Programming Packages HW3

David Roycroft

2023-10-23

Problem 1

Problem 2

Problem 3

- Part a

```
#Read in data
url <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/ThicknessGauge.dat"
sensory_raw<-read.table(url, header=F, skip=0, fill=T, stringsAsFactors = F)

#Data cleaning
sensory_raw <- sensory_raw[3:12,]
for(i in 1:10){
  sensory_raw$mean1[i] <- (sensory_raw[i,2]+sensory_raw[i,3])/2
  sensory_raw$mean2[i] <- (sensory_raw[i,4]+sensory_raw[i,5])/2
  sensory_raw$mean3[i] <- (sensory_raw[i,6]+sensory_raw[i,7])/2
}

sensory_raw <- sensory_raw %>%
  select(V1, mean1, mean2, mean3) %>%
  rename("Part" = "V1", "Operator 1" = "mean1", "Operator 2" = "mean2", "Operator 3" = "mean3")
```

In this dataset, the data for each operator in found in two columns next to each other (i.e. the first operator's measurements are found in the first two columns, second operator's measurements in the 3rd and 4th columns, etc.). To fix this dataset, we need to combine those two columns into a single column for each operator. Since there are two values to ensure measurement accuracy, I will take the average of those two measurements for each operator to get a singular representative value for every part. So I first calculated the mean values and then paired down the dataset to only the necessary variables. This gives a tidy dataset of 10 observations corresponding to the mean value of each operator's measurements for each of the 10 sampled parts.

```
kable(head(sensory_raw),caption="Part Thickness Data")
```

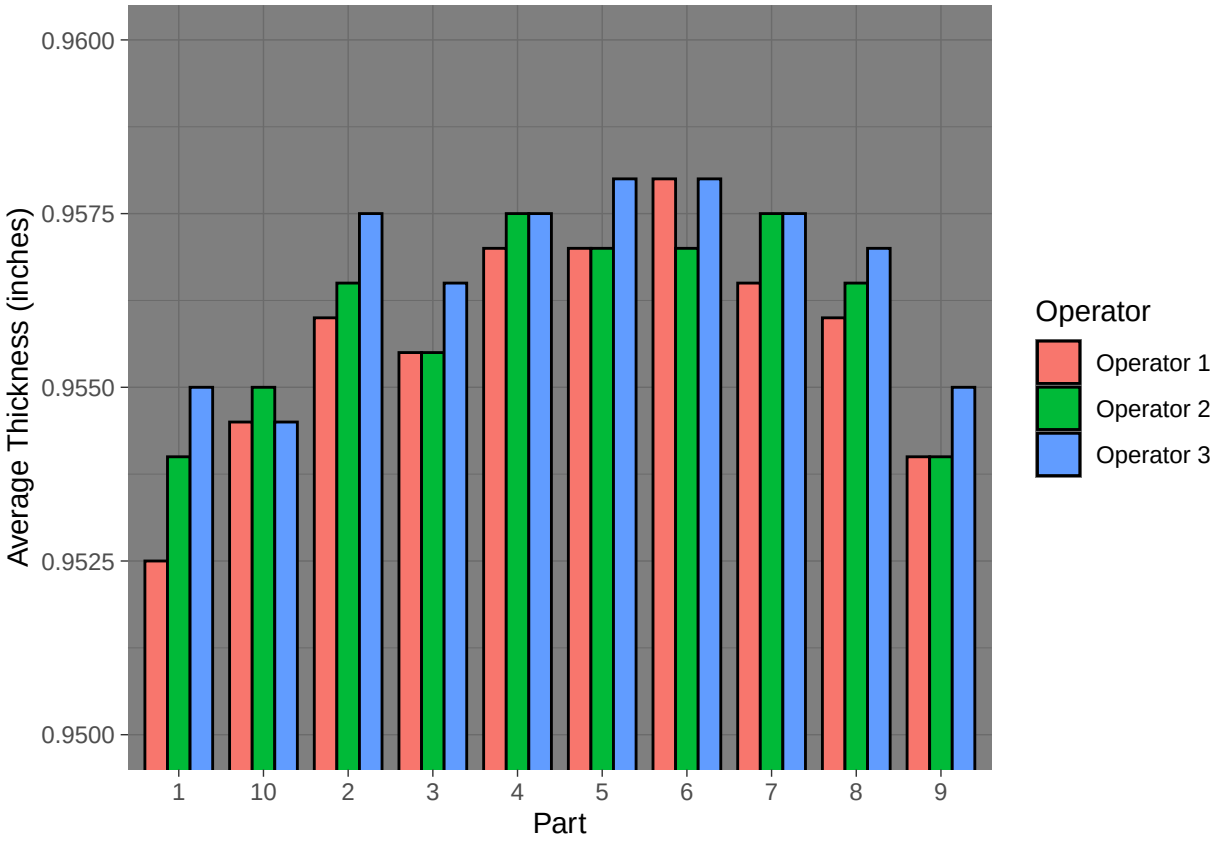
Since we have mean values, we will use a bar chart to show the differences between measurements across the different operators and parts measured.

```
sensory_plot <- gather(sensory_raw, key="Operator", value="value", 2:4)
ggplot(sensory_plot, aes(x=Part, y=value, fill = Operator)) +
  geom_bar(position = "dodge", stat = "identity", na.rm = TRUE, width = 0.8, color = "black") +
```

Table 1: Part Thickness Data

	Part	Operator 1	Operator 2	Operator 3
3	1	0.9525	0.9540	0.9550
4	2	0.9560	0.9565	0.9575
5	3	0.9555	0.9555	0.9565
6	4	0.9570	0.9575	0.9575
7	5	0.9570	0.9570	0.9580
8	6	0.9580	0.9570	0.9580

```
  labs(y="Average Thickness (inches)") +
  scale_y_continuous(limits = c(0.95,0.96), oob = rescale_none) +
  theme_dark()
```



- Part b

```
url <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
weight <- read.table(url, header = F, skip=0, fill=T, stringsAsFactors = F)

weight <- weight[-1, 1:6]

weight_final <- data.frame(c(weight$V1, weight$V3, weight$V5), c(weight$V2, weight$V4, weight$V6))
colnames(weight_final) <- c("Brain_Wt", "Body Weight")
```

Table 2: Brain Weight and Body Weight in Animals (kg)

Body Weight	Brain Weight
44.5	0.003385
15.5	0.000480
8.1	0.001350
423.0	0.465000
119.5	0.036330
115.0	0.027660

```
weight_final <- mutate(weight_final, "Brain Weight" = as.numeric(Brain_Wt)/1000) %>%
  select("Body Weight", "Brain Weight")
```

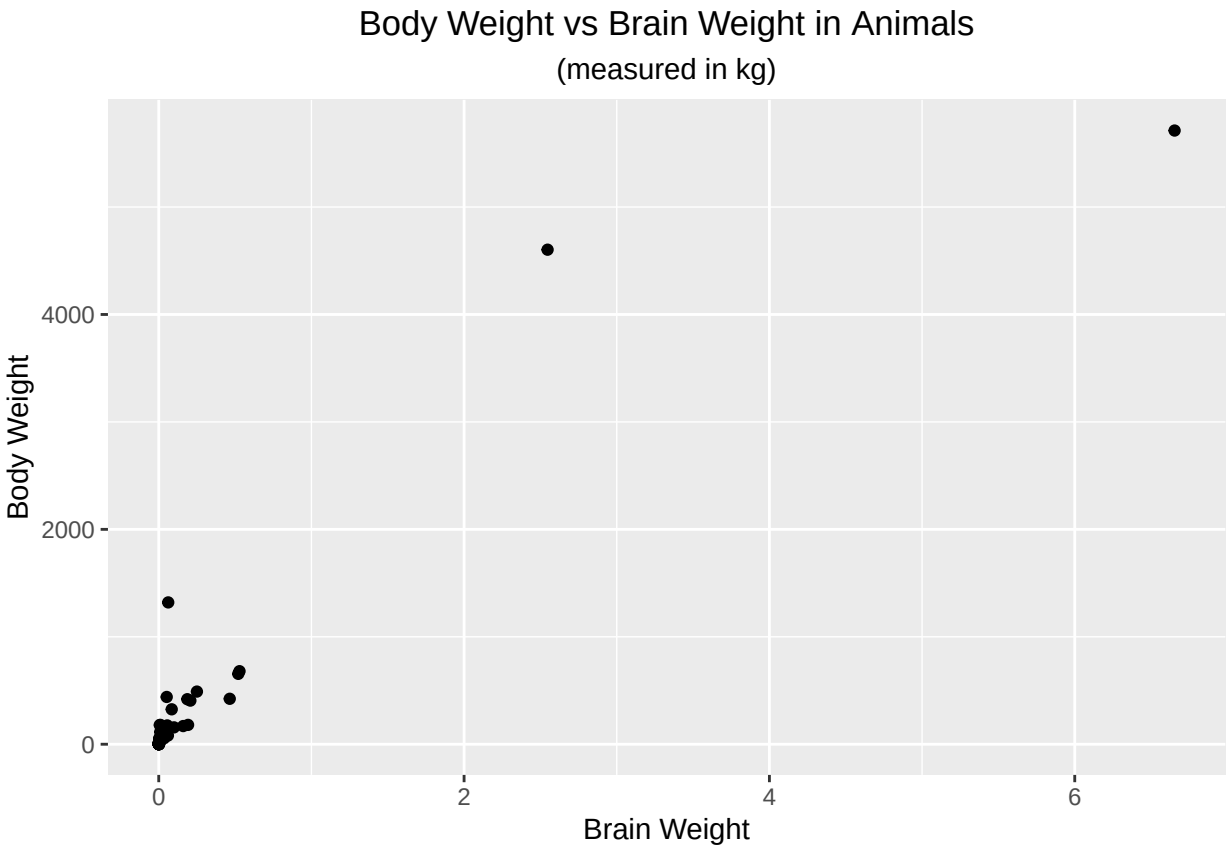
This dataset contains the brain and body weight for different animals. However, the brain and body weights were measured and recorded using different units. These records were also spread across 6 different columns rather than 2. So these columns must be combined and the the units changed to a uniform measurement. Once that is done, we get a resulting dataset which includes the body and brain weights of 63 animals all measured in kilograms.

```
kable(head(weight_final),caption="Brain Weight and Body Weight in Animals (kg)")
```

This data can be represented through a scatterplot to represent the relationship between brain and body weight. However, as can be seen below, there are two extreme values far away from the main cluster of datapoints. Thus, from a zoomed-out perspective, it is hard to determine if the data has a linear association. However,

```
ggplot(weight_final, aes(x=`Brain Weight`, y = as.numeric(`Body Weight`))) +
  geom_point() +
  labs(y = "Body Weight",
       title = "Body Weight vs Brain Weight in Animals",
       subtitle = "(measured in kg)") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))
```

```
## Warning: Removed 1 rows containing missing values (‘geom_point()‘).
```



- Part c

```
medals <- fread(input = "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat")

medals <- medals[,1:8]
colnames(medals) <- c("Year1", "Long Jump1", "Year2", "Long Jump2", "Year3", "Long Jump3", "Year4","Long

med_final <- data.frame("Year" = c(medals$Year1, medals$Year2, medals$Year3, medals$Year4),
                        "Long Jump Dist" = c(medals$`Long Jump1`, medals$`Long Jump2`, medals$`Long Jump3`, m
```

This data has two variables which are split into 8 columns which alternate between the two variables. Additionally, the column names were split by word and therefore created 4 empty columns to accommodate the expanded names. I removed the 4 empty columns and then stacked the alternating columns to shrink the dataset to only hold two columns.

```
kable(head(med_final),caption="Long Jump Distance by Olympic Gold Medalists")
```

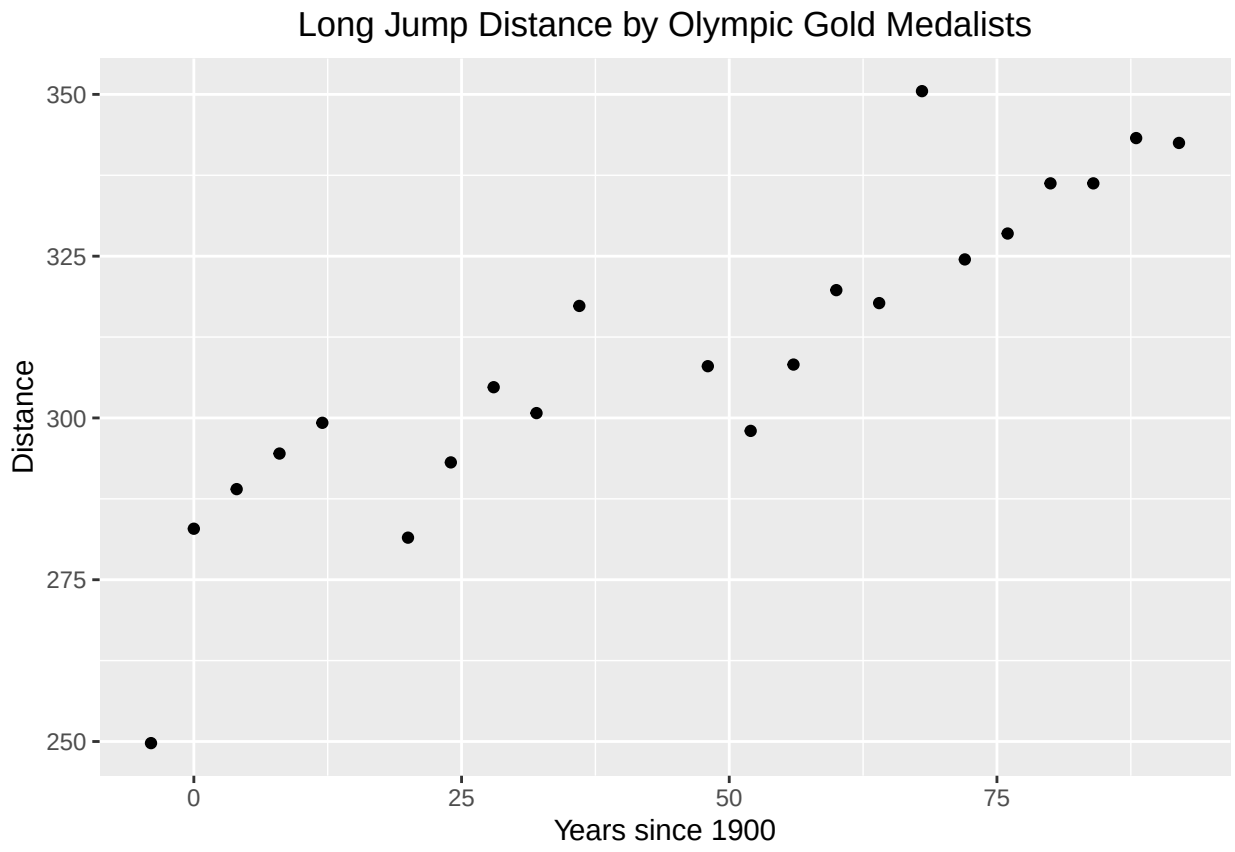
Since we are interested in seeing how the long jump distance has changed over time, I used a scatterplot to show the relationship between the two variables. We can see that the distance has increased over tikme and there is a pretty clear linear relationship.

```
ggplot(med_final, aes(x=Year, y=Long.Jump.Dist)) +
  geom_point() +
  labs(x = "Years since 1900",
```

Table 3: Long Jump Distance by Olympic Gold Medalists

Year	Long.Jump.Dist
-4	249.75
0	282.88
4	289.00
8	294.50
12	299.25
20	281.50

```
      y = "Distance",
      title = "Long Jump Distance by Olympic Gold Medalists") +
 theme(plot.title = element_text(hjust = 0.5))
```



- Part d

```
tomato <- fread(input = "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat")
```

```
## Warning in fread(input =
## "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"): Detected
## 3 column names but the data has 4 columns (i.e. invalid file). Added 1 extra
## default column name for the first column which is guessed to be row names or an
## index. Use setnames() afterwards if this guess is not correct, or fix the file
## write command that created the file to create a valid file.
```

Table 4: Tomato Densities by Variety

V1	10000_1	10000_2	10000_3	20000_1	20000_2	20000_3	30000_1	30000_2	30000_3
Ife\#1	16.1	15.3	17.5	16.6	19.2	18.5	20.8	18.0	21.0
PusaEarlyDwarf	8.1	8.6	10.1	12.7	13.7	11.5	14.4	15.4	13.7

```
tomato <- separate(tomato, col = "10000", into = c("10000_1", "10000_2", "10000_3"), sep=",")
```

```
## Warning: Expected 3 pieces. Additional pieces discarded in 1 rows [2].
```

```
tomato <- separate(tomato, col = "20000", into = c("20000_1", "20000_2", "20000_3"), sep=",")
tomato <- separate(tomato, col = "30000", into = c("30000_1", "30000_2", "30000_3"), sep=",")
```

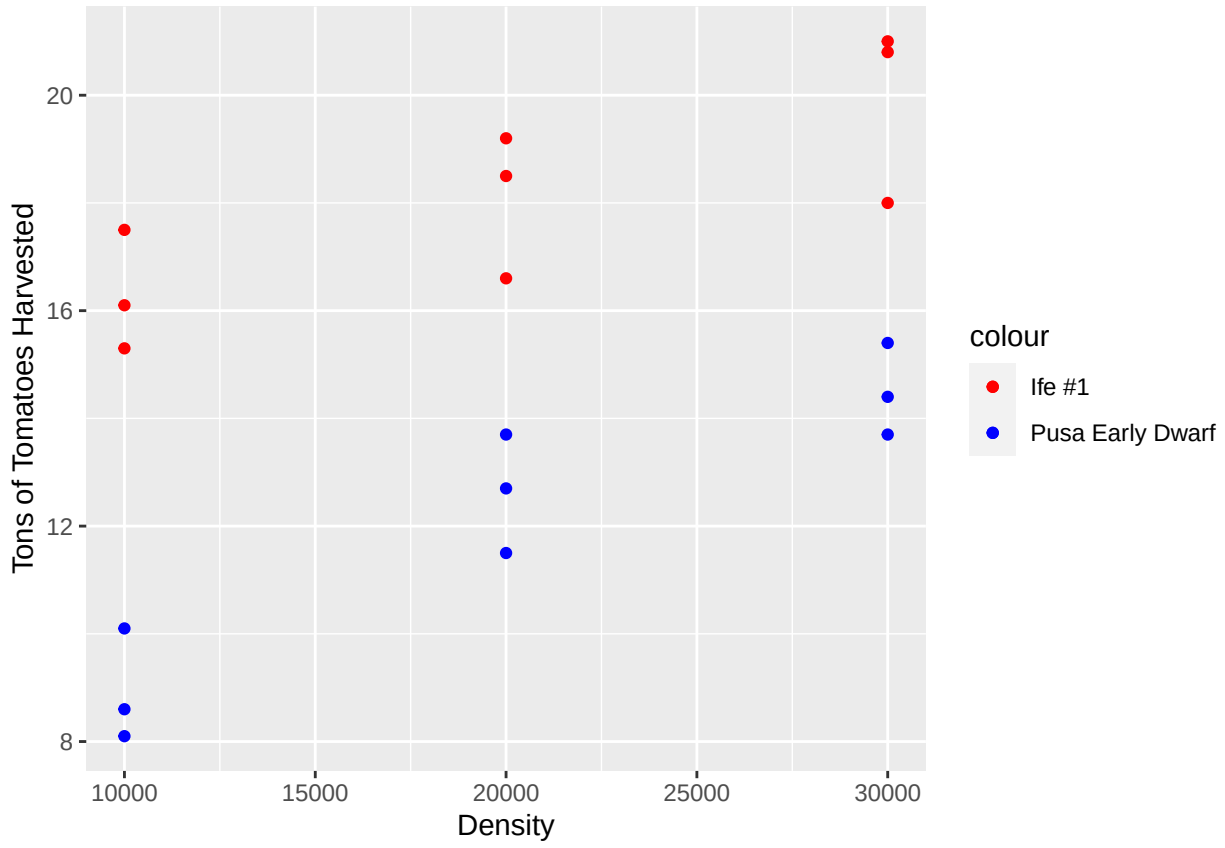
In this dataset, there are three different measurements for each tomato variety at each of the three tomato densities per hectare. I split those three different measurements into their own columns to facilitate easier plotting.

```
kable(head(tomato),caption="Tomato Densities by Variety")
```

I used a scatterplot to graph out the relationship between the amount of tomatoes harvested (in tons) and the density of tomatoes planted. Although there are only three distinct values used in the experiment, we can expect the relationship between the two variables to be similar linearly across all values of tomato density.

```
tomato_plot <- data.frame(t(tomato))
tomato_plot <- tomato_plot[-1,]
tomato_plot <- tomato_plot %>% mutate("Density" = c(10000,10000,10000,20000,20000,20000,30000,30000,300
tomato_plot$X1 <- as.numeric(tomato_plot$X1)
tomato_plot$X2 <- as.numeric(tomato_plot$X2)

ggplot(tomato_plot, aes(x=Density, y=X1)) +
  geom_point(aes(color = "Ife #1")) +
  geom_point(aes(x=Density, y=X2, color = "Pusa Early Dwarf")) +
  labs(y = "Tons of Tomatoes Harvested") +
  scale_colour_manual(breaks = c("Ife #1", "Pusa Early Dwarf"), values = c("red", "blue"))
```



- Part e

```
Larvae <- fread(input = "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LarvaeControl.dat")

Larvae$Age <- c(1,1,1,1,2,2,2,2)
colnames(Larvae) <- c("Block", "T1", "T2", "T3", "T4", "T5", "T1.1", "T2.1", "T3.1", "T4.1", "T5.1","Ag
Larvae_Final <- Larvae %>% group_by(Block) %>% reframe("Count" = c(T1,T2,T3,T4,T5, T1.1, T2.1, T3.1, T4
Larvae_Final$Age <- rep(c(rep(1,5),rep(2,5)),8)
Larvae_Final$Treatment <- rep(1:5,16)
Larvae_Final <- Larvae_Final %>% select("Block", "Age", "Treatment", "Count")
```

This data is larvae counts by block and treatment measured at two different ages. In order to tidy the data, I had to group the 10 larvae counts for the 5 treatment within each of the 8 blocks. I then created the age variable to show which counts and treatments corresponded to a specific age at the time of measurement. I also had to add an additional column to match the counts to their treatment group. This created a dataset with 80 observations of 4 variables.

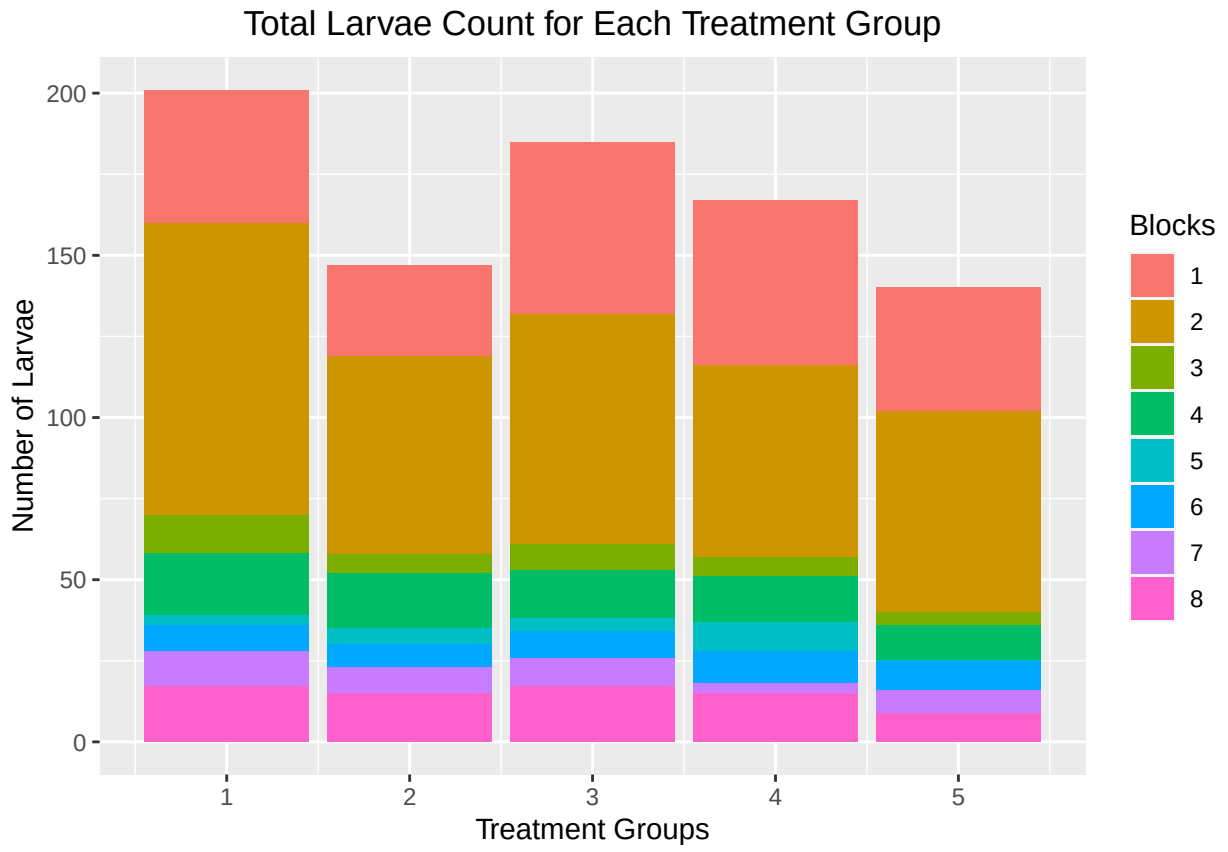
```
kable(head(Larvae_Final,10),caption="Larvae Counts by Age and Block")
```

I used a stacked bar chart to show the larvae counts for each treatment group broken down by block. These treatment counts are summed over both age groups.

Table 5: Larvae Counts by Age and Block

Block	Age	Treatment	Count
1	1	1	13
1	1	2	16
1	1	3	13
1	1	4	20
1	1	5	16
1	2	1	28
1	2	2	12
1	2	3	40
1	2	4	31
1	2	5	22

```
ggplot(Larvae_Final, aes(x=Treatment, y=Count, fill = as.factor(Block))) +
  geom_bar(stat = "identity") +
  labs(x = "Treatment Groups",
       y = "Number of Larvae",
       title = "Total Larvae Count for Each Treatment Group",
       fill = "Blocks") +
  theme(plot.title = element_text(hjust = 0.5))
```



Problem 4