dm24s1

Topic 07 : Regression1

Part 01 : Regression - Overview

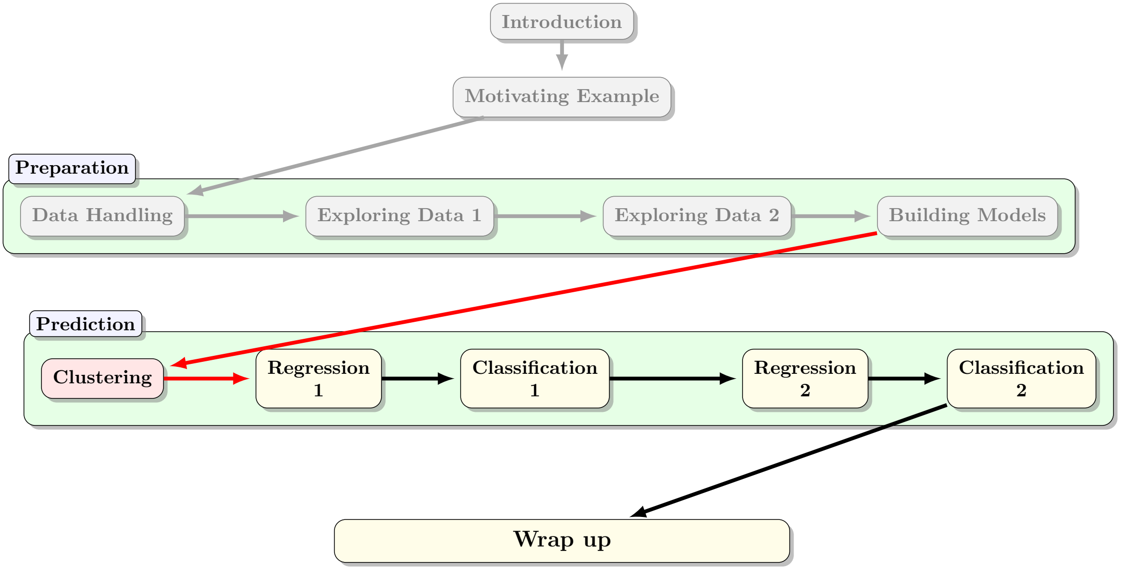
Dr Bernard Butler
Department of Computing and Mathematics, WIT.
(bernard.butler@setu.ie)

Autumn Semester, 2024

Outline

- Regression as a means of minimising sum of the squared errors
- Regression assumptions - what they mean, how they can be used for validation and model building
- Case studies from Diamond sales

Data Mining (Week 7)



Regression - Overview — Summary

This Week’s Aim

This week’s aim is to give an overview of the linear regression: fitting linear models to data, to predict a numeric value.

This Week's Aim

This week's aim is to give an overview of the linear regression: fitting linear models to data, to predict a numeric value.

- High level view of regression: where it came from, what it attempts to do.

This Week's Aim

This week's aim is to give an overview of the linear regression: fitting linear models to data, to predict a numeric value.

- High level view of regression: where it came from, what it attempts to do.
- Examine some extensions to the simplest case of linear regression.

This Week's Aim

This week's aim is to give an overview of the linear regression: fitting linear models to data, to predict a numeric value.

- High level view of regression: where it came from, what it attempts to do.
- Examine some extensions to the simplest case of linear regression.
- Consider how to check that the regression was successful, and make some improvements if necessary

This Week's Aim

This week's aim is to give an overview of the linear regression: fitting linear models to data, to predict a numeric value.

- High level view of regression: where it came from, what it attempts to do.
- Examine some extensions to the simplest case of linear regression.
- Consider how to check that the regression was successful, and make some improvements if necessary
- In this topic, we will cover the basics of Regression - we will return later in the module

This Week's Aim

This week's aim is to give an overview of the linear regression: fitting linear models to data, to predict a numeric value.

- High level view of regression: where it came from, what it attempts to do.
- Examine some extensions to the simplest case of linear regression.
- Consider how to check that the regression was successful, and make some improvements if necessary
- In this topic, we will cover the basics of Regression - we will return later in the module
- To provide context we will use the following datasets over this and Regression 2:

This Week's Aim

This week's aim is to give an overview of the linear regression: fitting linear models to data, to predict a numeric value.

- High level view of regression: where it came from, what it attempts to do.
- Examine some extensions to the simplest case of linear regression.
- Consider how to check that the regression was successful, and make some improvements if necessary
- In this topic, we will cover the basics of Regression - we will return later in the module
- To provide context we will use the following datasets over this and Regression 2:
  - Diamond dataset: predicting diamond prices given their weights
  - Generated data (various)

This Week's Aim

This week's aim is to give an overview of the linear regression: fitting linear models to data, to predict a numeric value.

- High level view of regression: where it came from, what it attempts to do.
- Examine some extensions to the simplest case of linear regression.
- Consider how to check that the regression was successful, and make some improvements if necessary
- In this topic, we will cover the basics of Regression - we will return later in the module
- To provide context we will use the following datasets over this and Regression 2:
  - Diamond dataset: predicting diamond prices given their weights
  - Generated data (various)
  - Advertising dataset: predicting widgets sold based on spending in different advertising channels
  - Credit dataset: predicting credit balance using income, status, etc.

Simple Linear Regression: Background

- Linear regression was discovered by Gauss and others around 1800. The “name” came later!
- With small data sets, calculations can be done by hand, but they are tedious and error-prone.
- The goal is simple: Given a training set of (x, y) data where y is assumed to have a linear relationship with x
  - Find the line that is the “best fit” to that data
  - Use the specification of that line to *predict* y for the test x values
- Note that the “linear relationship” of y upon x is just one of the underlying assumptions
- In practice, the data does not have an exact linear relationship, but it should be “close enough”—but what does that mean?

Review: Linear combinations (scalar product)

Definition 1 (Linear Combination of two vectors)

Given two vectors $\mathbf{a}$ and $\mathbf{b}$, each with n elements, the *linear combination* (c) of $\mathbf{a}$ and $\mathbf{b}$ is

$$c \equiv a_1 b_1 + a_2 b_2 + \ldots + a_n b_n = \sum_{i=1}^{n} a_i b_i \equiv |\mathbf{a}||\mathbf{b}| \cos(\mathbf{a}, \mathbf{b})$$

Remarks

Review: Linear combinations (scalar product)

Definition 1 (Linear Combination of two vectors)

Given two vectors **a** and **b**, each with n elements, the *linear combination* (c) of **a** and **b** is

$$c \equiv a_1 b_1 + a_2 b_2 + \ldots + a_n b_n = \sum_{i=1}^{n} a_i b_i \equiv |\mathbf{a}||\mathbf{b}| \cos(\mathbf{a}, \mathbf{b})$$

Remarks

- The linear combination of 2 vectors is a scalar, which can be seen as “mixing” two vectors.

Review: Linear combinations (scalar product)

Definition 1 (Linear Combination of two vectors)

Given two vectors **a** and **b**, each with n elements, the *linear combination* (c) of **a** and **b** is

$$c \equiv a_1 b_1 + a_2 b_2 + \ldots + a_n b_n = \sum_{i=1}^{n} a_i b_i \equiv |\mathbf{a}||\mathbf{b}| \cos(\mathbf{a}, \mathbf{b})$$

Remarks

- The linear combination of 2 vectors is a scalar, which can be seen as “mixing” two vectors.
- Matrix-vector multiplication $A\mathbf{x}$ can be seen as the linear combination of each row in the matrix A with the (column) vector $\mathbf{x}$.

Review: Linear combinations (scalar product)

Definition 1 (Linear Combination of two vectors)

Given two vectors $\mathbf{a}$ and $\mathbf{b}$, each with n elements, the *linear combination* (c) of $\mathbf{a}$ and $\mathbf{b}$ is

$$c \equiv a_1 b_1 + a_2 b_2 + \ldots + a_n b_n = \sum_{i=1}^{n} a_i b_i \equiv |\mathbf{a}||\mathbf{b}| \cos(\mathbf{a}, \mathbf{b})$$

Remarks

- The linear combination of 2 vectors is a scalar, which can be seen as “mixing” two vectors.
- Matrix-vector multiplication $A\mathbf{x}$ can be seen as the linear combination of each row in the matrix A with the (column) vector $\mathbf{x}$.
- Matrix-matrix multiplication AB can be seen as the linear combination of each row in the matrix A with each column in the matrix B.

Review: Linear combinations (scalar product)

Definition 1 (Linear Combination of two vectors)

Given two vectors $\mathbf{a}$ and $\mathbf{b}$, each with n elements, the *linear combination* (c) of $\mathbf{a}$ and $\mathbf{b}$ is

$$c \equiv a_1 b_1 + a_2 b_2 + \ldots + a_n b_n = \sum_{i=1}^{n} a_i b_i \equiv |\mathbf{a}||\mathbf{b}| \cos(\mathbf{a}, \mathbf{b})$$

Remarks

- The linear combination of 2 vectors is a scalar, which can be seen as “mixing” two vectors.
- Matrix-vector multiplication $A\mathbf{x}$ can be seen as the linear combination of each row in the matrix A with the (column) vector $\mathbf{x}$.
- Matrix-matrix multiplication AB can be seen as the linear combination of each row in the matrix A with each column in the matrix B.
- Two nonzero vectors $\mathbf{a}$ and $\mathbf{b}$ can have a scalar product that is zero if $\cos(\mathbf{a}, \mathbf{b}) = 0$, i.e., the $\mathbf{a}$ and $\mathbf{b}$ vectors are perpendicular to each other.

Review: Linear combinations (scalar product)

Definition 1 (Linear Combination of two vectors)

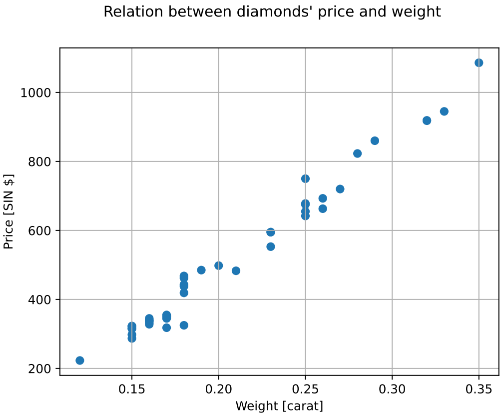
Given two vectors $\mathbf{a}$ and $\mathbf{b}$, each with n elements, the *linear combination* (c) of $\mathbf{a}$ and $\mathbf{b}$ is

$$c \equiv a_1 b_1 + a_2 b_2 + \ldots + a_n b_n = \sum_{i=1}^{n} a_i b_i \equiv |\mathbf{a}||\mathbf{b}| \cos(\mathbf{a}, \mathbf{b})$$

Remarks

- The linear combination of 2 vectors is a scalar, which can be seen as "mixing" two vectors.
- Matrix-vector multiplication $A\mathbf{x}$ can be seen as the linear combination of each row in the matrix A with the (column) vector $\mathbf{x}$.
- Matrix-matrix multiplication AB can be seen as the linear combination of each row in the matrix A with each column in the matrix B.
- Two nonzero vectors $\mathbf{a}$ and $\mathbf{b}$ can have a scalar product that is zero if $\cos(\mathbf{a}, \mathbf{b}) = 0$, i.e., the $\mathbf{a}$ and $\mathbf{b}$ vectors are perpendicular to each other.
- Linear combinations are used for prediction from linear (regression) models.

Motivating example: Diamond data



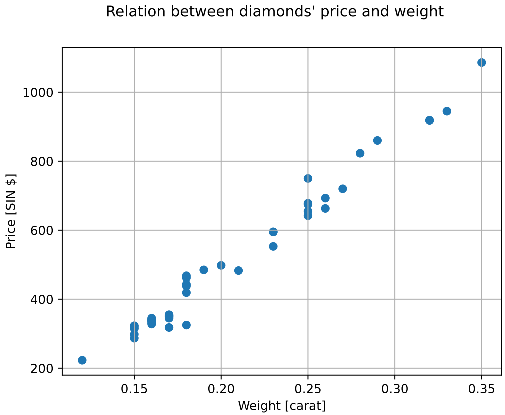
Diamond Prices by Weight

Motivating example: Diamond data

Diamond Prices by Weight

- Given the data on the left, can we use it to predict the price of a diamond that weighs 0.22 carat?

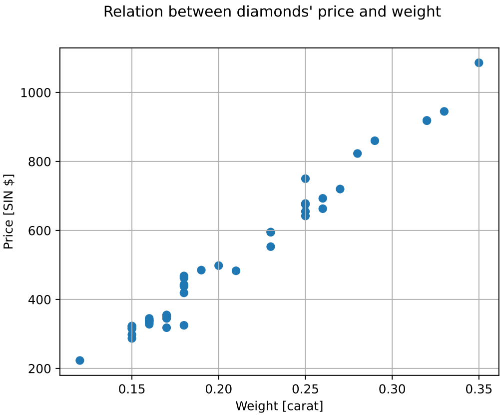
Motivating example: Diamond data



Diamond Prices by Weight

- Given the data on the left, can we use it to predict the price of a diamond that weighs 0.22 carat?
- NB - we have not seen a diamond with that weight before in the data

Motivating example: Diamond data

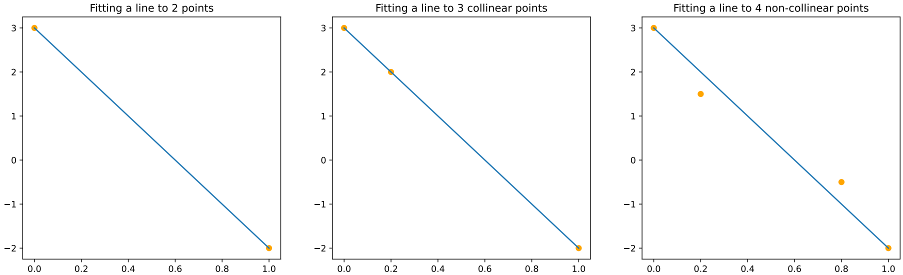


Diamond Prices by Weight

- Given the data on the left, can we use it to predict the price of a diamond that weighs 0.22 carat?
- NB - we have not seen a diamond with that weight before in the data
- Can you think of at least 4 other factors that might affect the price?

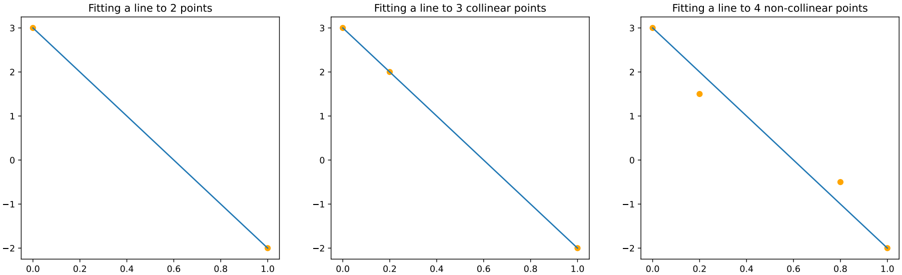
Simple Linear Regression: Geometric Intuition

- Given data $\{x_i, y_i\}$ where $i = 2, 3, \ldots, n$ and β_0, β_1 as the (unknown, but to be determined) *intercept* and *slope* of the regression line for this data.



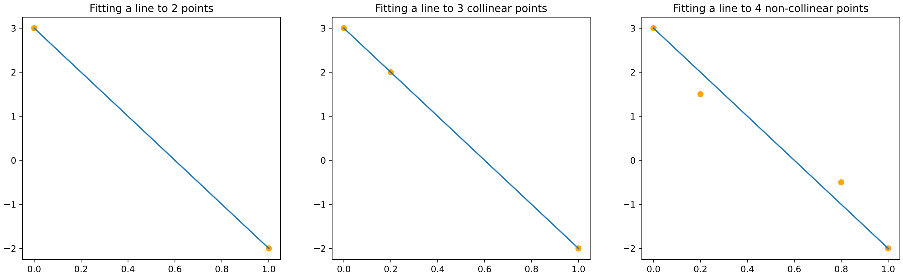
Simple Linear Regression: Geometric Intuition

- Given data $\{x_i, y_i\}$ where $i = 2, 3, \ldots, n$ and β_0, β_1 as the (unknown, but to be determined) *intercept* and *slope* of the regression line for this data.
- If $n = 1$, the problem is underdetermined: any line through the point will do - the solution is not unique.



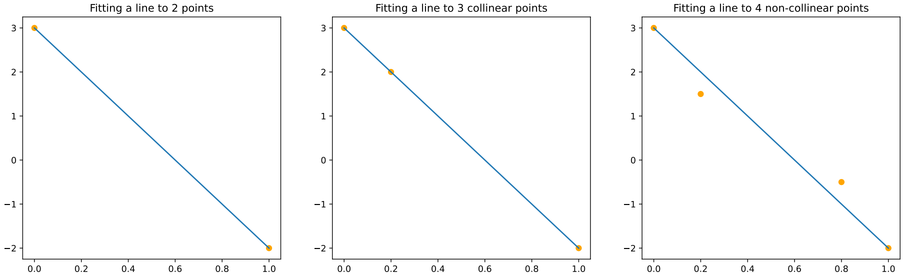
Simple Linear Regression: Geometric Intuition

- Given data $\{x_i, y_i\}$ where $i = 2, 3, \ldots, n$ and β_0, β_1 as the (unknown, but to be determined) *intercept* and *slope* of the regression line for this data.
- If $n = 1$, the problem is underdetermined: any line through the point will do - the solution is not unique.
- For $n = 2$ points with $x_2 \neq x_1$, this can be solved uniquely for β_0, β_1, using techniques you learnt for your Junior/Inter Cert.



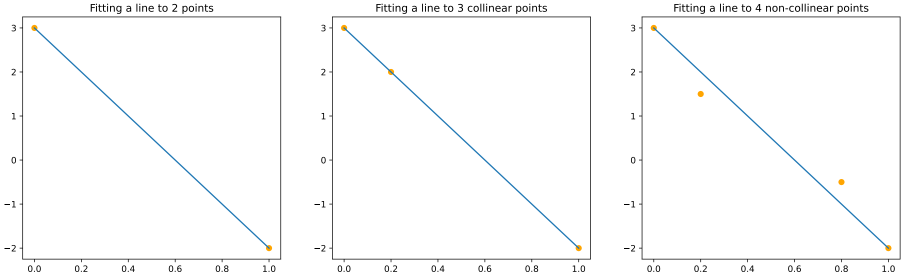
Simple Linear Regression: Geometric Intuition

- Given data $\{x_i, y_i\}$ where $i = 2, 3, \ldots, n$ and β_0, β_1 as the (unknown, but to be determined) *intercept* and *slope* of the regression line for this data.
- If $n = 1$, the problem is underdetermined: any line through the point will do - the solution is not unique.
- For $n = 2$ points with $x_2 \neq x_1$, this can be solved uniquely for β_0, β_1, using techniques you learnt for your Junior/Inter Cert.
- For $n > 2$ collinear points, just pick any two points and solve as before.



Simple Linear Regression: Geometric Intuition

- Given data $\{x_i, y_i\}$ where $i = 2, 3, \ldots, n$ and β_0, β_1 as the (unknown, but to be determined) *intercept* and *slope* of the regression line for this data.
- If $n = 1$, the problem is underdetermined: any line through the point will do - the solution is not unique.
- For $n = 2$ points with $x_2 \neq x_1$, this can be solved uniquely for β_0, β_1, using techniques you learnt for your Junior/Inter Cert.
- For $n > 2$ collinear points, just pick any two points and solve as before.
- Otherwise the problem is overdetermined so need a more general formulation to solve for β_0, β_1.



Simple Linear Regression: Formulation

Definition 2 (Matrix formulation)

- General equation is $y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \hat{y}_i + \epsilon_i$ (data = model + error), where $\hat{y}$ is the predicted y for these values of β_0, β_1.

Simple Linear Regression: Formulation

Definition 2 (Matrix formulation)

- General equation is $y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \hat{y}_i + \epsilon_i$ (data = model + error), where $\hat{y}$ is the predicted y for these values of β_0, β_1.
- Matrix form is $\mathbf{y} = X\beta$. Remember matrix-vector multiplication: inner product of i^{th} row of X times the vector β = $1 \times \beta_0 + x_i \times \beta_1 = \hat{y}_i$.

Simple Linear Regression: Formulation

Definition 2 (Matrix formulation)

- General equation is $y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \hat{y}_i + \epsilon_i$ (data = model + error), where $\hat{y}$ is the predicted y for these values of β_0, β_1.
- Matrix form is $\mathbf{y} = X\beta$. Remember matrix-vector multiplication: inner product of i^{th} row of X times the vector β = $1 \times \beta_0 + x_i \times \beta_1 = \hat{y}_i$.
- However, we don't know β yet, nor do we know $\hat{y}_i$, so we use y_i as an estimate of $\hat{y}_i$ and solve for all data in the training set.

Simple Linear Regression: Formulation

Definition 2 (Matrix formulation)

- General equation is $y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \hat{y}_i + \epsilon_i$ (data = model + error), where $\hat{y}$ is the predicted y for these values of β_0, β_1.
- Matrix form is $\mathbf{y} = X\beta$. Remember matrix-vector multiplication: inner product of i^{th} row of X times the vector β = $1 \times \beta_0 + x_i \times \beta_1 = \hat{y}_i$.
- However, we don't know β yet, nor do we know $\hat{y}_i$, so we use y_i as an estimate of $\hat{y}_i$ and solve for all data in the training set.
- So: our task is to solve the *overdetermined* (number of rows exceeds the number of columns) system of equations $\mathbf{y} = X\beta$ for β

Simple Linear Regression: Formulation

Definition 2 (Matrix formulation)

- General equation is $y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \hat{y}_i + \epsilon_i$ (data = model + error), where $\hat{y}$ is the predicted y for these values of β_0, β_1.
- Matrix form is $\mathbf{y} = X\beta$. Remember matrix-vector multiplication: inner product of i^{th} row of X times the vector β = $1 \times \beta_0 + x_i \times \beta_1 = \hat{y}_i$.
- However, we don't know β yet, nor do we know $\hat{y}_i$, so we use y_i as an estimate of $\hat{y}_i$ and solve for all data in the training set.
- So: our task is to solve the *overdetermined* (number of rows exceeds the number of columns) system of equations $\mathbf{y} = X\beta$ for β
- Our geometric intuition is that the errors should be "balanced": no benefit to changing intercept (sliding up or down) or slope (tilting the line).

Simple Linear Regression: Normal Equations

$$
\begin{aligned}
\mathbf{y} &\approx X\beta \\
\mathbf{y} &= X\beta + \epsilon \\
X^{\mathsf{T}}\mathbf{y} &= X^{\mathsf{T}}X\beta + X^{\mathsf{T}}\epsilon \\
X^{\mathsf{T}}\mathbf{y} &= X^{\mathsf{T}}X\beta
\end{aligned}
$$

Simple Linear Regression: Normal Equations

$$\begin{aligned} \mathbf{y} &\approx X\beta \\ \mathbf{y} &= X\beta + \epsilon \\ X^{\mathsf{T}}\mathbf{y} &= X^{\mathsf{T}}X\beta + X^{\mathsf{T}}\epsilon \\ X^{\mathsf{T}}\mathbf{y} &= X^{\mathsf{T}}X\beta \end{aligned}$$

because $X^{\mathsf{T}}\epsilon \equiv 0$ implies the fitted line gives balanced errors and so is 'best". Swapping sides, we have

$$\begin{aligned} (X^{\mathsf{T}}X)\beta &= X^{\mathsf{T}}\mathbf{y} \\ (X^{\mathsf{T}}X)^{-1}(X^{\mathsf{T}}X)\beta &= (X^{\mathsf{T}}X)^{-1}X^{\mathsf{T}}\mathbf{y} \end{aligned}$$

Simple Linear Regression: Normal Equations

$$
\begin{aligned}
\mathbf{y} &\approx X\beta \\
\mathbf{y} &= X\beta + \epsilon \\
X^{\mathsf{T}}\mathbf{y} &= X^{\mathsf{T}}X\beta + X^{\mathsf{T}}\epsilon \\
X^{\mathsf{T}}\mathbf{y} &= X^{\mathsf{T}}X\beta
\end{aligned}
$$

because $X^{\mathsf{T}}\epsilon \equiv 0$ implies the fitted line gives balanced errors and so is 'best''. Swapping sides, we have

$$
\begin{aligned}
(X^{\mathsf{T}}X)\beta &= X^{\mathsf{T}}\mathbf{y} \\
(X^{\mathsf{T}}X)^{-1}(X^{\mathsf{T}}X)\beta &= (X^{\mathsf{T}}X)^{-1}X^{\mathsf{T}}\mathbf{y}
\end{aligned}
$$

which is equivalent to the *Normal equations*

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

Simple Linear Regression: Normal Equations

$$
\begin{aligned}
\mathbf{y} &\approx X\beta \\
\mathbf{y} &= X\beta + \epsilon \\
X^{\mathsf{T}}\mathbf{y} &= X^{\mathsf{T}}X\beta + X^{\mathsf{T}}\epsilon \\
X^{\mathsf{T}}\mathbf{y} &= X^{\mathsf{T}}X\beta
\end{aligned}
$$

because $X^{\mathsf{T}}\epsilon \equiv 0$ implies the fitted line gives balanced errors and so is ‘best”. Swapping sides, we have

$$
\begin{aligned}
(X^{\mathsf{T}}X)\beta &= X^{\mathsf{T}}\mathbf{y} \\
(X^{\mathsf{T}}X)^{-1}(X^{\mathsf{T}}X)\beta &= (X^{\mathsf{T}}X)^{-1}X^{\mathsf{T}}\mathbf{y}
\end{aligned}
$$

which is equivalent to the *Normal equations*

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

Note that everything on the right is a set of operations on the data.

For more info, and an alternative construction of the Normal equations, see `https://goo.gl/TbLru3`.

Simple Linear Regression: Implementation

When implemented in software, the Normal equations are not used directly: faster and more numerically accurate algorithms are used instead, but the results are equivalent in exact arithmetic (remember: digital computers perform finite-precision arithmetic and so cannot be exact).

Simple Linear Regression: Implementation

When implemented in software, the Normal equations are not used directly: faster and more numerically accurate algorithms are used instead, but the results are equivalent in exact arithmetic (remember: digital computers perform finite-precision arithmetic and so cannot be exact).

One option is to use statsmodels: consistent with R (separate model specification), excellent diagnostics as standard

Another option is to use sklearn: consistent with other sklearn algorithms, more controls

Simple Linear Regression: Implementation

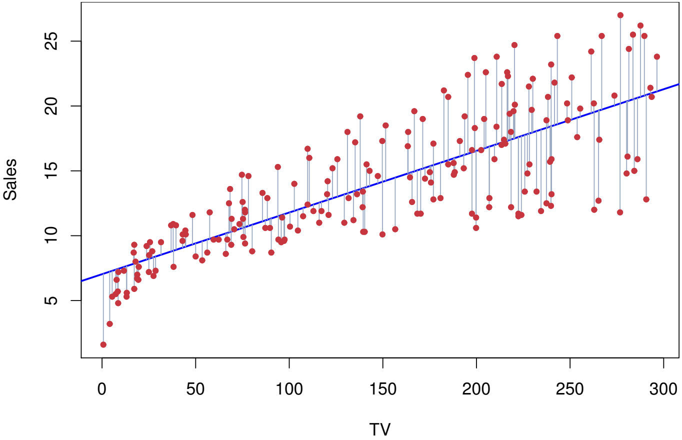
When implemented in software, the Normal equations are not used directly: faster and more numerically accurate algorithms are used instead, but the results are equivalent in exact arithmetic (remember: digital computers perform finite-precision arithmetic and so cannot be exact).

One option is to use statsmodels: consistent with R (separate model specification), excellent diagnostics as standard

Another option is to use sklearn: consistent with other sklearn algorithms, more controls

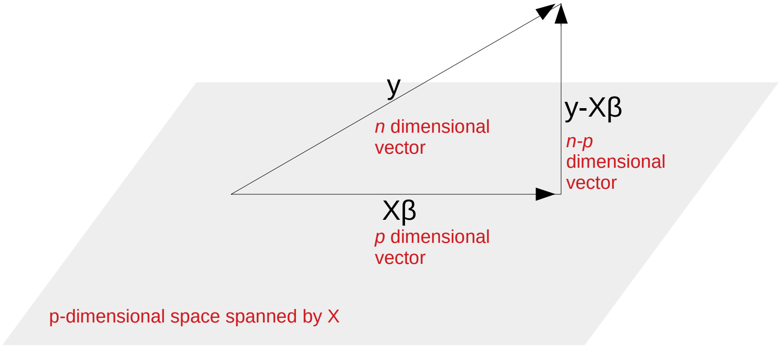
Remember: after *learning* the β parameters using the training data $\{\mathbf{x}_i, y_i\}$, with the model encoded in the feature matrix X, it is then possible to predict $\hat{y}_k$ for "new" (test) $\mathbf{x}_k$ values, using separate *prediction* function calls.

SLR: Residual Plot for the model



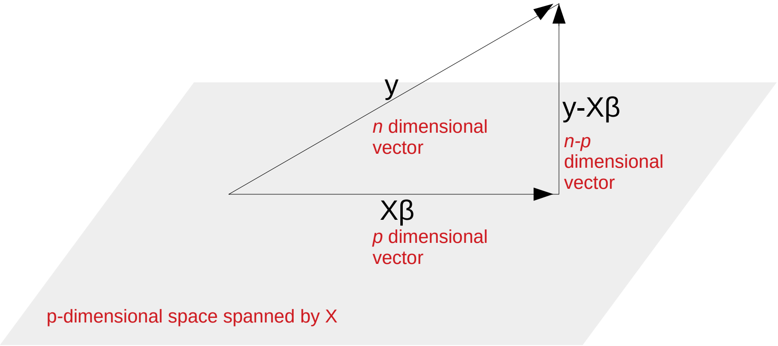
Source: ISLR, Fig 3.1: Advertising data with the model "Sales $\sim$ TV".
Note the vertical distance between the red dots (data points) $\mathbf{y}$ and the corresponding $\hat{\mathbf{y}}$ on the regression line, which is termed the *error* ϵ.

Geometrical interpretation of regression: n rows, p features, $n > p$



- Analogy: achieving photorealism with a limited palette of colours.
- Grey plane represents all the colours mixable from those colours.
- Point above plane: a colour that needs to be approximated.

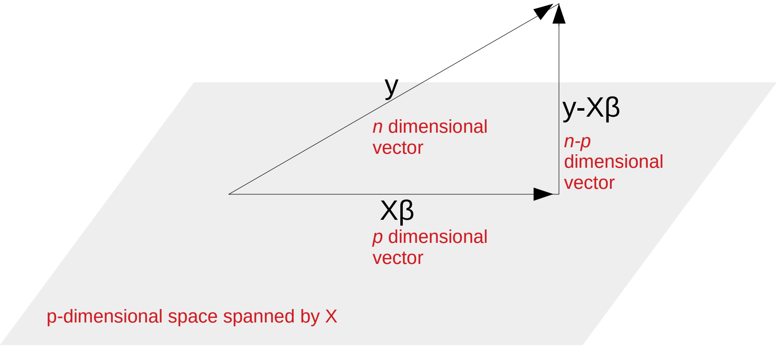
Geometrical interpretation of regression: n rows, p features, $n > p$



- Analogy: achieving photorealism with a limited palette of colours.
- Grey plane represents all the colours mixable from those colours.
- Point above plane: a colour that needs to be approximated.

Note that the X matrix spans the $p \times p$ space represented by the grey plane, but y has $n > p$ dimensions and so is represented by a point that lies outside the plane. When y is projected onto the X space, the projected point is $\hat{\mathbf{y}}$ and the residuals are represented by the vector $\mathbf{y} - X\beta \equiv \mathbf{y} - \hat{\mathbf{y}}$.

Geometrical interpretation of regression: n rows, p features, $n > p$



- Analogy: achieving photorealism with a limited palette of colours.
- Grey plane represents all the colours mixable from those colours.
- Point above plane: a colour that needs to be approximated.

Note that the X matrix spans the $p \times p$ space represented by the grey plane, but y has $n > p$ dimensions and so is represented by a point that lies outside the plane. When y is projected onto the X space, the projected point is $\hat{\mathbf{y}}$ and the residuals are represented by the vector $\mathbf{y} - X\beta \equiv \mathbf{y} - \hat{\mathbf{y}}$.
This decomposition of n data dimensions (observations) into p model parameters and n residuals with rank $n - p$ is helpful when interpreting regression diagnostics.

OLS and Linear Regression

Definition 3 (BLUE)

According to the Gauss-Markov theorem, *Ordinary Least Squares* (OLS), which uses the Normal equations to minimise the sum of the squares of the errors ($\|\epsilon\|_2 \equiv \sqrt{e_1^2 + e_2^2 + e_3^2 + \ldots + e_n^2}$), is the *Best, Linear, Unbiased, Estimator* of that model that can be derived from the training data, provided some reasonably loose assumptions hold.

OLS and Linear Regression

Definition 3 (BLUE)

According to the Gauss-Markov theorem, *Ordinary Least Squares* (OLS), which uses the Normal equations to minimise the sum of the squares of the errors ($\|\epsilon\|_2 \equiv \sqrt{e_1^2 + e_2^2 + e_3^2 + \ldots + e_n^2}$), is the *Best, Linear, Unbiased, Estimator* of that model that can be derived from the training data, provided some reasonably loose assumptions hold.

When we discuss Bias, Variance and Irreducible Error, it is clear that low bias is not enough. OLS might be BLUE but that does not guarantee low variance, because overfitting can still be a problem.

OLS and Linear Regression

Definition 3 (BLUE)

According to the Gauss-Markov theorem, *Ordinary Least Squares* (OLS), which uses the Normal equations to minimise the sum of the squares of the errors ($\|\epsilon\|_2 \equiv \sqrt{e_1^2 + e_2^2 + e_3^2 + \ldots + e_n^2}$), is the *Best, Linear, Unbiased, Estimator* of that model that can be derived from the training data, provided some reasonably loose assumptions hold.

When we discuss Bias, Variance and Irreducible Error, it is clear that low bias is not enough. OLS might be BLUE but that does not guarantee low variance, because overfitting can still be a problem.
In practice, the assumptions required for OLS to be appropriate can be stated in terms of properties of the residual vector ϵ.

OLS and Linear Regression

Definition 3 (BLUE)

According to the Gauss-Markov theorem, *Ordinary Least Squares* (OLS), which uses the Normal equations to minimise the sum of the squares of the errors ($\|\epsilon\|_2 \equiv \sqrt{e_1^2 + e_2^2 + e_3^2 + \ldots + e_n^2}$), is the *Best, Linear, Unbiased, Estimator* of that model that can be derived from the training data, provided some reasonably loose assumptions hold.

When we discuss Bias, Variance and Irreducible Error, it is clear that low bias is not enough. OLS might be BLUE but that does not guarantee low variance, because overfitting can still be a problem.
In practice, the assumptions required for OLS to be appropriate can be stated in terms of properties of the residual vector ϵ.
In the rest of this lecture, we will generalise from Simple to Multiple Linear Regression, where $\beta = (\beta_0, \beta_1, \beta_2, \ldots, \beta_p)$ and $2 \leq p \leq n$, so instead of fitting lines, we fit (hyper)planes to data.

Assumptions required for the linear model to be meaningful

Definition 4 (Linear Regression Assumptions)

1. The underlying relationship between the predictors and the response is linear in the regression parameters β.

Assumptions required for the linear model to be meaningful

Definition 4 (Linear Regression Assumptions)

1. The underlying relationship between the predictors and the response is linear in the regression parameters β.
2. The residual errors ϵ are drawn from a (multivariate) Normal distribution $N(\mu, \sigma^2)$ where $\mu = \mathbf{0}$.

Assumptions required for the linear model to be meaningful

Definition 4 (Linear Regression Assumptions)

1. The underlying relationship between the predictors and the response is linear in the regression parameters β.
2. The residual errors ϵ are drawn from a (multivariate) Normal distribution $N(\mu, \sigma^2)$ where $\mu = \mathbf{0}$.
3. The predictors are not pairwise collinear, i.e., each pair of predictors β_{j_1} and β_{j_2} (associated with columns $X(:, j_1)$ and $X(:, j_2)$) have low correlation (equivalently, the inner product of $X(:, j_1)$ and $X(:, j_2)$ is far from zero).

Assumptions required for the linear model to be meaningful

Definition 4 (Linear Regression Assumptions)

1. The underlying relationship between the predictors and the response is linear in the regression parameters β.
2. The residual errors ϵ are drawn from a (multivariate) Normal distribution $N(\mu, \sigma^2)$ where $\mu = \mathbf{0}$.
3. The predictors are not pairwise collinear, i.e., each pair of predictors β_{j_1} and β_{j_2} (associated with columns $X(:, j_1)$ and $X(:, j_2)$) have low correlation (equivalently, the inner product of $X(:, j_1)$ and $X(:, j_2)$ is far from zero).
4. There is no auto-correlation in $\mathbf{y}$: each observation is independent of its "neighbours".

Assumptions required for the linear model to be meaningful

Definition 4 (Linear Regression Assumptions)

1. The underlying relationship between the predictors and the response is linear in the regression parameters β.
2. The residual errors ϵ are drawn from a (multivariate) Normal distribution $N(\mu, \sigma^2)$ where $\mu = \mathbf{0}$.
3. The predictors are not pairwise collinear, i.e., each pair of predictors β_{j_1} and β_{j_2} (associated with columns $X(:, j_1)$ and $X(:, j_2)$) have low correlation (equivalently, the inner product of $X(:, j_1)$ and $X(:, j_2)$ is far from zero).
4. There is no auto-correlation in $\mathbf{y}$: each observation is independent of its "neighbours".
5. The errors are *homoscedastic* (i.e., $\mathrm{Var}(\epsilon)$ is constant over the range of $\mathbf{x}$ or $\mathbf{y}$).

Assumptions required for the linear model to be meaningful

Definition 4 (Linear Regression Assumptions)

1. The underlying relationship between the predictors and the response is linear in the regression parameters β.
2. The residual errors ϵ are drawn from a (multivariate) Normal distribution $N(\mu, \sigma^2)$ where $\mu = \mathbf{0}$.
3. The predictors are not pairwise collinear, i.e., each pair of predictors β_{j_1} and β_{j_2} (associated with columns $X(:, j_1)$ and $X(:, j_2)$) have low correlation (equivalently, the inner product of $X(:, j_1)$ and $X(:, j_2)$ is far from zero).
4. There is no auto-correlation in $\mathbf{y}$: each observation is independent of its “neighbours”.
5. The errors are *homoscedastic* (i.e., Var(ϵ) is constant over the range of $\mathbf{x}$ or $\mathbf{y}$).

Because these assumptions depend both on the data and on the model fitted to that data, it is meaningless to say that “Data set A does not satisfy the linear regression assumptions”, because this observation might not apply to all formulations of all models applied to that data.

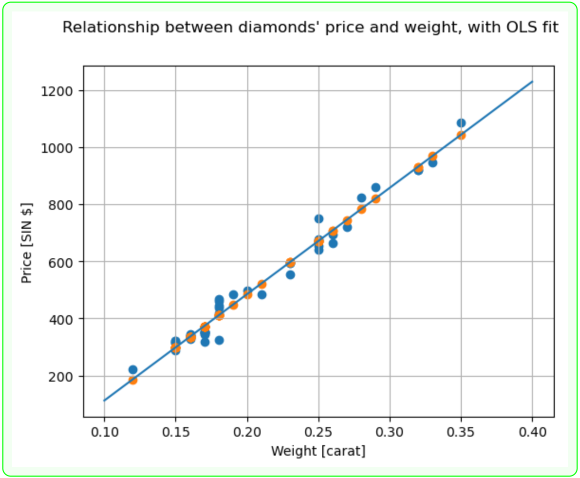
Assumptions required for the linear model to be meaningful

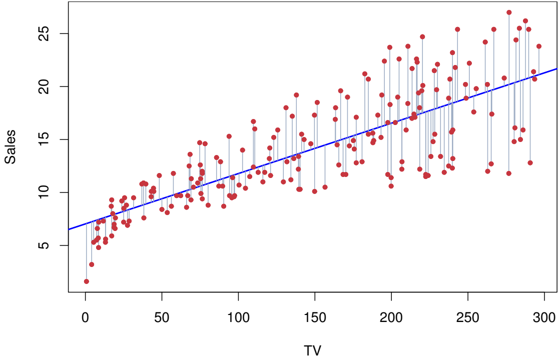
Definition 4 (Linear Regression Assumptions)

1. The underlying relationship between the predictors and the response is linear in the regression parameters β.
2. The residual errors ϵ are drawn from a (multivariate) Normal distribution $N(\mu, \sigma^2)$ where $\mu = \mathbf{0}$.
3. The predictors are not pairwise collinear, i.e., each pair of predictors β_{j_1} and β_{j_2} (associated with columns $X(:, j_1)$ and $X(:, j_2)$) have low correlation (equivalently, the inner product of $X(:, j_1)$ and $X(:, j_2)$ is far from zero).
4. There is no auto-correlation in $\mathbf{y}$: each observation is independent of its “neighbours”.
5. The errors are *homoscedastic* (i.e., $\mathrm{Var}(\epsilon)$ is constant over the range of $\mathbf{x}$ or $\mathbf{y}$).

Because these assumptions depend both on the data and on the model fitted to that data, it is meaningless to say that “Data set A does not satisfy the linear regression assumptions”, because this observation might not apply to all formulations of all models applied to that data.
Consequently, these assumptions can be used constructively, when model building, or as checks, when validating models.

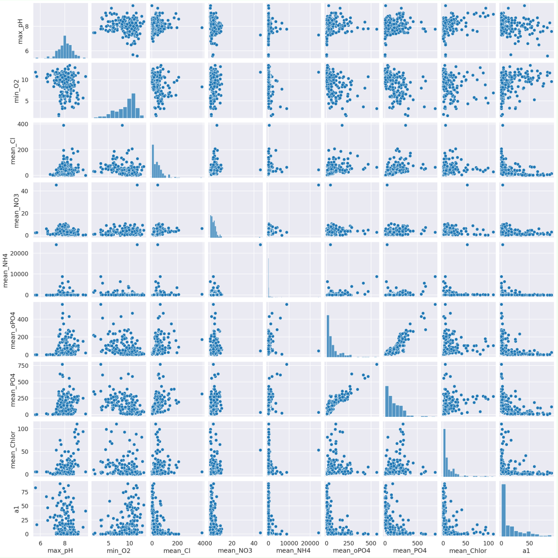
Linear relationship





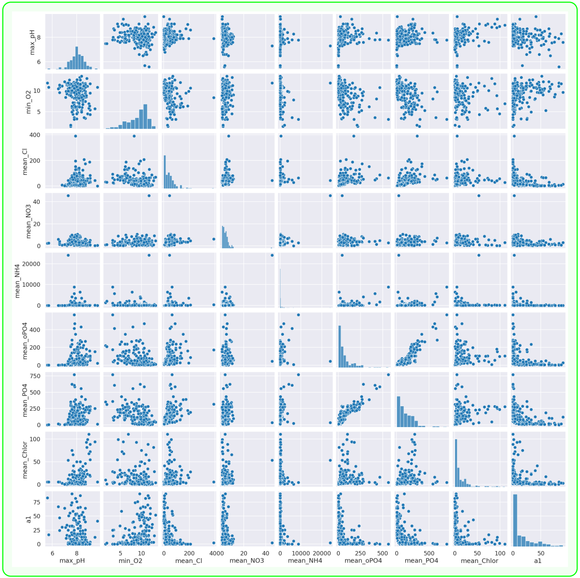
- In both cases, the relationship between predictor (feature) and target is approximately linear.
- Given a feature value, we can predict the target value using a simple linear formula.
- The predicted parameters are the *intercept* β_0 and *slope* β_1 of the line.
- Usually the vertical distance between a data point x_i and its predicted value $\hat{y}_i$ is $\epsilon_i \neq 0$.

Collinearity (high pairwise correlation) among the algae bloom predictors



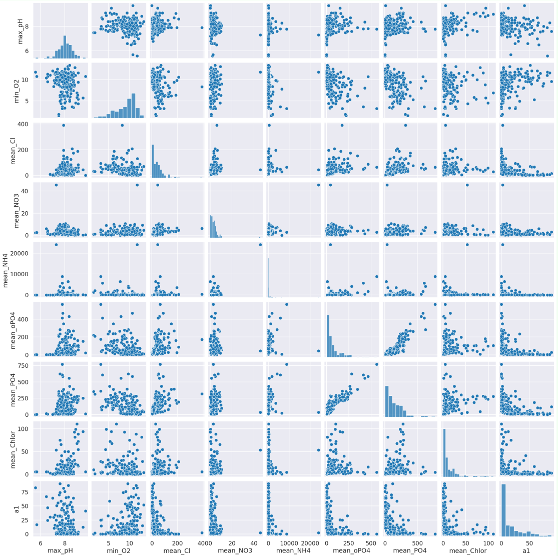
- The pairplot confirms what we saw in the corresponding correlation matrix: `mean_PO4` and `mean_OPO4` are highly correlated with each other (indeed, the relevant scatterplots indicate a strong linear relationship).

Collinearity (high pairwise correlation) among the algae bloom predictors



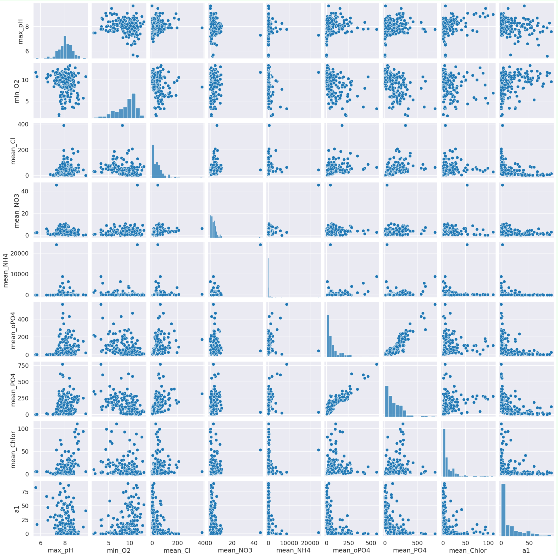
- The pairplot confirms what we saw in the corresponding correlation matrix: `mean_PO4` and `mean_OPO4` are highly correlated with each other (indeed, the relevant scatterplots indicate a strong linear relationship).
- Either `mean_PO4` or `mean_OPO4` can be included in the model, but not both of them.

Collinearity (high pairwise correlation) among the algae bloom predictors



- The pairplot confirms what we saw in the corresponding correlation matrix: `mean_PO4` and `mean_OPO4` are highly correlated with each other (indeed, the relevant scatterplots indicate a strong linear relationship).
- Either `mean_PO4` or `mean_OPO4` can be included in the model, but not both of them.
- Also, the individual predictors do not have a strong linear relationship with `a1` (look at the scatterplots in the last row and column) so, on their own, they are not likely to predict `a1` well with a linear model.

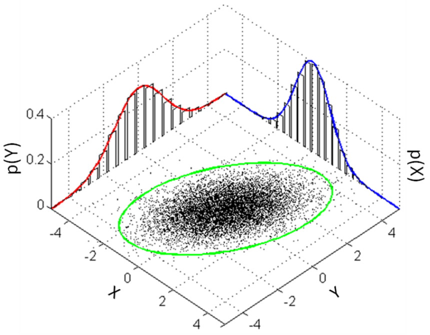
Collinearity (high pairwise correlation) among the algae bloom predictors

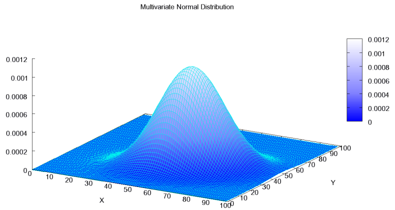


- The pairplot confirms what we saw in the corresponding correlation matrix: `mean_PO4` and `mean_OPO4` are highly correlated with each other (indeed, the relevant scatterplots indicate a strong linear relationship).
- Either `mean_PO4` or `mean_OPO4` can be included in the model, but not both of them.
- Also, the individual predictors do not have a strong linear relationship with `a1` (look at the scatterplots in the last row and column) so, on their own, they are not likely to predict `a1` well with a linear model.
- However, it is still possible that a combination of predictors might predict a1 well.

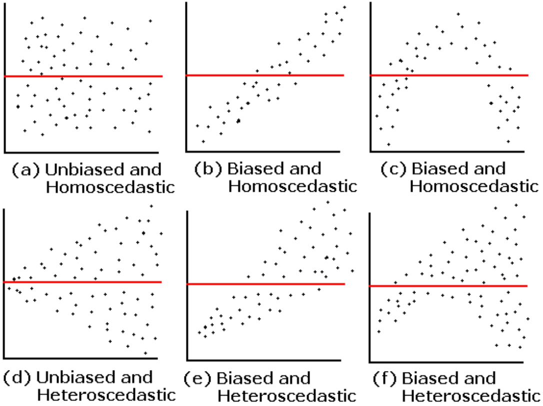
Errors are normally distributed

- Centred on zero so small errors are more common
- Symmetric so positive and negative balance out
- Normal distribution is a also called the Gaussian distribution and is “bell-shaped”.





Bias and variance in regression

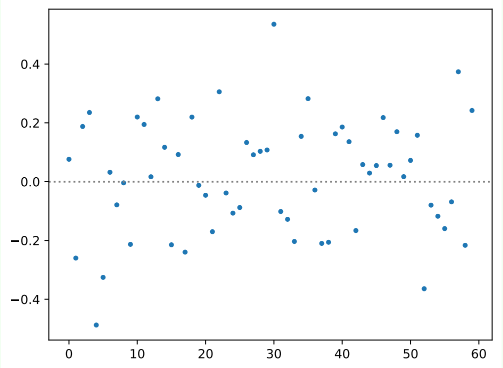


Source: `https://bit.ly/3vC9zK7`

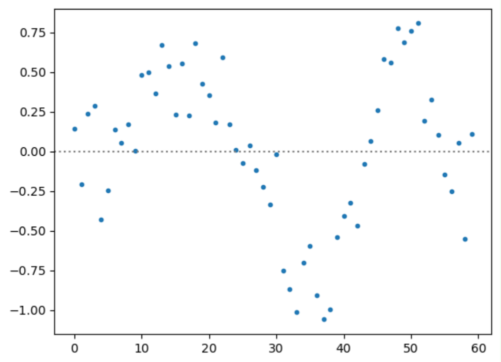
- Bias is caused by underfitting.
- Fix bias by adding suitable predictors.
- Overfitting causes large variance.
- If variance changes over the range, some errors get undue attention.
- Fix this by weighting the errors so they are balanced.

Errors should not be serially correlated

No serial correlation



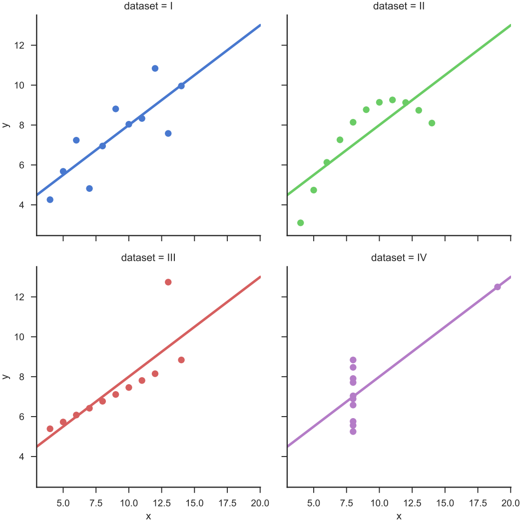
Positive serial correlation



Apparent seasonal effects - can they be removed?

1. Add feature to the model
2. Include autoregressive terms (but then it is no longer Ordinary Least Squares (OLS)!)

Anscombe's quartet (1973)



Francis Anscombe devised 4 data sets to show different forms of misalignment between data and models. Sets I,II,III share the same x values. All 4 sets share approximately the same descriptive statistics (mean and variance), but little else is common to all 4!

Only I appears suited as it stands. The other data sets require some work, particularly IV.

What do you think needs to be done for each data set?

Common Cost Functions in Regression Models

Remember: we are trying to minimise a loss function based on the error, which we approximate with the residuals of the training set.

Common Cost Functions in Regression Models

Remember: we are trying to minimise a loss function based on the error, which we approximate with the residuals of the training set.

Measure	Definition	Purpose
Mean square error (MSE)	$\frac{(p_1-a_1)^2+\cdots+(p_m-a_m)^2}{m}$	Mathematically tractable but places greater emphasise on observations with large error
Root mean square error (RMSE)	$\sqrt{\frac{(p_1-a_1)^2+\cdots+(p_m-a_m)^2}{m}}$	Has same units as data
Mean absolute error (RMAE)	$\frac{\lvert p_1-a_1\rvert+\cdots+\lvert p_m-a_m\rvert}{m}$	Does not overemphasise observations with large error (like MSE does)
Relative square error (RSE)	$\frac{(p_1-a_1)^2+\cdots+(p_m-a_m)^2}{(p_1-\bar{a})^2+\cdots+(p_m-\bar{a})^2}$	Relative metric compares the error in the predictions with errors in the simplest model possible (a model just always predicting the average value of y)
Root Relative square error (RRSE)	$\sqrt{\frac{(p_1-a_1)^2+\cdots+(p_m-a_m)^2}{(p_1-\bar{a})^2+\cdots+(p_m-\bar{a})^2}}$	
Relative absolute error (RAE)	$\frac{\lvert p_1-a_1\rvert+\cdots+\lvert p_m-a_m\rvert}{\lvert p_1-\bar{a}\rvert+\cdots+\lvert p_m-\bar{a}\rvert}$	

where a_j is the actual value, p_j is the predicted value, m is the number of observations, and $\bar{a}$ represents the mean of the a_j.

Choices of Vector norms

Definition 5 (Manhattan norm)

$\ell_1(\dots) = \|\dots\|_1$ is the *Manhattan* norm (length) of a vector. Let $\mathbf{x} = (x_1, x_2, \dots, x_m)$. Then $\ell_1(\dots) = \|\dots\|_1 = |x_1| + |x_2| + \dots + |x_m|$ is the *Manhattan* distance of $\mathbf{x}$ from the origin. Think of having to *walk* from one junction in Manhattan to another, the distance is the difference in the Street numbers plus the difference in the Avenue numbers.

Choices of Vector norms

Definition 5 (Manhattan norm)

$\ell_1(\ldots) = \|\ldots\|_1$ is the *Manhattan* norm (length) of a vector. Let $\mathbf{x} = (x_1, x_2, \ldots, x_m)$. Then $\ell_1(\ldots) = \|\ldots\|_1 = |x_1| + |x_2| + \ldots + |x_m|$ is the *Manhattan* distance of $\mathbf{x}$ from the origin. Think of having to *walk* from one junction in Manhattan to another, the distance is the difference in the Street numbers plus the difference in the Avenue numbers.

Definition 6 (Euclidean norm)

$\ell_2(\ldots) = \|\ldots\|_2$ is the *Euclidean* norm (length) of a vector. Let $\mathbf{x} = (x_1, x_2, \ldots, x_m)$. Then $\ell_2(\ldots) = \|\ldots\|_2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_m^2}$ is the *Euclidean* distance of $\mathbf{x}$ from the origin. Think of being able to *fly* over all the buildings using the shortest route (think: Pythagoras theorem!) from one junction in Manhattan to another.

Choices of Vector norms

Definition 5 (Manhattan norm)

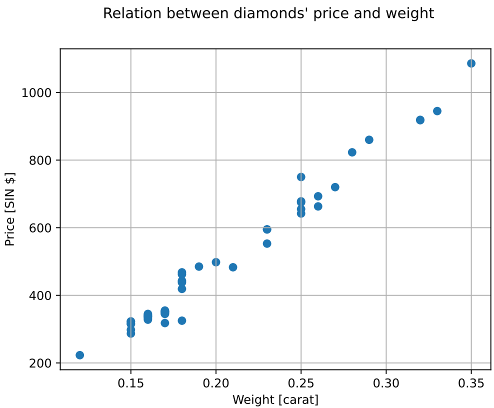
$\ell_1(\ldots) = \|\ldots\|_1$ is the *Manhattan* norm (length) of a vector. Let $\mathbf{x} = (x_1, x_2, \ldots, x_m)$. Then $\ell_1(\ldots) = \|\ldots\|_1 = |x_1| + |x_2| + \ldots + |x_m|$ is the *Manhattan* distance of $\mathbf{x}$ from the origin. Think of having to *walk* from one junction in Manhattan to another, the distance is the difference in the Street numbers plus the difference in the Avenue numbers.

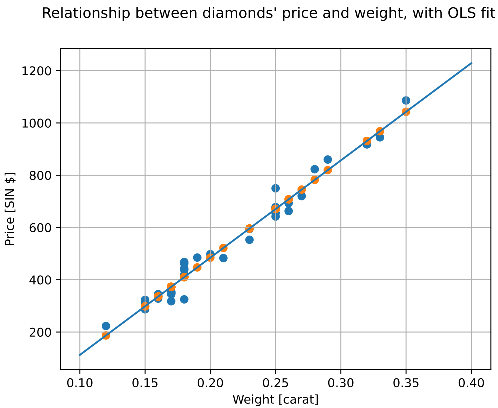
Definition 6 (Euclidean norm)

$\ell_2(\ldots) = \|\ldots\|_2$ is the *Euclidean* norm (length) of a vector. Let $\mathbf{x} = (x_1, x_2, \ldots, x_m)$. Then $\ell_2(\ldots) = \|\ldots\|_2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_m^2}$ is the *Euclidean* distance of $\mathbf{x}$ from the origin. Think of being able to *fly* over all the buildings using the shortest route (think: Pythagoras theorem!) from one junction in Manhattan to another.

The Euclidean norm is very common, but the Manhattan norm is gaining popularity, because it is robust to outliers and computers are becoming powerful enough. However we generally use Euclidean norm in this module.

Case Study 2: Diamonds - Check relationship





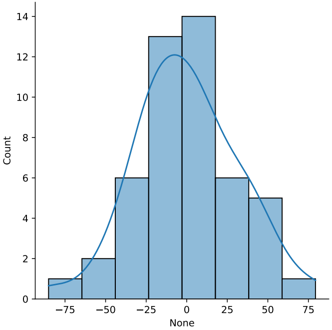
Clearly there is a linear relationship between a diamond's weight (in carats) and its price (in Singapore dollars, as here). So that is one assumption satisfied!

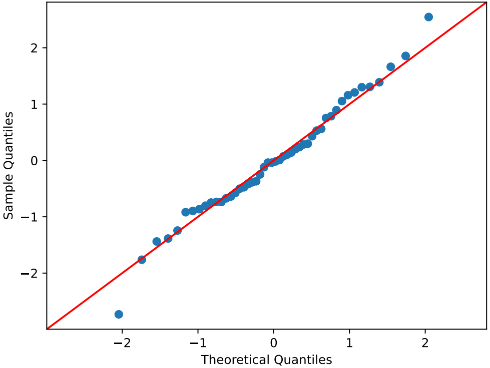
Sometimes the dependent variable has a linear dependence on a function of an attribute. Example functions include log, exp, sqrt, polynomial, etc. Even if the function is nonlinear in the attribute, that does not matter, as long as the model is linear in the regression parameters β.

Case Study 2: Diamonds - Check residual distribution

```python
import seaborn as sns
resFig = "res/residHist.pdf"
sns_plot = sns.displot(x = residuals, kde=True)
sns_plot.savefig(resFig)
```

```python
# Q-Q plot to verify the residuals distribution
resFig = "res/residualsqq.pdf"
fig = sm.qqplot(residuals, fit=True, line = '45')
fig.savefig(resFig)
```

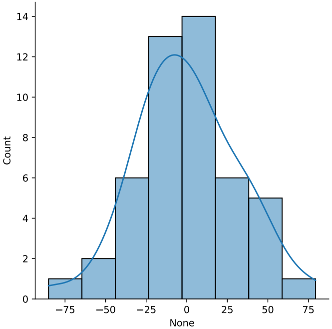


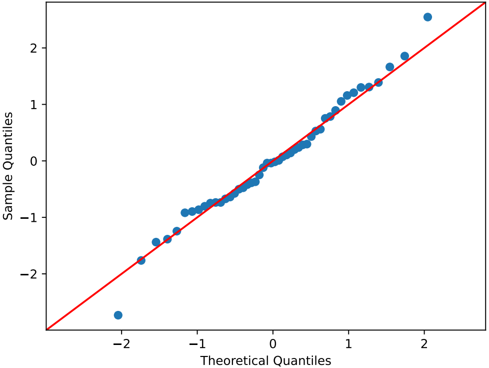


Case Study 2: Diamonds - Check residual distribution

```python
import seaborn as sns
resFig = "res/residHist.pdf"
sns_plot = sns.displot(x = residuals, kde=True)
sns_plot.savefig(resFig)
```

```python
# Q-Q plot to verify the residuals distribution
resFig = "res/residualsqq.pdf"
fig = sm.qqplot(residuals, fit=True, line = '45')
fig.savefig(resFig)
```



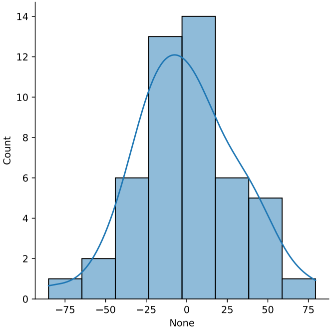


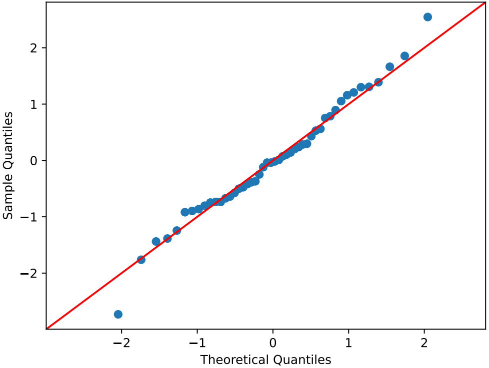
Both diagnostic plots indicate the residuals are reasonably close to Normal distribution centred on 0. The qqplot is perhaps more informative. Looking good so far!

Case Study 2: Diamonds - Check residual distribution

```python
import seaborn as sns
resFig = "res/residHist.pdf"
sns_plot = sns.displot(x = residuals, kde=True)
sns_plot.savefig(resFig)
```

```python
# Q-Q plot to verify the residuals distribution
resFig = "res/residualsqq.pdf"
fig = sm.qqplot(residuals, fit=True, line = '45')
fig.savefig(resFig)
```





Both diagnostic plots indicate the residuals are reasonably close to Normal distribution centred on 0. The qqplot is perhaps more informative. Looking good so far!

Case Study 2: Diamonds - model summary

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.978
Model:	OLS	**Adj. R-squared:**	0.978
Method:	Least Squares	**F-statistic:**	2070.
Date:	Fri, 25 Oct 2024	**Prob (F-statistic):**	6.75e-40
Time:	10:54:56	**Log-Likelihood:**	-233.20
No. Observations:	48	**AIC:**	470.4
Df Residuals:	46	**BIC:**	474.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-259.6259	17.319	-14.991	0.000	-294.487	-224.765
carats	3721.0249	81.786	45.497	0.000	3556.398	3885.651

Omnibus:	0.739	**Durbin-Watson:**	1.994
Prob(Omnibus):	0.691	**Jarque-Bera (JB):**	0.181
Skew:	0.056	**Prob(JB):**	0.913
Kurtosis:	3.280	**Cond. No.**	18.5

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
simpleModel.summary()
```

The output from Python's statsmodels.summary() call has lots of information!

Case Study 2: Diamonds - model summary

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.978
Model:	OLS	**Adj. R-squared:**	0.978
Method:	Least Squares	**F-statistic:**	2070.
Date:	Fri, 25 Oct 2024	**Prob (F-statistic):**	6.75e-40
Time:	10:54:56	**Log-Likelihood:**	-233.20
No. Observations:	48	**AIC:**	470.4
Df Residuals:	46	**BIC:**	474.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-259.6259	17.319	-14.991	0.000	-294.487	-224.765
carats	3721.0249	81.786	45.497	0.000	3556.398	3885.651

Omnibus:	0.739	**Durbin-Watson:**	1.994
Prob(Omnibus):	0.691	**Jarque-Bera (JB):**	0.181
Skew:	0.056	**Prob(JB):**	0.913
Kurtosis:	3.280	**Cond. No.**	18.5

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
simpleModel.summary()
```

The output from Python's statsmodels.summary() call has lots of information!

- **How much of the variability of the data is explained by the model?**

Case Study 2: Diamonds - model summary

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.978
Model:	OLS	**Adj. R-squared:**	0.978
Method:	Least Squares	**F-statistic:**	2070.
Date:	Fri, 25 Oct 2024	**Prob (F-statistic):**	6.75e-40
Time:	10:54:56	**Log-Likelihood:**	-233.20
No. Observations:	48	**AIC:**	470.4
Df Residuals:	46	**BIC:**	474.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-259.6259	17.319	-14.991	0.000	-294.487	-224.765
carats	3721.0249	81.786	45.497	0.000	3556.398	3885.651

Omnibus:	0.739	**Durbin-Watson:**	1.994
Prob(Omnibus):	0.691	**Jarque-Bera (JB):**	0.181
Skew:	0.056	**Prob(JB):**	0.913
Kurtosis:	3.280	**Cond. No.**	18.5

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
simpleModel.summary()
```

The output from Python's statsmodels.summary() call has lots of information!

- **How much of the variability of the data is explained by the model?**
- **What is the probability that such data arose if price does not increase with weight?**

Case Study 2: Diamonds - model summary

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.978
Model:	OLS	**Adj. R-squared:**	0.978
Method:	Least Squares	**F-statistic:**	2070.
Date:	Fri, 25 Oct 2024	**Prob (F-statistic):**	6.75e-40
Time:	10:54:56	**Log-Likelihood:**	-233.20
No. Observations:	48	**AIC:**	470.4
Df Residuals:	46	**BIC:**	474.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-259.6259	17.319	-14.991	0.000	-294.487	-224.765
carats	3721.0249	81.786	45.497	0.000	3556.398	3885.651

Omnibus:	0.739	**Durbin-Watson:**	1.994
Prob(Omnibus):	0.691	**Jarque-Bera (JB):**	0.181
Skew:	0.056	**Prob(JB):**	0.913
Kurtosis:	3.280	**Cond. No.**	18.5

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
simpleModel.summary()
```

The output from Python's statsmodels.summary() call has lots of information!

- **How much of the variability of the data is explained by the model?**
- **What is the probability that such data arose if price does not increase with weight?**
- **What score(s) indicate that the residuals are not serially correlated?**

Case Study 2: Diamonds - model summary

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.978
Model:	OLS	**Adj. R-squared:**	0.978
Method:	Least Squares	**F-statistic:**	2070.
Date:	Fri, 25 Oct 2024	**Prob (F-statistic):**	6.75e-40
Time:	10:54:56	**Log-Likelihood:**	-233.20
No. Observations:	48	**AIC:**	470.4
Df Residuals:	46	**BIC:**	474.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-259.6259	17.319	-14.991	0.000	-294.487	-224.765
carats	3721.0249	81.786	45.497	0.000	3556.398	3885.651

Omnibus:	0.739	**Durbin-Watson:**	1.994
Prob(Omnibus):	0.691	**Jarque-Bera (JB):**	0.181
Skew:	0.056	**Prob(JB):**	0.913
Kurtosis:	3.280	**Cond. No.**	18.5

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
simpleModel.summary()
```

The output from Python's statsmodels.summary() call has lots of information!

- **How much of the variability of the data is explained by the model?**
- **What is the probability that such data arose if price does not increase with weight?**
- **What score(s) indicate that the residuals are not serially correlated?**
- **What scores indicate that the distribution of the residuals is Normal?**

Model summary interpretation - 1

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.978
Model:	OLS	**Adj. R-squared:**	0.978
Method:	Least Squares	**F-statistic:**	2070.
Date:	Fri, 25 Oct 2024	**Prob (F-statistic):**	6.75e-40
Time:	10:54:56	**Log-Likelihood:**	-233.20
No. Observations:	48	**AIC:**	470.4
Df Residuals:	46	**BIC:**	474.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-259.6259	17.319	-14.991	0.000	-294.487	-224.765
carats	3721.0249	81.786	45.497	0.000	3556.398	3885.651

Omnibus:	0.739	**Durbin-Watson:**	1.994
Prob(Omnibus):	0.691	**Jarque-Bera (JB):**	0.181
Skew:	0.056	**Prob(JB):**	0.913
Kurtosis:	3.280	**Cond. No.**	18.5

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Definition 7 (Dep. variable)

This is synonymous with the *target*, which is `price` (in this dataset).

Definition 8 (Model)

statsmodels uses it here in the sense of *problem formulation*. We wish to solve an Ordinary Least Squares problem (assumes all the regression assumptions are met, so no special treatment was applied).

Definition 9 (No. Observations)

This is the number of rows (also known as instances or cases) in the training set.

Model summary interpretation - 2

Definition 10 (Df Model)

The model has one named feature (carats (weight of the diamond)) and one unnamed feature (constant, independent of carats). df, the number of degrees of freedom counts the named features.

Definition 11 (Df Residuals)

The number of degrees of freedom in the residuals is the number of residuals minus the number of features. A higher value tends to go with smaller model variance.

Definition 12 (Covariance Type)

If residuals have the same variance (homoscedastic), nonrobust covariance (the default) can be used.

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.978
Model:	OLS	**Adj. R-squared:**	0.978
Method:	Least Squares	**F-statistic:**	2070.
Date:	Fri, 25 Oct 2024	**Prob (F-statistic):**	6.75e-40
Time:	10:54:56	**Log-Likelihood:**	-233.20
No. Observations:	48	**AIC:**	470.4
Df Residuals:	46	**BIC:**	474.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-259.6259	17.319	-14.991	0.000	-294.487	-224.765
carats	3721.0249	81.786	45.497	0.000	3556.398	3885.651

Omnibus:	0.739	**Durbin-Watson:**	1.994
Prob(Omnibus):	0.691	**Jarque-Bera (JB):**	0.181
Skew:	0.056	**Prob(JB):**	0.913
Kurtosis:	3.280	**Cond. No.**	18.5

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Model summary interpretation - 3

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.978
Model:	OLS	**Adj. R-squared:**	0.978
Method:	Least Squares	**F-statistic:**	2070.
Date:	Fri, 25 Oct 2024	**Prob (F-statistic):**	6.75e-40
Time:	10:54:56	**Log-Likelihood:**	-233.20
No. Observations:	48	**AIC:**	470.4
Df Residuals:	46	**BIC:**	474.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-259.6259	17.319	-14.991	0.000	-294.487	-224.765
carats	3721.0249	81.786	45.497	0.000	3556.398	3885.651

Omnibus:	0.739	**Durbin-Watson:**	1.994
Prob(Omnibus):	0.691	**Jarque-Bera (JB):**	0.181
Skew:	0.056	**Prob(JB):**	0.913
Kurtosis:	3.280	**Cond. No.**	18.5

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Definition 13 (R-squared)

This is the ratio of the data variance explained by the model, to the variance of the data. It ranges from zero (model explains none of the data variance) to one (model explains all the data variance). A higher value is better, but be careful of overfitting the training set!

Definition 14 (Adj. R-squared)

Similar to R-squared, but it takes account of the number of features. Adding a feature generally increases R-squared, but if the feature did not help as much as its peers, adjusted R-squared shows a smaller increase than “normal” R-squared.

Model summary interpretation - 4

Definition 15 (F-statistic)

Ratio of the variance of a model with just the constant (intercept) feature to the variance of this model. Generally, large values of F are preferred.

Definition 16 (Prob (F statistic))

The value is assumed to follow the F distribution for given *dof*, so can lookup its probability. Small probability indicates that it is highly *unlikely* that the model is doing well purely by chance.

Definition 17 (log likelihood)

OLS is a special case of *maximum likelihood estimation*. Larger likelihood model fits the training data better.

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.978
Model:	OLS	**Adj. R-squared:**	0.978
Method:	Least Squares	**F-statistic:**	2070.
Date:	Fri, 25 Oct 2024	**Prob (F-statistic):**	6.75e-40
Time:	10:54:56	**Log-Likelihood:**	-233.20
No. Observations:	48	**AIC:**	470.4
Df Residuals:	46	**BIC:**	474.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-259.6259	17.319	-14.991	0.000	-294.487	-224.765
carats	3721.0249	81.786	45.497	0.000	3556.398	3885.651

Omnibus:	0.739	**Durbin-Watson:**	1.994
Prob(Omnibus):	0.691	**Jarque-Bera (JB):**	0.181
Skew:	0.056	**Prob(JB):**	0.913
Kurtosis:	3.280	**Cond. No.**	18.5

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Model summary interpretation - 5

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.978
Model:	OLS	**Adj. R-squared:**	0.978
Method:	Least Squares	**F-statistic:**	2070.
Date:	Fri, 25 Oct 2024	**Prob (F-statistic):**	6.75e-40
Time:	10:54:56	**Log-Likelihood:**	-233.20
No. Observations:	48	**AIC:**	470.4
Df Residuals:	46	**BIC:**	474.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-259.6259	17.319	-14.991	0.000	-294.487	-224.765
carats	3721.0249	81.786	45.497	0.000	3556.398	3885.651

Omnibus:	0.739	**Durbin-Watson:**	1.994
Prob(Omnibus):	0.691	**Jarque-Bera (JB):**	0.181
Skew:	0.056	**Prob(JB):**	0.913
Kurtosis:	3.280	**Cond. No.**	18.5

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Definition 18 (AIC and BIC)

Akaike and Bayesian Information Criterion. These are calculated from the residuals and are derived from *information theory*. They allow for the number of features. Lower values are better.

Definition 19 (Features table: const,carats in this example)

coef is the parameter value for that feature, e.g., const=-259.6 here. $P > |t| = 0$ so it is highly unlikely the coef is zero, given the training data. We also have the 2.5% and 97.5% quantiles, giving the expected range of coef.

Model summary interpretation - 6

Definition 20 (Skew, Kurtosis)

Measures of asymmetry and of peak shape of the residual distribution. Ideal values are 0 (skew) and 3 (kurtosis).

Definition 21 (Durbin-Watson)

Measures the serial correlation of the residuals. Ideal value is 2 (no serial correlation).

Definition 22 (Cond. no)

OLS implementation solves a linear system of equations. Condition number measures column (hence feature independence). Large values mean the features are not independent (they are correlated), making the system more difficult to solve.

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.978
Model:	OLS	**Adj. R-squared:**	0.978
Method:	Least Squares	**F-statistic:**	2070.
Date:	Fri, 25 Oct 2024	**Prob (F-statistic):**	6.75e-40
Time:	10:54:56	**Log-Likelihood:**	-233.20
No. Observations:	48	**AIC:**	470.4
Df Residuals:	46	**BIC:**	474.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-259.6259	17.319	-14.991	0.000	-294.487	-224.765
carats	3721.0249	81.786	45.497	0.000	3556.398	3885.651

Omnibus:	0.739	**Durbin-Watson:**	1.994
Prob(Omnibus):	0.691	**Jarque-Bera (JB):**	0.181
Skew:	0.056	**Prob(JB):**	0.913
Kurtosis:	3.280	**Cond. No.**	18.5

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

What does all this mean?

- Clearly, Linear Regression provides many diagnostics to assess how well the model works on the *training set*.

What does all this mean?

- Clearly, Linear Regression provides many diagnostics to assess how well the model works on the *training set*.
- We can test the model assumptions and many related desirable properties, when applied to the training set.

What does all this mean?

- Clearly, Linear Regression provides many diagnostics to assess how well the model works on the *training set*.
- We can test the model assumptions and many related desirable properties, when applied to the training set.
- However, in machine learning, our goal is to minimise prediction error on unseen data.

What does all this mean?

- Clearly, Linear Regression provides many diagnostics to assess how well the model works on the *training set*.
- We can test the model assumptions and many related desirable properties, when applied to the training set.
- However, in machine learning, our goal is to minimise prediction error on unseen data.
- As described last week, we need to perform a train-(validation)-test split and evaluate the errors on the test set.

What does all this mean?

- Clearly, Linear Regression provides many diagnostics to assess how well the model works on the *training set*.
- We can test the model assumptions and many related desirable properties, when applied to the training set.
- However, in machine learning, our goal is to minimise prediction error on unseen data.
- As described last week, we need to perform a train-(validation)-test split and evaluate the errors on the test set.
- *The best model is the one that minimises the error when predicting the target in the test set!*

What does all this mean?

- Clearly, Linear Regression provides many diagnostics to assess how well the model works on the *training set*.
- We can test the model assumptions and many related desirable properties, when applied to the training set.
- However, in machine learning, our goal is to minimise prediction error on unseen data.
- As described last week, we need to perform a train-(validation)-test split and evaluate the errors on the test set.
- *The best model is the one that minimises the error when predicting the target in the test set!*
- The Regression Model diagnostics can help

What does all this mean?

- Clearly, Linear Regression provides many diagnostics to assess how well the model works on the *training set*.
- We can test the model assumptions and many related desirable properties, when applied to the training set.
- However, in machine learning, our goal is to minimise prediction error on unseen data.
- As described last week, we need to perform a train-(validation)-test split and evaluate the errors on the test set.
- *The best model is the one that minimises the error when predicting the target in the test set!*
- The Regression Model diagnostics can help
  1. If a model does not do well, they can help to diagnose the problem(s).

What does all this mean?

- Clearly, Linear Regression provides many diagnostics to assess how well the model works on the *training set*.
- We can test the model assumptions and many related desirable properties, when applied to the training set.
- However, in machine learning, our goal is to minimise prediction error on unseen data.
- As described last week, we need to perform a train-(validation)-test split and evaluate the errors on the test set.
- *The best model is the one that minimises the error when predicting the target in the test set!*
- The Regression Model diagnostics can help
  1. If a model does not do well, they can help to diagnose the problem(s).
  2. They can also be used constructively, to help identify promising candidate models.

Summary

- We described linear models

Summary

- We described linear models
- We gave several ways to view what regression is: geometry, linear algebra, optimisation

Summary

- We described linear models
- We gave several ways to view what regression is: geometry, linear algebra, optimisation
- We described regression assumptions

Summary

- We described linear models
- We gave several ways to view what regression is: geometry, linear algebra, optimisation
- We described regression assumptions
- We looked at a simple example

Summary

- We described linear models
- We gave several ways to view what regression is: geometry, linear algebra, optimisation
- We described regression assumptions
- We looked at a simple example
- We consider ways of assessing the success of a regression model, even before checking its performance against the test set

Summary

- We described linear models
- We gave several ways to view what regression is: geometry, linear algebra, optimisation
- We described regression assumptions
- We looked at a simple example
- We consider ways of assessing the success of a regression model, even before checking its performance against the test set
- These quality metrics are not available for other problem formulations, but can help regression model building

Summary

- We described linear models
- We gave several ways to view what regression is: geometry, linear algebra, optimisation
- We described regression assumptions
- We looked at a simple example
- We consider ways of assessing the success of a regression model, even before checking its performance against the test set
- These quality metrics are not available for other problem formulations, but can help regression model building
- Next time - we return to classification…