pdf_HW_STA_445_Assignment 7

Nicholas Larson

4/9/2024

Load your packages here:

Problem 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 The information is relatively out of date, but will be fun to graph.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`

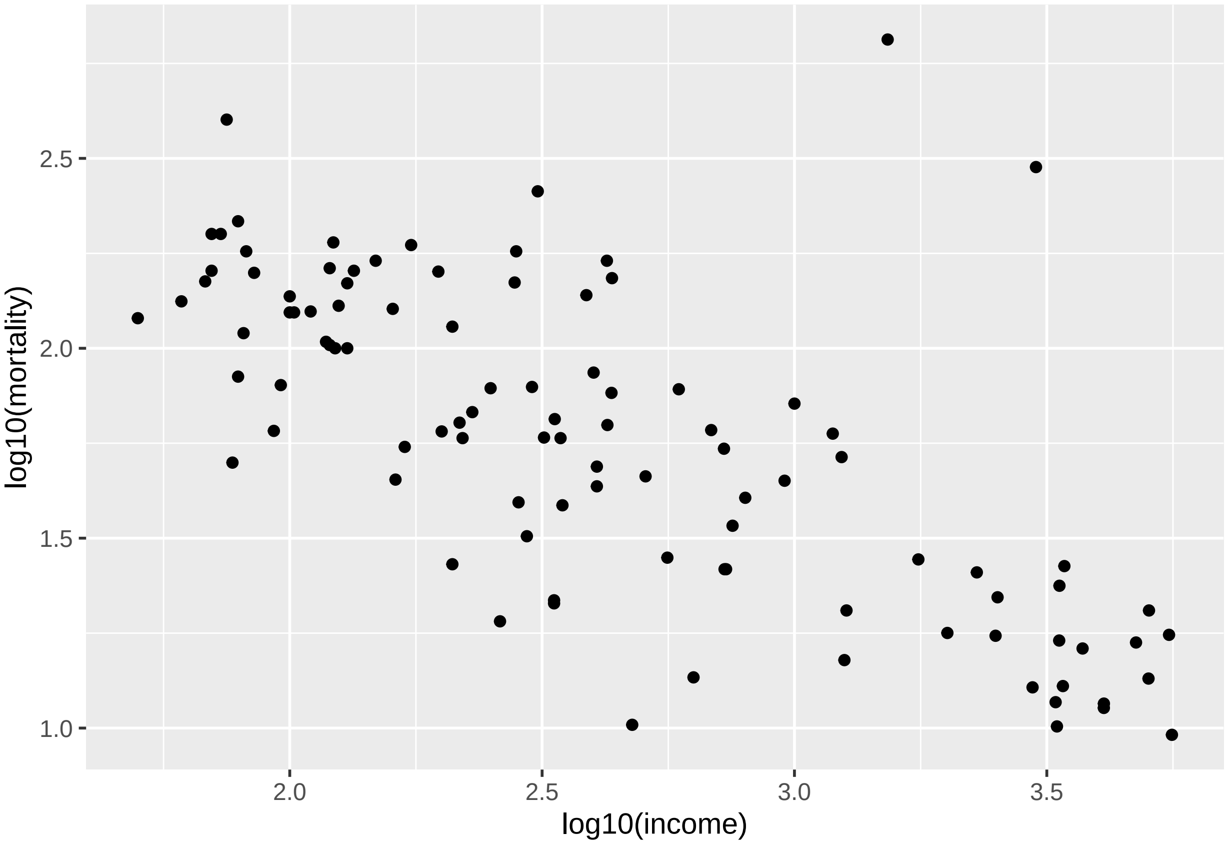
```
data('infmort', package='faraway')

infmort <- infmort %>% mutate(
  countrynames = rownames(infmort)
)
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
##                  countrynames
## Australia           Australia
## Austria             Austria
## Belgium             Belgium
## Canada              Canada
## Denmark             Denmark
## Finland             Finland
```

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

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

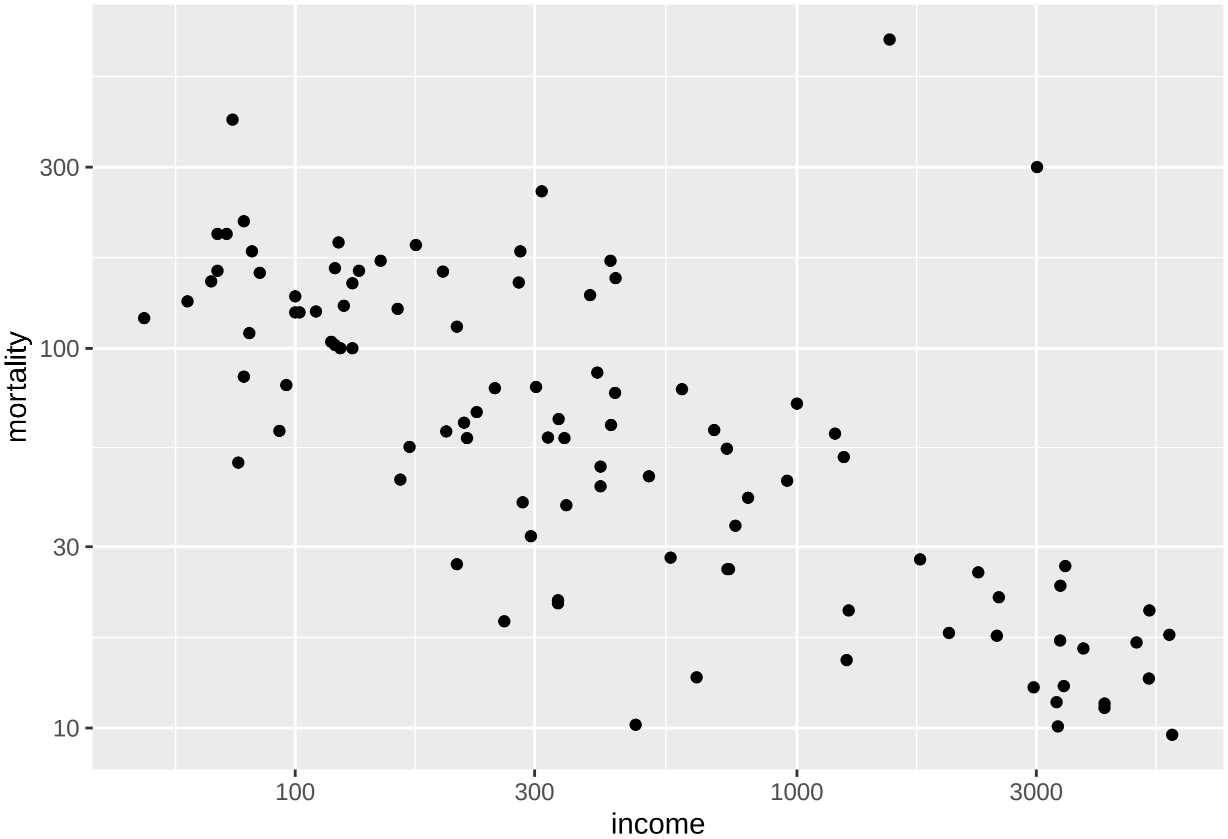
```
## Warning: Removed 4 rows containing missing values (`geom_point()`).
```



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(data=infmort, aes(x=income, y=mortality)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10()
```

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

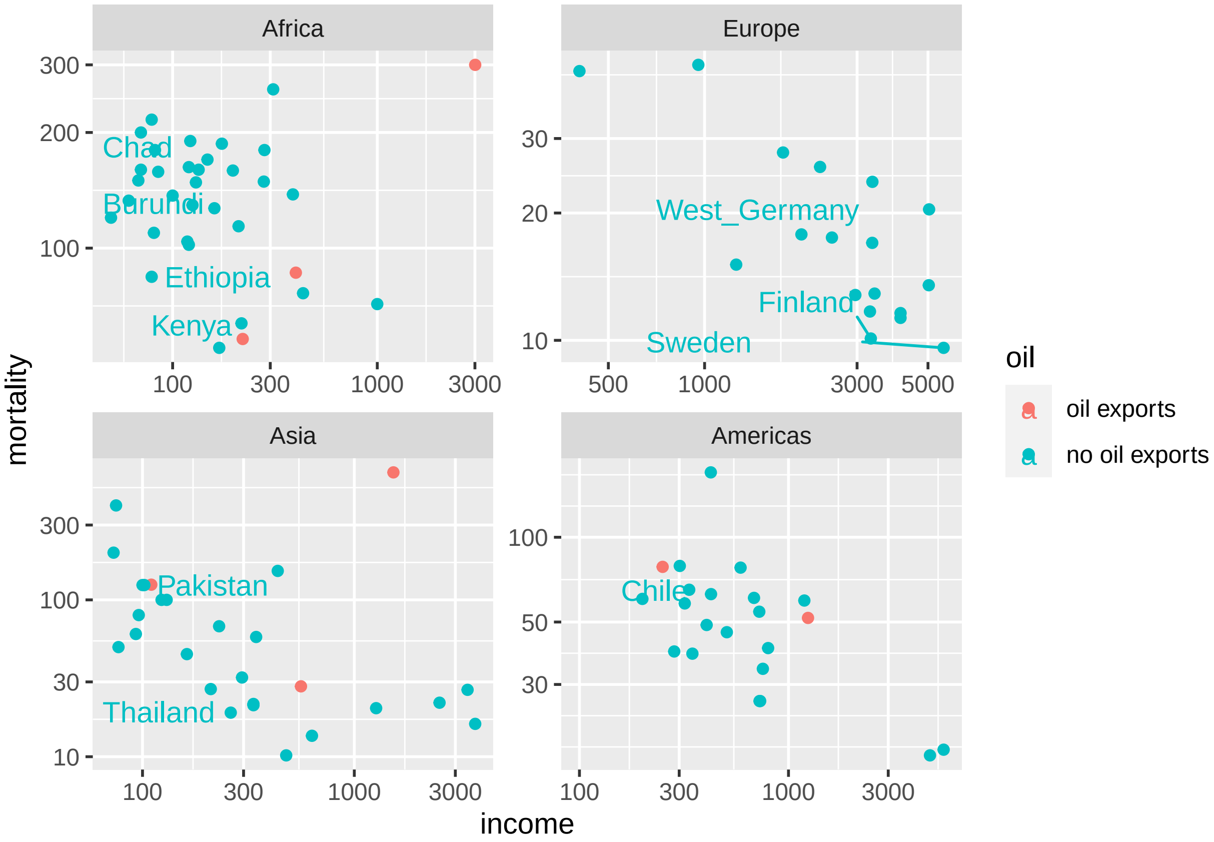


The second graph using scale_x/y_log10 looks better, both ways scale the graph down to comfortable proportions but scale_x/y_log10 retains the original data numbers and labels.

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.

```
set.seed(2)
countries <- slice_sample(infmort, n=10)
ggplot(data=infmort, aes(x=income, y=mortality, color=oil)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10()+
  geom_text_repel(data=countries, aes(x=income, y=mortality, label=countrynames)) +
  facet_wrap('region', nrow = 2, ncol = 2, scales = "free", shrink = TRUE)
```

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



Problem 2

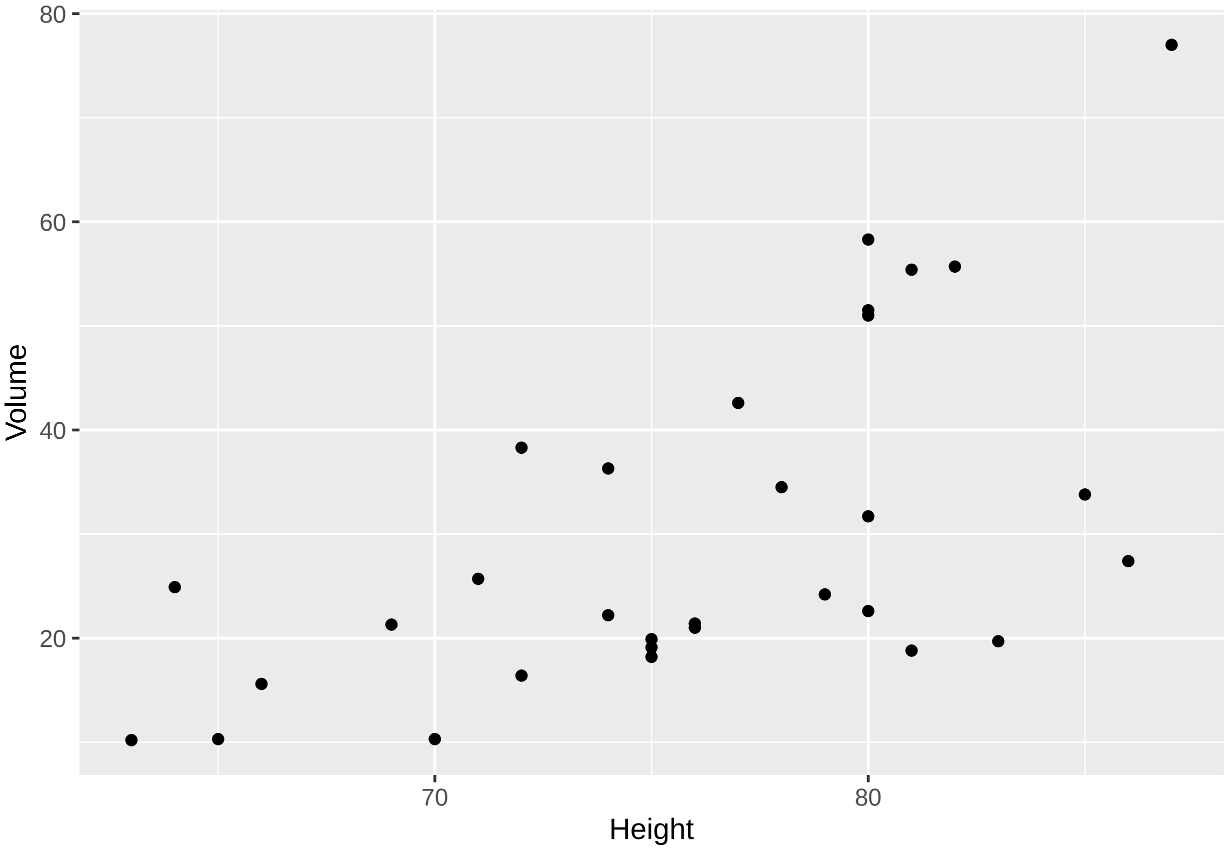
Using the `datasets::trees` data, complete the following:

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

```
data(trees)
head(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)
fitteddata <- trees %>% mutate(fit = fitted(model))
P1 <- ggplot(fitteddata, aes(x=Height)) +
  geom_point(aes(y=Volume))
P1
```



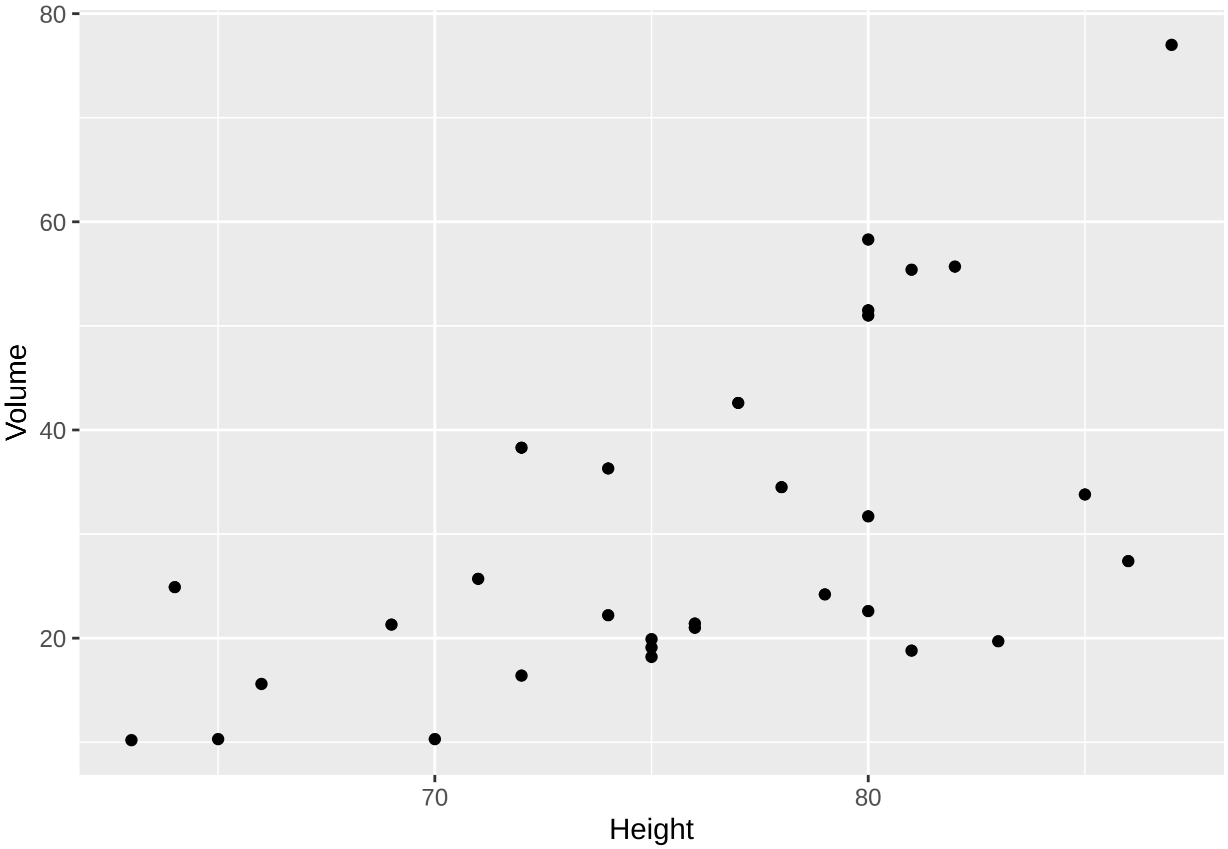
b. Using the str(your model's name) command, to get a list of all the information stored in the linear model object. Use $ to extract the slope and intercept of the regression line (the coefficients).

```
model$coefficients
```

```
## (Intercept)      Height
##   -87.12361     1.54335
```

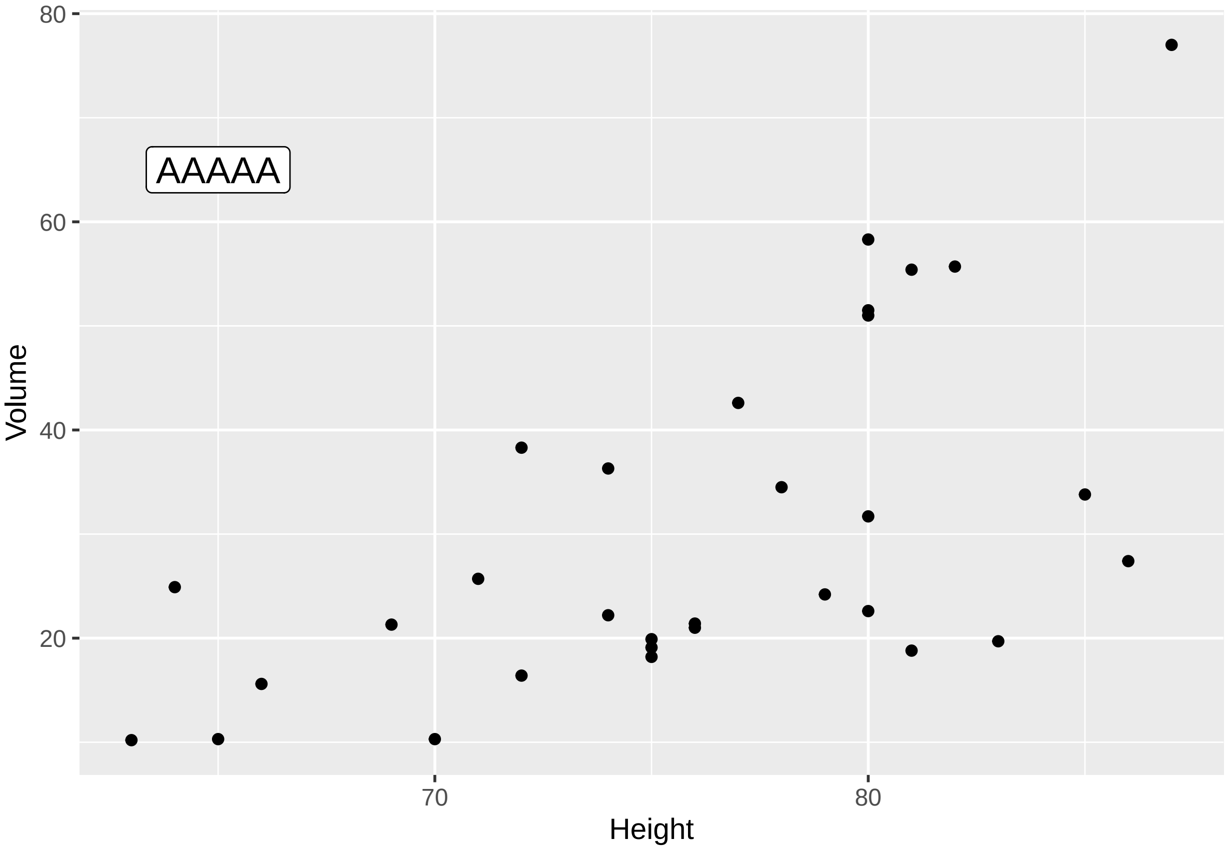
c. Using ggplot2, create a scatter plot of Volume vs Height.

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



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

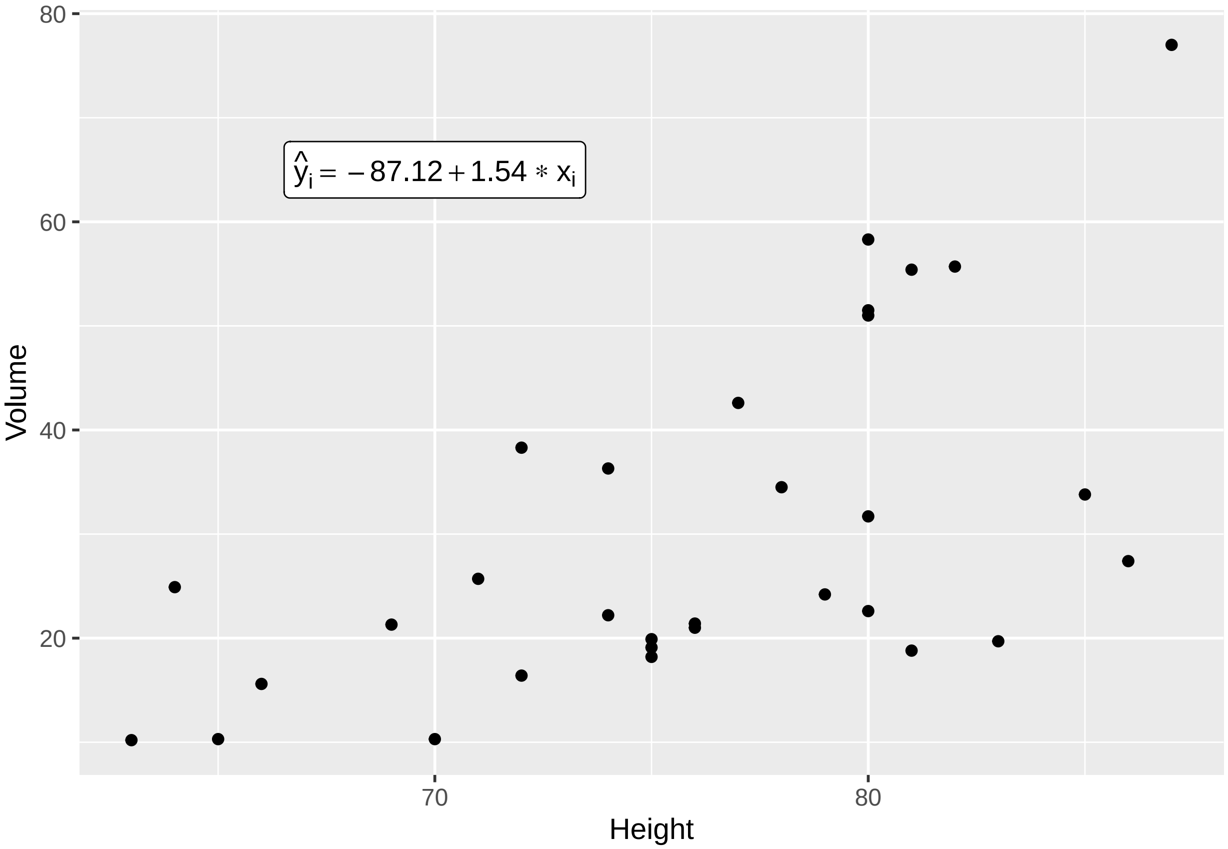
```
ggplot(trees, aes(y=Volume, x=Height))+
geom_point()+
annotate('label', x=65, y=65, label='AAAAA', size=5)
```



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(y=Volume, x=Height))+
geom_point()+
annotate('label', x=70, y=65, label=latex2exp::TeX("$\\hat{y}_i = -87.12 + 1.54 * x_i$"), size=4)
```

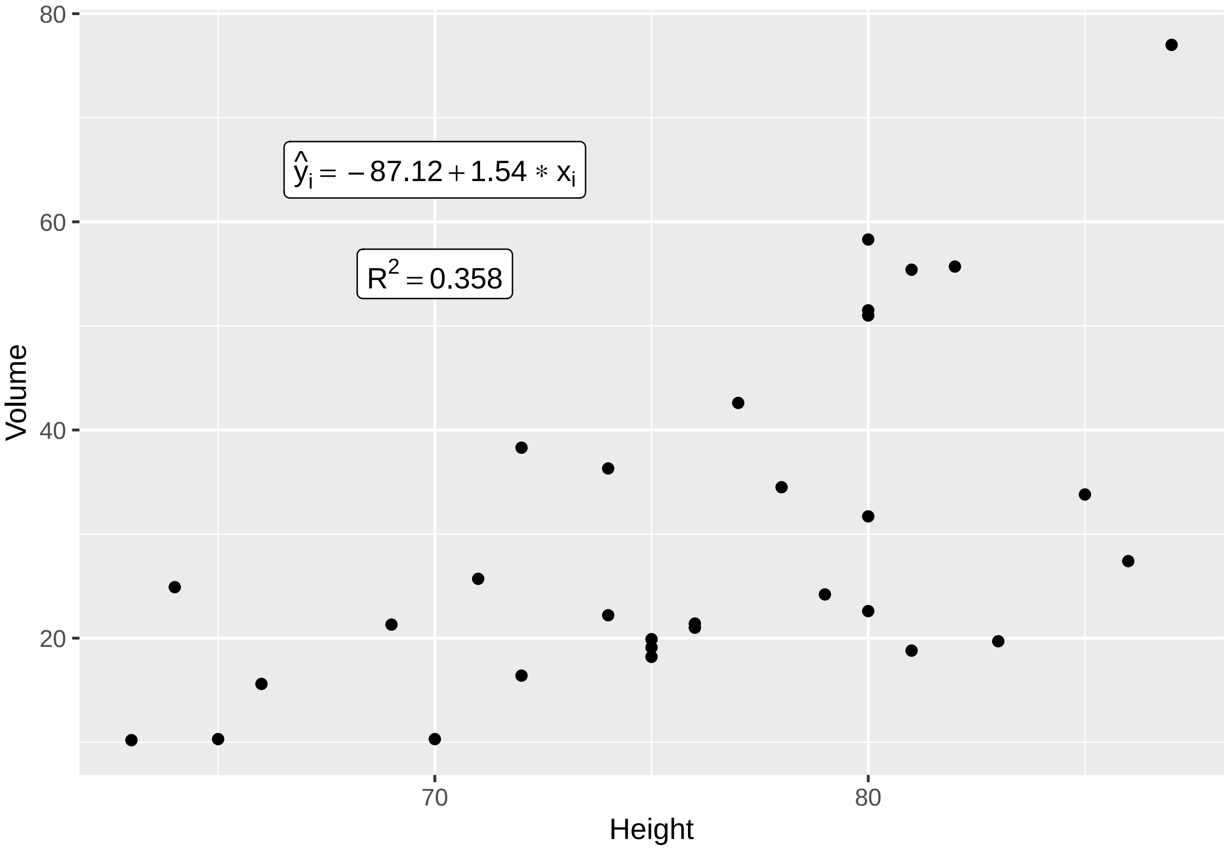
```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'
```



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

```
ggplot(trees, aes(y=Volume, x=Height))+
geom_point()+
annotate('label', x=70, y=65,
         label=latex2exp::TeX("$\\hat{y}_i = -87.12 + 1.54 * x_i$"), size=4)+
annotate('label', x=70, y=55, label=latex2exp::TeX("$R^2 = 0.358$"),
         size=4)
```

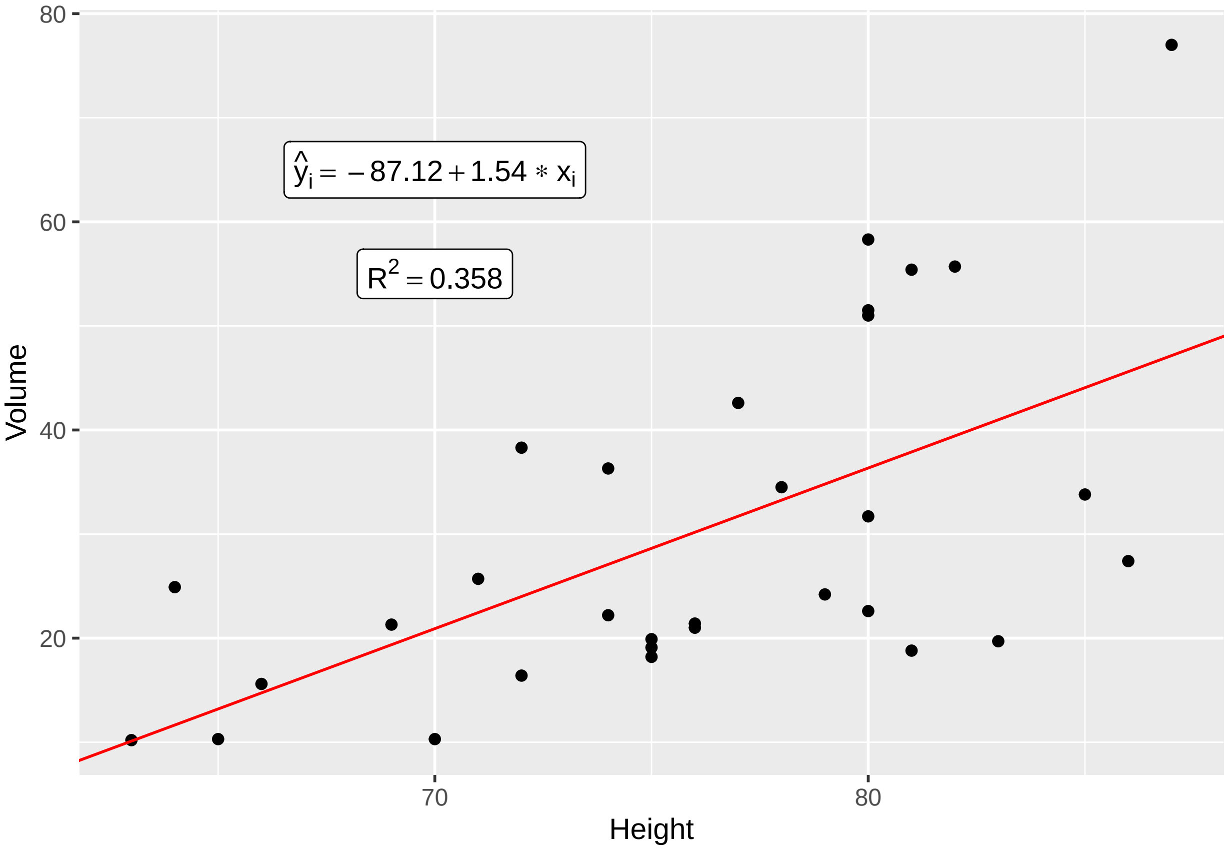
```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'

## 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()`.

```
ggplot(trees, aes(y=Volume, x=Height))+
geom_point()+
geom_abline(intercept=model$coefficients[1], slope=model$coefficients[2], color='red')+
annotate('label', x=70, y=65,
         label=latex2exp::TeX("$\\hat{y}_i = -87.12 + 1.54 * x_i$"),
         size=4)+
annotate('label', x=70, y=55, label=latex2exp::TeX("$R^2 = 0.358$"),
         size=4)
```

```
## 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'
```



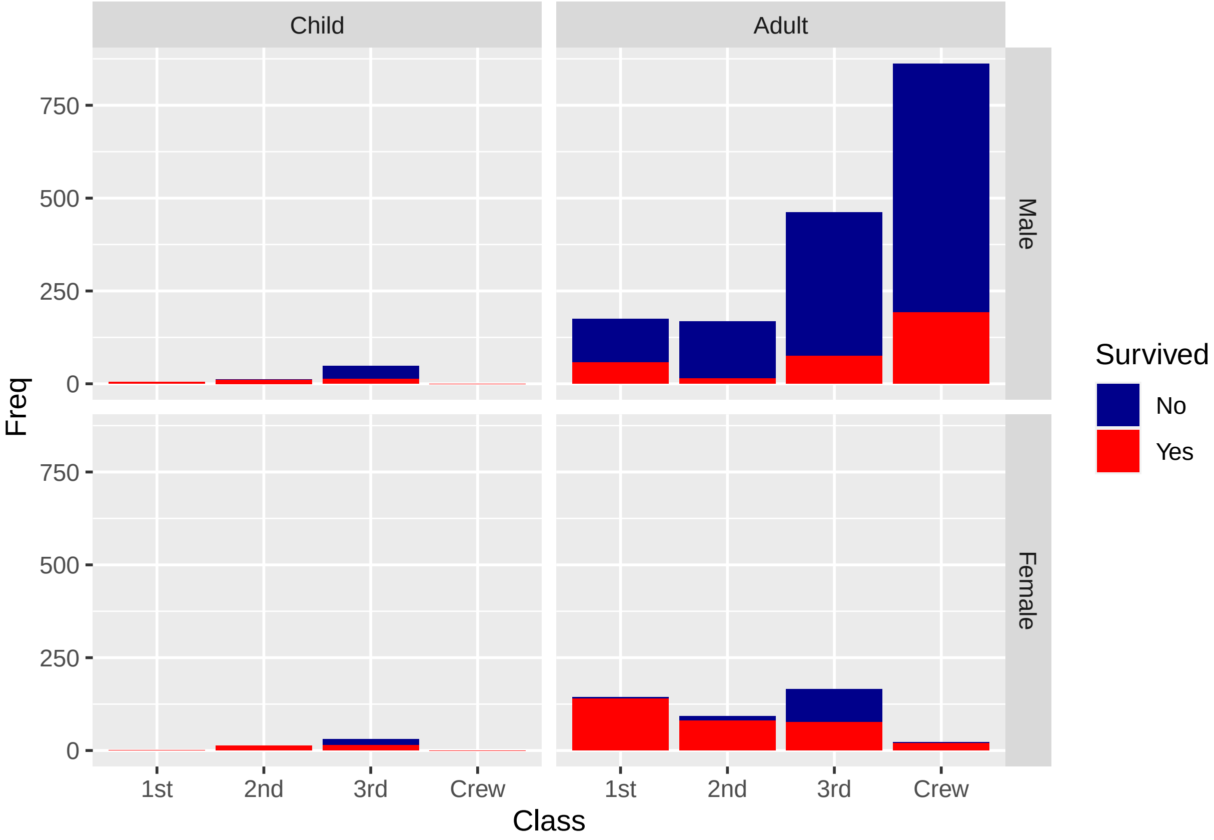
Problem 3

In `datasets::Titanic` table summarizes the survival of passengers aboard the ocean liner *Titanic*. It includes information about passenger class, sex, and age (adult or child). Create a bar graph showing the number of individuals that survived based on the passenger `Class`, `Sex`, and `Age` variable information. You'll need to use faceting and/or color to get all four variables on the same graph. Make sure that differences in survival among different classes of children are perceivable. *Unfortunately, the data is stored as a `table`and to expand it to a data frame, the following code can be used.*

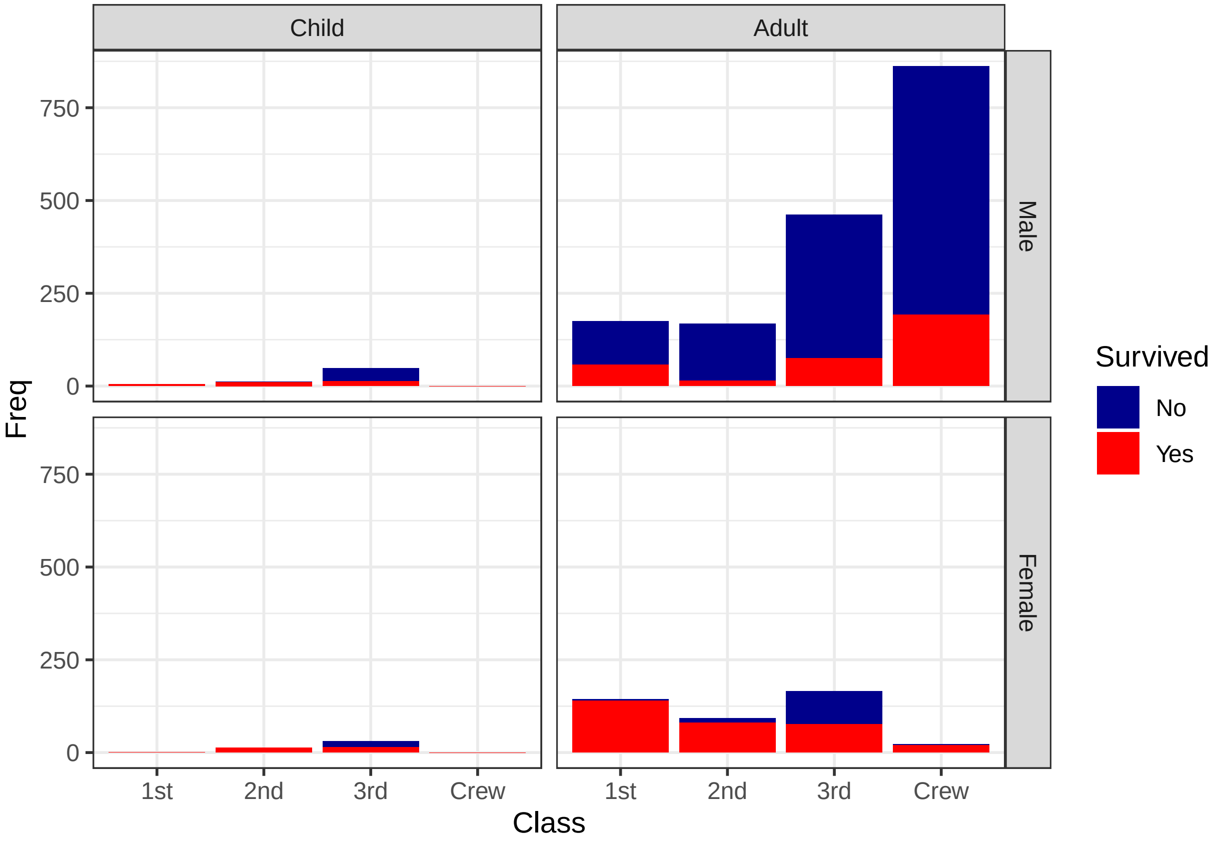
```
Titanic <- Titanic %>% as.data.frame()
head(Titanic)
```

a. Make this graph using the default theme. *If you use color to denote survivorship, modify the color scheme so that a cold color denotes death.*
b. Make this graph using the `theme_bw()` theme.
c. Make this graph using the `cowplot::theme_minimal_hgrid()` theme.
d. Why would it be beneficial to drop the vertical grid lines?

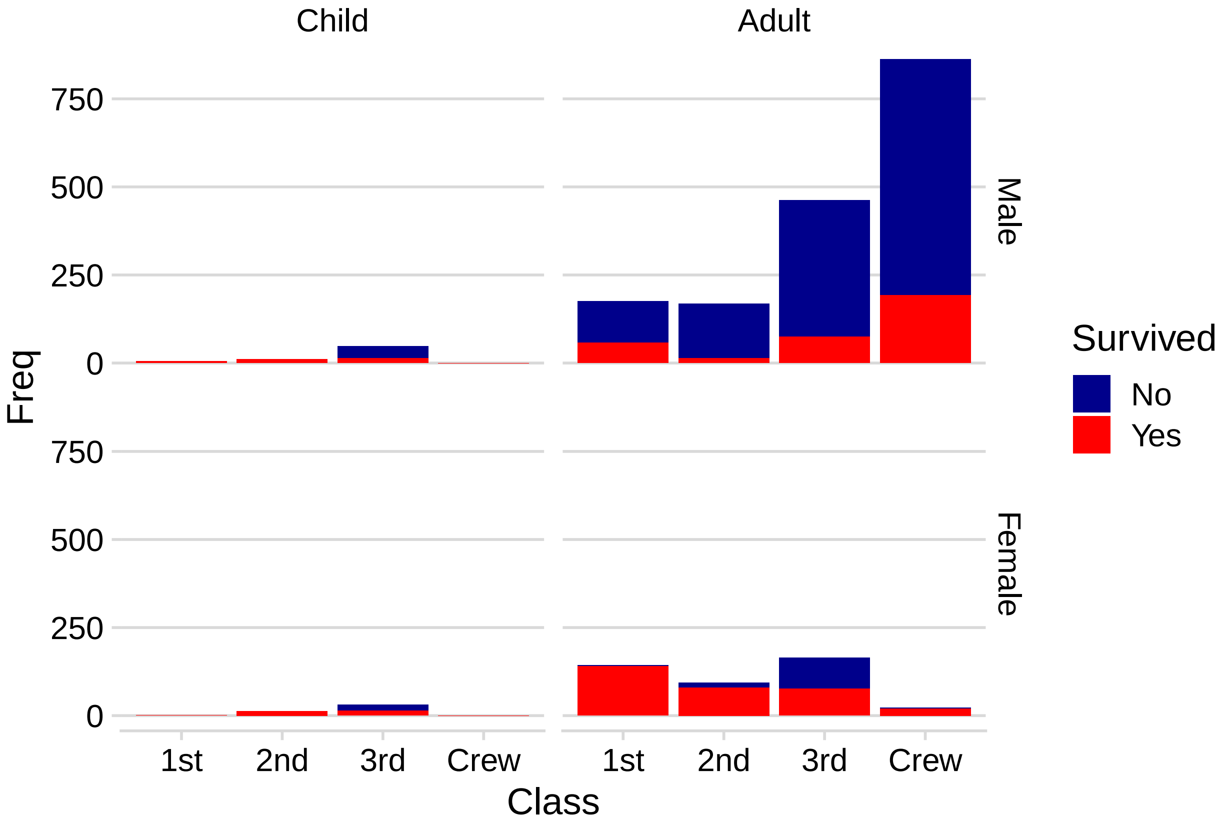
```
Titanic <- Titanic %>% as.data.frame()
T <- ggplot(Titanic) +
  geom_bar(aes(x=Class, y=Freq, fill=Survived), stat = 'identity')+
  facet_grid(Sex~Age)+
  scale_fill_manual(values= c('darkblue', 'red'))
T
```


Child
Adult
Male
Female
Freq
750
500
250
0
750
500
250
0
1st
2nd
3rd
Crew
1st
2nd
3rd
Crew
Class
Survived
No
Yes

```
T + theme_bw()
```



```
T+cowplot::theme_minimal_hgrid()
```



It

is beneficial to drop the vertical lines because we are distinguishing between factored variables not numerical. The lesser lines on the graph makes it easier to read.