CODING WITH R

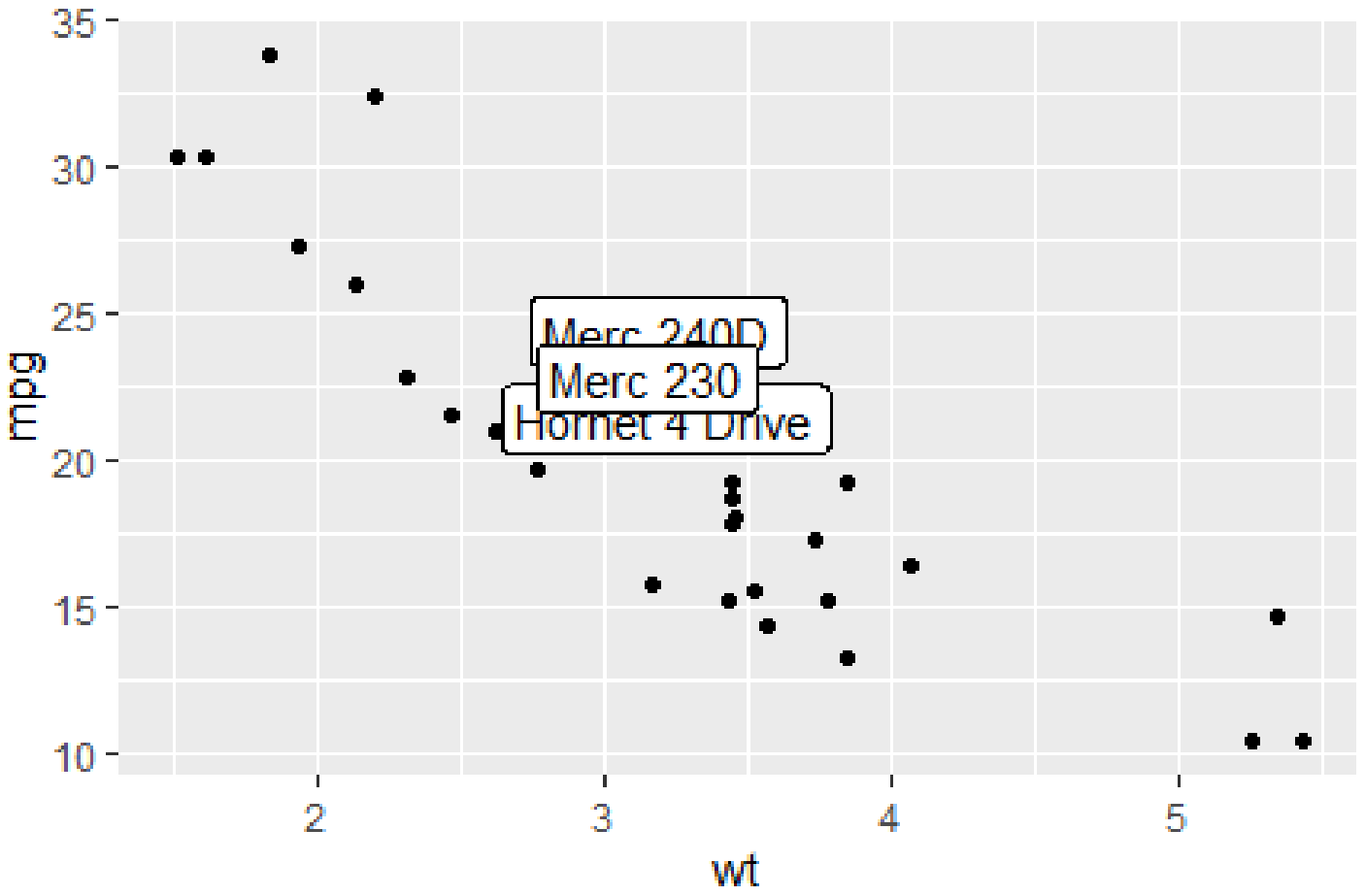
Niladri Chakraborty
University of the Free State

VISUALIZING DATA

```
# We want to annotate only cars with mpg > 20 and wt > 3.

# First we change data rownames as a real column called 'carName'

library(tidyverse)

str(mtcars)

data = as.data.frame(mtcars)

data = data%>%rownames_to_column(var="carName")

Or,

rownames_to_column(mtcars, var = "carName")
```

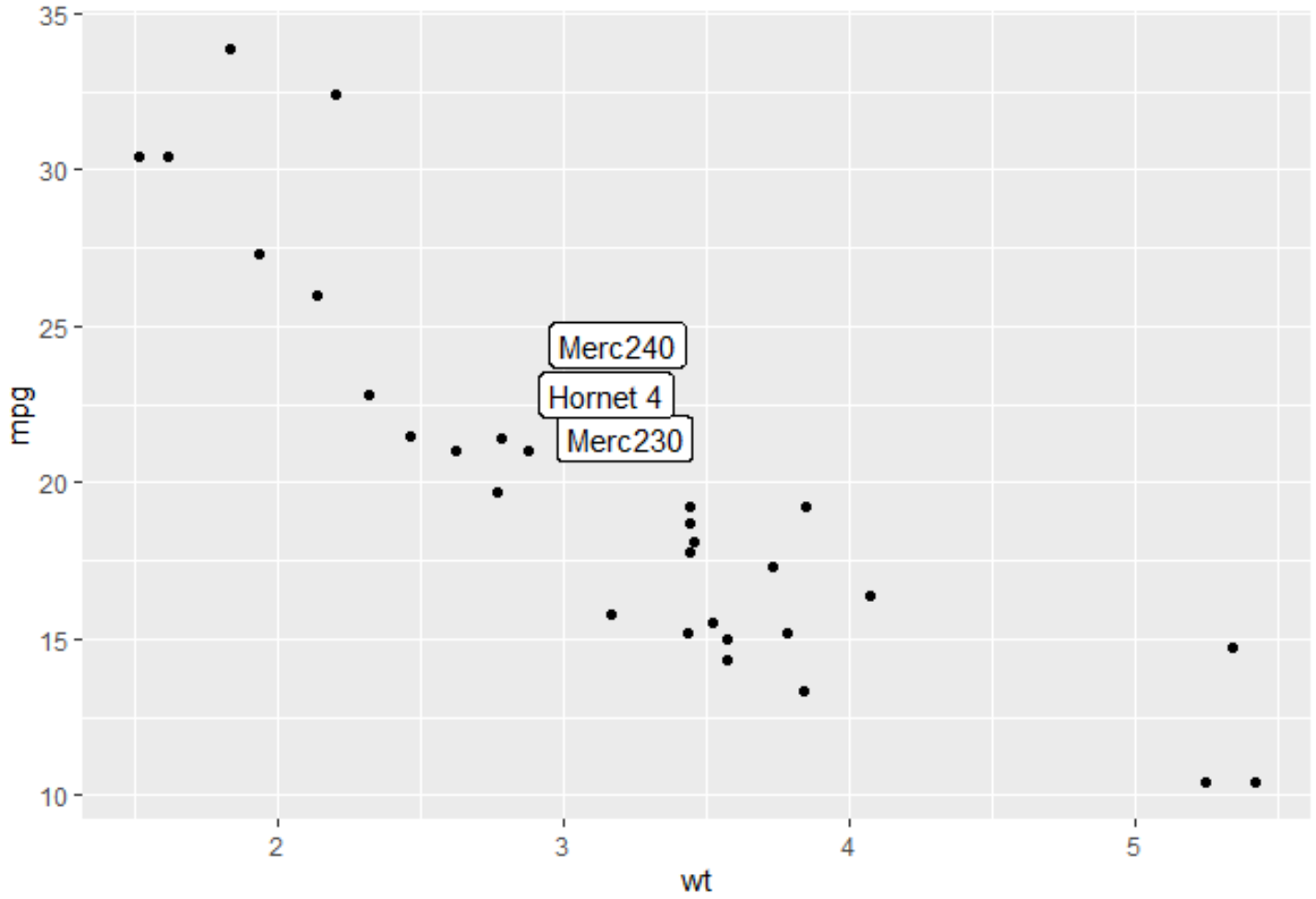
VISUALIZING DATA

```
# Plot

ggplot(data, aes(x=wt, y=mpg)) +

  geom_point() +

  geom_label(

    data=data %>% filter(mpg>20
& wt>3),

    aes(label=carName)

  )
```



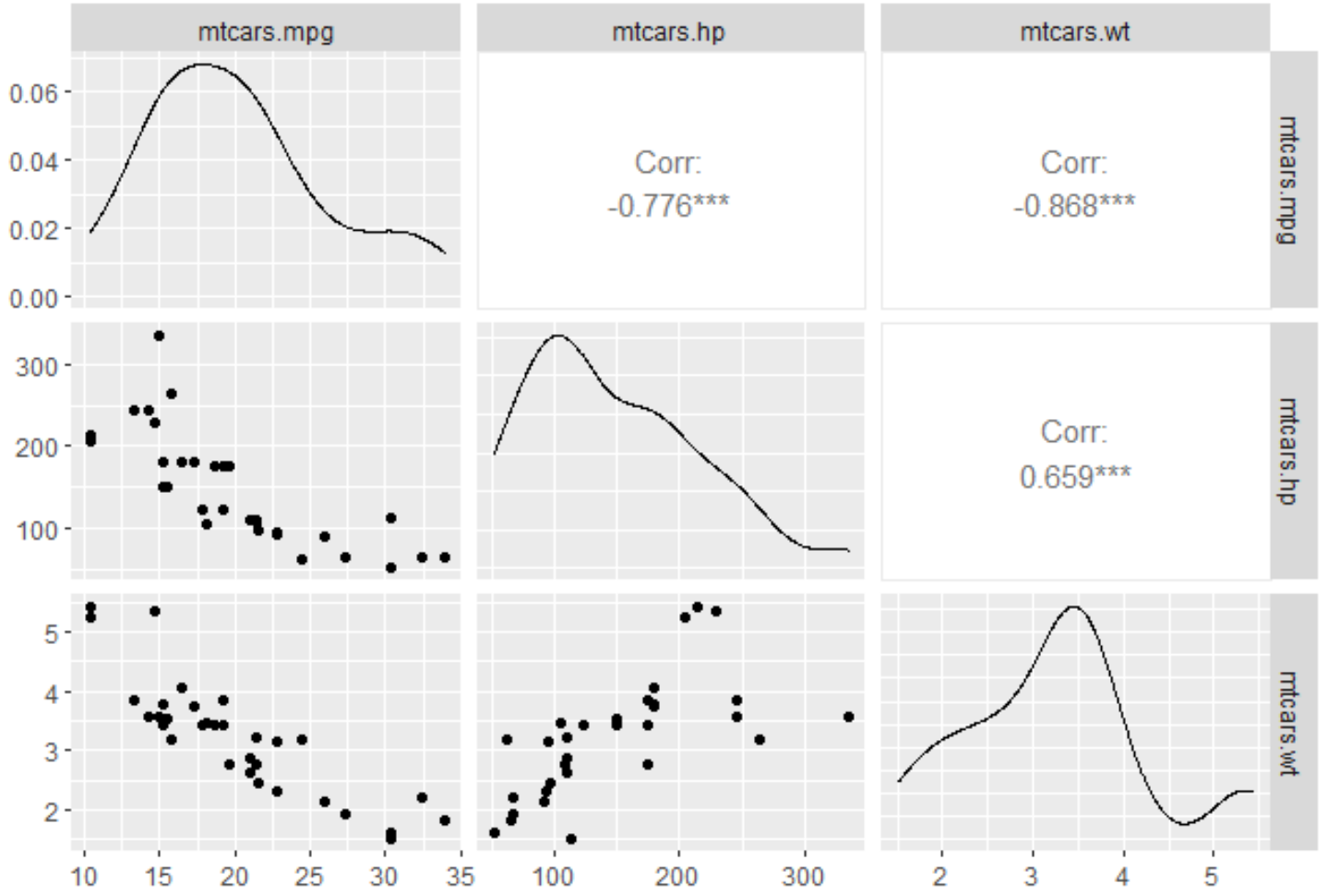
VISUALIZING DATA

```
ggplot(data, aes(x=wt, y=mpg)) +

  geom_point() +

  geom_label(

    data= filter(mtcars,mpg>20 &
wt>3), aes(label =
c("Merc230D","Merc240","Hornet
4"))

  )
```



CORRELATION PLOT

```
library(GGally)

data1 =
data.frame(mtcars$mpg,mtcars$hp,
mtcars$wt)

ggpairs(data1)
```

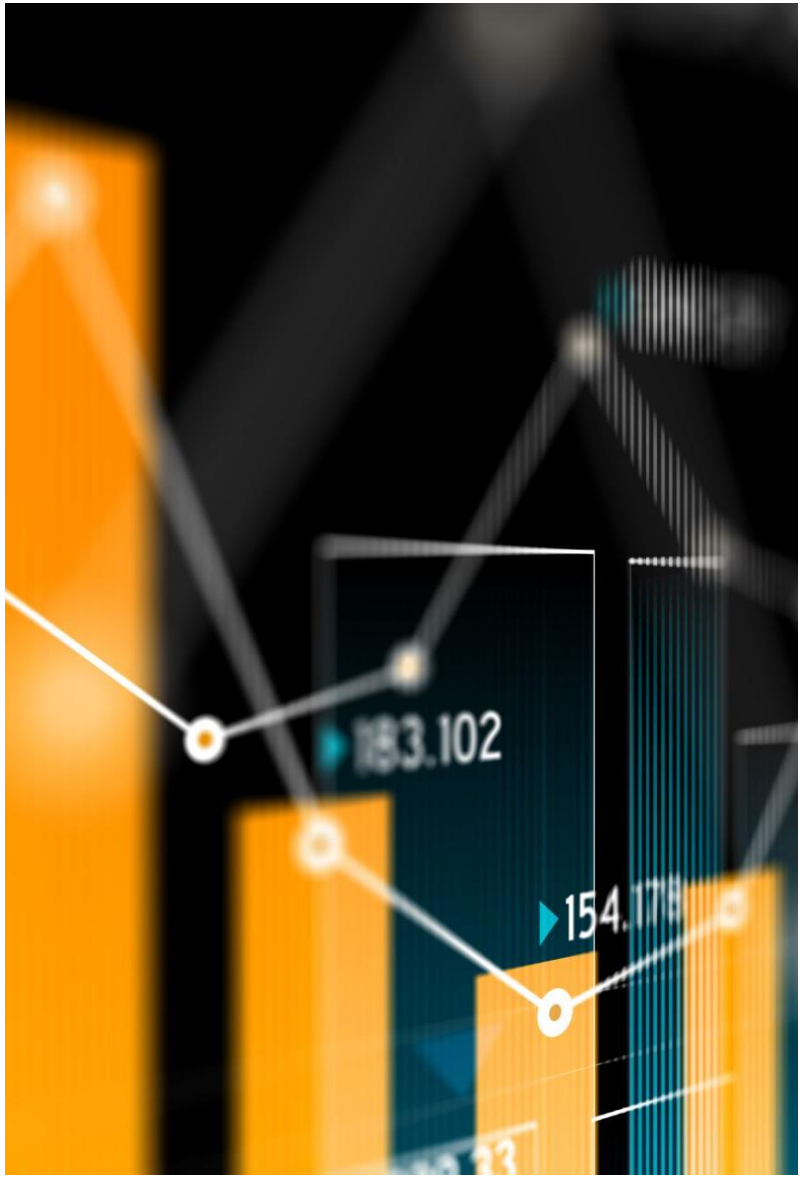


VISUALIZING DATA

Package ggrepel provides geoms for ggplot2 to repel overlapping text labels.

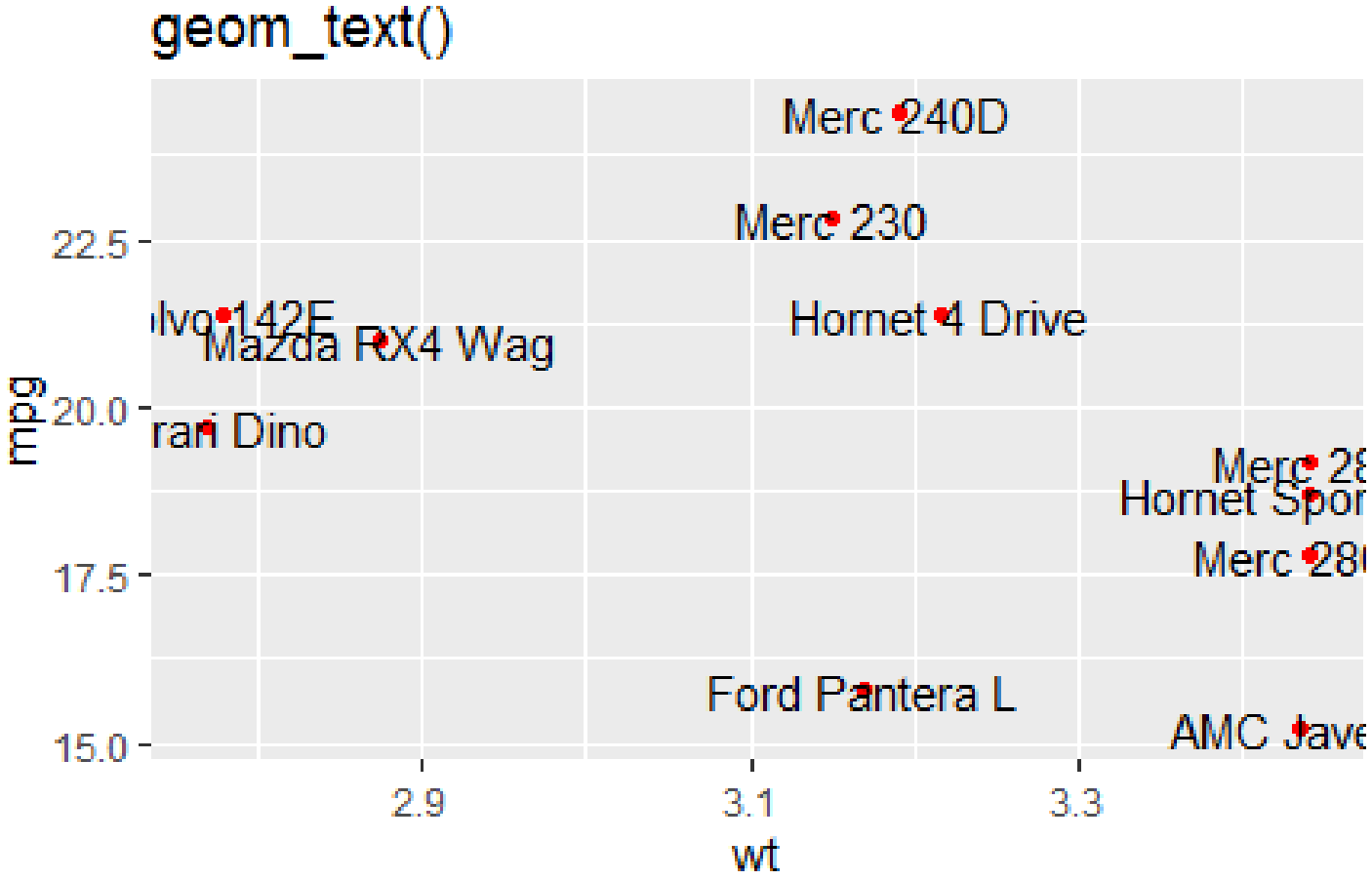
Let's compare geom_text() and geom_text_repel()

```
library(ggrepel)

dat = subset(mtcars, wt > 2.75 & wt < 3.45)

dat$car = rownames(dat)  ## store the row names to a variable

p = ggplot(dat, aes(wt, mpg, label = car)) +
geom_point(color = "red")

p
```



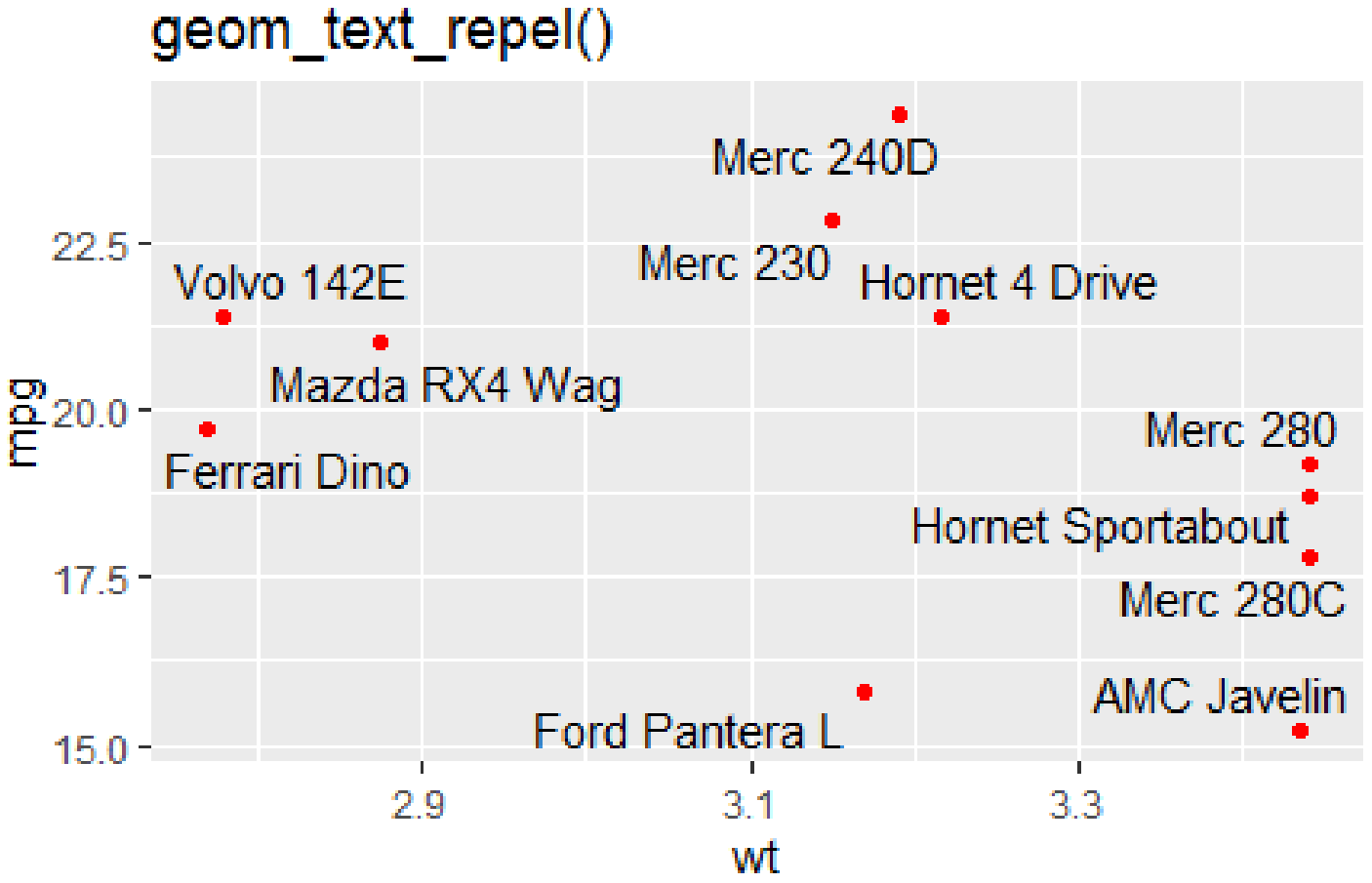
VISUALIZING DATA

```
p1 = p + geom_text() +
labs(title = "geom_text()")

p1
```



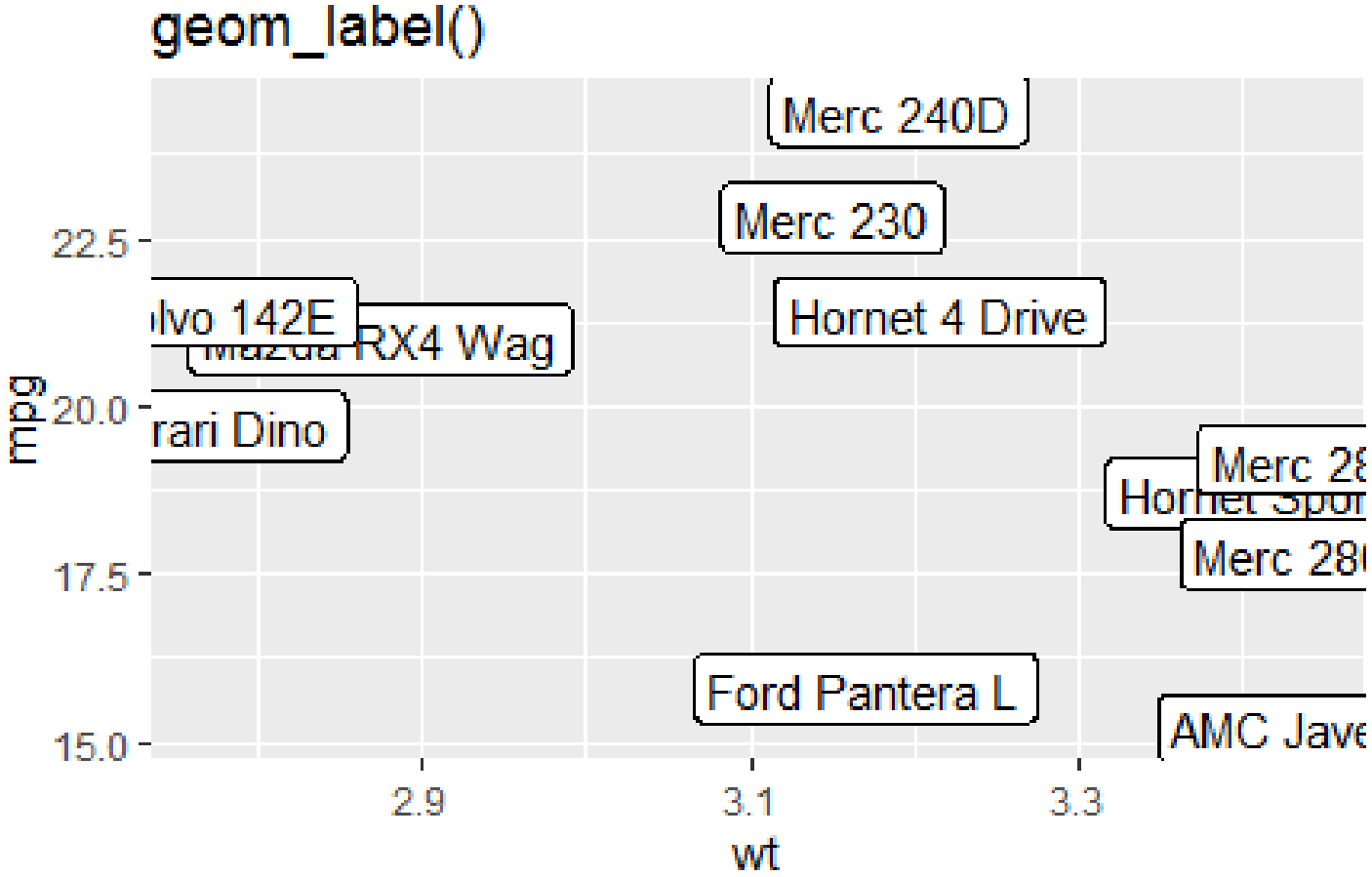
VISUALIZING DATA

```
p2 = p + geom_text_repel() +
labs(title = "geom_text_repel()")

p2
```



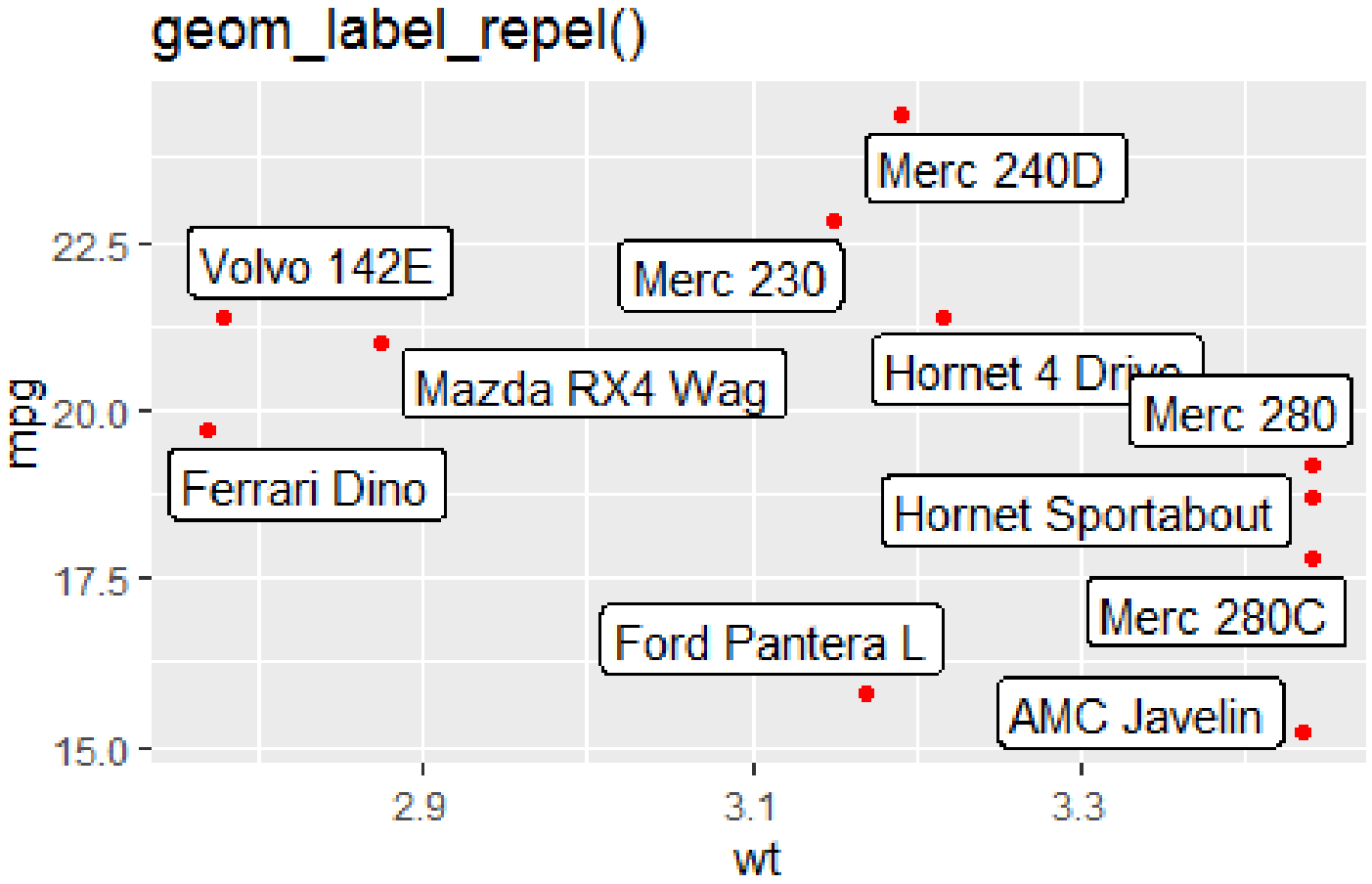
VISUALIZING DATA

```
p1 = p + geom_label() +
labs(title = "geom_label()")

p1
```



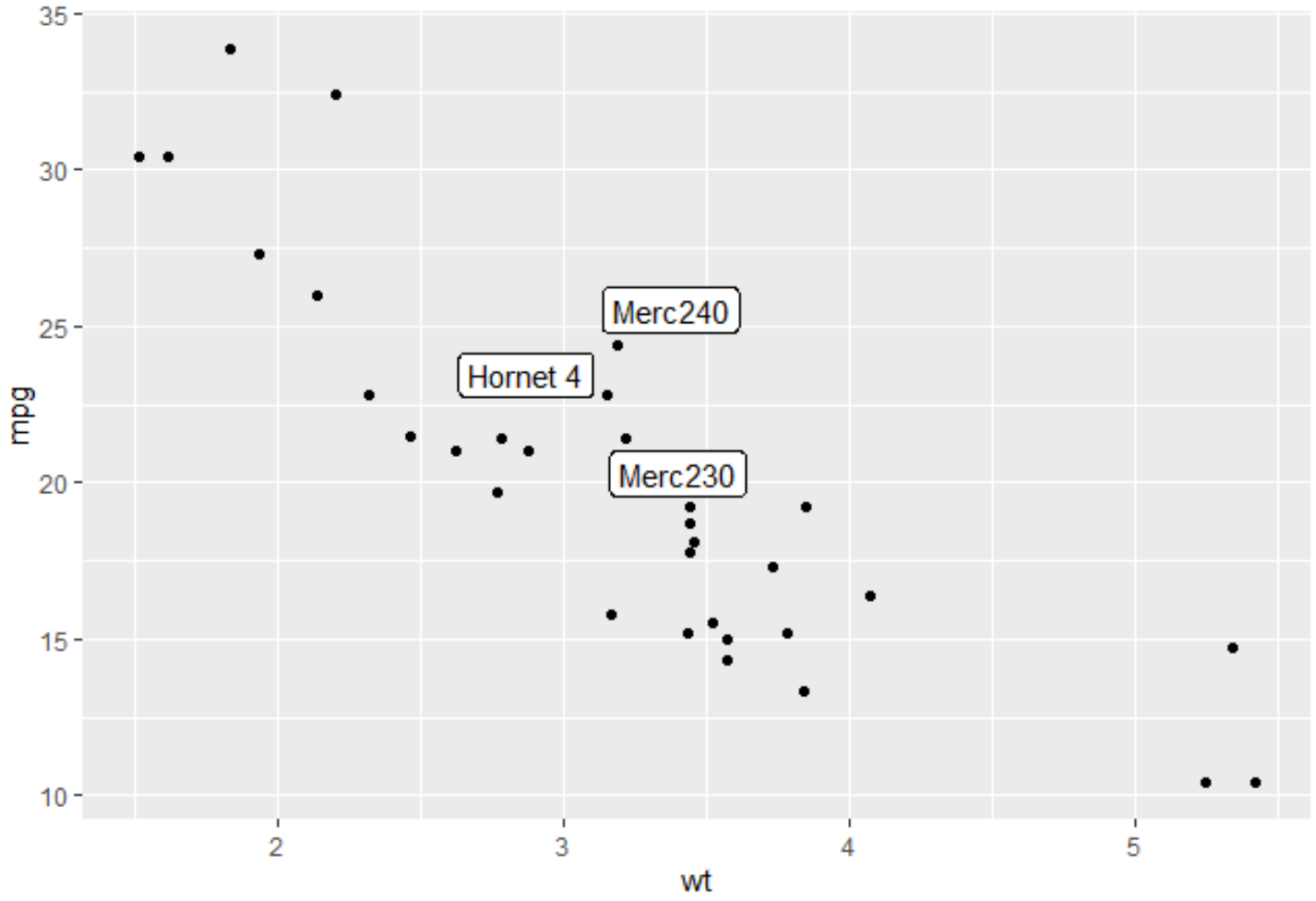
VISUALIZING DATA

```
p2 = p + geom_label_repel() +
labs(title = "geom_label_repel()")

p2
```



VISUALIZING DATA

```
ggplot(data, aes(x=wt, y=mpg)) +

  geom_point() +

  geom_label_repel(

    data= filter(mtcars,mpg>20 &
wt>3), aes(label =
c("Merc230","Merc240","Hornet
4"))

  )
```



VISUALIZING DATA

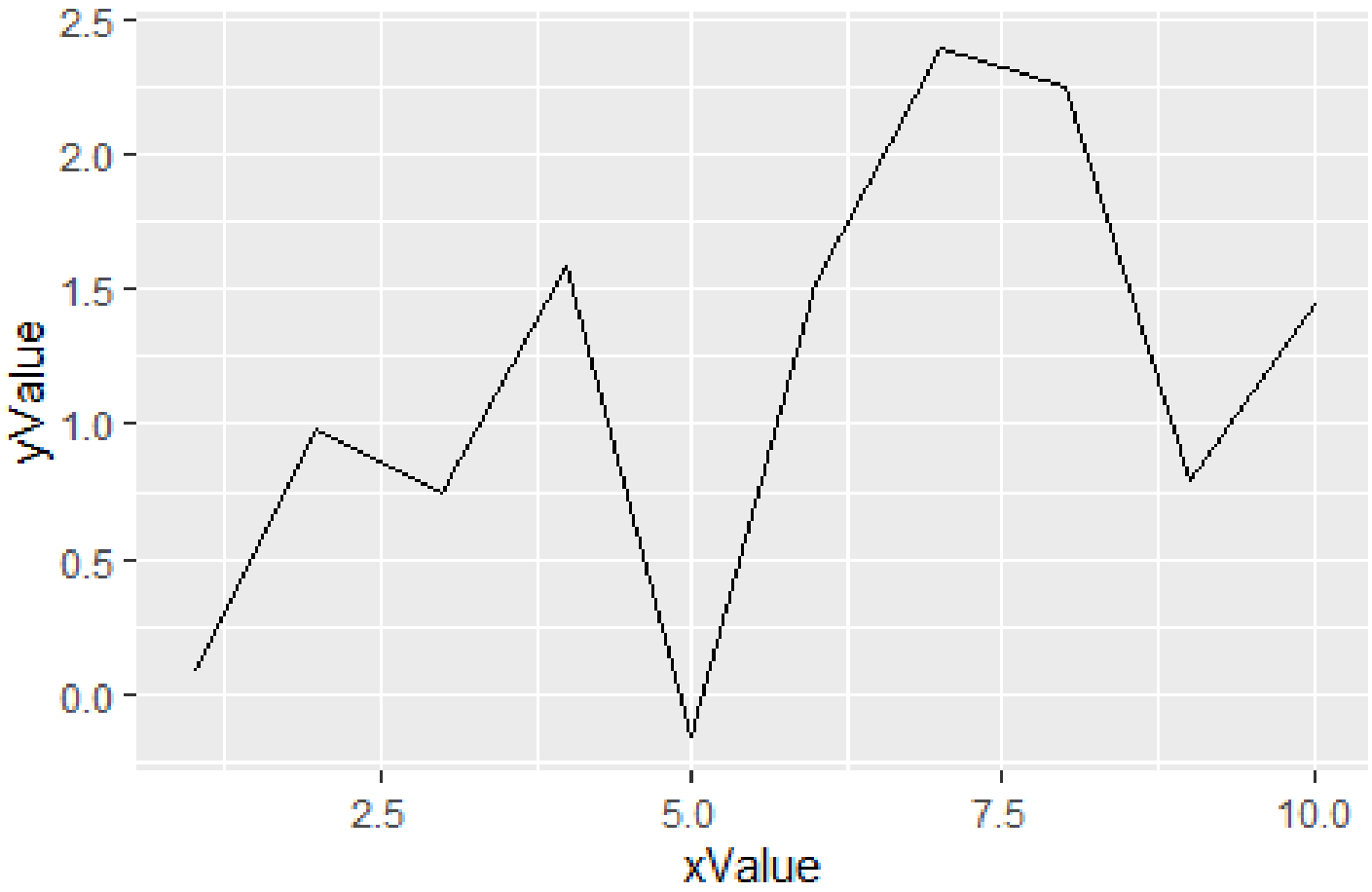
Creating line plot with ggplot()

The input data frame requires at least 2 columns:

- An *ordered* numeric variable for the X axis
- Another numeric variable for the Y axis

VISUALIZING DATA

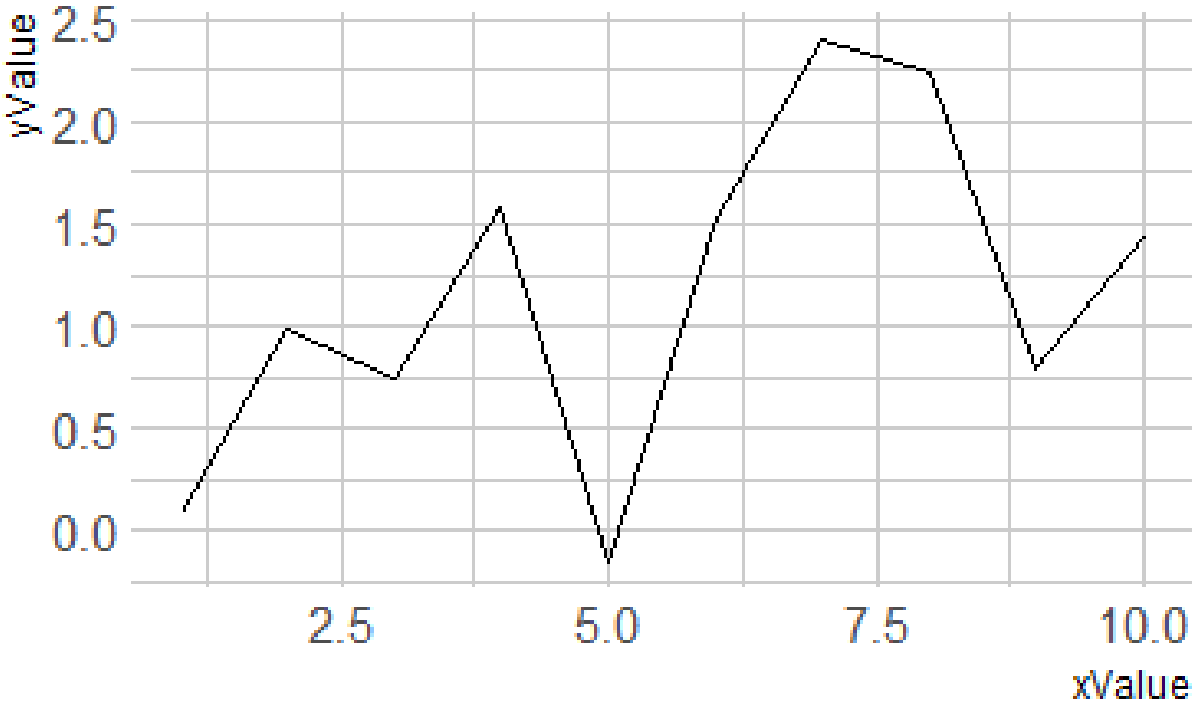
```
# create data

xValue = c(1:10)

yValue = cumsum(rnorm(10)) ## returns
cumulative sum of random samples

data = data.frame(xValue,yValue)


# Plot

p = ggplot(data, aes(x=xValue, y=yValue)) +

  geom_line()

p
```



VISUALIZING DATA

library(hrbrthemes)

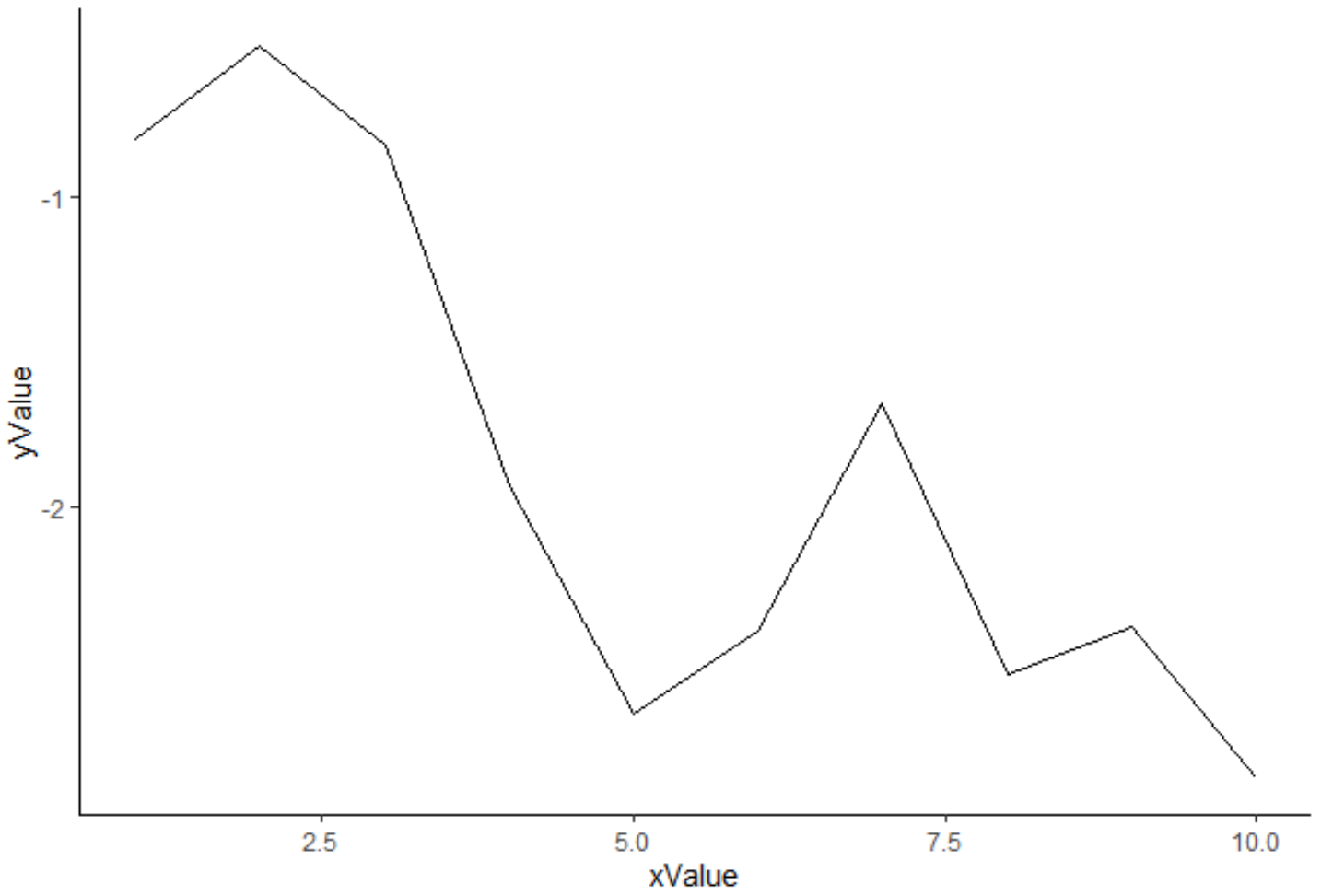
p+theme_ipsum()



VISUALIZING DATA

library(hrbrthemes)

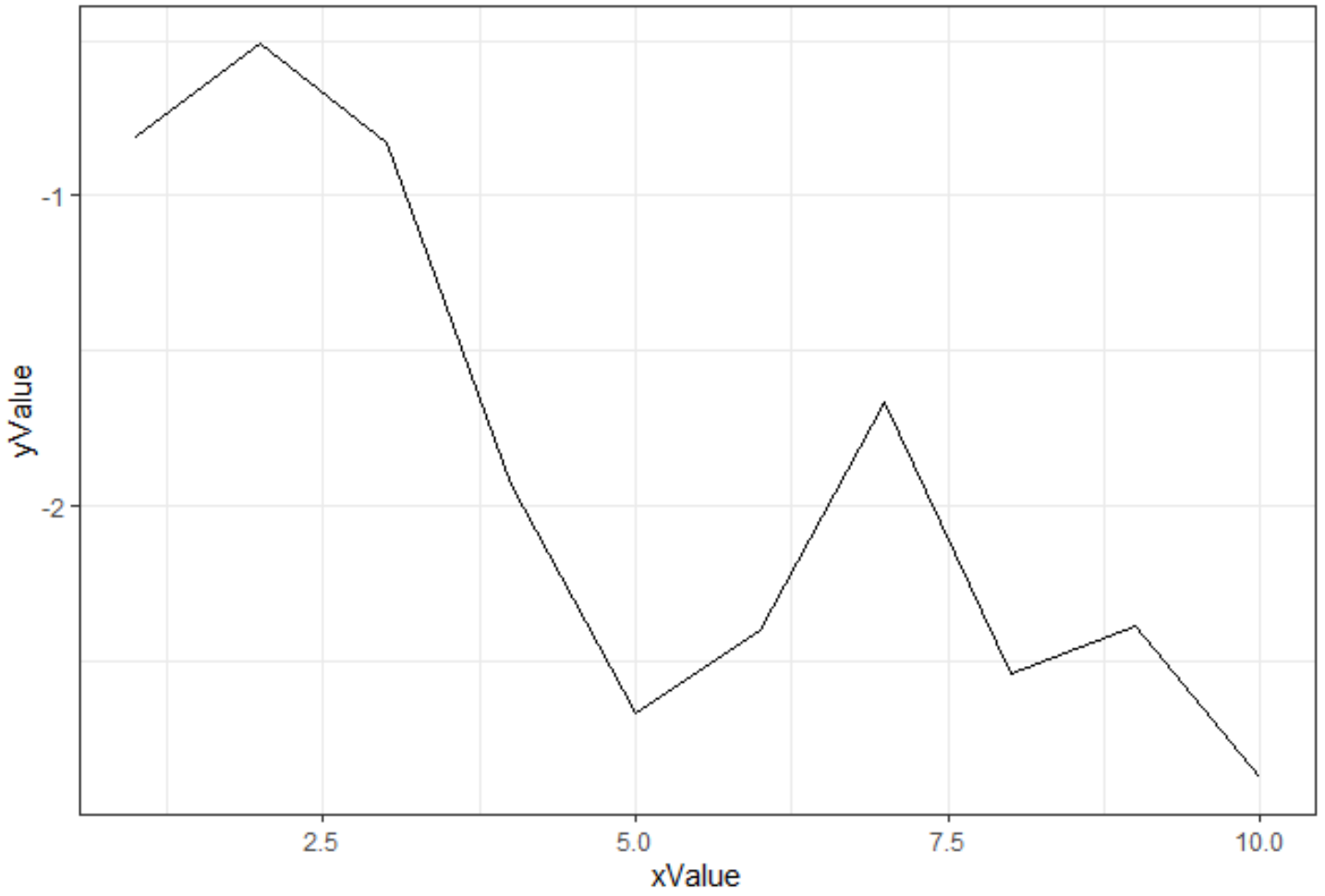
p+theme_classic()



VISUALIZING DATA

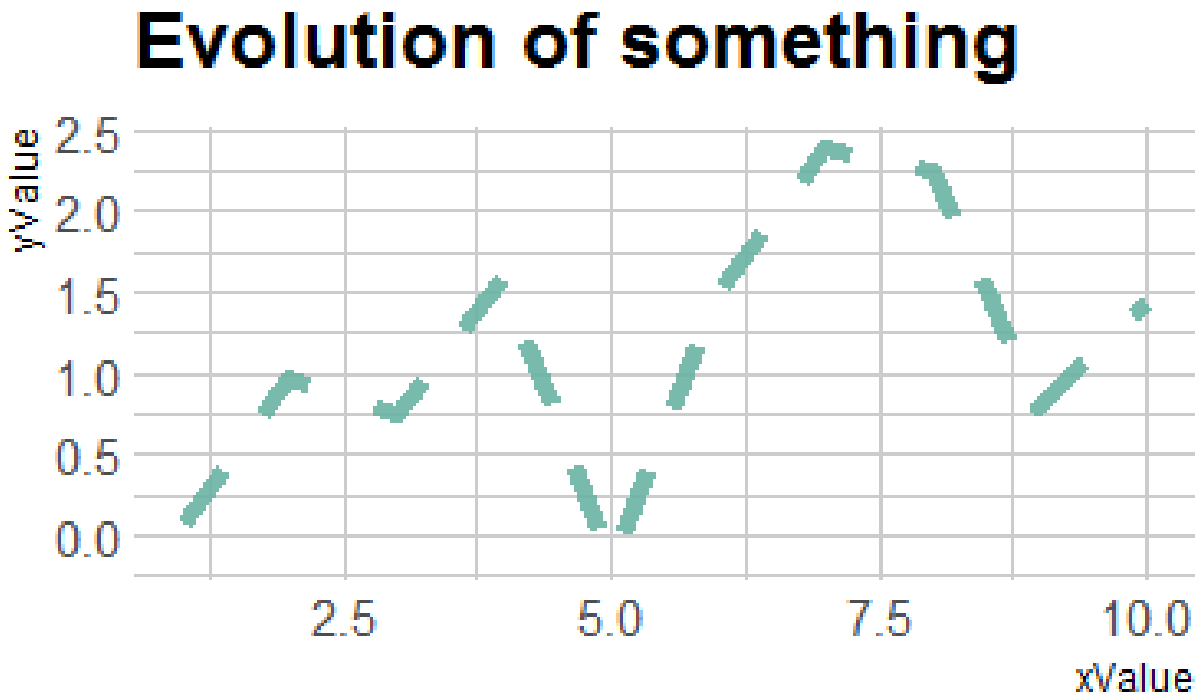
library(hrbrthemes)

p+theme_classic()



VISUALIZING DATA

```
# Customize the plot



# Plot

ggplot(data, aes(x=xValue,
y=yValue)) +

  geom_line( color="#69b3a2",
size=2, alpha=0.9, linetype=2) +

  theme_ipsum() +

  ggtitle("Evolution of something")
```

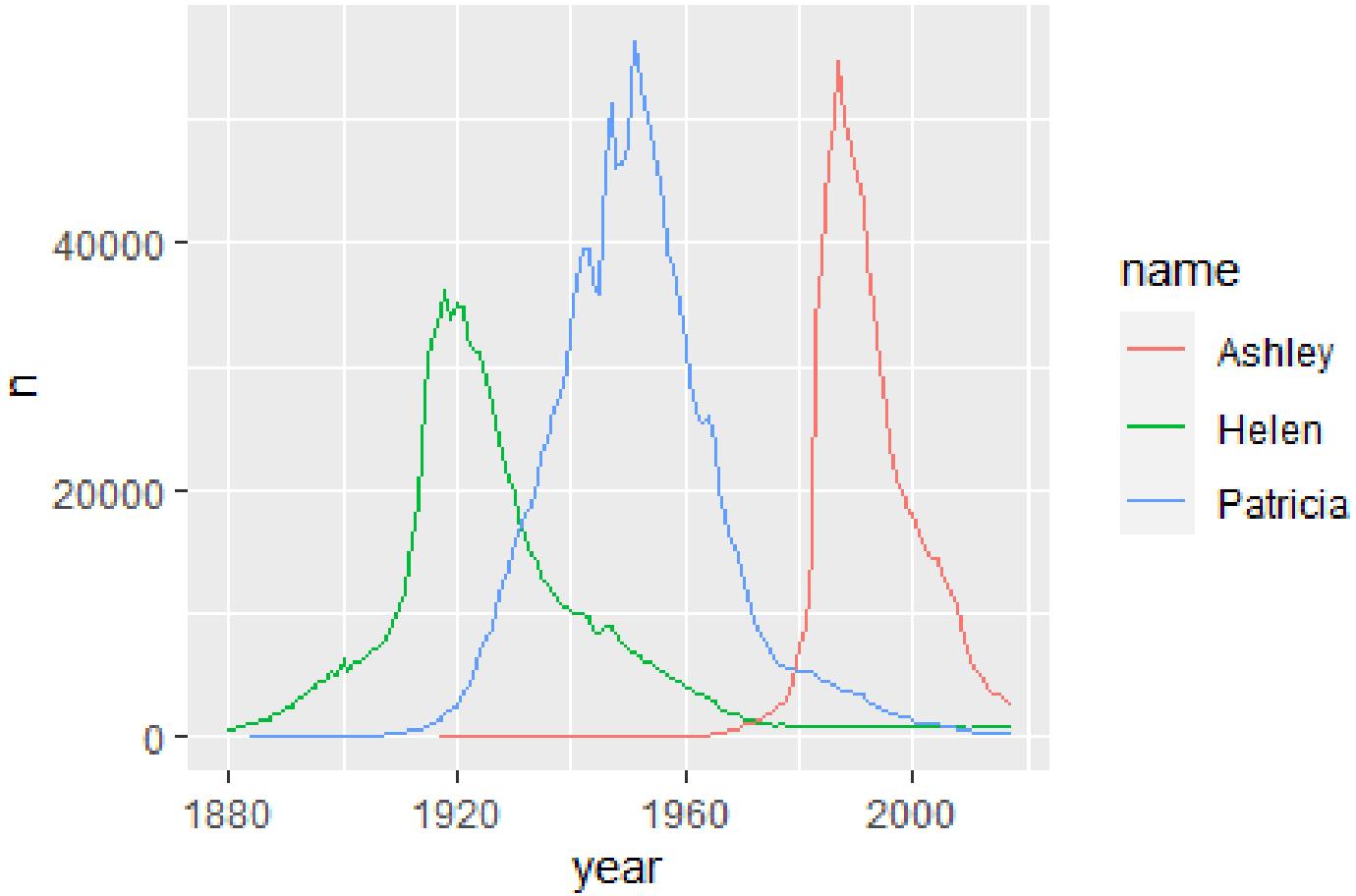


VISUALIZING DATA

```
# Multigroup line chart # Libraries
library(ggplot2)
library(babynames) # provide the dataset: a dataframe called babynames
library(dplyr)
# Keep only 3 names
str(babynames)
don = babynames %>%
  filter(name %in% c("Ashley", "Patricia", "Helen")) %>%
  filter(sex=="F")
```

VISUALIZING DATA

```
ggplot(don, aes(x=year, y=n,
group=name, color=name)) +

  geom_line()
```



VISUALIZING DATA

```
nmonths = 24
x = seq(1,24,1)
var1 = data.frame(
  x,
  Percent.Change = 25 + runif(nmonths,1,100)
)
var2 = data.frame(
  x,
  Percent.Change = 75 + runif(nmonths,1,50)
)
```

VISUALIZING DATA

```
cols = c("unit", "Stockmarket value")
colnames(var1) = cols
colnames(var2) = cols
var1
var2
```

DO NOT give space whem assigning column names in a data frame

VISUALIZING DATA

```
p = ggplot() + geom_line(data =
var1, aes(x = x, y =
Stockmarket.value), color =
"blue")+xlab('Units')
+ylab('percent.change')
```

```
p1 = p+geom_line(data = var2,
aes(x = x, y = Stockmarket.value),
color = "red") +

  xlab('Units') +

  ylab('percent.change')

p

p1
```