

# STA 445 Assignment 5

Kaylin McLiverty

2023-11-15

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 The information is relatively out of date (from 1970s?), but will be fun to graph.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

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

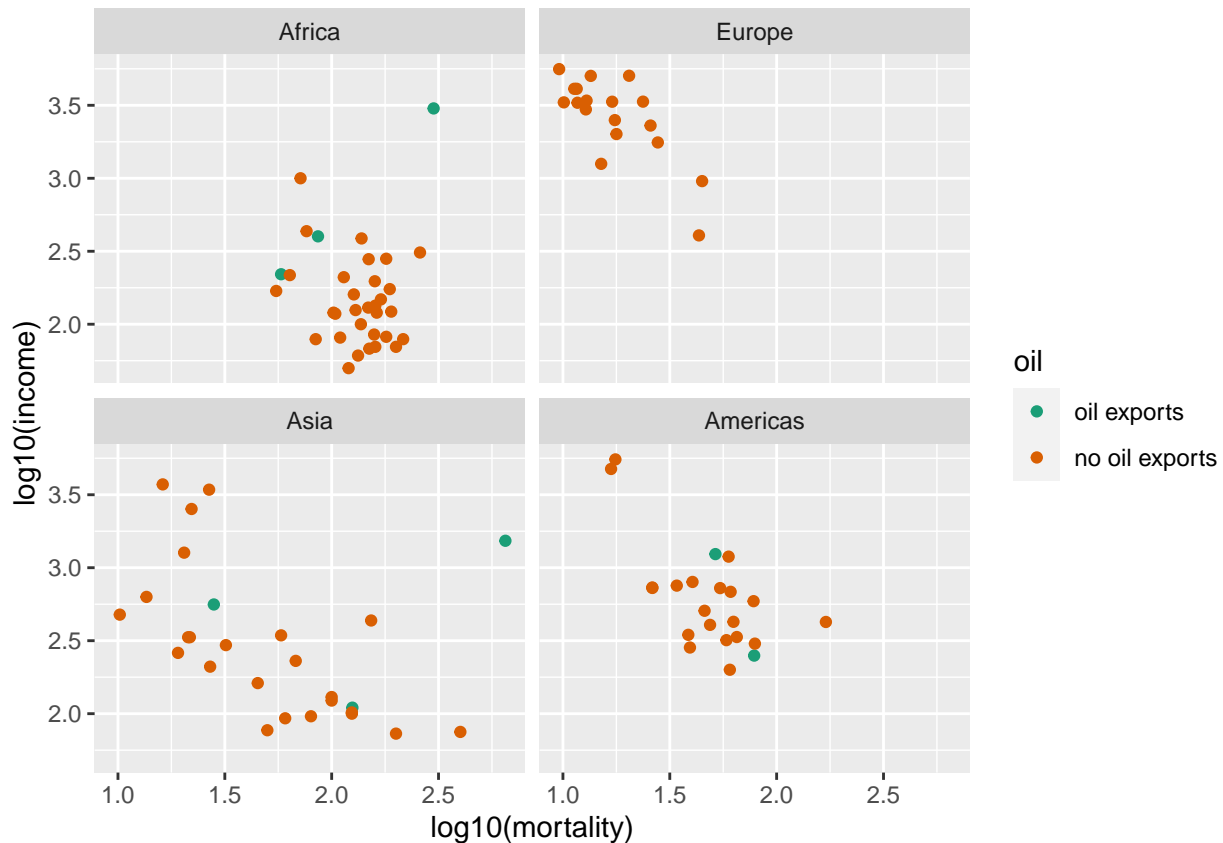
infmort1 <- infmort %>% mutate( country = rownames(infmort) )
head(infmort1)

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

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

```
P1 <- ggplot(infmort1, aes(x=log10(mortality), y=log10(income))) +
  geom_point(aes(color=oil)) +
  facet_wrap( ~region )+
  scale_color_brewer(palette='Dark2')
P1
```

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

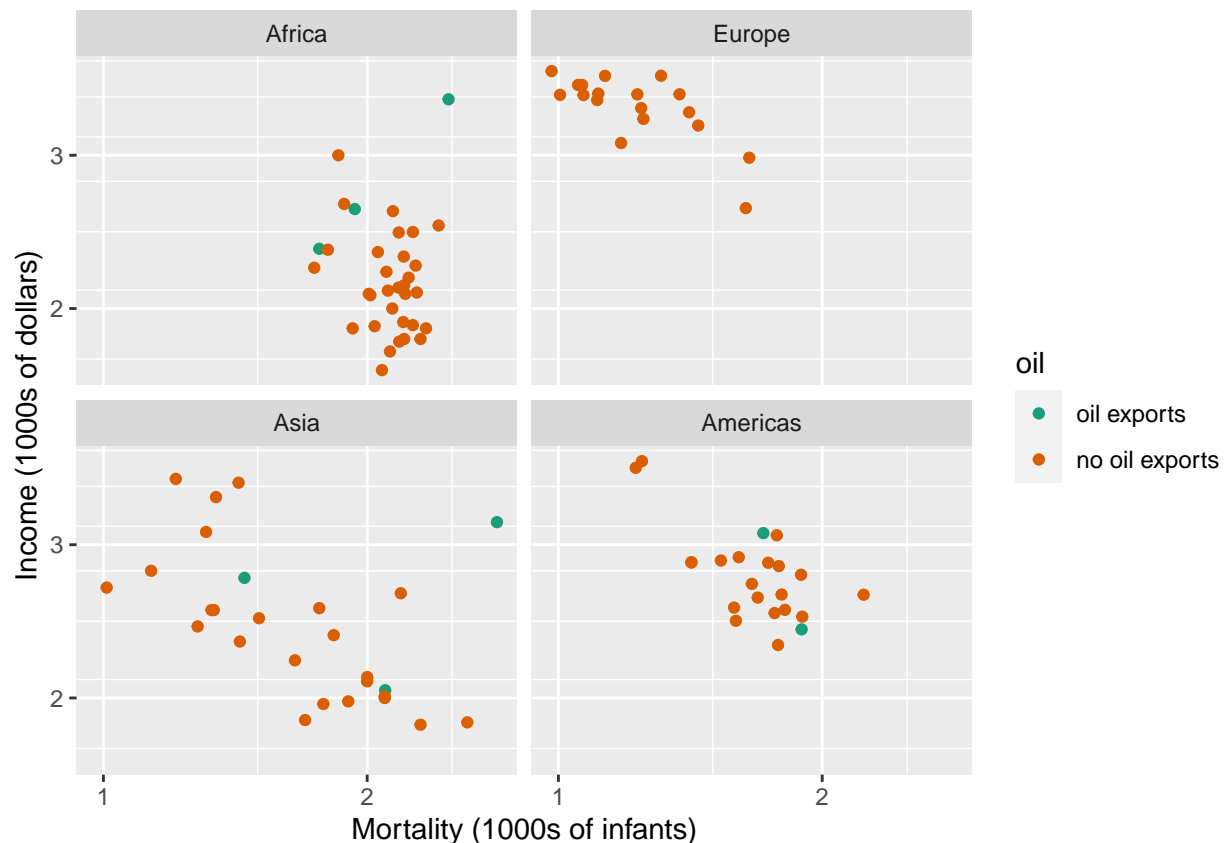


- c) Create the scatter plots using the ``scale_x_log10()`` and ``scale_y_log10()``. Set the major and minor breaks to be useful and aesthetically pleasing. Comment on which version you find easier to read.

*I personally think that the version without the `'scale_x_log10()'` is easier to read.*

```
P2 <- ggplot(infmort1, aes(x=log10(mortality), y=log10(income))) +
  geom_point(aes(color=oil)) +
  facet_wrap( ~region )+
  scale_color_brewer(palette='Dark2')+
  scale_x_log10(breaks=(0:3),
                minor_breaks=seq(0,3, by=.5))+
  xlab('Mortality (1000s of infants)')+
  scale_y_log10(breaks=(0:4),
                minor_breaks=seq(0,4, by=.35))+
  ylab('Income (1000s of dollars)')
P2
```

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



- 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.

```
P3 <- ggplot(infmort1, aes(x=log10(mortality), y=log10(income))) +
  geom_point(aes(color=oil)) +
  facet_wrap( ~region )+
  scale_color_brewer(palette='Dark2')+
  scale_x_log10(breaks=(0:3),
               minor_breaks=seq(0,3, by=.5))+
  xlab('Mortality (1000s of infants)')+
  scale_y_log10(breaks=(0:4),
               minor_breaks=seq(0,4, by=.35))+
  ylab('Income (1000s of dollars)')+
  geom_text_repel(aes(label=country))
```

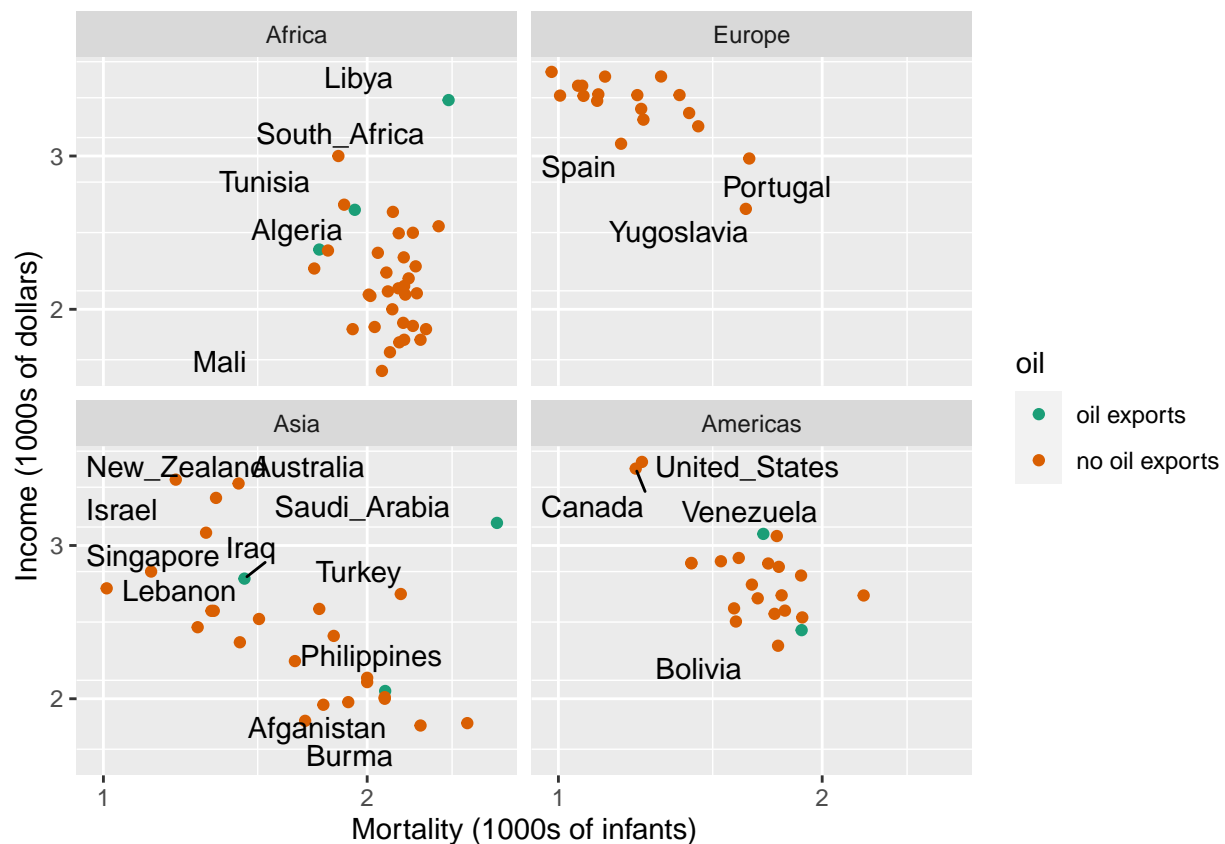
P3

```
## Warning: Removed 4 rows containing missing values (`geom_point()`).
## Warning: Removed 4 rows containing missing values (`geom_text_repel()`).
## Warning: ggrepel: 29 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
## Warning: ggrepel: 16 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
## Warning: ggrepel: 15 unlabeled data points (too many overlaps). Consider
```

```
## increasing max.overlaps
```

```
## Warning: ggrepel: 18 unlabeled data points (too many overlaps). Consider
```

```
## increasing max.overlaps
```



## Exercise 2

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

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

```
data(trees)
```

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

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

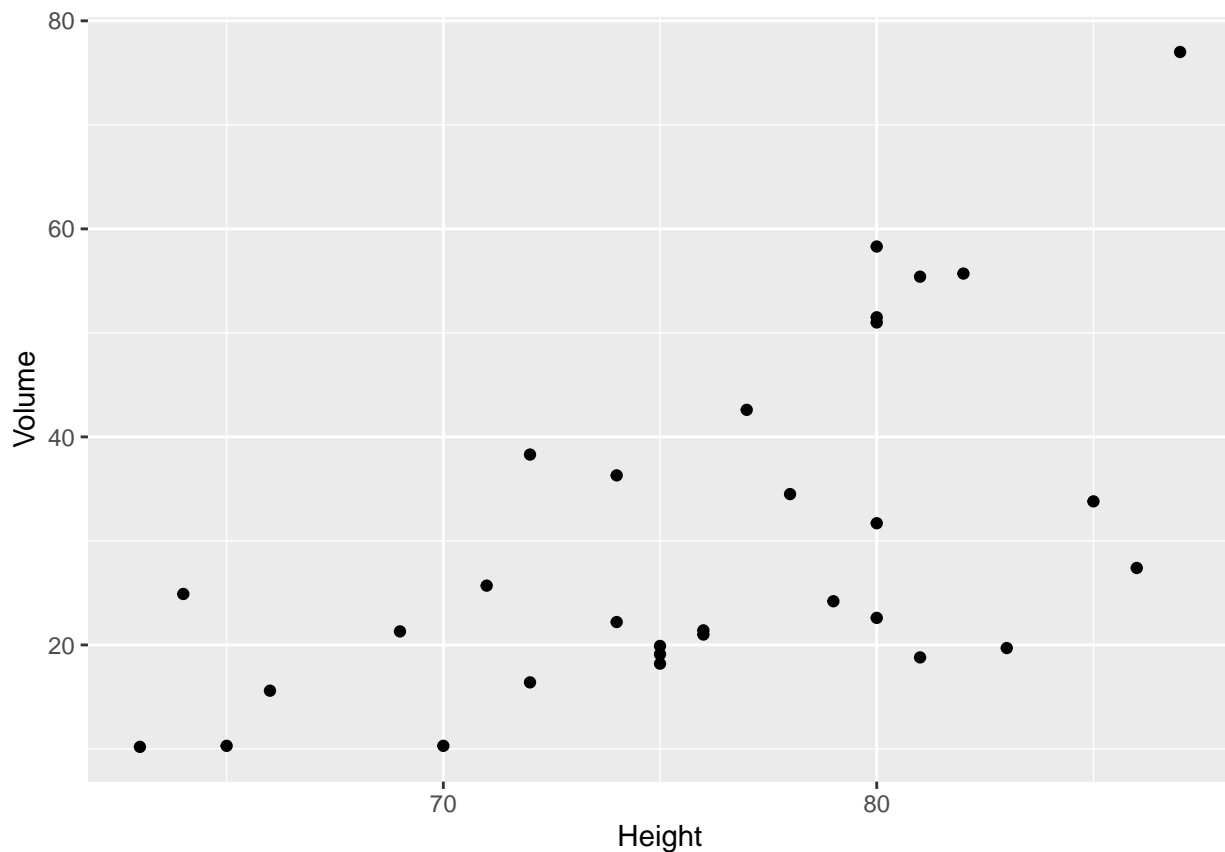
```
summary(model)
```

```
##
## Call:
## lm(formula = Volume ~ Height, data = trees)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.274  -9.894  -2.894   12.068   29.852
##
## Coefficients:
```

```
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -87.1236    29.2731  -2.976 0.005835 **
## Height      1.5433     0.3839   4.021 0.000378 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.4 on 29 degrees of freedom
## Multiple R-squared:  0.3579, Adjusted R-squared:  0.3358
## F-statistic: 16.16 on 1 and 29 DF,  p-value: 0.0003784
trees <- trees %>% mutate(fit=fitted(model))
```

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

```
P4 <- ggplot(trees, aes(x=Height, y=Volume)) +
  geom_point()
P4
```



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

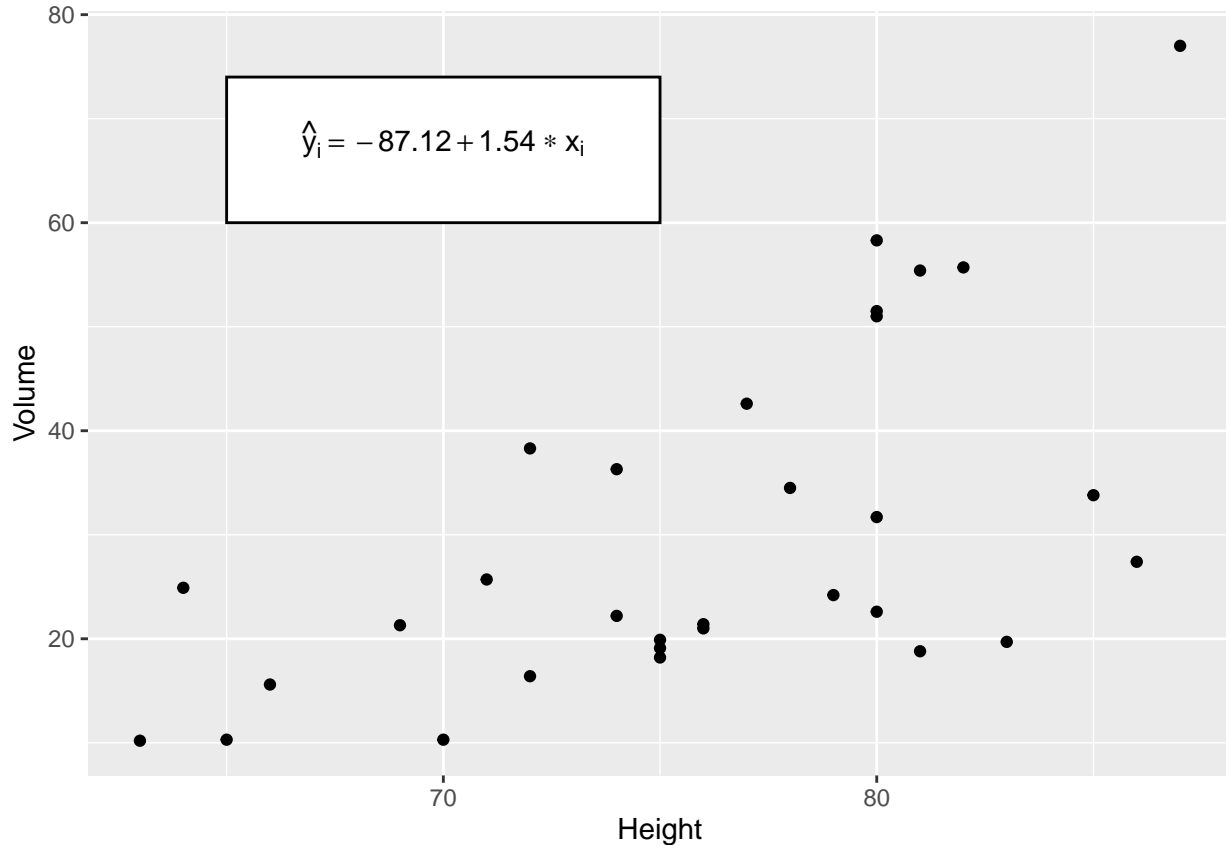
```
P4 <- ggplot(trees, aes(x=Height, y=Volume)) +
  annotate('rect', xmin=65, xmax=75, ymin=60, ymax=74,
         fill='white', color='black') +
  geom_point()
P4
```

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.

```
P4 <- ggplot(trees, aes(x=Height, y=Volume)) +
  annotate('rect', xmin=65, xmax=75, ymin=60, ymax=74,
          fill='white', color='black') +
  annotate('text', x=70, y=68,
          label=latex2exp::TeX('$\\hat{y}_i = -87.12 + 1.54 * x_i$')) +
  geom_point()
```

P4

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



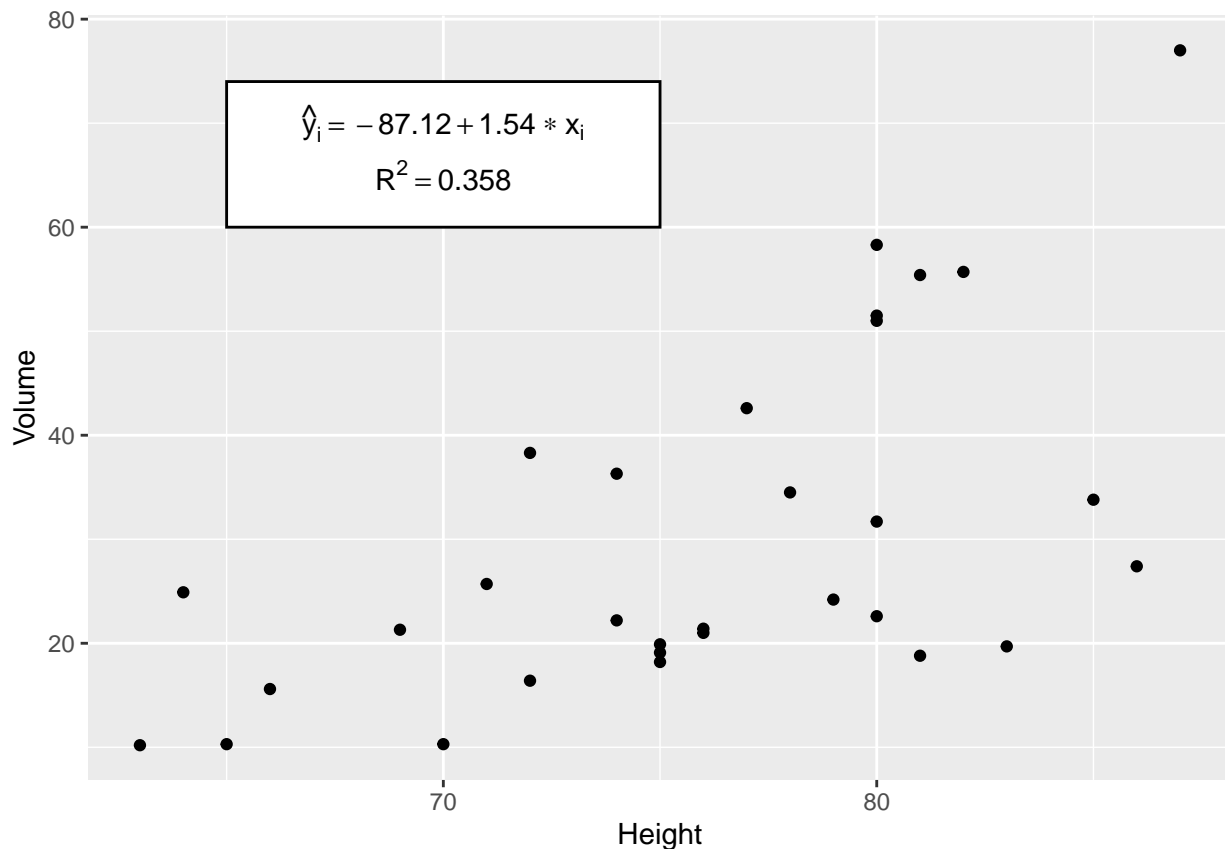
f) Add annotation to add  $R^2 = 0.358$

```
P4 <- ggplot(trees, aes(x=Height, y=Volume)) +
  annotate('rect', xmin=65, xmax=75, ymin=60, ymax=74,
          fill='white', color='black') +
  annotate('text', x=70, y=70,
          label=latex2exp::TeX('$\\hat{y}_i = -87.12 + 1.54 * x_i$')) +
  annotate('text', x=70, y=65, label=latex2exp::TeX('$R^2 = 0.358$')) +
  geom_point()
```

P4

```
## 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'
```



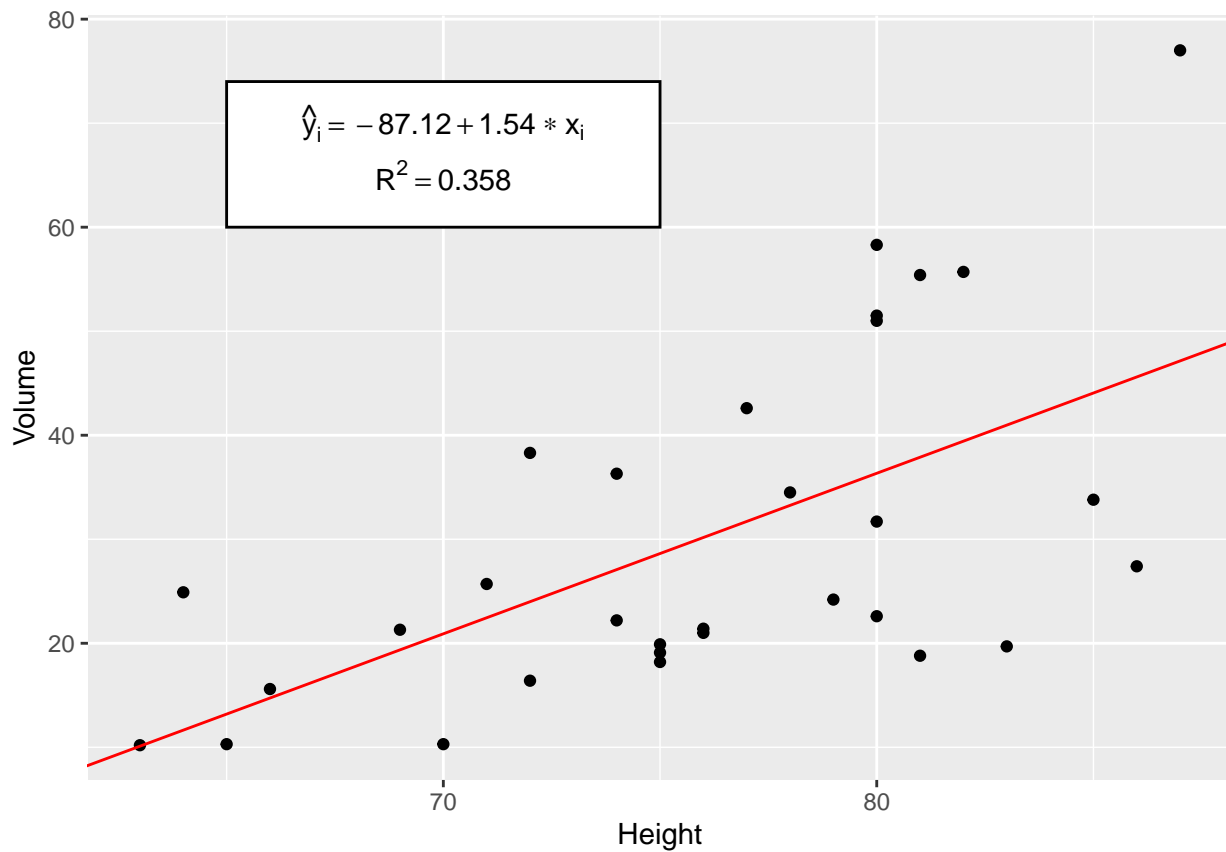
g) Add the regression line in red. The most convenient layer function to use is `geom_abline()`. It appears that the `annotate` doesn't work with `geom_abline()` so you'll have to call it directly.

```
P4 <- ggplot(trees, aes(x=Height, y=Volume)) +
  annotate('rect', xmin=65, xmax=75, ymin=60, ymax=74,
         fill='white', color='black') +
  annotate('text', x=70, y=70,
         label=latex2exp::TeX('$\\hat{y}_i = -87.12 + 1.54 * x_i$')) +
  annotate('text', x=70, y=65, label=latex2exp::TeX('$R^2 = 0.358$')) +
  geom_point() +
  geom_abline(intercept=-87.1236, slope=1.5433, color='red')
```

P4

```
## 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'
```



I wanted to compare the two graphs. The first one uses `geom_abline` and the second uses `geom_line` with the column of fitted values that I mutated

```
P4 <- ggplot(trees, aes(x=Height, y=Volume)) +
  annotate('rect', xmin=65, xmax=75, ymin=60, ymax=74,
         fill='white', color='black') +
  annotate('text', x=70, y=70,
         label=latex2exp::TeX('$\\hat{y}_i = -87.12 + 1.54 * x_i$')) +
  annotate('text', x=70, y=65, label=latex2exp::TeX('$R^2 = 0.358$')) +
  geom_point() +
  geom_line(aes(y=fit), color='red')
```

P4

```
## 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'
```



