

# Assignment Five

Rebecca Chavez

2023-11-09

## Chapter Fourteen

### Question One

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 The information is relatively out of date (from 1970s?), but will be fun to graph.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'`

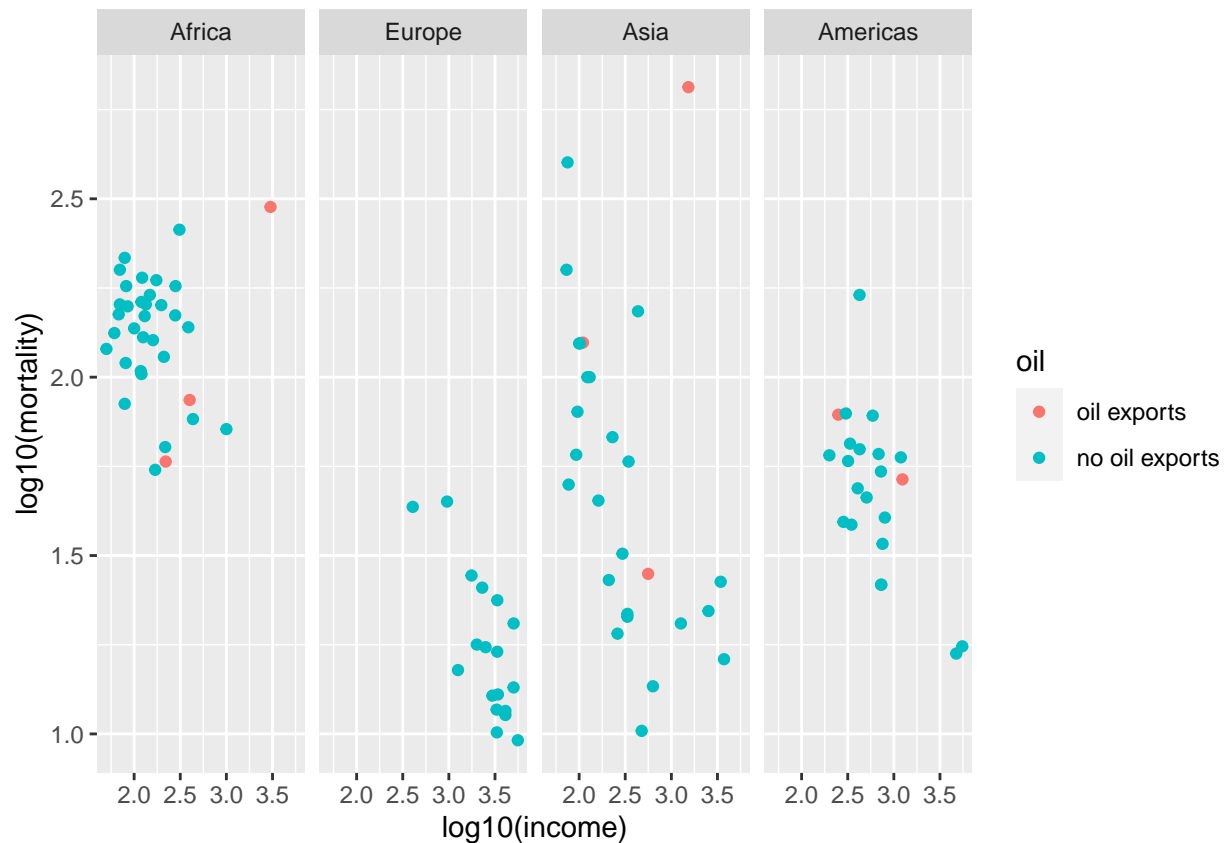
```
data('infmort', package = 'faraway')

infmort <- infmort %>%
  mutate(country = str_trim(rownames(infmort))) %>%
  mutate(country.labels = ifelse(country == 'Austria' | country == 'Ireland' |
                                country == 'Italy' | country == 'Malaysia' |
                                country == 'Taiwan' | country == 'Canada' |
                                country == 'Libya' | country == 'Nigeria' |
                                country == 'Ecuador' | country == 'Chile',
                                country, NA))
```

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

```
ggplot(infmort, aes(x = log10(income), y = log10(mortality))) +
  geom_point(aes(color = oil)) +
  facet_grid(. ~ region)
```

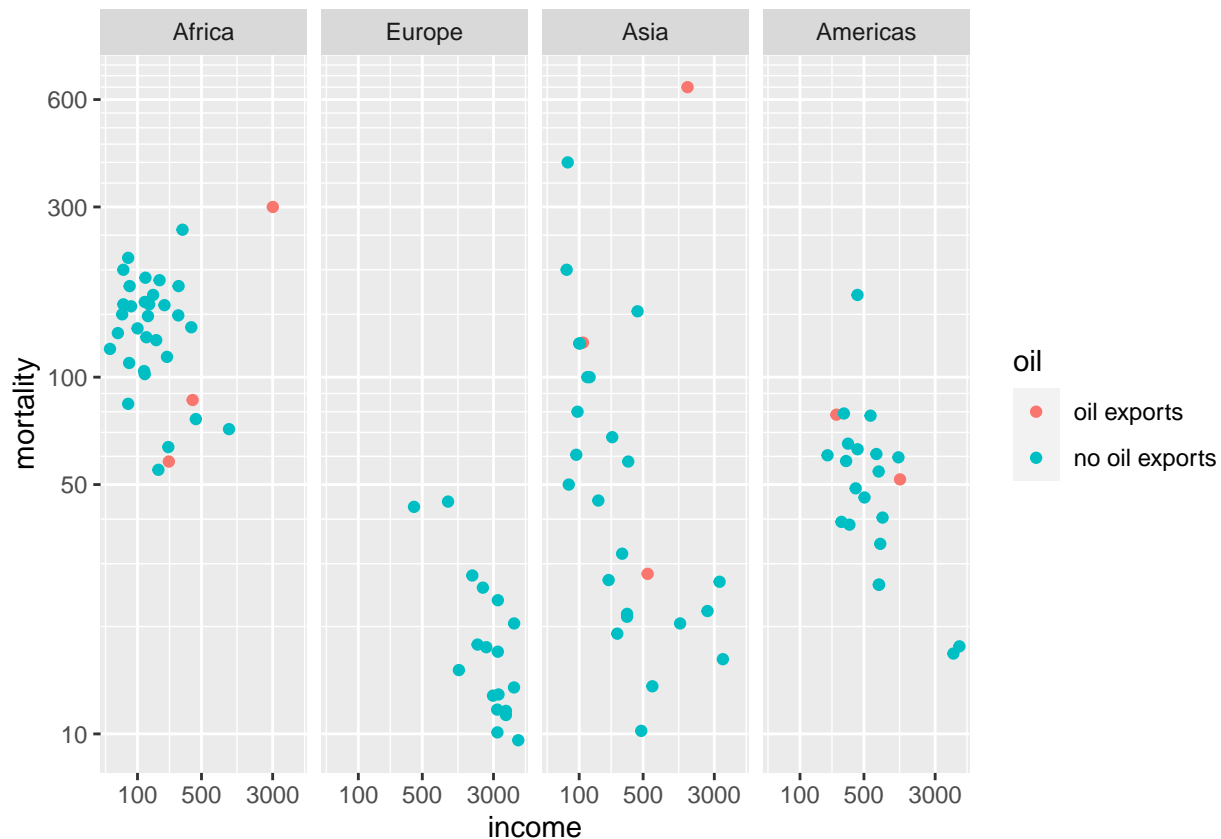
```
## Warning: Removed 4 rows containing missing values ('geom_point()').
```



- 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(infmort, aes(x = income, y = mortality)) +
  geom_point(aes(color = oil)) +
  scale_x_log10(breaks = c(100, 500, 3000)) +
  scale_y_log10(breaks = c(10, 50, 100, 300, 600),
    minor = c(seq(10, 100, by = 10),
      seq(100, 800, by = 50))) +
  facet_grid(. ~ region)
```

```
## Warning: Removed 4 rows containing missing values ('geom_point()').
```



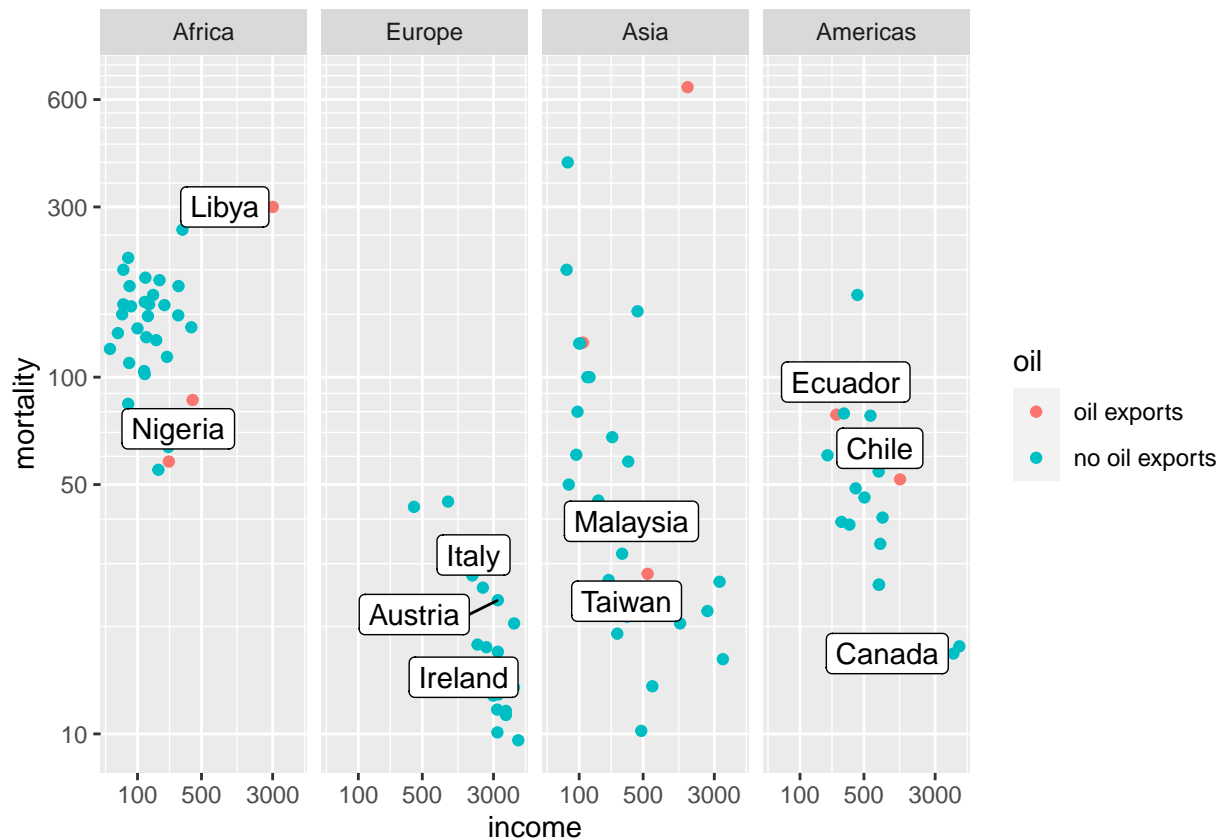
The `scale_x_log10()` and `scale_y_log10()` functions make the graphs easier to read by using the original scales instead of the `log10` values.

- 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.

```
ggplot(infmort, aes(x = income, y = mortality, label = country.labels)) +
  geom_point(aes(color = oil)) +
  scale_x_log10(breaks = c(100, 500, 3000)) +
  scale_y_log10(breaks = c(10, 50, 100, 300, 600),
    minor = c(seq(10, 100, by = 10),
      seq(100, 800, by = 50))) +
  facet_grid(. ~ region) +
  geom_label_repel()
```

```
## Warning: Removed 4 rows containing missing values ('geom_point()').
```

```
## Warning: Removed 95 rows containing missing values ('geom_label_repel()').
```



## Question Two

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

- a) Create a regression model for  $y = \text{'Volume'}$  as a function of  $x = \text{'Height'}$ .

```
data(trees)

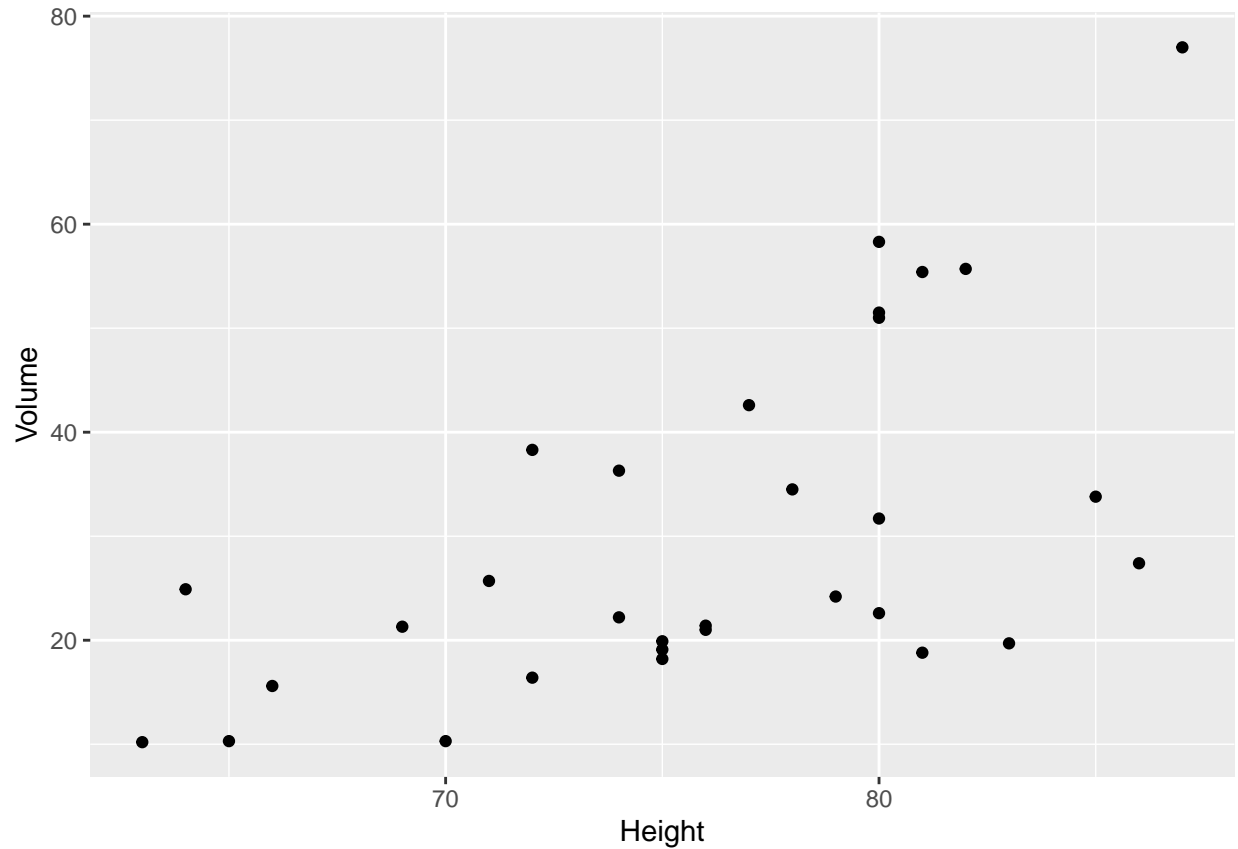
model <- lm(Volume ~ Height, data=trees)
```

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

```
modelCoef <- summary(model)$coef
```

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

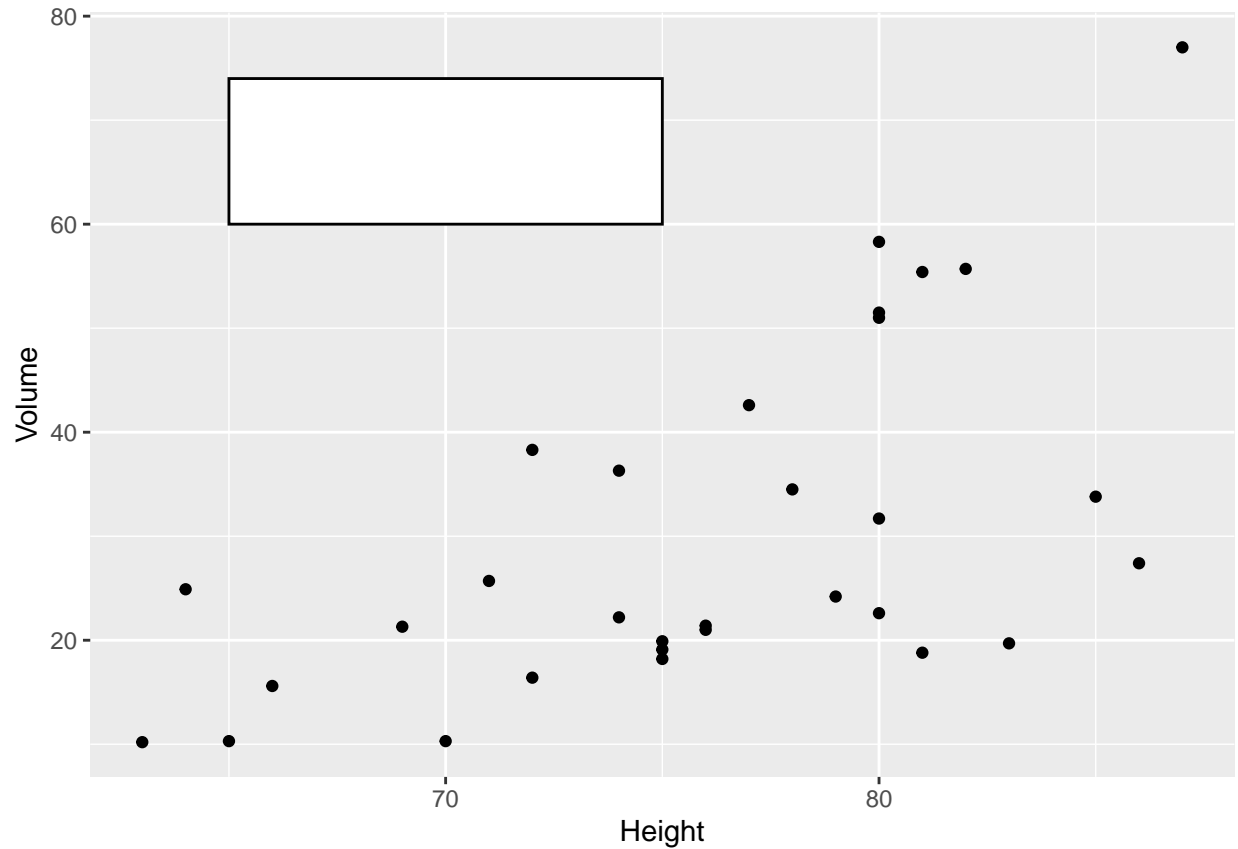
```
ggplot(trees, aes(x = Height, y = Volume)) +
  geom_point()
```



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

```
'''r
annotate('rect', xmin=65, xmax=75, ymin=60, ymax=74,
         fill='white', color='black') +
'''
```

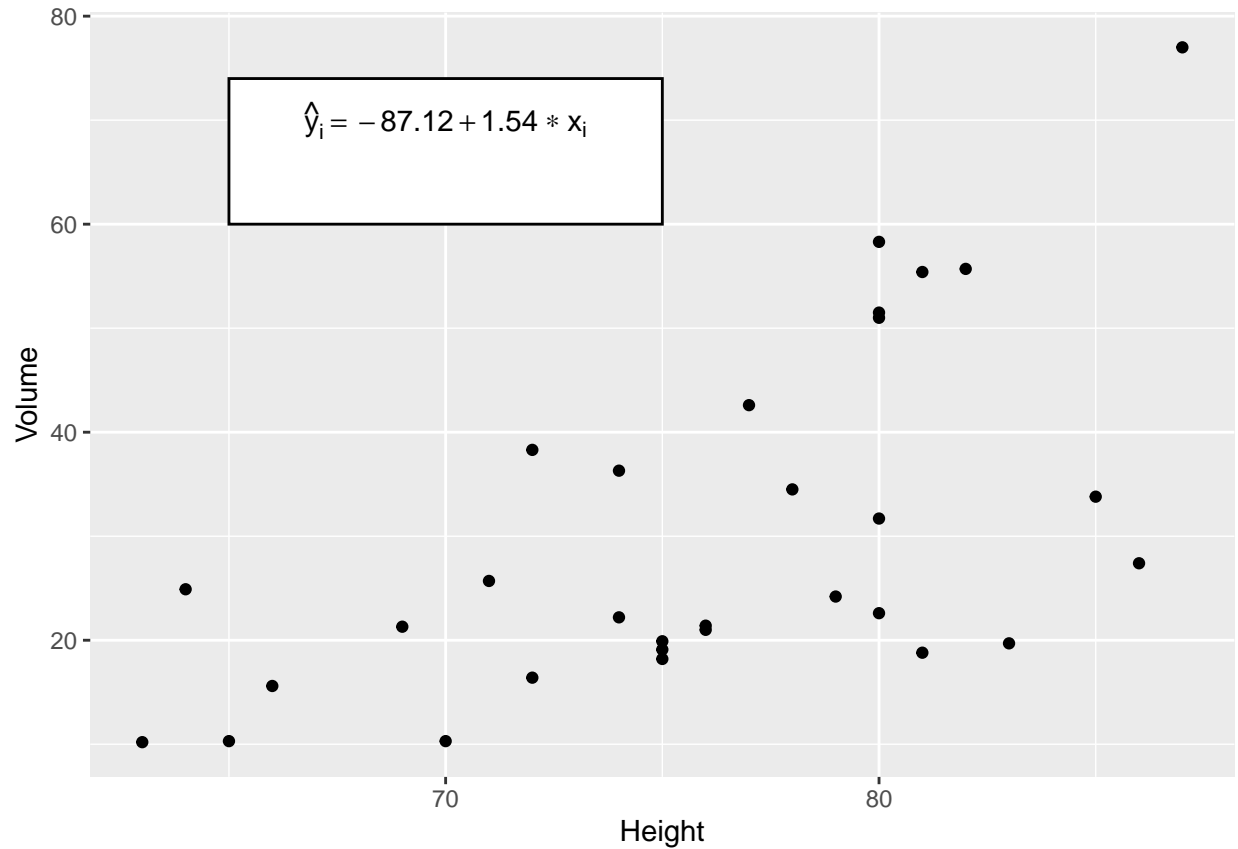
```
ggplot(trees, aes(x = Height, y = Volume)) +
  geom_point() +
  annotate('rect', xmin=65, xmax=75, ymin=60, ymax=74,
         fill='white', color='black')
```



- 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'
```



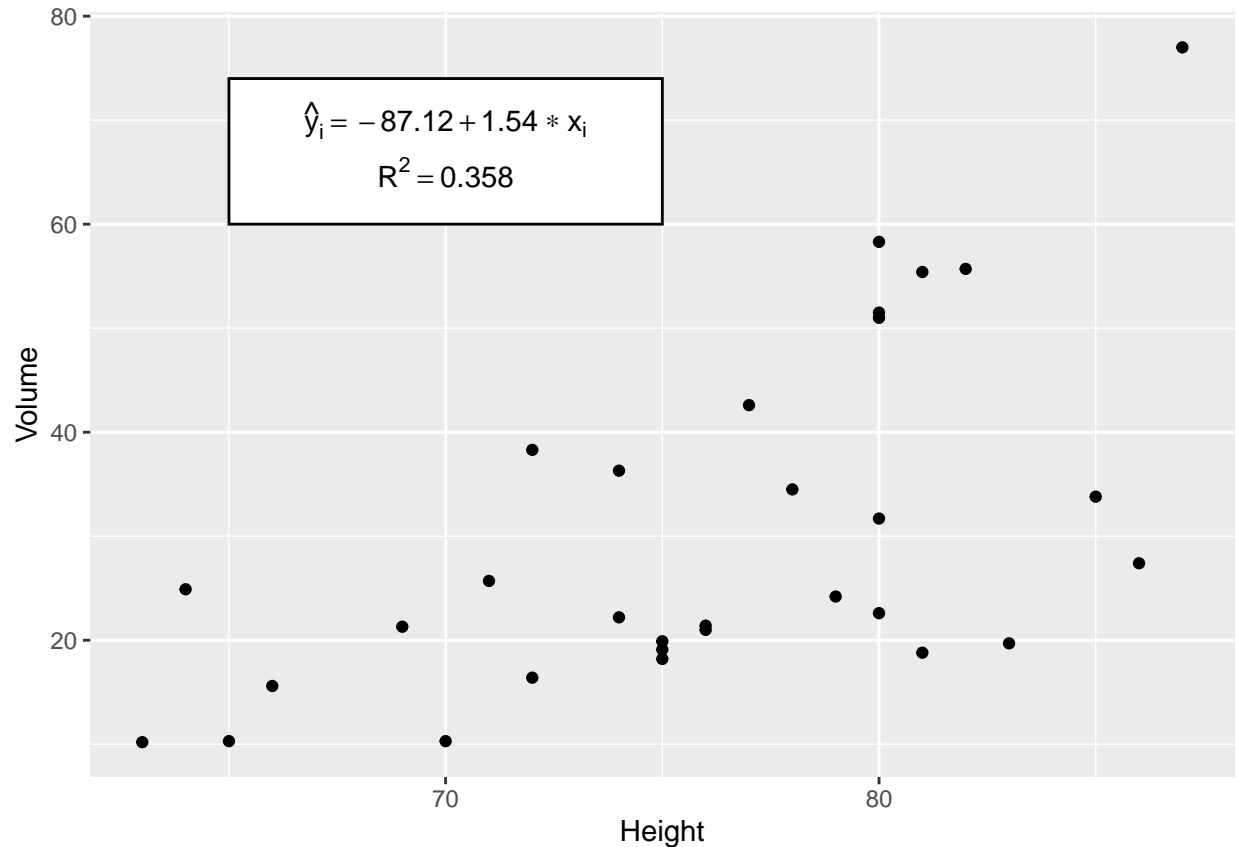
f) Add annotation to add  $R^2 = 0.358$

```
modelRsqr <- summary(model)$r.squared

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 = 65, label = TeX('$R^2 = 0.358$'))
```

```
## 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 use is 'geom\_abline()'. It appears that the 'annotate' doesn't work with 'geom\_abline()' so you'll have to call it directly.

```
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 = 65, label = TeX('$R^2 = 0.358$')) +
  geom_abline(intercept = modelCoef[1], slope = modelCoef[2])
```

```
## 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'
```



