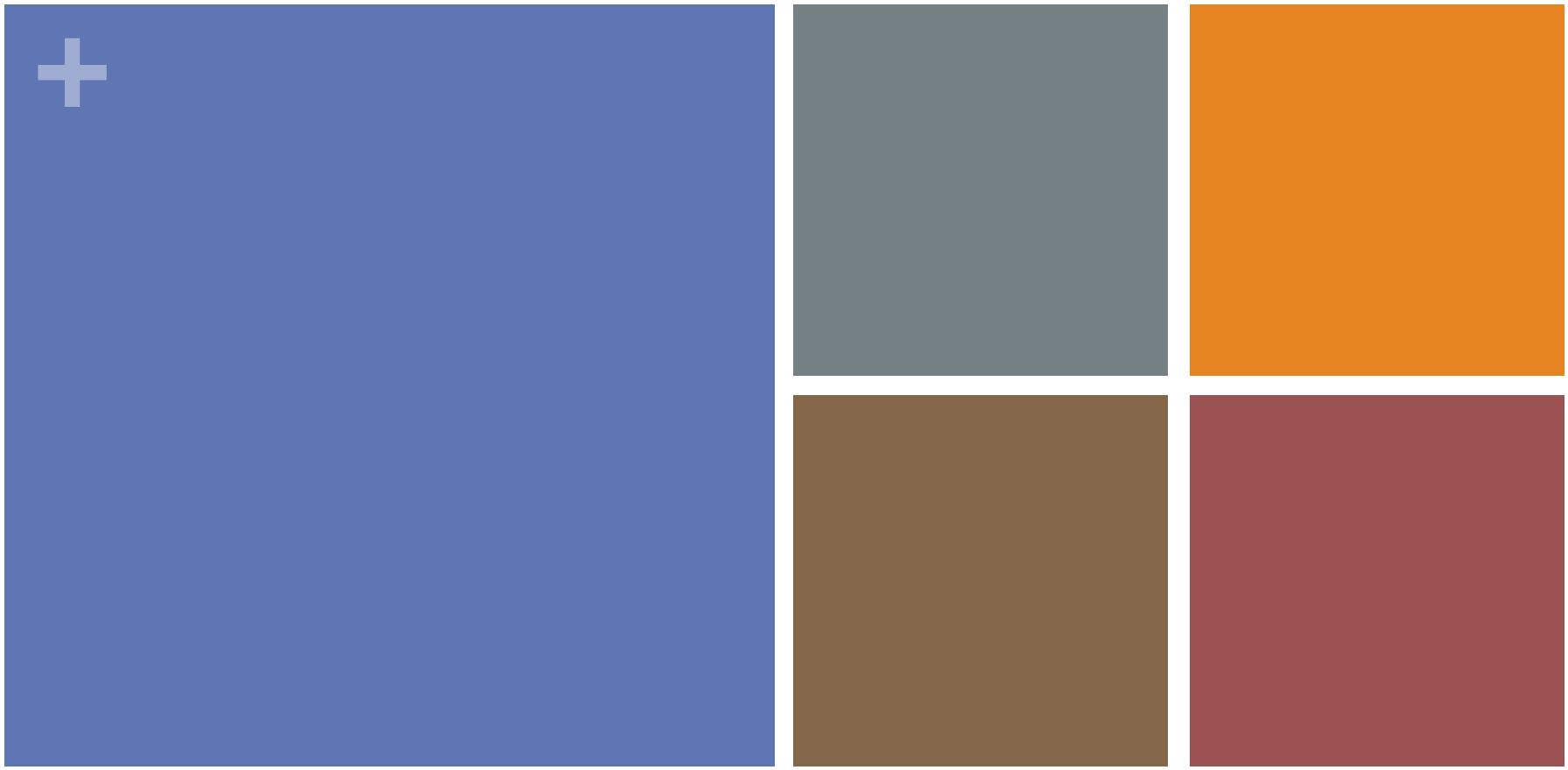


Lecture 2 – Predicting Wine Quality with Linear Regression I

IEOR 142 – Introduction to Machine Learning and Data Analytics

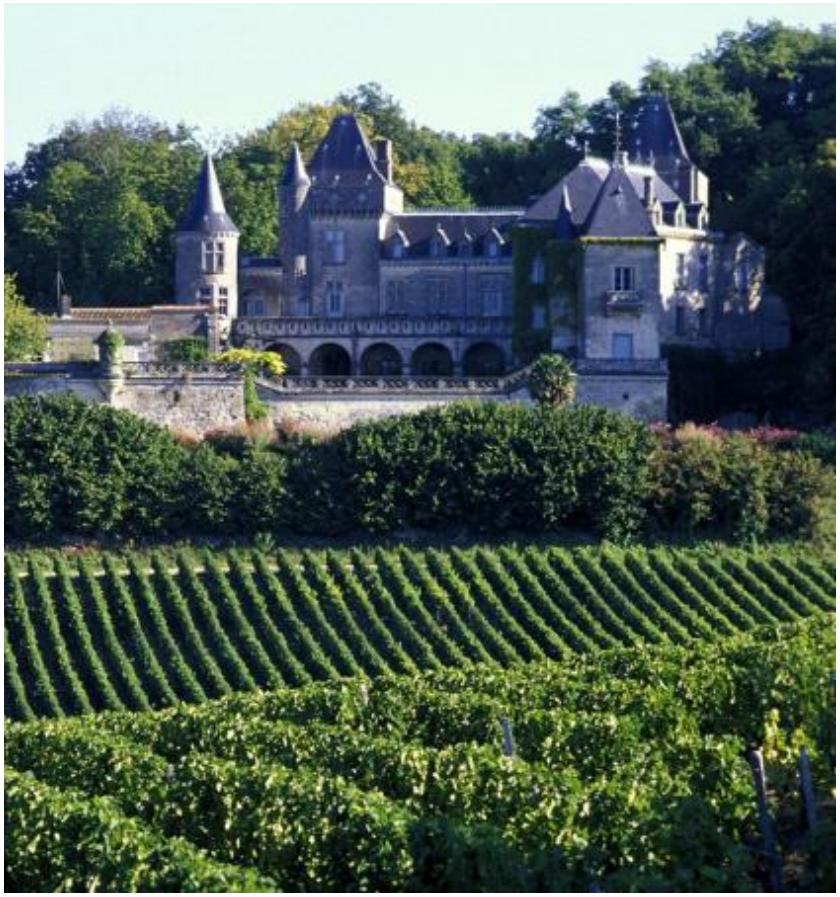
Fall 2018 – Paul Grigas

Today's Agenda

- Predicting wine quality with linear regression
- Introduction to Statistical Learning
- Model validation, overfitting, and other issues

Predicting Wine Quality

Vintage Bordeaux Wine

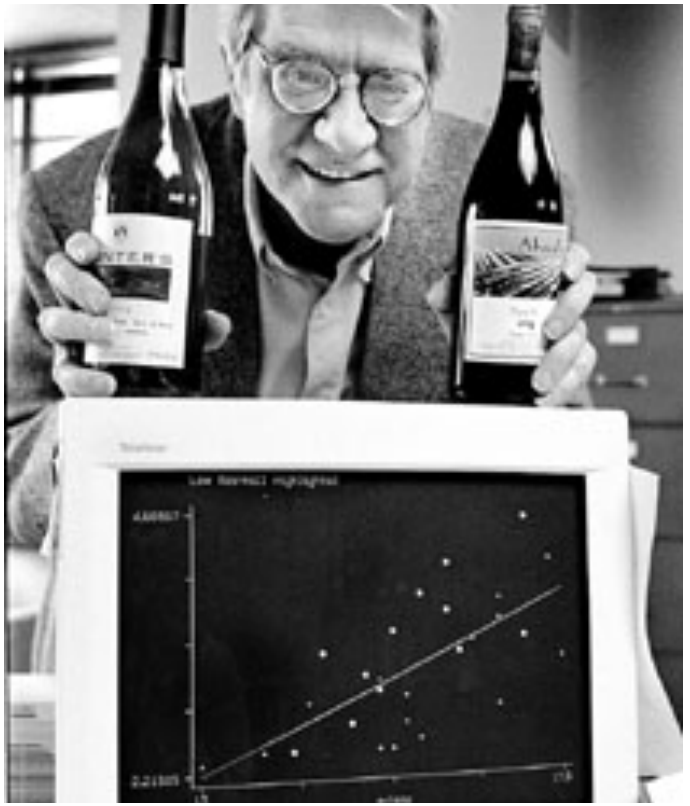


- Vintage wine vs. non-vintage wine?
- Large differences in price and quality in different years, even though wine is produced in a similar way
- Meant to be aged, so it is hard to know the quality of the wine when it initially goes on the market
- Expert tasters predict which wines will be good
- Can analytics be used to develop a different system for assessing the quality of wine?

Wine Quality – Ask the Expert

Predicting the Quality of Wine

- March 1990: Orley Ashenfelter, a Princeton economics professor, claims he can predict wine quality without tasting the wine



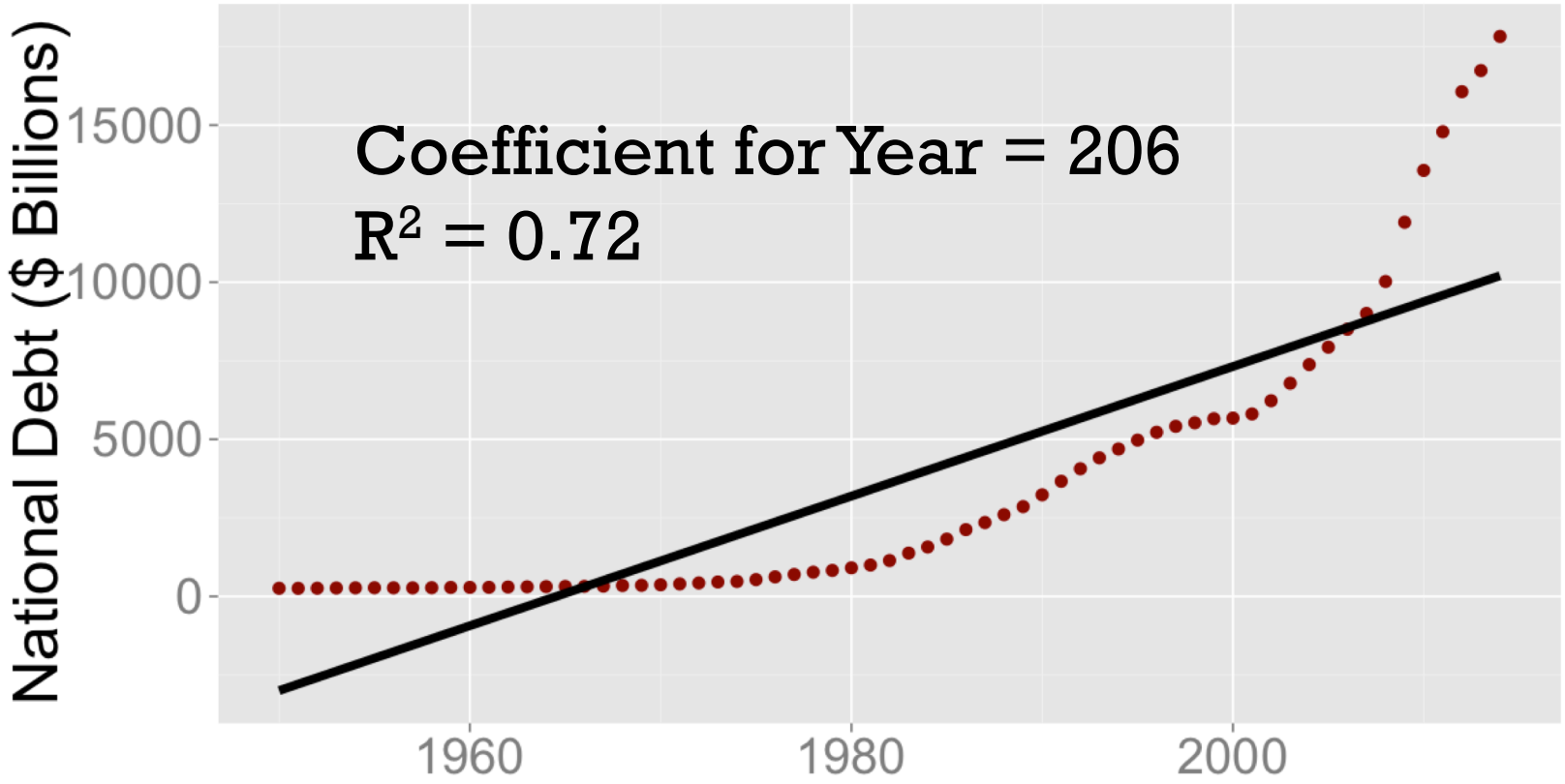
Using Linear Regression

- Ashenfelter used **(multiple) linear regression**
  - Predicts a continuous response variable – the *dependent variable*
  - Prediction is based on a set of *independent variables*
- Independent variables (features):
  - Age – older wines are more expensive
  - *Weather*
    - Average Growing Season Temperature
    - Harvest Rain
    - Winter Rain
- Dependent variable:
  - Price Index – composite metric of many different wineries in thousands of wine auctions in the years 1990-1991
  - His model used Log(Price Index)

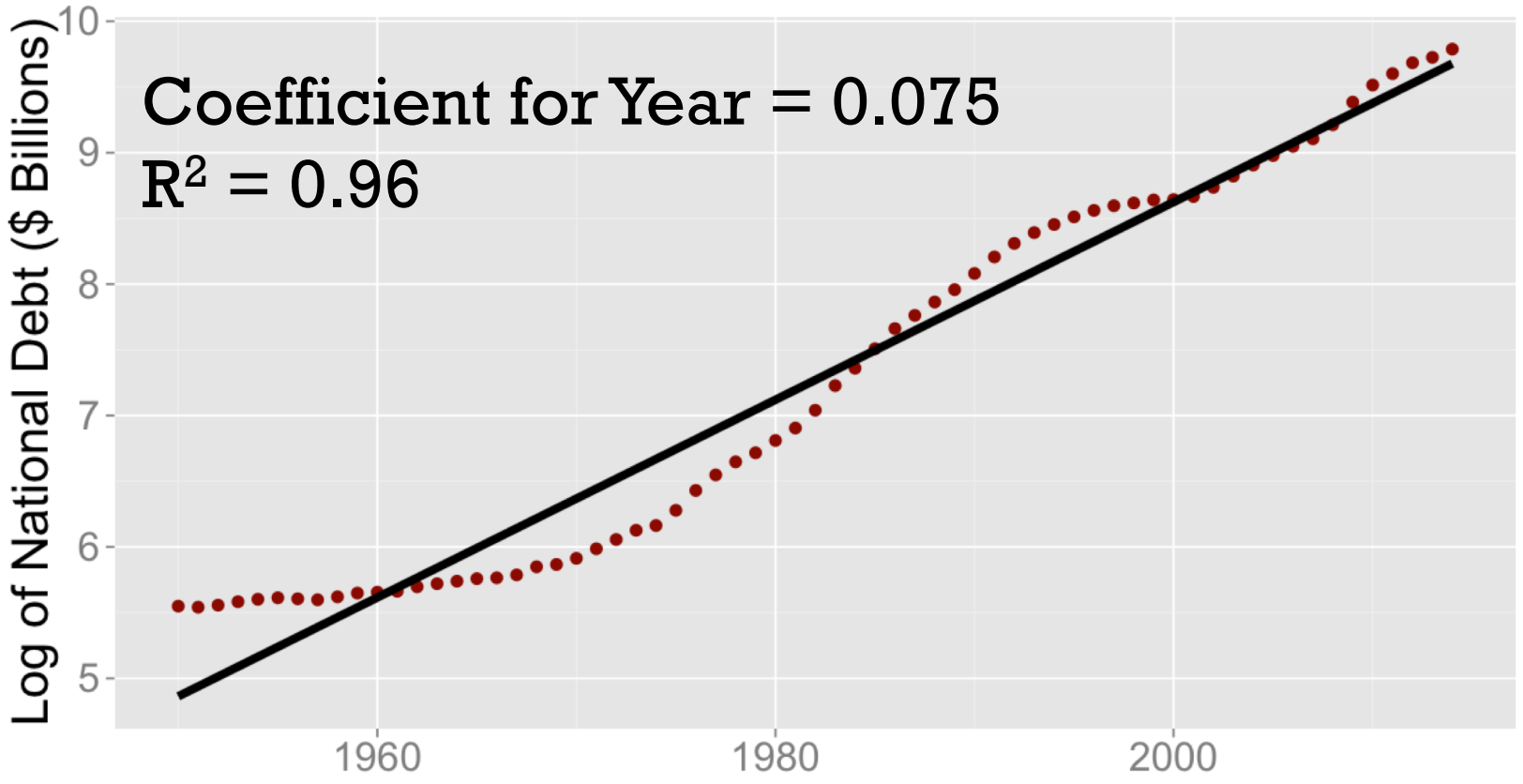
Why Log(Price Index)?

- Produces a better linear fit
  - Better fit revealed through plotting
  - The log( ) transformation also arises intrinsically, especially in settings where "growth" or "proportion" are natural phenomena

US National Debt (1950 – 2014)



US National Debt (1950 – 2014)



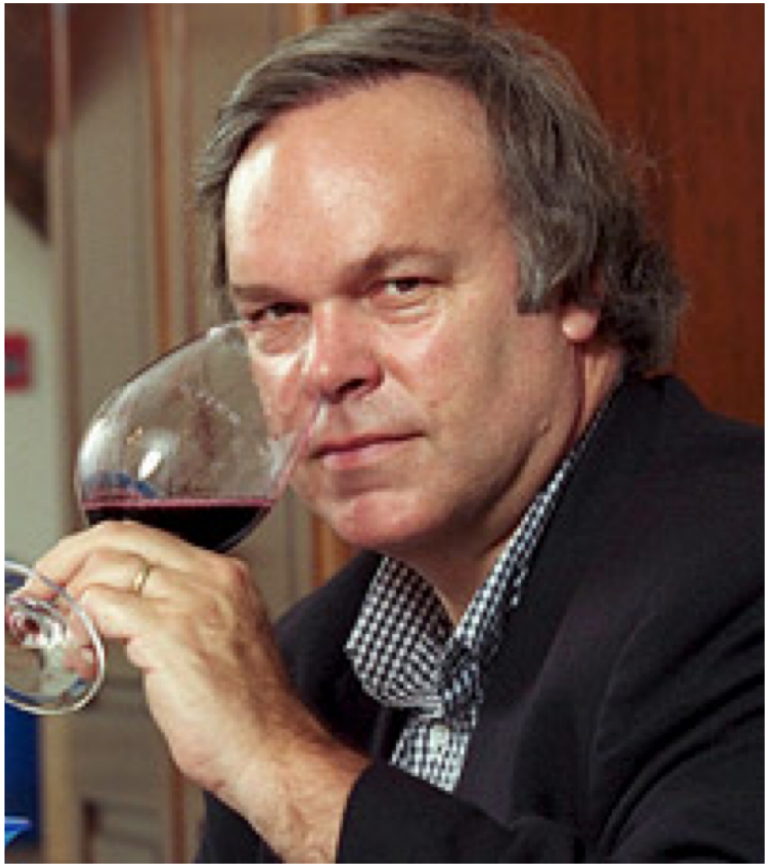
The Expert's Reaction

Robert Parker, the world's most influential wine expert at the time:

"Ashenfelter is an absolute total sham"

"Really a Neanderthal way of looking at wine"

"Rather like a movie critic who never goes to see the movie but tells you how good it is based on the actors and the director"



Vintage Wine Data

- Log(price index) based on 2015 auction prices
- Winter rain (mm)
- Harvest rain (mm)
- Average Temperature in growing season (Celsius)
- Average Temperature in harvest season (Celsius)
- Age of wine (years since vintage)
- Population of France
- US Alcohol Consumption (per capita, in liters of 100% alcohol)

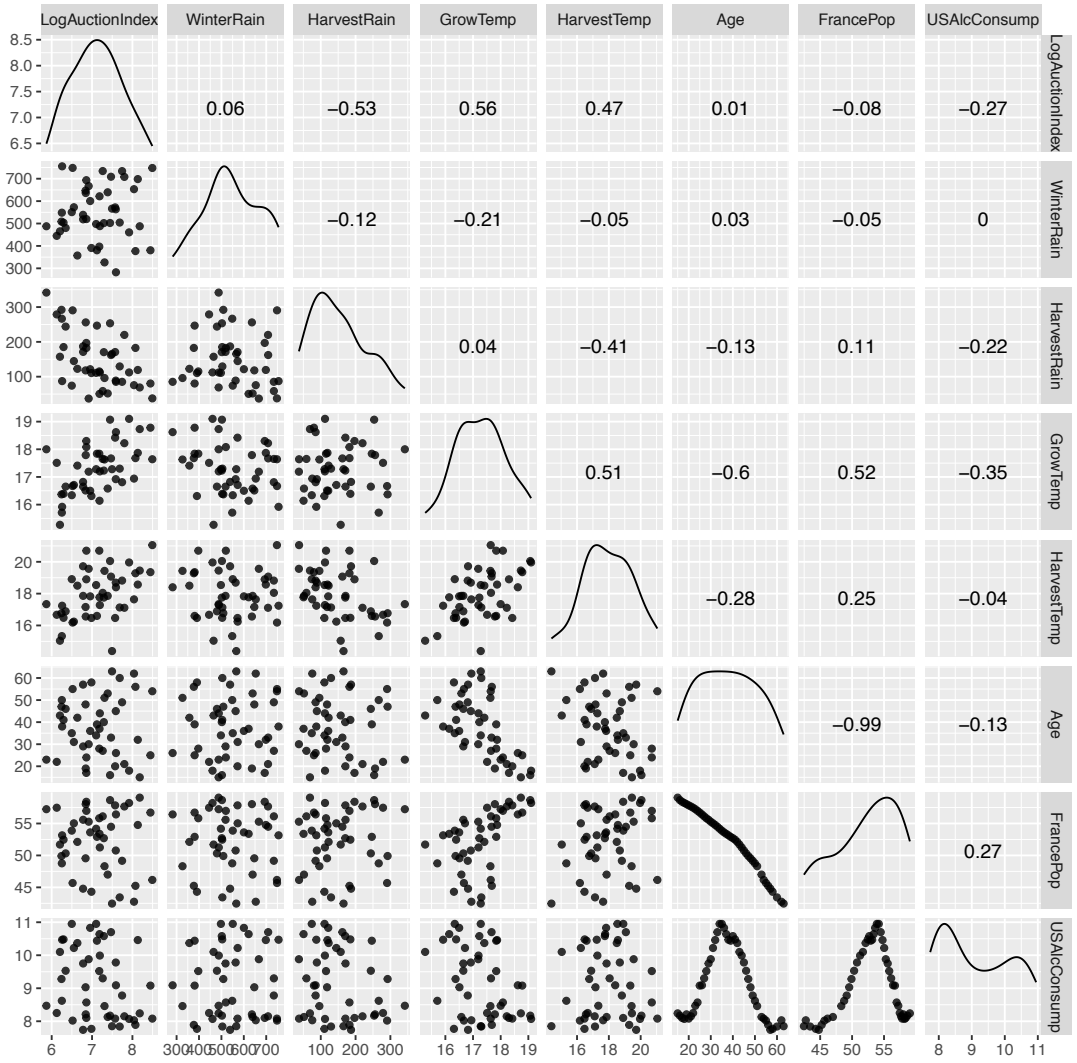
Vintage Wine Data, cont.

p = # of independent variables (p=7); n = # of observations (n=46)

		y	x_1	x_2	x_3	x_4	x_5	x_6	x_7
		LogAuctionIndex	WinterRain	HarvestRain	GrowTemp	HarvestTemp	Age	FrancePop	USAlcConsump
	1 1952	y_1 = 7.4950	x_{11} = 566.4	x_{21} = 165.5	x_{31} = 17.28	x_{41} = 14.39	x_{51} = 63	x_{61} = 42.46	x_{71} = 7.85
	2 1953	y_2 = 8.0393	x_{12} = 653.3	x_{22} = 75.6	x_{32} = 16.94	x_{42} = 17.64	x_{52} = 62	x_{62} = 42.75	x_{72} = 8.03
	3 1955	7.6858	504.3	129.5	17.30	17.13	60	43.43	7.84
	4 1957	6.9845	390.8	110.4	16.31	16.47	58	44.31	7.77
	5 1958	6.7772	538.8	187.0	16.82	19.72	57	44.79	7.74
	6 1959	8.0757	377.0	182.6	17.68	19.28	56	45.24	7.89
	7 1960	6.5188	748.2	290.6	16.67	16.18	55	45.68	8.02
	8 1961	8.4937	747.8	37.7	17.64	21.05	54	46.16	8.08
	9 1962	7.3880	639.4	51.8	16.58	17.86	53	47.00	8.13
	10 1964	7.3094	326.5	96.1	17.63	19.43	51	48.31	8.46
n=46	11 1965	6.2518	548.4	266.6	15.71	15.33	50	48.76	8.62
	12 1966	7.7443	734.0	85.2	16.81	18.82	49	49.16	8.78
	13 1967	6.8398	646.9	118.1	16.51	17.16	48	49.55	9.03
	14 1968	6.2435	508.6	292.1	16.37	16.77	47	49.91	9.28
	15 1969	6.3459	480.1	243.9	16.65	16.89	46	50.32	9.53
	16 1970	7.5883	563.5	88.8	16.92	18.69	45	50.77	9.78
	17 1971	7.1934	488.4	111.9	17.20	17.28	44	51.25	9.99
	18 1972	6.2049	465.1	157.3	15.27	15.04	43	51.70	10.10
	19 1973	6.6367	357.2	122.6	17.41	18.50	42	52.12	10.37
	20 1974	6.2941	503.6	185.1	16.39	16.48	41	52.46	10.48
	...	...	...	...	...	...	...	...	...
	46 2000	y_n = 8.1817	x_{1n} = 487.8	x_{2n} = 69.0	x_{3n} = 18.73	x_{4n} = 19.45	x_{5n} = 15	x_{6n} = 59.05	x_{7n} = 8.24

Why do the observations stop after the year 2000?

Vintage Wine Data, cont.



+

Linear Regression

Linear Regression

- Predict the value of the *dependent variable*:
  - Log(price index)
- Prediction as a linear function of the *independent variables*:
  - Winter rain (mm)
  - Harvest rain (mm)
  - Average Temperature in growing season (Celsius)
  - Average Temperature in harvest season (Celsius)
  - Age of wine (years since vintage)
  - Population of France
  - US alcohol consumption (per capita, in liters of 100% alcohol)

Multiple Linear Regression

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \epsilon$$

- Parametric method
- Observed data: $(x_i, y_i) \quad i = 1, \ldots, n$
- Each observed x_i is a feature vector: $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$
- Each observed y_i is a continuous response/dependent variable associated with x_i

+

Statistical Learning Interlude

General Statistical Learning Model

- Input variables: $X = (X_1, X_2, \ldots, X_p)$
  - Also often called features, predictors, or independent variables
- Output variable: Y
  - Also often called response or dependent variable
- Collected data in the form of n pairs:
  - $(x_i, y_i) \quad i = 1, \ldots, n$
  - $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$

A General Statistical Learning Model

- We presume that there is some relationship between X and Y:

$$Y = f(X) + \epsilon$$

- ϵ is a random error term that is **independent** of X and has mean 0
- f is a fixed but **unknown** function that represents the **systematic** information that X provides about Y
- (Supervised) statistical learning is a set of tools for estimating f

Why estimate f ?

- Two reasons to estimate f
  - Prediction
  - Inference
- If we have a good estimate for f, call it $\hat{f}$, then we can use $\hat{f}$ to make a prediction for a new value of X

$$\hat{Y} = \hat{f}(X)$$

- Think of the advertising example

Statistical Learning for Prediction

- True model: $Y = f(X) + \epsilon$
- Our prediction: $\hat{Y} = \hat{f}(X)$
- What does the accuracy of our prediction depend on?
  - Reducible error: $\hat{f}$ is not a perfect estimate of f
  - Irreducible error: $\mathrm{Var}(\epsilon) > 0$
- The aim of statistical learning techniques is to reduce the reducible error!

Statistical Learning for Inference

- Inference: how does Y change when X changes?
  - Which predictor variables are associated with the response?
  - What is the relationship between the response and each associated predictor? Positive or negative?
  - Is a linear equation adequate to describe the relationship between X and Y?
- These are all essentially questions about the **behavior** (e.g., slope) of f
- In this course, we will mostly be concerned with prediction, but inference is important!

How do we estimate f?

- As always, we start with data:
  - $(x_i, y_i) \quad i = 1, \ldots, n$
  - Often called the **training data**
- A statistical learning method is a procedure, applied to the training data, for estimating f
  - We'll cover a lot of these methods in this course
- Broadly speaking, two classes of methods:
  - Parametric methods
  - Non-parametric methods

Parametric Methods

- Start by assuming a particular functional form for f
  - For example, assume that f is linear:

$$f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

  - f is parameterized by $\beta = (\beta_0, \beta_1, \ldots, \beta_p)$
- Now apply a method that uses the training data to estimate β
  - We sometimes call this fitting the model
  - Classic example: ordinary least squares, i.e., linear regression
  - We will consider more sophisticated approaches as well

Parametric Methods

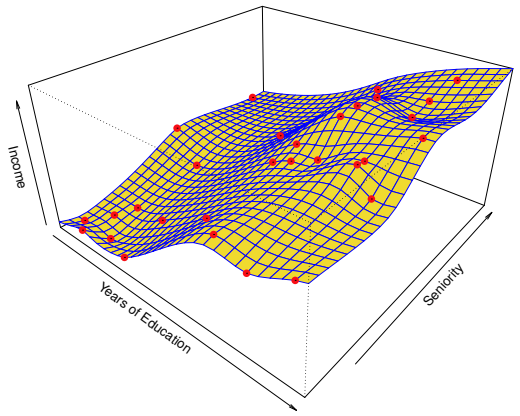
- Advantages of Parametric Methods:
  - Simplifies the problem of estimating f to the problem of estimating β
  - Potentially relatively less data needed to produce a reliable estimate of β
- Major Disadvantage of Parametric Methods:
  - The true functional form of f is usually more complicated than the model we chose
  - This may be remedied by selecting a flexible model class, but this comes at the danger of *overfitting*

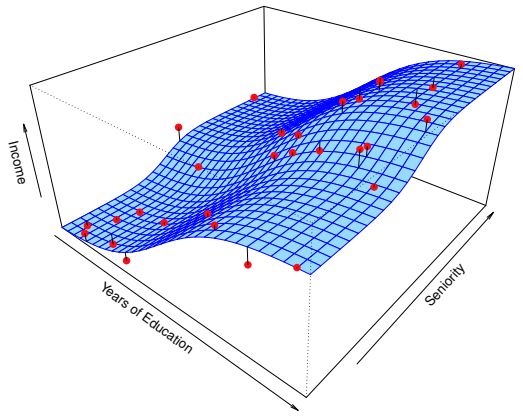
Non-parametric Methods

- Of course, non-parametric methods do not make parametric assumptions about f
- No explicit functional form is assumed
  - Allows for greater **flexibility**
  - Runs a greater risk of overfitting if you are not careful
  - Generally requires more data to produce an accurate estimate

Tradeoff Between Flexibility and Interpretability

- Why not just always use flexible, non-parametric methods?
  - One reason is that parametric models are more interpretable and thus better for inference
  - Even if you don't care about inference, non-parametric methods may overfit the training data





+

Back to Linear Regression

Linear Regression

- Predict the value of the *dependent variable*:
  - Log(price index)
- Prediction as a linear function of the *independent variables*:
  - Winter rain (mm)
  - Harvest rain (mm)
  - Average Temperature in growing season (Celsius)
  - Average Temperature in harvest season (Celsius)
  - Age of wine (years since vintage)
  - Population of France
  - US alcohol consumption (per capita, in liters of 100% alcohol)

Multiple Linear Regression

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \epsilon$$

- Parametric method
- Observed data: $(x_i, y_i) \quad i = 1, \ldots, n$
- Each observed x_i is a feature vector: $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$
- Each observed y_i is a continuous response/dependent variable associated with x_i

Multiple Linear Regression, cont.

- The (true) regression coefficients $\beta = (\beta_0, \beta_1, \ldots, \beta_p)$ are unknown to us
- How do we estimate the regression coefficients?
- Minimize prediction error, as measured by the **residual sum of squares (RSS):**

$$\mathrm{RSS}(\beta) := \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \ldots - \beta_p x_{ip})^2$$

Unconstrained Optimization Review

- Ingredients:
  - $w = (w_1, w_2, \ldots, w_m)^T \in \mathbb{R}^m$ is a vector of decision variables (often called parameters in ML/Stats)
  - $g : \mathbb{R}^m \to \mathbb{R}$ is the objective function (often called loss function or penalty function)
- Optimization problem:

$$\min_{w \in \mathbb{R}^m} g(w)$$

Unconstrained Optimization Review, cont.

- Optimization problem:

$$\min_{w \in \mathbb{R}^m} g(w)$$

- Definition of optimality: w^* solves the above optimization problem if and only if $g(w) \geq g(w^*)$ for all $w \in \mathbb{R}^m$

- Necessary Optimality Condition: If g is differentiable with gradient $\nabla g(w)$ and w^* solves the optimization problem, then:

$$\nabla g(w^*) = 0$$

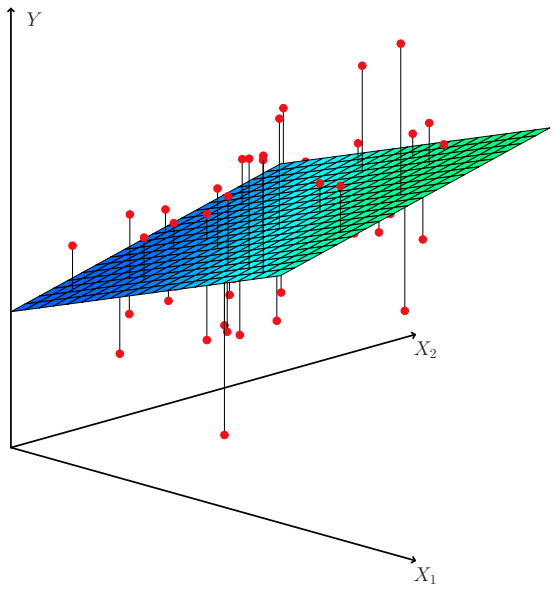
Multiple Linear Regression Coefficient Estimates

- The regression coefficient estimates $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ are chosen to minimize $\mathrm{RSS}(\beta)$

$$\min_{\beta \in \mathbb{R}^p} \mathrm{RSS}(\beta)$$

- Where:

$$\mathrm{RSS}(\beta) := \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \ldots - \beta_p x_{ip})^2$$



Multiple Linear Regression Coefficient Estimates

- Let $\mathbf{X}$ be the *n x p* matrix where the i^{th} row is the feature vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$
- Let $\mathbf{y}$ be the *n*-vector of responses y_i
- Then $\text{RSS}(\beta) = \|\mathbf{y} - \mathbf{X}\beta\|_2^2$ and (if $\text{rank}(\mathbf{X}) = p$), then one may use calculus/linear algebra to show that the solution of $\min_{\beta \in \mathbb{R}^p} \text{RSS}(\beta)$ is given by:

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Multiple Linear Regression, cont.

- **Prediction for the ith observation:**

$$\hat{y}_i := \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \ldots + \hat{\beta}_p x_{ip}$$

- **Residuals:** $e_i = y_i - \hat{y}_i$

- **RSS with respect to the estimated coefficients:**

$$\text{RSS} = \text{SSE} = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- SSE is the sum of squared errors (both conventions often used)

Vintage Wine Data

p = # of independent variables (p=7); n = # of observations (n=46)

n=46

		y	x_1	x_2	x_3	x_4	x_5	x_6	x_7
		LogAuctionIndex	WinterRain	HarvestRain	GrowTemp	HarvestTemp	Age	FrancePop	USAlcConsump
1	1952	y_1 = 7.4950	x_{11} = 566.4	x_{21} = 165.5	x_{31} = 17.28	x_{41} = 14.39	x_{51} = 63	x_{61} = 42.46	x_{71} = 7.85
2	1953	y_2 = 8.0393	x_{12} = 653.3	x_{22} = 75.6	x_{32} = 16.94	x_{42} = 17.64	x_{52} = 62	x_{62} = 42.75	x_{72} = 8.03
3	1955	7.6858	504.3	129.5	17.30	17.13	60	43.43	7.84
4	1957	6.9845	390.8	110.4	16.31	16.47	58	44.31	7.77
5	1958	6.7772	538.8	187.0	16.82	19.72	57	44.79	7.74
6	1959	8.0757	377.0	182.6	17.68	19.28	56	45.24	7.89
7	1960	6.5188	748.2	290.6	16.67	16.18	55	45.68	8.02
8	1961	8.4937	747.8	37.7	17.64	21.05	54	46.16	8.08
9	1962	7.3880	639.4	51.8	16.58	17.86	53	47.00	8.13
10	1964	7.3094	326.5	96.1	17.63	19.43	51	48.31	8.46
11	1965	6.2518	548.4	266.6	15.71	15.33	50	48.76	8.62
12	1966	7.7443	734.0	85.2	16.81	18.82	49	49.16	8.78
13	1967	6.8398	646.9	118.1	16.51	17.16	48	49.55	9.03
14	1968	6.2435	508.6	292.1	16.37	16.77	47	49.91	9.28
15	1969	6.3459	480.1	243.9	16.65	16.89	46	50.32	9.53
16	1970	7.5883	563.5	88.8	16.92	18.69	45	50.77	9.78
17	1971	7.1934	488.4	111.9	17.20	17.28	44	51.25	9.99
18	1972	6.2049	465.1	157.3	15.27	15.04	43	51.70	10.10
19	1973	6.6367	357.2	122.6	17.41	18.50	42	52.12	10.37
20	1974	6.2941	503.6	185.1	16.39	16.48	41	52.46	10.48
	...	...	...	...	...	...	...	...	...
46	2000	y_n = 8.1817	x_{1n} = 487.8	x_{2n} = 69.0	x_{3n} = 18.73	x_{4n} = 19.45	x_{5n} = 15	x_{6n} = 59.05	x_{7n} = 8.24

Best Practices: Out-of-Sample Testing

Full Dataset (n = 46)

	LogAuctionIndex	WinterRain	HarvestRain	GrowTemp	...
1 1952	7.4950	566.4	165.5	17.28	...
2 1953	8.0393	653.3	75.6	16.94	...
3 1955	7.6858	504.3	129.5	17.30	...
4 1957	6.9845	390.8	110.4	16.31	...
5 1958	6.7772	538.8	187.0	16.82	...
6 1959	8.0757	377.0	182.6	17.68	...
7 1960	6.5188	748.2	290.6	16.67	...
8 1961	8.4937	747.8	37.7	17.64	...
9 1962	7.3880	639.4	51.8	16.58	...
10 1964	7.3094	326.5	96.1	17.63	...
...	...	...	...	...	...
30 1984	6.5496	572.6	144.8	16.71	...
31 1985	6.9171	667.1	37.2	17.19	...
32 1986	6.7793	518.5	171.2	16.65	...
33 1987	7.1797	397.0	115.1	17.84	...
34 1988	7.2646	734.2	58.8	17.65	...
35 1989	7.5922	282.4	85.2	18.62	...
...	...	...	...	...	...
45 1999	7.4462	502.4	253.4	19.07	...
46 2000	8.1817	487.8	69.0	18.73	...

Training Set (n = 31)

	LogAuctionIndex	WinterRain	HarvestRain	GrowTemp	...
1 1952	7.4950	566.4	165.5	17.28	...
2 1953	8.0393	653.3	75.6	16.94	...
3 1955	7.6858	504.3	129.5	17.30	...
4 1957	6.9845	390.8	110.4	16.31	...
5 1958	6.7772	538.8	187.0	16.82	...
6 1959	8.0757	377.0	182.6	17.68	...
7 1960	6.5188	748.2	290.6	16.67	...
8 1961	8.4937	747.8	37.7	17.64	...
9 1962	7.3880	639.4	51.8	16.58	...
10 1964	7.3094	326.5	96.1	17.63	...
...	...	...	...	...	...
30 1984	6.5496	572.6	144.8	16.71	...
31 1985	6.9171	667.1	37.2	17.19	...

Testing Set (n = 15)

	LogAuctionIndex	WinterRain	HarvestRain	GrowTemp	...
32 1986	6.7793	518.5	171.2	16.65	...
33 1987	7.1797	397.0	115.1	17.84	...
34 1988	7.2646	734.2	58.8	17.65	...
35 1989	7.5922	282.4	85.2	18.62	...
...	...	...	...	...	...
45 1999	7.4462	502.4	253.4	19.07	...
46 2000	8.1817	487.8	69.0	18.73	...

Best Practices: Out-of-Sample Testing

- Set aside a "test set" of 20% – 50% of the observed data **before** creating the regression model(s)
- Typical practice: set aside the most recently observed data (for example the markets most recently entered or wines most recently matured)
- If there is no time-dependence in the observed data, select a random sample for the test set
- Keep the test set data "hands-off" until you are ready to asses the performance of your regression model

Best Practices: Out-of-Sample Testing

- Seriously, only use the test set once, when you have finished training your model, to estimate the performance of the model when you go to apply it in the real world
- All data used to help build the model is training data, and the training error (RSS) typically **underestimates** the performance error
- Soon in the course we will see how to use some of the training data as "validation data" to estimate the performance error during the training phase

Regression Output and Analysis

Regression Output (from R)

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.9662699  9.3823951  -0.529  0.60166
WinterRain   0.0011863  0.0005628   2.108  0.04616 *
HarvestRain -0.0033137  0.0010650  -3.112  0.00491 **
GrowTemp     0.6582753  0.1221937   5.387 1.79e-05 ***
HarvestTemp  0.0044212  0.0599935   0.074  0.94189
Age          0.0240080  0.0507587   0.473  0.64068
FrancePop   -0.0290258  0.1369627  -0.212  0.83403
USAlcConsump 0.1092561  0.1678945   0.651  0.52166
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3307 on 23 degrees of freedom
Multiple R-squared:  0.7894,   Adjusted R-squared:  0.7253
F-statistic: 12.31 on 7 and 23 DF,  p-value: 1.859e-06
```

Interpreting the Regression Coefficients

- Regression coefficients: $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$

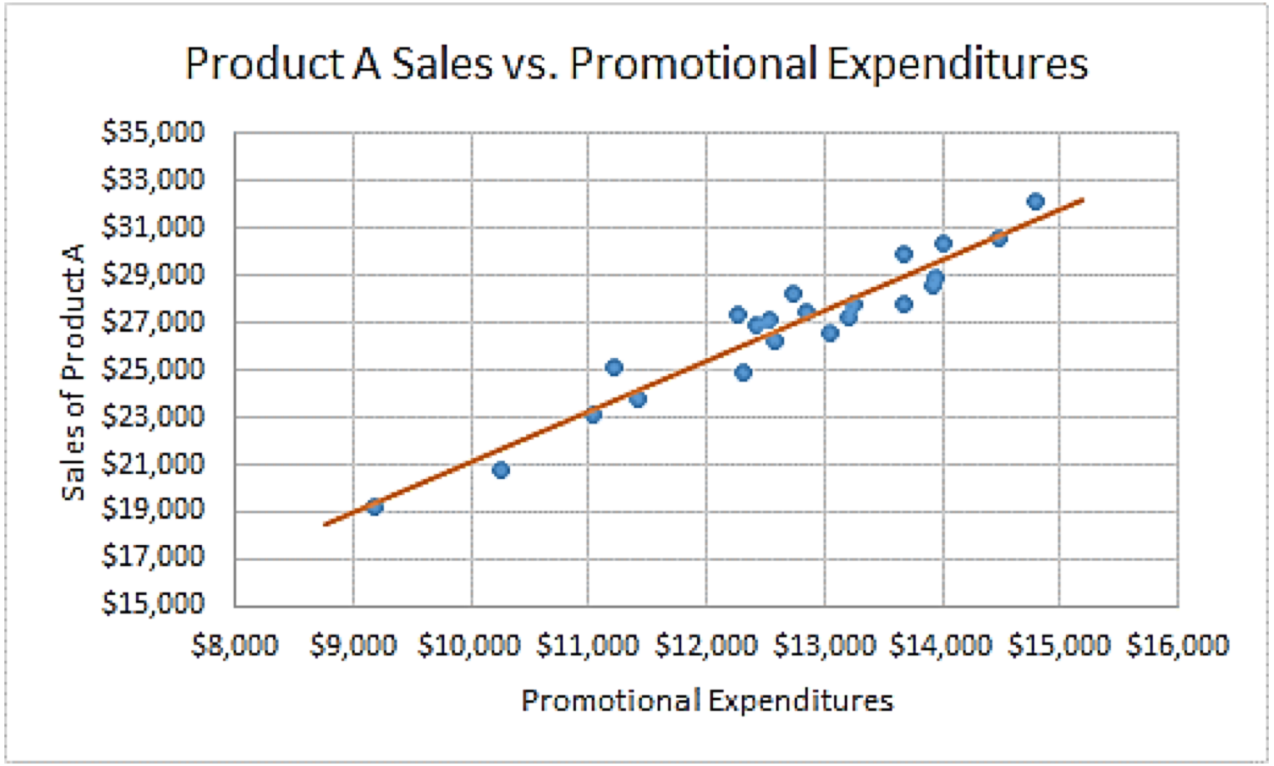
  are estimates of $\beta = (\beta_0, \beta_1, \ldots, \beta_p)$

  - $\hat{\beta}_0$ = -4.966
  - $\hat{\beta}_{\text{winter-rain}}$ = 0.0012 (An additional mm of winter rain is expected to result in an additional 0.0012 units of log(price index))
  - $\hat{\beta}_{\text{harvest-rain}}$= -0.0033 (An additional mm of winter rain is expected to result in a decrease of 0.0033 units of log(price index))
  - ….
  - $\hat{\beta}_{\text{USalc}}$ = 0.1093 (An additional liter of U.S. per capita alcohol consumption is expected to result in a increase of 0.1093 units of the log(price index))

Understanding R^2

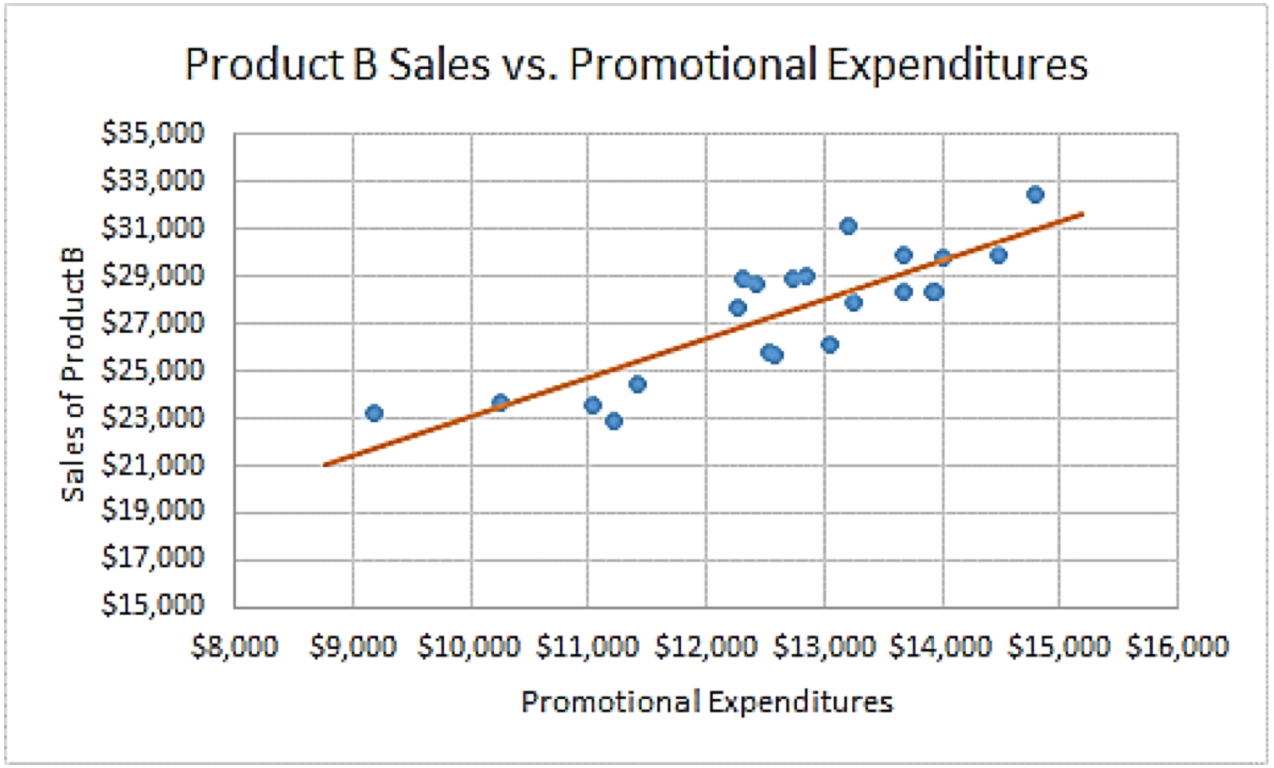
- R^2 is the **coefficient of determination**
- R^2 is a measure of the overall quality of the regression model
- R^2 is a number between 0.0 and 1.0
- A higher R^2 means the regression model is a better fit to the (training) data

Understanding R^2 , cont.



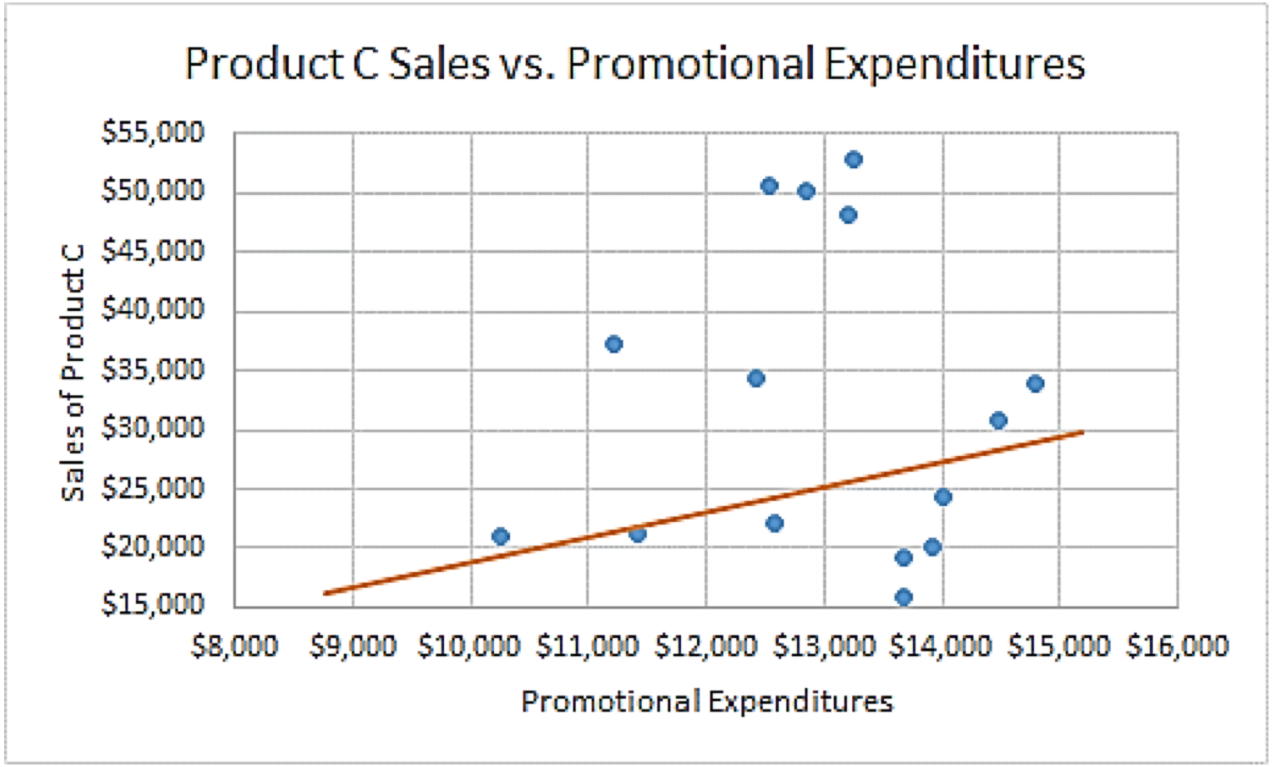
- R^2 = .924; very good linear model

Understanding R^2 , cont.



- R^2 = .710; good linear model

Understanding R^2 , cont.



- R^2 = .035; not a good model

What really is R^2 ?

- R^2 compares two models:
  - the regression model (the one determined by minimizing the RSS (residual sum of squares error), and
  - the "baseline" model. Think of the baseline model as a model you might have built using this data but without any real mathematical thinking.
- The **baseline model** predicts simplistically using only the mean/average of the sample outcomes:

$$\bar{y} = \frac{y_1 + \cdots + y_n}{n} = \frac{y_{1952} + \cdots + y_{1985}}{31} = 7.084$$

What really is R^2 , continued

$$R^2 = 1 - \frac{\text{Sum of squared residuals of regression model}}{\text{Sum of squared residuals of baseline model}}$$

$$= 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$$= 1 - \frac{SSE}{SST}$$

$$SST = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

Regression Output (from R)

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -4.9662699  9.3823951  -0.529  0.60166
WinterRain    0.0011863  0.0005628   2.108  0.04616 *
HarvestRain  -0.0033137  0.0010650  -3.112  0.00491 **
GrowTemp      0.6582753  0.1221937   5.387 1.79e-05 ***
HarvestTemp   0.0044212  0.0599935   0.074  0.94189
Age           0.0240080  0.0507587   0.473  0.64068
FrancePop    -0.0290258  0.1369627  -0.212  0.83403
USAlcConsump  0.1092561  0.1678945   0.651  0.52166
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3307 on 23 degrees of freedom
Multiple R-squared:  0.7894,    Adjusted R-squared:  0.7253
F-statistic: 12.31 on 7 and 23 DF,  p-value: 1.859e-06
```

Vintage Wine Data

p = # of independent variables (p=7); n = # of observations (n=46)

n=46

		y	x_1	x_2	x_3	x_4	x_5	x_6	x_7
		LogAuctionIndex	WinterRain	HarvestRain	GrowTemp	HarvestTemp	Age	FrancePop	USAlcConsump
1	1952	y_1 = 7.4950	x_{11} = 566.4	x_{21} = 165.5	x_{31} = 17.28	x_{41} = 14.39	x_{51} = 63	x_{61} = 42.46	x_{71} = 7.85
2	1953	y_2 = 8.0393	x_{12} = 653.3	x_{22} = 75.6	x_{32} = 16.94	x_{42} = 17.64	x_{52} = 62	x_{62} = 42.75	x_{72} = 8.03
3	1955	7.6858	504.3	129.5	17.30	17.13	60	43.43	7.84
4	1957	6.9845	390.8	110.4	16.31	16.47	58	44.31	7.77
5	1958	6.7772	538.8	187.0	16.82	19.72	57	44.79	7.74
6	1959	8.0757	377.0	182.6	17.68	19.28	56	45.24	7.89
7	1960	6.5188	748.2	290.6	16.67	16.18	55	45.68	8.02
8	1961	8.4937	747.8	37.7	17.64	21.05	54	46.16	8.08
9	1962	7.3880	639.4	51.8	16.58	17.86	53	47.00	8.13
10	1964	7.3094	326.5	96.1	17.63	19.43	51	48.31	8.46
11	1965	6.2518	548.4	266.6	15.71	15.33	50	48.76	8.62
12	1966	7.7443	734.0	85.2	16.81	18.82	49	49.16	8.78
13	1967	6.8398	646.9	118.1	16.51	17.16	48	49.55	9.03
14	1968	6.2435	508.6	292.1	16.37	16.77	47	49.91	9.28
15	1969	6.3459	480.1	243.9	16.65	16.89	46	50.32	9.53
16	1970	7.5883	563.5	88.8	16.92	18.69	45	50.77	9.78
17	1971	7.1934	488.4	111.9	17.20	17.28	44	51.25	9.99
18	1972	6.2049	465.1	157.3	15.27	15.04	43	51.70	10.10
19	1973	6.6367	357.2	122.6	17.41	18.50	42	52.12	10.37
20	1974	6.2941	503.6	185.1	16.39	16.48	41	52.46	10.48
	...	...	...	...	...	...	...	...	...
46	2000	y_n = 8.1817	x_{1n} = 487.8	x_{2n} = 69.0	x_{3n} = 18.73	x_{4n} = 19.45	x_{5n} = 15	x_{6n} = 59.05	x_{7n} = 8.24

Best Practices: Out-of-Sample Testing

Full Dataset (n = 46)

	LogAuctionIndex	WinterRain	HarvestRain	GrowTemp	...
1 1952	7.4950	566.4	165.5	17.28	...
2 1953	8.0393	653.3	75.6	16.94	...
3 1955	7.6858	504.3	129.5	17.30	...
4 1957	6.9845	390.8	110.4	16.31	...
5 1958	6.7772	538.8	187.0	16.82	...
6 1959	8.0757	377.0	182.6	17.68	...
7 1960	6.5188	748.2	290.6	16.67	...
8 1961	8.4937	747.8	37.7	17.64	...
9 1962	7.3880	639.4	51.8	16.58	...
10 1964	7.3094	326.5	96.1	17.63	...
...	...	...	...	...	...
30 1984	6.5496	572.6	144.8	16.71	...
31 1985	6.9171	667.1	37.2	17.19	...
32 1986	6.7793	518.5	171.2	16.65	...
33 1987	7.1797	397.0	115.1	17.84	...
34 1988	7.2646	734.2	58.8	17.65	...
35 1989	7.5922	282.4	85.2	18.62	...
...	...	...	...	...	...
45 1999	7.4462	502.4	253.4	19.07	...
46 2000	8.1817	487.8	69.0	18.73	...

Training Set (n = 31)

	LogAuctionIndex	WinterRain	HarvestRain	GrowTemp	...
1 1952	7.4950	566.4	165.5	17.28	...
2 1953	8.0393	653.3	75.6	16.94	...
3 1955	7.6858	504.3	129.5	17.30	...
4 1957	6.9845	390.8	110.4	16.31	...
5 1958	6.7772	538.8	187.0	16.82	...
6 1959	8.0757	377.0	182.6	17.68	...
7 1960	6.5188	748.2	290.6	16.67	...
8 1961	8.4937	747.8	37.7	17.64	...
9 1962	7.3880	639.4	51.8	16.58	...
10 1964	7.3094	326.5	96.1	17.63	...
...	...	...	...	...	...
30 1984	6.5496	572.6	144.8	16.71	...
31 1985	6.9171	667.1	37.2	17.19	...

Testing Set (n = 15)

	LogAuctionIndex	WinterRain	HarvestRain	GrowTemp	...
32 1986	6.7793	518.5	171.2	16.65	...
33 1987	7.1797	397.0	115.1	17.84	...
34 1988	7.2646	734.2	58.8	17.65	...
35 1989	7.5922	282.4	85.2	18.62	...
...	...	...	...	...	...
45 1999	7.4462	502.4	253.4	19.07	...
46 2000	8.1817	487.8	69.0	18.73	...

Training vs. Test Data

- Is R^2 really what we care about?
- R^2 is measured on the training data, the data that we used to fit the model
- What we really care about is predictive performance on *new* data
- Recall that we set aside some test data…
- We will use this test data to estimate the performance of our model on new data that we might see in the wild

Assessing "Real World" Performance of the Regression Model

- Here is our model, based on the training data observations (years 1952 through 1985):
  - **log(Price Index) = -4.966 + 0.001*(Winter Rain) - 0.003*(Harvest Rain) + 0.658*(Growing Temp) + 0.004*(Harvest Temp) + 0.024*(Age) - 0.029*(France Population) + 0.109*(US Alcohol)**
- Use the model to compute predictions and residuals for each observation in the test set (observation years 1986 through 2000)
  - Example: prediction for year 1998:

    6.932 = -4.966 + 0.001*(693.4) + … + 0.109*(8.10)
  - Actual 1998 log(Price Index) = 6.858
  - Residual = -0.074 = 6.858 – 6.932
- How good is this prediction? Well, let's look at all of the test set data records and compute a version of R^2 , which we call OSR^2

Out-of-Sample R^2 (OSR^2)

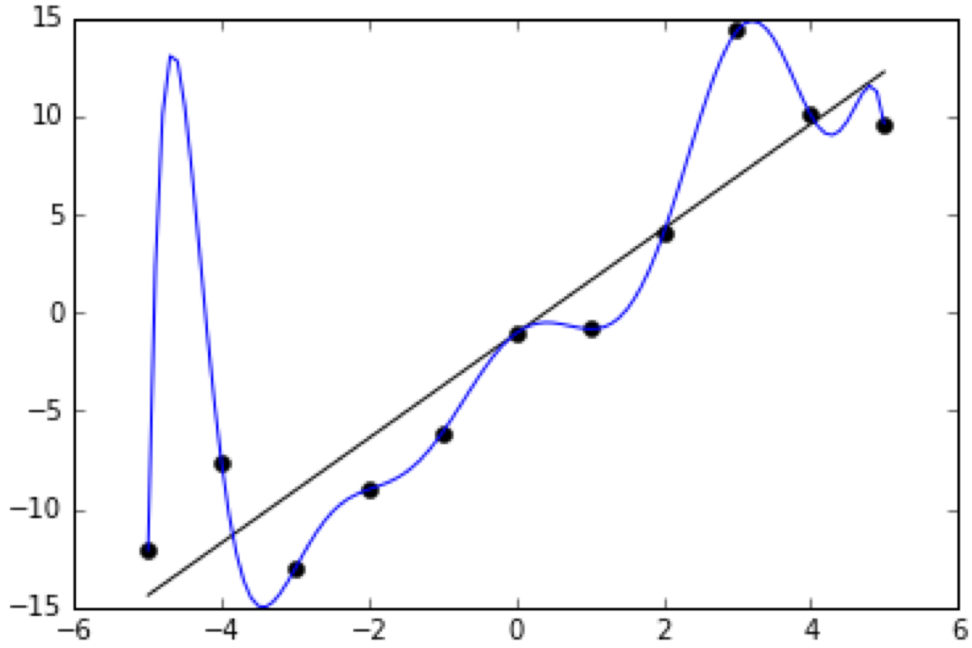
$$
\begin{aligned}
OSR^2 &= 1 - \frac{\text{Sum of squared residuals of regression model on the Test Set}}{\text{Sum of squared residuals of baseline model applied to the Test Set}} \\
&= 1 - \frac{\sum_{t=1986}^{2000} (y_t - \hat{y}_t)^2}{\sum_{t=1986}^{2000} (y_t - 7.084)^2} \\
&= 0.54
\end{aligned}
$$

Out-of-Sample R^2 (OSR^2)

- OSR^2 is an assessment of the real-world performance of the model we have built
- It should only be computed once, at the end of your analysis, as a final metric
- If OSR^2 is significantly smaller than R^2 (on the training data), this is an indicator of potential overfitting

Overfitting

- Overfitting occurs when the estimated model fits the noise in the training data
- All statistical learning methods are at risk for overfitting



Overfitting

- Overfitting is more likely when:
  - The number of parameters to be estimated is large
  - Data is limited
- Care must be taken to make sure that the model we estimate does not suffer from overfitting
  - We will see how to address this issue throughout the course, including today's lecture
- Overfitting is related to the "bias-variance tradeoff"

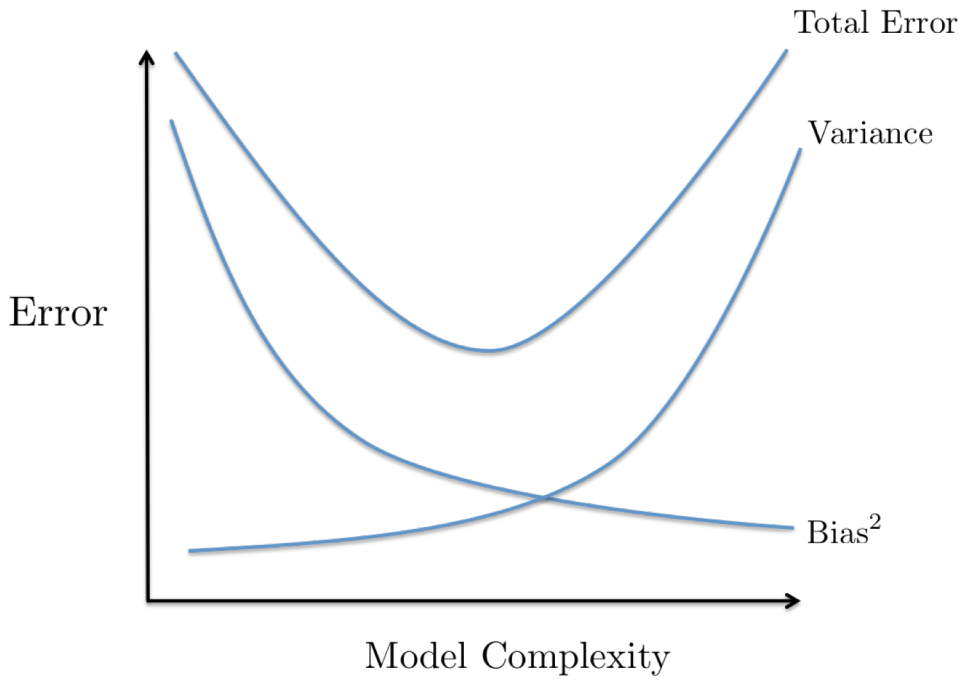
Flexible Statistical Learning Methods

- Flexible (usually non-parametric) statistical learning methods are able to capture complicated relationships
- Linear regression is relatively inflexible
- Flexibility usually implies that:
  - The resulting model is less interpretable
  - The method requires more data to produce an accurate estimate than a less flexible method
  - There is an increased risk of overfitting
- We will see examples of flexible, non-parametric methods later in the course

Bias and Variance of Learning Methods

- Bias refers to the error that is introduced by modeling a complicated relationship with a simple one
  - Less flexible methods have more bias
- Variance refers to the amount that our estimated function changes when you slightly change the dataset
  - More flexibility usually comes at the cost of higher variance
- The bias-variance tradeoff is a common theme in this course that we will continue discussing

The Bias-Variance Tradeoff



Error is measured on a test set

"Model Complexity" is a synonym for "Model Flexibility"

- Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani

- Thanks to Rob Freund and John Silberholz (MIT) for the wine datasets