Data Processing and Analysis in Python
Lecture 19
Regression Analysis

UNIVERSITY OF MARYLAND

ROBERT H. SMITH SCHOOL OF BUSINESS

DR. ADAM LEE

Regression Analysis

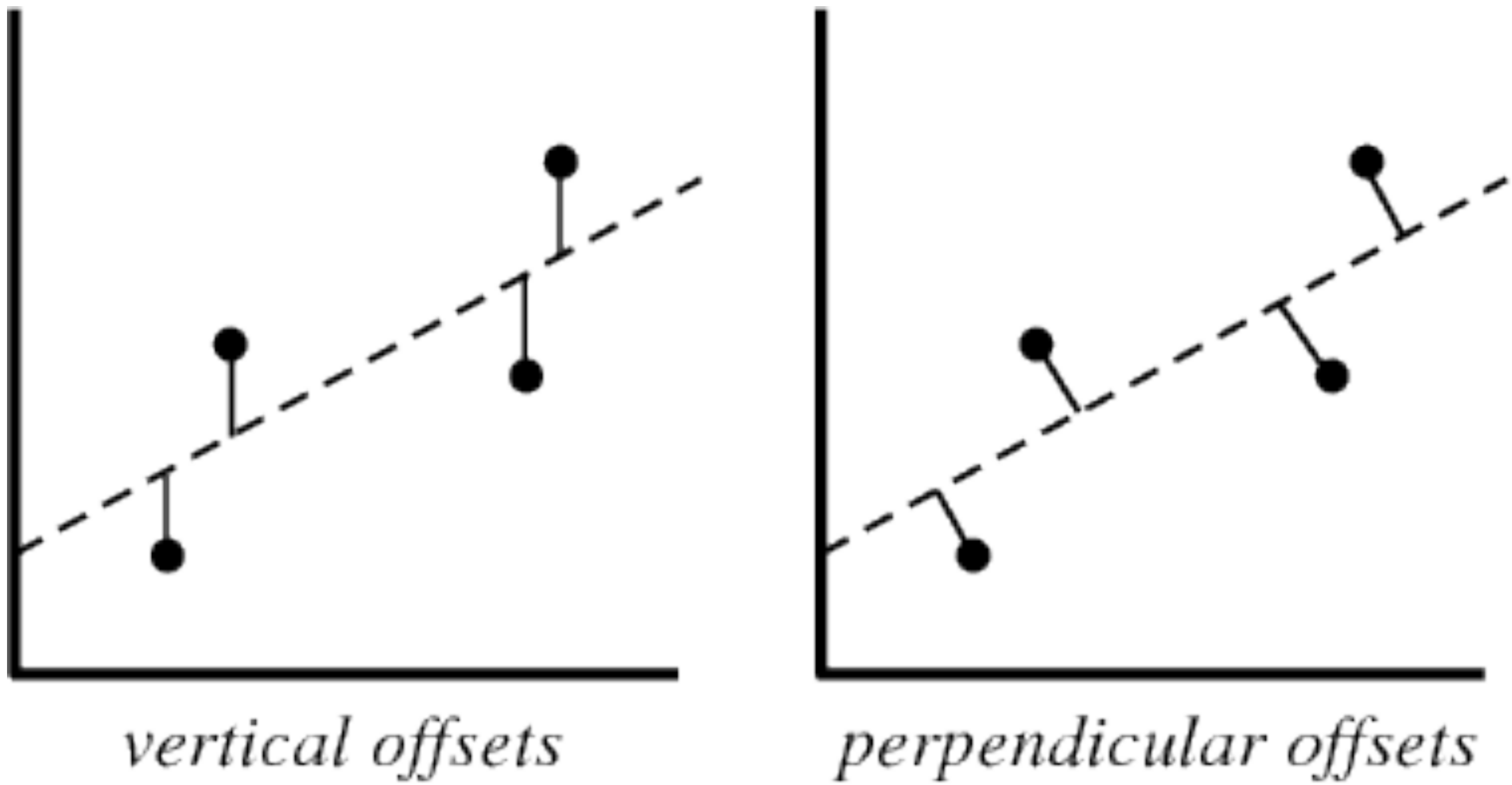
- To estimate the relationships among variables.
- To estimate a continuous dependent variable from a number of independent variables.
- The estimation target is a regression function of the independent variables.

- What differs from the **curve fitting**?
  - line fitting, polynomial fitting, geometric fitting, function fitting, etc.

Regression Techniques

- Parametric regression:
  - linear regression, ordinary least squares regression, etc.
- Semi-parametric regression:
  - partially linear model, index model, etc.
- Nonparametric regression:
  - Gaussian process regression, kernel regression, etc.

Errors



UNIVERSITY OF MARYLAND
ROBERT H. SMITH SCHOOL OF BUSINESS

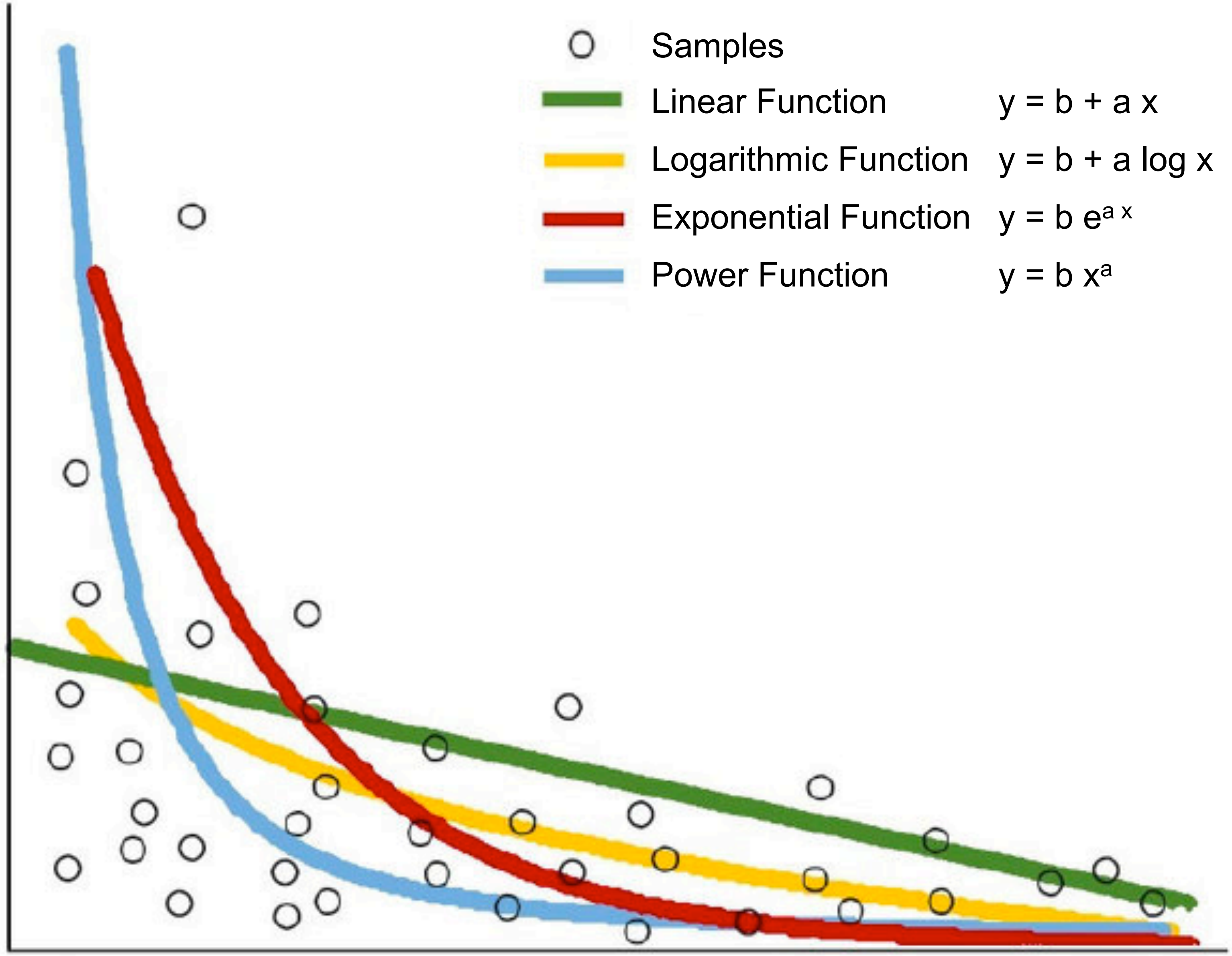
Estimation Accuracy

- Error (E_t) = estimated value (y_t') - actual value (y_t)
- Average of errors (ME) $$\bar{E} = \frac{\sum_t E_t}{N}$$
- Mean absolute error (MAE) $$MAE = \frac{\sum_t |E_t|}{N}$$
- Mean squared error (MSE) $$MSE = \frac{\sum_t E_t^2}{N}$$
- Root mean squared error (RMSE) $$RMSE = \sqrt{\frac{\sum_t E_t^2}{N}}$$

Price Elasticity of Demand (PED)

- A measure to show the responsiveness, or elasticity, of the quantity demanded of a good or service to a change in its price when nothing but the price changes
- Examples:
  - In economics, elasticity is a measure of how sensitive demand or supply is to price.
  - In marketing, it is how sensitive consumers are to a change in price of a product.

Regression



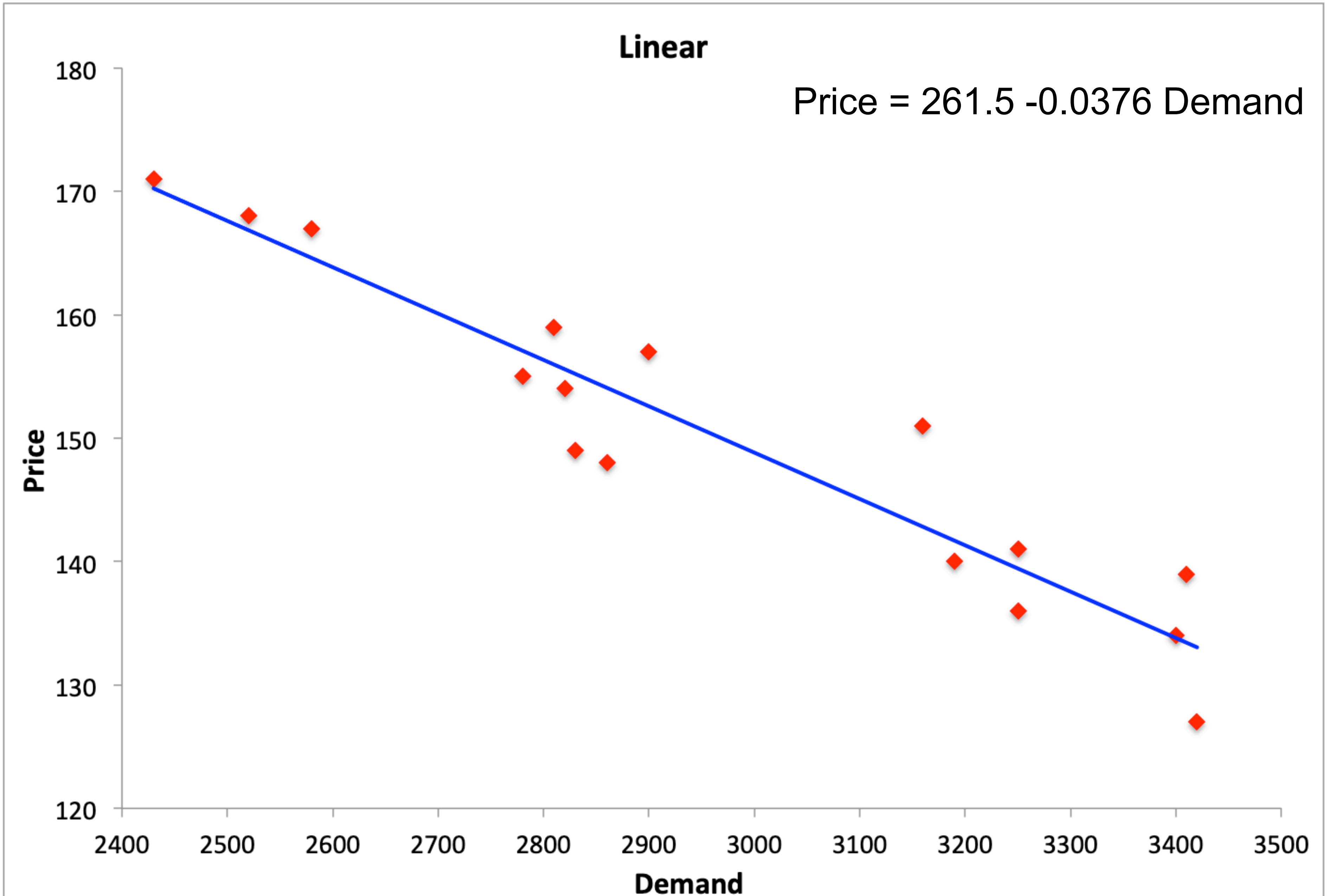
Samples
Linear Function $y = b + a x$
Logarithmic Function $y = b + a \log x$
Exponential Function $y = b e^{a x}$
Power Function $y = b x^a$

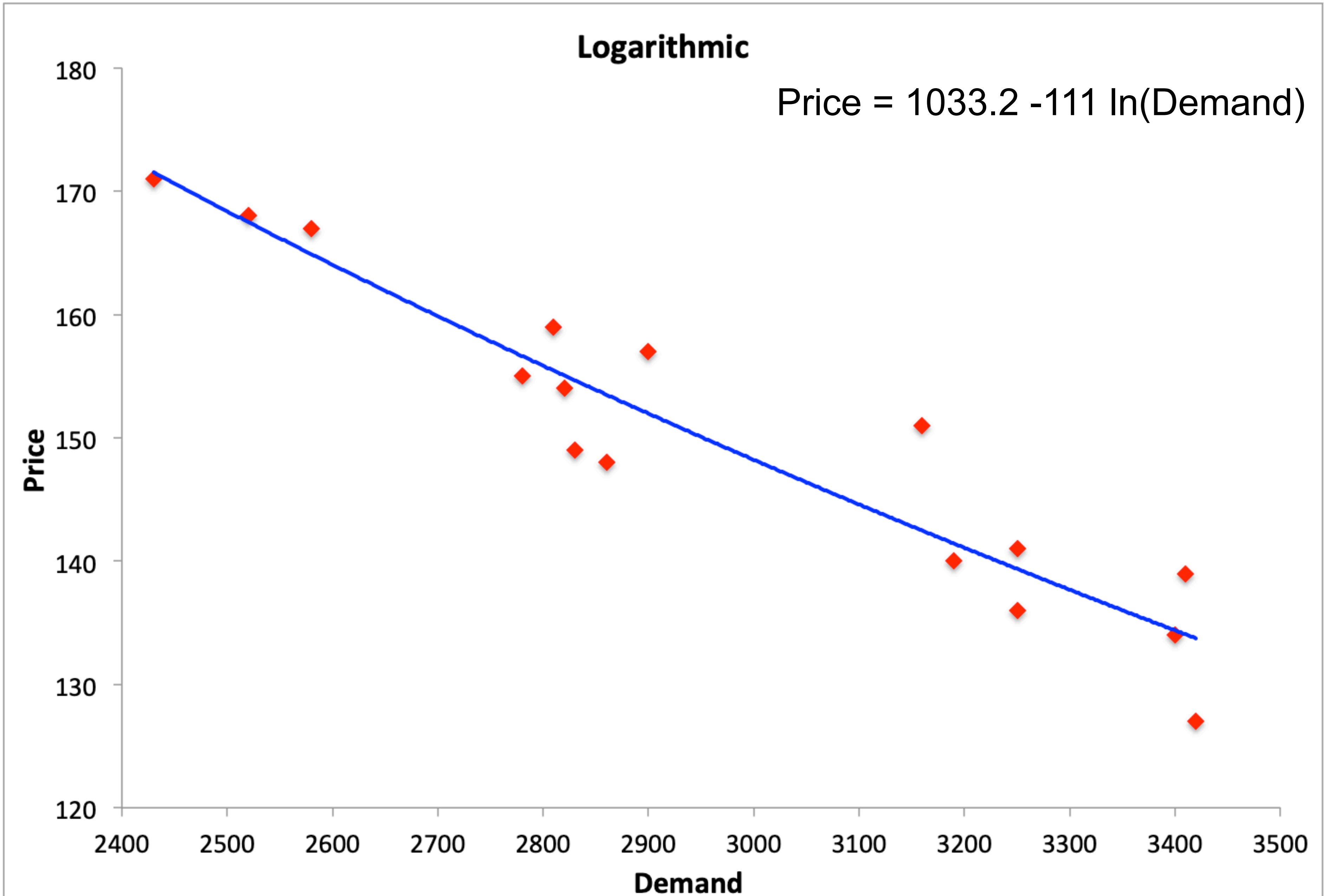
PED Example – Excel

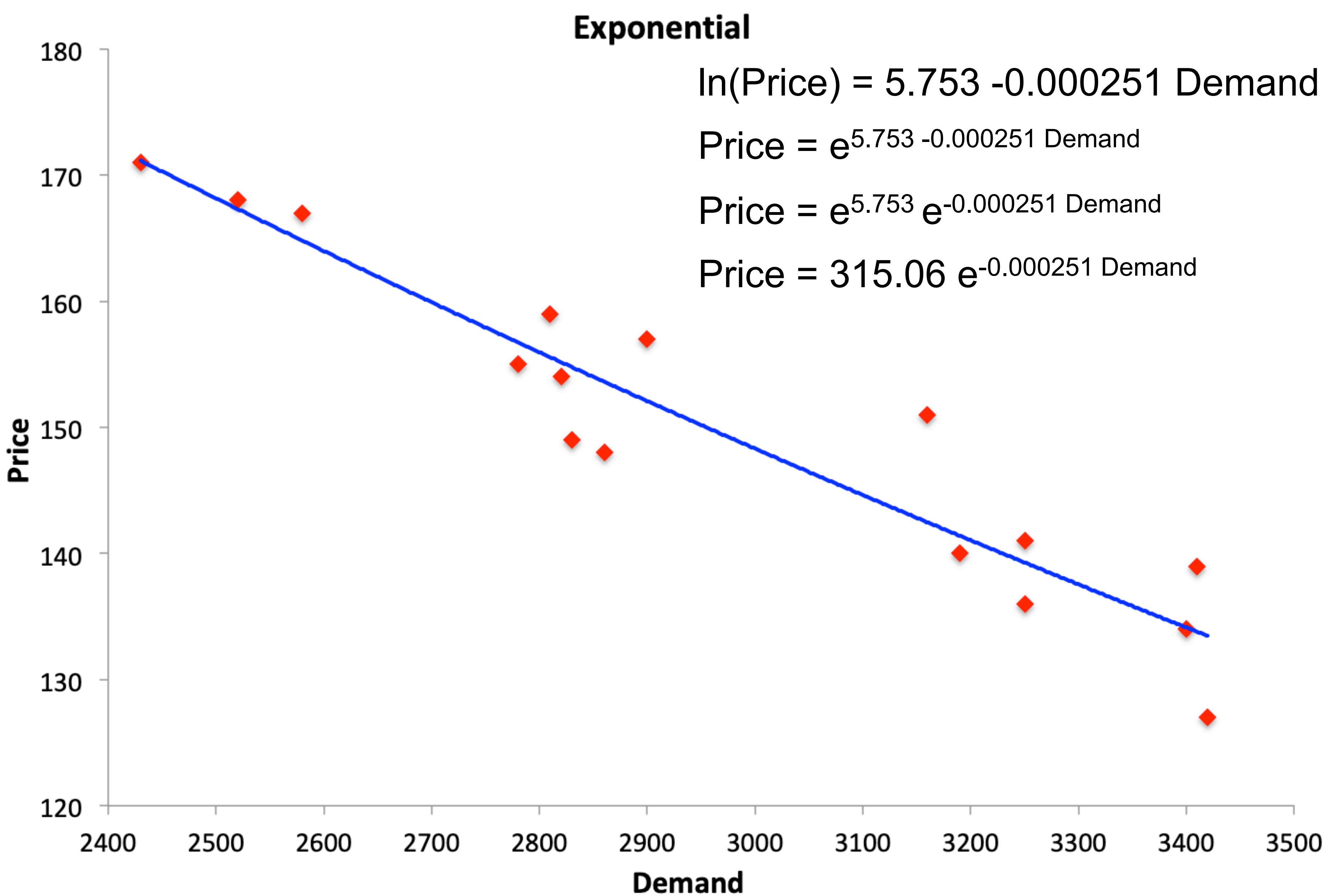
Price Elasticity of Demand (PED) Analysis Report

	Linear	Logarithmic	Exponential	Power
Intercept	261.537711	1033.23127	315.064425	54210.413
Slope	-0.0375678	-110.53858	-0.0002512	-0.7375878
ME	0.00079289	0.00080831	0.00040016	0.00040769
MAE	0.02320043	0.0225083	0.02241764	0.02244547
MSE	0.00080451	0.00080093	0.0007989	0.00081326
RMSE	0.02836384	0.02830071	0.02826476	0.02851775

Demand	Price	log(Demand	log(Price)	Error	Linear	Logarithmic	Exponential	Power
3420	127	8.13739583	4.84418709		0.0476835	0.05303193	0.05090782	0.05590373
3400	134	8.13153071	4.8978398		-0.0014391	0.00286102	0.00102521	0.00508322
3250	136	8.08641028	4.91265489		0.02531127	0.02478622	0.02416968	0.02381469
3410	139	8.13446757	4.93447393		-0.0400613	-0.0355486	-0.0374035	-0.0331675
3190	140	8.0677762	4.94164242		0.01211715	0.01021936	0.01001353	0.00832675
3250	141	8.08641028	4.94875989		-0.0110473	-0.0115537	-0.0121484	-0.0124908
2860	148	7.9585769	4.99721227		0.04117418	0.03717193	0.037977	0.03382601
2830	149	7.94803199	5.00394631		0.04175042	0.038034	0.03880823	0.03490565
3160	151	8.05832731	5.01727984		-0.0541494	-0.0564558	-0.0564814	-0.0585893
2820	154	7.94449216	5.0369526		0.01036682	0.00687242	0.00760813	0.00392257
2780	155	7.93020621	5.04342512		0.01354324	0.01056453	0.01121527	0.00801145
2900	157	7.97246602	5.05624581		-0.0280824	-0.0320627	-0.0313054	-0.0353708
2810	159	7.94093976	5.0689042		-0.019043	-0.0223206	-0.0216235	-0.0250962
2580	167	7.85554468	5.11799381		-0.0142948	-0.0126318	-0.0130991	-0.0114536
2520	168	7.83201418	5.12396398		-0.006745	-0.0030267	-0.0040785	-0.0001341
2430	171	7.79564654	5.14166356		-0.004398	0.00299145	0.00081752	0.0090312


Linear
Price = 261.5 -0.0376 Demand
Price
180
170
160
150
140
130
120
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
Demand


Logarithmic
Price = 1033.2 -111 ln(Demand)
Price
180
170
160
150
140
130
120
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
Demand

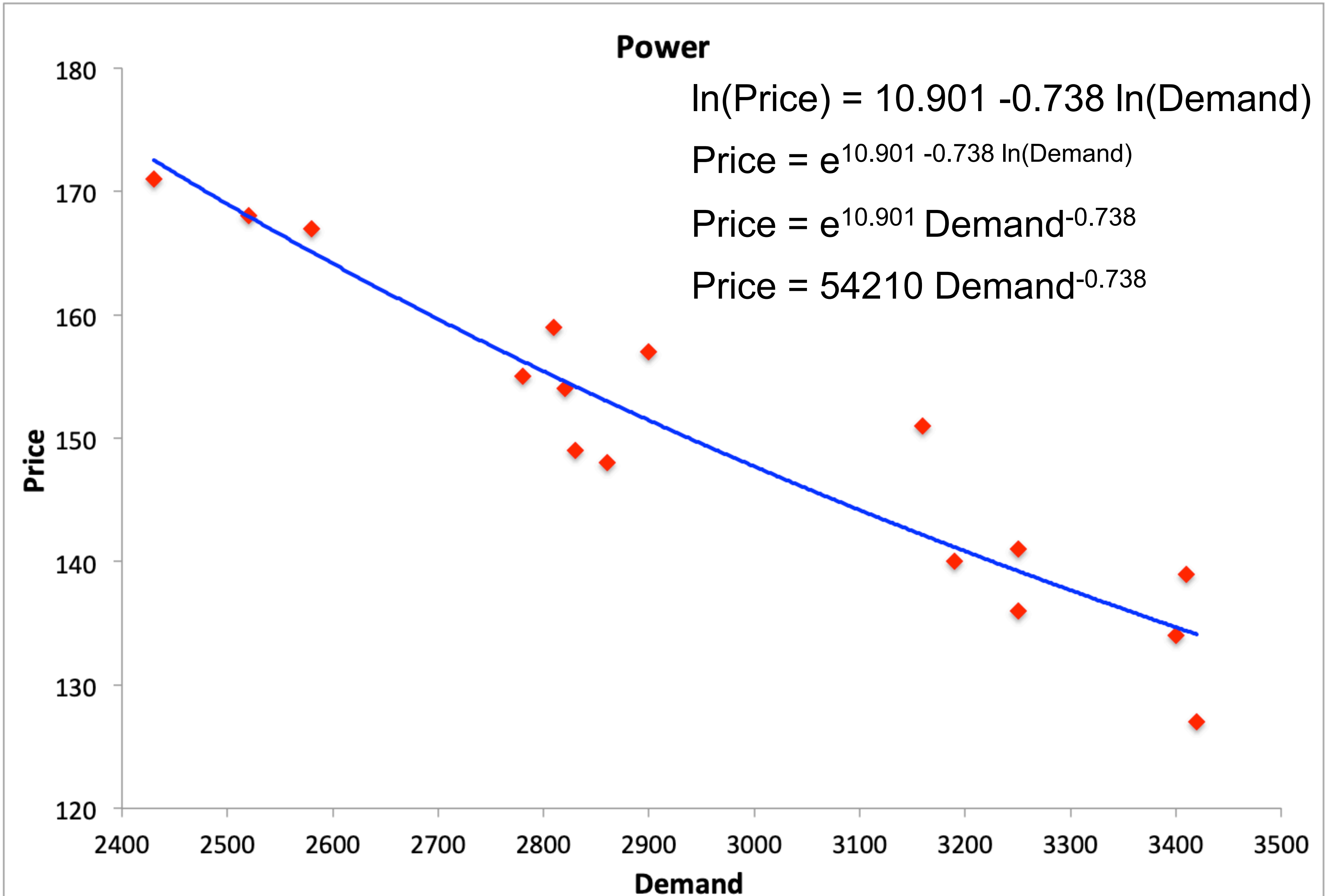


$$\ln(\text{Price}) = 5.753 - 0.000251\ \text{Demand}$$

$$\text{Price} = e^{5.753 - 0.000251\ \text{Demand}}$$

$$\text{Price} = e^{5.753}\, e^{-0.000251\ \text{Demand}}$$

$$\text{Price} = 315.06\, e^{-0.000251\ \text{Demand}}$$


Power
$\ln(\text{Price}) = 10.901 - 0.738 \ln(\text{Demand})$
$\text{Price} = e^{10.901 - 0.738 \ln(\text{Demand})}$
$\text{Price} = e^{10.901}\, \text{Demand}^{-0.738}$
$\text{Price} = 54210\, \text{Demand}^{-0.738}$
Price
180
170
160
150
140
130
120
2400
2500
2600
2700
2800
2900
3000
3100
3200
3300
3400
3500
Demand

PED Example – Python

```
import numpy as np
import pandas as pd
from scipy.stats import linregress
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt

# read data from file
dmpr = pd.read_csv("Demand Price.csv")
print(dmpr.describe())

# model comparison
```

PED Example – Python

```
# linear fit
linSlope, linIntercept, linR, linP, linSE = linregress
(dmpr["Demand"], dmpr["Price"])
print(linIntercept, linSlope)
att, var = curve_fit(lambda dem, itc, slp: itc + slp *
dem, dmpr["Demand"], dmpr["Price"])
print(att)
# logarithmic fit
logSlope, logIntercept, logR, logP, logSE = linregress
(np.log(dmpr["Demand"]), dmpr["Price"])
print(logIntercept, logSlope)
att, var = curve_fit(lambda dem, itc, slp: itc + slp *
np.log(dem), dmpr["Demand"], dmpr["Price"])
print(att)
```

PED Example – Python

```
# exponential fit
expSlope, expIntercept, expR, expP, expSE = linregress
(dmpr["Demand"], np.log(dmpr["Price"]))
print(np.exp(expIntercept), expSlope)
att, var = curve_fit(lambda dem, itc, slp: itc *
np.exp(slp * dem), dmpr["Demand"], dmpr["Price"])
print(att)
# power fit
powSlope, powIntercept, powR, powP, powSE = linregress
(np.log(dmpr["Demand"]), np.log(dmpr["Price"]))
print(np.exp(powIntercept), powSlope)
att, var = curve_fit(lambda dem, itc, slp: itc * dem **
slp, dmpr["Demand"], dmpr["Price"])
print(att)
```

PED Example – Python

```
# create figure for 2x2 grids
figure, axes = plt.subplots(nrows=2, ncols=2)
dem = np.linspace(2400, 3500, 111)
# plot linear fit
axes[0, 0].set_title("Linear Fit")
axes[0, 0].plot(dem, linIntercept + linSlope * dem, '-g',
linewidth=4)
axes[0, 0].plot(dmpr["Demand"], dmpr["Price"], 'om')
# plot logarithmic fit
axes[0, 1].set_title("Logarithmic Fit")
axes[0, 1].plot(dem, logIntercept + logSlope *
np.log(dem), '-y', linewidth=4)
axes[0, 1].plot(dmpr["Demand"], dmpr["Price"], 'om')
```

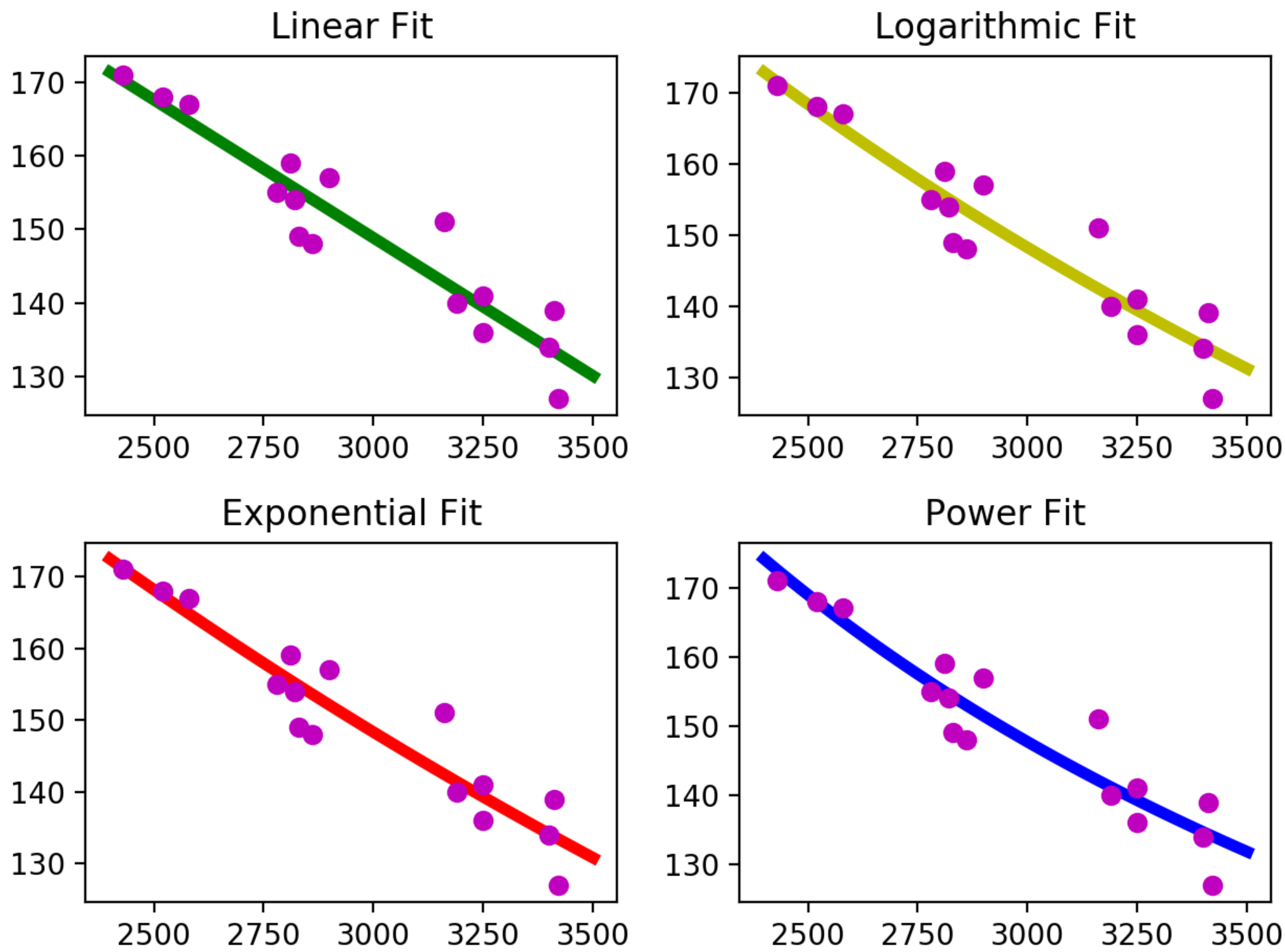
PED Example – Python

```
# plot exponential fit
axes[1, 0].set_title("Exponential Fit")
axes[1, 0].plot(dem, np.exp(expIntercept) *
np.exp(expSlope * dem), '-r', linewidth=4)
axes[1, 0].plot(dmpr["Demand"], dmpr["Price"], 'om')
# plot power fit
axes[1, 1].set_title("Power Fit")
axes[1, 1].plot(dem, np.exp(powIntercept) * dem **
powSlope, '-b', linewidth=4)
axes[1, 1].plot(dmpr["Demand"], dmpr["Price"], 'om')
# display figure
figure.tight_layout()
plt.show()
```

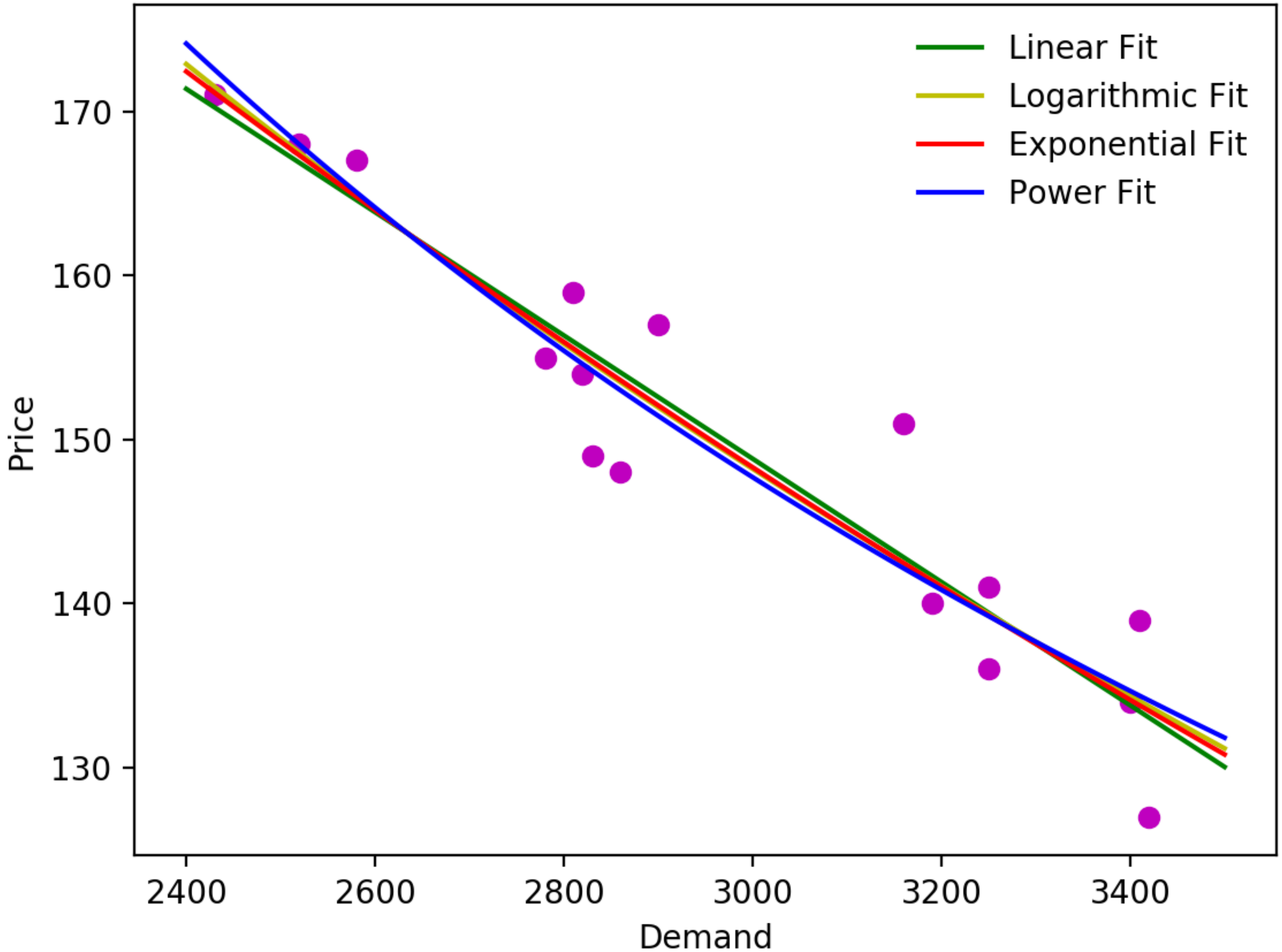
PED Example – Python

```
# plot four models altogether
plt.plot(dmpr["Demand"], dmpr["Price"], 'om')
plt.plot(dem, linIntercept + linSlope * dem, '-g',
label="Linear Fit")
plt.plot(dem, logIntercept + logSlope * np.log(dem), '-
y', label="Logarithmic Fit")
plt.plot(dem, np.exp(expIntercept) * np.exp(expSlope *
dem), '-r', label="Exponential Fit")
plt.plot(dem, np.exp(powIntercept) * dem ** powSlope, '-
b', label="Power Fit")
plt.xlabel("Demand")
plt.ylabel("Price")
plt.legend(frameon=False)
plt.show()
```

PED Example – Python



PED Example – Python



Motivating Example: Beer Preference

- Hacker-Pschorr
  - One of the oldest beer brewing companies in Munich
  - Collects data on beer preference (light/regular) and demographic information
- Goal:

  Determine demographic factors for preferring light beer

	A	B	C	D	E	F
1	Gender	Married	Income	Income in K	Age	Preference
2	0	0	31779	32	46	Regular
3	1	1	32739	33	50	Regular
4	1	1	24302	24	46	Regular
5	1	1	64709	65	70	Regular
6	1	1	41882	42	54	Regular
7	1	0	38990	39	36	Regular
8	1	0	22408	22	40	Regular
9	1	1	25440	25	51	Regular
10	0	1	30784	31	52	Regular
11	1	0	31916	32	43	Regular
12	1	0	23234	23	31	Regular
13	0	1	51094	51	46	Regular
14	1	0	38176	38	40	Regular
15	1	0	28513	29	34	Regular
16	0	1	44955	45	53	Regular
17	0	1	42051	42	58	Regular
18	1	1	40955	41	69	Regular
19	0	1	36451	36	44	Regular
20	1	0	20945	21	42	Regular
21	0	1	33880	34	48	Regular
22	1	1	47667	48	87	Regular
23	1	0	29293	29	43	Regular
24	1	1	30081	30	56	Regular
25	0	1	37263	37	49	Regular
26	1	1	40582	41	66	Regular

Multiple Linear Regression (MLR)

- Let us use MLR to predict preferences:
  - Code preference (the response) as: $$Y = \begin{cases} 1 & \text{if Light} \\ 0 & \text{if Regular} \end{cases}$$
  - Fit the model: $y = b_0 + b_1\, Gender + b_2\, Married + b_3\, Income + b_4\, Age$
- **scikit-learn**: Machine Learning in Python

```
>>> impoort sklearn as sk
```

- The results:

Input Variable	Coefficient	Std.Error	*p*-value
Intercept	0.38768652	0.18655577	0.04039437
Gender	-0.04586497	0.06949534	0.51086867
Married	0.02131404	0.07569433	0.77887774
Income	0.00002877	0.00002877	0
Age	-0.02323128	0.00296831	0

Motivating Example: Beer Preference

- What do you predict is the preference for a male (i.e. Gender=1), who is 25 years old, married (i.e. 1) with annual household income of $40,000?
  - 0.387... -0.045...x1 +0.021...x1 +0.000...x40000 -0.023...x25 = 0.9...

We are trying to predict a discrete outcome with a continuous function

- We need to be able to map values of independent variables into a response that makes sense!

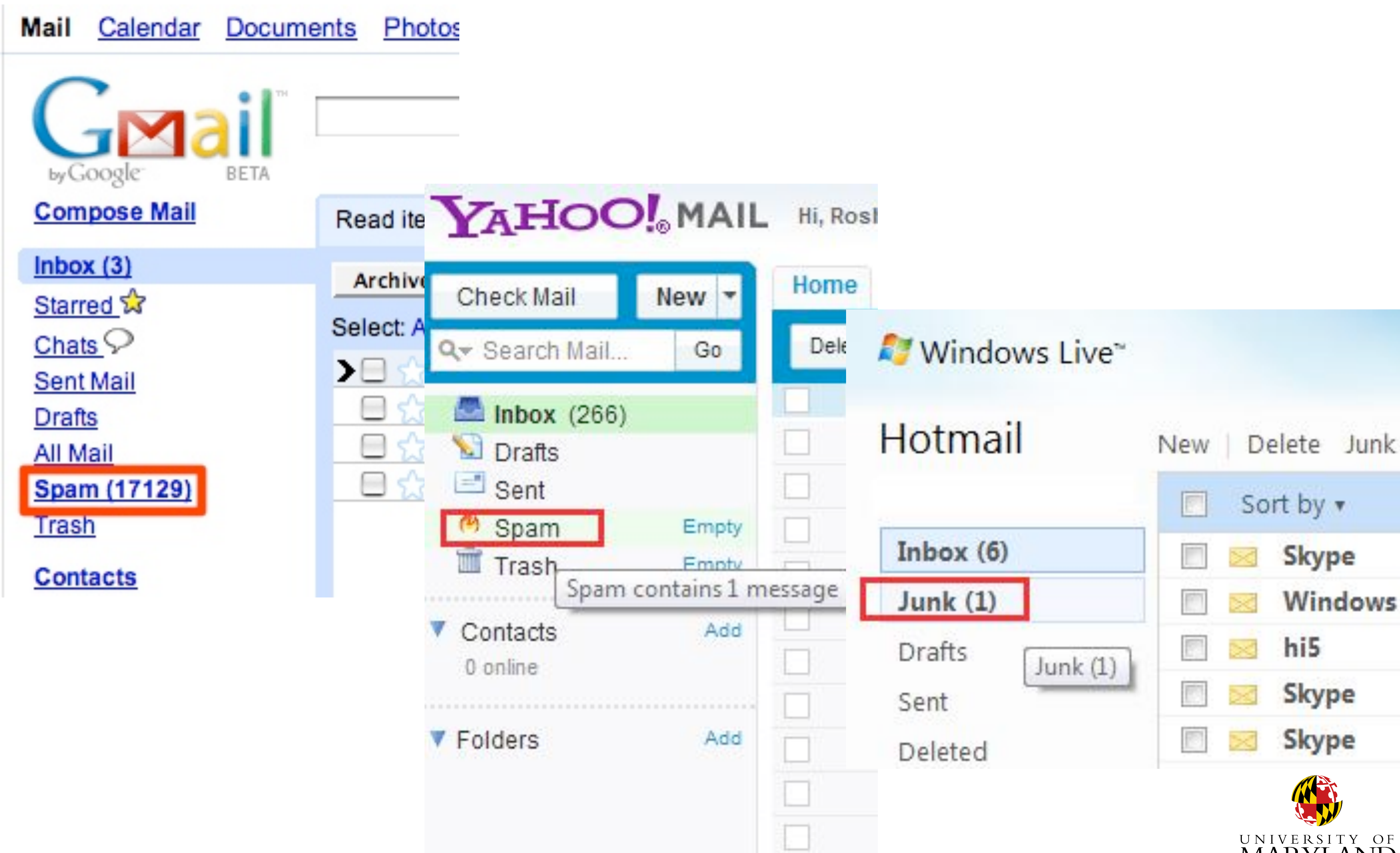
MLR Example – Python

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
# read data from file
beer = pd.read_csv("Beer.csv")
print(beer.describe())
# create dummy variables
dummy = pd.get_dummies(beer["Preference"])
# fit the model
res = LinearRegression().fit(beer.iloc[:, :4] ,
  dummy["Light"])
print(res.intercept_, res.coef_)
# predict new value
print(res.predict(np.array([[1, 1, 40000, 25]])))
```

Classification – Prediction

- The response (dependent) variable is qualitative:
  - "Hit" or "Flop" (e.g. a film or a song)
  - "Legitimate" or "Fraudulent" (e.g. credit card transaction)
  - "Positive" or "Negative" (e.g. medical test results)
  - "Agree", "Neutral", or "Disagree" (ordered categorical)
- The goal is to predict qualitative labels from one or more explanatory (independent) variables:
  - Predict whether a film will be a "hit" or "flop" using information about the cast, director, month of release, etc.
  - Predict whether a credit card transaction is "fraudulent" based on location of transaction, etc.

Example: Spam Detection



UNIVERSITY OF MARYLAND
ROBERT H. SMITH SCHOOL OF BUSINESS

More Examples

- Categorize news stories as finance, weather, entertainment, sports, etc. (think: Google News)
- Object recognition: What object is in the image? (think: Google image search)
- Medical diagnostic: Given list of symptoms, does a person have some disease? (think: Google pandemic trends)

Classifiers – Classification Algorithms

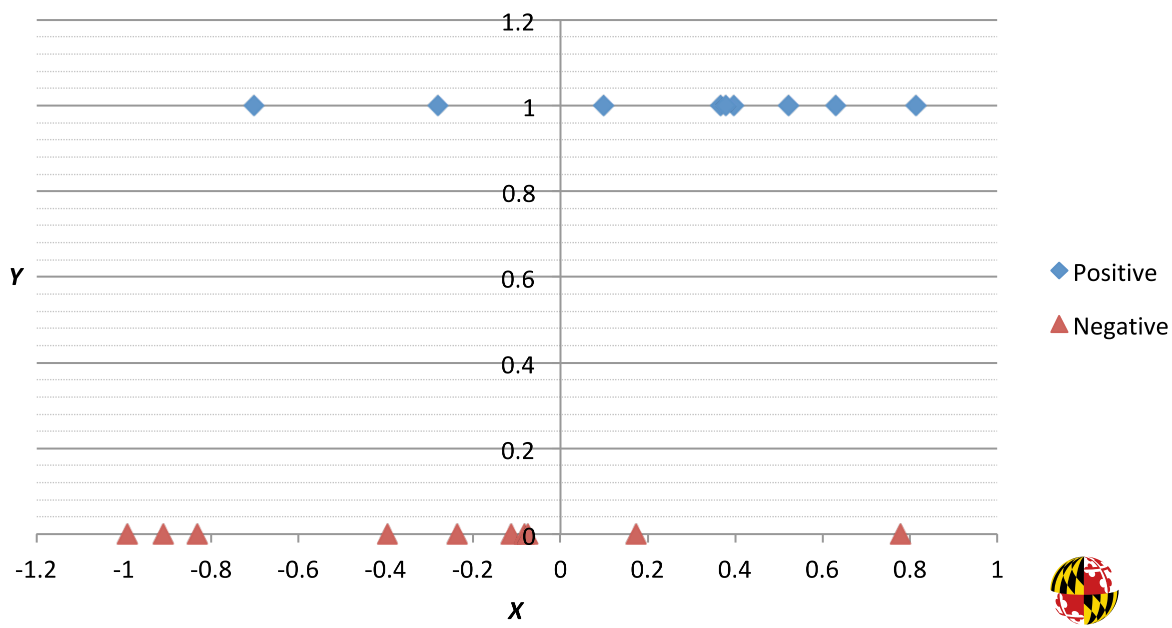
- Logistic Regression
- Naïve Bayes
- Stochastic Gradient Descent
- k-Nearest Neighbors (k-NN), Kernel-Based Methods
- Decision Tree (Classification and Regression Trees)
- Random Forest
- Support Vector Machine (SVM, SVR)
- Neural Nets
- Discriminant Analysis (Linear DA, Quadratic DA)
- … many more!

Qualitative Dependent Variables

- Simplest case is of a binary variable
  - Y is 1 or 0 (customer buys/does not buy; transaction is fraudulent/ not fraudulent; tumor is malignant/benign; …)
- Suppose we have a number of observations on such a variable Y
  - $y_i = 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, \ldots$
  - Their average is the proportion of 1's in the sample
  - In other words, it is the probability of "picking 1"
  - For indicator variables, $E[Y] = P[Y = 1]$, the probability of Y

Example

- We typically have available explanatory variables that are useful for predicting Y
  - In such cases $E(Y|X) = P[Y = 1|X]$
  - Here is a scatterplot (◆ corresponds to $Y = 1$; ▲ to $Y = 0$)
  - Does is appear as if Y depends on X?
  - Can we classify a case as positive or negative based on X?

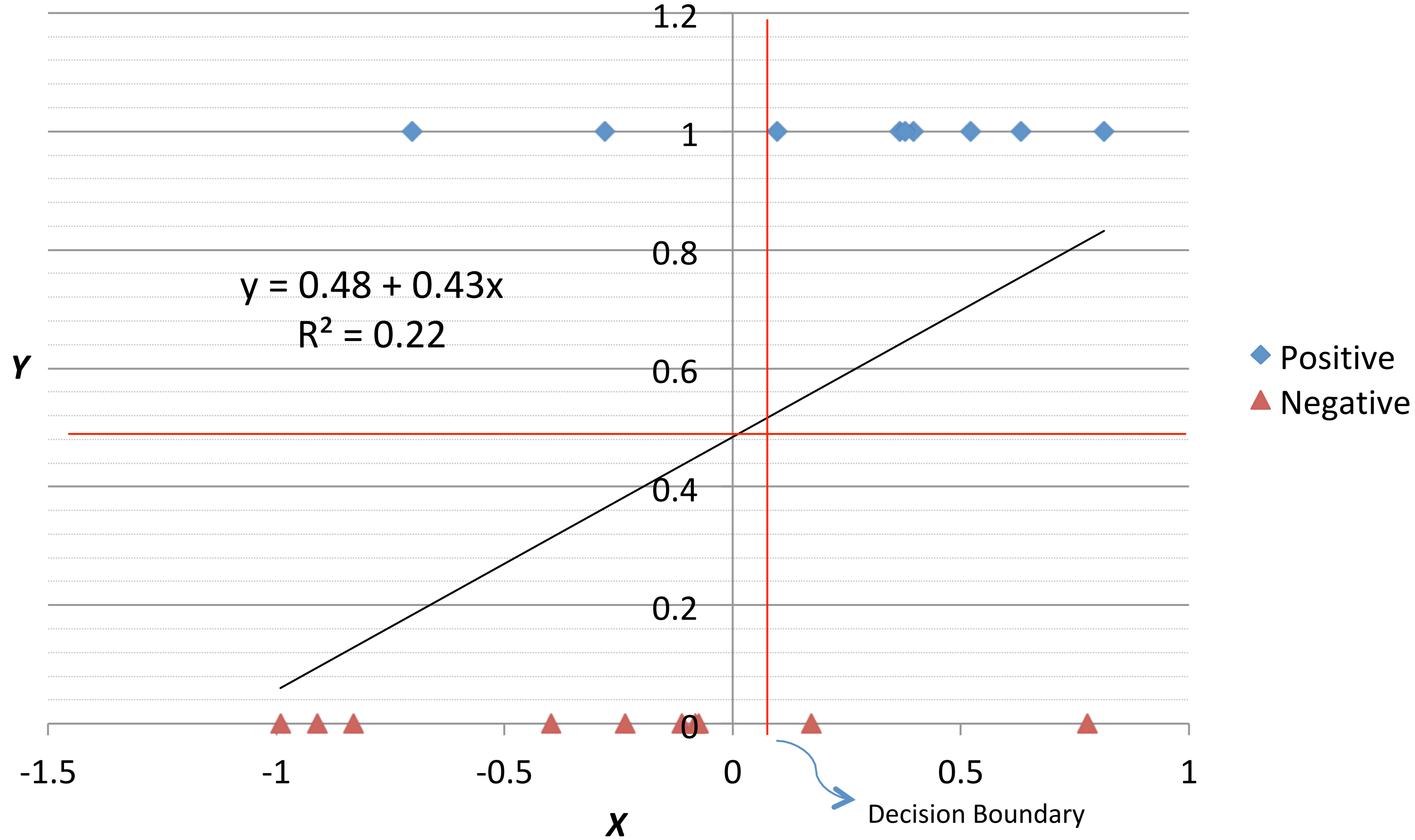


UNIVERSITY OF MARYLAND
ROBERT H. SMITH SCHOOL OF BUSINESS

Linear Probability Model

X	Y
0.521552	1
-0.70176	1
0.397391	1
-0.39701	0
0.36739	1
0.098193	1
0.813719	1
-0.27989	1
0.631199	1
0.378386	1
-0.11216	0
-0.08226	0
-0.83098	0
0.172896	0
-0.23603	0
-0.99109	0
-0.0739	0
-0.91048	0
0.777112	0
-0.32008	0

$$E[Y|X] = P[Y=1|X] = \alpha + \beta X$$



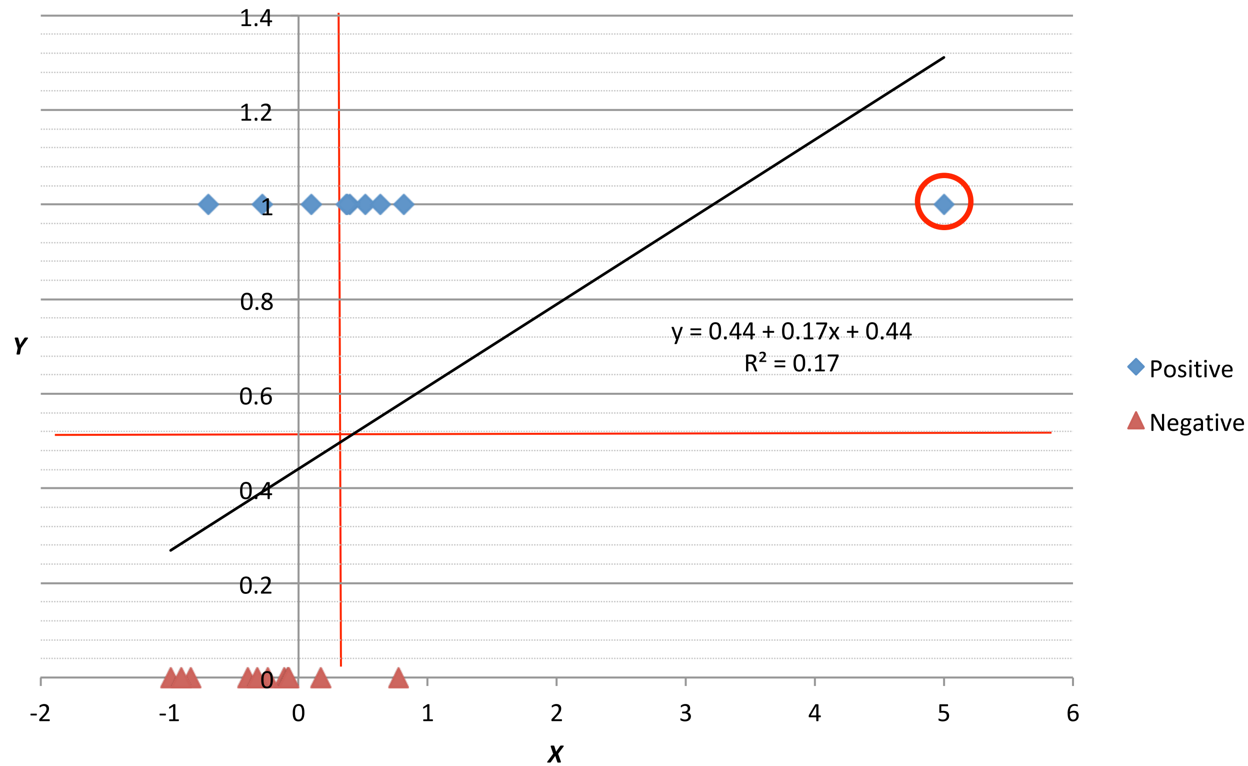
What is the interpretation of the slope 0.43 in the regression?

Predicting Cases Based on *X*

- Predicting outcomes isn't possible (directly)
  - The predicted value of the regression equation is the probability that $Y = 1$ (and not a value of Y)
- A "reasonable" approach would be:
  - If for any X the predicted value is greater than 0.5 then predict that $Y = 1$
  - 0.5 is said to be the cutoff
  - Values other than 0.5 could be used as well

Problems with Approach

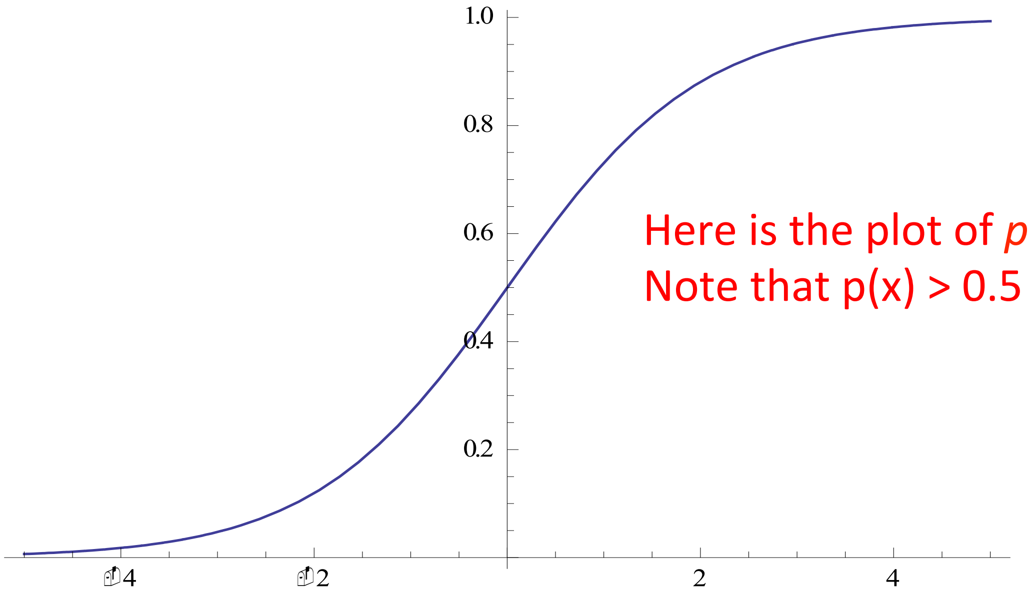
- We add one new point (very high *X* and positive case)
- Predicted probability is greater than one → Need a new strategy!



UNIVERSITY OF MARYLAND
ROBERT H. SMITH SCHOOL OF BUSINESS

Logistic Function

- We will use the logistic function instead of a linear probability function to estimate the probability of $Y = 1$
  - $P(Y=1) = 1 / (1 + e^{-(\alpha + \beta x)})$
- For now let us ignore how α and β are estimated, and focus on what they mean …

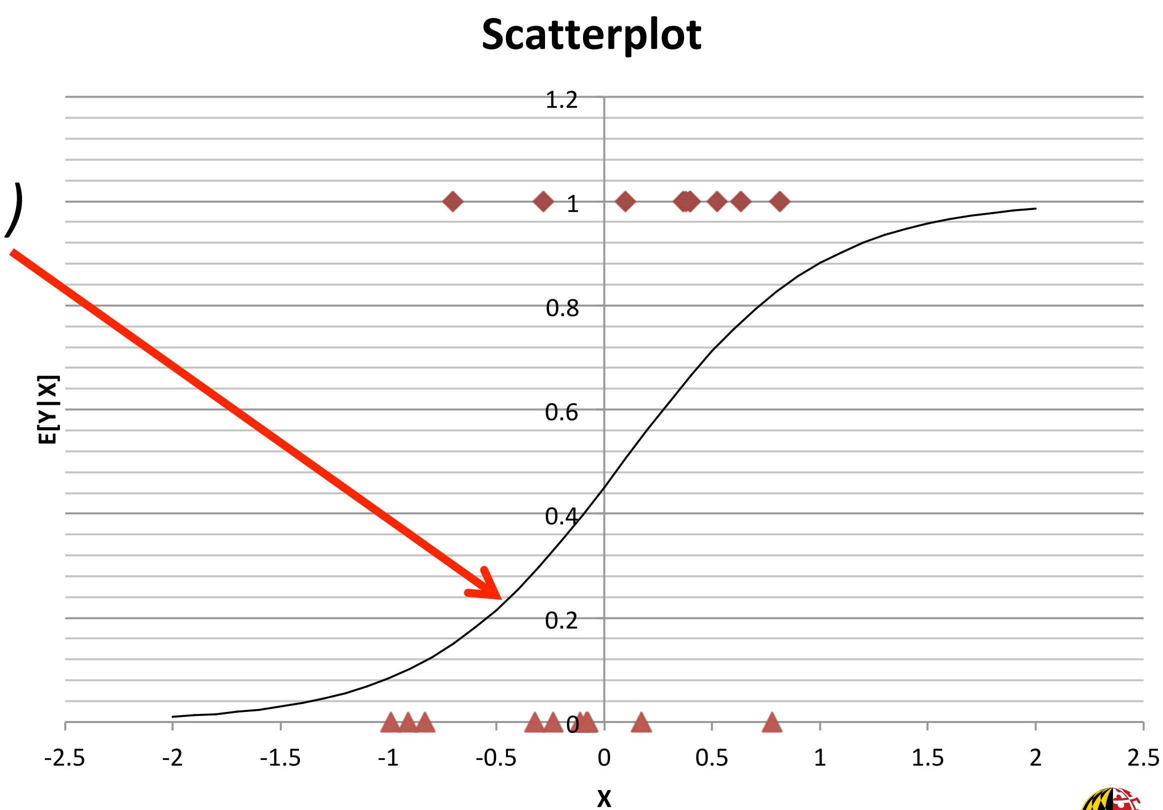


Here is the plot of $p(x) = 1 / (1 + e^{-x})$
Note that p(x) > 0.5 when x > 0

Predicted Values of *P(Y=1|X)*

- Superimposed on data

$$1 / (1 + e^{-(-0.195 + 2.20x)})$$

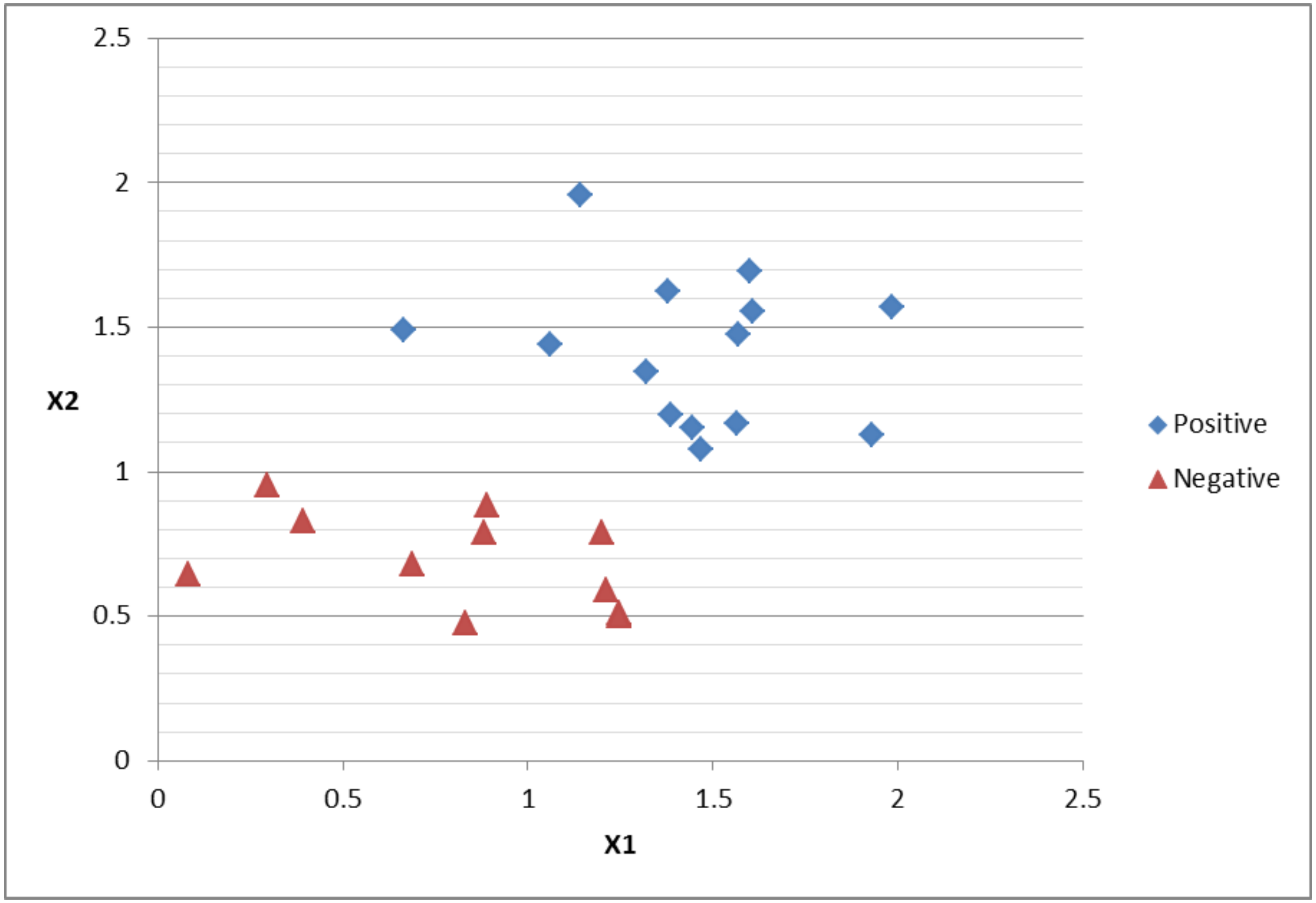


Decision Boundary

- This can be calculated from:
  - $\alpha + \beta X > 0$ (i.e. the value for which $P(Y=1|X) > 0.5$
  - $- 0.195 + 2.20 X > 0$ or $X > 0.0887$
  - This means that (in future instances) we will classify:
    cases with $X > 0.0887$ as positive and
    cases with $X <= 0.0887$ as negative
- Things get more interesting with more than one explanatory variable

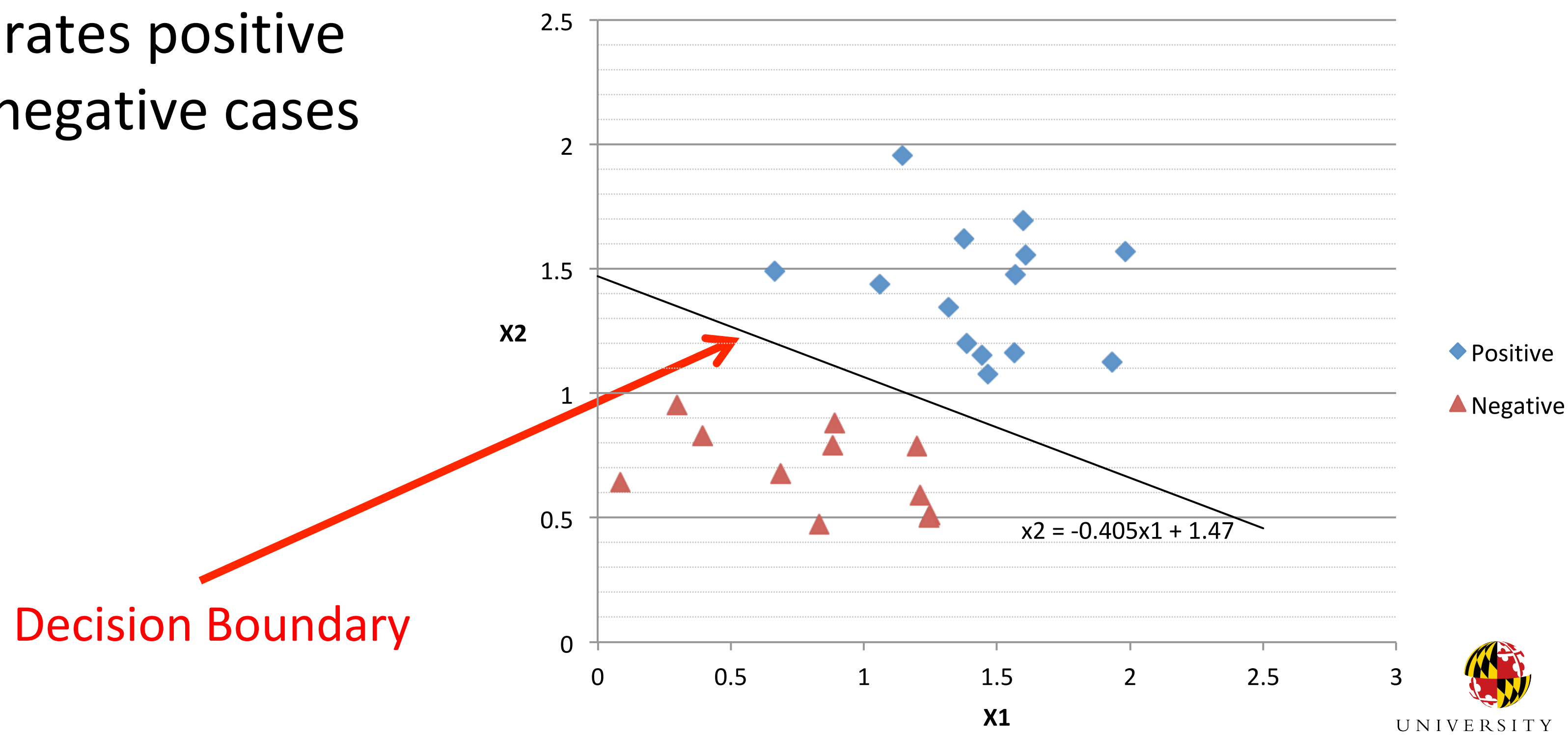
Example: Two Explanatory Variables

X1	X2	Y
1.377854	1.622107	1
1.599992	1.694089	1
1.565331	1.164109	1
1.443899	1.151105	1
1.609263	1.554958	1
1.983125	1.568665	1
1.143215	1.955096	1
1.568877	1.476863	1
1.93188	1.125688	1
1.319446	1.345884	1
1.385208	1.198732	1
0.664142	1.490022	1
1.46775	1.075994	1
1.05938	1.438602	1
0.883673	0.790297	0
1.211836	0.591293	0
1.198894	0.787916	0
0.686669	0.677238	0
1.246374	0.510375	0
1.245663	0.500476	0
0.083999	0.642928	0
0.831823	0.474497	0
0.39346	0.827828	0
0.296986	0.953236	0
0.890291	0.881744	0



Decision Boundary

- Decision Boundary
  - $-31.51 + 8.69\ X_1 + 21.44\ X_2 = 0$ (Why?)
  - Rearranging terms: $X_2 = -\ 0.405\ X_1 + 1.47$
  - Separates positive and negative cases



UNIVERSITY OF MARYLAND
ROBERT H. SMITH SCHOOL OF BUSINESS

Logistic Regression on Big Datasets

- Logistic Regression is actually very flexible!
- But its almost always better to stick with linear terms with big datasets

Interpreting Coefficients

- $P(Y=1|X_1,X_2) = 1 / (1 + e^{-(\alpha + \beta_1 X_1 + \beta_2 X_2)})$
  - The parameters α, β_1, and β_2 are estimated
  - Do they have meaningful interpretations?
- Denote $P(Y=1|X_1,X_2)$ by p
  - A little algebra allows us to rewrite this as
  - $p = 1 / (1 + e^{-(\alpha + \beta_1 X_1 + \beta_2 X_2)})$
  - $1/p - 1 = (1 - p) / p = e^{-(\alpha + \beta_1 X_1 + \beta_2 X_2)}$
  - $p / (1 - p) = e^{\alpha + \beta_1 X_1 + \beta_2 X_2}$
  - $\ln(p / (1 - p)) = \alpha + \beta_1 X_1 + \beta_2 X_2$
  - The LHS is called the ***logit***, and we study how to interpret this

Logistic Regression

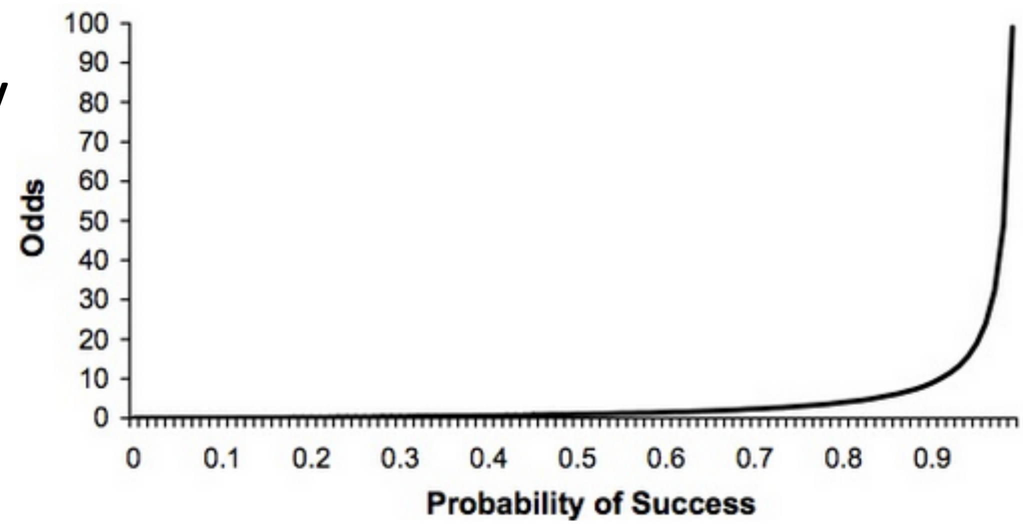
- **Goal:** Find a function of the predictor variables that relates them to a 0/1 outcome
- The main idea:
  - Instead of using *Y* (or *p*) as the dependent variable, we use a function of it, which is called the ***logit***.
  - The key advantage: ***logit*** maps any value of the dependent variables into a probability *[0,1]*.
- The ***logit***, it turns out, can be modeled as a linear function of the predictors
  - Once the ***logit*** has been predicted, it can be mapped back to a probability *p*.

Logistic Regression

- To find the ***logit*** value, we need two steps
  - First, the probability is converted to odds
    Step 1: Probabilities *[0,1]* → Odds *[0, ∞]*
  - Then, the log of the odds is taken
    Step 2: Odds *[0, ∞]* → Logit *[-∞, ∞]*
- These two steps transform:
  - The interval of probabilities in the range *[0,1]* to a numerical interval in the range *[-∞, ∞]*

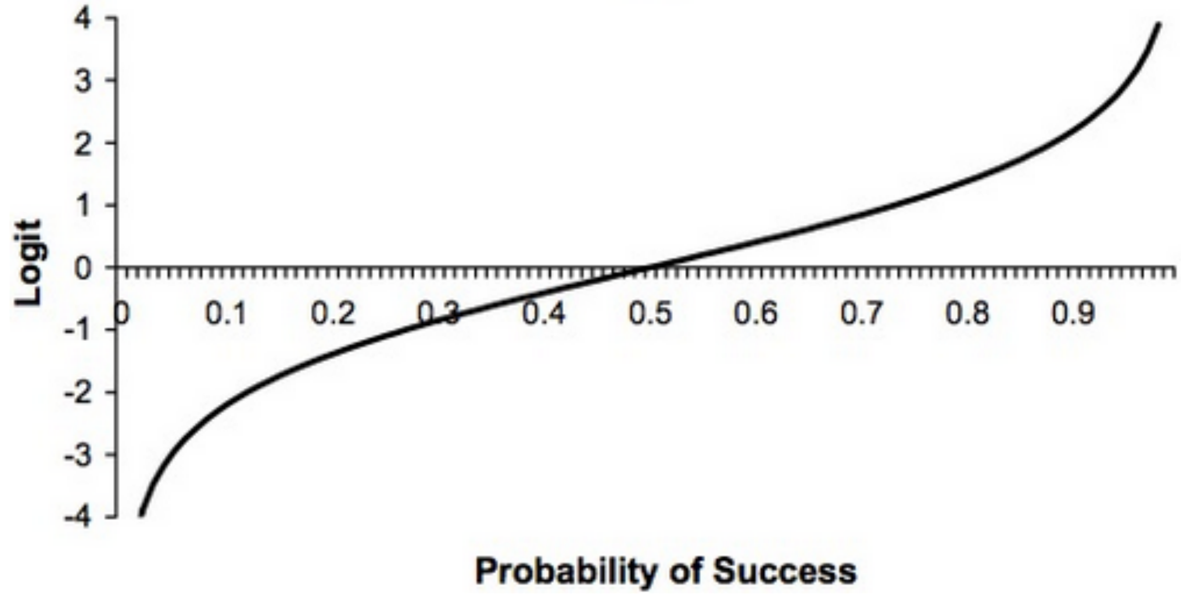
Step 1: Probabilities → Odds

- If an event has probability *p* then the odds of the event is:
  - *odds = p/(1-p)*
- Example:
  - If *p = .01*, the odds = *.01/.99 = .0101...*
  - If *p = .1*, the odds = *.1/.9 = .11...*
  - If *p = .5*, the odds = *.5/(1-.5) = 1*
  - If *p = .9*, the odds = *.9/.1 = 9*
- The range of odds:
  - $0 \leq \text{odds} \leq \infty$



Note in gambling, the "odds against" are often given, which is *1/odds(p)* or *odds(1-p)*.

Step 2: Odds → Logit



- Next, the natural log of the odds is taken:
  - *logit = log(odds) = ln(p/(1-p))*
- Example cont.:
  - If p = .01, the odds = .01/.99 = .0101..., logit = -4.6
  - If p = .1, the odds = .1/.9 = .11..., logit = -2.2
  - If p = .5, the odds = .5/(1-.5) = 1, logit = 0
  - If p = .9, the odds = .9/.1 = 9, logit = 2.2
- The range:
  - $-\infty \leq \log(\text{odds}) \leq \infty$

The Logit is:
negative if p < .5,
0 if p = .5,
positive if p > .5

The Logistic Regression Model

- A **nonlinear** regression model

  $$logit = \beta_0 + \beta_1 \, Gender + \beta_2 \, Married + \beta_3 \, Income + \beta_4 \, Age$$

- Inserting logit = log(odds)

  $$odds = e^{(\beta_0 + \beta_1 \text{Gender} + \beta_2 \text{Married} + \beta_3 \text{Income} + \beta_4 \text{Age})}$$

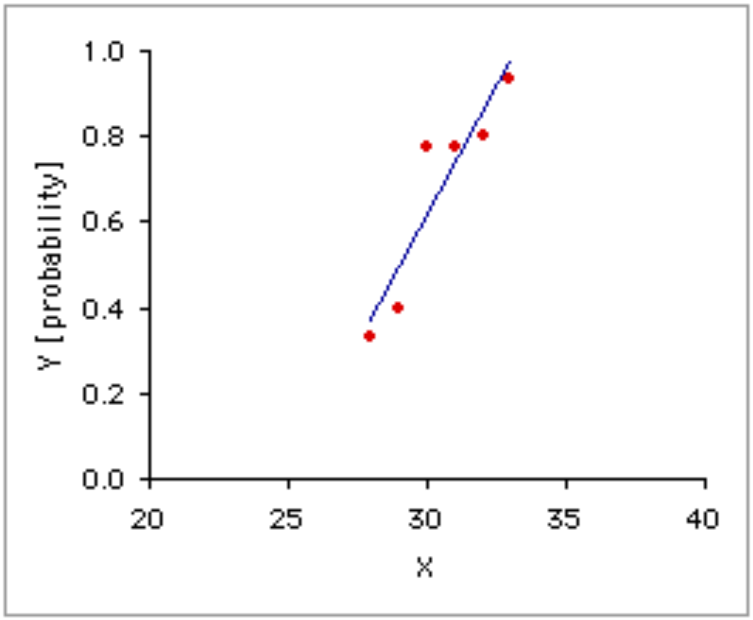
- Solving for p

  $$p = \frac{odds}{1 + odds} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \text{Gender} + \beta_2 \text{Married} + \beta_3 \text{Income} + \beta_4 \text{Age})}}$$

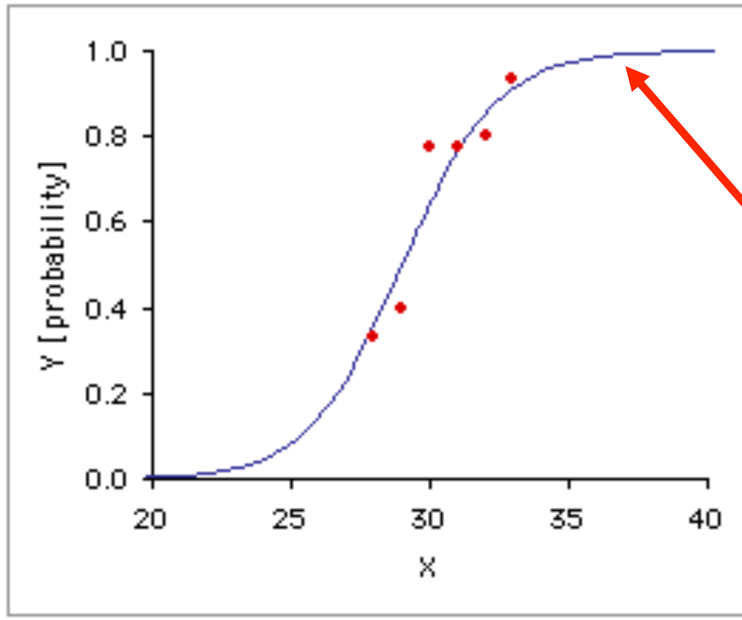
- Bottom-line: logistic regression is a nonlinear function that maps any values of the input variables into a probability

Plotting the Logistic Relationship

- Schematic for a single predictor:



Linear



S-shaped / sigmoidal function

Logistic

The Use of Logistic Regression

- Logistic Regression is used for predicting the probability of occurrence of an event
  - Can use numerous predictor variables that can be either numerical or categorical
- For example:
  - The probability that a person accepts a personal loan may be predicted from knowledge of the person's age, sex and annual income
- Used extensively in the medical and social sciences as well as marketing applications such as prediction of a customer's propensity to purchase a product or cease a subscription

Logistic Regression for Classification

- Logistic regression outputs the probability of a categorical outcome
- However it is most often used for classification
  - Example: Logistic regression outputs the probability of a customer accepting the loan
  - A classification labels a customer as an accepter/nonaccepter
- To end up with classification we need a cut-off value *c*
  - Observations with probabilities above *c* are classified as belonging to class *1* (accepter)
  - Observations with probabilities below *c* are classified as belonging to class *0* (nonaccepter)

What is the Right Value of *c*?

- A popular value is *0.5*
- Another option is to select the cut-off that maximizes accuracy (the number of correctly classified points)
  - Possible over-fitting
- Other options include minimizing false positives subject to restrictions on false negatives

Fitting the Model

- The relationship between p (or Y) and the β_i is nonlinear
- The parameters are estimated using the method of maximum likelihood estimation
  - This methods finds the parameters β_i, such that the chance of observing the data is maximized
  - Done iteratively → with big datasets, use a fast computer!

Note that logistic regression calculates changes in the log odds of the dependent variable, not changes in the dependent variable itself therefore we cannot use least squares estimation.

LR Example – Python

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
# read data from file
beer = pd.read_csv("Beer.csv")
print(beer.describe())
# create dummy variables
dummy = pd.get_dummies(beer["Preference"])
# fit the model
res = LogisticRegression().fit(beer.iloc[:, :4] ,
  dummy["Light"])
print(res.intercept_, res.coef_)
# predict new value and probabilities
print(res.predict(np.array([[1, 1, 40000, 25]])))
print(res.predict_proba(np.array([[1, 1, 40000, 25]])))
```

Interpreting Coefficients of Continuous Predictors: Beer Preference Example

Input Variable	Coefficient	Std.Error	*p*-value	Odds
Intercept	-0.00238108	1.93081641	0.72396708	*
Gender	-0.00279958	0.71664554	0.27772108	0.45937654
Married	-0.00040049	0.79447782	0.83089775	1.18490314
Income	0.00025542	0.00006335	0.00001103	1.00027847
Age	-0.22997591	0.05238947	0.00001323	0.79594839

- Estimated coefficient of Age: b_{Age} =________, or, $exp(b_{Age})$=_______.
- Implies that a 1 year increase in age _____creases the odds of preferring light beer by a factor of _______, for those with same gender, marital status and income
- If age increases by *10* years (but same gender, marital status & income), the odds of preferring light beer decreases by a factor of ________________________________

Interpreting Coefficients of Categorical Predictors: Beer Preference Example

Input Variable	Coefficient	Std.Error	*p*-value	Odds
Intercept	-0.00238108	1.93081641	0.72396708	*
Gender	-0.00279958	0.71664554	0.27772108	0.45937654
Married	-0.00040049	0.79447782	0.83089775	1.18490314
Income	0.00025542	0.00006335	0.00001103	1.00027847
Age	-0.22997591	0.05238947	0.00001323	0.79594839

- Estimated coefficient for Gender:
  $b_{Gender} = -0.003$, or,
  $odds_{Gender} = exp(b_{Gender}) = 0.997$
- Implies that the odds of a **male** customer preferring light beer are 0.997 times the odds of a **female** customer *of the same marital status, age and income* preferring light beer.

UNIVERSITY OF MARYLAND
ROBERT H. SMITH SCHOOL OF BUSINESS

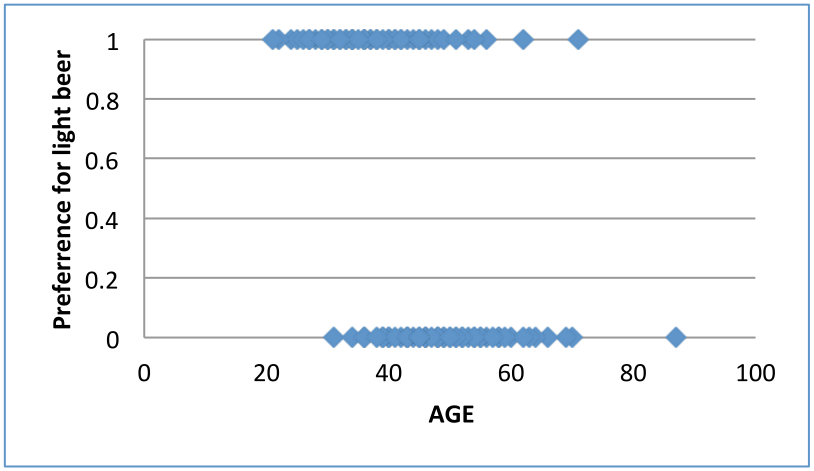
Interpretation: Bottom Line

- A single unit increase in x_i, holding all other independent variables constant, is:
  - associated with an change in the odds of the outcome being equal to one by a multiplicative factor of e^{β_ι}.
- In other words:
  - β_ι is the multiplicative factor by which the odds (of belonging to class 1) increase when the value of x_i is increased by 1 unit, holding all other predictors constant.

NOTE: The change in the probability of the outcome being one, *p*, for a unit increase in a particular explanatory variable, while holding all other variables constant, is ***not*** a constant – it depends on the specific values of that explanatory variable.

Practical Implication of Nonlinear Regression on Interpretation

- Probability that a 20-year-old married woman, earning \$40,000/year prefers light beer: $\hat{p}_{Light} = \dfrac{1}{1+e^{-6.06}} = 0.99767$
  - What if the same customer was **25 years old**? $\hat{p}_{Light} = \dfrac{1}{1+e^{-4.92}} = 0.9928$
  - What if the same customer was **40 years old**? $\hat{p}_{Light} = \dfrac{1}{1+e^{-1.50}} = 0.817$
  - What if the same customer was **45 years old**? $\hat{p}_{Light} = \dfrac{1}{1+e^{-0.356}} = 0.588$



Increased Annual Income is Associated with…

- … higher probability of preferring light beer
- … lower probability of preferring light beer
- … we do not have enough information to conclude about the effects of annual income on preferring light beer