# IntroToR

UCLA Collaboratory

10/21/2021

## Contents

## R Markdown and Code Chunks

### Overview of R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

In this lesson we will learn to install and load the packages necessary to do data analysis in R.

## Importing Data and Selecting Columns

We will learn to import data into R and reshape it into formats that are useful for data visualization. For our research question, we will work with public COVD19 data to visualize the number of COVID19 cases per million for every state in the form of a heatmap and imagine other ways we would want to visualize this kind of information.

```r
# This is just repeating the commands that we did in the first day(s) to import our data.

#Import the Covid Data Set
CountyVaxDataCA <- read.csv("cdph-vaccination-county-totals.csv", header = TRUE, sep = ",")

#Import the States data set (including DC and PR)
```

```r
CountyPop <- read.table("CountyData.csv", header = TRUE, sep = ",")
CountyPopFilter <- select(CountyPop, County, Per10K)


#Import MMR Vaccine Rate Data set from Tidy Tuesday
MMRVaccineRate <- read.csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/20
```

## Data Wrangling and Cleaning

We will learn to handle imperfect data sets, calculate new columns, clean and rename data for downstream uses.

```r
# Merging the tables
VaxDataMerged <- left_join(CountyPopFilter, CountyVaxDataCA, by = c("County"="county"))

# Calculating new columns and filtering data
NormalizedVaxData <- VaxDataMerged %>%
  mutate(VaxDensity = fully_vaccinated/Per10K) %>%
  filter(VaxDensity != "NA")

# Selecting and renaming columns
VaccinationRate <- NormalizedVaxData %>%
  select(date,
         County,
         VaxDensity,
         fully_vaccinated,
         Per10K,
         at_least_one_dose,
         new_doses_administered,
         new_pfizer_doses,
         new_moderna_doses,
         new_jj_doses)

# Summarizing groups
LastMonthSummarized <- CountyVaxDataCA %>%
  filter(date >  ymd("2021-05-23")) %>%
  group_by(county) %>%
  summarize(MeanNewDoses = mean(new_doses_administered),
            NewPfizerDoses = mean(new_pfizer_doses),
            NewModernaDoses = mean(new_moderna_doses),
            NewJJDoses = mean(new_jj_doses))

# Here we filter states that have populations over 10 million people
VaccinationRate <- VaccinationRate %>%
  filter(Per10K >= 100)

kable(LastMonthSummarized[1:10,], caption = "A data table with 10 rows")
```

Table 1: A data table with 10 rows

| county | MeanNewDoses | NewPfizerDoses | NewModernaDoses | NewJJDoses |
|---|---:|---:|---:|---:|
| Alameda | 3607.2013889 | 2814.0625000 | 672.36111 | 120.777778 |
| Alpine | 0.7222222 | 0.1597222 | 0.56250 | 0.000000 |
| Amador | 54.9444444 | 30.5277778 | 19.43750 | 4.979167 |
| Butte | 370.3750000 | 254.9375000 | 97.40278 | 18.034722 |
| Calaveras | 72.8402778 | 43.7430556 | 24.56944 | 4.527778 |
| Colusa | 46.6805556 | 24.0416667 | 21.50000 | 1.138889 |
| Contra Costa | 2678.5833333 | 2212.0000000 | 374.27778 | 92.305556 |
| Del Norte | 48.5486111 | 25.8888889 | 18.74306 | 3.916667 |
| El Dorado | 352.0763889 | 235.0277778 | 98.32639 | 18.722222 |
| Fresno | 2183.5972222 | 1606.5277778 | 511.50000 | 65.569444 |

## Visualizing trends using ggplot

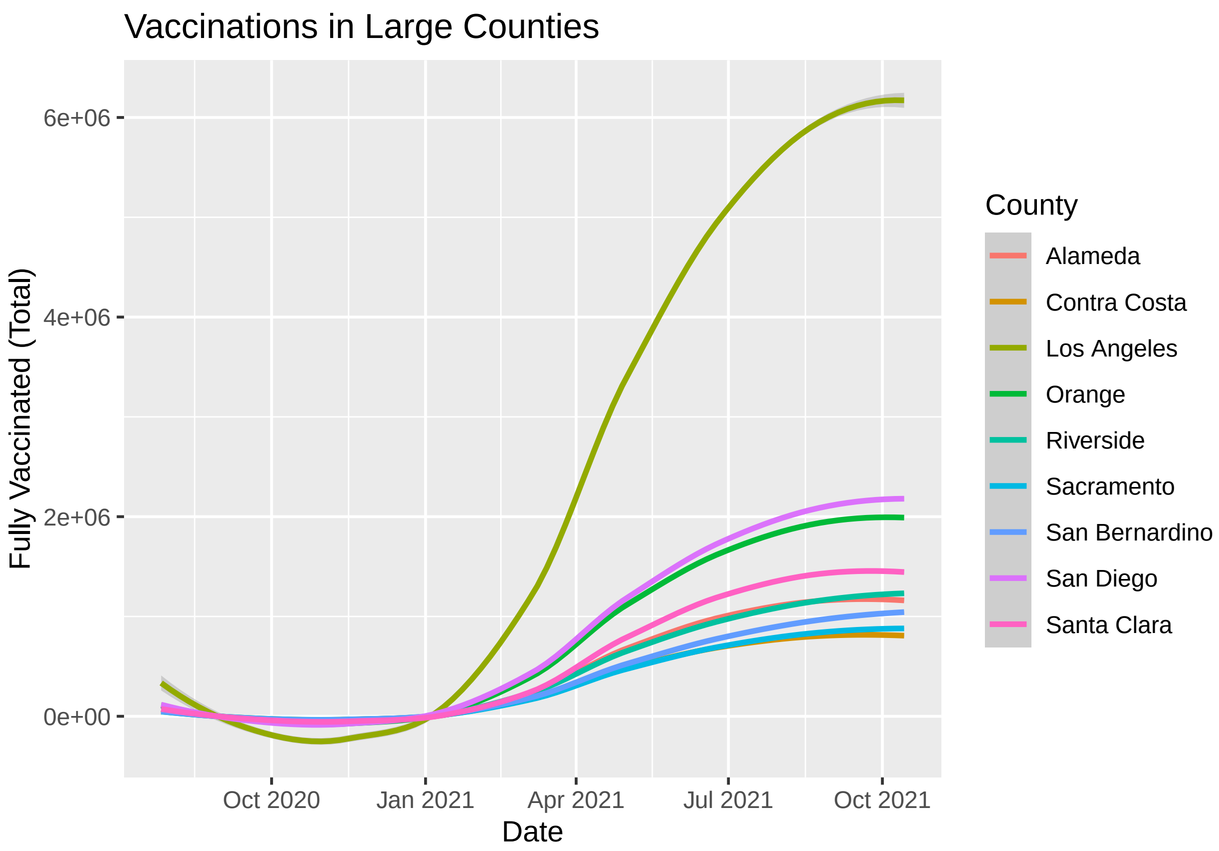
In this lesson, we use ggplot to create trendlines for data that we work with.

```r
# Here we write the ggplot code. We use the ymd() funciton on our x axis
# to make it easier to work with the dates in our table. We use geom_smooth
# to create a trend line and we set some axis labels.

VaxPlot <- ggplot(VaccinationRate, aes(x = ymd(date), y = fully_vaccinated, color = County)) +
  geom_smooth(method = "auto") +
  labs(x = "Date",
       y = "Fully Vaccinated (Total)",
       title = "Vaccinations in Large Counties")

VaxPlot
```

```
## ‘geom_smooth()‘ using method = ’loess’ and formula ’y ~ x’
```

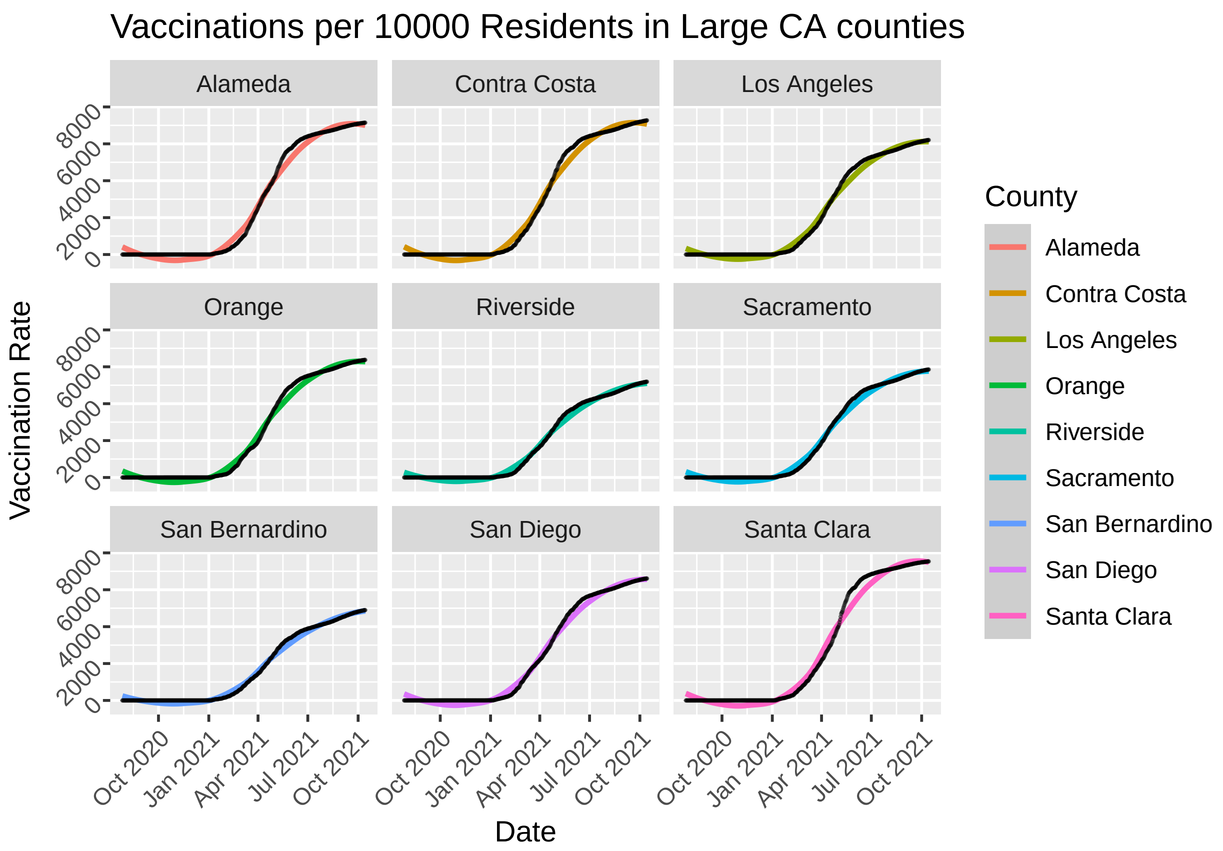


```
# We can easily separate plots so that we can analyze and compare
# data in different ways using facet_wrap()

VaxDensityPlotFacets <- ggplot(VaccinationRate, aes(ymd(date), VaxDensity, color = County)) +
  geom_smooth(method = "auto") +
  geom_point(color = "black", size = 0.1, alpha = 0.5)+
  facet_wrap(facets = "County") +
  theme(axis.text = element_text(angle = 45, hjust = 1)) +
  labs(title = "Vaccinations per 10000 Residents in Large CA counties",
       x = "Date",
       y = "Vaccination Rate")

VaxDensityPlotFacets
```

```
## ‘geom_smooth()‘ using method = ’loess’ and formula ’y ~ x’
```



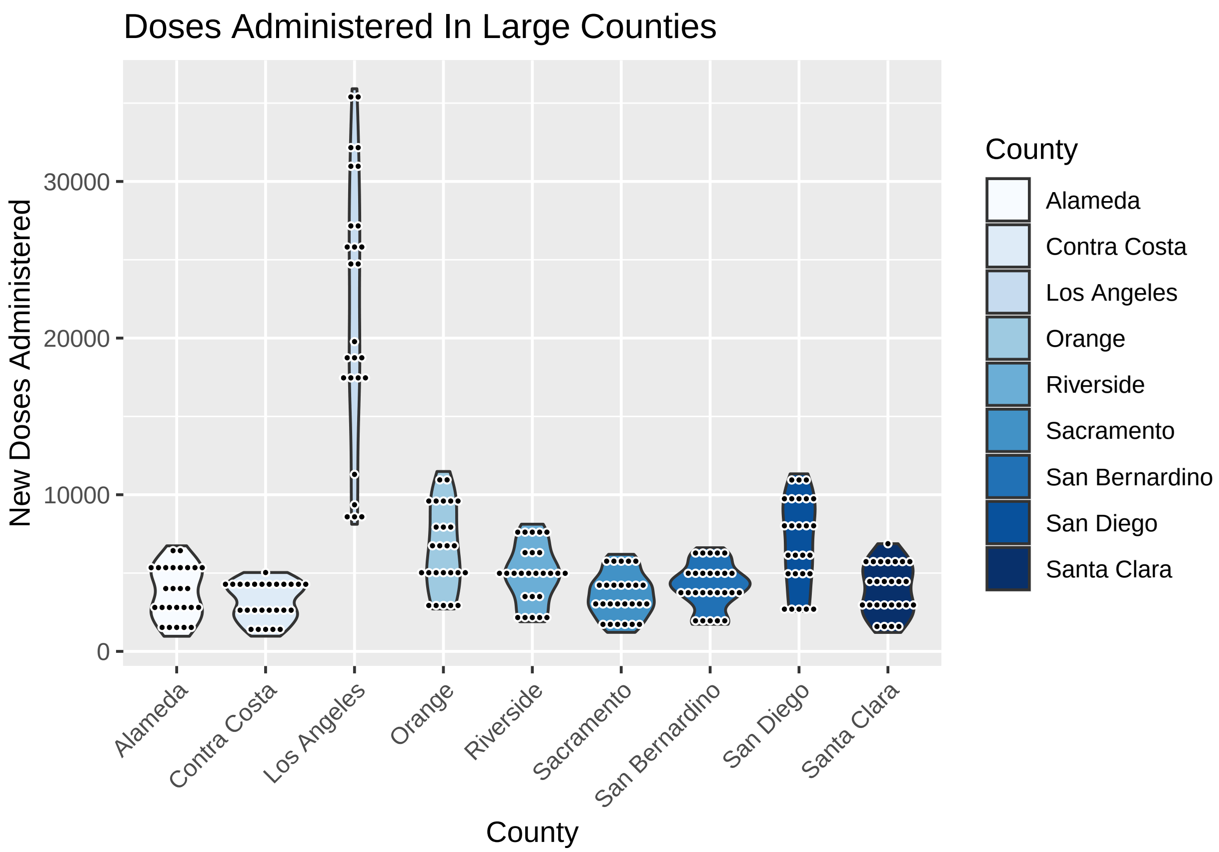
### Visualizing distribution with ggplot2

We created a violin plots to look at data distribution.

```
# Selecting the data
LastMonthVaccinations <- VaccinationRate %>%
  filter(ymd(date) >= ymd("2021-09-19")) %>%
  mutate(PfizerDensity = new_pfizer_doses/Per10K,
         ModernaDensity = new_moderna_doses/Per10K,
         JJDensity = new_jj_doses/Per10K)

# The Plot includes mean and stadnard deviation bars
CpmViolin <- ggplot(LastMonthVaccinations, aes(County, new_doses_administered, fill = County)) +
  geom_violin(scale = "area")+
  geom_dotplot(binaxis = "y", dotsize = 0.4, stackdir = "center", fill = "black", color = "white") +
  scale_fill_brewer(palette = "Blues")+
  labs(title = "Doses Administered In Large Counties",
       y = "New Doses Administered")+
  theme(axis.text.x=element_text(angle=45, hjust = 1))

CpmViolin
```



## T tests

Finally we learned to conduct some basic statistics on data that we want to compare.

```
# First we filter tables based on the data that we want to compare. In this # case we are
# comparing the number of cases per million in the last month between Los Angeles
# and San Diego.

LAvax <- LastMonthVaccinations %>%
  filter(County == "Los Angeles")

SDvax <- LastMonthVaccinations %>%
  filter(County =="San Diego")

# We can then use the t.test() function on specific columns to calculate
# our t-test comparing the two data sets.

t.test(LAvax$VaxDensity, SDvax$VaxDensity)
```

```
##
##  Welch Two Sample t-test
##
## data:  LAvax$VaxDensity and SDvax$VaxDensity
## t = -28.568, df = 50, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -434.5539 -377.4618
## sample estimates:
## mean of x mean of y
##  6131.963  6537.971
```

## Challenge

For your breakout room sessions. Take the code chunk below and create a plot that is different in some way compared to the plots shown above. Feel free to use any of the data we have imported over the course of the session. Try using a different Geom in ggplot, filtering the data differently, or changing some of the aes parameters.

## Instructions for optional homework assignment

Create a new code chunk below this text and name it “QuizChunk”. Create a new plot within the code chunk. You can copy and paste code from above, and modify it. Or you can write your own code or analysis using a different geometry or data set from the tidy tuesday space. Feel free to dig into the data set a bit further and be creative with the visualization you make. When you are finished, email me a copy of your R markdown file (the one that ends in .Rmd) by Thursday October 28th and I’ll take a look and give you feedback if you request. Please keep your code to less than 50 lines, or I won’t be able to evaluate it due to my own time limitations.