

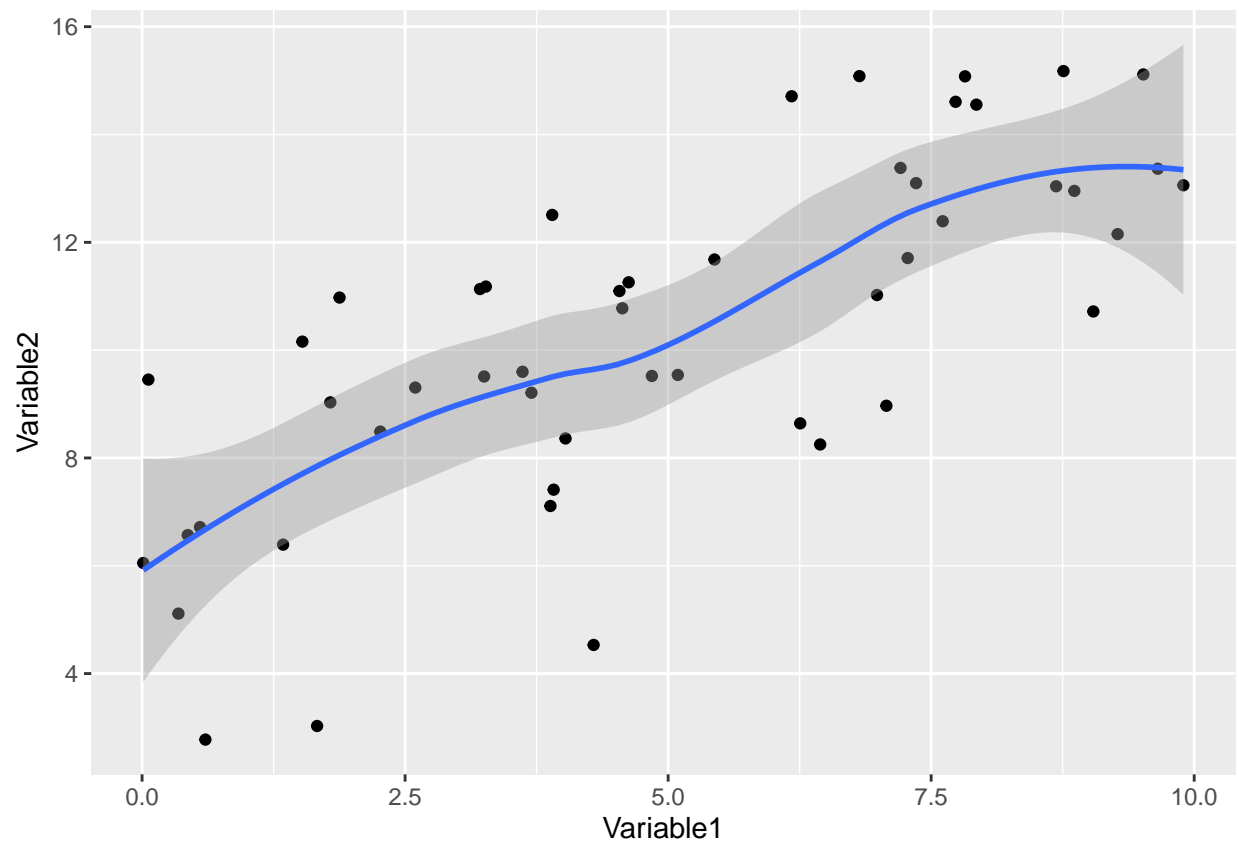
# More ggplot Figures Peer Review

Jonathan Talbot

## Problem 1

Create the figure in the solution for Problem 1, using the data included in the R Markdown file.

```
####Make sure you install any necessary libraries
#install.packages("ggplot2")
library(ggplot2)
####PUT YOUR CODE HERE
ggplot(data=dat1, mapping=aes(x=var1, y=var2))+
  geom_point()+
  geom_smooth()+
  labs(x="Variable1",y="Variable2")
```



## Problem 2

Create the figure in the solution for Problem 2, using the data included in the R Markdown file.

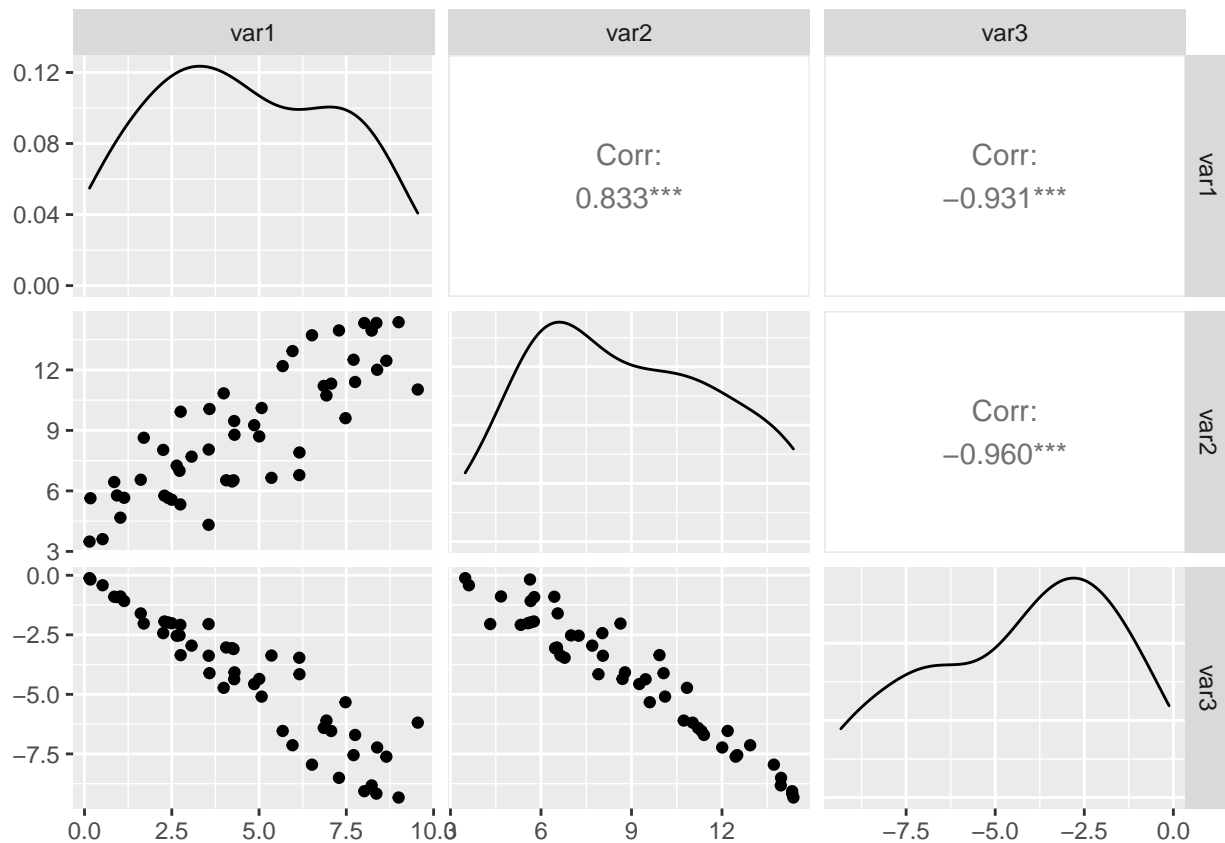
```
####Make sure you load any necessary libraries
```

```
library("GGally")
```

```
####PUT YOUR CODE HERE
```

```
#var2 <- as.factor(var2)
```

```
ggpairs(dat2)#, mapping=aes(color=var2))
```



```
#geom_area(color=factor)
```

### Problem 3

Create the figure in the solution for Problem 3, using the data included in the R Markdown file.

```
####Make sure you load any necessary libraries
#install.packages("ggcorrplot")           # Install ggcorrplot package
library(ggcorrplot)                       # Load ggcorrplot
####PUT YOUR CODE HERE

corr <- round(cor(dat3), 1)
corr
```

```
##      var1 var2 var3
## var1  1.0  0.9 -0.9
## var2  0.9  1.0 -1.0
## var3 -0.9 -1.0  1.0
```

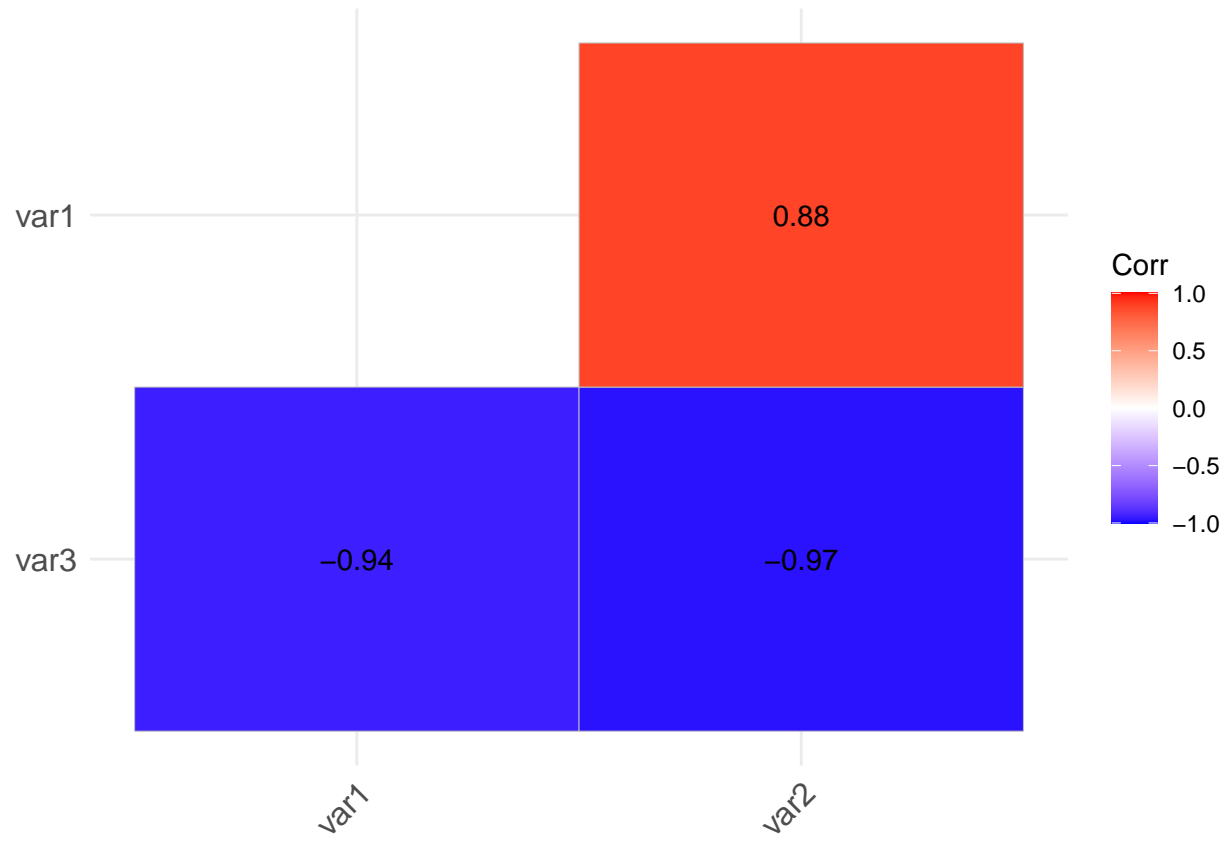
```
# Compute a matrix of correlation p-values
p.mat <- cor_pmat(dat3)
p.mat
```

```
##              var1              var2              var3
## var1 0.000000e+00 5.106500e-17 8.642971e-25
## var2 5.106500e-17 0.000000e+00 1.439856e-29
## var3 8.642971e-25 1.439856e-29 0.000000e+00
```

```
# Visualize the correlation matrix
# -----
# method = "square" or "circle"

ggcorrplot(cor(dat3), hc.order = TRUE, type = "upper", lab = TRUE)+
  coord_flip()
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```



## Problem 4

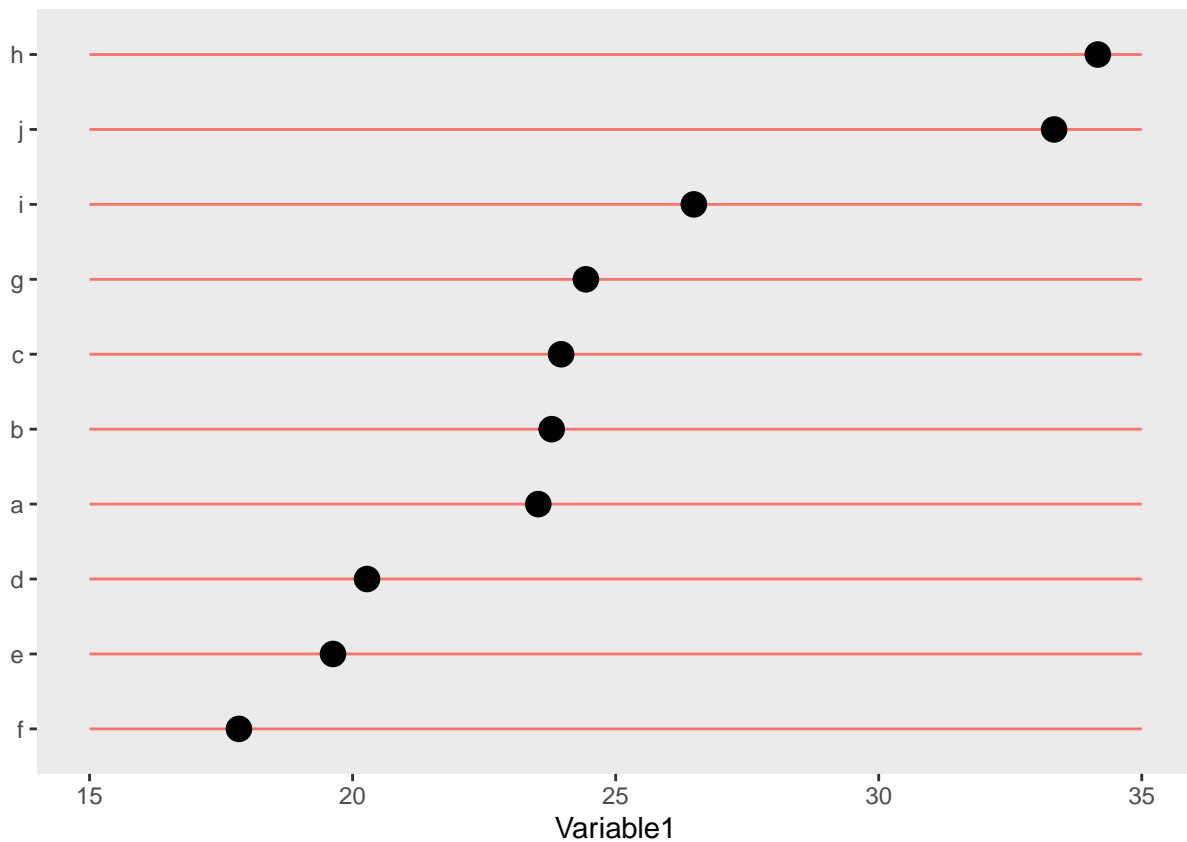
Create the figure in the solution for Problem 4, using the data included in the R Markdown file.

```
####Make sure you load any necessary libraries
#install.packages("ggthemes")
library(ggplot2)
library(ggthemes)
####PUT YOUR CODE HERE

dat4$names <- reorder(names, var1)

ggplot(dat4, mapping=aes(x=var1, y=names))+
  xlab("Variable1")+
  ylab("")+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())+
  theme(legend.position = "none")+
  scale_x_continuous(limits=c(15,35))+
  geom_segment(aes(x=15,xend=35,y=names,yend=names, color="red"))+
  geom_line(linetype="dotted")+
  geom_point(size=4)
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```



## Problem 5

Create the figure in the solution for Problem 5, using the data included in the R Markdown file.

```
####Make sure you load any necessary libraries
```

```
####PUT YOUR CODE HERE
```

```
dat5
```

```
## # A tibble: 10 x 2
##   names  var1
##   <chr> <dbl>
## 1 a      27.8
## 2 b      23.6
## 3 c      33.9
## 4 d      25.9
## 5 e      30.7
## 6 f      27.1
## 7 g      31.1
## 8 h      26.2
## 9 i      23.2
## 10 j     30.5
```

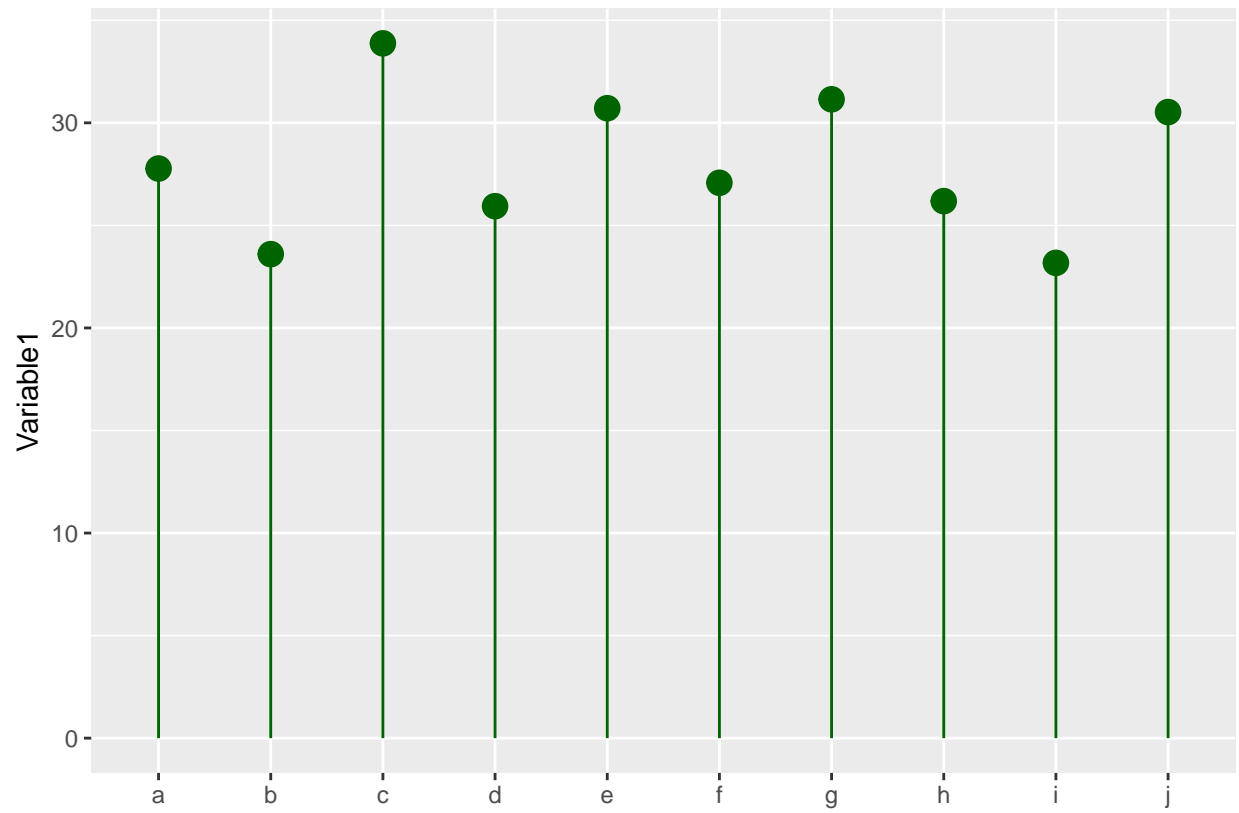
```
library(dplyr)
```

```
# create a vector with letters in the desired order
x <- c("i", "b", "d", "h", "f", "a", "j", "e", "g", "c")
```

```
dat5a <- dat5 %>%
  slice(match(x, names))
```

```
dat5$names <- reorder(names, var1)
```

```
ggplot(dat5a, aes(x=names, y=var1))+ # reorder(names, var1)) +
  geom_segment( mapping=aes(xend=names, yend=0), col="#006400" ) +
  geom_point( size=4, color="#006400" ) +
  xlab("")+
  ylab("Variable1")+
  theme(legend.position = "none")
```



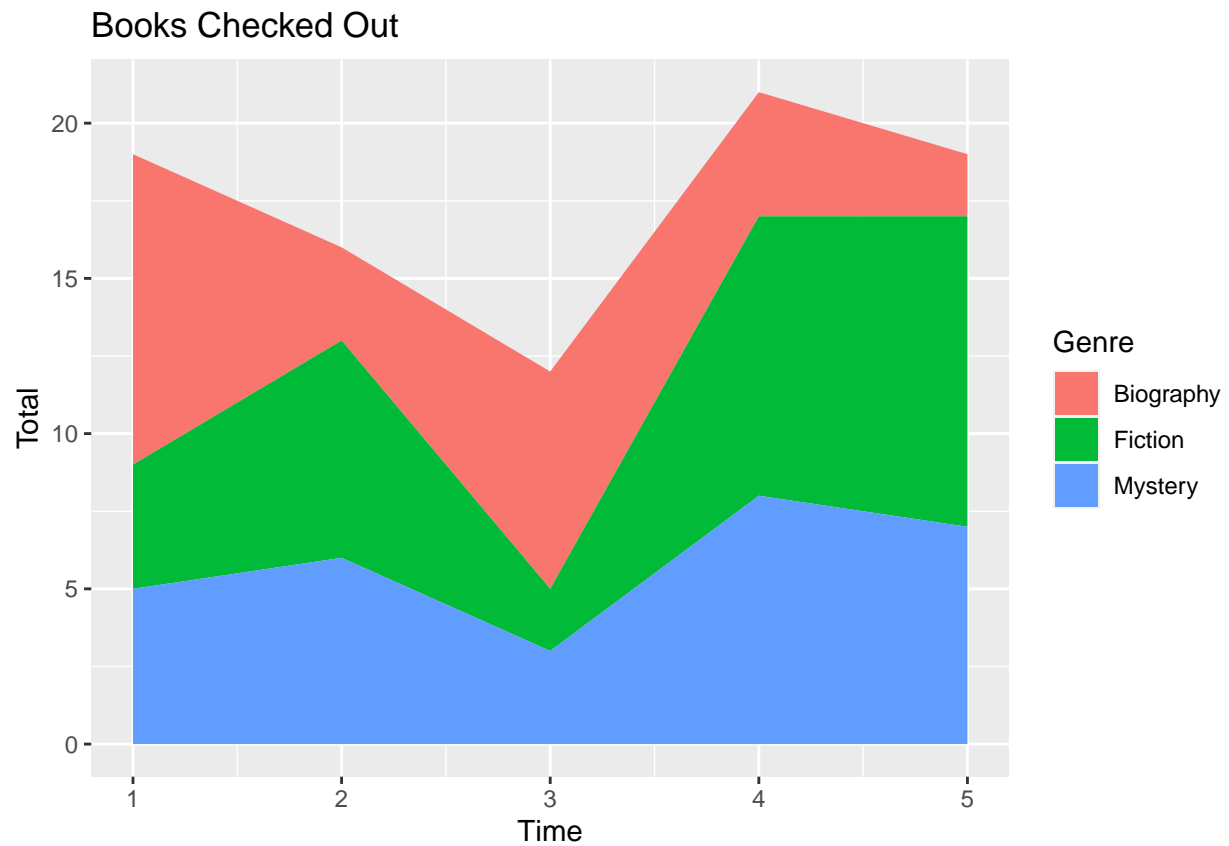
## Problem 6

Create the figure in the solution for Problem 6, using the data included in the R Markdown file.

```
####Make sure you load any necessary libraries
```

```
####PUT YOUR CODE HERE
```

```
ggplot(books_checked_out)+  
  geom_area(mapping=aes(x=Time,y=Total,fill=Genre))+  
  ggtitle("Books Checked Out")
```



## Problem 7



Create the figure in the solution for Problem 7, using the data included in the R Markdown file.

```
####Make sure you load any necessary libraries
#install.packages("SciencesPo")
library(ggalt)
#library(SciencesPo)
####PUT YOUR CODE HERE

# create a vector with letters in the desired order
x <- c("Biography", "Mystery", "Fiction")

books_checked_out2 <- books_checked_out2 %>%
  slice(match(x, Genre))

books_checked_out2
```

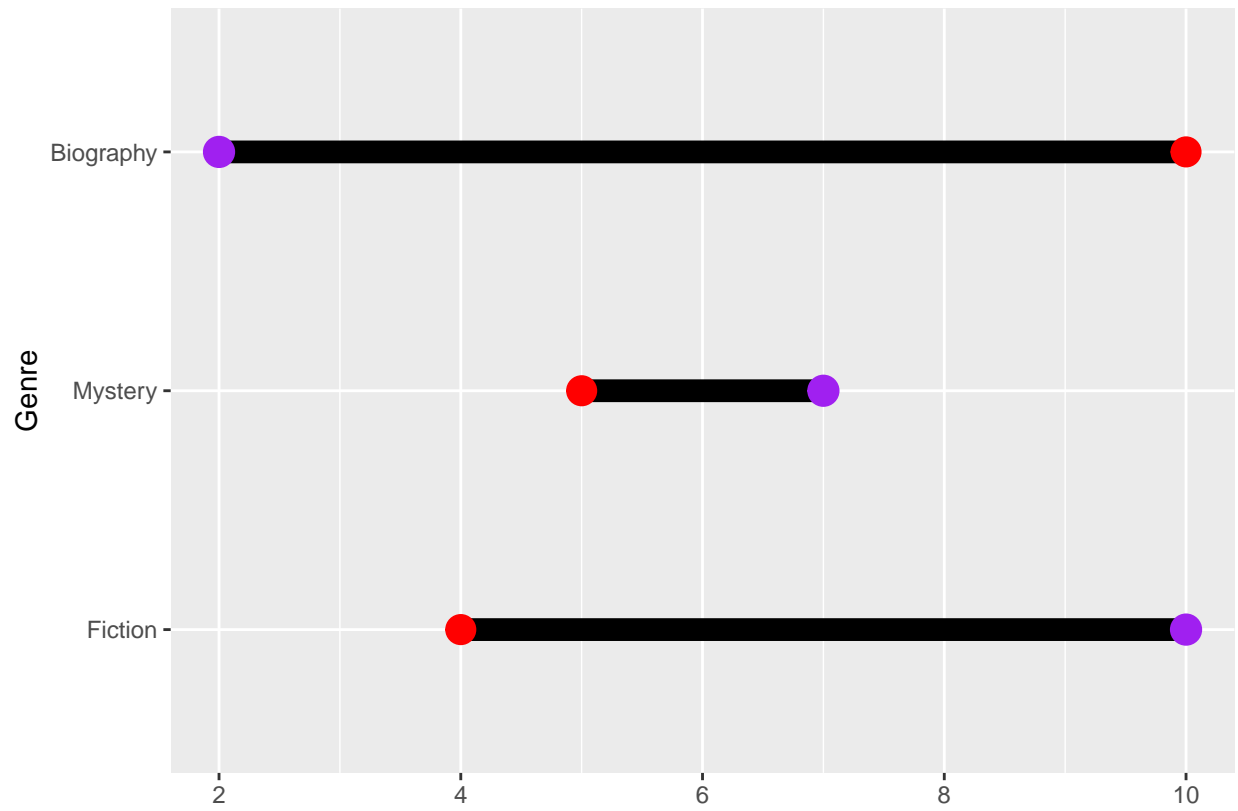
```
## # A tibble: 3 x 3
##   Genre      Time1 Time5
##   <chr>      <int> <int>
## 1 Biography    10     2
## 2 Mystery      5     7
## 3 Fiction      4    10
```

```
ggplot(books_checked_out2, aes(x=Time1, xend=Time5, y= reorder(Genre, Time1-Time5))) +
  #geom_point(size=5)+
  geom_dumbbell(size=4, point.size.r=5, point.size.l=5, size=0.5, color="black",
    colour_x = "red", colour_xend = "purple",
    dot_guide=FALSE, dot_guide_size=0)+

  ylab("Genre")+
  xlab("")
```

```
## Warning: Duplicated aesthetics after name standardisation: size
```

```
## Warning: Ignoring unknown parameters: point.size.r, point.size.l
```



## Problem 8

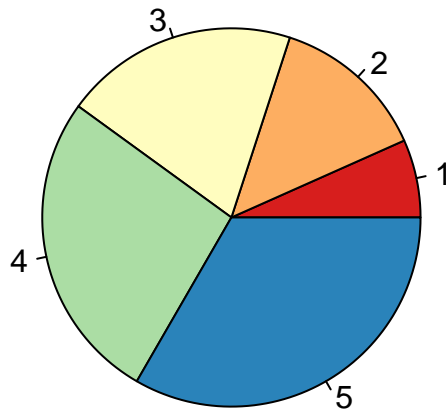
Create the figure in the solution for Problem 8, using the data included in the R Markdown file.

*###HINT: check the Help file for pie to see how to specify the color to the my\_color palette*

```
library(RColorBrewer)
```

```
my_color<-brewer.pal(5,"Paired")
```

```
pie(pie_dat, col=c("#d71e1d", "#fdae61", "#ffdbf", "#abdda4", "#2a83ba"))
```



*####Make sure you load any necessary libraries*

*####PUT YOUR CODE HERE*