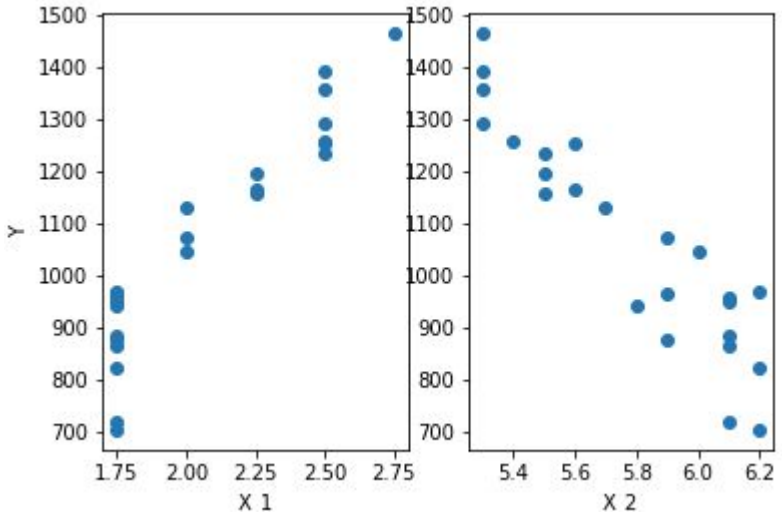
Multivariate Regression

GR 5205 / GU 4205
Section 3

Columbia University
Xiaofei Shi

With more predictors



2.75	5.3	1464
2.50	5.3	1394
2.50	5.3	1357
2.50	5.3	1293
2.50	5.4	1256
2.50	5.6	1254
2.50	5.5	1234
2.25	5.5	1195
2.25	5.5	1159
2.25	5.6	1167
2.00	5.7	1130
2.00	5.9	1075
2.00	6.0	1047
1.75	5.9	965
1.75	5.8	943
1.75	6.1	958
1.75	6.2	971
1.75	6.1	949
1.75	6.1	884
1.75	6.1	866
1.75	5.9	876
1.75	6.2	822
1.75	6.2	704
1.75	6.1	719

Generally speaking...

$$Y = \beta_0 + x^{(1)}\beta_1 + x^{(2)}\beta_2 + \cdots + x^{(p-1)}\beta_{p-1} + \epsilon$$

$$x = \begin{bmatrix} 1 & x_1^{(1)} & \cdots & x_1^{(p-1)} \\ 1 & x_2^{(1)} & \cdots & x_2^{(p-1)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n^{(1)} & \cdots & x_n^{(p-1)} \end{bmatrix} = \begin{bmatrix} 1 & x_1^\top \\ 1 & x_2^\top \\ \vdots & \vdots \\ 1 & x_n^\top \end{bmatrix} \in \mathbb{R}^{n \times p}$$

Recall our road map...

x - predictor (random) variables Y - response random variable

- Build your model:

  1) relationship: $Y = x\beta + \epsilon, \qquad \mathbb{E}[\epsilon] = 0,\ \mathrm{Var}[\epsilon] = \sigma^2 I_n.$

  2) preference: choose $\widehat{\beta}$ to minimize mean squared error

- Estimate your model parameters: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

  1) using observed data to express your preference: $Q = \sum_{i=1}^{n} (y_i - x_i^\top \beta)^2$

  2) get parameters estimation for your model:

- Understand your model:

  1) properties of estimations:

  2) predictions: $\widehat{Y}_0 = x_0^\top \widehat{\beta}, \qquad \widehat{y}_0 = x_0^\top b.$

Taking partial derivatives wrt a vector

Linear forms. If $f(\mathbf{X}) = \mathbf{X}^T\mathbf{a}$, with $\mathbf{a}$ not a function of $\mathbf{X}$, then

$$\nabla(\mathbf{X}^T\mathbf{a}) = \mathbf{a}$$

Quadratic forms. Let $\mathbf{C}$ be a $p \times p$ matrix which is not a function of $\mathbf{X}$, and consider the **quadratic form** $\mathbf{X}^T\mathbf{C}\mathbf{X}$. (You can check that this is scalar.) The gradient is

$$\nabla(\mathbf{X}^T\mathbf{C}\mathbf{X}) = (\mathbf{C} + \mathbf{C}^T)\mathbf{X}.$$

Back to optimize $Q(\beta) = \|y - x\beta\|^2$

$$\widehat{\beta} = (x^\top x)^{-1} x^\top Y$$

$$\widehat{\sigma}^2_{LS} = \frac{1}{n-p}\|Y - \widehat{Y}\|^2 = \frac{1}{n-p}\|Y - x\widehat{\beta}\|^2$$

Collinear

In order to invert $x^\top x$ we need on of the following:

- Its determinant is non-zero.
- It is of "full column rank", meaning all of its columns are linearly independent.
- It is of "full row rank", meaning all of its rows are linearly independent.

Collinear in terms of data if:

- If n < p.
- If one of the predictor variables is constant.
- If two of the predictor variables are proportional to each other.
- If two of the predictor variables are otherwise linearly related.

Multiple Regression Isn't Just a Bunch of Simple Regressions

Suppose the real model is $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$. (Nothing turns on $p = 2$, it just keeps things short.) What would happen if we did a simple regression of Y on just X_1? We know that the optimal (population) slope on X_1 is

$$\frac{\text{Cov}\left[X_1, Y\right]}{\text{Var}\left[X_1\right]}$$

Let's substitute in the model equation for Y:

$$\begin{aligned}
\frac{\text{Cov}\left[X_1, Y\right]}{\text{Var}\left[X_1\right]} &= \frac{\text{Cov}\left[X_1, \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon\right]}{\text{Var}\left[X_1\right]} \\
&= \frac{\beta_1 \text{Var}\left[X_1\right] + \beta_2 \text{Cov}\left[X_1, X_2\right] + \text{Cov}\left[X_1, \epsilon\right]}{\text{Var}\left[X_1\right]} \\
&= \beta_1 + \frac{\beta_2 \text{Cov}\left[X_1, X_2\right] + 0}{\text{Var}\left[X_1\right]} \\
&= \beta_1 + \beta_2 \frac{\text{Cov}\left[X_1, X_2\right]}{\text{Var}\left[X_1\right]}
\end{aligned}$$

Multiple Regression
Isn't Just a Bunch of Simple Regressions

Dep. Variable:	y	R-squared:	0.851
Model:	OLS	Adj. R-squared:	0.844
Method:	Least Squares	F-statistic:	125.4
Date:	Fri, 02 Oct 2020	Prob (F-statistic):	1.49e-10
Time:	15:43:15	Log-Likelihood:	-139.14
No. Observations:	24	AIC:	282.3
Df Residuals:	22	BIC:	284.6
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	4471.3393	304.254	14.696	0.000	3840.354	5102.324
x1	-588.9621	52.602	-11.196	0.000	-698.053	-479.871

Dep. Variable:	y	R-squared:	0.876
Model:	OLS	Adj. R-squared:	0.870
Method:	Least Squares	F-statistic:	155.0
Date:	Fri, 02 Oct 2020	Prob (F-statistic):	1.95e-11
Time:	20:28:29	Log-Likelihood:	-136.94
No. Observations:	24	AIC:	277.9
Df Residuals:	22	BIC:	280.2
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-99.4643	95.210	-1.045	0.308	-296.918	97.990
x_2	564.2039	45.317	12.450	0.000	470.221	658.186

Dep. Variable:	y	R-squared:	0.898
Model:	OLS	Adj. R-squared:	0.888
Method:	Least Squares	F-statistic:	92.07
Date:	Fri, 02 Oct 2020	Prob (F-statistic):	4.04e-11
Time:	20:31:04	Log-Likelihood:	-134.61
No. Observations:	24	AIC:	275.2
Df Residuals:	21	BIC:	278.8
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1798.4040	899.248	2.000	0.059	-71.685	3668.493
x_1	-250.1466	117.950	-2.121	0.046	-495.437	-4.856
x_2	345.5401	111.367	3.103	0.005	113.940	577.140

Something more about the coding

- Multivariate linear regression in R:

```
lm(y ~ x_1 + x_2 + x_3, data=dataset)
```

- Multivariate linear regression in Python:
  - ```
    import statsmodels.api as sm
    import statsmodels
    ```
  - Intercept is not set as default!

    ```
    X = sm.add_constant(X)
    ```
  - ```
    model = sm.OLS(Y, X).fit()
    ```