# Statistics and Estimation for Computer Science

İstanbul Teknik Üniversitesi

Mustafa Kamasak, PhD

Version: 2022.3.15

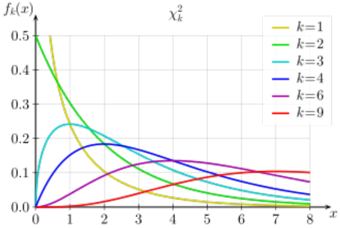
## Analysis of Categorical Data

## $\chi^2$ Statistics

- Chi-square test is used to analyze the goodness-of-fit of observed categorical data (such as gender, profession etc.) to its expected value (due to a known model, assumption etc.).
- $\chi^2$ statistics is used to determine the goodness-of-fit:

$$\chi^2 = \sum \frac{(\text{observed value} - \text{expected value})^2}{\text{expected value}}$$

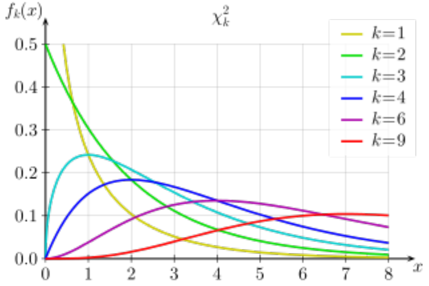
- $\chi^2$ statistics (rv.) has $\chi^2$ distribution. A critical value can be determined via computation or table-lookup.



## $\chi^2$ Statistics

- If $\chi^2$ value for an observation is lower than the critical value, it means the observed data is consistent with the expected model/assumption. Data fits model well.
- Very small expected values in the denominator may inflate the $\chi^2$ statistics.
- Rule of thumb: $\chi^2$ can be used whenever the expected value for all cases (cells) are at least 5. Otherwise, cells should be combined.

## $\chi^2$ Distribution



- The shape changes with dof (k in the figure)
- In t-distribution dof changes variation, dof does not change location (always centered at 0)
- In $\chi^2$ distribution location and variation changes with dof.

## Fair Die Example

- In reality, you might never know if a die is fair or not.
- It should be decided upon observations.
- A die is fair if $P(k) = 1/6$ for $k \in \{1, ..., 6\}$
- Assume two dice are thrown 600 times each

  Die 1:

| outcome $(k)$ | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| observation $(O_k)$ | 102 | 99 | 105 | 96 | 104 | 94 |

  Die 2:

| outcome $(k)$ | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| observation $(O_k)$ | 82 | 110 | 105 | 120 | 102 | 81 |

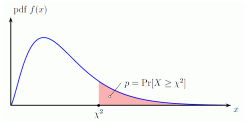
- Is die 1 fair?
- Is die 2 fair?

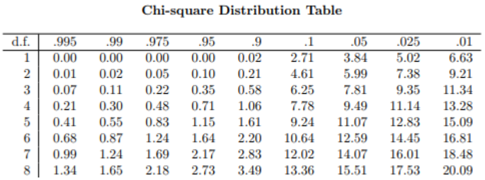
## Fair Die Example

- In a fair die, expected number of observations for each outcome (cell) is 100.
- For die 1:
$$\chi_1^2 = \sum_{i=1} \frac{(O_i - 100)^2}{100} = 0.98$$
- For die 2:
$$\chi_2^2 = \sum_{i=1} \frac{(O_i - 100)^2}{100} = 12.14$$
- dof $=$ # of cells - 1. Minus one is due to the fixed number of cells.
- dof $=$ 6 - 1 $=$ 5.

## Fair Die Example – Table Lookup

Table of upper-tail area (p-value) for $\chi^2$ distribution:





**Chi-square Distribution Table**

| d.f. | .995 | .99 | .975 | .95 | .9 | .1 | .05 | .025 | .01 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 2.71 | 3.84 | 5.02 | 6.63 |
| 2 | 0.01 | 0.02 | 0.05 | 0.10 | 0.21 | 4.61 | 5.99 | 7.38 | 9.21 |
| 3 | 0.07 | 0.11 | 0.22 | 0.35 | 0.58 | 6.25 | 7.81 | 9.35 | 11.34 |
| 4 | 0.21 | 0.30 | 0.48 | 0.71 | 1.06 | 7.78 | 9.49 | 11.14 | 13.28 |
| 5 | 0.41 | 0.55 | 0.83 | 1.15 | 1.61 | 9.24 | 11.07 | 12.83 | 15.09 |
| 6 | 0.68 | 0.87 | 1.24 | 1.64 | 2.20 | 10.64 | 12.59 | 14.45 | 16.81 |
| 7 | 0.99 | 1.24 | 1.69 | 2.17 | 2.83 | 12.02 | 14.07 | 16.01 | 18.48 |
| 8 | 1.34 | 1.65 | 2.18 | 2.73 | 3.49 | 13.36 | 15.51 | 17.53 | 20.09 |

- For die 1: $\chi_1^2 = 0.98$ and $0.975 < p_1 < 0.95$
- For die 2: $\chi_2^2 = 12.14$ and $0.025 < p_2 < 0.05$
- Critical value $\chi_c^2 = 11.07$ for dof=5 and $\alpha = 0.05$
  - As $\chi_1^2 < \chi_c^2$ (or as $p_1 > \alpha$), the observed data is not sufficient to claim that the die 1 is not fair.
  - As $\chi_2^2 > \chi_c^2$ (or as $p_2 < \alpha$), there is evidence that the die 2 is not fair.

## Fair Die Example – Python Implementation

```python
from scipy.stats import chisquare

# observations
dice1 = [102, 99, 105, 96, 104, 94]
dice2 = [82, 110, 105, 120, 102, 81]
# expected values
exp = [100, 100, 100, 100, 100, 100]

chi2,p = chisquare(dice1, exp)
print('Dice 1: chi2 stat: %.2f p: %f\n'%(chi2,p))
chi2,p = chisquare(dice2, exp)
print('Dice 2: chi2 stat: %.2f p: %f\n'%(chi2,p))
```

Output

```
Dice 1: chi2 stat: 0.98 p: 0.964163
Dice 2: chi2 stat: 12.14 p: 0.032919
```

## Fair Dice Example – Small sample size

- With 18 trials, test if Dice 3 is fair:

| outcome ($k$) | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| observation ($O_k$) | 4 | 0 | 4 | 4 | 2 | 4 |
| expected | 3 | 3 | 3 | 3 | 3 | 3 |

- $\chi^2 = 4.67$
- Rule of thumb: $\chi^2$ can be used whenever the expected value for all cases (cells) are at least 5. Otherwise, cells should be combined.

| outcome ($k$) | 1 - 2 | 3 - 4 | 5 - 6 |
|---|---|---|---|
| observation ($O_k$) | 4 | 8 | 6 |
| expected | 6 | 6 | 6 |

- $\chi^2 = 1.33$

## Mendel–Fisher Controversy

- On left: Gregor Mendel (1822–1884): Founder of genetics
- On right: Sir Ronald Fisher (1890–1962): Founder of statistics

Check out "A Statistical Model to Explain the Mendel–Fisher Controversy", Ana M. Pires and J. A. Branco,

`https://arxiv.org/pdf/1104.2975v1.pdf`

## Mendel–Fisher Controversy

- On 1865, Mendel published a paper about his experiments on garden peas.
- He breed pea wih pure yellow and green seeds. Then he breed them.
- In the first generation, all were yellow (as it is the dominant gene)
- In the second generation, out of 8023 peas:

| Color | Green | Yellow |
|---|---|---|
| # of obs. | 2001 | 6022 |
| expected | 2005.75 | 6017,25 |

## Mendel–Fisher Controversy

| Color | Green | Yellow |
|---|---|---|
| # of obs. | 2001 | 6022 |
| expected | 2005.75 | 6017,25 |

- $\chi^2 = 0.01$ – Way too good fit to theory
- In 1936, Ronald Fisher published a statistical analysis of Mendel's experiment. He computed the p-value of the Mendel's observations as 0.00004.
- He concluded that Mendel's data is manipulated.
  *"The data of most, if not all, of the experiments have been falsified so as to agree closely with Mendel's expectations."*
  *"Although no explanation can be expected to be satisfactory, it remains a possibility among others that Mendel was deceived by some assistant who knew too well what was expected."*

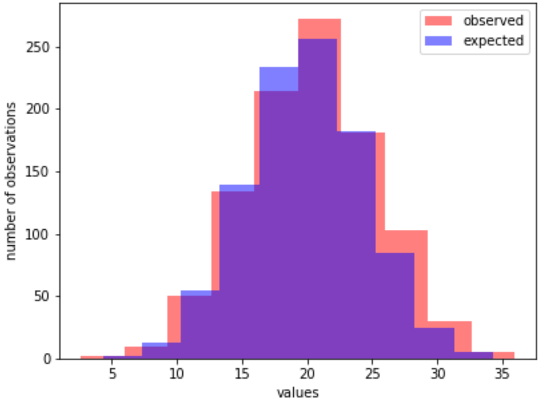
## Check if Data Comes from a Certain Distribution

- It is possible to check if data comes from a certain distribution
  - Continuous values are made discrete
  - Find number of data points within each bin
  - Find expected value of observations within each bin

$$Expected = N(F_X(upper) - F_X(lower))$$

    where $F_X$ is the cdf of the distribution.
  - Compute $\chi^2$ and find critical value $\chi_c^2$ from the loop-up table with significance level of $\alpha$.
  - If $\chi^2 > \chi_c^2$, reject distribution.

## Check for Normality



- Data come from Normal distribution $\sim \mathcal{N}(20, 25)$?

## Check for Normality – Python Implementation

```
from scipy.stats import chisquare, norm, t
import matplotlib.pyplot as plt

# sample size
N = 1000
mu = 20
std = 5

# generate sample from N(20,25)
x = std*norm.rvs(size=N)+mu

h = plt.hist(x,alpha=0.5,color='r')
bins = h[1]
vals = h[0]

# find expected values
expected = []
for i in range(len(bins)-1):
    ulimit = (bins[i+1]-mu)/std
    llimit = (bins[i]-mu)/std
    ev = N*(norm.cdf(ulimit)-norm.cdf(llimit))
    expected.append(ev)

# perform chi2 test
chi2,p=chisquare(vals,expected)
print(chi2,p)
```

## Check for Normality – Python Implementation

| Range | observed | expected |
|---|---|---|
| 3.99-7.19 | 4 | 4.5 |
| 7.19-10.39 | 14 | 22.1 |
| 10.39-13.60 | 72 | 72.8 |
| 13.60-16.80 | 164 | 160.9 |
| 16.80-20.00 | 257 | 239.1 |
| 20.00-23.20 | 252 | 239.1 |
| 23.20-26.41 | 148 | 160.8 |
| 26.41-29.61 | 62 | 72.7 |
| 29.61-32.81 | 24 | 22.1 |
| 32.81-36.02 | 3 | 4.5 |

- $\chi^2 = 8.42$ and $p = 0.493$
- No enough evidence to reject Normal distribution.

## Check for Normality – Python Implementation

```
from scipy.stats import chisquare, norm, t
import matplotlib.pyplot as plt

# sample size
N = 1000
mu = 20
std = 5

# generate sample from t with dof=18, mu=20, std=5
# x = std*norm.rvs(size=N)+mu
x = std*t.rvs(df=18,size=N)+mu

h = plt.hist(x,alpha=0.5,color='r')
bins = h[1]
vals = h[0]

# find expected values
expected = []
for i in range(len(bins)-1):
    ulimit = (bins[i+1]-mu)/std
    llimit = (bins[i]-mu)/std
    ev = N*(norm.cdf(ulimit)-norm.cdf(llimit))
    expected.append(ev)

# perform chi2 test
chi2,p=chisquare(vals,expected)
print(chi2,p)
```

## Check for Normality – Python Implementation

| Range | observed | expected |
|---|---|---|
| 4.81-7.85 | 16 | 6.4 |
| 7.85-10.89 | 27 | 26.7 |
| 10.89-13.94 | 88 | 78.3 |
| 13.94-16.98 | 146 | 160.1 |
| 16.98-20.02 | 220 | 228.8 |
| 20.02-23.06 | 241 | 228.3 |
| 23.06-26.10 | 143 | 159.1 |
| 26.10-29.14 | 79 | 77.4 |
| 29.14-32.19 | 27 | 26.3 |
| 32.19-35.23 | 13 | 6.2 |

- $\chi^2 = 27.08$ and $p = 0.00136$
- Reject Normal distribution for $\alpha = 0.05$.

## Independence of Bivariate Categorical Data

- $\chi^2$ statistics can also be used to investigate independence of bivariate categorical data.
- Assume we want to investigate the relation between *smoking during pregnancy* and *birth weight*.
- Observation are recorded as yes/no for smoking during pregnancy, underweight/normal/overweight for birth weights.
- Data looks like:
  {(yes, normal), (yes, underweight), (no, normal), ...}
- From this data, two-way frequency table is formed as follows:

| Observed | Underweight | Normal | Overweight | Total |
|---|---|---|---|---|
| Smoking | 55 | 44 | 8 | 107 |
| Non-smoking | 35 | 62 | 12 | 109 |
| Total | 90 | 106 | 20 | 216 |

## Independence of Bivariate Categorical Data

| Observed | Underweight | Normal | Overweight | **Total** |
|---|---|---|---|---|
| Smoking | 55 | 44 | 8 | 107 |
| Non-smoking | 35 | 62 | 12 | 109 |
| **Total** | 90 | 106 | 20 | 216 |

- This table is called *contingency table* [1].
- At the last row/column sum of the corresponding row/column is written.
- Grand total is written to the last cell.
- Hypothesis are:
  - $H_0$: data at rows and columns are independent from each other
  - $H_1$: data at rows and columns are **NOT** independent from each other
- $\chi^2$ dof is (number of rows-1)$\times$(number of columns-1)
- For this example, dof$=$(2-1)$\times$(3-1)$=$2

[1] contingency: a future event or circumstance which is possible but cannot be predicted with certainty.

## Independence of Bivariate Categorical Data

| Observed | Underweight | Normal | Overweight | **Total** |
|---|---|---|---|---|
| Smoking | 55 | 44 | 8 | 107 |
| Non-smoking | 35 | 62 | 12 | 109 |
| **Total** | 90 | 106 | 20 | 216 |

- If $H_0$ is correct, how should 90 underweight births distributed into smoking and non-smoking moms?
  - $90 \times 107/216$ should go to smoking cell,
  - $90 \times 109/216$ should go to non-smoking cell,
- If $H_0$ is correct, how should 107 smoking moms distributed into underweight/normal/overweight cells?
  - $107 \times 90/216$ should go to underweight cell,
  - $107 \times 106/216$ should go to normal cell,
  - $107 \times 20/216$ should go to overweight cell,
- The expected value for cell at row i and column j is:

$$E_{ij} = \frac{(\text{row total i}) \times (\text{column total j})}{\text{grand total}}$$

## Independence of Bivariate Categorical Data

| **Exp./Obs.** | Underweight | Normal | Overweight |
|---|---|---|---|
| Smoking | 45.58/55 | 52.51/44 | 9.91/8 |
| Non-smoking | 45.42/35 | 53.49/62 | 10.09/12 |

- Note that expected values in all cells are $\geq 5$.

```
from scipy.stats import chi2_contingency,
import matplotlib.pyplot as plt
import numpy as np

expected = np.array([ [45.58, 52.51, 9.91], [45.42, 53.49, 10.09] ])
observed = np.array([ [55, 44, 8], [35, 62, 12] ])

r = chi2_contingency(observed,expected)
print('chi2 stat: %.2f p-value: %f\n'%(r[0],r[1]))
```

- $\chi^2 = 8.28$ and p-value=0.0159.
- Hence $H_0$ should be rejected for significance level of $\alpha =$0.05.

## Yates Correction

- Consider 2×2 contingency matrix, the test statistics

$$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i}$$

- $O_i \sim Binomial(N, p)$
- Binomial distr is approximated with normal [2]
- Remember continuity correction when a discrete dist is approximated using cont. distr.
- For 2×2 contingency matrix, Yates correction is performed as follows [3]

$$\chi^2 = \sum_i \frac{(|O_i - E_i| - 0.5)^2}{E_i}$$

[2] Recall lower limit of 5 for each cell

[3] Yates, F (1934). "Contingency table involving small numbers and the chi-sq test". Supplement to the Journal of the Royal Statistical Society 1(2): 217–235. JSTOR

## Yates Correction

- Yates's correction prevents overestimation of statistical significance (underestimation of p-value) for small sample size
- Use Yate's correction when at least one cell of the table has an expected count smaller than 5
- Yates's correction is criticized for overcorrection [4]

[4] Larntz, K. (1978). Small sample comparisons of exact levels for chi-square goodness of fit statistics. Journal of the American Statistical Association, 73, 253-263.

## Regression

## Regression Analysis

- Regression[5] analysis analyzes relation between two or more variables

$$r(X) = E(Y|X = x) = \mu_{Y|X}$$

  where
  $X$ is called regressor, input, feature, predictor etc.
  $Y$ is called response, output, prediction
  $r(X)$ is called regression function
- Regression is used for
  - Analyze the relation between variables
  - To form a model $\implies$ predict a respond corresponding to an input

[5]term introduced by Francis Galton when he realized sons of extremely long/short fathers regress to mean.

## Simple Linear Regression Model

- Consider a linear relation between regressor and response

$$Y = \beta_0 + \beta_1 X + \epsilon$$

  where
  $\beta_0, \ \beta_1 \in \mathbb{R}$ are constant coefficients,
  $\epsilon$ is noise (error) with $\mu_\epsilon = 0$ and $\sigma_\epsilon^2$ is not a function of X.
- Typically Gaussian noise is assumed $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. Why? Multiple sources of error are superimposed $\implies$ CLT
- The regression function is

$$\begin{aligned} r(x) &= E(Y|X = x) \\ &= E(\beta_0 + \beta_1 x + \epsilon) \\ &= \beta_0 + \beta_1 x \end{aligned}$$

## Simple Linear Regression Model

- Coefficients $\beta_0$ and $\beta_1$ are the parameters of simple regression model
- They are estimated from a sample of observations

$$[(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)]$$

- Least squares estimation: Minimize the squared error

$$\mathcal{C}(x, y) = \sum_i \epsilon_i^2 = \sum_i (y_i - \beta_0 - \beta_1 x_i)^2$$

and

$$\hat{\beta}_0, \hat{\beta}_1 = \arg\min_{\beta_0, \beta_1} \mathcal{C}(x, y)$$

## Coefficients of Linear Regression

- Take partial derivative wrt to coefficients and make them zero

$$\frac{\partial}{\partial \beta_0}\mathcal{C}\Big|_{\beta_0=\hat{\beta}_0} = 0$$
$$\frac{\partial}{\partial \beta_1}\mathcal{C}\Big|_{\beta_0=\hat{\beta}_1} = 0$$

- Estimated coefficients are

$$\hat{\beta}_1 = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sum_i (x_i - \overline{x})^2} = \frac{S_{xy}}{S_{xx}}$$
$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}$$

where

$$S_{xx} = \sum_i x_i^2 - N\overline{x}^2$$
$$S_{xy} = \sum_i x_i y_i - N\overline{x}\ \overline{y}$$

## Linear Model Output and Residual

- Using parameters of model $\hat{\beta}_0$ and $\hat{\beta}_1$ model outputs can be computed for each regressor

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

- Difference between real response ($y_i$) and corresponding model output ($\hat{y}_i$) is called residual

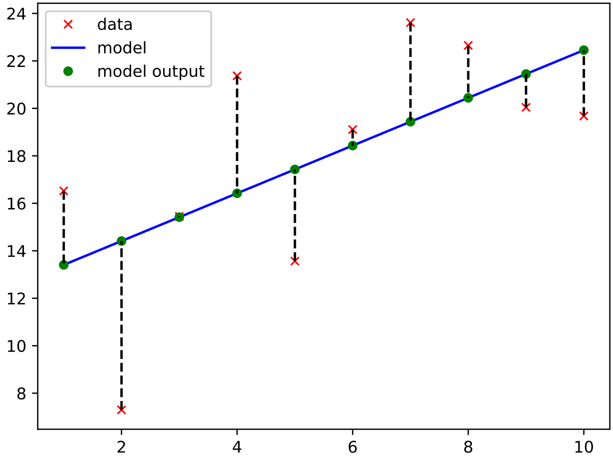
$$e_i = y_i - \hat{y}_i = y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i$$

- Do not get confused
  - $\epsilon$ is error that is inherent in the system
  - $e$ is error in the model/fit
- Residuals are the error in the model/fit. Causes are
  - Error in the system $\epsilon$
  - Error in the model
  - Other errors in computations/code etc.

# Linear Model Output and Residual

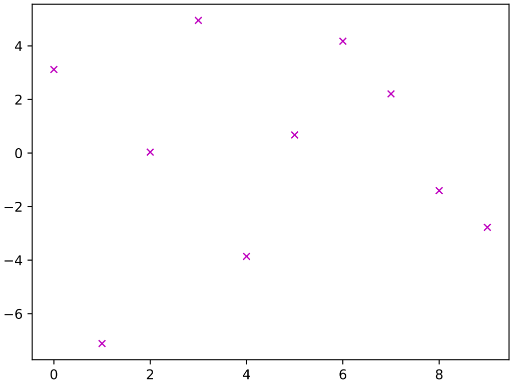
```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from scipy.stats import norm

beta0 = 3
beta1 = 2
x = np.arange(1,11)
y = beta0 + beta1*x + norm.rvs(loc=0, scale=6, size=x.size)

model = LinearRegression().fit(x.reshape((-1,1)), y)
beta0_est = model.intercept_
beta1_est = model.coef_
print('estimated beta0:', beta0_est )
print('estimated beta1:', beta1_est )
y_est = beta0_est + beta1_est*x

plt.plot(x,y,'rx',markersize=5)
plt.plot(x,y_est,'b-')
plt.plot(x,y_est,'go',markersize=5)
#plt.plot(x,beta0+beta1*x,'g--')
# draw residuals
for i in range(x.size):
    plt.plot([x[i],x[i]],[y[i],y_est[i]],'k--')

plt.legend(['data','model','model output'])
plt.show()
```

## Linear Model Output and Residual



## Linear Model Output and Residual

- Analysis of residuals is very useful to
  - Investigate the validity of assumptions (linearity, zero mean error, error variance not related to input)
  - Investigate regression performance (model adequacy)



## Coefficient of Determination

Few definitions:

- Amount of variation in response

$$S_{YY} = \sum_i (y_i - \overline{y})^2$$

- Amount of inherent variance of response

$$SS_E = \sum_i (y_i - \hat{y}_i)^2 = \sum_i e_i^2$$

  This is the variance that cannot be described by the model

- Amount of variance described by the model

$$SS_R = \sum_i (\hat{y}_i - \overline{y})^2 = S_{YY} - SS_E$$

- Coefficient of determination

$$R^2 = \frac{SS_R}{S_{YY}} = \frac{S_{YY} - SS_E}{S_{YY}}$$

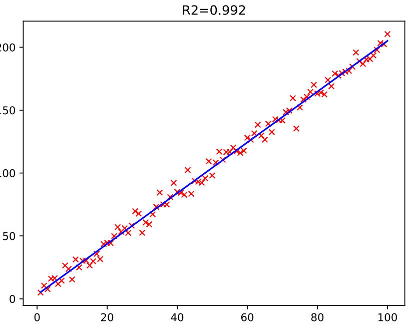
## Coefficient of Determination

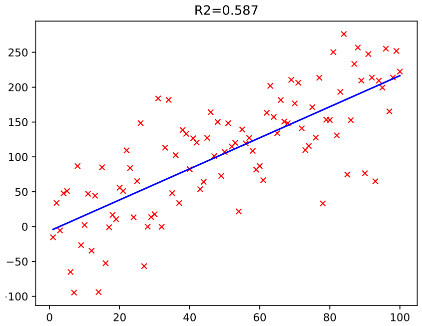
- Coefficient of determination

$$R^2 = \frac{SS_R}{S_{YY}} = \frac{S_{YY} - SS_E}{S_{YY}}$$

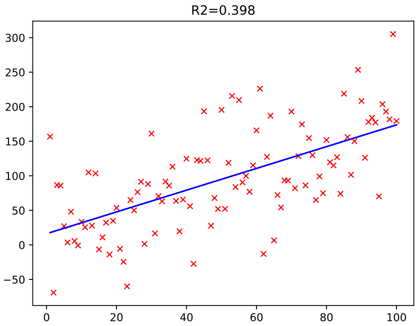
- $0 \leq R^2 \leq 1$
- $R^2$ represent the proportion of variance in response (output) that is explained by model
- $R^2$ is a measure of model adequacy
  - $R^2 = 0$ model is totally inadequate to describe output
  - $R^2 = 1$ model perfectly describes output

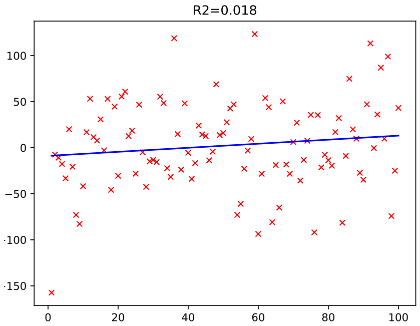
# Coefficient of Determination





# Coefficient of Determination





## Correlation Coefficient

- What is the correlation between $y$ and $x$?

$$r = \frac{Cov(x,y)}{\sigma_x \sigma_y} = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2 \sum_i (y_i - \overline{y})^2}}$$

- What is the relation between $\rho$ and $R^2$?

$$r = \frac{S_{XY}}{\sqrt{S_{XX} S_{YY}}}$$

$$SS_E = \frac{S_{XX} S_{YY} - S_{XY}^2}{S_{XX}}$$

$$|r| = \sqrt{1 - \frac{SS_E}{S_{YY}}} = \sqrt{R^2}$$

## Coefficient Coefficient

- Relation between $r$ and $R^2$

$$R^2 = r^2$$

- Relation between $r$ and $\hat{\beta}_1$

$$\hat{\beta}_1 = \frac{S_{XY}}{S_{XX}} \implies r = \hat{\beta}_1 \sqrt{\frac{S_{XX}}{S_{YY}}}$$

- Relation between $R^2$ and $\hat{\beta}_1$

$$R^2 = \hat{\beta}_1^2 \frac{S_{XX}}{S_{YY}}$$

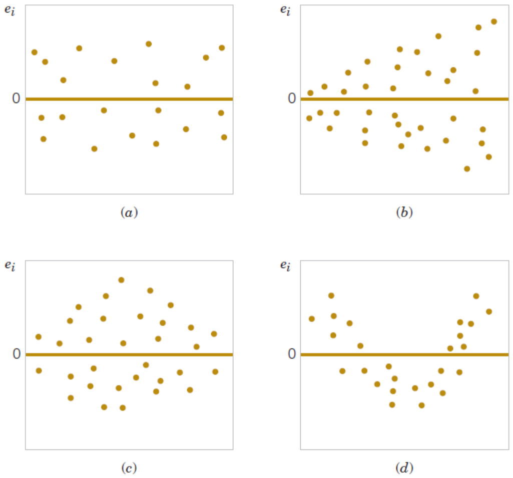
## Residual Analysis

- Residual values can have any variance
- Normalize to $\mathcal{N}(0, 1)$
- Standardized residuals

$$e'_i = \frac{y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)}{\sqrt{SS_E/(N-2)}}$$

- It is typically expected that $e'_i \in [-2, 2]$ as $P(|z| < 1.96) = 0.05$
- If there are few $e'_i \notin [-2, 2]$, corresponding outputs can be assumed to be outliers
- If there are too many $e'_i \notin [-2, 2]$, then the model is not adequate or assumptions are incorrect
- There should be no pattern in $e'_i$ and it should look random $\rightarrow$ Need quantitative method for measuring randomness

## Patterns in Residuals



(a) Ideal case (b) Increasing noise variance
(c) Varying noise variance (d) Model inadequacy

## Confidence Intervals for Model Parameters

- $\hat{\beta}_0$ and $\hat{\beta}_1$ are point estimates for model parameters $\beta_0$ and $\beta_1$
- What is $1-\alpha$ confidence intervals for estimations for model parameters?
- Variance of residuals is estimated as

$$\hat{\sigma}_e^2 = \frac{1}{N-2}\sum_i e_i^2 = \frac{SS_E}{N-2}$$

  $N-2$ dof is due to estimations of $\hat{\beta}_0$ and $\hat{\beta}_1$
- Recall

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}}$$
$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1\overline{x} = \overline{y} - \frac{S_{xy}}{S_{xx}}\overline{x}$$

## Confidence Intervals for Model Parameters

- If $\epsilon \sim \mathcal{N}$, least squares estimation is equal to MLE
- Recall asymtotical normality of MLE

$$\frac{\hat{\beta}_1 - \beta_1}{\mathrm{se}(\hat{\beta}_1)} \sim \mathcal{N}(0,1) \qquad \frac{\hat{\beta}_0 - \beta_0}{\mathrm{se}(\hat{\beta}_0)} \sim \mathcal{N}(0,1)$$

- Estimations are unbiased

$$E(\hat{\beta}_1) = \beta_1$$
$$E(\hat{\beta}_0) = \beta_0$$

- Standard errors (se) of estimations are

$$Var(\hat{\beta}_1) = \frac{\hat{\sigma}_e^2}{S_{XX}} \implies \mathrm{s.e.}(\hat{\beta}_1) = \sqrt{\frac{\hat{\sigma}_e^2}{S_{XX}}}$$

$$Var(\hat{\beta}_0) = \hat{\sigma}_e^2\Big(\frac{1}{N} + \frac{\overline{X}^2}{S_{XX}}\Big) \implies \mathrm{s.e.}(\hat{\beta}_0) = \sqrt{\hat{\sigma}_e^2\Big(\frac{1}{N} + \frac{\overline{X}^2}{S_{XX}}\Big)}$$

## Confidence Intervals for Model Parameters

- Recall asymtotic normality of MLE

$$\frac{\hat{\beta}_1 - \beta_1}{\mathrm{se}(\hat{\beta}_1)} \sim \mathcal{N}(0,1) \qquad \frac{\hat{\beta}_0 - \beta_0}{\mathrm{se}(\hat{\beta}_0)} \sim \mathcal{N}(0,1)$$

- Hence $1-\alpha$ confidence interval is
  - if $N \geq 20$

$$\left[\hat{\beta}_i - z_{\alpha/2}\mathrm{se}(\hat{\beta}_i), \quad \hat{\beta}_i + z_{\alpha/2}\mathrm{se}(\hat{\beta}_i)\right]$$

  - if $N < 20$

$$\left[\hat{\beta}_i - t_{N-2,\alpha/2}\mathrm{se}(\hat{\beta}_i), \quad \hat{\beta}_i + t_{N-2,\alpha/2}\mathrm{se}(\hat{\beta}_i)\right]$$

## Hypothesis Testing for Slope

- Consider following hypothesis:
  $\mathrm{H}_0 : \beta_1 = \beta_{1,0}$
  $\mathrm{H}_1 : \beta_1 \neq \beta_{1,0}$
- Recall, if $\sigma_e^2$ is known

$$\frac{\hat{\beta}_1 - \beta_{1,0}}{\sqrt{\sigma_e^2 / S_{XX}}} \sim \mathcal{N}(0,1)$$

- Use $\hat{\sigma}_e^2$ as estimate of $\sigma_e^2$
- Test statistics

$$T = \sqrt{\frac{(N-2)S_{XX}}{SS_E}}(\hat{\beta}_1 - \beta_{1,0})$$

- Reject $\mathrm{H}_0$ is $|t_{obs}| \geq t_{N-2,\alpha/2}$ for significance level of $\alpha$

## Hypothesis Testing for Null Slope

- An important hypothesis is $\beta_1 = 0$ which means there is no linear relation between input/output

$$Y = \beta_0 + \beta_1 X$$

- Hypothesis is then
  $\mathrm{H}_0 : \beta_1 = 0$
  $\mathrm{H}_1 : \beta_1 \neq 0$
- Recall, if $\sigma_e^2$ is known

$$\frac{\hat{\beta}_1}{\sqrt{\sigma_e^2 / S_{XX}}} \sim \mathcal{N}(0, 1)$$

- Use $\hat{\sigma}_e^2$ as estimate of $\sigma_e^2$
- Test statistics

$$T = \sqrt{\frac{(N-2)S_{XX}}{SS_E}}(\hat{\beta}_1)$$

- Reject $\mathrm{H}_0$ is $|T| \geq t_{N-2,\alpha/2}$ for significance level of $\alpha$

## Logistic Regression

- When response is binary ie. $y_i \in \{0, 1\}$, logistic regression is used
- Why not just use linear regression?
  - Error $\epsilon$ is not normal distributed

$$\epsilon_i = 1 - (\beta_0 - \beta_1 x_i) \text{ if } y_i = 1$$
$$\epsilon_i = -(\beta_0 - \beta_1 x_i) \text{ if } y_i = 0$$

  - With linear regression, response may lie outside [0,1]
- Model should be nonlinear

## Logit Function

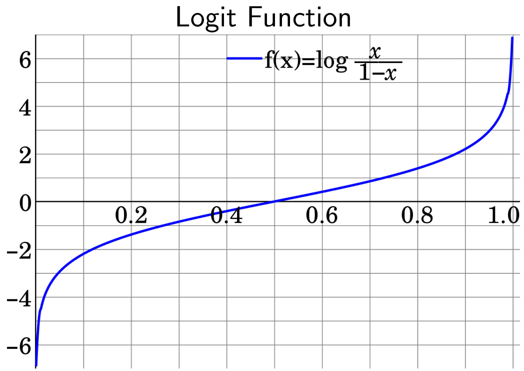
- Likelihood ratio is called odds

$$\text{odds} = \frac{E(y_i)}{1 - E(y_i)} = \frac{p}{1-p}$$

  where $E(y_i) = 1 \times P(y_i = 1) + 0 \times P(y_i = 0) = p$
- Logarithm of the odds maps a probability $p \in [0, 1]$ to $[-\infty,\ \infty]$
- **Log**istic un**it** – logit – function

$$\text{logit} = \log \frac{p}{1-p}$$

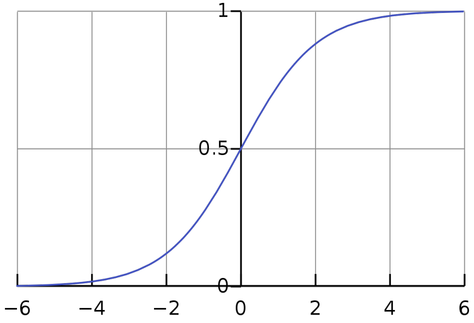
## Logit Function



- To map input $x_i \in [-\infty,\ \infty]$ to binary response $y_i \in \{0, 1\}$, inverse logit function is required

$$\text{logit}^{-1}(x) = \text{logistic}(x)$$

## Logistic Regression



- S-shaped Logistic/sigmoid function

$$
\begin{aligned}
P(y_i = 1) &= p \\
&= \text{logistic}(x_i) = \frac{1}{1 + \exp\{-(\beta_0 + \beta_1 x_i)\}}
\end{aligned}
$$

## Logistic Regression

- Logistic/sigmoid function

$$\begin{aligned} P(y_i = 1) &= p \\ &= \text{logistic}(x_i) = \frac{1}{1 + \exp\{-(\beta_0 + \beta_1 x_i)\}} \end{aligned}$$

- Remember

$$\begin{aligned} \text{logit} &= \log\left(\frac{p}{1-p}\right) \\ &= \log(\exp(\beta_0 + \beta_1 x_i)) \\ &= \beta_0 + \beta_1 x_i \end{aligned}$$

## Logistic Regression

- Model parameters are estimated by MLE
- No closed form expression: Use iterative methods
- Logistic regression can be used for binary classification

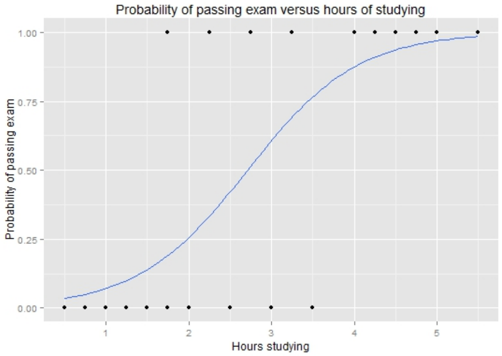
## Logistic Regression – Example

- A group of 20 students spends between 0 and 6 hours studying for an exam.[6]
- How does the number of hours spent studying affect the probability of the student passing the exam?
- Data

```
x=[
(0.50,0), (0.75,0), (1.00,0), (1.25,0), (1.50,0),
(1.75,0), (1.75,1),(2.00,0), (2.25,1) (2.50,0),
(2.75,1), (3.00,0), (3.25,1), (3.50,0), (4.00,1),
(4.25,1), (4.50,1), (4.75,1), (5.00,1), (5.50,1)
]
```

[6] Example from https://en.wikipedia.org/wiki/Logistic_regression

## Logistic Regression – Example



- Model parameters are $\hat{\beta}_0 = -4.0777$ and $\hat{\beta}_1 = 1.5046$
- Logistic function

$$y_i = \frac{1}{1 + \exp\{-(-4.0777 + 1.5046 x_i)\}}$$

## Logistic Regression – Example

- Logistic function

$$y_i = \frac{1}{1 + \exp\{-(-4.0777 + 1.5046 x_i)\}}$$

- Probability of passing exam with a 2-hour study is

$$y_i = \frac{1}{1 + \exp\{-(-4.0777 + 1.5046 \times 2)\}} = 0.26$$

- Probability of passing exam with a 4-hour study is

$$y_i = \frac{1}{1 + \exp\{-(-4.0777 + 1.5046 \times 4)\}} = 0.87$$

## Logistic Regression – Code

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LogisticRegression

# number of hours studied from exam
x = np.array([0.50, 0.75, 1.00, 1.25, 1.50, \
              1.75, 1.75, 2.00, 2.25, 2.50, \
              2.75, 3.00, 3.25, 3.50, 4.00, \
              4.25, 4.50, 4.75, 5.00, 5.50])
# pass(1)/fail(0)
y = np.array
    ([0,0,0,0,0,0,1,0,1,0,1,0,1,0,1,1,1,1,1,1])

model = LogisticRegression()
model.fit(x.reshape(-1,1),y)

print('beta0:', model.intercept_)
print('beta1:', model.coef_)
```

## Multiple Linear Regression

- Typically regressor is a vector

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k + \epsilon$$

- Assume *n* responses

$$\begin{aligned} y_1 &= \beta_0 + \beta_1 x_{1,1} + \cdots + \beta_k x_{1,k} + \epsilon_1 \\ \ldots \\ y_n &= \beta_0 + \beta_1 x_{n,1} + \cdots + \beta_k x_{n,k} + \epsilon_n \end{aligned}$$

- Use matrix notation

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\mathbf{Y} \in \mathbb{R}^n$, $\mathbf{X} \in \mathbb{R}^{n\times(k+1)}$, $\boldsymbol{\beta} \in \mathbb{R}^{(k+1)}$ and $\boldsymbol{\epsilon} \in \mathbb{R}^n$

## Multiple Linear Regression

- In matrix notation

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

- Parameter vector $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta} = [\beta_0, \beta_1, \cdots, \beta_k]^T \in \mathbb{R}^{(k+1)}$$

- Input matrix $\boldsymbol{X}$ is

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,k} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,k} \\ 1 & \cdots & \cdots & \cdots & \cdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,k} \end{bmatrix} \in \mathbb{R}^{n\times(k+1)}$$

- Error vector $\boldsymbol{\epsilon}$ is

$$\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \cdots, \epsilon_n]^T \in \mathbb{R}^n$$

# Multiple Linear Regression - Least Squares Estimate

- Least squares estimate is

$$\hat{\boldsymbol{\beta}} = \arg\min \mathcal{C}(\mathbf{X}, \mathbf{y})$$

where

$$\mathcal{C}(\mathbf{X}, \mathbf{y}) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)$$

- Take partial derivative

$$\frac{\partial \mathcal{C}(\mathbf{X}, \mathbf{Y})}{\partial \boldsymbol{\beta}} = 0$$

- Least squares estimate is

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

- Prediction becomes

$$\hat{\boldsymbol{y}} = \mathbf{X}\hat{\beta}$$

- Residuals are

$$\mathbf{e} = \mathbf{y} - \hat{\boldsymbol{y}}$$

## Multiple Linear Regression

- The variance estimate is

$$\sigma_e^2 = \frac{SS_E}{N-k-1}$$

  where $SS_E = e^T e$
- For $k+1$ model parameters, residual degrees of freedom is $N-k-1$
- Least square estimation bias and precision are

$$\begin{aligned} E(\hat{\beta}) &= \beta \\ \text{Cov}(\hat{\beta}) &= \sigma_e^2(\mathbf{X^T X})^{-1} \\ &= \sigma_e^2 \mathbf{C} \end{aligned}$$

- Standard error for $\hat{\beta}_j$ is

$$\text{se}(\hat{\beta}_j) = \sqrt{\sigma_e^2 \mathbf{C}_{jj}}$$

## Polynomial Regression

- A specific form of multiple linear regression is called polynomial regression
- Preferred with curvilinear relation between regressor and response
- Regressors of a polynomial model are generated from a single input as

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_k x^k$$

- This is a polynomial model of degree $k$
- Polynomial regression with degree $k = 1$ is equivalent to simple linear regression

## Better Fitting with More Parameters

- As the number of inputs (parameters) in a model increases, a better fit can be obtained
- Let model 1 has $K$ parameters and model 2 has $M$ parameters where $K < M$
- For the same response, let $\hat{y}_1$ be the output of model 1 and $\hat{y}_2$ be the output of model 2
- Sum of squared residuals for model 2 will be smaller than model 1

$$\|y - \hat{y}_1\|_2^2 > \|y - \hat{y}_2\|_2^2$$

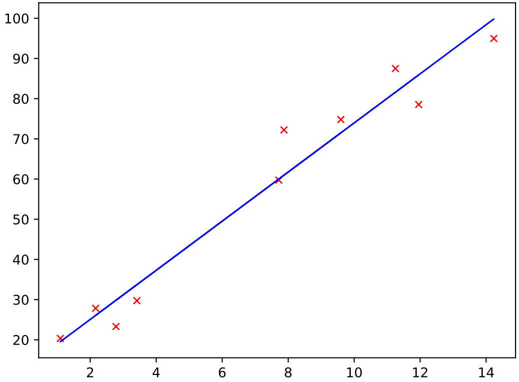
- Coefficient of determination of model 2 will be larger

$$R_1^2 > R_2^2$$

- With $N$ data points, a model with $N - 1$ inputs (hence $N$ parameters) will perfectly fit ie. $SS_E = 0$ and $R^2 = 1$

## Overfitting Example

- Assume an approximately linear relation between hours of study before final exam and final exam grade
- For 10 students following observations are made



- A linear model fits well with $R^2 = 0.96$

## Overfit Example

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.linear_model import LinearRegression

np.random.seed(1)

x = np.ndarray(shape=(10,10),dtype=float)
x[:,0] = np.arange(3, 13) + np.random.normal(0, 3, size=10)

for i in range(1,9):
    x[:,i] = np.random.normal(-3+i, 5, size=10)

y = x[:,0]*6 + 13 + np.random.normal(0, 10, size=10)

model = LinearRegression()
model.fit(x[:,0].reshape(-1,1), y)
hat_y = model.predict(x[:,0].reshape(-1,1))
r2 = r2_score(y, hat_y)
mse = mean_squared_error(y,hat_y)
print('degree: %d mse:%f  r square: %f'%(1,mse,r2))


for i in range(1,9):
    model.fit(x[:,0:i+1], y)
    hat_y = model.predict(x[:,0:i+1])
    r2 = r2_score(y, hat_y)
    mse = mean_squared_error(y,hat_y)
    print('degree: %d mse:%f  r square: %f'%(i+1,mse,r2))
```

## Overfit Example

output

```
degree: 1 mse:31.316861  r square: 0.958046
degree: 2 mse:23.079755  r square: 0.969081
degree: 3 mse:22.268891  r square: 0.970167
degree: 4 mse:22.237426  r square: 0.970210
degree: 5 mse:20.539075  r square: 0.972485
degree: 6 mse:11.652108  r square: 0.984390
degree: 7 mse:11.581630  r square: 0.984485
degree: 8 mse:8.566025  r square: 0.988525
degree: 9 mse:0.000000  r square: 1.000000
```

## Overfitting Example

- Even with totally random input, it is possible to increase $R^2$
- $R^2$ is a good metric for comparing models with same complexity
- $R^2$ is not a good metric for comparing models with different complexity $\implies$ Use adjusted $R^2$
- Few data points and high model complexity may cause overfitting $\implies$ Dropping coefficients in deep learning models
- Model memorizes data and cannot generalize
- How to decide model complexity?

## Adjusted $R^2$

- Use a revised $R^2$ metric that penalizes model complexity

$$\begin{aligned} R^2_{adjusted} &= 1 - \frac{SS_E/(N-K-1)}{S_{YY}/(N-1)} \\ &= 1 - \frac{(1-R^2)(N-1)}{N-K-1} \end{aligned}$$

  where $K$ is the number of regressors (parameters of the model excluding $\beta_0$)
- If $R^2$ does not increase substantially by added complexity, adjusted $R^2$ will decrease

## Overfit Example Revizited

output

```
degree: 1 mse:31.316861  r2: 0.958046 adj r2: 0.952802
degree: 2 mse:23.079755  r2: 0.969081 adj r2: 0.960247
degree: 3 mse:22.268891  r2: 0.970167 adj r2: 0.955251
degree: 4 mse:22.237426  r2: 0.970210 adj r2: 0.946377
degree: 5 mse:20.539075  r2: 0.972485 adj r2: 0.938091
degree: 6 mse:11.652108  r2: 0.984390 adj r2: 0.953171
degree: 7 mse:11.581630  r2: 0.984485 adj r2: 0.930181
degree: 8 mse:8.566025  r2: 0.988525 adj r2: 0.896721
degree: 9 mse:0.000000  r2: 1.000000 adj r2: nan
```

## Underfitting

- Model is too simple (ie. less than required number of parameters) $\rightarrow$ Underfitting
- Patterns can be seen at residuals

## Model Selection

- How to select a Model?
- Simpler models are preferred as they have improved generalization and lower risk of overfitting
  - Too simple $\implies$ underfitting (model inadequacy)
  - Too complicated $\implies$ overfitting
- If there are $K$ inputs, there are $2^K$ possible models
- Two step approach for model selection
  - Define a cost function $\mathcal{C}(M_k)$ where $\mathcal{C}$ is the cost function and $M_k$ is the model with $k$ regressors
  - Cost should have term for $SS_E$ and model complexity
  - Search through models to minimize cost
- Possible cost functions are
  - Mallow's C$_p$ statistic
  - AIC: Akaike information criteria
  - BIC: Bayesian information criteria
  - Negative $R^2_{adj}$ can be used as cost function

## Mallow's Cp Statistic

- Mallow's $C_p$ statistics is defined as follow

$$\mathcal{C}(M_k) = \frac{1}{N}(SS_E + 2k\hat{\sigma}_\epsilon^2)$$

or

$$\mathcal{C}(M_k) = \frac{SS_E}{\hat{\sigma}_\epsilon^2} - N + 2k$$

where $k$ is the number of regressors and $\hat{\sigma}_\epsilon^2$ is the estimate of noise variance

- Two definitions will give different cost values but they will lead to same model

## AIC: Akaike Information Criteria

- AIC considers model fits with MLE
- AIC cost function is

$$AIC(M_k) = N\log(\frac{SS_E}{N}) + 2k$$

- If noise has normal distribution ie $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ then Mallow $C_p$ and AIC will lead to the same model as MLE will be equivalent to LSE

## BIC: Bayesian Information Criteria

- BIC has more severe penalty for model complexity
- BIC cost function is

$$BIC(M_k) = N \log(\frac{SS_E}{N}) + k \log(N)$$

## Noise Variance Estimation

- Noise variance estimate is used for mentioned cost functions
- Noise variance estimate

$$\hat{\sigma}_\epsilon^2 = \frac{SS_E}{N - k - 1}$$

where $SS_E$ is obtained by all using all regressors (lowest possible $SS_E$ with most complicated model)

## Model Input Selection

- Which regressor combinations should be used for $SS_E$ for cost computation?
- For example to compute cost of $M_2$ which two regressor should be used?
- Exhaustive search $C(K, k)$ possible combinations and maximize $R^2$
- Use stepwise regressor selection
  - forward selection
    - Start with no regressors
    - Add one regressor that increases the cost the least
    - Iterate
  - backward elimination
    - Start with all regressors
    - Eliminate one regressor that decreases the cost the most
    - Iterate
- Stepwise regression selection does not guarantee best regressor combination

## Advantages and Disadvantages of Linear Regression

- Advantages:
  - Results are highly interpretable
  - Low computation cost
- Disadvantages:
  - Performance is typically worse compared to
    - Support vector regression
    - Random forest regression
    - ...[7]
  - Other issues (described at next slide)

[7] Taught in machine learning course.

## Problems with Linear Regression

- Data is typically nonlinear
- Error variances typically change with data (homoscedasticity/heteroscacity) $\rightarrow$ Use weighted least squares (WLS)
- Very small changes in data can result in very different models $\rightarrow$ Use regularization on model parameters
- Be cautious about interpretation of model. Models show associations, not causality.

## Other Types of Least Squares

- Ordinary least squares (OLS)
- Nonlinear least squares (NLS)
- Total least squares (TLS)
- Weighted least squares (WLS)
- Iterated reweighted least squares (IRLS)
- Others [8]

[8] Check out `https://en.wikipedia.org/wiki/Regression_analysis`

## Regression with Regularization

- Ordinary: No regression

$$\hat{\boldsymbol{\beta}} = \arg\min \|\mathbf{y} - \mathbf{X}\beta\|^2$$

where

$$\|\boldsymbol{M}\|^2 = \boldsymbol{M}^T\boldsymbol{M}$$

- Ridge regression ($L_2$ regression)

$$\hat{\boldsymbol{\beta}}_r = \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda\|\boldsymbol{\beta}\|^2$$

- Lasso regression ($L_1$ regression)

$$\hat{\boldsymbol{\beta}}_l = \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda\|\boldsymbol{\beta}\|$$

where

$$\|\boldsymbol{M}\| = \sum_i |M_i|$$

## Regression with Regularization

- Elasticnet regression (linear mixture of $L_1$ and $L_2$)

$$\hat{\boldsymbol{\beta}}_l = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda(\alpha\|\boldsymbol{\beta}\| + (1-\alpha)\|\boldsymbol{\beta}\|^2)$$