Lingnan 嶺南大學
University 香港 Hong Kong

School of Data Science
數據科學學院

CDS 533
Statistics for Data Science

Instructor: Lisha Yu

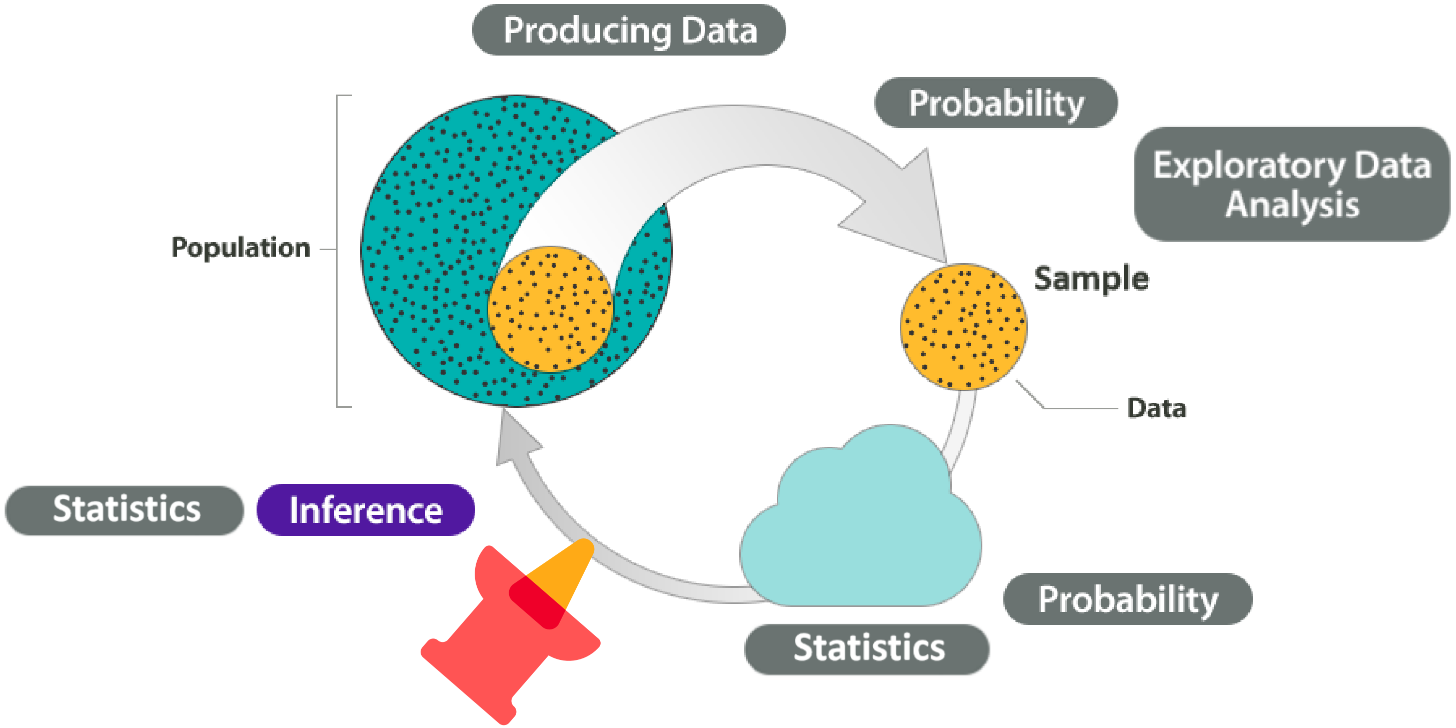
Division of Artificial Intelligence

School of Data Science

Lingnan University

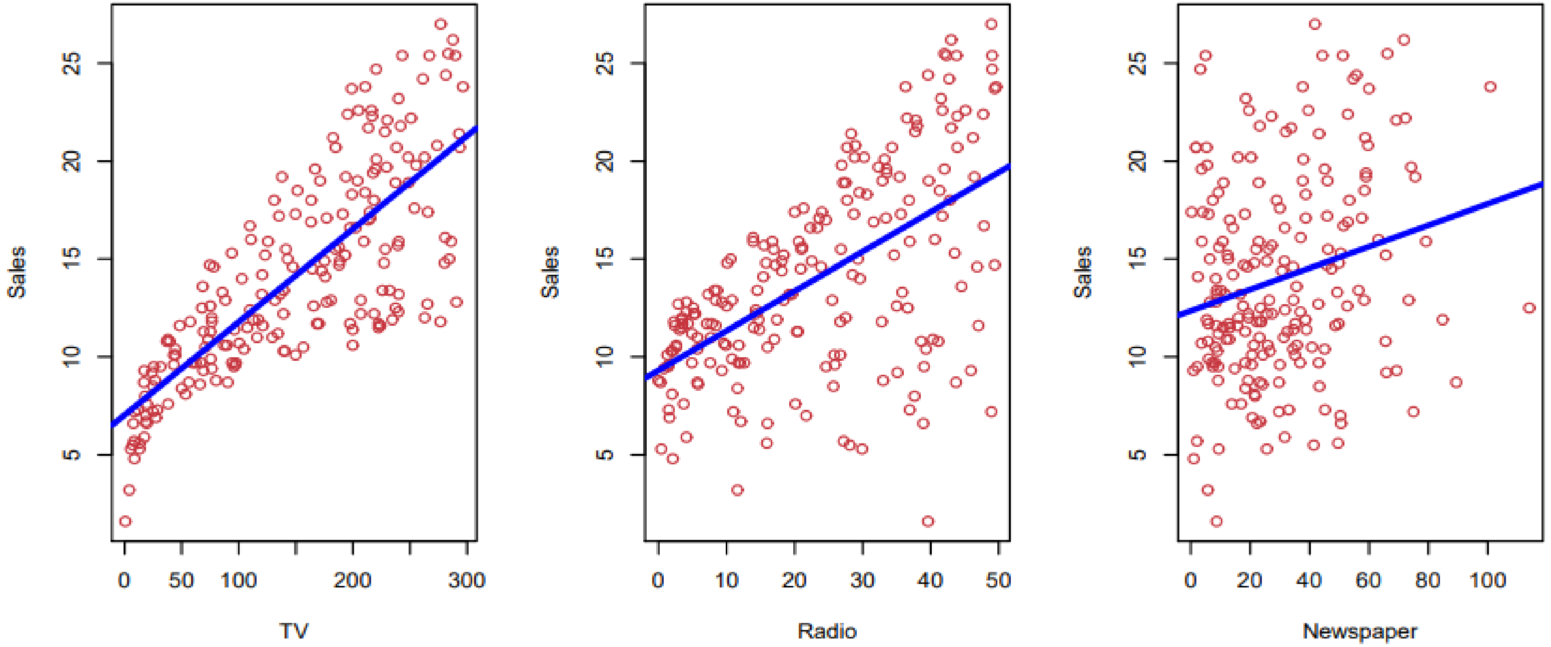
Fall 2024

Big Picture of Statistics



Statistical Inference (Regression)

Statistical Learning



Shown are Sales vs TV, Radio and Newspaper, with a blue linear-regression line fit separately. Can we predict Sales using these three?
Perhaps we can do better using a model

$$\texttt{Sales} \approx f(\texttt{TV}, \texttt{Radio}, \texttt{Newspaper})$$

Statistical Learning: Notation

- Here Sales is a *response* or *target* that we wish to predict. We generically refer to the response as Y . [**Dependent variable/Response**]
- TV is a *feature, or input*, or *predictor*; we name it X_1. [**Independent variable/Predictors**]
- Likewise name Radio as X_2, and so on.
- We can refer to the *input vector* collectively as

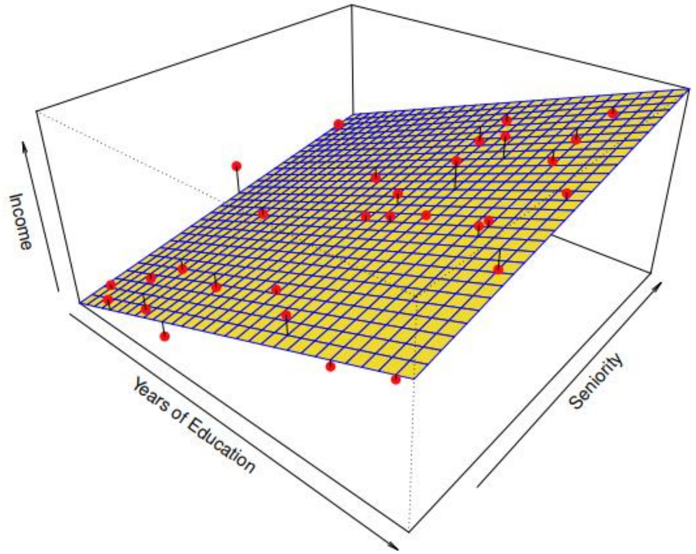
$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

Now we write our model as

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

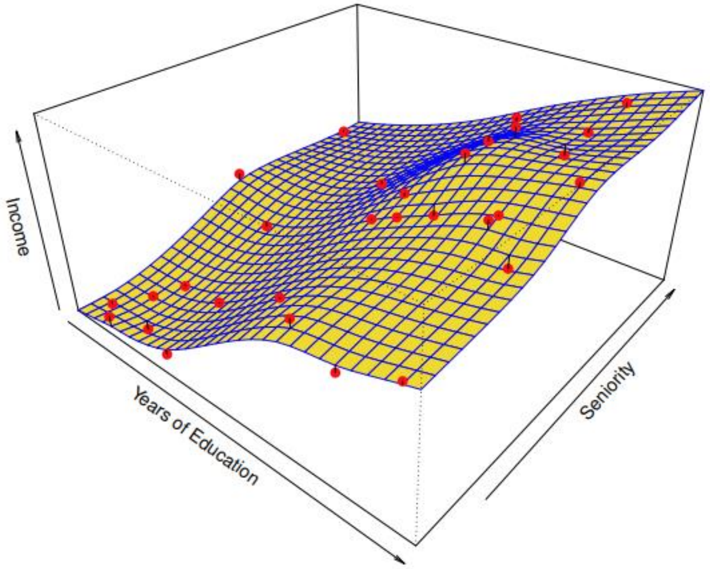
where ε captures measurement errors and other discrepancies.

Parametric and Structured Models

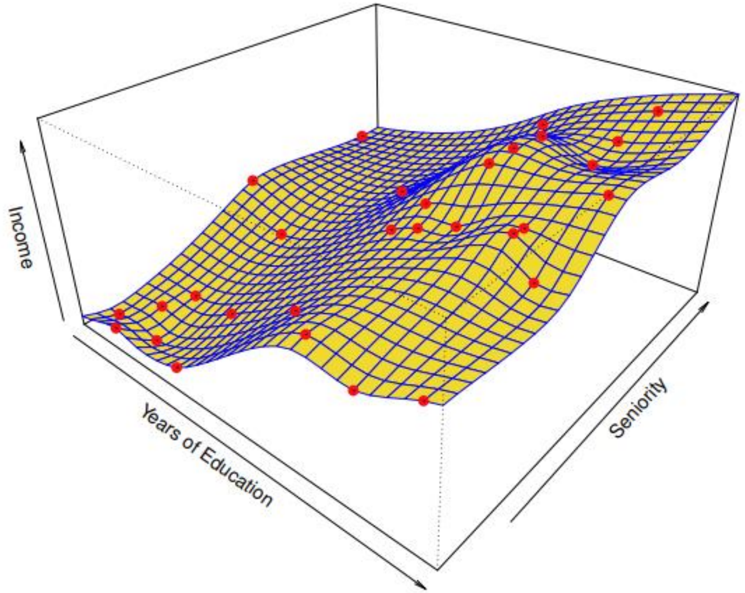


linear model fit by least squares

$\hat{f}_L(\texttt{education}, \texttt{seniority}) = \hat{\beta}_0 + \hat{\beta}_1 \times \texttt{education} + \hat{\beta}_2 \times \texttt{seniority}$



thin-plate spline regression model



more flexible spline regression model

overfitting

Simple linear regression (a single predictor)

- Given some estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ for the model coefficients, we predict future sales using

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x,$$

where $\hat{y}$ indicates a prediction of Y on the basis of $X = x$.

The hat symbol denotes an estimated values, some references use b_0, b_1.

Multiple Linear Regression

- Here our model is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon,$$

- We interpret β_j as the **average** effect on Y of a one unit increase in X_j , **holding all other predictors fixed**. In the advertising example, the model becomes

$$\texttt{sales} = \beta_0 + \beta_1 \times \texttt{TV} + \beta_2 \times \texttt{radio} + \beta_3 \times \texttt{newspaper} + \epsilon.$$

95% CI of the β_j

$$[\,\hat{\beta}_j - t_{1-\alpha/2,n-p}(\hat{\beta}_j)\,, \hat{\beta}_j + t_{1-\alpha/2,n-p}(\hat{\beta}_j)]$$

Qualitative Predictors

Example:

investigate differences in credit card balance between males and females, ignoring the other variables. We create a new variable

Dummy Variable $$x_i = \begin{cases} 1 & \text{if } i\text{th person is female} \\ 0 & \text{if } i\text{th person is male} \end{cases}$$

Resulting model:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is female} \\ \beta_0 + \epsilon_i & \text{if } i\text{th person is male.} \end{cases}$$

Interpretation?

Extensions of the Linear Model

Removing the additive assumption: **interactions**

Interactions:

- For example, the linear model

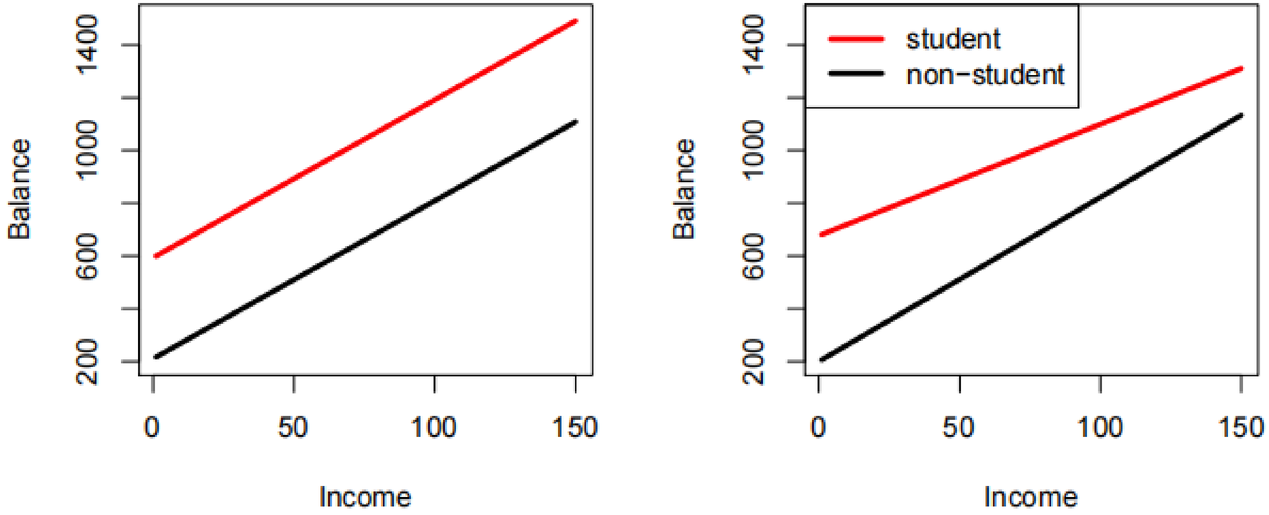
$$\begin{aligned}\texttt{sales} &= \beta_0 + \beta_1 \times \texttt{TV} + \beta_2 \times \texttt{radio} + \beta_3 \times (\texttt{radio} \times \texttt{TV}) + \epsilon \\ &= \beta_0 + (\beta_1 + \beta_3 \times \texttt{radio}) \times \texttt{TV} + \beta_2 \times \texttt{radio} + \epsilon.\end{aligned}$$

Linear Model: Interaction

Interactions between qualitative and quantitative variables

With an interaction term, the model takes the form

$$\begin{aligned}\texttt{balance}_i &\approx \beta_0 + \beta_1 \times \texttt{income}_i + \begin{cases}\beta_2 + \beta_3 \times \texttt{income}_i & \text{if student}\\ 0 & \text{if not student}\end{cases}\\ &= \begin{cases}(\beta_0 + \beta_2) + (\beta_1 + \beta_3) \times \texttt{income}_i & \text{if student}\\ \beta_0 + \beta_1 \times \texttt{income}_i & \text{if not student}\end{cases}\end{aligned}$$



Left: no interaction between income and student.

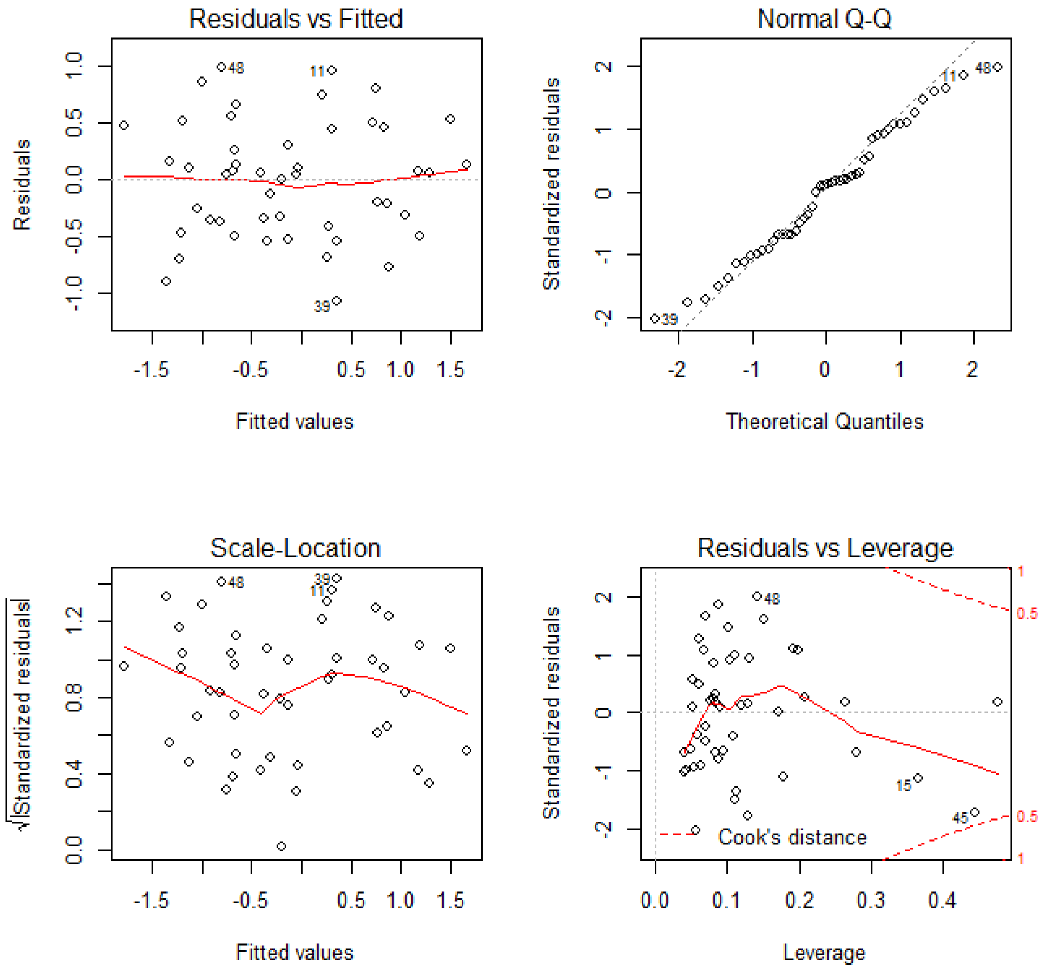
Assumptions of Regression

- **Linear Relationship**
- **Equal Variance (Homoscedasticity)**
  - The probability distribution of the errors has constant variance
- **Normality of Error**
  - Error values (ε) are normally distributed for any given value of X
- **Independence of Errors**
  - Error values are statistically independent
- **No or little Multicollinearity**

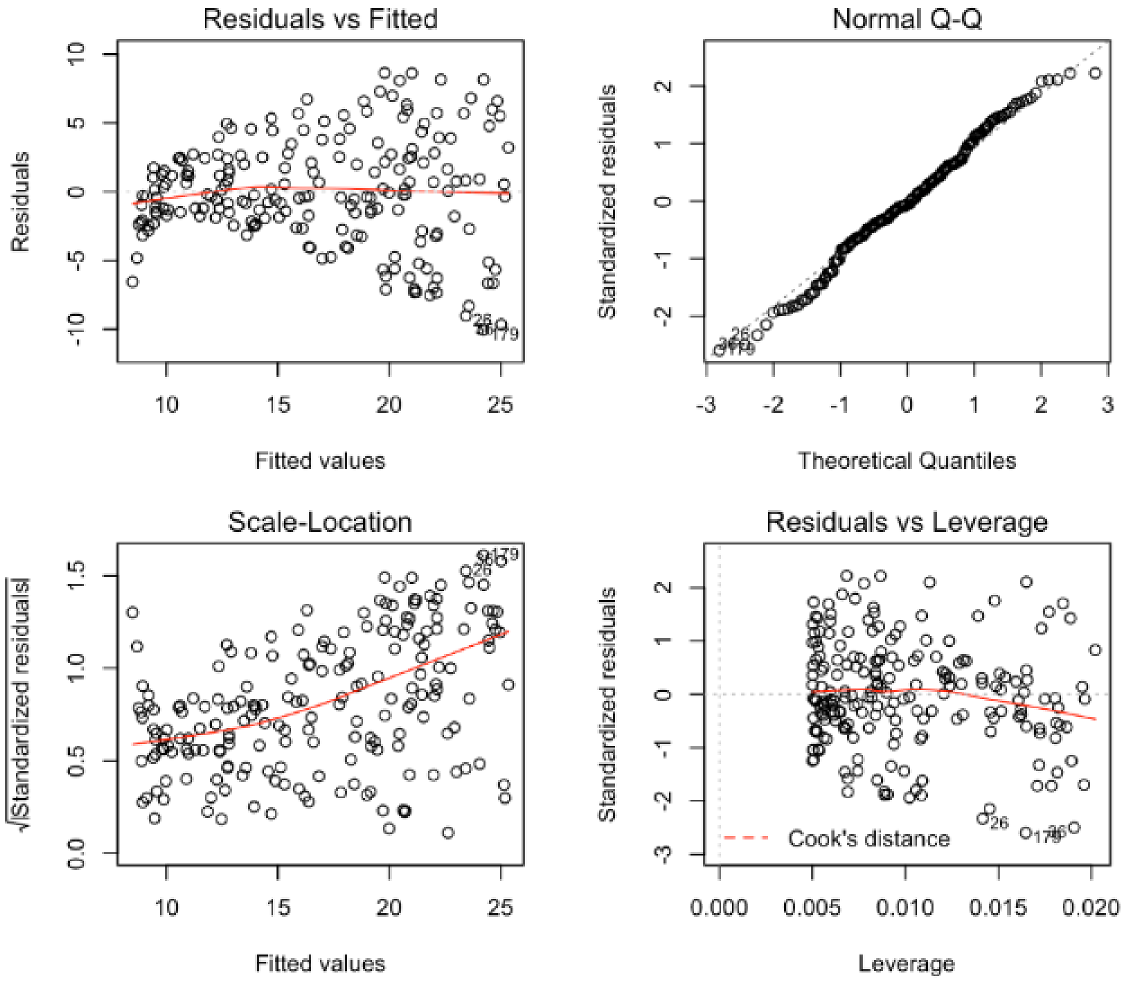
Use Graphical Analysis of Residuals!

Assumptions of Regression: R Examples

Example 1



Example 2

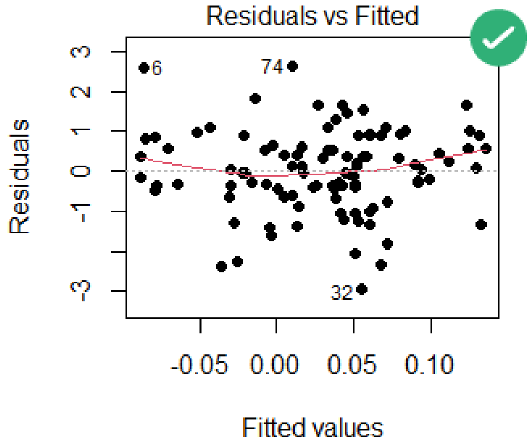


In this class....

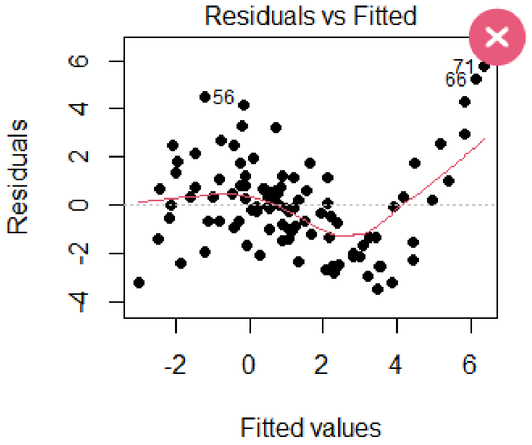
Fundamentals of Transformation

Motivation

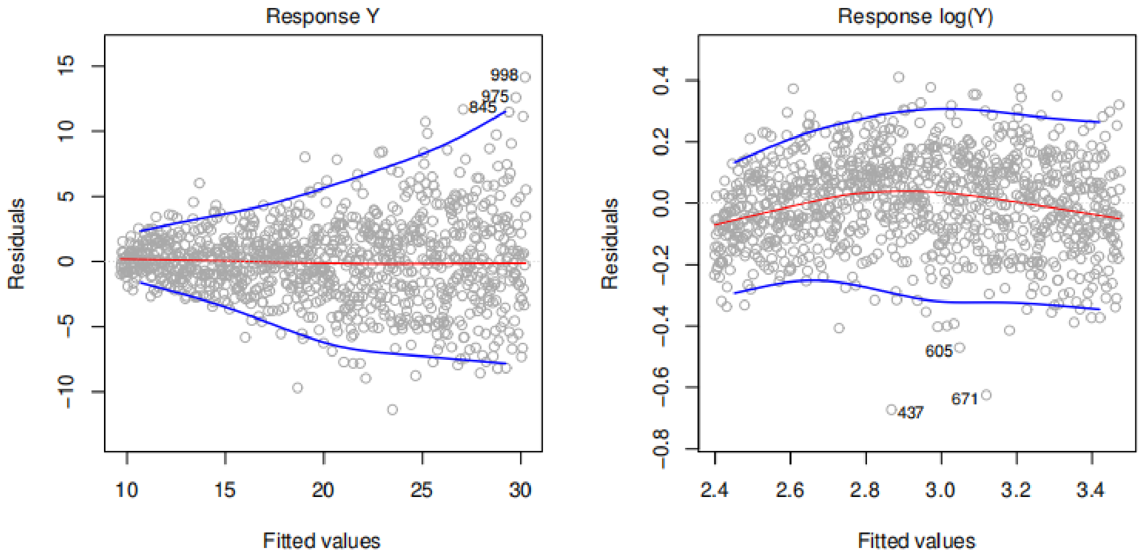
A: Linearity assumption satisfied:



B: Linearity assumption violated:



Heteroskedasticity -> Homoscedastic



- Transforming the predictor X (log, square root):
- $Y = \log(X)$

- Transforming the outcome Y (log, square root)
- $\log(Y) = X$

Fundamentals of Transformation

Why Transform Variables?

To conform to **regression assumptions** which amplifies predictive power and increase the overall quality of the model.

Four Primary Reasons

There are four primary reasons why we might want to transform continuous variables.:

1. To **even out the variance** of a variable if the assumption of homoscedasticity is violated
2. To **normalize** a variable if the assumption of normality is violated (by visual inspection, QQplot, normality test such as ShapiroWilks)
3. To **linearize** the regression model if it seems individual variables are non-linear.
4. Reduce the impact of **outliers** and **high-leverage observations.**

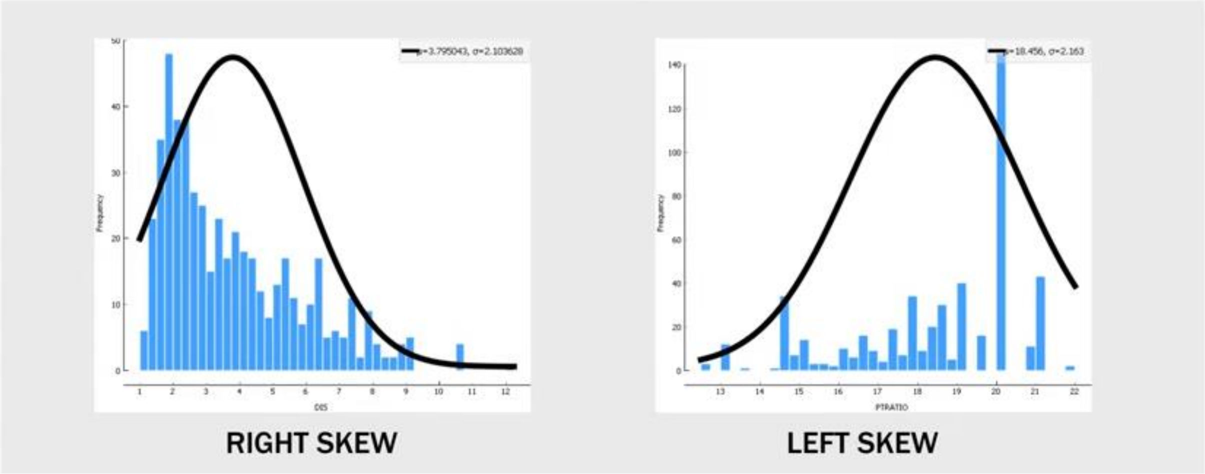
Fundamentals of Transformation

Methods

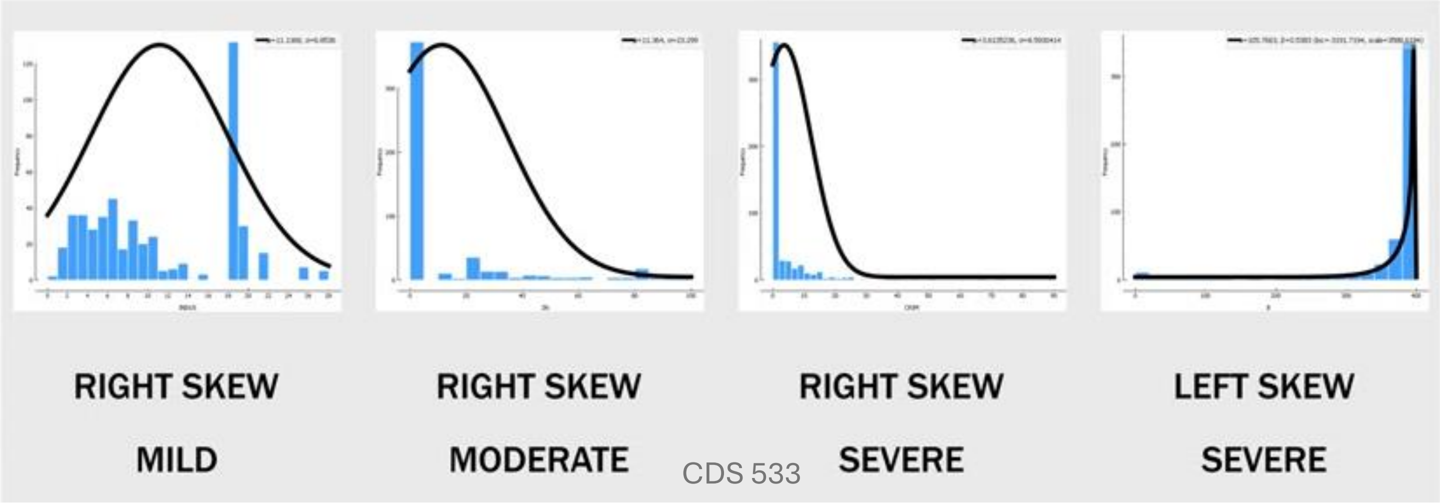
- In many case, a simple deterministic (nono-random) mathematical function is applied to one or more variables.
- Often, the transformation is a relatively simple algebraic function that it reversible:
  - Power functions: x^{λ} ,the most common which is $\sqrt{x} = x^{\frac{1}{2}}$
  - Logarithmic functions: $\log_{10} x$, $\log_{e} x = \ln x$
  - Complex transformations such as Box-Cox, Yeo-Johnson
  - …

Fundamentals of Transformation

Skewness Direction

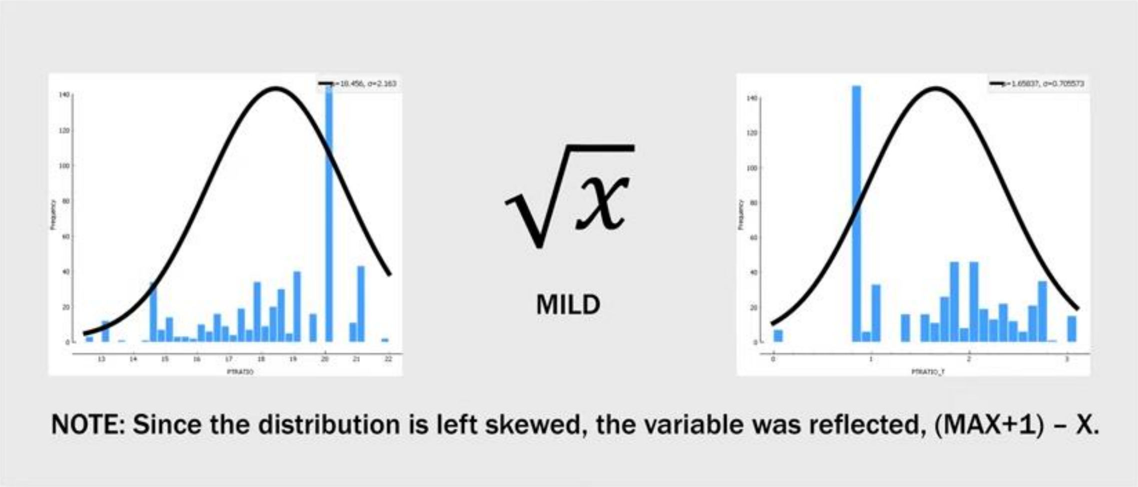


Skewness Magnitude



Fundamentals of Transformation

Square Root Transform

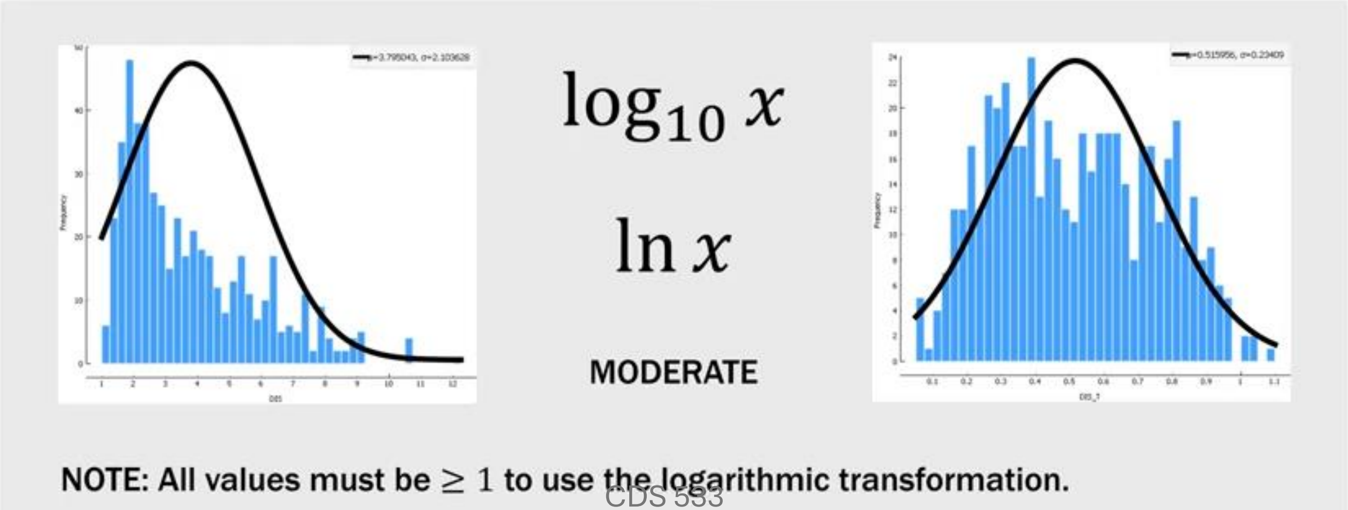


$\sqrt{x}$

MILD

NOTE: Since the distribution is left skewed, the variable was reflected, (MAX+1) – X.

Logarithmic Transformation



$\log_{10} x$

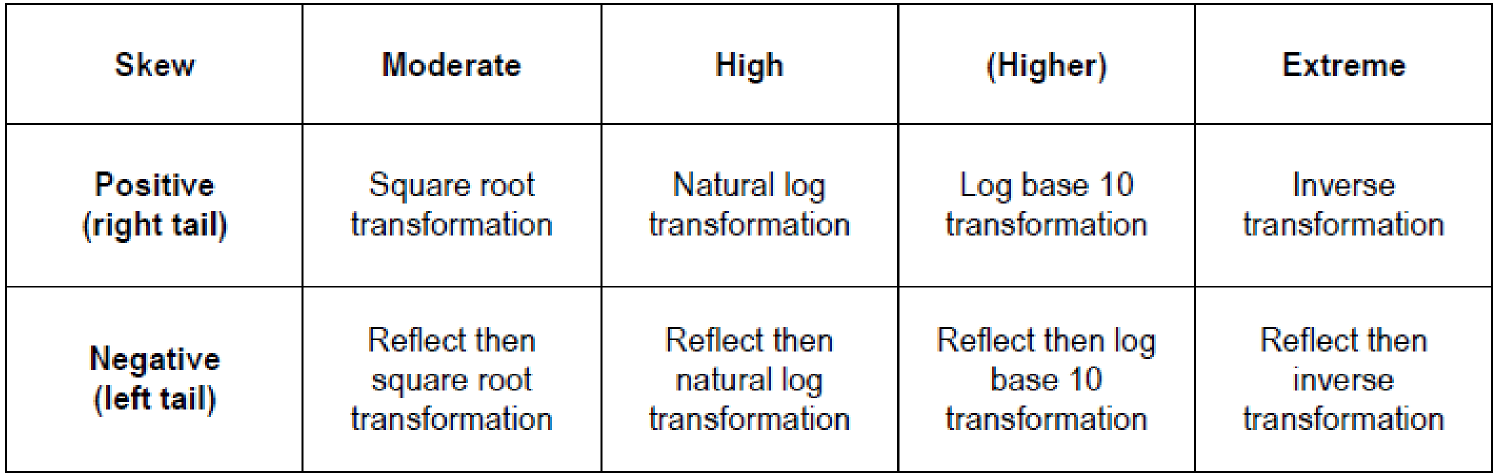
$\ln x$

MODERATE

NOTE: All values must be ≥ 1 to use the logarithmic transformation.

Fundamentals of Transformation

Summary on the Transformation

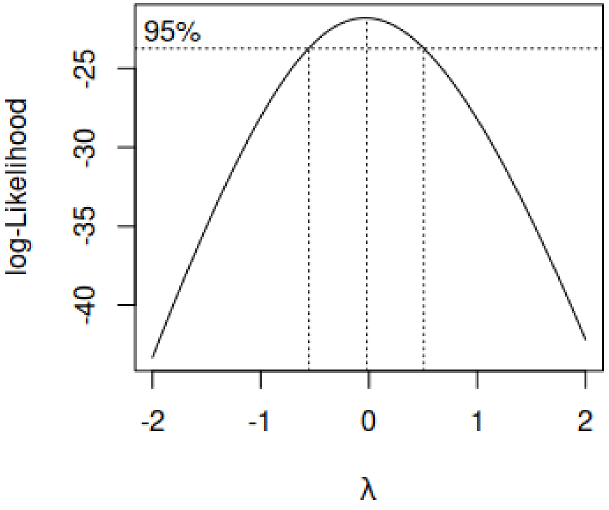


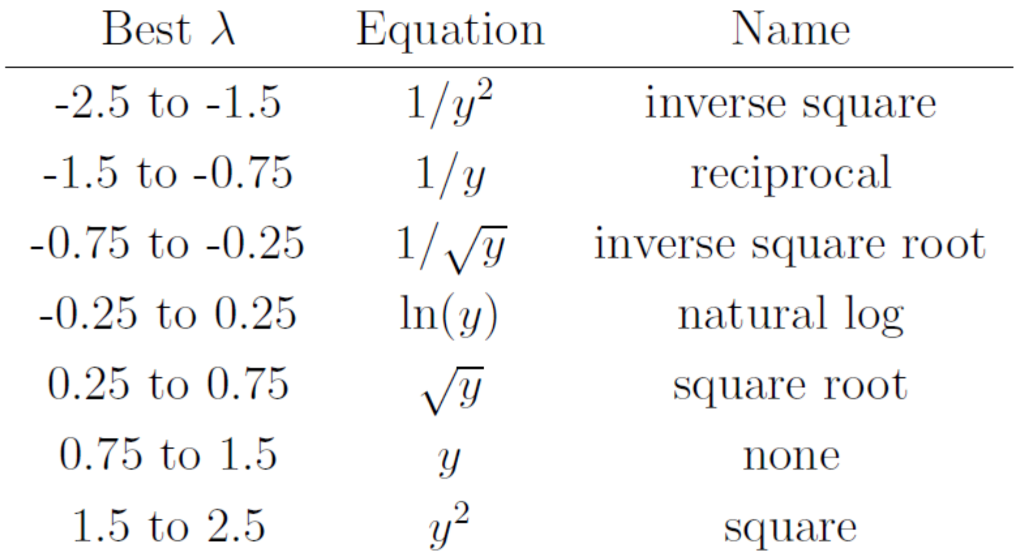
Skew	Moderate	High	(Higher)	Extreme
Positive (right tail)	Square root transformation	Natural log transformation	Log base 10 transformation	Inverse transformation
Negative (left tail)	Reflect then square root transformation	Reflect then natural log transformation	Reflect then log base 10 transformation	Reflect then inverse transformation

Fundamentals of Transformation

Box-Cox Method

- Power functions: x^{λ}
- George Box and David Cox developed a procedure to identify an appropriate exponent (Lamba=1) to use to transform data into a “normal shape” (searches from [-5,5]).
- The Lambda value indicates the power to which all data should be raised.





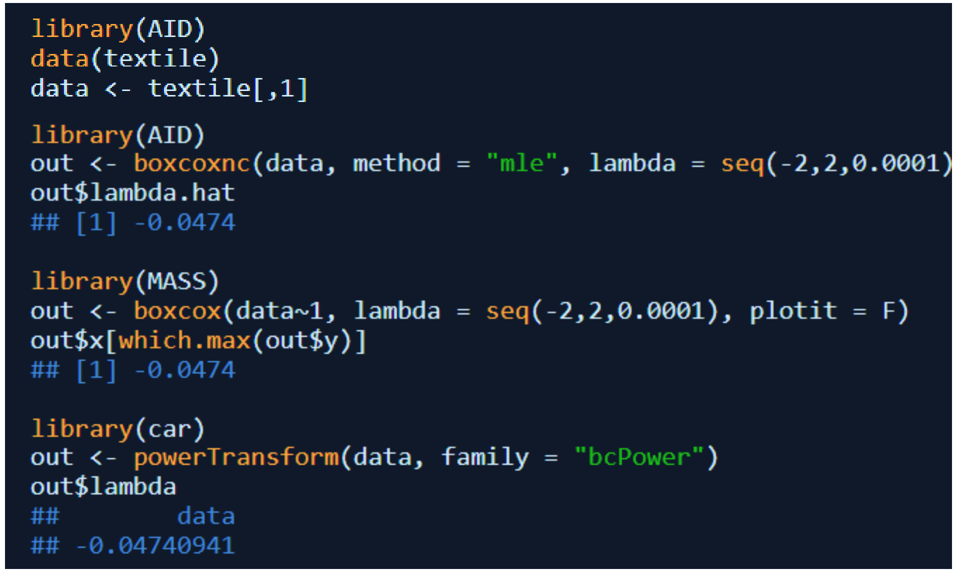
Best λ	Equation	Name
-2.5 to -1.5	$1/y^2$	inverse square
-1.5 to -0.75	$1/y$	reciprocal
-0.75 to -0.25	$1/\sqrt{y}$	inverse square root
-0.25 to 0.25	$\ln(y)$	natural log
0.25 to 0.75	$\sqrt{y}$	square root
0.75 to 1.5	y	none
1.5 to 2.5	y^2	square

Fundamentals of Transformation

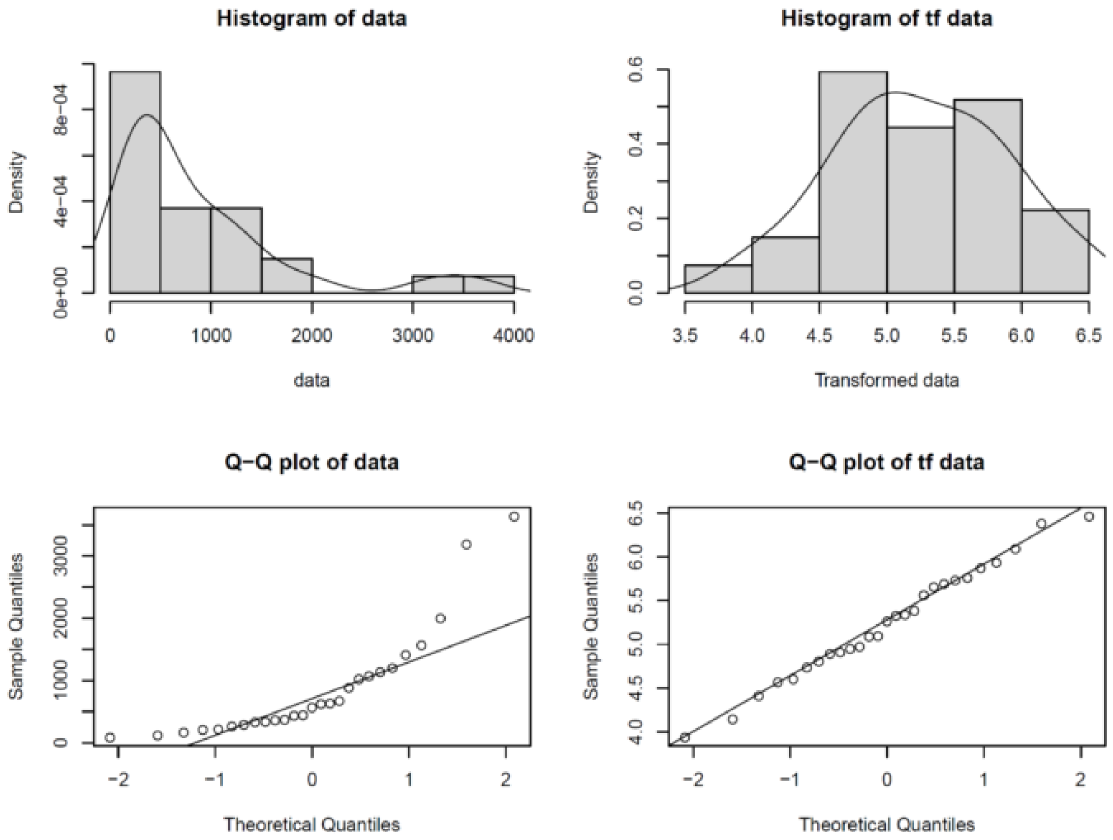
Box-Cox Method (R)

There are three ways of this estimation.

1. boxcoxnc() function in AID package
2. boxcox() function in MASS package
3. powerTransform function in car package



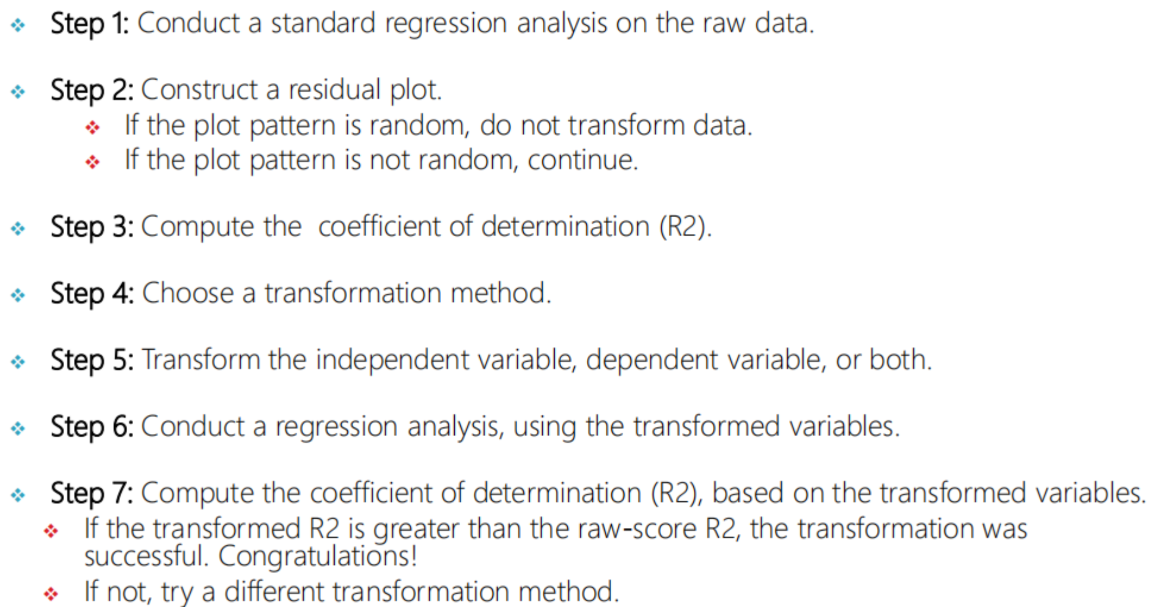
```
library(AID)
data(textile)
data <- textile[,1]

library(AID)
out <- boxcoxnc(data, method = "mle", lambda = seq(-2,2,0.0001)
out$lambda.hat
## [1] -0.0474

library(MASS)
out <- boxcox(data~1, lambda = seq(-2,2,0.0001), plotit = F)
out$x[which.max(out$y)]
## [1] -0.0474

library(car)
out <- powerTransform(data, family = "bcPower")
out$lambda
##        data
## -0.04740941
```



Fundamentals of Transformation

Transformations Guide

Transforming a data set to enhance linearity is a multi-step, trial-and-error process.



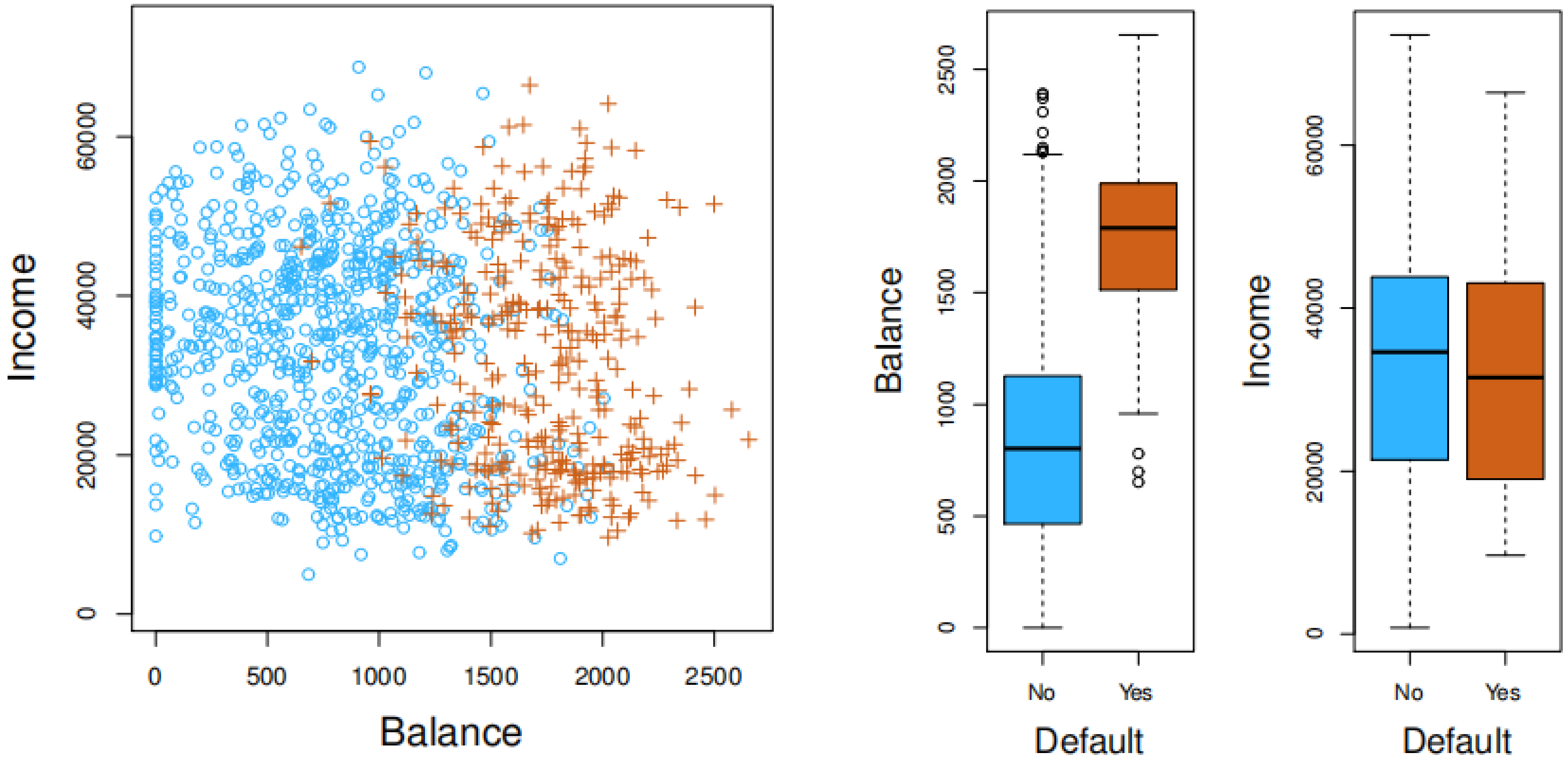
- **Step 1:** Conduct a standard regression analysis on the raw data.
- **Step 2:** Construct a residual plot.
  - If the plot pattern is random, do not transform data.
  - If the plot pattern is not random, continue.
- **Step 3:** Compute the coefficient of determination (R2).
- **Step 4:** Choose a transformation method.
- **Step 5:** Transform the independent variable, dependent variable, or both.
- **Step 6:** Conduct a regression analysis, using the transformed variables.
- **Step 7:** Compute the coefficient of determination (R2), based on the transformed variables.
  - If the transformed R2 is greater than the raw-score R2, the transformation was successful. Congratulations!
  - If not, try a different transformation method.

Challenges with Transformations

- Variables that have been transformed are sometimes harder to interpret.
- Can risk unintended consequences in variable relationships.

Logistic Regression

Classification



Logistic Regression

Logistic Regression

- Logistic Regression is used for solving a classification problem.
- Logistic Regression is pretty much the same as linear regression.
- Logistic Regression extends the idea of linear regression to a situation where the outcome variable is categorical.

i.e.,

Logistic regression:

– outcome is a categorical variable (usually binary – yes/no)

– risk factors are either continuous or categorical variables

Linear regression:

– outcome is a continuous variable

– risk factors are either continuous or categorical variables

Logistic Regression

Example

Let’s try to predict whether the credit card holder will default the payment.

The bank balance and the income are given as independent variables (feature), and the response variable has two values, ‘Yes’ and ‘No’

- Logistic regression models the probability of default. For example, the probability of default given balance can be written as

$$\Pr(\texttt{default} = \texttt{Yes}|\texttt{balance}).$$

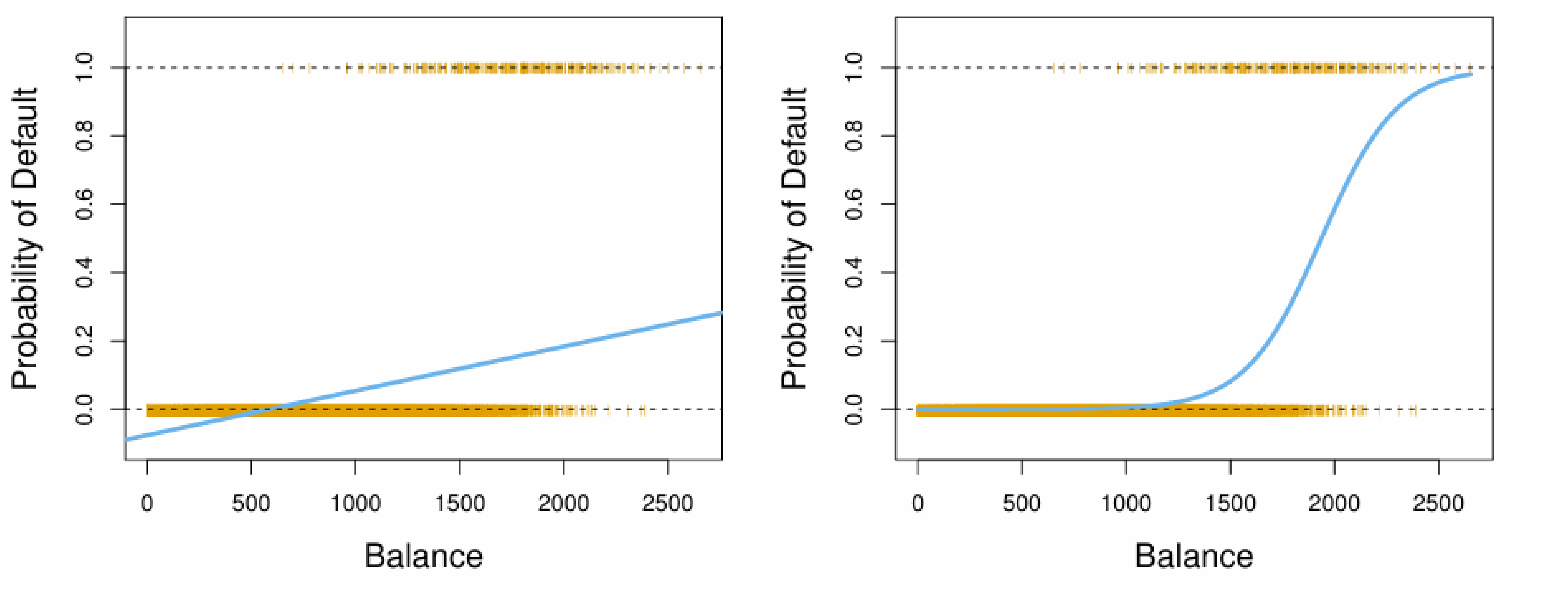
Suppose for the Default classification task that we code

$$Y = \begin{cases} 0 & \text{if } \texttt{No} \\ 1 & \text{if } \texttt{Yes}. \end{cases}$$

Can we simply perform a linear regression of Y on X and classify as Yes if $\hat{Y} > 0.5$?

Logistic Regression

Example (cont.)



The orange marks indicate the response Y, either 0 or 1.Linear regression does not estimate $\Pr(Y = 1|X)$ well. Logistic regression seems well suited to the task.

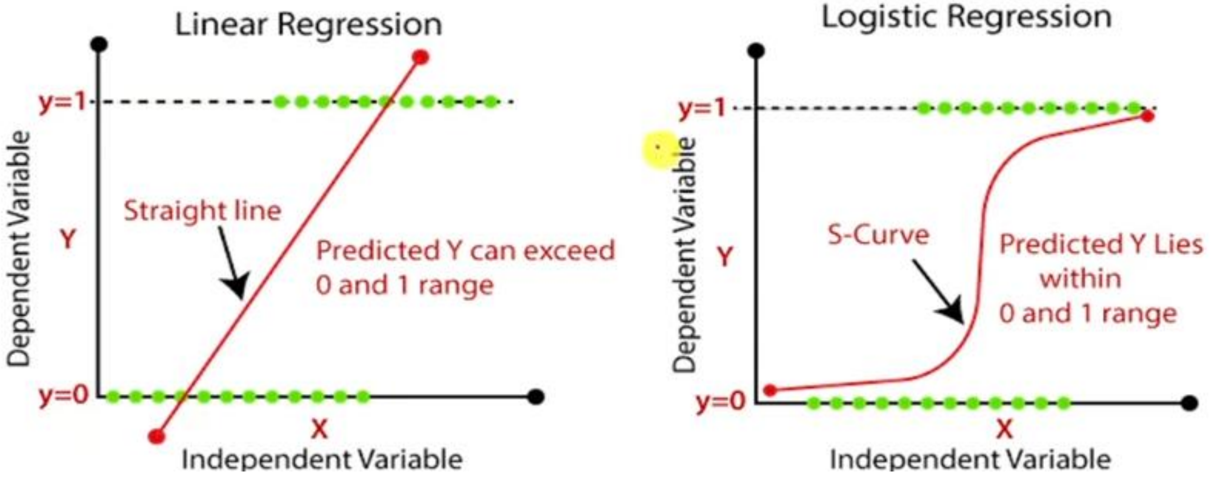
Logistic Regression

Model Expression

Let ‘s write $p(X) = \Pr(Y = 1|X)$ for short and consider using balance to predict default. Logistic regression uses the form

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

($e \approx 2.71828$ is a mathematical constant [Eulers number.]) It is easy to see that no matter what values 0, 1 or X take, $p(X)$ will have values between 0 and 1.



Logistic Regression

Odd Ratio

After a bit of manipulation, we find

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0 + \beta_1 X}.$$

The quantity $p(X)/[1 - p(X)]$ is called the **odds.**

$$Odds = \frac{proability\ of\ having\ an\ event}{proability\ of\ not\ having\ the\ event}$$

$$Odds = \frac{p}{1 - p}$$

- The value of odds range from zero to ∞ and the value of probability lies between zero and one.

Logistic Regression

Model Expression

A bit of rearrangement gives

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X.$$

- This monotone transformation is called the **log odds** or **logit** transformation of $p(X)$. (by log we mean *natural* log: ln.). [**Link function**]

➢ We use maximum likelihood to estimate the parameters.

$$\ell(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)).$$

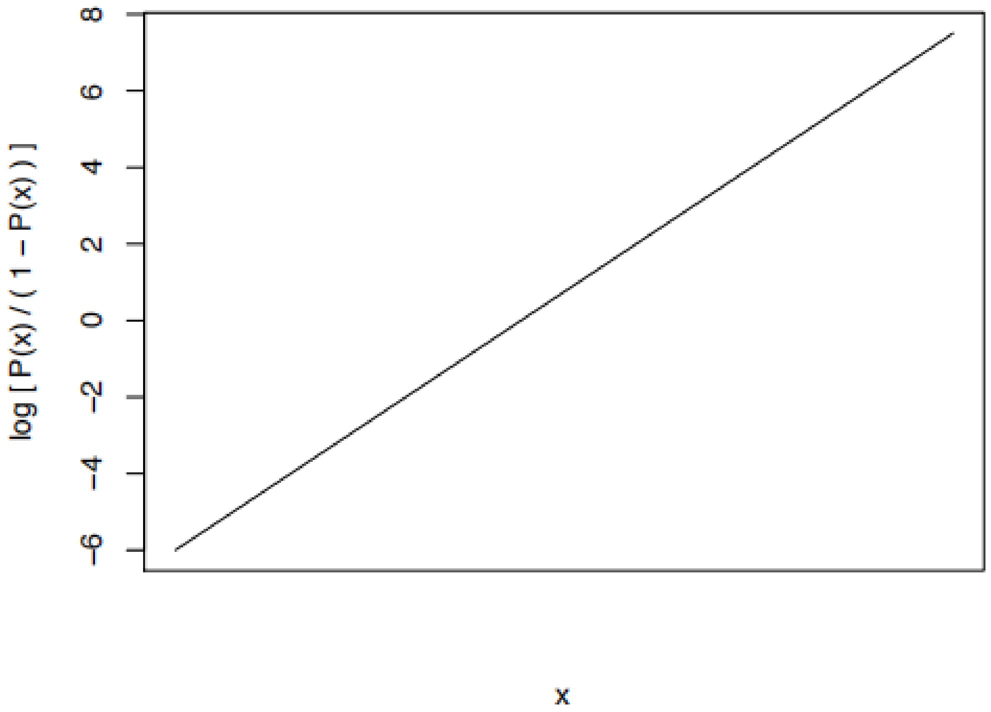
- This likelihood gives the probability of the observed zeros and ones in the data. We pick 0 and 1 to maximize the likelihood of the observed data.

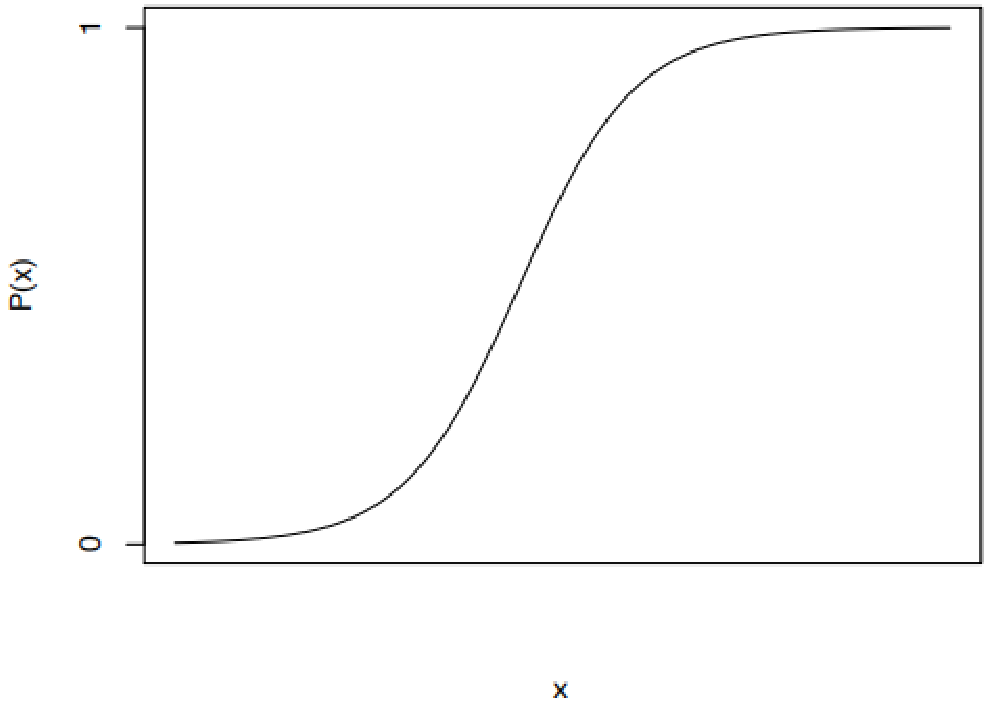
Logistic Regression

Relationships

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X.$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}.$$

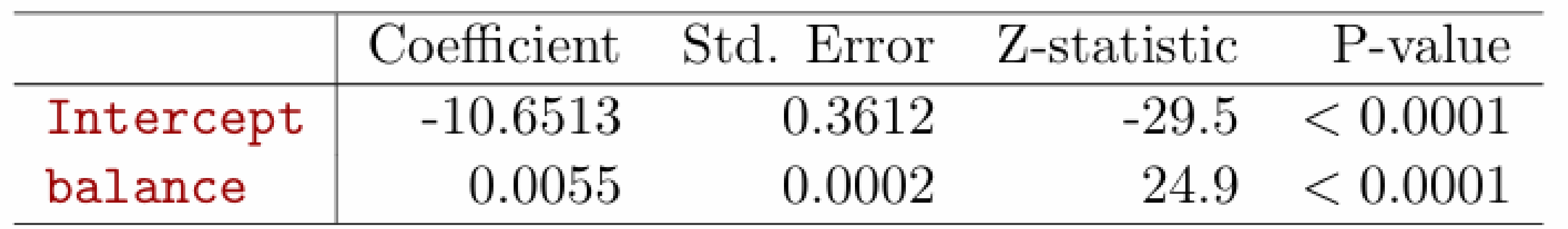




Logistic Regression

Logistic Regression in R

Most statistical packages can fit linear logistic regression models by maximum likelihood. In **R** we use the **glm** function.



	Coefficient	Std. Error	Z-statistic	P-value
`Intercept`	-10.6513	0.3612	-29.5	< 0.0001
`balance`	0.0055	0.0002	24.9	< 0.0001

- We see that β_1 = 0.0055; this indicates that an increase in balance is associated with an increase in the probability of default. To be precise, a one-unit increase in balance is associated with an increase in the log odds of default by 0.0055 units.

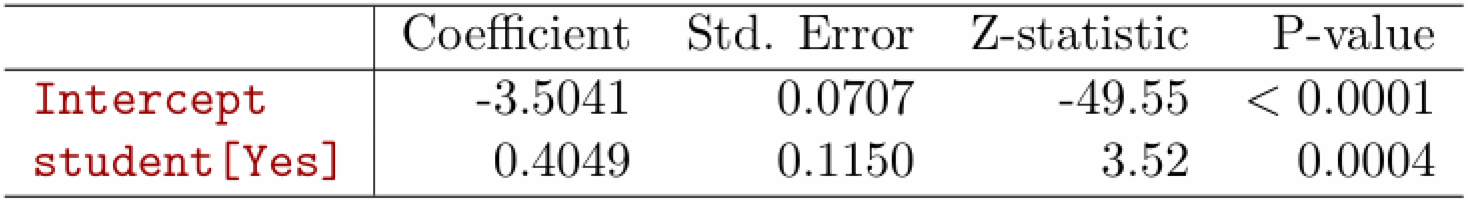
Logistic Regression

Make Predictions

What is our estimated probability of default for someone with a balance of $1000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times1000}}{1+e^{-10.6513+0.0055\times1000}} = 0.006$$

- One can use qualitative predictors with the logistic regression model using the dummy variable approach, using student as the predictor. [value of 1 for a student]



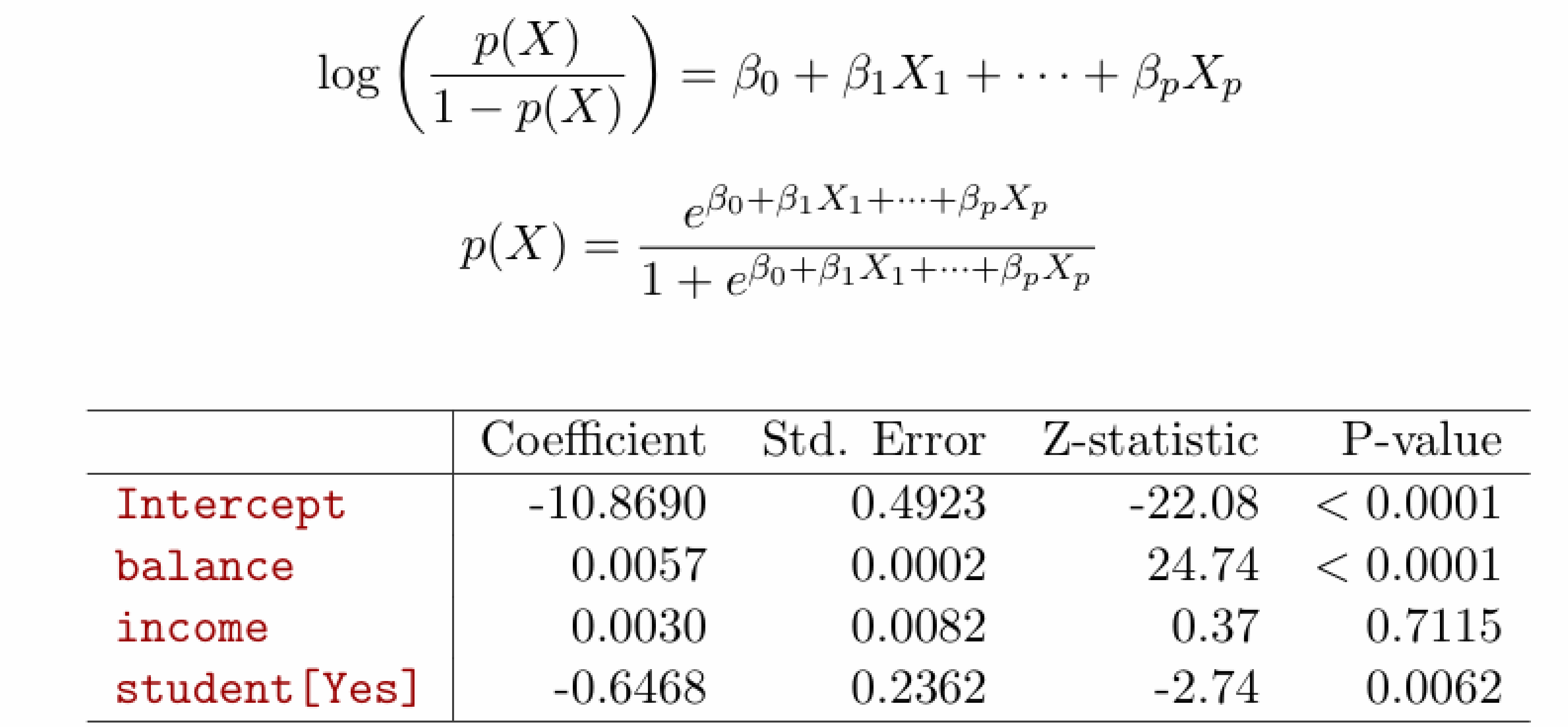
	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

  - The coefficient associated with the dummy variable is positive, and the associated p-value is statistically significant. This indicates that students tend to have higher default probabilities than non-students.

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=Yes}) = \frac{e^{-3.5041+0.4049\times1}}{1+e^{-3.5041+0.4049\times1}} = 0.0431, \quad \widehat{\Pr}(\texttt{default=Yes}|\texttt{student=No}) = \frac{e^{-3.5041+0.4049\times0}}{1+e^{-3.5041+0.4049\times0}} = 0.0292.$$

Logistic Regression

Multiple Logistic Regression



$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

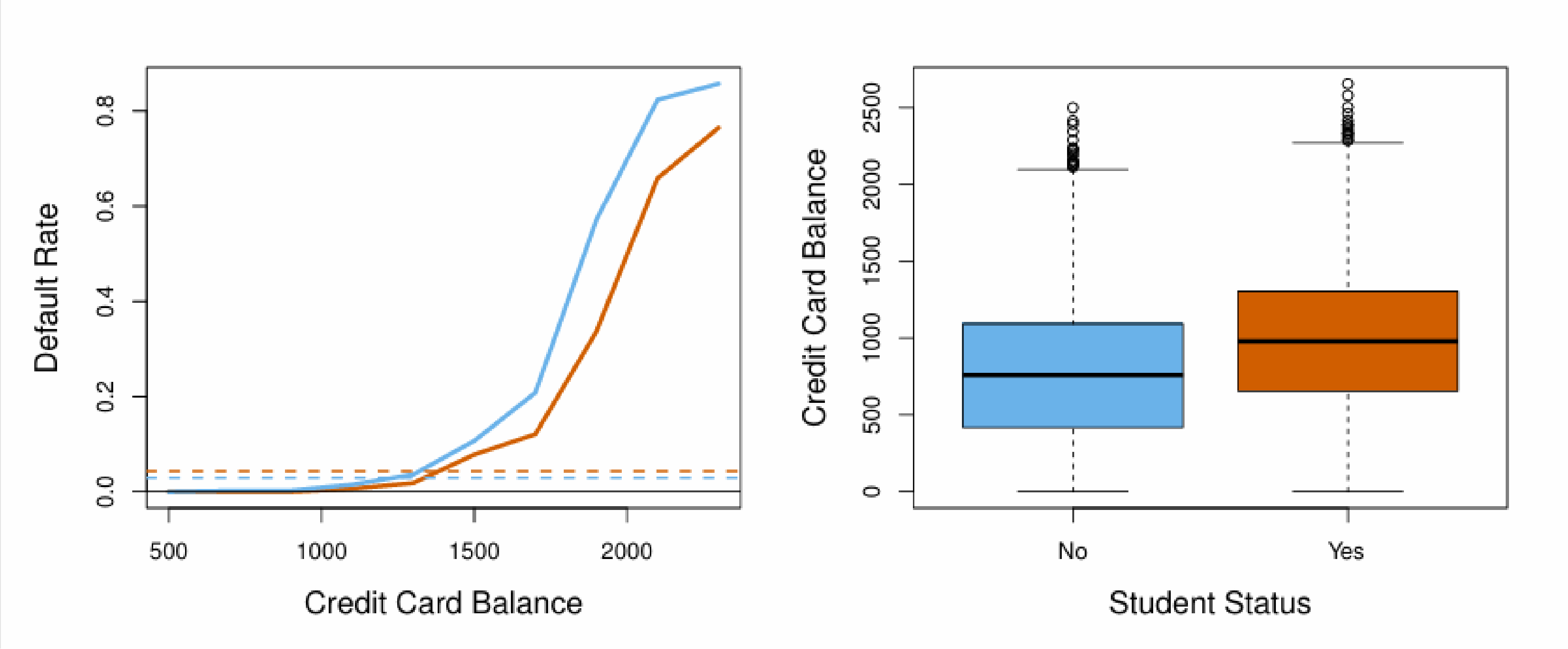
$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	-0.6468	0.2362	-2.74	0.0062

Why is the coefficient for student negative, while it was positive before?

Logistic Regression

Confounding



- Students tend to have higher balances than non-students, so their marginal default rate is higher than for non-students.
- But for each level of balance, students default less than non-students.
- Multiple logistic regression can tease this out.

Logistic Regression

Logistic regression with more than two classes

- It is easily generalized to more than two classes to the setting of $K > 2$ classes. This extension is sometimes known as **multinomial logistic regression**.
- To do this, we first select a single multinomial logistic regression class to serve as the baseline; without loss of generality, we select the K^{th} class for this role.
- One version (used in the R package glmnet) has the symmetric form

$$\Pr(Y = k|X) = \frac{e^{\beta_{0k}+\beta_{1k}X_1+\ldots+\beta_{pk}X_p}}{\sum_{\ell=1}^{K} e^{\beta_{0\ell}+\beta_{1\ell}X_1+\ldots+\beta_{p\ell}X_p}}$$

Generalized Linear Models

Motivation

Introduction of **linear models**, which assume a **linear relationship** between the mean of the response variable Y and a set of explanatory variables, with inference assuming that Y has a normal conditional distribution with constant variance.

- The **generalized linear model** *permits distributions for Y other than the normal and permits modelling nonlinear functions of the mean, such as non-negative responses, skewed distributions, and more.*
  - OLS regression
  - Logistic regression model
  - Poisson
  - ...

Generalized Linear Models

Introduction

Generalized linear models (GLMs) extend normal linear models to encompass non-normal response distributions and equating linear predictors to nonlinear functions of the mean. They provide a **unified theory** of modeling that includes the most important models for continuous and discrete response variables.

- **The Three Components of GLMs**
  - Response variable
  - Explanatory variable
  - Link function: This component is a function g applied to the *conditional expectation* $\mu_i = E(Y_i|x_{1i}, \dots, x_{pi})$of the response variable at explanatory variable values $(x_{1i}, \dots, x_{pi})$, relating it to the linear predictor,

$$g(\mu_i) = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi}$$

Generalized Linear Models

Example: Bikeshare Data

Linear regression with response bikers: number of hourly users in bikeshare program in Washington, DC.

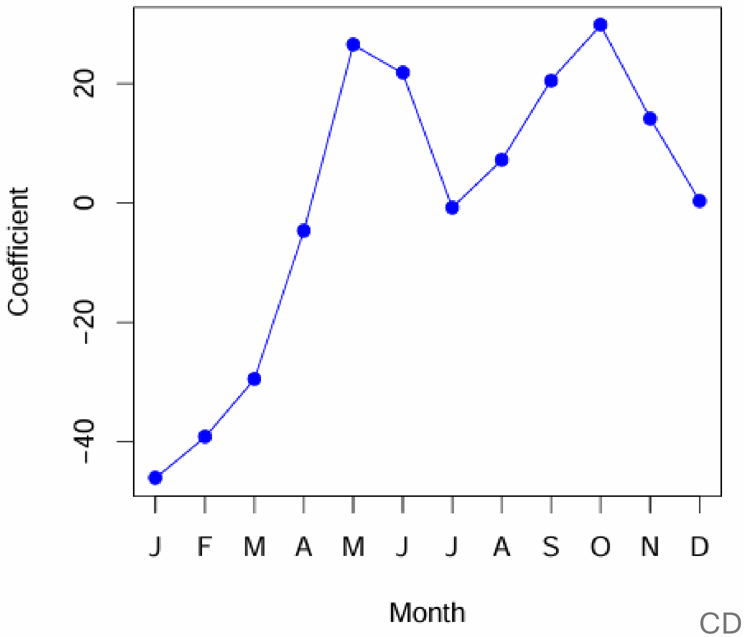
Predictors:

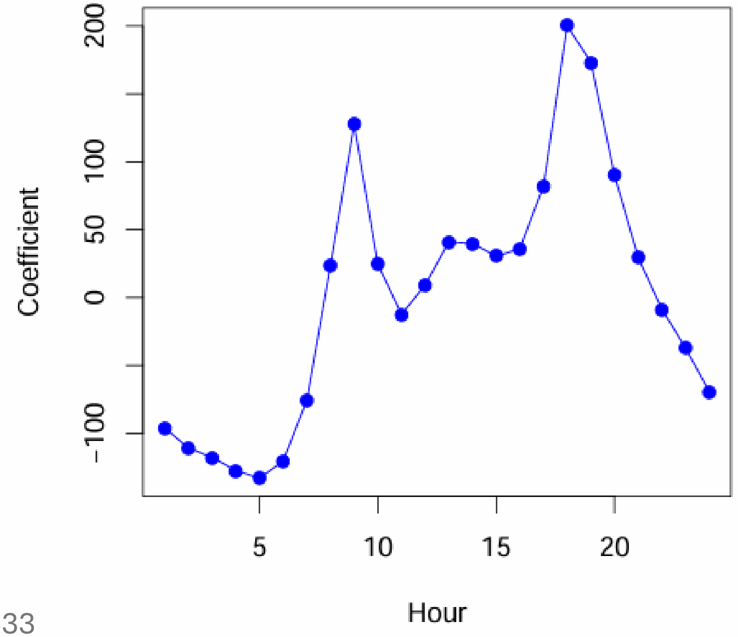
- mnth (month of the year)
- hr (hour of the day, from 0 to 23)
- workingday (an indicator variable that equals 1 if it is neither a weekend nor a holiday)
- temp (the normalized temperature in Celsius)
- weathersit (a qualitative variable that takes on one of four possible values: clear; misty or cloudy; light rain or light snow; or heavy rain or heavy snow.)

Generalized Linear Models

Example: Bikeshare Data

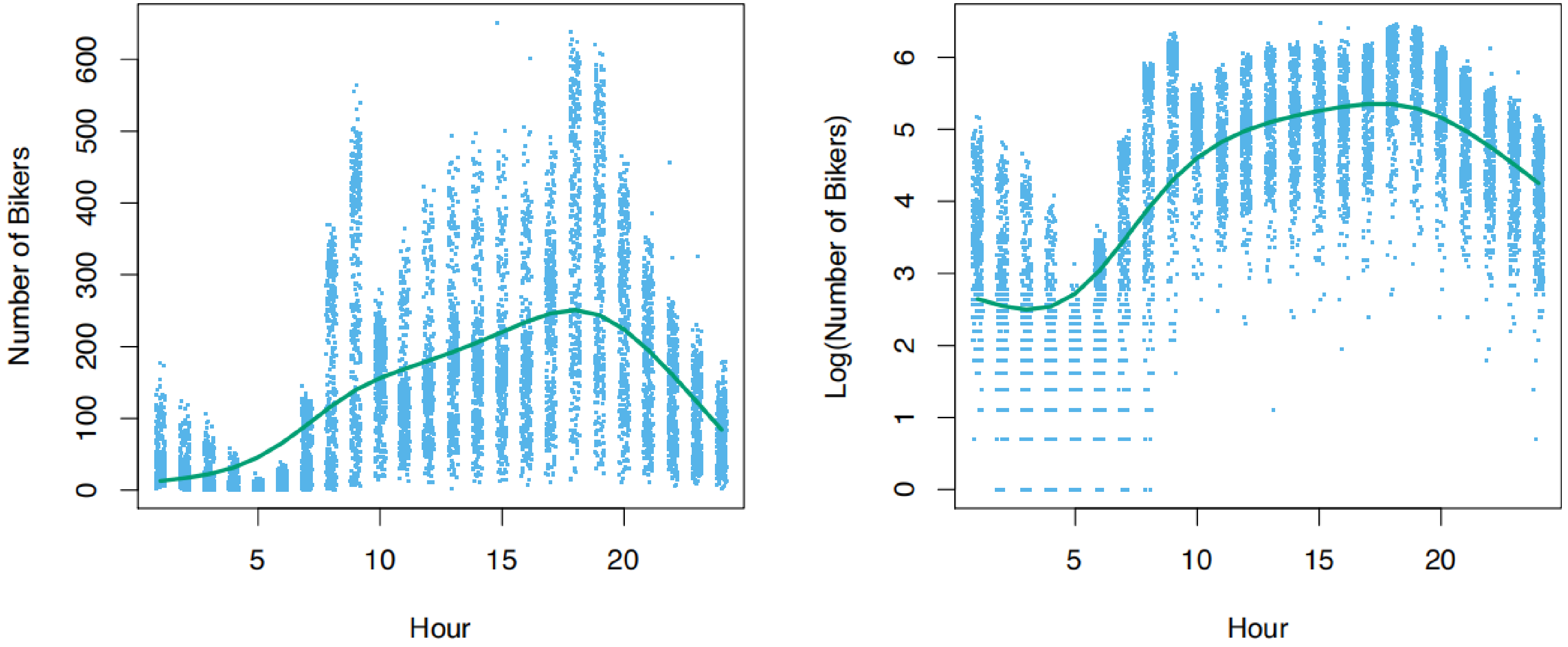
	Coefficient	Std. error	z-statistic	p-value
`Intercept`	73.60	5.13	14.34	0.00
`workingday`	1.27	1.78	0.71	0.48
`temp`	157.21	10.26	15.32	0.00
`weathersit[cloudy/misty]`	-12.89	1.96	-6.56	0.00
`weathersit[light rain/snow]`	-66.49	2.97	-22.43	0.00
`weathersit[heavy rain/snow]`	-109.75	76.67	-1.43	0.15





Generalized Linear Models

Example: Bikeshare Data



- In left plot we see that the variance mostly in creases with the mean.
- 10% of linear model predictions are negative (not shown here)
- Taking log(bikers) alleviates this, but has its own problems: e.g. predictions are on the wrong scale, and some counts are zero.

Poisson Regression

Motivation: Bikeshare Data

- Poisson distribution is useful for modeling counts:

$$\Pr(Y = k) = \frac{e^{-\lambda}\lambda^k}{k!} \quad \text{for } k = 0, 1, 2, \ldots$$

- $\lambda = E(Y) = Var(Y)$ – i.e., there is a mean/variance dependence.

- With covariates, we model

$$\log(\lambda(X_1, \ldots, X_p)) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

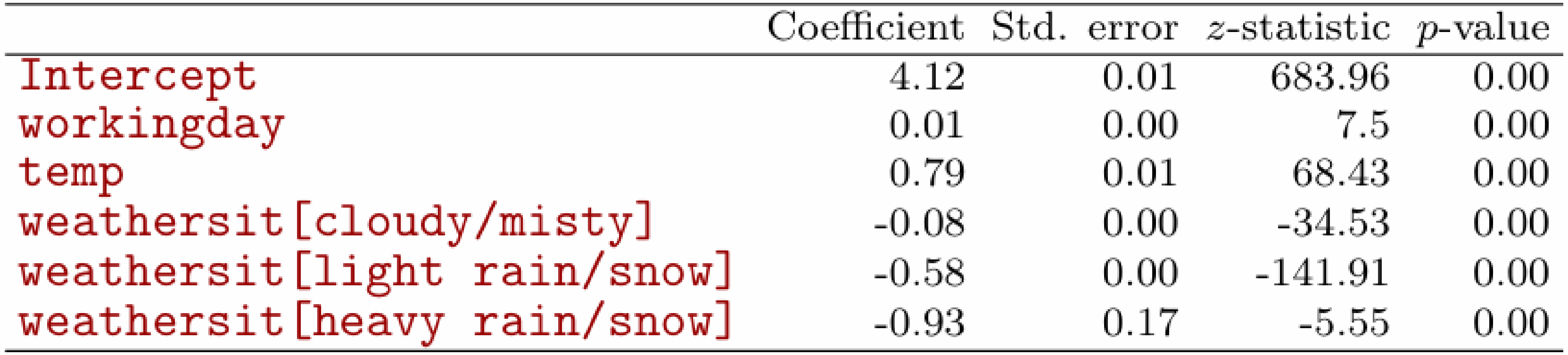
or equivalently,

$$\lambda(X_1, \ldots, X_p) = e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}.$$

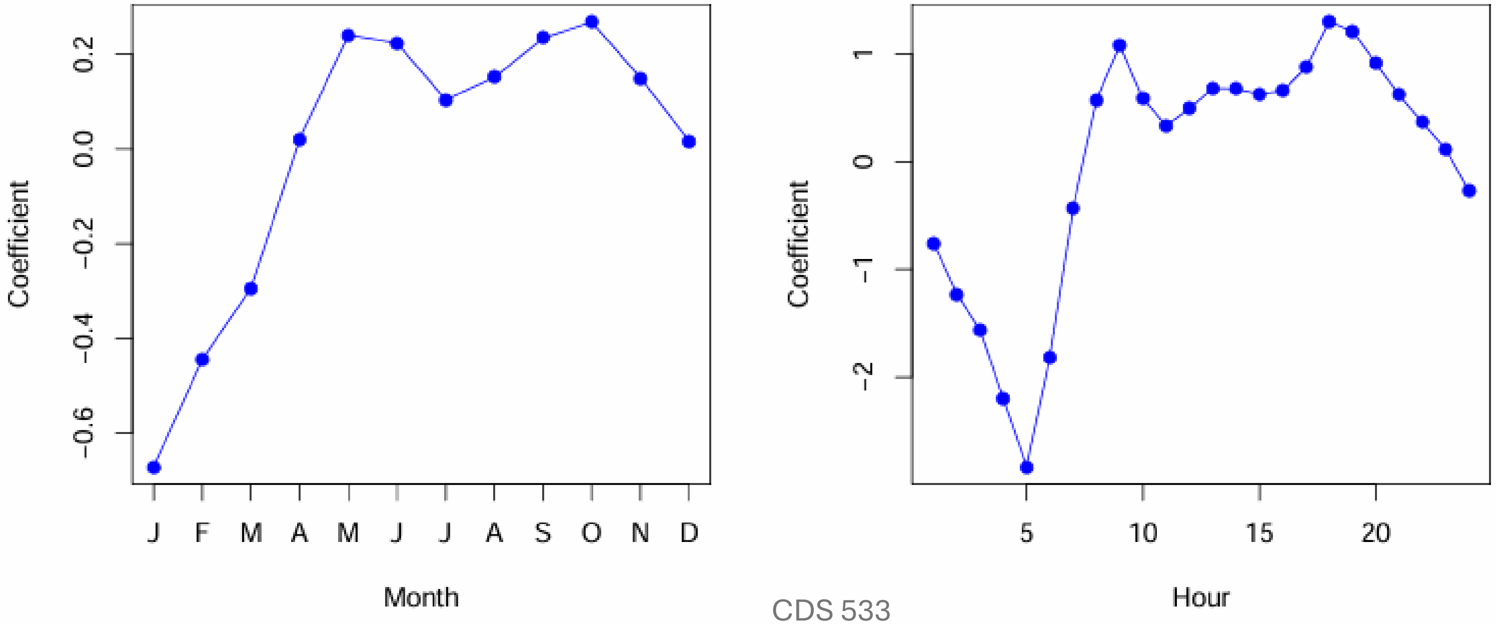
- Model automatically guarantees that the predictions are non-negative.

Poisson Regression

Motivation: Bikeshare Data

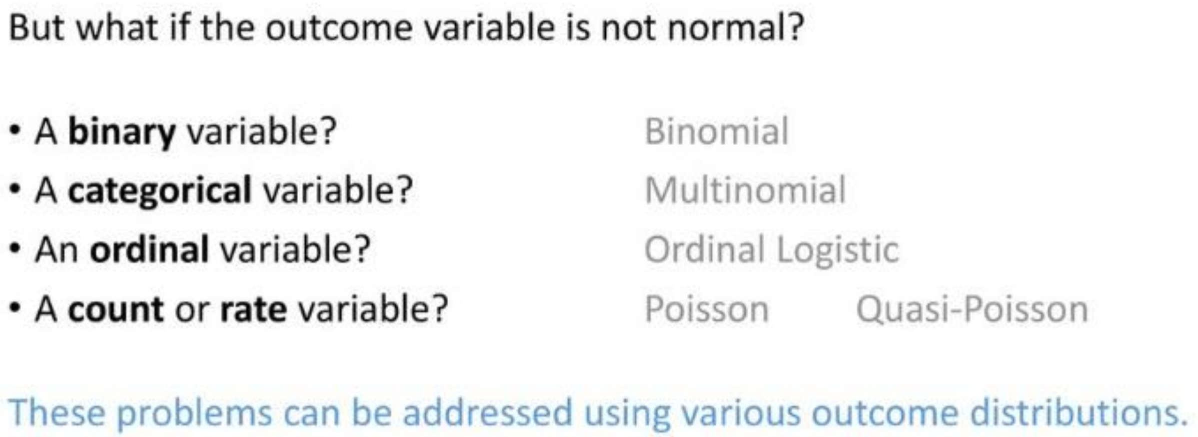


	Coefficient	Std. error	z-statistic	p-value
Intercept	4.12	0.01	683.96	0.00
workingday	0.01	0.00	7.5	0.00
temp	0.79	0.01	68.43	0.00
weathersit[cloudy/misty]	-0.08	0.00	-34.53	0.00
weathersit[light rain/snow]	-0.58	0.00	-141.91	0.00
weathersit[heavy rain/snow]	-0.93	0.17	-5.55	0.00



Generalized Linear Models (Limited)

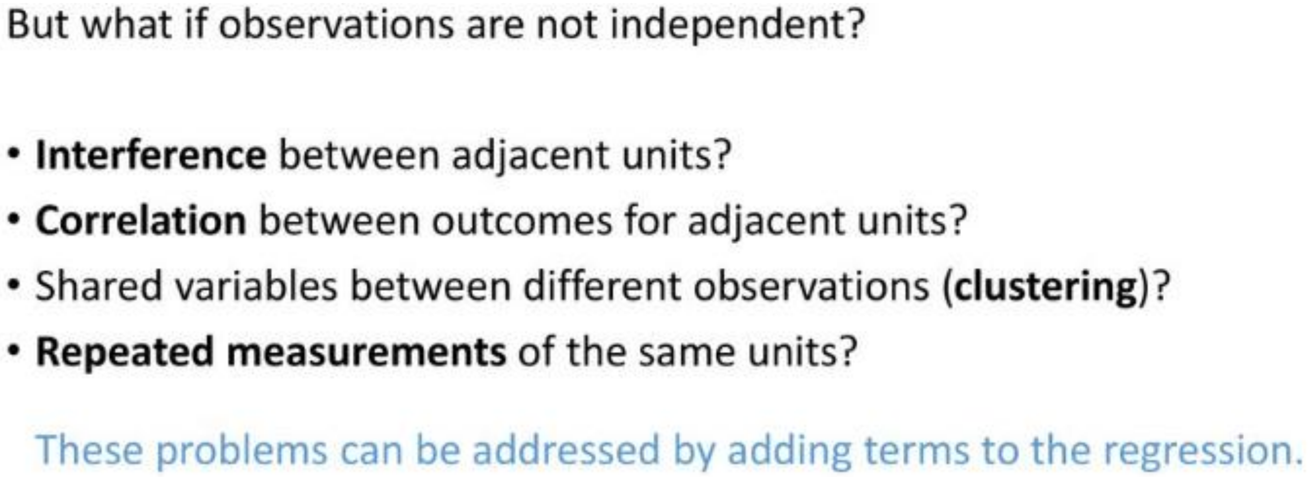
glm in R



But what if the outcome variable is not normal?

• A **binary** variable?	Binomial	
• A **categorical** variable?	Multinomial	
• An **ordinal** variable?	Ordinal Logistic	
• A **count** or **rate** variable?	Poisson	Quasi-Poisson

These problems can be addressed using various outcome distributions.



But what if observations are not independent?

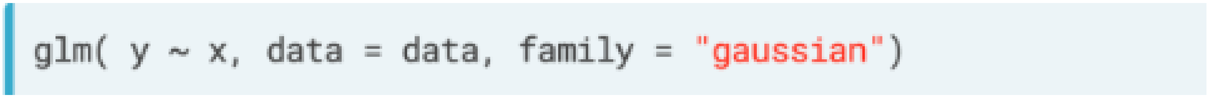
- **Interference** between adjacent units?
- **Correlation** between outcomes for adjacent units?
- Shared variables between different observations (**clustering**)?
- **Repeated measurements** of the same units?

These problems can be addressed by adding terms to the regression.

Generalized Linear Models (Limited)

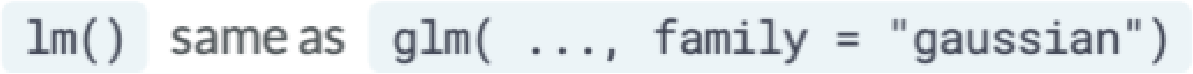
glm() syntax

GLMs are fit with function glm(). Like linear models (lm()s), glm()s have formulas and data as inputs, but also have a family input.

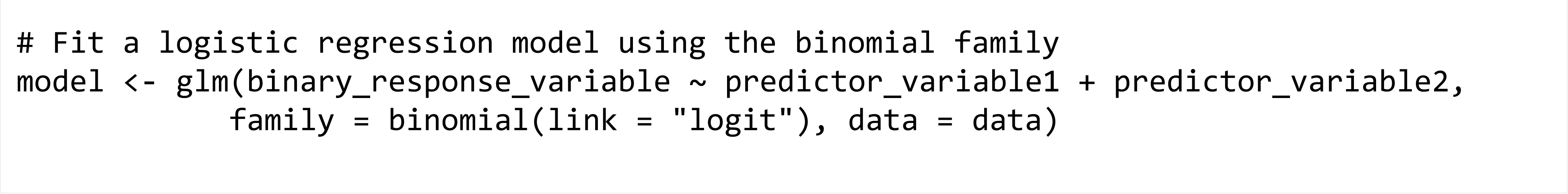


```
glm( y ~ x, data = data, family = "gaussian")
```

- The Gaussian family is how R refers to the normal distribution and is the default for a glm()



`lm()` same as `glm( ..., family = "gaussian")`



```
# Fit a logistic regression model using the binomial family
model <- glm(binary_response_variable ~ predictor_variable1 + predictor_variable2,
            family = binomial(link = "logit"), data = data)
```

https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/glm
https://docs.databricks.com/en/sparkr/glm-tutorial.html

Resampling Methods

Overview

- In the section, we discuss two resampling methods: **cross-validation and the bootstrap.**
- These methods refit a model of interest to samples formed from the training set, in order to obtain additional information about the fitted model.
- For example, they provide estimates of test-set prediction error, and the standard deviation and bias of our parameter estimates.

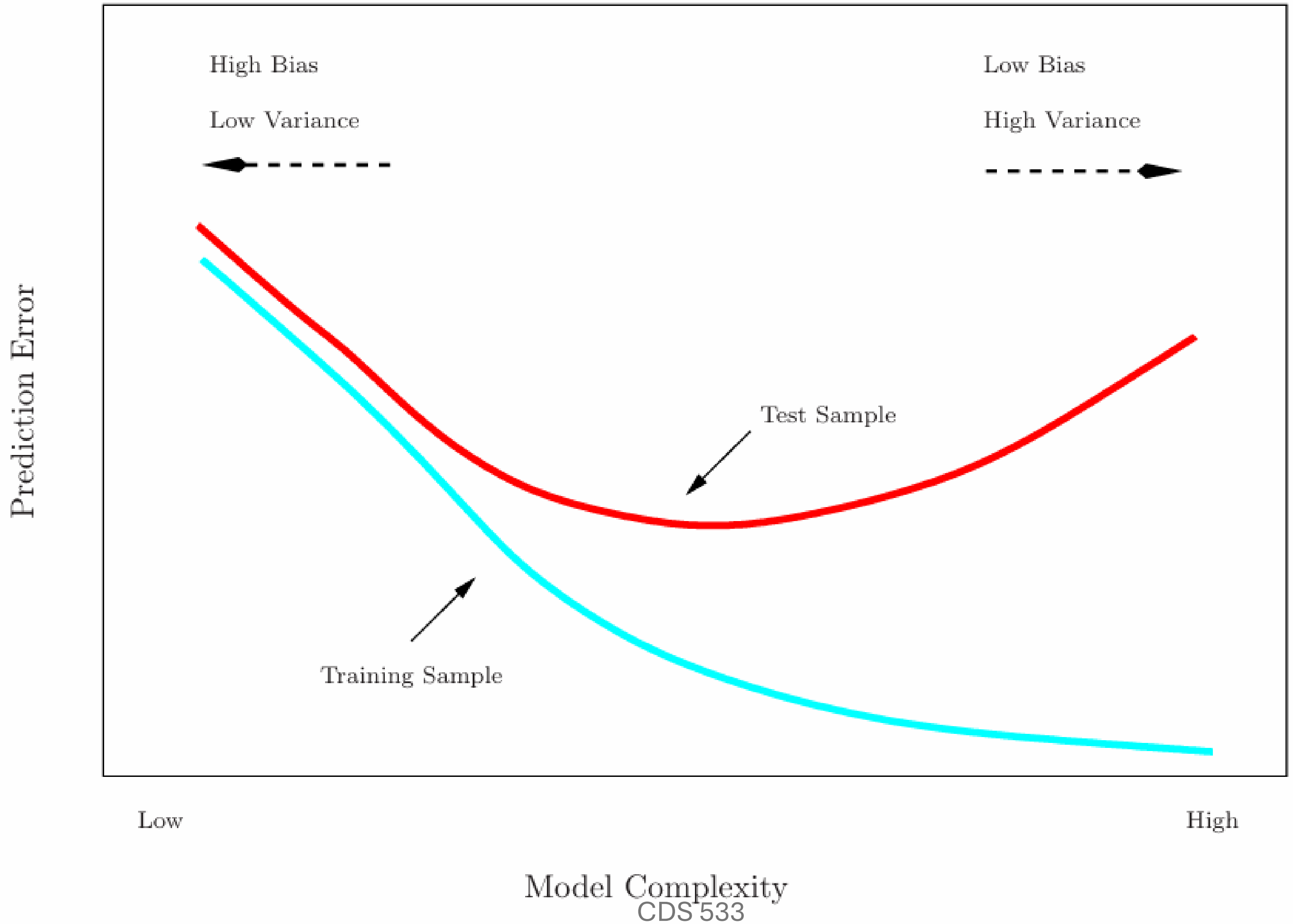
Resampling Methods

Training Error Versus Test Error

- Recall the distinction between the **test error** and the **training error**:
- The **test error** is the average error that results from using a statistical learning method to predict the response on a new observation, one that was not used in training the method.
- In contrast, the **training error** can be easily calculated by applying the statistical learning method to the observations used in its training.
- But the training error rate often is quite different from the test error rate, and in particular the former can dramatically underestimate the latter.

Resampling Methods

Training- versus Test-Set Performance



Prediction Error Estimates

Best solution: a large designated test set. Often not available.

Some methods make a mathematical adjustment to the training error rate in order to estimate the test error rate. These include the Cp statistic, AIC and BIC. They are discussed in the next class.

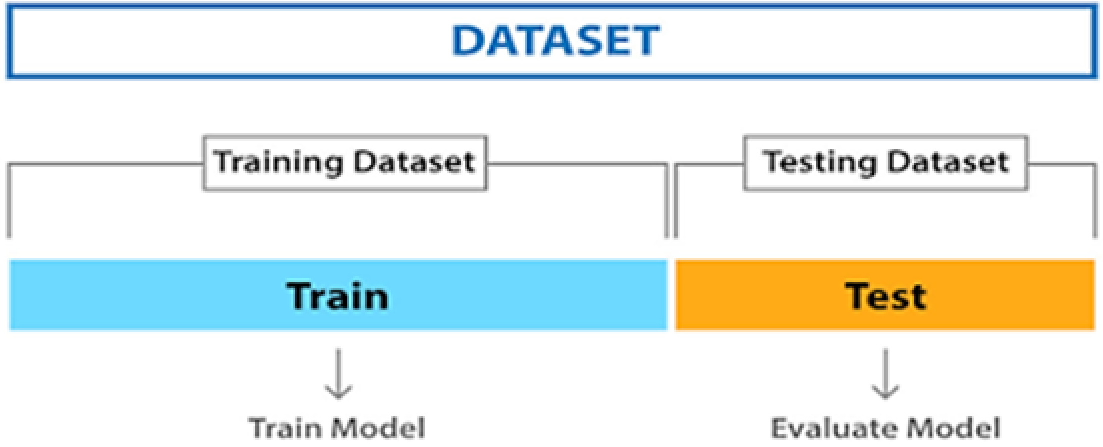
Here we instead consider a class of methods that estimate the test error by **holding out a subset of the training observations** from the fitting process, and then applying the statistical learning method to those held out observations.

Validation-set Approach

Here we randomly divide the available set of samples into two parts: a **training set** and a **validation** or **hold-out set**.

The model is fit on the training set, and the fitted model is used to predict the responses for the observations in the validation set.

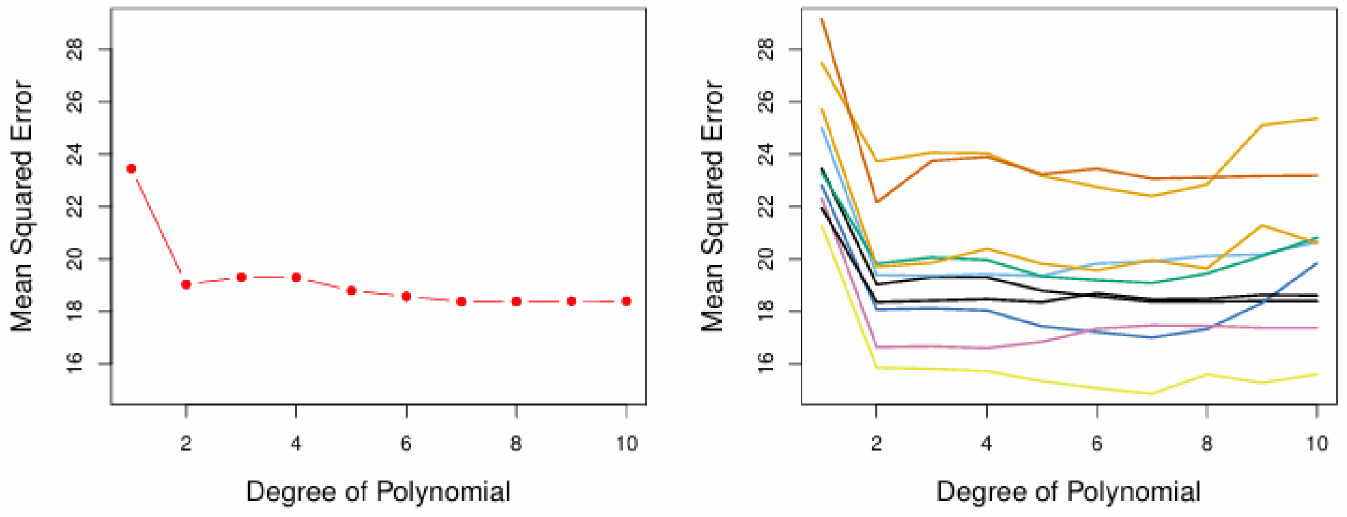
The resulting validation-set error provides an estimate of the test error. This is typically assessed using MSE in the case of a quantitative response and misclassification rate in the case of a qualitative (discrete) response.



Validation-set Approach

Example: Auto Data

- Want to compare linear vs higher-order polynomial terms in a linear regression
- We randomly split the 392 observations into two sets, a training set containing 196 of the data points, and a validation set containing the remaining 196 observations.



Left panel shows single split; right panel shows multiple splits

Validation-set Approach

Drawbacks of validation set approach

- The validation estimate of the test error can be highly variable, depending on precisely which observations are included in the training set and which observations are included in the validation set.
- In the validation approach, only a subset of the observations those that are included in the training set rather than in the validation set are used to fit the model.
- This suggests that the validation set error may tend to overestimate the test error for the model fit on the entire data set.

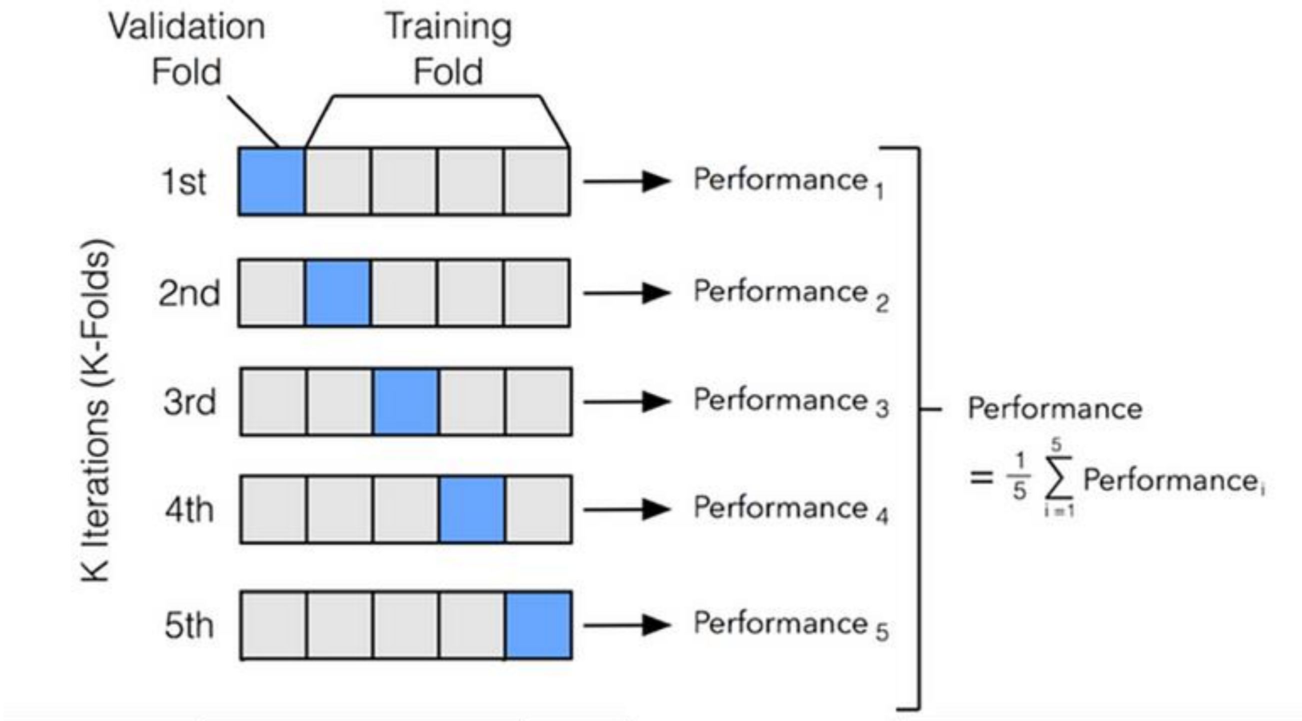
K-fold Cross-validation

K-fold Cross-validation

- Widely used approach for estimating test error.
- The dataset is divided into k equal-sized folds.
- The model is trained on $k-1$ folds and evaluated on the remaining fold.
- This procedure is repeated k times, ensuring each fold is used once as the test set.
- The overall performance is determined by averaging the results from all k iterations.

K-fold Cross-validation

K-fold Cross-validation



🟦 Test 🟥 Train

n = 12
k = 3

Data

1	2	3	4	5	6	7	8	9	10	11	12

K-fold Cross-validation

The Details

- Let the K parts be $C_1, C_2, \dots, C_k$, where C_k denotes the indices of the observation in part k. There are n_k observations in part k: if N is a multiple pf K, then $n_k = n/ K$.
- **[Regression]** Compute

$$\mathrm{CV}_{(K)} = \sum_{k=1}^{K} \frac{n_k}{n} \mathrm{MSE}_k$$

Where $\mathrm{MSE}_k = \sum_{i \in C_k} (y_i - \hat{y}_i)^2 / n_k$, and $\hat{y}_i$ is the fit for observation i, obtained from the data with part k removed.

- **[Classification]** Compute

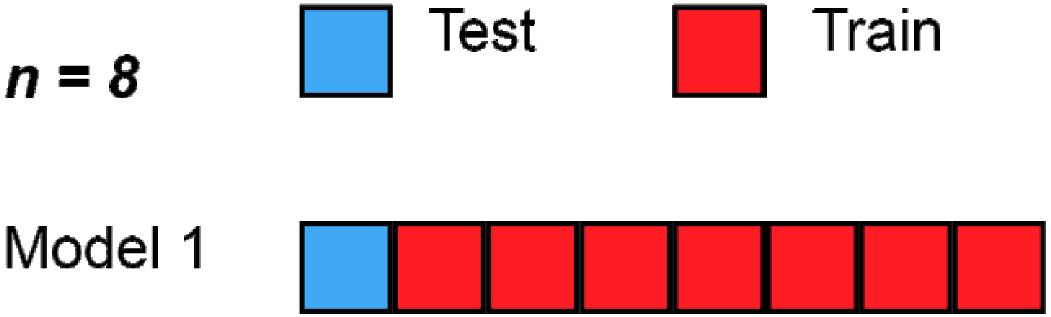
$$\mathrm{CV}_K = \sum_{k=1}^{K} \frac{n_k}{n} \mathrm{Err}_k$$

Where $\mathrm{MSE}_k = \sum_{i \in C_k} (y_i - \hat{y}_i)^2 / n_k$, and $\hat{y}_i$ is the fit for observation i, obtained from the data with part k removed.

Leave-one-out Cross-validation

Leave One Out Cross-validation

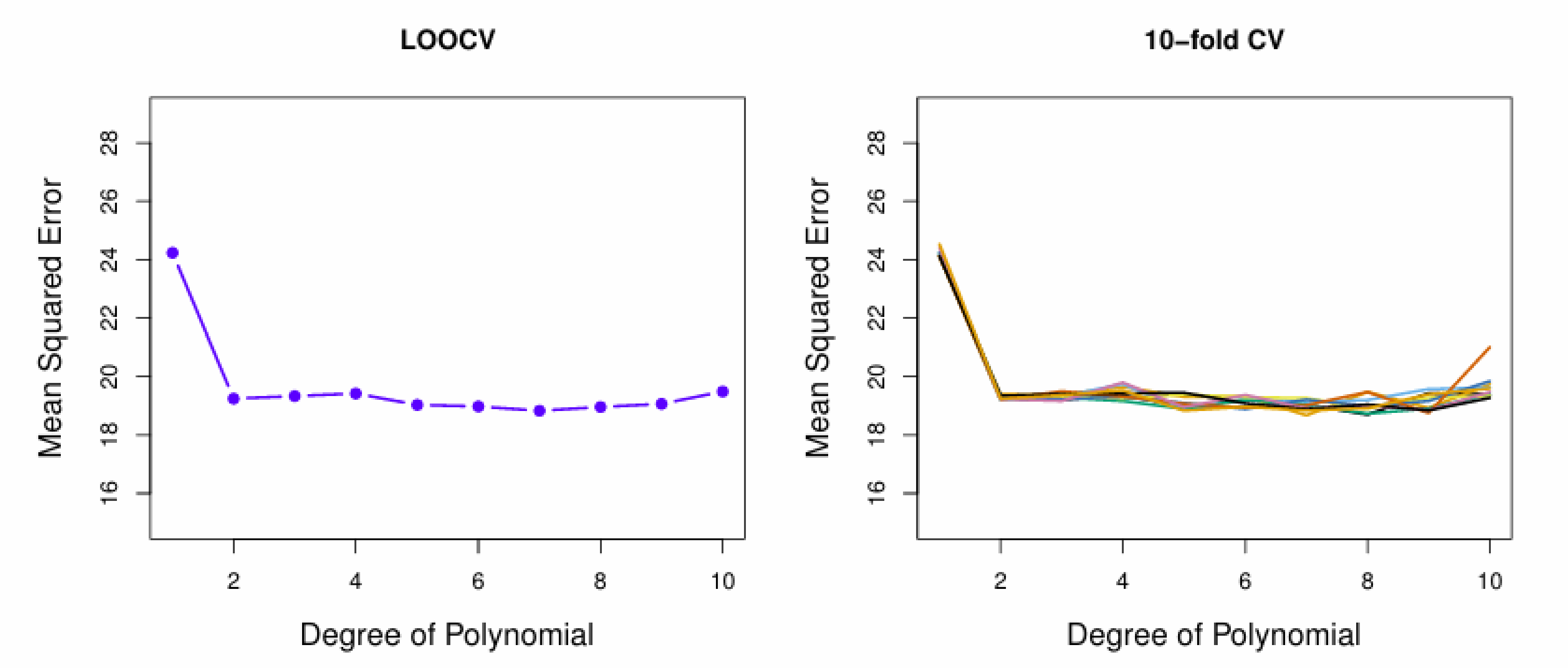
- Setting $K = n$ in k-fold cross-validation yields n-fold or leave-one-out cross validation **(LOOCV)**
  - Each data point is used once as the test set while the rest serve as the training set.
  - Provides an almost unbiased estimate of model performance but can be computationally expensive.



Leave-one-out Cross-validation

Example: Auto Data

- Setting $K = n$ in k-fold cross-validation yields n-fold or leave-one-out cross validation **(LOOCV)**

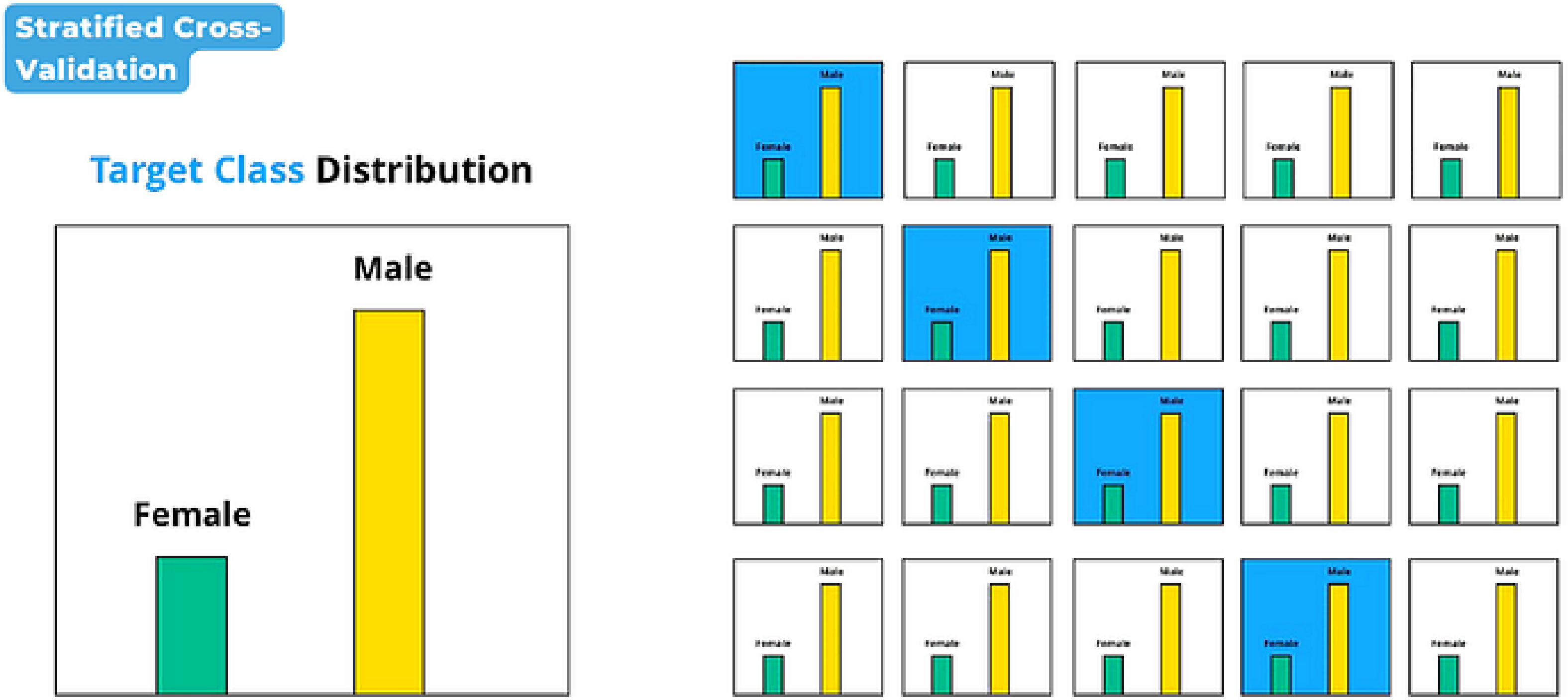


Stratified k-Fold Cross-Validation

Stratified k-Fold

Similar to k-Fold but ensures that each fold has approximately the same distribution of target classes as the entire dataset.

- Useful for imbalanced datasets.

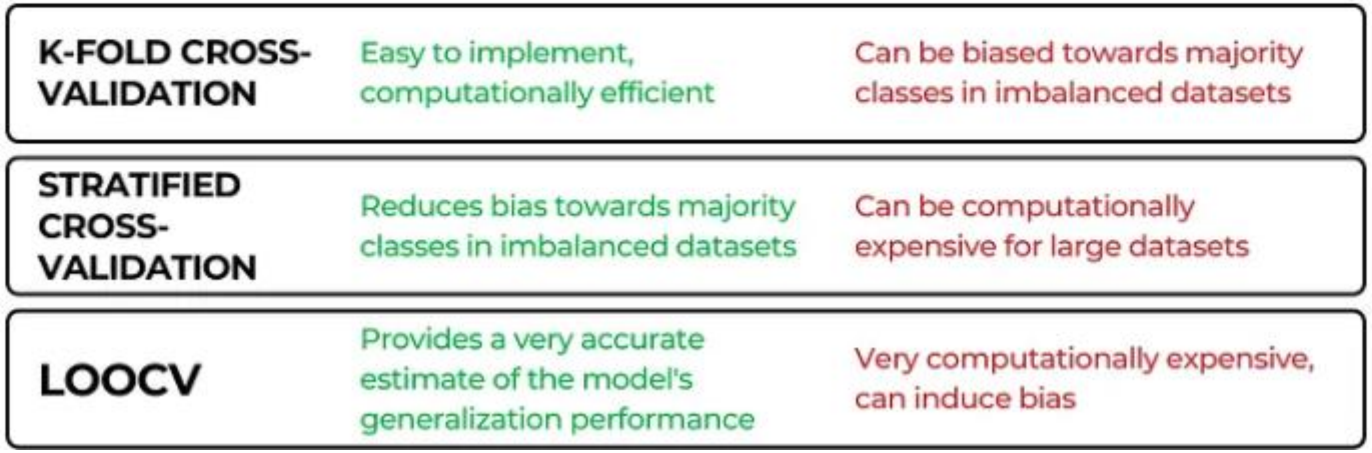


K-fold Cross-validation

Common choices of K: $K = 5$ or $K = 10$

- Advantage over LOOCV:
  1. Computationally lighter, especially for complex models with large data.
  2. Offers a good balance between bias and variance in model performance estimates.

  Advantage over validation set approach:
  1. Less variability resulting from the data-split, thanks to the averaging.



K-FOLD CROSS-VALIDATION	Easy to implement, computationally efficient	Can be biased towards majority classes in imbalanced datasets
STRATIFIED CROSS-VALIDATION	Reduces bias towards majority classes in imbalanced datasets	Can be computationally expensive for large datasets
LOOCV	Provides a very accurate estimate of the model's generalization performance	Very computationally expensive, can induce bias

The Bootstrap

The Bootstrap

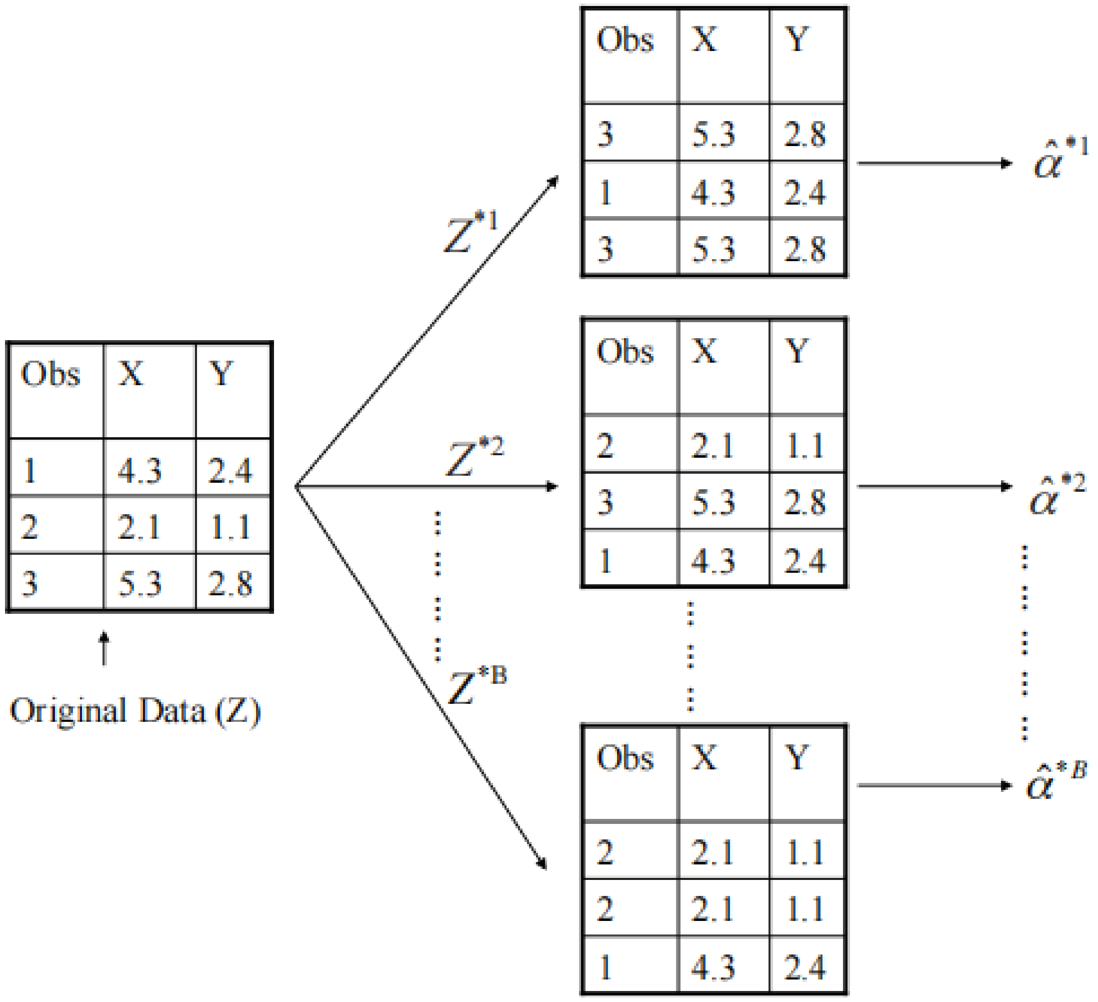
- The **bootstrap** is a flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method.
- It involves repeatedly drawing samples from the dataset with replacement and estimating model performance on these samples. It provides a way to assess the uncertainty in the performance metrics.

The Bootstrap

The details

1. **Generate Bootstrap Samples:**
   - Draw samples from the original dataset (with replacement) to create a bootstrap sample.
   - This process is repeated B times to create B bootstrap samples.
2. **Train and Evaluate the Model:**
   - Train the model on each bootstrap sample and evaluate it on the out-of-bag (OOB) data, which consists of data points not included in the bootstrap sample.
   - The OOB error estimate is used to gauge model performance.
3. **Aggregate Results:**
   - Calculate the performance metrics for each bootstrap sample.
   - Average the results to get an overall performance estimate.

The Bootstrap



Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

Original Data (Z)

Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

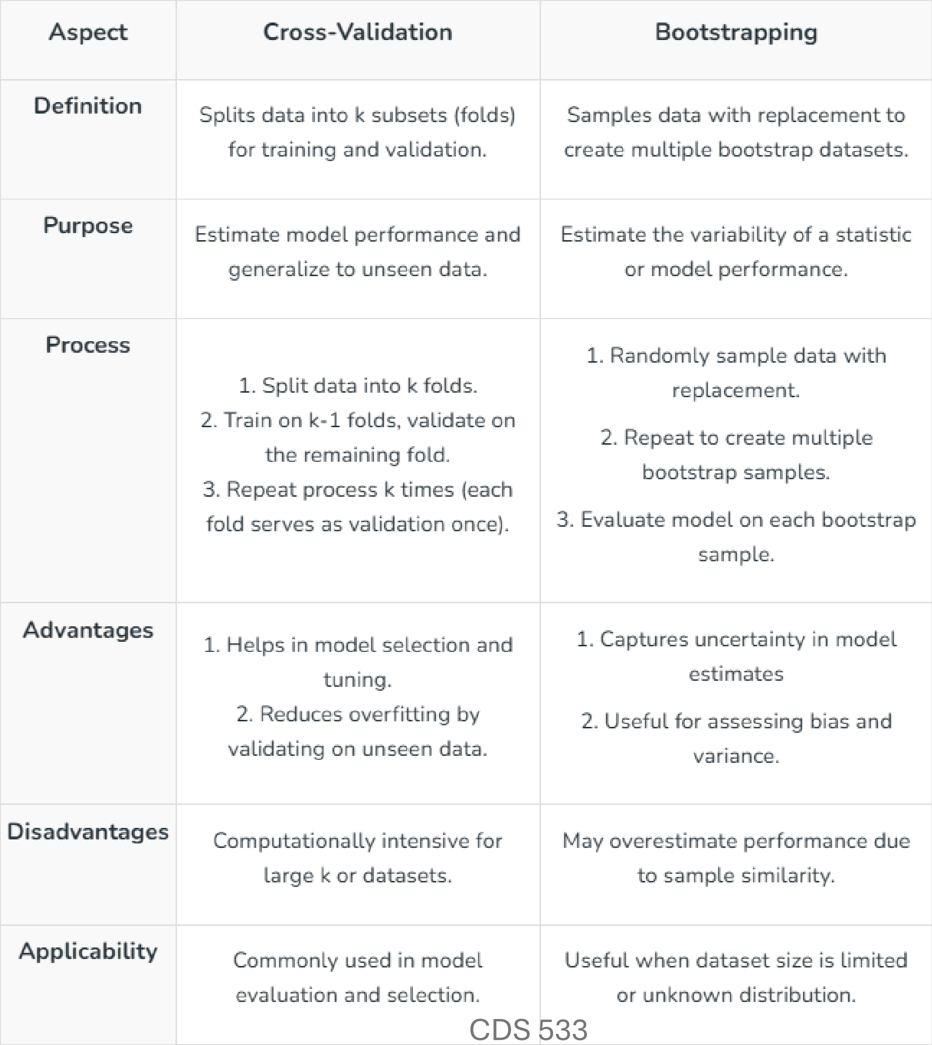
Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

A graphical illustration of the bootstrap approach on a small sample containing n = 3 observations. Each bootstrap data set contains n observations, sampled with replacement from the original data set. Each bootstrap data set is used to obtain an estimate of α.

Key Differences

Key Differences Between Cross-Validation and Bootstrapping



Aspect	Cross-Validation	Bootstrapping
Definition	Splits data into k subsets (folds) for training and validation.	Samples data with replacement to create multiple bootstrap datasets.
Purpose	Estimate model performance and generalize to unseen data.	Estimate the variability of a statistic or model performance.
Process	1. Split data into k folds. 2. Train on k-1 folds, validate on the remaining fold. 3. Repeat process k times (each fold serves as validation once).	1. Randomly sample data with replacement. 2. Repeat to create multiple bootstrap samples. 3. Evaluate model on each bootstrap sample.
Advantages	1. Helps in model selection and tuning. 2. Reduces overfitting by validating on unseen data.	1. Captures uncertainty in model estimates 2. Useful for assessing bias and variance.
Disadvantages	Computationally intensive for large k or datasets.	May overestimate performance due to sample similarity.
Applicability	Commonly used in model evaluation and selection.	Useful when dataset size is limited or unknown distribution.

Selection of Resampling Methods

When to Use Cross-Validation

- **Model Comparison:** When comparing multiple models or algorithms.
- **Hyperparameter Tuning:** When tuning hyperparameters to find the best model configuration.
- **Balanced Datasets:** Works well when the dataset is balanced and large enough to be split into meaningful folds.

When to Use Bootstrap

- **Small Datasets:** More effective for small datasets where splitting into multiple folds might not be feasible.
- **Variance Estimation:** When an estimate of the variability of the model performance is needed.
- **Uncertain Data:** When the dataset has significant noise or uncertainty.

Lab Time!