R Notebook

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

plot(cars)

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

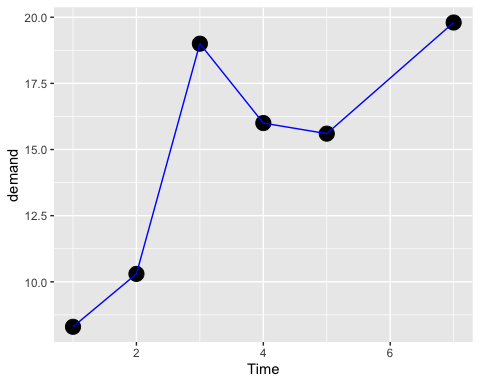
The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

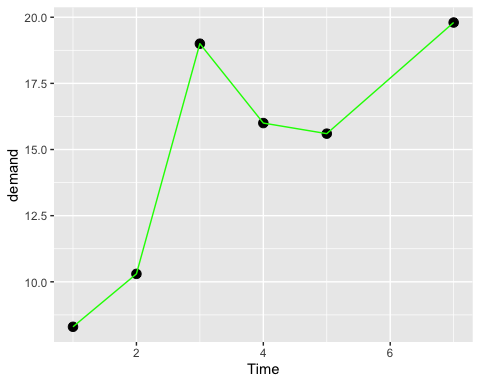
# loads the tidyverse library  
library(ggplot2)   
# loads the ggplot2 package frim tidyverse library  
data()   
# lists all the data sets built in R in new tab   
?BOD   
# data dictionary of BOD data set like description  
#this gives info about BOD in the bottam right section

# plotting the graph  
ggplot(data = BOD,  
 # choosing BOD as data  
 mapping = aes(x = Time,   
 y = demand))+   
 #setting x and y axes with time and demand respectively  
 geom\_point(size = 5) +   
 # adding points with size of 5  
 geom\_line(color = "blue")



# adds color line to the points  
# As time passes demand increases with slight decrease

# plotting the graph  
# in the below command it didn't specify the data and mapping beacuse ggplot   
# it automatically takes first two arugments as data and mapping  
  
ggplot(BOD, aes(Time, demand))+   
 # Assigning directly the data set, x and y axes.  
 geom\_point(size = 3)+   
 # adding points with size of 3  
 geom\_line(colour = "green")



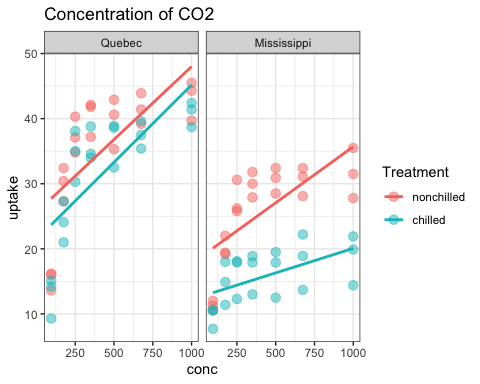
# adds color line to the points

head(CO2)

## Plant Type Treatment conc uptake  
## 1 Qn1 Quebec nonchilled 95 16.0  
## 2 Qn1 Quebec nonchilled 175 30.4  
## 3 Qn1 Quebec nonchilled 250 34.8  
## 4 Qn1 Quebec nonchilled 350 37.2  
## 5 Qn1 Quebec nonchilled 500 35.3  
## 6 Qn1 Quebec nonchilled 675 39.2

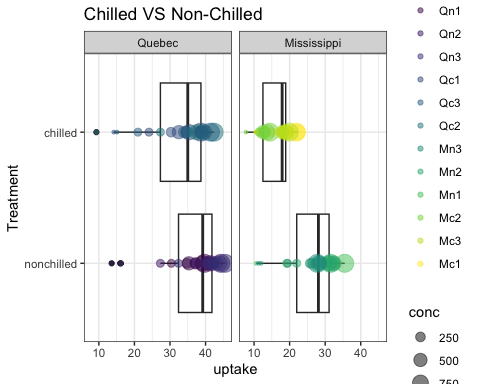
# displays first few rows from the data set  
CO2 %>%   
 #As we are using tidyverse package we can use pipe operator  
 # creates ggplot with co2 data set with pipe operator   
 # Create a scatter plot using ggplot  
 ggplot(aes(conc,uptake, # ggplot() X-axis as conc and Y-axis as uptake   
 colour = Treatment))+   
 # color to Treatment  
 # Add points to the scatter plot with specific aesthetics  
 geom\_point(size = 3, alpha = 0.5)+   
 # with the size 3 and transparency 0.5 to the points in plot  
 geom\_smooth(method = lm, se = F)+   
 # Add a trend line using linear regression (lm), excluding the confidence interval (se = FALSE)  
 facet\_wrap(~Type) +   
 # different panels are created for every plant depending on 'Type'  
 labs(title = "Concentration of CO2")+ # title to graph  
 theme\_bw()

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



# setting the black and white theme  
# It shows that as concentration of CO2 increases, CO2 uptake usually increases.  
# Additionally, nonchilled plants exhibit higher uptake rates compared to chilled plants.  
# This is evident from the partitioning of trend lines within each panel.

CO2 %>% # creates ggplot with co2 data set with pipe operator  
 ggplot(aes(Treatment, uptake)) +   
 # ggplot() with aes() mapping takes the Treatment and uptake  
 geom\_boxplot()+   
 # adds boxplot to the plot  
 geom\_point(alpha = 0.5, aes(size = conc,   
 #adding points with transparency and conc to size   
 colour = Plant))+ # color as per the plants  
 facet\_wrap(~Type)+   
 # different panels according to Type  
 # Different plant types are visualized in separate panels.  
 coord\_flip()+  
 # flips the plot  
 # Flipping the coordinates makes it easier to compare uptake levels across treatments.  
 theme\_bw()+   
 # black and white theme  
 labs(title = "Chilled VS Non-Chilled")



# adds the label

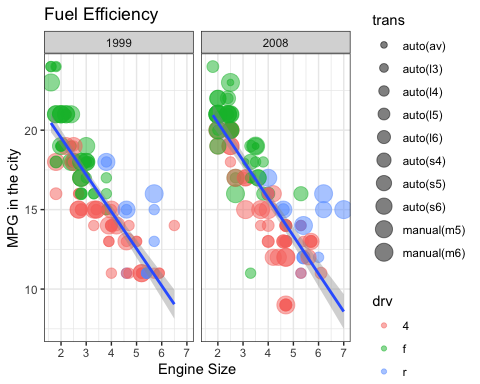
head(mpg)

## # A tibble: 6 × 11  
## manufacturer model displ year cyl trans drv cty hwy fl class   
## <chr> <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>   
## 1 audi a4 1.8 1999 4 auto(l5) f 18 29 p compa…  
## 2 audi a4 1.8 1999 4 manual(m5) f 21 29 p compa…  
## 3 audi a4 2 2008 4 manual(m6) f 20 31 p compa…  
## 4 audi a4 2 2008 4 auto(av) f 21 30 p compa…  
## 5 audi a4 2.8 1999 6 auto(l5) f 16 26 p compa…  
## 6 audi a4 2.8 1999 6 manual(m5) f 18 26 p compa…

# prints the first few rows  
  
# ggplot visualization using the mpg dataset  
mpg %>%   
 # uses mpg data set with pipe operator  
 filter(cty<25) %>%   
 # Filters dataset to include only cars with city mpg (cty) less than 25  
 ggplot(aes(displ, cty))+   
 # ggplot() with aes() mapping takes x and y axes  
 geom\_point(aes(colour = drv,   
 # adds the points with color 'drv'  
 size = trans),  
 alpha = 0.5)+   
 geom\_smooth(method = lm)+  
 # linear model method for trend  
 facet\_wrap(~year, nrow = 1)+   
 # Create separate panels for each 'year' of the cars, arranged in a single row  
 labs(x= "Engine Size", # x axis label  
 y = "MPG in the city", # y axis label  
 title = "Fuel Efficiency")+ # title for chart  
 theme\_bw()

## Warning: Using size for a discrete variable is not advised.

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



# black and white theme to chart