Assignment 5

Nathan Underwood

2023-11-14

```
library(ggrepel)
```

```
## Warning: package 'ggrepel' was built under R version 4.3.2
```

```
## Loading required package: ggplot2
```

```
library(latex2exp)
```

```
## Warning: package 'latex2exp' was built under R version 4.3.2
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(broom)

data(infmort, package = "faraway")
data(trees)
```

Question 1

The `infmort` data set from the package `faraway` gives the infant mortality rate for a variety of countries. The information is relatively out of date (from 1970s?), but will be fun to graph. Visualize the data using by creating scatter plots of mortality vs income while faceting using `region` and setting color by `oil` export status. Utilize a $\log_{10}$ transformation for both `mortality` and `income` axes. This can be done either by doing the transformation inside the `aes()` command or by utilizing the `scale_x_log10()` or `scale_y_log10()` layers. The critical difference is if the scales are on the original vs log transformed scale. Experiment with both and see which you prefer.

a. The `rownames()` of the table gives the country names and you should create a new column that contains the country names. *`rownames`

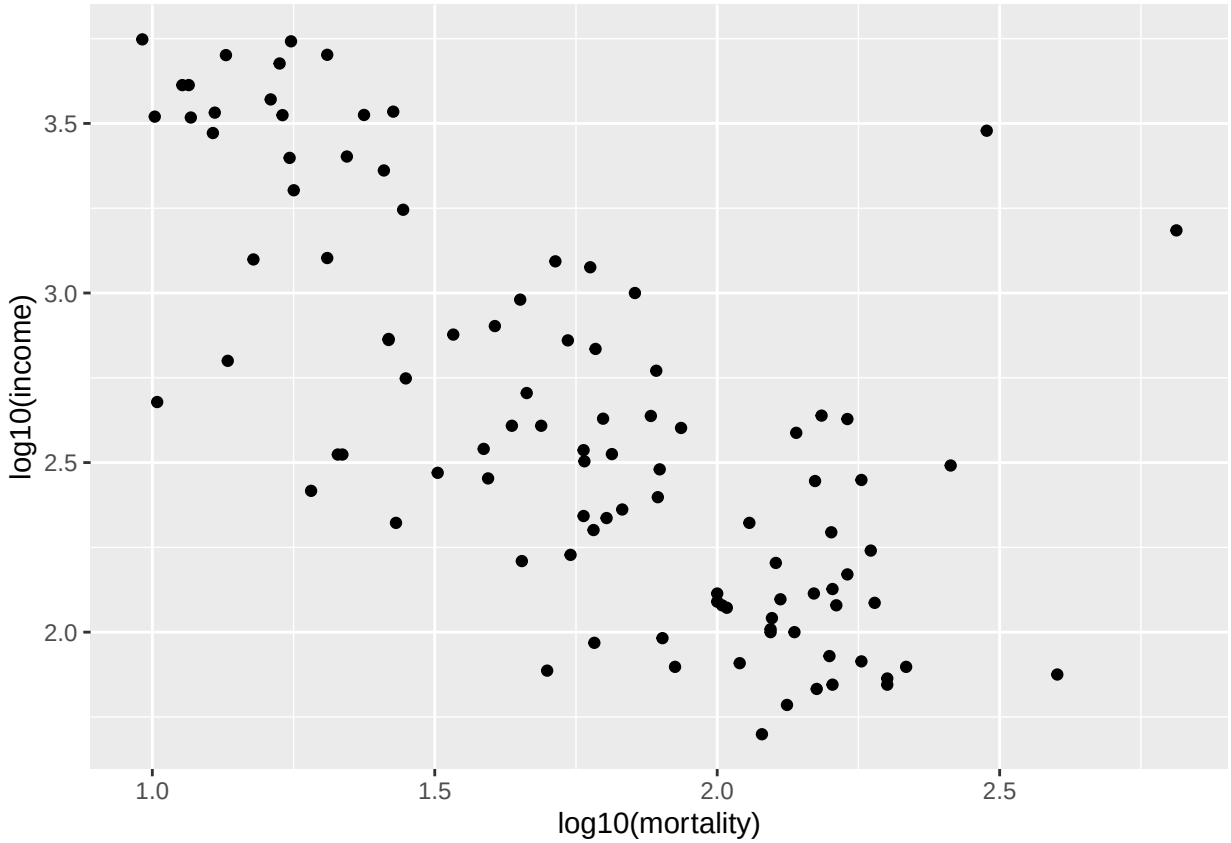
```
infmortNew <- infmort %>%
  mutate(Country = rownames(infmort)) %>%
  relocate(Country) %>%
  drop_na()

head(infmortNew)
```

```
##                                Country   region income mortality
## Australia           Australia               Asia   3426      26.7
## Austria             Austria               Europe   3350      23.7
## Belgium             Belgium               Europe   3346      17.0
## Canada              Canada              Americas   4751      16.8
## Denmark             Denmark               Europe   5029      13.5
## Finland             Finland               Europe   3312      10.1
##                                oil
## Australia           no oil exports
## Austria             no oil exports
## Belgium             no oil exports
## Canada              no oil exports
## Denmark             no oil exports
## Finland             no oil exports
```

b. Create scatter plots with the `log10()` transformation inside the `aes()`command.

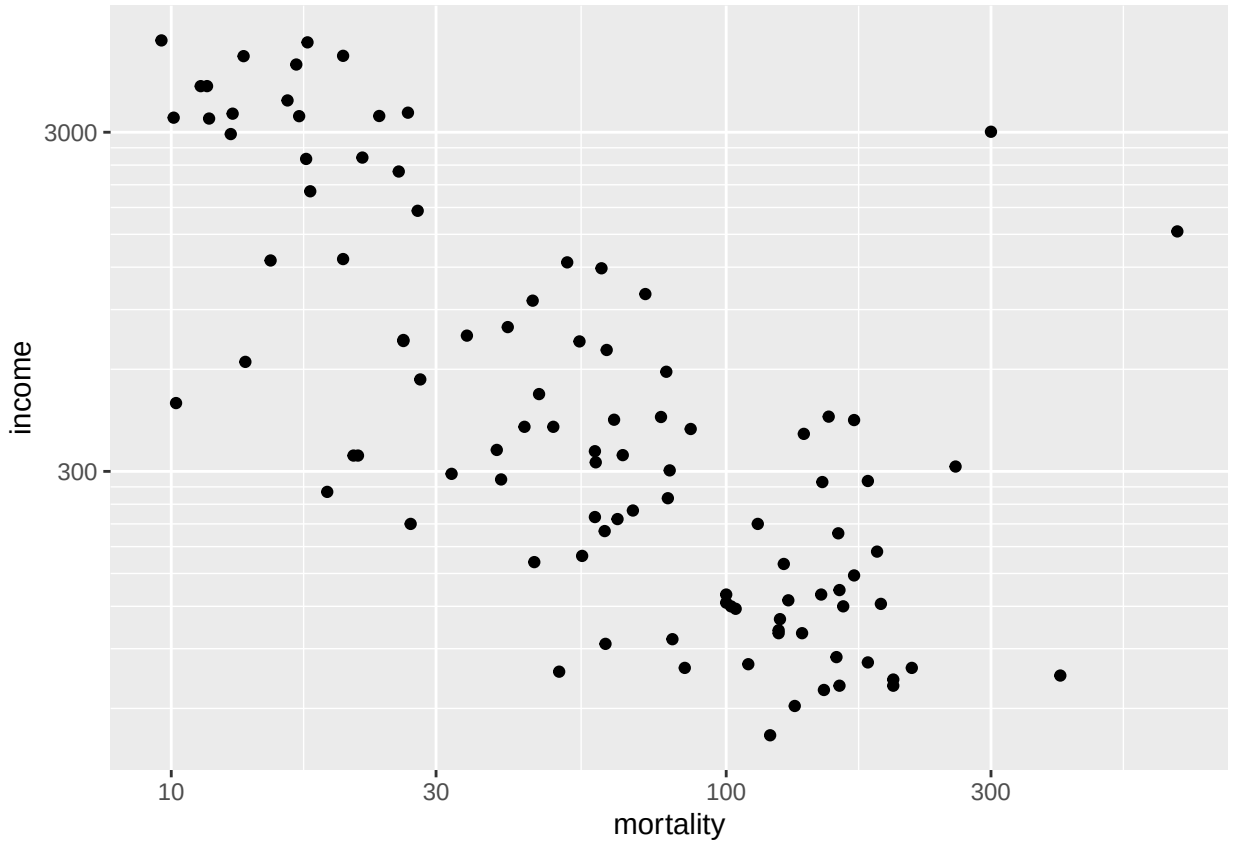
```
ggplot(infmortNew, aes(x = log10(mortality), y = log10(income))) +
  geom_point()
```



c. Create the scatter plots using the `scale_x_log10()` and `scale_y_log10()`. Set the major and minor breaks to be useful and aesthetically pleasing. Comment on which version you find easier to read.

```
ggplot(infmortNew, aes(x = mortality, y = income)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10(breaks = c(30, 300, 3000),
```

```
                minor = c(seq(30, 300, 30),
                          c(seq(300, 3000, 300))))
```

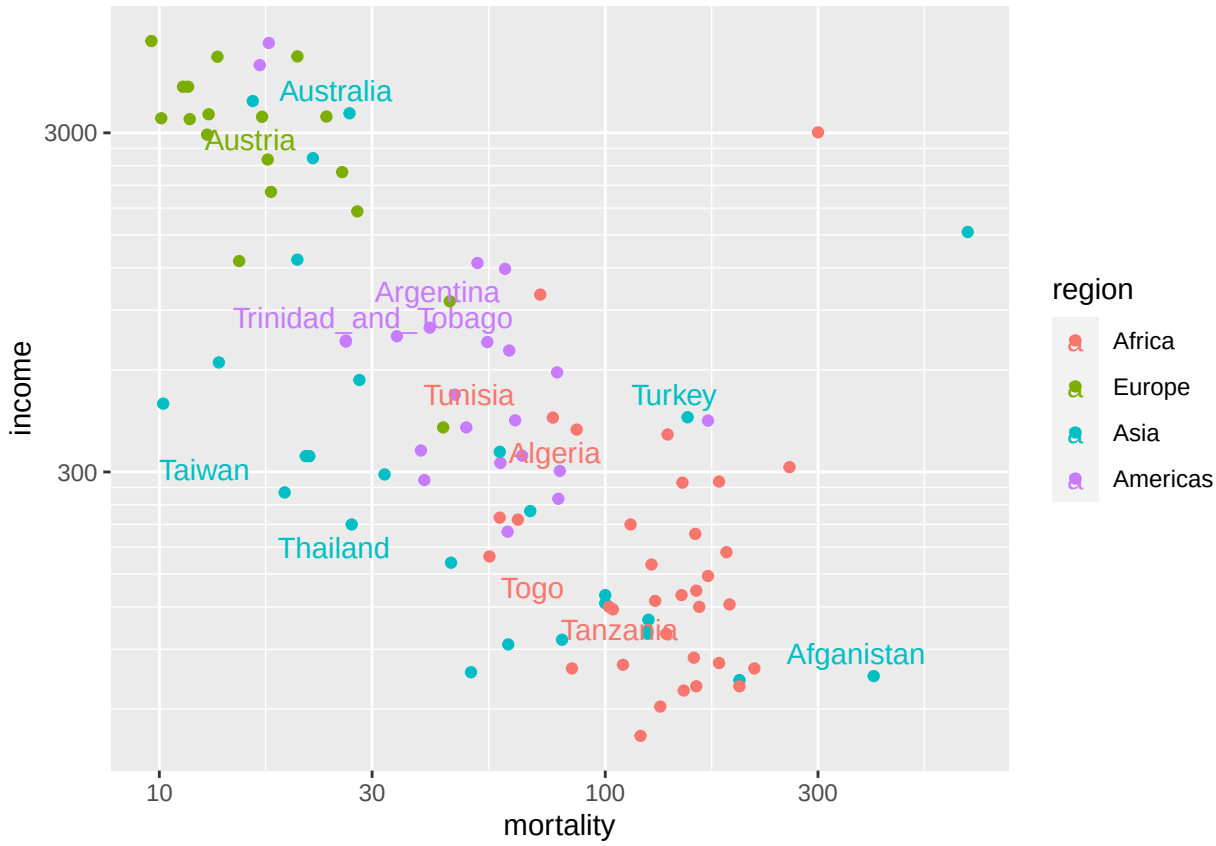


d. The package `ggrepel` contains functions `geom_text_repel()` and `geom_label_repel()` that mimic the basic `geom_text()` and `geom_label()` functions in `ggplot2`, but work to make sure the labels don't overlap. Select 10-15 countries to label and do so using the `geom_text_repel()` function.

```
infmortNew <- infmortNew %>%
  mutate(SelectCountries = ifelse(str_detect(Country, "^A|^T"), Country, NA))

ggplot(infmortNew, aes(x = mortality, y = income, color = region)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10(breaks = c(30, 300, 3000),
                minor = c(seq(30, 300, 30),
                          seq(300, 3000, 300))) +
  geom_text_repel(aes(label = SelectCountries))
```

```
## Warning: Removed 89 rows containing missing values (`geom_text_repel()`).
```



Question 2

Using the `datasets::trees` data, complete the following:

a. Create a regression model for $y =$ `Volume` as a function of $x =$ `Height`.

```
treeModel <- lm(Volume ~ Height, trees)

trees <- trees %>% mutate(fit = fitted(treeModel))
```

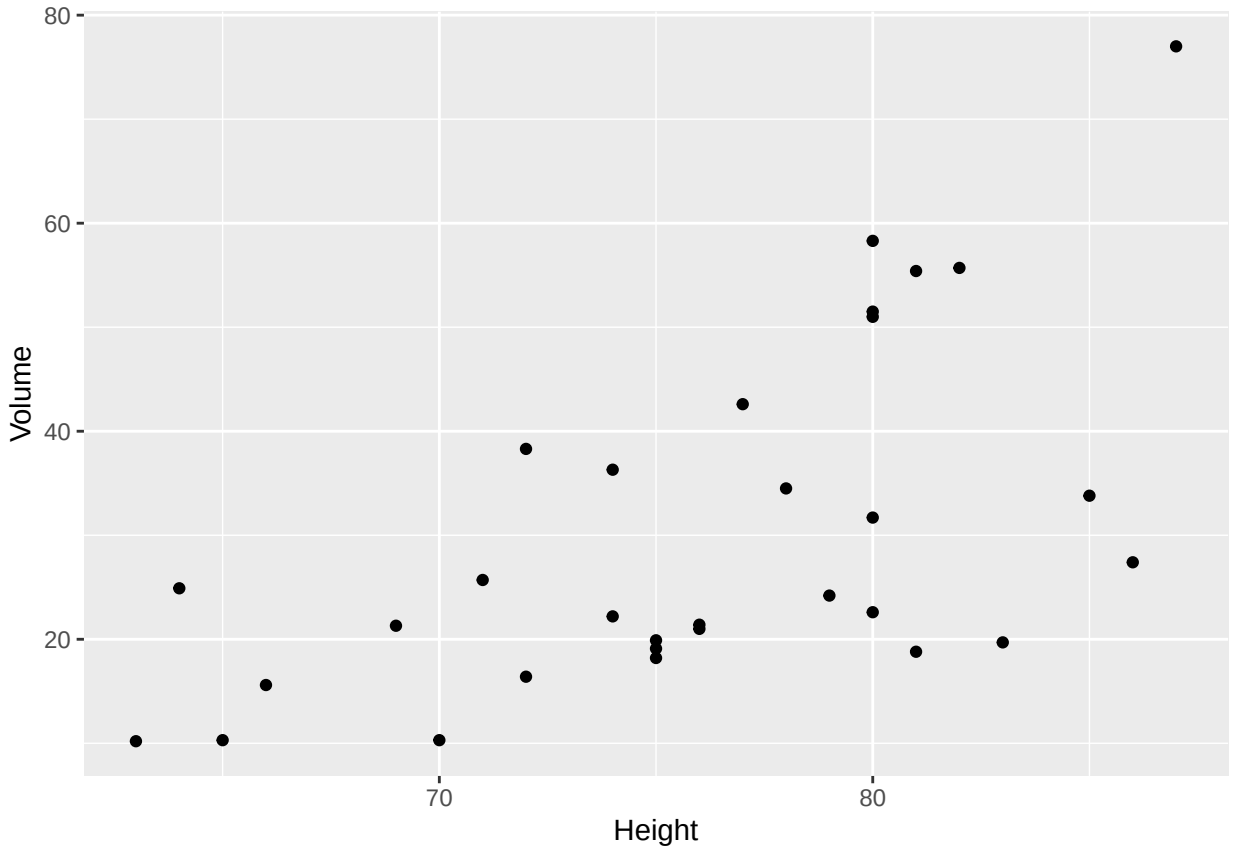
b. Using the `summary` command, get the y-intercept and slope of the regression line.

```
summary <- tidy(treeModel)

intercept <- summary[[1, 2]]
slope <- summary[[2, 2]]
```

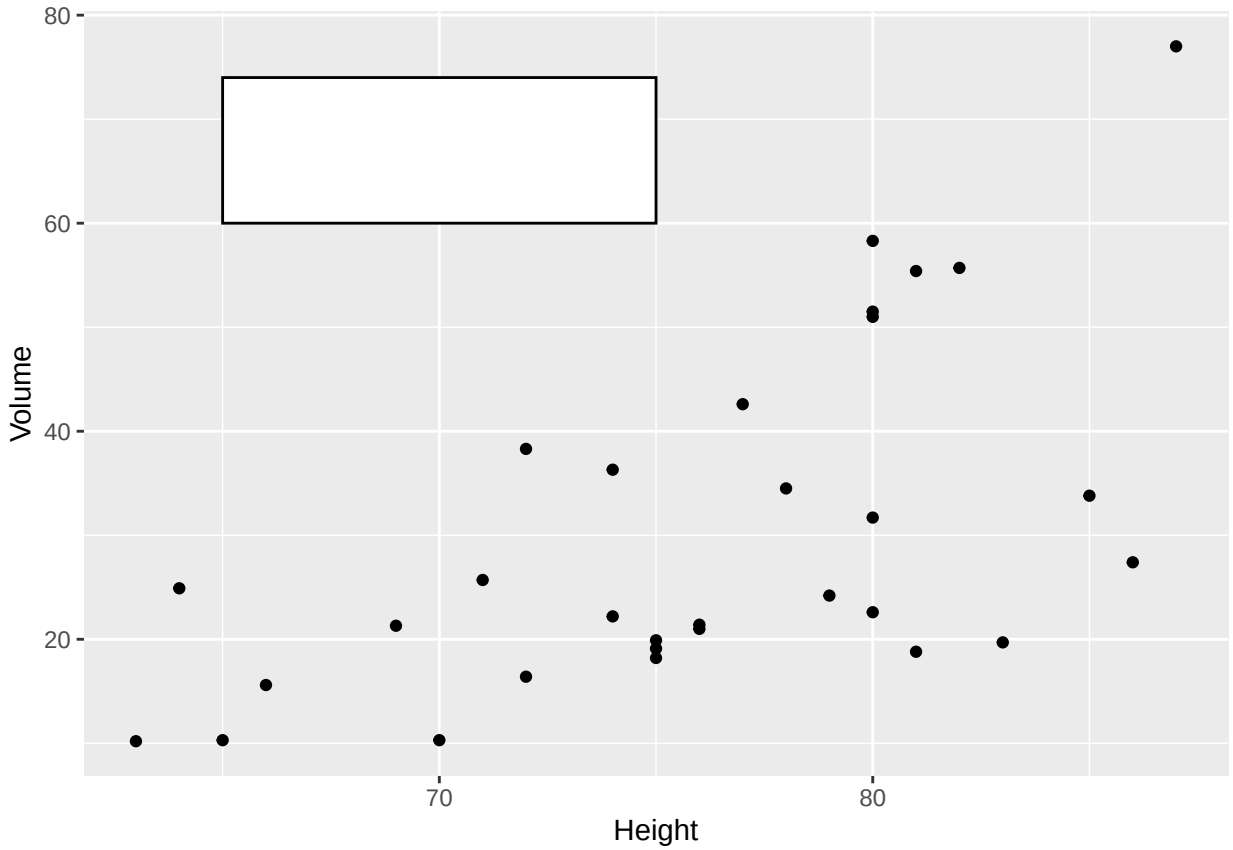
c. Using `ggplot2`, create a scatter plot of Volume vs Height.

```
ggplot(trees, aes(x = Height, y = Volume)) +
  geom_point()
```



d. Create a nice white filled rectangle to add text information to using by adding the following annotation layer.

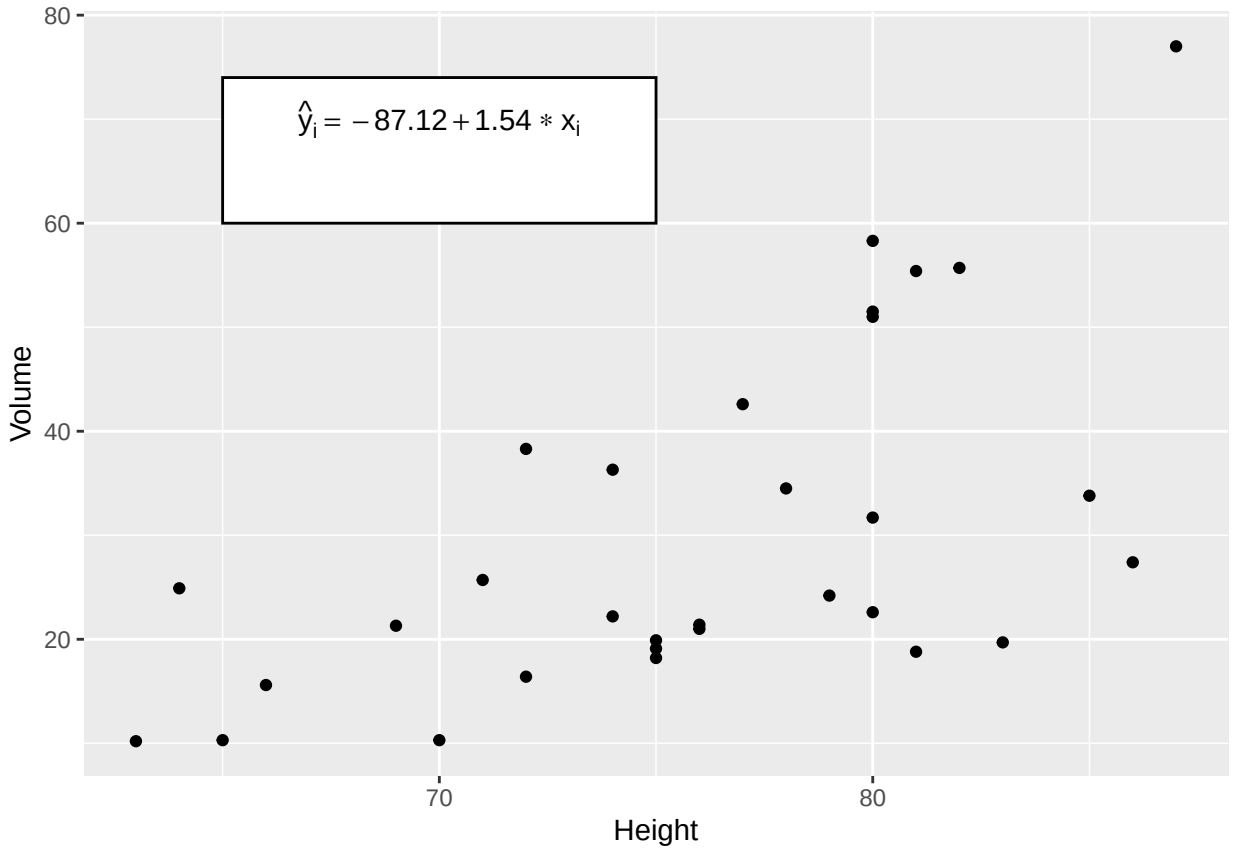
```
ggplot(trees, aes(x = Height, y = Volume)) +
  geom_point() +
  annotate('rect', xmin = 65, xmax = 75, ymin = 60, ymax = 74, fill = 'white', color = 'black')
```



e. Add some annotation text to write the equation of the line $\hat{y}_i = -87.12 + 1.54 * x_i$ in the text area.

```
ggplot(trees, aes(x = Height, y = Volume)) +
  geom_point() +
  annotate('rect', xmin = 65, xmax = 75, ymin = 60, ymax = 74, fill = 'white', color = 'black') +
  annotate("text", x = 70, y = 70, label = TeX("$\\hat{y}_i = -87.12 + 1.54 * x_i$"))
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'
```



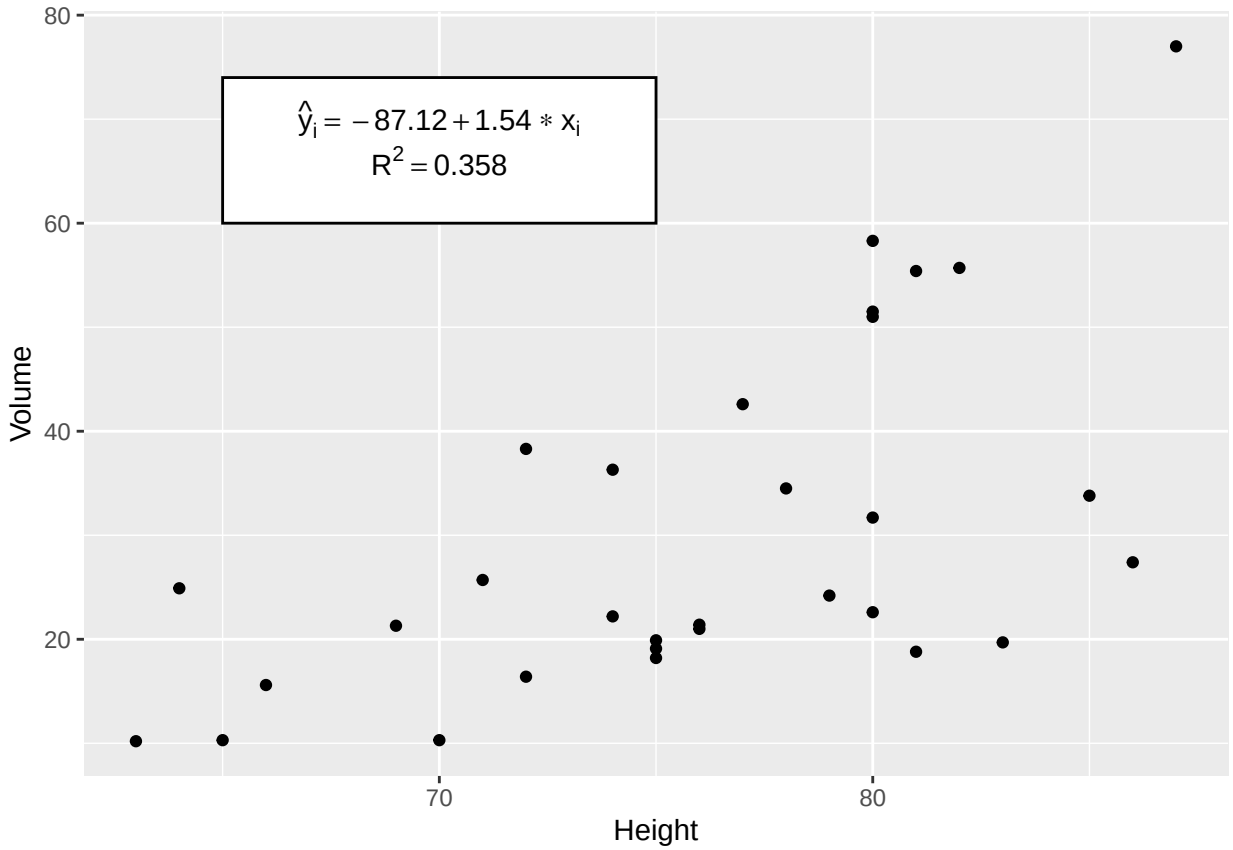
f. Add annotation to add $R^2 = 0.358$

```
plot <- ggplot(trees, aes(x = Height, y = Volume)) +
  geom_point() +
  annotate('rect', xmin = 65, xmax = 75, ymin = 60, ymax = 74, fill = 'white', color = 'black') +
  annotate("text", x = 70, y = 70, label = TeX("$\\hat{y}_i = -87.12 + 1.54 * x_i$")) +
  annotate("text", x = 70, y = 66, label = TeX("$R^2 = 0.358$"))

plot
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'
```



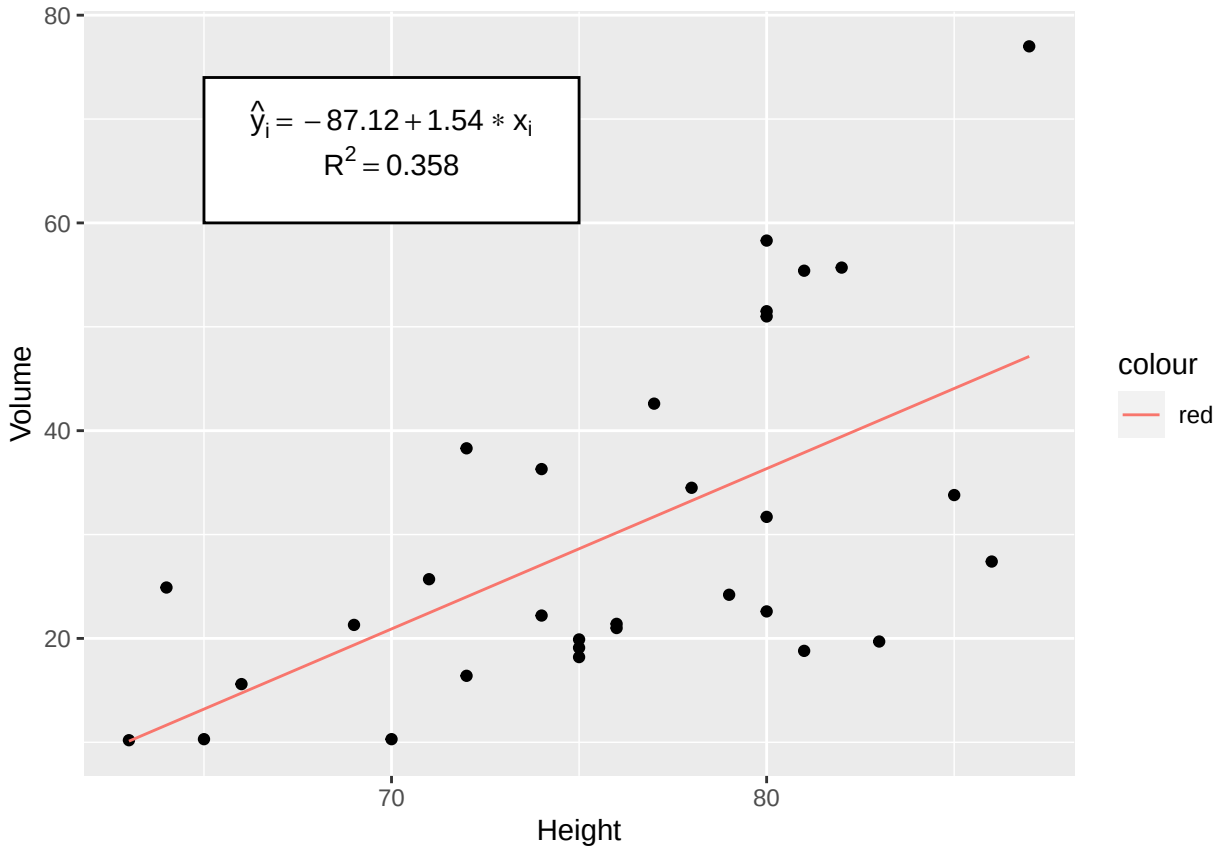
g. Add the regression line in red. The most convenient layer function to uses is `geom_abline()`. It appears that the `annotate` doesn't work with `geom_abline()` so you'll have to call it directly.

```
plot +
  geom_line(aes(y = trees$fit, color = "red"))
```

```
## Warning: Use of `trees$fit` is discouraged.
## i Use `fit` instead.
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'
```




```
# plot + geom_abline(aes(intercept = intercept, slope = slope, color = "red"))
# plot + geom_smooth(method = "lm", se = FALSE, aes(color = "red"))
```