NEWCASTLE UNIVERSITY

SCHOOL OF MATHEMATICS, STATISTICS & PHYSICS

SEMESTER 2 2025/26

MAS2908

Data Visualisation

(This document contains solutions)

Time allowed: 2 hours

Candidates should attempt all questions. Marks for each question are indicated. However you are advised that marks may be adjusted in accordance with the University's Moderation and Scaling Policy.

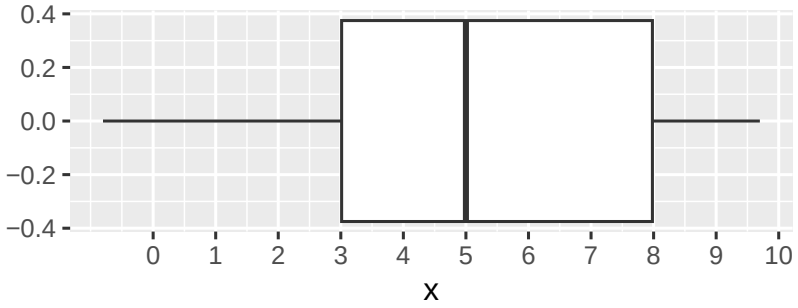
There are THREE questions in Section A and THREE questions in Section B.

This is a specimen exam paper demonstrating the style of questions you may encounter.

Calculators may be used.

SECTION A

A1. The boxplot of variable x is shown below:



(a) What is the median of x?

(i) 3.0
(ii) 4.0
(iii) 5.0
(iv) Cannot tell from the boxplot

Answer:

5.0 [1]

[1 mark]

(b) What is the first quartile (Q1) of x?

(i) 2.0
(ii) 3.0
(iii) 4.0
(iv) Cannot tell from the boxplot

Answer:

3.0 [2]

[2 marks]

(c) Can you tell whether the distribution of x is unimodal, bimodal, or something else from the boxplot alone? Justify your answer.

Answer:

No. A boxplot only displays the five-number summary (minimum, Q1, median, Q3, and maximum, subject to the 1.5 IQR whisker rule) and flags potential outliers. It does not reveal how the data

Question A1 continued on next page

are distributed within each quartile, so there is no way to determine the number of modes. [2]

[2 marks]

[Total: 5 marks]

A2. You will use the `penguins` dataset from the `palmerpenguins` package for this question. This dataset contains measurements of 344 penguins from three species. The key variables are:

- `species`: Penguin species (Adelie, Chinstrap, or Gentoo)
- `island`: Island of observation (Biscoe, Dream, or Torgersen)
- `bill_length_mm`: Bill length in mm
- `bill_depth_mm`: Bill depth in mm
- `flipper_length_mm`: Flipper length in mm
- `body_mass_g`: Body mass in g
- `sex`: Sex of the penguin (female or male)
- `year`: Year of observation (2007, 2008, or 2009)

Run the following lines of code in RStudio console before answering the questions:

```
library(ggplot2)
library(dplyr)
library(palmerpenguins)
penguins <- palmerpenguins::penguins
```

If a package is not yet available i.e. you get an error when running the above code, please first install the missing package(s). For example, if the `ggplot2` package is missing, run `install.packages("ggplot2")` before running the above code again. If you are prompted to answer whether you want to install from sources the packages which need compilation, answer **No** (or equivalent).

For this question, when writing code to fix the identified problems, you can assume that the font size and aspect ratio of the plots are unchanged.

(a) To examine the number of penguins according to `species`, the following code is run but an error is returned:

```
ggplot(penguins, aes(x = species)) + geom_histogram()
```

Write down the reason for the error.

Answer:

`geom_histogram()` is for continuous variables; `species` is a categorical variable. [1]

[1 mark]

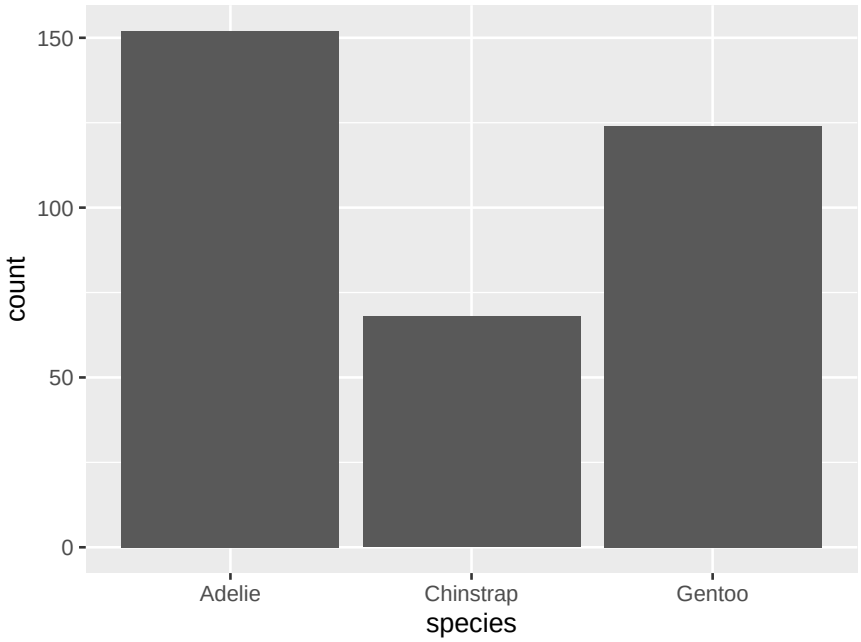
Question A2 continued on next page

(b) Edit the code in the previous part to remedy the issue you identified.

Answer:

data and mapping [1]; `geom_bar()` [1]

```
ggplot(penguins, aes(x = species)) + geom_bar()
```

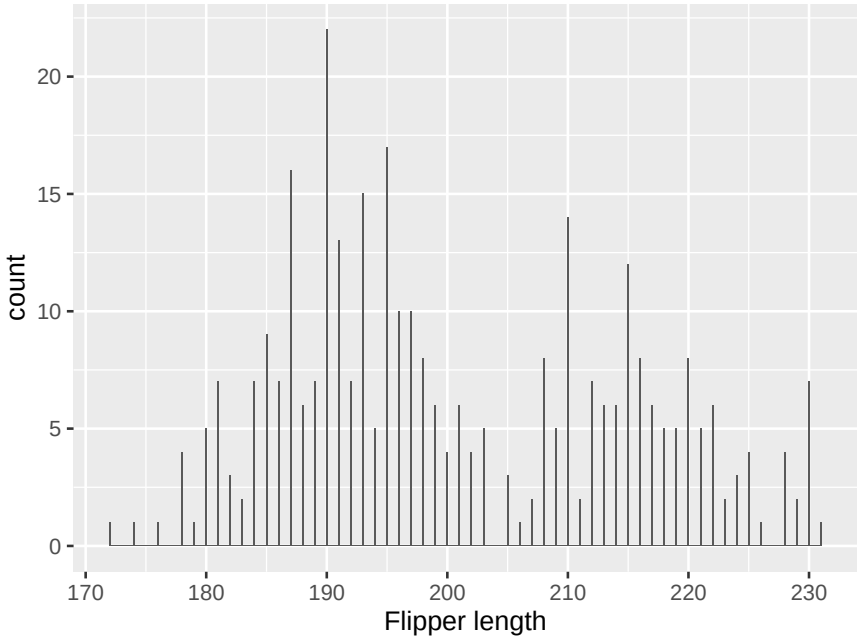


[2 marks]

(c) To examine the shape of the distribution of `flipper_length_mm`, the following code is run:

```
ggplot(penguins, aes(x = flipper_length_mm)) +
  geom_histogram(binwidth = 0.1) +
  labs(x = "Flipper length")
```

Question A2 continued on next page



Write down the reason why the plot is showing very thin lines.

Answer:

The `binwidth` of 0.1 is too small relative to the range of `flipper_length_mm`, so each bin contains almost no observations and appears as a thin line. [1]

[1 mark]

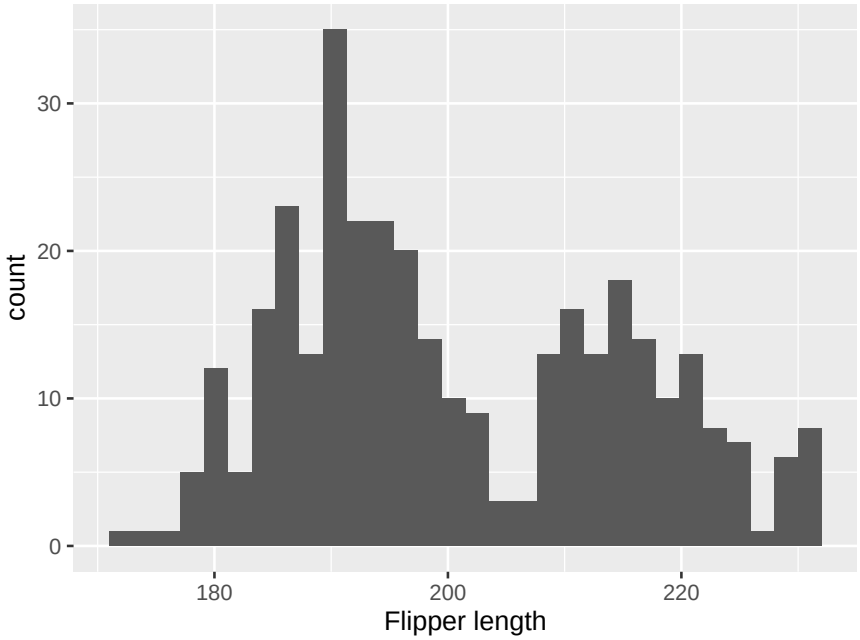
(d) Edit the code in the previous part to remedy the issue you identified.

Answer:

geom_histogram() [1], use a larger (or default) `binwidth` [1]

```
ggplot(penguins, aes(x = flipper_length_mm)) +
  geom_histogram() +
  labs(x = "Flipper length")
```

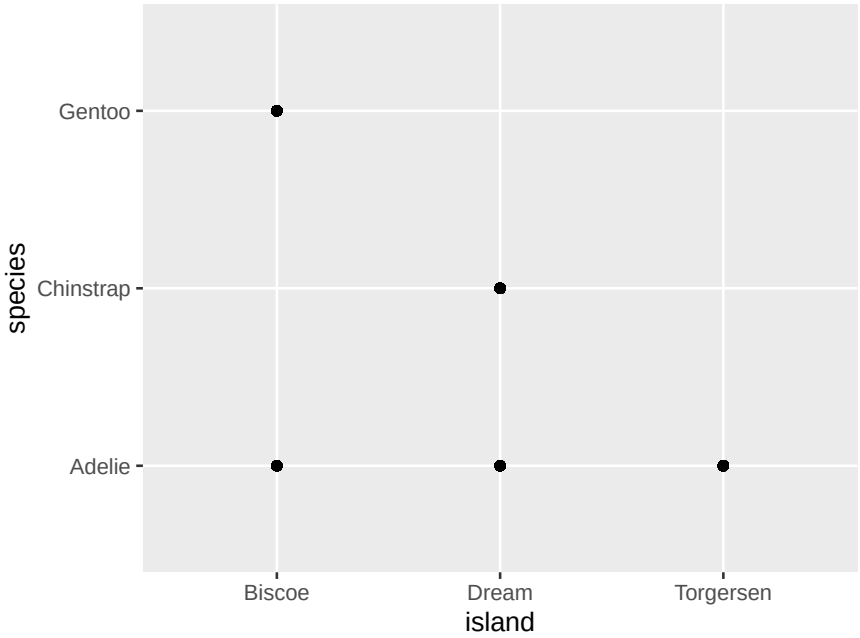
Question A2 continued on next page



[2 marks]

(e) To examine the pairwise relationships between `species` and `island` without using colours, the following code is run:

```
ggplot(penguins, aes(x = island, y = species)) +
  geom_point()
```



Write down one potential issue with this plot and briefly justify your answer.

Answer:

Question A2 continued on next page

The points are on top of each other (overplotting): because both `island` and `species` are categorical, many penguins share the same coordinates and are drawn as a single point, hiding the density of observations. [1]

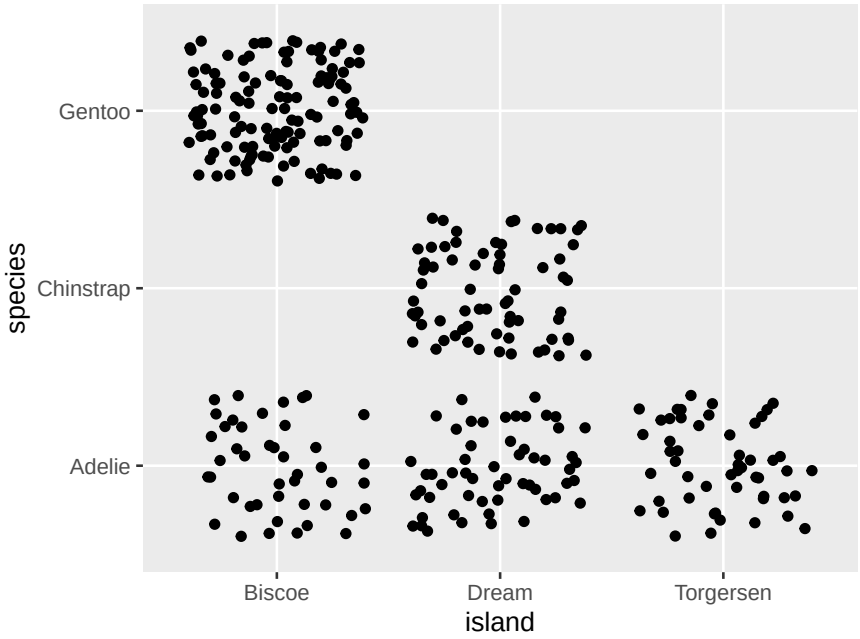
[1 mark]

(f) Edit the code in the previous part to remedy the issue you identified.

Answer:

`geom_jitter()` [2]

```
ggplot(penguins, aes(x = island, y = species)) +
  geom_jitter()
```



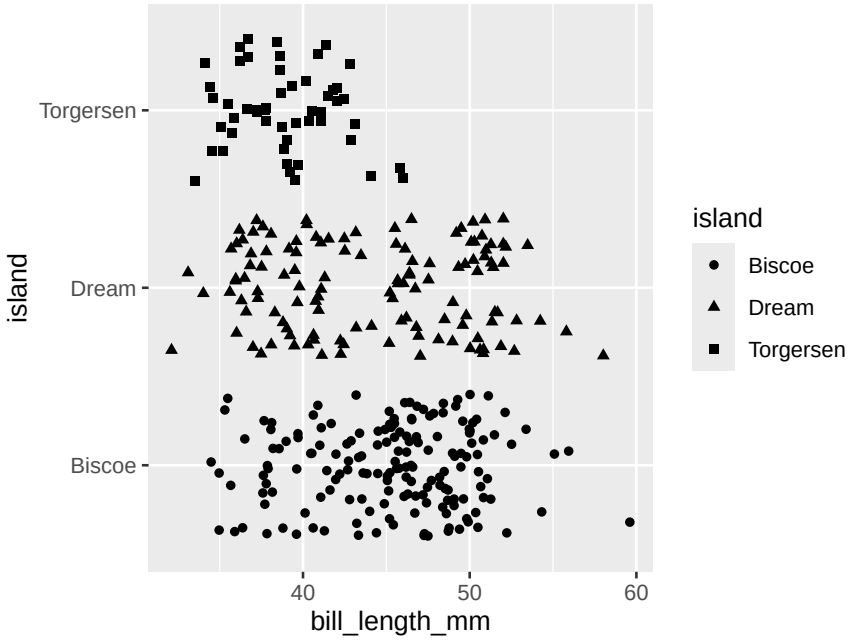
Alternative: `geom_count()` [2] (plot not shown)

```
ggplot(penguins, aes(x = island, y = species)) +
  geom_count()
```

[2 marks]

(g) To examine the relationship between `island` and `bill_length_mm`, the following code is run:

```
ggplot(penguins, aes(x = bill_length_mm, y = island,
                     shape = island)) +
  geom_jitter()
```



Write down two potential issues with this plot and briefly justify your answer.

Answer:

(1) `shape = island` is redundant because `island` is already on the y-axis; mapping it to shape as well adds no new information [1]. (2) The x-axis label `bill_length_mm` is the raw variable name and not informative; it should be a descriptive label such as `"Bill length"` [1].

[2 marks]

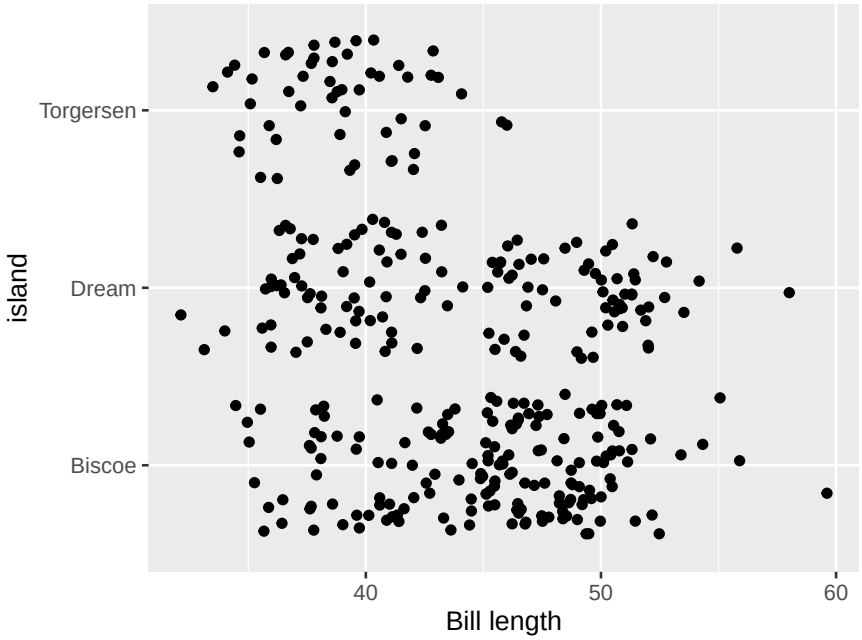
(h) Edit the code in the previous part to remedy the issues you identified, and label the x-axis label `"Bill length"`.

Answer:

`aes(x, y)` [1]; Remove `shape = island` from `aes()` [1]; add `labs(x = "Bill length")` [1]

```
ggplot(penguins, aes(x = bill_length_mm, y = island)) +
  geom_jitter() +
  labs(x = "Bill length")
```

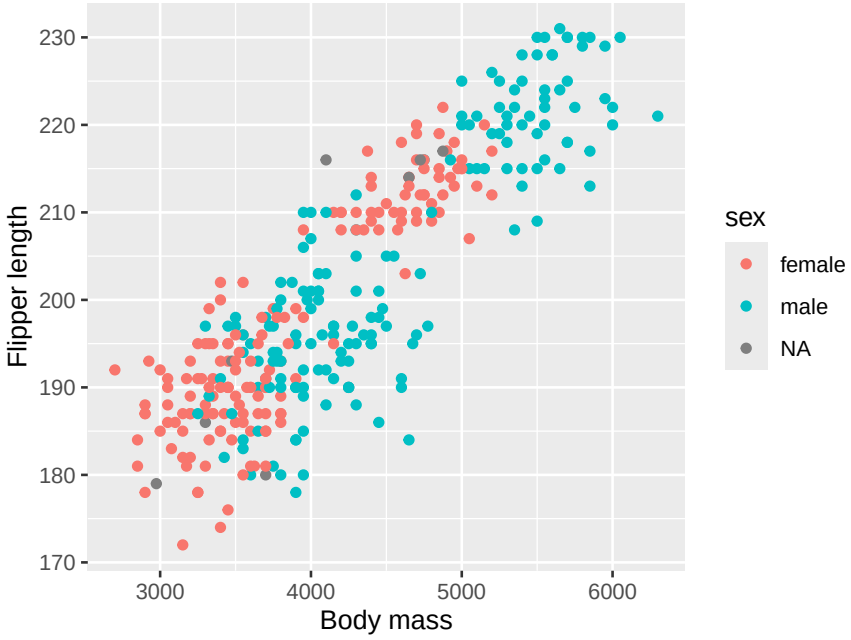
Question A2 continued on next page



[3 marks]

(i) To examine the relationship between `body_mass_g` and `flipper_length_mm` by `sex`, the following code is run:

```
ggplot(penguins, aes(x = body_mass_g, y = flipper_length_mm,
                     colour = sex)) +
  geom_point() +
  labs(x = "Body mass", y = "Flipper length")
```



Explain why there are grey points in the scatterplot.

Answer:

Question A2 continued on next page

The grey points are observations where `body_mass_g` or `flipper_length_mm` is missing (`NA`); `ggplot2` renders these as grey by default. [2]

[2 marks]

(j) Following from the previous part, we would like to use a colour-blind friendly palette, but the following code produces an error:

```
ggplot(penguins, aes(x = body_mass_g, y = flipper_length_mm,
                     colour = sex)) +
  geom_point() +
  labs(x = "Body mass", y = "Flipper length") +
  scale_colour_viridis_b()
```

Write down the reason for the error.

Answer:

`scale_colour_viridis_b()` is a binned scale that only supports continuous data [1]; `sex` is a discrete/categorical variable [1].

[2 marks]

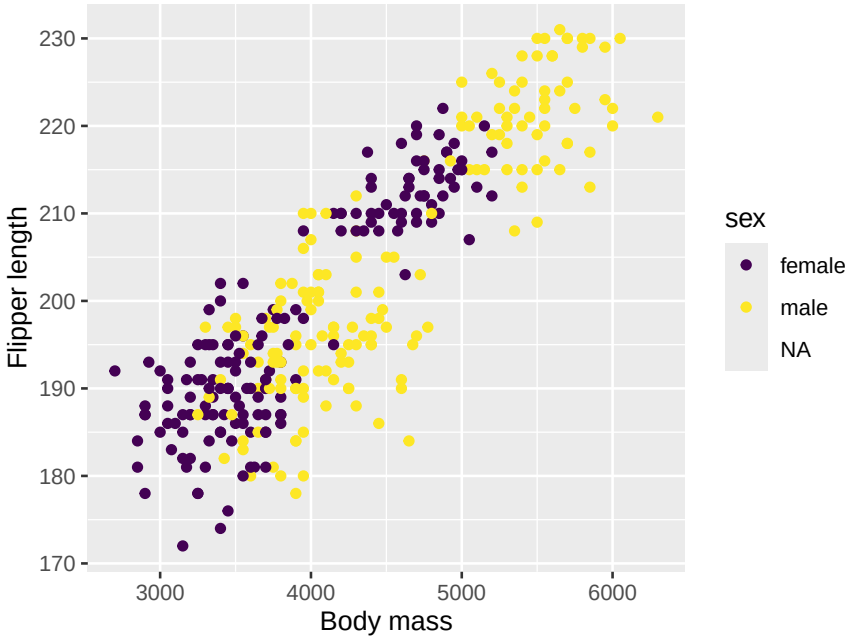
(k) Edit the code in the previous part to resolve the error you identified.

Answer:

`scale_colour_viridis_d()` [1]

```
ggplot(penguins, aes(x = body_mass_g, y = flipper_length_mm,
                     colour = sex)) +
  geom_point() +
  labs(x = "Body mass", y = "Flipper length") +
  scale_colour_viridis_d()
```

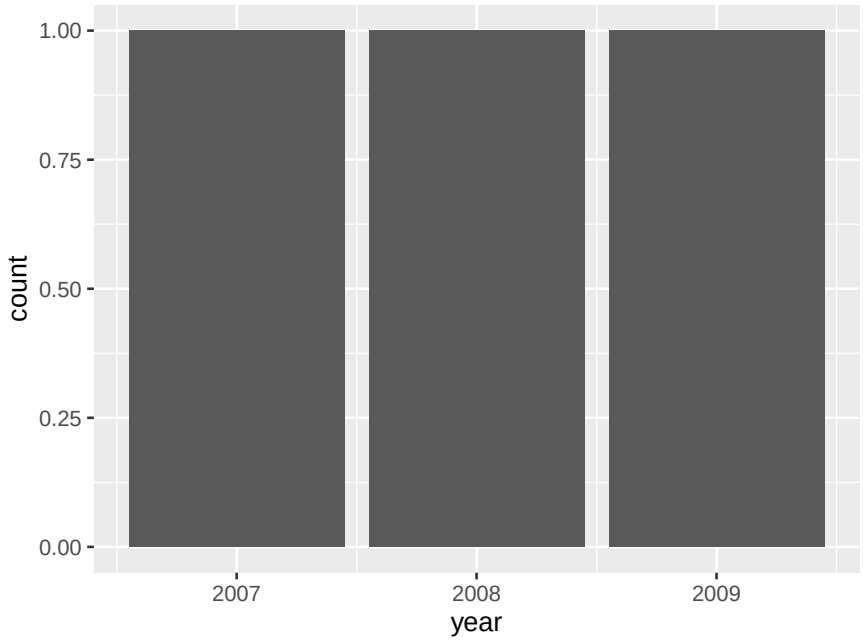
Question A2 continued on next page



[1 mark]

(l) To visualise the counts according to `year`, the following code is run:

```
penguins |>
  count(year) |>
  ggplot(aes(x = year)) +
  geom_bar()
```



Write down one potential issue with this plot and briefly justify your answer.

Answer:

Question A2 continued on next page

`geom_bar()` counts the number of rows it receives; after `count(year)`, there is one row per year, so each bar has height 1 rather than the actual number of penguins observed that year. [2]

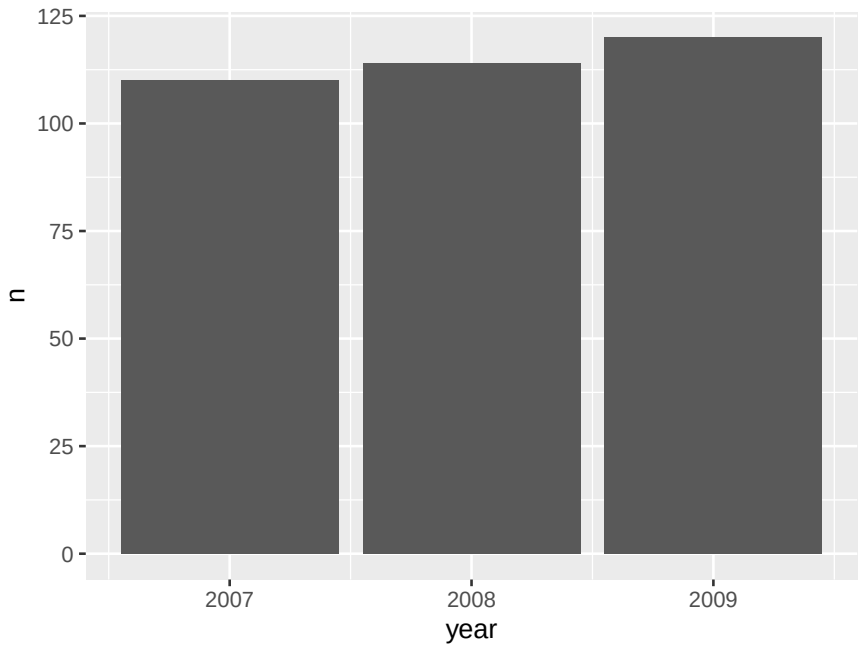
[2 marks]

(m) Edit the code in the previous part to remedy the issue you identified.

Answer:

`geom_col()` with `y = n` in `aes()` [2]; alternatively `geom_bar(stat = "identity")` [2]; or remove `count(year)` and plot `penguins` directly [2]

```
penguins |>
  count(year) |>
  ggplot(aes(x = year, y = n)) +
  geom_col()
```



Alternative: remove `count(year)` and plot directly (plot not shown)

```
ggplot(penguins, aes(x = year)) + geom_bar()
```

[2 marks]

(n) The following incomplete code attempts to visualise the distribution of `bill_length_mm` according to `species` using a boxplot:

```
ggplot(penguins) +
  geom_boxplot(aes(x = species))
```

Question A2 continued on next page

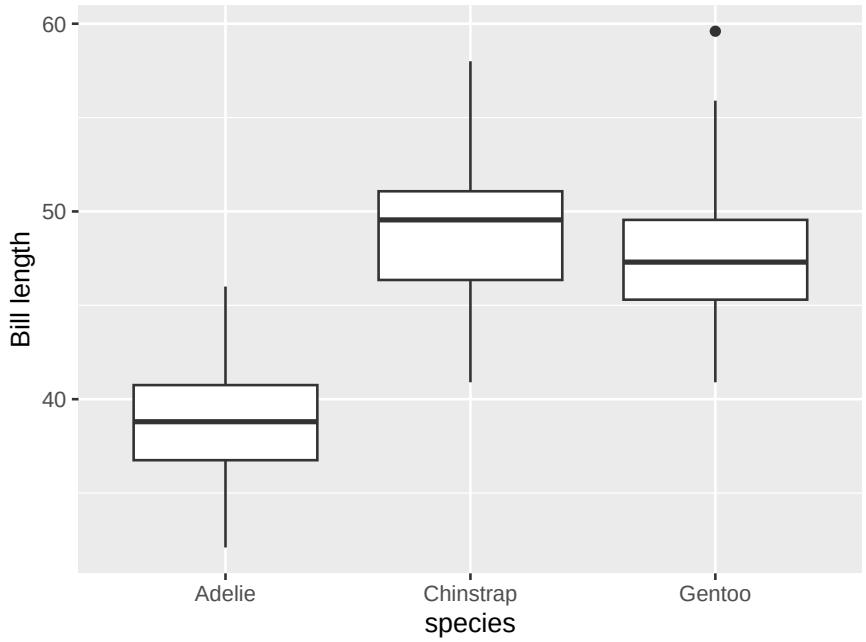
Complete the code, and label the y-axis `"Bill length"`.

Answer:

`, y = bill_length_mm)` to complete the `aes()` [1]; `+ labs(y = "Bill length")` [1]

```
ggplot(penguins) +
  geom_boxplot(aes(x = species, y = bill_length_mm)) +
  labs(y = "Bill length")

## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



[2 marks]

[Total: 25 marks]

A3. You will use the `economics` dataset from the `ggplot2` package for this question. This dataset records monthly US economic indicators from July 1967 to April 2015. The key variables are:

- `date`: Date of observation (monthly)
- `pce`: Personal consumption expenditures (billion USD)
- `pop`: Total US population (thousands)
- `psavert`: Personal savings rate (%)
- `uempmed`: Median duration of unemployment (weeks)
- `unemploy`: Number of unemployed (thousands)

Run the following lines of code in RStudio console before answering the questions:

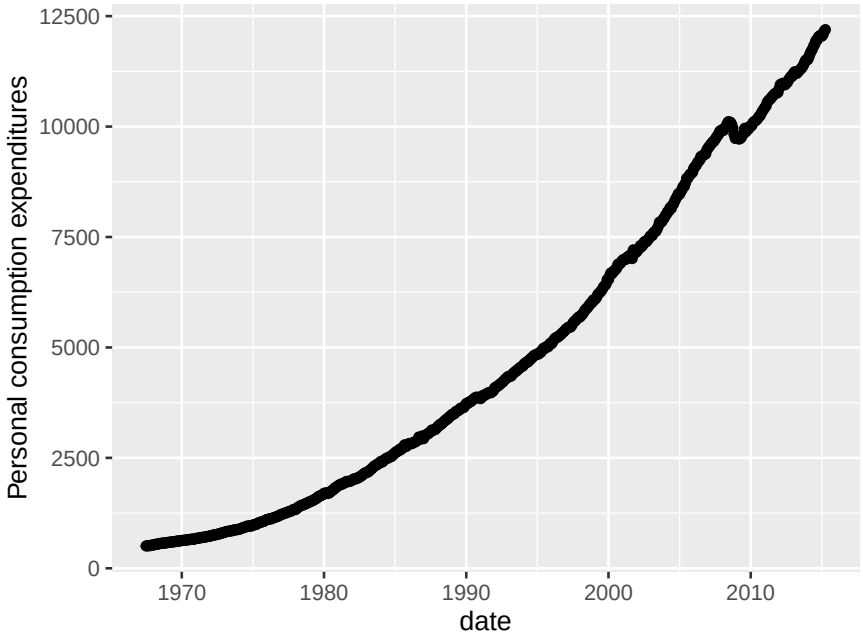
```
library(ggplot2)
library(dplyr)
```

If a package is not yet available i.e. you get an error when running the above code, please first install the missing package(s). For example, if the `ggplot2` package is missing, run `install.packages("ggplot2")` before running the above code again. If you are prompted to answer whether you want to install from sources the packages which need compilation, answer **No** (or equivalent).

(a) To examine the trend of personal consumption expenditures over time, the following code is run:

```
ggplot(economics, aes(x = date, y = pce)) +
  geom_point() +
  labs(y = "Personal consumption expenditures")
```

Question A3 continued on next page



Write down two potential issues with this plot.

Answer:

(1) The points overlap each other heavily, making the trend hard to see [1]. (2) The x-axis label defaults to `date` (the variable name), which is not informative at this scale — `"Year"` would be clearer [1].

[2 marks]

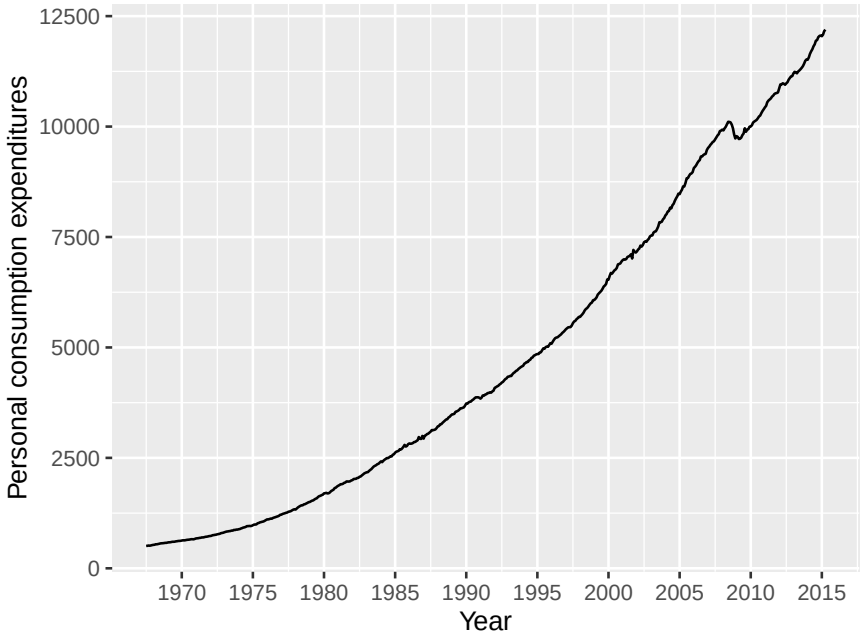
(b) Edit the code in the previous part to remedy the issues you identified, **and** change the x-axis tick mark labels to every 5 years (e.g. $1970, 1975, \ldots$).

Answer:

`geom_line()` [1], `labs(x = "Year")` [1], `scale_x_date()` [1], `date_breaks = "5 years"` [1], `date_labels = "%Y"` [1]

```
ggplot(economics, aes(x = date, y = pce)) +
  geom_line() +
  scale_x_date(
    date_breaks = "5 years",
    date_labels = "%Y"
  ) +
  labs(x = "Year", y = "Personal consumption expenditures")
```

Question A3 continued on next page



[5 marks]

(c) The following code plots both the personal savings rate and the number of unemployed over time:

```
ggplot(economics) +
  geom_line(aes(x = date, y = psavert)) +
  geom_line(aes(x = date, y = unemploy), colour = "red") +
  labs(x = "Personal savings rate / number of unemployed", y = NULL)
```

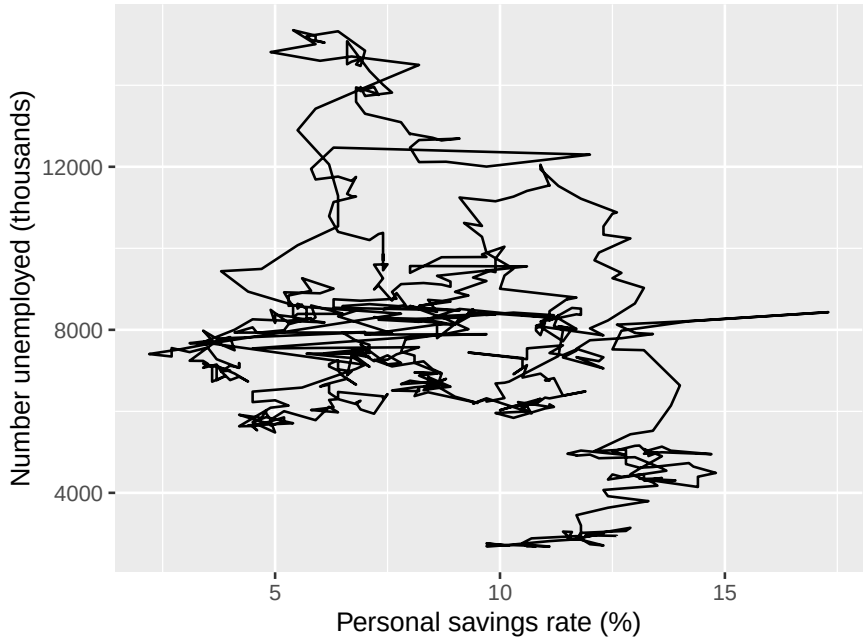
Write down the potential issue with this plot that is unrelated to the use of colours, briefly justify your answer, and suggest a way to remedy this issue.

Question A3 continued on next page

Answer:

`psavert` appears as a near-flat line at the bottom of the plot [1] because the two variables are on very different scales: `psavert` ranges from roughly 2% to 17%, while `unemploy` ranges from about 2,000 to 15,000 thousand [1]. Either standardise both variables so they share a common scale, or use `geom_path()` to plot `psavert` against `unemploy` directly, with time as the implicit sequence [1].

```
ggplot(economics, aes(x = psavert, y = unemploy)) +
  geom_path() +
  labs(x = "Personal savings rate (%)",
       y = "Number unemployed (thousands)")
```



Alternative: standardise both series (plot not shown)

```
economics |>
  mutate(
    psavert_std = scale(psavert)[, 1],
    unemploy_std = scale(unemploy)[, 1]
  ) |>
  ggplot(aes(x = date)) +
  geom_line(aes(y = psavert_std)) +
  geom_line(aes(y = unemploy_std), colour = "red") +
  labs(x = NULL, y = "Standardised value")
```

[3 marks]

[Total: 10 marks]

SECTION B

B4. This question assesses your ability to create data visualisations and embed them in dynamic document generation using R Markdown. You will create plots using `ggplot2` to answer the questions, and submit both the generated file **and** the R Markdown file (.Rmd). There are marks allocated for following these instructions — see Question B4(d).

You will use the `msleep` dataset from the `ggplot2` package for this question. This dataset contains sleep data for 83 mammals. The variables are:

- `name`: Common name of the mammal
- `genus`: Taxonomic genus
- `vore`: Diet type (carni, herbi, insecti, omni; `NA` if unknown)
- `order`: Taxonomic order
- `conservation`: Conservation status (IUCN category)
- `sleep_total`: Total amount of sleep per day (hours)
- `sleep_rem`: REM sleep per day (hours)
- `sleep_cycle`: Length of sleep cycle (hours)
- `awake`: Time spent awake per day (hours)
- `brainwt`: Brain weight (kg)
- `bodywt`: Body weight (kg)

Open a new R Markdown file, clear any existing content, and copy and paste the following template:

Question B4 continued on next page

````
---
title: "Question B4"
author: "Your student number"
output: html_document
---

```{r setup}
#| echo: false
library(ggplot2)
library(dplyr)
knitr::opts_chunk$set(
  out.width = "80%",
  fig.align = "center",
  fig.asp = 0.7
)
```

# Part (a)

i. Write answers to part (i) here.

ii. Write answers to part (ii) here.

```{r aii}
# write code here to make necessary plots
```
````

Edit the template as appropriate and according to the following parts.

(a) Figure 1 attempts to visualise the distribution of `sleep_total` by `vore`.

(i) Comment on two different aspects in which this plot could be improved, or is not well suited to the stated purpose.

Answer:

The font size is too small relative to the figure width; for each diet type, there are multiple observations, causing overplotting and making the scatterplot poor for showing the distribution of `sleep_total` within each group.

(ii) Improve this plot by using a different `geom_*()` and making necessary changes.

Answer:

To show the distribution by `vore`, use a boxplot or jitter plot. To fix the font size, reduce `fig.width` or increase `base_size` in

Question B4 continued on next page

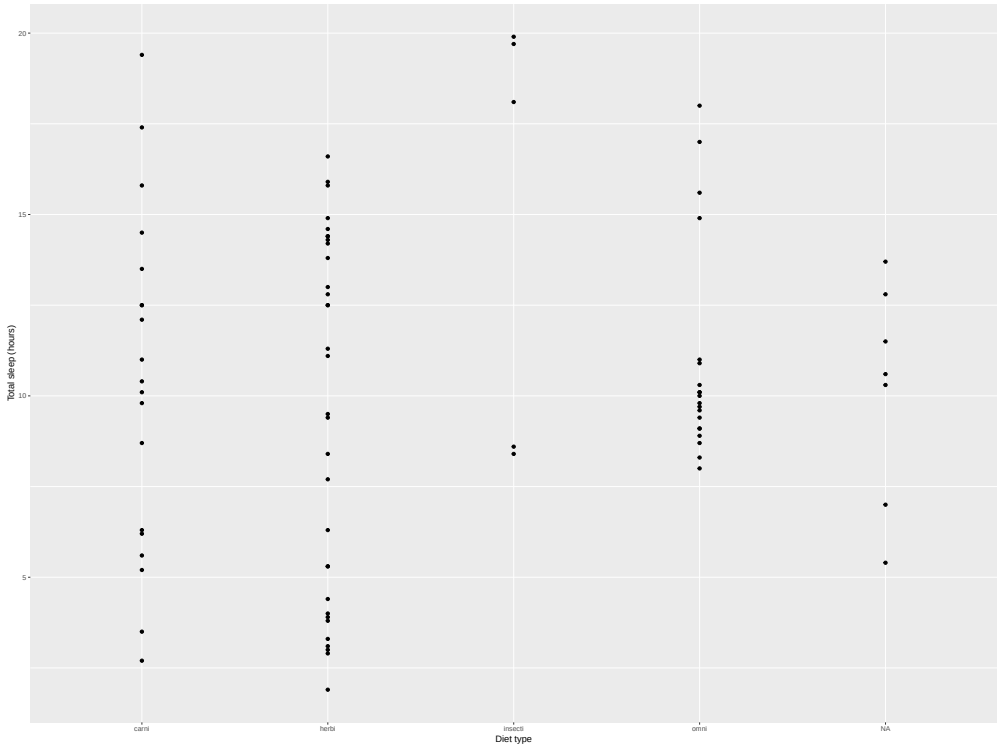


Figure 1: Scatterplot of diet type and total sleep.

a theme. See Figure 2.

```
ggplot(msleep, aes(vore, sleep_total)) +
  geom_boxplot() +
  labs(x = "Diet type", y = "Total sleep (hours)")
```

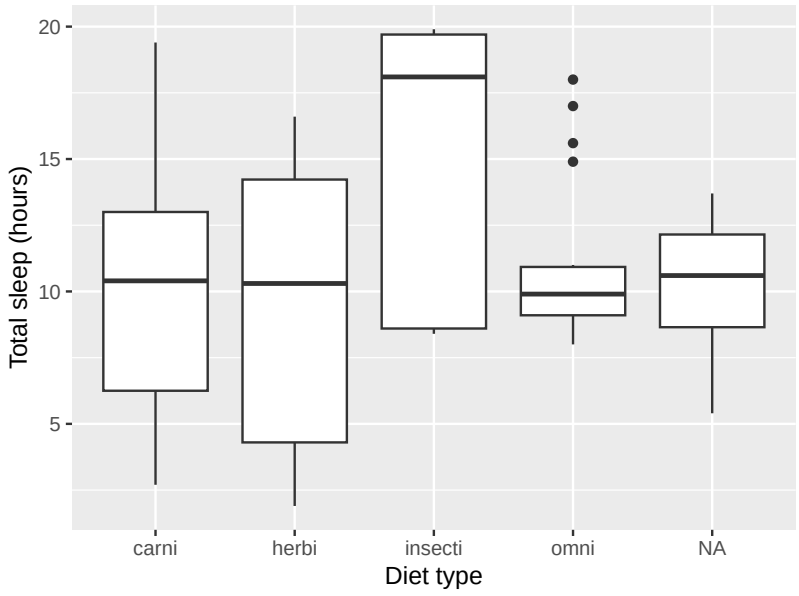


Figure 2: Boxplot of total sleep by diet type.

Question B4 continued on next page

[4 marks]

(b) We want to show the counts of mammals by `vore` and `order` using `geom_count()`.

(i) Create the count plot. Make sure you pair the axes and the variables appropriately so that the tick mark labels do not overlap each other.

Answer:

Place `vore` on the x-axis (few short labels) and `order` on the y-axis (many labels, which stack without overlapping vertically). See Figure 3.

```
msleep |>
  ggplot() +
  geom_count(aes(vore, order)) +
  labs(x = "Diet type", y = "Taxonomic order", size = "Count")
```

(ii) Suggest a way to improve the tick mark labels for `vore`.

Answer:

The values `carni`, `herbi`, `insecti`, and `omni` are truncated; appending `vore` to each (except `NA`) gives the full words *carnivore*, *herbivore*, *insectivore*, and *omnivore*. This can be done with `scale_x_discrete()`:

[3 marks]

(c) Identify a variable in `msleep` that is strongly positively correlated with `bodywt`.

(i) Name the variable.

Answer:

`brainwt` is strongly positively correlated with `bodywt`. [1 mark]

(ii) Create a scatterplot of the variable with `bodywt` on the x-axis. Adjust the axes appropriately using `scale_x_*()` and `scale_y_*()` to improve the visibility of all the data points.

Answer:

See Figure 4. Both variables span several orders of magnitude, so without log scales most points are squashed into the bottom-left corner. Adding `scale_x_log10()` and `scale_y_log10()` reveals the full spread of the data. [1 mark each, 2 marks]

Question B4 continued on next page

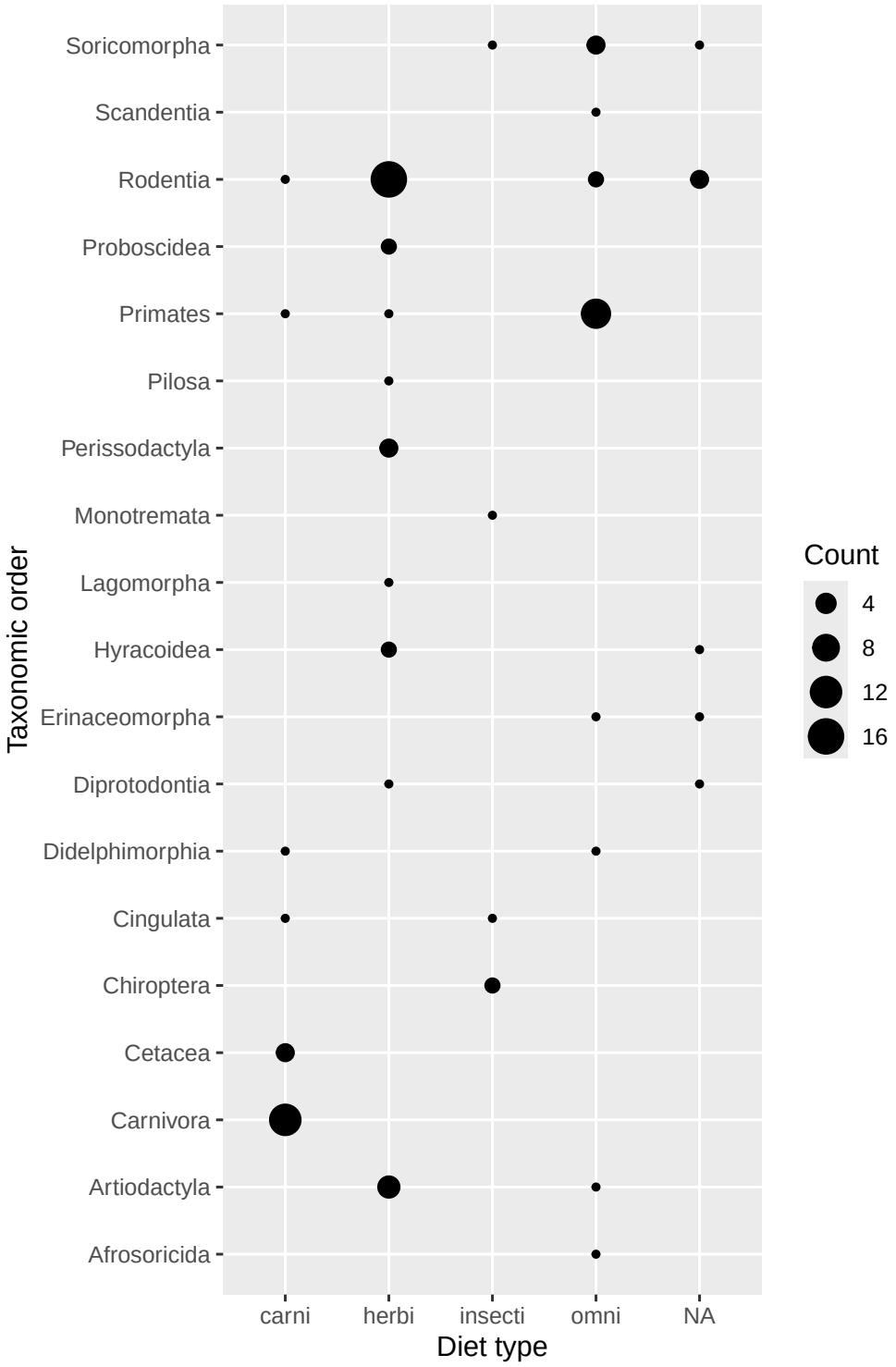


Figure 3: Count plot of taxonomic order by diet type.

Question B4 continued on next page

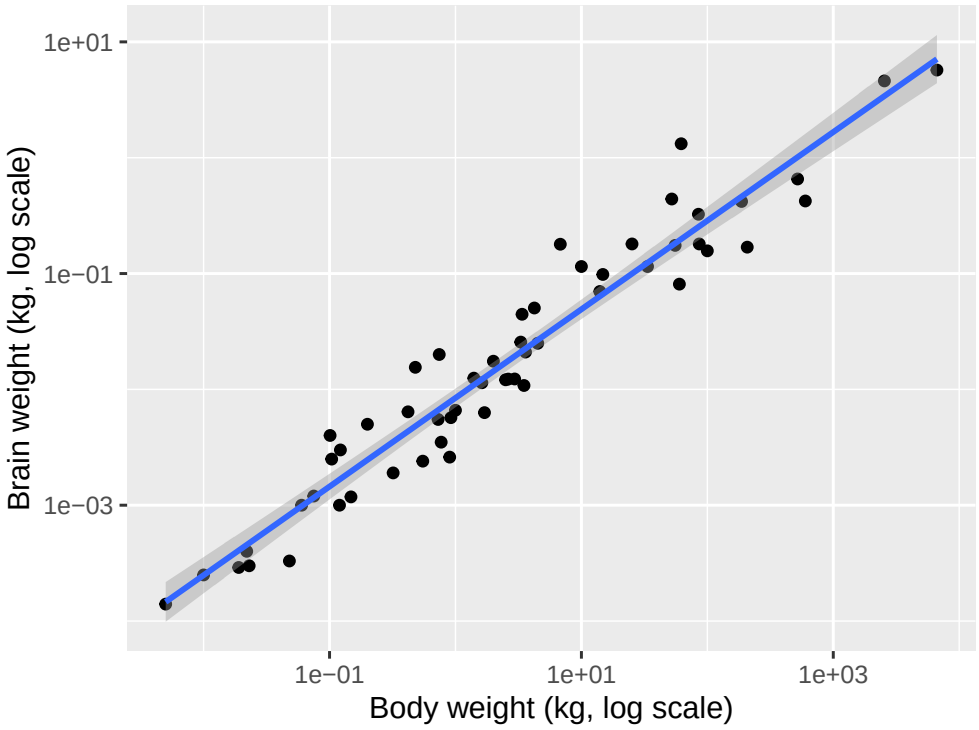


Figure 4: Scatterplot of brain weight against body weight (log–log scale) with line of best fit.

(iii) Overlay the straight line of best fit.

Answer:

Use `geom_smooth(method = "lm")` to overlay the linear regression line on the log–log scale. [1 mark]

```
ggplot(msleep, aes(x = bodywt, y = brainwt)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_x_log10() +
  scale_y_log10() +
  labs(x = "Body weight (kg, log scale)",
       y = "Brain weight (kg, log scale)")
```

[4 marks]

(d) Overall presentation:

(i) Your submission should be a single report in **HTML, and the R Markdown (.Rmd) file that generates the report**.

(ii) Ensure that you have included sensible and clear axis labels, figure captions, and readable font sizes in well-proportioned plots.

Answer:

Question B4 continued on next page

2 marks for submitting the correct files, 1 mark for reproducibility (not reproducible if absolute paths used), 2 marks each for axis labels, figure captions, and font sizes + figure aspect ratios.

[9 marks]

[Total: 20 marks]

B5. For this question, use a sensible label for each scale used.

This question assesses your ability to visualise geospatial data. First load the required packages and data by running the following lines of code in RStudio:

```
library(ggplot2)
library(dplyr)
library(sf)
library(units)
library(geodaData)
library(rnaturalearth)
library(rnaturalearthdata)
data(ncovr, package = "geodaData")
```

If a package is not yet available i.e. you get an error when running the above code, please first install the missing package(s). For example, if the `ggplot2` package is missing, run `install.packages("ggplot2")` before running the above code again. If you are prompted to answer whether you want to install from sources the packages which need compilation, answer **No** (or equivalent).

You will use the `ncovr` dataset from the `geodaData` package. This dataset contains county-level socioeconomic and demographic data for the contiguous United States across four decades (1960, 1970, 1980, 1990). The key variables are:

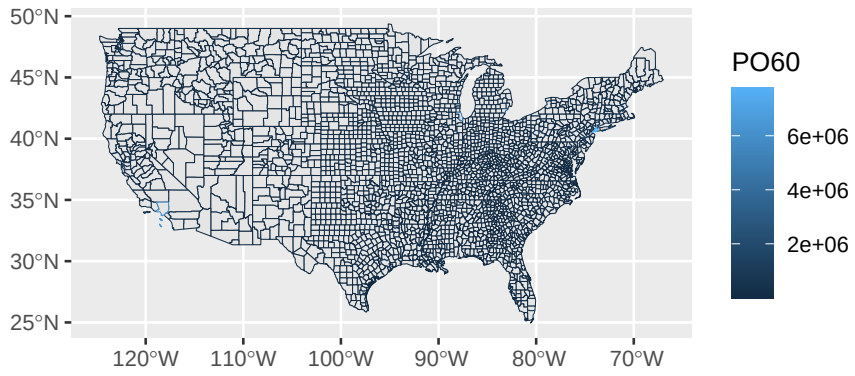
- `NAME`: County name
- `STATE_NAME`: State name
- `HR`*yy*: Homicide rate per 100,000 population in 19*yy* (e.g. `HR60` for 1960)
- `PO`*yy*: Population in 19*yy* (e.g. `PO60` for 1960)
- `UE`*yy*: Unemployment rate in 19*yy*
- `MA`*yy*: Median age in 19*yy*
- `POL`*yy*: Log population in 19*yy*
- `DNL`*yy*: Log population density in 19*yy*
- `GI`*yy*: Gini coefficient (income inequality) in 19*yy*

where *yy* is one of 60, 70, 80, or 90.

(a) We want to visualise the population, with the interior of the counties coloured by `PO60`, but the following code does not achieve the purpose:

Question B5 continued on next page

```
ggplot(ncovr, aes(colour = PO60)) + geom_sf()
```



Write down the reason.

Answer:

In `ggplot2`, `colour` controls the colour of the *boundary* (outline) of geometric shapes, whereas `fill` controls the colour of the *interior*. Using `colour = PO60` therefore colours the county boundaries rather than the filled areas. [2]

[2 marks]

(b) Edit the code in the previous part to remedy the issue, and label the fill scale `"Population (1960)"`.

Answer:

Replace `colour` with `fill` [1] and add `labs(fill = "Population (1960)")` [1]:

```
ggplot(ncovr, aes(fill = PO60)) +
  geom_sf() +
  labs(fill = "Population (1960)")
```

[2 marks]

(c) Explain why the resulting plot is not very informative, in relation to the distribution of `PO60`.

Answer:

`PO60` is highly right-skewed: a small number of counties (large metropolitan areas) have populations far larger than the vast majority. On a linear colour scale this compresses almost all counties

Question B5 continued on next page

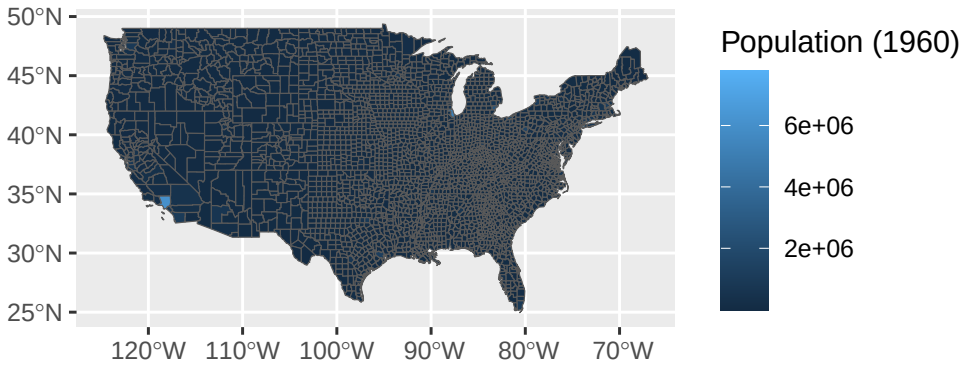


Figure 5: Population in 1960 by US county.

into the same low-value colour band, making differences between them invisible. [2]

[2 marks]

(d) Edit the code to improve the plot by using a variable that is different but related to `POP60`. Use an appropriate label for the fill scale.

Answer:

Use `POL60` (log population), which compresses the skew and reveals variation across counties [1], and label accordingly [1]:

```
ggplot(ncovr, aes(fill = POL60)) +
  geom_sf() +
  labs(fill = "Log population (1960)")
```

[2 marks]

(e) The following code creates a new dataset and a new variable:

```
ncovr_area <- ncovr |> mutate(area = st_area(geometry) |> drop_units())
```

Explain what the code does.

Answer:

`st_area(geometry)` computes the area of each county polygon, returning values with area units attached (e.g. square metres) [1]. `drop_units()` removes the units attribute, converting the result to

Question B5 continued on next page

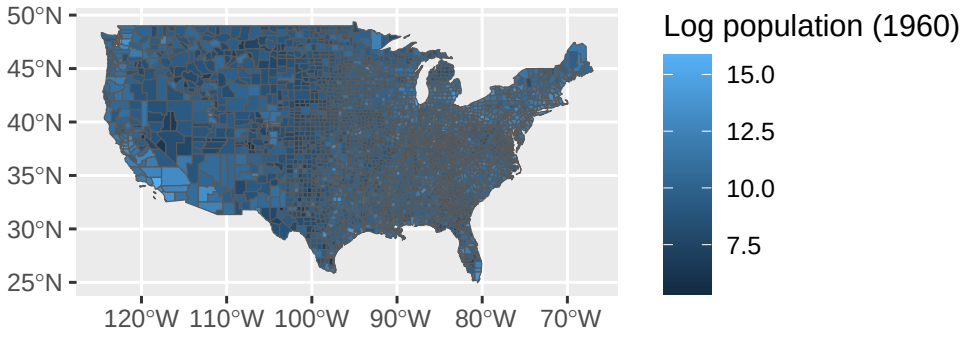


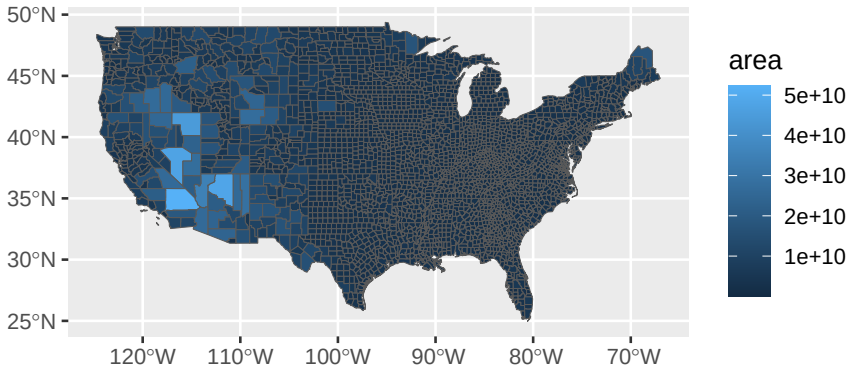
Figure 6: Log population in 1960 by US county.

a plain numeric vector [1]. The result is a new dataset `ncovr_area` with an additional column `area` giving each county's area as a unitless number.

[2 marks]

(f) Explain why the plot created by the following code is not very informative, in relation to the nature of the created variable:

```
ncovr_area |> ggplot() + geom_sf(aes(fill = area))
```



Answer:

The variable `area` represents the geographic size of each county, which is already fully encoded in the shape and extent of each poly-

Question B5 continued on next page

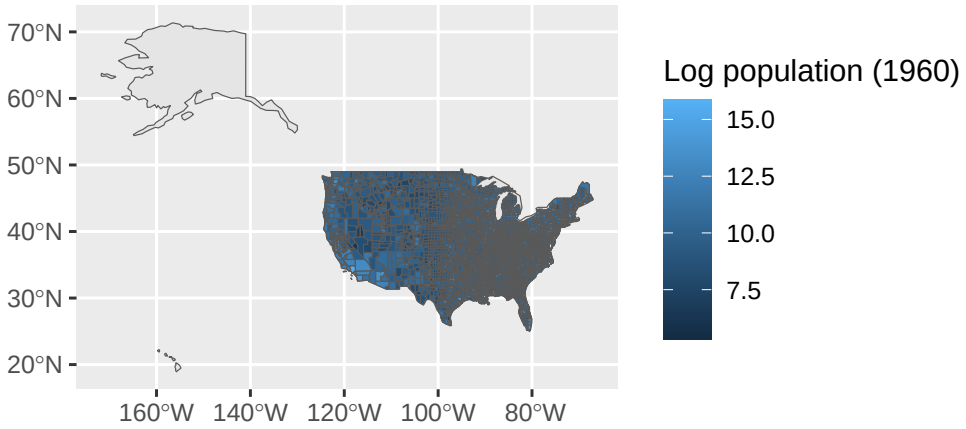


Figure 7: Log population in 1960 overlaid on the United States background map.

gon on the map. The fill colour is therefore perfectly correlated with the visual size of each county, adding no information beyond what is already visible. [2]

[2 marks]

(g) The following lines of code create a map of the United States:

```
usa_map <-
  ne_countries(country = "united states of america", returnclass = "sf")
```

Using `geom_sf()` and `ncovr`, overlay your answer to Question 5(d) on `usa_map`.

Answer:

`ggplot()` with no default data [1], `geom_sf(data = usa_map)` as base layer [1], `geom_sf(data = ncovr, aes(fill = POL60))` on top [1], `labs(fill = ...)` [1]:

```
ggplot() +
  geom_sf(data = usa_map) +
  geom_sf(aes(fill = POL60), data = ncovr) +
  labs(fill = "Log population (1960)")
```

[4 marks]

(h) The county boundaries in the previous plot make it difficult to see the colours on this scale. Edit the code to remove the county boundaries.

Question B5 continued on next page

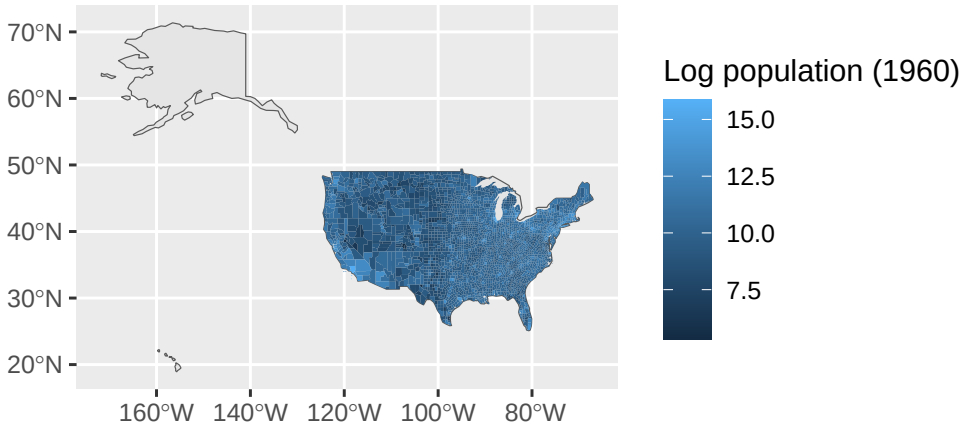


Figure 8: Log population in 1960 without county boundaries.

Answer:

Add `colour = NA` inside the `ncovr` `geom_sf()` call [2]:

```
ggplot() +
  geom_sf(data = usa_map) +
  geom_sf(aes(fill = POL60), data = ncovr, colour = NA) +
  labs(fill = "Log population (1960)")
```

[2 marks]

(i) We extract the geometry of the first county in the data using the following code:

```
ncovr$geometry[[1]]
```

Explain what the list of pairs of numbers in the output means.

Answer:

Each pair of numbers is a coordinate (longitude, latitude) of a vertex on the county boundary. By connecting these vertices in order, we trace the outline (polygon boundary) of the county; the complete list of pairs defines the shape of that county. [2]

[2 marks]

[Total: 20 marks]

B6. This question assesses your ability to make a Shiny app that visualises data. For parts B6(a) and B6(b), type your answers in the given text boxes. For parts B6(c) and B6(d), you will upload a single file named **app.R** that creates the Shiny app.

You will use the `seatbelts` dataset created from the code below. Run the following lines of code to load the required packages and data:

```
library(shiny)
library(dplyr)
library(ggplot2)
library(tsbox)
library(tidyr)
library(lubridate)
seatbelts <-
  Seatbelts |>
  ts_tbl() |>
  pivot_wider(names_from = id, values_from = value) |>
  mutate(law = factor(law), year = year(time))
```

If a package is not yet available i.e. you get an error when running the above code, please first install the missing package(s). For example, if the `ggplot2` package is missing, run `install.packages("ggplot2")` before running the above code again. If you are prompted to answer whether you want to install from sources the packages which need compilation, answer **No** (or equivalent).

The `seatbelts` dataset contains monthly road casualty figures for Great Britain from January 1969 to December 1984. The variables are:

- `time`: Date of observation (monthly, January 1969 to December 1984)
- `DriversKilled`: Monthly number of car drivers killed
- `drivers`: Monthly number of car drivers killed or seriously injured
- `front`: Monthly number of front-seat passengers killed or seriously injured
- `rear`: Monthly number of rear-seat passengers killed or seriously injured
- `kms`: Distance driven (millions of kilometres)
- `PetrolPrice`: Petrol price (pence per litre, in 1984 prices)
- `VanKilled`: Monthly number of van drivers killed
- `law`: Indicator for whether the compulsory seat belt law was in effect (0 = no, 1 = yes; law came into effect on 31 January 1983)
- `year`: Year of observation

(a) Using `seatbelts` and appropriate `ggplot2` functions, make a plot of the monthly counts of `DriversKilled` over time that shows both

Question B6 continued on next page

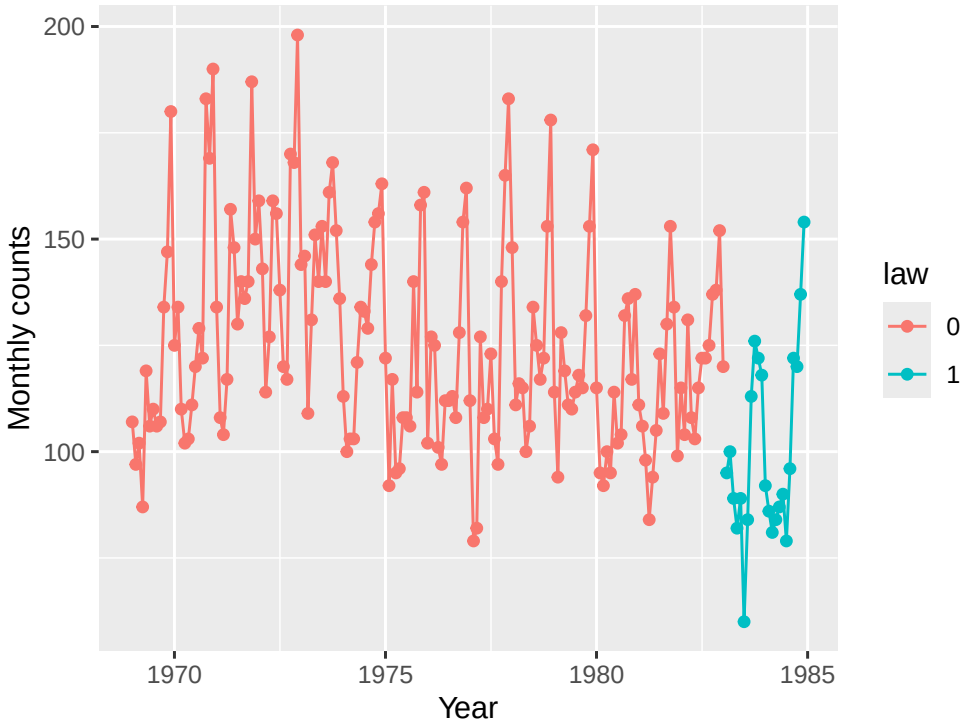


Figure 9: Monthly counts of drivers killed, coloured by whether the seat belt law was in effect.

the points and the line connecting them, with the x-axis and y-axis labelled `"Year"` and `"Monthly counts"`, respectively. Also colour the line and points according to `law`.

```
ggplot(seatbelts, aes(x = time, y = DriversKilled, colour = law)) +
  geom_line() +
  geom_point() +
  labs(x = "Year", y = "Monthly counts")
```

Answer:

Correct `aes(x = time, y = DriversKilled)` [1], `colour = law` in `aes()` [1], `geom_line()` [1], `geom_point()` [1], axis labels [1].

[5 marks]

(b) Describe, if any, the seasonal pattern of the monthly casualty counts, and suggest potential reasons for such a pattern.

Answer:

Monthly casualty counts tend to peak in autumn and winter (roughly October–January) and dip in summer (June–August). Potential

Question B6 continued on next page

reasons include: reduced daylight hours in winter leading to poorer visibility; icy or wet road conditions increasing stopping distances and loss of control; and increased holiday travel around Christmas and New Year raising exposure. [1 for describing the seasonal pattern; 1 for a plausible reason]

[2 marks]

(c) Create a Shiny app that takes a `year` from a slider input or select input, and returns the line-and-point plot of the monthly counts of `DriversKilled` over time for that `year`, essentially a zoomed-in version of your answer to Question B6(a) but without the colours. You can copy and paste the following code into the file `app.R`:

```
library(shiny)
library(dplyr)
library(ggplot2)
library(tsbox)
library(tidyr)
library(lubridate)

seatbelts <-
  Seatbelts |>
  ts_tbl() |>
  pivot_wider(names_from = id, values_from = value) |>
  mutate(law = factor(law), year = year(time))

server <- function(input, output, session) {
  ## write your code here
}

ui <- fluidPage(
  ## write your code here
)

shinyApp(ui = ui, server = server)
```

Answer:

`renderPlot()` [1], `filter(year == input$year)` [2], correct `aes(x = time, y = DriversKilled)` [1], `geom_line()` + `geom_point()` [1], axis labels [1], `sliderInput()` or `selectInput()` with correct `min/max/step` or `choices` [1], `plotOutput()` [1].

[8 marks]

(d) In the same Shiny app, and in addition to the plot in Question B6(c), use `seatbelts` and `geom_path()` to make a plot that visualises the

Question B6 continued on next page

co-evolution of `PetrolPrice` (on the x-axis) and `kms` (on the y-axis) over the selected `year`. Label the x- and y-axes `"Petrol price (pence per litre)"` and `"Distance driven (millions of kilometres)"` respectively.

Answer:

`renderPlot()` [1], correct filtering by `input$year` [1], `geom_path()` with correct `aes(x = PetrolPrice, y = kms)` [1], axis labels [1], `plotOutput()` [1].

[5 marks]

[Total: 20 marks]

```
library(shiny)
library(dplyr)
library(ggplot2)
library(tsbox)
library(tidyr)
library(lubridate)

seatbelts <-
  Seatbelts |>
  ts_tbl() |>
  pivot_wider(names_from = id, values_from = value) |>
  mutate(law = factor(law), year = year(time))

server <- function(input, output, session) {
  output$casualty_plot <-
    renderPlot({
      seatbelts_year <- seatbelts |> filter(year == input$year)
      ggplot(seatbelts_year, aes(x = time, y = DriversKilled)) +
        geom_line() +
        geom_point() +
        labs(x = "Year", y = "Monthly counts")
    })
  output$phase_plot <-
    renderPlot({
      seatbelts_year <- seatbelts |> filter(year == input$year)
      ggplot(seatbelts_year, aes(x = PetrolPrice, y = kms)) +
        geom_path() +
        labs(x = "Petrol price (pence per litre)",
             y = "Distance driven (millions of kilometres)")
    })
}

ui <-
```

```
  fluidPage(
    title = "MAS2908 Specimen Exam",
    sliderInput(
      inputId = "year",
      label   = "Select year",
      min     = 1969,
      max     = 1984,
      value   = 1983,
      step    = 1
    ),
    fluidRow(
      column(width = 6, plotOutput("casualty_plot", height = 500)),
      column(width = 6, plotOutput("phase_plot",    height = 500))
    )
  )

shinyApp(ui = ui, server = server)
```

THE END