class05.R

Jocelyn Olvera

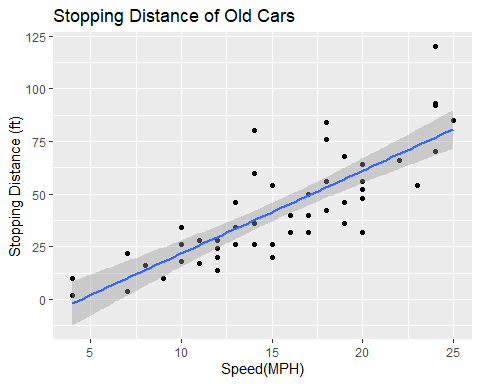
2021-10-13

# Class 05: Data Visualization  
# Author: Jocelyn Olvera  
# Date: October 13 2021  
# Description: Today we are going to use ggplot2 package  
  
# First we need to loud the package!  
# Install.packages("ggplot2")  
library(ggplot2)  
  
# We will use this inbuilt "cars" dataset first  
# "head" allows viewing of only a small subset of the data   
head(cars)

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

# All ggplots have at least 3 layers  
# data + aes + geoms  
  
ggplot(data=cars) +  
 aes(x=speed, y=dist) +   
 geom\_point()+   
# Side-Note: geom\_line()+ draws lines connecting data points   
 geom\_smooth(method="lm")+  
 labs(title="Stopping Distance of Old Cars",  
 x="Speed(MPH)",  
 y="Stopping Distance (ft)")

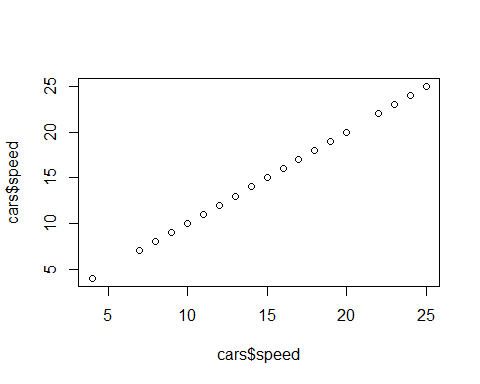
## `geom\_smooth()` using formula 'y ~ x'



# Side-Note: ggplot is not the only graphics system  
# A very popular is a good old "base" R graphics  
plot(cars)



plot(cars$speed, cars$speed)



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

# Q1.how many genes are in this dataset?  
nrow(genes)

## [1] 5196

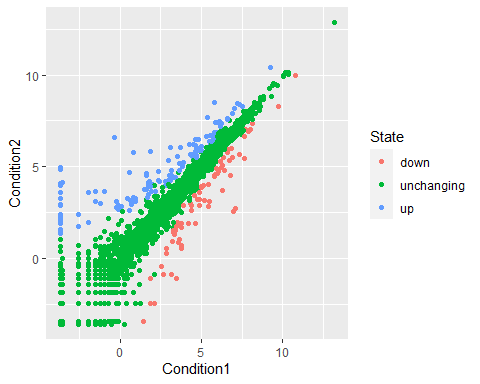
# Q2. how many genes are "up"?  
# this function list all genes: genes$State  
table(genes$State)

##   
## down unchanging up   
## 72 4997 127

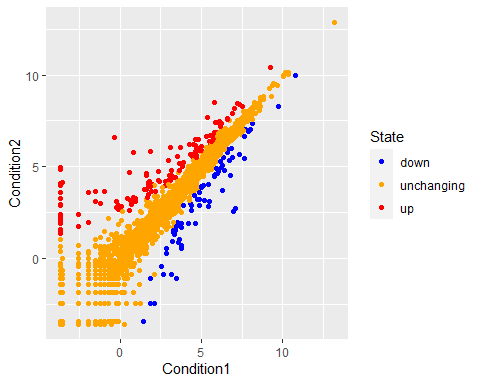
# Q3. What % are up?  
# this updates as data updates, adding sig-figs is the #3  
# round=rounding up percentage  
round(table(genes$State)/nrow(genes)\*100,2)

##   
## down unchanging up   
## 1.39 96.17 2.44

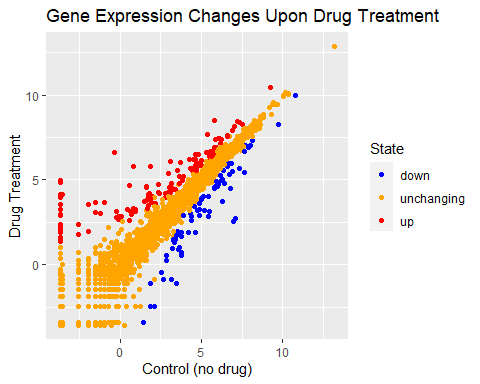
# Lets make a figure  
# Sates are the down/up/unchanging gene expressions  
p <- ggplot(genes) +  
 aes(x=Condition1, y=Condition2, col=State) +  
 geom\_point()  
  
p



# I like it but not the default colors, lets change them  
# reminder to re-assign p in orde to keep the color change  
p <- p + scale\_colour\_manual(values=c("blue", "orange", "red"))  
p



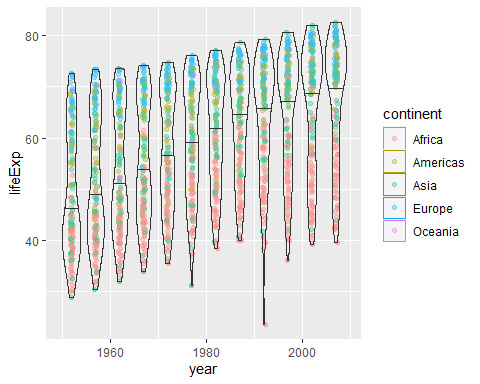
# Changing the main title and titles of the X and Y axis  
p + labs(title="Gene Expression Changes Upon Drug Treatment", x="Control (no drug)", y="Drug Treatment")



# Lets explore the gapminder dataset  
# install.packages("gapminder")  
library(gapminder)  
head(gapminder)

## # A tibble: 6 x 6  
## country continent year lifeExp pop gdpPercap  
## <fct> <fct> <int> <dbl> <int> <dbl>  
## 1 Afghanistan Asia 1952 28.8 8425333 779.  
## 2 Afghanistan Asia 1957 30.3 9240934 821.  
## 3 Afghanistan Asia 1962 32.0 10267083 853.  
## 4 Afghanistan Asia 1967 34.0 11537966 836.  
## 5 Afghanistan Asia 1972 36.1 13079460 740.  
## 6 Afghanistan Asia 1977 38.4 14880372 786.

# Lets make a new plot of year vs lifeExp  
# running first 2 lines just displays plot layout  
# alpha is transparency  
# reminder "aes" aesthetics   
ggplot(gapminder) +   
 aes(x=year, y=lifeExp, col=continent) +  
 geom\_jitter(width=0.3, alpha=0.4) +  
# geom\_boxplot(aes(group=year), alpha=0.4)  
# mean line = quantiles  
 geom\_violin(aes(group=year),alpha=0.2,   
 draw\_quantiles = 0.5)



# Install the plotly  
# install.packages("plotly")  
# plotly is an interactive plot function (AMAZING!)  
library(plotly)

##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:stats':  
##   
## filter

## The following object is masked from 'package:graphics':  
##   
## layout

# adding (#) to ggplotly to be able to save as PDF  
# ggplotly()