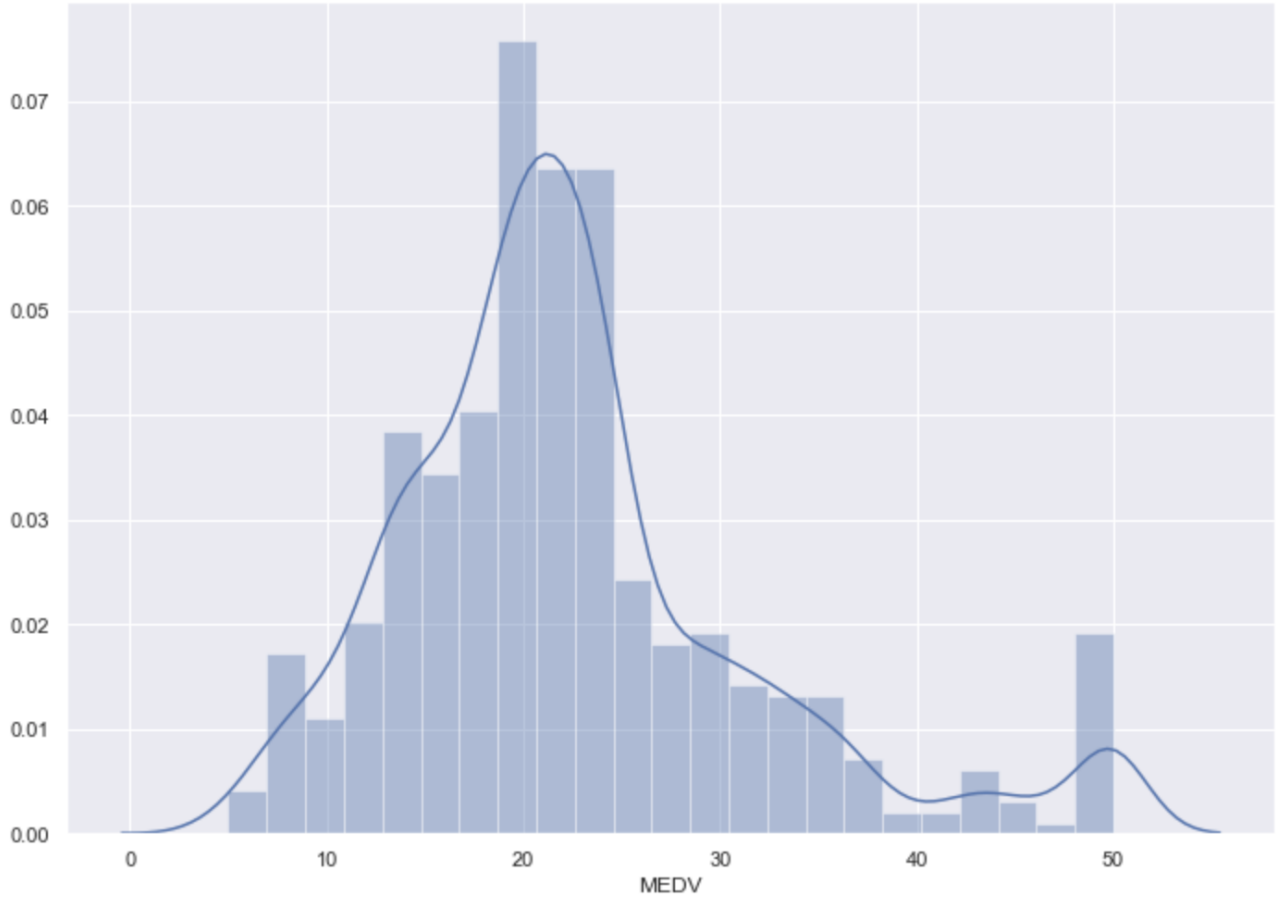

Figure 3: Plot response variable

```
correlation_matrix = boston.corr().round(2)
## can try print(correlation_matrix), or alternatively do the following
sns.heatmap(data=correlation_matrix, annot=False)
# annot = True to print the values inside the square
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a238eec50>
```

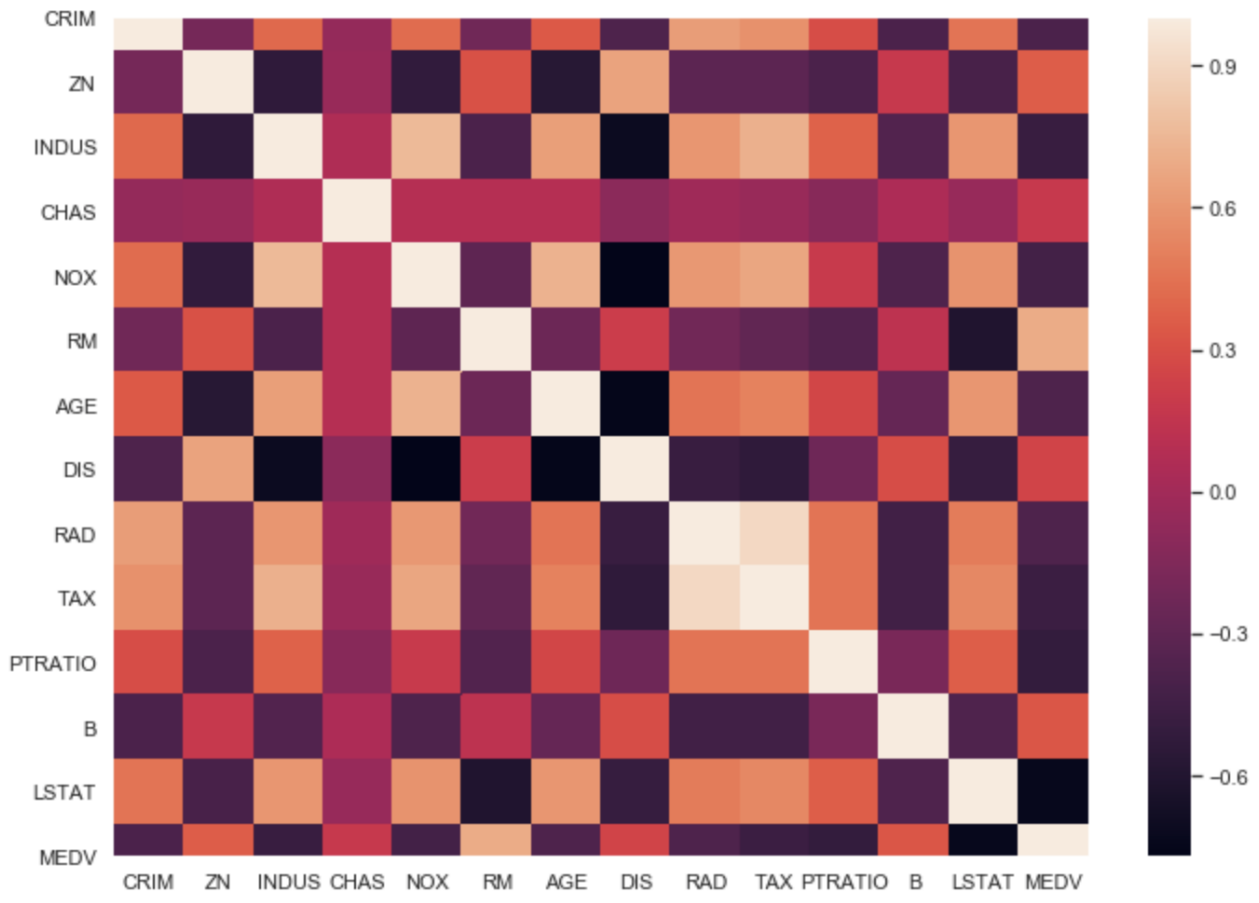

Figure 4: Correlation matrix

- For illustration purpose we only use `PTRATIO` and `RM` as covariates
- It looks like one has negative correlation with `MEDV` and the other has positive correlation with `MEDV`
- If for some reason, we have to choose only one of these two predictors, which one would you like to choose?

```
plt.figure(figsize=(20, 5))
features = ['PTRATIO', 'RM']; target = boston['MEDV']
for i, col in enumerate(features):
    plt.subplot(1, len(features) , i+1) # subplot(nrows, ncols, index). Three separate integers describing
    #the position of the subplot. If the three integers are nrows, ncols, and index in order,
    #the subplot will take the index position on a grid with nrows rows and ncols columns.
    #index starts at 1 in the upper left corner and increases to the right.
    x = boston[col]; y = target
    plt.scatter(x, y)
    plt.title('MEDV' + " vs. " + col)
    plt.xlabel(col); plt.ylabel('MEDV')
```

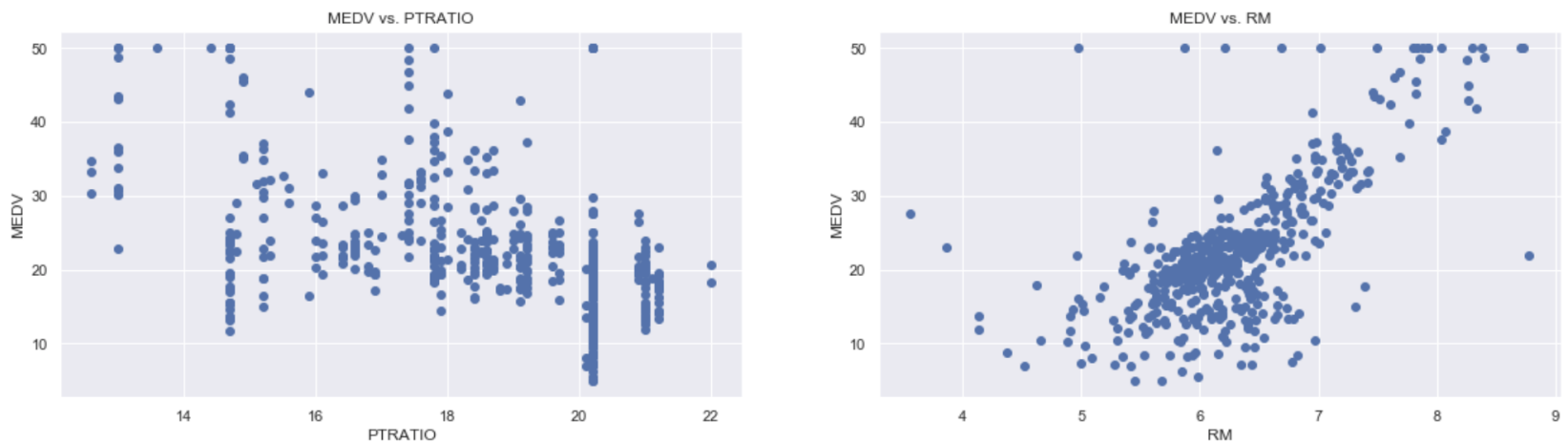

Figure 5

```
X2 = boston[['PTRATIO','RM']].values
y2 = boston['MEDV'].values
linear_model_2 = LinearRegression(); linear_model_2.fit(X2, y2)
r_sq_house = linear_model_2.score(X2, y2); print('coefficient of determination:', r_sq_house)
```

```
coefficient of determination: 0.5612534621272917
```

```
print(linear_model_2.coef_); print(linear_model_2.intercept_)
```

```
[-1.2671614   7.71407021]
-2.5611648168336067
```

Figure 6: Fitting a multiple linear regression model

- The model fitting is not much different from fitting a simple linear regression model
- Except that one does not need to worry about the covariates being a 1D object
- Try regress `MEDV` on `RM` only, do you see a larger or smaller R square?
- The Python `sklearn` library focuses on prediction, is not yet fully developed for some typical statistical inferential tasks.

```
import statsmodels.api as sm
X3 = sm.add_constant(X2)4
ols = sm.OLS(y2, X3)
ols_result = ols.fit()
ols_result.summary()
```

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.561
Model:	OLS	**Adj. R-squared:**	0.560
Method:	Least Squares	**F-statistic:**	321.7
Date:	Sun, 26 Jan 2020	**Prob (F-statistic):**	1.04e-90
Time:	17:22:15	**Log-Likelihood:**	-1631.8
No. Observations:	506	**AIC:**	3270.
Df Residuals:	503	**BIC:**	3282.
Df Model:	2		
Covariance Type:	nonrobust		

Figure 7

- The df for residuals is $n - p - 1$; here the total df $= 1 + 1 + 503 = 505$

- continue from the previous page. The summary also contains:

	coef	std err	t	P>|t|	[0.025	0.975]
const	-2.5612	4.189	-0.611	0.541	-10.791	5.669
x1	-1.2672	0.134	-9.440	0.000	-1.531	-1.003
x2	7.7141	0.414	18.650	0.000	6.901	8.527

Figure 8

- Compare the coefficients fitted by `statmodels` and `sklearn`
- What are the second to the last columns about?
- We see some t-statistics and their p-values here. What are the null hypotheses?

- continue from the previous page. The summary also contains:

	coef	std err	t	P>|t|	[0.025	0.975]
const	-2.5612	4.189	-0.611	0.541	-10.791	5.669
x1	-1.2672	0.134	-9.440	0.000	-1.531	-1.003
x2	7.7141	0.414	18.650	0.000	6.901	8.527

Figure 8

- Compare the coefficients fitted by `statmodels` and `sklearn`
- What are the second to the last columns about?
- We see some t-statistics and their p-values here. What are the null hypotheses?

When we use more than one input variables

- Make predictions

$$\hat{y} = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_j$$

- Model the data as

$$y = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \varepsilon$$

  where ε is noise

- As in simple linear regression, choose noise distribution based on
  - Convenience (assume it is normally distributed)
  - Prior knowledge (rarely)

Fitting a multiple regression

- Making predictions using

$$\hat{y} = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_j$$

- Fit as in single variable case. Solve (least squares criterion or OLS)

$$\underset{\beta_0 \in \mathbb{R}, \beta \in \mathbb{R}^p}{\text{minimize}} \sum_{i=1}^{n} (y_i - \beta_0 - \beta^T x_i)^2$$

- Since each observation has p values, $x_i = (x_{i1}, \cdots, x_{ip})^T$, where $i = 1, \cdots, n$. In matrix notation, let $y = (y_1 \cdots, y_n)^T$ and

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & x_{23} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & x_{n3} & \cdots & x_{np} \end{bmatrix}$$

where x_{ij} is the ith observation of the jth variable

Is there a relationship between the reponse and predictors?

- $H_0 : \beta_1 = \cdots = \beta_p = 0$
- H_a: at least one of β_j is non-zero
- Did we miss β_0?
- This hypothesis test is performed by computing the F-statistic,

$$F = \frac{(TSS - RSS)/p}{RSS/(n - p - 1)}$$

- The larger the F-statistic, the strong evidence against null hypothesis
- Why do we need F-statistic and its p-values? Because inspecting individual t-statistic and p-value is NOT enough to answer the question posted on the top of the slide
- F-statistic is used only when p is small
- See Equation (3.24) in `ISLR` for more general definition of the F-statistic (Optional)

RSE and R^2 for multiple linear regression

- RSE and R^2 are common measures of model fit.
- R^2 is used most often.
- R^2 will always increase when more variables are added to the model
- Reason: adding another variable to the least squares equations allows us to fit the training data more accurately
- Residual standard error (RSE)

$$RSE = \sqrt{\frac{1}{n-p-1}RSS}$$

- Does RSE always increase/decrease when more variables are added to the model?
- RSE has the same unit as the response. So we cannot compare RSEs across different responses.
- $RSE/\bar{y}$ could be more interpretable than just RSE.
- In this course, we focus on R^2

Making prediction

- There are three sorts of uncertainty associated with making prediction using linear model
- the least squares plane

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1, + \cdots + \hat{\beta}_p x_p$$

  is an estimate for the true population regression plane

$$f(x) = \beta_0 + \beta_1 x_1, + \cdots + \beta_p x_p$$

- assuming a linear model for $f(X)$ is almost always an *approximation* of reality
- Even if you know $f(x)$, the response value cannot be determined perfectly because of the random error ε in the model ($y = f(x) + \varepsilon$)
- The first two kind of errors are *reducible errors* and the last is *irreducible error*
- Why is (the same level) prediction interval for y wider than confidence interval for $f(x)$? p.82 of `ISLR`

Qualitative predictors and interaction terms

Qualitative predictors:

- Sometimes, categorical variables can be useful in making predictions.
- We usually code qualitative variables by *dummy variables* (variables taking values 0 and 1). Why are they called dummy variables?
- Suppose you a categorical variable with three *levels*. We need to code it with two dummy variables. See Python Tutorial 2 for more details

Interaction terms:

- Sometimes, influence of the predictors can not be decoupled into additive terms. In addition to *main effects*, we need *interaction effect*:

$$sales = \beta_0 + \beta_1 \cdot TV + \beta_2 \cdot radio + \beta_3 \cdot (radio \cdot TV) + \varepsilon$$

-Rewrite the above equation as:

$$sales = \beta_0 + (\beta_1 + \beta_3 \cdot radio) \cdot TV + \beta_2 \cdot radio + \varepsilon$$

- How do we interpret β_3? the increase in the effectiveness of `TV` advertising for a one unit increase in `radio` advertising (or vice-versa)

Nonlinear relationship

- True relationship between response and predictors might be nonlinear
- A first extension to linear regression to handle non-linear relationship is to use *polynomial regression*
- When we use higher order polynomial, **overfitting** is a concern.
- Suppose we use a polynimal of degree 5 to fit a data set with 6 points.
- All 6 residuals are zero. R^2 is 1. But this is clearly not the model you want.
- Q: what is the value of *simulation* in machine learning, pilot training, construction. . . ..? For example, if you don't believe the above statement. . . .

Collinearity

- Collinearity: two or more predictor variables are closely related to one another
- Collinearity causes problems to separate out the individual effects of collinear variables on the response
- multicollinearity: collinearity that exists between three or more variables; hard to detect by looking at the correlation matrix
- variance inflation factor(VIF):

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}}$$

  where $R^2_{X_j|X_{-j}}$ is the R^2 from a regression of X_j onto all other predictors
- A VIF value bigger than 5 or 10 indicates problematic amount of collinearity. Solution 1: drop variable with high VIF; solution 2: eliminate predictor with the highest p-value in its t-test; solution
- Does collinearity matter for prediction? Not as much as for inference

Performance measure on the (training) dataset does not tell the whole story

- So far, all the performance evaluations are on training data
- They tell us how well the model performs on the SAME data that was used to fit it
- By overfitting the model (e.g., using lots of predictors), we can make the RSS arbitrarily low.
- But this model will not accurately predict future datasets. Prediction of future datasets is called "generalization".
- For fixed amount of data, the larger the model, the larger the R^2. Therefore, R^2 cannot be used to compare models with different sizes.
- There are two solutions
  - create model-size adjusted performance measures, such as adj-R^2.
  - evaluate the performance on data NOT used for fitting (training).

Some quotes from a practitioner regarding linear regression

- Must clean data beforehand. Raw data often contains missing values, zeros values and erroneous extreme values. Don't let these values distort regression results, make sure no "garbage-in-garbage-out" scenario.
- Linear regression, like all other methodologies, is just a tool. What matters more is how to find more signals (x) that makes a stronger relationship. Nowadays there are many, many advanced modeling techniques, which can potentially improve the rsq over linear regression. But my advice is to first make enough effort in looking for new significant signals, and then explore newer methodologies.
- Ratio of number of samples to number of parameters: as a rule of thumb it's 30 to 1 but in practice the comfort level does depend on whether the signals are strong enough. For significant relationships 30 to 1 should be enough, for noisy data like financial data, the rsq is usually smaller than 10% or sometimes even smaller than 3%. In this case one really need more samples, like 100 to 1, to be comfortable not to get into spurious relationships.

Some quotes from a practitioner regarding linear regression

- Must clean data beforehand. Raw data often contains missing values, zeros values and erroneous extreme values. Don't let these values distort regression results, make sure no "garbage-in-garbage-out" scenario.
- Linear regression, like all other methodologies, is just a tool. What matters more is how to find more signals (x) that makes a stronger relationship. Nowadays there are many, many advanced modeling techniques, which can potentially improve the rsq over linear regression. But my advice is to first make enough effort in looking for new significant signals, and then explore newer methodologies.
- Ratio of number of samples to number of parameters: as a rule of thumb it's 30 to 1 but in practice the comfort level does depend on whether the signals are strong enough. For significant relationships 30 to 1 should be enough, for noisy data like financial data, the rsq is usually smaller than 10% or sometimes even smaller than 3%. In this case one really need more samples, like 100 to 1, to be comfortable not to get into spurious relationships.

Some quotes from a practitioner regarding linear regression

- Must clean data beforehand. Raw data often contains missing values, zeros values and erroneous extreme values. Don't let these values distort regression results, make sure no "garbage-in-garbage-out" scenario.
- Linear regression, like all other methodologies, is just a tool. What matters more is how to find more signals (x) that makes a stronger relationship. Nowadays there are many, many advanced modeling techniques, which can potentially improve the rsq over linear regression. But my advice is to first make enough effort in looking for new significant signals, and then explore newer methodologies.
- Ratio of number of samples to number of parameters: as a rule of thumb it's 30 to 1 but in practice the comfort level does depend on whether the signals are strong enough. For significant relationships 30 to 1 should be enough, for noisy data like financial data, the rsq is usually smaller than 10% or sometimes even smaller than 3%. In this case one really need more samples, like 100 to 1, to be comfortable not to get into spurious relationships.

Textbook reading

- Note that the Lectures 2a and 2b do not cover the entirety of Chapter 3 of ISLR
- You are not responsible for contents in the chapter beyond what is covered in the lectures and tutorials
- It is recommended that you read the sections 3.1 and 3.2 (excluding the general definition of F statistics) in ISLR