MONASH University
MALAYSIA

MONASH
BUSINESS

Exploratory Data Analysis

ETW2001 Foundations of Data Analysis and Modelling

Manjeevan Singh Seera

Accredited by:

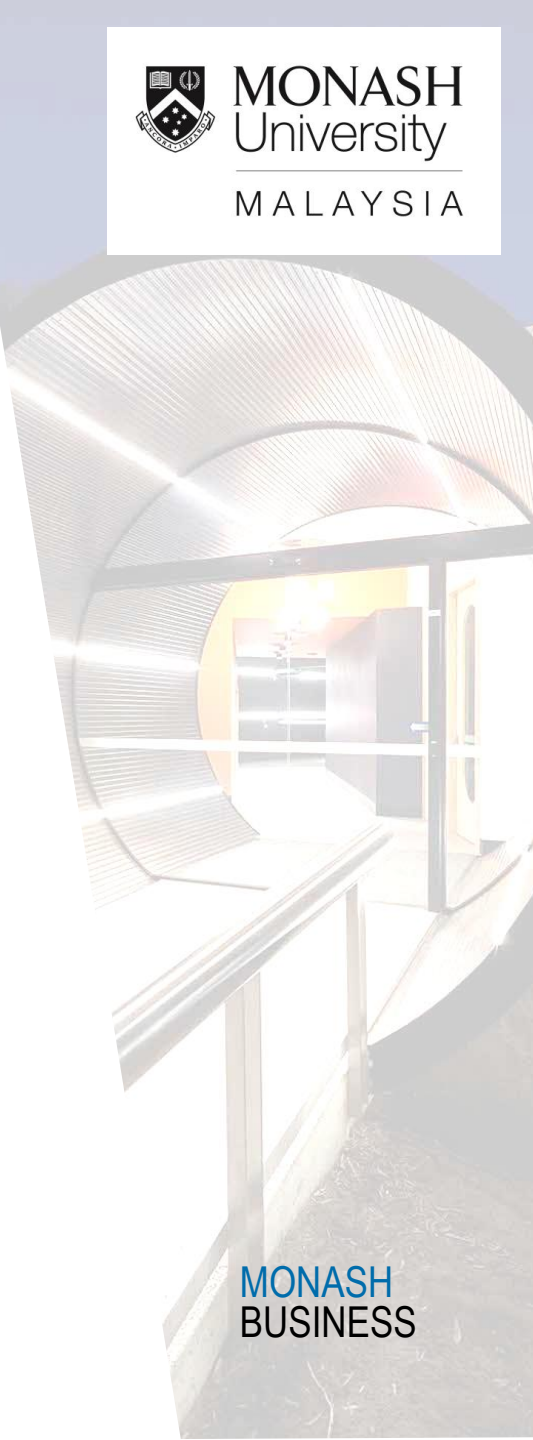
AACSB ACCREDITED

Advanced Signatory:

PRME
an initiative of the
United Nations Global Compact

Outline

- Exploratory Data Analysis
  - Variation
  - Covariation
  - Patterns and Models



MONASH University MALAYSIA

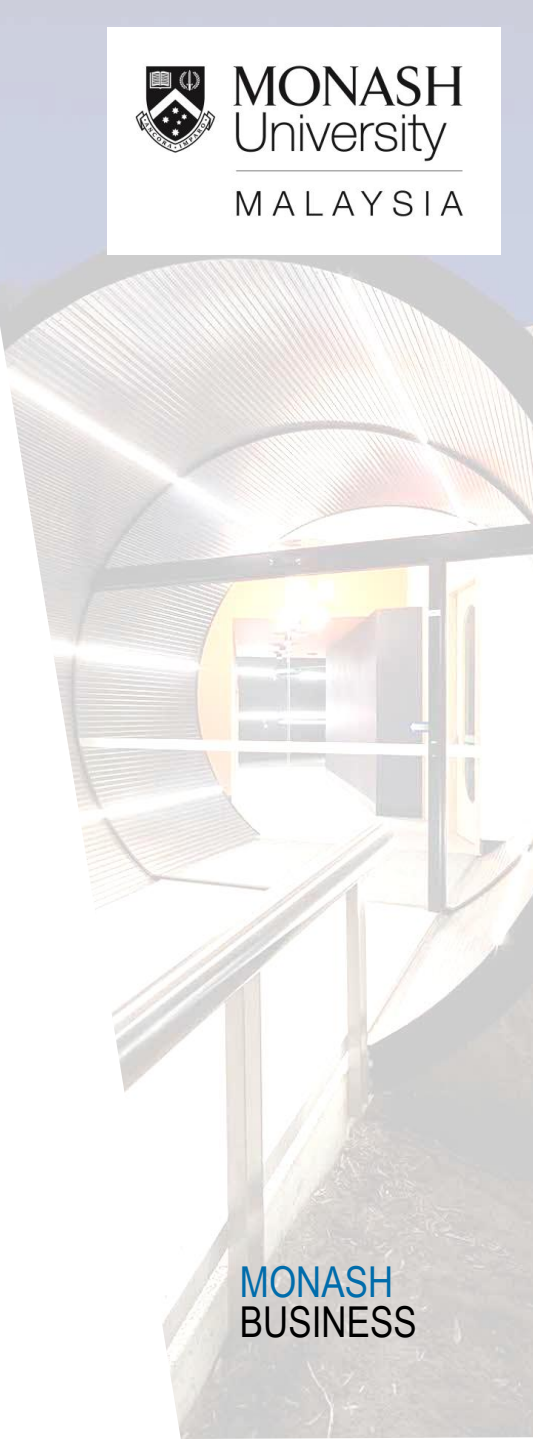
MONASH BUSINESS

Introduction

In this slides, you will learn how to use visualization and transformation to explore your data in a systematic way, a task that statisticians call **E**xploratory **D**ata **A**nalysis (EDA).

EDA is not a formal process with a strict set of rules. During the initial phases of EDA you should **feel free to investigate** every idea that occurs to you. Some of these ideas will pan out, and some will be dead ends (but don't worry!). You will eventually write up and communicate to others.

EDA is a vital part of any data analysis, as you always need to investigate the quality of your data. **Data cleaning** is just one application of EDA, as in data cleaning, you need to use all EDA tools: visualization, transformation, and modelling.



MONASH University MALAYSIA

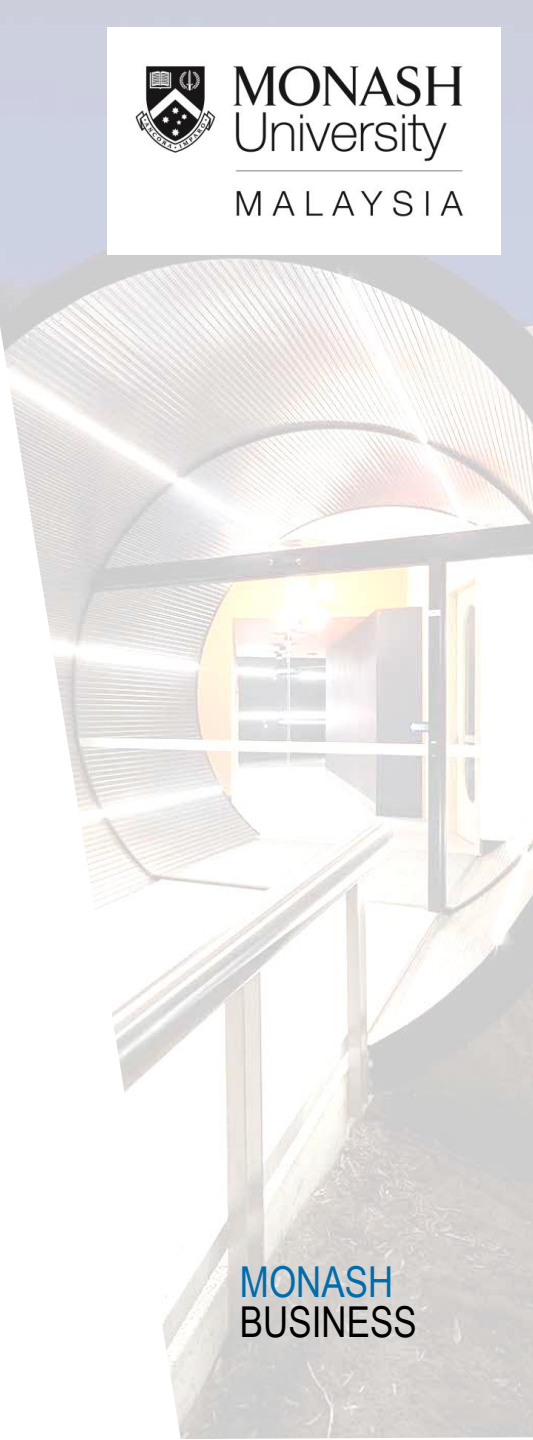
MONASH BUSINESS

Prerequisites

We will use what we have learned in dplyr and ggplot2 to interactively **ask** questions, **answer** them with data, and then **ask new** questions.

```
library(tidyverse)
```

```
── Attaching packages ─────────────────
✓ ggplot2 3.3.2     ✓ purrr   0.3.4
✓ tibble  3.0.4     ✓ dplyr   1.0.2
✓ tidyr   1.1.2     ✓ stringr 1.4.0
✓ readr   1.4.0     ✓ forcats 0.5.0
```



MONASH University MALAYSIA

MONASH BUSINESS

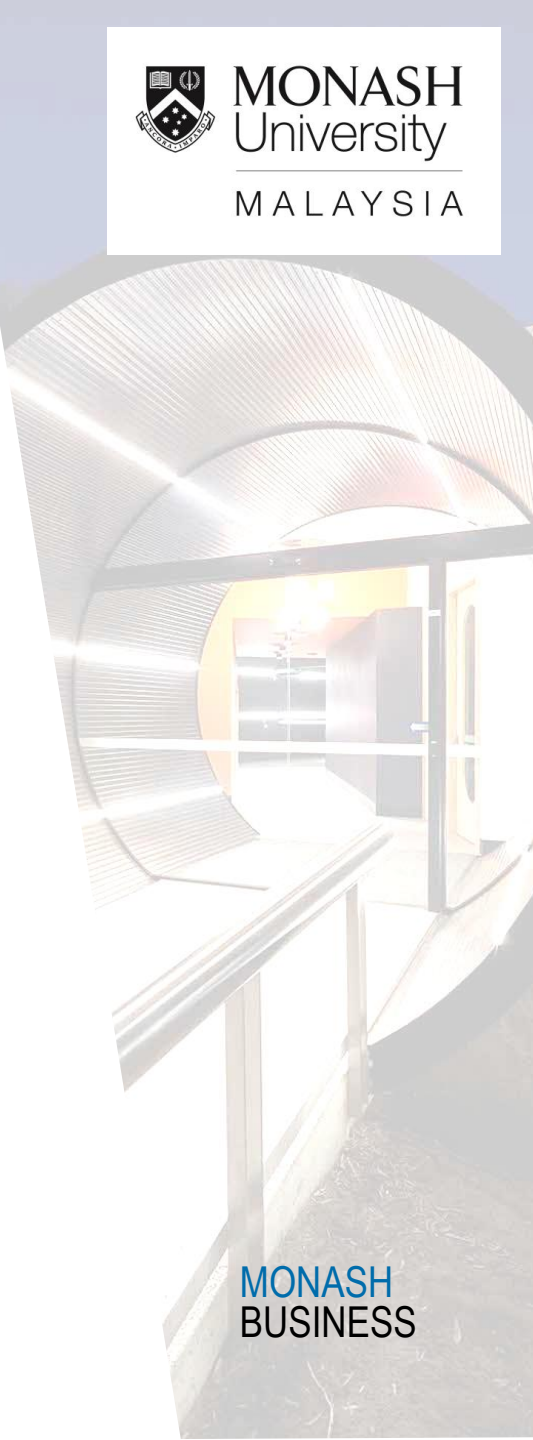
Questions

Your goal in EDA is to develop an **understanding** of your data. Simplest way to do so is using questions as tools in guiding your investigation.

- By asking questions, it focuses your attention to a specific part in the dataset, which will help you to decide which **graphs**, **models**, or **transformations** to make.

Fundamentally, EDA is a creative process. The key is to keep asking quality questions, as it will expose you to new aspects of your data, which increases your chances of making a discovery.

- You may be able to drill down to the interesting parts of the data and develop thought-provoking questions, if you follow-up each question with a new one.



Questions

When asking questions to guide the research, there are **no specific rules**. You can use these questions as a starting point:

- What type of variation occurs within my variables?
- What type of covariation occurs between my variables?

The two questions listed above will be looked at in the slides. Some terms are defined as follows.

- A **variable** is a quantity, quality, or property that **you can measure**.
- A **value** is the state of a variable **when you measure** it. The value of a variable may change from measurement to measurement.

Observation

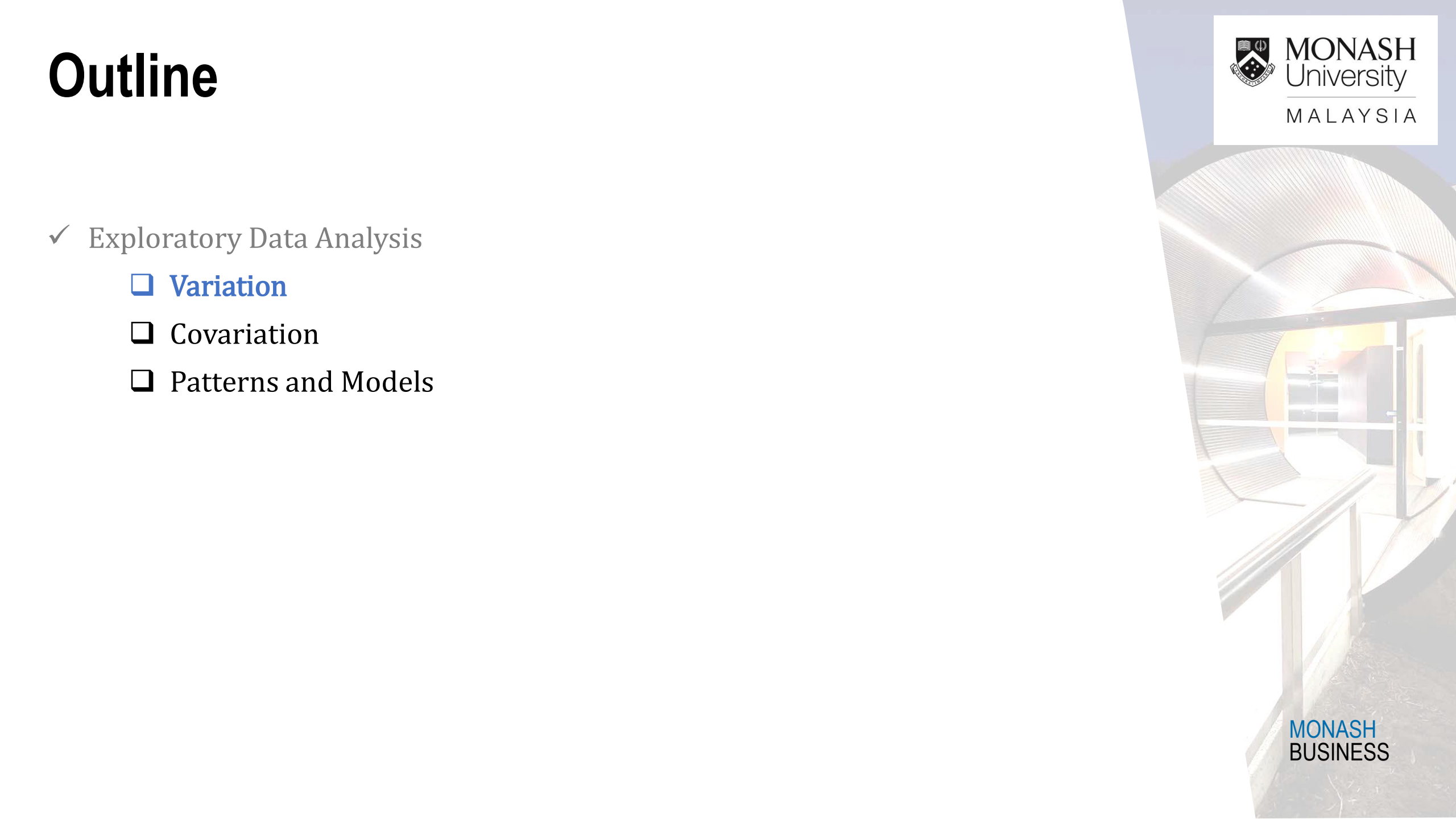
An observation is a set of measurements made under similar conditions (same time, same object). The observation will contain several values, each associated with a different variable. It is sometimes referred as a **data point**.

Tabular data is a set of values, each associated with a variable and an observation. It is **tidy** if each value is placed in its own "cell", each variable in its own column, and each observation in its own row.

As of now, you have been using data which is **tidy**. Remember that in real-life, most data **isn't tidy**.

Outline

- ✓ Exploratory Data Analysis
  - ❑ **Variation**
  - ❑ Covariation
  - ❑ Patterns and Models



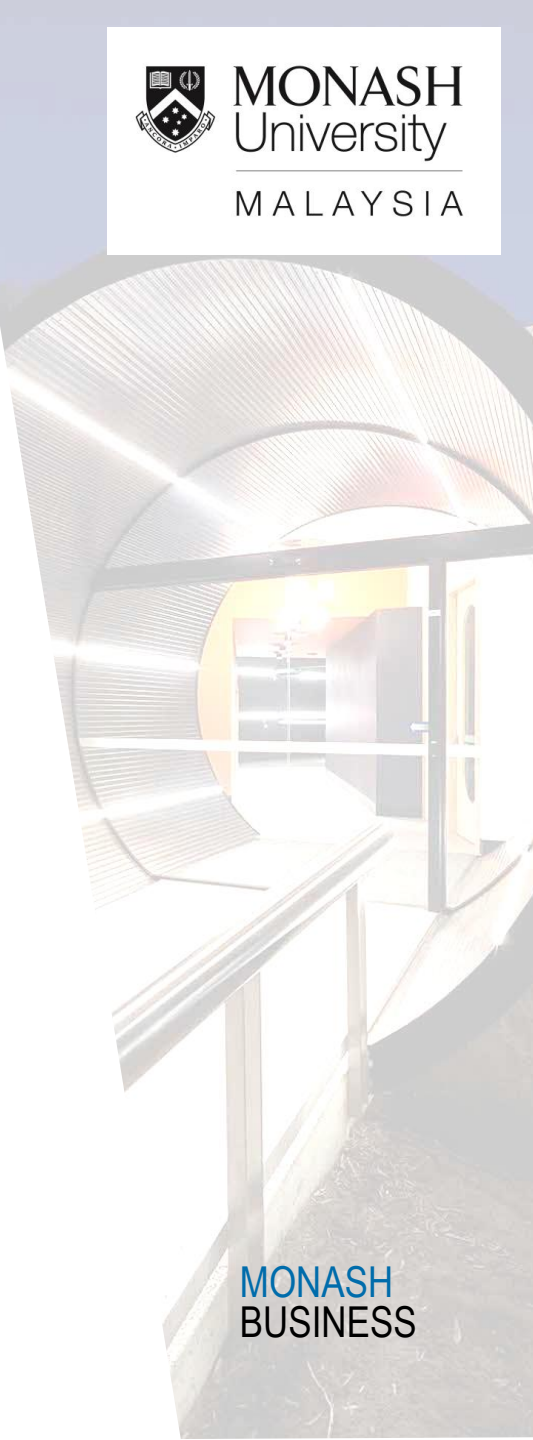
Variation

Variation is tendency of values of a variable to change from measurement to measurement.

- In real-life, if you measure any continuous variable **twice**, you will get two **different results**.
- It is **true** even when you measure quantities that are **constant**, like the speed of light.

Each measurements include a small amount of **error** that varies from measurement to measurement. Each variable has its own pattern of variation, which can reveal interesting information.

The best way to **understand** that pattern is to visualize the distribution of the variable's values.



MONASH University MALAYSIA

MONASH BUSINESS

diamonds

We will be using the diamonds dataset (53940 rows with 10 variables).

price: in US dollars (\$326 – \$18,823)
carat: weight of the diamond (0.2 – 5.01)
cut: quality (Fair, Good, Very Good, Premium, Ideal)
color: from D (best) to J (worst)
clarity: how clear the diamond, from I1 (worst) to IF (best)
x: length in mm (0 – 10.74)
y: width in mm (0 – 58.9)
z: depth in mm (0 – 31.8)
depth: total depth percentage = z / mean(x, y) = 2 * z / (x + y) (43 – 79)
table: width of top of diamond relative to widest point (43 – 95)

MONASH University MALAYSIA

MONASH BUSINESS

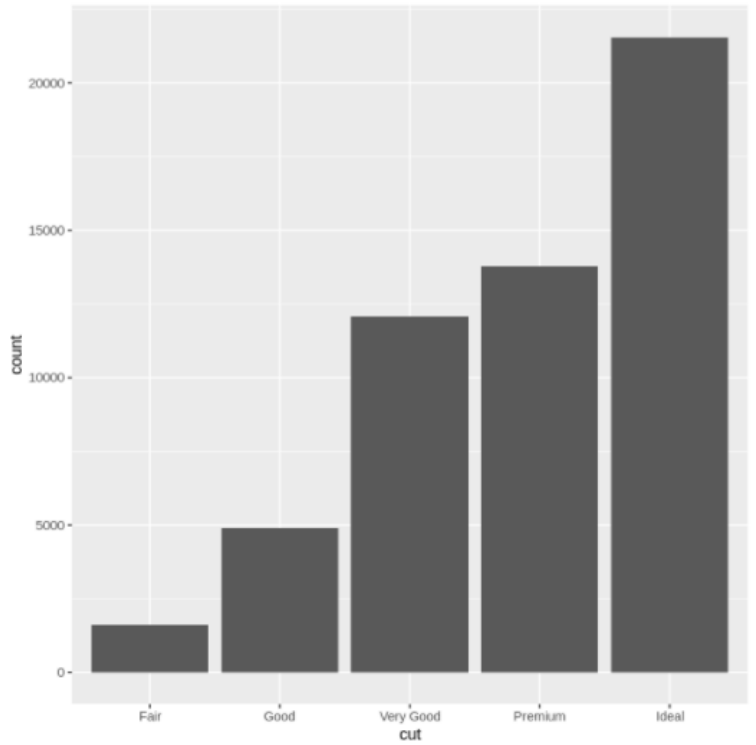
Visualizing distributions

How to visualize distribution of a variable, depends on whether it is categorical or **continuous**.

A variable is categorical if it can only take one of a small set of values. Categorical variables are usually saved as factors or character vectors.

A bar chart can be used to examine the distribution.

```
library(tidyverse)
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut))
```



MONASH University MALAYSIA

MONASH BUSINESS

Visualising distributions

The **height** of the bars displays **how many** observations occurred with each x value. Let's compute these values manually with `dplyr::count()`:

```
diamonds %>%
  count(cut)
```

A tibble: 5 × 2

cut	n
<ord>	<int>
Fair	1610
Good	4906
Very Good	12082
Premium	13791
Ideal	21551

MONASH University MALAYSIA

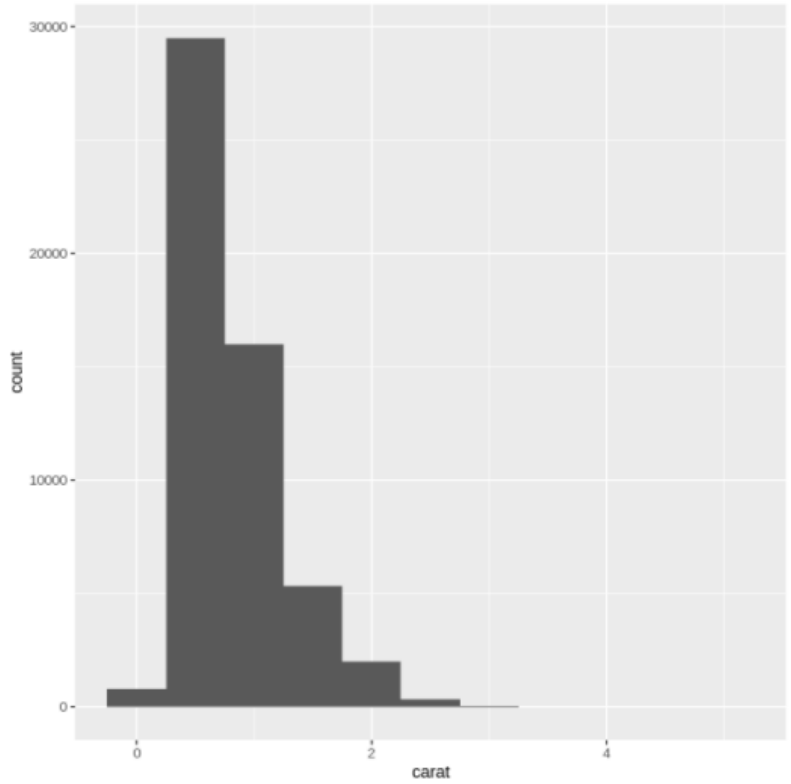
MONASH BUSINESS

Histogram

A variable is continuous if it can take any of an **infinite set** of ordered values. Numbers and date-times are two examples of continuous variables.

To examine the distribution of a continuous variable, use a histogram.

```
ggplot(data = diamonds) +
  geom_histogram(mapping = aes(x = carat), binwidth = 0.5)
```



MONASH University
MALAYSIA

MONASH
BUSINESS

Histogram

You can also compute by combining `dplyr::count()` and `ggplot2::cut_width()`:

```
diamonds %>%
  count(cut_width(carat, 0.5))
```

A tibble: 11 × 2

cut_width(carat, 0.5)	n
<fct>	<int>
[-0.25,0.25]	785
(0.25,0.75]	29498
(0.75,1.25]	15977
(1.25,1.75]	5313
(1.75,2.25]	2002
(2.25,2.75]	322
(2.75,3.25]	32
(3.25,3.75]	5
(3.75,4.25]	4
(4.25,4.75]	1
(4.75,5.25]	1

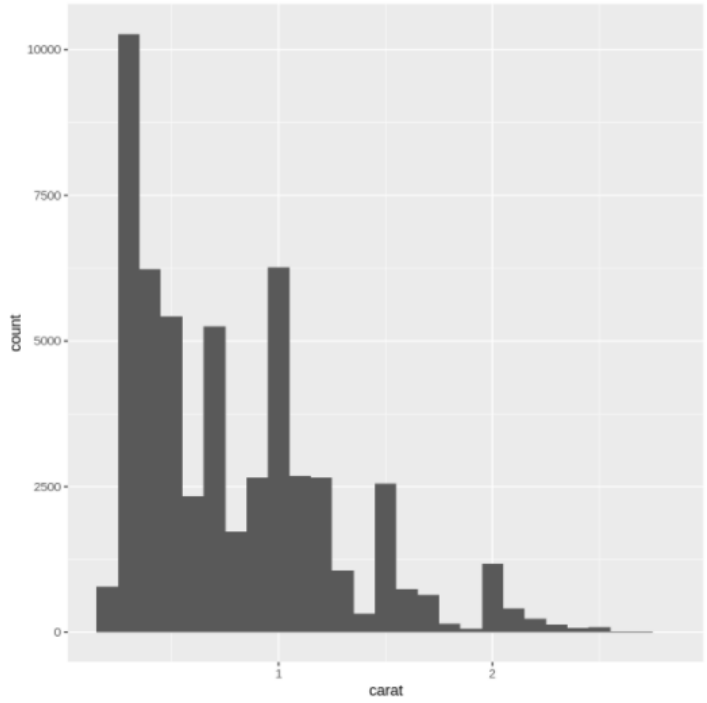
Histogram

A histogram divides the x-axis into equally spaced **bins** and then uses the **height** of a bar to display the **number of observations** that fall in each bin. In the previous graph, the tallest bar shows that almost 30,000 observations have a carat value between 0.25 and 0.75, which are the left and right edges of the bar (binwidth = 0.5).

We can set the width of the intervals in a histogram with the **binwidth** argument, which is measured in the units of the x variable. The graph on the right shows a smaller binwidth (0.1), and zooming into diamonds with a size of less than 3 carats.

```
smaller <- diamonds %>%
  filter(carat < 3)

ggplot(data = smaller, mapping = aes(x = carat)) +
  geom_histogram(binwidth = 0.1)
```



MONASH University MALAYSIA

MONASH BUSINESS

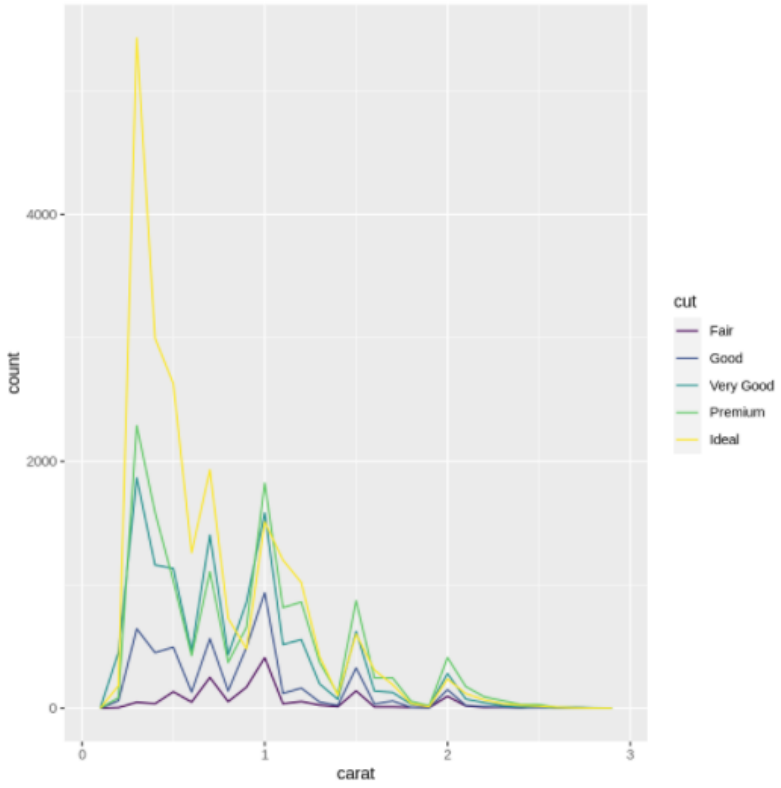
Histogram

To overlay **multiple histograms** in the same plot, it is recommended to use `geom_freqpoly()` instead of `geom_histogram()`.

`geom_freqpoly()` performs the same calculation as `geom_histogram()`, but instead of displaying the counts with bars, uses lines instead.

It's easier to understand overlapping lines than bars.

```
ggplot(data = smaller, mapping = aes(x = carat, colour = cut)) +
  geom_freqpoly(binwidth = 0.1)
```



MONASH University MALAYSIA

MONASH BUSINESS

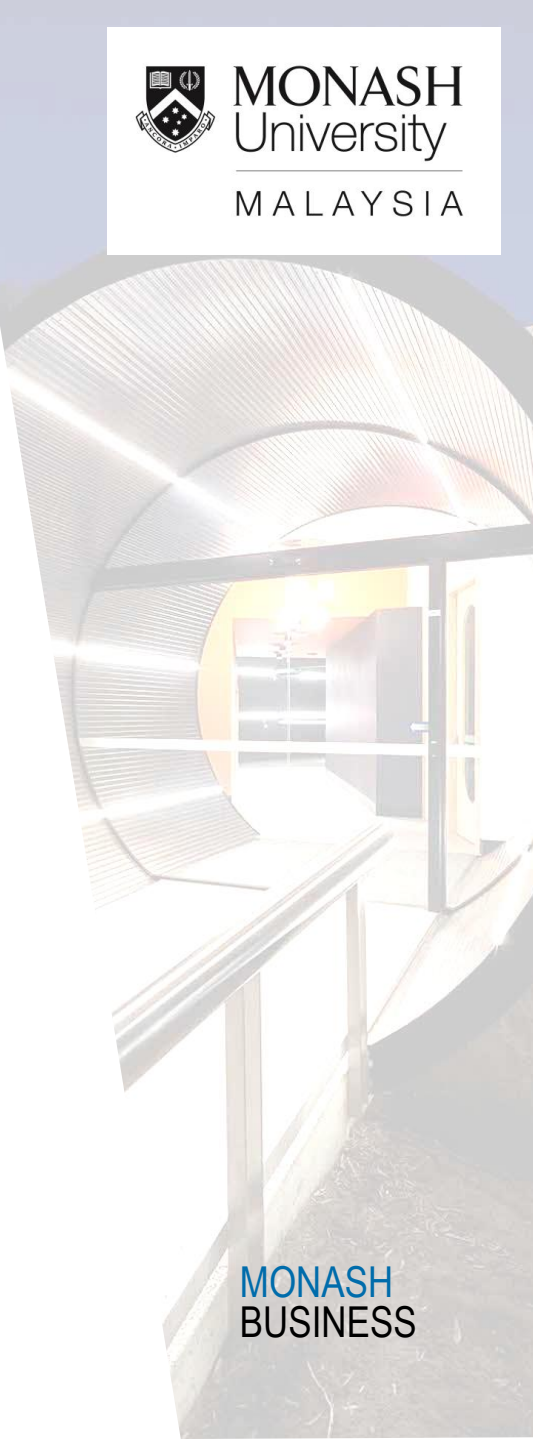
Variation

There are a few challenges associated with this type of plot.

Now that you can visualize **variation**, what should you look for in your plots? What type of follow-up questions should you ask?

Let's look at the following slides. The key to asking good follow-up questions:

- Your curiosity: what do you want to learn more about?
- Your skepticism: how could this be misleading?



Typical values

In both bar charts and histograms, **tall bars** show the **common** values of a variable, and **shorter bars** show **less-common** values. Places that do not have bars reveal values that were not seen in your data.

To turn this information into useful questions, look for anything unexpected:

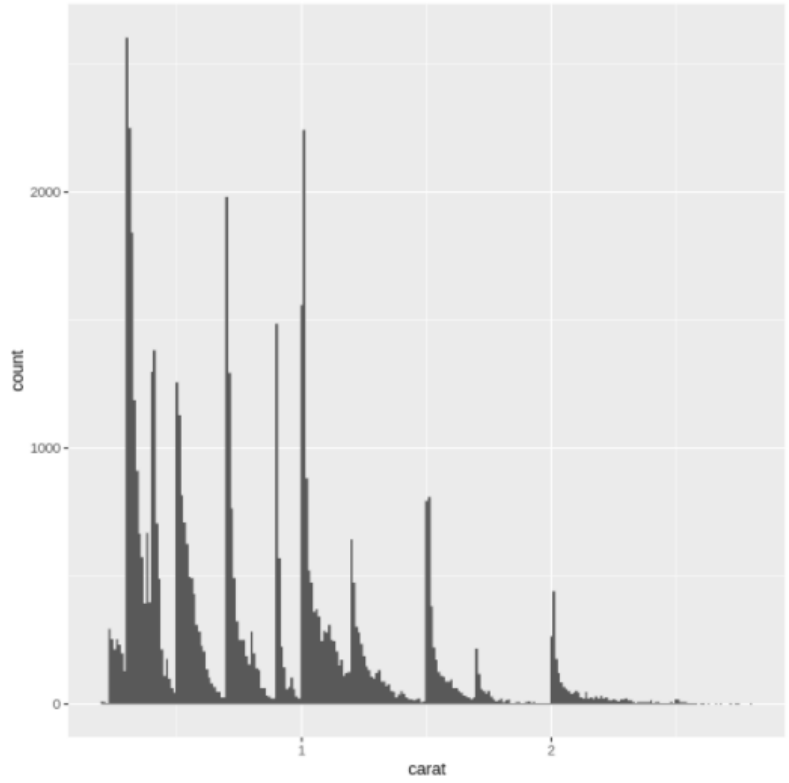
- Which values are the most common? Why?
- Which values are rare? Why? Does that match your expectations?
- Can you see any unusual patterns? What might explain them?

Typical values

As an example, the histogram shows several interesting questions:

- Why are there more diamonds at **whole carats** and **common fractions** of carats?
- Why are there more diamonds slightly to the **right** of each peak than there are slightly to the **left** of each peak?
- Why are there no diamonds bigger than **3 carats**?

```
ggplot(data = smaller, mapping = aes(x = carat)) +
  geom_histogram(binwidth = 0.01)
```



MONASH University MALAYSIA

MONASH BUSINESS

Clusters

Clusters of similar values suggest that subgroups exist in your data. To understand the subgroups, ask:

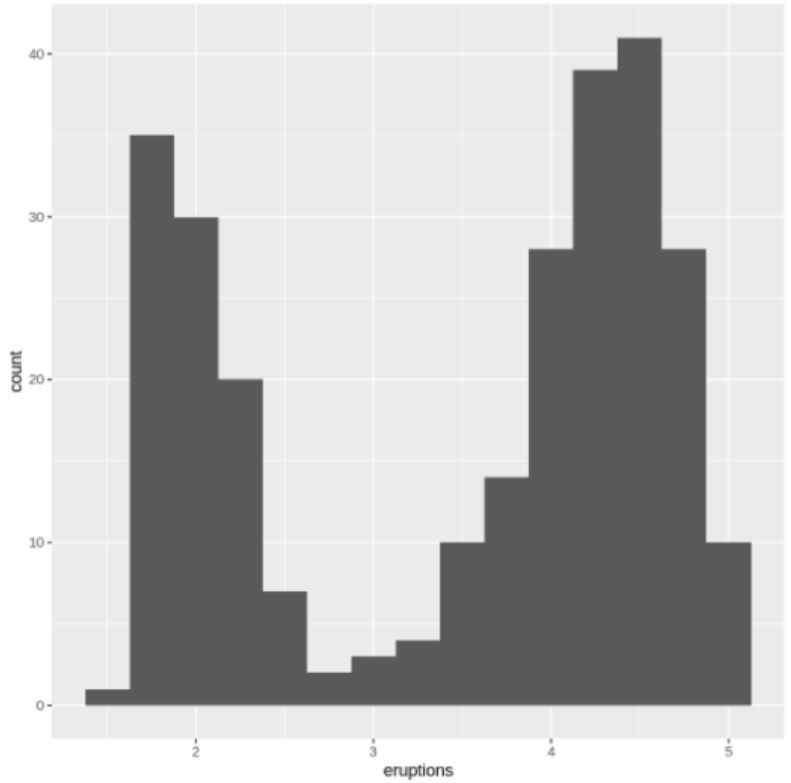
- How are the observations within **each** cluster **similar** to each other?
- How are the observations in **separate** clusters **different** from each other?
- How can you **explain** the clusters?
- Why might the **appearance** of clusters be **misleading**?

Clusters

The histogram below shows the length (in minutes) of 272 eruptions of the Old Faithful Geyser in Yellowstone National Park.

Eruption times appear to be clustered into two groups: short eruptions (~2 minutes) and long eruptions (4-5 minutes), but little in between.

```
ggplot(data = faithful, mapping = aes(x = eruptions)) +
  geom_histogram(binwidth = 0.25)
```



MONASH University MALAYSIA

MONASH BUSINESS

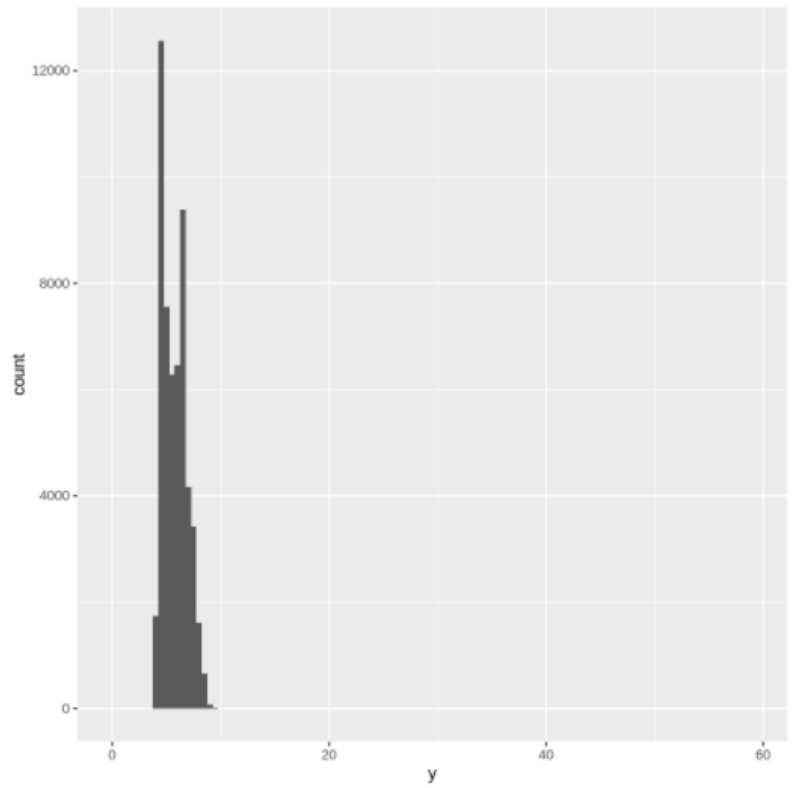
Unusual values

Outliers are observations that are unusual; data points that don't seem to fit the pattern.

- It can be data entry errors; other times outliers suggest important new science.

When you have a lot of data, outliers are sometimes difficult to see in a histogram. As an example, the outlier in histogram can be seen with the unusually wide limits on the x-axis.

```
ggplot(diamonds) +
  geom_histogram(mapping = aes(x = y), binwidth = 0.5)
```



MONASH University MALAYSIA

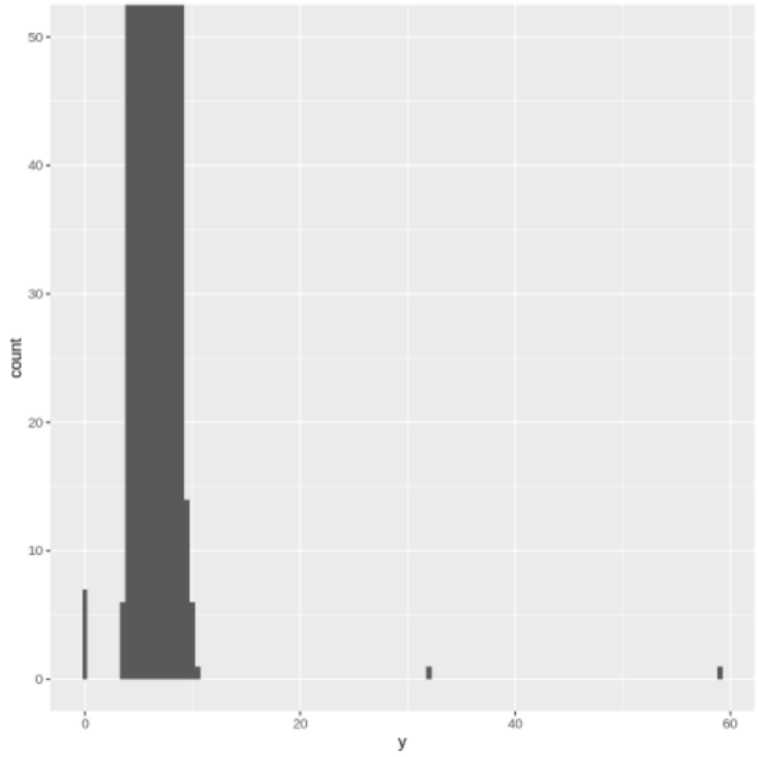
MONASH BUSINESS

Unusual values

There are many observations in the common bins that the **rare bins** are so short that you can't see them (though if you stare intently at 0 you'll spot something).

To make it easy to see the **unusual values**, we need to **zoom** to small values of the y-axis with `coord_cartesian()`.

```
ggplot(diamonds) +
  geom_histogram(mapping = aes(x = y), binwidth = 0.5) +
  coord_cartesian(ylim = c(0, 50))
```



MONASH University MALAYSIA

MONASH BUSINESS

Cartesian

`coord_cartesian()` also has an `xlim()` argument for when you need to zoom into the x-axis.

ggplot2 also has `xlim()` and `ylim()` functions that work slightly differently: they throw away the data outside the limits.

This allows us to see that there are three **unusual values**: 0, ~30, and ~60. We can show them using dplyr.

```
unusual <- diamonds %>%
  filter(y < 3 | y > 20) %>%
  select(price, x, y, z) %>%
  arrange(y)
unusual
```

A tibble: 9 × 4

price	x	y	z
<int>	<dbl>	<dbl>	<dbl>
5139	0.00	0.0	0.00
6381	0.00	0.0	0.00
12800	0.00	0.0	0.00
15686	0.00	0.0	0.00
18034	0.00	0.0	0.00
2130	0.00	0.0	0.00
2130	0.00	0.0	0.00
2075	5.15	31.8	5.12
12210	8.09	58.9	8.06

Results

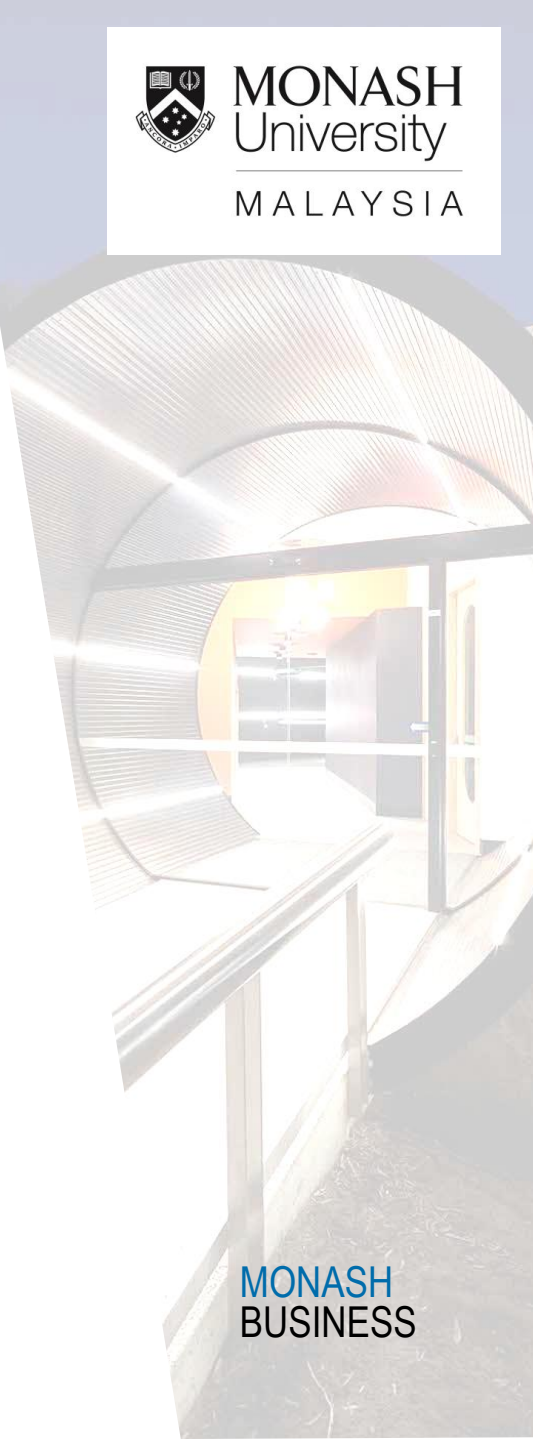
The y variable measures one of the three dimensions of these diamonds, in mm. Diamonds cannot have a width of **0mm**, so these values must be **incorrect**. We might also suspect that measurements of **32mm** and **59mm** are **implausible**: those diamonds are >1" long, but don't cost hundreds of thousands of dollars!

It's good practice to **repeat** your analysis with and without the **outliers**.

- If they have minimal effect on the results, and you can't figure out why they're there, it's reasonable to replace them with missing values, and move on.
- However, if they have a substantial effect on your results, you shouldn't drop them without justification. You'll need to **figure out** what caused them (e.g. a data entry error) and **disclose** that you removed them in your write-up.

Outline

- ✓ Exploratory Data Analysis
  - ✓ Variation
  - ❑ **Covariation**
  - ❑ Patterns and Models



MONASH University MALAYSIA

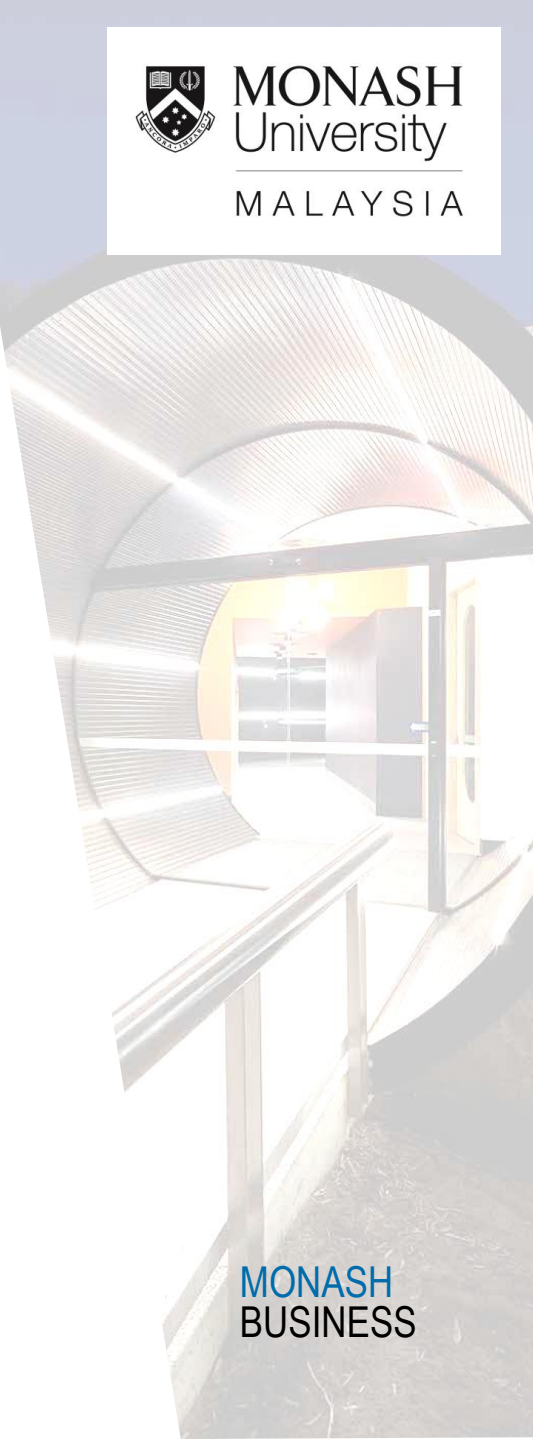
MONASH BUSINESS

Covariation

If **variation** describes the behavior **within** a variable, covariation describes the behavior between variables.

Covariation is the tendency for the values of two or more variables to **vary** together in a **related way**.

The best way to spot covariation is to visualize the relationship between two or more variables. How to do it all depends on the type of variables involved.



MONASH University MALAYSIA

MONASH BUSINESS

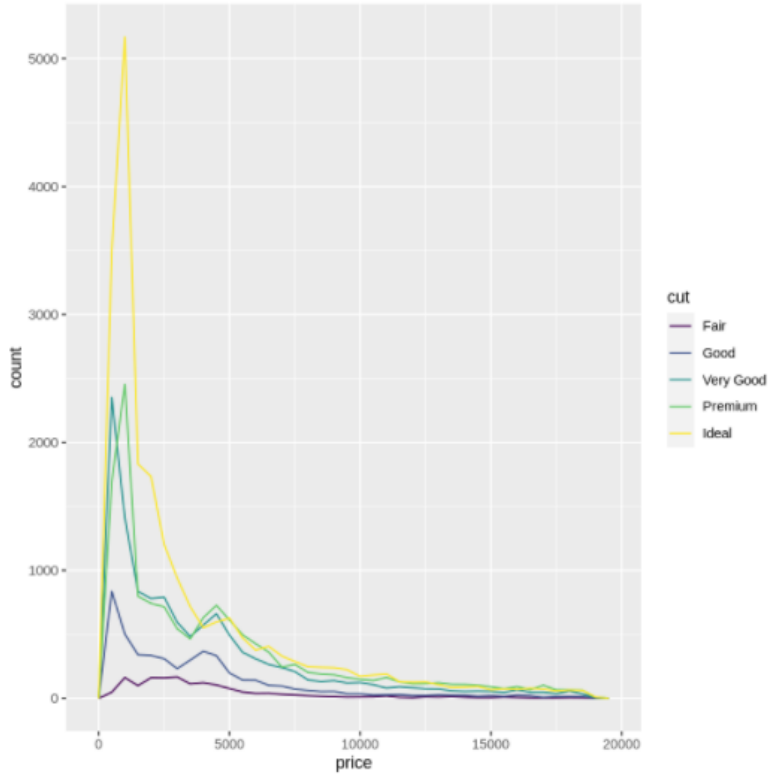
A categorical and continuous variable

It is normal to explore distributions of continuous variable broken down by a categorical variable, as in the previous frequency polygon.

The default appearance of `geom_freqpoly()` is not that useful for that sort of comparison because the height is given by the count.

As an example, let's explore how the **price** of a diamond varies with its **quality**.

```
ggplot(data = diamonds, mapping = aes(x = price)) +
  geom_freqpoly(mapping = aes(colour = cut), binwidth = 500)
```



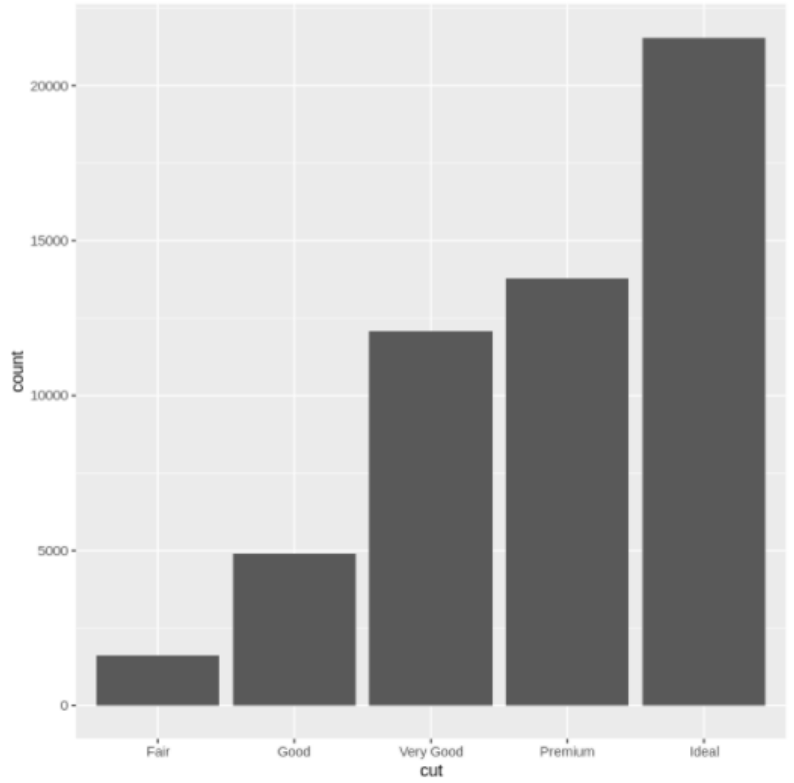
MONASH University MALAYSIA

MONASH BUSINESS

A categorical and continuous variable

It's **hard** to see the **difference** in distribution because the overall counts differ so much.

```
ggplot(diamonds) +
  geom_bar(mapping = aes(x = cut))
```

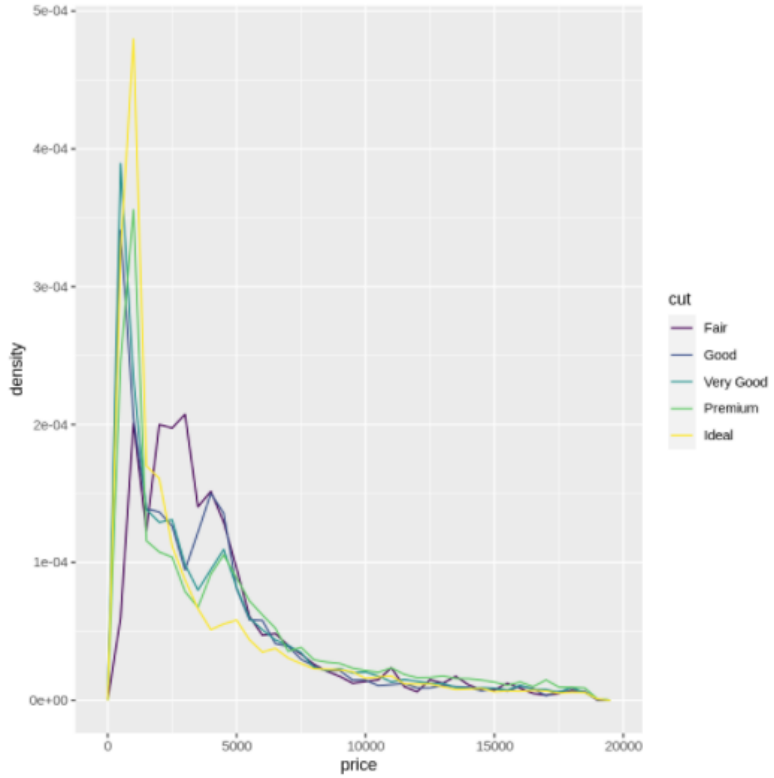


Comparison

As to make an simpler comparison, we can swap what is displayed on the y-axis.

Instead of displaying count, we'll display density, which is the count standardized so that the area under each frequency polygon is one.

```
ggplot(data = diamonds, mapping = aes(x = price, y = ..density..)) +
  geom_freqpoly(mapping = aes(colour = cut), binwidth = 500)
```



MONASH University MALAYSIA

MONASH BUSINESS

Boxplot

There's something rather surprising about this plot: it appears that **fair diamonds** (the lowest quality) have the **highest average price**! But maybe that's because frequency polygons are a little hard to interpret: there's a lot going on in this plot.

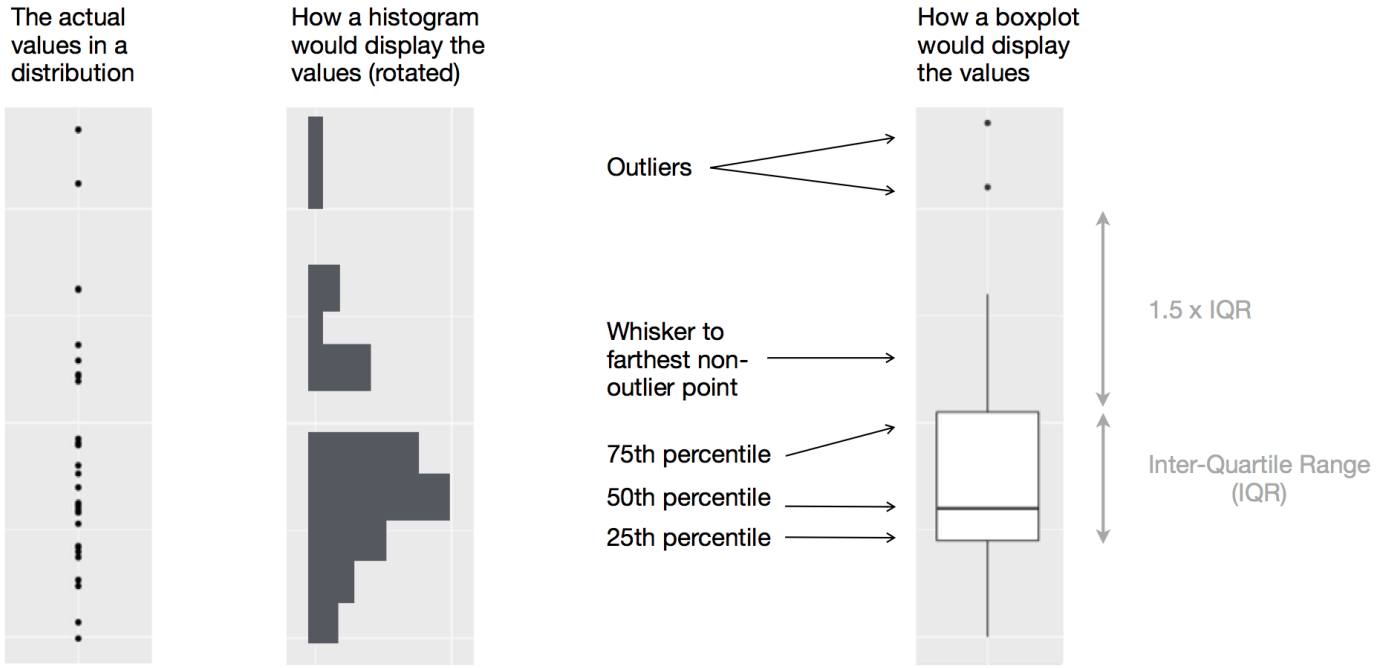
Another way to display the distribution is using boxplot. Boxplot is another alternative to display the distribution of a continuous variable broken down by a categorical variable is the boxplot. It is a type of visual shorthand for a distribution of values that is popular among statisticians. It consists of:

- Box that stretches from the 25th percentile of the distribution to the 75th percentile, a distance known as the **interquartile range** (IQR),
- Middle of the box is a line that displays the **median**, i.e. 50th percentile, of the distribution.

These three lines give you a sense of the spread of the distribution and whether or not the distribution is symmetric about the median or skewed to one side.

Boxplot

Visual points that display observations that fall more than 1.5 times the IQR from either edge of the box. These outlying points are unusual so are plotted individually. A line (or whisker) that extends from each end of the box and goes to the farthest non-outlier point in the distribution.



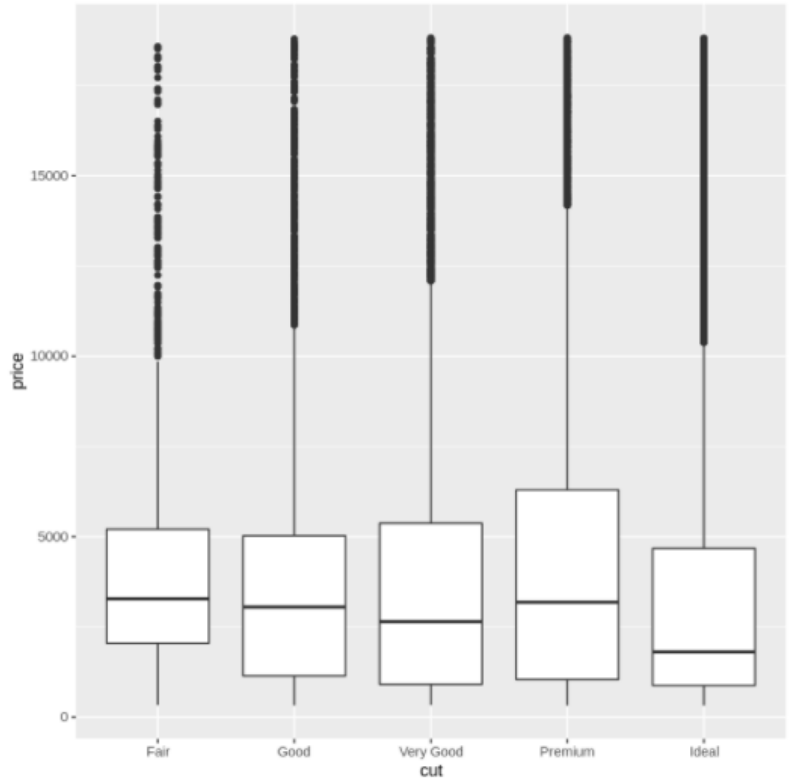
MONASH University MALAYSIA

MONASH BUSINESS

Boxplot

Take a look at the distribution of **price by cut** using `geom_boxplot()`.

```
ggplot(data = diamonds, mapping = aes(x = cut, y = price)) +
  geom_boxplot()
```



MONASH University
MALAYSIA

MONASH
BUSINESS

Boxplot

While we get less information on the distribution, boxplots are much compact, so we can easily compare them.

It supports the contradictory finding that better quality diamonds are cheaper on average!

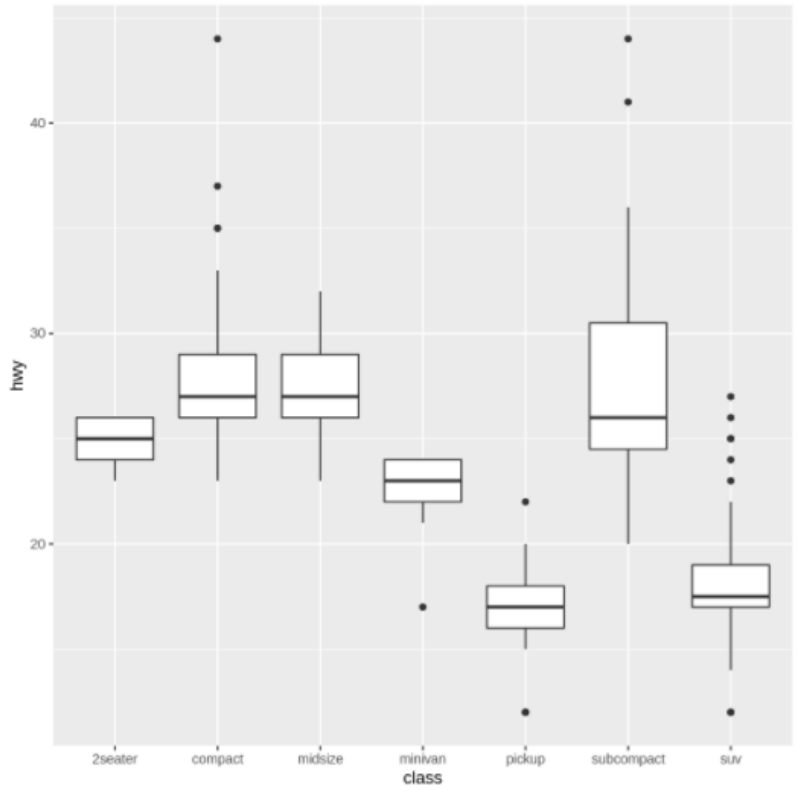
cut is an ordered factor: fair is worse than good, which is worse than very good and so on. Many categorical variables don't have such an intrinsic order, so you might want to **reorder them** to make a more informative display. One way to do that is with the `reorder()` function.

Trend

As an example, take the class variable in the mpg dataset.

Let's look at how **highway mileage** varies across classes.

```
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) +
  geom_boxplot()
```



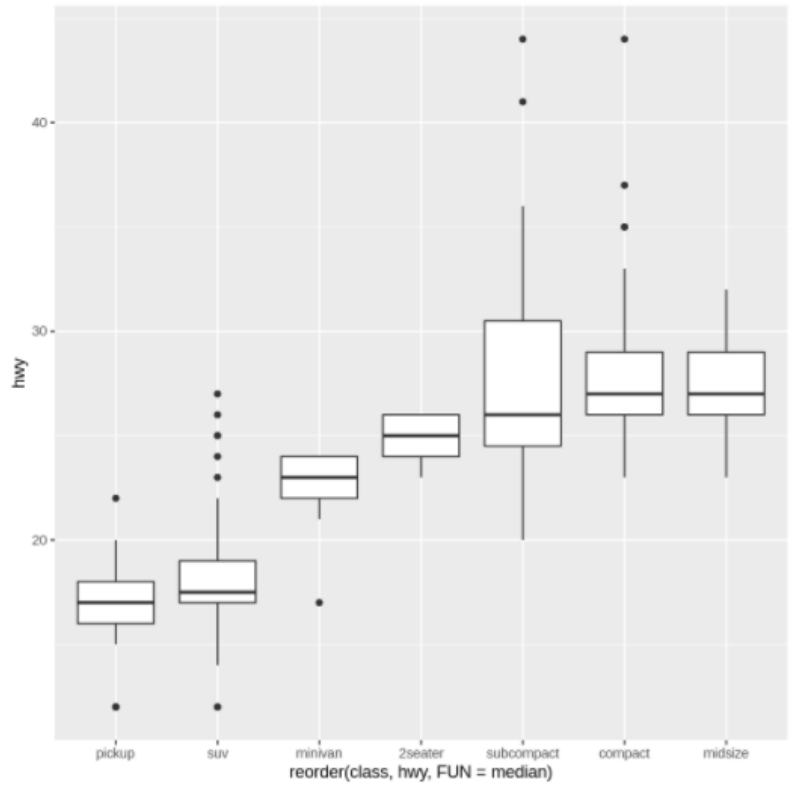
MONASH University MALAYSIA

MONASH BUSINESS

Trend

To make the trend easier to see, we can reorder class based on the median value of hwy.

```
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(x = reorder(class, hwy, FUN = median), y = hwy))
```



MONASH University MALAYSIA

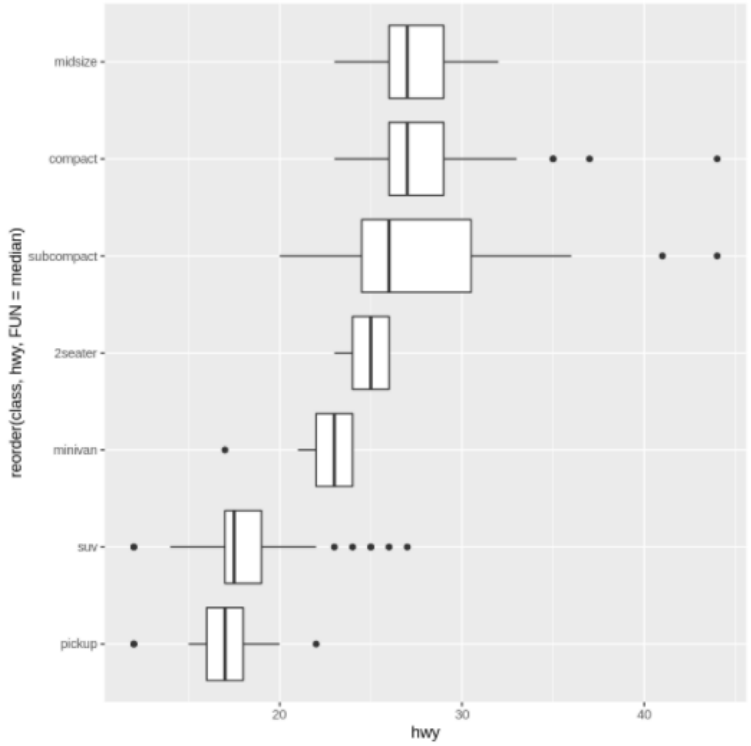
MONASH BUSINESS

Flip

Should we have **long variable names**, `geom_boxplot()` will work better if you flip it 90°.

We can do it using `coord_flip()`.

```
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(x = reorder(class, hwy, FUN = median), y = hwy)) +
  coord_flip()
```

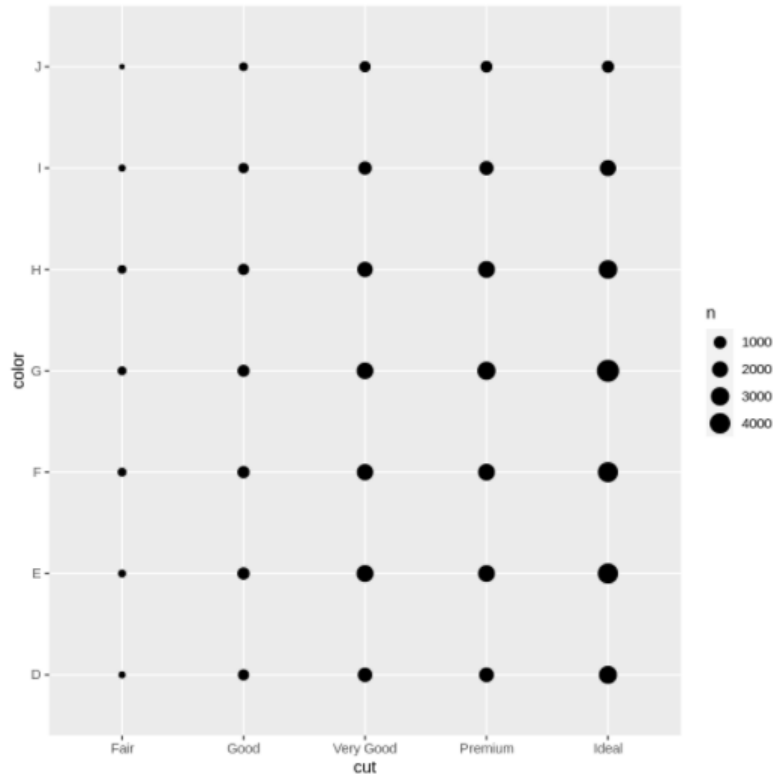


Two categorical variables

To **visualize** the covariation between categorical variables, you'll need to **count** the number of observations for each combination.

One way to do that is to rely on the built-in `geom_count()`.

```
ggplot(data = diamonds) +
  geom_count(mapping = aes(x = cut, y = color))
```



MONASH University MALAYSIA

MONASH BUSINESS

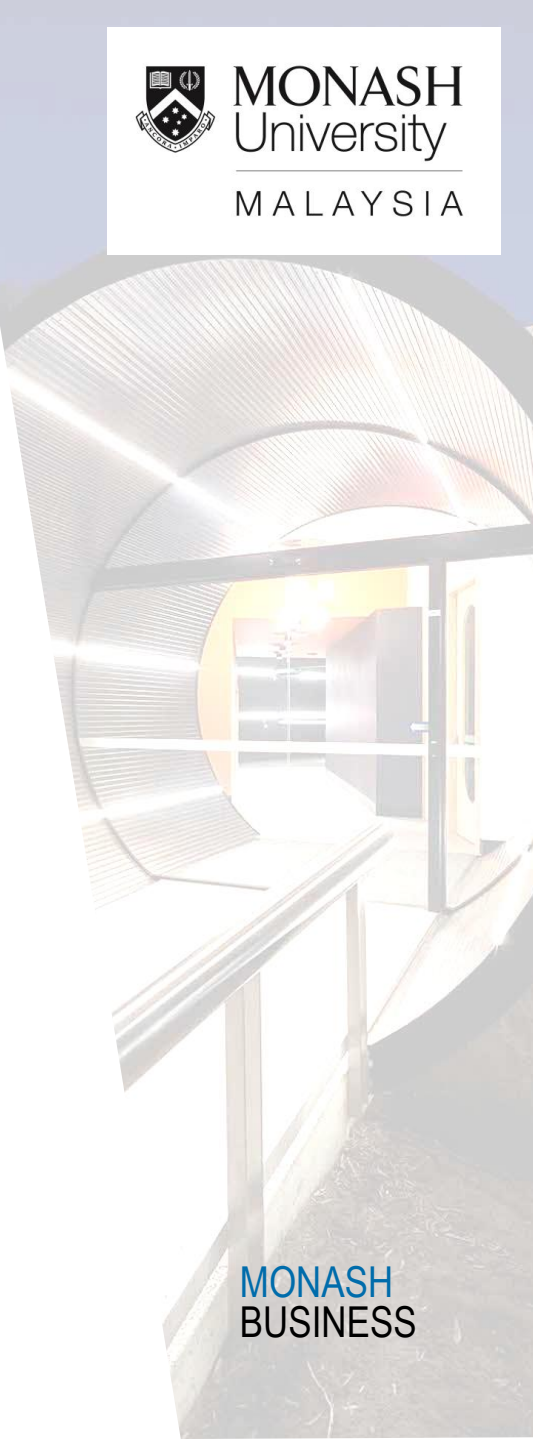
Count

Size of each circle in the plot displays number of observations occurred at each combination of values. **Covariation** will appear as a strong correlation between specific x values and specific y values. To compute count with dplyr:

```
diamonds %>%
  count(color, cut)
```

A tibble: 35 × 3

color	cut	n
<ord>	<ord>	<int>
D	Fair	163
D	Good	662
D	Very Good	1513
D	Premium	1603
D	Ideal	2834
E	Fair	224
E	Good	933



MONASH University MALAYSIA

MONASH BUSINESS

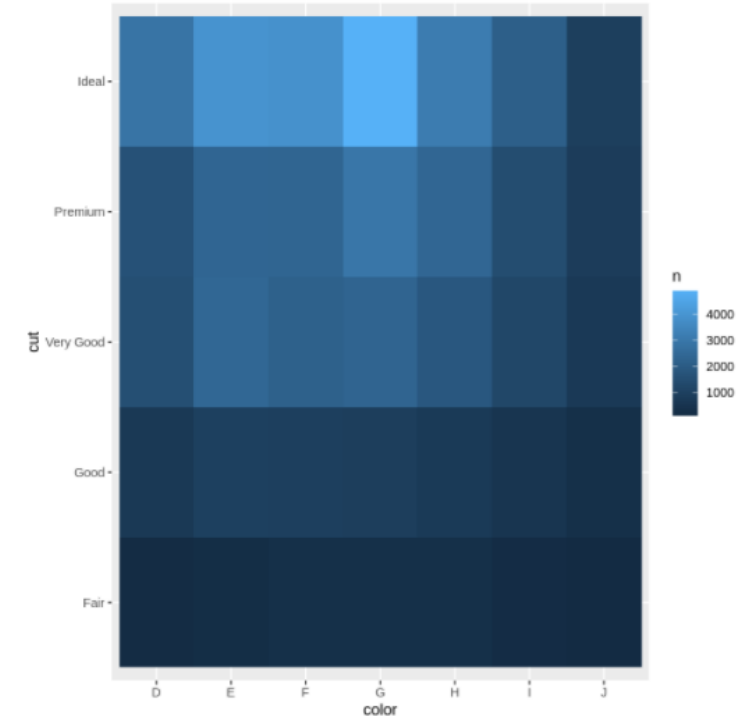
Visualise

Let's visualize with `geom_tile()` and the fill aesthetic.

If the categorical variables are **unordered**, you might want to use the **seriation package** to simultaneously reorder the rows and columns in order to more clearly reveal interesting patterns.

For larger plots, you might want to try the **d3heatmap** or **heatmaply** packages, which create interactive plots.

```
diamonds %>%
  count(color, cut) %>%
  ggplot(mapping = aes(x = color, y = cut)) +
    geom_tile(mapping = aes(fill = n))
```



MONASH University MALAYSIA

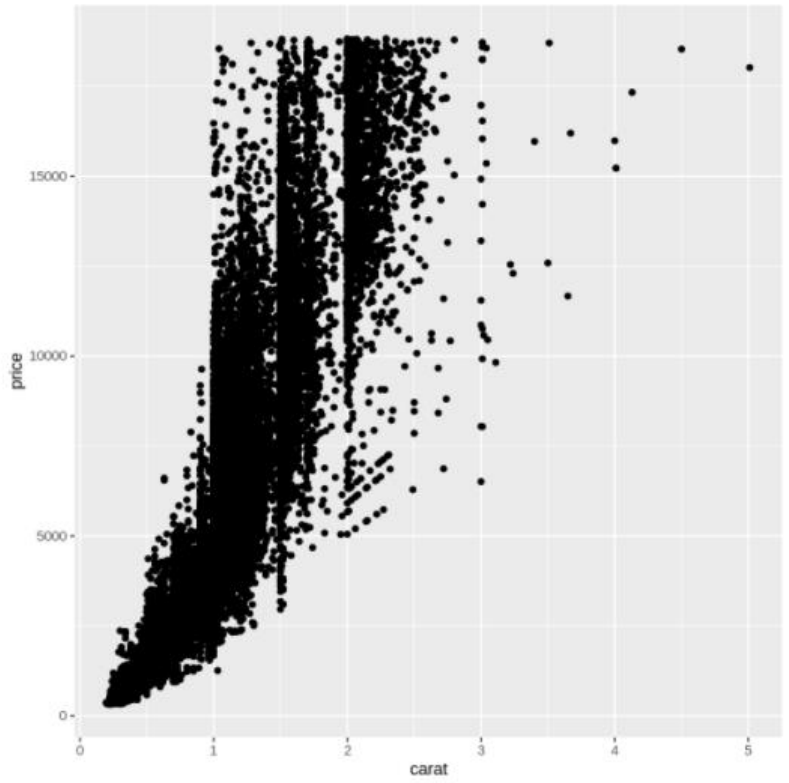
MONASH BUSINESS

Two continuous variables

We've seen a great way to visualize the **covariation** between **two continuous variables**: draw a scatterplot with `geom_point()`.

We can see covariation as a pattern in the points. As an example, we see an **exponential** relationship between the **carat size** and **price** of a diamond.

```
ggplot(data = diamonds) +
  geom_point(mapping = aes(x = carat, y = price))
```



MONASH University MALAYSIA

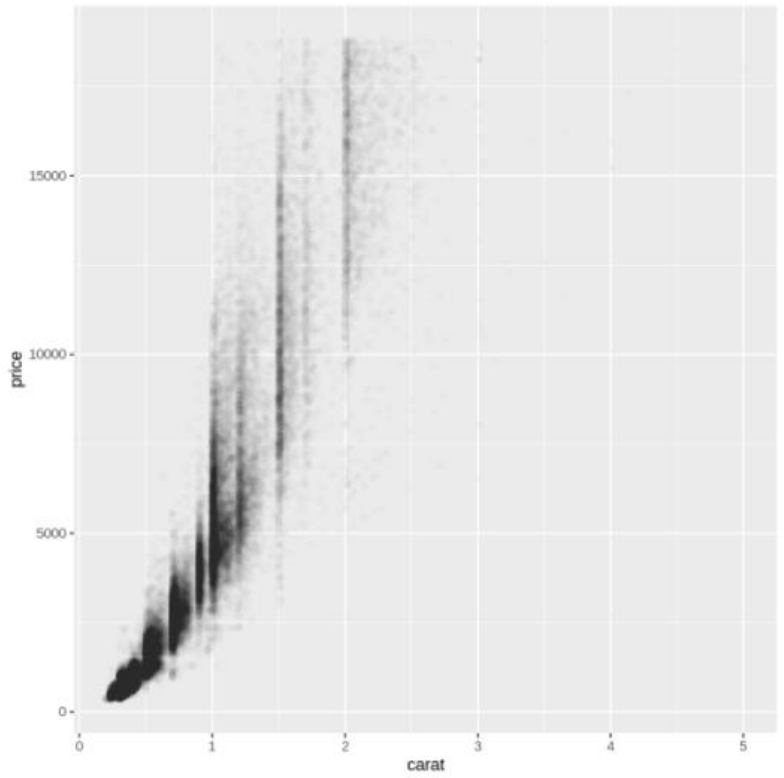
MONASH BUSINESS

Scatterplots

Scatterplots are less useful as size of your dataset grows, as points begin to overplot, and pile up into areas of uniform black.

We've already seen one way to fix the problem: using the **alpha aesthetic** to add **transparency**.

```
ggplot(data = diamonds) +
  geom_point(mapping = aes(x = carat, y = price), alpha = 1 / 100)
```



MONASH University MALAYSIA

MONASH BUSINESS

Scatterplots

However, using transparency can be difficult for very large datasets. An alternative is to use **bin**. Previously we used `geom_histogram()` and `geom_freqpoly()` to bin in one dimension.

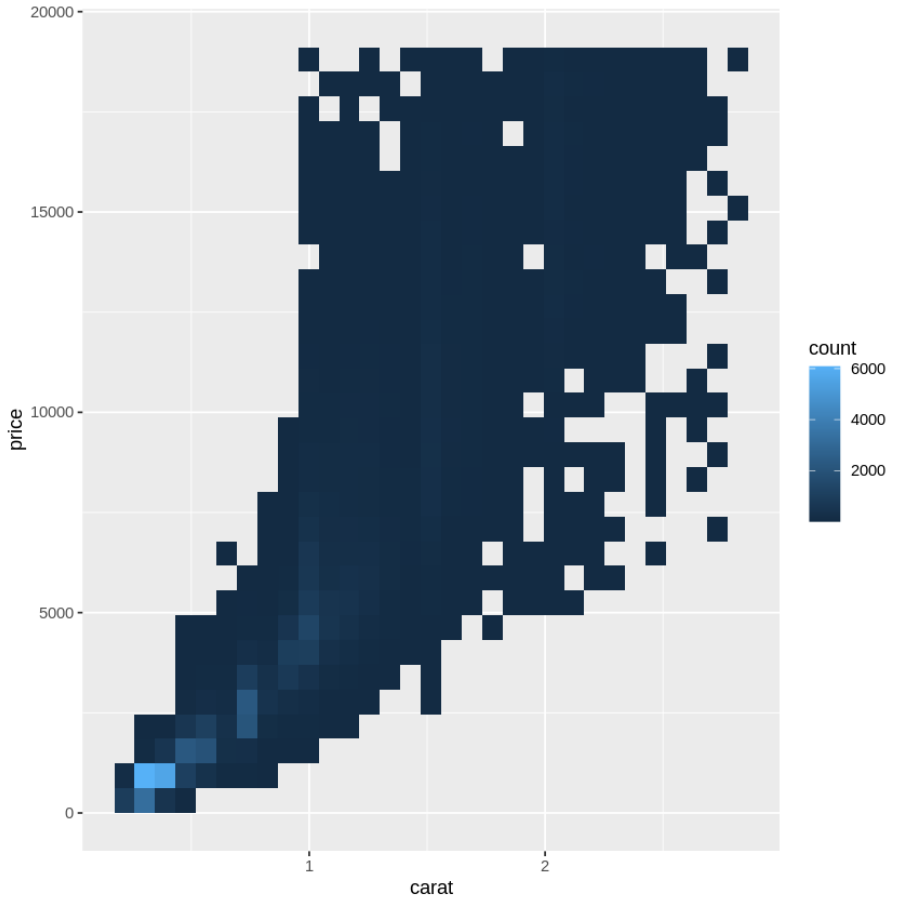
`geom_bin2d()` and `geom_hex()` divide the coordinate plane into **2-dimension bins** and then use a fill color to display how many points fall into each bin.

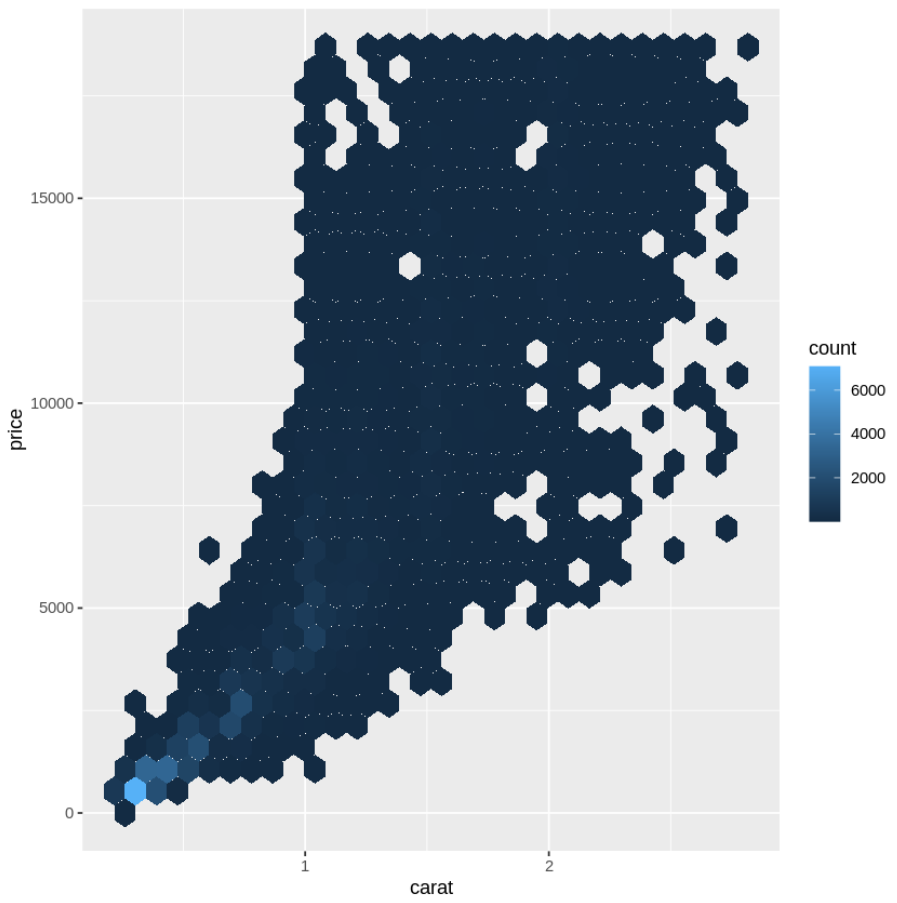
- `geom_bin2d()` creates rectangular bins,
- `geom_hex()` creates hexagonal bins (hexbin package is needed).

```
ggplot(data = smaller) +
  geom_bin2d(mapping = aes(x = carat, y = price))

install.packages("hexbin")
ggplot(data = smaller) +
  geom_hex(mapping = aes(x = carat, y = price))
```

Scatterplots





MONASH University
MALAYSIA

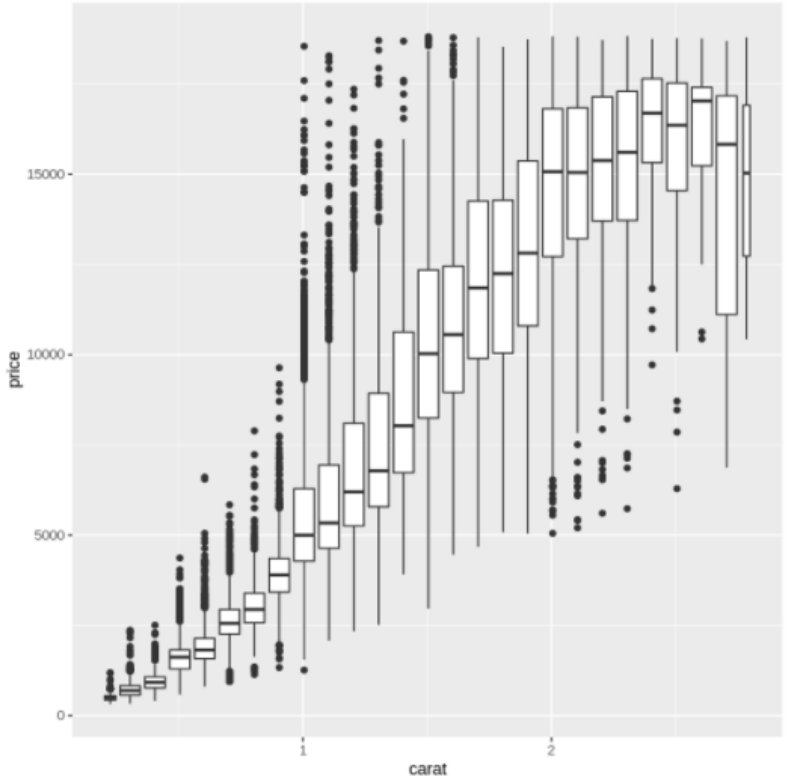
MONASH
BUSINESS

Boxplot

Another option is to **bin** one continuous variable so it acts like a categorical variable. We then use one of the techniques for **visualizing** the combination of a categorical and a continuous variable that you learned about.

As an example, we can **bin** **carat** and then for each group, display a **boxplot**.

```
ggplot(data = smaller, mapping = aes(x = carat, y = price)) +
  geom_boxplot(mapping = aes(group = cut_width(carat, 0.1)))
```



MONASH University MALAYSIA

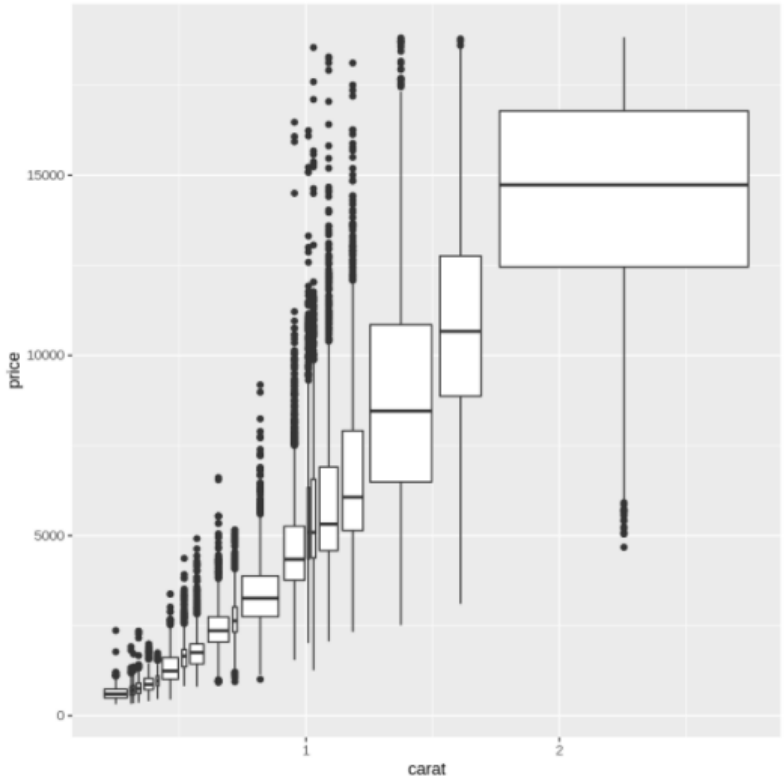
MONASH BUSINESS

Boxplot

`cut_width(x, width)` divides x into bins of width. **Boxplots** look roughly the **same** regardless of how many observations there are, so it's difficult to tell that each boxplot summarizes a different number of points.

- One way is to make the width of the boxplot proportional to the number of points with `varwidth = TRUE`.
- Another way uses `cut_number()` to display approximately the same number of points in each bin.

```
ggplot(data = smaller, mapping = aes(x = carat, y = price)) +
  geom_boxplot(mapping = aes(group = cut_number(carat, 20)))
```

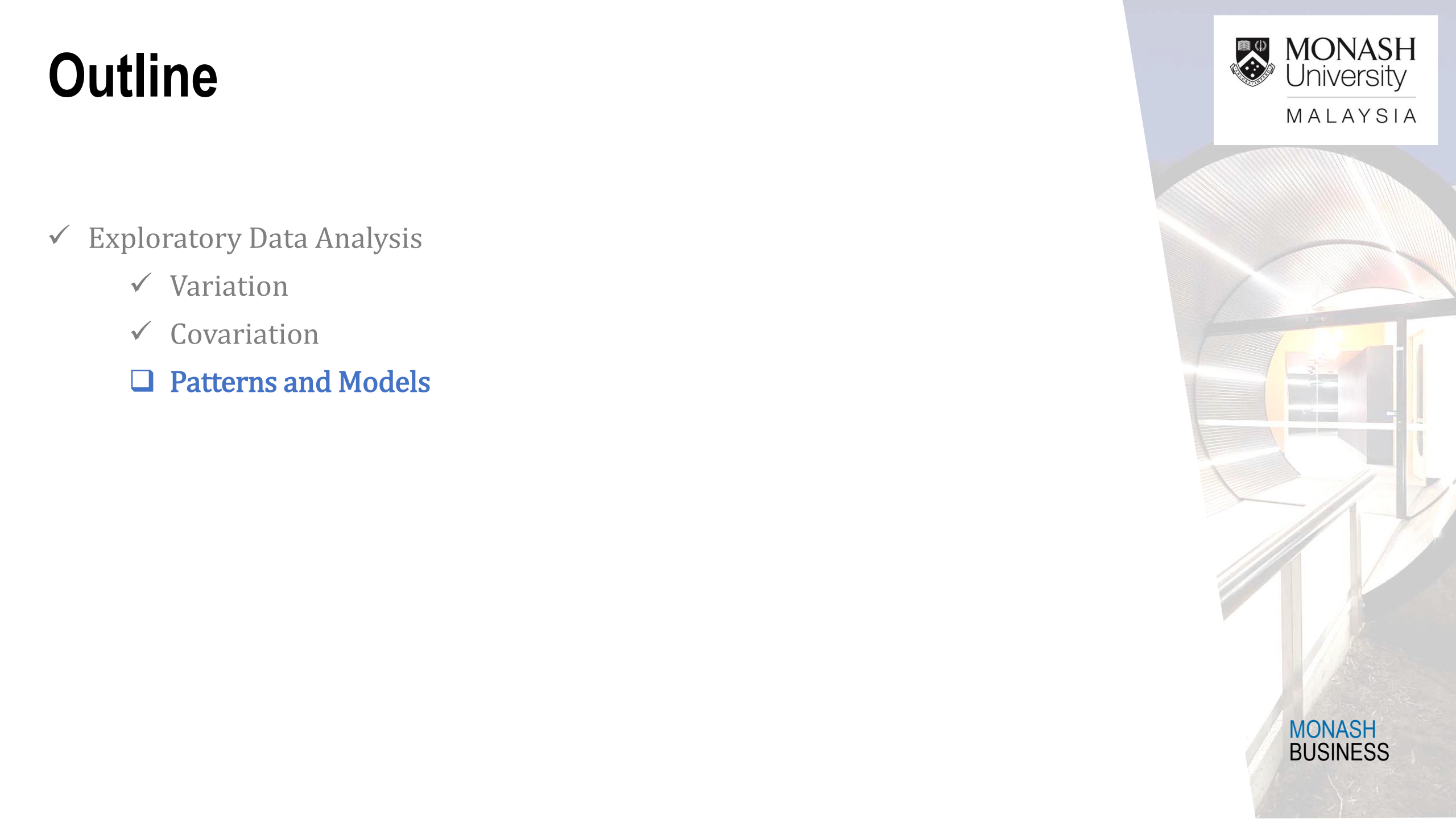


MONASH University MALAYSIA

MONASH BUSINESS

Outline

- ✓ Exploratory Data Analysis
  - ✓ Variation
  - ✓ Covariation
  - ❑ **Patterns and Models**



MONASH University MALAYSIA

MONASH BUSINESS

Patterns and models

Patterns in the data provide clues about relationships. When a systematic relationship exists between two variables it will appear as a pattern in the data. If you spot a pattern, ask yourself:

- Could this pattern be due to **coincidence** (i.e. random chance)?
- How can you **describe** the relationship implied by the pattern?
- How **strong** is the relationship implied by the pattern?
- What **other variables** might affect the relationship?
- Does the **relationship change** if you look at individual subgroups of the data?

MONASH University MALAYSIA

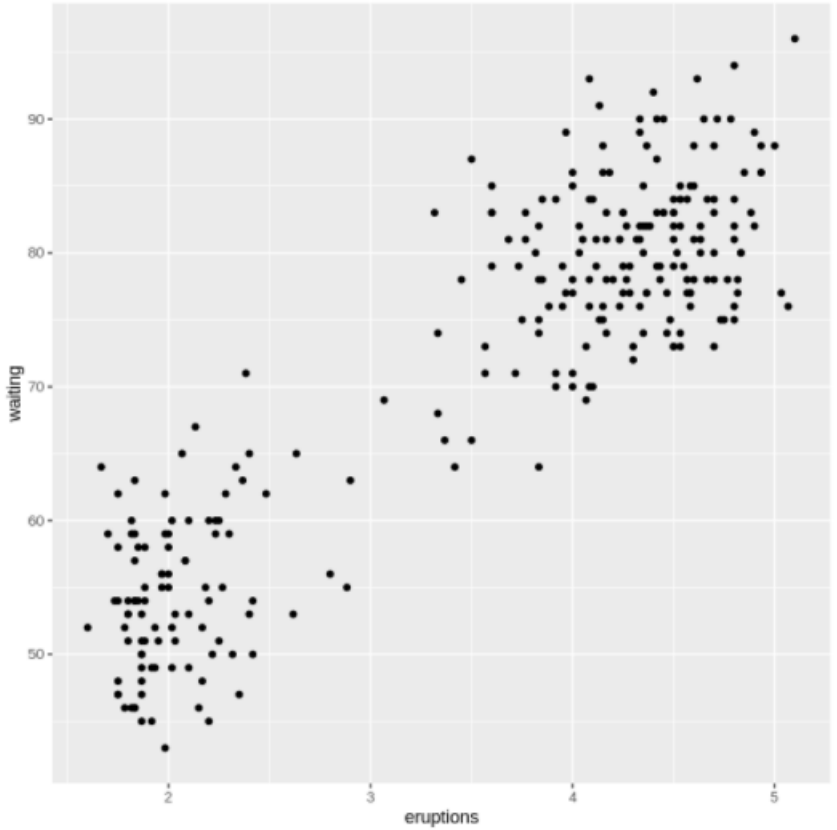
MONASH BUSINESS

Patterns and models

A scatterplot of Old Faithful eruption lengths versus the wait time between eruptions shows a pattern: **longer wait times** are associated with **longer eruptions**.

The scatterplot also displays the two clusters.

```
ggplot(data = faithful) +
  geom_point(mapping = aes(x = eruptions, y = waiting))
```



MONASH University MALAYSIA

MONASH BUSINESS

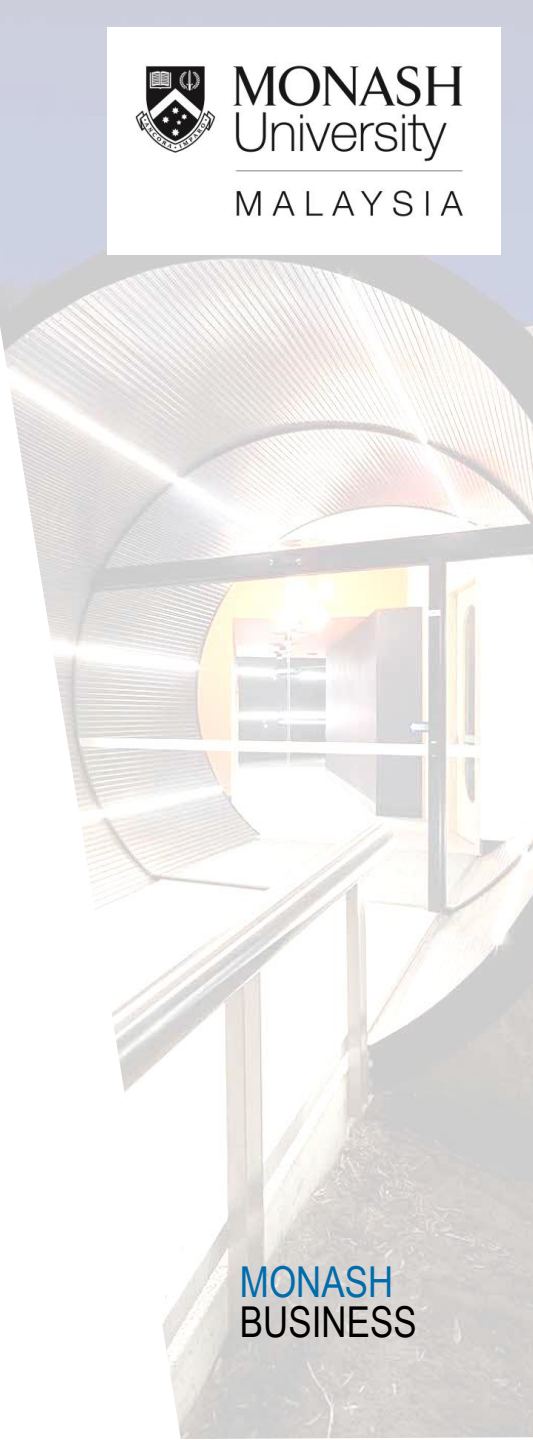
Patterns and models

Patterns provide one of the most useful tools for data scientists because they reveal covariation.

- Variation as a phenomenon that creates **uncertainty**,
- Covariation is a phenomenon that **reduces** it.

If two variables covary, you can use the values of one variable to make better predictions about the values of the second.

If the covariation is due to a causal relationship (a special case), then you can use the value of one variable to **control** the value of the **second**.



MONASH University MALAYSIA

MONASH BUSINESS

Patterns and models

Models are a tool for extracting patterns out of data.

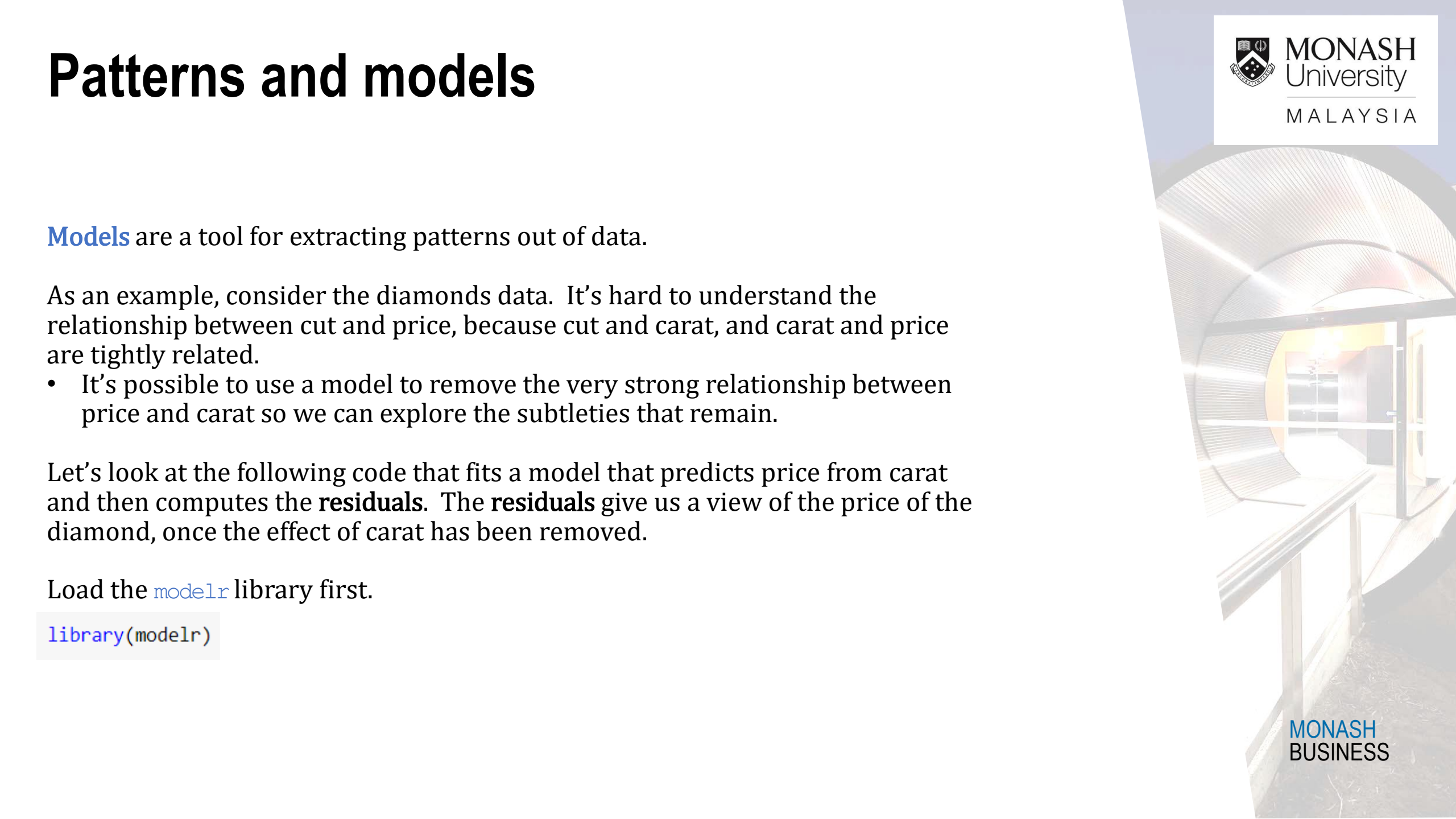
As an example, consider the diamonds data. It's hard to understand the relationship between cut and price, because cut and carat, and carat and price are tightly related.

- It's possible to use a model to remove the very strong relationship between price and carat so we can explore the subtleties that remain.

Let's look at the following code that fits a model that predicts price from carat and then computes the **residuals**. The **residuals** give us a view of the price of the diamond, once the effect of carat has been removed.

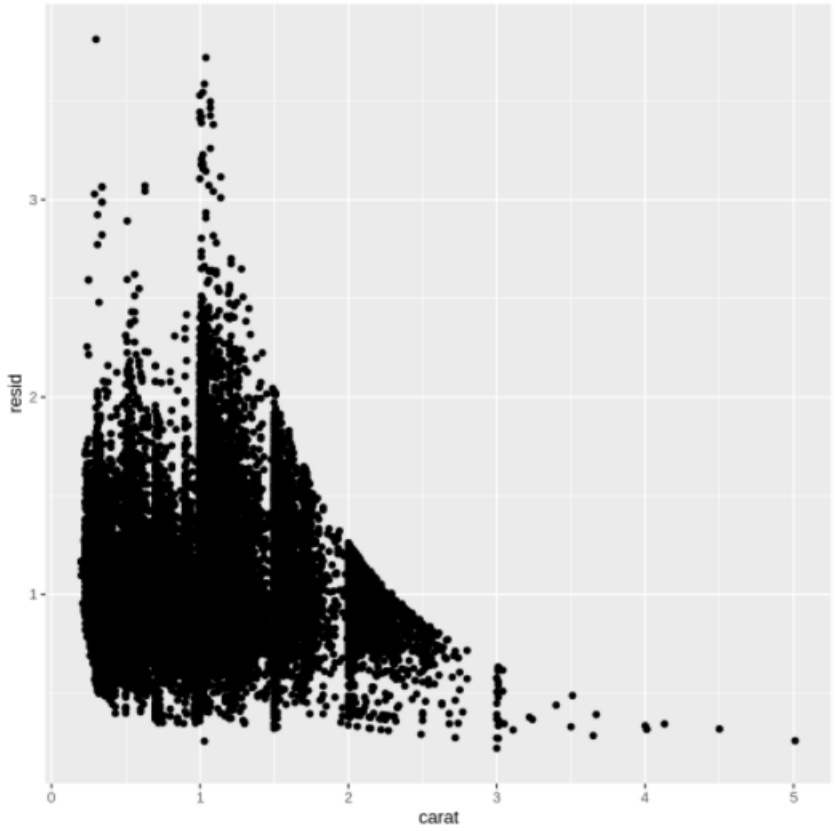
Load the `modelr` library first.

```
library(modelr)
```



Patterns and models

```
mod <- lm(log(price) ~ log(carat), data = diamonds)
diamonds2 <- diamonds %>%
  add_residuals(mod) %>%
  mutate(resid = exp(resid))
ggplot(data = diamonds2) +
  geom_point(mapping = aes(x = carat, y = resid))
```



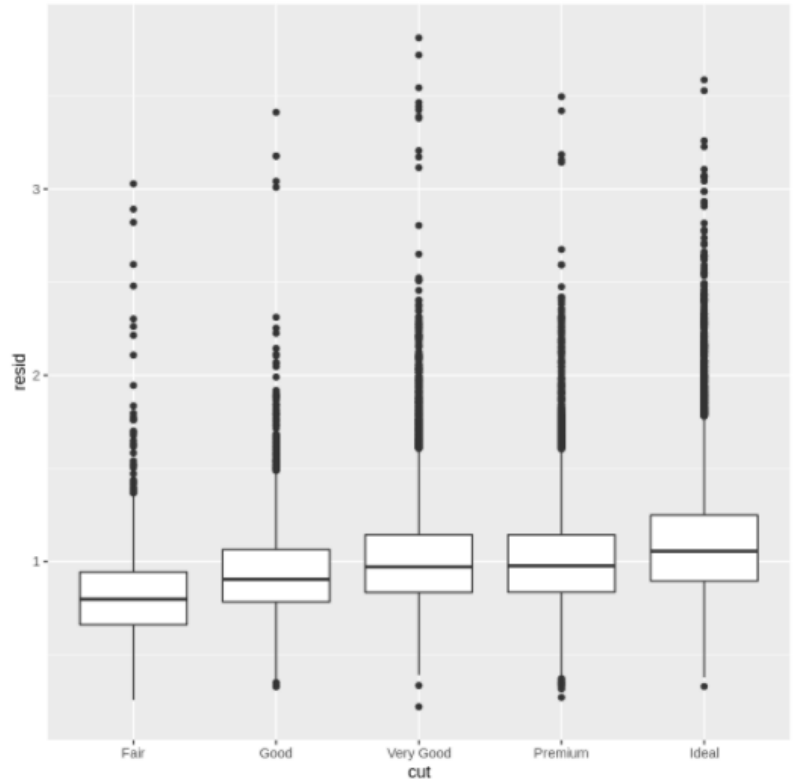
MONASH University
MALAYSIA

MONASH
BUSINESS

Patterns and models

Once we've removed the strong relationship between carat and price, we can see what you expect in the relationship between cut and price: relative to their size, **better quality diamonds are more** expensive.

```
ggplot(data = diamonds2) +
  geom_boxplot(mapping = aes(x = cut, y = resid))
```



MONASH University MALAYSIA

MONASH BUSINESS

THANK YOU

FIND OUT MORE AT MONASH.EDU.MY
LIKE @MONASH UNIVERSITY MALAYSIA ON FACEBOOK
FOLLOW @MONASHMALAYSIA ON TWITTER