445-14

2023-11-09

Exercises

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 The information is relatively out of date (from 1970s?), but will be fun to graph.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

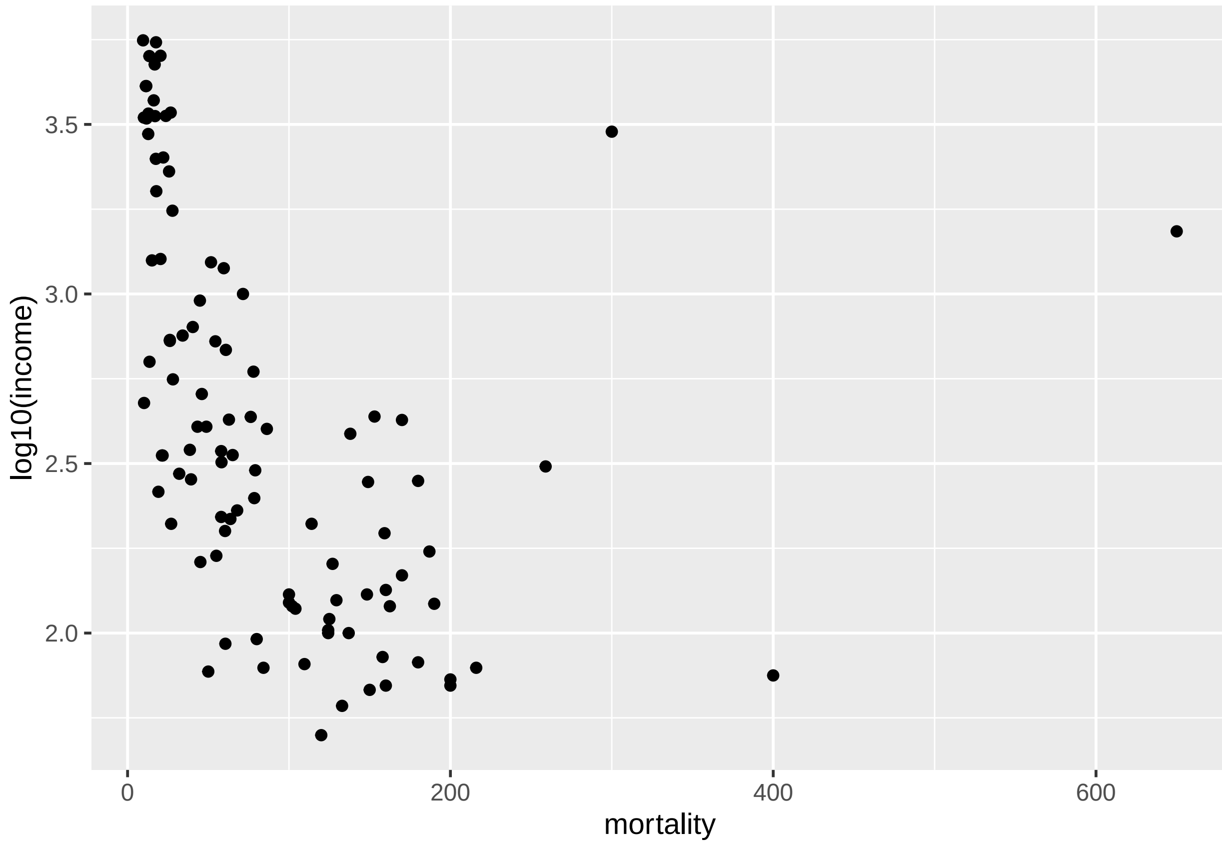
   a) The `rownames()` of the table gives the country names and you should create a new column that contains the country names. *`rownames`

```
data(infmort, package='faraway')
rownames <- rownames(infmort) # new vector with country names
infmort <- infmort %>%
  cbind(rownames) # make 'rownames' a new col
head(infmort)
```

```
##                        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
##                          rownames
## Australia              Australia
## Austria                Austria
## Belgium                Belgium
## Canada                 Canada
## Denmark                Denmark
## Finland                Finland
```

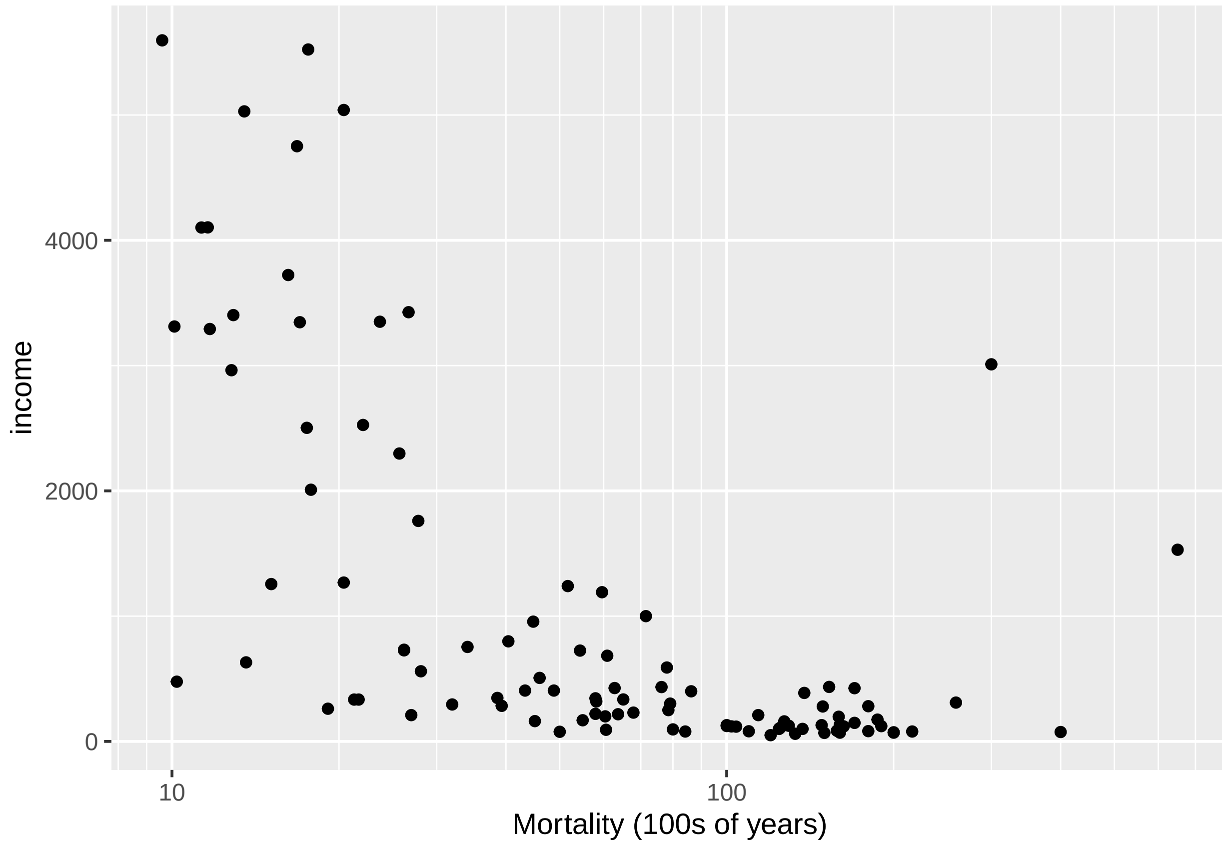
```
b)  Create scatter plots with the `log10()` transformation inside the
`aes()` command.
```

```
infmort <- infmort %>%
  drop_na() # remove NA values
ggplot(infmort, aes(x=mortality, y=log10(income))) +
  geom_point()     # mortality vs income with log10
```

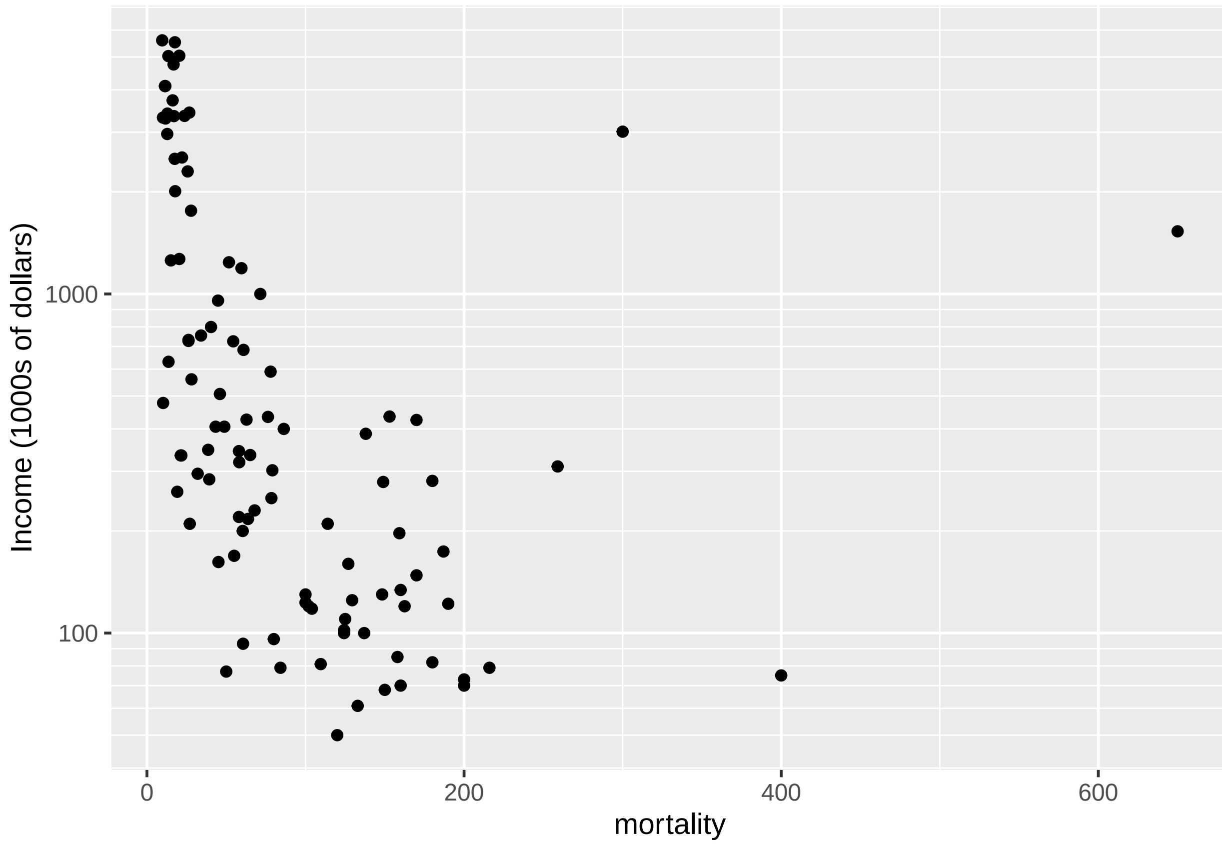


c) Create the scatter plots using the `scale_x_log10()` and `scale_y_log10()`. Set the major and minor breaks to be useful and aesthetically pleasing. Comment on which version you find easier to read. I found scaling the y-axis with log10 to be more helpful because the y-axis deals with a much larger range of numbers.

```
ggplot(infmort, aes(x=mortality, y=income)) +
  geom_point() +
  scale_x_log10(breaks=10^(0:3), # major breaks from 0 -> 1,000
                minor  = outer(seq(0,10,by=1), 10^(0:3)) # 9 minor breaks
               ) +                                       # in each major break
               xlab('Mortality (100s of years)')
```



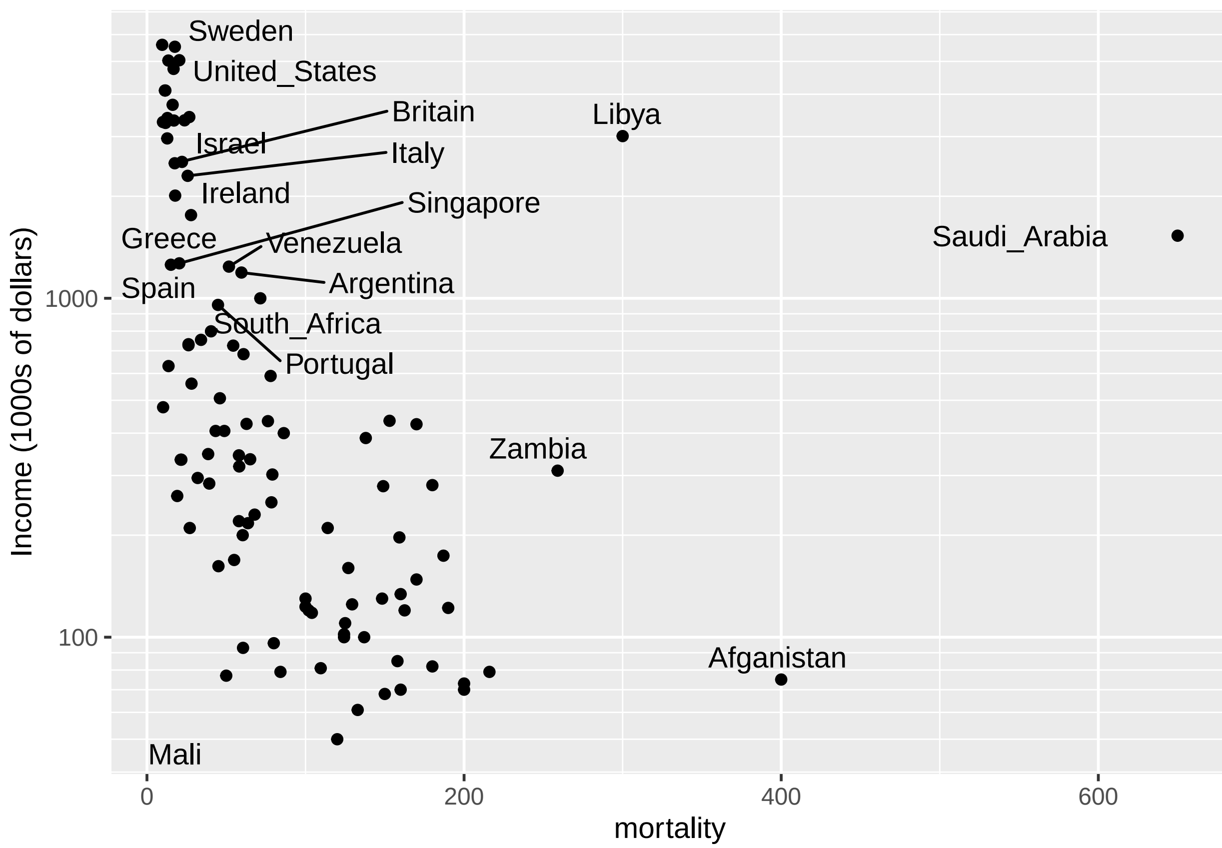

```
ggplot(infmort, aes(x=mortality, y=income)) +
  geom_point() +
  scale_y_log10(breaks=10^(0:4), # major breaks from 0 -> 10,000
               minor  = outer(seq(0,10,by=1), 10^(0:4)) # 9 minor breaks
              ) +                                       # in each major break
              ylab('Income (1000s of dollars)')
```



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=mortality, y=income)) +
  geom_point() +
  scale_y_log10(breaks=10^(0:4),
                minor  = outer(seq(0,10,by=1), 10^(0:4))) +
  ylab('Income (1000s of dollars)') +
  ggrepel::geom_text_repel(data = infmort,aes(label = rownames))
```

```
## Warning: ggrepel: 83 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



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

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

```
head(datasets::trees)
```

```
##   Girth Height Volume
## 1   8.3     70   10.3
## 2   8.6     65   10.3
## 3   8.8     63   10.2
## 4  10.5     72   16.4
## 5  10.7     81   18.8
## 6  10.8     83   19.7
```

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

```
##
## Call:
## lm(formula = Volume ~ Height, data = trees)
##
## Coefficients:
## (Intercept)       Height
##     -87.124        1.543
```

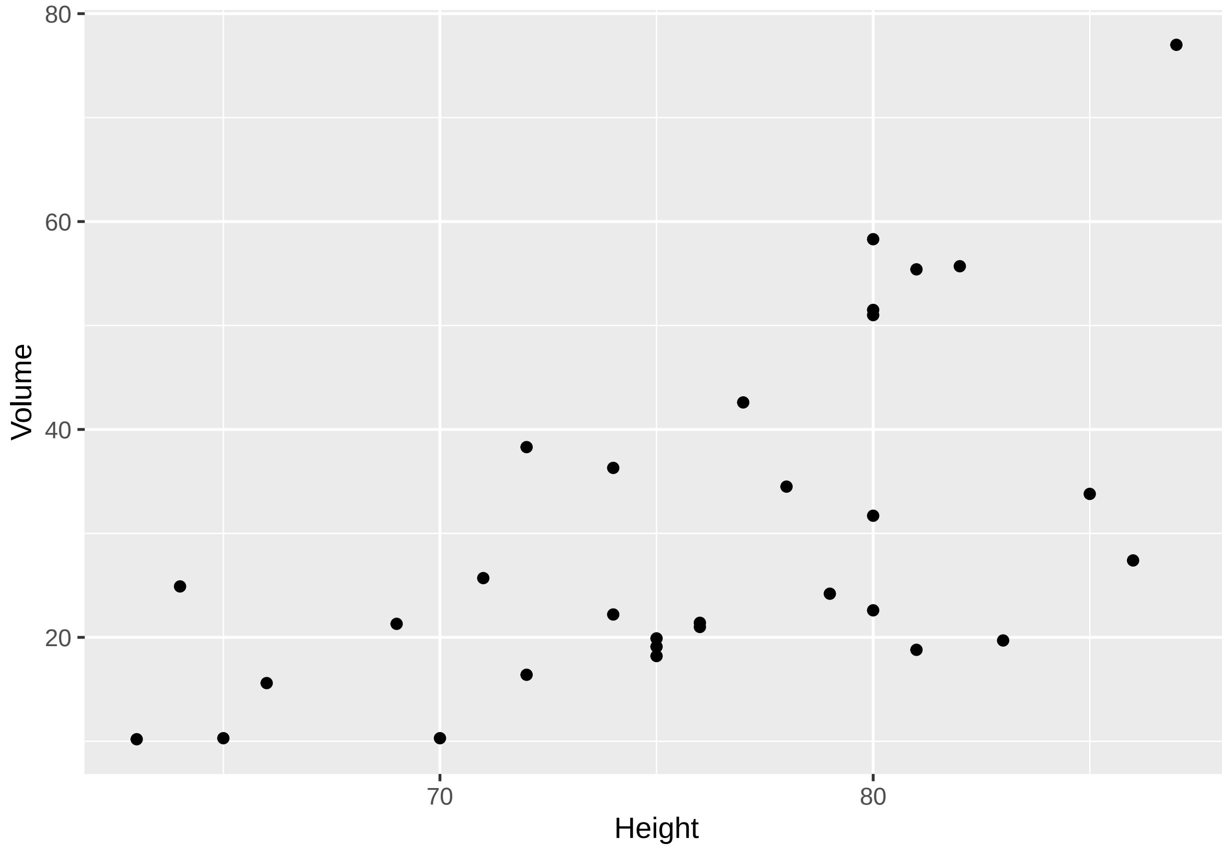
b) Using the ‘summary‘ command, get the y-intercept and slope of the regression line.

```r
summary(model) # summary of 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
```

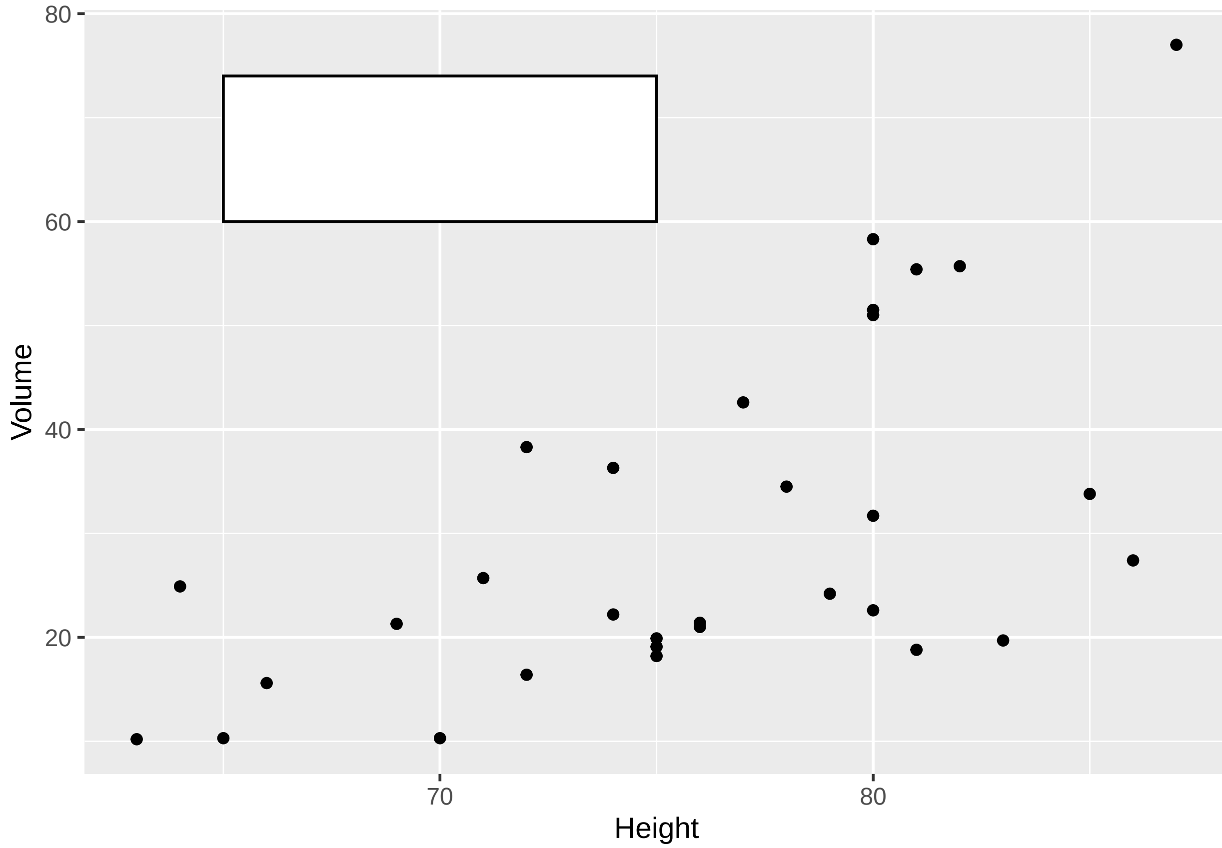
c) Using ‘ggplot2‘, create a scatter plot of Volume vs Height.

```r
ggplot(data=trees, aes(y=Volume, x=Height)) + geom_point() # vol vs height
```



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

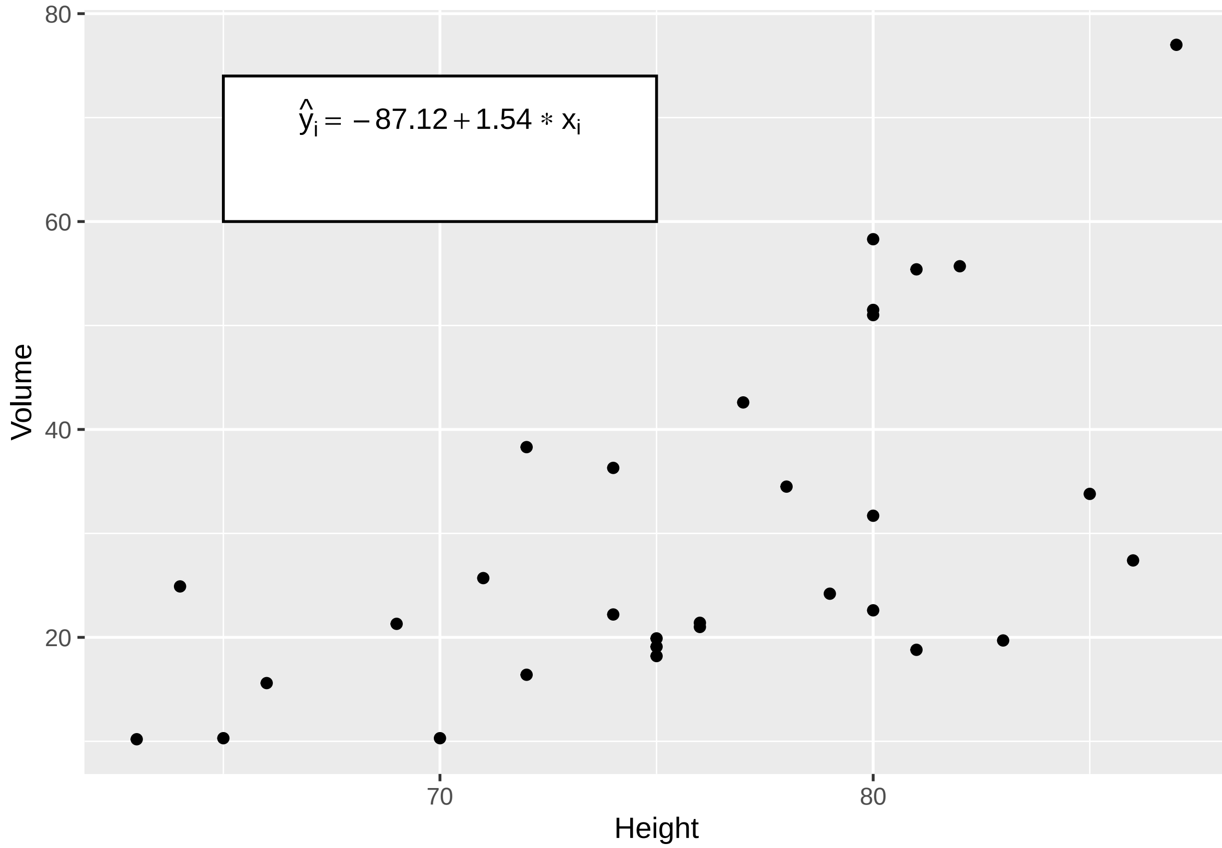
```
ggplot(data=trees, aes(y=Volume, x=Height)) + geom_point() +
  annotate('rect', xmin=65, xmax=75, ymin=60, ymax=74, # add rectangle
          fill='white', color='black')
```



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(data=trees, aes(y=Volume, x=Height)) + geom_point() +
  annotate('rect', xmin=65, xmax=75, ymin=60, ymax=74,
           fill='white', color='black') +
  annotate('text',x=70, y=70, # equation of line
           label = latex2exp::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'
```

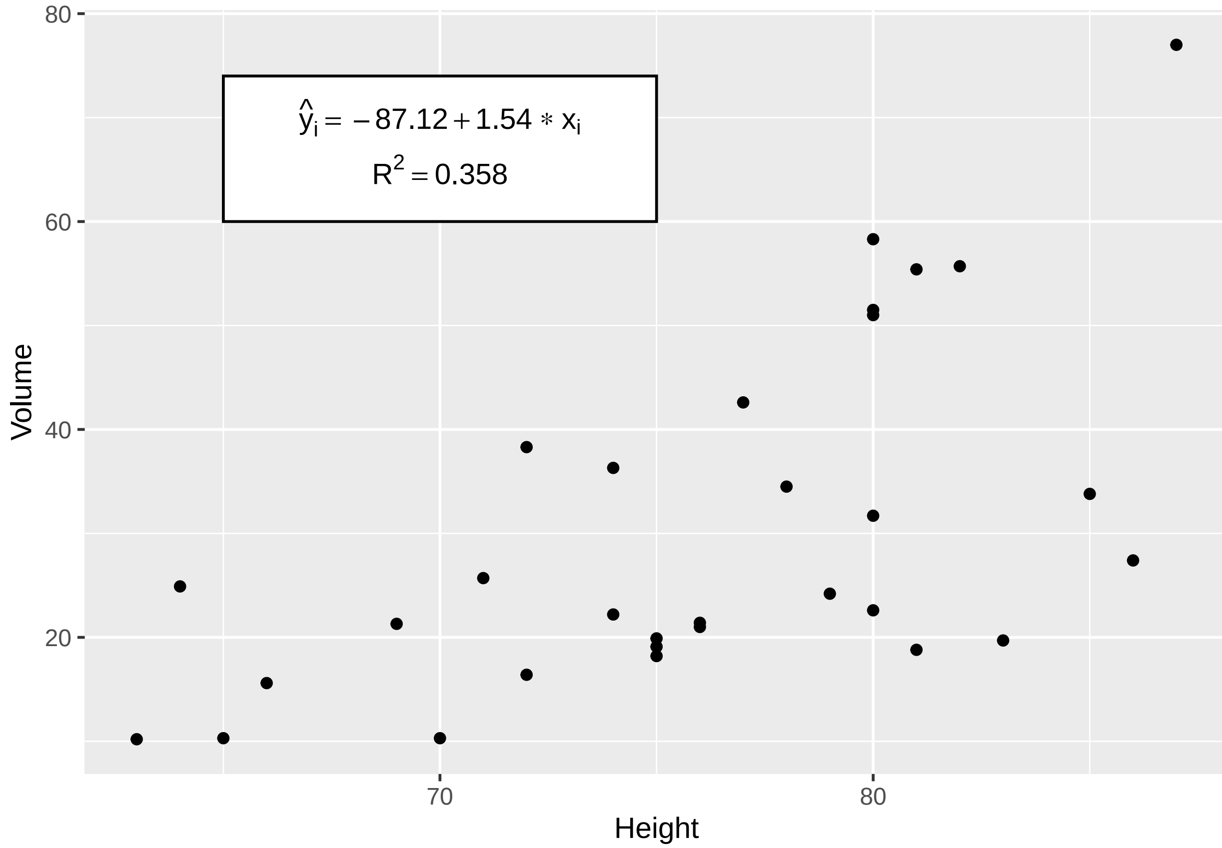


f) Add annotation to add $R^2 = 0.358$

```
ggplot(data=trees, aes(y=Volume, x=Height)) + geom_point() +
  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('$R^2 = 0.358$')) # r^2 label
```

```
## 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'
```

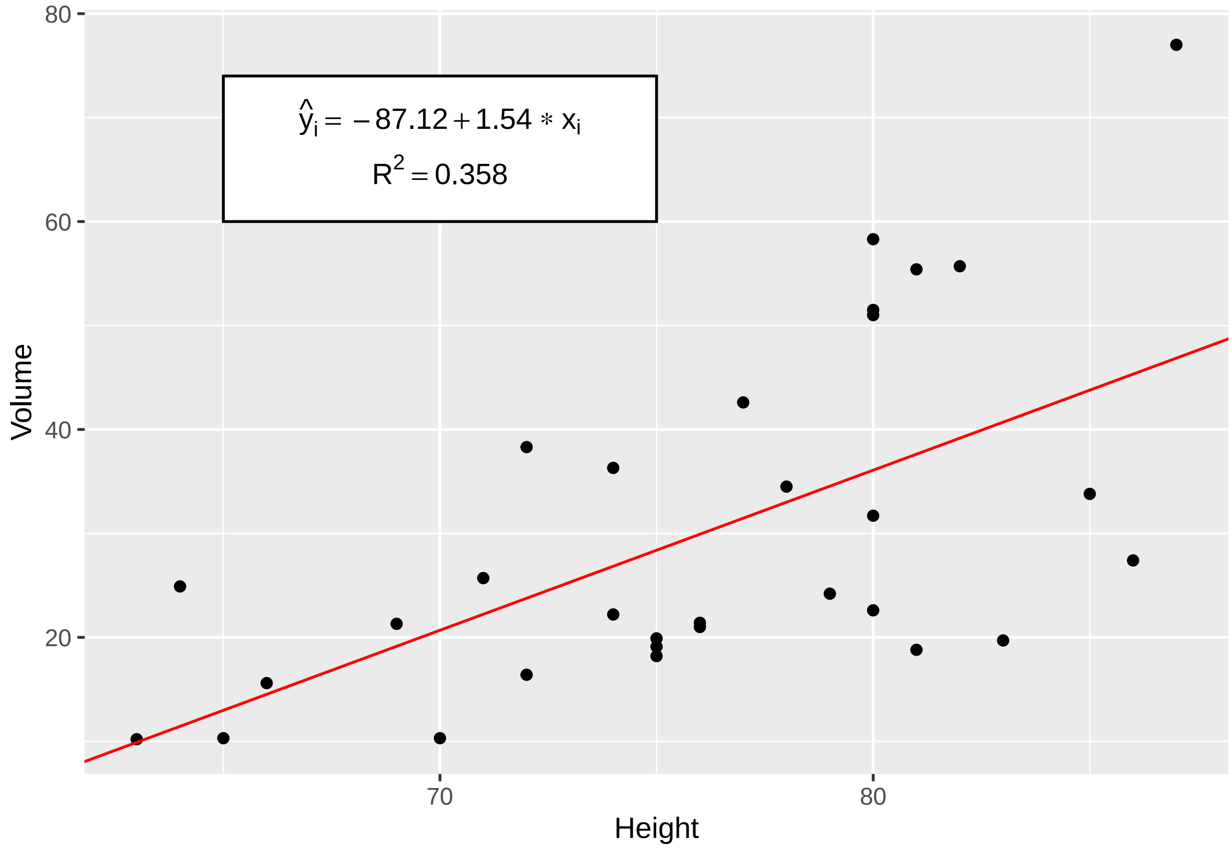


g) Add the regression line in red. The most convenient layer function to uses is `geom_abline()`. It appears that the `annotate` doesn't work with `geom_abline()` so you'll have to call it directly.

```
ggplot(data=trees, aes(y=Volume, x=Height)) + geom_point() +
  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('$R^2 = 0.358$')) + # regression line in red
  ggplot2::geom_abline(intercept = -87.12, slope = 1.54, color='red')
```

```
## 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'
```


$\hat{y}_i = -87.12 + 1.54 * x_i$
$R^2 = 0.358$
80
60
40
20
Volume
70
80
Height