

car_viz

call built-in data mtcars.

```
data(mtcars)
```

Select only car models where mpg<20

```
mtcars_mpg2 <- mtcars[mtcars$mpg < 20,]  
#use of $ to select the desired column
```

Reduce the variables to mpg, cyl, disp, hp, gears

```
mtcars_mpg2 <- mtcars_mpg2[, c(1,2,3,4,10)]  
#columns are vectors so we are calling the columns we want by  
#the number vector they are in the data frame
```

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)  
  
sum_special <- function(df_x){  
  
  ## sum_special calculates data summary statistics  
  ## the input param df_x is the data frame of input values  
  
  # browser() # browser() will start the debugger  
  # if the line is uncommented
```

```

## test the input data to assure that it is a data frame.
try(if(!is.data.frame(df_x)) stop("Input data must be a data frame.))

sp_means <- apply(df_x, MARGIN = 2, FUN = mean)
sp_var <- apply(df_x, MARGIN = 2, FUN = var)
sp_cov <- cov(df_x)
sp_cor <- cor(df_x)

## Note that defining a list with the
## syntax list(list_name = list_content) produces
## named list items
sp_outputs <- list(sp_means=sp_means,
                   sp_var = sp_var,
                   sp_cov = sp_cov,
                   sp_cor = sp_cor)

return(sp_outputs)
}

```

```

#call the sp_out list

```

```

sp_out

```

```

## $sp_means
##      mpg      cyl      disp      hp      gear
## 15.900000  7.555556 313.811111 191.944444  3.444444
##
## $sp_var
##      mpg      cyl      disp      hp      gear
##  7.5258824  0.7320261 9438.7645752 3253.5849673  0.6143791
##
## $sp_cov
##      mpg      cyl      disp      hp      gear
## mpg      7.5258824 -1.3176471 -188.79529 -75.81176  0.6352941
## cyl     -1.3176471  0.7320261  64.71111  28.44444 -0.2614379
## disp   -188.7952941 64.7111111 9438.76458 2679.60065 -34.1934641
## hp      -75.8117647 28.4444444 2679.60065 3253.58497 15.2026144
## gear     0.6352941 -0.2614379 -34.19346  15.20261  0.6143791
##
## $sp_cor
##      mpg      cyl      disp      hp      gear
## mpg      1.0000000 -0.5613802 -0.7083614 -0.4844811  0.2954459
## cyl     -0.5613802  1.0000000  0.7784989  0.5828450 -0.3898406
## disp   -0.7083614  0.7784989  1.0000000  0.4835389 -0.4490217
## hp      -0.4844811  0.5828450  0.4835389  1.0000000  0.3400314
## gear     0.2954459 -0.3898406 -0.4490217  0.3400314  1.0000000

```

```

#This shows you the mean, variance, covariance, and
#correlation between the variables mpg, cyl, disp, hp, and #gear.

```

```

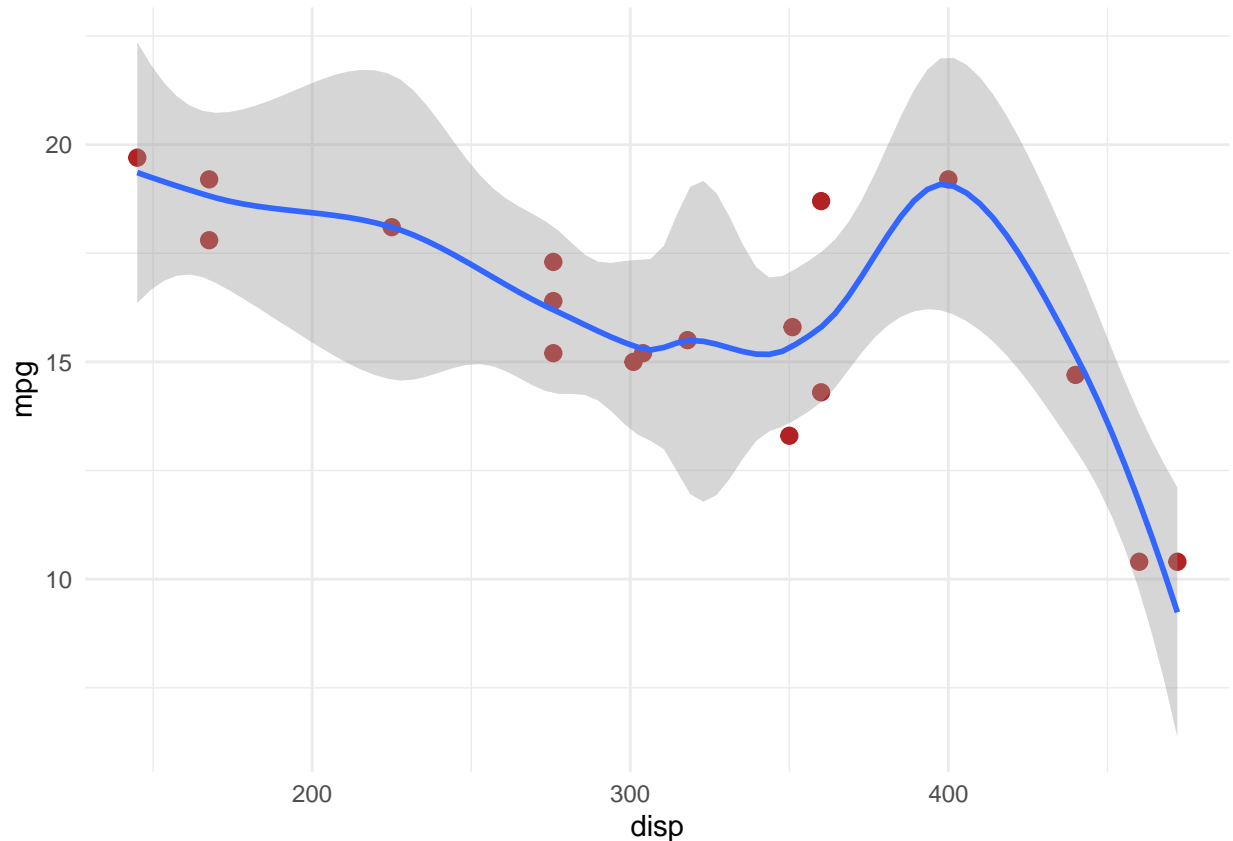
# library(esquisse)
#
# esquisser(data = mtcars_mpg2, viewer = "browser")

```

*#Using esquisser allows us to easily manipulate the plot
#without having to worry about the code. Once we have the #desired plot we can copy the ggplot code into*

```
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?

We need to use cylinders as the x input rather than disp, and used the data set `mtcars_mpg2` which is a subset of `mtcars` that only includes car models where `mpg < 20`.

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

