Using RStudio and Rmarkdown

Upload the data file we want to work from

In the bottom-right panel, we see all the files associated with this project. Right now, there aren't many! We need to upload the data file so that we can work with it.

Load the data into R

Our data now lives in the cloud, but R is still not ready to work with it. Notice, the environment is empty. We can load the data in using "Import Dataset" button. Let's give it the name "scr" for short. Now it's in the environment and ready to be used!

The data is in a format called a data frame, which is a fancy way of saying it is arranged with each variable in a column. We can inspect it by clicking on it in the environment.

We can use code chunks to embed analyses

If we click "Insert Chunk" then we can embed R code into our document like so, and run it!

```
mean(scr$hap)
```

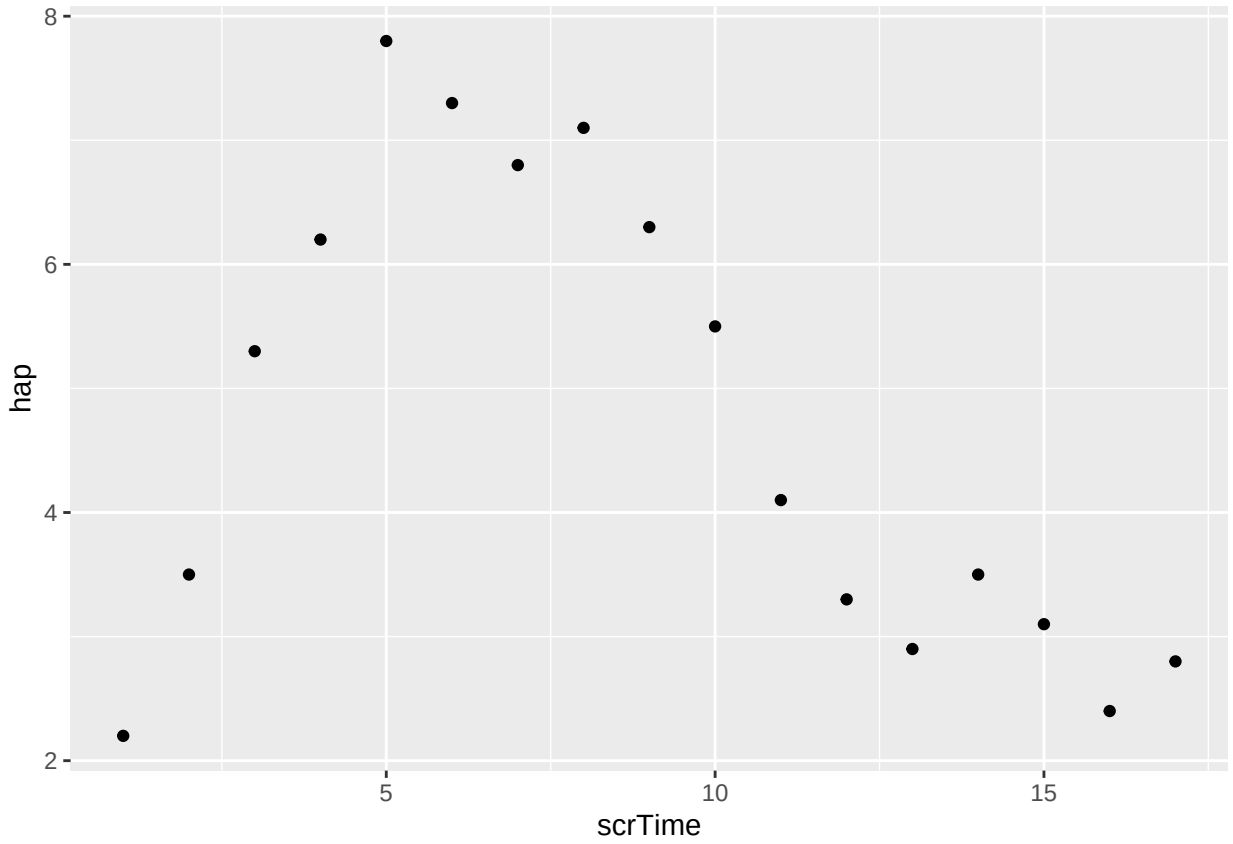
```
## [1] 4.711765
```

```
#Note: when we use $hap, we are extracting that column from the data frame!
```

We can make basic plots

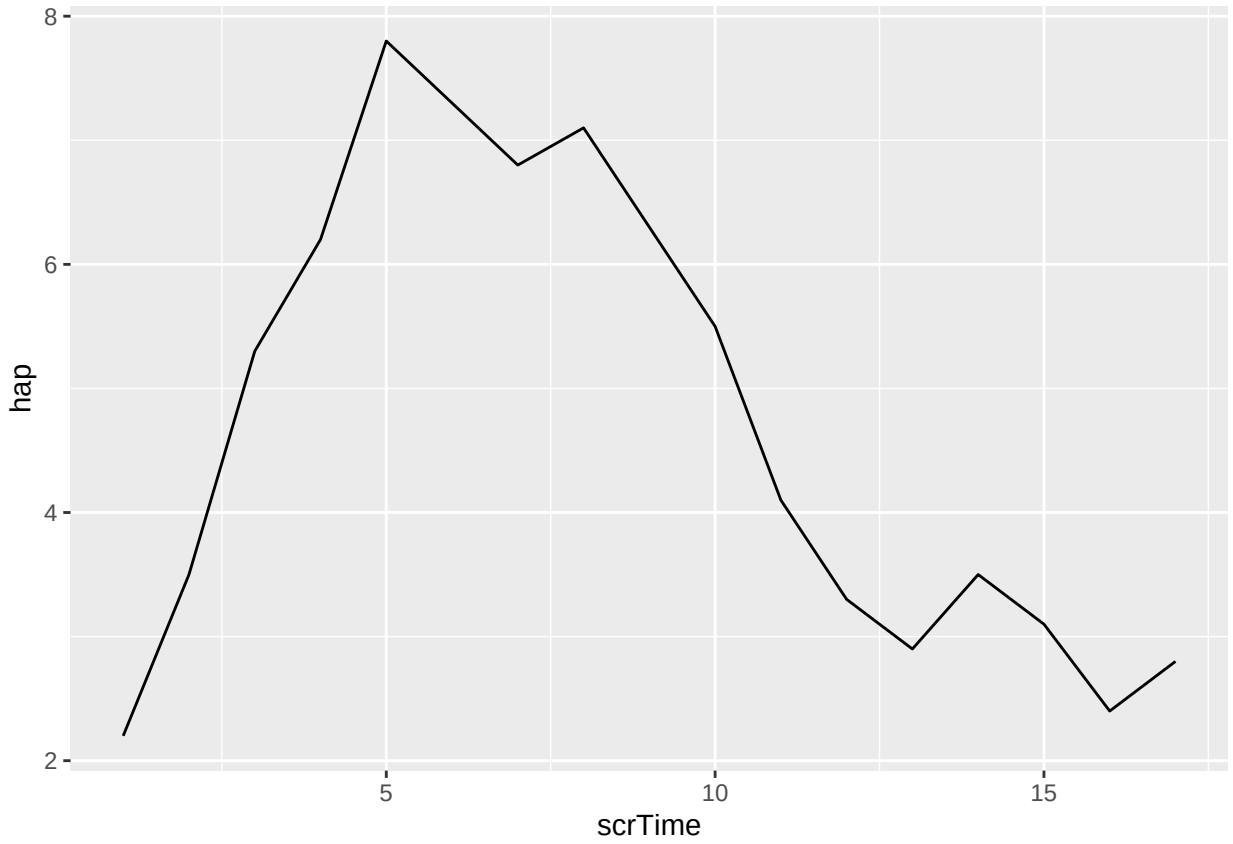
We can use ggplot to make a simple scatterplot of screen time vs happiness.

```
ggplot(scr, aes(x=scrTime,y=hap)) + geom_point()
```



Or we could make a line graph.

```
ggplot(scr, aes(x=scrTime,y=hap)) + geom_line()
```

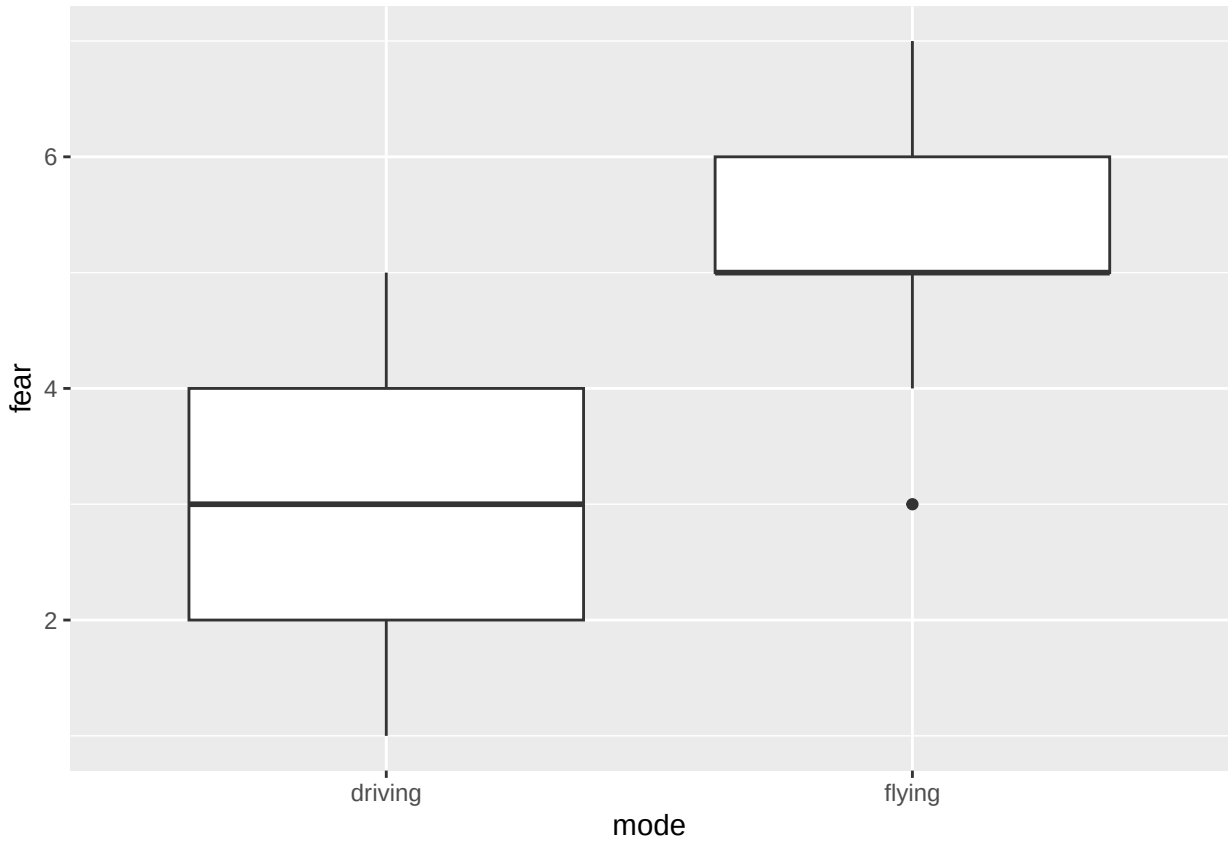


We can make lots of different types of plots

Let's look at another type of data: group comparisons. In data science, we sometimes call this an A/B test. We'll load in a new dataset where we can compare fear of flying vs fear of driving.

We'll make boxplots of the two groups's data.

```
ggplot(travel, aes(x=mode,y=fear))+geom_boxplot()
```



Compare the means using a t-test

We can do a statistical test with a single line of code.

```
t.test(fear~mode,
       data=travel,
       var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  fear by mode
## t = -8.4, df = 48, p-value = 5.511e-11
## alternative hypothesis: true difference in means between group driving and group flying is not equal
## 95 percent confidence interval:
##  -2.924889 -1.795111
## sample estimates:
## mean in group driving  mean in group flying
##                  2.84                  5.20
```

Confidence interval plots

We can summarize our findings using a confidence interval plot. First we need to obtain the two relevant quantities: means and confidence intervals for each travel mode.

In R we can do this using the group_by() and summarize() functions. We will use the pipe operator %>% to apply these functions in turn.

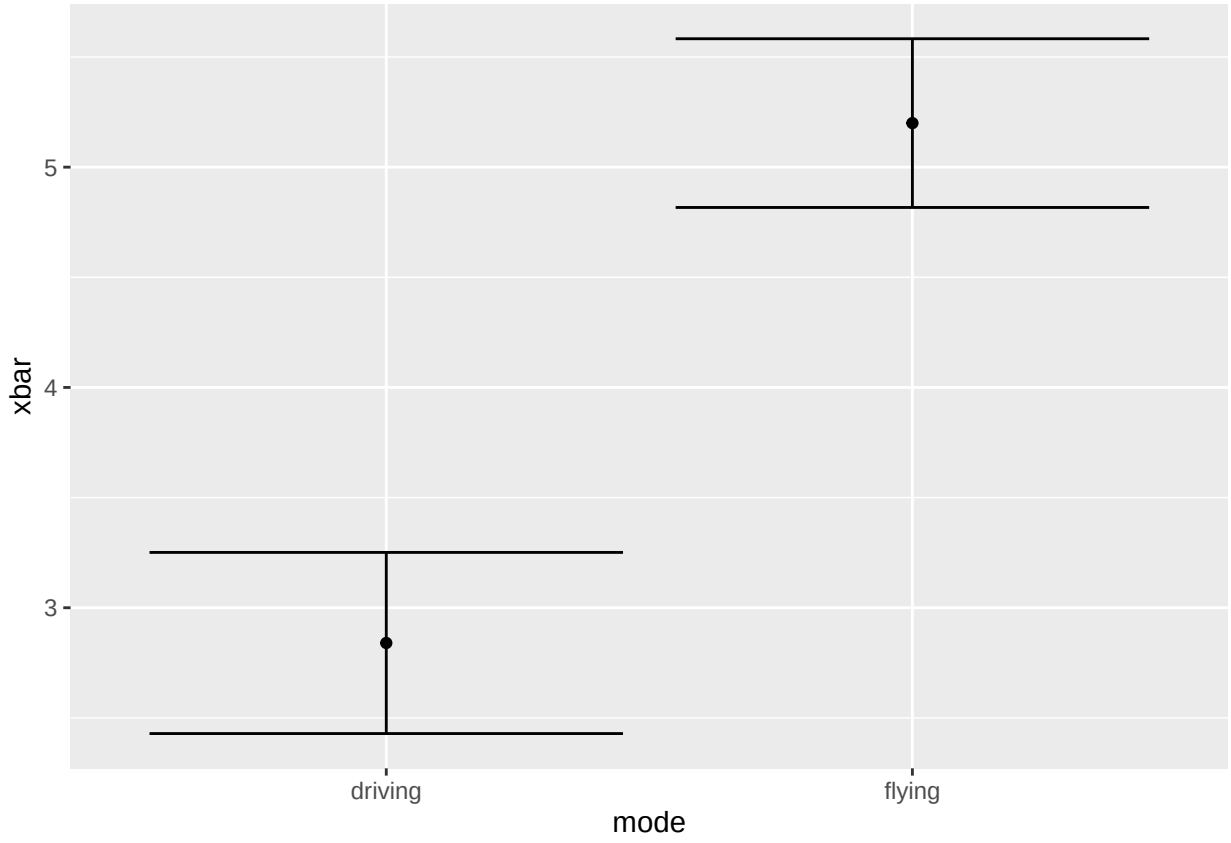
```
estimates <- travel %>%
              group_by(mode) %>%
              summarize(n = n(),
                        xbar = mean(fear),
                        se = sd(fear)/sqrt(n),
                        ci = 2*se)

print(estimates)
```

```
## # A tibble: 2 x 5
##   mode        n  xbar    se    ci
##   <chr>   <int> <dbl> <dbl> <dbl>
## 1 driving    25  2.84 0.206 0.411
## 2 flying     25  5.2  0.191 0.383
```

Let's make a plot with the estimates and confidence intervals. We'll use our new summary table, called "estimates."

```
ggplot(estimates, aes(x=mode,y=xbar))+
  geom_point() +
  geom_errorbar(aes(ymin= xbar - ci, ymax = xbar + ci))
```



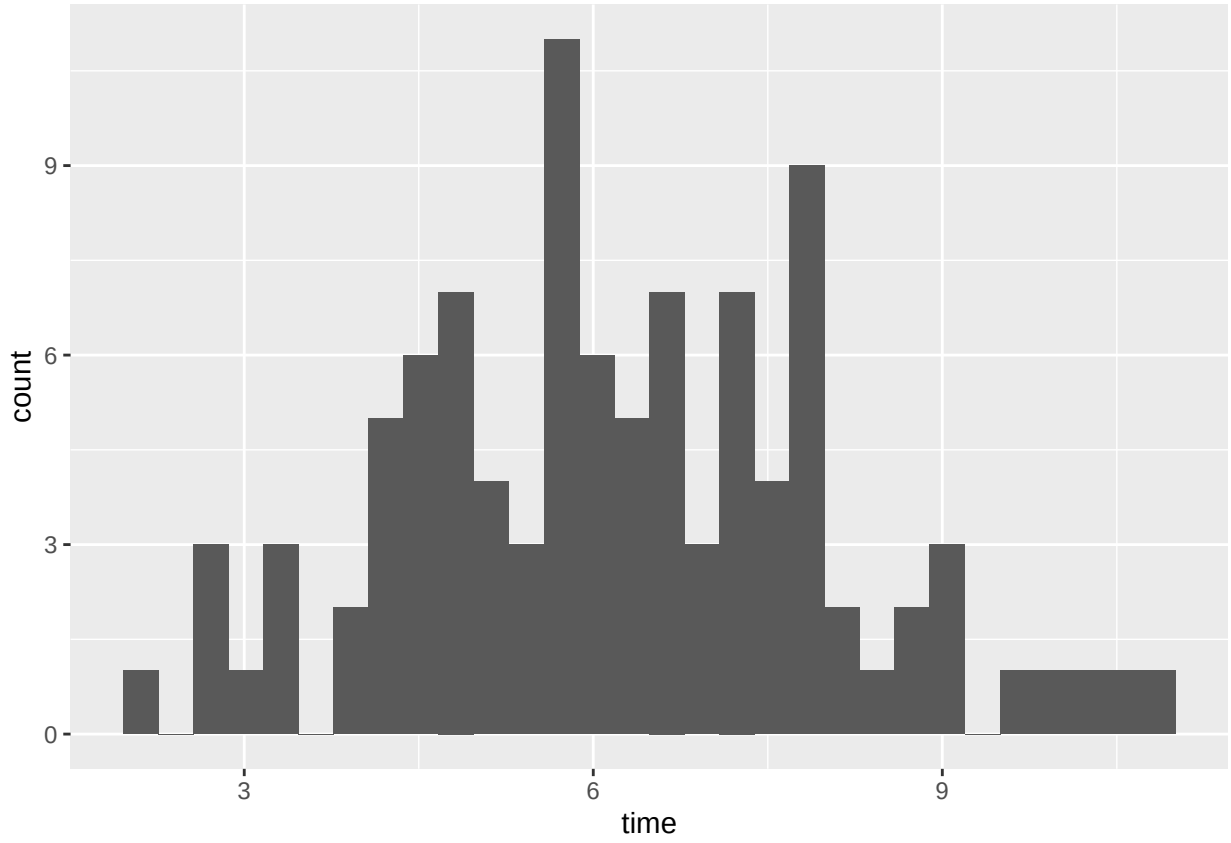
Let's do a basic regression analysis

We've seen how to make a scatterplot. Now let's do a formal regression analysis of some X/Y data.

We'll use a dataset that measured people's time spent on social media as well as their level of depression and social media addiction. Let's make some basic descriptive plots.

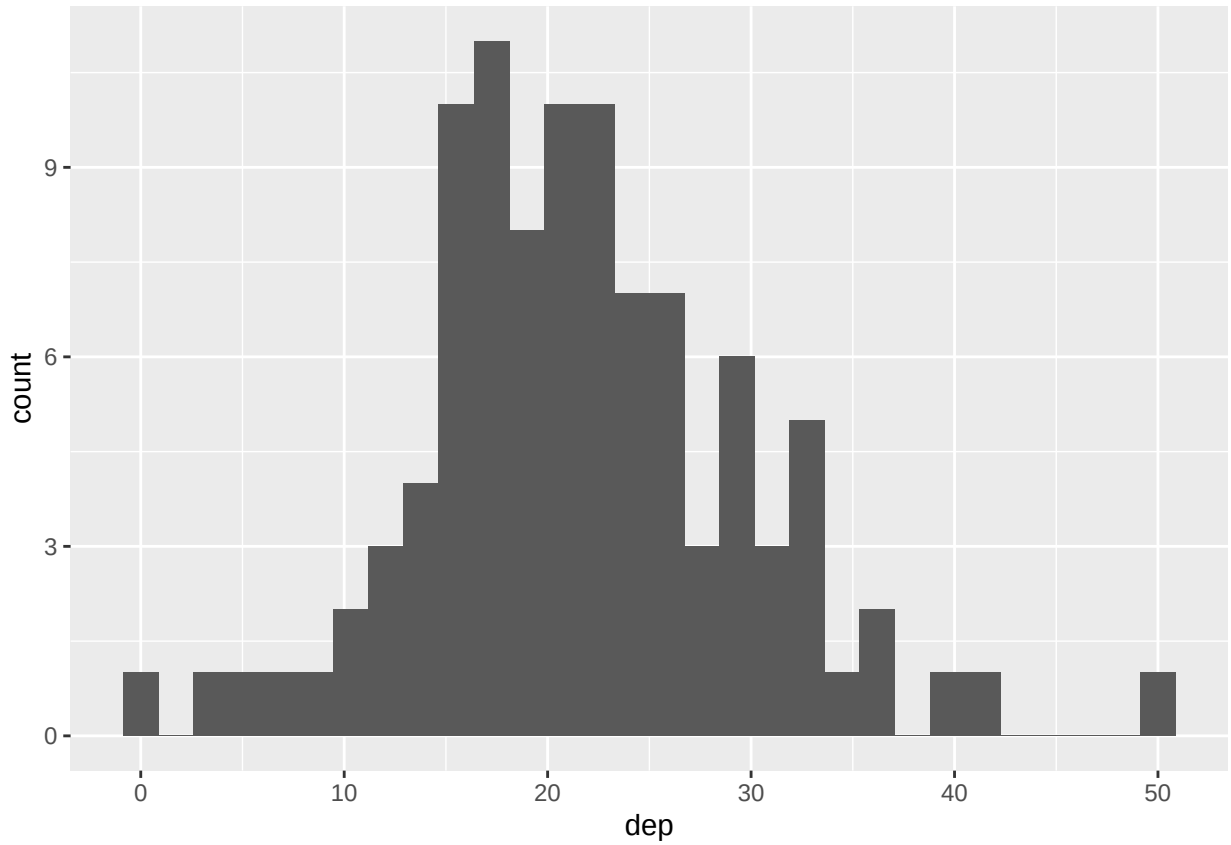
```
ggplot(socmed, aes(x=time))+geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



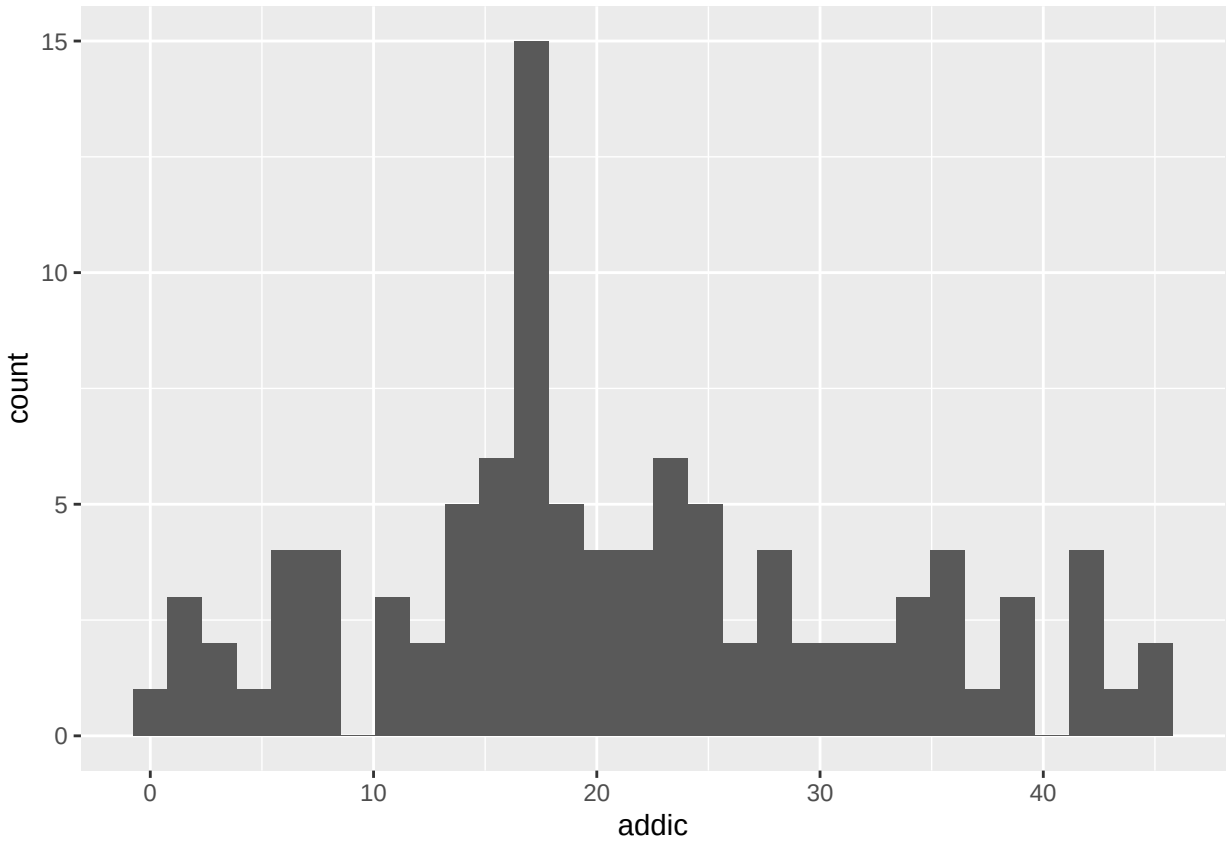

```
ggplot(socmed, aes(x=dep))+geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(socmed, aes(x=addic))+geom_histogram()
```

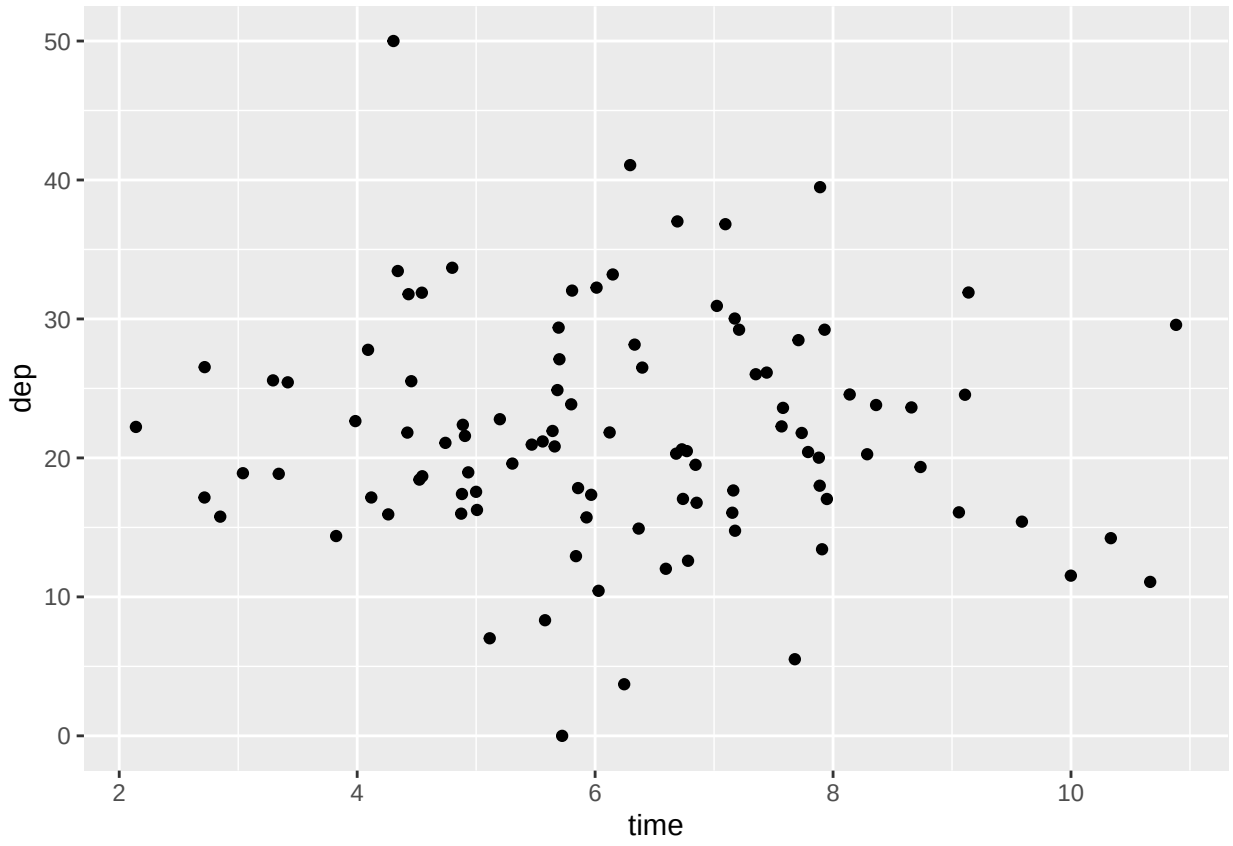
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Hypothesis time!

We might hypothesize that those who spend more time on social media may have higher levels of depression. We'll make a scatter plot showing time spent on the x-axis and depression score on the y-axis.

```
ggplot(socmed, aes(x = time, y = dep)) + geom_point()
```



Fit the linear model

We'll fit a linear regression model to these data with depression as the outcome and time as the predictor.

```
model <- lm(dep ~ time, data = socmed)
summary(model)
```

```
##
## Call:
## lm(formula = dep ~ time, data = socmed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.8169  -4.9542  -0.9255   4.6367  27.8376
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  23.2111     2.8750   8.073 1.77e-12 ***
## time         -0.2436     0.4443  -0.548    0.585
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.121 on 98 degrees of freedom
## Multiple R-squared:  0.003059,   Adjusted R-squared:  -0.007114
## F-statistic: 0.3007 on 1 and 98 DF,  p-value: 0.5847
```

We'll also add the estimated regression line to our plot.

```
ggplot(socmed, aes(x = time, y = dep)) + geom_point() + geom_smooth(method="lm")
```

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