Stat. 651 Quiz 1 Solution

Prof. Eric A. Suess

Instructions: This is an open book, open notes, and open Google/internet test. You may use R on your own computer or the lab computer. You may use a calculator.

Type your answers to the questions into an R Notebook. Answer the questions in order. Answer each the question that is asked above your R code chunks. You must write a sentence containing your answer.

Your files should have a name in the usual form for the class **lastname_firstname_Stat651_Quiz1.Rmd** and **lastname_firstname_Stat651_Quiz1.docx** or **lastname_firstname_Stat651_Quiz1.pdf**. Submit **both** files in Backboard before the end of class period. Also, submit the .pptx file of the Tableau Story.

Academic Honesty: As a student at CSU East Bay you are held to the standards stated in the Academic Dishonesty Policy. Copying another student's work or allowing another student to copy your work is academically dishonest. I expect you to be academically honest while taking the test.

These question is related to the homework from Chapter 3.

For the *RailTrail* dataset from the *mosaicData* R package answer the following questions. The data is from Northampton, MA.

```
library(tidyverse)
library(mosaic)
```

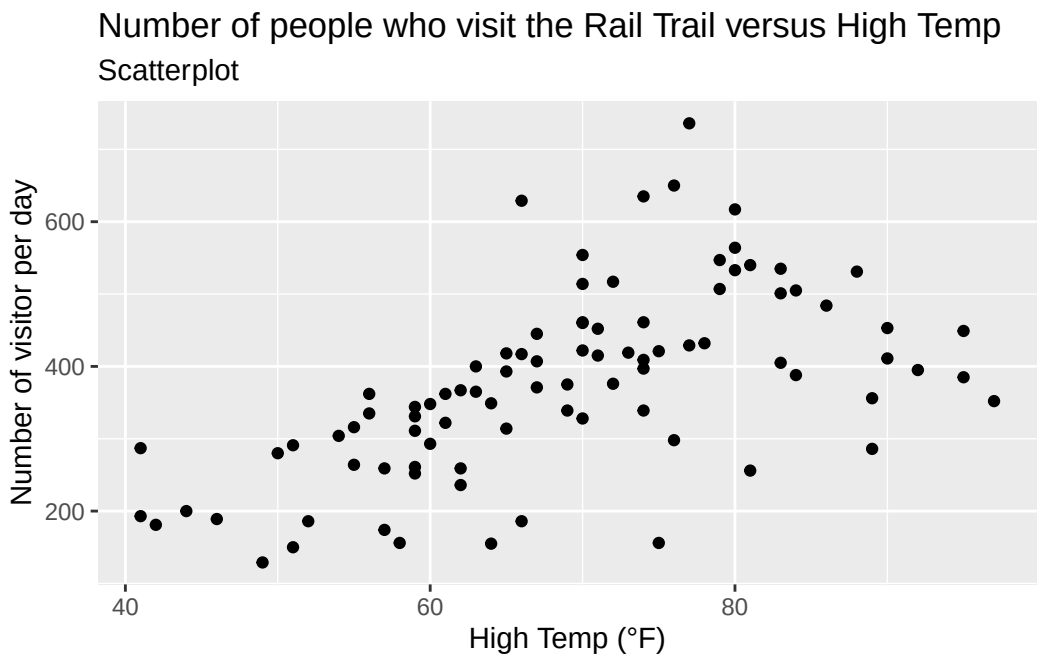
1. Create a scatterplot of the *volume* (number of crossing per day) against the *high temperature* that day.

Answer: Note that "y against x" and "y vs x" and "y on the y-axis and x on the x-axis" and "y depends on x" all mean the same thing.

```
head(RailTrail)
```

```
  hightemp lowtemp avgtemp spring summer fall cloudcover precip volume weekday
1       83      50    66.5      0      1    0        7.6   0.00    501    TRUE
2       73      49    61.0      0      1    0        6.3   0.29    419    TRUE
3       74      52    63.0      1      0    0        7.5   0.32    397    TRUE
4       95      61    78.0      0      1    0        2.6   0.00    385   FALSE
5       44      52    48.0      1      0    0       10.0   0.14    200    TRUE
6       69      54    61.5      1      0    0        6.6   0.02    375    TRUE
  dayType
1 weekday
2 weekday
3 weekday
4 weekend
5 weekday
6 weekday
```

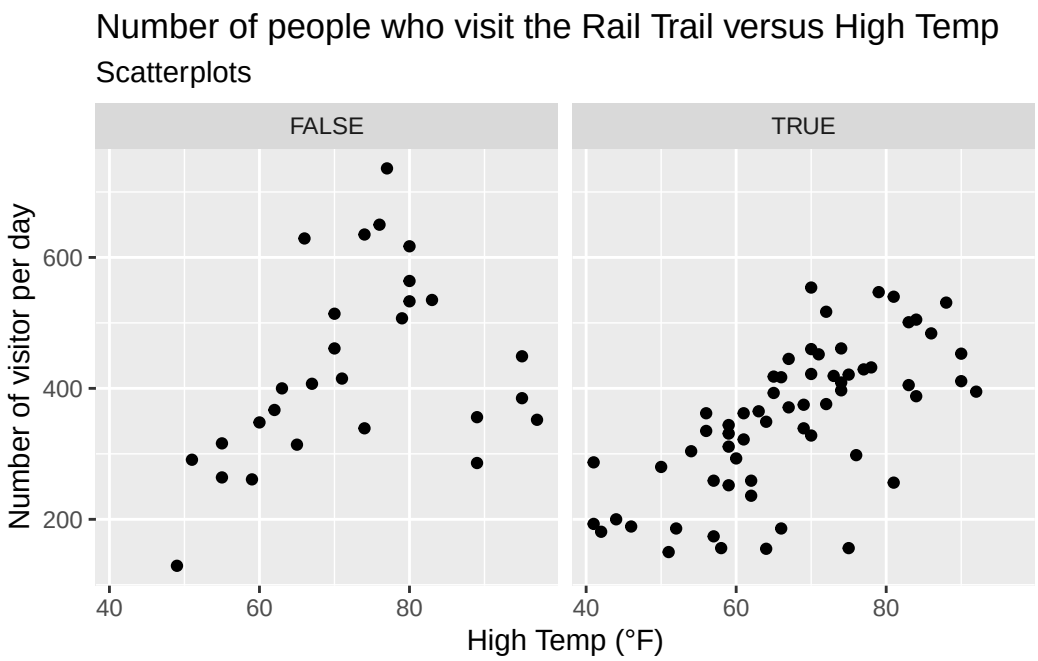
```
p <- RailTrail %>% ggplot(aes(y = volume, x = hightemp)) +
  geom_point() +
  labs(title = "Number of people who visit the Rail Trail versus High Temp",
       subtitle = "Scatterplot"
       ) +
  ylab("Number of visitor per day") + xlab("High Temp (°F)")
p
```



2. Separate your previous plot into facets by *weekday*.

Answer: Note that you can use *facet_wrap* or *facet_grid*.

```
p2 <- p + facet_wrap(~weekday) +
  labs(subtitle = "Scatterplots")
p2
```



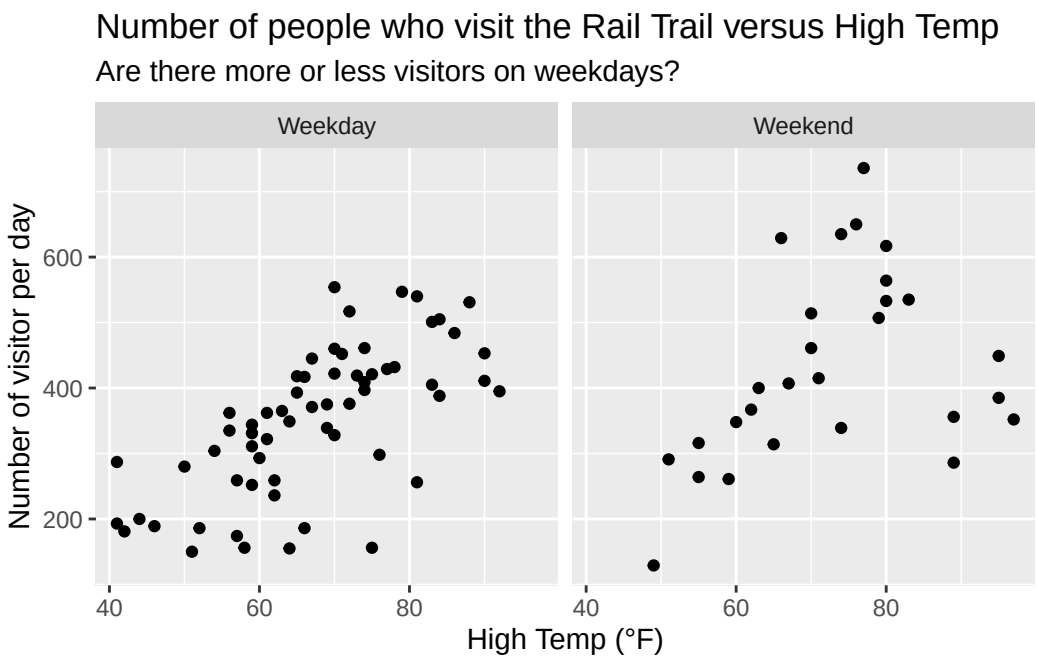
3. Examine the plot you have created in the previous question, is there anything about the plot that is unclear? Suggest a way to fix the issue you have described. Make a improved plot.

Answer: Yes, FALSE and TRUE do not mean anything in the plot. So change the values of FALSE and TRUE to weekday and weekend. Upon further reading the weekday variable is an indicator of non-holiday weekdays, which is slightly more specific than what has been asked. The *dayType* variable is more appropriate to use.

```
RailTrail2 <- RailTrail %>% mutate( dayType = fct_recode(dayType,
                                                         "Weekday" = "weekday",
                                                         "Weekend" = "weekend") )

p <- p %+% RailTrail2
```

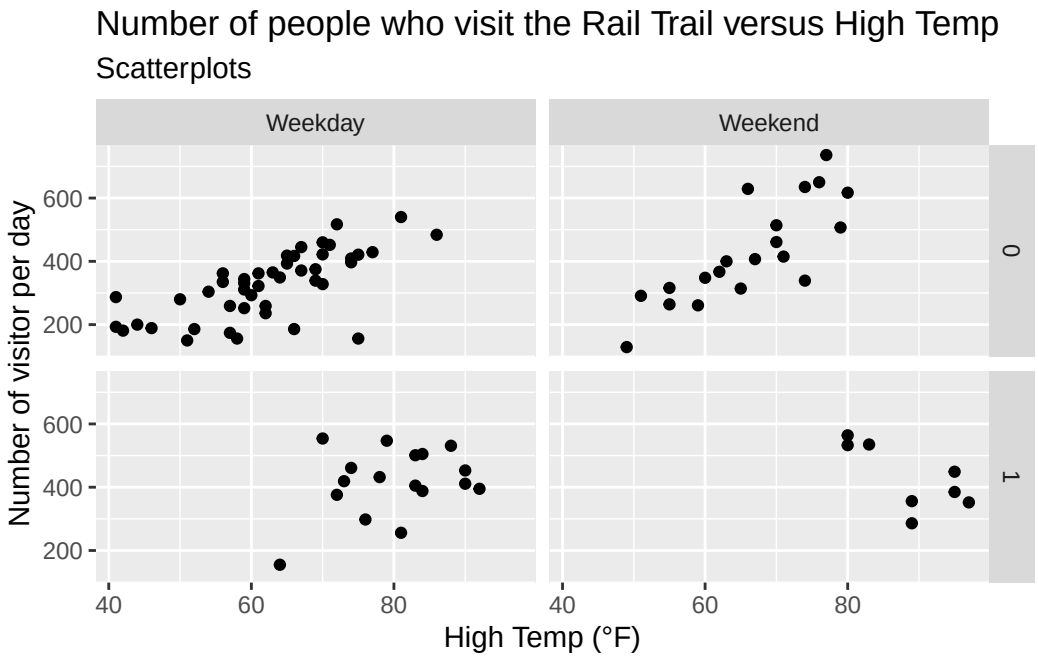
```
p2 <- p + facet_wrap(~ dayType) +
  labs(subtitle = "Are there more or less visitors on weekdays?")
p2
```



4. Separate your plot into facets by summer and weekday. The summer variable used for the rows and the weekday variable for the columns.

Answer: Again you can use *facet_wrap* or *facet_grid*, but you should be consistent.

```
p3 <- p + facet_grid(summer ~ dayType) +
  labs(subtitle = "Scatterplots")
p3
```



5. Examine the plot you have created in the previous question, is there anything about the plot that is unclear? Suggest a way to fix the issue you have described. Make a improved plot. Hint: Change the *summer* variable to a factor and use %+% to replace the data in the original plot p.

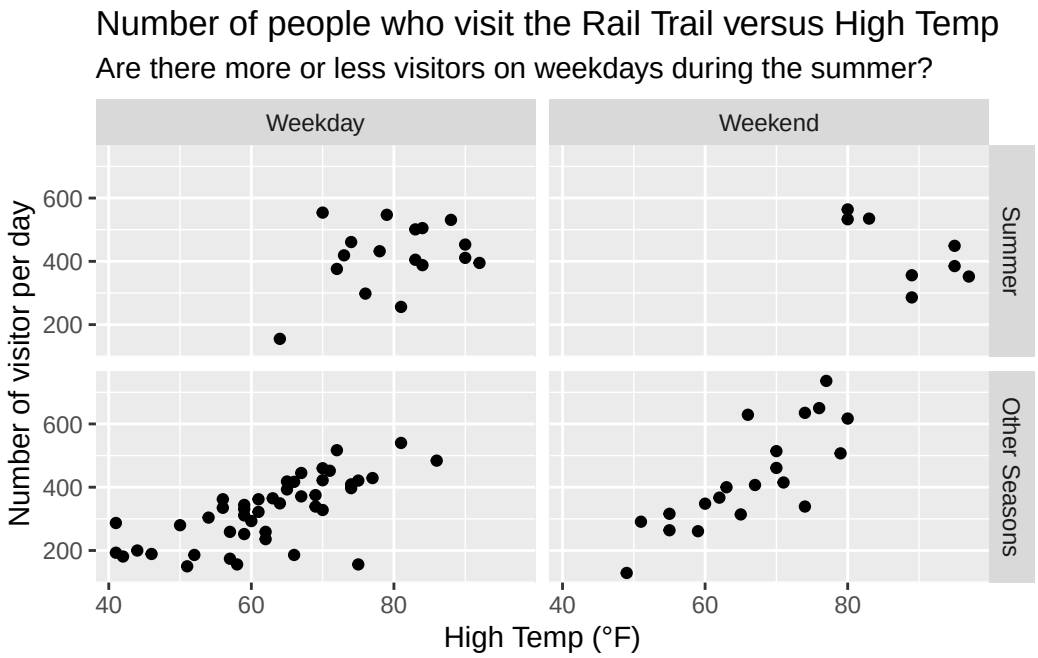
Answer: Yes, the values 0 and 1 have no meaning. Use the labels *not summer* and *summer*. This can be done in many ways, using *factor* or *ifelse* or the *labeller* option in ggplot. My hint was to use the previous plot and just replace the data with %+%.

```
RailTrail2 <- RailTrail2 %>% mutate( summer = fct_rev(factor(summer,
                                                levels = c(0,1),
                                                labels = c("Other Seasons", "Summer")))

# Note that fct_rev reverse the order of the factor so Summer is on top of the plot.

p <- p %+% RailTrail2

p3 <- p + facet_grid(summer ~ dayType) +
  labs(subtitle = "Are there more or less visitors on weekdays during the summer?")
p3
```

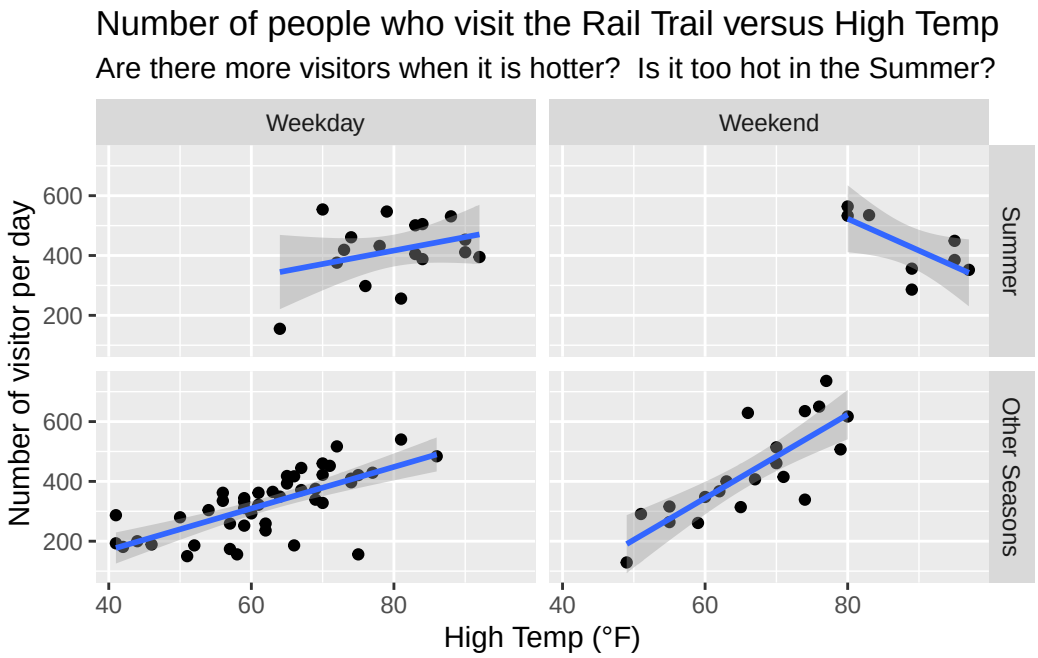


6. Add regression lines to the four facets. When does the relationship between *volume* and *hightemp* change?

Answer: The relationship is postive, but changes in the Summer on weekends.
When it gets too hot the volume of people visiting the park goes down.

```
p4 <- p3 + geom_smooth(method=lm) +
  labs(subtitle = "Are there more visitors when it is hotter?  Is it too hot in the Summer
p4
```

```
`geom_smooth()` using formula 'y ~ x'
```



7. (Extra Credit) Compute the slope of each regresion line. Hint: Use the map function from the **purrr* R package.

Answer: This is a nice opportunity to use the *map* function from the *purrr* R package.

```
library(purrr)
library(kableExtra)
```

```
Attaching package: 'kableExtra'
```

```
The following object is masked from 'package:dplyr':

    group_rows
```

```
mod_fn <- function(df) {
  lm(volume ~ hightemp, data = df)
}

b_fun <- function(mod){
  coefficients(mod)[[2]]
}
```

```
RailTrail2_m <- RailTrail2 %>% unite(SummerDayType, c("summer", "dayType")) %>%
  group_by(SummerDayType) %>%
  nest() %>%
  mutate(model = map(data, mod_fn))

RailTrail2_m %>% transmute(SummerDayType, beta = map_dbl(model, b_fun)) %>%
  arrange(desc(SummerDayType)) %>%
  kable()
```

SummerDayType	beta
Summer_Weekend	-10.641566
Summer_Weekday	4.494804
Other Seasons_Weekend	13.959904
Other Seasons_Weekday	6.950167

8. Plot *volume* versus each of the following variables: *hightemp*, *lowtemp*, *cloudcover*, *precip*. Add regression lines. Put the 4 plots into one ggplot using a function from the **cowplot** R package.

Answer: It is always best to use a matrix of plots for scatterplot so the scales are not so different.

```
library(cowplot)
```

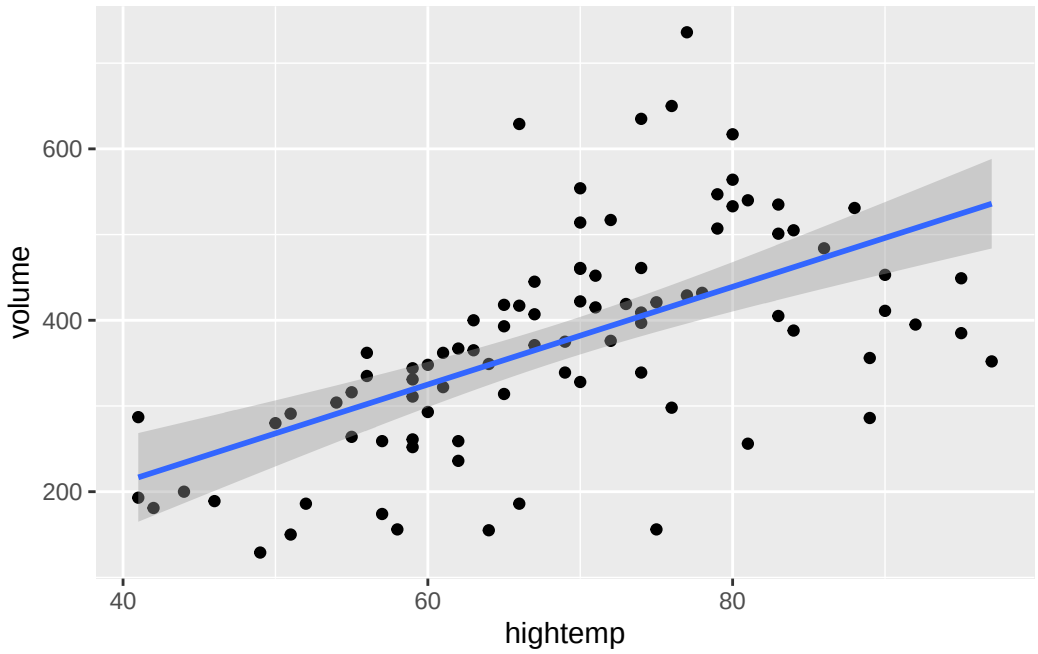
```
Attaching package: 'cowplot'
```

```
The following object is masked from 'package:mosaic':

    theme_map
```

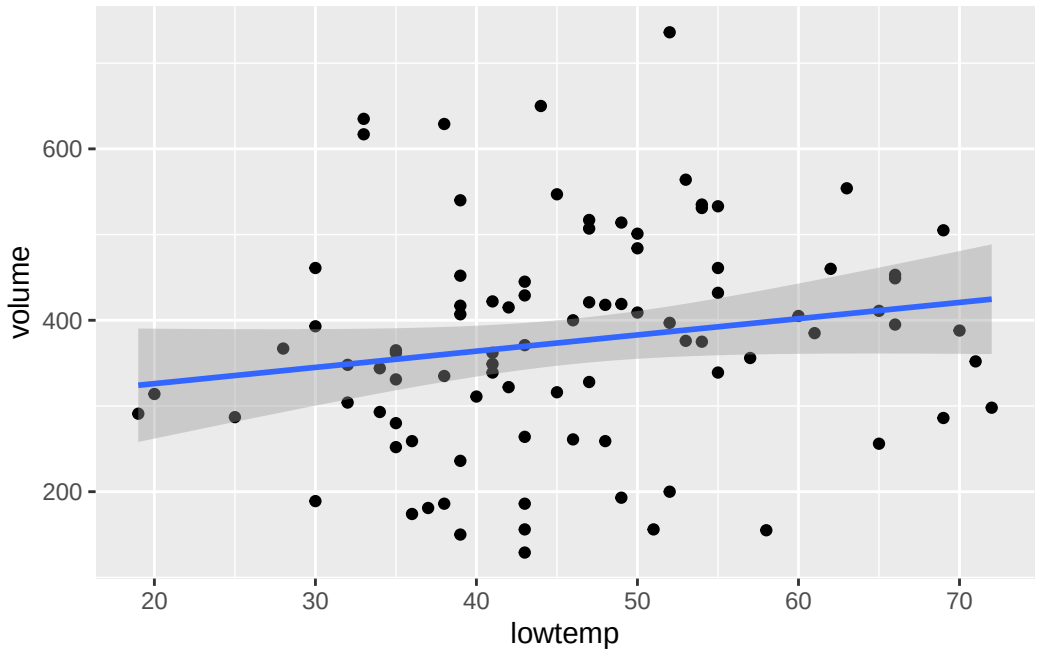
```
g1 <- RailTrail %>% ggplot(aes(y = volume, x = hightemp)) +
  geom_point() +
  geom_smooth(method=lm)
g1
```

```
`geom_smooth()` using formula 'y ~ x'
```



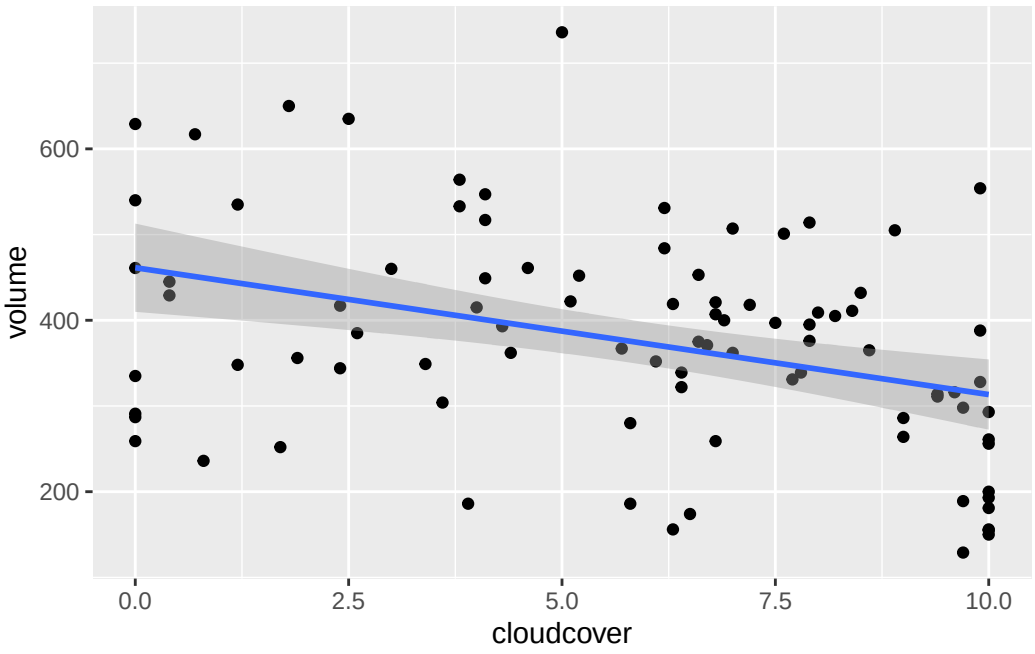

```
g2 <- RailTrail %>% ggplot(aes(y = volume, x = lowtemp)) +
  geom_point() +
  geom_smooth(method=lm)
g2
```

```
`geom_smooth()` using formula 'y ~ x'
```



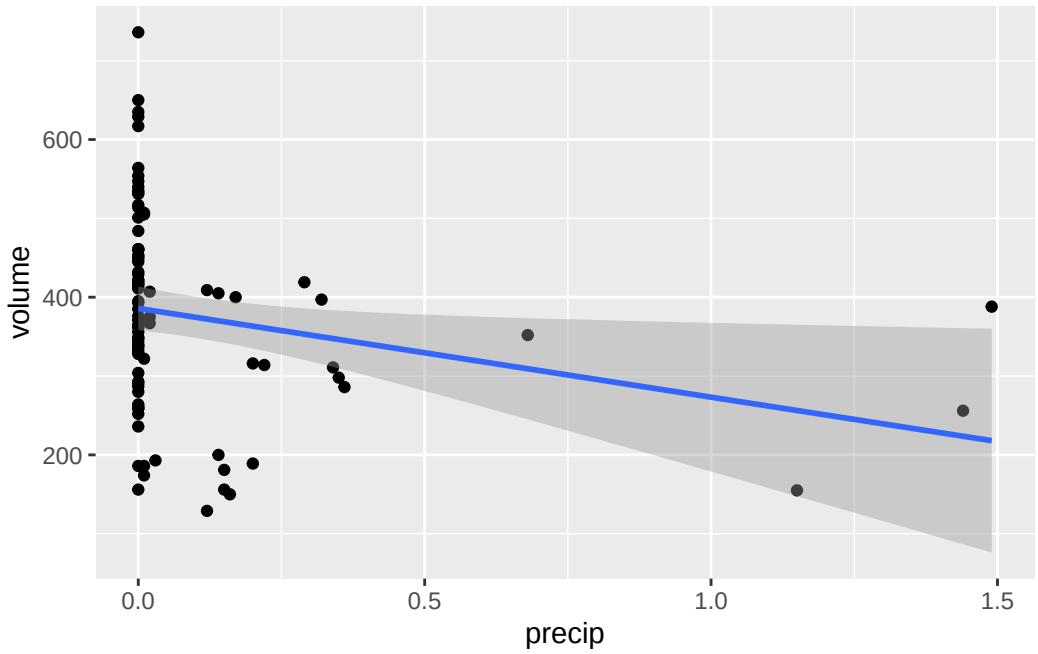

```
g3 <- RailTrail %>% ggplot(aes(y = volume, x = cloudcover)) +
  geom_point() +
  geom_smooth(method=lm)
g3
```

```
`geom_smooth()` using formula 'y ~ x'
```




```
g4 <- RailTrail %>% ggplot(aes(y = volume, x = precip)) +
  geom_point() +
  geom_smooth(method=lm)
g4
```

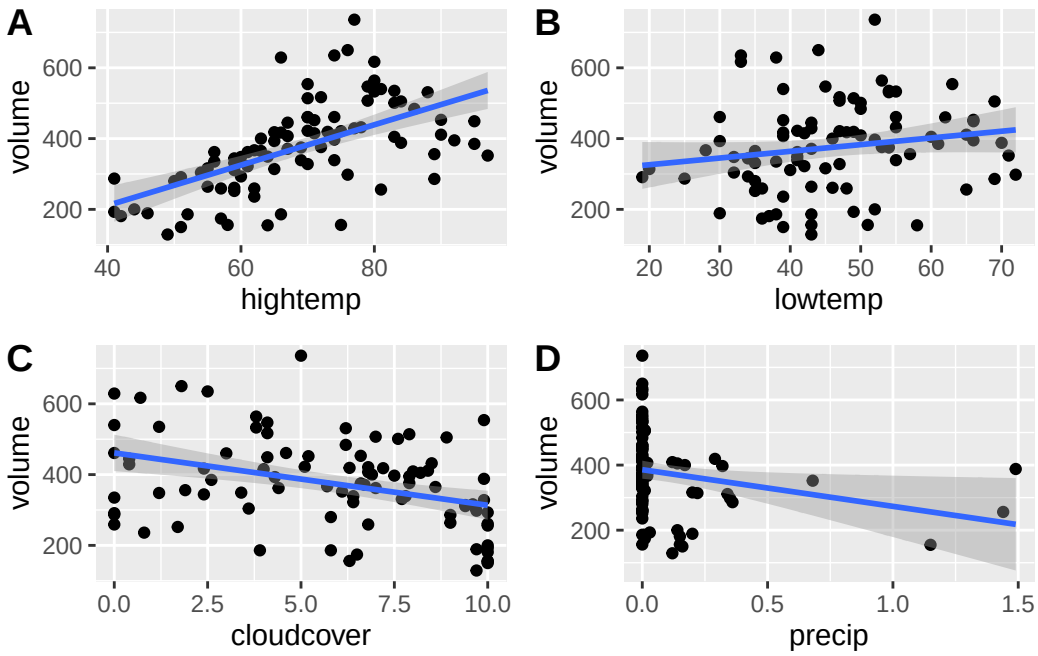
```
`geom_smooth()` using formula 'y ~ x'
```




```
plot_grid(g1, g2, g3, g4, labels = c('A', 'B', 'C', 'D'))
```

```
`geom_smooth()` using formula 'y ~ x'
```

```
`geom_smooth()` using formula 'y ~ x'
`geom_smooth()` using formula 'y ~ x'
`geom_smooth()` using formula 'y ~ x'
```



If you want to add the formula for the regression lines to the plot, here is one suggestion.

```
library(ggpubr)
```

```
Attaching package: 'ggpubr'
```

```
The following object is masked from 'package:cowplot':

    get_legend
```

```
p1 <- ggscatter(RailTrail,
  y = "volume", x = "hightemp",
  add = "reg.line") +
  stat_regline_equation(label.x = 45, label.y = 550)

p2 <- ggscatter(RailTrail,
  y = "volume", x = "lowtemp",
  add = "reg.line") +
  stat_regline_equation(label.x = 50, label.y = 650)
```

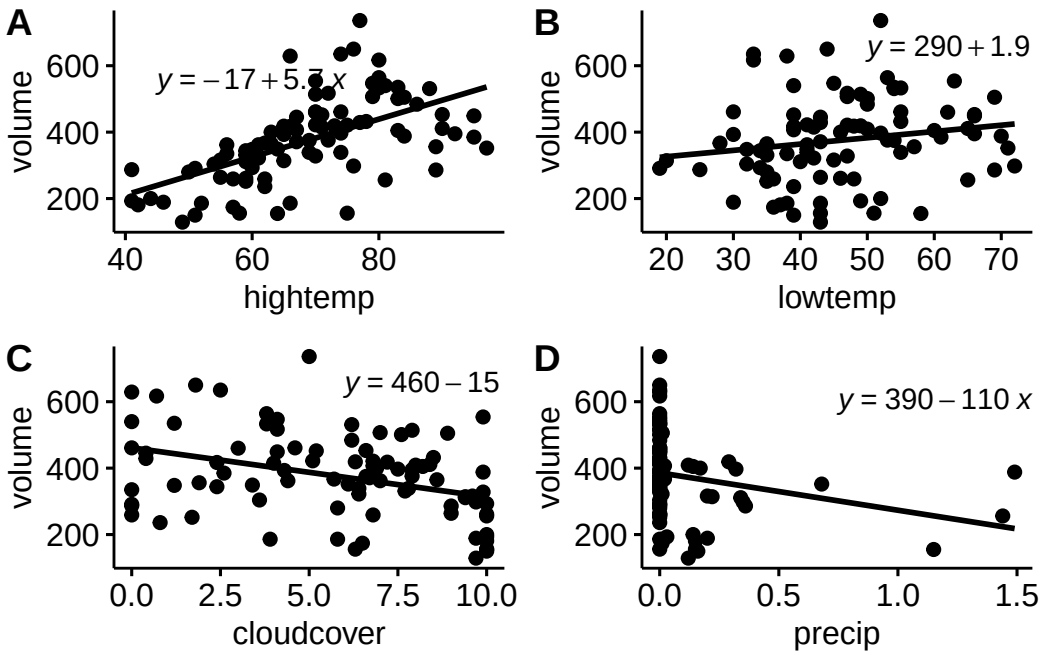
```r
p3 <- ggscatter(RailTrail,
  y = "volume", x = "cloudcover",
  add = "reg.line") +
  stat_regline_equation(label.x = 6, label.y = 650)

p4 <- ggscatter(RailTrail,
  y = "volume", x = "precip",
  add = "reg.line") +
  stat_regline_equation(label.x = .75, label.y = 600)

plot_grid(p1, p2, p3, p4, labels = c('A', 'B', 'C', 'D'))
```

```
`geom_smooth()` using formula 'y ~ x'
```

```
`geom_smooth()` using formula 'y ~ x'
`geom_smooth()` using formula 'y ~ x'
`geom_smooth()` using formula 'y ~ x'
```



9. Export the RailTrail data to a .csv file and load it into the Tableau.

Answer: Run the code and use the .csv file in Tableau.

```
write_csv(RailTrail, "RailTrail.csv")
```

10. Make the same plots in Tableau. Arrange them into a Story. Export the Story as a .pptx file.

Answer: Export to .pptx.