

Class 5:Data Viz with ggplot

Nancy Leon-Rivera

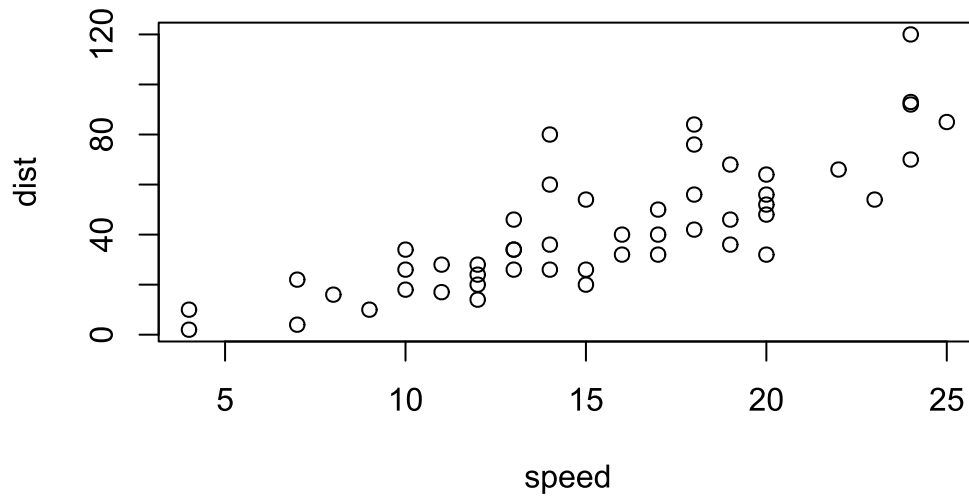
#Graphs and plots in R R has a lot of different graphics systems which include “**base R**”(e.g. the `plot()` function) and add on packages like **ggplot2**. Let’s start with plotting a simple data set in “base R and then ggplot2 to see how they differ.

```
#head() allows to print out first 6 lines and not all lines of data  
head(cars)
```

	speed	dist
1	4	2
2	4	10
3	7	4
4	7	22
5	8	16
6	9	10

To plot this in base R I just use `plot()`

```
plot(cars)
```

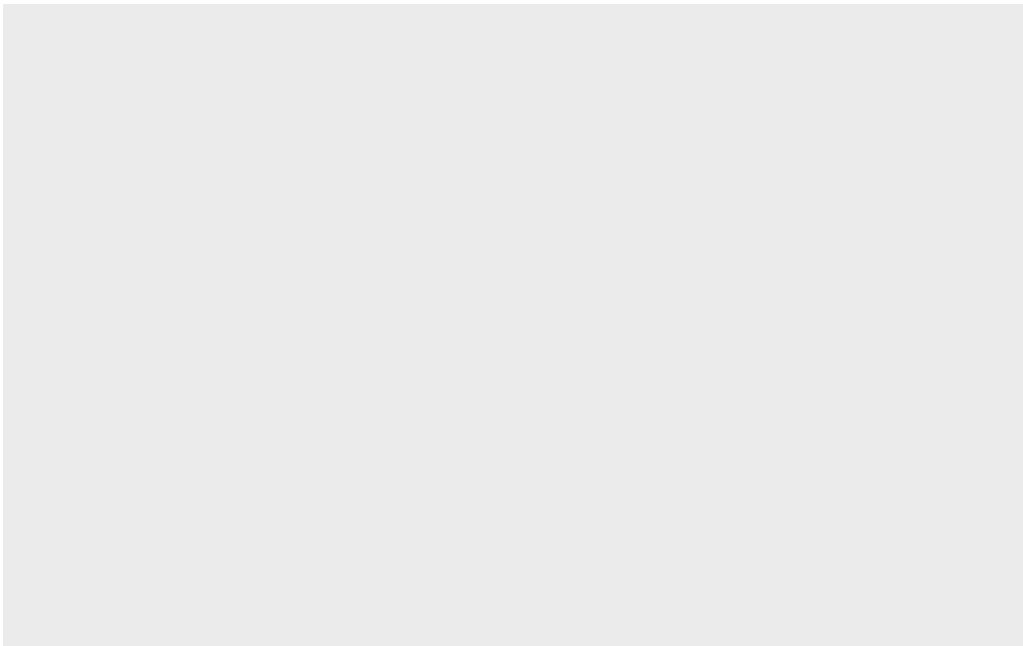


First to use ggplot2, I need to install the package. For his I use the `install.packages()` function.

I don't want to run `install.packages()` in my quarto dic as this would re-install the package every time I render the document. Before I can use this function I need to load the package with a `library` call.

The main function in the ggplot2 package is `ggplot()`.

```
#ggplot(cars) won't work becuae ggplot is not loaded in the library
library(ggplot2)
ggplot(cars)
```



There are at least 3 things that every ggplot needs:

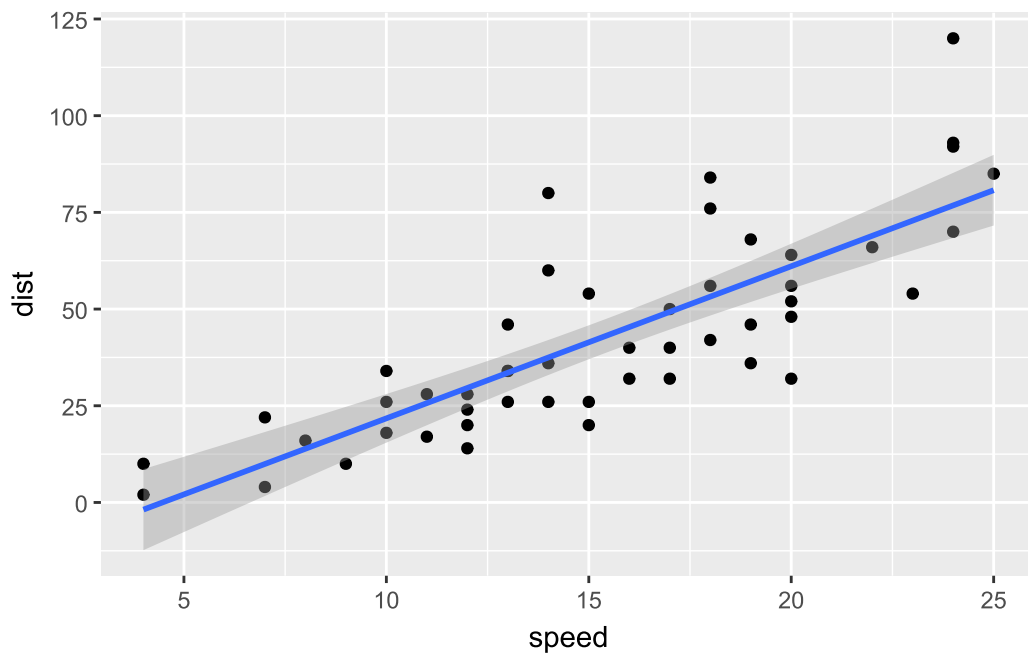
- the **data**(the actual data set that I want to plot)
- the **aesthetics** (how the data maps to my plot)
- the **geoms** or geometries (the type of plot)

I can add more layers to build up more complicated plots:

```
p <- ggplot(cars) +  
  aes(x=speed, y=dist) +  
  geom_point() +  
  geom_smooth(method = "lm")
```

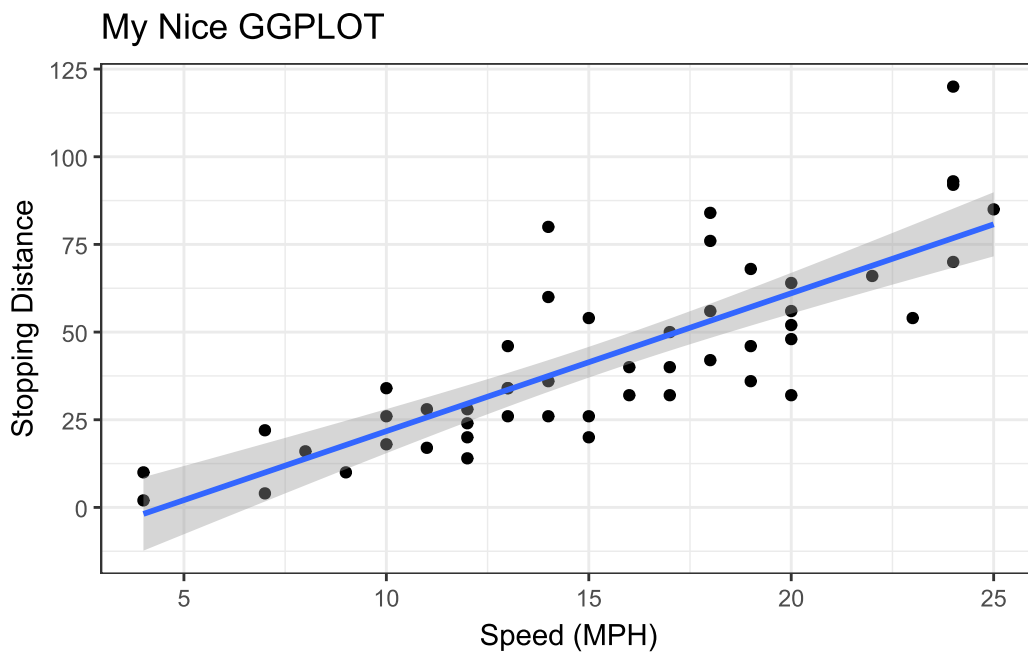
p

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



```
p + labs(title= "My Nice GGPlot", x= "Speed (MPH)", y = "Stopping Distance") +  
  theme_bw()
```

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



```
url <- "https://bioboot.github.io/bimm143_S20/class-material/up_down_
expression.txt"
genes <- read.delim(url)
head(genes)
```

	Gene	Condition1	Condition2	State
1	A4GNT	-3.6808610	-3.4401355	unchanging
2	AAAS	4.5479580	4.3864126	unchanging
3	AASDH	3.7190695	3.4787276	unchanging
4	AATF	5.0784720	5.0151916	unchanging
5	AATK	0.4711421	0.5598642	unchanging
6	AB015752.4	-3.6808610	-3.5921390	unchanging

```
nrow(genes)
```

```
[1] 5196
```

```
colnames(genes)
```

```
[1] "Gene"      "Condition1" "Condition2" "State"
```

```
# wrong table("State")
sum(genes$State=="up")
```

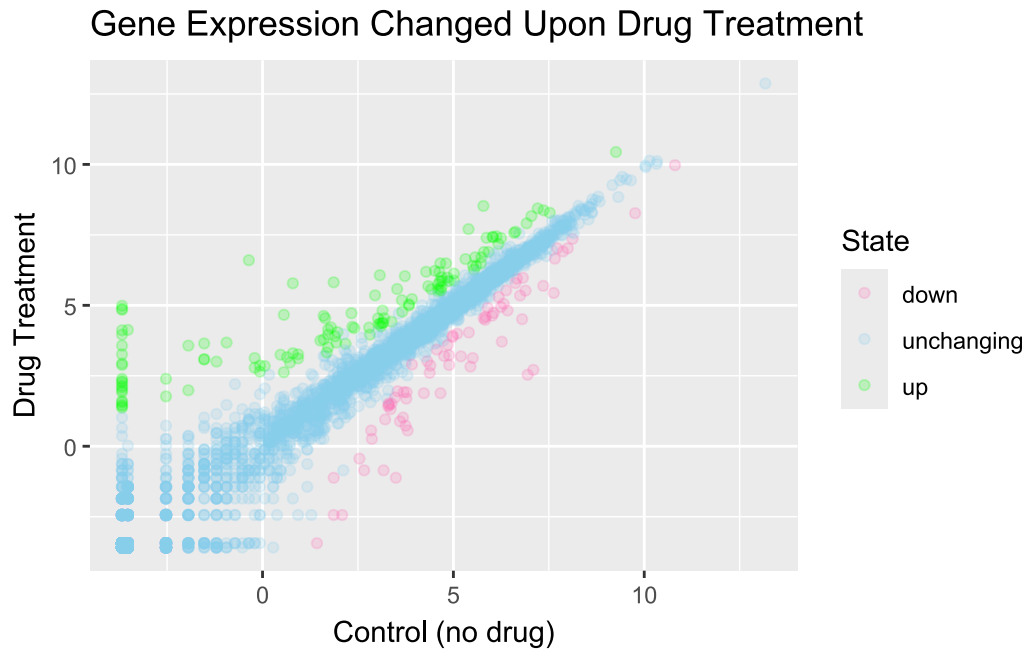
```
[1] 127
```

```
table(genes$State)/nrow(genes) *100
```

down	unchanging	up
1.385681	96.170131	2.444188

```
#watch out for spaces, spelling, and Caps
b <- ggplot(genes) +
  aes(x= Condition1, y=Condition2, col=State) +
  #alpha=0.5 makes point transparent
  geom_point(alpha=0.2)
```

```
b + scale_color_manual(values = c("Hot Pink", "Sky Blue", "Green"))+
  labs(title = "Gene Expression Changed Upon Drug Treatment", x= "Control (no
drug)", y="Drug Treatment")
```



##Gapminder dataset plots We can get exposure to setting more aes() parameters with data sets that include more columns

```
# File location online
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/
extdata/gapminder.tsv"

gapminder <- read.delim(url)
```

Have a small peak at the first 6 rows

```
head(gapminder)
```

	country	continent	year	lifeExp	pop	gdpPercap
1	Afghanistan	Asia	1952	28.801	8425333	779.4453
2	Afghanistan	Asia	1957	30.332	9240934	820.8530
3	Afghanistan	Asia	1962	31.997	10267083	853.1007
4	Afghanistan	Asia	1967	34.020	11537966	836.1971
5	Afghanistan	Asia	1972	36.088	13079460	739.9811
6	Afghanistan	Asia	1977	38.438	14880372	786.1134

```
#number of rows  
nrow(gapminder)
```

```
[1] 1704
```

```
#number of unique countries  
sum(gapminder$country)
```

```
[1] 0
```

```
table(gapminder$country)
```

Afghanistan	Albania	Algeria
12	12	12
Angola	Argentina	Australia
12	12	12
Austria	Bahrain	Bangladesh
12	12	12
Belgium	Benin	Bolivia
12	12	12
Bosnia and Herzegovina	Botswana	Brazil
12	12	12
Bulgaria	Burkina Faso	Burundi
12	12	12
Cambodia	Cameroon	Canada
12	12	12
Central African Republic	Chad	Chile
12	12	12
China	Colombia	Comoros
12	12	12
Congo, Dem. Rep.	Congo, Rep.	Costa Rica
12	12	12
Cote d'Ivoire	Croatia	Cuba
12	12	12
Czech Republic	Denmark	Djibouti
12	12	12
Dominican Republic	Ecuador	Egypt
12	12	12
El Salvador	Equatorial Guinea	Eritrea
12	12	12
Ethiopia	Finland	France
12	12	12
Gabon	Gambia	Germany

12	12	12
Ghana	Greece	Guatemala
12	12	12
Guinea	Guinea-Bissau	Haiti
12	12	12
Honduras	Hong Kong, China	Hungary
12	12	12
Iceland	India	Indonesia
12	12	12
Iran	Iraq	Ireland
12	12	12
Israel	Italy	Jamaica
12	12	12
Japan	Jordan	Kenya
12	12	12
Korea, Dem. Rep.	Korea, Rep.	Kuwait
12	12	12
Lebanon	Lesotho	Liberia
12	12	12
Libya	Madagascar	Malawi
12	12	12
Malaysia	Mali	Mauritania
12	12	12
Mauritius	Mexico	Mongolia
12	12	12
Montenegro	Morocco	Mozambique
12	12	12
Myanmar	Namibia	Nepal
12	12	12
Netherlands	New Zealand	Nicaragua
12	12	12
Niger	Nigeria	Norway
12	12	12
Oman	Pakistan	Panama
12	12	12
Paraguay	Peru	Philippines
12	12	12
Poland	Portugal	Puerto Rico
12	12	12
Reunion	Romania	Rwanda
12	12	12
Sao Tome and Principe	Saudi Arabia	Senegal
12	12	12
Serbia	Sierra Leone	Singapore
12	12	12
Slovak Republic	Slovenia	Somalia
12	12	12
South Africa	Spain	Sri Lanka

12	12	12
Sudan	Swaziland	Sweden
12	12	12
Switzerland	Syria	Taiwan
12	12	12
Tanzania	Thailand	Togo
12	12	12
Trinidad and Tobago	Tunisia	Turkey
12	12	12
Uganda	United Kingdom	United States
12	12	12
Uruguay	Venezuela	Vietnam
12	12	12
West Bank and Gaza	Yemen, Rep.	Zambia
12	12	12
Zimbabwe		
12		

```
#can't do sum since the data set is chars not numbers
length(unique(gapminder$country))
```

```
[1] 142
```

What years does the data set cover

```
table(gapminder$year)
```

```
1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 2002 2007
142  142  142  142  142  142  142  142  142  142  142  142
```

```
#or
unique(gapminder$year)
```

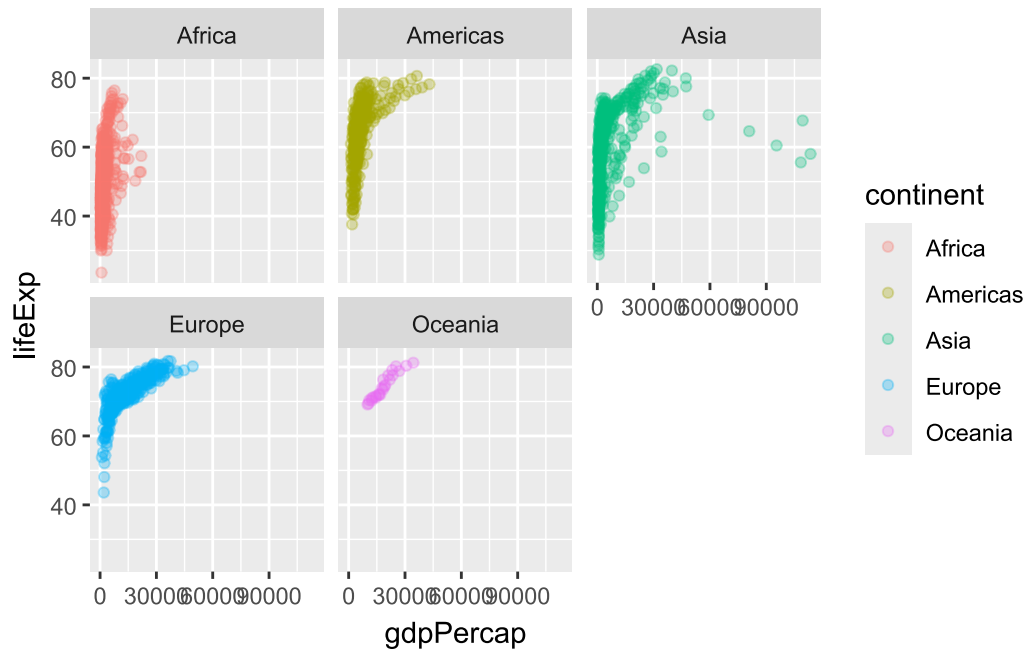
```
[1] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 2002 2007
```

Key functions that will be useful in our journey include:

-nrow() -ncol() -length() -unique() -table()

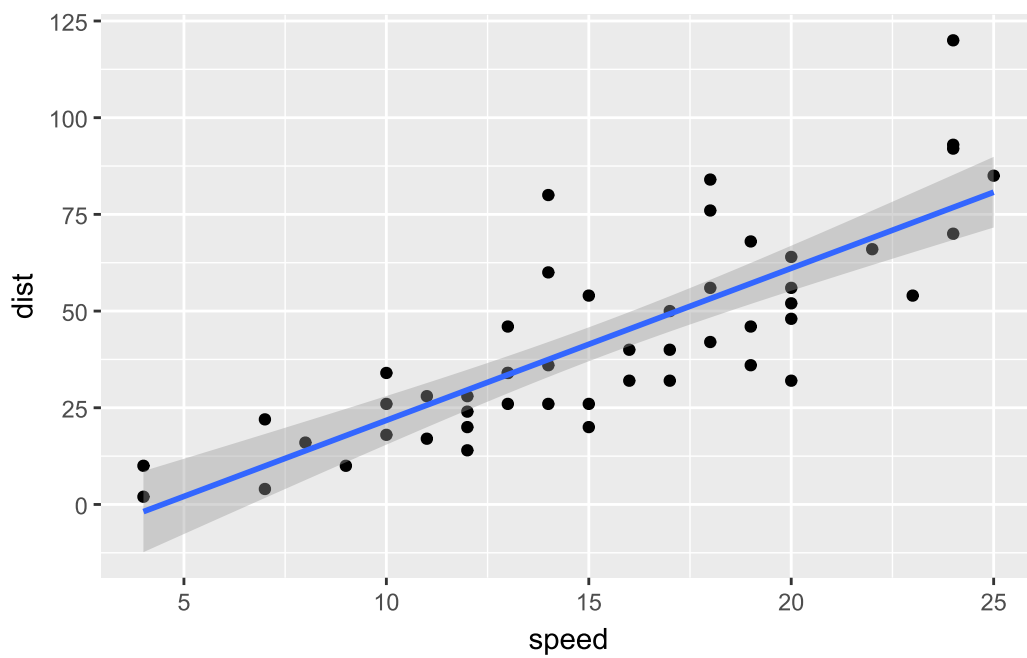
```
ggplot(gapminder)+
  aes(x=gdpPercap, y=lifeExp, col= continent) +
```

```
geom_point(alpha=0.3) +  
facet_wrap(~continent)
```



p

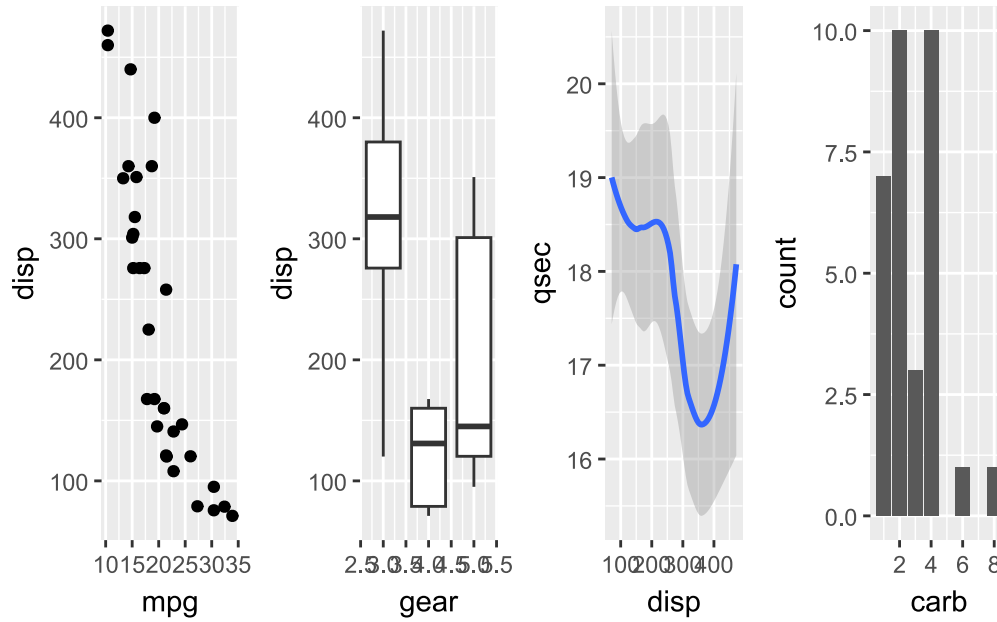
```
`geom_smooth()` using formula = 'y ~ x'
```



```
library(patchwork)
p1 <- ggplot(mtcars) + geom_point(aes(mpg, disp))
p2 <- ggplot(mtcars) + geom_boxplot(aes(gear, disp, group = gear))
p3 <- ggplot(mtcars) + geom_smooth(aes(displacement, qsec))
p4 <- ggplot(mtcars) + geom_bar(aes(carb))

(p1 | p2 | p3 | p4)
```

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



Animation!!

```
library(gapminder)
```

Attaching package: 'gapminder'

The following object is masked _by_ '.GlobalEnv':

gapminder

```
library(gganimate)
```