

car_viz

Andrew Sisitzky

9/22/2021

The code begins by calling both the package ‘tidyverse’ and the built-in dataset ‘mtcars’

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --
## v ggplot2 3.3.3      v purrr  0.3.4
## v tibble  3.1.0      v dplyr  1.0.5
## v tidyr   1.1.3      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

# call built-in data mtcars.
data(mtcars)
```

We then create a subset of the ‘mtcars’ dataset of only the observations in which the variable mpg has a value over 20. After that, we trim this dataset further, selecting only certain columns to display in the new dataset ‘mtcars_mpg2’.

```
# Select only car models where mpg > 20
mtcars_mpg2 <- mtcars[mtcars$mpg > 20,]

# Reduce the variables to mpg, cyl, disp, hp, gears
mtcars_mpg2 <- mtcars_mpg2[, c(1,2,3,4,10)]
```

We then use the R function ‘source()’ to link this file with the ‘hand_functions.R’ file in order to utilize a function within that file.

The following chunk calls the function ‘sum_special’ and uses it to calculate the data summary statistics of the ‘mtcars_mpg2’ dataset. This function calculates the mean, var, cov, and cor of ‘mtcars_mpg2’

```
# read the R file hand_functions.R so that it can be used
# notice that with echo = TRUE
source(file = "hand_functions.R", echo = TRUE)

##
## > sum_special <- function(df_x) {
## +   try(if (!is.data.frame(df_x))
## +     stop("Input data must be a data frame."))
## +   sp_means <- apply(df_ .... [TRUNCATED]

# Now use the function from hand_functions.R
sp_out <- sum_special(mtcars_mpg2)
```

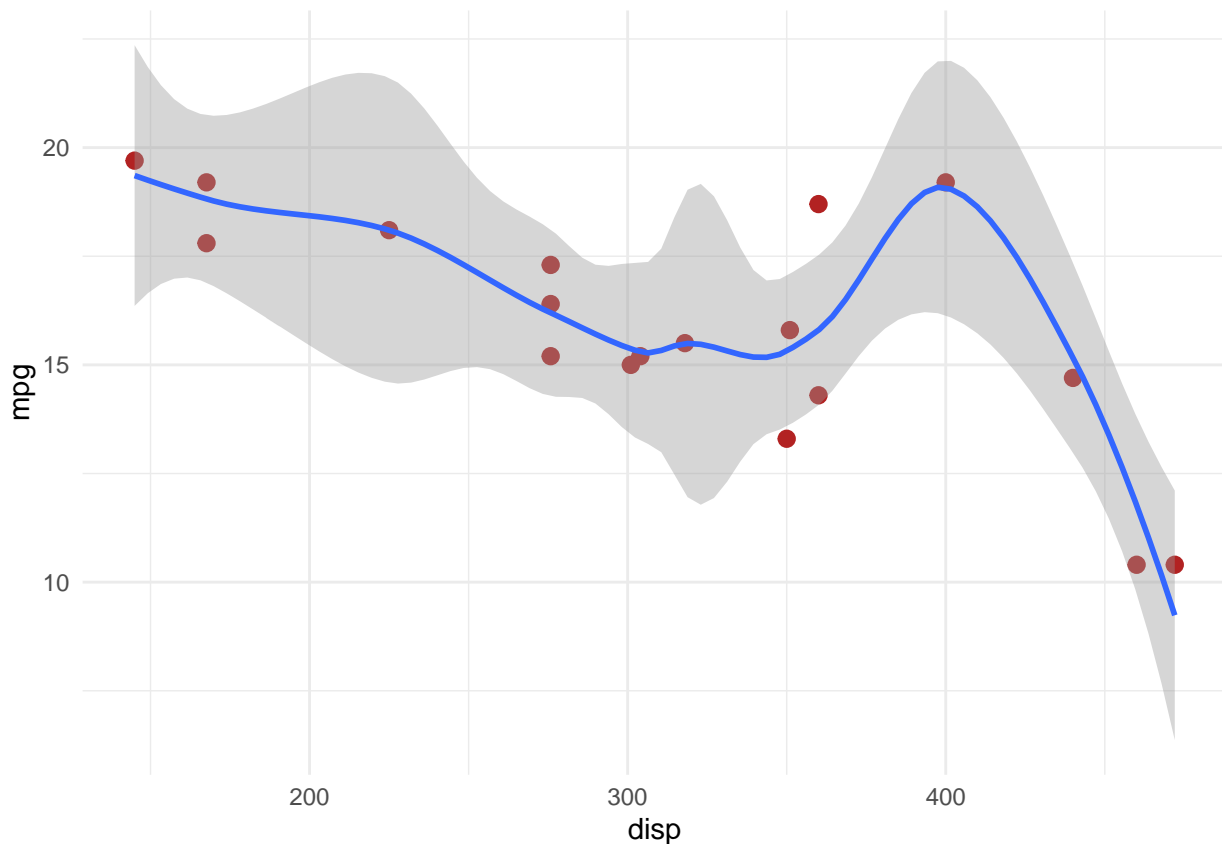
The next chunk of code shows the use of the `esquisser` function in the `esquisse` package, however this code was kept as a comment as to not allow the chunk to compile. If it were to run, its purpose would be to create a plot with `ggplot2`, as we will do in the next section of code.

```
# library(esquisse)  
# esquisser(data = mtcars_mpg2, viewer = "browser")
```

In the next section, we create a plot of the `mtcars_mpg2` dataset, where the variable `disp` is plotted on the x axis and `mpg` is plotted on the y axis. After the plot is created, this section of code uses `geom_point` to create a scatterplot of the data (where the points are bullet shaped, of size 4L, and of the color B22222) and `geom_smooth` is used to overlay the smoothed conditional means of the data. Finally, `theme_minimal()` sets the theme of the plot.

```
ggplot(mtcars_mpg2) +  
  aes(x = disp, y = mpg) +  
  geom_point(shape = "bullet", size = 4L, colour = "#B22222") +  
  geom_smooth(span = 0.5) +  
  theme_minimal()
```

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



```
# note that this boxplot cannot be made with esquisse() unless  
# the data is adjusted. What adjustment is needed?
```

In this final section of code, we use `ggplot` to once again plot data from the `mtcars_mpg2` dataset. This time, the data on the x axis is the `cyl` variable converted to a factor and the `mpg` variable on the y axis. This time, `geom_boxplot` is used to generate boxplots, as opposed to the scatterplot we saw in the previous plot.

```
ggplot(mtcars_mpg2, aes(x=as.factor(cyl), y=mpg)) +  
  geom_boxplot(fill="slateblue", alpha=0.2) +  
  xlab("cyl")
```

