Report

Marcus Nunes

18 November, 2019

Contents

1 Introduction

We will be using this .Rmd file as a template for this short course. Please save a copy and edit it as we go through. Every change we make to this source file can be visualized using the command ctrl + shift + K (command + shift + K on a Mac).

2 Report

Let's take a look at the cars dataset, present in R's memory. From now on, every snippet of this `.Rmd` file that starts with ` ```{r} ` will be called `chunk`. Each `chunk` is a snippet of R code that will be executed and whose output will go directly to the final report.

```
cars
```

```
##    speed dist
## 1      4    2
## 2      4   10
## 3      7    4
## 4      7   22
## 5      8   16
## 6      9   10
## 7     10   18
## 8     10   26
## 9     10   34
## 10    11   17
## 11    11   28
## 12    12   14
## 13    12   20
## 14    12   24
## 15    12   28
## 16    13   26
## 17    13   34
## 18    13   34
## 19    13   46
```

```
## 20    14   26
## 21    14   36
## 22    14   60
## 23    14   80
## 24    15   20
## 25    15   26
## 26    15   54
## 27    16   32
## 28    16   40
## 29    17   32
## 30    17   40
## 31    17   50
## 32    18   42
## 33    18   56
## 34    18   76
## 35    18   84
## 36    19   36
## 37    19   46
## 38    19   68
## 39    20   32
## 40    20   48
## 41    20   52
## 42    20   56
## 43    20   64
## 44    22   66
## 45    23   54
## 46    24   70
## 47    24   92
## 48    24   93
## 49    24  120
## 50    25   85
```

This is a 50 row data set. The lines in the output take up a lot of space on the page. So instead of displaying it in its entirety, let's calculate some statistics about it.

```
summary(cars)
```

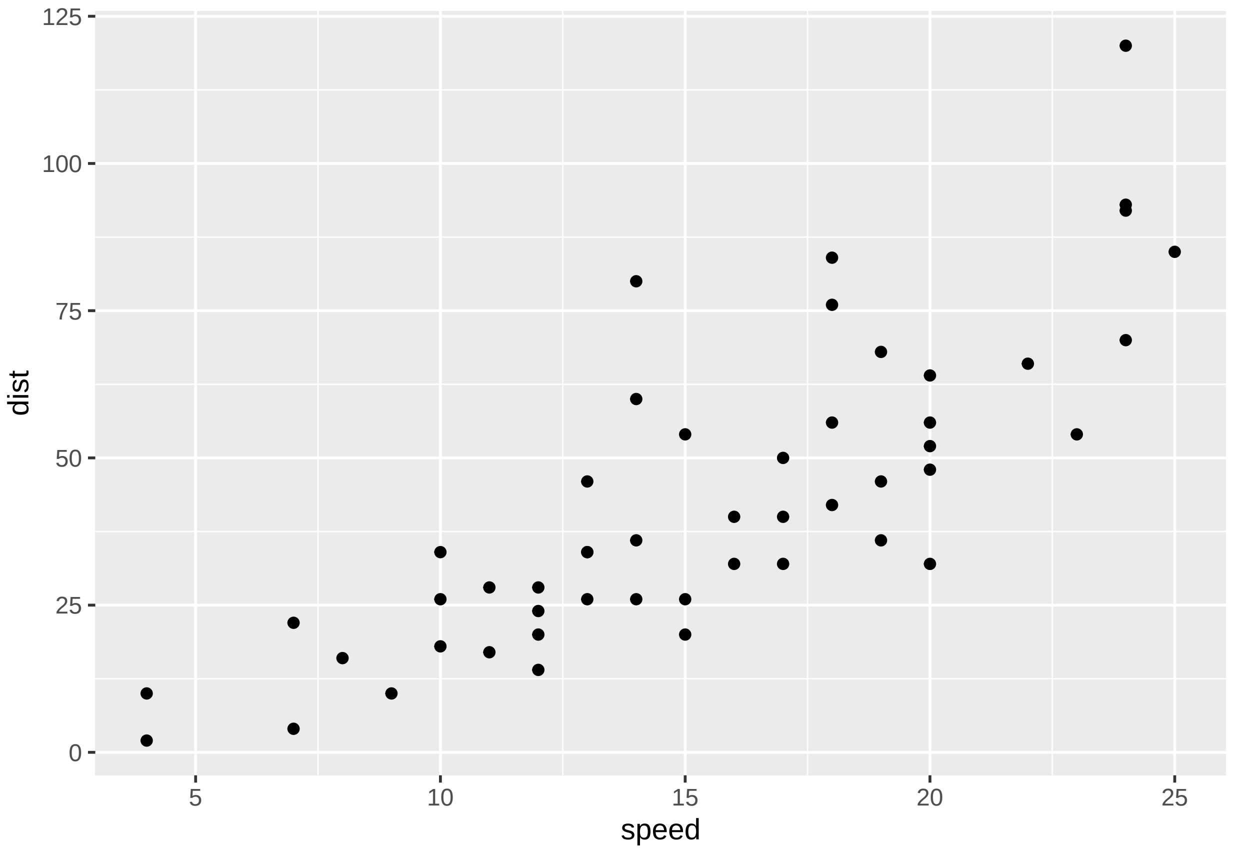
```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

Interesting. We found the *Five Numbers Summary* of the dataset `cars`. By the way, notice how I took advantage of this short paragraph to demonstrate some ~~pocibilities~~ possibilities of text formatting using R Markdown.

> É possível até mesmo escrever citações com a linguagem!
>
> Nunes, M.A. (2019)

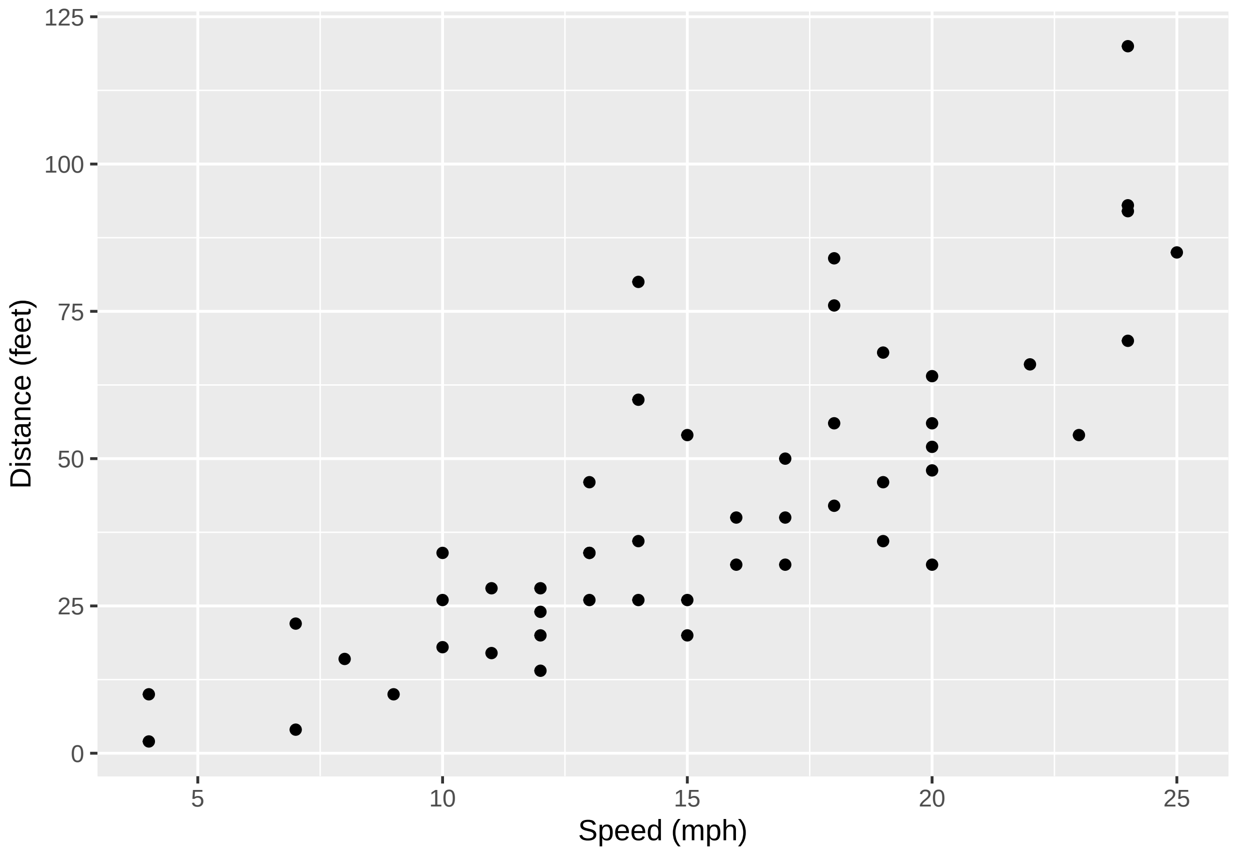
But let's go back to the analysis, because this is what matters here. The next logical step is to make a scatter plot.

```r
ggplot(cars, aes(x = speed, y = dist)) +
  geom_point()
```

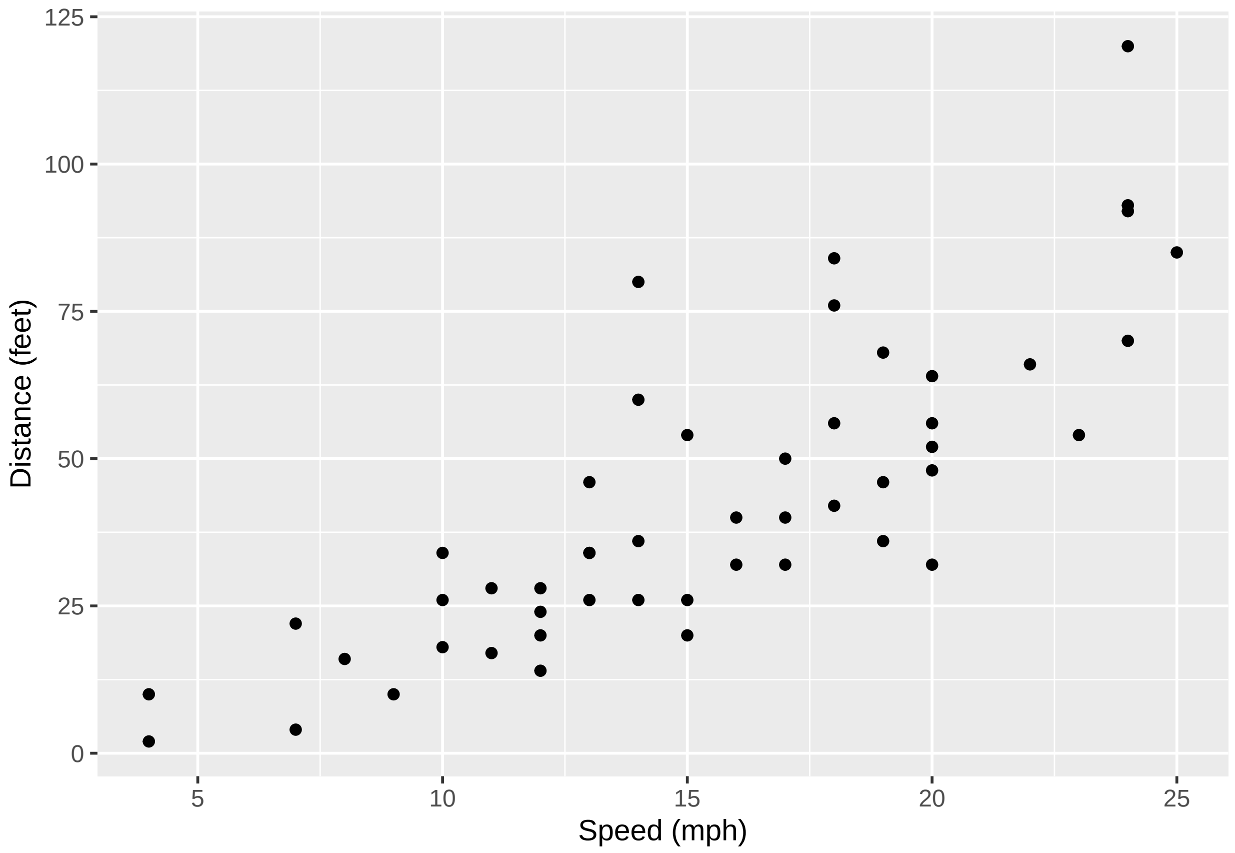


Note that the chunks reproduce exactly what the R code should reproduce. Therefore, we need to identify correctly the plot axes:

```r
ggplot(cars, aes(x = speed, y = dist)) +
  geom_point() +
  labs(x = "Speed (mph)", y = "Distance (feet)")
```



Note that this chunk displays both the code and the result of the plot. If we were writing a report for a client, the code part should be deleted. Fortunately, R Markdown allows us to put the option `echo = FALSE` in the chunk definition, so only the plot is shown:



If we set the `eval = FALSE` option, only the code is displayed, without the plot being displayed:

```r
ggplot(cars, aes(x = speed, y = dist)) +
  geom_point() +
  labs(x = "Speed (mph)", y = "Distance (feet)")
```

If we want, we can put a caption on the picture. This figure can even be referenced later:

Here's how I can reference Figure 1 as if I were using LaTeX. Note that I must put `fig:` as a reference to call the figure correctly. This should be done for **all** figures in the text.

I can even do it again in another color, to show the numbering is updated as in traditional LaTeX:

Therefore, Figure 2 shows what was promised in the previous paragraph. In addition to graphics, we can fit models to our data. Let's do a linear regression on the `cars` dataset:

```r
fit <- lm(dist ~ speed, data = cars)
summary(fit)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
```

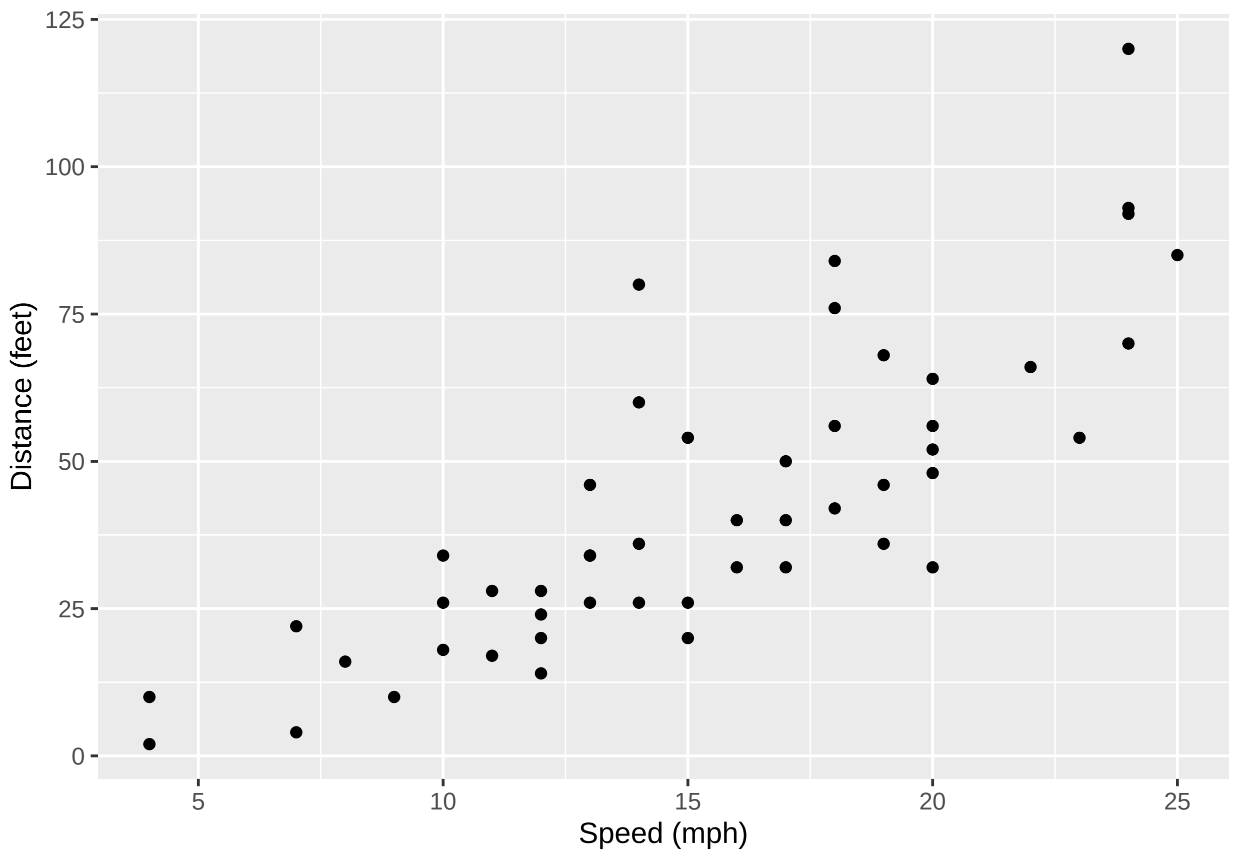


Figure 1: Scatter plot between the distance required to completely stop a car (in feet) and its speed (in miles per hour).

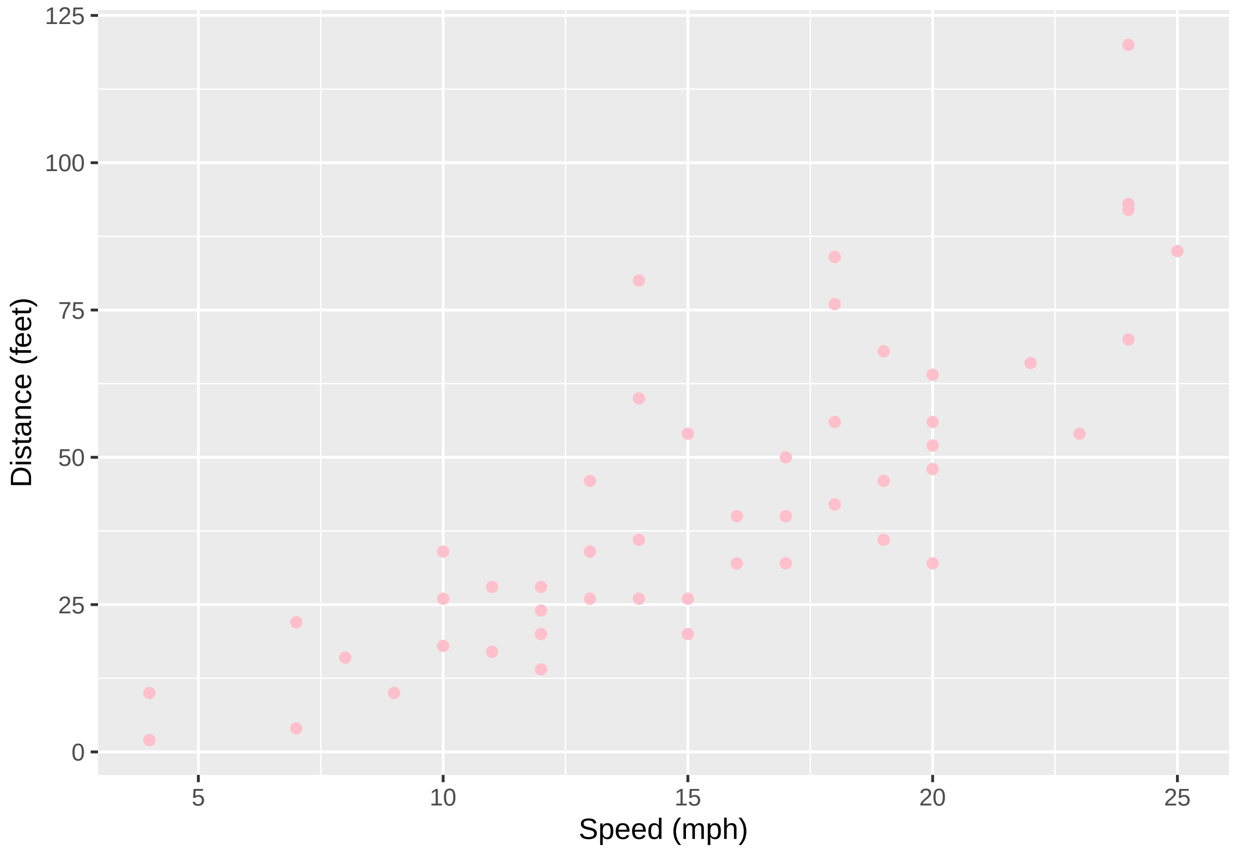


Figure 2: Scatter plot with pink dots between the distance required to completely brake a car (in feet) and its speed (in miles per hour).

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

This output would be interesting to us statisticians. But what about the average user? How can I get him to see the result of the adjustment we made without using the standard output of R? First, we need to identify what part of the object `fit` has the information we want:

```
summary(fit)$coefficients
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -17.579095  6.7584402 -2.601058 1.231882e-02
## speed         3.932409  0.4155128  9.463990 1.489836e-12
```

Now, let's use the `knitr` package and its `kable` function to get a better table with our results:

```
library(knitr)
kable(summary(fit)$coefficients,
      format = "latex")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-17.579095	6.7584402	-2.601058	0.0123188
speed	3.932409	0.4155128	9.463990	0.0000000

Notice that we can make a table, but it's ugly. This table

1. is too close to the text
2. is off center
3. has no title
4. has too many digits
5. has too many lines
6. but look on the bright side: at least we learned to make a numbered list in R Markdown.

We can fix all of these problems in a fairly reasonable way. Library `kableExtra` comes to help us:

```
library(kableExtra)
```

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

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

Table 1: Regression model result when applied to data ‘cars‘.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-17.5791	6.7584	-2.6011	0.0123
speed	3.9324	0.4155	9.4640	0.0000

Table 2: Regression model result when applied to data ‘cars‘.

	Estimate	Std. Error	t value	Pr(>|t|)
Intercept	-17.5791	6.7584	-2.6011	0.0123
Speed	3.9324	0.4155	9.4640	0.0000

```r
kable(summary(fit)$coefficients,
      format = "latex",
      booktabs = TRUE,
      caption = "Regression model result when applied to data `cars`.",
      digits = 4) %>%
  kable_styling(position = "center")
```

Suppose we do not want the rows to be identified as *(Intercept)* and *speed*. Let’s say we want them to be identified as *Intercept* (without paranthesis) and *Speed* (capitalized).

```r
fit.results <- as.data.frame(summary(fit)$coefficients)
row.names(fit.results) <- c("Intercept", "Speed")
kable(fit.results,
      format = "latex",
      booktabs = TRUE,
      caption = "Regression model result when applied to data `cars`.",
      digits = 4) %>%
  kable_styling(position = "center")
```

Now Table 2 looks exactly as we thought it would. Note that, similar to what we did with the figures, we must put `tab:` as a reference to call the table. This should also be done for **all** tables in the text.

Something similar can be done to the columns of any data frame. Notice how the column names have changed in 3.

```r
names(fit.results) <- c("Estimated Value", "Standard Error", "Statistic", "p-value")
kable(fit.results,
      format = "latex",
      booktabs = TRUE,
      caption = "Regression model result when applied to data `cars`.",
      digits = 4) %>%
  kable_styling(position = "center")
```

Now we can make the final plot of our analysis by plotting the data together with the fitted line. This result is found in Figure 3.

Table 3: Regression model result when applied to data ‘cars‘.

	Estimated Value	Standard Error	Statistic	p-value
Intercept	-17.5791	6.7584	-2.6011	0.0123
Speed	3.9324	0.4155	9.4640	0.0000

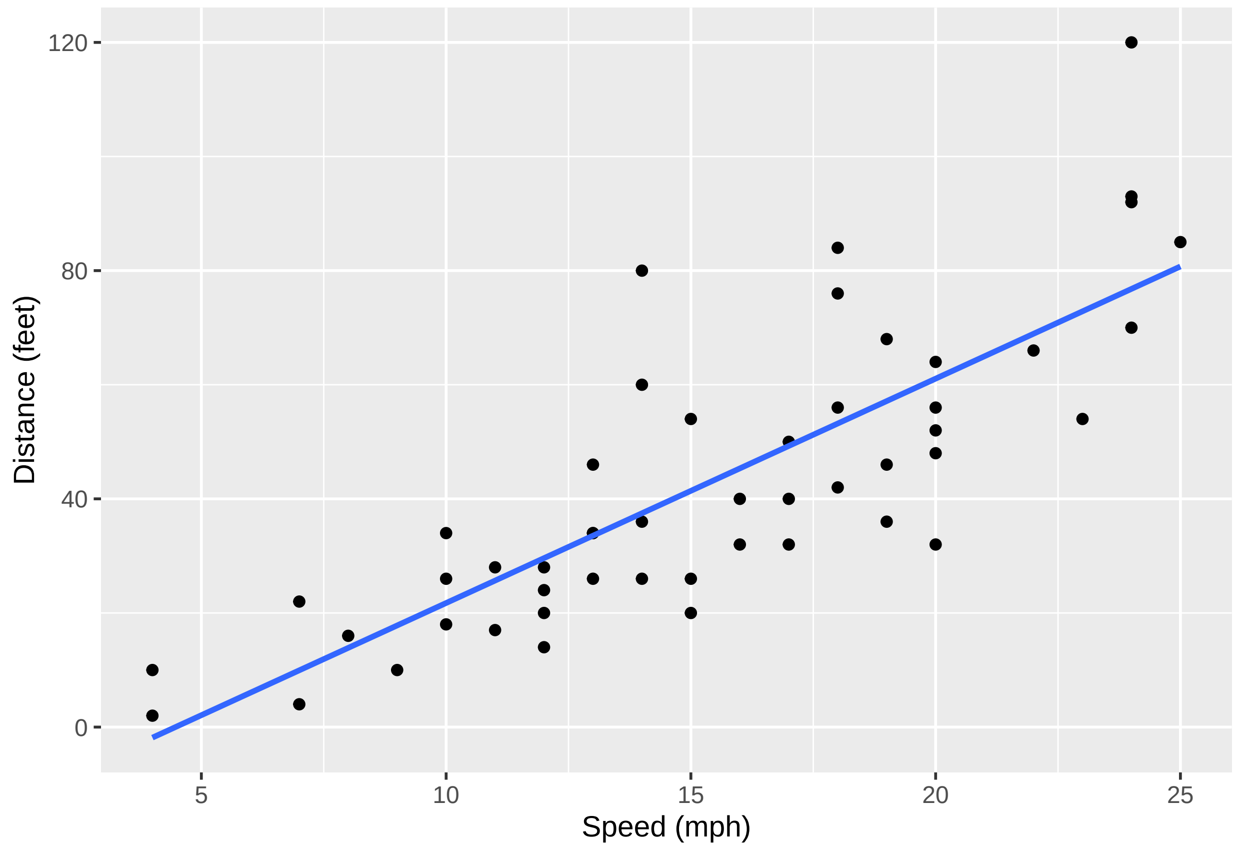


Figure 3: Linear regression fit between the distance required to completely brake a car (in feet) and its speed (in miles per hour).

```r
ggplot(cars, aes(x = speed, y = dist)) +
  geom_point() +
  labs(x = "Speed (mph)", y = "Distance (feet)") +
  geom_smooth(method = "lm", se = FALSE)
```

If we consider Figure 3 too small, we can change its dimensions, as shown in Figure 4.

```r
ggplot(cars, aes(x = speed, y = dist)) +
  geom_point() +
  labs(x = "Speed (mph)", y = "Distance (feet)") +
  geom_smooth(method = "lm", se = FALSE, colour = "black")
```

We only need to run the residuals anaysis to finish our report. Figure 5 shows how to do it in one line of code.

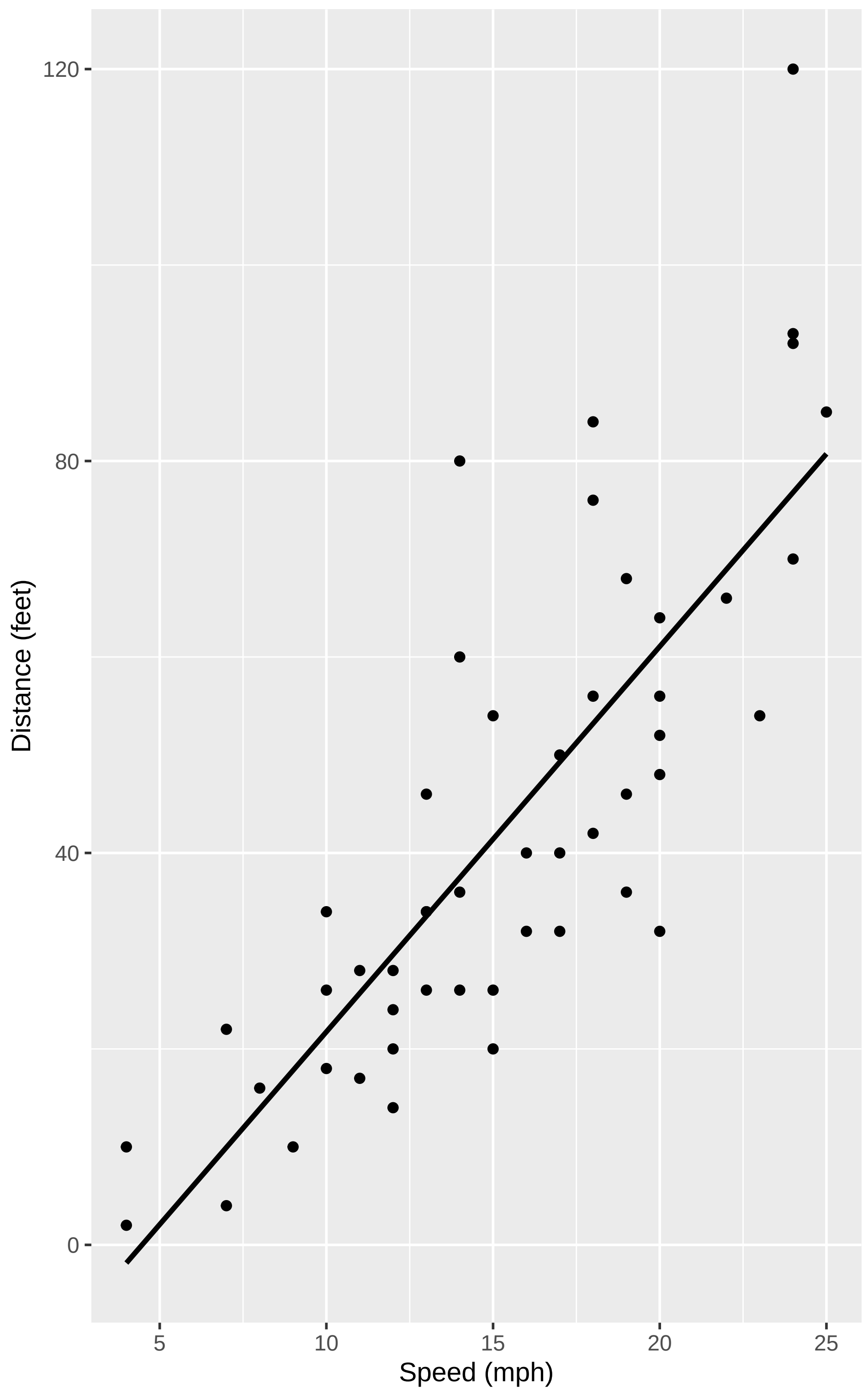


Figure 4: Linear regression fit between the distance required to completely brake a car (in feet) and its speed (in miles per hour).

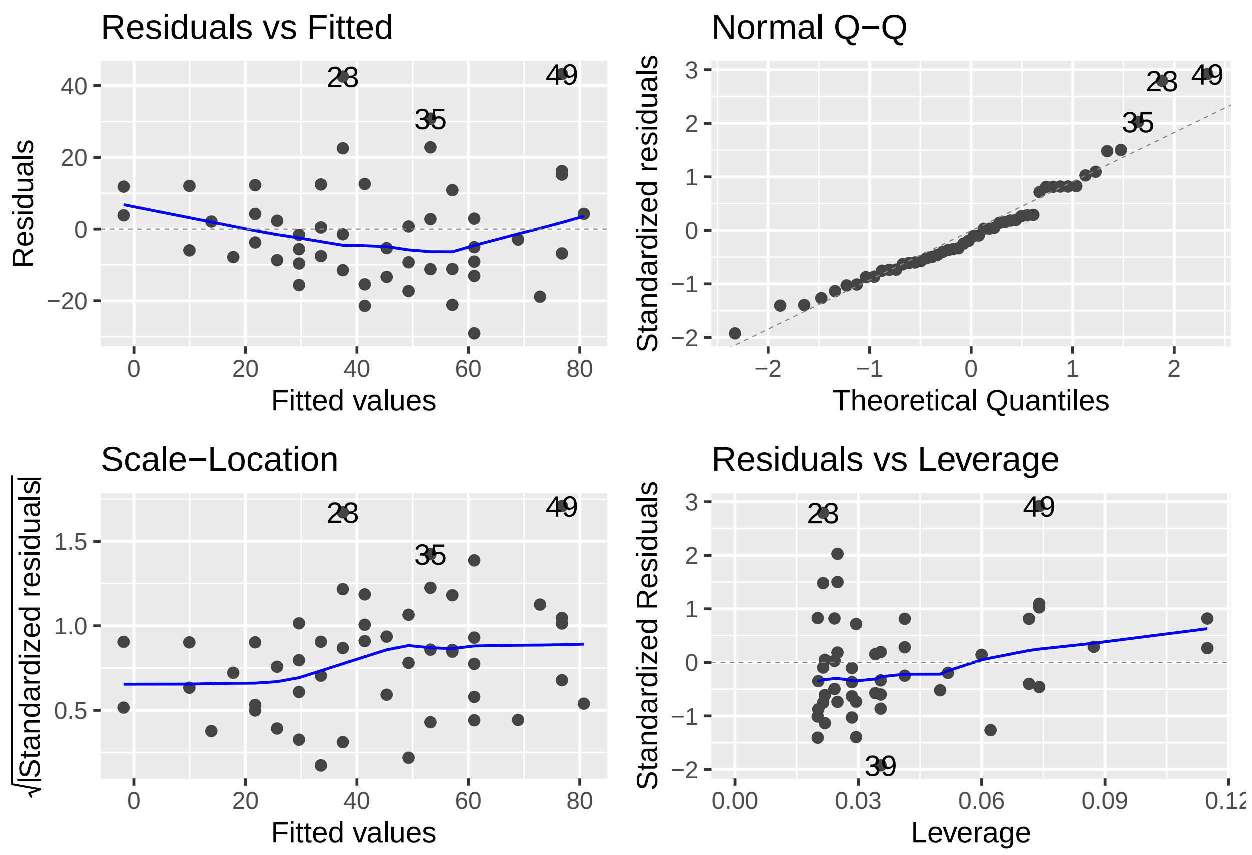


Figure 5: Residual analysis of our regression model.

```r
library(ggfortify)
autoplot(fit)
```

We can use R Markdown to mix R results inside text. For exemple, in this example, the regression intercept is -17.5790949 and its slope is 3.9324088. As any R code output, we can limit the number of digits very easily: -17.5791 and 3.9324

3 Conclusion

That's it. There are many other resources available in R Markdown, but we covered the basics in this tutorial.